

Juego serio con realidad aumentada destinado a la generación de ejercicios estructurados en complejidad e intensidad para el tratamiento de niños con TDAH



Universidad del Cauca

**Rosmer Hasan Yepes Hoyos
Jhoan Sebastian Hurtado Campo**

**Director: Diego Enrique Guzmán Villamarin, Msc.
Codirector: Carlos Felipe Rengifo Rodas, PhD.**

Universidad del Cauca
Facultad de Ingeniería Electrónica y Telecomunicaciones
Departamento de Electrónica, Instrumentación y Control
Popayán
2022

Juego serio con realidad aumentada destinado a la generación de ejercicios estructurados en complejidad e intensidad para el tratamiento de niños con TDAH

Rosmer Hasan Yepes Hoyos
Jhoan Sebastian Hurtado Campo

Trabajo de grado presentado a la Facultad de Ingeniería Electrónica y Telecomunicaciones de la Universidad del Cauca para la obtención del título de

Ingenieros en:
Automática Industrial

Universidad del Cauca
Facultad de Ingeniería Electrónica y Telecomunicaciones
Departamento de Electrónica, Instrumentación y Control
Popayán
2022

Resumen

Este trabajo tiene como objetivo generar una herramienta para ayudar en el tratamiento de niños con TDAH, en el cual los terapeutas puedan configurar ejercicios estructurados en complejidad e intensidad usando los beneficios que presentan los juegos serios y la realidad aumentada. En primera instancia se hizo una selección de características por parte de un grupo de terapeutas utilizando la metodología Scrum, estas características permitieron diseñar la arquitectura de la herramienta la cual se compone de tres módulos que se comunican mediante una base de datos: el primer módulo es el juego serio en realidad aumentada, en donde el infante podrá realizar los ejercicios asignados; el segundo módulo es el de configuración, en este el terapeuta podrá estructurar los ejercicios en complejidad e intensidad y asignarlos a los infantes; el tercer módulo es el de reportes, el cual permite generar reportes de desempeño de los infantes, para que el terapeuta pueda realizar un análisis de la información obtenida y tomar las decisiones pertinentes en el tratamiento. Con la finalidad de evaluar la usabilidad de la herramienta, se seleccionó un grupo de especialistas en el área a los que se les suministró la información y los archivos necesarios para ejecutarla y posteriormente efectuar una encuesta con base en las características de usabilidad mostradas en la norma ISO/IEC 25010. Los resultados arrojaron calificaciones que dejan la puerta abierta a más herramientas de este tipo por todos los beneficios que presentan.

Palabras clave: TDAH, juego serio, realidad aumentada, ejercicio estructurado.

Abstract

This project aims to generate a tool to help in the treatment of children with ADHD, where therapists can set exercises structured in complexity and intensity using the benefits of serious games and augmented reality (AR) for the acquisition of behaviors and skills. In the first instance, a selection of characteristics was made by a group of therapists using the scrum methodology, these characteristics allowed the design of the structure of the tool, which is made up of three modules that communicate through a database, the first module is the serious game in augmented reality (AR) where the infant will be able to perform the assigned exercises, the second module is the configuration module, in which the therapist will be able to structure the exercises in complexity and intensity and assign them to the infants, the third module is the report module which It allows generating performance reports of infants so that the therapist can perform an analysis of the information obtained and make the corresponding decisions in the treatment. In order to evaluate the usability of the tool, a group of specialists in the area was selected, who were provided with the information and files necessary to execute it and subsequently carry out a google form, the results showed qualifications that leave the door open to more tools of this type for all the benefits they present.

Keywords: ADHD, serious game, augmented reality, structured exercise.

Tabla de contenido

1. Introducción	1
2. Materiales y métodos	6
2.1. Metodología Scrum	6
2.2. Selección de características	9
2.3. Arquitectura del software propuesto	11
2.3.1. Lenguaje Unificado de Modelado	12
2.4. Justificación de la herramienta software	14
2.4.1. Herramientas de desarrollo	14
2.4.2. Kit de desarrollo de software (SDK) para realidad aumentada	17
2.4.3. Motor de base de datos	19
2.4.4. Librerías utilizadas	22
2.5. Creación de la base de datos	23
3. Resultados	25
3.1. Módulo de configuración	25
3.2. Módulo de juego serio	36
3.3. Módulo de reportes	46
3.4. Pilotaje del módulo de reportes	53
3.5. Protocolo experimental	54
3.5.1. Selección de participantes.	54
3.6. Parámetros de validación	57
3.7. Medición de valores para respuestas	60
3.8. Análisis de resultados	60
4. Discusión y conclusiones	67
4.1. Discusión	67
4.2. Conclusiones	68
4.3. Trabajos futuros	69

Índice de tablas

1.	Ejemplo de historia de usuario.	7
2.	Historias de usuario.	9
3.	Motores de desarrollo.	17
4.	Kit de realidad aumentada compatible con Unity.	19
5.	Motores de base de datos.	20
6.	Comparación de datos obtenidos	54
7.	Comparación media y varianza de datos obtenidos	54
8.	Especialistas voluntarios	56
9.	Preguntas del cuestionario	59
10.	Valores en escala de likert	60

Índice de figuras

1.	Arquitectura generalizada.	12
2.	Diagrama de componentes MiniSuper AR.	13
3.	Diagrama de caso de uso.	13
4.	Jerarquía base de datos.	24
5.	Flujo de interfaces del módulo de configuración.	26
6.	Interfaz inicial.	27
7.	Interfaz de registro.	28
8.	Interfaz de bienvenida.	29
9.	Interfaz de pacientes.	29
10.	Interfaz de registro de pacientes.	30
11.	Interfaz de ejercicio.	31
12.	Interfaz de creación de ejercicio.	32
13.	Interfaz de visualización de ejercicios.	33
14.	Interfaz de selección de configuración.	33
15.	Interfaz de configuración de reto.	34
16.	Interfaz visualización de reto.	35
17.	Interfaz selección de lado de estante.	35
18.	Interfaz configuración de estante.	36
19.	Interfaz de inicio juego serio	37
20.	Interfaz de inicio de sesión.	38
21.	Interfaz de perfil de usuario.	38
22.	Interfaz de misiones en las fechas correctas.	39
23.	Interfaz de misiones en las fechas incorrectas.	39
24.	Agregar cámara en AR.	40
25.	Agregar herramientas de para detección de planos.	40
26.	Inspector de Plane Finder.	41
27.	Jerarquía de Ground Plane Stage.	41
28.	Puntero para colocar objeto.	42
29.	Estante con objetos.	42
30.	Jerarquía de objetos desactivados.	42
31.	Jerarquía del monster desactivado.	43
32.	Asignación de tag.	43
33.	Progreso de las tareas.	44
34.	Progreso con una tarea completada.	44
35.	Imagen de reloj utilizado.	44

36.	Reloj en pantalla de juego.	45
37.	Interfaz de victoria.	45
38.	Interfaz de derrota.	46
39.	Interfaz acceso/login módulo de reportes.	47
40.	Interfaz de extracción de datos paciente.	48
41.	Interfaz de generación de reporte y reutilización.	48
42.	Página introductoria.	50
43.	Página descriptiva.	51
44.	Página evaluativa.	52
45.	Prueba de realización de ejercicio.	53
46.	Gestión de pacientes	61
47.	Nube de palabras gestión de pacientes.	61
48.	Gestión de ejercicios.	62
49.	Nube de palabras gestión de ejercicios.	63
50.	Gestión de reportes.	63
51.	Nube de palabras gestión de reportes.	64
52.	Jugabilidad.	65
53.	Nube de palabras de jugabilidad.	65

Capítulo 1

Introducción

El trastorno por déficit de atención e hiperactividad (TDAH) es uno de los trastornos más comunes en edades tempranas. Su sintomatología comprende la inatención, la hiperactividad y la impulsividad [1, 2]. El instituto neurológico de Colombia describe el TDAH como una alteración neurológica crónica, la cual se descubre después de los cinco años de edad [3]. Sus orígenes pueden ser genéticos, neuroquímicos y medioambientales [4, 5]. Este trastorno afecta al 5 % de la población en niños y adolescentes de todo el mundo [4], además de que persiste en la adultez, generando un alto índice de riesgo en la aparición de otras dificultades conductuales, emocionales y físicas [6, 7, 8, 9]. En Colombia [4], alude que entre un 15 % y 17 % de la población colombiana padece este trastorno, evidenciando un panorama de alto riesgo en la salud mental de escolares y adolescentes.

Por otra parte, al ser un trastorno que afecta diferentes áreas y procesos como lo son los pedagógicos y psicosociales, el niño por sí mismo difícilmente podrá resolverlo de forma autónoma, requiere ayuda e intervención ya sea farmacológica o no farmacológica, para reintegrarse y desarrollarse de acuerdo a su potencial y talento [3]. Entre las intervenciones para este trastorno, las farmacológicas actualmente son las más usadas, aunque realmente no abarcan en totalidad las afectaciones que el TDAH genera en quienes lo poseen [10]. Dentro de éstas se podrían mencionar las habilidades sociales, los déficits cognitivos y emocionales [11, 12], además de que puede dar lugar a largo plazo a efectos secundarios, adversos y/o desconocidos [13].

En concordancia con lo anterior, la medicación es el tratamiento más usado, aún así hay otros métodos como por ejemplo el implementado en [14] en el cual después de realizar el debido diagnóstico, se procedió a programar sesiones terapéuticas en las que se realizaban 3 tareas programadas con anterioridad, las cuales buscaban potenciar la función reguladora del lenguaje para lograr una actividad voluntaria y organizada. Por otra parte, técnicas alternativas como la musicoterapia son empleadas como ayudantes en tratamientos psicoterapéuticos del TDAH, donde se integra la música con técnicas mixtas tanto alternativas como farmacológicas tradicionales [15].

Para finalizar, el TDAH presenta características, unas más comunes que otras. Como por ejemplo el bajo desempeño en las funciones ejecutivas y las habilidades motoras, los cuales son abordados a través de tratamientos que mediante la implementación de ejercicios físicos y cognitivos, pretenden generar un mejoramiento en la realización de estas actividades [16]. El uso de los procedimientos basados en ejercicios físicos tienen la ventaja, no hay la necesidad de recurrir a fármacos que puedan afectar negativamente el proceso del paciente [17].

Estado del arte

Con base en lo descrito anteriormente, el TDAH es un trastorno que ha sido abordado con diferentes estrategias dentro de las cuales las más relevantes son las relacionadas con el juego. Por ende, el siguiente apartado busca realizar una descripción breve de lo que se considera el estado del arte, en el cual se involucran terapias alternativas cuyo beneficio en el tratamiento del TDAH ha sido comprobado, como lo son; la estimulación cerebral, la estimulación cognitiva, la incorporación del ejercicio físico, la terapia de juego entre otros [13].

Para iniciar, se considera indispensable aclarar que el juego es una actividad que los niños realizan desde sus primeros años de vida, por lo que la incorporación de estos en las terapias alternativas se presta como una herramienta de aprendizaje para la adquisición de conductas, competencias y capacidades [18, 19]. Esto nos lleva a los juegos serios (JS) los cuales utilizan tecnologías de juegos electrónicos con metodologías pedagógicas, aportando una alta flexibilidad, adaptabilidad y alcance; características necesarias en tratamientos constantes y de rehabilitación [20]. Además, el uso de juegos serios como una herramienta de tratamiento en niños con discapacidades de desarrollo [21], es una forma que repercute de manera positiva porque se presenta de manera divertida para el infante.

En relación con lo anterior, el uso de herramientas tecnológicas se considera entre los tratamientos alternativos y se pueden dar por medio de plataformas software, las cuales muestran una gran practicidad y una mayor cobertura para su aplicabilidad [22]. El uso de estas plataformas esta teniendo un alto impacto en los métodos usuales de recepción de datos, como lo son las encuestas o programas agendados que buscan rigurosidad y poca flexibilidad en los tratamientos [23].

Un ejemplo de estos es Brain-Computer-Interface (BCI) utilizado para el tratamiento y rehabilitación en las funciones de memoria y la concentración mediante la utilización de juegos de computadora que relacionan el control de las señales cerebrales y la computadora, logrando así aumentar el nivel de atención del jugador [24]. Otro estudio relacionado con el tema, muestra el sistema de telemedicina para el TDAH basado en múltiples agentes, en los cuales se realiza a los pacientes un tratamiento mediante biofeedback EEG instrumentados en el hogar, con el fin de llegar a un mejoramiento en las condiciones de niveles de sus funciones operativas, como: la memoria operativa, la atención y demás aspectos,

los cuales muestran resultados prometedores en los avances e implementación de aspectos biomédicos [25].

El desarrollo de juegos serios para la rehabilitación y tratamiento del TDAH requiere tener en cuenta estándares en su diseño como son reglas, retos, interacciones, objetivos específicos y objetivos implícitos [26, 27], que son la parte más importante y lo que caracteriza a un juego serio. Entre los objetivos implícitos están la mejora de habilidades, obtención de conocimiento, obtención de experiencias y recuperación física [26]. Con respecto a la implementación del entorno, la inmersividad junto con la simplicidad son claves para una adecuada adquisición de las habilidades objetivo [28], asimismo, el monitorear este proceso con adquisición de datos permite realizar seguimiento del usuario. Por otro lado, está demostrado que los niños con TDAH responden de mejor manera a estímulos positivos [29] además de apuntar al mejoramiento de las habilidades requeridas por el juego serio como lo son: autocontrol, planificación, autorregulación emocional, retraso en la respuesta impulsiva, reacción controlada y atención a los detalles [30].

En la actualidad existen algunas aplicaciones utilizadas en la rehabilitación de TDAH como los son: Plan-It Commander [29], Braingame Brian [31], Antonyms [30], BRAVO [32], entre otras [33, 34, 35]. Presentan deficiencias similares en cuanto a interoperabilidad de los datos o contenido, flexibilidad cognitiva, personalización, modularización del contenido (dividir episodios o escenas del juego en trozos más pequeños para una reorganización flexible), la medición técnica de los estados intrínsecos o capacidades físicas de los jugadores (ej. medir la inmersión de los jugadores), así como la transferencia de habilidades adquiridas al mundo real [36]. Para este último [37, 38], muestran que la transferencia de habilidades puede tener mayores beneficios si se implementan las tecnologías que ligan lo virtual con lo real, tales como realidad virtual y realidad aumentada, esta segunda siendo más relevante pues liga la realidad de una manera interactiva.

En relación con la utilización del software, se evidencian algunos ejemplos relevantes, dentro de los cuales se encuentra el desarrollado en [39] donde mediante la utilización de un software de realidad aumentada desarrollado en framework ARKit 2, propone validar la idea que la realidad aumentada puede ayudar a mejorar el nivel de atención. Se probó en una población de niños con TDAH de entre 7 y 11 años, en la cual mediante un test de cancelación se logró una mejora de 42 % en el promedio del puntaje obtenido en el juego. Otro estudio relacionado en el cual mediante la implementación de una aplicación móvil basada en realidad aumentada, donde el problema a tratar era la atención, se validó con una muestra de 9 niños divididos en en dos grupos: 5 de control y 4 de validación. Para la evaluación, se hicieron dos pruebas psicológicas: figuras y stroopen. La prueba de figuras mostró resultados positivos con una mejora en el puntaje global del 47,92 % para el grupo de control y del 9,82 % para el grupo de validación [40].

Sumado a lo anterior, se podría mencionar otro ejemplo: este es el caso de ATHYNOS, otro software basado en realidad aumentada donde se incluye un espacio de actividades diseñado según prácticas de intervención adecuadas establecidas por expertos en la materia del TDAH, además de una interfaz de usuario natural (NUI) para ayudar a la integración

de movimientos corporales de los niños. La prueba utilizó un total de 11 niños de entre 7 y 10 años, diagnosticados con TDAH. Los resultados obtenidos mediante un análisis estadístico, mostraron una mejora significativa en el nivel de concentración [33]. Además, en el proyecto AHA (ADHD-Augmented), entorno integrado de aprendizaje en realidad aumentada basado en la web y que tiene como objetivo mejorar las dificultades de lectura y ortografía de niños diagnosticados con TDAH, hicieron un estudio piloto con 117, en el cual se apreció mediante un análisis estadístico, una mejora de desempeño en los resultados de aprendizaje [41].

Por otra parte, se encuentra a AR-Therapist, donde se presentan una integración terapéutica dentro del entorno de realidad aumentada en línea. Los pacientes juegan un juego que ha sido diseñado para atraer su atención con una guía de instrucciones de voz/señas, mientras que los resultados y desempeño del paciente en el juego son almacenados en una base de datos donde el terapeuta pueda hacer control del progreso del tratamiento. Con esto se logra un excelente nivel de accesibilidad para las necesidades de cada paciente, ya que imita los roles del terapeuta a través de un entorno de realidad aumentada y el monitoreo en tiempo real por parte del terapeuta [42].

En el marco de la personalización se sugiere una gama más amplia de niveles y dificultad para mantener la atención de los niños y mejorar sus destrezas individuales [33]. Una de las soluciones implementadas para este problema es realizar una configuración automática de los parámetros del juego; sin embargo [43] menciona que esto no es lo ideal, pues es recomendable entregar al terapeuta y al usuario, dado el caso, el control de la configuración de parámetros [44, 26]. Lo anterior se conoce como un enfoque de rehabilitación orientado al paciente en el cual el terapeuta puede diseñar las tareas o ejercicios específicos para cada paciente impulsado por su experiencia y apoyado por la tecnología [45]. Como por ejemplo en [46], donde se genera protocolos acordes a las características de la patología y sesiones de ejercicios que aumentan su complejidad e intensidad conforme avanza la terapia.

Continuando con la personalización de los ejercicios en los juegos, el control sobre la configuración de niveles de dificultad y una fuerte relación entre lo real y lo virtual, este trabajo de grado busca proponer una aplicación con AR como herramienta de apoyo a los terapeutas en el tratamiento del TDAH, con el fin de responder la pregunta ¿Qué características se requieren en un juego serio con realidad aumentada destinado a la generación de ejercicios estructurados en complejidad e intensidad para el tratamiento de niños con TDAH?

Objetivos

Para el presente trabajo se proponen los siguientes objetivos.

Objetivo general

Proponer una aplicación de realidad aumentada para la realización de ejercicios estructurados en complejidad e intensidad en niños con déficit de atención e hiperactividad.

Objetivos específicos

1. Concebir una interfaz de usuario para que el terapeuta configure la complejidad e intensidad de un ejercicio cognitivo en realidad aumentada.
2. Desarrollar un módulo de generación de reportes de desempeño en el cual se especifique la media y la varianza del tiempo requerido para realizar el ejercicio cognitivo.
3. Evaluar la usabilidad de la aplicación propuesta con base en una muestra de especialistas en el área.

Estructura de la monografía

La presente monografía se divide en un total de 4 capítulos, el primer capítulo (abordado previamente) presenta una introducción conceptual al tema de investigación, el estado del arte y los objetivos planteados para el proyecto. El capítulo 2 esta compuesto por los materiales y métodos del proyecto, estos son la metodología utilizada, la selección de características para el desarrollo de un juego serio, el diseño de la arquitectura del software, la justificación de herramientas software, la creación de la base de datos. El capítulo 3 abarca la descripción de módulos, el protocolo experimental, la evaluación realizada a un grupo de expertos, la validación de parámetros, la escala de medición para dichos parámetros y el análisis de los resultados. Finalmente, en el capítulo 4 se presentan la discusión y conclusiones, además de trabajos futuros.

Capítulo 2

Materiales y métodos

Los juegos serios (JS) están pensados explícitamente con un propósito educativo, de entrenamiento o rehabilitación y no están destinados a ser jugados principalmente por diversión; esto no significa que dichos juegos no sean o no deban ser entretenidos, más aun siendo destinados para niños que son los usuarios finales de esta aplicación. Existen unos criterios en la literatura que dan unas pautas, sin ser estrictas, para el diseño de un JS; en este caso se utilizó lo planteado en [47], en el cual se recopila y caracteriza 24 juegos serios en los últimos 20 años, clasificando sus metodologías con base en características comunes encontradas, además del porcentaje de JS que usaron ese criterio teniendo como margen mínimo para este proyecto un 50 %.

Este capítulo abordará la metodología scrum, la cual fue seleccionada para la realización de este proyecto, la selección de criterios de diseño que requiere el JS obtenidos de [47], la arquitectura del software propuesto, la justificación de herramientas software, en donde se mencionan los motores de desarrollo para juegos, los software de desarrollo para la realidad aumentada, los motores de base de datos, además de las librerías y paquetes utilizados para todos los mencionados. Para finalizar se abordará una sección para la creación de la base de datos. Las recomendaciones sobre los requisitos computacionales para la ejecución de las aplicaciones se encuentran al final de los resultados.

2.1. Metodología Scrum

La metodología Scrum fue la base para el proyecto en mención. Ésta metodología consta de una serie de reuniones, que tienen como objetivo construir las características más importantes de un producto. A medida que avanzan las reuniones se van agregando otras características y funcionalidades; éstas deben ir adaptadas de acuerdo a los comentarios de clientes y las partes interesadas. Básicamente está basada en un desarrollo incremental, en donde se construye un producto o sus características, a medida que ocurren las reuniones.

La metodología cuenta con unos roles o perfiles específicos para ayudar con el desarrollo de las actividades. Dichos roles son los siguientes:

- **Product Owner:** Es el encargado de agregar valor al producto, definiendo los

objetivos asegurando que se cumplan para obtener los resultados deseados.

- **Scrum Master:** Es la persona que ayuda al equipo a mantenerse centrado en los objetivos del proyecto. Ayuda a resolver los problemas que puedan surgir, además de aconsejar al product owner para maximizar el retorno de la inversión.
- **Developers:** Son las personas que realizan las tareas para cumplir con los objetivos, comprometiéndose con el desarrollo de las actividades de cada reunión. Este rol es el encargado de darle la mejor calidad al producto final.

Con base en la metodología descrita, a continuación se describe el procedimiento realizado en la presente investigación, así como los roles que asumieron diferentes profesionales que aportaron en la planeación y desarrollo del mismo.

Para el seguimiento de tareas de cada reunión se utilizó un registro de historias de usuario, como se muestra en el formato de la tabla 1, se registraron las funcionalidades o características que los product owners consideraron que eran necesarias para incluir en la herramienta a desarrollar. Para dicha característica se plantearon además unos criterios de aceptación con los cuales se evaluó el cumplimiento de la misma dentro del software. El formato de la tabla 1 permitió llevar un control de las funcionalidades y características requeridas, al numerarlas y documentarlas para su implementación, se pudieron recopilar en la tabla 2. Por otra parte el equipo o scrum team que ayudo a formar estas características fue el siguiente:

- **Product Owner:** Elizabeth Roldán Gonzales, fisioterapeuta, Mg. en Neurorehabilitación y Mg. en educación. Luisa Fernanda Ordoñez, Fonoaudiologa Magister en educación desde la diversidad y diplomada en neurodesarrollo. Saidy Rosero, terapeuta ocupacional, Mg. en Educación desde la Diversidad.
- **Scrum Master:** Msc.Diego Enrique Guzmán Villamarin.
- **Developers:** Jhoan Sebastian Hurtado Campo y Rosmer Hasan Yepes Hoyos.

#03	Como «Cliente» quiero «Ambiente de supermercado» para poder «Atraer la atención de niños de una edad entre los 7 y 10 años»
CRITERIOS DE ACEPTACIÓN	
1. Ambiente tridimensional	Generar un estante con productos de supermercado para su búsqueda.
2. Ambiente estático	Una vez generado el ambiente se conserve la posición de este para realizar el ejercicio de búsqueda.
3. Objetos variados	Tener diferentes tipos de objetos en el estante y una variante de este en color para su búsqueda

Tabla 1: Ejemplo de historia de usuario.

Característica	Beneficio	Criterio de aceptación
Que la temática del juego sea de búsqueda contrarreloj.	Enfocar el tratamiento en las funciones ejecutivas y la atención.	Establecimiento de un tiempo límite para la ejecución del ejercicio.
Diseñar las tareas a realizar.	Estructurar un plan de actividades en torno al juego.	Asignación de nombre al ejercicio, configurar la complejidad e intensidad mediante retos/misiones de búsqueda de uno o varios objetos específicos en realidad aumentada.
La actividad a realizar incluya el movimiento físico	Incluir beneficios en el comportamiento y de coordinación motora	Establecer el objeto en realidad aumentada, de manera que la búsqueda de los objetos implique efectuar movimiento en el espacio de realización.
Ambiente de supermercado.	Atraer la atención de niños de una edad entre los 7 y 10 años.	Generar un estante tridimensional con productos de supermercado. Una vez generado el estante, conserve su posición.
Interacciones simples.	Facilitar el uso de la aplicación y el desarrollo de la actividad en AR.	No usar gestos complejos para la interacción, usando simplemente un toque sobre el objeto presentado en pantalla para su selección. Mantener la pantalla bloqueada en una sola orientación.
Variedad de objetos.	Estructurar la actividad en torno a un objeto.	Se cuente con objetos orgánicos e inorgánicos y a su vez con una variante visual de cada objeto para mayor personalización.
Que se pueda trabajar con varios pacientes.	Tener un manejo factible de los pacientes.	Amplia capacidad en el registro de pacientes.
Acceso simple al juego serio.	Facilitar el ingreso del infante al juego.	Realizar el ingreso al juego mediante un único usuario fácil y corto, para que sea sencillo de recordar e ingresar.

Continúa en la siguiente página

Característica	Beneficio	Criterio de aceptación
Gestión controlada de los datos del paciente.	Permitir la manipulación de los datos registrados de los pacientes.	Agregar botones para modificación de los datos y para la eliminación de los pacientes.
Establecer intervalo de tiempo para el ejercicio.	Estructurar un plan de ejercicios personalizado.	Los rangos de tiempo establecidos en el ejercicio se configuran desde días, semanas y meses.
Fijar un estímulo adicional al ejercicio.	Ayudar a la concentración y la atención selectiva.	Que este estímulo sea un objeto no convencional en el ambiente de minisúper.
Escoger la distribución de los objetos.	Tener un mayor grado de personalización.	Los objetos puedan ubicarse en cualquiera de las partes disponibles del estante.
Interfaces simples.	Facilitar el entendimiento de las opciones del juego.	Las interfaces no estén sobrecargadas de información y muestren solamente lo relevante.
Indicador del progreso en tiempo real del ejercicio	Estimular al niño a finalizar el ejercicio	Mostrando la cantidad de objetos faltantes.
Incentivo visual al terminar el ejercicio.	Estimular al niño a seguir con el juego.	Agregando interfaces visualmente llamativas.
Recopilar datos.	generar estadísticas de progreso.	Generar un reporte en PDF con las estadísticas del progreso de acuerdo al ejercicio realizado, en estas estadísticas deben incluir la media y la varianza del tiempo requerido para realizar el ejercicio.
Generar diagramas de los resultados.	Facilitar el análisis de resultados.	Agregar al PDF gráficos de líneas con la información específica de los resultados obtenidos de cada ejercicio.

Tabla 2: Historias de usuario.

2.2. Selección de características

Teniendo en cuenta lo mostrado en [47] y la información de la tabla 2, la cual contiene las historias de usuario para el desarrollo del proyecto, se dividió la selección de características en 3 grupos principales, en donde se combinan tanto las propiedades que debe tener un juego serio y las características específicas de las historias de usuario.

- Establecer objetivos del juego.
- Caracterizar al público objetivo.
- Incorporar elementos del juego.

Establecer objetivos del juego

Esta característica determina los objetivos y propósitos a alcanzar por medio del JS. Se establece que el juego debe enfocarse en factores determinados para un tratamiento más amplio como la atención selectiva, la atención sostenida, la velocidad de procesamiento y la relación espacial, por lo que se discute un modelo de juego utilizado en terapias denominado “¡Pictureka!”. Este juego de mesa, sugerido por las product owner, tiene como objetivo la búsqueda de un personaje en el tablero que se selecciona al azar de unas tarjetas con un tiempo específico de búsqueda; la observación, la atención y la velocidad de procesamiento son el factor principal de este juego.

Por otro lado, en esta sección de establecimiento de objetivo se determinaron también las conductas deseadas. Una éstas conductas deseadas en el desarrollo es el movimiento o actividad física, que según [48] al ser integrada en los tratamientos cognitivo-conductuales, puede producir beneficios a nivel cognitivo, ejecutivo, coordinación motora, rendimiento académico, comportamiento y en las habilidades sociales de los niños con TDAH, así como la posibilidad de reducir los síntomas propios de este trastorno.

Caracterizar al público objetivo

Se determinó como público objetivo, niños y niñas con edad entre los 7 y 10 años. Junto con los product owners se estableció este margen debido a que los niños en esta edad pueden afrontar de mejor manera el reto del primer contacto con un entorno AR. Además, debido a que la tipología del juego se enfoca en actividades regladas como lo es la búsqueda en un tiempo definido, también se estableció que la temática del juego debería ser enfocada a tareas un poco más adultas pues los niños con estas edades tienden a querer realizar actividades que los adultos hacen, como por ejemplo; ir de compras, efectuar una profesión, entre otras. En respuesta a esto, se definió el ambiente del juego como un entorno de supermercado en el cual la búsqueda de objetos de una lista se acopla de manera ideal a las necesidades requeridas. Teniendo en cuenta que los niños con TDAH presentan dificultades al momento de concentrarse en una actividad específica, se agregan estímulos visuales que los ayuden a permanecer interesados en el desarrollo de los ejercicios y en la utilización de juego.

Incorporar elementos del juego

Una de las primeras incorporaciones del juego es la parte del diseño y mecánicas. Ya se ha definido el ambiente y el entorno en el que se va a realizar, siendo este un supermercado en realidad aumentada; al tratarse de búsqueda de objetos lo recomendable es no implementar mecánicas complejas de selección, porque siendo una búsqueda contrarreloj, el enfoque principal es encontrar el objeto haciendo uso de la atención sostenida.

En esta sección también se establece el diseño instruccional, lo que desemboca en la configuración de las tareas a realizar; tareas que se ligan totalmente al ajuste de la complejidad e intensidad, pudiendo ajustar tanto los intervalos de tiempo en el que se pueden realizar los ejercicios, así como también la distribución de los objetos en el ambiente de supermercado. Además se toman aspectos como la limitación de tiempo, para realizar los objetivos establecidos; de igual manera el establecer los modelos 3D de los objetos con calidad gráfica reducida para su funcionamiento en dispositivos de gama media. Las características de recopilación de datos que se generen con respecto a las tareas realizadas por el infante, van más enfocadas a la generación de reportes de resultados.

2.3. Arquitectura del software propuesto

Para el diseño de la arquitectura del software se estableció una relación modular. Esta se agrupa en tres que son los encargados de efectuar y cumplir las necesidades establecidas en las historias de usuario realizadas en conjunto con el grupo de terapeutas. Estos 3 módulos se efectuaron de tal manera que cada módulo es una aplicación/software única que se comunica con las otras dos. Teniendo en cuenta que el uso de Python facilita la generación de reportes por todas las librerías matemáticas que posee, se estableció como el lenguaje utilizado para el módulo de reportes y el uso de C# en Unity para la creación del juego serio y la configuración de este; teniendo en cuenta que el paciente no tendrá acceso a ninguna función del terapeuta, se optó por separarlo en dos aplicaciones. El primer módulo es una aplicación únicamente del juego serio para el paciente, en el segundo una aplicación que permite configurar aspectos de complejidad e intensidad dentro del juego serio y un tercer módulo para la generación de reportes de desempeño y progreso, por tanto, el terapeuta tendrá acceso al módulo de configuración y de generación de reportes. Por otro lado, el paciente solo tendrá acceso al juego serio.

La figura 1 presenta la arquitectura propuesta de una manera generalizada, las 3 aplicaciones se comunican por medio de la base de datos y cada una tiene un objetivo que combinado a las otras dos permite cumplir cada necesidad requerida.

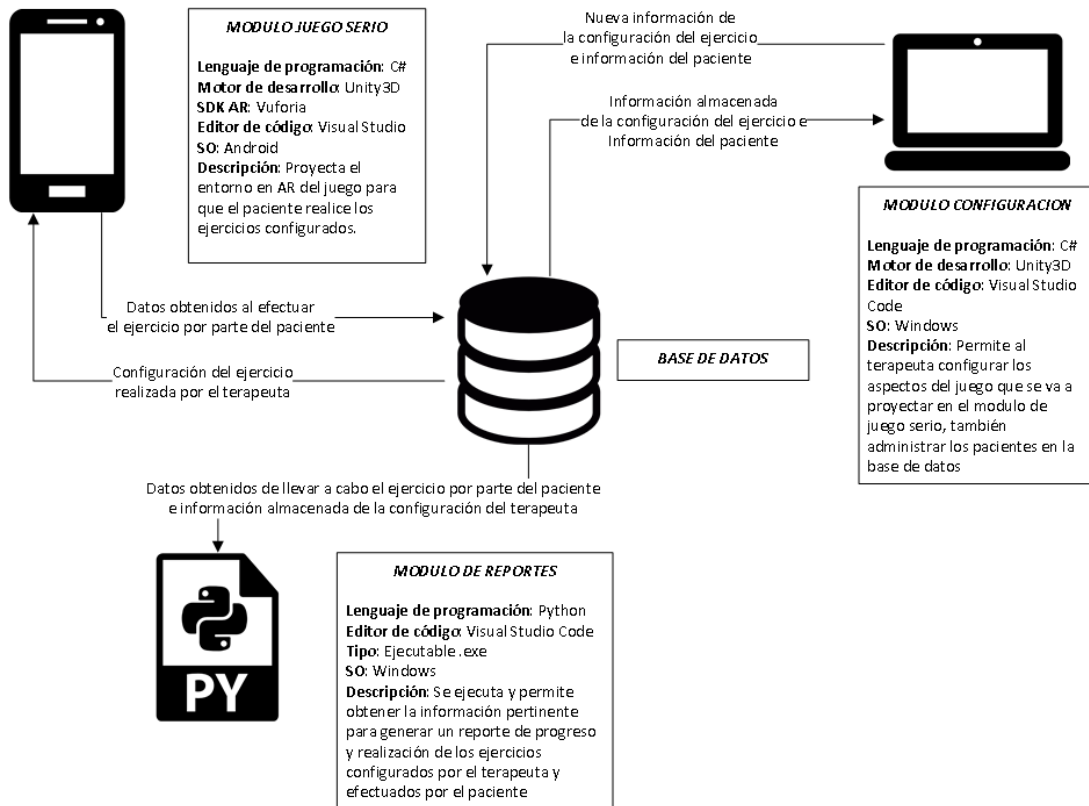


Figura 1: Arquitectura generalizada.

