



Universidad Politécnica
de Madrid



**Escuela Técnica Superior de
Ingenieros Informáticos**

Grado en Ingeniería Informática

Trabajo Fin de Grado

**Creación de una Biblioteca de Test
Cognitivos Online Optimizados para
Dispositivos Móviles**

Autor: Santiago Federico Caballero Pastor

Tutor(a): Raúl Alonso Calvo

Madrid, junio 2023

Este Trabajo Fin de Grado se ha depositado en la ETSI Informáticos de la Universidad Politécnica de Madrid para su defensa.

Trabajo Fin de Grado

Grado en Ingeniería Informática

Título: Creación de una Biblioteca de Test Cognitivos Online Optimizados para Dispositivos Móviles

Junio 2023

Autor: Santiago Federico Caballero Pastor

Tutor:

Raúl Alonso Calvo

Lenguajes y Sistemas Informáticos e Ingeniería del Software

ETSI Informáticos

Universidad Politécnica de Madrid

Resumen

En este Proyecto se pretende crear una batería de pruebas cognitivas para evaluar a pacientes con posibles deterioros cognitivos. Se trata de una aplicación web optimizada para dispositivos móviles con un total de siete pruebas distintas.

En la aplicación, el usuario podrá inscribir sus datos y elegir una de las pruebas. Estas pruebas, guardan en un servidor REST una serie de parámetros para el análisis posterior de algún profesional médico. Los parámetros se guardan con el fin de simular la interacción del usuario con la aplicación y para medir las distintas capacidades cognitivas del individuo: la memoria, la capacidad de cómputo, y la coordinación ojo-mano entre otras.

Dichos tests están compuestos por una prueba de memorización de imágenes, al que le sigue una de arrastrar un objeto en su correspondiente lugar, después se memorizan unas palabras, hay un cómputo matemático, continua con la pulsación de un círculo, sigue con la detección de una bola hasta llegar al final de un trayecto, y por último una implementación propia de un test MOCA.

Esta memoria pretende recolectar todo lo relacionado con el desarrollo de este trabajo. Se comenzará con una introducción, en la que se presentarán las principales motivaciones y objetivos. Se continuará analizando aplicaciones similares en la literatura, así como las herramientas que se han usado en el proyecto.

Después se tratará todo lo relacionado con la implementación de la aplicación. Comenzando con los requisitos y casos de uso de cada prueba y finalizando con el diseño, interfaz, uso y almacenamiento de parámetros de cada prueba.

Para finalizar, se analizarán los resultados obtenidos tras el trabajo y se hará un estudio del impacto de este proyecto.

Abstract

The aim of this project is to create a series of cognitive tests to evaluate patients with possible cognitive impairment. It is a web application optimized for mobile devices with a total of seven different tests.

In the application, the user will be able to register his data and choose one of the tests. These tests store a series of parameters on an REST server for future analysis by a medical professional. Parameters are stored to simulate the user's interaction with the application and measure the different cognitive capacities of the individual: memory, computational capacity, and hand-eye coordination, among others.

The tests are composed of an image memorization test, followed by a test of dragging an object to its corresponding place, then memorizing some words, next a mathematical computation, continued by pressing a circle, then detecting a ball until reaching the end of a path, and finally an implementation of a MOCA test.

This report aims to collect everything related to the development of this work. It will begin with an introduction, in which the main motivations and objectives will be presented. It will continue by analyzing similar applications in the literature, as well as the tools that have been used in the project.

After that, everything related to the implementation of the application will be discussed. Starting with the requirements and use cases of each test and ending with the design, interface, use, and storage of parameters of each test.

Finally, the results obtained after the work will be analyzed and a study of the impact of this project will be made.

Tabla de contenidos

1	Introducción.....	1
1.1	Motivación.....	1
1.2	Objetivos	2
2	Estado de la cuestión.....	3
2.1	Aplicaciones Similares	3
2.1.1	Tests Cognitivos	3
2.1.1.1	Mini-Cog	3
2.1.1.2	MMSE.....	3
2.1.1.3	MOCA	4
2.1.2	Cognifit	5
2.1.3	CANTAB.....	7
2.1.4	CoGNIT	9
2.2	Tecnologías utilizadas	10
2.2.1	HTML5	10
2.2.2	CSS.....	11
2.2.3	JAVASCRIPT	11
2.2.4	API REST	12
2.2.5	Tecnologías Móviles.....	12
2.2.6	Multiplataforma Móviles.....	13
2.2.7	Visual Studio Code.....	13
2.2.8	IntelliJ IDEA Community Edition	14
2.2.9	Draw.io	14
3	Requisitos.....	15
3.1	Prueba 1, Parejas	15
3.1.1	Requisitos Funcionales	15
3.1.2	Casos de Uso	16
3.2	Prueba 2, Arrastra Y Suelta.....	16
3.2.1	Requisitos Funcionales	16
3.2.2	Casos de Uso	17
3.3	Prueba 3, Recuerda Palabras.....	17
3.3.1	Requisitos Funcionales	18
3.3.2	Casos de Uso	18
3.4	Prueba 4, Noquea Las Torres.....	19
3.4.1	Requisitos Funcionales	19
3.4.2	Casos de Uso	20
3.5	Prueba 5, CentroBola	21
3.5.1	Requisitos Funcionales	21
3.5.2	Casos de Uso	22

3.6	Prueba 6, MetaBola	22
3.6.1	Requisitos Funcionales	22
3.6.2	Casos de Uso	23
3.7	Prueba 7, MOCA.....	24
3.7.1	Requisitos Funcionales	24
3.7.2	Casos de Uso	26
4	Desarrollo	30
4.1	Prueba 1, Parejas	30
4.1.1	Diseño	30
4.1.2	Interfaz y Uso.....	31
4.1.3	Almacenamiento de Datos.....	33
4.2	Prueba 2, Arrastra y Suelta	34
4.2.1	Diseño	34
4.2.2	Interfaz y Uso.....	35
4.2.3	Almacenamiento de Datos.....	37
4.3	Prueba 3, Recuerda Palabras.....	38
4.3.1	Diseño	38
4.3.2	Interfaz y Uso.....	40
4.3.3	Almacenamiento de Datos.....	42
4.4	Prueba 4, Noquea las Torres.....	43
4.4.1	Diseño	43
4.4.2	Interfaz y Uso.....	44
4.4.3	Almacenamiento de Datos.....	46
4.5	Prueba 5, CentroBola	47
4.5.1	Diseño	47
4.5.2	Interfaz y Uso.....	48
4.5.3	Almacenamiento de Datos.....	50
4.6	Prueba 6, MetaBola	51
4.6.1	Diseño	51
4.6.2	Interfaz y Uso.....	52
4.6.3	Almacenamiento de Datos.....	54
4.7	Prueba 7, MOCA.....	55
4.7.1	Diseño	55
4.7.2	Interfaz y Uso.....	59
4.7.3	Almacenamiento de Datos.....	66
4.8	Menú Principal	66
4.8.1	Diseño	66
4.8.2	Interfaz y Uso.....	67
4.8.3	Almacenamiento de Datos.....	68

5	Resultados y conclusiones	70
6	Análisis de Impacto	73
7	Bibliografía	74

Tabla de Ilustraciones

Ilustración 1: Mini-Cog	3
Ilustración 2: MMSE	4
Ilustración 3: MOCA	5
Ilustración 4: Logo Cognifit	5
Ilustración 5: Logo CANTAB	7
Ilustración 6: Logo HTML5	11
Ilustración 7: Logo CSS	11
Ilustración 8: Logo JavaScript	11
Ilustración 9: REST	12
Ilustración 10: Logo IONIC	13
Ilustración 11: Logo Visual Studio Code	14
Ilustración 12: Logo IntelliJ Idea	14
Ilustración 13: Logo Draw.io	14
Ilustración 14: Casos de Uso Parejas	16
Ilustración 15: Casos de Uso Arrastra Y Suelta	17
Ilustración 16: Casos de Uso Recuerda Palabras	19
Ilustración 17: Casos de Uso Noquea Las Torres	21
Ilustración 18: Casos de Uso CentroBola	22
Ilustración 19: Casos de Uso MetaBola	23
Ilustración 20: Casos de Uso MOCA Sección 1	26
Ilustración 21: Casos de Uso MOCA Sección 2	26
Ilustración 22: Casos de Uso MOCA Sección 3	27
Ilustración 23: Casos de Uso MOCA Sección 4	27
Ilustración 24: Casos de Uso MOCA Sección 5	28
Ilustración 25: Casos de Uso MOCA Sección 6	28
Ilustración 26: Casos de Uso MOCA Sección 7	29
Ilustración 27: Diagramas Parejas	30
Ilustración 28: Interfaz Parejas	32
Ilustración 29: Usuario selecciona dos cartas distintas	32
Ilustración 30: Usuario seleccionada una pareja	33
Ilustración 31: Usuario finaliza la prueba Parejas en la dificultad 2	33
Ilustración 32: Parámetros Parejas	33
Ilustración 33: Diagramas Arrastra Y Suelta	34
Ilustración 34: Interfaz Arrastra Y Suelta	35
Ilustración 35: Usuario arrastra la imagen de la naranja en su caja	36
Ilustración 36: Usuario completa el primer nivel de Arrastra Y Suelta	36
Ilustración 37: Interfaz del segundo nivel de Arrastra Y Suelta	37
Ilustración 38: Parámetros Arrastra Y Suelta	38
Ilustración 39: Diagramas Recuerda Palabras	39
Ilustración 40: Interfaz Recuerda Palabras	40
Ilustración 41: Interfaz Recuerda Palabras con 8 palabras a memorizar, y segundos	40
Ilustración 42: Recuerda Palabras en ejecución con 8 palabras	41
Ilustración 43: Tablero de las palabras a encontrar	41
Ilustración 44: Finalización de Recuerda Palabras con la puntuación	42
Ilustración 45: Parámetros Recuerda Palabras	42
Ilustración 46: Diagramas Noquea Las Torres	43
Ilustración 47: Interfaz Noquea Las Torres	44
Ilustración 48: Enemigo derrotado	45
Ilustración 49: Jugador Derrotado, Prueba Finalizada	45
Ilustración 50: Cuarta Torre	46
Ilustración 51: Finalización de Noquea Las Torres	46

Ilustración 52: Parámetros Noquea Las Torres	47
Ilustración 53: Diagramas CentroBola	48
Ilustración 54: Interfaz CentroBola	49
Ilustración 55: Bola aparece aleatoriamente.....	49
Ilustración 56: Usuario pulsa cerca del centro de la bola	50
Ilustración 57: Finalización CentroBola	50
Ilustración 58: Parámetros CentroBola	51
Ilustración 59: Diagramas MetaBola	52
Ilustración 60: Interfaz MetaBola	53
Ilustración 61: Nivel 1de MetaBola.....	53
Ilustración 62: Nivel 2 de MetaBola.....	54
Ilustración 63: Nivel 3 de MetaBola con bola en la meta.....	54
Ilustración 64: Usuario finaliza la prueba en la dificultad 2	55
Ilustración 65: Diagramas MOCA Parte 1	56
Ilustración 66: Diagramas MOCA Parte 2	56
Ilustración 67: Diagramas MOCA Parte 3	57
Ilustración 68: Diagramas MOCA Parte 4	57
Ilustración 69: Diagramas MOCA Parte 5	58
Ilustración 70: Diagramas MOCA Parte 6	59
Ilustración 71: Diagramas MOCA Parte 6	59
Ilustración 72: Interfaz MOCA Parte 1	60
Ilustración 73: Interfaz MOCA Parte 2	60
Ilustración 74: Interfaz MOCA Parte 3	61
Ilustración 75: Interfaz MOCA Parte 4	61
Ilustración 76: Interfaz MOCA Parte 5	61
Ilustración 77: Interfaz MOCA Parte 6	62
Ilustración 78: Interfaz MOCA Parte 7	62
Ilustración 79: Dibujo realizado en MOCA Parte 1.....	63
Ilustración 80: Dibujo realizado en MOCA Parte 2.....	63
Ilustración 81: Identificación de los animales MOCA Parte 3.....	64
Ilustración 82: Temporizador tras primera grabación MOCA Parte 4	64
Ilustración 83: Finalización de MOCA Parte 5	65
Ilustración 84: Finalización de MOCA Parte 6	65
Ilustración 85: Finalización con pista de MOCA Parte 7	66
Ilustración 86: Diagramas Menú Principal	67
Ilustración 87: Interfaz Menú, Registro de Datos.....	68
Ilustración 88: Interfaz Menú Principal	68
Ilustración 89: Parámetros Menú Principal	69

1 Introducción

¿Qué son las pruebas cognitivas? Las pruebas cognitivas son pruebas para detectar si una persona sufre un problema de cognición. ¿Qué entendemos por cognición? La cognición es un conjunto de procesos que intervienen en el pensamiento, la memoria, el lenguaje, el juicio y la capacidad de aprender. A los problemas en el desarrollo de estos procesos se los denomina deterioros cognitivos, variando entre leves y graves. [1]

Estos deterioros son originados por múltiples causas como, por ejemplo, efectos secundarios de medicamentos, enfermedades en vasos sanguíneos, depresión y demencia (pérdida grave de una función mental). Destaca entre estas enfermedades el Alzheimer, una demencia que reduce la capacidad de pensar, la memoria, y la habilidad de realizar tareas sencillas con el tiempo. [1]

Las pruebas cognitivas no indican la causa específica del deterioro, pero sirven a los profesionales como punto de partida para realizar más pruebas médicas y responder al problema. Por ello, estas pruebas sirven para detectar un deterioro cognitivo leve, cambios en memoria y otras funcionalidades, pero son cambios que no tienen un efecto importante en la vida diaria. Aun así, puede conllevar a un deterioro más grave, por lo que los médicos podrían realizar más pruebas para controlar el deterioro. [1]

Hay diferentes tipos de pruebas cognitivas, cada una con un formato, preguntas y puntuación distinto, pero en esencia, consisten en pruebas sencillas diseñadas para medir funcionalidades mentales. Cuanto más puntuación se tenga en las pruebas, menos indicios de deterioro cognitivo existen en el paciente. Las tres pruebas cognitivas más famosas son el Mini-Cog (prueba rápida, fácil de usar y ampliamente disponible), el MMSE (que es usada para encontrar problemas cognitivos más serios) y el MOCA (la prueba ideal para encontrar deterioros cognitivos leves). [1]

1.1 Motivación

Tome la decisión de realizar este proyecto pensando en crear una aplicación atractiva y funcional en formato móvil. A partir de ahí, fui desarrollando la idea hasta llegar a la idea actual, la creación de una serie de test cognitivos online optimizados para dispositivos móviles. Además, este proyecto me daba la oportunidad de indagar en el tema de deterioro cognitivo y así poder ayudar a personas que tienen este tipo de problema.

Partiendo del estudio “Alimentos ricos en polifenoles como estrategia de envejecimiento saludable y sostenible. Kephenol” del CSIC comencé el proyecto. En dicho estudio se necesitan voluntarios mayores de 50 años y con índice de masa corporal entre 18 y 32 kg/m², para observar sus hábitos alimenticios y como el consumo de polifenoles afecta a su salud neuropsicológica y cerebrovascular. [2]

La tesis “Estimación de la ingesta de polifenoles y contaminantes alimentarios en diferentes poblaciones: Mujeres embarazadas y niños” también ha sido fuente de inspiración para poder desarrollar el proyecto. En dicha tesis doctoral, se parte de la idea de la lucha contra la obesidad y la desnutrición sobre todo en mujeres embarazadas y niños. Por lo que se buscó herramientas fiables como

registros dietéticos, bases de datos de composición de alimentos o programas informáticos para la ayuda de una nutrición más personalizada. Por ello, el objetivo de la tesis era crear y validar diferentes herramientas dietéticas para poder estimar la ingesta de polifenoles o la concentración de contaminantes procedentes de los alimentos, especialmente en mujeres embarazadas y niños. [3]

La tesis concluye con un desarrollo de herramientas fiables para el cálculo de ingesta de contaminantes y polifenoles en diferentes poblaciones, con una base de datos "CONT11" y "S4H" desarrolladas que permitieron realizar las estimaciones de una manera más eficiente. Además, se encontraron relaciones entre un consumo elevado de contaminantes y un aumento de peso o deterioro de la salud. También, se demostró que un consumo mayor de polifenoles se asocia a un mejor estado de salud como un índice de masa corporal materno menor. [3]

1.2 Objetivos

En este proyecto se creará una aplicación web adaptada a un dispositivo móvil. En ella, se pretende insertar distintas pruebas para poder evaluar posibles problemas cognitivos en pacientes con determinadas características. Para ello, cada persona podrá hacer estos test tanto en un navegador web como en un dispositivo móvil, y durante la ejecución de estas pruebas, se guardarán diferentes tipos de parámetros, dependiendo de la prueba, para que los evaluadores puedan obtenerlos y usarlos debidamente en su diagnóstico.

De este modo se podrá relacionar la calidad de la dieta de los distintos pacientes con respecto a su capacidad cognitiva. Asimismo, evaluar cómo afectan los polifenoles en pacientes con algún tipo de deterioro, y llevar un estudio de la evolución de los pacientes con la dieta.

El proyecto se dividirá en dos partes. Por un lado, una batería de pruebas en la que se usará la combinación de HTML5, CSS y JavaScript y así desarrollar unas pruebas dedicadas a los distintos navegadores. Posteriormente, para adaptar la aplicación a dispositivos móviles, se hará uso de tecnologías multiplataforma. Por otro lado, se realizará un servicio web de tipo REST para poder recoger los distintos datos obtenidos de la interfaz del usuario en las pruebas.

Cada prueba es un subobjetivo distinto, compuesto por tareas, pero con un mismo objetivo final, evaluar la cognición del individuo. Se desarrollarán pruebas centradas en la memoria, tiempo de reacción, percepción auditiva y visual, tiempo de respuesta, coordinación y velocidad de procesamiento. Además, las tareas deben ser sencillas y fáciles de entender y manejar aptas para cualquier público sin importar el dispositivo que se use.

Con todo ello, el objetivo final es desarrollar una aplicación web fiable que les sirva a los profesionales como herramienta para evaluar el estado cognitivo de sus pacientes. Por lo tanto, se pretende también llevar un plan de pruebas para garantizar la máxima calidad posible.

2 Estado de la cuestión

2.1 Aplicaciones Similares

En este apartado se mencionan algunas aplicaciones similares a la que se ha realizado en este proyecto. Aplicaciones que sirven para la evaluación cognitiva de una persona.

2.1.1 Tests Cognitivos

A continuación, veremos en qué consisten estas pruebas con más detalle.

2.1.1.1 Mini-Cog

El Mini-Cog es una prueba de detección rápida para la enfermedad del Alzheimer con una duración de tres minutos. Es una herramienta de cribado cognitivo desarrollada por el Dr. Soo Borson que mide dos aspectos: recuerdo a corto plazo y el dibujo de un reloj. Está compuesta de tres secciones: la primera de ellas es repetir tres palabras no relacionadas, la segunda dibujar un reloj con sus números y manecillas, y la tercera es recordar las tres palabras de la primera sección. De un total de 5 puntos, una puntuación de cero, uno o dos, indica una preocupación en el funcionamiento cognitivo. [4]

Mini-Cog®		Instrucciones de administración y puntuación	Dibujo del reloj		
		Id.: _____ Fecha: _____	Id.: _____ Fecha: _____		
Paso 1: Registro de tres palabras					
Mire directamente a la persona y diga: "Escuche con atención. Voy a decirle tres palabras y quiero que las repita ahora e intente recordarlas. Las palabras son [seleccione una lista de palabras de las versiones de abajo]. ¿Podría repetirlas ahora?". Si la persona no puede repetir las palabras después de tres intentos, avance al paso 2 (dibujo del reloj).					
Estas listas de palabras, y otras más, se han usado en uno o más estudios clínicos. ^{1,3} Si se repite el ejercicio, se recomienda usar una lista de palabras alternativa.					
Versión 1	Versión 2	Versión 3	Versión 4	Versión 5	Versión 6
Banana	Líder	Aldea	Río	Capitán	Hija
Amanecer	Temporada	Cocina	Nación	Jardín	Cielo
Silla	Mesa	Bebé	Dedo	Foto	Montaña
Paso 2: Dibujo del reloj					
Diga: "Ahora quiero que dibuje un reloj. Primero, ponga todos los números en su lugar correspondiente". Cuando termine, diga: "Ahora dibuje las agujas para indicar que son las 11:10".					
Use el círculo preimpreso (consulte la página siguiente) para este ejercicio. Repita las instrucciones según sea necesario, ya que esta no es una prueba de memoria. Avance al paso 3 si no se completa el reloj en tres minutos.					
Paso 3: Recordar tres palabras					
Pídale a la persona que repita las palabras que usted le dijo en el paso 1. Diga: "¿Cuáles son las tres palabras que le pedí que recordara?". Anote abajo el número de versión de la lista de palabras y las respuestas de la persona.					

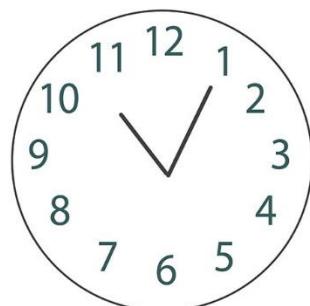


Ilustración 1: Mini-Cog

El Mini-Cog, a diferencia del MMSE, no está influenciado por el nivel educativo o por ninguna habilidad lingüística. En esta prueba destaca la precisión para identificar a personas que padecen la enfermedad de Alzheimer. [4]

2.1.1.2 MMSE

La prueba Mini Mental de Folstein, MMSE, se publicó en el año 1975 por Marshal F. Folstein, Susan Folstein y Paul R. McHung, con el objetivo de crear una prueba estandarizada para diferenciar los trastornos presentes en los pacientes de psiquiátricos. Asimismo, esta prueba cuenta con una adaptación

para la población española, realizada por Lobo en 1975, además de algunas pruebas más de evaluación. [5]

El MMSE permite detectar una demencia en pacientes psiquiátricos utilizando cribados sistemáticos. A su vez, se puede usar para seguir un seguimiento de la evolución del deterioro cognitivo de un paciente con demencia.

Mini-Mental State Examination (MMSE)

Patient's Name: _____ Date: _____

Instructions: Ask the questions in the order listed.
Score one point for each correct response within each question or activity.

Maximum Score	Patient's Score	Questions
5		"What is the year? Season? Date? Day of the week? Month?"
5		"Where are we now? State? County? Town/city? Hospital? Floor?"
3		The examiner names three unrelated objects clearly and slowly, then asks the patient to name all three of them. The patient's response is used for scoring. The examiner repeats them until patient learns all of them, if possible. Number of trials: _____
5		"I would like you to count backward from 100 by sevens." (93, 86, 79, 72, 65,...) Stop after five answers. Alternative: "Spell WORLD backwards." (D-L-R-O-W)
3		"Earlier I told you the names of three things. Can you tell me what those were?"
2		Show the patient two simple objects, such as a wristwatch and a pencil, and ask the patient to name them.
1		"Repeat the phrase: 'No ifs, ands, or buts.'"
3		"Take the paper in your right hand, fold it in half, and put it on the floor." (The examiner gives the patient a piece of blank paper.)
1		"Please read this and do what it says." (Written instruction is "Close your eyes.")
1		"Make up and write a sentence about anything." (This sentence must contain a noun and a verb.)
1		"Please copy this picture." (The examiner gives the patient a blank piece of paper and asks him/her to draw the symbol below. All 10 angles must be present and two must intersect.)
30		TOTAL

Ilustración 2: MMSE

Esta prueba contiene preguntas sobre la orientación espacio temporal, la atención, memoria, concentración, cálculo matemático, lenguaje, percepción viso espacial y seguir instrucciones básicas. En cada sección se reciben una serie de puntos con un máximo de 30 en toda la prueba. Si se tiene entre 30 y 27 puntos, no existe deterioro cognitivo, entre 26 y 25 puntos, existen dudas sobre un posible deterioro, entre 24 y 10 puntos, existe un deterioro cognitivo de leve a moderado, entre 9 y 6 puntos, existe un deterioro cognitivo de moderado a severo, y si se obtienen menos de 6 puntos, existe un deterioro cognitivo severo. [5]

2.1.1.3 MOCA

La prueba de evaluación cognitiva de Montreal, MOCA, es una prueba de 30 preguntas de duración entre 10 y 12 minutos, que ayuda a evaluar las disfunciones cognitivas leves. Publicado en 2005 por un grupo de la Universidad de McGill, evalúa la orientación, la memoria a corto plazo, la función ejecutiva, las habilidades del lenguaje, la abstracción, la denominación de los animales, la atención y el dibujo de un reloj. Cada una de estas habilidades cognitivas se organizan en secciones respectivamente, donde cada parte tiene una puntuación distintas. El máximo de puntos es 30, y se considera normal un puntuaje igual o superior a 26 donde se añade un punto si el sujeto tiene 12 años o menos. [6]

MONTREAL COGNITIVE ASSESSMENT (MOCA) (EVALUACIÓN COGNITIVA MONTREAL)		NOMBRE: Nivel de Educación: Sexo:	Fecha de nacimiento: FECHA:
VISUOESPACIAL / EJECUTIVA Copiar el cubo Dibujar un reloj (Once y diez) (3 puntos)		Puntos	
IDENTIFICACIÓN [] [] []		/3	
MEMORIA Leer la lista de palabras, el paciente debe reportarlas. Haga dos intentos. Recomendaciones: 5 minutos más tarde		ROSTRO SEDA IGLESIA CLAVEL ROJO	Sin puntos
ATENCIÓN Leer la serie de números (1 número/sig.) El paciente debe dar un golpeo con la mano cada vez que se diga la letra A. No se asignan puntos si > 2 errores. [] 93 [] 86 [] 79 [] 72 [] 65 4 o 5 sustacciones correctas 3 puntos, 2 o 3 correctas 2 puntos, 1 correcta 1 punto, 0 correctas 0 puntos.		El paciente debe repetir la inversa [] 7 4 2	/2
LENGUAJE Repita: El gato se esconde bajo el sofá cuando los perros entran en la sala. [] Espera que él le entregue el mensaje una vez que él dice lo que [] Juego del lenguaje. Decir el mayor número posible de palabras que comienzan por la letra 'P' en 1 minuto. [] (N = 11 palabras)		[] FBACMNAAJKLBFAKDEAAAJAMOFAAB	/1
ABSTRACCIÓN Similitud entre p. ej. manzana-naranja = fruta. [] trencadista [] relaj-ergle []		/2	
RECUERDO DIFERIDO Debe acordarse de las palabras SIN PISTAS ROSTRO [] SEDA [] IGLESIA [] CLAVEL [] ROJO [] Punto por acuerdo SIN PISTAS únicamente		/5	
Optativo Pista de categoría Pista elección múltiple			
ORIENTACIÓN [] Día del mes [] Mes [] Año [] Día de la semana [] Lugar [] Localidad []		Normal > 26 / 30	TOTAL /30
<small>© Z. Nasreddine MD Versión 07 noviembre 2004 www.mocatest.org</small>			

Ilustración 3: MOCA

El MOCA es una prueba que ayuda a los profesionales de la salud a determinar si una persona tiene una función cognitiva anormal y con ello, comprobar si tiene la enfermedad de Alzheimer con un examen de diagnóstico más completo. [6]

2.1.2 Cognifit

La primera de ellas es “Cognifit”, una aplicación móvil de gimnasia cerebral y estimulación cognitiva para evaluar, entrenar la memoria y agilidad mental. Para ello, estimula las capacidades cognitivas del usuario con distintos tipos de pruebas como rompecabezas, retos mentales o juegos de ingenio y con ello, identificar el riesgo de padecer alteraciones cognitivas. Está destinada a realizar una evaluación como la de un especialista y para ayudar a la salud cognitiva de los usuarios sanos adaptándose automáticamente a las necesidades cognitivas de cada usuario ya sean niños, adultos o personas mayores. [7]



Ilustración 4: Logo Cognifit

“Cognifit” contiene una batería de pruebas neuropsicológica orientada a la detección de alteraciones o deficiencias concretas. Dirigida a detectar y valorar diferentes áreas cognitivas como la atención y concentración, percepción,

memoria, funciones ejecutivas, coordinación, bienestar físico, psicológico y social del usuario. A continuación, se describirán algunas de las pruebas. [8]

“El Rectángulo” es una prueba para medir el tiempo de respuesta y velocidad de procesamiento donde el objetivo es pulsar un rectángulo en la pantalla la mayor cantidad de veces en un tiempo. [8]

“Círculos y Hexágonos” evalúa la atención focalizada, percepción espacial, escaneo visual, velocidad de procesamiento, coordinación y tiempo de respuesta. Consiste en pulsar el centro de cada círculo que aparece en pantalla, de la manera más precisa y rápida posible. [8]

“Cifras y Formas” analiza la inhibición, monitorización, flexibilidad cognitiva, velocidad de procesamiento y tiempo de reacción del usuario. En esta prueba, el usuario tiene que pulsar el circulo con mayor tamaño ignorando el número que tenga, y cuando se indique, se debe pulsar sobre el circulo con mayor número. [8]

“Los Números” mide la memoria a corto plazo, memoria fonológica a corto plazo, memoria de trabajo, velocidad de procesamiento y tiempo de respuesta. Consiste en una serie de bolas con distintos números donde se tiene que memorizar un patrón y luego replicarlo. [8]

“Figuras Musicales” es una prueba para medir la estimación cuyo objetivo es reproducir el sonido durante el mismo tiempo que el modelo. [8]

“Sigue la Bola” evalúa la monitorización, flexibilidad cognitiva, coordinación ojo-mano y velocidad de procesamiento. Consiste en mantener el cursor lo más cerca del centro de una bola que se va moviendo. [8]

“Laberintos” analiza la planificación, percepción espacial, y escaneo visual. Consiste en llegar a la meta de cada laberinto en el menor tiempo posible y con el menor número de pasos posibles. [8]

“Tres figuras” mide la percepción espacial, escaneo visual, reconocimiento, memoria no verbal, memoria de trabajo, velocidad de procesamiento y el tiempo de respuesta. El objetivo es identificar que grupo de tres figuras es idéntico al modelo mostrado. [8]

“Colores y Palabras” evalúa la inhibición, monitorización, percepción visual, denominación, velocidad de procesamiento y el tiempo de respuesta. Consiste en pulsar o no pulsar un botón tan rápido como el nombre del color que corresponda con el color en el que está escrito. [8]

“La bola” analiza la coordinación ojo-mano donde el objetivo es mantener el cursor lo más centrado posible en la bola. [8]

“Círculos Iluminados” mide la percepción espacial, memoria a corto plazo, memoria visual a corto plazo, memoria no verbal, memoria de trabajo, planificación, velocidad de procesamiento, y el tiempo de respuesta donde el objetivo es recordar la mayor secuencia de estímulos visuales posibles. [8]

“Las Letras” evalúa la percepción visual, denominación, velocidad de procesamiento y el tiempo de respuesta del usuario. Consiste en identificar el nombre del objeto mostrado y elegir la primera letra de su nombre lo más rápido posible. [8]

“Objetivos Vistos o Escuchados” mide la monitorización, percepción visual, percepción auditiva, reconocimiento, memoria contextual, memoria no-verbal, denominación, memoria de trabajo, velocidad de procesamiento y el tiempo de respuesta. Consiste en memorizar los objetos presentados y la forma en la que se presentan para después, indicar si fueron presentados en forma de imagen, sonido, o si es la primera vez que sale el objeto en cuestión. [8]

“Imágenes y Palabras” analiza la percepción visual, percepción auditiva, memoria contextual, memoria no verbal, denominación, velocidad de procesamiento y el tiempo de reacción. Consiste en identificar si el estímulo mostrado ha sido mostrado con anterioridad. [8]

“Rápido y Curioso” evalúa la estimación y el objetivo es estimar la velocidad de los estímulos. [8]

“Perspectivas y Distancias” mide la estimación y tiene como objetivo estimar la distancia entre estímulos. [8]

“La Bola y los Colores” evalúa a tención dividida, inhibición, monitorización, percepción visual, flexibilidad cognitiva, velocidad de procesamiento, coordinación ojo-mano y tiempo de respuesta. Consiste en realizar una tarea de coordinación a la vez que una tarea tipo “Go, No-Go”. [8]

2.1.3 CANTAB

“CANTAB” es un software que incluye medidas altamente sensibles, precisas y objetivas de la función cognitiva correlacionadas con redes neuronales. Contiene una serie de pruebas que incluyen pruebas de memoria, aprendizaje, función ejecutiva, memoria visual, memoria episódica y memoria verbal, atención, procesamiento de información, tiempo de reacción reconocimiento social y de emociones, toma de decisiones, y control de la respuesta. Asimismo, las pruebas muestran una alta sensibilidad a los efectos farmacológicos, genéticos y ambientales tanto positivos como negativos en individuos sanos y poblaciones de pacientes. [9]



Ilustración 5: Logo CANTAB

“CANTAB” contiene las tareas más validadas y usadas para la medición de la cognición. Estas pruebas se pueden agrupar en cuatro secciones: velocidad de atención y psicomotriz, función ejecutiva, memoria y cognición emocional y social.

