



FACULTAD DE INGENIERÍA
DEPARTAMENTO DE CIENCIAS DE DATOS E
INFORMÁTICA

COGNIGAME: Videojuegos Interactivos para la Evaluación y Rehabilitación
Cognitiva en Adultos Mayores con ACV

Proyecto de Título II para optar al Título de Ingeniero Informático con mención en
Ciencias de la Información.

Camilo Jaime Briones Quezada
Profesora Guía: Mg. Delia Ibacache B.
Valparaíso, Chile

2024

DEDICATORIA

Partiré, como una vez leí por ahí, agradeciéndome por luchar cada día por mí mismo y nunca dejar de mirar más allá del cielo.

Le dedico esta tesis a mi hija, por ser parte de mi vida, uno de mis pilares fundamentales en mantenerme firme cada día, que con su propia sonrisa hace que quiera superarme cada vez más. A mis hermanos, papá y mamás por siempre brindarme su apoyo, en especial, a Mario Briones, hermano con quien me críe y siempre ha estado cada vez que necesito un abrazo, palabras y apoyo.

Quiero mencionar también a mis amigos: Pedro Roblero, Cristóbal Brito y Hugo Leal, por siempre animarme y aconsejarme en este camino llamado proyecto de título, quienes, siempre están dispuesto a responder mis dudas y en especial a Pedro Roblero por abrirme la puerta de su casa y apoyar día a día mis objetivos.

Finalmente nombrar a mi profesora guía Delia Ibacache, quien me aceptó como su alumno tesista y me ha guiado de buena manera, escuchando atentamente mis dudas y nutriéndome de observaciones claves para el desarrollo del proyecto.

RESUMEN

La investigación en el campo de la salud cognitiva de adultos mayores ha cobrado un interés creciente en los últimos años. Los estudios revisados ofrecen una perspectiva prometedora sobre el papel de los videojuegos en la estimulación cognitiva y la preservación de la función cognitiva en este grupo etario. Es por esto, que se busca innovar en el ámbito terapéutico con enfoque a adultos mayores con problemas cognitivos.

El proyecto consiste en crear una plataforma web de videojuego que permita a los pacientes, del departamento de geriatría del Hospital de Limache, entretenerse como trabajo de rehabilitación, ya que al estimularse de esta manera generen nuevas conexiones nerviosas evitando el deterioro cognitivo, tanto por la edad como por alguna patología. Además, la aplicación generará informes de los pacientes en base a la captura de métricas desde el juego que serán utilizadas por los profesionales del área para un mejor monitoreo y/o evaluación del paciente.

ABSTRACT

Research in the field of cognitive health among older adults has garnered increasing interest in recent years. Reviewed studies offer a promising perspective on the role of video games in cognitive stimulation and the preservation of cognitive function in this age group. Therefore, there is a quest for innovation in the therapeutic realm focusing on older adults with cognitive issues.

The project aims to create a web-based gaming platform allowing patients from the geriatrics department of Hospital de Limache to engage in rehabilitation activities. By stimulating in this manner, patients develop new neural connections, thus avoiding cognitive decline due to age or pathology. Additionally, the application will generate patient reports based on metrics captured during gameplay, which will be used by professionals of field for better patient monitoring and evaluation.

TABLA DE CONTENIDO

RESUMEN.....	iii
ABSTRACT	iv
TABLA DE CONTENIDO.....	v
TABLA DE ILUSTRACIONES.....	viii
INTRODUCCIÓN	1
CAPÍTULO I: FORMULACIÓN DEL PROBLEMA	2
1.1 DESAFÍOS EN EL MONITOREO DE PACIENTES.....	2
1.2 PROPUESTA DE SOLUCIÓN	3
1.3 OBJETIVOS	4
1.3.1 OBJETIVO GENERAL	4
1.3.2 OBJETIVOS ESPECÍFICOS.....	4
CAPÍTULO II: MARCO TEÓRICO.....	5
2.1 CONCEPTO DE VIDEOJUEGO COGNITIVO.....	6
2.2 ENTORNOS TECNOLÓGICOS.....	6
2.2.1 JAVASCRIPT	7
2.2.2 REACT.....	7
2.2.3 MONGO DB	7
2.2.4 PHASER	8
2.2.5 GITHUB.....	8
2.3 METODOLOGÍA DE TRABAJO.....	9
CAPÍTULO III: ANÁLISIS.....	10
3.1 PLATAFORMAS DE ENTRENAMIENTO COGNITIVO.....	10

3.2 REQUERIMIENTOS.....	11
3.2.1 REQUERIMIENTOS FUNCIONALES.....	11
3.2.2 REQUERIMIENTOS NO FUNCIONALES	12
CAPÍTULO IV: DISEÑO DEL SISTEMA	13
4.1 PATRÓN DE DISEÑO.....	13
4.2 PERFILES DE USUARIO.....	13
4.3 VISTAS DE USUARIO.....	14
4.3.1 INICIO DE SESIÓN.....	14
4.3.2 VISTA SUPERADMIN	15
4.3.3 VISTA KINESIOLOGO.....	17
4.3.4 VISTA PACIENTE.....	21
CAPÍTULO V: CONSTRUCCIÓN DEL SISTEMA.....	25
5.1 IMPORTACIÓN Y DECLARACIÓN DE REFERENCIAS	25
5.2 CONFIGURACIÓN DEL JUEGO	27
5.3 PRELOAD: CARGA DE RECURSOS	29
5.4 CREATE: CREACIÓN DE LA ESCENA DEL JUEGO	30
5.5 MANEJO DE EVENTO DEL BOTON.....	32
5.6 DEFINICIÓN DE POSICIONES	33
5.7 CREACIÓN DE LÉMURES	34
5.8 LÓGICA DEL JUEGO	35
5.9 ALMACENAMIENTO DE LOS DATOS	37
5.9.1 ESQUEMA USUARIO.....	38
5.9.2 ESQUEMA STATS	40
CAPÍTULO VI: PRUEBAS.....	42
6.1 TIPOS DE PRUEBAS	42

6.1.1 PRUEBAS UNITARIAS	42
6.1.2 PRUEBAS DE INTEGRACIÓN	42
6.1.3 PRUEBAS FUNCIONALES	42
6.1.4 PRUEBAS DE USABILIDAD	43
6.1.5 PRUEBAS DE RENDIMIENTO.....	43
6.2. PASOS DEL PLAN	43
6.3 RESULTADOS DE LAS PRUEBAS	43
CAPÍTULO VII: CONCLUSIONES.....	45
7.1 CUMPLIMIENTO DEL OBJETIVO GENERAL.....	45
7.2 REFUTACIÓN BASADA EN ESTUDIOS TEÓRICOS.....	46
7.3 BENEFICIOS PARA PACIENTES Y ESPECIALISTAS.....	47
7.4 IMPORTANCIA DEL SISTEMA CREADO.....	47
REFERENCIAS	48
BIBLIOGRAFÍA	48
LINKOGRAFÍA	48

TABLA DE ILUSTRACIONES

Ilustración 1: Vista login page CogniGame.	15
Ilustración 2: Vista Superadmin.	16
Ilustración 3: Formulario de registro.	17
Ilustración 4: Vista kinesiólogo.	18
Ilustración 5: Listado de stats a observar.	18
Ilustración 6: Stats de un determinado juego y paciente.	19
Ilustración 7: Gráfico de línea en base a respuesta en cada aparición de lémur.	20
Ilustración 8: Gráfico de línea en base a respuesta de clic entre lémures.	20
Ilustración 9: Vista paciente.	22
Ilustración 10: Aparición de un lémur.	23
Ilustración 11: Aparición de dos lémures.	24
Ilustración 12: Importación y declaración de referencias.	27
Ilustración 13: Código con la configuración del juego.	29
Ilustración 14: Código con la carga de recursos.	30
Ilustración 15: Código con la creación de escena del juego.	32
Ilustración 16: Código para manejo	33
Ilustración 17: Código de definición de posiciones.	34
Ilustración 18: Código para la creación de lémures.	36
Ilustración 19: Esquema usuario	37
Ilustración 20: Esquema stats.	40

INTRODUCCIÓN

En las últimas décadas, el envejecimiento poblacional ha sido una tendencia global, y Chile no es la excepción. Este fenómeno trae consigo un aumento significativo en la prevalencia de enfermedades cognitivas, como el Alzheimer y otras demencias, así como un incremento en los casos de accidentes cerebrovasculares (ACV). Frente a esta realidad, se hace imperativo desarrollar herramientas que no solo permitan un monitoreo eficaz de los pacientes, sino que también ofrezcan alternativas para su estimulación cognitiva y rehabilitación tras un ACV.

El presente proyecto tiene como objetivo, desarrollar una plataforma web de videojuegos orientada a la estimulación cognitiva de adultos mayores con problemas cognitivos que permita a los especialistas obtener datos para analizar el estado del paciente durante el tratamiento

El sistema web se desarrollará utilizando tecnologías modernas como React para la interfaz de usuario, MongoDB para la gestión de la base de datos y Phaser para la creación de los juegos, todos ellos elegidos por sus ventajas en términos de rendimiento, flexibilidad y escalabilidad.

A lo largo de este documento, se detallará cada una de estas etapas, así como los resultados obtenidos y las conclusiones derivadas del trabajo realizado. Además, se discutirán las implicaciones y posibles futuras aplicaciones de esta herramienta

CAPÍTULO I: FORMULACIÓN DEL PROBLEMA

Los ACV son una de las principales causas de discapacidad en adultos mayores a nivel mundial, afectando considerablemente su calidad de vida. Estos eventos pueden provocar secuelas cognitivas y físicas severas, requiriendo una rehabilitación extensa y personalizada.

En Chile, la incidencia de ACV ha aumentado, lo que subraya la necesidad de enfoques terapéuticos innovadores que puedan complementar las estrategias de rehabilitación tradicionales.

1.1 DESAFÍOS EN EL MONITOREO DE PACIENTES

Los kinesiólogos del Hospital geriátrico La Paz de la Tarde de la comuna de Limache, que trabajan en el ámbito de la rehabilitación de pacientes con accidentes cerebrovasculares (ACV), enfrentan varios desafíos significativos en su labor diaria. Uno de los problemas más críticos es la dificultad para obtener y analizar datos precisos y en tiempo real sobre el progreso cognitivo y motor de sus pacientes. Estos desafíos incluyen:

Falta de Herramientas de Monitoreo Eficaces: La mayoría de los métodos tradicionales de evaluación se basan en observaciones subjetivas y pruebas estandarizadas que no siempre reflejan de manera precisa el estado cognitivo y motor del paciente. Estas evaluaciones pueden ser esporádicas y no proporcionan un seguimiento continuo ni detallado del progreso del paciente.

Recopilación de Datos Insuficiente: La recopilación manual de datos a través de notas clínicas y formularios puede ser ineficiente y propensa a errores. Esto dificulta la obtención de una visión clara y completa del avance del paciente, limitando la capacidad de los kinesiólogos para ajustar y personalizar los planes de rehabilitación de manera efectiva.

- **Falta de Interacción con Tecnología Avanzada:** Aunque existen tecnologías avanzadas para el monitoreo y la rehabilitación, no cuentan con acceso a estas herramientas debido a restricciones presupuestarias o falta de capacitación del personal. Esto impide que los kinesiólogos aprovechen al máximo las ventajas que la tecnología puede ofrecer para el seguimiento y tratamiento de sus pacientes.
- **Necesidad de Datos Objetivos y Cuantificables:** Para optimizar los planes de tratamiento y justificar las decisiones clínicas, los kinesiólogos necesitan datos objetivos y cuantificables sobre el desempeño de los pacientes. Sin estos datos, es difícil medir con precisión el impacto de las intervenciones terapéuticas y realizar ajustes informados.
- **Evaluaciones Discontinuas:** La naturaleza episódica de las evaluaciones clínicas tradicionales significa que cualquier mejora o deterioro entre las sesiones puede pasar desapercibida. Esto puede resultar en una falta de intervención temprana y, por lo tanto, en oportunidades perdidas para maximizar la recuperación del paciente.