2.3.1. Lenguaje Unificado de Modelado

El Lenguaje Unificado de Modelado (UML) permite una visualización más clara y estandarizada del sistema o software por medio de varios tipos de diagramas que lo comprenden. Con este lenguaje se puede expresar de manera gráfica tanto la estructura como el comportamiento, por lo que se usaron unos tipos específicos de diagramas para representar la estructura de MiniSuper AR. El UML se divide en dos tipos de diagramas para representar estructura y comportamiento; cada uno de estos puede ser representado con unas subclases de diagramas. Para motivos explicativos se tomó un representante de cada uno, siendo el de estructura el “Diagrama de componentes” y el de comportamiento el “Diagrama de caso de uso”.

Diagrama de componentes

El diagrama de componentes representa la relación de los componentes individuales del sistema, mediante una vista de diseño estática. En este diagrama se ilustran aspectos de modelado lógico y físico; al estar refiriéndonos a UML, estos componentes son partes modulares independientes y se comunican con otros por medio de interfaces. Utilizando interfaces propias o suministradas por otro módulo, este diagrama se puede ver en la figura 2.

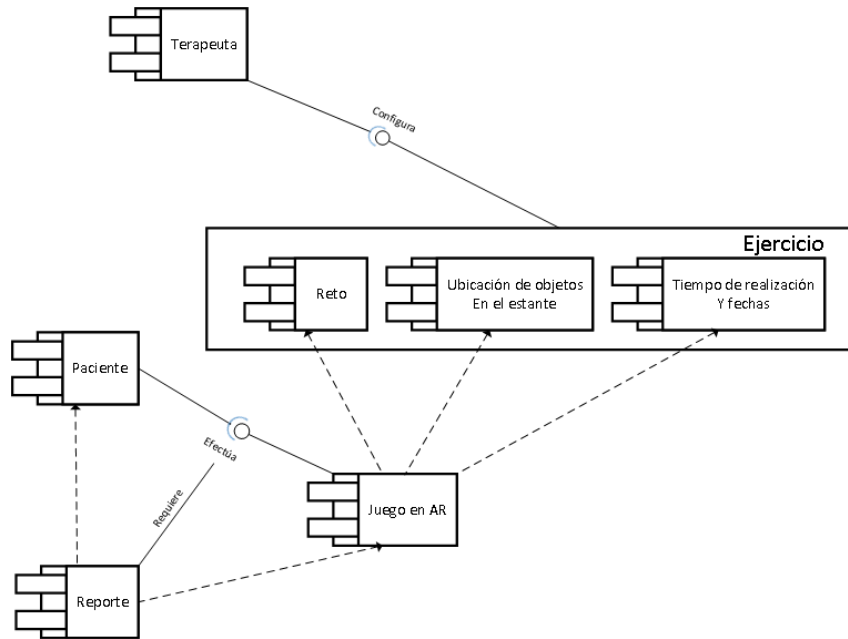


Figura 2: Diagrama de componentes MiniSuper AR.

Diagrama de caso de uso

Los diagramas de caso de uso sirven para mostrar de una manera más específica la comunicación y el comportamiento de un sistema mediante la interacción con los actores, usuarios y/u otros sistemas. En otras palabras, un diagrama de caso de uso nos muestra la relación entre quienes usan la herramienta y las funcionalidades o casos de uso que esta brinda, el diagrama de caso de uso está representado en la figura 3.

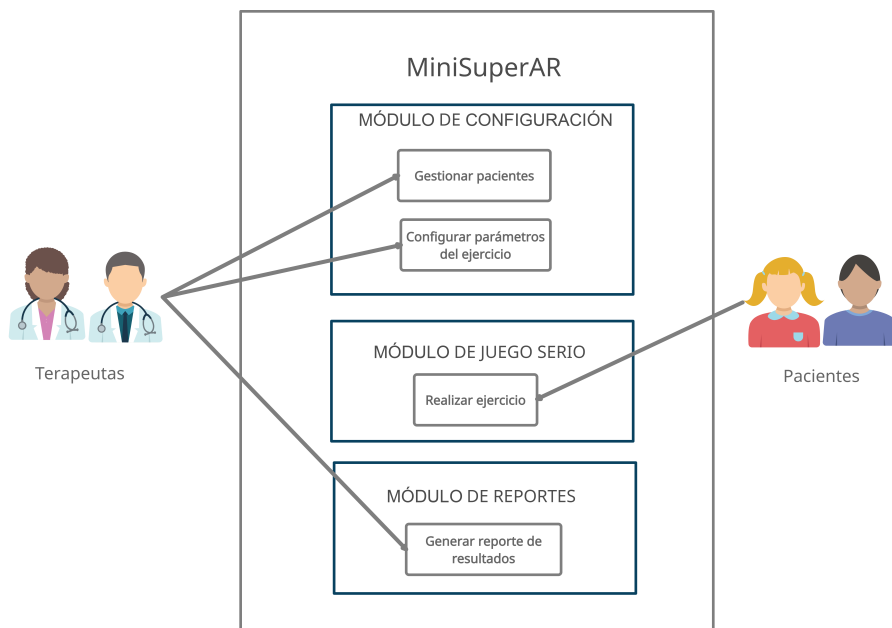


Figura 3: Diagrama de caso de uso.

2.4. Justificación de la herramienta software

2.4.1. Herramientas de desarrollo

Para el desarrollo de las aplicaciones, fue necesario obtener un motor de desarrollo de videojuegos con las características necesarias que ayudaran al cumplimiento de los objetivos propuestos. Se hizo una búsqueda de los software de desarrollo más destacados y se establecieron en la tabla 3.

Nombre	Descripción	Requerimientos	Tipo de exportación	Gratuito
Game maker studio[49].	Este software de desarrollo de videojuegos está enfocado en el desarrollo de juegos 2D, está basado en el lenguaje de programación C y con un kit de desarrollo (SDK).	Ram: 8 GB. CPU: Intel quadcore 64bit o más. GPU: Tarjeta gráfica dedicada OpenGL4.	Depende del tipo de licencia que se adquiera, el más básico permite Windows, Ubuntu y Mac.	No, el trial es muy limitado.
Unity 3D[50].	Es multiplataforma y cuenta con una amplia popularidad, lo que hace que la documentación que se encuentre sea abundante, permite trabajar en juegos 2D y 3D, también basados en cámaras 3D y muchas librerías (paquetes) con funcionalidades adicionales gratuitas como realidad aumentada; en lenguajes de programación está C, Unity Script, Python y Boo.	Ram: 4 GB. CPU: Soporte para el conjunto de instrucciones SSE2 o más. GPU: Capacidad para DX10. Principalmente la complejidad del proyecto establece las características, por lo que un proyecto sencillo requiere menos recursos computacionales y uno grande, más.	Permite exportar a una amplia variedad de plataformas, Windows, Mac, Linux, Android, iOS, Playstation, Xbox, Nintend3ds, etc.	Sí, su licencia gratuita tiene la condición de que al inicio del proyecto salga el logo de Unity, también una versión de paga que suprime este logo al inicio.

Continúa en la siguiente página

Nombre	Descripción	Requerimientos	Tipo de exportación	Gratuito
Godot[51].	Es multiplataforma, permite el desarrollo de juegos en 2D y 3D, también como con Unity permite crear juegos 2D basados en cámaras 3D. Utiliza lenguaje de programación en C, C++ y GDScript (similar a Python).	Ram: 3 GB. CPU: Intel Celeron Dual-Core T3300 @ 2.00GHz o más GPU: Tarjeta gráfica dedicada OpenGL2.1.	Permite exportar a Windows, Mac, Linux, Android, iOS, HTML5.	Sí.
Gamesalad[52].	Este motor se centra en el desarrollo de juegos en 2D, sin necesidad de codificar pues usa programación en bloques lo que lo limita a proyectos de no mucha complejidad y facilita la enseñanza de la programación a educadores.	Ram: 2 GB. CPU: Core 2 Duo o más. GPU: Tarjeta gráfica dedicada OpenGL2.0.	Permite exportar a Windows, Mac, Android.	Sí, posee una licencia paga con más características.

Continúa en la siguiente página

Nombre	Descripción	Requerimientos	Tipo de exportación	Gratuito
Stencyl[53].	Se enfoca en la creación de juegos en 2D, utilizando una técnica similar a la de Gamesalad, que es programación por bloques arrastrando y conectando entre sí las funciones que vienen predefinidas.	Ram: 2 GB. CPU: 2GHz CPU con uno o más núcleos. GPU: Tarjeta gráfica dedicada OpenGL2.0.	Permite exportar a Windows, Linux, Mac, Android, iOS, HTML5.	Sí, funciona igual que Unity pues da todas las funcionalidades, pero en el proyecto al inicio se muestra el logo de Stencyl. Esto se puede evitar, adquiriendo una licencia paga.
CryEngine[54].	Es un potente motor de desarrollo de juegos en 3D maneja lenguajes como C, C++ y LUA, especialmente este motor se usa para juegos de alta exigencia gráfica pues en un principio fue un motor de demostración para la empresa Nvidia.	Ram: 4 GB. CPU: Intel Dual-Core 2GHz o AMD Dual-Core 2GHz. GPU: NVIDIA GeForce 400 series o AMD Radeon HD 6000 series.	Permite exportar a Windows, Ps4, Xbox one y Oculus.	Sí, posee una licencia de pago.
Amazon lumberyard[55].	Es un motor de videojuegos AAA multiplataforma integrada a AWS y está basado en CryEngine, maneja lenguaje C++ y LUA.	Ram: 8 GB. CPU: 3 GHz quad-core. GPU: 2 GB VRAM DirectX 11 o más.	Permite exportar a Windows, Linux, Android, iOS y Mac.	Sí, sin embargo, para juegos que requieran conexión a internet se debe usar un servidor de AWS o uno propio.

Continúa en la siguiente página

Nombre	Descripción	Requerimientos	Tipo de exportación	Gratuito
Cocos 2D-x[56]	Este motor se enfoca en el desarrollo de juegos en 2D, libros interactivos, demostraciones y otras aplicaciones gráficas. Utiliza principalmente el lenguaje de programación de Python, pero también hay versiones de C++.	Ram: 8 GB. CPU: 3 GHz quad-core. GPU: 2 GB VRAM DirectX 11 o más.	Permite exportar a iOS, Android, Windows y Mac otras versiones permiten además exportar a Xbox.	Sí, al igual que Godot está bajo licencia de MIT, se debe dar crédito en la documentación del juego o proyecto realizado.

Tabla 3: Motores de desarrollo.

Según lo mencionado anteriormente y teniendo en cuenta los siguientes criterios de selección:

- Permita el desarrollo de juegos en 3D.
- Cuente con amplia documentación sobre el motor de desarrollo.
- Sea gratuito.
- Permita exportar a Windows, Android e iOS.
- Con librerías para trabajar con realidad aumentada, con soporte y documentación.

Se optó por elegir a Unity3D, ya que cumple con todas las características, además de que se tiene un conocimiento previo y posee una gran variedad de guías para sus diferentes herramientas y librerías.

2.4.2. Kit de desarrollo de software (SDK) para realidad aumentada

La realidad aumentada es una tecnología que permite mezclar el entorno real y virtual, de este modo permite interactuar con la realidad física en tiempo real, lo que brinda amplias posibilidades de uso gracias a su flexibilidad [57]. Se hizo una investigación teniendo en cuenta que el motor de desarrollo que se escogió fue Unity3D, buscando así kits de desarrollo de software, dedicados a la realidad aumentada, compatibles con el motor de desarrollo para el juego. Esta información se presenta en la tabla 4.

Para el juego serio se requiere que el SDK de realidad aumentada sea flexible, con la documentación sobre sus herramientas, además debe de contar con licencia gratuita. El SDK de vuforia para Unity fue la opción que cumple con estas necesidades de manera más acertada y cómoda, además de que cuenta con amplia guía sobre desarrollo y se encuentran variedad de proyectos abiertos.

Nombre	Descripción	Características	Gratuito
Vuforia [58]	Este kit de desarrollo software para dispositivos móviles permite el desarrollo e implementación de aplicaciones en realidad aumentada de una manera sencilla.	Vuforia utiliza una tecnología concreta de visión artificial la cual le permite rastrear y reconocer imágenes planas y objetos 3D en tiempo real, reconocimiento de texto, el objetivo que se fija con este SDK no se pierde ni siquiera cuando el dispositivo móvil se mueve, detección y rastreo simultáneo, acceder a base de datos locales y en la nube.	Sí, pero presenta una marca de agua de la compañía en una esquina cuando se usa la cámara. Esto se puede evitar, adquiriendo una licencia.
ARCore[59]	Es un kit de desarrollo de Google creada para experimentar con realidad aumentada, permitiendo a los dispositivos móviles detectar su entorno, comprender el mundo e interactuar con la información que se encuentra en él.	Utiliza 3 capacidades clave con la cámara del dispositivo móvil, que son el seguimiento del movimiento, la comprensión del medio ambiente y la estimación de la luz. Cuenta con rastreo de posición del dispositivo mientras se mueve, junto con su orientación, incluye también un sistema de detección de superficies planas.	Sí, su licencia gratuita tiene la condición de que al inicio del proyecto salga el logo de ARCore.

Continúa en la siguiente página

Nombre	Descripción	Características	Gratuito
ARkit [60]	Brinda a los usuarios una experiencia más realista pues utiliza Visual Inertial Idometry (VIO) para rastrear con precisión el entorno, VIO combina los datos del sensor de la cámara del dispositivo a los datos de algoritmos incorporados a Arkit.	Funciona únicamente en dispositivos con iOS, no solo puede detectar planos horizontales, sino que también rastrea y coloca objetos en un punto de característica más pequeño en el espacio, otra característica interesante que tiene es el de emplear el sensor de la cámara para estimar la cantidad correcta de iluminación de los objetos.	Sí.

Tabla 4: Kit de realidad aumentada compatible con Unity.

2.4.3. Motor de base de datos

Para la selección del motor de base de datos se realizó la búsqueda de las especificaciones sobre 4 motores, los cuales son: Firebase, SQLite, MySQL y MongoDB. En la tabla 5 se logra evidenciar un cuadro comparativo, donde se realizó la definición, características, ventajas y desventajas de los motores mencionados.

Nombre	Definición	Característica	Ventaja	Desventaja
Firestore	Es una plataforma ubicada en la nube que permite crear proyectos sin necesidad de un servidor y entrega funciones propias como autenticación vía email, redes sociales, Google, etc. Real time data base, Gosting y Cloud Messaging.	Es una base de datos no relacional y además posee ciertas características usando su licencia gratuita: 100 conexiones simultáneas, 1 GB de almacenamiento, transferencia mensual de 10 GB.	Es desarrollado por Google, es sin servidor, monitoreo de errores, es seguro, es compatible con plataformas como iOS, Android, aplicaciones web, Unity y C++.	Consultas lentas, no es de código abierto.

Continúa en la siguiente página

Nombre	Definición	Característica	Ventaja	Desventaja
SQLite	Es un motor de base de datos relacional, autónomo, está integrado en todos los teléfonos móviles y se usan para almacenamiento interno de las aplicaciones.	Tiene un formato estable, es multiplataforma y está contenido en una biblioteca relativamente pequeña escrita en lenguaje C y es totalmente autocontenida (sin dependencias externas).	Fácil de utilizar, ideal para desarrollo de aplicaciones móviles, ocupa poco espacio de almacenamiento, es de código abierto, accesible a través de una gran variedad de herramientas.	Es mono usuario, tiene limitaciones en el almacenamiento de datos, no tiene funciones de seguridad y administración de usuario, lo cual ocasiona problemas de seguridad, está orientada a aplicaciones que tenga un tráfico de datos bajo.
MySQL	Es el sistema de gestión de base de datos relacionales más usado del mundo, además de que cuenta con doble licencia, una open source y otra comercial con soporte. Adicionalmente, está integrado con PHP lo que lo hace más seguro y simple.	Está disponible en todas las plataformas y sistemas, utiliza varias herramientas para portabilidad, soporta procedimientos almacenados, triggers y funciones, tiene una sencilla curva de aprendizaje.	Posee una alta seguridad en los datos, bajo mantenimiento para proyectos pequeños.	Es de esquema fijo (Lo cual no permite almacenar datos distintos), una baja documentación a comparación de otras bases de datos, no es intuitivo.
MongoDB	Es un sistema de base de datos no relacional orientada a documentos, lo que significa que guarda los datos en documentos (BSON que es una representación binaria de JSON) y no en registros.	Está escrito en C++, se usa generalmente para aplicaciones que necesiten almacenar datos semi estructurados, se pueden realizar todo tipo de consultas además de poder realizarlas utilizando JavaScript.	Requiere pocos recursos de computación, tiene gran documentación, es de código abierto, tiene buen soporte para estructuras distribuidas.	Falta de estandarización, no hay soporte para transacciones por defecto.

Tabla 5: Motores de base de datos.

Se hizo una exploración a la base de datos de Sqlite y MySQL, pero teniendo en cuenta la arquitectura que se diseñó, la cual se aprecia en la figura 1, se concluyó que una base de datos de tipo relacional no favorecía el proyecto porque al no requerir una cantidad de datos muy elevada, las no relacionales cuentan con el beneficio de manejar arquitecturas jerárquicas, en especial Firebase que cuenta con funciones ya establecidas lo cual facilita la implementación usando Realtime Database, la cual no necesita de un servidor y aunque tenga limitaciones en su cuota de transferencia mensual y de almacenamiento, al manejar tan pocos datos se hace más que suficiente para el proyecto. Firebase fue la base de datos elegida para el proyecto.

Indagando más en la base de datos escogida, Firebase es una plataforma digital gratuita, comienza a cobrar únicamente cuando se maneja de manera más avanzada y experimentada con mayor flujo de datos y usuarios registrados en los proyectos creados, además de soportar diferentes plataformas móviles, lo cual es una característica muy acertada para el proyecto. Los servicios que ofrece Firebase cuenta con una categoría de desarrollo que incluye diferentes funciones y servicios necesarios para la creación de un proyecto de aplicación móvil y otras funciones tales como:

- **Real time data base:** En una base de datos en tiempo real alojada en la nube y que almacena los datos como JSON, permitiendo escribir y leer datos desde la aplicación a la base de datos en tiempo real.
- **Autenticación:** Es un sistema de autenticación de usuarios mediante email o redes sociales, lo que les permite identificarlo y poder acceder a sus características de manera segura.
- **Nube de almacenamiento:** Sistema donde los desarrolladores almacenan y envían archivos de sus aplicaciones (Como por ejemplo fotografías).
- **Hosting:** Servidor para publicar y alojar las aplicaciones de manera rápida y sencilla, además de ser un hosting estático y seguro.
- **Remote config:** Se utiliza para modificar diferentes aspectos de la aplicación sin tener que publicar una actualización de la misma.
- **Test lab:** Mediante dispositivos virtuales permite testear y realizar pruebas antes de publicar la aplicación, lo que ayuda a detectar posibles errores previo a subida.
- **Crash reporting:** Se utiliza para enviar notificaciones y mensajes reportando errores de las aplicaciones de varios usuarios en tiempo real.

Tal como se evidencia anteriormente, Firebase proporciona una gran cantidad de herramientas para el desarrollo de aplicaciones; sin embargo, para este proyecto fueron de mayor interés 2 de ellas: Real time data base y Autenticación, las cuales se usaron para la autenticación de usuarios registrados a la base de datos y la base de datos donde se guardan y extraen los datos que se manejan en los 3 diferentes módulos.

2.4.4. Librerías utilizadas

- **Librerías y paquetes utilizados en el módulo de configuración y el módulo juego serio.**

- **Vuforia Engine AR:** Plataforma para el desarrollo de realidad aumentada, con soporte para teléfonos, tabletas y gafas líderes. Se pueden agregar fácilmente la funcionalidad avanzada de visión por computadora a las aplicaciones de Android, iOS y UWP para crear mejores experiencias en realidad aumentada para interactuar de manera realista con los objetos y el entorno.
- **Unity UI:** Conjunto de herramientas para desarrollar interfaces de usuario para juegos y aplicaciones.
- **Visual Studio Editor:** Integración del editor de código para admitir visual studio como editor de código para Unity.
- **Unity Collaborate:** Sistema de control de versiones oficial de Unity. Facilita que se guarden, compartan y sincronicen proyectos de Unity.
- **Timeline:** Utilizado para crear contenido cinematográfico, secuencias de juegos, secuencias de audio y efectos de partículas complejos.
- **TextMeshPro:** Es una alternativa para el texto de la interfaz de usuario de Unity y la malla de texto heredada.
- **Test Framework:** Marco de prueba para ejecutar pruebas en modo de edición y reproducción en Unity.
- **Firebase:** Permite realizar acciones para conectar, almacenar y sincronizar datos en tiempo real con el servidor de base de datos Firebase.

- **Librerías utilizadas en el módulo de reportes.**

- **PyQt5 (5.15.4):** Es uno de los módulos más utilizados en la creación de interfaces gráficas para aplicaciones en Python y esto es debido a su simplicidad, se deben realizar instalaciones aparte de PyQt5 para poder usar este módulo, específicamente las siguientes con sus versiones compatibles a la 5.15.4; pyqt5-plugins (5.15.4.2.2), PyQt5-Qt5 (5.15.2), PyQt5-sip (12.9.0), pyqt5-tools (5.15.4.3.2), PyQt5Designer (5.14.1).
- **pyrebase4 (4.4.3):** Es un módulo que nos permite la comunicación con la API de Firebase, en otras palabras es un "wrapper" que contiene funciones de comunicación y de escritura con la base de datos que se va a utilizar y se vera adelante en el servidor de base de datos.
- **matplotlib (3.4.3):** Es una biblioteca para la generación de gráficos a partir de datos contenidos en listas o arrays.
- **numpy (1.21.2):** Es la extensión matemática de matplotlib, proporciona una API que recuerda a la de matlab para crear vectores y matrices, también con varias funciones matemáticas de alto nivel.

- **reportlab** (3.6.1): Es el motor encargado de crear documentos PDF complejos y basados en datos, como los que se obtienen en matplotlib y numpy.
- **random**: Nos permite utilizar la función random para manipular números aleatorios y está presente en la versión de Python.
- **math**: Es el módulo que permite usar funciones matemáticas estándar y está presente en la versión de Python.
- **pathlib** (1.0.1): Proporciona el manejo de rutas en los directorios para almacenar los reportes PDF y las carpetas donde se obtiene información visual gráfica para añadir a los reportes (Como los gráficos).
- **datetime**: Es un módulo que permite la manipulación de fechas y de tiempo, además de que forma parte de la versión de python.
- **Sys**: Es uno de los módulos principales de Python, viene con la versión instalada y provee funciones y variables que se utilizan para manipular diferentes partes del entorno de ejecución de Python.

2.5. Creación de la base de datos

El registro y creación inicial de la base de datos se encuentra en los anexos A y B, una vez creada la base de datos vacía, se empieza por definir la jerarquía y las diferentes ramas que la conforman, como se muestra en la figura 4. El nodo principal está en el motor para la base de datos, que como se mencionó anteriormente es Firebase. Después se agrega el usuario administrador, en donde estarán almacenados todos los datos de la jerarquía de la base de datos. De este usuario se desprenden las ramas para los terapeutas, los cuales tienen el rango más alto, debido a que tendrán toda la información relacionada a sus pacientes y los ejercicios. Las ramas que se desprenden de los terapeutas son: los usuarios registrados, la cantidad de usuarios registrados al terapeuta e información de registro del terapeuta.

La rama de usuarios que se desprende de cada terapeuta, contendrá a cada uno de los pacientes registrados para ese terapeuta específico. De cada usuario se desprende: el nombre completo del paciente, el documento del paciente, los ejercicios del paciente y los errores de los ejercicios. Las ramas siguientes se desprenderán de los ejercicios y de los errores.

De los errores se desprende únicamente el nombre de cada ejercicio, para saber los datos independientes de los ejercicios; de cada nombre de ejercicio se desprenden todos los intentos que se realizaron para desarrollar el ejercicio, y de cada intento contiene la información del número de aciertos, de fallas y el tiempo que se utilizó en ese intento.

De la rama de usuarios desprende igualmente el nombre de cada ejercicio. Para obtener la información independiente de éstos, de cada uno se desprende la información del intervalo de fechas en las que se pueden realizar (FechaInicio y FechaFin), la información de configuración del ejercicio (Estante) y el reto que se le asigna, donde se desprende la información sobre los objetos que requiere buscar y la cantidad de cada uno de ellos. En la configuración

del ejercicio se guarda la información de la posición del *Monster* en el estante, el tiempo que se tiene para efectuar el ejercicio y el nombre de los objetos que van en cada posición del estante.

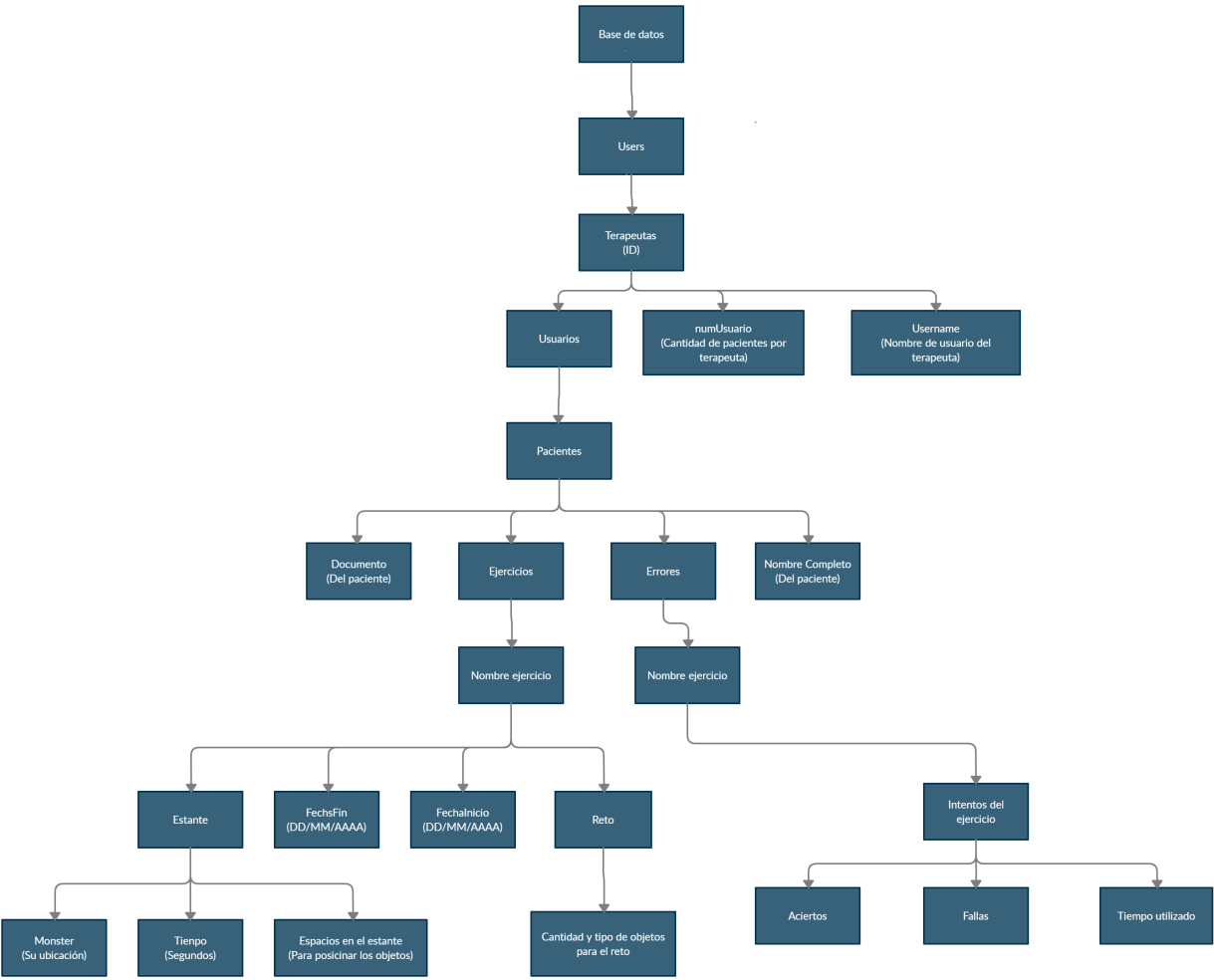


Figura 4: Jerarquía base de datos.

Capítulo 3

Resultados

Este capítulo abordará la descripción de los módulos mencionados en la sección de arquitectura, a su vez también la validación del módulo de reportes por medio de una prueba piloto. Seguido a esto se encuentra la selección del protocolo experimental, en donde se encuentra la selección de participantes, la selección de las preguntas de la encuesta y su método de medición. Para finalizar se procede a efectuar un análisis de los resultados obtenidos.

3.1. Módulo de configuración

El módulo de configuración es el que permite al terapeuta realizar todos los cambios a las variables que presenta el ejercicio para poder efectuarlo acorde a estas modificaciones. Este módulo es el que posibilita que los ejercicios puedan ser estructurados en complejidad e intensidad, por consiguiente y teniendo en cuenta la comodidad que presenta, este se ejecuta en computador. Para su diseño no se requirieron animaciones en la definición de las características a configurar, ya que esto favorece a que el mismo pueda funcionar bien en computadores con componentes no muy elevados en términos de potencia.

Adicional a lo anterior, también podrá administrar los pacientes que estén bajo su tratamiento, recordando que el paciente tendrá como acceso para el juego serio un usuario único que le suministrará el terapeuta de forma personalizada a sus gustos, de una manera hace que sea más fácil para recordar. También con esto se evitan los correos electrónicos o contraseñas que pueden ser más complejos para un niño en esos rangos de edad. Respecto al color usado para el módulo, se seleccionó una gama de colores azules, pues al ser un software relacionado con el sector salud este color lo representa.

Estructura del módulo de configuración

Buscando la simplicidad, se identificó que el software debería funcionar de manera jerárquica, haciendo que el terapeuta tenga que hacer unos ciertos pasos de manera secuencial, esto con el fin de no crear muchas rutas opcionales y facilitar el acceso a todas las funcionalidades que se vayan a presentar. Para visualizar esto de mejor manera, la figura 5 es

un diagrama de flujo simple que muestra la estructura de las interfaces del módulo (Los cuadros grandes representan una interfaz principal y los pequeños una interfaz secundaria).

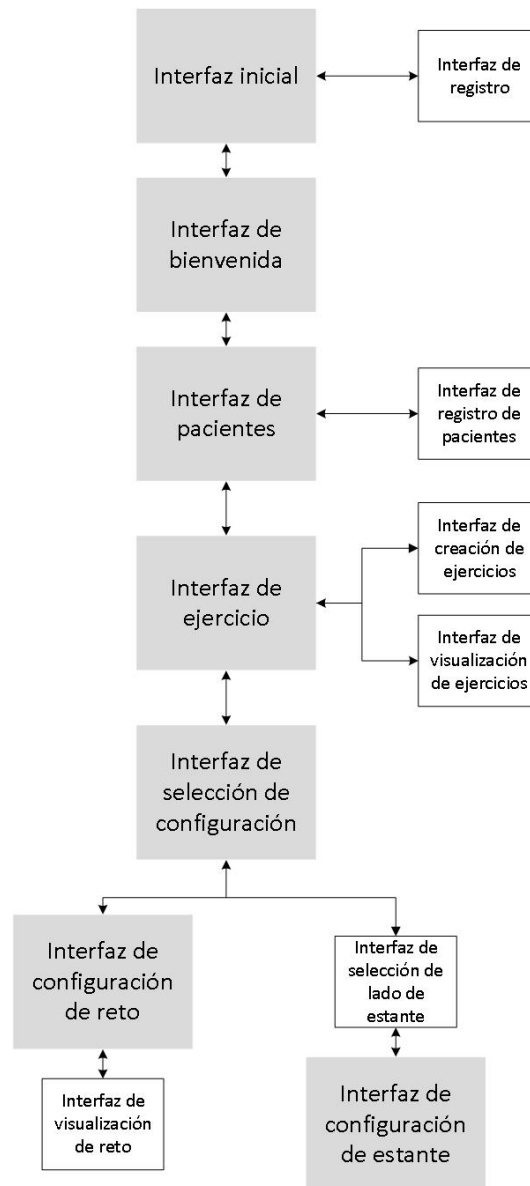


Figura 5: Flujo de interfaces del módulo de configuración.

Interfaz inicial

La interfaz inicial es la primera que verá el terapeuta, se aprecia en la figura 6, una vez ejecute este módulo, esta interfaz es la encargada de la etapa de autenticación del usuario. Esto quiere decir que desde esta interfaz se podrá iniciar sesión o en dado caso de no poseer una cuenta (siendo su primera vez en el módulo), llevarlo a la interfaz de registro. La interfaz inicial cuenta con un total de 2 campos de texto y 4 botones.



Figura 6: Interfaz inicial.

El primer campo de texto es donde se debe colocar el correo con el cual se registró en el módulo y en el segundo se coloca la contraseña asociada al mencionado correo. En dado caso de no estar registrado deberá presionar el botón llamado “Registro” el cual lo llevará a la interfaz de registro. En cuanto a los otros tres botones, el botón “Salir” es una equis blanca dentro de un círculo rojo en la esquina superior derecha, en el que como su nombre lo indica, permite salir del módulo y volver al escritorio del computador; cabe mencionar que este mismo botón aparece en distintas interfaces dentro del módulo. El botón “Olvide mi contraseña” envía al correo electrónico en el primer campo de texto un enlace que lleva a un formulario a través de este se podrá restablecer la contraseña asociada a ese correo (si no pone el correo en el primer campo de texto, no podrá realizar esta función). Por último está el botón de “Acceso”, en donde se valida la información en los dos campos de texto, si la información es correcta se llevará a la interfaz de bienvenida o en su defecto que alguno de estos datos en los campos de texto sea incorrecto, visualizará en pantalla el fallo para poder solventarlo. Un ejemplo de un posible fallo sería “La contraseña es incorrecta”.

Interfaz de registro

La interfaz de registro se presenta como un formulario, que cuenta con un total de 4 campos de texto y dos botones. Esta interfaz se puede ver en la figura 7, el formulario solicita unos datos para poder hacer uso del módulo. Lo que datos solicitados en orden son:

- Usuario: Un nombre distintivo en el sistema (Tipo alfanumérico y con un mínimo de caracteres de 1), un ejemplo de usuario es: *Hugo21*.
- Correo: Una dirección de correo electrónico (Debe poder acceder a este correo en caso de que olvide la contraseña para poder restablecerla), un ejemplo de correo es: *uncorreo@mail.com*.
- Contraseña: Una contraseña (Tipo alfanumérico con un mínimo de caracteres de 6), un ejemplo de contraseña es: *Contrsñ4*.

- Confirmar contraseña: Repetir la contraseña anterior para confirmarla.