La atención cubre nuestra habilidad de seleccionar la información a la que queremos atender ignorando el resto. Las habilidades psicomotoras sirven para relacionar las funciones cognitivas con los movimientos. Las pruebas de esta sección son las siguientes: en primer lugar, “Reaction Time” que mide la velocidad de respuesta motora y mental, tiempo de reacción, precisión,

impulsividad y medidas del tiempo de movimiento. Consiste en mantener un botón apretado en la parte más baja de la pantalla hasta que se ilumine uno de los cinco posibles círculos, que es cuando el usuario tendrá que presionar ese círculo lo antes posible. En segundo lugar, “Rapid Visual Information Processing” que mide la atención sostenida. Consiste en detectar un patrón de números en una secuencia pseudoaleatoria de números. En tercer lugar, “Motor Screening Task” mide las habilidades sensoriomotoras y consiste en pulsar sobre las cruces que van saliendo en pantalla de la manera más precisa y rápida posible. En cuarto lugar, “Match to Sample Visual Search” evalúa la atención y la búsqueda visual. Consiste en encontrar un patrón de los posibles, que coincidía con el del patrón complejo que se encuentra en medio de la pantalla. En último lugar, “Adaptive Tracking Task” mide la coordinación visomotora y la vigilancia. Consiste en mantener el dedo lo más cerca del centro de una bola que se va moviendo, y automáticamente, la velocidad de ésta, se ajusta al rendimiento del usuario. [9]

La función ejecutiva es el dominio cognitivo que abarca el pensamiento de alto nivel y la toma de decisiones. Las pruebas de esta sección son las siguientes: en primer lugar, “Cambridge Gambling Task” que se desarrolló para evaluar la toma de decisiones y comportamiento de riesgos fuera de un contexto de aprendizaje. Consiste en elegir en qué color de caja, roja o azul, se esconde una ficha de color amarillo de un total de 10 cajas cuyo color es aleatorio. En segundo lugar, “Intra-Extra Dimensional Set Shift” que presenta discriminación visual y mantenimiento de la formación de conjuntos de atención, desplazamiento y flexibilidad de atención. Consiste en utilizar retroalimentación para elaborar reglas que determinen qué estímulo es el correcto a partir de figuras rosas y línea blancas. En tercer lugar, “Multitasking Test” es una prueba para evaluar la capacidad de gestionar la información contradictoria proporcionada por la dirección de una flecha y su ubicación en la pantalla. Por ello, se debe seleccionar el botón de la izquierda o derecha dependiendo de las indicaciones en pantalla, lado en el que se encuentra la flecha o dirección de la flecha. En cuarto lugar “One Touch Stockings of Cambridge” mide la función ejecutiva basándose en la prueba de las “Torres de Hanoi”. En esta prueba, se dispone de dos secciones que contienen tres bolas de colores, y el objetivo es que la sección de abajo tenga el mismo patrón que el de arriba con movimientos parecidos a la prueba de las “Torres de Hanoi”. En quinto lugar, “Spatial Working Memory” que requiere de la retención y manipulación de información visoespacial. Consiste en descubrir una ficha de color amarillo que se encuentra escondida en una de las cajas de colores que aparecen en pantalla con el objetivo de llenar una columna. En sexto lugar, “Stop Signal Task” que mide el control de impulsos. El usuario debe pulsar el botón de la izquierda o derecha dependiendo de la dirección de una flecha que aparece en medio de la pantalla, pero no debe pulsar nada si se oye una señal de audio (beep). En último lugar, “CognitionKit Digit Symbol Substitution” es una medida global de la capacidad cognitiva, que requiere de múltiples dominios cognitivos. En esta prueba se muestra un array de símbolos abstractos con números, y se presenta uno de esos números el cual, el usuario tendrá que dibujar su correspondiente símbolo en un tiempo determinado. [9]

La memoria es la habilidad de almacenar información, a largo o corto plazo. Las pruebas de esta sección son las siguientes: la primera prueba, “Delayed Matching to Sample” que evalúa, simultáneamente, la capacidad de emparejamiento visual y la memoria a corto plazo para patrones no verbalizables. El usuario es enseñado un patrón complejo seguido de cuatro

similares y el individuo tiene que escoger el que coincide. La segunda prueba, “Digit Span” es una medida de la memoria verbal a corto plazo y funcional. El participante oirá una secuencia de dígitos e inmediatamente tendrá que repetir dicha secuencia, en el orden que se escuchó o en orden inverso. La tercera prueba, “Paired Associates Learning” que evalúa memoria visual y el aprendizaje. En la pantalla se muestran una serie de cajas en la que algunas tendrán un patrón que se mostrará al usuario, y después de ello, saldrá en el medio de la pantalla un patrón, y el individuo deberá seleccionar en qué caja está ese mismo patrón. La cuarta prueba “Spatial Span” evalúa la capacidad de memoria de trabajo visoespacial mediante una serie de cuadrados blanco y uno de ellos, en un cierto momento, cambia de color, y el usuario debe pulsarlo lo antes posible. La quinta prueba “Verbal Paired Associates” es una evaluación de la memoria asociativa y episódica en la que la prueba consiste en aprender un conjunto de pares de palabras tras haberlos escuchado. La sexta prueba “Verbal Recognition Memory” evalúa la memoria verbal y el aprendizaje. Esta prueba enseña al individuo una secuencia de palabras en pantalla una por una, y después, el usuario debe recordarlas. La última prueba “CognitionKit N-Back” mide la atención y la memoria de trabajo. El individuo verá un flujo de formas abstractas (estímulos) una tras otra y tiene que determinar si el estímulo que aparece en la pantalla coincide con el estímulo mostrado justo antes. [9]

La cognición emocional y social se centra en entender como procesamos la información afectiva. Las pruebas definidas a continuación, evalúan la capacidad de responder a estímulos cargados de emociones. La primera prueba es “Emotional Bias Task” que detecta sesgos perceptivos en la percepción de emociones faciales. En esta tarea se visualizan unas caras durante un escueto tiempo y el usuario tiene que elegir entre dos emociones la correcta. La segunda prueba es “Emotion Recognition Task” que mide la habilidad de identificar los seis emociones faciales básicas. En esta tarea se visualizan una expresión facial durante 200ms y el individuo tiene que escoger la expresión enseñada entre las seis (tristeza, felicidad, miedo, enfado, asco o sorpresa). [9]

2.1.4 CoGNIT

“CoGNIT” es una aplicación web para la evaluación automática, estandarizada y eficiente de la función cognitiva. Las pruebas incluidas abarcan los dominios cognitivos como la memoria, el lenguaje, velocidad psicomotora, función ejecutiva, atención, capacidad visoespacial, destreza manual y síntomas de depresión. [10]

En un estudio, se evalúó las pruebas cognitivas de “CoGNIT” para 36 pacientes de avanzada edad con síntomas de deterioro cognitivo leve reclutados en el “Blekinge Institute of Technology” en Suecia. Los resultados concluyeron que un 42% realizaron mal la prueba de pulsación con cuatro dedos, hubo problemas en la claridad de las instrucciones en dos pruebas, y se encontraron algunos bugs menores. Por lo que la aplicación mostró buenos resultados. [10]

La batería de pruebas de esta aplicación se divide en memoria, función ejecutiva y velocidad psicomotriz, atención, lenguaje, velocidad de motor, habilidad visoespacial, y depresión. Ahora se verá algunas de las pruebas de cada sección que estuvieron presentes en el estudio mencionado anteriormente. [10]

Para la memoria, se realizó una prueba en la que se presentaban 10 sustantivos por audio y texto. El paciente debía recordar las palabras y decirlas en alto

después de su presentación. Después de 15 minutos y dos tareas de distracción, se le muestra al paciente 20 palabras, 10 que salieron anteriormente y 10 que no, y él tendrá que decidir si la palabra se incluyó en la secuencia de sustantivos del inicio. [10]

Para la función ejecutiva y velocidad psicomotriz, se realizó el “Stroop test” que consta de dos partes. Por un lado, el paciente debe presionar uno de dos posibles botones de colores indicado por texto. Por otro lado, el texto es coloreado y el individuo es preguntado que presione sobre el botón indicado por el color del texto y no lo que está escrito. Además, se realizó el “Trail Making Test” que incluye también dos pruebas. Por una parte, los números del 1 al 25 son presentados en pantalla y el paciente debe pulsar los botones en orden numérico. Por otra parte, ambos números (del 1 al 13) y letras (de la “A” hasta la “L”) aparecen en pantalla. El individuo debe pulsar sobre las letras y números en orden alternando entre números sucesivos y letras del alfabeto (1-A-2-B-3-C...). [10]

Para la atención, el paciente debe presionar sobre uno de dos posibles botones indicado por una flecha lo más rápido posible. [10]

Para el lenguaje, el paciente tiene que decir en voz alta todas las palabras que pueda relacionadas por una categoría durante un minuto. [10]

Para la velocidad de motor, en la prueba de pulsación con cuatro dedos, se requiere que el paciente toque un teclado pequeño con dígitos del 2 al 5 en un orden específico. El orden correcto de pulsación es 2-3-4-5-4-3-2-3... En la prueba de pulsación con un dedo, el paciente debe tocar con un dedo repetidamente entre dos círculos que hay en pantalla. [10]

Para la habilidad visoespacial, la prueba de diseño bloqueante se compone de una figura con cuatro bloques de colores. El individuo debe reorganizar los bloques para que coincidan con un patrón que aparece en la parte de arriba de la pantalla. [10]

Para la depresión, se preguntan 15 preguntas y el cliente debe responder con un “sí” o un “no”. [10]

2.2 Tecnologías utilizadas

Para poder realizar la biblioteca de pruebas cognitivas, se han usado una serie de herramientas que facilitan el trabajo. A continuación, se describirán las principales herramientas usadas en este proyecto, así como su uso en el mismo.

2.2.1 HTML5

HTML5 es la versión más reciente de HTML (HyperText MarkUp Language), el componente más básico de las web. Permite definir el significado y estructura del contenido de una web, y habitualmente, se mezcla con CSS, para describir la apariencia de la web, o JavaScript para determinar la funcionalidad de la web. [11]

Esta tecnología contiene enlaces que conectan distintas páginas web entre sí. Además, usa unas marcas para etiquetar texto, imágenes y otros contenidos. Estas marcas incluyen elementos como <head>, <title> o <body> entre muchos, que se distingue de otro texto porque está rodeado por “<” y ”>”. [11]



Ilustración 6: Logo HTML5

2.2.2 CSS

CSS (Cascading Style Sheets) es una tecnología que permite estilizar una página web. Permite añadir cualidades visuales y estéticas a un documento HTML, es decir, puede manipular los elementos de un HTML para darle forma, color o posición entre otras muchas opciones. [12]

CSS se usa para separar la presentación y el contenido. Es decir, mientras los documentos HTML se encargan de la información y datos, los documentos CSS se encargan de aspectos relacionados con el estilo. Con ello, se consigue una aplicación web con código más limpio, organizado y sin duplicaciones de estilo. [12]



Ilustración 7: Logo CSS

2.2.3 JAVASCRIPT

JavaScript (JS) es un lenguaje de programación ligero que se compila en tiempo de ejecución. Principalmente, está compuesto de scripts (secuencias de comandos) para páginas web, pero también se usa en otros entornos como Node.js o Adobe Acrobat. Es un lenguaje de programación basado en eventos, multiparadigma, de un solo hilo y dinámico. [13]

Se utiliza en el lado del cliente de la web, pudiéndose usar para customizar o programar el comportamiento de una página web cuando ocurre un evento. [13]



Ilustración 8: Logo JavaScript

2.2.4 API REST

REST (Representational State Transfer) es un servicio sin estado, el estado lo mantiene el cliente. Sirve como interfaz para conectar varios sistemas basados en el protocolo HTTP y permite obtener y generar datos devolviendo los datos en formatos como XML o JSON. [14]

Una API REST permite tener un servicio en el cual almacenar la lógica de negocio y servir los datos mediante recursos URL. Además, se apoya en los métodos básicos HTTP como POST (crear nuevo recurso), GET (obtener recursos), PUT (modificar recurso), o DELETE (borrar recurso). [14]

Se ha utilizado esta tecnología, para poder almacenar parámetros de las distintas pruebas de una forma cómoda y sencilla. Con este fin, los médicos podrán así evaluar a los pacientes accediendo a este servicio. [14]



Ilustración 9: REST

2.2.5 Tecnologías Móviles

La tecnología móvil hace referencia a los dispositivos y herramientas portátiles que permiten a los usuarios compartir información sin necesidad de cables. Esta tecnología está compuesta de tres componentes: dispositivos celulares, redes de telecomunicaciones y sistemas software. [15]

El uso principal de esta tecnología es la comunicación entre dispositivos bidireccional. Acceder a contenido multimedia, comunicarse con familiares, gestionar negocios a largas distancias entre otros ejemplos. [15]

La tecnología móvil dispone de cuatro características fundamentales que le permiten adaptarse a cualquier tipo de contexto. En primer lugar, la portabilidad, ya que, gracias a los avances en electrónica y circuitos integrados, se puede disponer de dispositivos más compactos y poderosos, y por ello, portables. En segundo lugar, la capacidad de integración, ya que la tecnología móvil es fácil de adaptar para cualquier dispositivo móvil, como por ejemplo las aplicaciones web. En tercer lugar, movilidad eficiente, lo que permite mantener procesos diarios en funcionamiento a la vez que un usuario se moviliza. En último lugar, escalabilidad, ya que tiene la capacidad de adaptarse a nuevas herramientas y hacer frente a la innovación con versiones mejoradas de productos ya funcionales. [15]

Los tecnologías móviles operan sobre una plataforma de red específica. Estas redes tienen una serie de limitaciones y ventajas. A continuación, se hablará de las siguientes redes: radiofrecuencia celular, redes móviles de proveedor, Wifi y Bluetooth. [15]

En primer lugar, los teléfonos celulares tienen la capacidad de utilizar redes de radiofrecuencia por lo que cada móvil puede comunicarse dentro de un mismo radio a través de llamadas de voz y mensajes de texto. En segundo lugar, las

redes móviles permiten recibir y enviar paquetes de datos. En este sentido se ha avanzado mucho: 1G, 2G, 2.5G, 3G, 4G y 5G cada versión mejorando a la anterior en cuanto a velocidad de navegación, velocidades de descarga, ancho de banda y conexiones más estables. En tercer lugar, la conexión wifi es un punto de acceso inalámbrico a una conexión por cable como fibra óptica siendo una de las alternativas más rápidas y confiables. Por último, la conexión Bluetooth sirve para la transmisión de datos a bajas frecuencias en un rango corto de distancia. [15]

A su vez, esta “tecnología móvil” dispone de diferentes funciones dependiendo del contexto en la que se aplique. Algunas de las funciones más conocidas son: acceso a la información, desarrollo basado en la nube, acceso a bienes y servicios, y gestión de tiempo productivo. [15]

2.2.6 Multiplataforma Móviles

En este proyecto, se quiere desarrollar una aplicación que se pueda ejecutar tanto en un ordenador convencional como en un dispositivo móvil o una tablet. Por ello, para pasar de una aplicación de ordenador a una de móvil, la aplicación debe ser multiplataforma. [16]

Las aplicaciones multiplataforma se caracterizan por estar creadas en un único lenguaje de programación que facilita su exportación y visualización en cualquier dispositivo. Solo necesitan unos ligeros cambios para su adaptación a cualquier aparato. Son muy cómodas puesto que no es necesario desarrollar la misma aplicación en otro lenguaje permitiendo un desarrollo más fluido. [16]

Para este proyecto, se decidió usar la herramienta Ionic, debido a su facilidad para adaptarse a un proyecto en JavaScript. Ionic es un SDK de código abierto que viene con componentes JavaScript y CSS. [16]



Ilustración 10: Logo IONIC

2.2.7 Visual Studio Code

Visual Studio Code es un editor de código fuente gratuito que se ejecuta en tu ordenador o en la web y está disponible en una variedad de sistemas operativos. Incluye soporte para los lenguajes de programación como son JavaScript, TypeScript y Node.js pero, tiene extensiones para otros lenguajes de programación como C++ o Java. Aparte, tiene un sistema llamado IntelliSense para el autocompletado de código como variables o métodos y otras potentes funcionalidades de edición. [17]

Se escogió este programa para la parte de la web ya que es una herramienta sencilla y cómoda de usar que facilita la edición de archivos HTML, CSS y JavaScript permitiendo crear aplicaciones web de calidad. Además, tiene una extensión denominada “Live Server”, que permite lanzar un servidor local con recarga de contenidos en directo, la cual ha servido para agilizar el proceso de creación de las interfaces y funcionalidades. [17]



Ilustración 11: Logo Visual Studio Code

2.2.8 IntelliJ IDEA Community Edition

IntelliJ Idea es un entorno de desarrollo integrado para Java. Es un programa fácil de usar, flexible y tiene un diseño sólido.[2] Además, se ha usado SpringBoot dentro de IntelliJ Idea para poder disponer de funcionalidades de los servicios REST. [18]

Por un lado, IntelliJ se ha usado por la potencia y facilidad que tiene y, personalmente, por tener conocimientos previos de su uso y experiencia. Por otro lado, se decidió usar SpringBoot ya que se pretendía usar Java como lenguaje para manejar un servicio REST, por lo que SpringBoot facilita inmensamente el disponer de un servidor REST y manejarlo. [18]

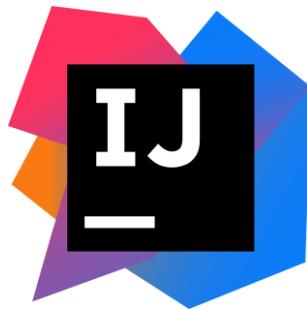


Ilustración 12: Logo IntelliJ Idea

2.2.9 Draw.io

Draw.io es un software de dibujo gráfico para la creación de diagramas y gráficas. Se ha usado este programa para poder crear los distintos diagramas del proyecto destacando en su simpleza y organización.



Ilustración 13: Logo Draw.io

3 Requisitos

Una vez entendida la motivación y el propósito del proyecto, así como realizado el estudio de las herramientas que se utilizarán, se realizó el diseño y desarrollo de la aplicación.

El primer paso fue concretar los tests cognitivos que formarían la batería de pruebas. Con ello, el cuerpo de la aplicación, así como los parámetros a guardar de los test que permitieran a posterior valorar la capacidad cognitiva del paciente.

Se establecieron siete pruebas distintas para la evaluación cognitiva, cada una con su lógica y su estructura, pero a su vez, se ha intentado mantener una interfaz lo más parecida posible. En este aspecto, cabe también destacar que se ha realizado un estudio para la homogeneización de los datos a guardar en cada test. Dado que el sentido del trabajo es evaluar a los pacientes, se debe guardar los parámetros necesarios. Por ello, el servicio REST también deberá tener una estructura similar para el almacenamiento de los datos de las distintas pruebas.

Antes de ver cómo están compuestos las distintas pruebas, se presentarán los distintos requisitos de las pruebas, así como los casos de uso.

3.1 Prueba 1, Parejas

En esta prueba denominada “PAREJAS”, el usuario deberá encontrar todas las cartas parejas de un tablero antes de que se acabe el tiempo. Esta prueba mide la memoria a corto plazo, así como la velocidad de procesamiento.

3.1.1 Requisitos Funcionales

P-RF.01: Mostrar un tablero con las casillas en blanco.

P-RF.02: Mostrar un temporizador, número de aciertos y número de movimientos.

P-RF.03: Permitir cambiar nivel dificultad.

P-RF.04: Mostrar contenido de una casilla.

P-RF.05: Desactivar los botones de cambio de nivel cuando comience la prueba.

P-RF.06: Habilitar los botones de cambio de nivel cuando finalice la prueba.

P-RF.07: Comprobar si dos casillas seleccionadas son iguales.

P-RF.08: Impedir pulsado cuando dos casillas no son iguales.

P-RF.09: Finalizar prueba cuando el temporizador llegue a cero.

P-RF.10: Finalizar prueba tras encontrar todas las parejas del tablero.

P-RF.11: Modificar el tablero, y temporizador tras cambio de dificultad.

P-RF.12: Mostrar todas las casillas si el temporizador llega a cero.

P-RF.13: Enviar parámetros tras cada movimiento.

P-RF.14: Permitir volver al menú principal.

3.1.2 Casos de Uso

El usuario cambia la dificultad de la prueba en el número de parejas a encontrar, por lo que el sistema cambia el nivel. Por defecto, está configurado el nivel uno. A continuación, el usuario pulsa sobre una de las casillas(cartas) de un tablero. El objetivo es ir seleccionando dos cartas para comprobar si son parejas. Cuando el usuario encuentra todas las parejas, la prueba termina. En paralelo al usuario, se comienza un temporizador. Cuando este llegue a cero, y el usuario no ha encontrado todas las parejas, la prueba finalizara. Por último, el usuario puede desplazarse al menú principal.

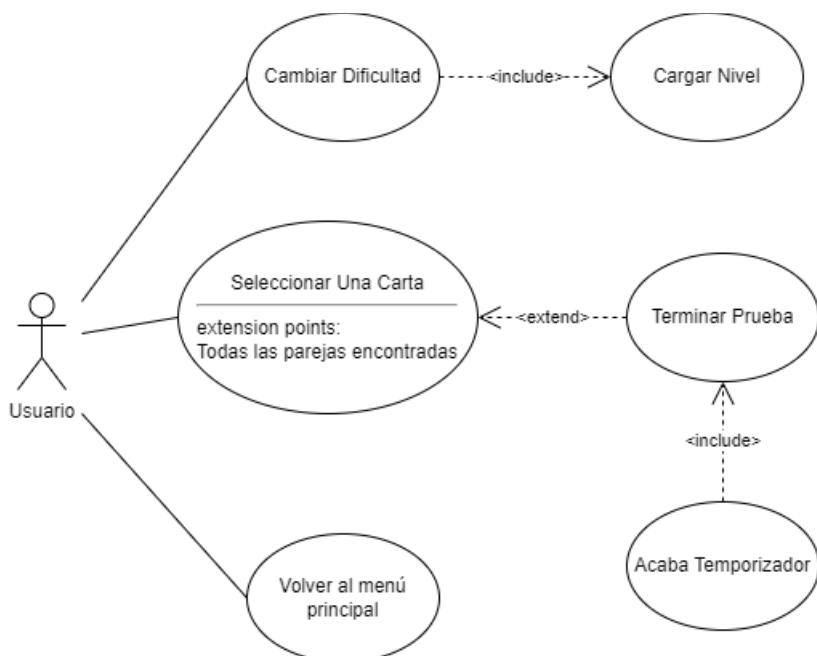


Ilustración 14: Casos de Uso Parejas

3.2 Prueba 2, Arrastra Y Suelta

En esta prueba denominada “Arrastrar y Soltar”, el usuario deberá arrastrar un elemento y depositarlo en su caja correspondiente. Este test mide la coordinación ojo-mano y el conocimiento de frutas y verduras.

3.2.1 Requisitos Funcionales

- AS-RF.01:** Mostrar un lista de imágenes.
- AS-RF.02:** Mostrar una lista de cajas etiquetadas.
- AS-RF.03:** Mostrar instrucciones de la prueba.
- AS-RF.04:** Permitir arrastrar las imágenes.
- AS-RF.05:** Permitir depositar las imágenes en cajas.
- AS-RF.06:** Detectar si una imagen corresponde con una caja.
- AS-RF.07:** Insertar imagen dentro de una caja.

AS-RF.08: Cambiar de nivel cuando todas las imágenes han sido depositadas en una caja.

AS-RF.09: Mostrar 5 frutas en el primer nivel.

AS-RF.10: Mostrar 10 verduras en el segundo nivel.

AS-RF.11: Finalizar prueba cuando en el segundo nivel se haya completado.

AS-RF.12: Enviar parámetros tras cada movimiento.

AS-RF.13: Permitir volver al menú principal.

3.2.2 Casos de Uso

El usuario inicia la prueba, cogiendo y arrastrando uno de los posibles objetos. Estos objetos los debe depositar en su caja oportuna. Cuando deposita todos los objetos en sus cajas, el sistema automáticamente carga el siguiente nivel. En este nuevo nivel, el usuario debe hacer lo mismo, pero cuando consigue depositar todos los objetos, la prueba habrá finalizado. Por último, el usuario puede desplazarse al menú principal.