1.2 PROPUESTA DE SOLUCIÓN

La estimulación cognitiva a través de videojuegos ha demostrado ser una buena alternativa en la rehabilitación de pacientes con ACV, ayudando a mejorar habilidades como la memoria, la atención y el razonamiento lógico. La literatura científica sugiere que los videojuegos no solo son una fuente de entretenimiento, sino también herramientas efectivas para mantener y fortalecer la salud cognitiva (Irazoki, y otros, 2020).

Para abordar estos desafíos, se propone la implementación de "CogniGame", una plataforma de videojuegos diseñada para el monitoreo y rehabilitación de pacientes con ACV.

1.3 OBJETIVOS

La plataforma de videojuego "CogniGame" nace con la intención de facilitar la labor de los kinesiólogos con una herramienta tecnológica que permita monitorear de manera continua el estado cognitivo y motor de los pacientes. A través de la gamificación de las tareas de rehabilitación, "CogniGame" no solo hará que el proceso sea más entretenido y motivador para los pacientes, sino que también proporciona datos detallados y objetivos.

1.3.1 OBJETIVO GENERAL

Generar una plataforma de videojuegos con enfoque cognitivo, que permita capturar los datos del adulto mayor a medida que avanza el juego que permita a los especialistas obtener datos para analizar el estado del paciente durante el tratamiento.

1.3.2 OBJETIVOS ESPECÍFICOS

Desarrollar una plataforma de videojuego accesible y efectiva que pueda ser utilizada por adultos mayores con diferentes niveles de habilidad tecnológica.

Crear un juego para la estimulación cognitiva y rehabilitación, enfocándose en áreas específicas afectadas por enfermedades neurodegenerativas y ACV.

Desarrollar una base de datos para proporcionar datos detallados y precisos que ayuden a los profesionales de la salud en la toma de decisiones.

COGNIGAME, permitirá un seguimiento constante del estado cognitivo y motor del paciente a través de juegos interactivos, proporcionando datos en tiempo real, el almacenamiento de datos de manera automática de tal modo que se puedan generar métricas sobre el desempeño del paciente, facilitando la evaluación y personalización de los planes de tratamiento. La naturaleza interactiva y constante de la plataforma permite detectar mejoras o deterioros de manera oportuna, permitiendo intervenciones tempranas y más efectivas.

CAPÍTULO II: MARCO TEÓRICO.

Para sustentar el proyecto han sido revisados diversos estudios que refutan y promueven la idea de generar alternativas tecnológicas para el tratamiento de adultos mayores con problemas cognitivos.

La investigación de (Irazoki, y otros, 2020) proporciona evidencia sustancial sobre los efectos positivos de los videojuegos en la cognición, especialmente en adultos mayores. Los hallazgos de este estudio indican que los videojuegos pueden mejorar varias habilidades cognitivas, incluida la memoria, la atención y el procesamiento visual-espacial. Además, el trabajo de (Fernández-Calvo, Rodríguez-Pérez, Contador, Rubio-Santorum, & Ramos Campos, 2011) resalta la capacidad de los videojuegos para mejorar la flexibilidad cognitiva y la función ejecutiva, aspectos fundamentales en la preservación de la cognición a lo largo del envejecimiento.

La revisión de (Pazán Jurado, 2015) destaca la importancia de los videojuegos como herramientas de intervención terapéutica en el contexto de la salud mental de los adultos mayores. Se observa que los videojuegos pueden ofrecer una forma efectiva y accesible de estimulación cognitiva, lo que potencialmente puede retrasar el deterioro cognitivo asociado con el envejecimiento y las enfermedades neurodegenerativas. Además, el trabajo de (Moscardi, 2018) subraya la diversidad de aplicaciones clínicas de los videojuegos, haciendo hincapié en su utilidad para mejorar la salud mental y la cognición en adultos mayores.

En conjunto, estos estudios sugieren que los videojuegos ofrecen un medio eficaz para la estimulación cognitiva en adultos mayores, ofreciendo una variedad de desafíos mentales que pueden mantener y mejorar diversas funciones cognitivas. La interactividad, la retroalimentación y la adaptabilidad de los videojuegos los convierten en herramientas atractivas y efectivas para la preservación de la cognición en el envejecimiento.

2.1 CONCEPTO DE VIDEOJUEGO COGNITIVO.

Un videojuego cognitivo, se puede definir como una forma de entretenimiento digital diseñada específicamente para estimular y mejorar las funciones cognitivas del jugador, ya que, a diferencia de los videojuegos tradicionales, cuyo principal objetivo es el entretenimiento. Los videojuegos cognitivos están diseñados con actividades que desafían y ejercitan la mente, con el propósito de mejorar habilidades cognitivas como la memoria, la atención, el razonamiento y la concentración, entre otros. Estos suelen presentar una variedad de desafíos y actividades que requieren que el jugador use y mejore sus habilidades mentales mientras se divierte.

Los videojuegos cognitivos presentan desafíos diseñados para estimular áreas cognitivas específicas, como la memoria de trabajo, la atención selectiva o la flexibilidad mental. Los niveles de dificultad suelen adaptarse para desafiar al jugador de manera progresiva, aumentando gradualmente la complejidad de los desafíos a medida que avanza en el juego sin tener que perder, lo cual es una de las variantes más relevantes a diferencia de los videojuegos tradicionales, ya que utiliza el tiempo como forma de determinar la finalización del juego.

Proporcionan retroalimentación inmediata sobre el desempeño del jugador, lo que permite identificar áreas de mejora y motivar al jugador a seguir avanzando.

2.2 ENTORNOS TECNOLÓGICOS.

Para el desarrollo de la plataforma de videojuego se utilizarán distintas tecnologías que crearán en conjunto un ecosistema sostenible y escalable para futuras modificaciones que requieran implementar.

2.2.1 JAVASCRIPT

Es uno de los lenguajes de programación más populares y ampliamente utilizados en el desarrollo web, esto significa que hay una gran cantidad de recursos disponibles, incluidas bibliotecas y frameworks como Phaser. Además, es ejecutado por todos los navegadores web modernos sin necesidad de complementos adicionales, lo que garantiza que el juego a desarrollar sea accesible, independientemente del navegador que utilicen.

Es un lenguaje versátil que permite crear una variedad de aplicaciones interactivas y dinámicas en la web, tiene una capacidad para crear interactividad y proporcionar respuestas en tiempo real en las aplicaciones web. Esto es fundamental para los juegos cognitivos, ya que se necesita que los juegos respondan de manera rápida y precisa a las |

2.2.2 REACT

Es una librería que utiliza JavaScript como lenguaje de programación y es conocido por su capacidad para crear interfaces de usuario altamente interactivas y receptivas. Lo cual, es crucial para mantener comprometidos a los usuarios, especialmente a adultos mayores, al proporcionar una experiencia de usuario fluida y atractiva. Es altamente flexible y se integra bien con otras bibliotecas y frameworks, lo que permite la incorporación de nuevas características y funcionalidades de manera eficiente, facilitando la adaptación del sistema a futuras necesidades y requisitos. Además, cuenta con una gran comunidad de desarrolladores y una amplia variedad de recursos disponibles, incluidos tutoriales, documentación y bibliotecas de código abierto, brindando acceso a soporte y ayuda adicional en caso de problemas o desafíos durante el desarrollo.

2.2.3 MONGO DB

Es una base de datos NoSQL, lo que significa que no requiere un esquema de datos fijo y predefinido, permitiendo una mayor flexibilidad en la estructura de los datos, lo que puede

ser beneficioso cuando se trabaja con información variada y en constante cambio, como en un sistema de juegos cognitivos. Utiliza un modelo de datos orientado a documentos, que se asemeja más a la estructura de datos utilizada en la programación orientada a objetos. Esto facilita la representación de datos complejos y anidados, lo que puede ser útil para almacenar información relacionada con perfiles de usuarios, progreso en juegos y resultados de evaluaciones cognitivas. También ofrece una rápida velocidad de desarrollo debido a su esquema flexible y su capacidad para manejar datos semiestructurados, acelerando el proceso de desarrollo del sitio web al permitir una iteración más rápida y una adaptación fácil a los cambios en los requisitos del proyecto.

2.2.4 PHASER

Está específicamente diseñado para el desarrollo de juegos en HTML5, lo que lo hace ideal para crear experiencias interactivas y de entretenimiento, posee una curva de aprendizaje suave y una documentación exhaustiva que facilita su adopción, incluso para desarrolladores con poca experiencia en juegos. Ofrece una amplia gama de funciones y herramientas que permiten una gran flexibilidad y personalización en el desarrollo de juegos, siendo una solución sólida y efectiva para crear experiencias interactivas y entretenidas destinadas a adultos mayores. Su facilidad de uso, flexibilidad y soporte lo convierten en una opción excelente para este tipo de proyecto, especialmente teniendo en cuenta que se utilizará JavaScript como lenguaje de programación.

2.2.5 GITHUB

GitHub es una plataforma ampliamente utilizada para el desarrollo colaborativo de software, lo que facilita el trabajo en equipo y la gestión de versiones del código fuente. Esto significa que se puede trabajar con otros colaboradores de manera eficiente, supervisar los cambios realizados y mantener un historial completo de las modificaciones realizadas en el proyecto. Si bien es cierto que la plataforma será desarrollada solamente por mi persona, luego de terminar el presente proyecto de título, lo pueden tomar otros desarrolladores e implementar nuevas funciones que se integren a la aplicación que se generará, lo cual hace más imprescindible su uso. GitHub permitirá mantener un historial

completo de todos los cambios que se realicen en el código de la plataforma de videojuego, brindando la posibilidad de retroceder a versiones anteriores en caso de necesidad.

2.3 METODOLOGÍA DE TRABAJO.

Para desarrollar el proyecto de manera eficiente es necesario utilizar una metodología de trabajo que permita gestionar eficazmente la construcción de la plataforma de juego para mantener un flujo de trabajo fluido. Para este proyecto se escogió el modelo de procesos Cascada, con una metodología de desarrollo de software lineal y secuencial que se divide en distintas fases, cada una de las cuales debe completarse antes de pasar a la siguiente. Dentro de sus principales características se puede destacar que el desarrollo progresa de manera lineal a través de las distintas fases sin necesidad de volver atrás. Además, es secuencial, por lo que cada etapa debe terminar por completo antes de continuar con la siguiente. Finalmente, esto permite planificar y ejecutar el desarrollo del videojuego de manera estructurada y ordenada.

CAPÍTULO III: ANÁLISIS.

Como parte del proceso investigativo, se realizaron búsquedas de aplicaciones similares con el objetivo de comparar lo que existe, versus lo que se desarrollará, encontrando varias diferencias en el mercado global.

3.1 PLATAFORMAS DE ENTRENAMIENTO COGNITIVO.

En el ámbito de la rehabilitación y el entrenamiento cognitivo, diversas plataformas en línea han emergido como herramientas efectivas para mejorar y mantener la salud cognitiva de personas de todas las edades. A continuación, se describen tres de las principales plataformas que han ganado reconocimiento por sus enfoques innovadores y su efectividad en la estimulación y rehabilitación de funciones cognitivas.