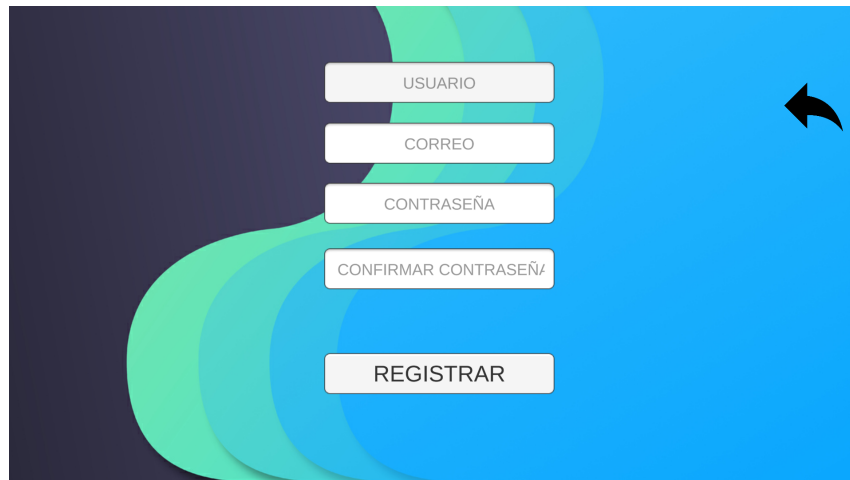


Figura 7: Interfaz de registro.

El botón “*Volver*” está en la parte superior derecha, como una flecha negra, donde se efectúa un cambio de interfaz regresando a la inicial. Por último el botón “*Registrar*” válida los datos que están en los 4 campos de texto, si la información es correcta volverá automáticamente a la interfaz inicial para poder iniciar sesión, en dado caso que algún dato sea incorrecto se visualizará en la pantalla el fallo para que pueda solventarlo. Un ejemplo de un posible fallo sería “*Las contraseñas no coinciden*”.

Interfaz de bienvenida

En la interfaz de bienvenida, si los datos de acceso ingresados en los campos de texto de la interfaz de inicio son correctos y por supuesto se cuenta con conexión a internet, la interfaz de bienvenida se mostrará de tal manera que se ve en la figura 8. Esta presenta un texto de “*Bienvenido (a)*” junto con el nombre de usuario del terapeuta que se registró en el módulo. Además de este texto hay dos botones, uno para cerrar la aplicación que funciona de la misma manera que el visto en la interfaz inicial y un botón de “*Ir a pacientes*”; este botón es el encargado de cargar los datos directamente inferiores en jerarquía del terapeuta de la base de datos los cuales son el número de pacientes que tiene a cargo junto con sus datos (el nombre de usuario del paciente, el nombre y el documento).

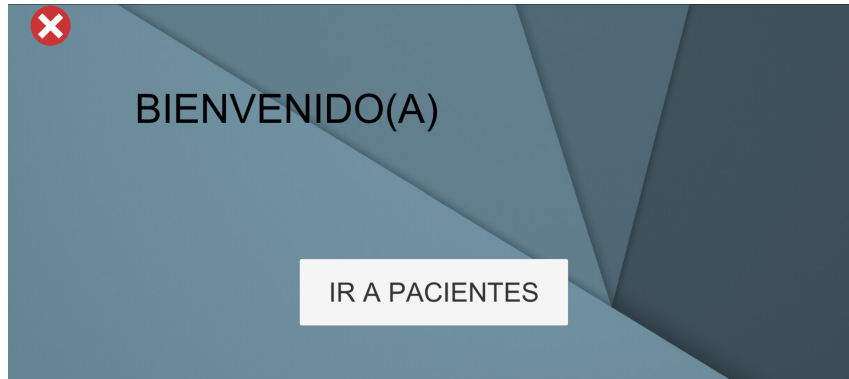


Figura 8: Interfaz de bienvenida.

Interfaz de pacientes

La interfaz de pacientes es la que permite todo lo referente a la administración o gestión del paciente, más específicamente permite el registro, modificación y eliminación del paciente en el sistema de MiniSuper AR. El registro es la función que añade al paciente a la base de datos del sistema y lo registra como uno de los pacientes del terapeuta que previamente accedió al módulo. La edición es la función que permite modificar los datos de un paciente, por último la función de eliminación la cual borrará al paciente del sistema junto con los datos que se hayan almacenado de este.

En el diseño de esta interfaz se priorizó que la realización de las funciones fuera dentro de la misma interfaz, dividiendo el espacio en uno de visualización y otro de gestión. En el lado izquierdo se representan todos los pacientes que el terapeuta tiene registrados en forma de botones individuales. Debajo de este espacio, un botón con un símbolo de más que lleva a la interfaz de registro de pacientes y al lado de este botón un contador de pacientes, en la figura 9 que presenta la interfaz y con fines explicativos ya se cuenta con un paciente registrado.



Figura 9: Interfaz de pacientes.

Presionar el botón con el nombre del paciente generará en el lado derecho de gestión la información asociada a este, su usuario, su nombre y su documento, acto seguido podrá elegir entre editar, eliminar o seleccionar para configurarle los ejercicios. Si el terapeuta desea editar debe presionar el botón “*Editar*”, este habilitará los campos de texto de nombre y documento para modificarlo (El usuario al ser un dato único no podrá realizarle cambios una vez establecido), luego podrá presionar el botón de “*Ok*” para actualizar la información en la base de datos y la interfaz, si al contrario lo va a eliminar solo presionará el botón “*Eliminar*” y se actualizará en la base de datos y la interfaz. Por último si lo va a seleccionar para avanzar y configurar el ejercicio de ese paciente tendrá que presionar el botón de “*Ir*” que lo llevará a la siguiente interfaz. Para concluir esta interfaz, el botón de “*Cerrar sesión*” lo llevará de nuevo a la interfaz de inicio.

Interfaz de registro de pacientes

La interfaz de registro de pacientes se visualiza en la figura 10, el cual presenta un formulario de tres campos de texto y 3 botones, los datos solicitados en los campos de texto son:

- **Usuario:** Un identificativo que funciona como llave de acceso del paciente para hacer los ejercicios en el juego serio (de tipo alfanumérico con un mínimo de caracteres de 4).
- **Nombre completo:** El nombre completo del paciente.
- **Tarjeta de identidad:** El documento de identidad del paciente.



Figura 10: Interfaz de registro de pacientes.

El botón de “*Verificar*” busca en la base de datos si no hay otro paciente con el mismo usuario, esta verificación es obligatoria, ya que una vez se verifique que el usuario está disponible se habilitará el botón de “*Guardar*”, donde se valida esta información y almacena en la base de datos el paciente nuevo; después de esta validación automáticamente vuelve a la interfaz de paciente. El botón “*Volver*” simplemente permite el regreso a la interfaz de paciente actualizando la información de nuevo.

Interfaz de ejercicio

La interfaz de ejercicio se puede visualizar en la figura 11, en donde se presenta una vez se haya presionado el botón con el nombre del paciente, seguido del botón de “ir” en lado derecho de gestión en la interfaz de pacientes. Después de eso en la parte superior de la interfaz de ejercicio se visualizan dos campos de texto no interactivos. El primer campo muestra el ejercicio que se está a punto de configurar y el segundo al paciente que se seleccionó anteriormente; si no sale ningún ejercicio debe seleccionarlo yendo a la interfaz de visualización de ejercicios con el botón de “Ver ejercicios” o creando el ejercicio con el botón de “Crear ejercicio” el cual lleva a la interfaz de crear ejercicio. Cuando ya se tenga en la parte superior de la pantalla el nombre del paciente y el nombre del ejercicio que se va a configurar, podrá presionar el botón de “Configurar”, el cual lleva a la interfaz de selección de configuración.

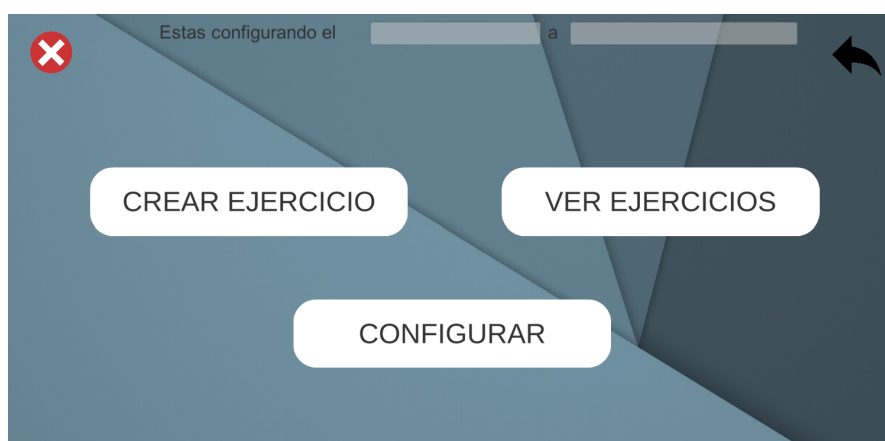


Figura 11: Interfaz de ejercicio.

Interfaz de creación de ejercicios

La interfaz de creación de ejercicio como muestra la figura 12 permite por medio de un formulario crear un ejercicio acotado a un tiempo establecido. Los datos que solicita este formulario son:

- Nombre ejercicio: El nombre identificativo que tendrá este ejercicio, servirá para darle indicaciones al paciente y para reconocerlo más fácilmente en el reporte.
- Fecha inicio: Una fecha de inicio deberá digitar de tal forma que cumpla la siguiente estructura DD/MM/AAAA (Día/Mes/Año).
- Fecha fin: Una fecha de fin deberá digitar de tal manera que cumpla la siguiente estructura DD/MM/AAAA (Día/Mes/Año).



Figura 12: Interfaz de creación de ejercicio.

Además de un botón de “*Crear*” que confirmará la información del formulario y establecerá ese ejercicio para la configuración poniéndolo en el campo de texto de la parte superior que no se puede interactuar y volviendo a la interfaz de ejercicio automáticamente.

Interfaz de visualización de ejercicios

La interfaz de visualización ejercicios tal como lo indica la figura 13, es la que permite cargar todos los ejercicios que previamente se encuentren hechos para el paciente seleccionado, con fines demostrativos en la figura 13 cuenta con un ejercicio creado y además seleccionado (ya que por defecto después de su creación aparece en el primer campo de texto no interactuable). En este caso el ejercicio es *Ejercicio1* y el paciente que lo tiene asignado es *Jose Torres*. En esta interfaz podrá seleccionar uno de los 7 ejercicios que se pueden tener al tiempo; este límite se estableció para efectuar cambios a los ejercicios que ya se hayan realizado en la totalidad del tiempo fijado. Esto permite ir construyendo cambios de dificultad, alterándolos y modificando su fecha para que sean vueltos a realizar o por otra parte, eliminarlo y crear uno nuevo en la interfaz de crear ejercicio.

Adicional a lo anterior, esta interfaz permite visualizar los ejercicios hechos para el paciente, también permite seleccionarlos para configurarlos, editarlos para modificar la fecha o eliminarlos por completo. Para acceder a cualquiera de estas tres funcionalidades se debe presionar la barra blanca que contiene el nombre del ejercicio y sus fechas, para así poder usar los botones de la derecha. El botón de “*Establecer*” asigna el ejercicio a la parte de arriba de esta manera cuando se presione el botón de “*Configurar*” de la interfaz de ejercicio lo llevará a la interfaz de selección de configuración, el botón de “*Editar*” permite asignar los cambios que se hagan a las fechas porque después de presionar la barra blanca se habilitan los campos de fecha inicio y fecha fin del ejercicio y por último, el botón de “*Eliminar*”, eliminará el ejercicio en su totalidad.

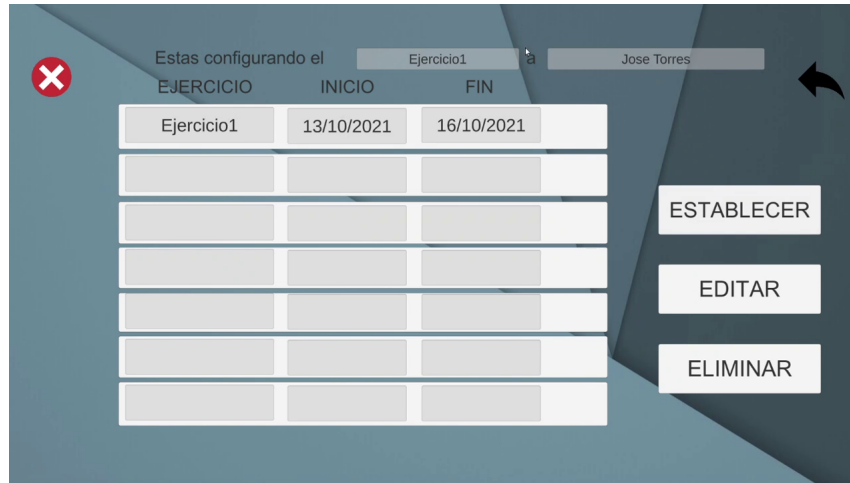


Figura 13: Interfaz de visualización de ejercicios.

Interfaz de selección de configuración

En la interfaz de selección de configuración, al tener los campos de texto no interactivos de la parte superior el ejercicio y el paciente que lo tiene asignado, se puede presionar en el botón de “Configurar” en la interfaz de ejercicio, esto hará que la interfaz de selección de configuración se muestre como se ve en la figura 14. Esta interfaz cuenta con dos botones que acceden a los dos aspectos que se pueden modificar del ejercicio, uno que es “Ir a estante” en donde se cambia a la interfaz de selección de lado de estante y se puede realizar las configuraciones del entorno en realidad aumentada, el otro “Configurar reto” el cual lleva a la interfaz de configuración de reto donde se establecen los objetivos del ejercicio.



Figura 14: Interfaz de selección de configuración.

Interfaz de configuración de reto

La interfaz de configuración de reto permite establecer el objetivo para el ejercicio, lo cual se hace evidente en la figura 15, donde se observa la interfaz. Al respecto, se puede mencionar un ejemplo de configuración del objetivo del *Ejercicio1*: para hacer la configuración del reto se presionará el objeto de todos los disponibles en el recuadro izquierdo, seguido de la cantidad en el recuadro derecho siendo un máximo de 12 por objeto, se puede confirmar viendo en los campos de texto no intratables el objeto y la cantidad acompañados de una oración que dice *Encontrar **objeto** con una cantidad de **número***; acto seguido deberá presionar el botón de “*Guardar*” para cargar esta información a la base de datos. Una vez hecho podrá ver si se realizó el guardado al entrar a la interfaz de visualización de reto con el botón de “*Ver retos*”; se puede colocar un máximo de objetivos de 15 y cada uno con un máximo de búsqueda de 12, pero esta cantidad sería superior a los espacios que ofrece el estante de objetos que es de 38 por lo que el reto debe configurarse teniendo en cuenta esta restricción o no podrá completarse satisfactoriamente.



Figura 15: Interfaz de configuración de reto.

Interfaz de visualización de reto

La interfaz de visualización de reto, permite divisar el objetivo o reto asociado a ese ejercicio (Los objetos junto a la cantidad a buscar). De igual manera, permite borrar todo el objetivo o reto presionando el botón “*Borrar*” para establecerlo de nuevo volviendo a la interfaz de configuración de reto. Un ejemplo de esto se observa en la figura 16 en la cual se indica el ejercicio 1 se deben buscar 2 bananas y 4 uvasVariantes.

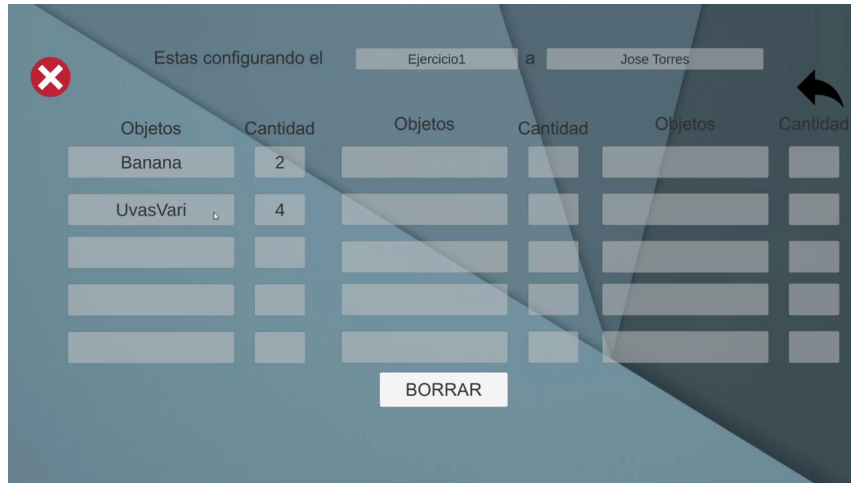


Figura 16: Interfaz visualización de reto.

Interfaz de selección de lado de estante

El presionar el botón “*Ir al estante*” en la interfaz de selección de configuración, llevará a esta interfaz como se ve en la figura 17, donde nos presenta los 4 lados configurables del estante. En cada uno de estos lados se puede seleccionar el objeto que tendrá cada espacio; el frontal y el posterior cuentan con el mismo número de espacios, siendo 12 para un total de 24 y los laterales que tienen menor número de espacios, siendo el rojo el que menos tiene con una cantidad de 6 y el amarillo con 8, el total de espacios es de 38. Los 4 funcionan de la misma manera, por lo que al presionar un lado llevará a la interfaz de configuración de estante adaptado para ese lado en específico.

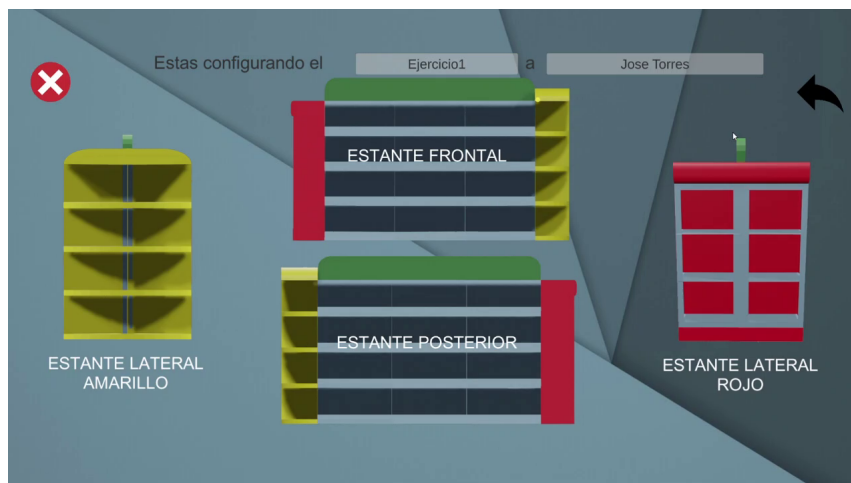


Figura 17: Interfaz selección de lado de estante.

Interfaz de configuración de estante

La interfaz de configuración de estante se presenta de la misma manera como se ve la figura 18, solamente que varía el lado del estante a configurar en la mitad de la interfaz dependiendo cuál se seleccionó en la interfaz de selección de lado. Cuenta con unos botones en cada espacio configurable del lado, en donde no hay objeto se visualiza una X verde, pero si ya cuenta con un objeto este mismo se mostrará en el espacio. Al presionar un espacio con o sin objeto se desplegará un cuadro a la izquierda con los objetos los cuales pueden cambiar a sus formas variantes que tienen un cambio de color presionando el botón “Variantes”. El lado derecho permite configurar donde aparecerá el personaje extra a buscar llamado “Monster” (El cual es de un color rosado) este “Monster” solo se puede ubicar en uno de los 4 sitios preestablecidos en el estante; además del “Monster” también se configura el tiempo de búsqueda para todo el ejercicio ingresando el número de segundos en el campo de texto. Por último, esta interfaz cuenta con 4 botones más, los cuales son: “Guardar”, el cual almacena toda la información de configuración del ejercicio del paciente en la base de datos, “Cerrar sesión” que permite volver a la pantalla inicial, “Botón mágico” para seleccionar varios espacios a la vez para una configuración más rápida y uno con la forma del estante que permitirá volver a la interfaz de selección de lado.

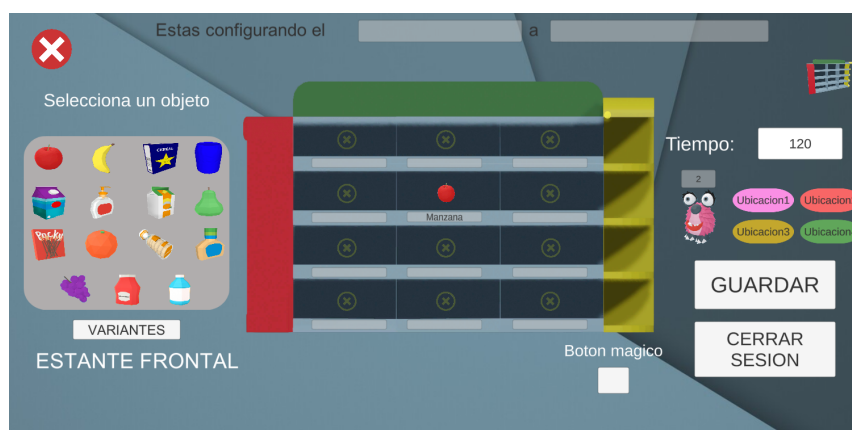


Figura 18: Interfaz configuración de estante.

3.2. Módulo de juego serio

Para el desarrollo del módulo del juego serio, se realizó una configuración de interfaz simple y llamativa, debido a que va dirigida específicamente a niños. Como ya se mencionó anteriormente, se optó por realizar un juego de búsqueda de objetos en un mini supermercado para incitar y motivar a los niños a encontrar los objetos.

Interfaz de inicio

Se utilizaron colores y diseños llamativos para la creación de la ambientación de las pantallas del juego, tal como lo indica la figura 19. Para el fondo de la interfaz de inicio se realizó un diseño de un supermercado, lo cual le agrega la ambientación necesaria al juego, junto con el logo personalizado del juego llamado, MiniSuper AR, asimismo el logo del semillero de realidad aumentada y realidad virtual en la industria 4.0. Igualmente se cuenta con dos botones para interactuar, el primero “*Jugar*” que tiene asociado la acción que lleva a la siguiente escena del juego y el segundo “*Salir*” que es para el cierre de la aplicación.



Figura 19: Interfaz de inicio juego serio

Inicio de sesión

Se visualiza en la figura 20, la interfaz de inicio de sesión, en donde se optó por dejar el mismo diseño de fondo que la interfaz inicial. Esta interfaz cuenta con un botón “*Volver*”, que tiene asociada la acción para volver a la escena anterior. Se le agregó un cuadro de texto con el mensaje para indicar que ingrese el nombre de usuario y otro cuadro de texto vacío, para ingresar dicho usuario requerido, este lo registra y genera el terapeuta. Asimismo se cuenta con otro botón, “*Login*”, el cual verificará si el usuario es correcto o incorrecto; de ser correcto se redirigirá a la siguiente escena, por el contrario, de no ser correcto se avisará que el usuario es incorrecto y deberá ingresar uno correcto.



Figura 20: Interfaz de inicio de sesión.

Perfil de Usuario

La interfaz del perfil de usuario se visualiza en la figura 21 en donde se presentan dos secciones: al lado izquierdo se muestran los datos de usuario como lo son el nombre y el documento, al lado derecho se aprecian los ejercicios asignados al usuario, estos se podrán realizar de acuerdo al intervalo de tiempo definido previamente cuando se generó el ejercicio. Además, también se asigna un botón “*Salir*” con la acción para volver a la pantalla de inicio. Para acceder a los ejercicios se debe presionar el botón correspondiente al nombre del ejercicio creado, por ejemplo en el caso de la figura 21, el botón tiene asignado el nombre del ejercicio Peras y solo se debe presionar para dirigirlo hacia la siguiente escena, que mostrará las condiciones del reto para ese ejercicio, tal como indica la figura 22.



Figura 21: Interfaz de perfil de usuario.

Pantalla de objetivos

Se puede visualizar en la figura 22, en la cual se presenta la escena que tiene la información del reto/misión asignada al ejercicio. Se agregaron dos botones, el primero “*Volver*”, para volver a la escena anterior, el segundo botón “*Siguiente*” para avanzar a la siguiente escena. Dependiendo del intervalo de fechas en las que se asignó la realización del ejercicio, se podrá presionar el botón siguiente; sin embargo, si el ejercicio no se encuentra en el intervalo indicado, se mostrará la escena de la figura 23, para evitar que se realice fuera de tiempo.

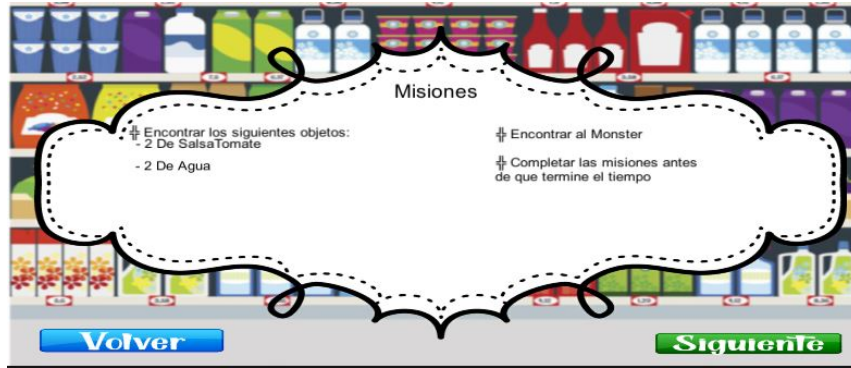


Figura 22: Interfaz de misiones en las fechas correctas.



Figura 23: Interfaz de misiones en las fechas incorrectas.

Escena de juego

En la escena de juego en realidad aumentada, se utilizó la herramienta de Vuforia para detección de planos que ya viene incorporada con el paquete de instalación. Empezando se eliminan la cámara que trae la escena por defecto y se agrega una cámara, como se observa en la figura 24. Para la detección de planos de Vuforia se agregan dos herramientas: *Ground plane stage* y *Plane Finder*, como se ve en la figura 25. El *Ground plane stage* actúa como una superficie virtual que asocia el plano detectado, y se puede posicionar objetos en ese espacio; *Plane Finder* es el buscador de planos en el entorno real.

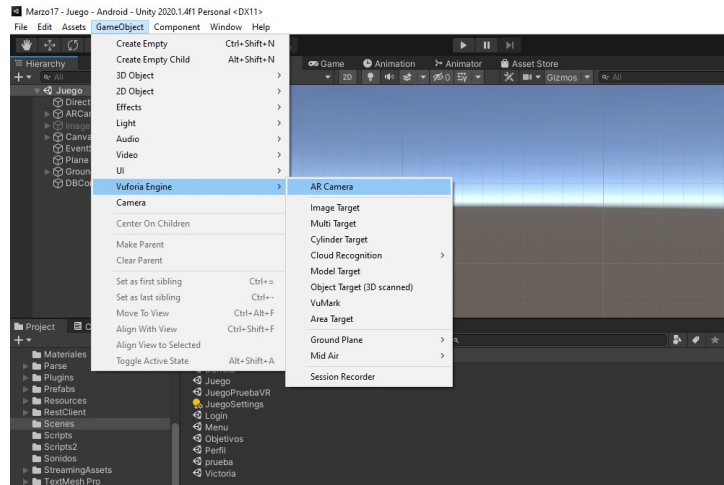


Figura 24: Agregar cámara en AR.

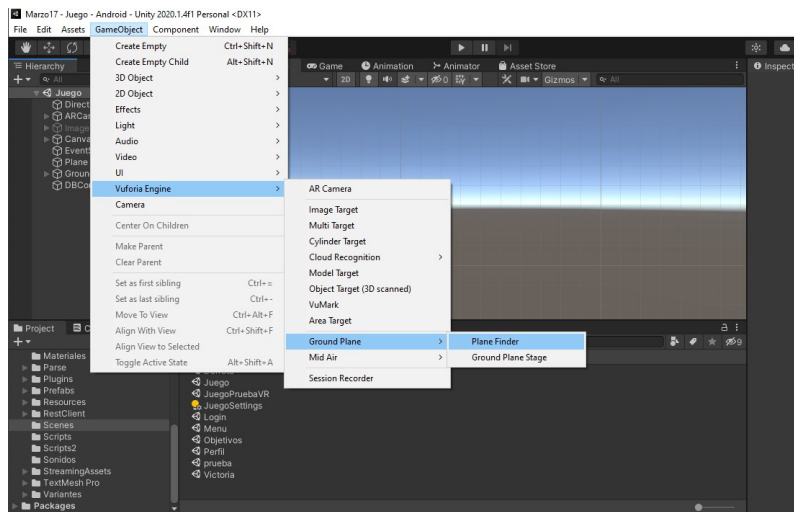


Figura 25: Agregar herramientas de para detección de planos.

En el inspector de *Plane Finder* se debe colocar en la parte de *Content Positioning Behaviour* y dentro de *Anchor Stage* arrastrar el *Grund Plane Stage*, esto se puede visualizar en la figura 26. Cuando el buscador de planos detecte un clic, posicionará la superficie de *Ground plane stage* en el plano detectado en el entorno real, por ende los objetos asociados a este se proyectaran en el entorno real, debido a eso, el objeto que se requiere proyectar en la realidad aumentada debe estar dentro de la jerarquía de *Ground Plane Stage*, como se observa en la figura 27.

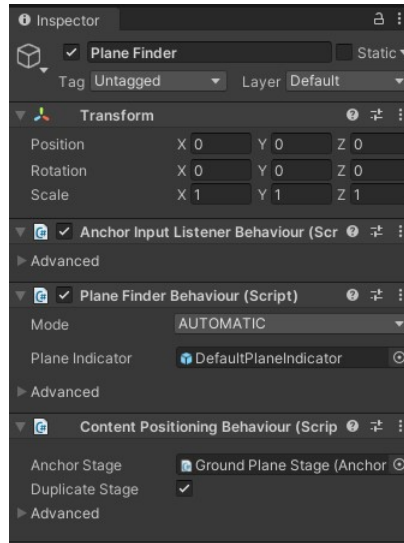


Figura 26: Inspector de Plane Finder.

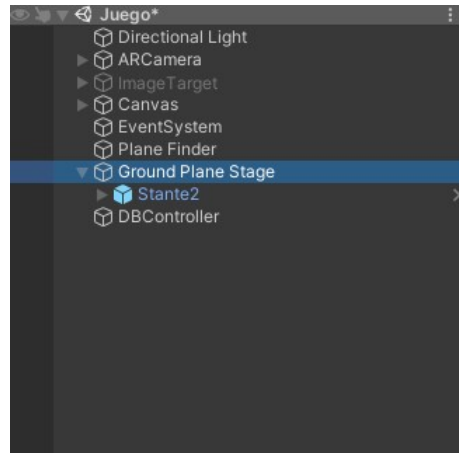


Figura 27: Jerarquía de Ground Plane Stage.

Con esto realizado, se puede visualizar en la figura 28, un indicador para saber donde proyectar el objeto, en este caso se proyectará el estante con los objetos correspondientes, tal como se observa en la figura 29.



Figura 28: Puntero para colocar objeto.

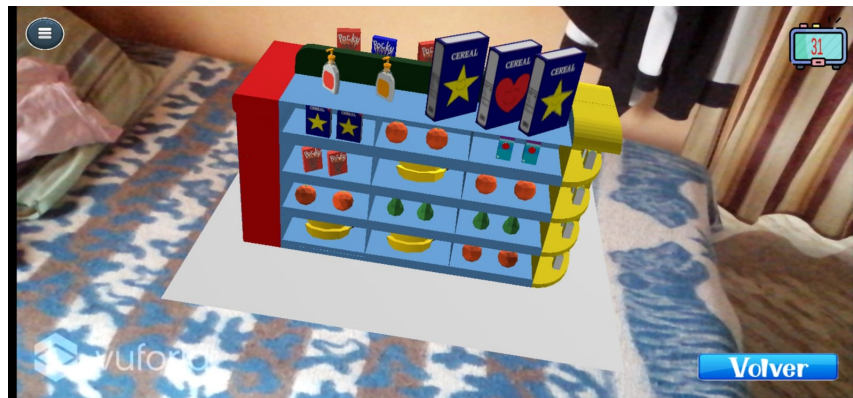


Figura 29: Estante con objetos.

Tal como se observa en la figura 30, todos los objetos se encuentran desactivados. Debido a que el *Monster* es un objeto especial para motivar al usuario a buscar en lugares más específicos, se le asignaron 4 posiciones predefinidas para su aparición, las cuales se encuentran desactivadas como se muestra en la figura 31.



Figura 30: Jerarquía de objetos desactivados.



Figura 31: Jerarquía del monster desactivado.

Interacción de objetos

Con respecto a la interacción de los objetos, se tiene que todos los objetos tiene la misma interacción, dependiendo del reto programado en el ejercicio, lo que indica que: si el objeto está en la lista de búsqueda, entonces al seleccionarlo desaparecerá y contara el progreso, de no estar en la lista de búsqueda, entonces simplemente se emitirá un sonido de fallo y no contará al progreso del juego. Para identificar cada objeto y poder relacionarlo con la interacción correcta, cada uno de ellos tiene un tag con su nombre, lo cual se puede ver la figura 32.

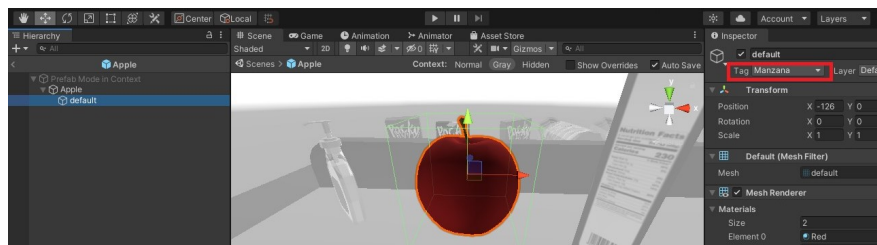


Figura 32: Asignación de tag.

Progreso en el juego

Tal como se observa en la figura 33, para el progreso se realizó un menú desplegable, en donde se muestran los objetos que se deben buscar, asimismo la cantidad que requiere buscar de cada uno de ellos, este menú se actualiza conforme vaya cumpliendo los aciertos del juego, lo que quiere decir que cada que se encuentren los objetos se va restando en el indicador, y así mostrar el progreso a medida que juega. Una vez se complete la búsqueda de un objeto, este se marcará completado, como lo indica la figura 34.