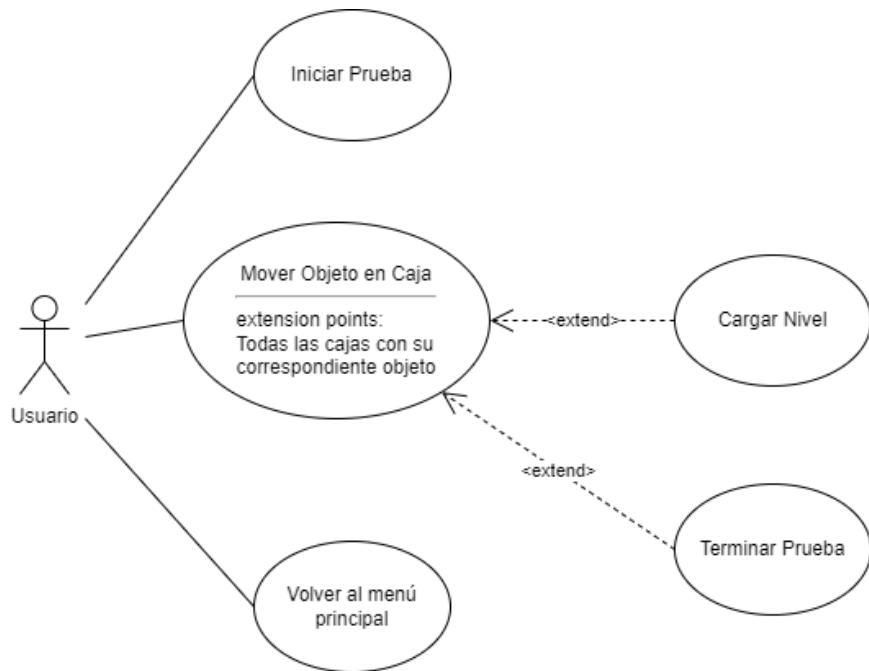


Ilustración 15: Casos de Uso Arrastra Y Suelta

3.3 Prueba 3, Recuerda Palabras

En esta prueba denominada “Recuerda Palabras”, el usuario deberá memorizar una serie de palabras y después, encontrarlas entre otro grupo más amplio de palabras. Este test mide la memoria.

3.3.1 Requisitos Funcionales

- RP-RF.01:** Mostrar opciones de la prueba.
- RP-RF.02:** Permitir cambiar nivel dificultad.
- RP-RF.03:** Permitir cambiar tiempo del temporizador.
- RP-RF.04:** Permitir visualizar o no el temporizador.
- RP-RF.05:** Cambiar número de palabras a recordar.
- RP-RF.06:** Cambiar tiempo del temporizador.
- RP-RF.07:** Esconder temporizador.
- RP-RF.08:** Mostrar palabras a recordar.
- RP-RF.09:** Desactivar los botones de configuración cuando comience la prueba.
- RP-RF.10:** Habilitar los botones de configuración cuando finalice la prueba.
- RP-RF.11:** Mostrar tablero que contenga las palabras a recordar y otras que no lo sean.
- RP-RF.12:** Permitir seleccionar las palabras.
- RP-RF.13:** Mostrar cuantas palabras quedan por escoger.
- RP-RF.14:** Mostrar puntuación final.
- RP-RF.15:** Mostrar las palabras correctas en caso de no acertar todas.
- RP-RF.16:** Permitir reiniciar la prueba.
- RP-RF.17:** Enviar parámetros tras pulsar una palabra.
- RP-RF.18:** Permitir volver al menú principal.

3.3.2 Casos de Uso

El usuario cambia la dificultad de la prueba en el número de palabras a recordar, lo que produce que el sistema cambie cargue el nuevo nivel. Puede además cambiar el tiempo total del temporizador o decidir si quiere visualizar el temporizador o no. Tras iniciar la prueba, el usuario puede seleccionar una palabra de las mostradas. Cuando pulse todas las palabras permitidas, la prueba finalizara. Cuando esto ocurra, el individuo puede cambiar la dificultad o reiniciar la prueba cargando de nuevo el nivel. Por último, el usuario puede desplazarse al menú principal.

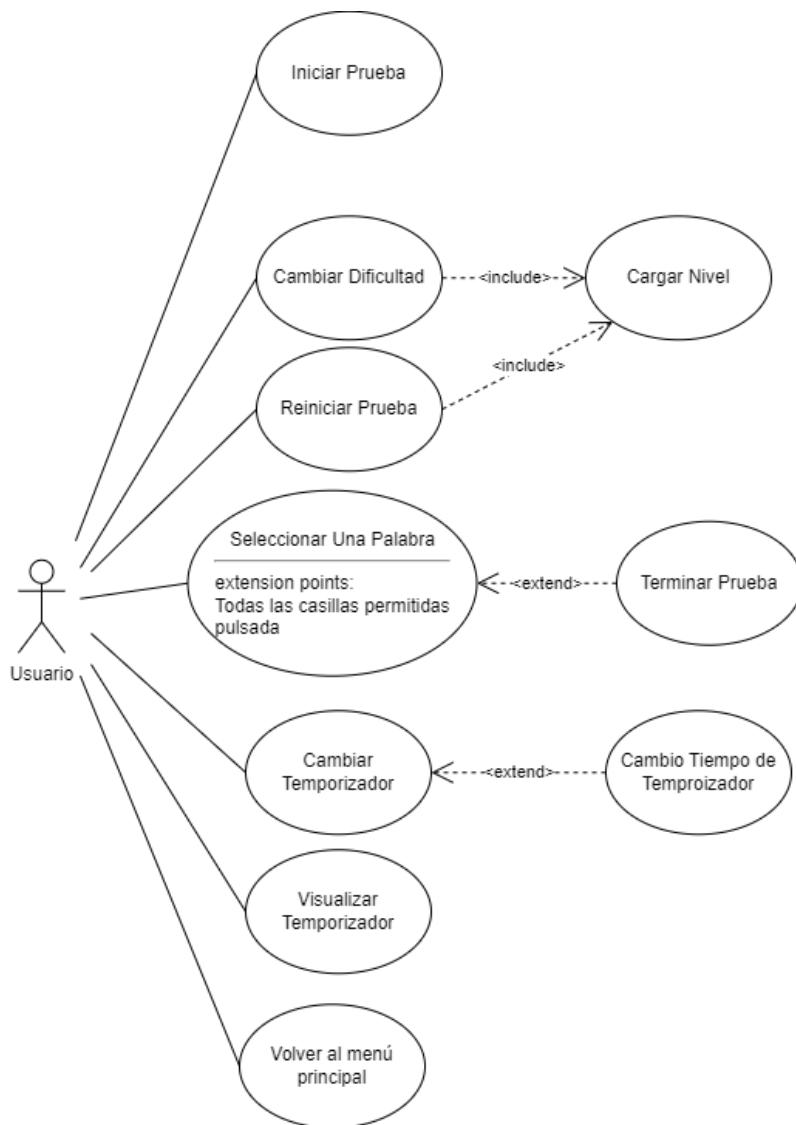


Ilustración 16: Casos de Uso Recuerda Palabras

3.4 Prueba 4, Noquea Las Torres

En esta prueba denominada “Noquea las Torres”, el usuario deberá superar las distintas torres derrotando a los enemigos hasta llegar al final. El jugador así como los distintos enemigos tienen asociado un número que simboliza su fuerza. Esta prueba mide las habilidades matemáticas y cálculo mental.

3.4.1 Requisitos Funcionales

NT-RF.01: Permitir activar o desactivar la música de fondo.

NT-RF.02: Encender o apagar la música de fondo.

NT-RF.03: Mostrar por pantalla el tiempo empleado en la prueba.

NT-RF.04: Mostrar por pantalla al jugador con su fuerza.

NT-RF.05: Mostrar por pantalla a los enemigos con su fuerza.

NT-RF.06: Mostrar por pantalla las casillas disponibles.

NT-RF.07: Permitir seleccionar las casillas de los enemigos y de la gema.

- NT-RF.08:** Mostrar por pantalla la gema cuando se esté en la última torre.
- NT-RF.09:** Sumar fuerza del enemigo al jugador si esté tiene más fuerza que el enemigo.
- NT-RF.10:** Avanzar a la siguiente torre si todos los enemigos de la actual torre han sido derrotados.
- NT-RF.11:** Indicar que un enemigo ha sido derrotado.
- NT-RF.12:** Indicar que el jugador ha sido derrotado.
- NT-RF.13:** Tras obtención de la gema, finalizar la prueba
- NT-RF.14:** Finalizar prueba si enemigo es más fuerte que el jugador.
- NT-RF.15:** Permitir reiniciar la prueba.
- NT-RF.16:** Enviar parámetros tras pulsar, en un enemigo o en la gema.
- NT-RF.17:** Permitir volver al menú principal.

3.4.2 Casos de Uso

El usuario inicia la prueba seleccionando sobre uno de los enemigos. Cuando derrote a los enemigos de una de las torres, el sistema automáticamente carga la siguiente torre. El usuario avanzara en las torres hasta llegar a la última, donde graba una casilla especial con una gema. Si consigue obtener la gema, la prueba terminara. Si el usuario selecciona un enemigo que es más fuerte que él, la prueba finalizara. Cuando esto ocurra, el usuario puede reiniciar la prueba y durante toda la prueba, puede decidir encender o apagar la música de fondo. Por último, el usuario puede desplazarse al menú principal.

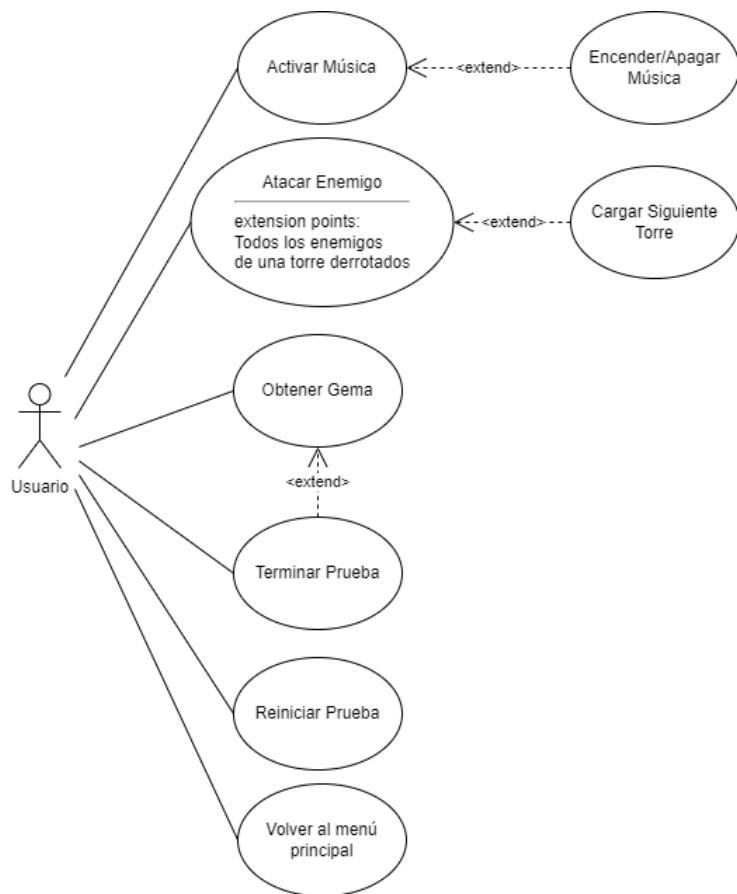


Ilustración 17: Casos de Uso Noquea Las Torres

3.5 Prueba 5, CentroBola

En esta prueba denominada “CentroBola”, el usuario tiene que pulsar el centro de los círculos que vayan apareciendo en pantalla. Esta prueba mide la monitorización y la coordinación ojo-mano.

3.5.1 Requisitos Funcionales

- CB-RF.01:** Mostrar las instrucciones de la prueba.
- CB-RF.02:** Permitir comenzar la prueba.
- CB-RF.03:** Mostrar círculos de diferentes dimensiones y colores.
- CB-RF.04:** Mostrar únicamente una bola en cada nivel.
- CB-RF.05:** Permitir avanzar en un nivel.
- CB-RF.06:** Permitir a la bola ser pulsada.
- CB-RF.07:** Desactivar el botón de siguiente nivel cuando comience un nivel.
- CB-RF.08:** Habilitar botón de siguiente nivel cuando se pulse sobre una bola.
- CB-RF.09:** Calcular la distancia desde el punto pulsado hasta el medio.
- CB-RF.10:** Mostrar la distancia por pantalla, indicando si es muy lejana en rojo y cercana en azul.
- CB-RF.11:** Finalizar la Prueba

CB-RF.12: Permitir reiniciar la prueba.

CB-RF.13: Enviar parámetros tras pulsar, un círculo.

CB-RF.14: Permitir volver al menú principal.

3.5.2 Casos de Uso

El usuario inicia la prueba y pulsa sobre la bola que haya en pantalla. Cuando esto ocurra, el usuario puede pasar al siguiente nivel, donde aparece otra bola. Cuando el usuario haya pulsado sobre todas las bolas, la prueba habrá finalizado. En este instante, el usuario puede reiniciar la prueba. Por último, el usuario puede desplazarse al menú principal.

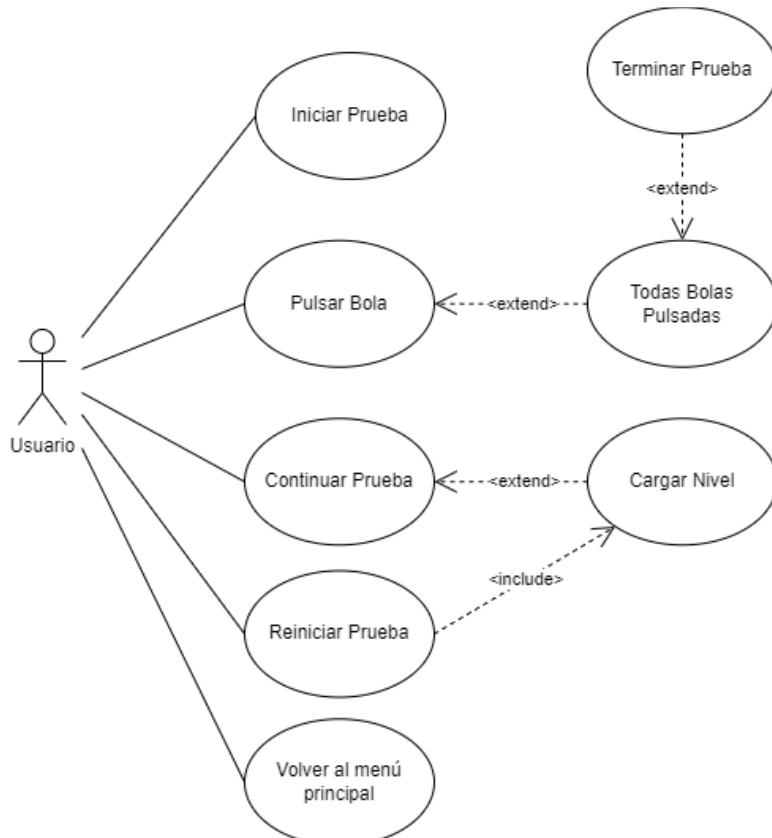


Ilustración 18: Casos de Uso CentroBola

3.6 Prueba 6, MetaBola

En esta prueba denominada “MetaBola”, el usuario deberá detener la bola en el momento oportuno, es decir, en la meta. Esta prueba mide la anticipación, cálculo mental, percepción espacial, flexibilidad cognitiva, y velocidad de procesamiento.

3.6.1 Requisitos Funcionales

MB-RF.01: Mostrar las instrucciones de la prueba.

MB-RF.02: Permitir cambiar nivel de dificultad.

MB-RF.03: Permitir iniciar la prueba.

- MB-RF.04:** Permitir detener la bola en cualquier momento del recorrido.
- MB-RF.05:** Cambiar la ruta cuando se cambie el nivel.
- MB-RF.06:** Desactivar los botones de cambio de nivel cuando comience la prueba.
- MB-RF.07:** Habilitar los botones de cambio de nivel cuando finalice la prueba
- MB-RF.08:** Mostrar camino que recorrerá la bola.
- MB-RF.09:** Mostrar la bola en movimiento.
- MB-RF.10:** Desaparecer la bola y ruta pasados unos segundos.
- MB-RF.11:** Calcular trayectoria de la bola dependiendo del nivel.
- MB-RF.12:** Pausar la bola cuando lo deseé el usuario.
- MB-RF.13:** Pausar bola cuando llegue al final del trayecto.
- MB-RF.14:** Enviar parámetros tras pulsar, detener la bola.
- MB-RF.15:** Permitir volver al menú principal.

3.6.2 Casos de Uso

El usuario cambia la dificultad de la prueba en el trayecto de la esfera, por lo que el sistema cambia el nivel. Cuando inicie la prueba, el usuario podrá detener la bola cuando lo vea oportuno. Cuando lo haga, terminara la prueba. Paralelamente, si la bola llega al final del recorrido, y el usuario no detuvo la bola, el sistema la detendrá y finalizará la prueba. Por último, el usuario puede desplazarse al menú principal.

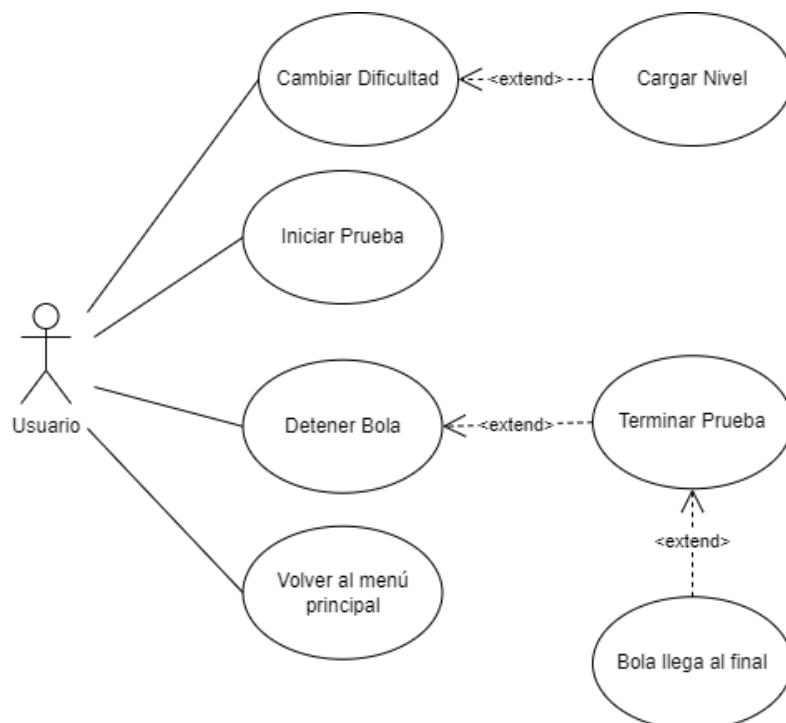


Ilustración 19: Casos de Uso MetaBola

3.7 Prueba 7, MOCA

En esta prueba denominada “MOCA”, se pretende simular el test de Evaluación Cognitiva de Montreal(MoCA) vista anteriormente. Esta prueba mide la memoria a corto plazo y recuperación de memoria retardada, la orientación, la función ejecutiva, la habilidad visoespacial, las habilidades del lenguaje, la abstracción, la denominación de animales y la atención.

3.7.1 Requisitos Funcionales

Debido a que esta prueba esta dividida en secciones, se detallaran primero, los requisitos funcionales que todas las secciones deben tener, y después los requisitos individuales de cada prueba. Para ello, los requisitos globales tendrán la forma “M-RF.OX” siendo X el número del requisito, y los requisitos individuales de la forma “MY-RF.OX” siendo Y el número de la sección y X el número del requisito.

M-RF.01: Estar compuesto por siete secciones distintas.

M-RF.02: Mostrar instrucciones de cada sección.

M-RF.03: Permitir avanzar a la siguiente sección una vez acaba la actual.

M-RF.04: Generar la siguiente sección.

M-RF.05: Enviar parámetros tras pasar al siguiente nivel.

M-RF.06: Finalizar la prueba cuando se llegue a la última sección.

Sección 1:

M1-RF.01: Mostrar un área de dibujo.

M1-RF.02: Permitir dibujar al usuario.

M1-RF.03: Mostrar lo que se debe dibujar.

M1-RF.04: Permitir borrar el dibujo.

M1-RF.05: Borrar el dibujo.

M1-RF.06: Permitir volver al menú principal.

Sección 2:

M2-RF.01: Mostrar un área de dibujo.

M2-RF.02: Permitir dibujar al usuario.

M2-RF.03: Permitir borrar el dibujo.

M2-RF.04: Borrar el dibujo.

Sección 3:

M3-RF.01: Mostrar un león, un rinoceronte y un camello.

M3-RF.02: Mostrar tres campos de texto.

M3-RF.03: Identificar que campo de texto corresponde a que animal.

M3-RF.04: Permitir escribir sobre los campos de texto.

Sección 4:

- M4-RF.01:** Mostrar una pista de audio
- M4-RF.02:** Permitir escuchar una pista de audio.
- M4-RF.03:** Permitir grabar la voz.
- M4-RF.04:** Permitir detener la grabación.
- M4-RF.05:** Mostrar un temporizador de un minuto después de la primera grabación.
- M4-RF.06:** Permitir escuchar la pista de audio dos veces.
- M4-RF.07:** Permitir realizar una grabación de voz.

Sección 5:

- M5-RF.01:** Mostrar tres pistas distintas de audio.
- M5-RF.02:** Permitir escuchar las distintas pistas de audio.
- M5-RF.03:** Permitir realizar dos grabaciones de voz.
- M5-RF.04:** Permitir detener una grabación.
- M5-RF.05:** Mostrar un botón para detectar cuantas “A” se han dicho en una grabación.
- M5-RF.06:** Contar número de veces que se pulsa el botón para detectar “A”.
- M5-RF.07:** Mostrar cinco campos de texto.
- M5-RF.08:** Permitir únicamente escribir números en los campos de texto.

Sección 6:

- M6-RF.01:** Mostrar dos frases en azul.
- M6-RF.02:** Permitir realizar tres grabaciones de voz.
- M6-RF.03:** Permitir detener una grabación.
- M6-RF.04:** Mostrar un contador de un minuto que muestre el tiempo en el que el usuario debe pronunciar palabras empezando por la letra “P”.
- M6-RF.05:** Detener la tercera grabación de voz cuando pase un minuto.
- M6-RF.06:** Mostrar dos parejas de palabras en azul.
- M6-RF.07:** Mostrar dos campos de texto.
- M6-RF.08:** Permitir escribir en los campos de texto la relación de los pares de palabras.

Sección 7:

- M7-RF.01:** Mostrar once campos de texto.
- M7-RF.02:** Mostrar las instrucciones de cada campo de texto.
- M7-RF.03:** Permitir escribir en los campos de texto.
- M7-RF.04:** Mostrar diez botones, cinco con “Pista 1” y cinco con “Pista 2”.
- M7-RF.05:** Mostrar una pista si se pulso sobre alguno de los botones.

3.7.2 Casos de Uso

En este apartado, se presentará los casos de uso de las distintas secciones:

Sección 1:

El usuario dibuja un camino, y el sistema lo va mostrando por pantalla. Ocurre lo mismo cuando intenta dibujar un cubo. Además, puede decidir borrar el dibujo entero, donde el sistema vaciará el área de dibujo. Cuando acabe de dibujar, puede continuar con el MOCA. Por último, el usuario puede volver al menú principal.

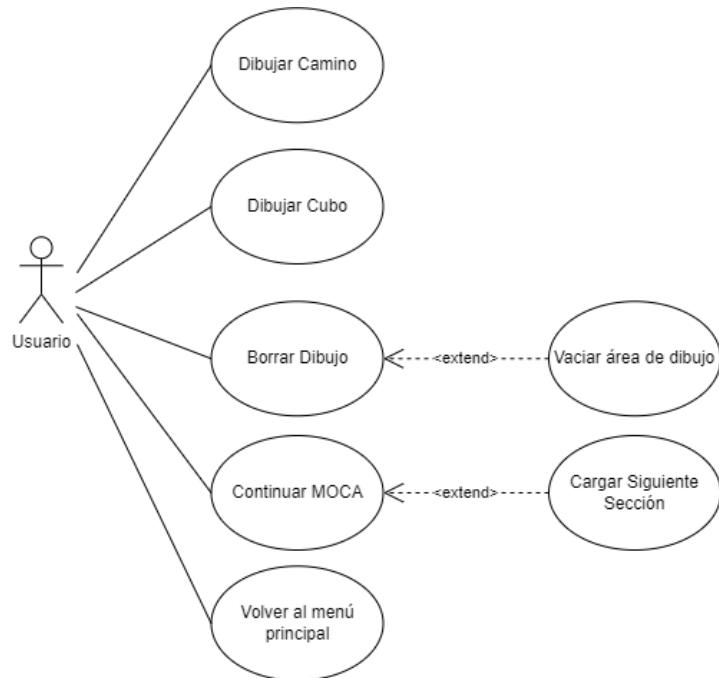


Ilustración 20: Casos de Uso MOCA Sección 1

Sección 2:

El usuario dibuja un reloj y el sistema se lo muestra por pantalla. Puede decidir borrar el dibujo entero, donde el sistema vaciará el área de dibujo. Cuando acabe de dibujar, puede continuar con el MOCA.

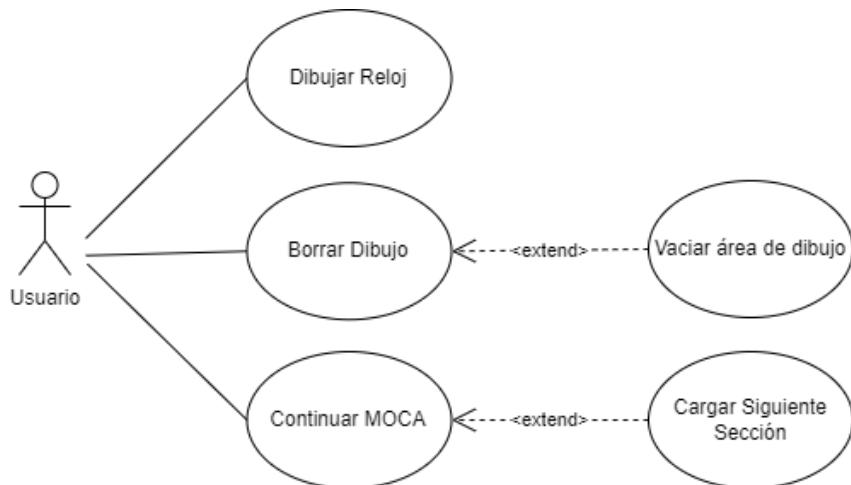


Ilustración 21: Casos de Uso MOCA Sección 2

Sección 3:

El usuario escribe los nombres de los animales que aparecen en pantalla. Cuando acabe de identificarlos, puede continuar con el MOCA.

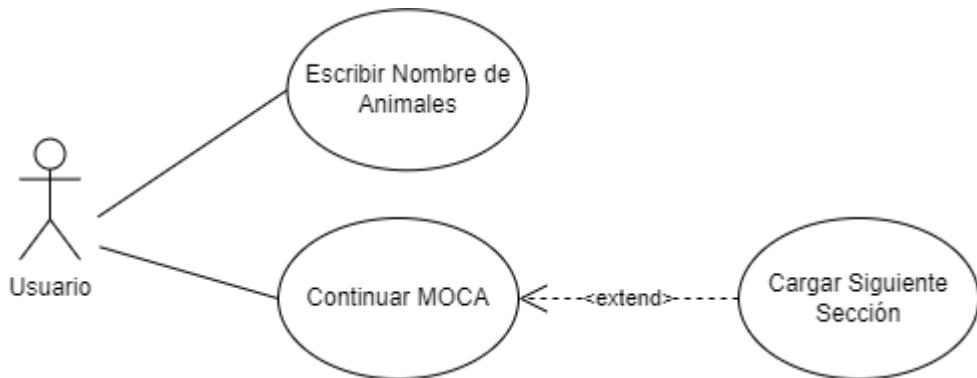


Ilustración 22: Casos de Uso MOCA Sección 3

Sección 4:

El usuario reproduce una pista de audio. A continuación, se realiza una grabación de su voz de lo que se le ha pedido. Cuando acabe, lo para y después de un minuto, volverá a escuchar la pista de audio y realizará otra vez, una grabación. Cuando acabe las dos grabaciones, puede continuar con el MOCA.

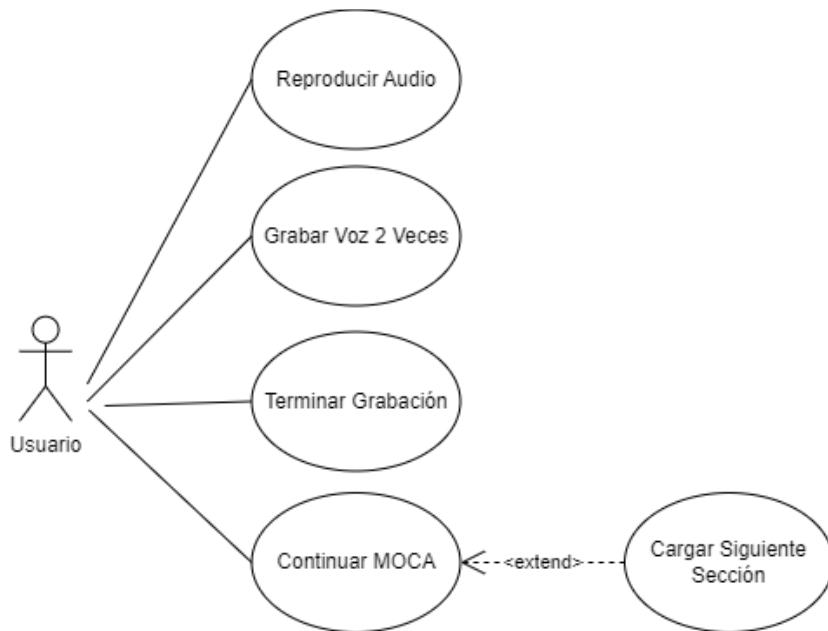


Ilustración 23: Casos de Uso MOCA Sección 4

Sección 5:

El usuario reproduce el primer audio y se graba con las indicaciones requeridas. Después reproduce el segundo audio y se graba con las indicaciones requeridas. Seguidamente, escucha otro audio distinto donde por cada letra “A” escuchada, debe pulsar un botón. A continuación, realiza la resta de 7 en 7 empezando desde 100 escribiendo los resultados en los campos de texto habilitados. Cuando acabe con las restas, puede continuar con el MOCA.