- **Braingymer:** Es un programa de entrenamiento cerebral en línea que ofrece una variedad de juegos y ejercicios diseñados para mejorar y mantener la salud cognitiva. Está dirigida a personas de todas las edades que desean ejercitar su cerebro y mejorar sus habilidades cognitivas.
- **CogniFit:** Es una plataforma en línea de entrenamiento cerebral y evaluación cognitiva que ofrece una variedad de juegos y ejercicios diseñados para estimular y mejorar diversas funciones cognitivas. Está dirigida tanto a profesionales de la salud como a usuarios individuales interesados en mejorar su salud cerebral y su rendimiento cognitivo.
- **NeuronUP:** Es una plataforma online de estimulación cognitiva diseñada para profesionales de la salud, que trabajan con personas que presentan dificultades cognitivas o trastornos neurológicos. Ofrece una amplia variedad de actividades y herramientas diseñadas para mejorar y rehabilitar funciones cognitivas específicas, como la memoria, la atención, el razonamiento, la percepción y las funciones ejecutivas.

Las aplicaciones mencionadas anteriormente buscan estimular cognitivamente mediante videojuegos, pero ninguna se enfoca específicamente en adultos mayores. Además, para adquirir todas sus funciones, como la de obtener informes en base a la data, generar un tratamiento e incluso acceder a profesionales la persona debe suscribirse a un plan con costo mensual o anual.

Por lo anterior, es que el presente proyecto de título se diferencia de las otras aplicaciones, ya que, está hecha para ser utilizada por profesionales de la salud del departamento de geriatría del Hospital de Limache y extendida a sus pacientes para recopilar datos y ayudarlos en el tratamiento cognitivo sin costo asociado, innovando tecnológicamente en la estimulación cognitiva.

3.2 REQUERIMIENTOS

3.2.1 REQUERIMIENTOS FUNCIONALES

- RF1: El sistema debe permitir a los administradores agregar, eliminar y gestionar cuentas de kinesiólogos.
- RF2: Cada kinesiólogo debe poder registrar y gestionar los perfiles de sus pacientes.
- RF3: Monitoreo y Evaluación. La plataforma debe proporcionar herramientas para el monitoreo continuo del estado cognitivo y motor de los pacientes.
- RF4: Deben generarse informes detallados sobre el progreso de los pacientes, accesibles tanto para los kinesiólogos como para los administradores.
- RF5: Los datos de monitoreo deben actualizarse en tiempo real para reflejar el estado actual del paciente.
- RF6: La plataforma debe incluir al menos un juego diseñado para estimular funciones cognitivas específicas, como memoria, atención, razonamiento y percepción.

- RF7: Los juegos deben adaptarse al nivel de habilidad del paciente, aumentando en dificultad a medida que el paciente progresa.

3.2.2 REQUERIMIENTOS NO FUNCIONALES

- RNF1: La interfaz debe ser intuitiva y accesible para usuarios con diferentes niveles de habilidad tecnológica.
- RNF2: La plataforma debe garantizar la protección de los datos personales y médicos de los pacientes mediante técnicas de cifrado y control de acceso.
- RNF3: Deben implementarse medidas para asegurar la integridad y confidencialidad de los datos.
- RNF4: La plataforma debe garantizar la protección de los datos personales y médicos de los pacientes mediante técnicas de cifrado y control de acceso.
- RNF5: Deben implementarse medidas para asegurar la integridad y confidencialidad de los datos.

CAPÍTULO IV: DISEÑO DEL SISTEMA

En este capítulo se abordará el diseño del sistema para la plataforma de videojuego "CogniGame". Se detallarán los patrones de diseño aplicados y la interacción entre los diferentes componentes del sistema.

4.1 PATRÓN DE DISEÑO

Los patrones de diseño, son esquemas generales que se pueden aplicar para resolver problemas de diseño específicos en diversas situaciones. Los patrones de diseño no son implementaciones completas, sino que proporcionan una descripción de cómo organizar clases y objetos para resolver un problema particular de manera flexible y eficiente, es por esto, que para la implementación del juego se utilizará el patrón de diseño MVC. Este es uno de los patrones arquitectónicos más conocidos y utilizados en el desarrollo de aplicaciones, especialmente en el contexto de interfaces de usuario y aplicaciones web. Este patrón separa una aplicación en tres componentes principales: el modelo, la vista y el controlador.

- **Modelo:** Representa los datos de la aplicación y la lógica de negocio. Es responsable de recuperar, almacenar y manipular los datos.
- **Vista:** Representa la interfaz de usuario y la presentación de los datos del modelo. La vista muestra los datos y recibe la entrada del usuario.
- **Controlador:** Actúa como intermediario entre el modelo y la vista, recibe la entrada del usuario desde la vista, la procesa y realiza las actualizaciones necesarias en el modelo.

4.2 PERFILES DE USUARIO

La **plataforma** tendrá un sistema de acceso, la cual discrimina entre los 3 roles:

- **SUPERADMIN:** Su vista contempla su propia información, una tabla con el registro de kinesiólogos, opción de agregar kinesiólogos y modificar el estado de éstos, entre activo e inactivo.

- **KINESIOLOGO:** Tendrá una vista similar a la del Superadmin, la diferencia está en que verá una tabla de pacientes, pudiendo agregar y modificar el estado de éstos. Además, podrá observar la data obtenida del juego por cada paciente y en cada partida, observando ciertas estadísticas que serán relevante para tomar decisiones que promuevan un enfoque y adaptación del tratamiento en el adulto mayor.
- **PACIENTE:** Su vista contemplará su información al igual que los otros roles, como también, la visualización del juego, observará las instrucciones y detalle del funcionamiento del juego. Una vez termina de jugar, podrá observar algunos datos básicos que servirán de motivación para enfrentar nuevamente el videojuego de manera lúdica, promoviendo el interés de este usuario a mejorar cognitiva y recreativamente.

4.3 VISTAS DE USUARIO

El uso del patrón de diseño implementado se presentan las diferentes vistas de la aplicación.

- **Vista Login Page:** La pantalla de inicio de sesión de la plataforma.
- **Vista Superadmin:** La vista para el administrador del sistema que incluye el Formulario de Registro utilizado para registrar nuevos usuarios.
- **Vista Kinesiólogo:** La vista para los kinesiólogos que utilizan la plataforma para monitorear a los pacientes. Se incluye un listado de estadísticas relevantes que deben ser observadas.
- **Vista Paciente:** La vista utilizada por los pacientes para interactuar con los juegos.

4.3.1 INICIO DE SESIÓN

El acceso al sistema se apreciará como muestra la ilustración 1, minimalista y dentro del marco de la cognición. El lado derecho de la ilustración asemeja las conexiones neuronales

con un tono suave para dar sensación de relajación y, por el lado izquierdo, podemos apreciar el Logo de la plataforma, el cual posee el nombre del proyecto, representando los videojuegos cognitivos tanto en su nombre como en su contenido.

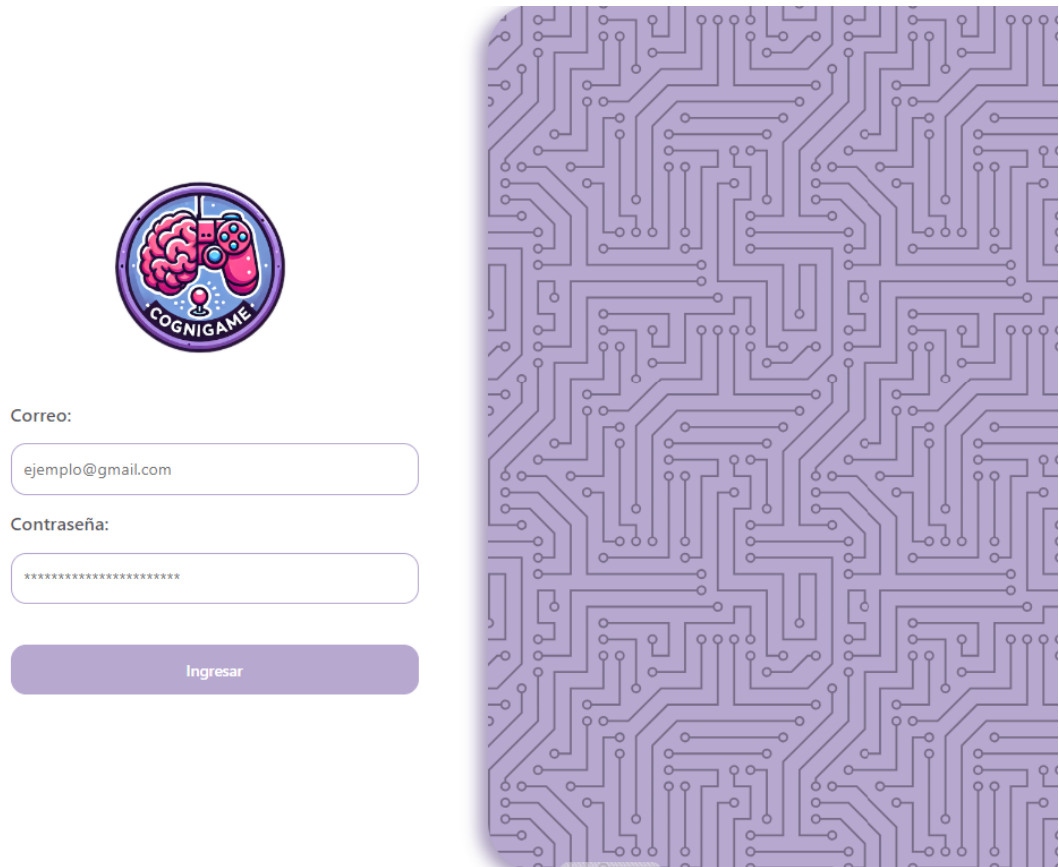
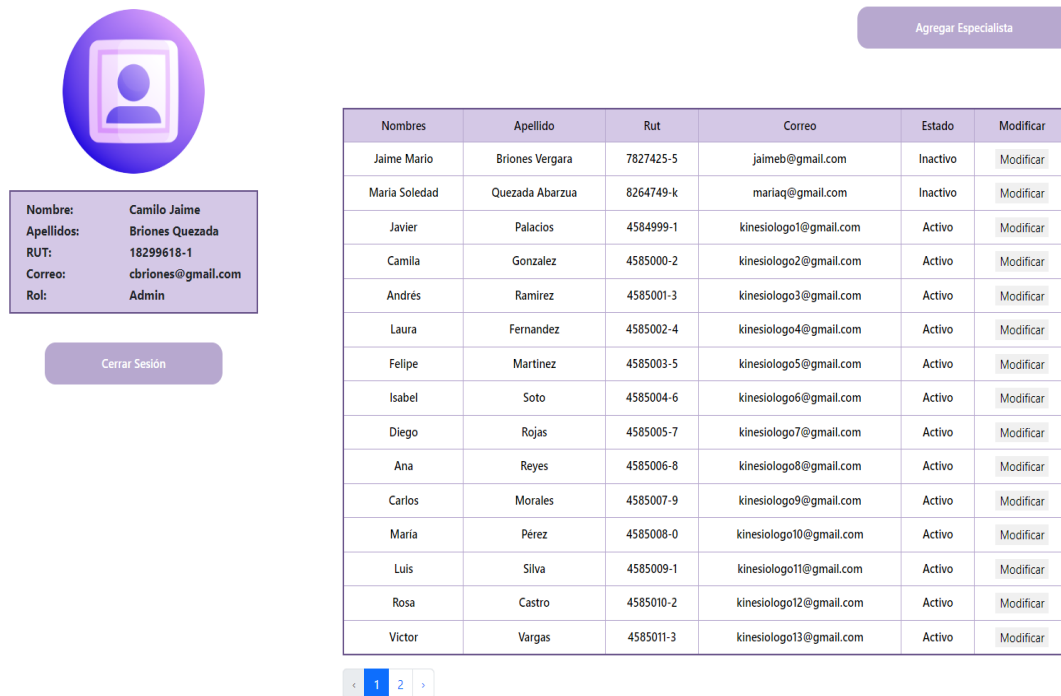


Ilustración 1: Vista login page CogniGame.