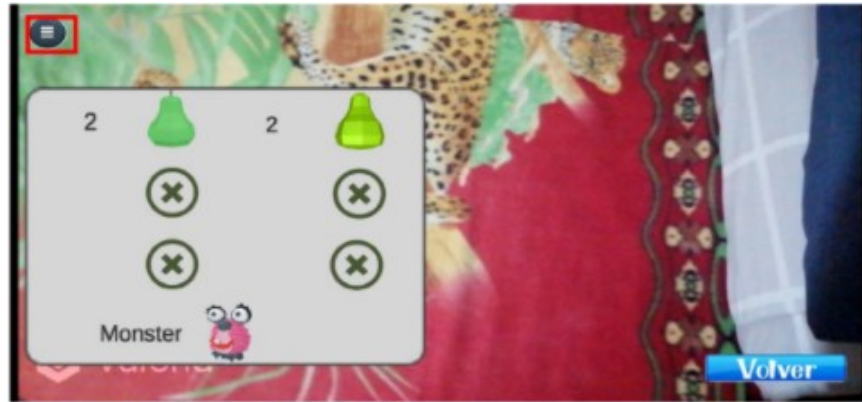


Figura 33: Progreso de las tareas.



Figura 34: Progreso con una tarea completada.

Tiempo

La visualización del tiempo requerido para la realización del ejercicio, utiliza una imagen de un reloj digital genérico, como se visualiza en la figura 35. En un cuadro de texto se imprime constantemente el cálculo del tiempo, cuyo valor se obtiene de la base de datos, tal como indica la figura 36. El tiempo se mostrará en la parte superior derecha de la pantalla, para que el jugador en cada momento pudiera tener en cuenta el tiempo restante.



Figura 35: Imagen de reloj utilizado.

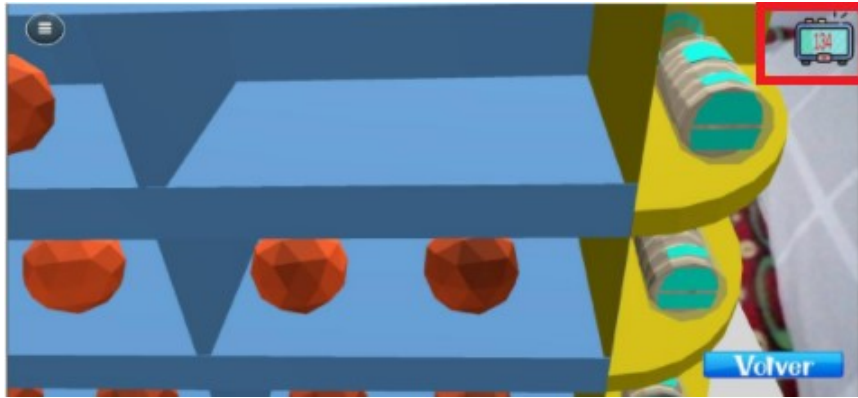


Figura 36: Reloj en pantalla de juego.

Condición de victoria y envío de datos

Una vez terminado el juego solo se puede llegar a dos escenas siguientes: la primera es una escena de victoria, tal como se observa en la figura 37 y la segunda una escena de derrota, tal como se visualiza en la figura 38. La escena que se muestre depende de la condición de victoria, que está definida por el reto asignado al ejercicio. Para que ocurra el cambio a la escena de victoria se deben cumplir todas las condiciones del reto, las cuales siempre salen en la escena de la figura 22, lo que indica que debe encontrar todos los objetos de cada tipo de objeto de la lista de búsqueda, debe encontrar el *Monster* y hacer todo antes de que se termine el tiempo; al cumplirse estas condiciones habrá ganado el juego y cambiará a la escena de victoria. Debido a que no se tiene una cantidad limitada de fallos, la única condición para perder el juego y cambiar a la escena de derrota, será que se termine el tiempo y no haya cumplido con el reto. Inmediatamente, cambia a la escena de victoria o derrota, se enviará la información a la base de datos y dicha se utilizará para el análisis en el módulo de reportes. Los datos que se envían son el tiempo utilizado, el número de fallas, y el número de aciertos.



Figura 37: Interfaz de victoria.



Figura 38: Interfaz de derrota.

3.3. Módulo de reportes

El módulo de reportes permite la generación de un documento en formato PDF en el que el terapeuta podrá ver los resultados del paciente, al efectuar los diferentes ejercicios que fueron configurados en el módulo de configuración y realizados en el juego serio. Entre la información presentada en el documento está la media y la varianza del tiempo utilizado para efectuar cada ejercicio.

El módulo de reportes cuenta solamente con dos interfaces. Estas al no presentar figuras grandes o formularios extensos no fueron diseñadas para ser en pantalla completa, por lo que el módulo es en modo ventana. Las dos interfaces fueron nombradas como interfaz inicial e interfaz principal; como previamente se dijo, este módulo se trabajó Python, el cual permite utilizar librerías matemáticas y librerías de conversión y generación de documentos en PDF. En cuanto al color usado para este módulo, se escogieron rojo para el fondo y azul para los botones, haciendo representación de los colores institucionales.

Interfaz inicial

Tal como se observa en la figura 39, la interfaz inicial cuenta con dos campos de texto y único botón llamado “Acceder”. Al estar el módulo en modo ventana no se requiere un botón de salir como puede ser en el caso del módulo de configuración por ende para salir y cerrar el módulo simplemente debe presionar la “X” en la ventana. En los campos de texto se deberá poner el correo y la contraseña del terapeuta, estos datos son los que se introducen al registrarse en el módulo de configuración, ya que es el mismo terapeuta el que está accediendo a este. Al igual que al acceder al módulo de configuración, si uno de estos datos es incorrecto saldrá un aviso indicando que la información introducida no es válida. En dado caso que la información sea correcta, la interfaz cambiará a la principal.



Figura 39: Interfaz acceso/login módulo de reportes.

Interfaz principal

Se visualiza en la figura 40, la interfaz principal, la cual cuenta con dos botones y un campo de texto, además de un escrito de bienvenida y una advertencia; ya que en esta interfaz simplemente se debe ingresar en el campo de texto el nombre de usuario del paciente del que se quiera generar el reporte, ese usuario es el identificativo único del paciente recordando que también es la llave con la que él puede acceder y realizar el ejercicio en el juego serio. La advertencia indica que el usuario se debe poner tal y como fue asignado respetando espacios, mayúsculas y minúsculas. Una vez puesto el nombre de usuario del paciente en el único campo de texto de esta interfaz se presiona el botón de *“Cargar datos”*, si este no cuenta con datos (No ha hecho algún ejercicio) saldrá un aviso de error, al contrario, si cuenta con datos (Ha hecho algún ejercicio) la interfaz cambiará a la que se ve en la figura 41, donde mostrará un aviso diciendo que los datos se han cargado y ocultando el botón de *“Cargar datos”* para dar paso al botón de *“Generar reporte”* el cual efectuará la acción de su nombre y generará un reporte en la carpeta donde se encuentre el ejecutable tipo EXE de este módulo. Si el terapeuta quiere generar otro reporte de otro paciente, presionará el botón de *“Limpiar”* y volverá a la interfaz de la figura 40.



Figura 40: Interfaz de extracción de datos paciente.



Figura 41: Interfaz de generación de reporte y reutilización.

Estructura reporte PDF

El reporte PDF tiene como estructura una página introductoria única en todo el reporte, seguida de una página descriptiva del ejercicio y una página evaluativa del ejercicio, la cantidad de estas dos últimas dependerá de la cantidad de ejercicios realizados por el paciente. En caso de tener 2 ejercicios, como ejemplo, se tendrá la siguiente estructura:

- 1- Página introductoria
- 2- Página descriptiva del primer ejercicio

- 3- Página evaluativa del primer ejercicio
- 4- Página descriptiva del segundo ejercicio
- 5- Página evaluativa del segundo ejercicio

A continuación se presenta la información pertinente para las páginas del reporte en PDF.

- **Página introductoria:** Esta página tiene la estructura presentada en la figura 42. El título contiene el nombre de usuario del paciente, seguido va la información básica del paciente (nombre completo, documento de identidad y el nombre de usuario del terapeuta), después de esto va la información del número de ejercicios configurados y el nombre de estos ejercicios. Por último, un identificador de dos columnas, la primera con los sitios del estante proyectado en AR con sus ubicaciones y la segunda columna los objetos disponibles para asociar a esas ubicaciones.
- **Página descriptiva del ejercicio:** Visible en la figura 43. Esta página descriptiva en la parte superior tiene el nombre del ejercicio, el tiempo configurado del ejercicio, la ubicación del *Monster*; seguido están todas las ubicaciones del estante con el objeto asignado a estas y para concluir en el pie de la página se encuentra el reto o el objetivo configurado del ejercicio (la búsqueda de el o los objetos).
- **Página evaluativa del ejercicio:** Presentada en la figura 44, muestra en la parte superior los rangos de fecha en la que el ejercicio está o estuvo habilitado, le siguen el número de intentos junto con el número de aciertos, fallas y el tiempo utilizado. Después 3 gráficos, el primer gráfico muestra el factor de aciertos (objetos acertados/objetos a buscar) por intento, el segundo muestra el número de fallos por intento y el último el tiempo empleado por intento. Por último, en la parte inferior izquierda la media del tiempo utilizado del ejercicio y la varianza del tiempo utilizado del mismo.

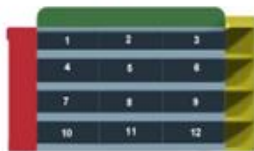
Reporte de _____

Nombre completo: _____
Documento: _____
Username terapeuta asignado: _____

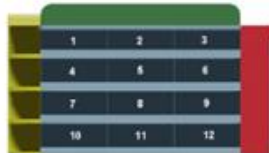
La cantidad de ejercicios configurados fueron: _____
Los nombres de estos ejercicios son: _____, _____, _____

Estante

Estante frontal (EF) posiciones:



Estante posterior (EP) posiciones:



Estante lateral rojo (ELR) posiciones:



Estante lateral amarillo (ELA) posiciones:



Objetos



Figura 42: Página introductoria.

Reporte de _____
 _____ configuración

El tiempo configurado fue de: ____
 La ubicación del monster fue la numero: ____

EF1 tiene un(a): ____ EF2 tiene un(a): ____ EF3 tiene un(a): ____ EF4 tiene un(a): ____ EF5 tiene un(a): ____ EF6 tiene un(a): ____ EF7 tiene un(a): ____ EF8 tiene un(a): ____ EF9 tiene un(a): ____ EF10 tiene un(a): ____ EF11 tiene un(a): ____ EF12 tiene un(a): ____	EP1 tiene un(a): ____ EP2 tiene un(a): ____ EP3 tiene un(a): ____ EP4 tiene un(a): ____ EP5 tiene un(a): ____ EP6 tiene un(a): ____ EP7 tiene un(a): ____ EP8 tiene un(a): ____ EP9 tiene un(a): ____ EP10 tiene un(a): ____ EP11 tiene un(a): ____ EP12 tiene un(a): ____	ELA1 tiene un(a): ____ ELA2 tiene un(a): ____ ELA3 tiene un(a): ____ ELA4 tiene un(a): ____ ELA5 tiene un(a): ____ ELA6 tiene un(a): ____ ELA7 tiene un(a): ____ ELA8 tiene un(a): ____
---	---	--

El numero de retos fue de: __ incluyendo el monster y fueron los siguientes objetos.

 El objeto fue monster con una cantidad no modificable de 1
 El objeto fue _____ con una cantidad de ____
 ...

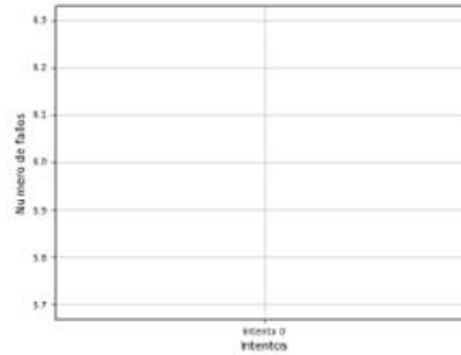
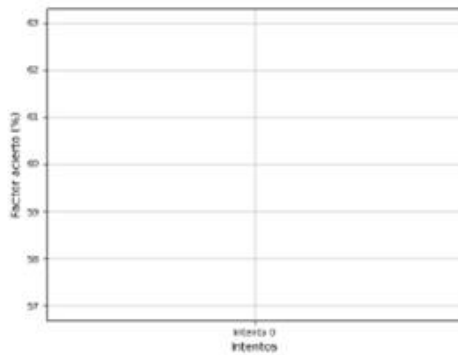
Figura 43: Página descriptiva.

Reporte de _____
_____ resultados

Este ejercicio tuvo como fecha de inicio: _____ y su fecha de finalización fue: _____

Realizo un numero de intentos de: ____

En el intento __: tuvo un total de __ acierto, __ fallas y un tiempo utilizado de ____
...



La media del tiempo utilizado es: ____
La varianza del tiempo utilizado es: ____

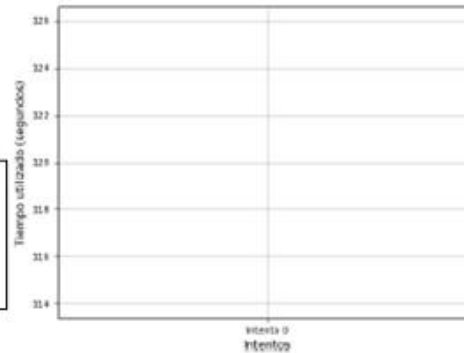


Figura 44: Página evaluativa.

3.4. Pilotaje del módulo de reportes

La prueba de funcionalidad tiene como fin verificar la precisión de los datos que se obtienen en la herramienta. Los datos a evaluar en cuestión fueron el número de fallos y aciertos por intento, además del tiempo utilizado para la realización del ejercicio. Para esta prueba se tomó una muestra de 3 sujetos mayores de edad, cada uno de ellos realizó 2 ejercicios distintos y con 5 intentos en cada ejercicio, los cuales fueron previamente configurados distintos para cada sujeto. En la figura 45 se observa uno de los sujetos realizando el ejercicio; la prueba se llevó a cabo por medio de una toma manual de datos, en el cual se usó un cronómetro de mano para la toma del tiempo utilizado. Igualmente se hicieron anotaciones por acierto y fallo en una libreta, teniendo en cuenta el sonido respectivo del juego serio al presionar un objeto correcto o uno incorrecto. Posteriormente, se hizo una comparación con los datos de la herramienta dando así los resultados de la tabla 6.

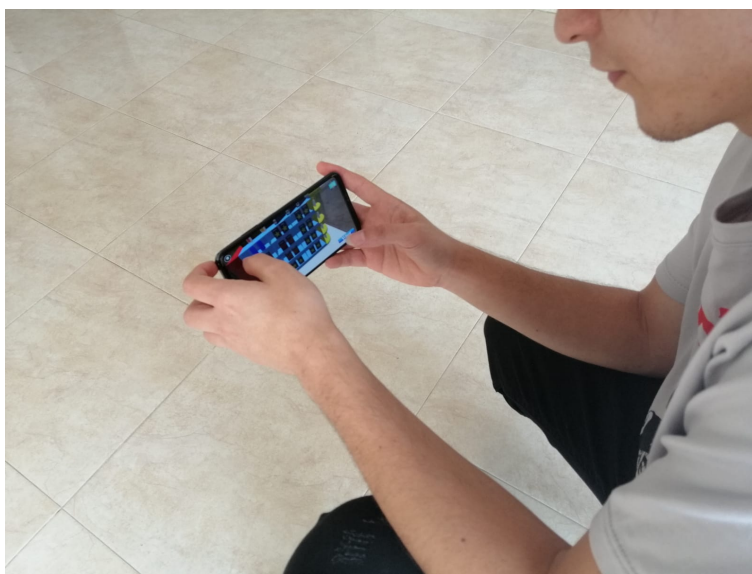


Figura 45: Prueba de realización de ejercicio.

Los 3 sujetos tuvieron victorias en cada uno de sus intentos, esto teniendo en cuenta que el tiempo configurado para todos fue de 2 minutos. La toma manual de aciertos y de derrotas al ser datos exactos fue precisa, pero la toma del tiempo presentó variaciones en algunos de los intentos con variaciones, que se identificaron como factor humano. Validando la información, la tabla 7 muestra una comparativa de los datos obtenidos manualmente y los datos que arroja el módulo de reportes de MiniSuper AR; los datos a comparar son la media y la varianza del tiempo utilizado por ejercicio de los 3 sujetos.

Sujeto	Ejercicio e intento	Aciertos		Fallos		Tiempo utilizado (seg)	
		Manual	MiniSuper AR	Manual	MiniSuper AR	Manual	MiniSuper AR
Sujeto 1	Ejercicio 1 intento 1	11	11	1	1	40	39
	Ejercicio 1 intento 2	11	11	1	1	25	25
	Ejercicio 1 intento 3	11	11	3	3	22	22
	Ejercicio 1 intento 4	11	11	0	0	16	16
	Ejercicio 1 intento 5	11	11	1	1	12	12
	Ejercicio 2 intento 1	12	12	2	2	19	18
	Ejercicio 2 intento 2	12	12	0	0	35	36
	Ejercicio 2 intento 3	12	12	2	2	26	25
	Ejercicio 2 intento 4	12	12	2	2	35	35
Sujeto 2	Ejercicio 2 intento 5	12	12	2	2	44	40
	Ejercicio 1 intento 1	16	16	2	2	93	94
	Ejercicio 1 intento 2	16	16	6	6	63	62
	Ejercicio 1 intento 3	16	16	1	1	25	23
	Ejercicio 1 intento 4	16	16	3	3	22	21
	Ejercicio 1 intento 5	16	16	31	31	81	81
	Ejercicio 2 intento 1	12	12	2	2	39	40
	Ejercicio 2 intento 2	12	12	3	3	45	45
	Ejercicio 2 intento 3	12	12	1	1	43	44
Sujeto 3	Ejercicio 2 intento 4	12	12	0	0	32	33
	Ejercicio 2 intento 5	12	12	1	1	33	34
	Ejercicio 1 intento 1	12	12	1	1	52	51
	Ejercicio 1 intento 2	12	12	0	0	45	45
	Ejercicio 1 intento 3	12	12	1	1	34	33
	Ejercicio 1 intento 4	12	12	2	2	35	34
	Ejercicio 1 intento 5	12	12	4	4	42	41
	Ejercicio 2 intento 1	13	13	4	4	61	60
	Ejercicio 2 intento 2	13	13	2	2	44	44
Sujeto 3	Ejercicio 2 intento 3	13	13	4	4	52	51
	Ejercicio 2 intento 4	13	13	1	1	45	44
	Ejercicio 2 intento 5	13	13	1	1	45	45

Tabla 6: Comparación de datos obtenidos

Sujeto	Ejercicio	Media		Varianza	
		Manual	MiniSuperAR	Manual	MiniSuperAR
Sujeto 1	Ejercicio 1	23 seg	22.8 seg	92.8 seg	86.16 seg
	Ejercicio 2	31.8 seg	30.8 seg	73.36 seg	65.36 seg
Sujeto 2	Ejercicio 1	56.8 seg	56.0 seg	831.36 seg	874 seg
	Ejercicio 2	38.4 seg	39.2 seg	27.04 seg	24.56 seg
Sujeto 3	Ejercicio 1	41.6 seg	40.8 seg	44.24 seg	45.76 seg
	Ejercicio 2	49.4 seg	48.8 seg	41.84	38.16 seg

Tabla 7: Comparación media y varianza de datos obtenidos

3.5. Protocolo experimental

3.5.1. Selección de participantes.

Se seleccionó un grupo de especialistas enfocados en áreas específicas en las que el proyecto se enfoca. Los especialistas fueron voluntarios para la realización de la encuesta, y se les proporcionó la información necesaria, como lo es:

- Manual de usuario para poder entender paso a paso el funcionamiento de los módulos.
- Vídeos tutoriales para guiarlos en las acciones a realizar.
- Carpeta en drive con todos los archivos de instalación y sus correspondientes tutoriales para ello.
- Encuesta de usabilidad realizada con la herramienta de formularios de Google.

A continuación, en la tabla 8. Se presentan los especialistas a los cuales se les envió un correo con la información mencionada anteriormente.

Participante	Profesión	Experiencia
Sujeto 1	Psicóloga, Universidad Cooperativa de Colombia	Especialista en neuropsicología infantil, Pontificia Universidad Javeriana, Sede Cali. Magister en desarrollo infantil, Universidad de Manizales. Experiencia 6 años en evaluación neuropsicológica infantil
Sujeto 2	Pisioterapeuta. Universidad de Pamplona	Especialista y Magister en Neurorehabilitación. Estudiante de doctorado en Ciencias Cognitivas con experiencia en el manejo de niños con parálisis cerebral, TEA y otros desordenes neuromotores y de conducta, con 7 años de experiencia en investigación, integrante del grupo Fisioter clasificado en categoría A
Sujeto 3	Fisioterapeuta Universidad del Valle	Especialista y Magister en Neurorehabilitación. Docente Universitaria de pre y postgrado Universidad del Valle y UAM. Gerente y terapeuta Centro de Neurorehabilitación Surgir Cali
Sujeto 4	Fisioterapeuta Fundación Universitaria María Cano	Especialista y Magister en neurorehabilitación. Docente del área de neurodesarrollo y neurorehabilitación 10 años
Sujeto 5	Terapeuta Ocupacional	Magister en intervención en la discapacidad y dependencia (Universidad da Coruña, España), certificada en Integración Sensorial (Universidad del Sur de California), Entrenamiento en Neurodesarrollo básico y pediátrico (NDTA). Investigadora integrante del grupo BIOH del programa de Terapia Ocupacional de la Universidad Mariana.
Sujeto 6	Fonoaudióloga	Especialista en gestión de la rehabilitación Candidata a Magister en Psicopedagogía

Continúa en la siguiente página

Participante	Profesión	Experiencia
Sujeto 7	Fonoaudióloga	Especialista en Educación y discapacidad
Sujeto 8	Terapeuta Ocupacional	Especialista en Educación y discapacidad. Diplomado en Neuropsicología Infantil y trastornos en el aprendizaje con énfasis en pedagogía
Sujeto 9	Terapeuta Ocupacional	Especialista en Educación y discapacidad Diplomado en Neurodesarrollo Diplomado en Neuropsicología Infantil y trastornos en el aprendizaje con énfasis en pedagogía.
Sujeto 10	Fonoaudiólogo de la universidad de Talca, Chile.	Doctorando en Fonoaudiología de la UMSA, Buenos Aires. Académico del departamento de ciencias de la Fonoaudiología, universidad de Talca. Magister en educación especial y master en intervención ABA para trastornos del desarrollo. Profesional que tiene experiencia en las atenciones clínicas a niños y niñas con trastornos severos de la comunicación en etapas iniciales y en apoyo psicoeducativo y familiar
Sujeto 11	Licenciada en Fonoaudiología, de la Universidad del Museo Social Argentino	Master en análisis conductual aplicado, Briding PROMPT trained. Especializada en lectura funcional mediante método CABAS y formada en DENVER y PECS.
Sujeto 12	Fonoaudióloga de la Universidad del Cauca	Magister en educación mediadas por las TIC Especialista en pedagogía de la lectura y la escritura Académica de la Universidad del Cauca
Sujeto 13	Fonoaudióloga de la Universidad del Cauca.	Magister en discapacidad. Académica de la Universidad del Cauca
Sujeto 14	Fonoaudióloga	Maestría en Educación: Desarrollo Humano (Valle del Cauca). Reconocimiento por la colaboración, apoyo y trabajo responsable como asesora de práctica del programa de Fonoaudiología

Tabla 8: Especialistas voluntarios

3.6. Parámetros de validación

Para la evaluación de la usabilidad por parte de los usuarios, uno de los objetivos de este trabajo, se usó un método de validación fundamentado en la familia de normas ISO/IEC 25000, también llamada SQuaRE (System and software Quality Requirements and Evaluation), el cual propone un marco de trabajo común para evaluar la calidad de un producto software. La familia de normas ISO/IEC 25000 está compuesta por cinco divisiones: División para la gestión de calidad, división para el modelo de calidad, división para la medición de la calidad, división para los requisitos de calidad y división para la evaluación de la calidad. La división para el modelo de calidad (ISO/IEC 2501n), específicamente en la ISO/IEC 25010(System and software quality models) la cual describe el modelo de calidad para el producto software y para la calidad en uso que tiene las varias categorías, como lo son: Funcionalidad, seguridad, rendimiento, mantenimiento, usabilidad, portabilidad, fiabilidad, compatibilidad. Teniendo en cuenta estas categorías, se basa la evaluación de este proyecto en esta norma.

La usabilidad es la característica a validar en este proyecto y se subdivide en las siguientes características: Capacidad para reconocer su adecuación, capacidad de aprendizaje, capacidad para ser usada, protección contra errores de usuario, estética de la interfaz de usuario y accesibilidad.

Se realizó una encuesta para el funcionamiento completo de los 3 módulos que componen el proyecto, dividiéndolo en 4 secciones:

- **Gestión de pacientes:** Evaluación de características referentes a la manipulación de pacientes dentro de la aplicación
- **Gestión de ejercicios:** Evaluación de características referentes a toda la parte de generación de ejercicios para asignar a cada paciente.
- **Gestión de reportes:** Evaluación de características referentes a la generación de reportes de desempeño de los pacientes.
- **Jugabilidad:** Evaluación de características generales del juego tanto estética como funcional.

Teniendo en cuenta la distribución hecha para la evaluación y las secciones, se crearon una serie de preguntas dependiendo de las características que se requieren valorar y la funcionalidad que tengan, estas preguntas se aprecian mejor en la tabla 9.

Funcionalidad	Pregunta	Descripción
Gestión de pacientes	¿Considera usted que el procedimiento de registro de pacientes en la plataforma es sencillo?	En esta pregunta se hace referencia al procedimiento, entendiéndose que es la acción de registrar al paciente en la aplicación, o sea llenar sus datos e ingresarlo a la aplicación.
	¿Considera usted que el procedimiento de eliminación de pacientes en la plataforma en caso de ser necesario es fácil?	En esta pregunta se refiere a la acción de borrar dado el caso un paciente de la plataforma y así, su registro se cancele.
	¿Considera usted que el procedimiento para el ingreso y modificación de los datos del usuario (nombre y documento de identidad) es simple?	Se considera como edición a la modificación y actualización de los datos registrados del paciente, o sea puede ser su nombre o su documento de identidad.
	¿Considera que la ejecución de la aplicación respecto al acceso, la visualización de la pantalla, alcance y manejo de botones y acceso a diferentes funciones es cómoda?	Con esto se quiere dar a entender que su visualización en pantalla sea clara, que no tenga que acceder a muchas pestañas o ventanas para ver la información que desea. O sea que todo esté al alcance, como sus datos y los botones que le permitan gestionar al paciente.
Gestión de ejercicios	¿Considera usted que el proceso de configuración para crear un ejercicio dentro de la aplicación y variar la complejidad de este es sencillo?	En esta pregunta, se desea saber si el proceso o la acción de crear el ejercicio y configurarlo le parece muy complejo o fácil.
	¿Considera que el nivel de personalización (tiempo, objetos y ubicación) que posee la aplicación para la configuración de las actividades para cada usuario es el acertado?	Aquí la pregunta va enfocada a si las modificaciones y/o configuración del ejercicio le da suficiente personalización a este (tiempo, objetos y ubicación).
	¿Considera que la variación de dificultad en cada ejercicio configurado respecto a la selección de objetos, ubicación y tiempo es acertada?	Aquí se desea saber si el especialista en el área confirma si dependiendo lo que se busque en el ejercicio y los demás objetos que se encuentran en el estante se varía la dificultad de manera acorde. Ejemplo. Buscar una manzana verde entre solo manzanas rojas con un tiempo de 10 segundos es más fácil que buscar una manzana roja entre diferentes objetos de diferentes formas en 10 segundos.

Continúa en la siguiente página

Funcionalidad	Pregunta	Descripción
Gestión de reportes	¿Considera usted el reporte de resultados de cada paciente generado por la aplicación es acertado?	Esta pregunta se refiere a si usar el módulo de reportes (entiéndase como la aplicación que genera el PDF con los resultados del paciente) le parece difícil o poco claro teniendo en cuenta que solo debe acceder con su correo y contraseña e ingresar el nombre de usuario del paciente seguido de un clic en el botón generar.
	¿Considera que el acceso (correo-contraseña- usuario) en la aplicación a los reportes de resultados de cada paciente es sencillo?	Esta pregunta considera que el inicio de sesión en el módulo de reportes se puede realizar de manera sencilla.
	¿Considera que la información generada en el reporte de resultados y la conducta a seguir para cada paciente es acertada?	En esta pregunta se refiere a si la información que se presenta en el reporte le parece suficiente para establecer el próximo ejercicio o sesión de acuerdo a los resultados que se ven.
	¿Considera que los gráficos reflejan de manera acertada el desempeño de los pacientes?: Factor de éxito, Factor de fallas y tiempo en los ejercicios	Aquí con gráficos nos referimos a las ilustraciones que muestran los resultados, si al usar número de errores, aciertos y tiempo sobrante para generar el desempeño es suficiente para determinar la efectividad del ejercicio configurado.
Jugabilidad	¿Considera que el procedimiento para acceder y realizar el ejercicio es sencillo?	Se pregunta como le parece el proceso de acceder al juego y realizarlo, si sería sencillo o complicado para el paciente.
	¿Considera que es fácil la interacción del paciente con la aplicación respecto a la realidad aumentada?	Si la interacción con los objetos en el entorno de realidad aumentada AR sería fácil o complicada para el paciente, si es claro cuando es un acierto o un error.
	¿Considera que sería fácil la recepción del usuario ante la realización de una actividad y la relación entre respuestas acierto/error?	Qué tan bien se percibe el juego y las actividades a realizar, por ejemplo que quede claro que debe buscar manzanas y que no debe seleccionar otro objeto diferente.
	¿Considera que la estética y el ambiente de juego son agradables?	Se quiere saber si para el paciente los modelos de objetos, los sonidos, los colores son factibles.

Tabla 9: Preguntas del cuestionario

3.7. Medición de valores para respuestas

Para la medición se utilizó como base la escala de Likert, que se utiliza para medir el nivel de aceptación de las personas, cuestionándolas sobre su nivel de acuerdo o desacuerdo sobre una declaración, por tanto, a diferencia de un simple “sí” o “no”, esta permite a los encuestados calificar sus respuestas. La forma de calificación designada para esta encuesta se observa en la tabla 10.

Respuesta	Puntaje
Muy de acuerdo	5
Algo de acuerdo	4
Ni de acuerdo ni en desacuerdo	3
Algo en desacuerdo	2
Muy en desacuerdo	1

Tabla 10: Valores en escala de likert

3.8. Análisis de resultados

De los 14 seleccionados para la encuesta de valoración, solo los sujetos 6, 9, 10, 14 realizaron la encuesta, por tanto a continuación, se presentan los resultados obtenidos mediante el cuestionario para evaluar la usabilidad de las herramientas software, teniendo en cuenta los 4 especialistas que completaron la evaluación, que se realizó de manera virtual mediante una encuesta desarrollada en la herramienta de formularios de Google.

Gestión de pacientes

Cantidad de respuestas: 4.

En la figura 46 se visualizan los resultados de las preguntas de la sección de gestión de pacientes, se observa una media de 4.7 en las respuestas, mostrando un nivel de aceptación del 94% por parte de los encuestados.

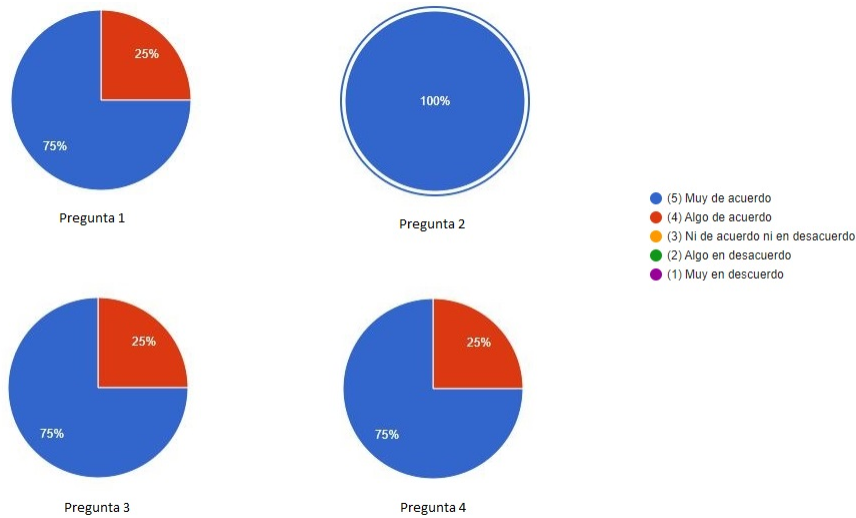


Figura 46: Gestión de pacientes

Pregunta abierta ¿Qué otras recomendaciones propone usted con respecto a la parte de Gestión de pacientes?

Cantidad de respuestas por pregunta: 4.

En la figura 47, se muestran las respuestas de los participantes en una nube de palabras, de la cual se realizó un análisis de las características similares y más llamativas. Con este análisis se obtuvieron los siguientes comentarios: Se sugiere para el módulo de gestión de pacientes más rapidez en la interfaz, igualmente que el usuario sea la identificación de los niños para más facilidad de los terapeutas; sin embargo, la idea del nombre de usuario es algo creativo y hace el juego más llamativo.



Figura 47: Nube de palabras gestión de pacientes.

Gestión de ejercicios

Cantidad de respuestas por pregunta: 4.

En la figura 48 se visualizan los resultados de las preguntas de la sección de gestión de ejercicios, donde se observa una media de 4.5 en las respuestas, mostrando un nivel de aceptación del 90 % por parte de los encuestados.

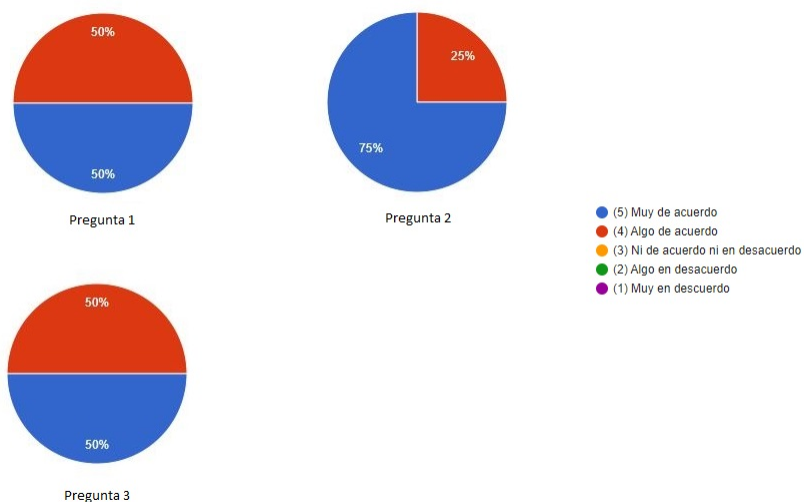


Figura 48: Gestión de ejercicios.