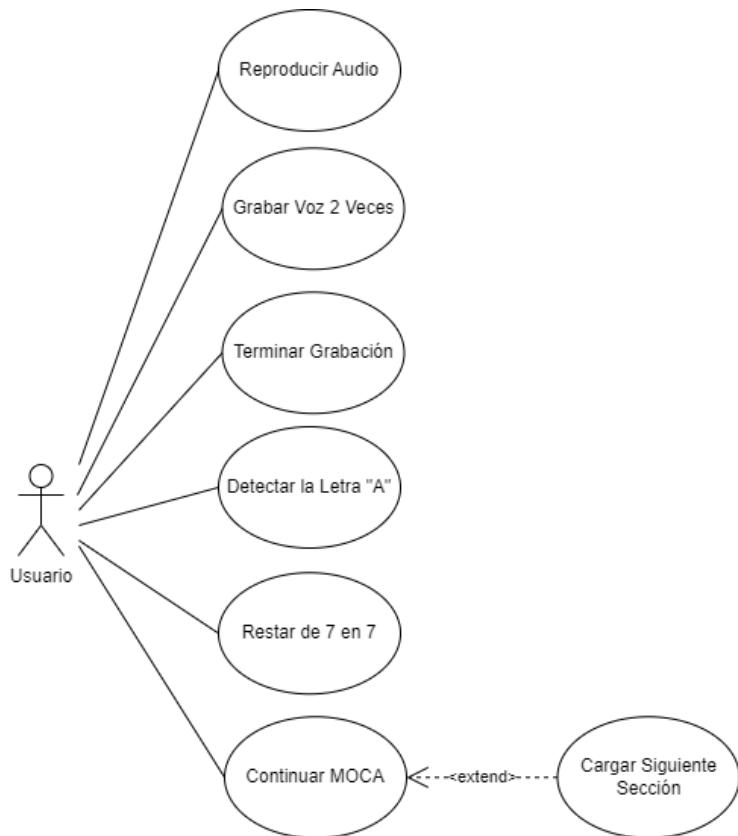


Ilustración 24: Casos de Uso MOCA Sección 5

Sección 6:

El usuario lee dos frase y se graba diciéndolas, en dos grabaciones distintas. A continuación, se graba durante un minuto diciendo palabras que comiencen con la letra “P”. Por último, debe escribir la similitud entre dos parejas. Cuando acabe, puede continuar con el MOCA.

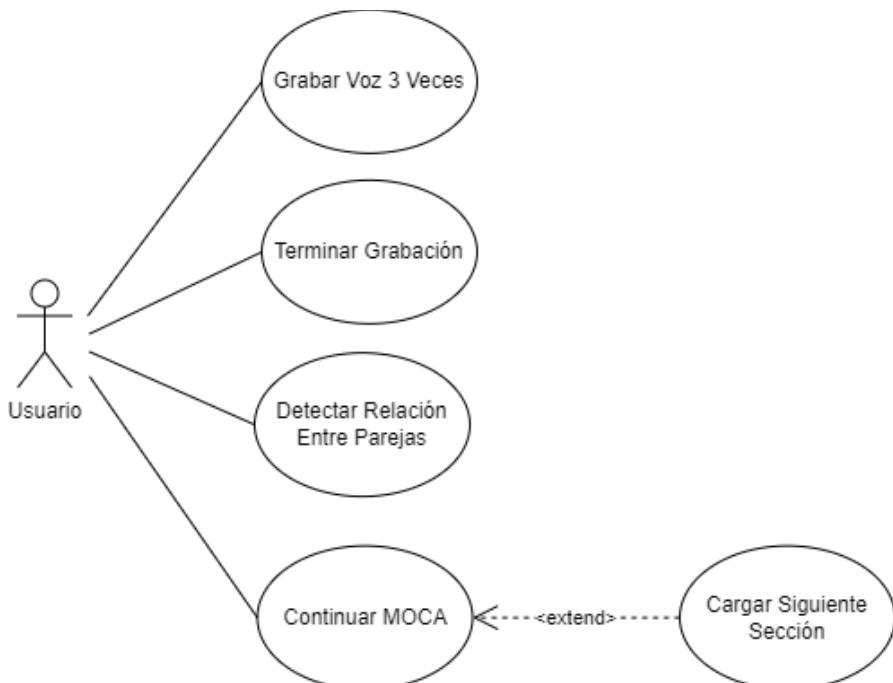


Ilustración 25: Casos de Uso MOCA Sección 6

Sección 7:

El usuario escribe en los campos de texto la información solicitada. Hay cinco campos que requieren escribir unas palabras que se tendría que haber acordado, pero si no se acuerda, puede usar una serie de botones que le darán pistas para recordar las palabras. Cuando finalice de llenar los campos, puede finalizar la prueba.

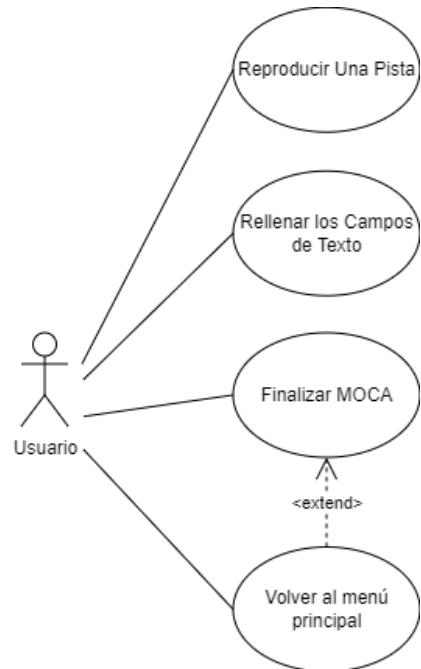


Ilustración 26: Casos de Uso MOCA Sección 7

4 Desarrollo

En esta sección, se indica en qué consiste individualmente cada una de estas pruebas: qué evalúan, en qué consisten, que información obtenemos, como están construidas y como es su apariencia. Además, se detalla un menú principal para inscribir los datos del individuo y permitir seleccionar cualquiera de las pruebas.

4.1 Prueba 1, Parejas

4.1.1 Diseño

El usuario tiene dos opciones: pulsar una casilla y después otra, o puede cambiar la dificultad de la prueba. Cuando se pulsa en una casilla, el sistema desvelará su contenido. Si se selecciona una segunda carta, el sistema detectará si son iguales o no, y enviará la información al momento. Por un lado, si no son iguales, el sistema esconderá las dos cartas seleccionadas. Por otro, si son iguales, se puntuará con un acierto. En ambos casos, se enviará información al servicio REST.

Cuando se llegue al número de aciertos necesarios para finalizar la prueba, se enviarán los resultados. A su vez, cuando se selecciona la primera carta de todas, se activa el temporizador de tiempo. Si éste se acaba, se detendrá la prueba y se puntuaran los resultados.

Toda esta lógica viene representada en un diagrama de alto nivel como se puede ver en la ilustración 27.

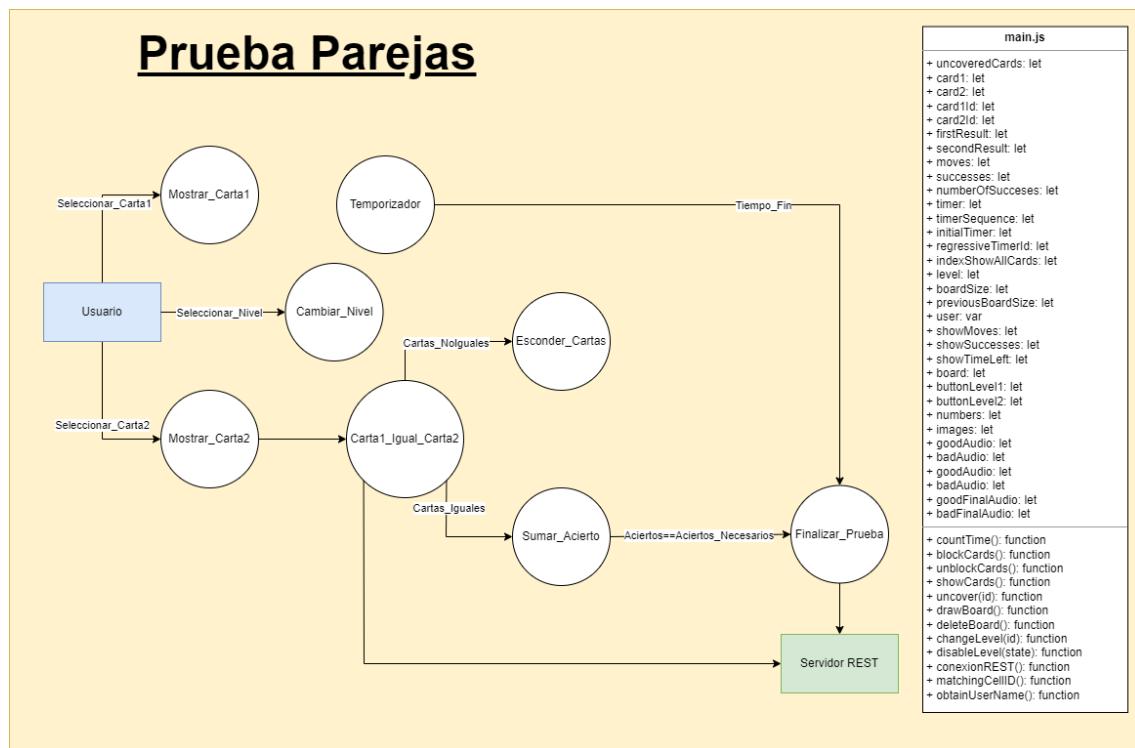


Ilustración 27: Diagramas Parejas

Respecto al diseño de bajo nivel de la ilustración 27, se empezará hablando de las variables.

En primer lugar, se necesita guardar cuantas cartas desveladas en un turno, una o dos, así como qué cartas son y su contenido.

En segundo lugar, se deben guardar el nivel, la dificultad de la prueba, los movimientos realizados, aciertos, el tamaño del tablero, las imágenes, y los números que representan a las imágenes, para facilitar la comparación de las cartas.

En tercer lugar, se necesitan variables para acceder a los elementos HTML con lo que poder modificar el temporizador, el número de movimientos, el tablero, número de aciertos y los botones.

En cuarto lugar, variables para controlar el tiempo y, por último, variables de los efectos de sonido.

Sobre los métodos, “countTime()” contiene la lógica del temporizador, “blockCards()” bloquea al usuario de seleccionar más cartas mientras se están comparando dos cartas, “unblockCards()” desbloquea todas las cartas excepto las parejas ya encontradas, “showCards()” muestra el contenido de todas las cartas. La función “uncover(id)” es la más importante ya que es el método al que se accede cuando se selecciona una carta y se encarga de comparar las dos cartas para ver si son parejas o no. El resto de las funciones: “drawBoard()” muestra el tablero necesario dependiendo del nivel, “deleteBoard()” elimina el tablero que hay en pantalla cuando se cambia de nivel, “changeLevel(id)” cambia el nivel, “disableLevel(state)” impide al usuario cambiar de nivel, “conexionREST()” y “matchingCellID()” sirven para enviar los parámetros necesarios al servicio REST, y “obtainUserName()” obtiene el nombre del usuario realizando la prueba.

4.1.2 Interfaz y Uso

El sistema muestra por pantalla un tablero de 4x4 formado por botones blancos denominados cartas. Debajo del título, aparecen dos botones para cambiar la dificultad, un botón para volver al menú principal , el número de aciertos, el tiempo disponible, y el número de movimientos realizados.

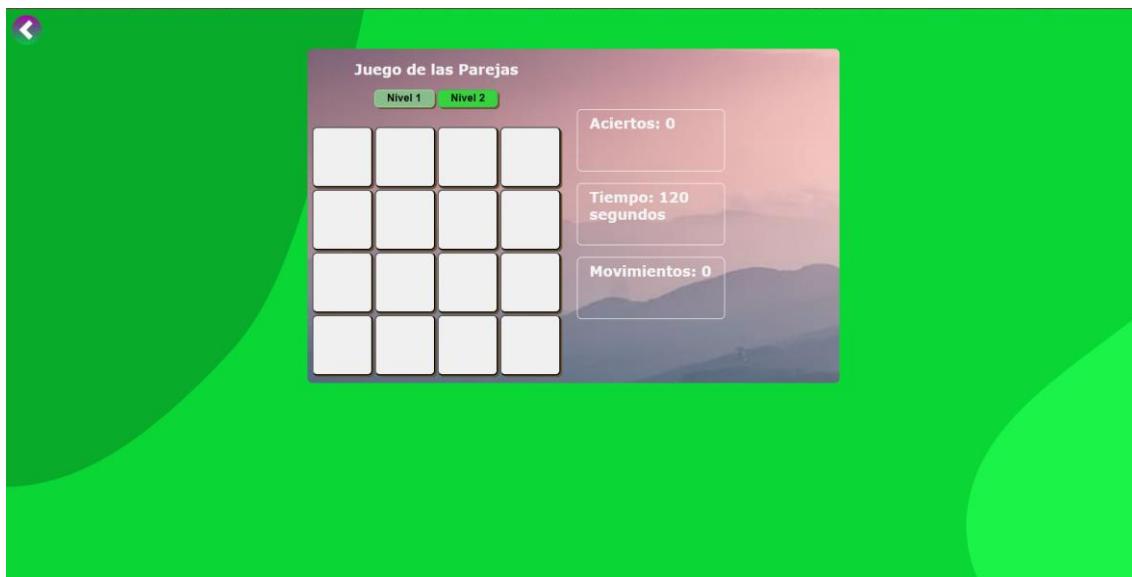


Ilustración 28: Interfaz Parejas

El usuario, al comenzar la prueba, tiene tres opciones: volver al menú principal, cambiar la dificultad (donde se aumentará el tablero a uno de 6x6, el tiempo del temporizador y el número de parejas), o seleccionar una de las cartas del tablero. Si se selecciona una carta, ésta mostrará una imagen y el temporizador comenzará, impidiendo cambiar de nivel hasta que se complete el tablero o finalice el tiempo. A continuación, el usuario selecciona otra carta, desvelándose su contenido. Tras ello, se impide pulsar sobre cualquier carta por un periodo corto de tiempo, en la que se comparan los resultados de las dos cartas seleccionadas. Si son iguales, se mantendrán destapadas y el número de aciertos aumentará y subirá al marcador, mientras que, si no son iguales, se ocultaran las cartas seleccionadas. De cualquier modo, el temporizador seguirá en funcionamiento, el número de movimientos aumentará, y se desbloquearán todas las cartas para poder pulsarlas excepto, las parejas ya encontradas. Una vez se encuentran todas las parejas, finaliza la prueba.

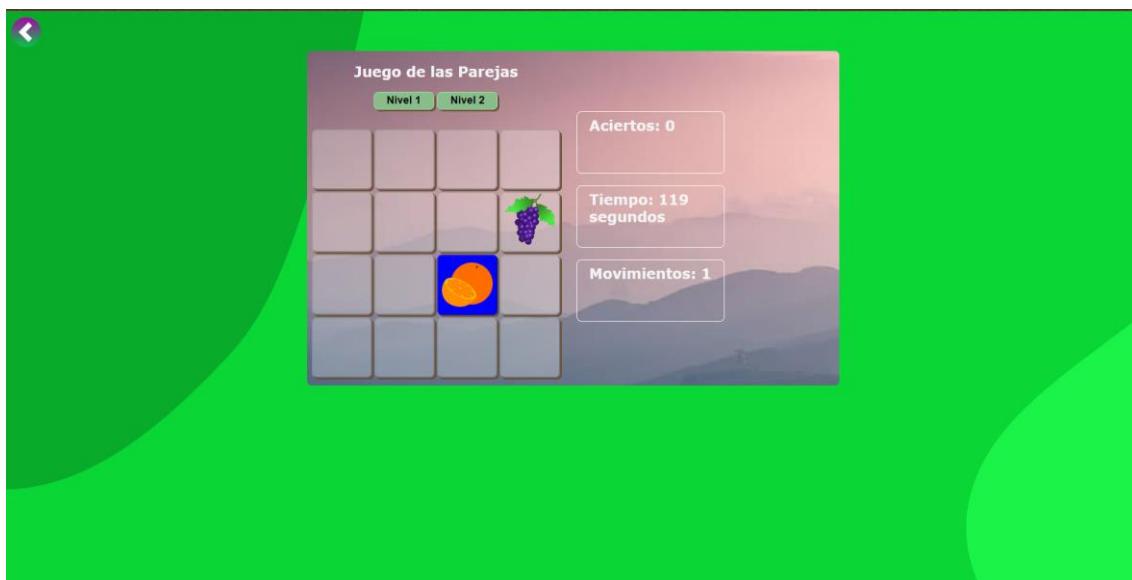


Ilustración 29: Usuario selecciona dos cartas distintas

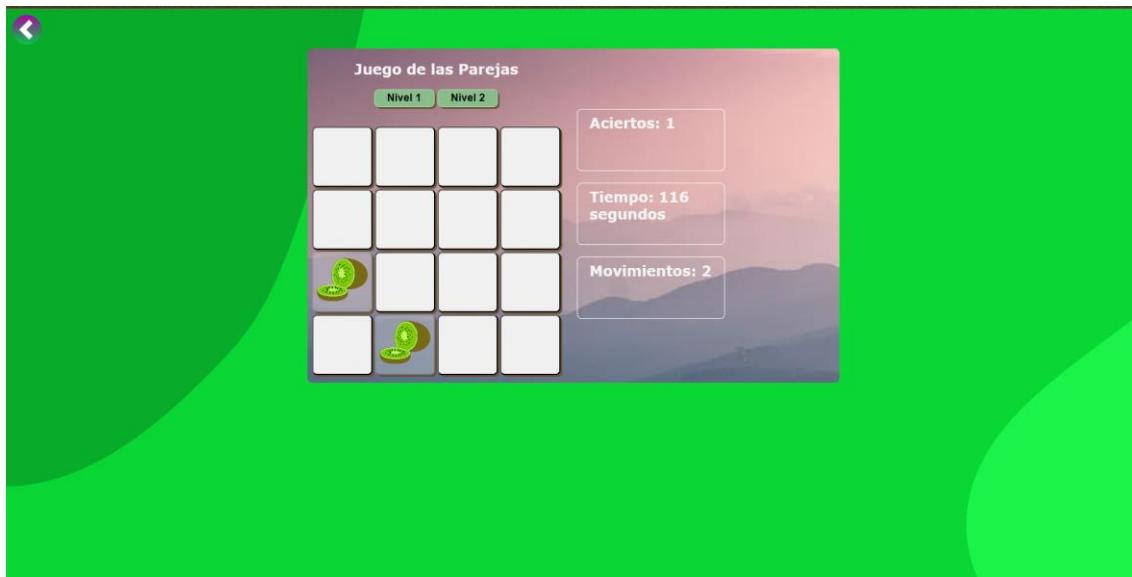


Ilustración 30: Usuario seleccionada una pareja

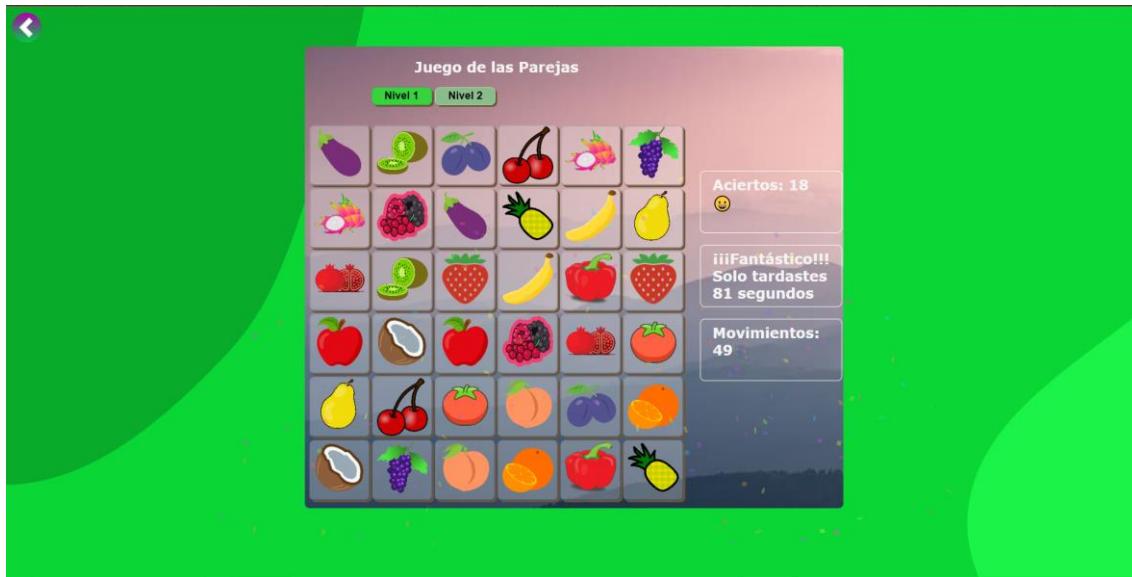


Ilustración 31: Usuario finaliza la prueba Parejas en la dificultad 2

4.1.3 Almacenamiento de Datos

Para esta prueba se requiere almacenar distintos parámetros cada vez que se realiza un movimiento, es decir, cuando se seleccionan dos cartas. Estos parámetros están compuestos por los movimientos realizados, el tiempo empleado, el usuario, el nivel de dificultad, número de aciertos, las dos casillas seleccionadas, la casilla válida para formar una pareja con respecto a la primera carta seleccionada, el tablero y el PPI de la pantalla.

```
public class Parejas {  
    1 usage  
    @JsonProperty("usuario")  
    private String usuario;  
    1 usage  
    @JsonProperty("tiempo")  
    private double tiempo=0.0;  
    1 usage  
    @JsonProperty("movimientos")  
    private int movimientos=0;  
    1 usage  
    @JsonProperty("nivel")  
    private int nivel=0;  
    1 usage  
    @JsonProperty("aciertos")  
    private int aciertos=0;  
    1 usage  
    @JsonProperty("casillaSelecciona1")  
    private Casilla casillaSelecciona1;  
    1 usage  
    @JsonProperty("casillaSelecciona2")  
    private Casilla casillaSelecciona2;  
    1 usage  
    @JsonProperty("casillaValida")  
    private Casilla casillaValida;  
    1 usage  
    @JsonProperty("ppi")  
    private double ppi;  
    1 usage  
    @JsonProperty("tablero")  
    private List<Integer> tablero;
```

Ilustración 32: Parámetros Parejas

4.2 Prueba 2, Arrastra y Suelta

4.2.1 Diseño

El usuario puede seleccionar un objeto y arrastrarlo a cualquier zona. Solo puede depositarlo en una de las cajas disponible. Cuando deja de pulsar el objeto en una de las cajas, el sistema compara el objeto con la caja. Si no son iguales, deja el objeto donde esta, mientras que, si son iguales, el sistema deposita el elemento dentro de la caja sumando un acierto. En ambos casos, se enviará información al servicio REST. Cuando se llegue a cinco aciertos, se cambiará de nivel. Este nuevo nivel es exactamente igual al anterior, a diferencia de los objetos que se encuentran en él, el número de objetos y los valores de las cajas. Cuando se llegue a diez aciertos, la prueba finalizará.

Toda esta lógica viene representada en un diagrama de alto nivel como se muestra en la ilustración 33.

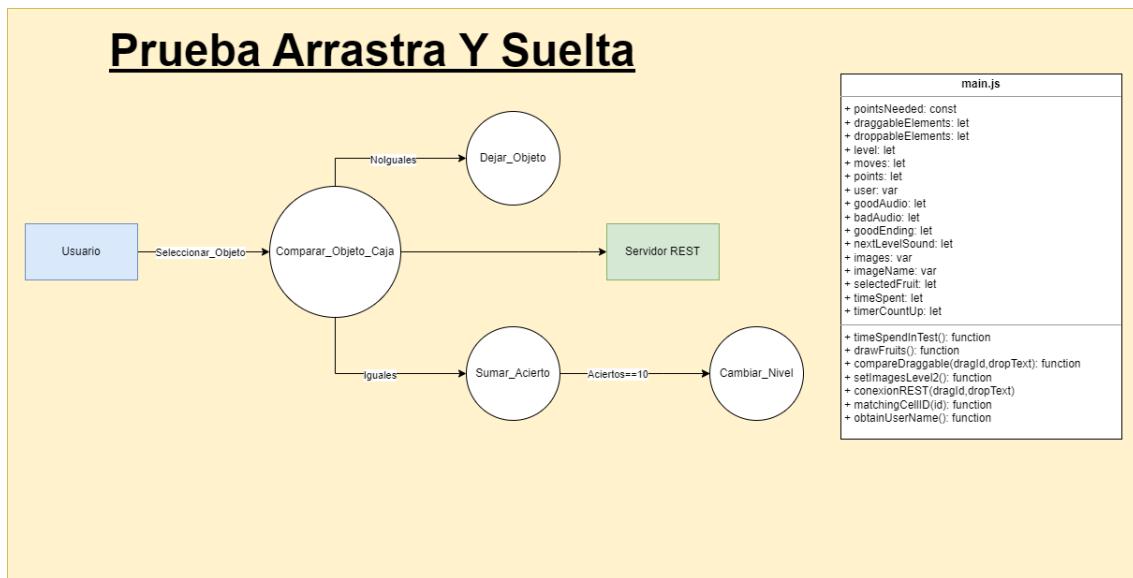


Ilustración 33: Diagramas Arrastra Y Suelta

Respecto al diseño de bajo nivel de la ilustración 33, se empezará hablando de las variables.

En primer lugar, se necesita guardar el objeto que se está arrastrando, y el elemento donde se quiere depositar.

En segundo lugar, se deben guardar el nivel, movimientos realizados, aciertos, las imágenes y nombre de las imágenes.

En tercer lugar, variables para manejar el tiempo, y en último lugar, variables de los efectos de sonido.

Sobre los métodos, “timeSpendInTest()” sirve para comenzar el contador del tiempo transcurrido en la prueba, ”drawFruits()” dibuja los objetos así como las cajas y les añade la capacidad de arrastrar y soltar, ”compareDraggable(dragId,dropText)” compara el objeto con la caja donde se desea depositar, ”setImagesLevel2()” cambia las imágenes de los objetos en el nivel 2, ”conexionREST()” y ”matchingCellID()” sirven para enviar los parámetros necesarios al servicio REST, y ”obtainUserName()” obtiene el nombre del usuario realizando la prueba.

4.2.2 Interfaz y Uso

La prueba muestra por pantalla las instrucciones: un botón para volver al menú principal , una lista de objetos, y una lista de cajas. Esta lista de objetos en primera instancia está compuesta de frutas mientras que las cajas aparecen el nombre de cada una de ellas. En cambio, en el segundo nivel, los objetos son verduras y las cajas contienen los nombres de cada una de ellas.



Ilustración 34: Interfaz Arrastra Y Suelta

El usuario, para empezar, tiene dos opciones: volver al menú principal o seleccionar y arrastrar un objeto. Si selecciona un objeto y lo mantiene pulsado, podrá moverlo por la pantalla. Si lo suelta en cualquier sitio que no sea dentro del área correspondiente, el objeto se quedará en su sitio inicial. En cambio, si lo suelta dentro del área de una caja, si la imagen coincide con el nombre, el objeto se moverá y se mantendrá en la caja. En caso contrario, el objeto permanecerá en su lugar inicial.

Cuando todos los objetos estén dentro de una caja, automáticamente comenzará el segundo nivel donde el usuario tiene la misma misión que en el primer nivel. Al acabar el segundo nivel, finaliza la prueba.



Ilustración 35: Usuario arrastra la imagen de la naranja en su caja



Ilustración 36: Usuario completa el primer nivel de Arrastra Y Suelta



Ilustración 37: Interfaz del segundo nivel de Arrastra Y Suelta

4.2.3 Almacenamiento de Datos

Para esta prueba, se requiere almacenar distintos parámetros cada vez que se realiza un movimiento, es decir, cuando se arrastra un objeto y se suelta en una caja. Estos parámetros están compuestos por los movimientos realizados, el tiempo empleado, el usuario, el nivel, número de aciertos, objeto escogido, la caja donde se soltó el elemento, la caja válida donde se debe soltar el objeto, y el PPI de la pantalla.

```

public class ArrastraYSuelta {
    2 usages
    @JsonProperty("usuario")
    private String usuario;
    2 usages
    @JsonProperty("tiempo")
    private double tiempo=0.0;
    2 usages
    @JsonProperty("movimientos")
    private int movimientos=0;
    2 usages
    @JsonProperty("nivel")
    private int nivel=0;
    2 usages
    @JsonProperty("aciertos")
    private int aciertos=0;
    2 usages
    @JsonProperty("nombreElementoEscogido")
    private String nombreElementoEscogido;
    2 usages
    @JsonProperty("casillaSelecciona")
    private Casilla casillaSelecciona;
    2 usages
    @JsonProperty("casillaValida")
    private Casilla casillaValida;
    2 usages
    @JsonProperty("listaObjetos")
    private List<String> listaObjetos;
    2 usages
    @JsonProperty("listaCajas")
    private List<String> listaCajas;
    2 usages
    @JsonProperty("ppi")
    private double ppi;
}

```

Ilustración 38: Parámetros Arrastra Y Suelta

4.3 Prueba 3, Recuerda Palabras

4.3.1 Diseño

El usuario puede cambiar el nivel de dificultad de la prueba, reiniciarla, esconder o desvelar el temporizador disponible, cambiar el tiempo del temporizador de la prueba, o comenzar la prueba.