4.3.2 VISTA SUPERADMIN

La presente vista, como se muestra en la ilustración 2, contendrá un sidebar mostrando la información del Superadmin que ingresó al sistema, mostrando el listado de los especialistas teniendo opción de modificar los campos del kinesiólogo, exceptuando el Rut y el Rol. Además de un buscador por Rut para dar con alguien en específico.

Es de vital importancia que siempre deba quedar registro de quienes han sido parte del sistema por temas de seguridad, por lo mismo, es que, el estado que pueden poseer los especialistas son 2, Activo e Inactivo.



The screenshot displays the Superadmin interface. On the left, there is a user profile for Camilo Jaime Briones Quezada, with details including RUT (18299618-1), email (cbriones@gmail.com), and role (Admin). A 'Cerrar Sesión' button is located below the profile. On the right, there is a table listing specialists, with columns for Nombres, Apellido, Rut, Correo, Estado, and Modificar. A 'Agregar Especialista' button is positioned above the table. At the bottom, there is a pagination control showing page 1 of 2.

Nombres	Apellido	Rut	Correo	Estado	Modificar
Jaime Mario	Briones Vergara	7827425-5	jaimeb@gmail.com	Inactivo	Modificar
Maria Soledad	Quezada Abarzua	8264749-k	mariaq@gmail.com	Inactivo	Modificar
Javier	Palacios	4584999-1	kinesiologo1@gmail.com	Activo	Modificar
Camila	Gonzalez	4585000-2	kinesiologo2@gmail.com	Activo	Modificar
Andrés	Ramirez	4585001-3	kinesiologo3@gmail.com	Activo	Modificar
Laura	Fernandez	4585002-4	kinesiologo4@gmail.com	Activo	Modificar
Felipe	Martinez	4585003-5	kinesiologo5@gmail.com	Activo	Modificar
Isabel	Soto	4585004-6	kinesiologo6@gmail.com	Activo	Modificar
Diego	Rojas	4585005-7	kinesiologo7@gmail.com	Activo	Modificar
Ana	Reyes	4585006-8	kinesiologo8@gmail.com	Activo	Modificar
Carlos	Morales	4585007-9	kinesiologo9@gmail.com	Activo	Modificar
Maria	Pérez	4585008-0	kinesiologo10@gmail.com	Activo	Modificar
Luis	Silva	4585009-1	kinesiologo11@gmail.com	Activo	Modificar
Rosa	Castro	4585010-2	kinesiologo12@gmail.com	Activo	Modificar
Victor	Vargas	4585011-3	kinesiologo13@gmail.com	Activo	Modificar

Ilustración 2: Vista Superadmin.

Por otra parte, como se observó en la ilustración 2, posee un botón que permite registrar nuevos kinesiólogos, el cual, renderizará un formulario para el registro a realizar por el Superadmin y posterior ingreso al sistema, tal como se refleja en la ilustración 3.

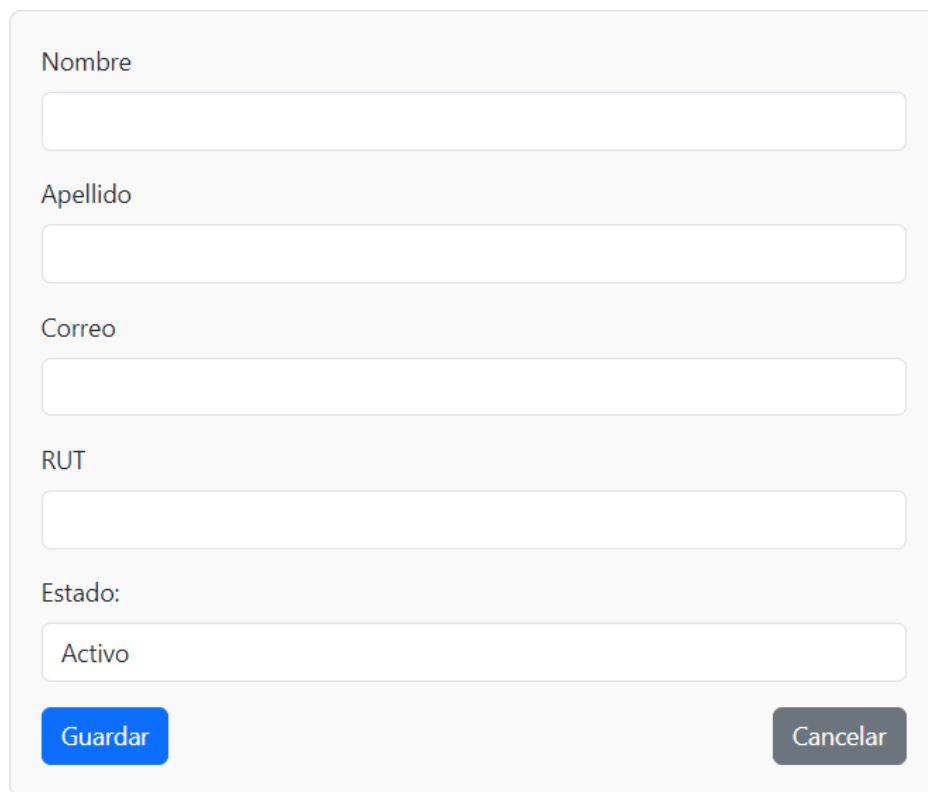
Un formulario de registro con un fondo gris claro y bordes redondeados. Contiene cinco campos de entrada blancos con bordes grises, uno para cada campo de texto. Los campos están etiquetados como 'Nombre', 'Apellido', 'Correo', 'RUT' y 'Estado:'. El campo 'Estado:' tiene un menú desplegable que muestra la opción 'Activo'. En la parte inferior del formulario hay dos botones: 'Guardar' (azul) y 'Cancelar' (gris oscuro).

Ilustración 3: Formulario de registro.

4.3.3 VISTA KINESIOLOGO

De manera muy similar a la vista del Superadmin, la vista del kinesiólogo, también visualizará sus datos, la diferencia está en que su tabla mostrará el listado de pacientes activos e inactivos del sistema, su diagnóstico, como también la opción para modificar sus datos exceptuando el Rut, lo que se puede apreciar en la ilustración 4. También, podrá observar los stats que cada paciente posee al presionar sobre el ícono en el campo Stats de la tabla.



Nombre: Jaime Mario

Apellidos: Briones Vergara

RUT: 7827425-5

Correo: jbriones@gmail.com

Rol: Kinesiólogo

Cerrar Sesión

Agregar Paciente

Nombres	Apellido	Rut	Correo	Estado	Diagnóstico	Modificar	Stats
Mario	Briones	18914887-9	mario@gmail.com	Activo	Alzheimer	Modificar	👁
Jose	Perez	20356789-3	jose@gmail.com	Activo	Alzheimer	Modificar	👁
Ana	Sanchez	19456723-6	ana@gmail.com	Activo	Demencia Frontotemporal	Modificar	👁
Laura	Gomez	17890234-1	laura@gmail.com	Activo	Amnesia global transitoria	Modificar	👁
Luis	Ramirez	18456789-2	luis@gmail.com	Activo	Alzheimer	Modificar	👁
Marta	Lopez	18234567-8	marta@gmail.com	Activo	Encefalopatía traumática crónica	Modificar	👁
Pedro	Hernandez	18765432-0	pedro@gmail.com	Activo	Amnesia global transitoria	Modificar	👁
Carmen	Martinez	19012345-6	carmen@gmail.com	Activo	Alzheimer	Modificar	👁
Juan	Fernandez	18923456-1	juan@gmail.com	Activo	Demencia Frontotemporal	Modificar	👁
Alicia	Torres	18345678-9	alicia@gmail.com	Activo	Encefalopatía traumática crónica	Modificar	👁
Carlos	Diaz	19567890-2	carlos@gmail.com	Activo	Alzheimer	Modificar	👁
Patricia	Vega	19123456-7	patricia@gmail.com	Activo	Encefalopatía traumática crónica	Modificar	👁
Antonio	Molina	19678901-3	antonio@gmail.com	Activo	Demencia Frontotemporal	Modificar	👁
Rosa	Morales	19789012-4	rosa@gmail.com	Activo	Demencia Frontotemporal	Modificar	👁
Miguel	Pardo	19890123-5	miguel@gmail.com	Activo	Alzheimer	Modificar	👁

< 1 2 >

Ilustración 4: Vista kinesiólogo.

La vista, como muestra la ilustración 5, incluye la fecha y hora del sistema de cada juego finalizado que ha tenido el paciente y una opción donde observar dichos resultados. Esto facilita un seguimiento cronológico y detallado del desempeño del paciente en las diferentes sesiones de juego.

Mario Andres Briones Quezada				18914887-9
ID	Fecha	Hora	Resultado	
1	20-05-2024	09:20	👁	
2	21-05-2024	09:21	👁	
3	22-05-2024	09:22	👁	
4	23-05-2024	09:23	👁	
5	24-05-2024	09:24	👁	

« < 1 2 3 ... 9 > »

Ilustración 5: Listado de stats a observar.

Al presionar sobre los resultados mostrados en la ilustración anterior, el sistema proporcionará acceso a una recopilación detallada de la data obtenida por el adulto mayor durante la sesión de juego. Esta información incluirá diversos parámetros relevantes como el número total de estímulos respondidos y no respondidos, tiempos de reacción específicos, y otros indicadores de rendimiento.

Este enfoque no solo facilitará una evaluación continua y personalizada, sino que también proporcionará una base para ajustar y personalizar las intervenciones terapéuticas. La capacidad de observar y analizar estos datos en intervalos regulares asegura que cualquier cambio en el estado cognitivo del paciente sea detectado tempranamente, permitiendo una intervención oportuna.

Además, la presentación visual de estos datos, como se muestra en el sistema, es diseñada para ser intuitiva y accesible, permitiendo a los profesionales de la salud extraer información significativa con facilidad, la cual, también puede ser impresa a través de la funcionalidad implementada, tal como se aprecia en la ilustración 6.

Mario Andres Briones Quezada18914887-9

Fecha: 20-05-2024

Hora: 09:20

Imprimir

RESULTADOS DEL JUEGO

Lémures Totales:

16

Lémures Clickeados:

14

Lémures No Clickeados:

2

Tiempo(s) Transcurrido(s) En No Clockear Un Lémur:

[29617, 43228]

Tiempo De Respuesta En Cada Aparición De Lémur:

[2278, 5629, 3668, 1679, 2292, 4747, 2456, 1330, 1778, 1803, 799, 1212, 1414, 727]

Tiempo De Respuesta En Hacer Click Entre Un Lémur y Otro:

[6954, 1625, 1614, 4216, 11262, 1112, 2277, 10040, 3227, 2199, 3616, 3404, 2516]

Ilustración 6: Stats de un determinado juego y paciente.

Bajo la presente vista también se podrán apreciar gráficos, como se evidencia en las ilustraciones 7 y 8, que ayuden a brindar de manera visual los stats obtenidos en el

videojuego completado por el paciente como también permite la comparativa aplicando filtros de partidas como las últimas 7 del día, todas las del día y por tramo de días. Estos gráficos proporcionan una representación clara y concisa del rendimiento del paciente en diferentes aspectos del juego, permitiendo identificar fácilmente tendencias y áreas que requieren mejora.