Pregunta abierta ¿Qué otras recomendaciones propone usted con respecto a la parte de Gestión de ejercicios?

Cantidad de respuestas: 4.

En la figura 49, se muestran las respuestas de los participantes en una nube de palabras, con las que se realizó un análisis de las características similares y más llamativas. Los comentarios recolectados con este análisis fueron los siguientes: Una vez más implican una mayor rapidez en la interfaz, además de centrarse en los objetos y que estos tengan una mejor visualización, tanto en calidad, colores y selección de imágenes. Respecto a los ejercicios, sugieren una mayor diversidad en las actividades que se desarrollan debido a que se debe tener en cuenta las edades de los niños; sin embargo, la aplicación se puede usar de manera sencilla para la parte de creación de ejercicios y la variación de colores es adecuada y ayuda a la atención de los niños, las opciones de selección de objetos y número de objetos a buscar brindan una buena oportunidad para los terapeutas de ajustar la dificultad.



Figura 49: Nube de palabras gestión de ejercicios.

Gestión de reportes

Cantidad de respuestas por pregunta: 4.

En la figura 50 se visualizan los resultados de las preguntas de la sección de gestión de reportes, donde se observa una media de 4.6 en las respuestas, mostrando un nivel de aceptación del 92 % por parte de los encuestados.

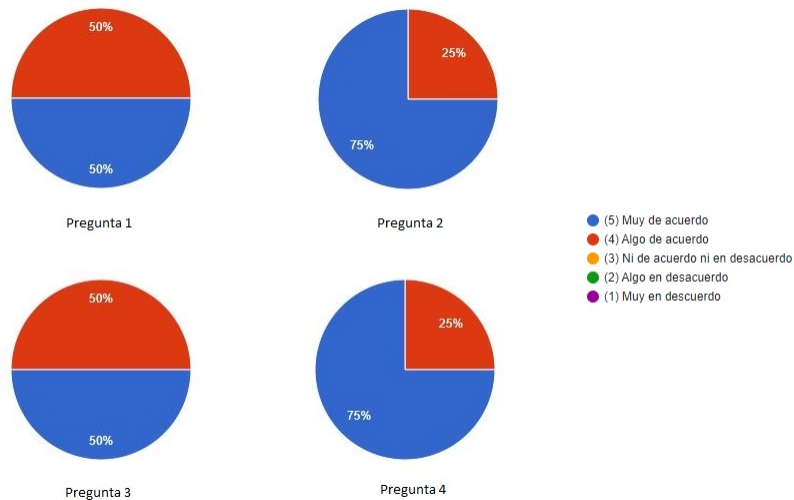


Figura 50: Gestión de reportes.

Pregunta abierta ¿Qué otras recomendaciones propone usted con respecto a la parte de Gestión de reportes?

Cantidad de respuestas: 3.

En la figura 51, se muestran las respuestas de los participantes en una nube de palabras, con las cuales se realizó un análisis de las características similares y más llamativas. Con dicho análisis se obtuvieron los siguientes comentarios por parte de los encuestados: Sugieren inicialmente tener en cuenta una puntuación dependiendo del desempeño en el juego, así como también el área específica que se quiere trabajar con el niño; sin embargo, que el reporte se genere automáticamente y que tenga gráficas hace que sea posible un mejor análisis de la actividad del usuario.

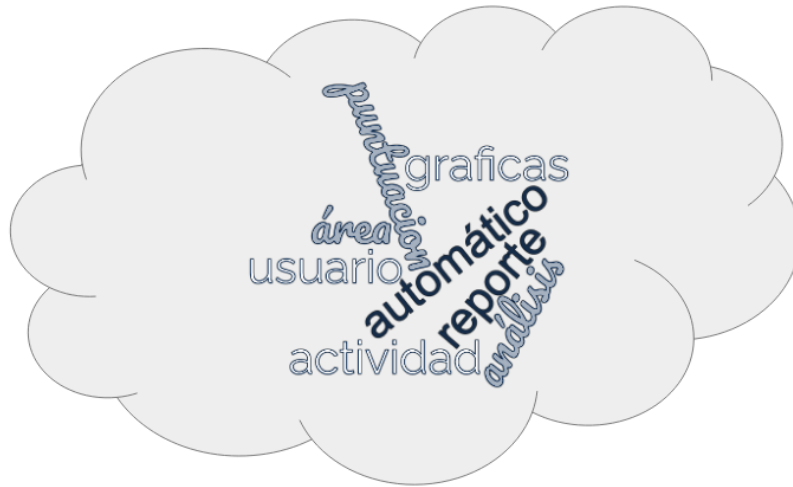


Figura 51: Nube de palabras gestión de reportes.

Jugabilidad

Cantidad de respuestas por pregunta: 4.

En la figura 52 se visualizan los resultados de las preguntas de la sección de jugabilidad, donde se observa una media de 4.3 en las respuestas, mostrando un nivel de aceptación del 86% por parte de los encuestados.

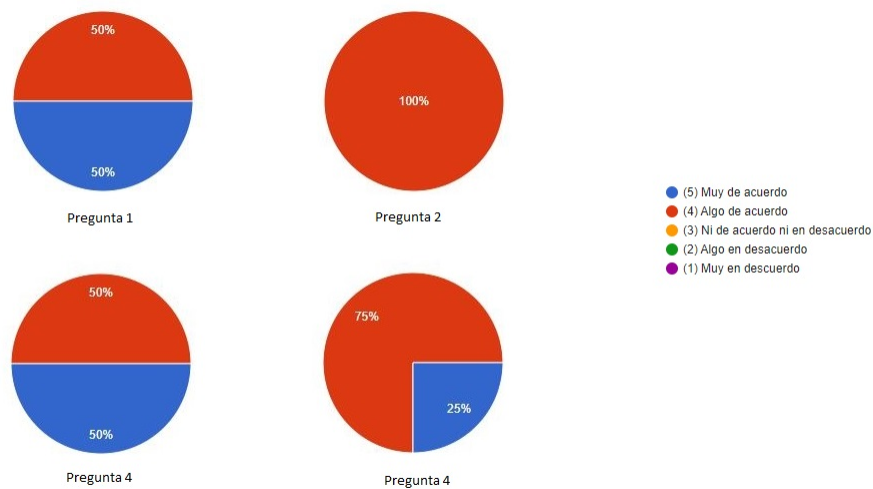


Figura 52: Jugabilidad.

Pregunta abierta ¿Qué otras recomendaciones propone usted con respecto a la parte de jugabilidad?

Cantidad de respuestas: 3.

En la figura 53, se muestran las respuestas de los participantes en una nube de palabras, con las cuales se realizó un análisis de las características similares y más llamativas. Dicho análisis arrojó los siguientes comentarios: Sugieren que el tamaño de las figuras tenga más parecido a la realidad, además, teniendo en cuenta que está dirigido a niños, es posible que no tengan aún la capacidad de leer y entender las instrucciones escritas, por tanto, recomiendan añadir comandos de voz para estas instrucciones y así el niño pueda entender mejor las actividades y tenga mejor relación con las instrucciones visuales que están en el juego.

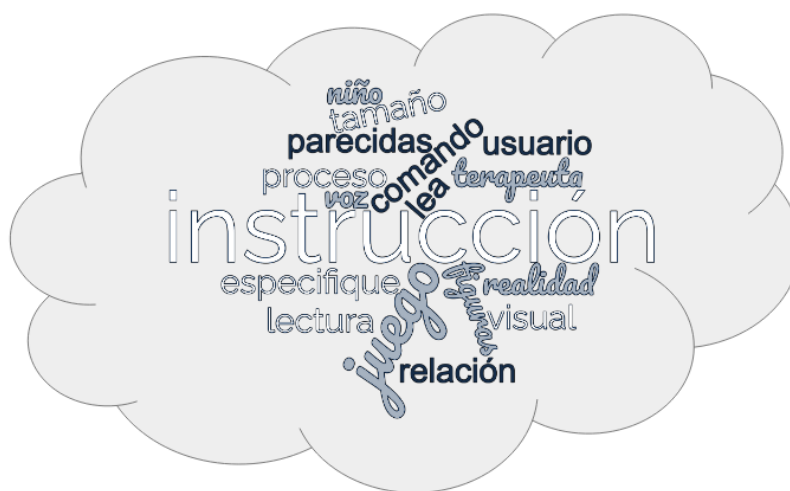


Figura 53: Nube de palabras de jugabilidad.

Recomendaciones

- Módulo de configuración:
 - Espacio en disco duro: 130 Mb
 - OS: Windows 7 SP1+, 8, 10, 64-bit
 - Tarjeta de video con capacidad para DX10 (shader modelo 4.0)
 - CPU: X64 arquitectura con soporte de conjunto de instrucciones SSE2
 - Memoria ram de 4GB o más

- Módulo de juego serio:
 - Versión de Android: 7.0 o más
 - Giroscopio
 - Acelerómetro
 - Magnetómetro
 - Espacio en el dispositivo: 110 Mb
 - memoria ram 2 GB

- Módulo de reportes:
 - Espacio en disco duro: 440 Mb
 - OS: Windows7 o posteriores
 - Procesador de 64 bits a 2 GHz o más
 - Memoria ram de 2GB o más

Capítulo 4

Discusión y conclusiones

4.1. Discusión

Sobre el módulo de gestión de pacientes, los expertos calificaron con una media de 4.7 la herramienta presentada; sin embargo, en la nube de palabras, se evidencia aspectos a tener en cuenta en el futuro, como por ejemplo: para tener una mejor experiencia de uso, se debe tener más rapidez en la interfaz, además que se use la identificación del paciente para tener un mayor orden por parte de los terapeutas, en el momento de interactuar con los pacientes que tienen registrados.

En la sección de gestión de ejercicios, calificaron positivamente con un valor de 4.5 de media, esto puede deberse a que cada interfaz correspondiente a la generación de los ejercicios es sencilla en su uso, además que las mecánicas del juego seleccionadas, son adecuadas para manipular la complejidad e intensidad del ejercicio; sin embargo, la reducción de puntos se aprecia en las recomendaciones de la nube de palabras en dicha sección, en donde al igual que la sección anterior se sugiere mayor rapidez en las interfaces, agregando también la idea de mejorar los objetos, tanto en tamaño, calidad, colores e imágenes ilustrativas tanto de los objetos, como de las acciones que se van realizando.

Teniendo en cuenta las recomendaciones de las terapeutas, el juego se creó con un ejercicio de búsqueda de objetos, el cual se asemeja al juego comercial “*¡Pictureka!*”, el que suelen usar para las terapias de niños con TDAH; sin embargo, los participantes sugieren agregar una mayor variedad de actividades y se asignen dependiendo de las edades de los niños.

Los usuarios calificaron la sección de gestión de reportes con una media de 4.6, esto puede deberse a que los resultados propuestos en el reporte son claros y adecuados, la información mostrada y las gráficas generadas son acordes y suficientes para mostrar el desempeño del paciente; sin embargo, en las recomendaciones mostradas en la nube de palabras de dicha sección, agregan, que si bien la información mostrada es correcta, también se podría agregar más información relevante como lo es una puntuación dependiendo del desempeño que tuvo dentro del juego, así como también mostrar información más detallada dependiendo de lo que se requiera trabajar específicamente con el paciente.

La sección de jugabilidad fue valorada con un valor de media de 4.3, la más baja entre las 4 secciones. Si bien el juego va dirigido al paciente, la encuesta fue realizada por los terapeutas para evaluar la usabilidad con base en el conocimiento y experiencia en los niños con TDAH. Entre las recomendaciones mostradas en la nube de palabras está: primero tener en cuenta las condiciones de los niños con TDAH, ya que al agregar instrucciones escritas se asume que la totalidad de los niños tiene la habilidad de lectura; sin embargo, está la posibilidad de que según las características específicas del trastorno, el niño no cuenta con la capacidad de entender las instrucciones escritas, por tanto sugieren que sería necesario agregar instrucciones auditivas para un mejor entendimiento; segundo, debido a que la realidad aumentada integra objetos virtuales en el mundo real, para tener una mejor inmersión, la similitud de los objetos del estante con los objetos reales debería ser más exacta, sin ser necesario tanto detalle a nivel visual, se pueden acoplar el tamaño de los objetos para una mayor similitud con los objetos reales.

En [61] presentan un ejemplo de cómo analizar las emociones de los pacientes mientras se utiliza una herramienta software, esta información permite tener en cuenta aspectos que no son evaluados en las encuestas; sin embargo, en este proyecto dado que esta fase ocurrió en perdido de pandemia, no pudo realizarse de manera presencial, razón por la cual no hubo retroalimentación directa sobre las emociones, lo cual no nos permitió realizar este tipo de análisis.

4.2. Conclusiones

Este trabajo de grado presentó el desarrollo de un juego serio que permite al terapeuta configurar en dificultad e intensidad los ejercicios para niños con TDAH, la sinergia al relacionar al grupo de desarrollo con los especialistas, permitió obtener un desarrollo con alto grado de aceptación, como evidencian las encuestas.

De este trabajo de grado se concluye que las características a tener en cuenta para el desarrollo de este tipo de aplicación son: primero, tener en cuenta el uso de objetos con alteraciones visuales de los mismos, como un cambio de color o etiqueta, pues de esta forma se puede trabajar la atención selectiva. Segundo, la configuración de las mecánicas del juego, como el tiempo, la ubicación de los objetos y la cantidad de los ejercicios, en el caso de este proyecto, son necesarios para diseñar un plan de trabajo personalizado. Tercero, usar interfaces intuitivas y que no estén saturadas de información para generar mayor facilidad de entendimiento tanto para el terapeuta como para el paciente. Cuarto, utilizar una arquitectura modular para separar el acceso a diferentes funcionalidades de la herramienta entre el paciente y el terapeuta es factible.

En cuanto a la realidad aumentada, se vio una alta aceptación por parte de los terapeutas; estas tecnologías emergentes se están presentando como una opción realmente viable y que

da resultados positivos no solo en el tratamiento o su uso específico, si no en términos de manejo y apropiación pues de manera eficaz induce al movimiento, lo cual en conjunto con tratamientos conductuales es ideal para hacer de los resultados algo más realistas; pues cuando se habla de TDAH la hiperactividad también se ve en el comportamiento físico.

El uso de Python para la evaluación de datos fue muy útil, pues presenta librerías especializadas para el manejo de estos, además de contar con librerías para su posterior visualización analítica, debido a que Python es ampliamente usado en ciencia de datos. Eso nos deja concluir que en términos de desarrollo para una herramienta evaluativa, separarla y utilizar Python es viable.

4.3. Trabajos futuros

Teniendo en cuenta los resultados obtenidos a continuación se enumeran los trabajos futuros:

- Aumentar la cantidad de objetos y la variedad de los mismos en el juego.
- Implementar un sistema de interacciones con rotaciones, desplazamientos, cambios de tamaños, entre otros.
- Implementar más señales visuales que vayan indicando el progreso y las acciones realizadas en la aplicación de generación de ejercicios.
- Implementar un algoritmo de ajuste automático de la complejidad e intensidad basado en inteligencia artificial.
- Incluir un sistema de comandos de voz para presentar las instrucciones al usuario en la aplicación móvil.

Bibliografía

- [1] J. Do Austerman, “Adhd and behavioral disorders: assessment, management, and an update from dsm-5,” *Cleveland Clinic journal of medicine*, vol. 82, p. S3, 2015.
- [2] S. M. Palma, A. C. M. Natale, and H. M. Calil, “A 4-year follow-up study of attention-deficit hyperactivity symptoms, comorbidities, and psychostimulant use in a brazilian sample of children and adolescents with attention-deficit/hyperactivity disorder,” *Frontiers in psychiatry*, vol. 6, p. 135, 2015.
- [3] “Inatención e hiperactividad (tdah).” [Online]. Available: <http://institutoneurologico.org/index.php/component/sppagebuilder/?view=page&id=27>
- [4] L. J. León Monguí, L. F. Reyes Villamil *et al.*, “Videojuego piano tiles estrategia para mejorar la atención sostenida en niños de 7 a 11 años diagnosticados con (tdah) de villavicencio.” 2021.
- [5] C. R. Sánchez and I. C. Muñoz, “Efectos del ejercicio físico terapéutico en jóvenes con trastorno por déficit de atención e hiperactividad: revisión sistemática,” *Fisioterapia*, 2021.
- [6] C. Colomer-Diago, A. Miranda-Casas, P. Herdoiza-Arroyo, and M. J. Presentación-Herrero, “Funciones ejecutivas y características estresantes de niños con trastorno por déficit de atención/hiperactividad: influencia en los resultados durante la adolescencia,” *Rev Neurol*, vol. 54, no. Supl 1, pp. 117–26, 2012.
- [7] A. H. Zúñiga, T. de Santos Moreno, F. J. Q. G. del Alamo, P. M. Ruiz-Lázaro, J. A. A. Díez, A. F. Jaén, and J. A. Ramos-Quiroga, “Consenso delphi sobre el trastorno por déficit de atención e hiperactividad (tdah): valoración por un panel de expertos,” *Actas españolas de psiquiatría*, vol. 44, no. 6, pp. 231–243, 2016.
- [8] J. Cornejo, O. Osío, Y. Sánchez, J. Carrizosa, G. Sánchez, H. Grisales, H. Castillo-Parra, and J. Holguín, “Prevalencia del trastorno por déficit de atención-hiperactividad en niños y adolescentes colombianos,” *Rev neurol*, vol. 40, no. 12, pp. 716–722, 2005.
- [9] L. J. L. Lizcanao, D. J. G. Ruiza, H. J. González, and P. P. R. Torresb, “Attention deficit hyperactivity disorder (adhd) in schoolchildren aged 6 to 17 years,” *Rev Pediatr*, vol. 21, pp. e101–e108, 2019.

- [10] S. Correa Franco *et al.*, “Alcances y nivel de efectividad en tratamientos para el trastorno por déficit de atención e hiperactividad (tdah): Revisión de literatura,” B.S. thesis, Universidad EAFIT, 2018.
- [11] L. C. Saiz Fernández, “Psicoestimulantes para el tdah: análisis integral para una medicina basada en la prudencia,” *Revista de la Asociación Española de Neuropsiquiatría*, vol. 38, no. 133, pp. 301–330, 2018.
- [12] D. Hernández Torres and A. Licon Oliver, “Intervención no farmacológica sobre el control inhibitorio en adolescentes con trastorno por déficit de atención/hiperactividad,” *Arch. neurocienc.*, vol. 26, no. 01, Mar. 2021.
- [13] A. A. Pablo and A. G. Hayes, “El papel de la musicoterapia y las terapias alternativas en el tratamiento del TDAH: un estudio exploratorio,” *Medicina naturista*, vol. 13, no. 1, pp. 15–20, 2019.
- [14] V. Solis and M. C. Quijano, “Rehabilitación neuropsicológica en un caso de tdah con predominio impulsivo,” *Revista chilena de neuropsicología*, vol. 9, no. 2, pp. 67–71, 2014.
- [15] M. M. Robayo Espinosa, “La música como coadyuvante en el tratamiento psicoterapéutico del tdah en niños, niñas y adolescentes,” 2017.
- [16] V. Benzing and M. Schmidt, “The effect of exergaming on executive functions in children with adhd: A randomized clinical trial,” *Scandinavian journal of medicine & science in sports*, 2019.
- [17] B. Valentin and S. Mirko, “Cognitively and physically demanding exergaming to improve executive functions of children with attention deficit hyperactivity disorder: a randomised clinical trial,” *BMC pediatrics*, vol. 17, no. 1, pp. 1–8, 2017.
- [18] C. B. G. Calleros, J. G. García, and Y. N. Rangel, “Uso de juegos serios como herramienta educativa para la enseñanza a niños con tdah,” *RD-ICUAP*, vol. 5, no. 13, 2019.
- [19] V. Inoue, “Abrimos la puerta a los juegos serios,” *Learning Review Latinoamérica*, vol. 2, pp. 12–17, 2008.
- [20] F. Juca Maldonado, M. B. García Saltos, and O. Burgo Bencomo, “Los juegos serios y su influencia en el uso responsable de energía y cuidado del medio ambiente,” *Revista Universidad y Sociedad*, vol. 9, no. 1, pp. 129–136, 2017.
- [21] P. Kokol, H. Vošner, J. Završnik, J. Vermeulen, S. Shohieb, and F. Peinemann, “Serious game-based intervention for children with developmental disabilities,” *Current pediatric reviews*, 2019.
- [22] E. Garnica and B. Quiroga, “Diseño de un sistema de juegos interactivos para niños y niñas con déficit atencional e hiperactividad,” *Experiencias en innovación educativa: Convirtiendo conocimiento en nuevas oportunidades*, p. 69, 2019.

- [23] E. G. Estrada, B. F. Q. López, P. M. Castro, and Á. P. M. León, “Diseño de módulos interactivos para tratar el trastorno por déficit de atención con hiperactividad-tdah,” *Revista Ingeniería, Matemáticas y Ciencias de la Información*, vol. 3, no. 6, 2016.
- [24] K. P. Thomas, A. P. Vinod, and C. Guan, “Design of an online eeg based neurofeedback game for enhancing attention and memory,” in *2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE, 2013, pp. 433–436.
- [25] “Estudio sobre el sistema de telemedicina de la terapia de biorretroalimentación tdah [j],” *Revista de la Universidad de Ciencia y Tecnología Electrónica de China*, 2011.
- [26] V. Wattanasoontorn, I. Boada, R. García, and M. Sbert, “Serious games for health,” *Entertainment Computing*, vol. 4, no. 4, pp. 231–247, 2013.
- [27] J. Huizinga, “Homo ludens: A study of the play-element in culture (translated from the 1944 german ed.),” 1955.
- [28] C. Z. Lara, J. M. G. Calleros, J. G. García, Y. N. Rangel, and C. A. C. Ordóñez, “Guías para el desarrollo de espacios virtuales para entrenamiento cognitivo.”
- [29] K. C. Bul, I. H. Franken, S. Van der Oord, P. M. Kato, M. Danckaerts, L. J. Vreeke, A. Willems, H. J. Van Oers, R. Van den Heuvel, R. Van Slagmaat *et al.*, “Development and user satisfaction of “plan-it commander,” a serious game for children with adhd,” *Games for health journal*, vol. 4, no. 6, pp. 502–512, 2015.
- [30] V. Colombo, D. Baldassini, S. Mottura, M. Sacco, M. Crepaldi, and A. Antonietti, “Antonyms: A serious game for enhancing inhibition mechanisms in children with attention deficit/hyperactivity disorder (adhd),” in *2017 International Conference on Virtual Rehabilitation (ICVR)*. IEEE, 2017, pp. 1–2.
- [31] P. J. Prins, E. T. Brink, S. DAVIS, A. Ponsioen, H. M. Geurts, M. De Vries, and S. Van Der Oord, ““braingame brian”: toward an executive function training program with game elements for children with adhd and cognitive control problems,” *GAMES FOR HEALTH: Research, Development, and Clinical Applications*, vol. 2, no. 1, pp. 44–49, 2013.
- [32] M. C. Barba, A. Covino, V. De Luca, L. T. De Paolis, G. D’Errico, P. Di Bitonto, S. Di Gestore, S. Magliaro, F. Nunnari, G. I. Paladini *et al.*, “Bravo: a gaming environment for the treatment of adhd,” in *International Conference on Augmented Reality, Virtual Reality and Computer Graphics*. Springer, 2019, pp. 394–407.
- [33] D. Avila-Pesantez, L. A. Rivera, L. Vaca-Cardenas, S. Aguayo, and L. Zuñiga, “Towards the improvement of adhd children through augmented reality serious games: Preliminary results,” in *2018 IEEE Global Engineering Education Conference (EDU-CON)*. IEEE, 2018, pp. 843–848.

- [34] P. J. Prins, S. Dovis, A. Ponsioen, E. Ten Brink, and S. Van Der Oord, “Does computerized working memory training with game elements enhance motivation and training efficacy in children with adhd?” *Cyberpsychology, behavior, and social networking*, vol. 14, no. 3, pp. 115–122, 2011.
- [35] L. A. Rivera, “Augmented reality serious games design to improve attention of children with adhd,” *IEEE 11CCC*, pp. 297–303, 2016.
- [36] A. Streicher and J. D. Smeddinck, “Personalized and adaptive serious games,” in *Entertainment Computing and Serious Games*. Springer, 2016, pp. 332–377.
- [37] K. Laver, S. George, S. Thomas, J. Deutsch, and M. Crotty, “Virtual reality for stroke rehabilitation: an abridged version of a cochrane review.” 2015.
- [38] R. Proffitt and B. Lange, “Considerations in the efficacy and effectiveness of virtual reality interventions for stroke rehabilitation: moving the field forward,” *Physical therapy*, vol. 95, no. 3, pp. 441–448, 2015.
- [39] D. F. Oruna Cabrera, “Neuroeducación y realidad aumentada con arkit para incrementar el nivel de atención en niños de 7 a 11 años con tda en la asociación peruana de déficit de atención de la ciudad de lima en el semestre 2019-10,” 2021.
- [40] J. D. Orozco Mogollón, “Software de realidad aumentada para niños con trastornos de déficit de atención e hiperactividad,” 2021.
- [41] C. Tosto, T. Hasegawa, E. Mangina, A. Chifari, R. Treacy, G. Merlo, and G. Chiazzese, “Exploring the effect of an augmented reality literacy programme for reading and spelling difficulties for children diagnosed with adhd,” *Virtual Reality*, vol. 25, no. 3, pp. 879–894, 2021.
- [42] S. Alqithami, M. Alzahrani, A. Alzahrani, and A. Mostafa, “Modeling an augmented reality game environment to enhance behavior of adhd patients,” in *International conference on brain informatics*. Springer, 2019, pp. 179–188.
- [43] “Exergaming y rehabilitación: una metodología para el diseño de exergames terapéuticos eficaces y seguros,” *Entertainment Computing*.
- [44] C. M. Dean and R. B. Shepherd, “Task-related training improves performance of seated reaching tasks after stroke: a randomized controlled trial,” *Stroke*, vol. 28, no. 4, pp. 722–728, 1997.
- [45] G. Lupinacci, G. Gatti, C. Melegari, and S. Fontana, “Interactive design of patient-oriented video-games for rehabilitation: concept and application,” *Disability and Rehabilitation: Assistive Technology*, vol. 13, no. 3, pp. 234–244, 2018.
- [46] M. J. Cano-Mañas, S. Collado-Vázquez, and R. Cano-de-la Cuerda, “Videojuegos comerciales en la rehabilitación de pacientes con ictus subagudo: estudio piloto,” *Revista de Neurología*, vol. 65, no. 8, pp. 337–347, 2017.

- [47] L. M. Londoño and M. D. Rojas, “Determinación de criterios generales para el diseño de juegos serios: modelo metodológico integrador,” *Información tecnológica*, vol. 32, no. 1, pp. 123–132, 2021.
- [48] L. Gámez-Calvo, J. M. Gamonales, V. Hernández-Beltrán, and J. Muñoz-Jiménez, “Beneficios de la hipoterapia para personas con trastorno por déficit de atención e hiperactividad en edad escolar. revisión sistemática exploratoria (benefits of hypotherapy for people with attention deficit and hyperactivity disorder in school age. expl,” *Retos digit.*, vol. 43, pp. 88–97, Jun. 2021.
- [49] Y. G. Ltd. (1999). [Online]. Available: <https://www.yoyogames.com/es/gamemaker>
- [50] U. Technologies. (2005). [Online]. Available: <https://unity.com/es>
- [51] A. M. Juan Linietsky and contributors. (2014). [Online]. Available: <https://godotengine.org>
- [52] I. GameSalad. (2009). [Online]. Available: <https://gamesalad.com>
- [53] L. Stencyl. (2011). [Online]. Available: <https://www.stencyl.com>
- [54] C. GmbH. (2002). [Online]. Available: <https://www.cryengine.com>
- [55] I. Amazon Web Services. (2006). [Online]. Available: <https://aws.amazon.com/es/lumberyard/details/>
- [56] C. Technologies. (2008). [Online]. Available: <https://www.cocos.com/en/>
- [57] M. Vidal Ledo, B. Lío Alonso, A. Santiago Garrido, A. Muñoz Hernández, I. d. R. Morales Suárez, and A. M. Toledo Fernández, “Realidad aumentada,” *Educación Médica Superior*, vol. 31, no. 2, pp. 0–0, 2017.
- [58] P. Inc. (2015). [Online]. Available: <https://developer.vuforia.com>
- [59] G. LLC. (2018). [Online]. Available: <https://developers.google.com/ar>
- [60] A. Inc. (2017). [Online]. Available: <https://developer.apple.com/augmented-reality/arkit/>
- [61] A. Bangor, P. T. Kortum, and J. T. Miller, “An empirical evaluation of the system usability scale,” *Intl. Journal of Human–Computer Interaction*, vol. 24, no. 6, pp. 574–594, 2008.

ANEXO A

CREAR LOGIN Y REGISTRO A LA APLICACIÓN USANDO UNITY Y FIREBASE

Para esta guía se requiere que se haya instalado con anticipación el motor de videojuegos unity en cualquier versión, es recomendable que esta versión se instale junto con los SDK de Android, SDK iOS y Windows Build usando UnityHub que permite una gestión más limpia de la descarga de este motor de videojuegos junto con los SDK ya mencionados. A continuación, se anexan las imágenes de estos módulos junto con la versión utilizada para esta guía.

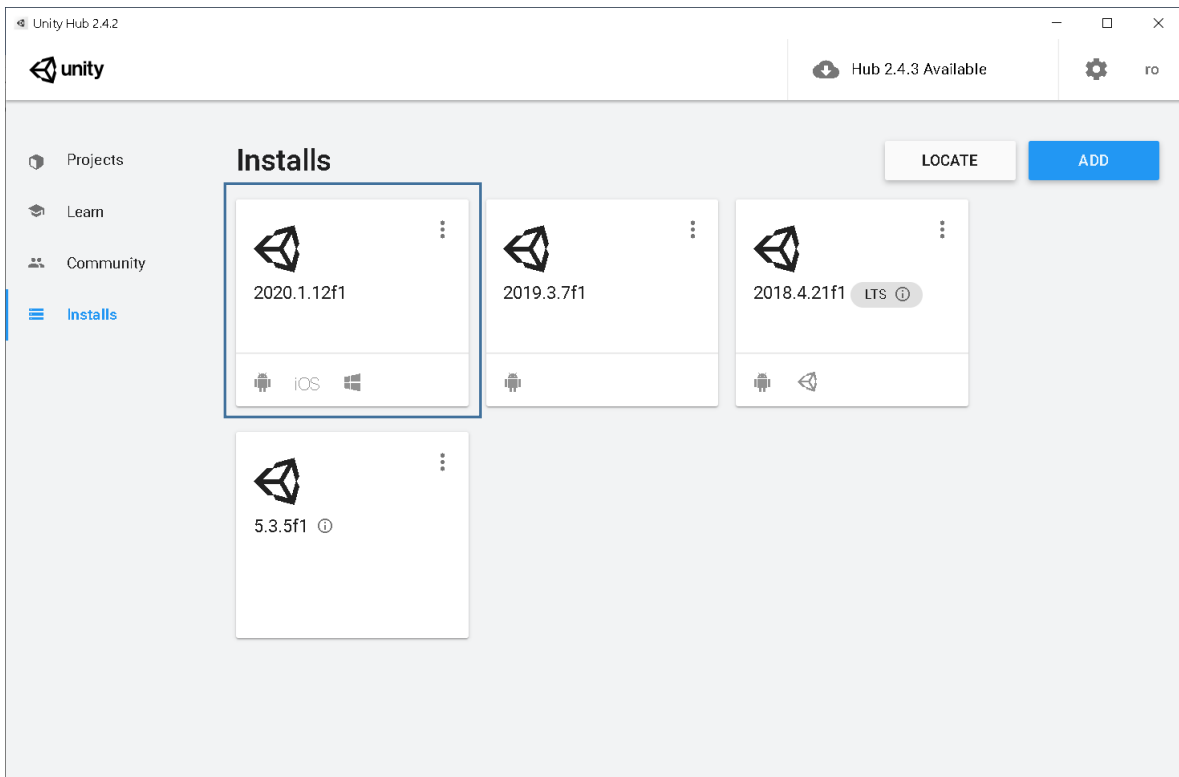


Fig. 1. UnityHub con la instalación de la versión y módulos

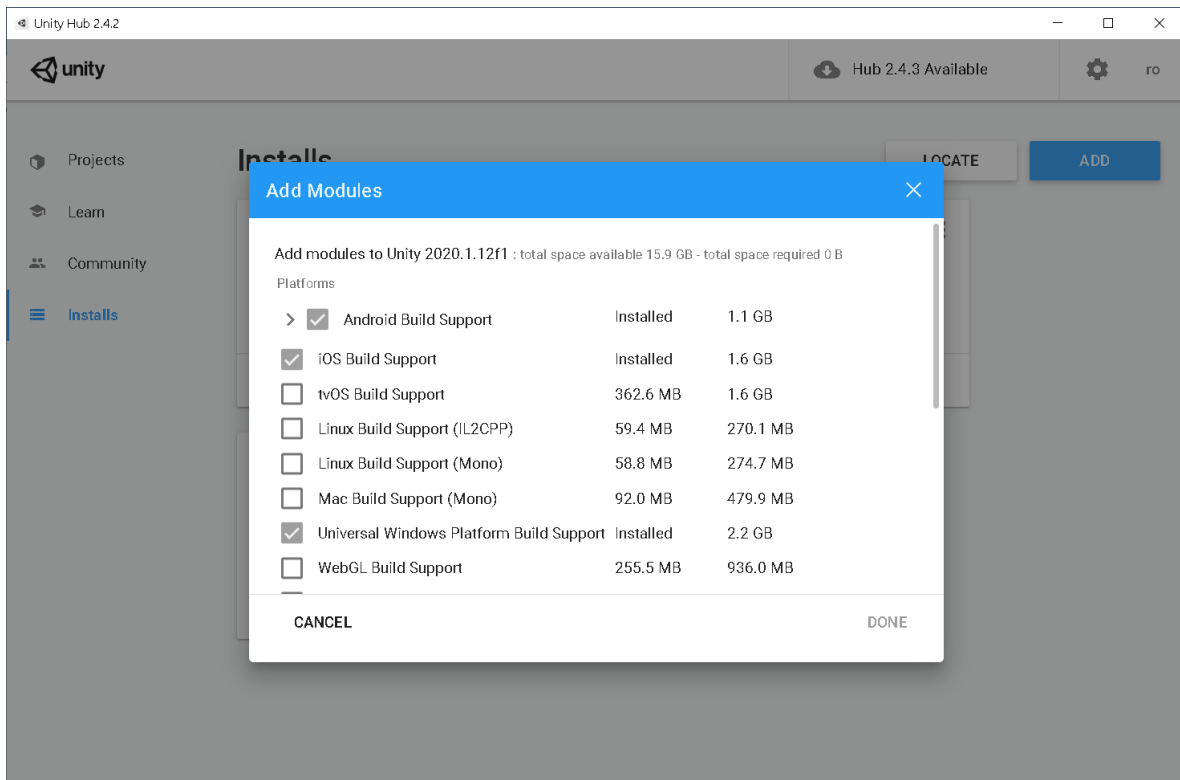


Fig. 2. Módulos específicos instalados para esta versión (Android, iOS, Windows)

Una vez ya tengamos instalados el motor de videojuegos, unity, procederemos a crear un proyecto en firebase que es nuestra base de datos donde podremos almacenar las credenciales que nos permitirán logearnos al aplicativo o registrarnos en su defecto. Usando el navegador de preferencia nos dirigimos al enlace “firebase.com” donde nos encontraremos con una interfaz donde deberemos ir al botón “acceder” en la esquina superior derecha; para logearnos, recomendablemente con una cuenta Gmail (Si no se posee una, crearla para poder logearse).