Al iniciar la prueba, el temporizador empieza a contar marcha atrás y se muestran unas palabras. Al llegar a cero aparece una nueva tabla que contiene las anteriores y otras de relleno. El usuario puede pulsar sobre las celdas de la tabla. Una vez hecho esto, se enviará información al servicio REST y se guardará la palabra seleccionada. Cuando se hayan escogido todas las palabras permitidas por el nivel de dificultad, el sistema comprobará si son las palabras buscadas o no, finalizando la prueba.

Toda esta lógica viene representada en un diagrama de alto nivel en la ilustración 39.

Prueba Recuerda Palabras

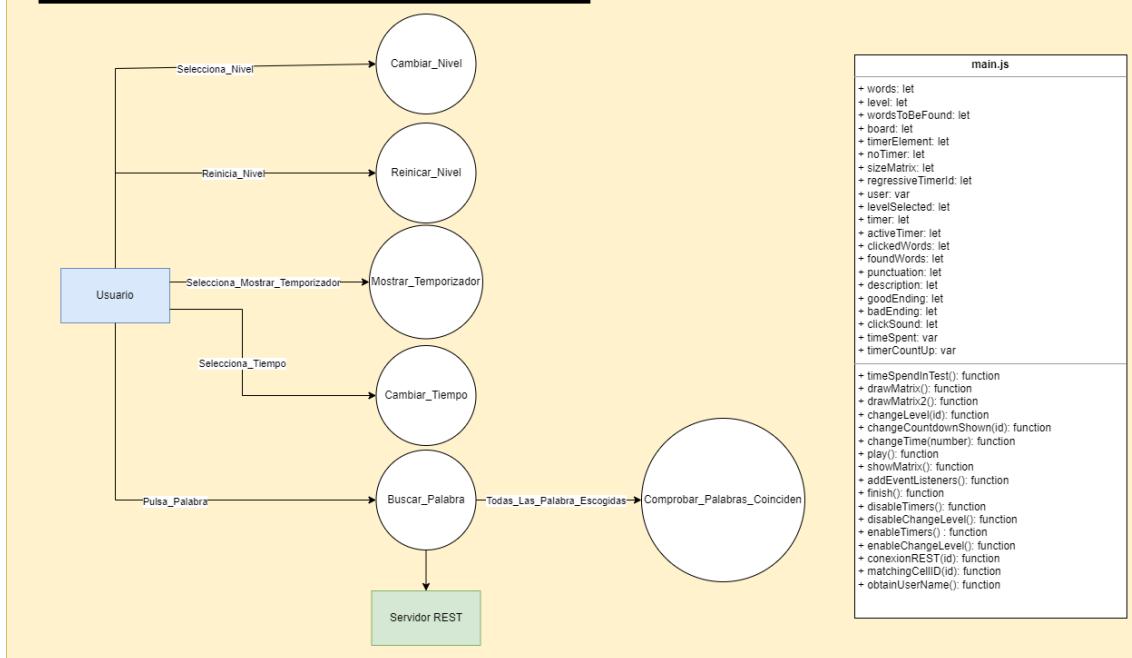


Ilustración 39: Diagramas Recuerda Palabras

Respecto al diseño de bajo nivel de la ilustración 39, se empezará hablando de las variables.

En primer lugar, se necesita almacenar todas las posibles palabras que pueden aparecer en la prueba, así como, dentro de las palabras disponibles, una variable que contenga las palabras a buscar donde su tamaño será correspondiente al de nivel de dificultad y en serán aleatorias.

En segundo lugar, se deben guardar el nivel, número de palabras seleccionadas, las palabras encontradas, tamaño de la tabla, y si el temporizador es visible o no.

En tercer lugar, se necesitan variables para acceder a los elementos HTML para poder modificar el temporizador, el tablero, puntuación (número de palabras seleccionadas del total disponible y número de aciertos), los botones de cambio de nivel y tiempo, y la descripción de lo que debe hacer el usuario.

En cuarto lugar, variables para manejar el tiempo transcurrido en la prueba, y en último lugar, variables de los efectos de sonido.

Sobre los métodos, “timeSpendInTest()” sirve para comenzar el contador del tiempo transcurrido en la prueba, “drawMatrix()” dibuja las palabras a buscar, “drawMatrix2()” dibuja el tablero que contiene las palabras a buscar y otras palabras irrelevantes, “changeLevel(id)” cambia la dificultad de la prueba, “changeCountdownShown(id)” esconde o muestra el temporizador, “changeTime(number)” cambia el tiempo del temporizador, “play()” sirve para comenzar la prueba, “showMatrix()” muestra por pantalla el tablero, “addEventListeners()” añade a las palabras del tablero funciones cuando se pulsa sobre una de ellas, “finish()” comprueba si se han encontrado todas las palabras o no, “disableTimers()” inhabilita los botones de cambio de tiempo, “disableChangeLevel()” inhabilita los botones de cambio de dificultad, “enableTimers()” habilita los botones de cambio de tiempo, “enableChangeLevel()” habilita los botones de cambio de tiempo, “conexionREST(id)”,

“matchingCellID(id)” sirven para enviar los parámetros necesarios al servicio REST, y “obtainUserName()” obtiene el nombre del usuario realizando la prueba.

4.3.2 Interfaz y Uso

La prueba muestra por pantalla un temporizador que puede aparecer con números o no dependiendo de la selección hecha y con distintos valores iniciales dependiendo del número de segundos seleccionados para el temporizador. También aparecen unas instrucciones, un botón para volver al menú principal ↺ y un tablero con casillas en blanco donde el número de casillas cambia respecto a la dificultad seleccionada.

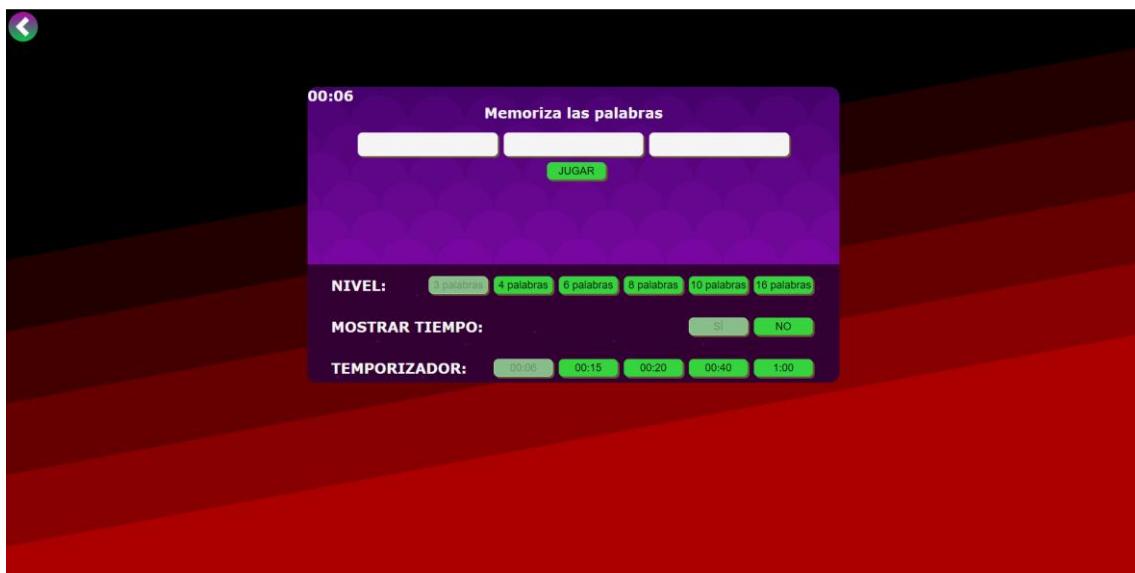


Ilustración 40: Interfaz Recuerda Palabras

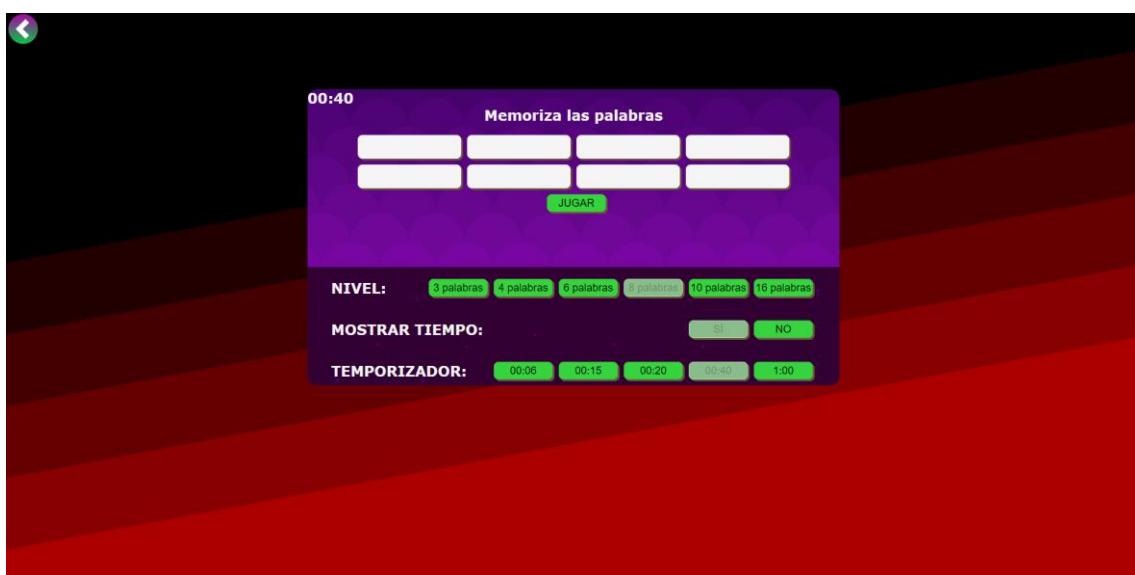


Ilustración 41: Interfaz Recuerda Palabras con 8 palabras a memorizar, y 40 segundos

El usuario, para empezar, tiene cinco opciones: volver al menú principal, cambiar el número de palabras a recordar y por tanto la dificultad, visualizar o no el temporizador, cambiar el tiempo disponible para visualizar las palabras a

recordar, y comenzar la prueba. Cuando haya hecho las configuraciones queridas y comience la prueba con el botón “JUGAR”, aparecerán en pantalla una serie de palabras y el temporizador comenzara su ejecución, donde el individuo tiene un determinado tiempo para memorizar las palabras. Una vez acabe el temporizador, aparecerán una lista nueva de palabras. Aquí, el usuario debe intentar seleccionar aquellas palabras que tenía que memorizar obviando el resto. Cuando seleccione las palabras, la prueba habrá acabado en un éxito si selecciono todas las palabras correctas y en fracaso si no.

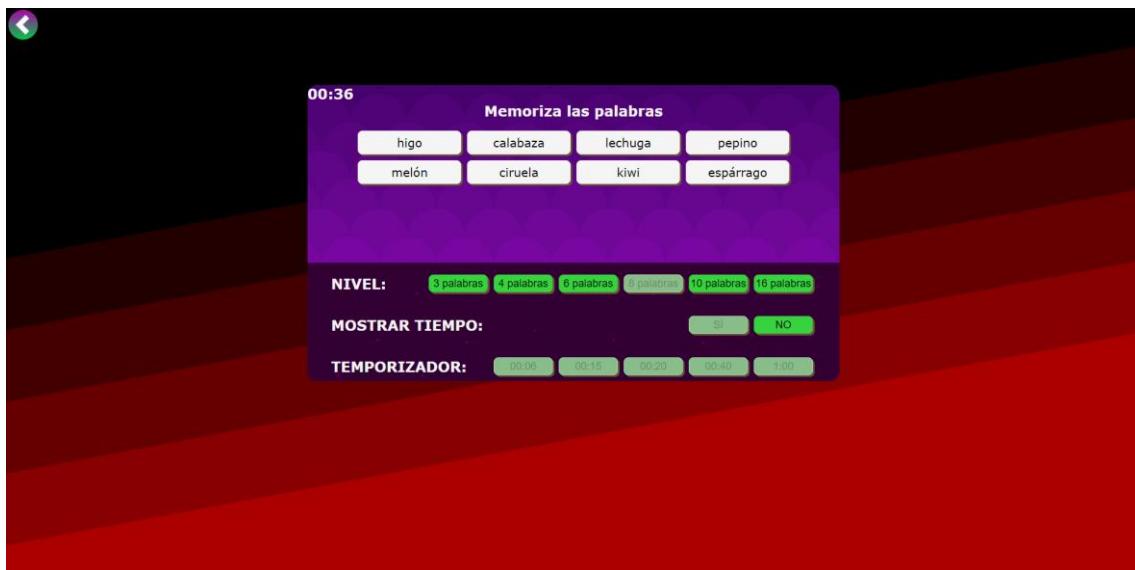


Ilustración 42: Recuerda Palabras en ejecución con 8 palabras

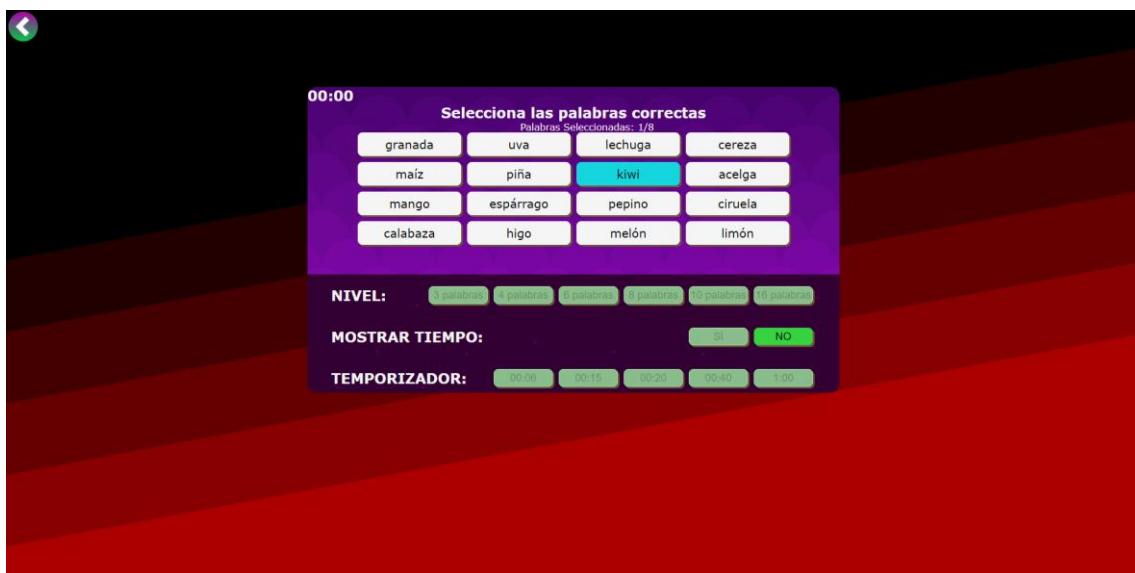


Ilustración 43: Tablero de las palabras a encontrar



Ilustración 44: Finalización de Recuerda Palabras con la puntuación

4.3.3 Almacenamiento de Datos

Para esta prueba, se requiere almacenar distintos parámetros cada vez que se realiza selecciona una palabra. Estos parámetros están compuestos por el número de palabras pulsadas, el tiempo empleado, el usuario, el nivel de dificultad, número de aciertos, casilla de la palabra escogida, las casillas donde se encuentran las palabras a buscar, y el PPI de la pantalla.

```
public class Memoria {
    2 usages
    @JsonProperty("usuario")
    private String usuario;
    2 usages
    @JsonProperty("tiempo")
    private double tiempo=0.0;
    2 usages
    @JsonProperty("movimientos")
    private int movimientos=0;
    2 usages
    @JsonProperty("nivel")
    private String nivel;
    2 usages
    @JsonProperty("aciertos")
    private int aciertos=0;
    2 usages
    @JsonProperty("temporizador")
    private String temporizador;
    2 usages
    @JsonProperty("casillaSelecciona")
    private Casilla casillaSelecciona;
    2 usages
    @JsonProperty("casillasValida")
    private List<Casilla> casillasValida;
    2 usages
    @JsonProperty("ppi")
    private double ppi;
    2 usages
    @JsonProperty("tablero")
    private List<String> tablero;
```

Ilustración 45: Parámetros Recuerda Palabras

4.4 Prueba 4, Noquea las Torres

4.4.1 Diseño

El usuario puede activar o desactivar una música de fondo, reiniciar la prueba o seleccionar una casilla de un enemigo. Cuando se hace esta acción, el sistema compara las fuerzas del enemigo y del jugador. Si el jugador tiene fuerza menor, el usuario perderá, en cambio, si su fuerza es mayor, derrotara al enemigo y su fuerza aumentara con la del enemigo. De cualquier modo, se enviará información al servicio REST. Cada torre tiene una serie de enemigos a derrotar. Si derrotas a todos, avanza a la siguiente torre hasta llegar a la última que se diferencia del resto por ser la más alta y contener una casilla especial donde alberga una gema. Si el individuo llega a obtener la gema, el usuario habrá superado la prueba con éxito.

Toda esta lógica viene representada en un diagrama de alto nivel como se puede ver en la ilustración 46.

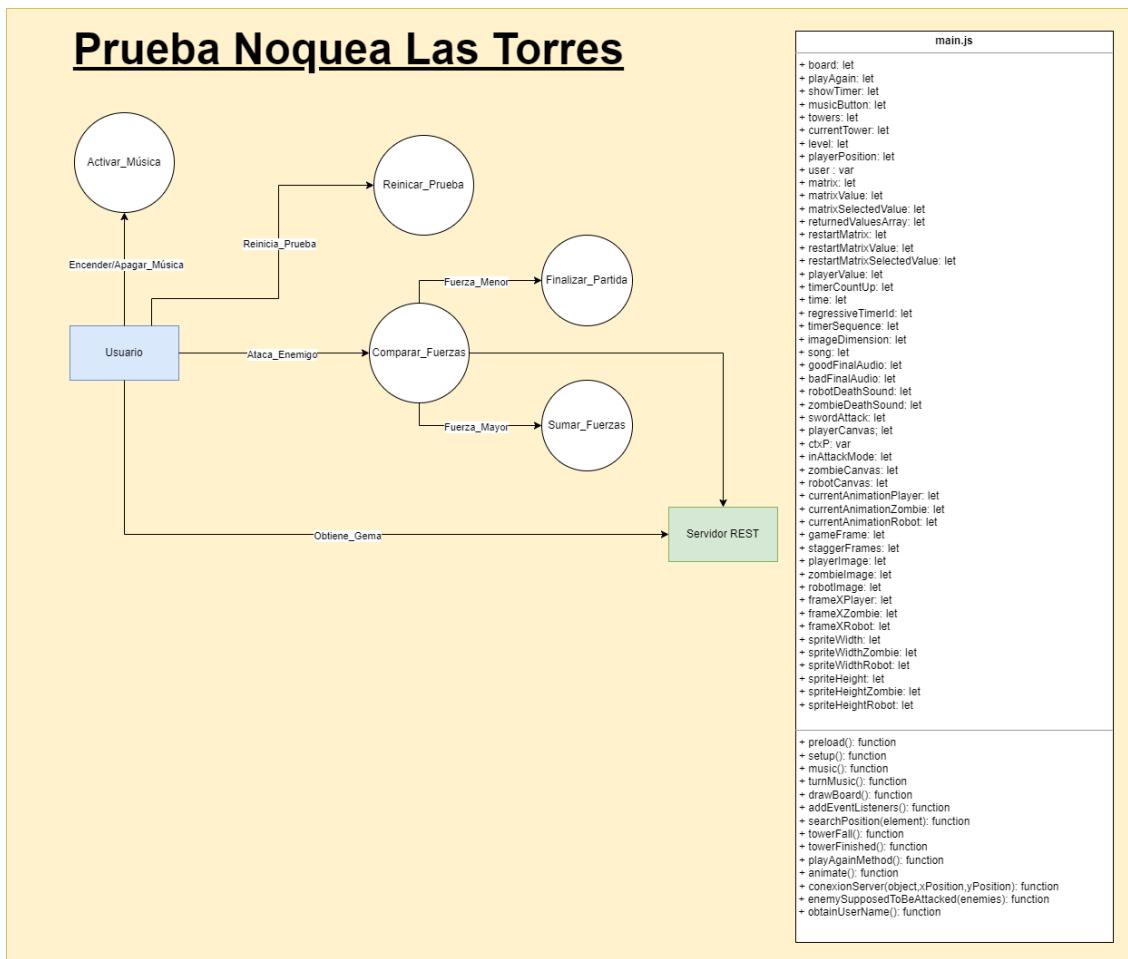


Ilustración 46: Diagramas Noquea Las Torres

Respecto al diseño de bajo nivel de la ilustración 46, se empezará hablando de las variables.

En primer lugar, se necesitan variables para acceder a los elementos HTML para poder modificar el tiempo empleado en la prueba, el botón de reintentar, las torres, el botón de la música, las casillas de los enemigos, y la casilla del jugador.

En segundo lugar, se deben guardar el número de torres, el número de torre en la que está el jugador y el valor del jugador.

En tercer lugar, variables para manejar el contenido de las torres, elemento que debe ir dentro de cada casilla, su valor, y variables para reiniciar las torres.

En cuarto lugar, variables para manejar el tiempo transcurrido en la prueba, y variables para los efectos de sonido.

En último lugar, variables para controlar las animaciones tanto de los enemigos como del jugador.

Sobre los métodos, “preload()”, “setup()”, y “music()” son funciones para configurar la canción, y “turnMusic()” sirve para encender o apagar la música de fondo. Seguidamente, esta “drawBoard()” que dibuja las torres y al jugador, “addEventListeners()” añade a los enemigos y a la gema funciones cuando se pulsa sobre uno de ellos, “searchPosition(element)” busca la posición de un elemento en las torres, “towerFall()” avanza al jugador a la siguiente torre, “towerFinished()” detecta si se derrotaron a todos los enemigos de la torre actual, y “playAgainMethod()” permite reintentar la prueba. Por último, “animate()” permite animar tanto al jugador como a los enemigos y a su vez, cambiar de animación cuando sea oportuno, “conexionServer(object,xPosition,yPosition)”, “enemySupposedToBeAttacked(enemies)”, sirven para enviar los parámetros necesarios al servicio REST, y “obtainUserName()” obtiene el nombre del usuario realizando la prueba.

4.4.2 Interfaz y Uso

La prueba muestra por pantalla el tiempo empleado en la prueba, un botón para encender o apagar la música de fondo, unas indicaciones, un botón para volver al menú principal , un caballero y unos zombis ambos con un número sobre sus cabezas. El caballero es el jugador que comienza con una fuerza de valor 2 mientras que los zombis son los enemigos por batir para avanzar en las torres. A parte de los zombis, también se encuentran robots como enemigos, y una gema roja en la última torre, la torre con la mayor altura. Además, cada enemigo y jugador tiene diferentes animaciones, una cuando son derrotados, una cuando siguen vivos, y adicionalmente, el jugador tiene un ataque de espada.



Ilustración 47: Interfaz Noquea Las Torres

El usuario, para empezar, tiene tres opciones: volver al menú principal, encender o apagar la música, o seleccionar a un enemigo. Cuando se selecciona al enemigo, si el poder del jugador es mayor que el enemigo, lo derrotará y obtendrá su poder. Cuando todos los enemigos de una torre hayan caído, el sistema automáticamente avanzará el usuario a la siguiente torre donde tendrá que realizar la misma acción. Cuando llegue al final, encontrará una gema, si consigue pulsar sobre esa gema habiendo derrotado a todos los enemigos, el individuo habrá completado la prueba con éxito. En caso de que un enemigo lo venza porque tiene más poder, la prueba habrá finalizada en fracaso.



Ilustración 48: Enemigo derrotado



Ilustración 49: Jugador Derrotado, Prueba Finalizada



Ilustración 50: Cuarta Torre



Ilustración 51: Finalización de Noquea Las Torres

4.4.3 Almacenamiento de Datos

Para esta prueba, se guardan los parámetros cada vez que se ataca a un enemigo. Se guardan el tiempo empleado, el usuario, las coordenadas x e y donde el usuario ha pulsado, las coordenadas x e y de los puntos validos (el punto arriba a la izquierda, y el punto abajo a la derecha de la casilla que se debe seleccionar), el PPI, el nivel (la torre en la que se está), y la fuerza del jugador.

```

public class Torre {
    2 usages
    @JsonProperty("tiempo")
    private double tiempo=0.0;
    2 usages
    @JsonProperty("usuario")
    private String usuario;
    2 usages
    @JsonProperty("casillaPulsada")
    private Casilla casillaPulsada;
    2 usages
    @JsonProperty("casillaValida")
    private Casilla casillaValida;
    2 usages
    @JsonProperty("ppi")
    private double ppi;
    2 usages
    @JsonProperty("nivel")
    private int nivel=0;
    2 usages
    @JsonProperty("fuerzaJugador")
    private int fuerzaJugador=0;
    2 usages
    @JsonProperty("tablero")
    private List<List<String>> tablero;

```

Ilustración 52: Parámetros Noquea Las Torres

4.5 Prueba 5, CentroBola

4.5.1 Diseño

El usuario puede pulsar sobre la bola que este en pantalla y reiniciar la prueba. Cuando lo hace, el sistema calcula la distancia desde el punto pulsado hasta el centro enviando la información al servicio REST, comentando el resultado al usuario y apareciendo otro círculo cuando lo deseé el usuario. Una vez que el usuario haya pulsado sobre todas las bolas de la prueba, esta habrá finalizado, y el individuo obtendrá su resultado.

Toda esta lógica viene representada en un diagrama de alto nivel como se muestra en la ilustración 53.

Prueba CentroBola

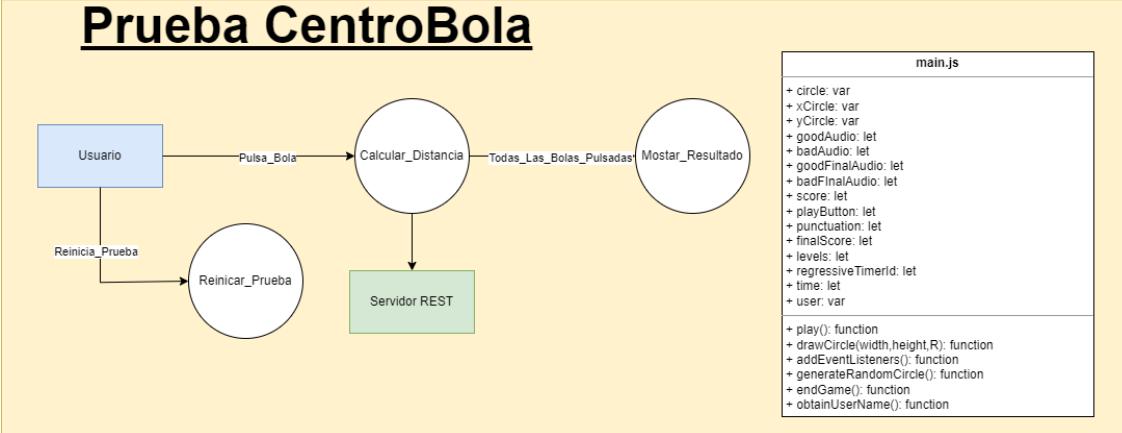


Ilustración 53: Diagramas CentroBola

Respecto al diseño de bajo nivel de la ilustración 53, se empezará hablando de las variables.

En primer lugar, se necesitan variables para acceder a los elementos HTML para poder modificar los círculos que aparecen en pantalla, la puntuación, número de bolas pulsadas, y el botón de iniciar/continuar la prueba.

En segundo lugar, se deben guardar las coordenadas del círculo, la puntuación final, el nivel (cuantos círculos han aparecido) y variables para manejar el tiempo.

En último lugar. Variables para los efectos de sonido.

Sobre los métodos, “play()” comienza la prueba así como el tiempo en la prueba, “drawCircle(width,height,R)” dibuja los círculos, “addEventListeners()” añade funciones cuando se pulsa sobre un círculo y envía los parámetros necesarios al servicio REST, “generateRandomCircle()” configura las dimensiones de las bolas y posición de las bolas, “endGame()” finaliza la prueba, y “obtainUserName()” obtiene el nombre del usuario realizando la prueba.