Respuesta en Cada Aparición de Lémur

R = Respuesta



Ilustración 7: Gráfico de línea en base a respuesta en cada aparición de lémur.

Respuesta en Hacer Click Entre Lémures

R = Respuesta



Ilustración 8: Gráfico de línea en base a respuesta de clic entre lémures.

4.3.4 VISTA PACIENTE

La vista de este rol está diseñada, como se observa en la ilustración 9, para proporcionar una comprensión integral del videojuego creado, sus reglas y la información relevante sobre los beneficios de utilizar CogniGame. En la parte izquierda de la pantalla, se presentan los datos del paciente, lo cual incluye su nombre, apellido, RUT, correo electrónico y rol, junto con una breve descripción del proyecto y los beneficios que este videojuego puede aportar a la salud cognitiva. Esta sección es crucial, ya que ofrece un contexto personalizado y asegura que cada paciente entienda cómo su participación y progreso son monitorizados y valorados.

La parte central de la vista está dedicada al videojuego en sí. Aquí es donde el paciente interactúa directamente con CogniGame. La interfaz de usuario está diseñada para ser intuitiva y atractiva, permitiendo que los pacientes se sumerjan en la experiencia del juego sin dificultades. El botón central de "Game" inicia la partida, y la pantalla muestra de manera dinámica y visualmente atractiva el entorno del juego, los elementos interactivos y los desafíos que el paciente debe completar.

Finalmente, en la parte derecha de la pantalla, se encuentran detalladas las reglas del juego. Esta sección es fundamental para guiar al paciente en cómo jugar correctamente y maximizar los beneficios cognitivos de la actividad. Las reglas están explicadas de manera clara y concisa, asegurando que el paciente entienda cada paso del proceso del juego, desde el inicio hasta la finalización de una partida. Además, la repetición y el incremento gradual en la dificultad de las tareas están diseñados para mantener el interés del paciente y proporcionar una progresión continua en la estimulación cognitiva.

En conjunto, esta vista no solo facilita la interacción del paciente con CogniGame, sino que también asegura que el paciente esté informado y cómodo con su uso, maximizando así la eficacia de la herramienta como un recurso de estimulación cognitiva.

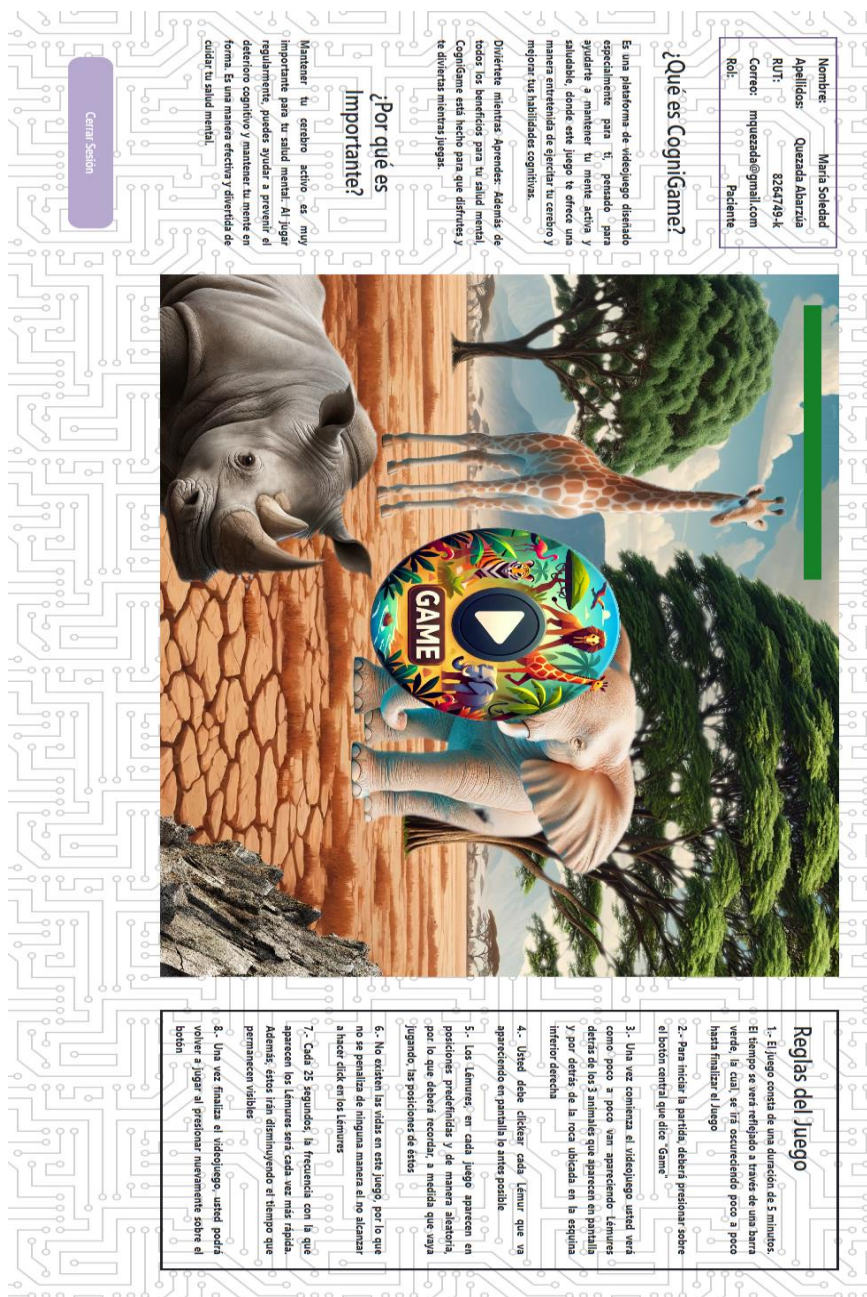


Ilustración 9: Vista paciente.

Las imágenes utilizadas en el juego han sido generadas mediante inteligencia artificial, lo que garantiza que cada una de ellas es completamente original y de creación propia. Este enfoque no solo añade un toque único y personalizado al diseño del juego, sino que también asegura que no se infringen derechos de autor de terceros. Al emplear inteligencia artificial para la generación de las imágenes, hemos podido obtener gráficos detallados y

visualmente atractivos que se integran perfectamente en el entorno del juego, ofreciendo una experiencia visualmente enriquecedora para los jugadores. Esta metodología innovadora de creación de imágenes contribuye significativamente a la singularidad y autenticidad del proyecto, destacando nuestro compromiso con la originalidad y la creatividad en el desarrollo del videojuego.

Una vez que el juego ha comenzado, el paciente deberá estar constantemente atento a la aparición de los lémures dentro del entorno del juego, lo que requiere un alto nivel de concentración, acorde se ve en la ilustración 10. En cada aparición, cada lémur lleva consigo un reloj que se detiene en el momento en que es clickeado. A medida que el juego avanza, el adulto mayor, a través del trabajo cognitivo continuo, irá fortaleciendo su agilidad mental, reflejada en tiempos de respuesta cada vez más rápidos.



Ilustración 10: Aparición de un lémur.

Este proceso no solo aplica a la aparición inicial de cada lémur, sino también al periodo entre clics, es decir, desde que se hace clic por primera vez en un lémur hasta que se hace clic en el siguiente. Este ciclo se repite de manera iterativa, desafiando constantemente la capacidad del jugador para reaccionar rápidamente a las apariciones de los lémures, como se aprecia en la ilustración 11.

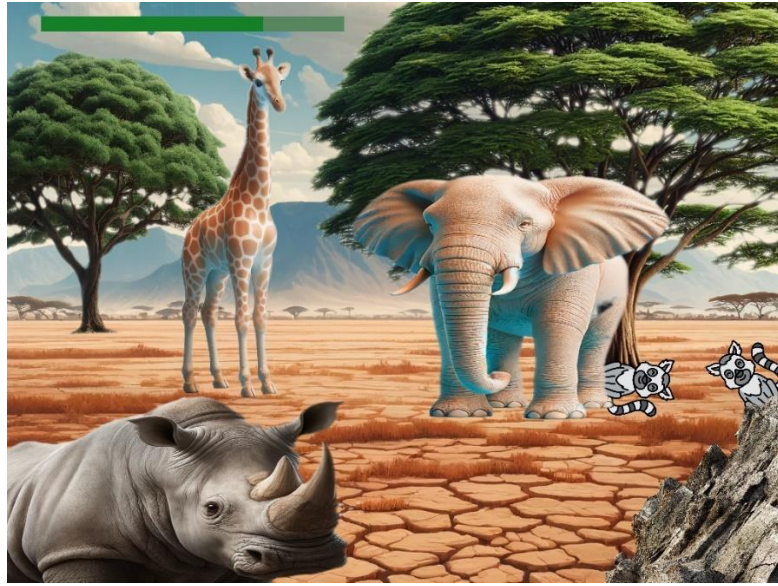


Ilustración 11: Aparición de dos lémures.

Además, el juego está diseñado para trabajar la memoria del jugador. Las posiciones en las que los lémures aparecen están predefinidas, pero la aparición en estas posiciones es aleatoria. A medida que el paciente juega repetidamente, comenzará a recordar las posiciones de los lémures, lo que le permitirá anticiparse y mejorar su atención espacial. Este proceso de recordar y anticipar las posiciones mejora significativamente su capacidad de memoria y concentración.

Conforme el juego avanza y el paciente se familiariza más con las mecánicas del mismo, se dará cuenta de que, basándose en la barra de tiempo, el ritmo del juego se acelera en ciertos puntos. Esta aceleración requiere que el paciente incremente aún más su concentración y agilidad, proporcionando un mayor desafío cognitivo. Esta dinámica no solo mantiene al paciente comprometido con el juego, sino que también asegura un constante esfuerzo mental, lo que es esencial para la estimulación y mejora de sus habilidades cognitivas.

CAPÍTULO V: CONSTRUCCIÓN DEL SISTEMA.

Para el desarrollo del videojuego, se utilizaron dos tecnologías fundamentales que permitieron crear una experiencia interactiva y atractiva para los usuarios. Estas tecnologías son Phaser y JavaScript. A continuación, se detalla cómo cada una de estas herramientas contribuyó al proyecto y se proporcionan ejemplos de código para ilustrar su implementación.

Phaser es un framework de desarrollo de juegos en HTML5 que se utiliza ampliamente debido a su flexibilidad y potencia. Este framework permite a los desarrolladores crear juegos que pueden ejecutarse en navegadores web modernos, ofreciendo soporte para una variedad de dispositivos y plataformas. Una de las principales ventajas de Phaser es su capacidad para manejar gráficos y animaciones de manera eficiente, lo que es esencial para crear una experiencia de juego fluida y atractiva.

5.1 IMPORTACIÓN Y DECLARACIÓN DE REFERENCIAS

La importación y declaración de referencias en el código del juego son etapas fundamentales que configuran la base sobre la cual se construye todo el desarrollo del videojuego. En primer lugar, según muestra la ilustración 12, se importan las bibliotecas y módulos necesarios, en este caso, Phaser y React. Phaser es una de las bibliotecas más populares y potentes para el desarrollo de juegos en HTML5, mientras que React proporciona la estructura para la interfaz del juego, facilitando la integración de los elementos interactivos en la página web. La importación de estas bibliotecas es esencial para acceder a sus funcionalidades y métodos que permiten la creación y gestión de escenas, sprites, eventos y la lógica del juego.