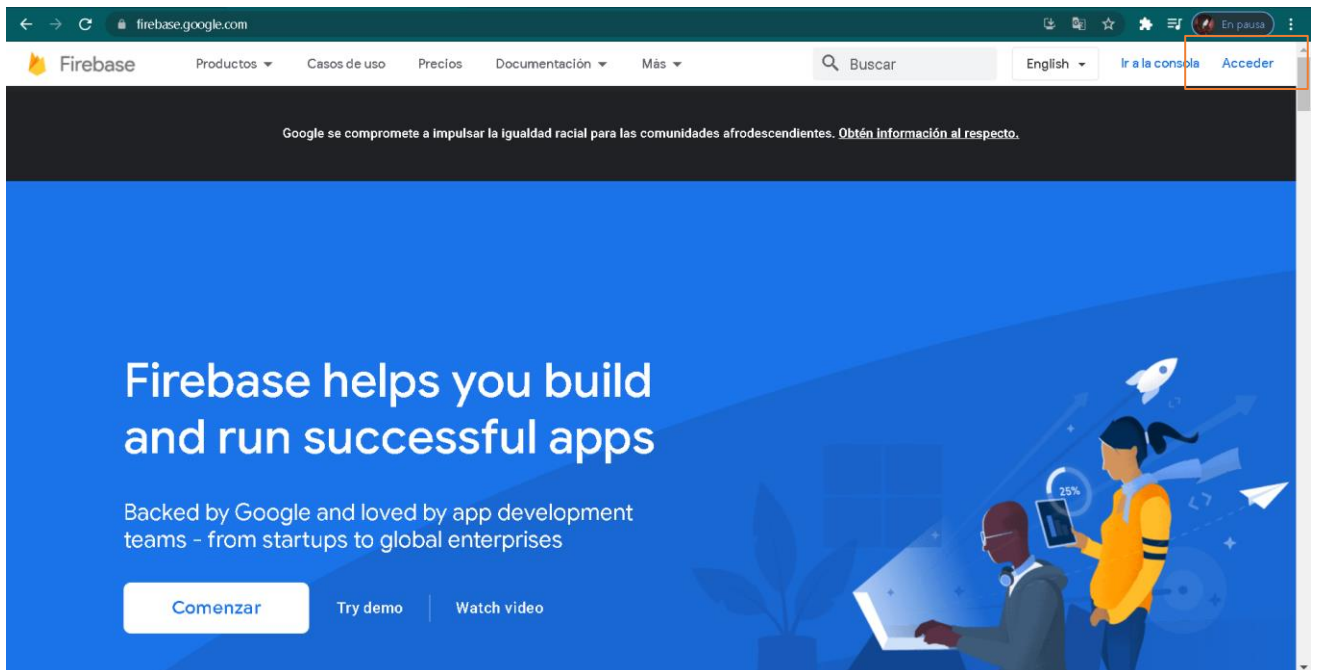


Fig. 3. Interfaz Firebase

Al logearnos a Firebase podremos crear nuestro proyecto dando al lado de donde estaba el botón acceder, que se llama “Ir a la consola”

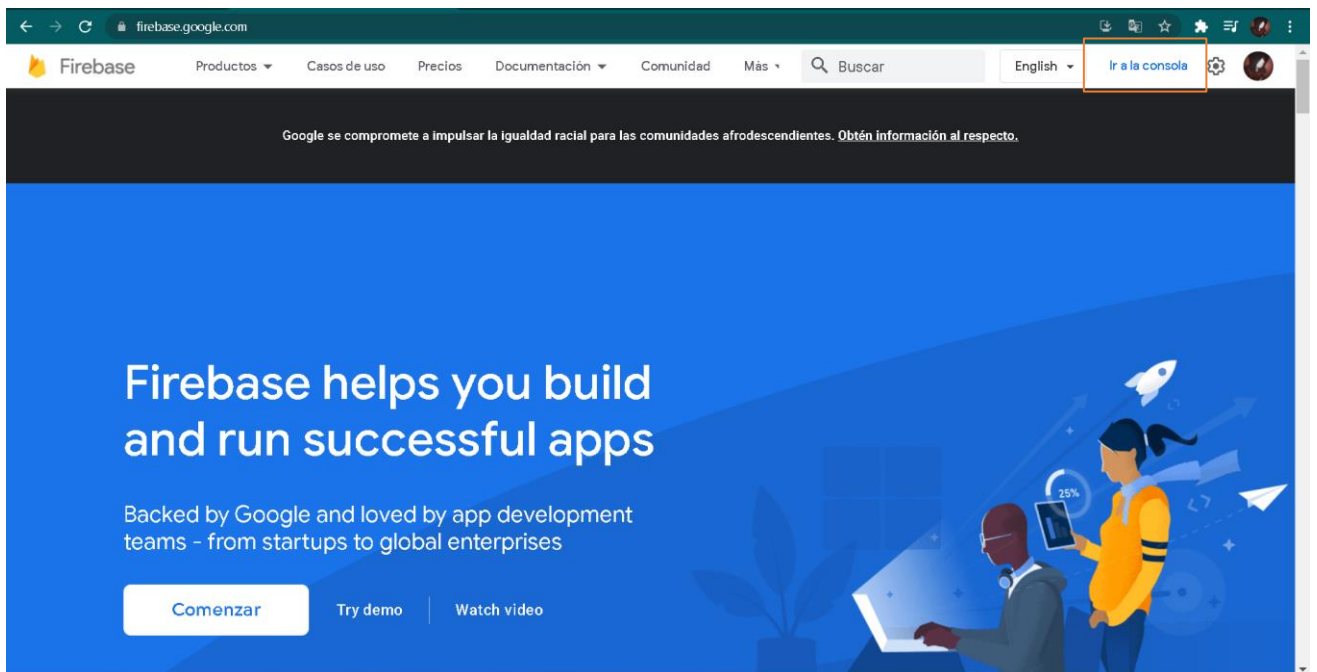


Fig. 4. Logeado e Ir a la consola

Luego de dar click al botón “Ir a la consola” Como no tenemos un proyecto previo aparecerá un botón que dice “Crear un nuevo proyecto” donde deberemos dar click para rellenar un

formulario con los datos de nuestro proyecto, el primero de estos será el nombre de nuestro proyecto y aceptar las condiciones del servicio de firebase para poder aceptar (Fig. 6).

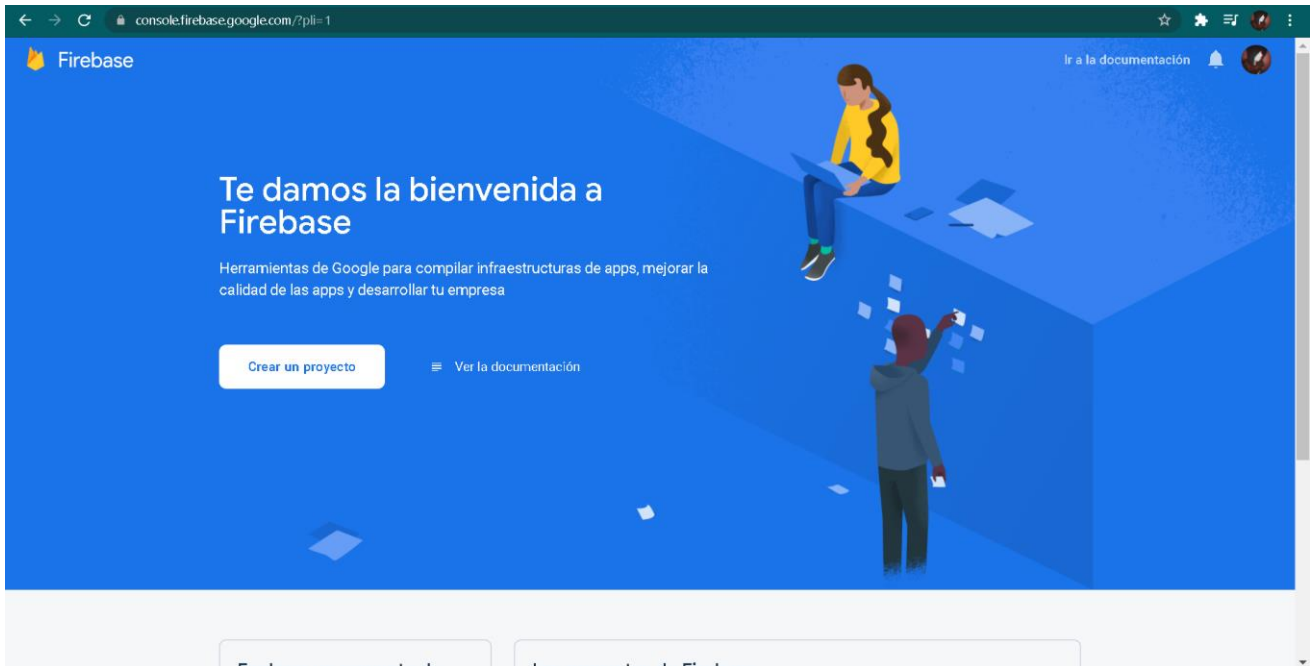


Fig. 5. Crear un nuevo proyecto



Fig. 6. Nombre del proyecto y condiciones

Después de dar en continuar nos dirá si queremos activar google analytics para este proyecto, le daremos que no porque no lo necesitamos para esta guía y podremos darle en "Crear proyecto" donde deberemos esperar un poco.

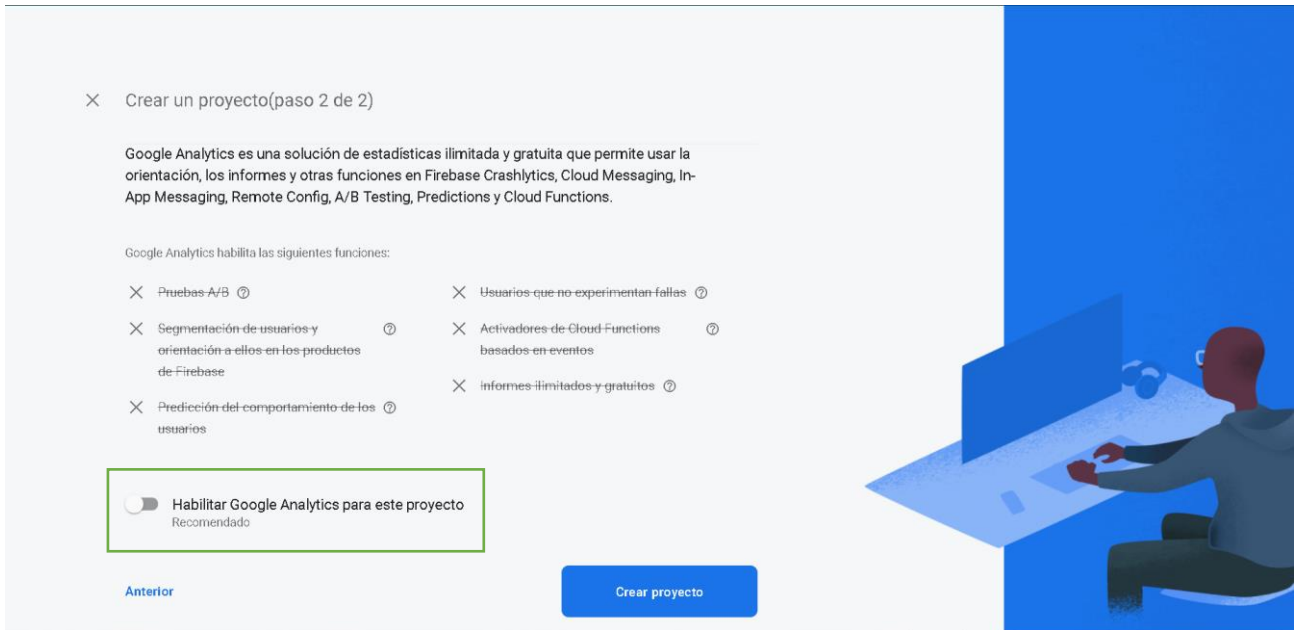


Fig. 7. Google Analytics

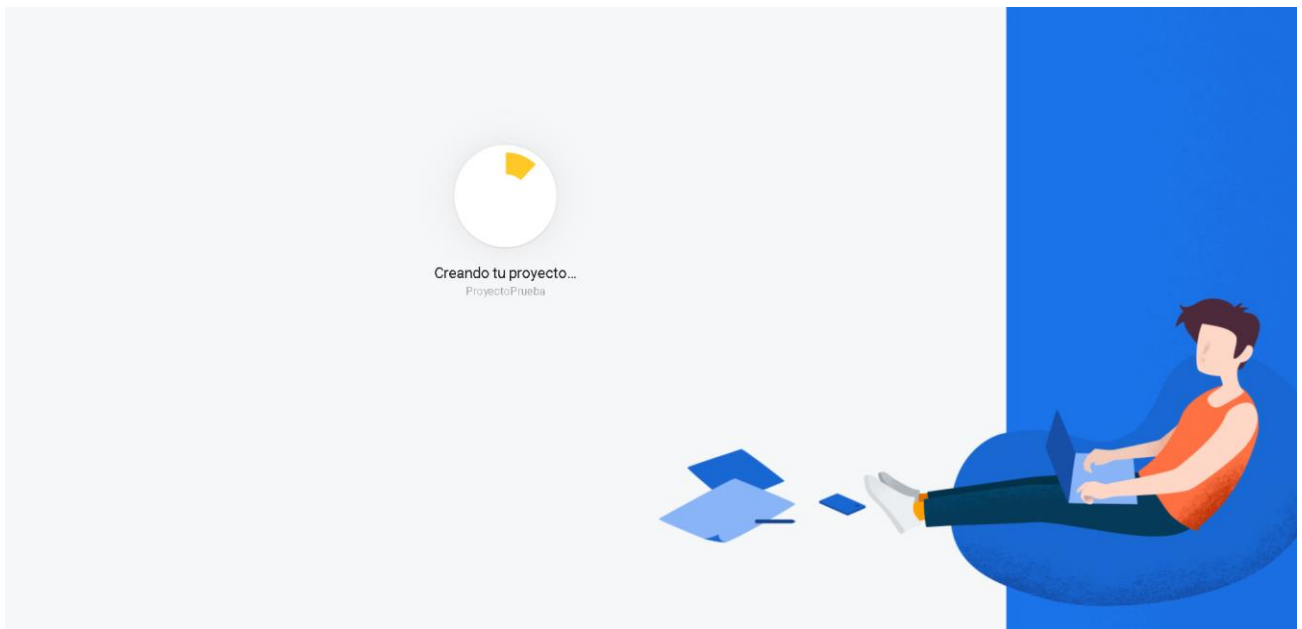


Fig. 8. Esperar que se cree el proyecto

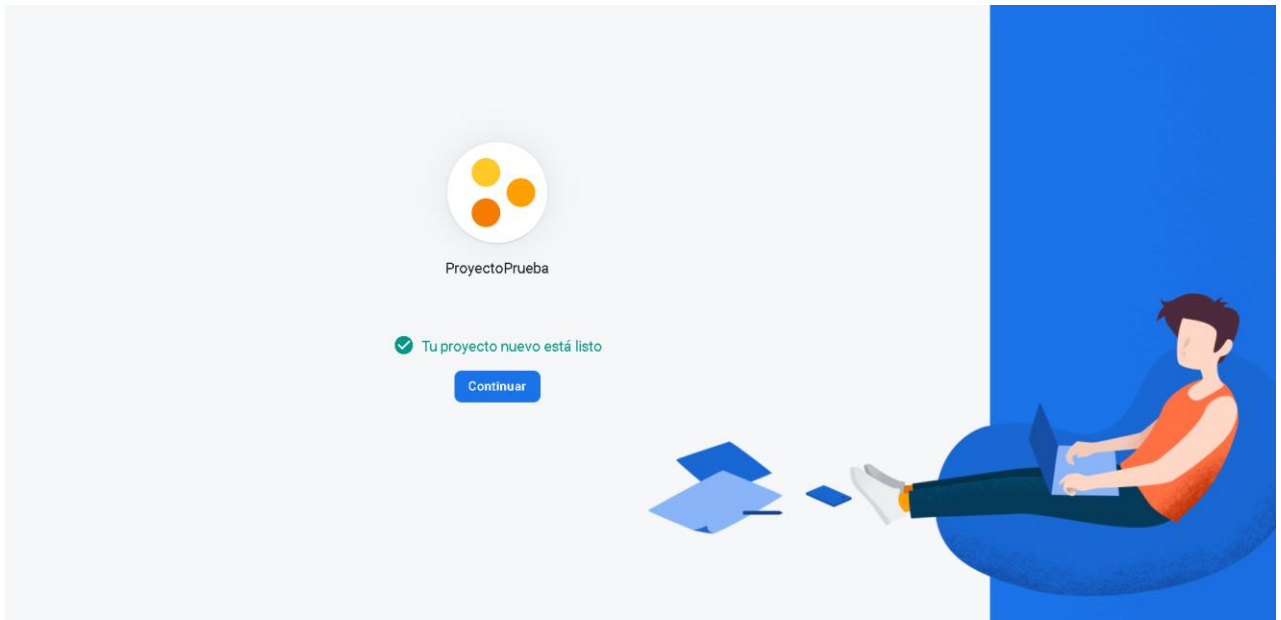


Fig. 9. Proyecto creado

Una vez le demos en continuar aparecerá la interfaz del proyecto donde deberemos realizar unas configuraciones antes. Veremos un icono de Unity y deberemos dar click en el.



Fig. 10. Interfaz del proyecto

Una vez se presione el icono de unity aparecerá una interfaz con diferentes opciones, vamos a hacer el login para escritorio, pero igual sirve para móvil por lo que seleccionaremos la opción de Android App, ya que funciona bien para ambos casos. Se deben poner algunas configuraciones que nos arroja unity por lo que antes de seguir deberemos crear nuestro proyecto base en unity.

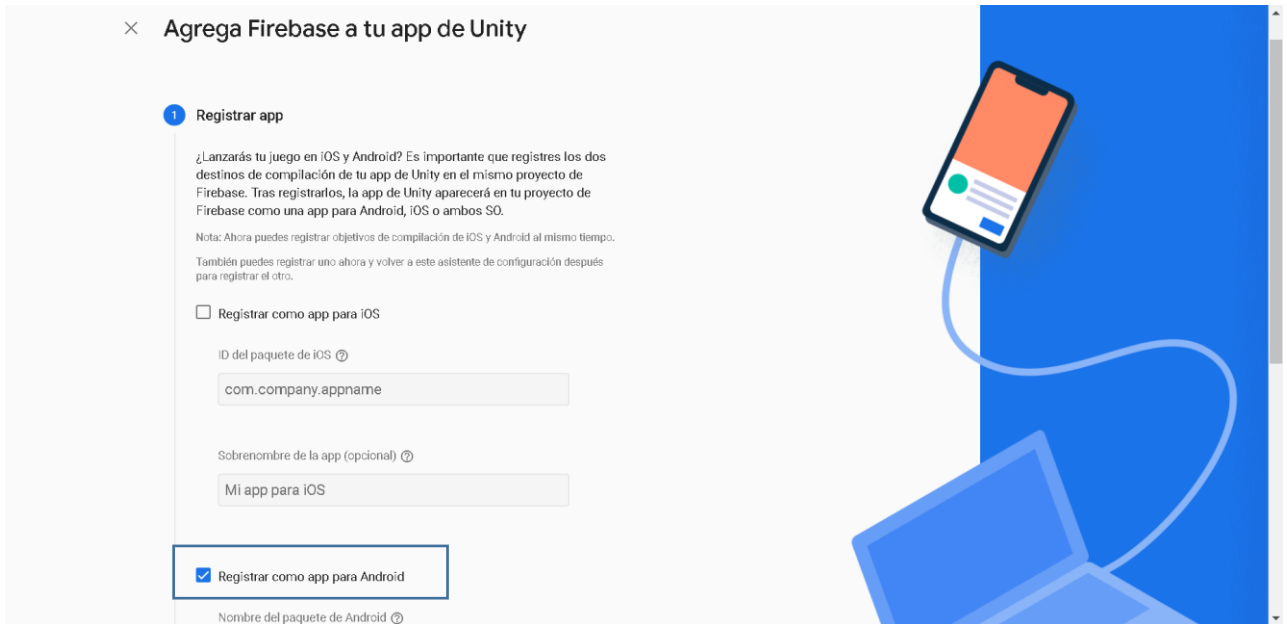


Fig. 11. Seleccionar Android App

Crearemos un proyecto en unity, que sea 3D aunque puede ser 2D dando en el botón “New” de UnityHub, lo nombraremos como queramos y crearemos una interfaz simple con botones, inputfield y textos como se ve en las siguientes figuras.

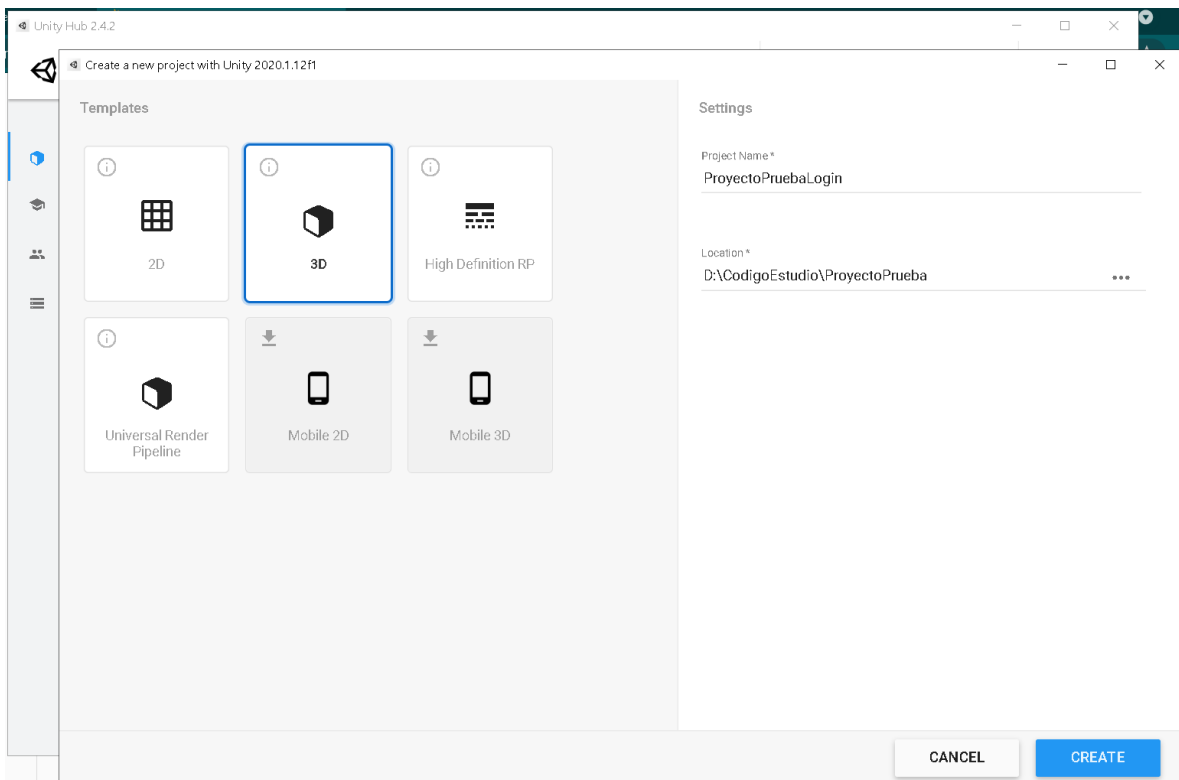


Fig. 12. Crear nuevo proyecto en unity

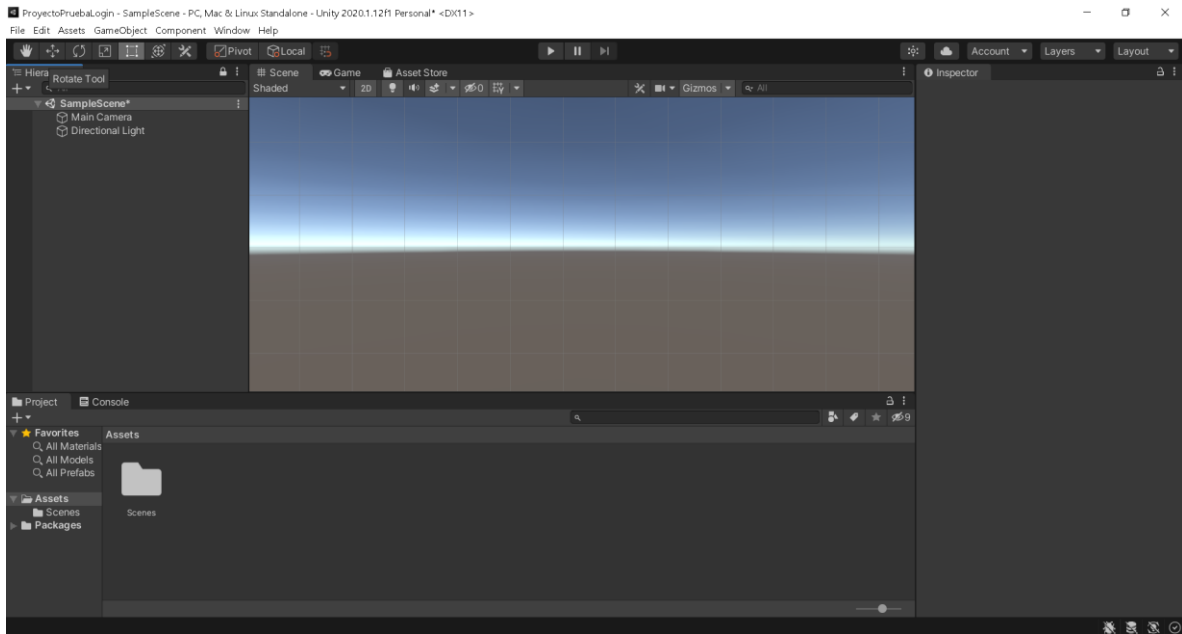


Fig. 13. Interfaz Unity

Seleccionaremos la opción de 2D al lado del bombillo para una mejor visualización y crearemos dos “GameObjects” que serán nuestra interfaz, se nombrara uno “Login” y el otro “Register”, cada uno con un canvas y un panel. De la siguiente manera.

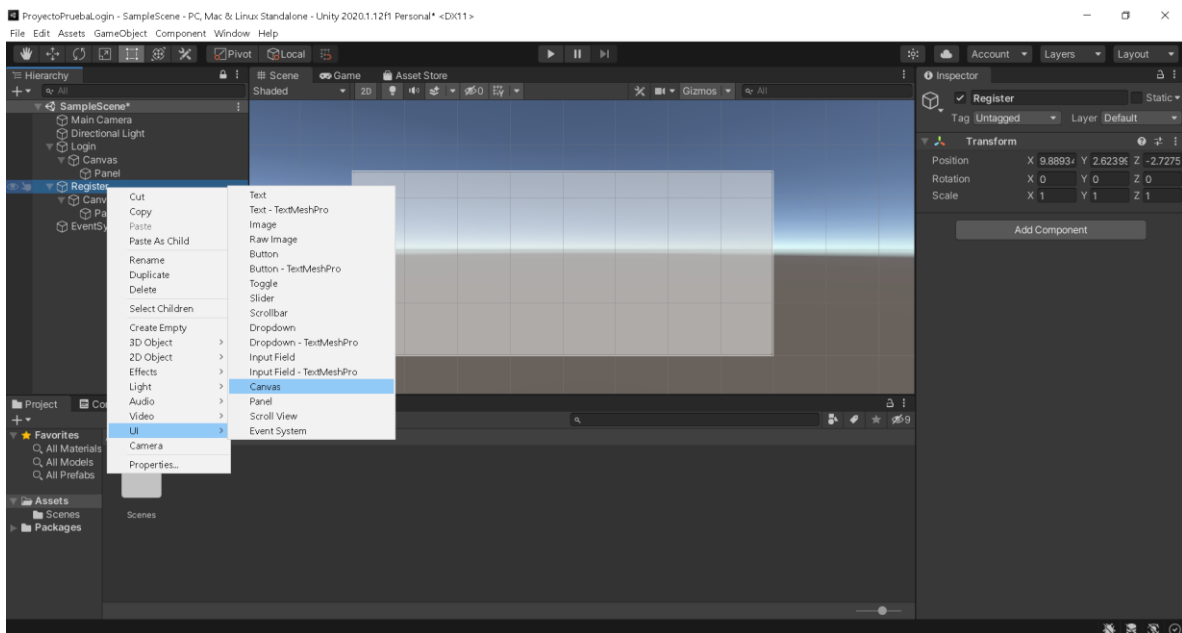


Fig. 14. Crear componentes

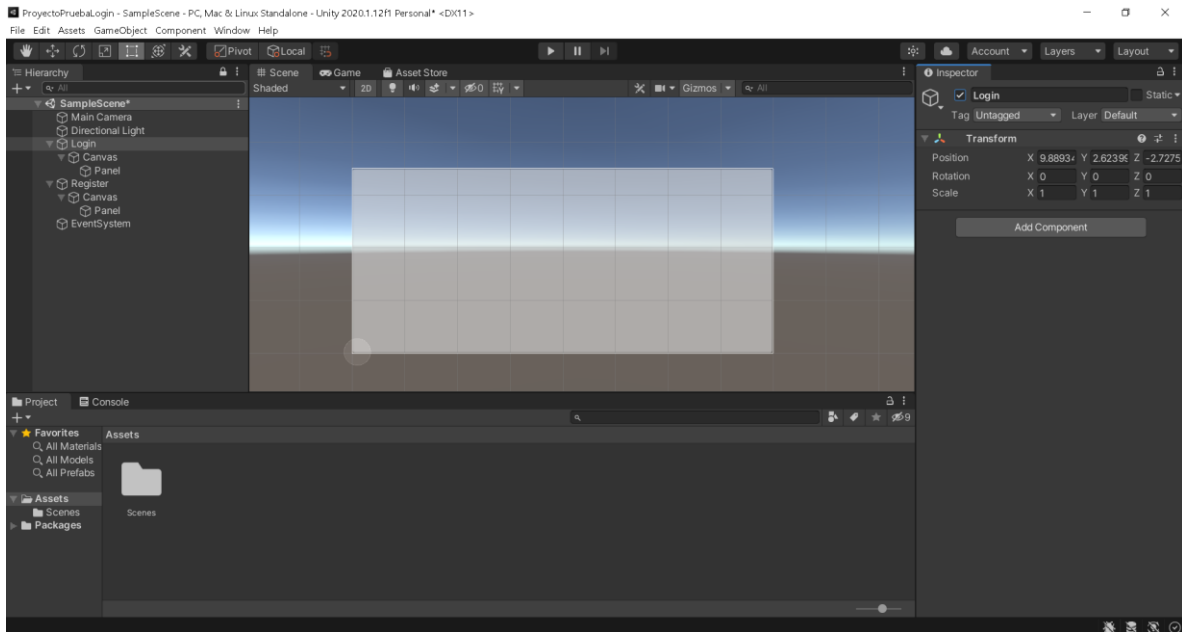


Fig. 15. Componentes creados

Desactivamos el register para poder trabajar con el login, creamos dos inputfield donde colocaremos el correo y la contraseña, y dos botones que sean para o logearse con las credenciales de los input o registrarnos llevándonos a la interfaz que después haremos de registrar.

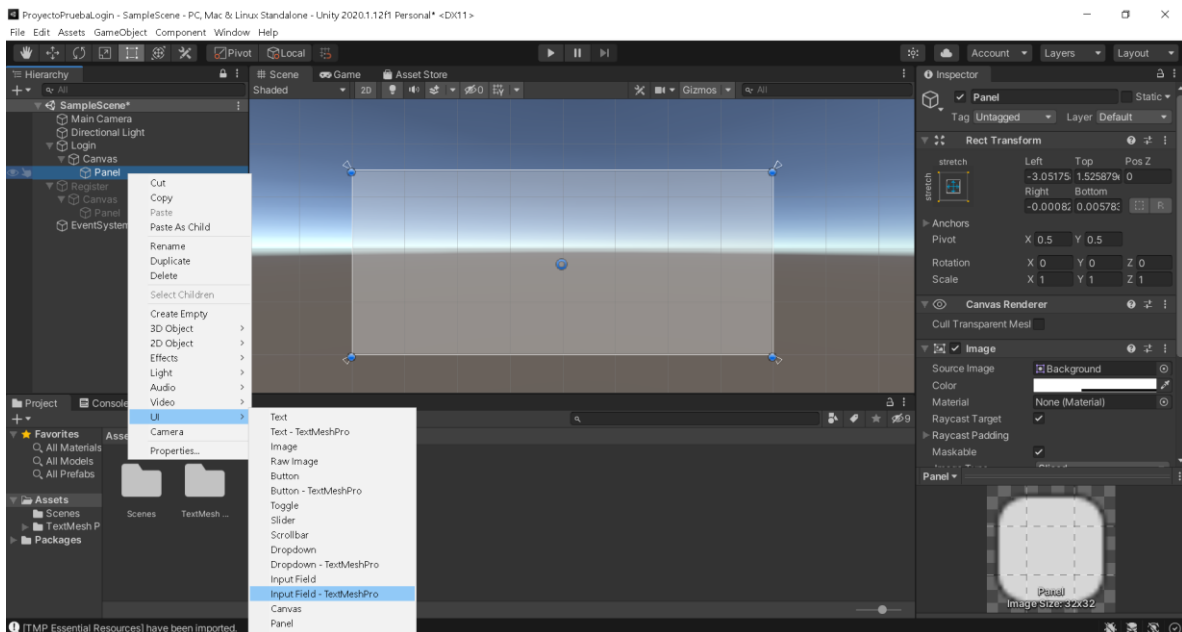


Fig. 16. Crear inputfield

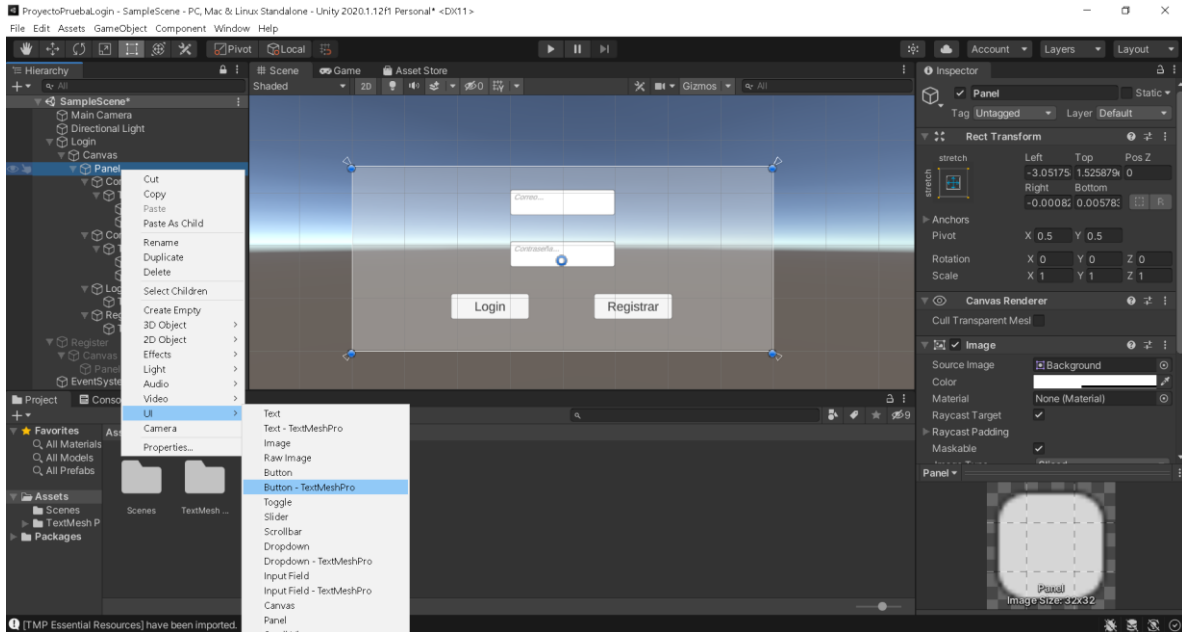


Fig. 17. Crear botones

Se renombran los botones, y los inputs para hacerlos más acorde al propósito y para ordenar. Dando como resultado lo siguiente.