4.5.2 Interfaz y Uso

La prueba muestra por pantalla las instrucciones de la prueba, un botón y un botón para volver al menú principal . Cuando comienza la prueba, un círculo de unas dimensiones aleatorias y en una posición aleatoria aparecerá en pantalla y cuando se pulse sobre él, la puntuación obtenida se mostrará en pantalla, de color azul si estuvo cerca del centro, y en color rojo si no fue el caso.

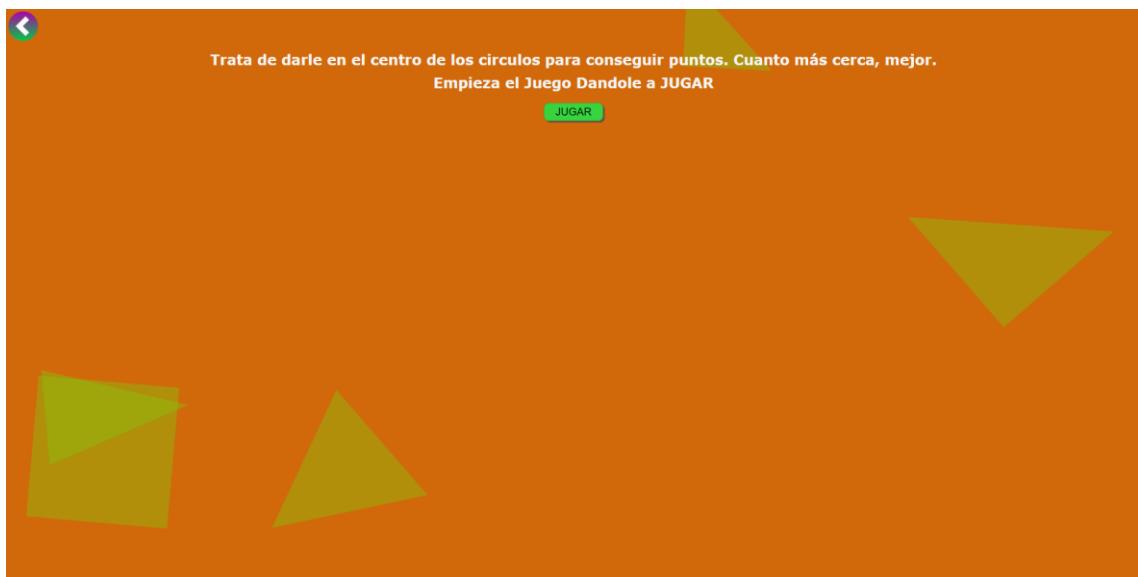


Ilustración 54: Interfaz CentroBola

El usuario, para empezar, tiene dos opciones: volver al menú principal, o comenzar la prueba. Para ello, si le da el botón de “JUGAR”, una bola aparecerá en pantalla. El usuario debe pulsarlo lo más cerca posible del centro. Una vez hecho, saltará la puntuación, y el sistema le dirá cuántas bolas quedan por pulsar. Después, el usuario debe darle al botón “SIGUIENTE” para continuar la prueba, donde aparecerá otro círculo y el usuario debe hacer lo mismo que con el anterior. Cuando pulse todos los círculos, obtendrá un resultado final en color rojo o azul. Si está en azul, significa que la prueba se realizó con éxito, si sale en rojo, fracaso la prueba.

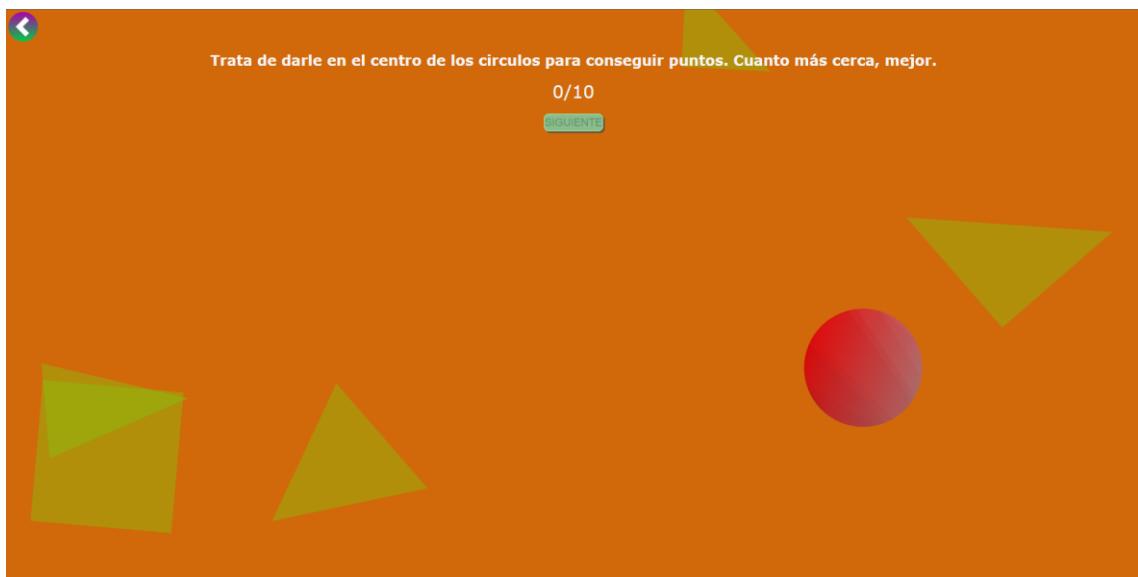


Ilustración 55: Bola aparece aleatoriamente

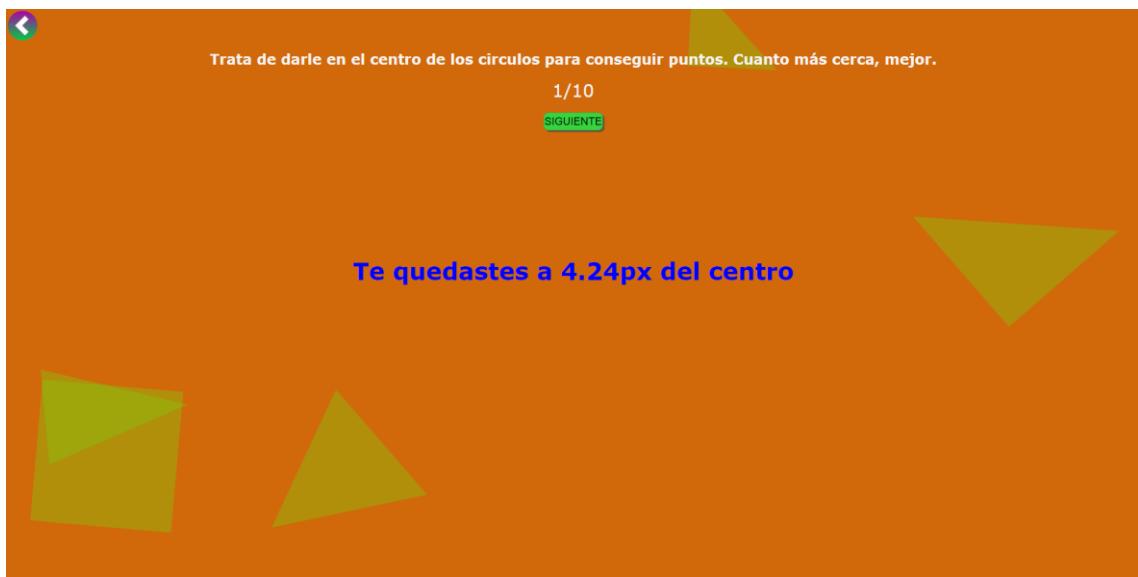


Ilustración 56: Usuario pulsa cerca del centro de la bola

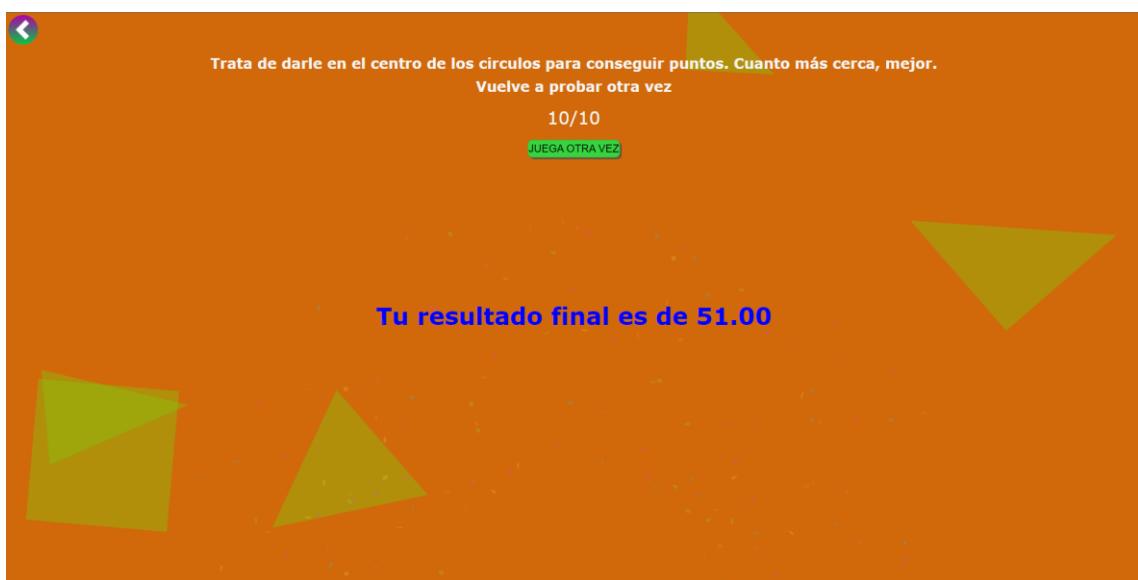


Ilustración 57: Finalización CentroBola

4.5.3 Almacenamiento de Datos

Para esta prueba, se guardan los parámetros cada vez que se pulsa sobre una bola. Se guardan el tiempo empleado, el usuario, el nivel, la distancia desde el punto pulsado hasta el centro, el punto pulsado, y el PPI.

```

public class CentroBola {
    3 usages
    @JsonProperty("tiempo")
    private double tiempo=0.0;
    3 usages
    @JsonProperty("usuario")
    private String usuario;
    3 usages
    @JsonProperty("puntoPulsado")
    private Coordenadas puntoPulsado;
    3 usages
    @JsonProperty("distancia")
    private double distanciaRespectoAlMedio;
    2 usages
    @JsonProperty("ppi")
    private double ppi;
    2 usages
    @JsonProperty("nivel")
    private int nivel=0;
}

```

Ilustración 58: Parámetros CentroBola

4.6 Prueba 6, MetaBola

4.6.1 Diseño

El usuario puede realizar tres acciones, comenzar la prueba, cambiar la dificultad, o detener la bola. Cambiando de dificultad, el trayecto de la bola es modificado. Una vez cambiado la dificultad e inicializo la prueba, cuando el usuario detenga la bola, el sistema finalizara la prueba, calculando la distancia a la que se ha quedado la bola respecto al final del trayecto y enviará información al servicio REST. Al iniciar la prueba, se mostrará el camino que seguirá la bola. Al cabo de unos segundos ambos desaparecerán.

Toda esta lógica viene representada en un diagrama de alto nivel en la ilustración 59.

Prueba MetaBola

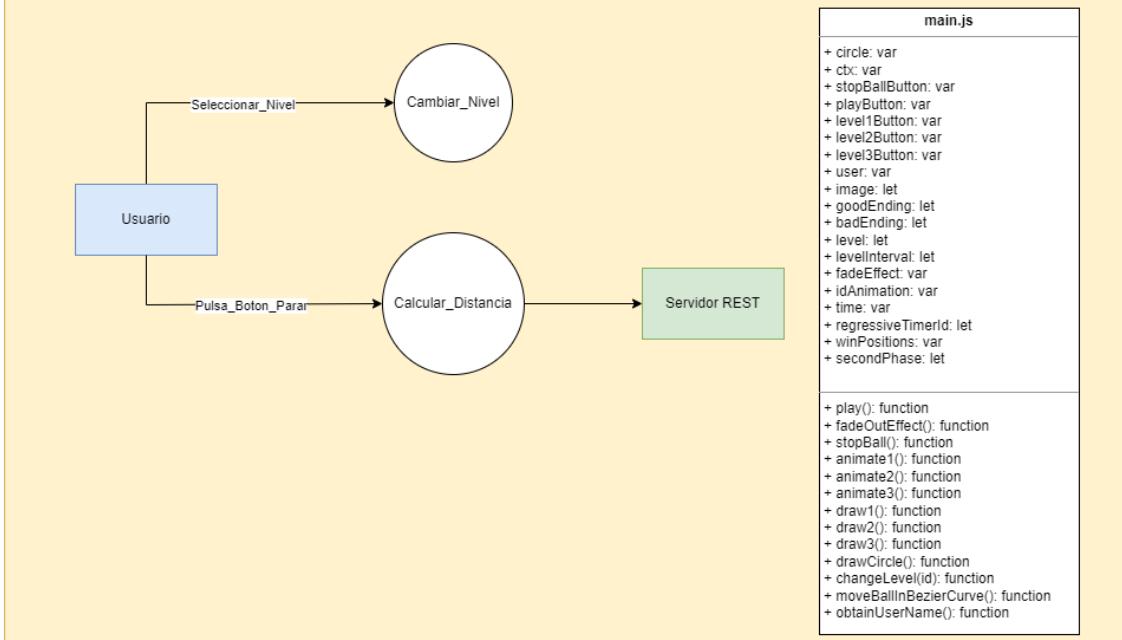


Ilustración 59: Diagramas MetaBola

Respecto al diseño de bajo nivel de la ilustración 59, se empezará hablando de las variables.

En primer lugar, se necesitan variables para acceder a los elementos HTML para poder modificar los botones de cambio de dificultad, así como la bola y el camino que seguirá.

En segundo lugar, se deben guardar el nivel de dificultad, el intervalo de segundos en los que se producirá el efecto de desvanecimiento y variables para manejar el tiempo. En último lugar. variables para los efectos de sonido.

Sobre los métodos, “play()” inicializa la prueba y “fadeOutEffect()” realiza el efecto de desvanecimiento de la bola y del camino. La función “stopBall()” detiene la bola, calcula la distancia desde la bola hasta la meta y envía los parámetros necesarios al servicio REST. Las funciones “animate1()”, “animate2()”, “animate3()” realizan el movimiento de la bola en los distintos niveles respectivamente, “draw1()”, “draw2()” y “draw3()” dibujan los caminos de los distintos niveles, “drawCircle()” dibuja la bola en pantalla, “changeLevel(id)” cambia el nivel de la prueba, “moveBallInBezierCurve()” ayuda al movimiento de la esfera en cuanto a las curvas, y “obtainUserName()” obtiene el nombre del usuario realizando la prueba.

4.6.2 Interfaz y Uso

La prueba muestra por pantalla una serie de botones, dos deshabilitados y tres hábiles, además de un botón para volver al menú principal y unas instrucciones. Los botones hábiles permiten cambiar la dificultad de la prueba y cuando uno de ellos es elegido, se habilitará el botón de “JUGAR”. Después aparece un camino distinto y una bola recorriéndolo.

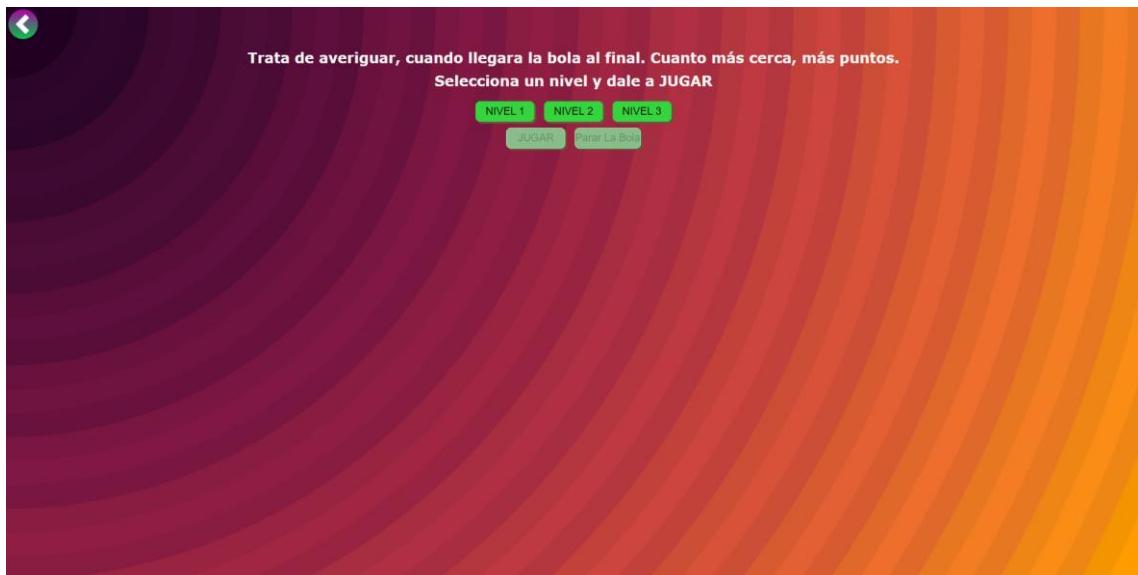


Ilustración 60: Interfaz MetaBola

El usuario, al comenzar la prueba, tiene dos opciones: elegir la dificultad de la prueba o volver al menú principal. Si selecciona un nivel, el usuario podrá comenzar la prueba. Cuando lo haga, aparecerá una trayectoria en negro con una bola que se mueve por dicho camino. Este camino depende del nivel elegido. A partir de ahora, el objetivo del usuario es detener la bola lo más cerca del final. Esta tarea se dificulta ya que, al cabo de unos segundos, tanto la bola como el camino empiezan a desvanecer hasta que desaparecen completamente. Cuando decida detener la bola, el sistema calculará la distancia a la que se quedó respecto a la meta, notificara al usuario si se quedó llegó a la meta o no, y mostrara el camino y la posición bola. Con ello finalizaría la prueba.

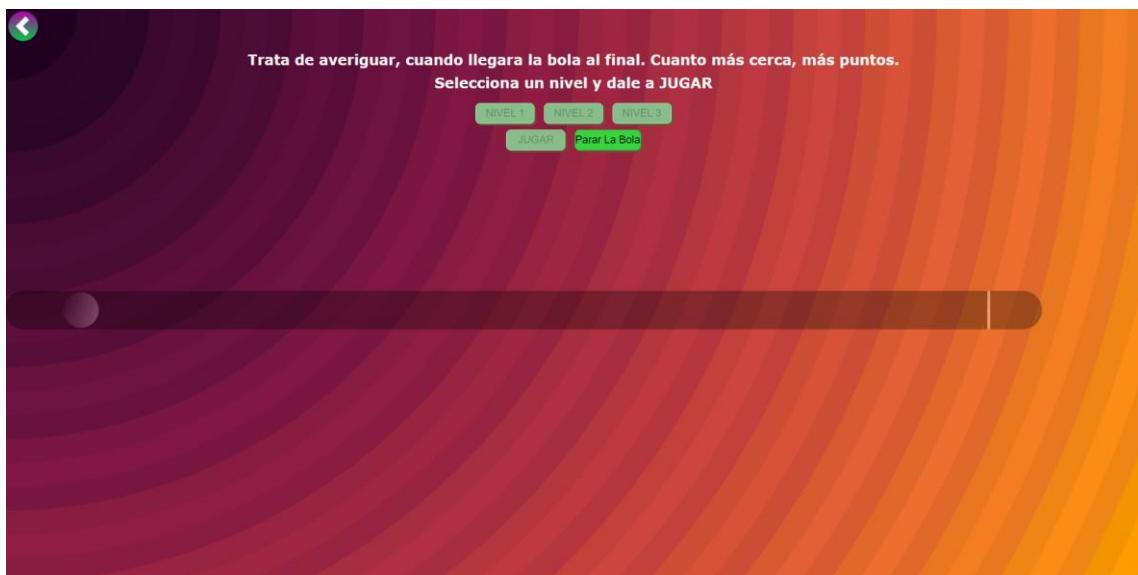


Ilustración 61: Nivel 1 de MetaBola

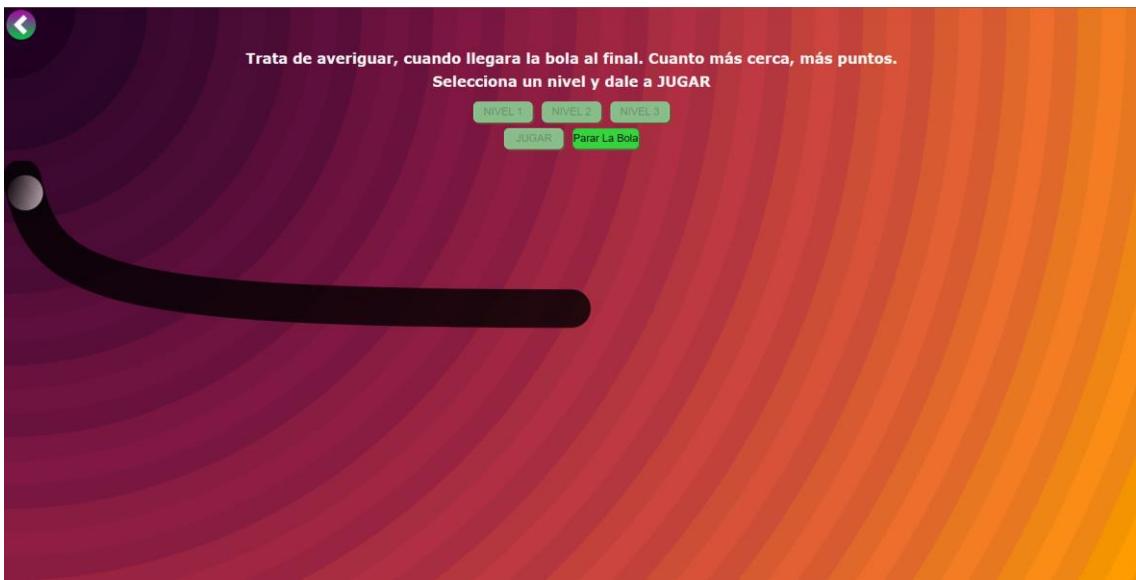


Ilustración 62: Nivel 2 de MetaBola

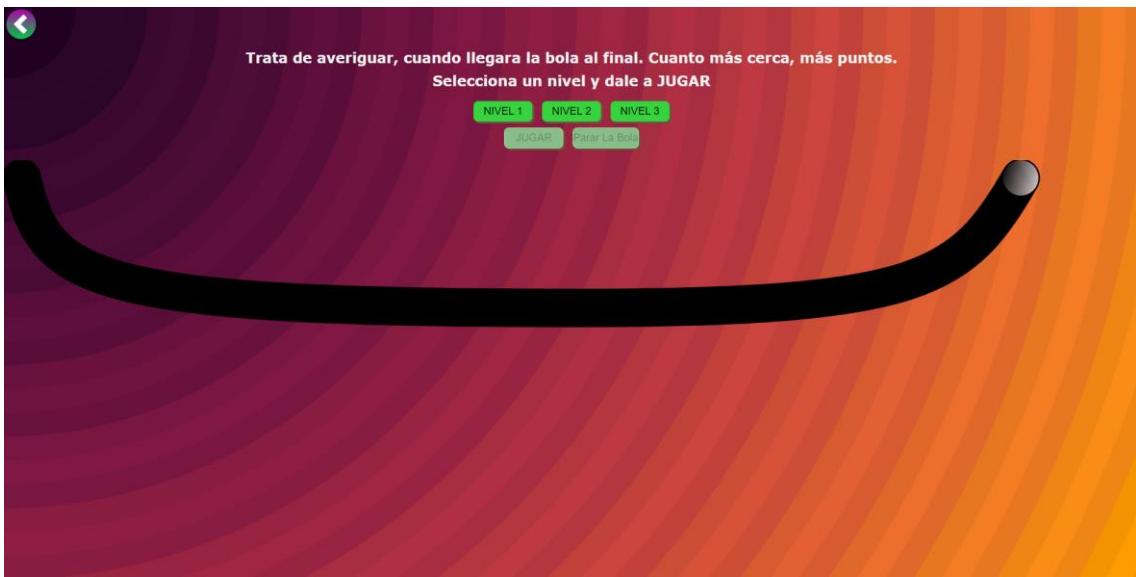


Ilustración 63: Nivel 3 de MetaBola con bola en la meta

4.6.3 Almacenamiento de Datos

Para esta prueba se requiere almacenar distintos parámetros cada vez que se detenga la bola. Estos parámetros están compuestos por el tiempo empleado, el usuario, si el usuario detuvo la bola o lo hizo el sistema, las coordenadas del punto donde se detuvo la bola, los puntos donde está la meta, la distancia respecto la bola y la meta, el ppi, y el nivel de dificultad.

```

public class MetaBola {
    2 usages
    @JsonProperty("tiempo")
    private double tiempo=0.0;
    2 usages
    @JsonProperty("usuario")
    private String usuario;
    2 usages
    @JsonProperty("usuarioDetuvoLaBola")
    private boolean usuarioDetuvoLaBola;
    2 usages
    @JsonProperty("puntoPulsado")
    private Coordenadas puntoPulsado;
    2 usages
    @JsonProperty("puntosValidos")
    private PuntosValidos puntosValidos;
    2 usages
    @JsonProperty("distancia")
    private double distanciaRespectoARespuestaMasCercana;
    2 usages
    @JsonProperty("ppi")
    private double ppi;
    2 usages
    @JsonProperty("nivel")
    private int nivel=0;
}

```

Ilustración 64: Usuario finaliza la prueba en la dificultad 2

4.7 Prueba 7, MOCA

La prueba MOCA, como se vio anteriormente, está compuesta por varias secciones. Por ello, se ha realizado un diseño para cada sección, intentando mantener en cada sección, una interfaz similar.

4.7.1 Diseño

Sección 1:

En esta sección, el usuario tiene cuatro opciones, dibujar sobre un área delimitada, borrar el dibujo, volver al menú principal o pasar a la siguiente sección.

MOCA Visuoespacial

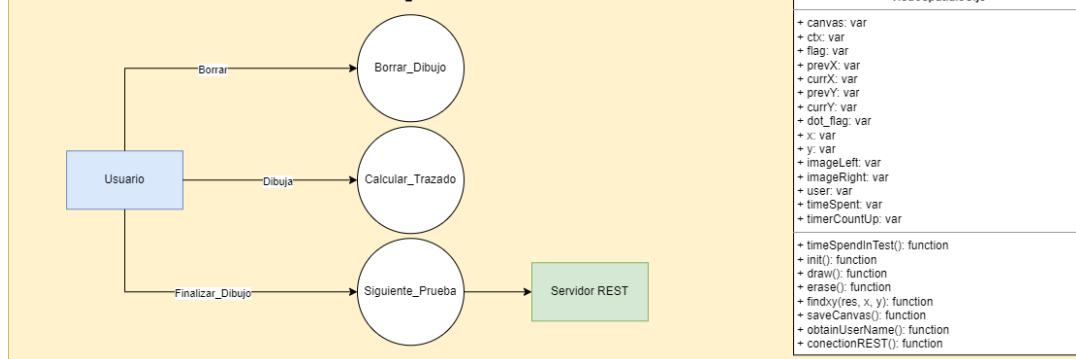


Ilustración 65: Diagramas MOCA Parte 1

Respecto al diseño de bajo nivel, se necesitan variables para manejar el área dibujable, variables para tener control del tiempo y para las imágenes del cubo y grafo.

Además, contiene funciones para inicializar el área de dibujo, así como las imágenes, para cuando se está dibujando, para borrar el dibujo, para controlar el tiempo, para obtener al usuario y por último para guardar los resultados y enviarlos al servicio REST.

Sección 2:

Esta sección es idéntica a la primera, pero cambia el objetivo, ya que se tiene que dibujar un reloj.

MOCA Dibujo Reloj

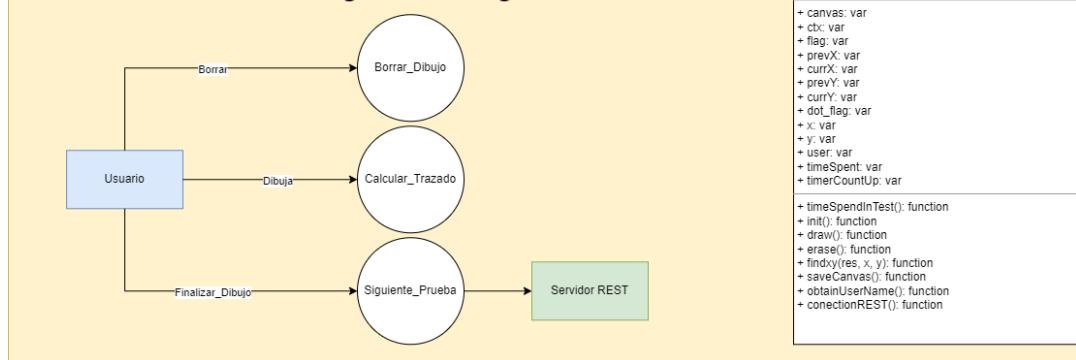


Ilustración 66: Diagramas MOCA Parte 2

Respecto al diseño de bajo nivel, se necesitan variables para manejar el área dibujable, y variables para tener control del tiempo.