La declaración de referencias dentro del código es igualmente crucial. Estas referencias, almacenadas en variables utilizando `useRef`, permiten un control preciso sobre los elementos del juego. Por ejemplo, `gameContainer` es una referencia al contenedor del

juego en el DOM, asegurando que Phaser se inicialice correctamente dentro del elemento designado. Otras referencias, como `recentPositions`, `lastArrayUsed`, `lastPositionUsed`, `lemurEvent`, entre otras, son utilizadas para almacenar y manipular estados y datos que cambian dinámicamente durante el juego. Estas referencias facilitan el seguimiento de las posiciones de los lémures, el manejo de eventos de aparición y desaparición, y la gestión del tiempo y la interacción del jugador con los lémures.

Al definir estas referencias desde el inicio, el código mantiene un alto grado de organización y eficiencia. Por ejemplo, `clickTimes`, `nonClickTimes`, `clickIntervals`, `lemursClicked`, `lemursNotClicked` y `totalLemurs` son referencias que permiten registrar y analizar las acciones del jugador durante el juego. Estas referencias no solo ayudan en la gestión del juego en tiempo real, sino que también son esenciales para la recolección de datos que pueden ser utilizados posteriormente para evaluar el rendimiento cognitivo del jugador. La declaración de `startTime` y `lastClickTime` como referencias asegura que los tiempos de reacción y las duraciones de las interacciones sean precisas y fácilmente accesibles para cálculos posteriores.

Además, al utilizar `useRef` en lugar de variables regulares, se asegura que estas referencias mantengan su valor entre renderizados sin desencadenar nuevas renderizaciones del componente. Esto es particularmente importante en un entorno de juego, donde la eficiencia y la rapidez de respuesta son cruciales. En resumen, la importación y declaración de referencias establecen la infraestructura necesaria para un desarrollo de juego robusto, eficiente y organizado, proporcionando una base sólida sobre la cual se construye toda la lógica y la interacción del juego.

```
import { useEffect, useRef } from 'react';
import Phaser from 'phaser';

const Game = () => {
  const gameContainer = useRef(null);
  const recentPositions = useRef([]);
  const lastArrayUsed = useRef(null);
  const lastPositionUsed = useRef(null);
  const lemurEvent = useRef(null);
  const elapsedTime = useRef(0);
  const currentDelay = useRef(3600);
  const currentDuration = useRef(2600);
  const currentHold = useRef(1500);
  const gameStarted = useRef(false);
  const gameEnded = useRef(false);
  const clickTimes = useRef([]);
  const nonClickTimes = useRef([]);
  const clickIntervals = useRef([]);
  const lemursClicked = useRef(0);
  const lemursNotClicked = useRef(0);
  const totalLemurs = useRef(0);
  const lastClickTime = useRef(null);
  const startTime = useRef(Date.now());

```

Ilustración 12: Importación y declaración de referencias.

5.2 CONFIGURACIÓN DEL JUEGO

La configuración del juego es un paso esencial para establecer el entorno en el que se desarrollará la experiencia del jugador. En esta etapa, se define cómo se verá y se comportará el juego, tal como indica la ilustración 13. Asegurando que todos los elementos gráficos y lógicos funcionen correctamente. En el código proporcionado, la

configuración del juego se realiza utilizando un objeto de configuración que especifica varios parámetros clave.

Primero, se define el tipo de renderizado con `type: Phaser.AUTO`. Esto permite a Phaser elegir automáticamente entre WebGL y Canvas para renderizar el juego, asegurando el mejor rendimiento posible según las capacidades del navegador del usuario. La flexibilidad de elegir automáticamente entre estos dos métodos garantiza que el juego funcione de manera óptima en una amplia gama de dispositivos y navegadores.

Luego, se configuran las dimensiones del juego con `width: 1024` y `height: 768`. Estas dimensiones establecen el tamaño del lienzo de juego, determinando el área visible y jugable. Esta configuración es crucial para el diseño del juego, ya que define el espacio en el que se colocarán y moverán los elementos del juego. Un tamaño adecuado del lienzo asegura que todos los elementos sean visibles y accesibles para el jugador, mejorando la experiencia de juego.

El parámetro `parent: gameContainer.current` asocia el lienzo de juego con un contenedor específico en el DOM. Esto es esencial para integrar el juego dentro de una página web, asegurando que se renderice en el lugar correcto y sea parte del flujo del documento HTML. Al especificar este contenedor, se facilita la gestión del juego desde el componente React, permitiendo una integración más fluida con otros elementos de la interfaz web.

Finalmente, se definen las funciones `preload`, `create` y `update` dentro del objeto de configuración bajo la propiedad `scene`. Estas funciones son responsables de gestionar diferentes fases del ciclo de vida del juego. La función `preload` se encarga de cargar todos los recursos necesarios, como imágenes y sonidos, antes de que comience el juego. La función `create` inicializa los elementos del juego, como sprites y botones, y configura la lógica inicial. La función `update` se ejecuta en un bucle, actualizando constantemente el estado del juego y manejando la lógica dinámica y las interacciones del jugador.

Esta configuración detallada del juego asegura que todos los elementos necesarios estén preparados y gestionados correctamente, proporcionando una base sólida sobre la cual se puede construir y desarrollar toda la lógica del juego. Al definir claramente estos parámetros y funciones, se crea un entorno de desarrollo estructurado y eficiente, facilitando tanto el desarrollo inicial como el mantenimiento y la expansión del juego en el futuro.

```
useEffect(() => {  
  const config = {  
    type: Phaser.AUTO,  
    width: 1024,  
    height: 768,  
    parent: gameContainer.current,  
    scene: {  
      preload,  
      create,  
      update  
    }  
  };  
  
  const game = new Phaser.Game(config);
```

Ilustración 13: Código con la configuración del juego.

5.3 PRELOAD: CARGA DE RECURSOS

En la función preload, se cargan todas las imágenes que se utilizarán en el juego, asegurando que los recursos estén disponibles cuando se necesiten, como muestra la ilustración 14.

```
function preload() {
  this.load.image('background', './src/assets/back.jpg');
  this.load.image('arbol', './src/assets/arbol3.png');
  this.load.image('arbol2', './src/assets/arbol4.png');
  this.load.image('jirafa', './src/assets/jirafaa.png');
  this.load.image('lemur', './src/assets/lemur.png');
  this.load.image('piedra', './src/assets/piedra.png');
  this.load.image('rinoceronte', './src/assets/rinoceronte.png');
  this.load.image('elefante', './src/assets/elefante.png');
  this.load.image('playButton', './src/assets/play.png');
  this.load.image('gameOver', './src/assets/endgame.png');
}
```

Ilustración 14: Código con la carga de recursos.

5.4 CREATE: CREACIÓN DE LA ESCENA DEL JUEGO

La creación de la escena del juego es un proceso crucial que establece el entorno visual y funcional donde se desarrollará la interacción del jugador. Este proceso, como se aprecia en la ilustración 15, comienza con la carga de los recursos gráficos necesarios, como imágenes de fondo y sprites de los personajes y objetos del juego. Estos elementos visuales se integran para formar un entorno cohesivo y atractivo que mejora la experiencia del usuario. La función create es responsable de configurar y posicionar estos elementos en la pantalla del juego, asegurando que cada componente esté en su lugar y listo para la interacción.

En esta etapa, se agregan varios sprites al juego, como árboles, animales y rocas, cada uno con propiedades específicas de escala, ángulo y profundidad para crear una escena tridimensional convincente. Por ejemplo, se añaden sprites de un elefante, una jirafa, un rinoceronte y una piedra, ubicados estratégicamente en la pantalla para formar un paisaje natural. Estos elementos no solo mejoran la estética del juego, sino que también sirven como puntos de referencia para la aparición de los lémures, añadiendo una capa de complejidad y desafío a la experiencia de juego.

Además de los elementos visuales, la función create también incluye la configuración de botones interactivos, como el botón de inicio del juego y el botón de reinicio. El botón de inicio, una vez presionado, comienza el juego destruyendo el botón mismo para evitar

múltiples activaciones y desencadenando la secuencia de eventos del juego. El botón de reinicio, por su parte, aparece solo al finalizar el juego, permitiendo al jugador reiniciar la partida y comenzar de nuevo. Esta funcionalidad asegura que el jugador tenga control sobre el inicio y la repetición del juego, facilitando una experiencia de usuario fluida y sin interrupciones.

La creación de la escena del juego también incluye la configuración de una barra de tiempo visual que muestra la duración restante del juego. Esta barra es un indicador crucial para el jugador, proporcionando una referencia constante del tiempo disponible y aumentando la urgencia conforme el juego avanza. La barra de tiempo se actualiza dinámicamente durante la partida, disminuyendo gradualmente a medida que el tiempo pasa, lo que agrega una capa adicional de tensión y desafío.

La función create también establece los eventos temporizados que gestionan la aparición y movimiento de los lémures. Estos eventos aseguran que los lémures aparezcan y desaparezcan en intervalos regulares, ajustando dinámicamente la frecuencia y duración de sus apariciones para incrementar la dificultad del juego. Esta lógica es esencial para mantener el interés y la participación del jugador.

En resumen, la creación de la escena del juego es un proceso integral que establece el entorno visual y funcional del juego, configura los elementos interactivos y prepara el terreno para la dinámica de juego. Este proceso garantiza que cada componente esté correctamente posicionado y listo para la interacción, proporcionando una experiencia de usuario inmersiva y desafiante. La configuración de botones, la barra de tiempo y los eventos temporizados forman la base sobre la cual se desarrolla el juego, asegurando una experiencia de juego coherente y atractiva que fomenta la participación y el compromiso del jugador.

```

function create() {
    this.add.image(512, 384, 'background').setDepth(0);

    const arbol = this.add.sprite(100, 250, 'arbol').setScale(0.3).setAngle(-2).setDisplaySize(500,
600).setDepth(0).setInteractive(this.input.makePixelPerfect(100));
    const arbol2 = this.add.sprite(800, 200, 'arbol2').setScale(0.3).setAngle(-2).setDisplaySize(900,
800).setDepth(0).setInteractive(this.input.makePixelPerfect(100));
    const jirafa = this.add.sprite(300, 300, 'jirafa').setScale(0.3).setAngle(-2).setDisplaySize(500,
600).setDepth(2).setInteractive(this.input.makePixelPerfect(100));
    const piedra = this.add.sprite(956, 675,
'piedra').setScale(0.1).setAngle(0).setDepth(2).setInteractive(this.input.makePixelPerfect(100));
    const rinoceronte = this.add.sprite(120, 730, 'rinoceronte').setScale(0.3).setAngle(-5).setDisplaySize(700,
400).setDepth(2).setInteractive(this.input.makePixelPerfect(100));
    const elefante = this.add.sprite(660, 380, 'elefante').setScale(0.3).setAngle(0).setDisplaySize(700,
400).setDepth(2).setInteractive(this.input.makePixelPerfect(100));

    const playButton = this.add.sprite(512, 384,
'playButton').setInteractive().setDepth(5).setScale(0.3).setInteractive(this.input.makePixelPerfect(100));
    const gameOverImage = this.add.sprite(512, 150, 'gameOver').setDepth(5).setScale(0.4).setVisible(false);
    const restartButton = this.add.sprite(512, 480, 'playButton').setInteractive().setDepth(5).setDisplaySize(200,
200).setVisible(false);

```

Ilustración 15: Código con la creación de escena del juego.

5.5 MANEJO DE EVENTO DEL BOTON

Se definen los eventos para los botones de inicio y reinicio del juego, lo cuales se pueden observar en la ilustración 16. Estos eventos manejan la lógica para iniciar o reiniciar el juego y actualizar las variables de estado necesarias, eventos que aseguran que el juego se desarrolle de manera ordenada.

```

playButton.on('pointerdown', () => {
  if (!gameStarted.current) {
    gameStarted.current = true;
    startGame();
    playButton.destroy();
  }
});

restartButton.on('pointerdown', () => {
  if (gameEnded.current) {
    gameEnded.current = false;
    gameOverImage.setVisible(false);
    restartButton.setVisible(false);
    startGame();
  }
});

```

Ilustración 16: Código para manejo del botón.