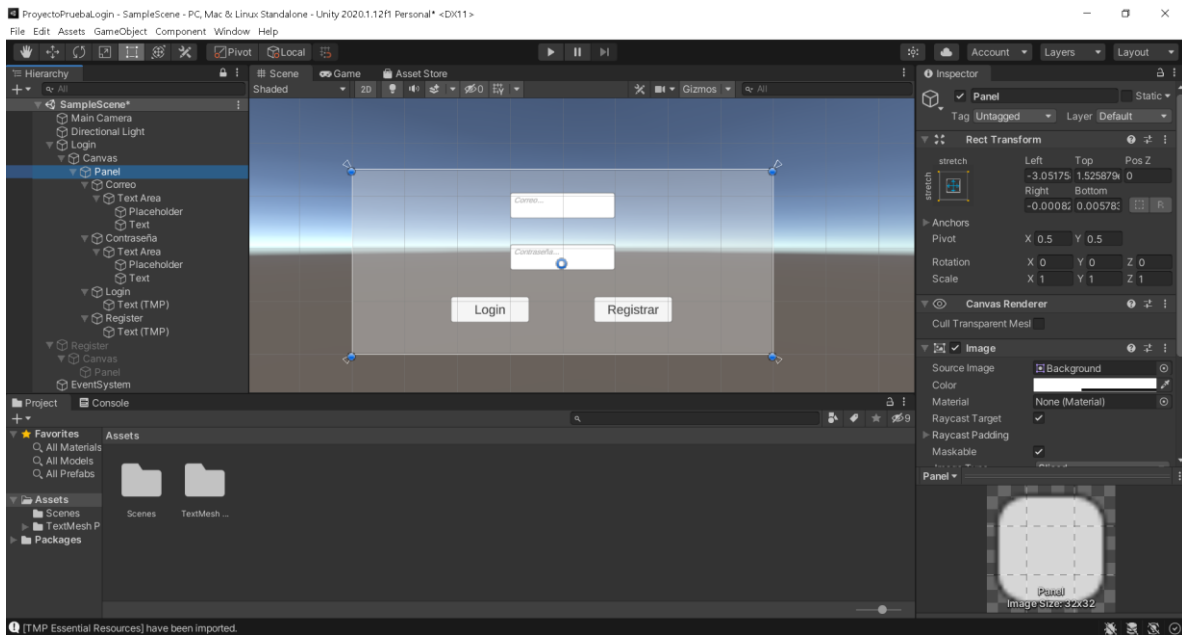


Fig. 18. Interfaz de login

Ahora desactivamos esta interfaz y construimos la de registro que contendrá 4 inputfields, usuario, correo, contraseña y verificar contraseña. Además, un botón que sea de registrar y así ya poderse logear. Quedando de la siguiente manera.

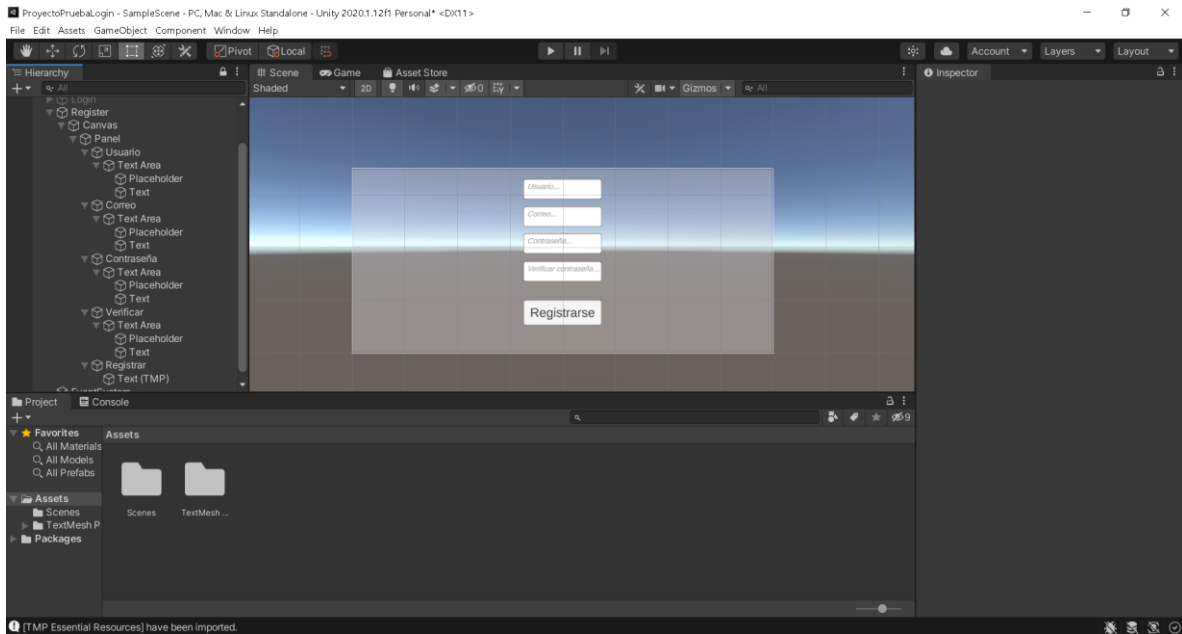


Fig. 19. Interfaz de registro

Ya teniendo listo nuestro proyecto base en unity, vamos a la sección file de unity, luego a build settings y player settings

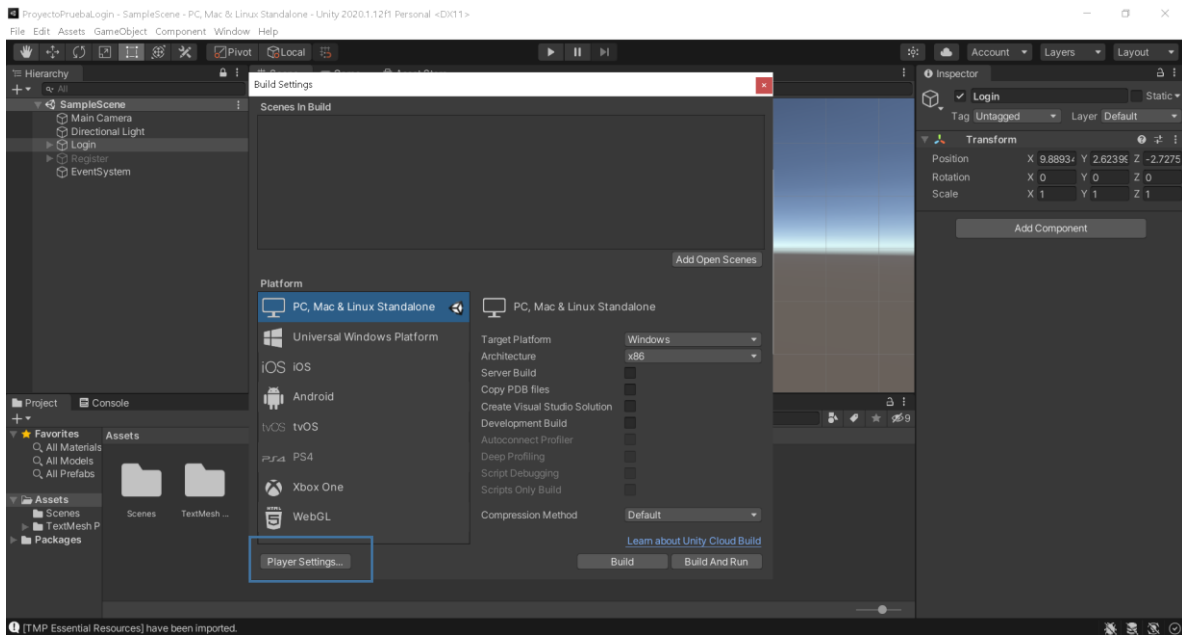


Fig. 20. Player settings

Buscamos la sección other settings y copiamos lo siguiente para colocarlo en el proyecto de firebase y luego registrar la app.

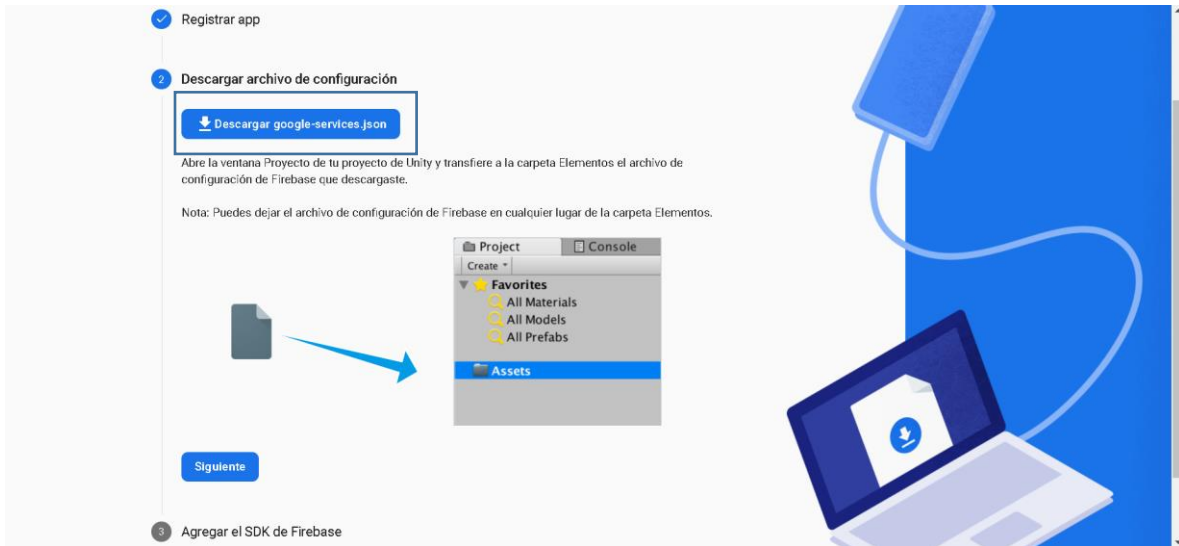


Fig. 23. Descargar json

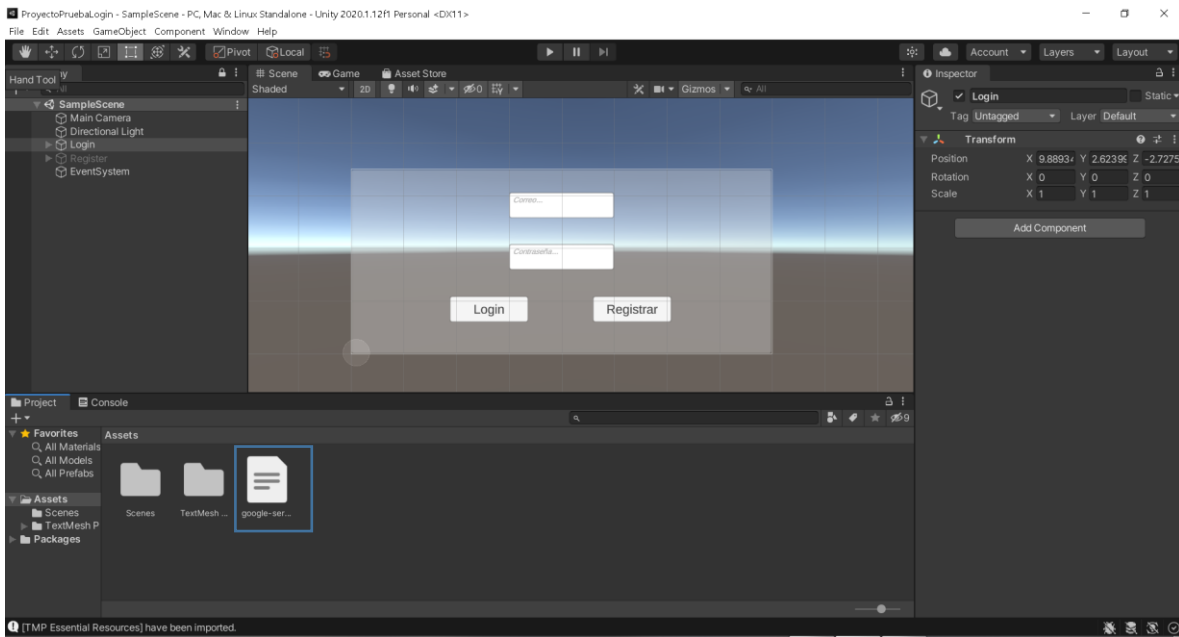


Fig. 24. Pegar en assets el archivo descargado

Luego podremos dar en siguiente en firebase para descargar el SDK que nos permitirá la comunicación de unity con firebase.

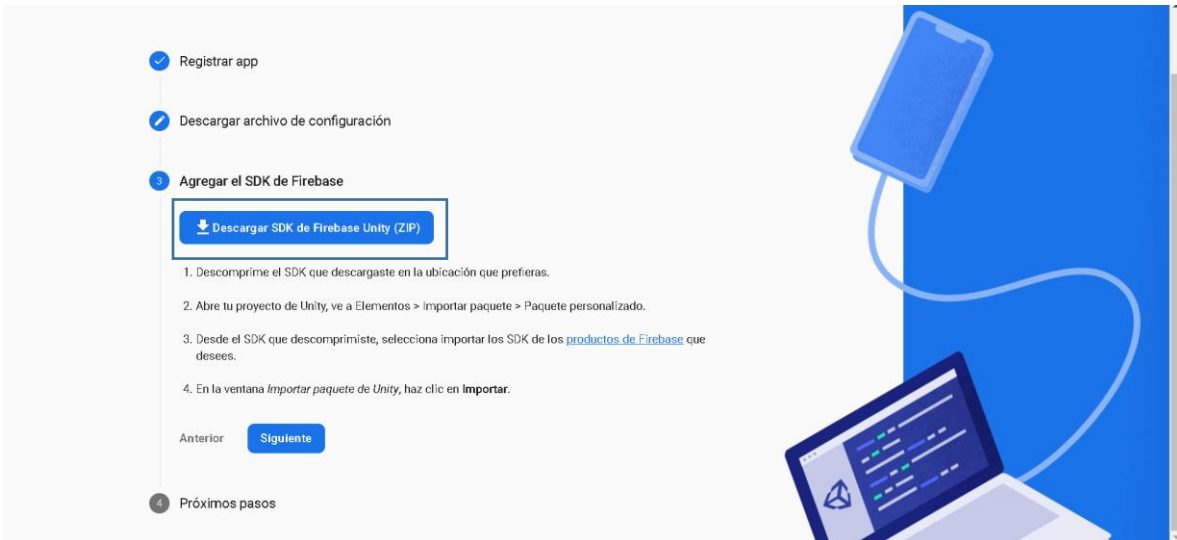


Fig. 25. Descargar SDK.zip

Después de descargar, lo extraemos y desde unity accedemos a la carpeta donde lo descomprimimos e importamos en la pestaña Assets, importpackage donde deberemos importar “FirebaseAuth.unitypackage”

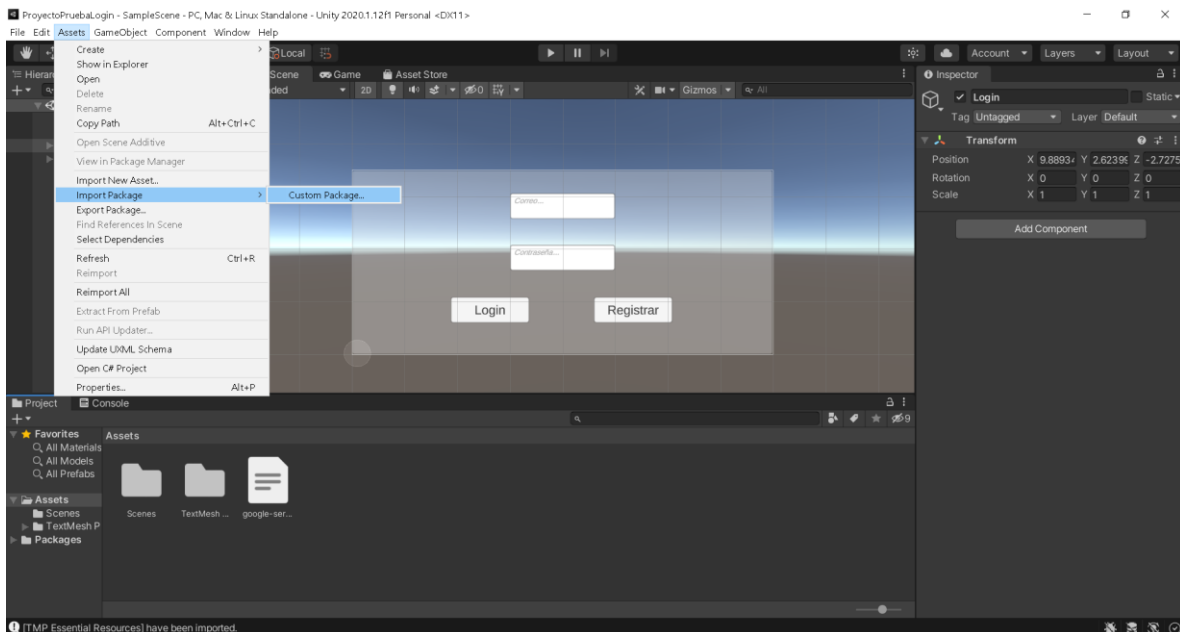


Fig. 26. Importar paquetes

🔍	FirebaseAnalytics.unitypackage	30/04/2021 01:40 ...	Unity package file
🔍	FirebaseAuth.unitypackage	30/04/2021 01:40 p. m.	Unity package file
🔍	FirebaseCrashlytics.unitypackage	30/04/2021 01:40 ...	Unity package file
🔍	FirebaseDatabase.unitypackage	30/04/2021 01:41 ...	Unity package file
🔍	FirebaseDynamicLinks.unitypackage	30/04/2021 01:41 ...	Unity package file
🔍	Firestore.unitypackage	30/04/2021 01:42 ...	Unity package file
🔍	FirebaseFunctions.unitypackage	30/04/2021 01:41 ...	Unity package file
🔍	FirebaseInstallations.unitypackage	30/04/2021 01:42 ...	Unity package file
🔍	FirebaseInstanceId.unitypackage	30/04/2021 01:42 ...	Unity package file
🔍	FirebaseMessaging.unitypackage	30/04/2021 01:41 ...	Unity package file
🔍	FirebaseRemoteConfig.unitypackage	30/04/2021 01:41 ...	Unity package file
🔍	FirebaseStorage.unitypackage	30/04/2021 01:41 ...	Unity package file

Fig. 27. Paquetes a importar

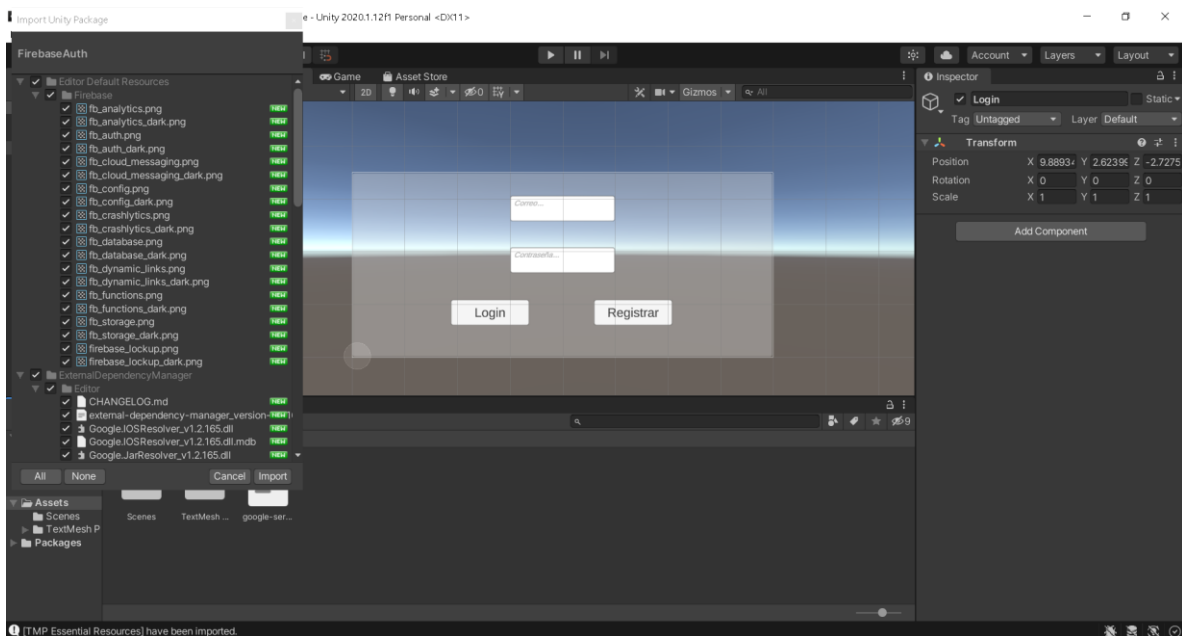


Fig. 28. Importar

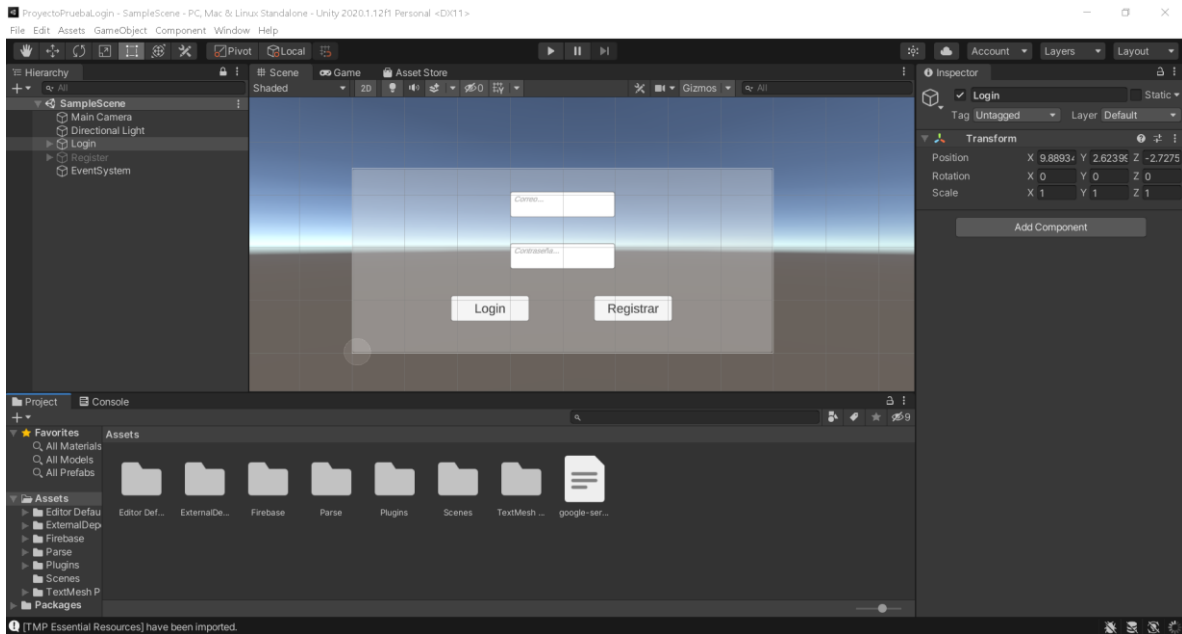


Fig.29. Carpetas que se crean al importar

Después de importar los paquetes, vamos a firebase donde ya podemos avanzar a l último paso que nos llevara a la consola de nuevo. En la consola iremos al lado izquierdo donde está la sección de autenticación»

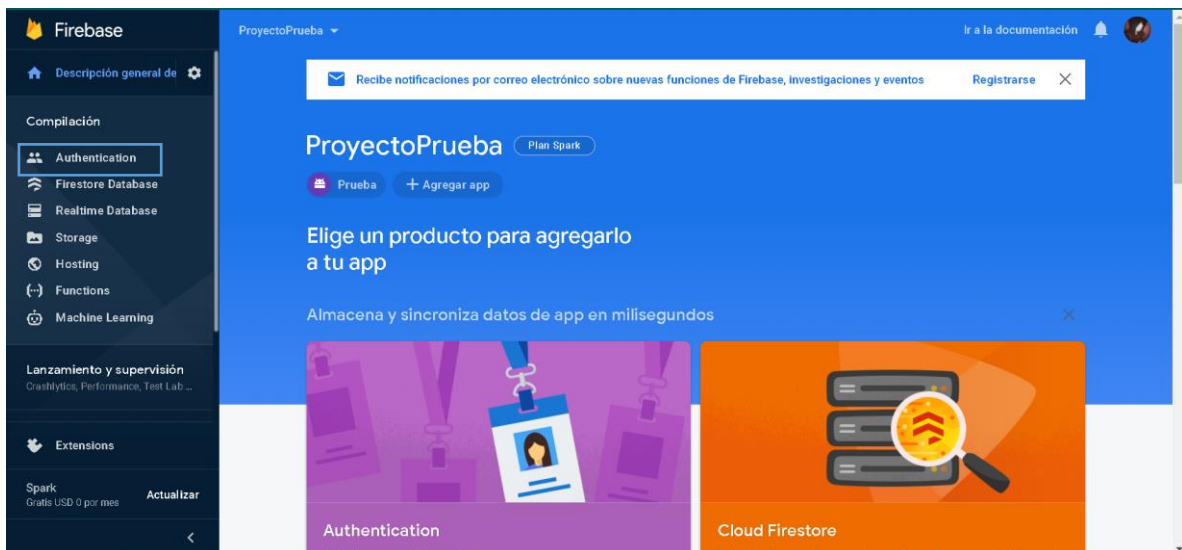


Fig. 30. Sección autenticación

Daremos en comenzar y escogeremos un método de autenticación, para esta guía será como email, por lo que habilitamos la primera opción dando en el lápiz al final de la opción dicha.

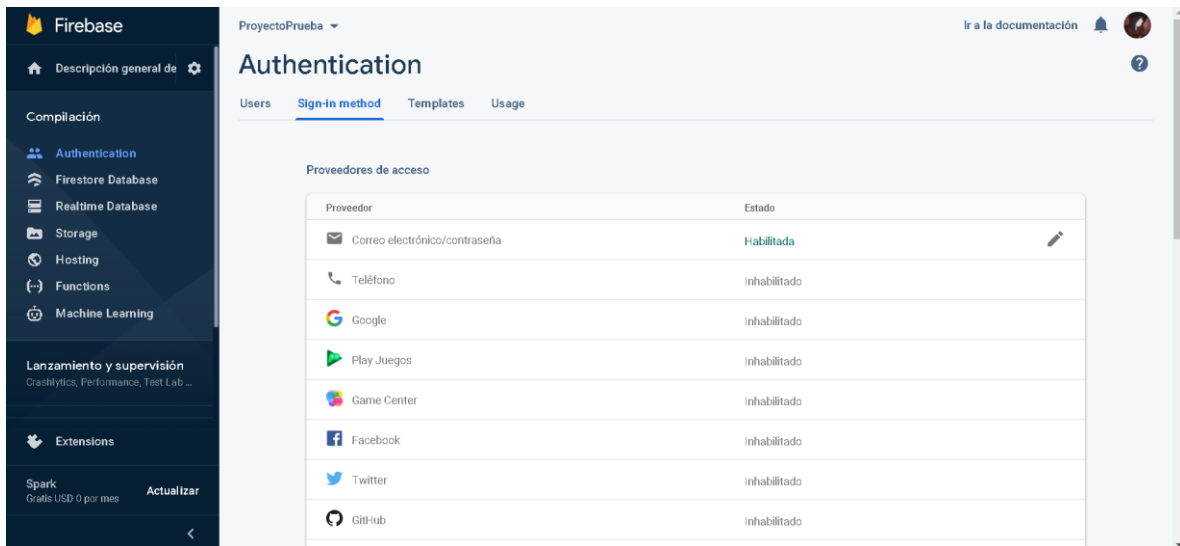


Fig. 31. Autenticación vía correo electrónico con contraseña

Una vez habilitada, vamos a unity donde creamos dos “GameObject” que llamaremos “AuthManager” y “UiManagaer” también crearemos una carpeta llamada “Scripts” donde se guardaran los códigos de esta guía.

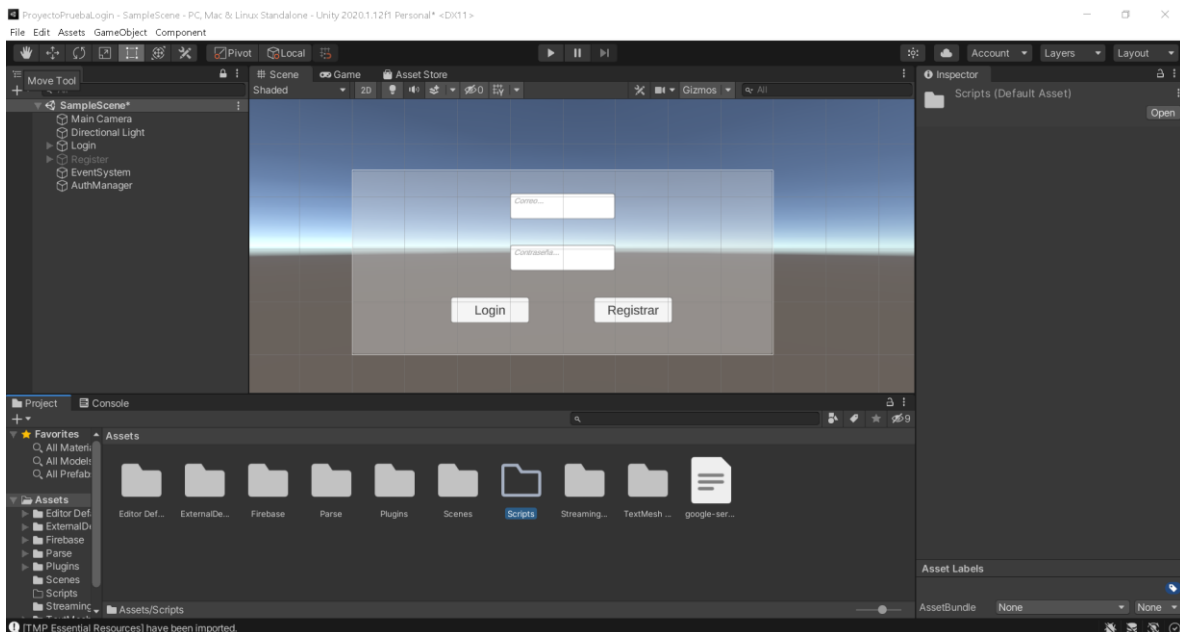


Fig. 32. Crear carpeta y GameObject.

Crearemos dos scripts llamado “AuthManager” y “UiManager” en la carpeta previamente creada y lo abriremos, usando la herramienta visual studio que ya viene con la instalación de unity. A continuación todo el código que forma parte de AuthManager.

ANEXO B

Registro y creación de Base de datos en Firebase.

Inicialmente se accedió a la página principal de firebase (<https://firebase.google.com>) y registro con una cuenta, como firebase es una dependencia de Google solo hace falta entrar con un correo Gmail para poder acceder a las funciones que ofrece firebase.

Una vez iniciada la sesión se agrega un nuevo proyecto como se muestra en la figura 1,

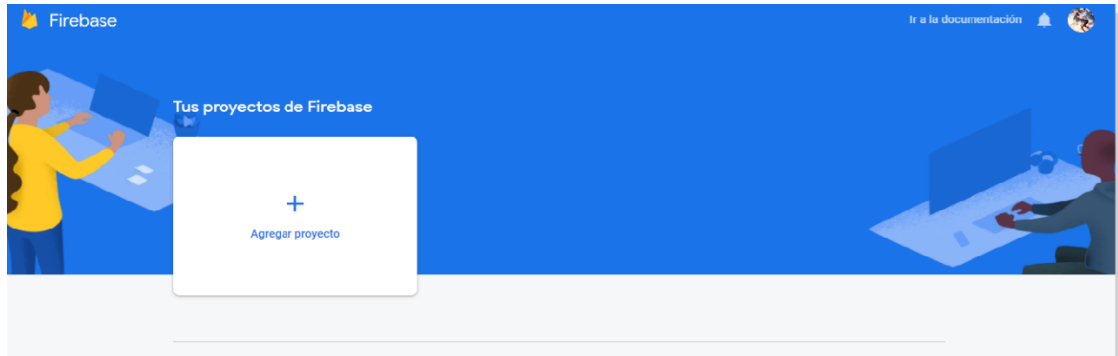


Figura 1. Pantalla de creación de proyecto.