Además, contiene funciones para inicializar el área de dibujo, así como las imágenes, para cuando se está dibujando, para borrar el dibujo, para controlar el tiempo, para obtener al usuario y por último para guardar los resultados y enviarlos al servicio REST.

Sección 3:

En esta sección, el usuario puede continuar con la prueba, o llenar los campos de texto con los nombres de los animales que aparecen en pantalla.

MOCA Identificación

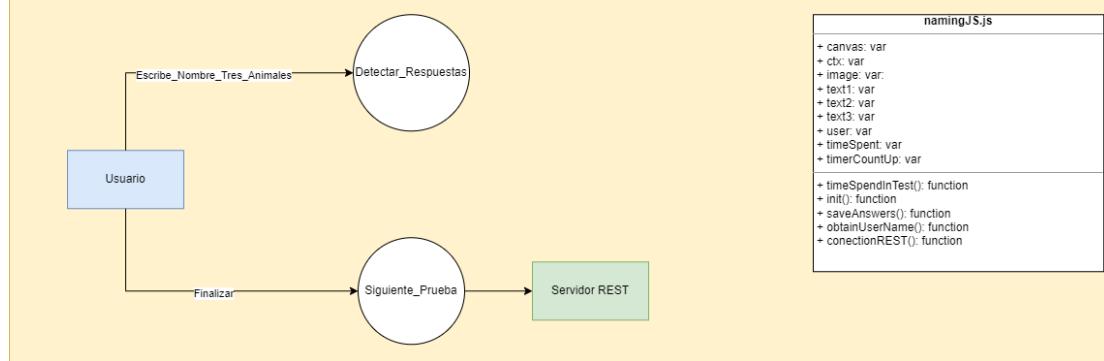


Ilustración 67: Diagramas MOCA Parte 3

Respecto al diseño de bajo nivel, se necesitan variables para mostrar en pantalla las imágenes de los animales, para obtener los valores de los campos de texto, y tener control del tiempo.

Además, contiene funciones para mostrar las imágenes, para controlar el tiempo, para obtener al usuario y por último para guardar los resultados y enviarlos al servicio REST.

Sección 4:

En esta sección, el usuario puede reproducir una pista de audio dos veces, y realizar una grabación de su voz dos veces. Finalmente, puede avanzar a la siguiente sección.

MOCA Memoria

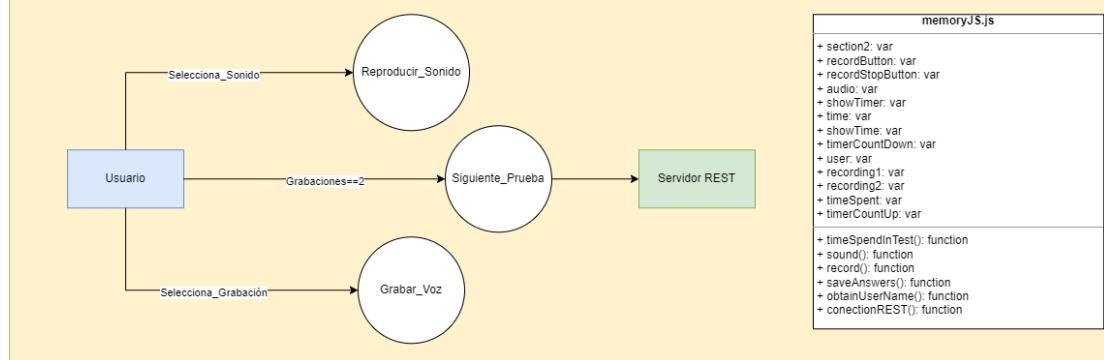


Ilustración 68: Diagramas MOCA Parte 4

Respecto al diseño de bajo nivel, se necesitan variables para manejar la grabación y detención de la grabación, así como de la pista de audio disponible. También hace falta variables para el temporizador y el tiempo empleado en la prueba.

Además, contiene funciones para grabar la voz del usuario, manejar el temporizador, reproducir el audio, para controlar el tiempo empleado, para obtener al usuario y por último para guardar los resultados y enviarlos al servicio REST.

Sección 5:

En esta sección, el usuario puede reproducir tres pistas de audio distintas además de grabar su voz dos veces, presionar un botón para

detectar el número de letras “A” dichas en un audio, y llenar una serie de campos de texto. Asimismo, puede avanzar en el MOCA.

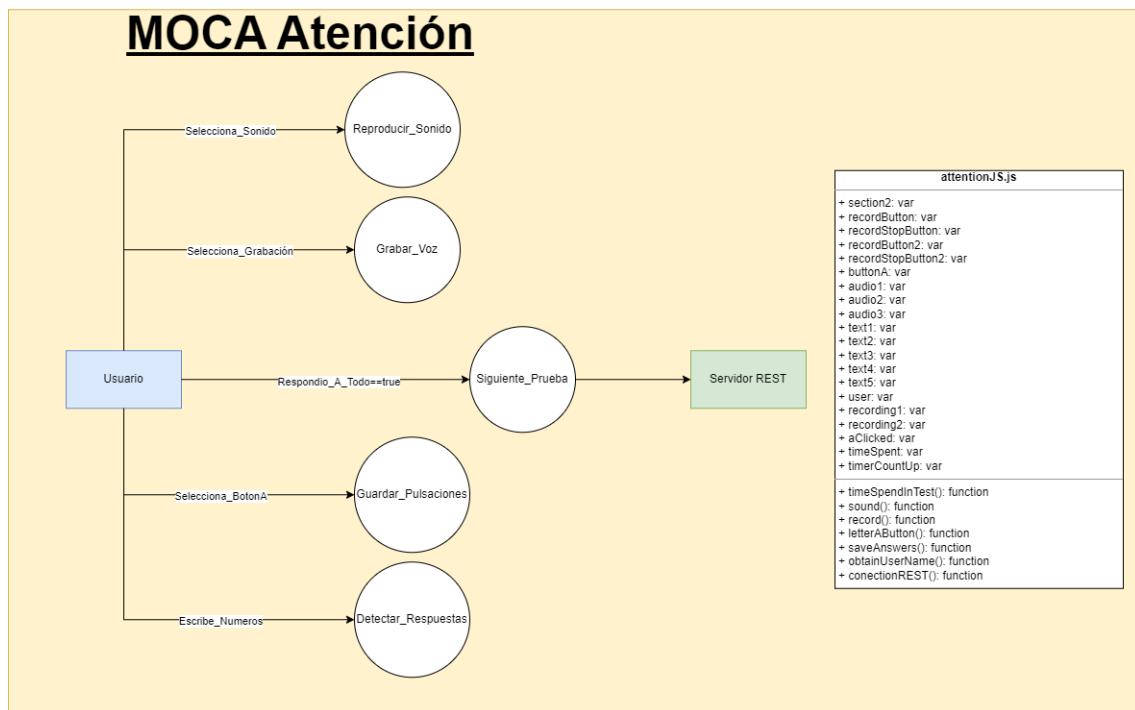


Ilustración 69: Diagramas MOCA Parte 5

Respecto al diseño de bajo nivel, se necesitan variables para manejar la grabaciones y detención de las grabaciones, así como de las pistas de audio disponibles. También hace falta variables para los campos de texto, para detectar el número de “A” pulsadas, y el tiempo empleado en la prueba.

Además, contiene funciones para grabar la voz del usuario, reproducir los audios, para controlar el tiempo empleado, para obtener al usuario y por último para guardar los resultados y enviarlos al servicio REST.

Sección 6:

En esta sección, el usuario puede grabar su voz tres veces y llenar dos campos de texto además de avanzar a la siguiente parte del MOCA.

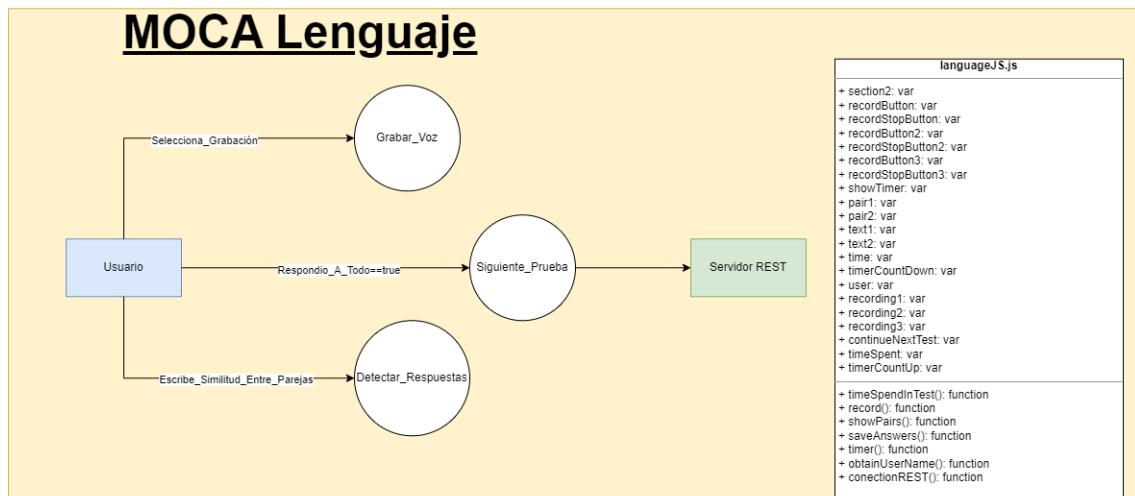


Ilustración 70: Diagramas MOCA Parte 6

Respecto al diseño de bajo nivel, se necesitan variables para manejar la grabaciones y detención de las grabaciones, así como de las pistas de audio disponibles. También hace falta variables para los campos de texto, para las parejas, y el tiempo empleado en la prueba.

Además, contiene funciones para grabar la voz del usuario, reproducir los audios, para mostrar las parejas, para controlar el tiempo empleado, para obtener al usuario y por último para guardar los resultados y enviarlos al servicio REST.

Sección 7:

En esta sección el usuario puede llenar una serie de campos de texto, así como solicitar una serie de pistas. Finalmente, puede finalizar la prueba.

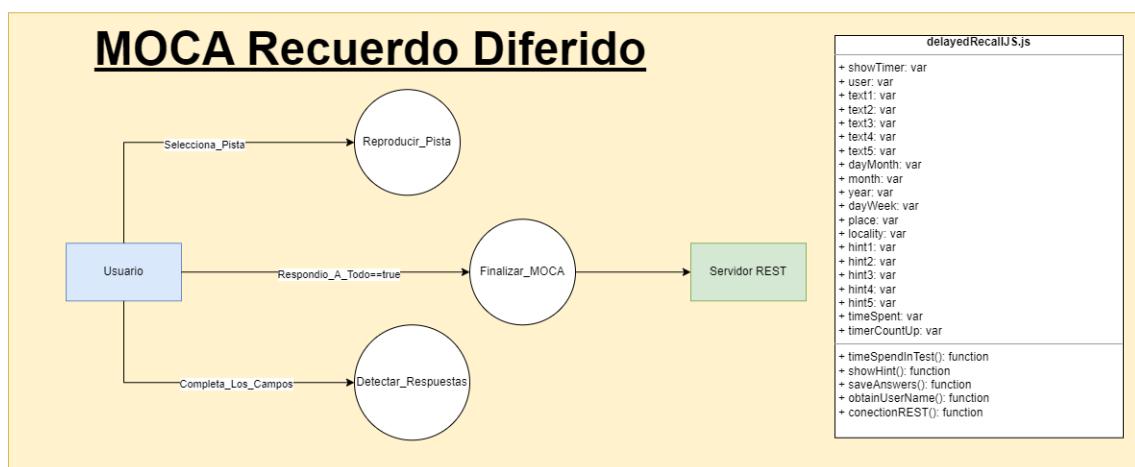


Ilustración 71: Diagramas MOCA Parte 6

Respecto al diseño de bajo nivel, se necesitan variables para manejar los campos de texto, así como las pistas y el tiempo empleado en la prueba.

Además, contiene funciones para mostrar por pantalla las distintas pistas, para controlar el tiempo empleado, para obtener al usuario y por último para guardar los resultados y enviarlos al servicio REST.

4.7.2 Interfaz y Uso

La prueba en general muestra una serie de instrucciones encima de un área blanca donde se alberga el contenido de cada una de las secciones y un botón debajo del área para continuar o finalizar la prueba. Además, en la sección 1 se hallan un botón para volver al menú principal , un botón para borrar el dibujo, y contiene dos imágenes, una de un cubo, y otra de un grafo sin completar. En la sección 2 se halla un botón para borrar el dibujo. En la sección 3, aparecen 3 dibujos de tres animales distintos, y tres campos de texto con las indicaciones de que escribir. En la sección 4, dentro del área se encuentra una pista de audio y unos botones para realizar una grabación de voz. En la sección 5, se puede apreciar tres pistas de audio, cuatro botones para realizar las dos grabaciones, un botón denominado “LETRA”, cinco campos de texto y una serie de instrucciones. En la sección 6 se hallan dos frases resaltadas en azul, cinco

botones para realizar las dos grabaciones, uno de los cuales no tiene botón para detener la grabación, dos campos de texto y una serie de instrucciones. Por último, en la sección 7, se presentan una serie de campos de texto con indicaciones sobre que escribir en ellos, además de una serie de botones para dar pistas.

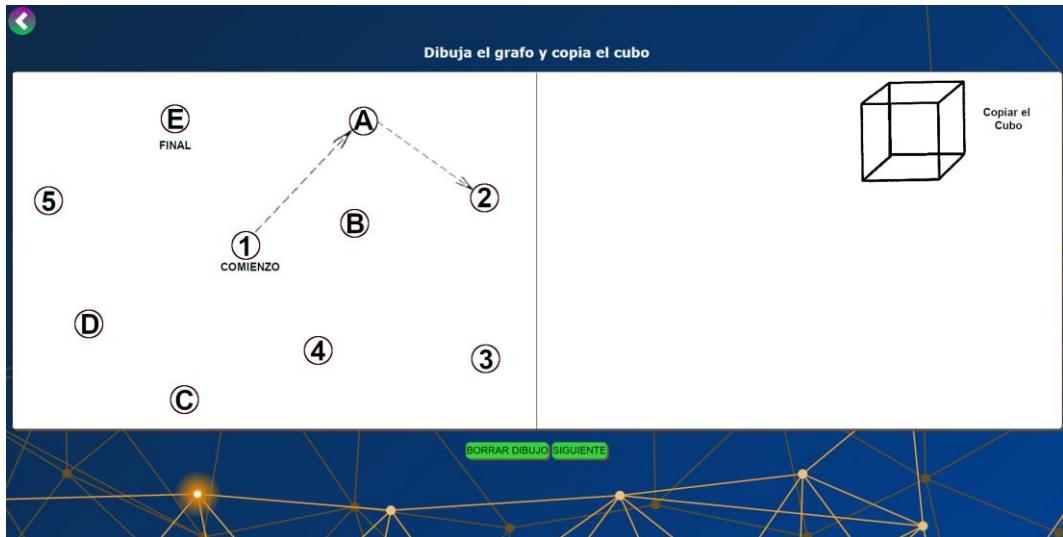


Ilustración 72: Interfaz MOCA Parte 1

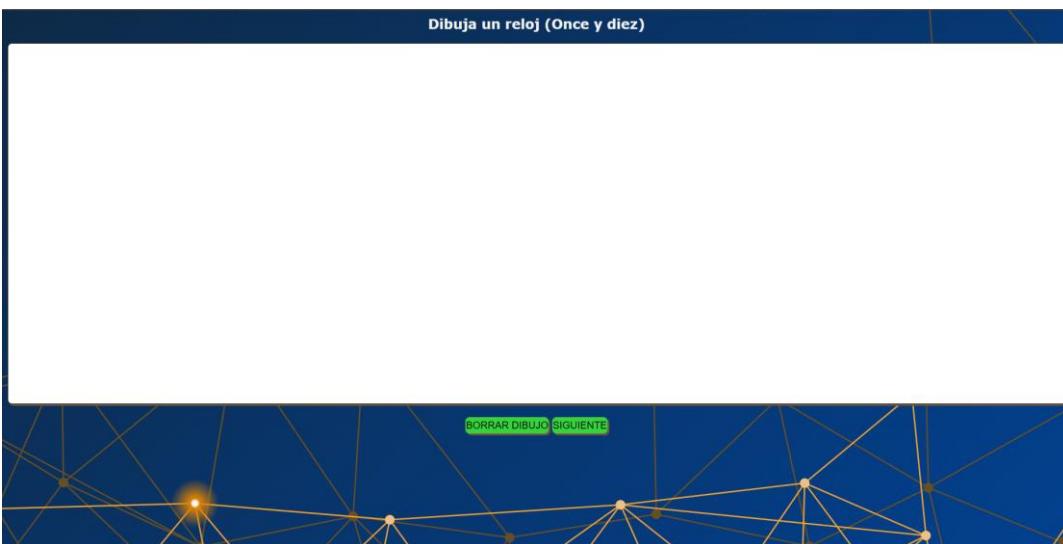


Ilustración 73: Interfaz MOCA Parte 2

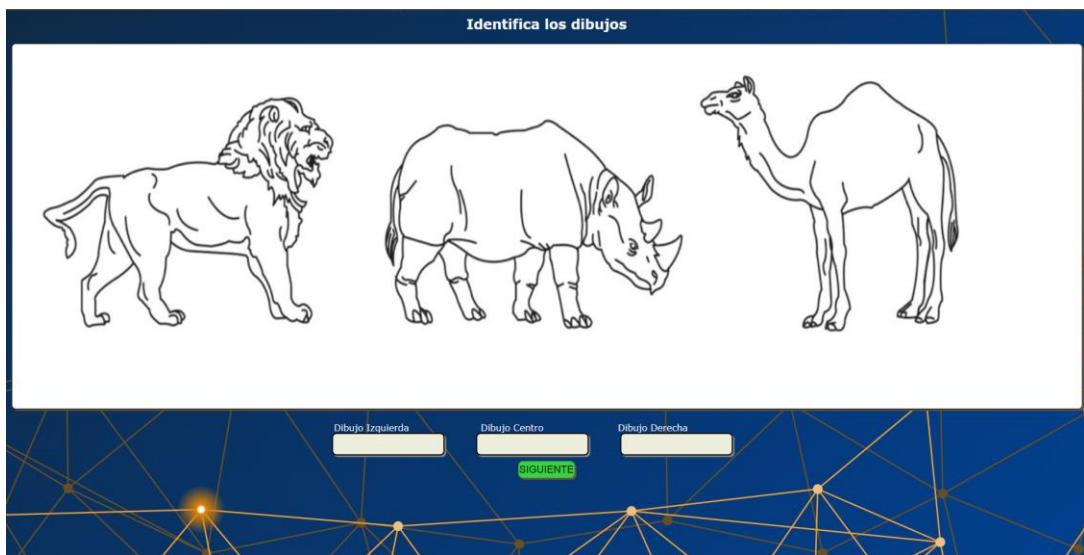


Ilustración 74: Interfaz MOCA Parte 3



Ilustración 75: Interfaz MOCA Parte 4

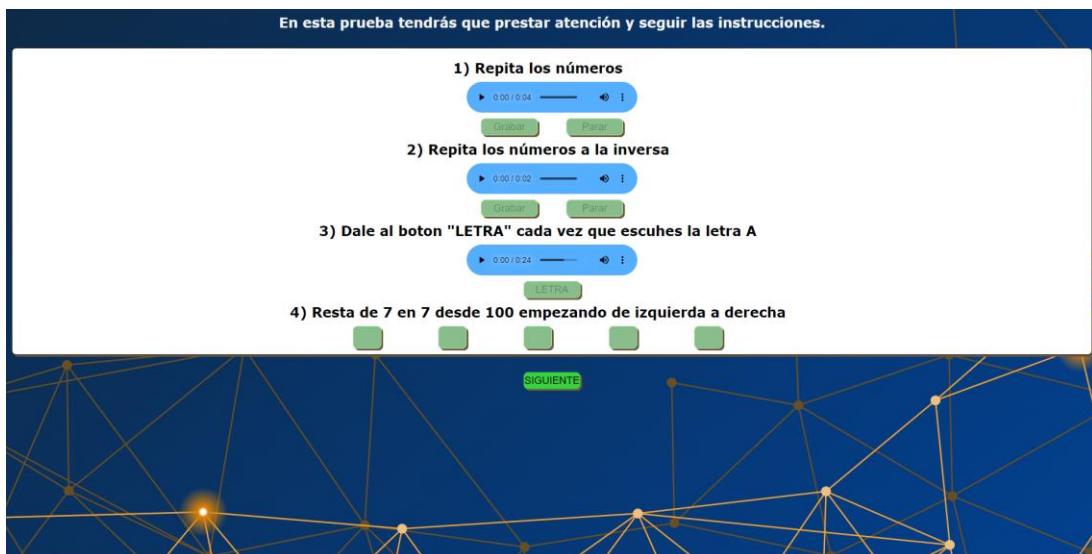


Ilustración 76: Interfaz MOCA Parte 5

En esta prueba leerás dos frases y tendrás que repetirlas. Después, deberás decir el mayor número posible de palabras que comiencen por la "P" en un minuto. Por último, escribir la similitud entre un par de parejas.

1) Repita la frase:
El gato se esconde bajo el sofá cuando los perros entran en la sala.

2) Repita la frase:
Espero que él entregue el mensaje una vez que ella se lo pida.

3) Di el mayor número posible de palabras que comiencen por la letra "P".

4) Di la similitud entre las parejas, por ejemplo:
manzana-naranja = fruta



Ilustración 77: Interfaz MOCA Parte 6

En esta prueba deberá escribir las 5 palabras memorizadas intentando hacerlo sin pistas. Después, escriba la fecha y lugar donde está haciendo las pruebas.

1) Complete los campos con las palabras que se acuerde:
 1^a Palabra 2^a Palabra 3^a Palabra 4^a Palabra 5^a Palabra

2) Complete los campos:
 Día del mes Mes Año
 Día de la semana Lugar Localidad

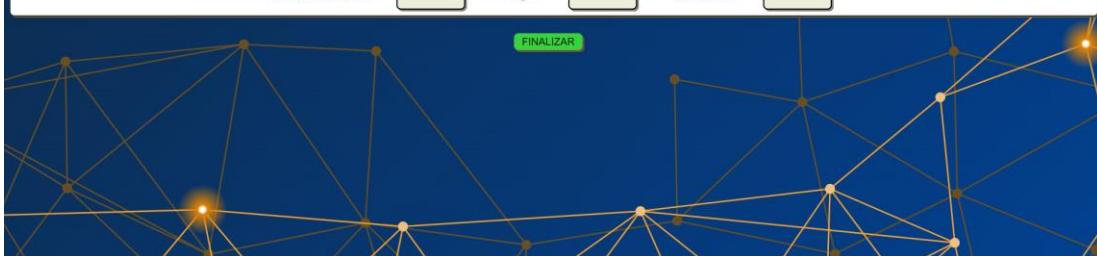


Ilustración 78: Interfaz MOCA Parte 7

El usuario comienza la prueba y tiene dos opciones: volver al menú principal, o dibujar sobre el área limitada. El usuario deberá dibujar por un lado un cubo, y por otro, un camino para acabar el grafo empezado. Si se equivoca puede borrarlo y empezarlo de nuevo, y cuando acabe, puede avanzar a la siguiente prueba del MOCA. En la segunda prueba, el usuario debe dibujar un reloj con las agujas apuntando a las once y diez. Si se equivoca, puede borrarlo y cuando acabe, puede avanzar a la siguiente parte. En la parte 3, el usuario debe identificar a los animales de las imágenes. En el primer campo de texto debe escribir el nombre del animal de la imagen de la izquierda, el segundo campo de texto debe escribir el nombre del animal de la imagen del centro, y el último campo de texto debe escribir el nombre del animal de la imagen de la derecha. Una vez finalice, puede avanzar a la siguiente sección. En la sección 4, el usuario reproduce la pista de audio, y debe grabarse diciendo las palabras que acaba de escuchar. Después de un minuto, debe de volver a realizar este procedimiento y continuar con el MOCA recordando en su mente estas palabras para una futura prueba. En la prueba 5, el usuario primero escucha unos números y debe grabarse repitiendo los mismos números. A continuación,

escucha otros números, pero ahora debe grabarse diciéndolos a la inversa. Después, reproducirá la tercera pista de audio y debe pulsar el botón “LETRA” cada vez que oiga una “A”. Por último, debe de restar de 7 en 7 empezando desde el número 10 y escribiendo los resultados en los campos de texto. Una vez hecho esto, podrá avanzar a la siguiente parte. En la parte 6, el usuario se graba dos veces diciendo dos frases distintas que están resaltadas en azul. Seguidamente, en un minuto, debe de decir el mayor número posible de palabras empezando por la letra “P”. Por último, aparecerán dos parejas de palabras y el usuario deberá llenar en los campos de texto disponibles, la relación entre las parejas avanzado así a la última sección. En la última sección, el usuario debe recordar las palabras que memorizó en la prueba 4 y escribirlas en los campos de texto adecuados. Si no se acuerda de alguna de ellas, puede solicitar ayuda con una serie de pistas, dos por cada palabra. Cuando haya escrito las palabras, debe llenar los últimos datos con el día del mes, mes, año, día de la semana, lugar y localidad donde está realizando el MOCA finalizando así la prueba.

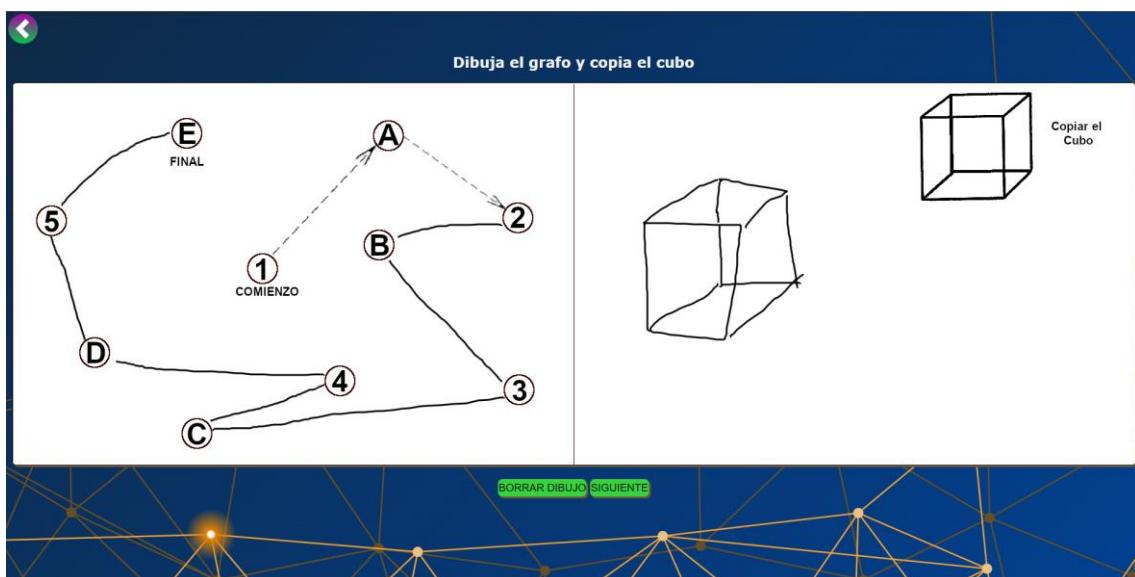


Ilustración 79: Dibujo realizado en MOCA Parte 1

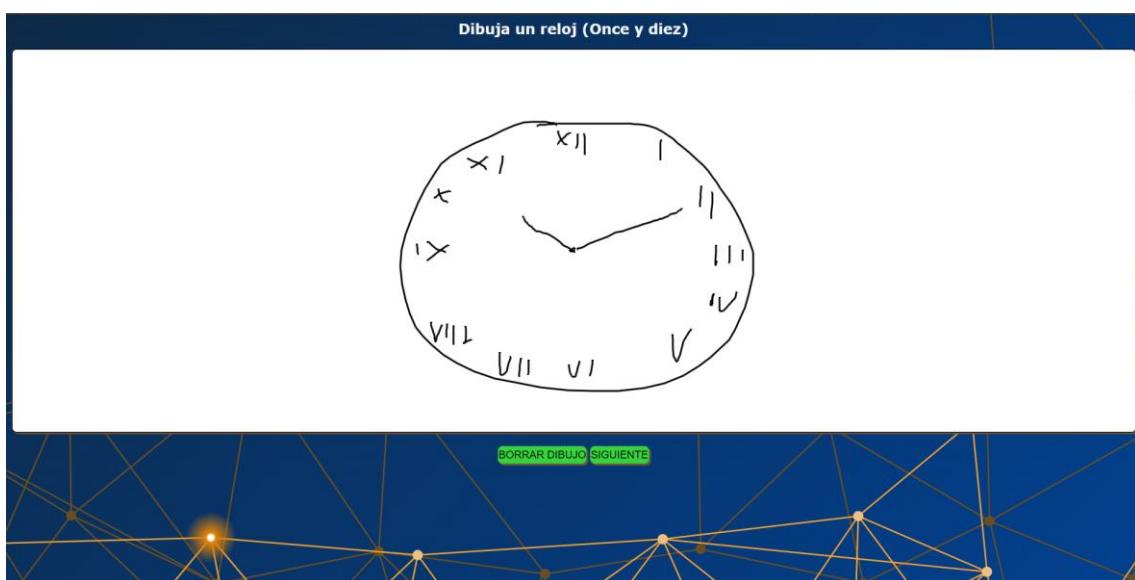


Ilustración 80: Dibujo realizado en MOCA Parte 2

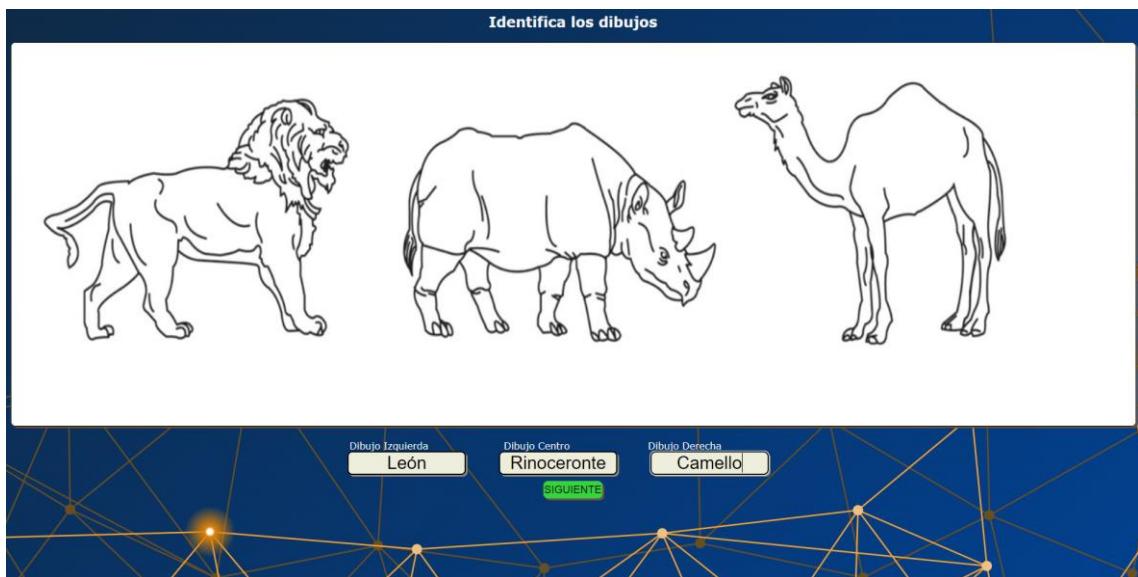


Ilustración 81: Identificación de los animales MOCA Parte 3

En esta prueba escucharás unas palabras y tendrás que repetirlas sin importar el orden. Despúes de un minuto, escúchala otra vez e intenta repetirlas. Memoriza las palabras porque te preguntarán sobre ellas en una prueba futura.