5.6 DEFINICIÓN DE POSICIONES

Se La definición de posiciones en el videojuego es un aspecto crucial para garantizar una experiencia dinámica y desafiante para el jugador. En este juego, las posiciones de los lémures están predefinidas en varias áreas estratégicas del entorno, estando detrás del elefante, la piedra, el rinoceronte y la jirafa, tal como muestra la ilustración 17. Estas posiciones no solo están diseñadas para mantener al jugador alerta y enfocado, sino también para variar las apariciones de los lémures de manera que no sean completamente predecibles. Cada conjunto de posiciones incluye coordenadas específicas y otros atributos, como el ángulo y la escala, que determinan cómo y dónde aparecerán los lémures en la pantalla.

```

const elefantePositions = [
  {startX: 640, startY: 280, endX: 640, endY: 180, angle: 0, scale: 0.2},
  {startX: 540, startY: 300, endX: 500, endY: 195, angle: -22, scale: 0.2},
  {startX: 770, startY: 280, endX: 810, endY: 185, angle: 15, scale: 0.2},
  {startX: 780, startY: 377, endX: 870, endY: 377, angle: 90, scale: 0.2},
  {startX: 770, startY: 500, endX: 840, endY: 500, angle: 90, scale: 0.2},
  {startX: 600, startY: 500, endX: 524, endY: 500, angle: -90, scale: 0.2}
];

const piedraPositions = [
  {startX: 840, startY: 700, endX: 790, endY: 680, angle: -60, scale: 0.2},
  {startX: 860, startY: 650, endX: 825, endY: 605, angle: -45, scale: 0.2},
  {startX: 1000, startY: 550, endX: 970, endY: 490, angle: -45, scale: 0.2}
];

const rinocerontePositions = [
  {startX: 600, startY: 500, endX: 524, endY: 500, angle: -90, scale: 0.2},
  {startX: 50, startY: 632, endX: 50, endY: 530, angle: 0, scale: 0.2},
  {startX: 380, startY: 670, endX: 484, endY: 670, angle: 90, scale: 0.2}
];

const jirafaPositions = [
  {startX: 250, startY: 300, endX: 217, endY: 250, angle: -48, scale: 0.2},
  {startX: 275, startY: 330, endX: 180, endY: 330, angle: -85, scale: 0.2},
  {startX: 325, startY: 300, endX: 400, endY: 270, angle: 70, scale: 0.2}
];

```

Ilustración 17: Código de definición de posiciones.

5.7 CREACIÓN DE LÉMURES

En la creación de lémures, conforme se evidencia en la ilustración 18, se genera una serie de sprites que representan a estos animales en el juego. Estos lémures se crean inicialmente como objetos invisibles y desactivados, esperando ser activados conforme avanza el juego. Cada lémur es un objeto interactivo que puede detectar eventos de clic del usuario, lo que es esencial para la funcionalidad del juego. La implementación de eventos en cada lémur permite registrar el tiempo exacto en que son clickeados por el jugador, calculando así el tiempo de reacción desde su aparición hasta el momento del clic. Este proceso de

captura de tiempos de reacción es crucial para evaluar la agilidad mental del jugador y su capacidad de respuesta.

Además, se configuran diferentes posiciones predefinidas para los lémures detrás de los elementos visuales del juego como el elefante, la jirafa, el rinoceronte y la piedra. Estas posiciones son aleatorias dentro de los límites establecidos para garantizar que los lémures aparezcan en ubicaciones variadas, lo que aumenta el desafío y previene que el jugador anticipe los movimientos. Este sistema de aparición y reaparición constante de los lémures también implica un elemento de memoria espacial, ya que el jugador deberá recordar las posibles ubicaciones de los lémures a medida que el juego progresa. Esta variabilidad en las posiciones de los lémures no solo mantiene el juego interesante y desafiante, sino que también estimula la memoria y la capacidad de adaptación del jugador.

5.8 LÓGICA DEL JUEGO

La lógica del juego se basa en la creación de eventos temporizados que gestionan la aparición y movimiento de los lémures, así como en la captura de las interacciones del jugador. Una de las funciones principales es `createLemurEvent`, la cual se encarga de seleccionar un lémur disponible y una posición de aparición, evitando repeticiones inmediatas para mantener el desafío. Esta función utiliza arrays de posiciones que se eligen de manera pseudoaleatoria, asegurando que cada lémur tenga un recorrido y una ubicación específicos al aparecer y desaparecer en la pantalla.

La función `startGame` inicia el juego al configurar todos los eventos necesarios para la aparición de lémures y la actualización del temporizador del juego. Esto incluye establecer intervalos regulares para que los lémures aparezcan y desaparezcan, y ajustarlos dinámicamente para aumentar la dificultad conforme el tiempo avanza. La lógica de incremento de velocidad es fundamental para mantener el interés del jugador, ya que los

lémures empiezan a aparecer con mayor frecuencia y a una velocidad mayor, lo cual incrementa la dificultad y el nivel de desafío del juego.

```
this.lemurs = Array.from({ length: 15 }, () => {
  const lemur = this.add.sprite(0, 0, 'lemur').setInteractive().setScale(0.2);
  lemur.visible = false;
  lemur.setDepth(1);
  lemur.on('pointerdown', () => {
    if (!gameEnded.current) {
      const clickTime = Date.now();
      const appearanceTime = lemur.appearanceTime;
      const reactionTime = clickTime - appearanceTime;
      clickTimes.current.push(reactionTime);

      if (lastClickTime.current !== null) {
        const interval = clickTime - lastClickTime.current;
        clickIntervals.current.push(interval);
      }

      lastClickTime.current = clickTime;
      lemur.visible = false;
      lemur.active = false;
      lemursClicked.current += 1;
    }
  });
  return lemur;
});

const allPositions = {
  elefante: elefantePositions,
  piedra: piedraPositions,
  rinoceronte: rinocerontePositions,
  jirafa: jirafaPositions
};
```

Ilustración 18: Código para la creación de lémures.

Además, la función `endGame` se activa cuando el tiempo del juego ha finalizado. Esta función maneja la finalización del juego, asegurando que todos los lémures sean ocultados y registrando las estadísticas finales como el número de lémures clickeados y no

cliqueados, así como los tiempos de respuesta y los intervalos entre clics. Este registro es esencial para analizar el desempeño del jugador y para proporcionar datos cuantitativos que pueden ser utilizados para monitorear el progreso cognitivo del paciente a lo largo del tiempo. La recopilación y análisis de estos datos permite personalizar y ajustar las sesiones de juego para maximizar los beneficios terapéuticos, ofreciendo una experiencia de juego que se adapta a las necesidades específicas de cada jugador.

5.9 ALMACENAMIENTO DE LOS DATOS

MongoDB es una base de datos NoSQL altamente flexible y escalable, ideal para manejar grandes volúmenes de datos no estructurados o semi-estructurados. Para el desarrollo del sistema web de videojuegos cognitivos dirigido a adultos mayores, MongoDB se presenta como una opción adecuada por varias razones:

1. **Flexibilidad de Esquemas:** MongoDB permite una estructura de datos flexible, lo que es esencial para proyectos donde la naturaleza de los datos puede evolucionar con el tiempo. Esta característica es particularmente útil para aplicaciones de salud y bienestar donde los requisitos y los tipos de datos pueden cambiar.
2. **Escalabilidad:** MongoDB está diseñado para escalar horizontalmente. Esto significa que se pueden manejar grandes volúmenes de datos y tráfico de usuarios sin comprometer el rendimiento, lo cual es crucial para aplicaciones web con potencial de crecimiento.
3. **Consultas Rápidas y Eficientes:** Gracias a su modelo de documentos, MongoDB permite realizar consultas complejas y rápidas, facilitando el acceso a los datos críticos en tiempo real. Esto es fundamental para aplicaciones que necesitan

analizar y mostrar métricas rápidamente, como es el caso de nuestra plataforma de videojuegos cognitivos.

5.9.1 ESQUEMA USUARIO

- **Identificación Única (rut):** Cada usuario tiene un RUT único que sirve como identificador principal que a su vez permite la autenticación del usuario. Esto asegura que no haya duplicados y facilita el seguimiento individual de los usuarios.
- **Información de Contacto (email):** El campo de email es obligatorio y único, lo que permite la comunicación directa con el usuario.
- **Autenticación (password):** El campo de contraseña es necesario para la seguridad del usuario y la integridad del sistema.
- **Fecha de Creación (createdAt):** Este campo registra cuándo se creó el usuario, lo cual es útil para el seguimiento y auditoría.
- **Información Personal (firstName y lastName):** Estos campos son necesarios para identificar al usuario de manera personal y profesional.
- **Roles y Permisos (role):** El rol del usuario define su nivel de acceso y permisos dentro del sistema, asegurando que solo los usuarios autorizados puedan acceder a ciertas funciones.
- **Estado de Actividad (isActive):** Este campo permite activar o desactivar cuentas de usuario según sea necesario, facilitando la gestión de usuarios inactivos.
- **Diagnósticos (diagnostic):** Un array de diagnósticos proporciona flexibilidad para almacenar múltiples condiciones médicas. Además, no es un campo requerido dado que en muchas ocasiones puede suceder que el diagnóstico aún no se ha determinado.

```

import mongoose from 'mongoose'

const userSchema = new mongoose.Schema({
  rut: {
    type: String,
    required: true,
    unique: true
  },
  email: {
    type: String,
    required: true,
    unique: true
  },
  password: {
    type: String,
    required: true
  },
  createdAt: {
    type: Date,
    default: Date.now
  },
  firstName: {
    type: String,
    required: true
  },
  lastName: {
    type: String,
    required: true
  },
  role: {
    type: String,
    required: true,
    enum: ['Superadmin', 'Kinesiologo', 'Paciente', 'Neurologo']
  },
  isActive: {
    type: Boolean,
    default: true
  },
  diagnostic: {
    type: [String],
    default: []
  }
})

export const User = mongoose.model('User', userSchema)

```

Ilustración 19: Esquema usuario

5.9.2 ESQUEMA STATS

- **Total de Lémures (totalLemur):** Este campo registra la cantidad total de lémures que aparecieron durante el juego, lo que es crucial para evaluar la participación del paciente.
- **Lémures Clickeados (lemurClicked):** Este campo contabiliza cuántos lémures fueron clickeados por el paciente, proporcionando una métrica directa de su desempeño.
- **Lémures No Clickeados (lemurNotClicked):** Este campo cuenta los lémures que no fueron clickeados, lo que ayuda a identificar posibles dificultades en la respuesta del paciente.
- **Tiempos de No Clickeados (nonClickedTime):** Un array que registra los tiempos en los que los lémures no fueron clickeados, ofreciendo una visión detallada de los lapsos de inactividad.
- **Tiempos de Clickeados (clickedTime):** Un array que registra los tiempos de reacción de los clics, proporcionando datos sobre la rapidez y agilidad del paciente.
- **Intervalos entre Clics (clickedIntervals):** Este array almacena los intervalos de tiempo entre cada clic, permitiendo el análisis de la consistencia y velocidad de respuesta.
- **Fecha de Creación (createdAt):** Registra cuándo se generaron las estadísticas, lo cual es útil para el seguimiento temporal y la evolución del paciente.
- **Identificación del Usuario (userRut):** Relaciona las estadísticas con el usuario correspondiente, asegurando que cada conjunto de datos esté vinculado a un paciente específico.

```

import mongoose from 'mongoose'

const statsSchema = new mongoose.Schema({
  totalLemur: {
    type: Number,
    required: true
  },
  lemurClicked: {
    type: Number,
    required: true
  },
  lemurNotClicked: {
    type: Number,
    required: true
  },
  nonClickedTime: {
    type: Array,
    required: true
  },
  clickedTime: {
    type: Array,
    required: false
  },
  clickedIntervals: {
    type: Array,
    required: false
  },
  createdAt: {
    type: Date,
    default: Date.now
  },
  userRut: {
    type: String,
    required: true
  }
})

export const Stats = mongoose.model('Stats', statsSchema)

```

Ilustración 20: Esquema stats.