Dale al botón de audio para comenzar, y después, al botón "Grabar".

Reproducir audio

00:58

Grabar Parar

SIGUIENTE

Ilustración 82: Temporizador tras primera grabación MOCA Parte 4

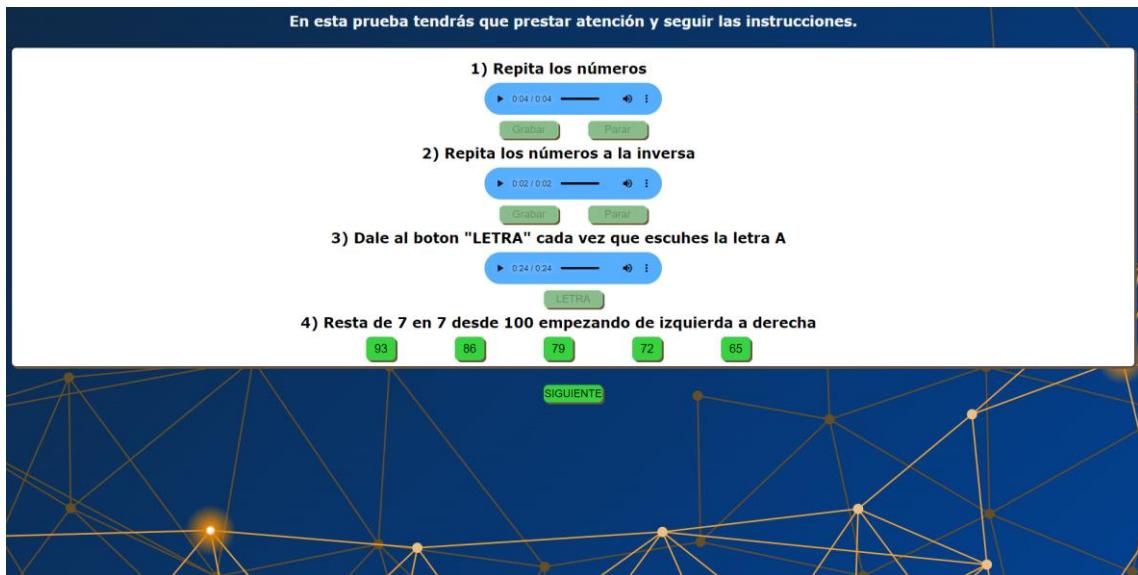


Ilustración 83: Finalización de MOCA Parte 5

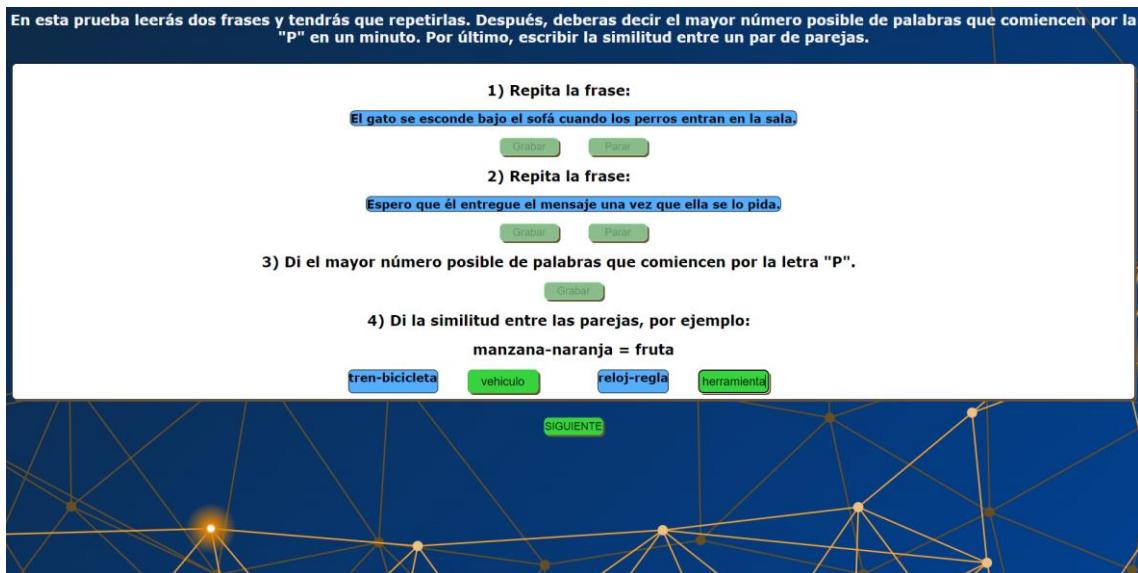


Ilustración 84: Finalización de MOCA Parte 6

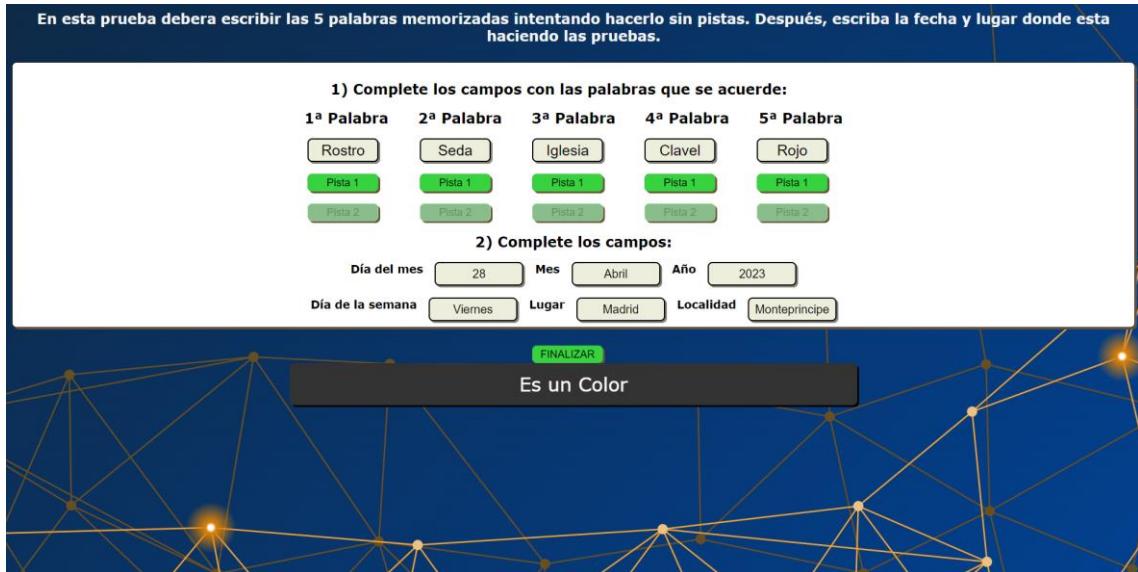


Ilustración 85: Finalización con pista de MOCA Parte 7

4.7.3 Almacenamiento de Datos

Para esta prueba se requiere almacenar distintos parámetros cada vez que se pase de sección o se finalice la prueba. Además, cada sección tiene distintos parámetros a guardar por lo que se mostrarán los parámetros que se guardan en cada sección individualmente en formato de lista:

- **Sección 1:** usuario, tiempo empleado y dibujo.
- **Sección 2:** usuario, tiempo empleado y dibujo.
- **Sección 3:** usuario, tiempo empleado, el animal identificado de la izquierda, centro y derecha.
- **Sección 4:** usuario, tiempo empleado, y las dos grabaciones de voz.
- **Sección 5:** usuario, tiempo empleado, las dos grabaciones de voz, el número de “A” encontradas, y las restas de 7 en 7 realizadas.
- **Sección 6:** usuario, tiempo empleado, tres grabaciones de voz, y las relaciones de las parejas.
- **Sección 7:** usuario, tiempo empleado, las palabras memorizadas, si seleccionaron alguna de las pistas, y el resto de los datos procedentes de los campos de texto.

4.8 Menú Principal

4.8.1 Diseño

El usuario utilizará un menú con el que registrar sus datos, así como acceder a las distintas pruebas de la aplicación. Cuando haya registrado sus datos, la información se enviará al servicio REST, y se le redirigirá a un menú donde pueda acceder a las pruebas. En este instante podrá elegir cualquier prueba y será redirigido a ella.

Toda esta lógica viene representada en un diagrama de alto nivel como se muestra en la ilustración 86.

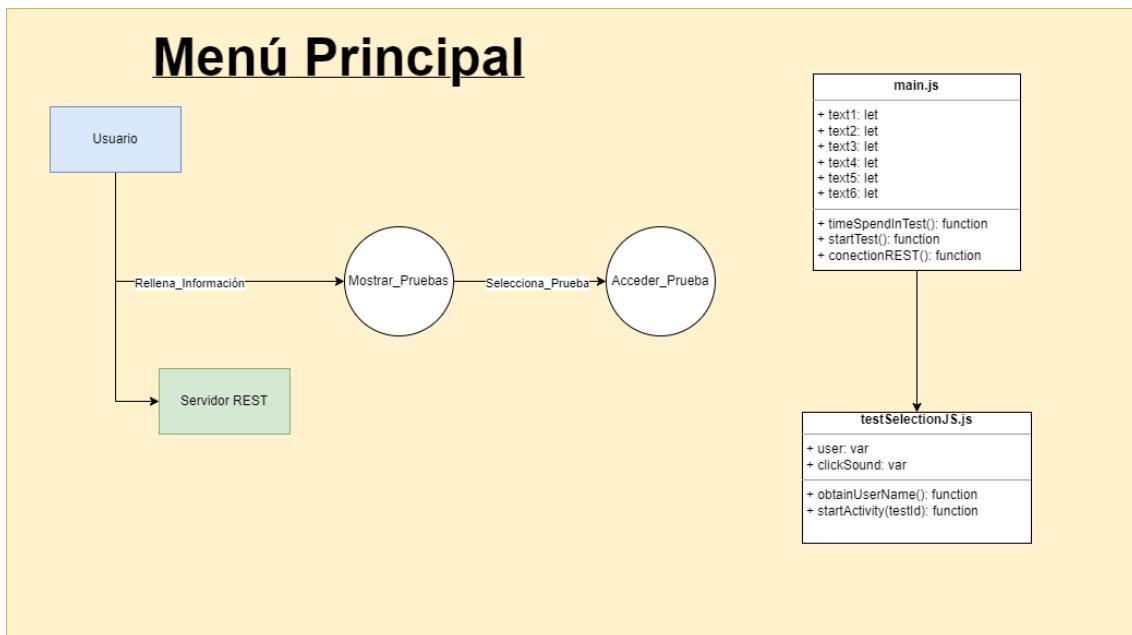


Ilustración 86: Diagramas Menú Principal

Respecto al diseño de bajo nivel de la ilustración 86, se empezará hablando de las variables.

En primer lugar, en la pantalla de registro, se necesitan variables para acceder al contenido de los campos de texto en los cuales el usuario introduce su información.

En segundo lugar, en el menú, se deben guardar el usuario y los efectos de sonido.

Sobre los métodos, por un lado, “timeSpendInTest()” sirve para comenzar el contador del tiempo transcurrido en la prueba, “startTest()” redirige al usuario al menú principal, “conexionREST()”, sirve para enviar los parámetros necesarios al servicio REST. Por el otro lado, “obtainUserName()” obtiene el nombre del usuario realizando la prueba y “startActivity(testId) accede a una de las pruebas”.

4.8.2 Interfaz y Uso

El menú muestra en la pantalla, un rectángulo donde se hayan una serie de campos de texto de diferentes tipos con indicaciones sobre que escribir en ellos. Además, tienen un botón para redirigir al usuario al menú. En el menú, se hallan siete botones, uno para cada prueba, con una serie de dibujos.

Rellene los campos con su información

Nombre:

EDAD:

Nivel de Estudios:

Sexo: Varón

Fecha de Nacimiento: dd/mm/aaaa

Fecha: dd/mm/aaaa

Ilustración 87: Interfaz Menú, Registro de Datos

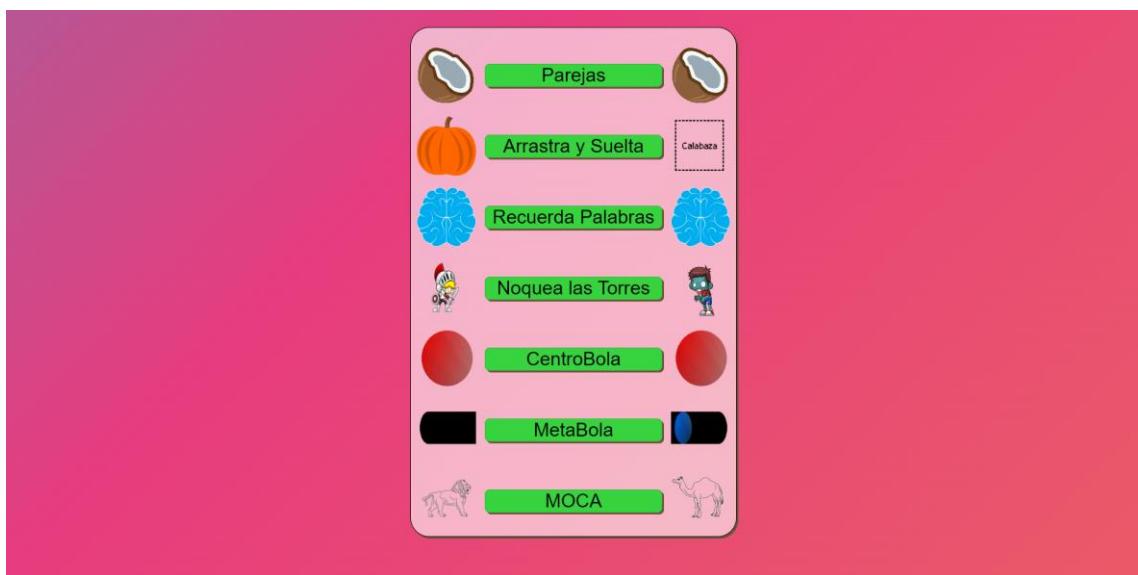


Ilustración 88: Interfaz Menú Principal

El usuario, rellena los campos de texto disponibles con su información, y después, accede a un menú donde se hallan botones que redirigen al usuario a las distintas pruebas. En esta parte, el individuo puede elegir libremente cuál de las pruebas realizar pulsando sobre una de ellas.

4.8.3 Almacenamiento de Datos

En el menú principal, se almacena la información del usuario que va a realizar la batería de pruebas. Por ello, se guardará el nombre del usuario, edad, nivel de estudios, sexo, fecha de nacimiento y fecha actual.

```
+ usages
public class Usuario {
    2 usages
    @JsonProperty("usuario")
    private String usuario;
    2 usages
    @JsonProperty("estudios")
    private String estudios;
    2 usages
    @JsonProperty("sexo")
    private String sexo;
    2 usages
    @JsonProperty("nacimiento")
    private LocalDate nacimiento;
    2 usages
    @JsonProperty("fecha")
    private LocalDate fecha;
    2 usages
    @JsonProperty("edad")
    private int edad;
```

Ilustración 89: Parámetros Menú Principal

5 Resultados y conclusiones

Tras la realización de este trabajo, se ha obtenido una aplicación web basada en las tecnologías HTML, CSS y JavaScript que contiene un total de siete pruebas distintas para evaluar la cognición de una persona.

En la aplicación, el usuario puede registrar sus datos y seleccionar una de las pruebas. Estas pruebas evalúan distintos aspectos de la cognición, pero están desarrolladas para mantener un mismo aspecto común en la medida de lo posible.

Acompañando a esta aplicación, se ha conseguido desarrollar un servicio REST capaz de comunicarse con la aplicación web y guardar los distintos parámetros de las pruebas en una base de datos, para su futuro análisis.

Por último, gracias a Ionic, se implementó una versión de la aplicación para dispositivos móviles. Por lo que este trabajo, contiene una aplicación web móvil que es capaz de conectarse a un servicio REST y almacenar información en una base de datos.

Por tanto, se ha logrado cumplir los objetivos propuestos ya que estamos ante un proyecto que permite recopilar y guardar distintos parámetros de los pacientes para su posterior uso. Un proyecto dividido en dos partes, la aplicación web y el servicio REST que puede ser de mucha utilidad a los médicos para diagnosticar enfermedades del cerebro como el Alzheimer y analizar los efectos de una dieta basada en polifenoles.

Durante el desarrollo, se experimentaron diversas adversidades que se pudieron resolver a tiempo. La primera a mencionar sería el poco conocimiento sobre algunas de las tecnologías implementadas, como pueden ser JavaScript, CSS o Ionic. La segunda dificultad sería las pruebas “MetaBola” y “MOCA”, una por el cálculo de las curvas de Bezier, y la otra por las distintas secciones que conllevaba. Otra dificultad para destacar sería en el aspecto visual, ya que ha sido un tarea ardua conseguir una vista funcional a la vez que adecuada para el usuario al que va dirigido esta aplicación y bonita, adaptada también a dispositivos móviles. Dichos problemas requirieron de más horas de las queridas, pero gracias a la investigación y a mi constante empeño, puede solventar los problemas.

Cronológicamente, tras conocer el número de pruebas a realizar y como serían, se comenzó por la aplicación web como primer objetivo, es decir, con los JavaScript, CSS y HTML. Al ser una tecnología que desconocía bastante, me informé y aprendí a cómo usarlas. Después, comencé con la primera prueba “Parejas”. Una prueba que ha sufrido diversos cambios estéticos, pero ha sido de gran inspiración para el resto de las pruebas, así como de aprendizaje. En dicha prueba, se consiguió crear una prueba que mide la memoria y velocidad de procesamiento a través de la memorización de parejas en un tablero, intentando hallar todas las parejas del tablero en un determinado tiempo. Añadiendo a su vez, dos niveles de dificultad.

A continuación, la prueba “Arrastra Y Suelta”, una prueba que costo por su mecánica principal, la capacidad de arrastrar un objeto y depositarlo en algún lugar. Con ello, se creó una prueba que mide el conocimiento de frutas y verduras y, sobre todo, la coordinación ojo-mano. Para ello, se ha conseguido una prueba donde se arrastren frutas y verduras en su respectivo lugar,

siendo el nivel de las verduras de mayor dificultad por su número de elementos a arrastrar.

Seguidamente, “Recuerda Palabras”, una prueba donde lo que más costo fue la coordinación de los distintos botones que había con respecto a la tabla de palabras y temporizador. Aun así, se alcanzó una prueba que se basa en la memoria recordando una serie de palabras en un determinado tiempo, ambos dependientes de las ajustes elegidos.

Posteriormente, la prueba de “Noquea Las Torres”. Aquí, lo más costoso fueron las animaciones de los personajes, así como la configuración de las distintas torres. Finalmente, se obtuvo una prueba que se centra en las matemáticas. Para ello, existen unas torres con un valor matemático en cada piso y el individuo deberá seleccionar cuál de ellas tiene menor número.

Más adelante, se empezó la prueba “CentroBola”. Una de las pruebas que menos dificultad me han supuesto, pero a la vez muy efectiva para medir la monitorización y coordinación ojo-mano mediante la simple idea de pulsar lo más cerca posible al centro de un círculo.

Luego, con la prueba “MetaBola”, una de las pruebas que más desafío me han supuestos por la curvas de Bezier. Superadas las adversidades, se logró una prueba donde una bola avanza en un camino y el individuo debe pararla en el momento justo y así, se mide la anticipación, cálculo mental, percepción espacial, flexibilidad cognitiva y velocidad de procesamiento.

A continuación, el “MOCA”, una adaptación en formato digital de la prueba evaluación cognitiva de Montreal. Precisamente por ello, fue un test en el que las ideas estaban claras y estructuradas, pero por su elevado contenido no fue tarea fácil. Después de todo, se quedó en una prueba con distintas secciones que simula adecuadamente el test de Montreal.

Por último, para hacer la aplicación más práctica y sencilla, se creó un menú principal, donde cada individuo puede introducir sus datos y seleccionar la prueba a realizar, y con ello, enlazar toda la batería de pruebas.

Una vez realizadas las dos primeras pruebas, paralelamente, se trabajó en el servicio REST, consiguiendo así vincular la aplicación web con la obtención de datos para su posterior análisis por parte de los profesionales. Ha sido un servicio que ha estado sufriendo constantes cambios para poder comunicarse perfectamente con la aplicación web, así como los propios datos importantes a guardar. Más adelante, también se consiguió conectar a una base de datos creada para la ocasión con el objetivo de guardar todo lo recibido en un sitio seguro y permanente. Finalmente, una vez se comprobó el correcto funcionamiento de la aplicación, así como su comunicación con el servicio REST, se adaptó a una aplicación móvil mediante Ionic, finalizando y completando de esta manera, los objetivos de este trabajo.

En líneas futuras, esta aplicación podría incorporar nuevas y más variadas pruebas clasificadas en secciones. De esta manera no solo ayudaría a los médicos a detectar deterioros cognitivos, si no que ayudaría a las personas a mantener su mente activa y sana.

A nivel personal, este proyecto me ha permitido aprender sobre distintas tecnologías, especialmente JavaScript, CSS, y HTML las cuales me han supuesto un desafío, pero al final, he disfrutado aprendiendo con ellas.

Además, he podido aplicar conocimientos adquiridos durante el grado de Ingeniería Informática como el manejo de una base de datos, el servicio REST o simplemente el poder adaptarme e investigar sobre nuevas herramientas, consiguiendo un producto final del cual estoy bastante satisfecho.

6 Análisis de Impacto

Personalmente, este proyecto me ha impactado positivamente. Me ha servido como una gran experiencia de aprendizaje que puede serme de gran utilidad en el futuro. Asimismo, me ha permitido mejorar numerosas competencias y habilidades como la adaptación ante un problema, búsqueda de soluciones rápidas o creatividad y a reforzar conocimientos adquiridos en la carrera.

Este trabajo podría tener un gran impacto social en el aspecto de la salud y el bienestar según la Agenda de 2030 de la ODS ya que permitirá diagnosticar enfermedades relacionadas con el cerebro y detectar que alimentos ayudan a eliminar deterioros cognitivos de las personas fomentando así, un gran bienestar en aquellas personas que tengan problemas cognitivos permitiendo de esta manera mejorar su calidad de vida.

Este tipo de pruebas normalmente son realizadas con papel y junto a un especialista, por lo que reunirlas en aplicaciones digitales supondría una mejora en el medio ambiente, así como de transporte, recursos y tiempo.

7 Bibliografía

- [1] «Pruebas Cognitivas,» MedlinePlus, [En línea]. Available: <https://medlineplus.gov/spanish/pruebas-de-laboratorio/pruebas-cognitivas/>.
- [2] «Estudio Observacional KEPHENOL,» ICTAN, 13 Mayo 2022. [En línea]. Available: <https://www.ictan.csic.es/8724/seguimos-necesitando-voluntarios-para-el-estudio-observacional-kephenol/>.
- [3] D. J. H. Nogueira, «Estimación de la ingesta de polifenoles y contaminantes alimentarios en diferentes poblaciones: Mujeres embarazadas y niños,» Universidad de Granada, 2022. [En línea]. Available: <https://digibug.ugr.es/bitstream/handle/10481/80017/73926%281%29.pdf?sequence=4&isAllowed=y>.
- [4] N. Carcavilla, «Mini-Cog test para detectar la demencia,» Comunicación & Demencias, [En línea]. Available: <https://comunicacionydemencias.com/mini-cog/>.
- [5] «Mini Mental de Folstein (MMSE),» Stimulus, 28 mayo 2019. [En línea]. Available: <https://stimuluspro.com/blog/mini-mental-de-folstein-mmse/#:~:text=El%20MMSE%20es%20una%20prueba,de%20un%20paciente%20con%20demencia..>
- [6] N. Carcavilla, «MoCA: Test de Evaluación Cognitiva Montreal,» Comunicación & Demencias, [En línea]. Available: <https://comunicacionydemencias.com/test-moca-demencia/#:~:text=El%20MoCA%20es%20una%20prueba,para%20la%20enfermedad%20de%20Alzheimer..>
- [7] «Analizamos...,» Orientatech, [En línea]. Available: <https://www.orientatech.es/cognifit#:~:text=Cognifit%20es%20una%20aplicaci%C3%B3n%20de,la%20memoria%20y%20agilidad%20mental..>
- [8] Cognifit, [En línea]. Available: <https://www.cognifit.com/es/evaluacion-cognitiva/test-cognitivo>.
- [9] «CANTAB,» CAMBRIDGE COGNITION, [En línea]. Available: <https://www.cambridgecognition.com/cantab/>.
- [1] J. S. B. a. P. A. Anders Behrens, «CoGNIT Automated Tablet Computer 0 Cognitive Testing in Patients With Mild Cognitive Impairment: Feasibility Study,» National Library of Medicine, 11 Marzo 2022. [En línea]. Available: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8957010/>.
- [1] «HTML: Lenguaje de etiquetas de hipertexto,» MDN web docs, 30 Noviembre 1] 2022. [En línea]. Available: <https://developer.mozilla.org/es/docs/Web/HTML>.
- [1] Manz, «¿Qué es CSS?,» LenguajesCSS, [En línea]. Available: 2] <https://lenguajecss.com/css/introduccion/que-es-css/>.
- [1] «Acerca de JavaScript,» MDN web docs, 12 Diciembre 2022. [En linea]. 3] Available: <https://developer.mozilla.org/es/docs/conflicting/Web/JavaScript>.

- [1] J. M. R. Moncayo, «Qué es REST: Conoce su potencia,» OpenWebinars, 17 4] Mayo 2018. [En línea]. Available: <https://openwebinars.net/blog/que-es-rest-conoce-su-potencia/>.
- [1] C. Cedeño, «Tecnología móvil: qué es, desarrollo, usos y tipos de tecnología 5] celular actual,» Cinconoticias, 6 Septiembre 2022. [En línea]. Available: <https://www.cinconoticias.com/tecnologia-movil/>.
- [1] «¿Cómo desarollar apss multiplataforma?,» KEEPCODING, 8 Agosto 2022. 6] [En línea]. Available: <https://keepcoding.io/blog/desarrollar-aplicaciones-moviles-multiplataform/>.
- [1] M. Heller, «What is Visual Studio Code? Microsoft's extensible code editor,» 7] InforWorld, 8 Julio 2022. [En línea]. Available: <https://www.infoworld.com/article/3666488/what-is-visual-studio-code-microsofts-extensible-code-editor.html>.
- [1] «IntelliJ IDEA,» Techopedia, [En línea]. Available: 8] <https://www.techopedia.com/definition/7755/intellij-idea> .