CAPÍTULO VI: PRUEBAS

Las pruebas son una fase crucial en el desarrollo de software, ya que aseguran que la aplicación cumpla con los requisitos especificados y funcione de manera correcta y eficiente. En este capítulo, se describen las pruebas realizadas en la plataforma "CogniGame" para validar su funcionalidad, rendimiento y usabilidad.

6.1 TIPOS DE PRUEBAS

6.1.1 PRUEBAS UNITARIAS

Descripción: Estas pruebas se centran en verificar el correcto funcionamiento de las unidades más pequeñas del código, como funciones y métodos individuales. Ejemplo de Prueba: Validación de la función de cálculo de puntuaciones en los juegos cognitivos.

6.1.2 PRUEBAS DE INTEGRACIÓN

Descripción: Estas pruebas validan la interacción entre diferentes módulos y componentes del sistema. para pruebas de interfaz, Postman para pruebas de API. Ejemplo de Prueba: Verificación de la correcta comunicación entre el módulo de autenticación y el módulo de perfil de usuario.

6.1.3 PRUEBAS FUNCIONALES

Descripción: Estas pruebas aseguran que la funcionalidad del sistema cumpla con los requisitos especificados. Ejemplo de Prueba: Comprobación de que los juegos se adaptan al nivel de habilidad del paciente.

6.1.4 PRUEBAS DE USABILIDAD

Descripción: Estas pruebas evalúan la facilidad de uso de la plataforma desde la perspectiva del usuario final. Método: Ejemplo de Prueba: Evaluación de la de la interfaz de usuario para pacientes mayores.

6.1.5 PRUEBAS DE RENDIMIENTO

Descripción: Estas pruebas miden cómo se comporta el sistema bajo diferentes condiciones de carga. Ejemplo de Prueba: Evaluación del tiempo de respuesta del sistema cuando múltiples usuarios acceden simultáneamente.

6.2. PASOS DEL PLAN

- Definición del Alcance
- Identificar las funcionalidades críticas del sistema.
- Determinar los escenarios de uso más comunes y críticos.
- Diseño de Casos de Prueba
- Crear casos de prueba detallados que cubran todas las funcionalidades y escenarios identificados.
- Incluir criterios de aceptación claros para cada caso de prueba.
- Ejecución de Pruebas: Realizar las pruebas unitarias e integración durante la fase de desarrollo.
- Repetición de Pruebas: Volver a ejecutar pruebas después de la corrección de defectos para asegurar que los problemas se hayan resuelto.

6.3 RESULTADOS DE LAS PRUEBAS

- **Pruebas Unitarias:** Se lograron cubrir el 95% de las funciones críticas con pruebas unitarias exitosas. Se encontraron y corrigieron varios errores menores relacionados con cálculos de puntuación.

- **Pruebas de Integración:** Resultado: Todas las pruebas de integración fueron exitosas, confirmando la correcta interacción entre módulos. Se mejoró la documentación de la API para facilitar la integración.
- **Pruebas Funcionales:** Las funcionalidades clave de la plataforma cumplieron con los requisitos especificados. Se realizaron ajustes menores en la lógica de adaptación de los juegos.
- **Pruebas de Usabilidad:** Los usuarios finales encontraron la interfaz intuitiva y fácil de usar. Se identificaron mejoras en la navegación, que fueron implementadas.

CAPÍTULO VII: CONCLUSIONES

El desarrollo y la implementación de la plataforma CogniGame representa un avance significativo en el campo de la estimulación cognitiva para adultos mayores, cumpliendo con los objetivos generales y específicos planteados al inicio de esta tesis. La creación de una plataforma de videojuegos con enfoque cognitivo, diseñada para capturar y utilizar de manera efectiva los datos del adulto mayor durante el juego, no solo es innovadora sino también esencial para el tratamiento y monitoreo de este grupo etario.

7.1 CUMPLIMIENTO DEL OBJETIVO GENERAL

CogniGame ha sido diseñado para abordar el deterioro cognitivo en adultos mayores, utilizando videojuegos como una herramienta terapéutica. La plataforma no solo proporciona una forma entretenida y estimulante de ejercitar el cerebro, sino que también captura datos valiosos durante el juego. Estos datos incluyen tiempos de reacción, intervalos entre clics y patrones de interacción, los cuales son fundamentales para evaluar el progreso cognitivo del paciente. Al integrar estos datos en un sistema accesible para médicos y especialistas, CogniGame facilita una intervención más personalizada y efectiva, optimizando el tratamiento de cada paciente.

1. Desarrollo del Sistema Web con Distintas Interfaces

Se ha desarrollado un sistema web robusto con interfaces específicas para el login, vista del paciente y vista del médico especialista. Estas interfaces han sido diseñadas para ser intuitivas y fáciles de usar, permitiendo a los usuarios navegar y acceder a la información de manera eficiente. La interfaz del paciente ofrece acceso directo al juego y muestra los resultados y métricas relevantes, mientras que la interfaz del médico permite un análisis detallado de los datos capturados, facilitando el seguimiento y la evaluación del progreso cognitivo del paciente.

2. Creación del Juego

El juego desarrollado dentro de CogniGame ha sido diseñado específicamente para estimular funciones cognitivas clave como la memoria, la atención y la flexibilidad cognitiva. Utilizando tecnologías avanzadas como Phaser y JavaScript, se ha creado un entorno interactivo y atractivo que mantiene a los pacientes comprometidos y motivados. Además, la utilización de inteligencia artificial para la generación de imágenes del juego asegura un diseño único y original, que enriquece la experiencia del usuario.

3. Desarrollo de la Base de Datos

Una base de datos sólida y bien estructurada ha sido implementada para almacenar tanto la información de los usuarios como los datos capturados durante el juego. Utilizando MongoDB, una base de datos NoSQL, se ha asegurado la escalabilidad y flexibilidad necesarias para manejar grandes volúmenes de datos de manera eficiente. La capacidad de almacenar datos detallados sobre el desempeño de los pacientes permite a los médicos realizar un seguimiento preciso y tomar decisiones informadas sobre el tratamiento.

7.2 REFUTACIÓN BASADA EN ESTUDIOS TEÓRICOS

La importancia y efectividad de CogniGame están respaldadas por diversos estudios teóricos que destacan los beneficios de los videojuegos en la estimulación cognitiva. Según la investigación de Irazoki et al. (2020), los videojuegos pueden mejorar varias habilidades cognitivas, incluida la memoria, la atención y el procesamiento visual-espacial. Fernández-Calvo et al. (2011) resalta la capacidad de los videojuegos para mejorar la flexibilidad cognitiva y la función ejecutiva, aspectos fundamentales en la preservación de la cognición a lo largo del envejecimiento. Además, Pazán Jurado (2015) y Moscardi (2018) subrayan la importancia de los videojuegos como herramientas de

intervención terapéutica, destacando su utilidad para mejorar la salud mental y la cognición en adultos mayores.

7.3 BENEFICIOS PARA PACIENTES Y ESPECIALISTAS

COGniGame ofrece múltiples beneficios tanto para los pacientes como para los especialistas. Para los adultos mayores, el juego proporciona una forma divertida y efectiva de mantener su mente activa, ayudando a prevenir el deterioro cognitivo y mejorando su calidad de vida. Los especialistas, por su parte, pueden acceder a datos detallados y precisos sobre el rendimiento cognitivo de sus pacientes, lo que les permite personalizar los tratamientos y realizar un seguimiento más efectivo del progreso.

7.4 IMPORTANCIA DEL SISTEMA CREADO

El sistema CogniGame no solo representa una herramienta terapéutica innovadora, sino que también marca un avance en la integración de tecnologías en el cuidado de la salud. La captura y análisis de datos en tiempo real, la creación de un entorno de juego atractivo y la facilidad de acceso a la información para los especialistas son elementos que diferencian a CogniGame y demuestran su valor en el tratamiento de problemas cognitivos en adultos mayores.

En resumen, CogniGame cumple con los objetivos propuestos al inicio de esta tesis, proporcionando una plataforma efectiva y accesible para la estimulación cognitiva de adultos mayores. Su desarrollo y aplicación no solo benefician a los pacientes al ofrecer una herramienta terapéutica divertida y efectiva, sino que también facilitan el trabajo de los especialistas del departamento de geriatría del Hospital de Limache al proporcionar datos precisos y útiles para el tratamiento personalizado.

REFERENCIAS

BIBLIOGRAFÍA

- Fernández-Calvo, B., Rodríguez-Pérez, R., Contador, I., Rubio-Santorum, A., & Ramos Campos, F. (2011). Efficacy of cognitive training programs based on new software technologies in patients with Alzheimer- Type dementia.
- Irazoki, E., Van der Roest, H., Contreras-Somoza, L. M., Toribio-Guzmán, J. M., Jenaro-Río, C., & Franco-Martín, M. A. (2020). Technologies for Cognitive Training and Cognitive Rehabilitation for People With Mild Cognitive Impairment and Dementia. A Systematic Review.
- Moscardi, R. (2018). VIDEOJUEGOS Y HABILIDADES COGNITIVAS. Buenos Aires: Argentina.
- Pazán Jurado, L. R. (2015). Videojuegos de entrenamiento cerebral como factores de estimulación de la esfera cognitiva en adultos mayores en el centro geriátrico sagrado corazon de jesús.

LINKOGRAFÍA

- Braingymmer, << braingymmer.com - Sobre nosotros >> 2024. [En Línea]. Disponible en: <https://www.braingymmer.com/es/content/sobre-braingymmer/>
- CogniFit, << cognifit.com - ¿Qué es CogniFit? >> 2024. [En Línea]. Disponible en: <https://www.cognifit.com/cl/que-es-cognifit>

- NeuronUP, << neuronup.com – Sobre nosotros >> 2024. [En Línea]. Disponible en: <https://neuronup.com/sobre-nosotros/>
- JavaScript, << amazon.com/ - ¿Qué es JavaScript? >> 2024. [En Línea]. Disponible en: <https://aws.amazon.com/es/what-is/javascript/>
- React, << hostinger.es - ¿Qué es React? >> 2024. [En Línea]. Disponible en: <https://www.hostinger.es/tutoriales/que-es-react>
- MongoDB, << mongodb.com – ¿Qué es MongoDB? >> 2024. [En Línea]. Disponible en: <https://www.mongodb.com/es/company/what-is-mongodb>
- Phaser, << phaser.io - ¿Qué es Phaser? >> 2024. [En Línea]. Disponible en: <https://phaser.io/tutorials/making-your-first-phaser-3-game-spanish>
- Github, << hostinger.es - ¿Qué es GitHub? >> 2024. [En Línea]. Disponible en: <https://www.hostinger.es/tutoriales/que-es-github>
- Cascada, << safetyculture.com - ¿Qué es la metodología en Cascada? >> 2024. [En Línea]. Disponible en: <https://safetyculture.com/es/temas/metodologia-en-cascada/>
- Modelo MVC, << desarrolloweb.com – Modelo MVC >> 2024. [EN Línea]. Disponible en: <https://desarrolloweb.com/articulos/que-es-mvc.html/>