Se le asigna un nombre como se indica en la figura 2, luego se aceptan las analíticas las cuales habilitan las funciones de la figura 2, y por último se asigna la región y se aceptan los términos y condiciones para darle a la creación del proyecto.



Figura 2. Asignación de nombre al proyecto



Figura 3. Habilitación de analíticas de Firebase

Una vez creado el proyecto se deberá la ampliación asociada al proyecto, con la cual se va interactuar con firebase, como se muestra en la figura 4, firebase permite agregar una aplicación exclusiva para Android, iOS, Web y para Unity, en este caso se crea una aplicación para Unity

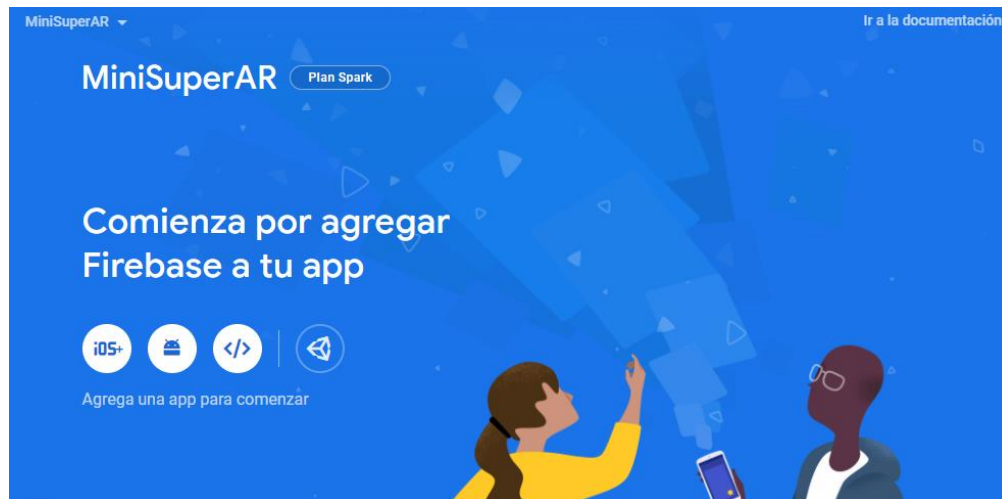


Figura 4. Interfaz para agregar una app

Para el registro de la aplicación de Unity, se piden los datos de registro de la app para Apple, para Android o para las dos al tiempo, como se muestra en la figura 5.

Registrar como app para Apple

ID del paquete de Apple ⓘ

com.supermarker.MinisuperAR

Sobrenombre de la app (opcional) ⓘ

MinisuperAR

Registrar como app para Android

Nombre del paquete de Android ⓘ

com.supermarker.MinisuperAR

Sobrenombre de la app (opcional) ⓘ

MiniuperAR

Figura 5. Registro de aplicación para Android o Apple

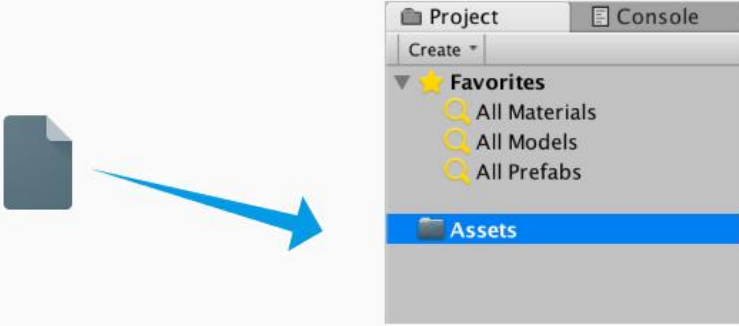
Una vez registrados los datos se deberán descargar los archivos necesarios, como se muestra en la figura 6, para el reconocimiento de los datos y conexión de la aplicación de Unity con firebase, estos archivos se agregan en la carpeta Assets del proyecto de Unity.

[Descargar google-services.json](#)

[Descargar GoogleService-Info.plist](#)

Abre la ventana Proyecto de tu proyecto de Unity y transfiere a la carpeta Elementos los dos archivos de configuración de Firebase que descargaste.

Nota: Puedes dejar los archivos de configuración de Firebase en cualquier lugar de la carpeta Elementos.



[Siguiente](#)

Figura 6. Ventana para descargar GoogleService

Como se muestra en la figura 7, luego se descarga el SDK de Firebase para Unity.

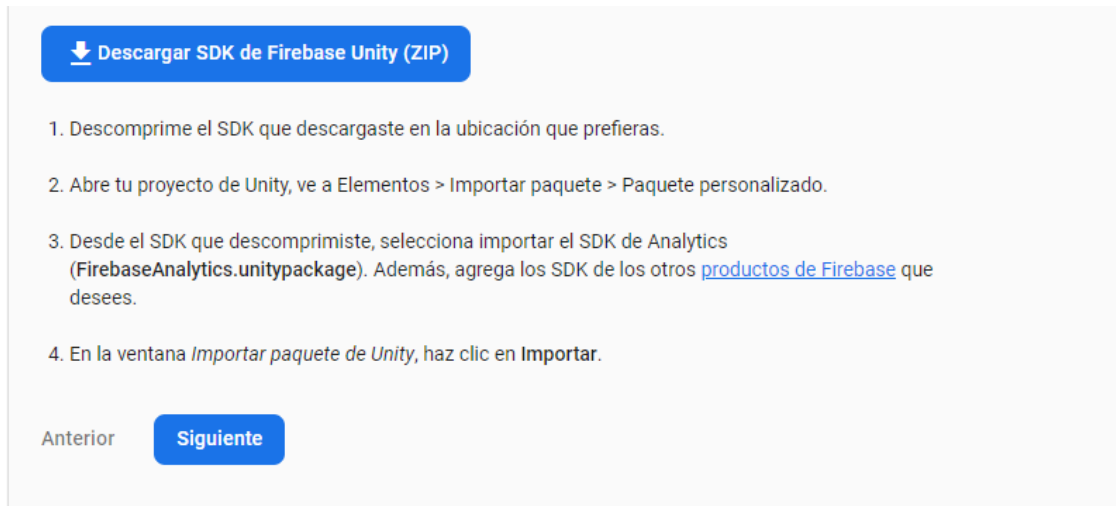


Figura 7.ventana de descarga de SDK de Firebase

Ahora se creará la base de datos, en el lado izquierdo en la parte de compilación, como se ve en la figura 8, se selecciona la base de datos en tiempo real (Realtime Database) y luego en crear una base de datos



Figura 8. Interfaz para creación de base de datos

Luego se selecciona una ubicación para la base de datos, como se ve en la figura 9, y se seleccionara el modo en el que empezarán las reglas de la base de datos, los cuales se muestran en la figura 10.

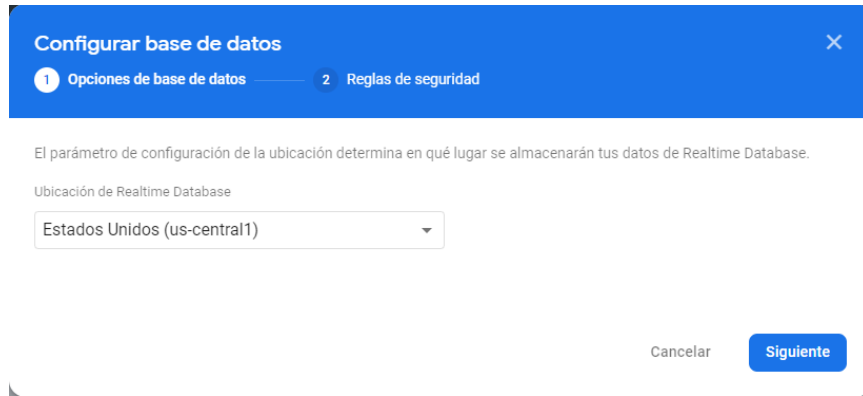


Figura 9. Ubicación de la base de datos



Figura 10. Modo de reglas de la base de datos

En la figura 11, se muestra la base de datos creada lista para configurar.



Figura 11. Base de datos creada

Instalación de paquete base de datos de firebase en Unity

Una vez descargado el sdk de firebase para Unity quedara un archivo .zip y al descomprimirlo quedara una carpeta con el mismo nombre, como se muestra en la figura 12.

Nombre	Fecha de modificación	Tipo	Tamaño
firebase_unity_sdk	15/02/2022 8:10 a. m.	Carpeta de archivos	
firebase_unity_sdk_8.8.0	15/02/2022 8:39 a. m.	Archivo WinRAR Z...	1.987.258 KB

Figura 12. SDK de firebase para Unity

Al interior de la carpeta, como se observa en la figura 13, se encontrarán 2 carpetas (dotnet3 y dotnet4), dependiendo de la compatibilidad de la versión de Unity con Net3 o Net4 se escogerá la carpeta necesaria.

Nombre	Fecha de modificación	Tipo	Tamaño
dotnet3	31/01/2022 4:00 p. m.	Carpeta de archivos	
dotnet4	31/01/2022 4:00 p. m.	Carpeta de archivos	
LICENSE	28/01/2022 4:51 p. m.	Archivo	15 KB
readme.md	28/01/2022 4:51 p. m.	Archivo MD	82 KB

Figura 13. Archivos dentro del .zip

Como se observa en la figura 14, dentro de ambas carpetas (dонтnet3 y dontnet4) se encuentras los paquetes para importar las funcionalidades de firease a Unity, en este caso se instalará el paquete de base de datos el cual corresponde al archivo FirebaseDatabase.

Nombre	Fecha de modificación	Tipo	Tamaño
FirebaseAnalytics	28/01/2022 4:49 p. m.	Unity package file	86.821 KB
FirebaseAuth	28/01/2022 4:50 p. m.	Unity package file	90.359 KB
FirebaseCrashlytics	28/01/2022 4:50 p. m.	Unity package file	86.280 KB
FirebaseDatabase	28/01/2022 4:51 p. m.	Unity package file	96.824 KB
FirebaseDynamicLinks	28/01/2022 4:50 p. m.	Unity package file	86.973 KB
Firestore	28/01/2022 4:51 p. m.	Unity package file	102.939 KB
FirebaseFunctions	28/01/2022 4:51 p. m.	Unity package file	88.668 KB
FirebaseInstallations	28/01/2022 4:51 p. m.	Unity package file	86.646 KB
FirebaseMessaging	28/01/2022 4:50 p. m.	Unity package file	87.406 KB
FirebaseRemoteConfig	28/01/2022 4:50 p. m.	Unity package file	89.216 KB
FirebaseStorage	28/01/2022 4:51 p. m.	Unity package file	94.095 KB

Figura 14. Paquetes de Firebase para Unity

Una vez tenemos los archivos, creamos un nuevo proyecto en Unity con el nombre de “Ejemplo FireBase”, como se ve en la figura 15.

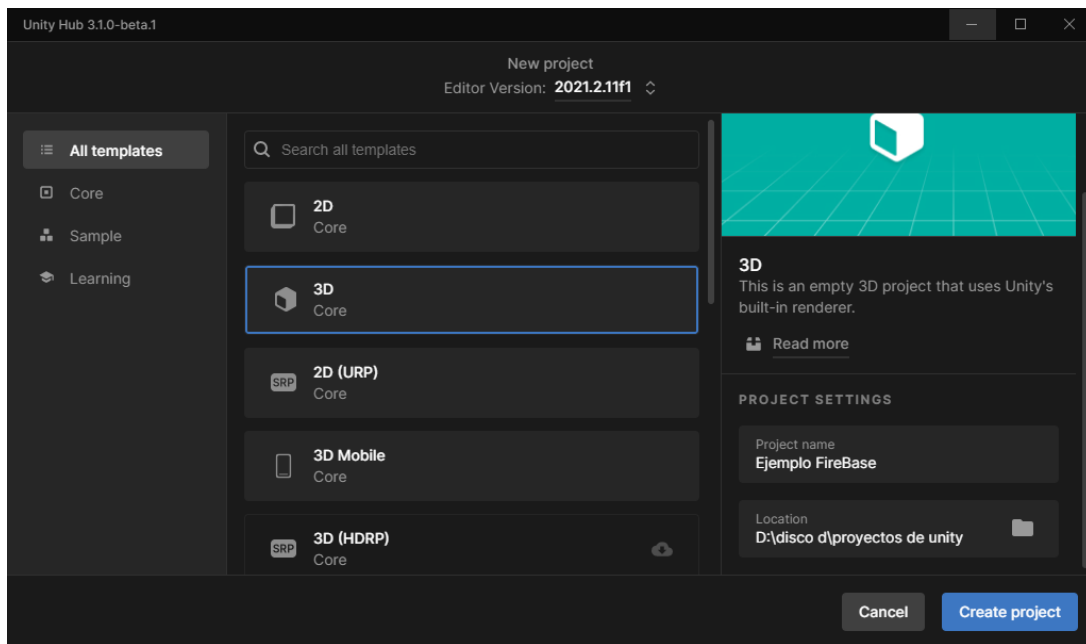


Figura 15. Creación de nuevo proyecto en Unity

En este caso, como es un proyecto para Android, vamos a la a la pestaña File y luego a Build Settings...(figura 16)

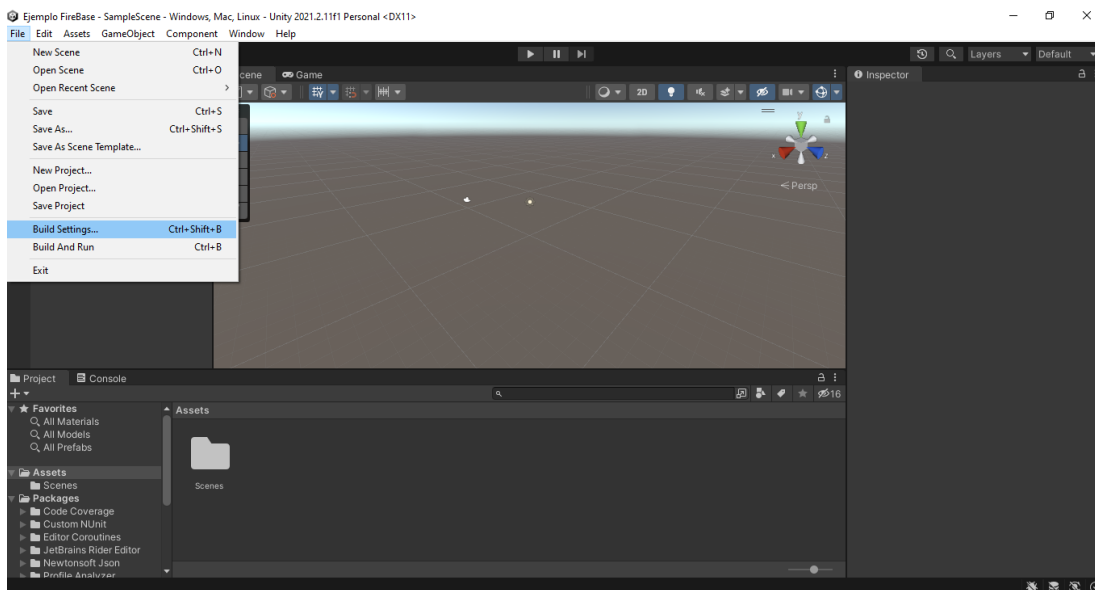


Figura 16. Opción de Build Settings

Luego en la ventana de Build Settings vamos a Android y luego a Switch Plataforma, como se ve en la figura 17, para cambiar la plataforma a la que va dirigida la app.

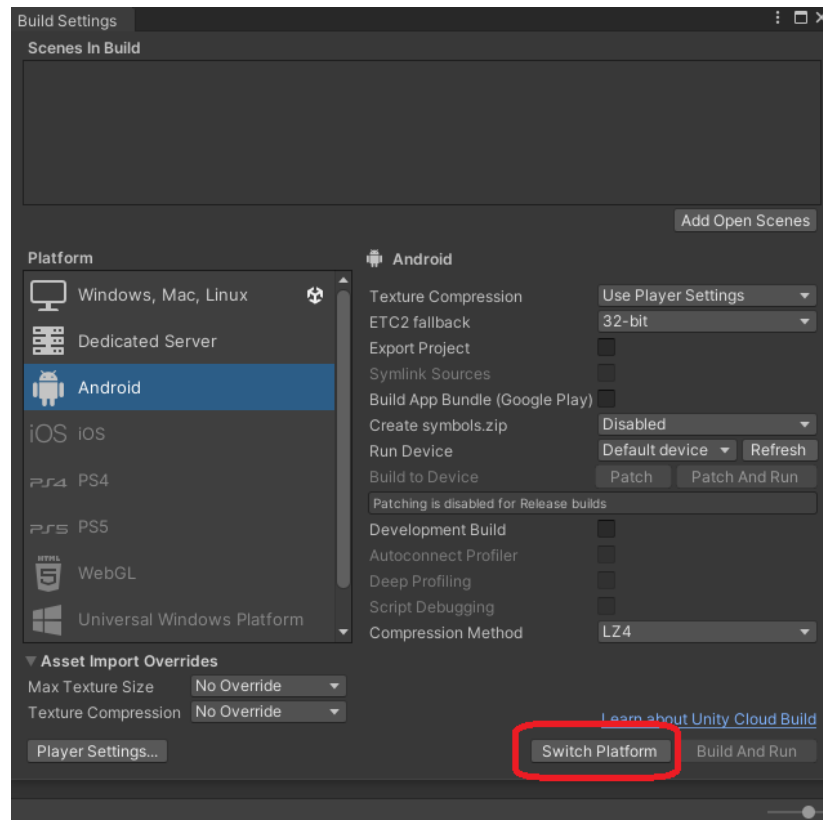


Figura 17. Cambio de plataforma

Una vez echo el cambio de plataforma, se agrega el paquete de Base de datos de Firebase, mediante la pestaña Assesst luego Import Package y Custom Package, como se ve en la figura 18.

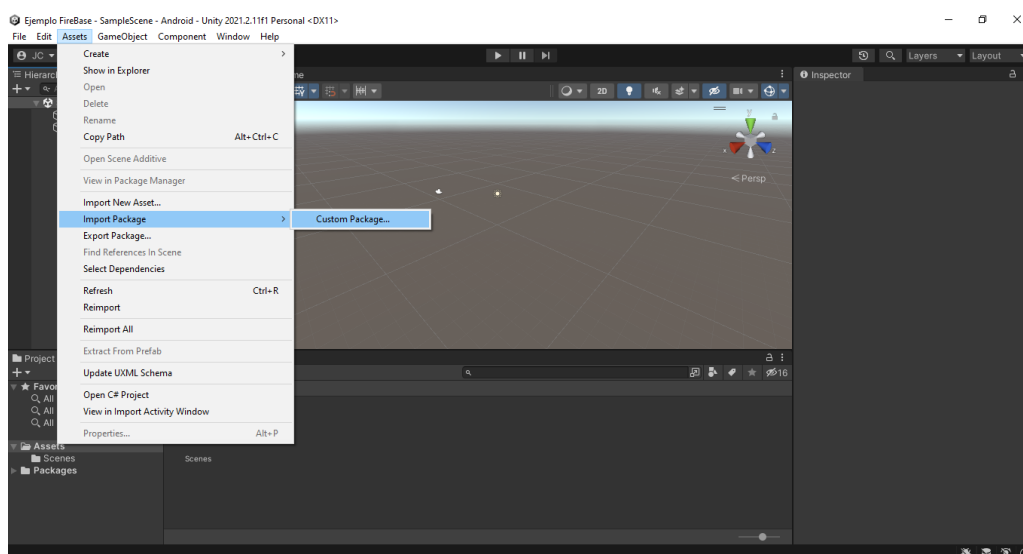


Figura 18. Importar Assets al proyecto de Unity

En la figura 19 se muestra una ventana para seleccionar el paquete, vamos a la carpeta donde están guardados los archivos de firebase descargados anteriormente, seleccionamos FirebaseDatabase y le damos en abrir.

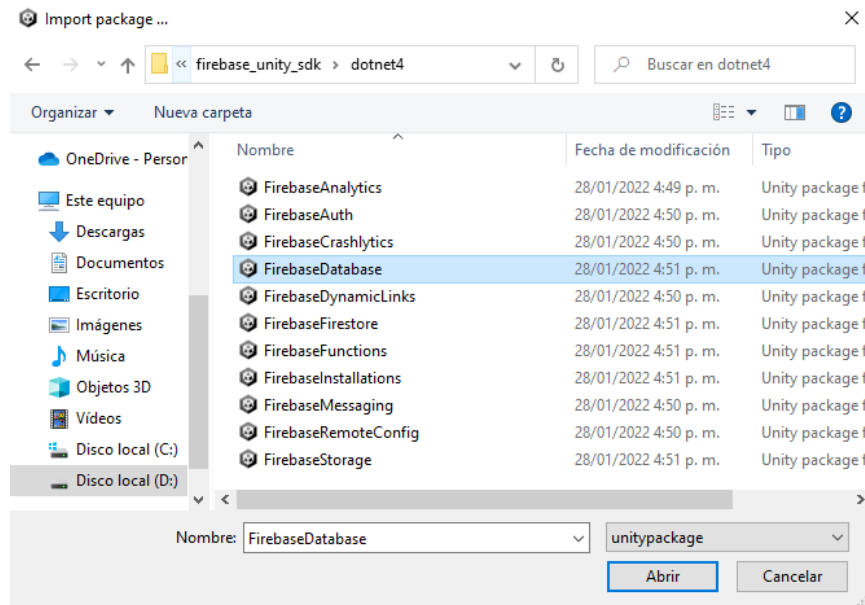


Figura 19. Selección de Paquete FirebaseDatabase

En la figura 20 se ve una ventana donde se muestran los archivos del paquete de FirebaseDatabase, y le damos en import para que empiece a importar el paquete a Unity.

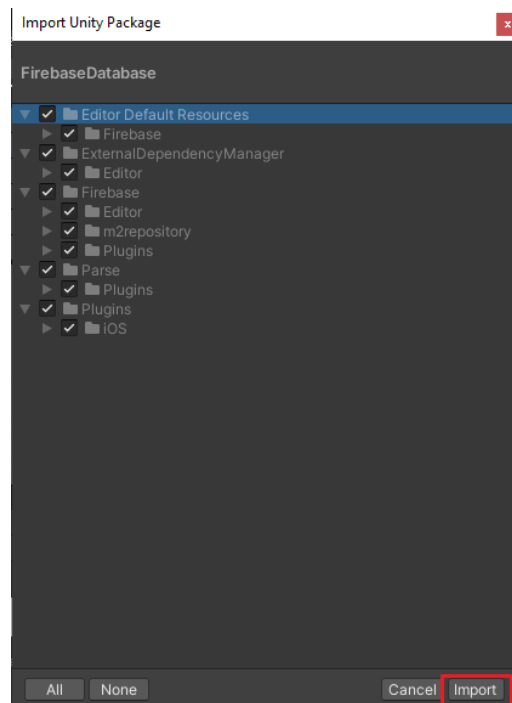


Figura 20. Ventana con archivos a importar

Como se ve en la figura 21, una vez haya finalizado el proceso de importación saldrán los archivos agregados a la carpeta Assets del proyecto de Unity y se podrán usar las Funcionalidades de FirebaseDatabase.

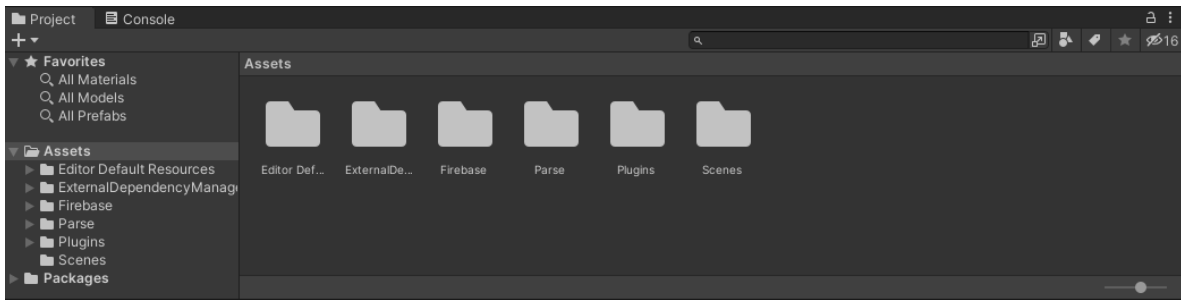


Figura 21. Carpetas importadas al proyecto

ANEXO C

Instalación de Paquete de Vuforia en Unity

Ingresamos a la página de vuforia, <https://developer.vuforia.com>, y nos dirigimos a la parte de registro como se ve en la figura 1.

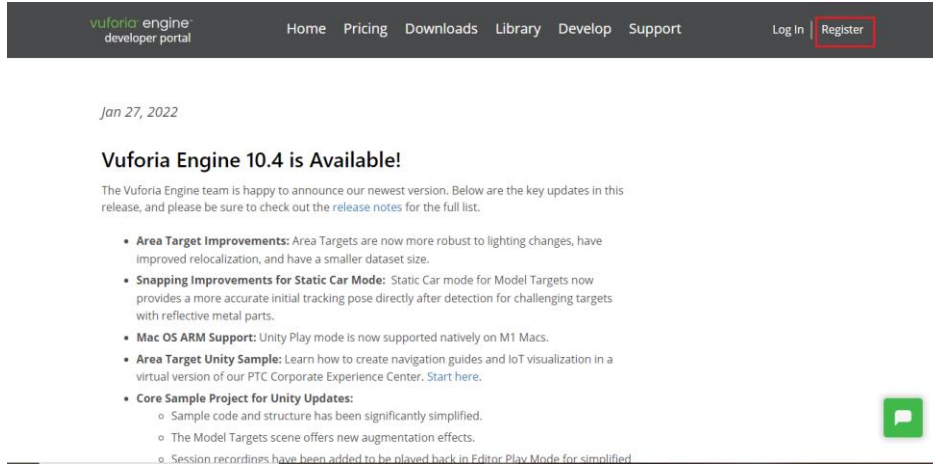


Figura 1. Selección de registro en página de Vuforia

Nos llevara a la ventana de registra, mostrada en la figura 2, e ingresaremos los datos necesarios para el registro

The image shows the registration form for a Vuforia Developer Account. The form is titled 'Register for a Vuforia Developer Account' and includes the following fields and options:

- First Name * (Nombre)
- Last Name * (Apellido)
- Company * (Independiente)
- Country * (Colombia)
- Email Address * (Correojemplo@gmail.com)
- Username * (Usuariocualquiera)
- Password *
- Confirm Password *

Below the form, there is a CAPTCHA section with a green checkmark and the text 'No soy un robot'. There are two checkboxes for terms and conditions:

- I agree to the terms of the Vuforia Developer Agreement.
- I acknowledge that my personal details will be processed in accordance with PTC's privacy policy and may be used for marketing purposes by PTC Inc. its subsidiaries and members of the PTC Partner Network, solely for the promotion of PTC's products and associated services.

At the bottom of the form, there is a green 'Create account' button.

Figura 2. Interfaz de registro en vuforia

Como se ve en la figura 3, se enviara un correo de verificación al correo ingresado en el registro

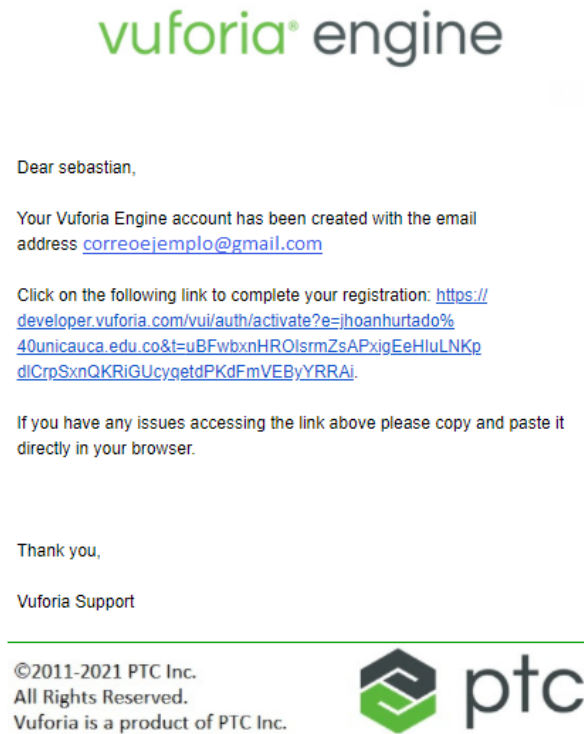


Figura 3. Correo de verificación

Como se ve en la figura 4, el enlace redirigirá a la pagina de vuforia y mostrara que la cuenta a sido activada con éxito.

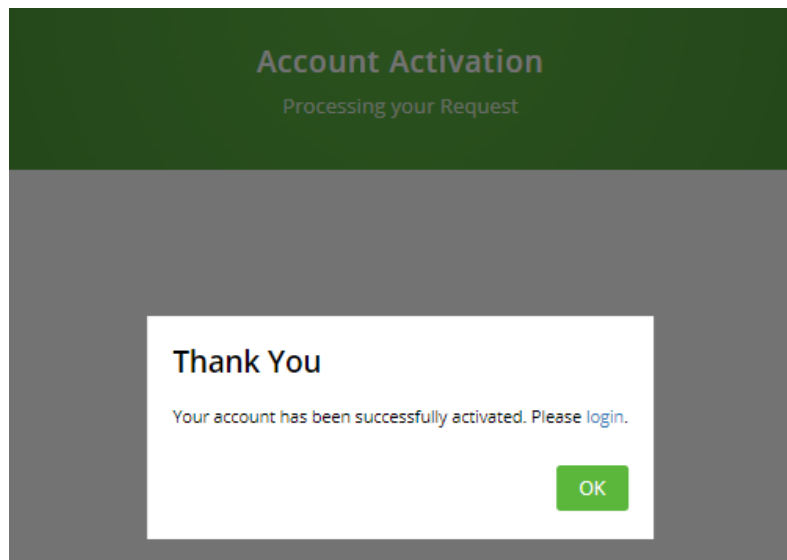
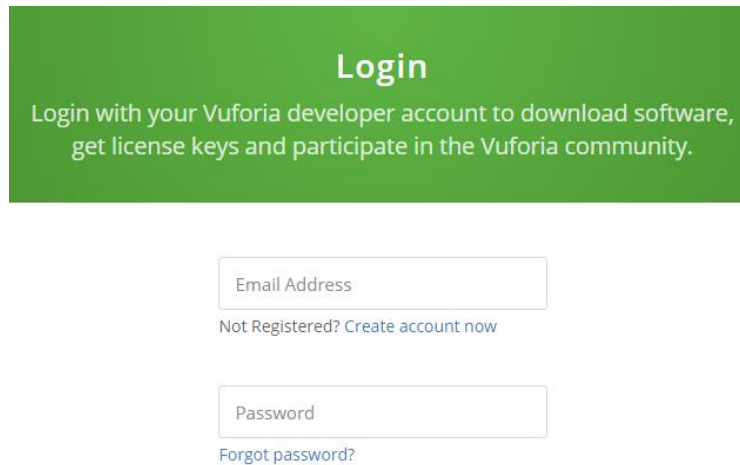


Figura 4. Cuenta creada con éxito

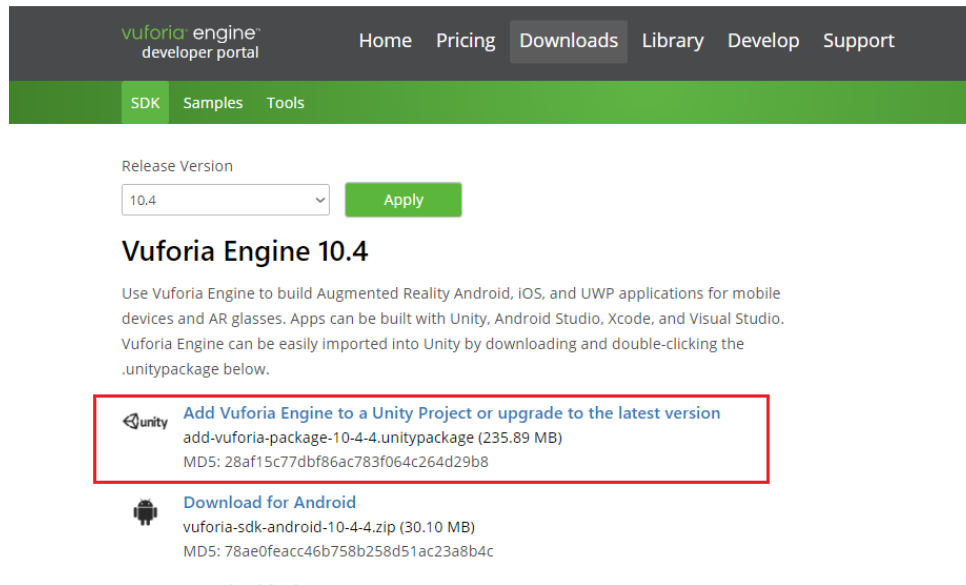
Ahora podremos ingresar la cuenta creada, con el correo y contraseña



The login form is contained within a green rectangular box. At the top, the word "Login" is centered in white. Below it, a white text box contains the instruction: "Login with your Vuforia developer account to download software, get license keys and participate in the Vuforia community." Below this box are two input fields: "Email Address" and "Password". Under the "Email Address" field, there is a link that says "Not Registered? Create account now". Under the "Password" field, there is a link that says "Forgot password?".

Figura 5. Interfaz de Login

Una vez ingresado a la cuenta nos dirigimos a la pestaña de Downloads y en SDK y descargaremos el paquete de vuforia para Unity el cual se ve en la figura 6.



The screenshot shows the "Downloads" section of the Vuforia Engine developer portal. At the top, there is a navigation bar with "vuforia engine developer portal" on the left and "Home Pricing Downloads Library Develop Support" on the right. Below this is a sub-navigation bar with "SDK Samples Tools". A "Release Version" dropdown menu is set to "10.4" with an "Apply" button. The main heading is "Vuforia Engine 10.4". Below this, there is a paragraph of text explaining the engine's use. A red box highlights the "Add Vuforia Engine to a Unity Project or upgrade to the latest version" link, which points to a .unitypackage file (235.89 MB) with MD5: 28af15c77dbf86ac783f064c264d29b8. Below this, there is a "Download for Android" link pointing to a zip file (30.10 MB) with MD5: 78ae0feacc46b758b258d51ac23a8b4c.

Figura 6. Descarga del paquete de vuforia para Unity

Una vez descargado el SDK, crearemos un nuevo proyecto en Unity llamado Ejemplo Vuforia, como se muestra en la figura 7.

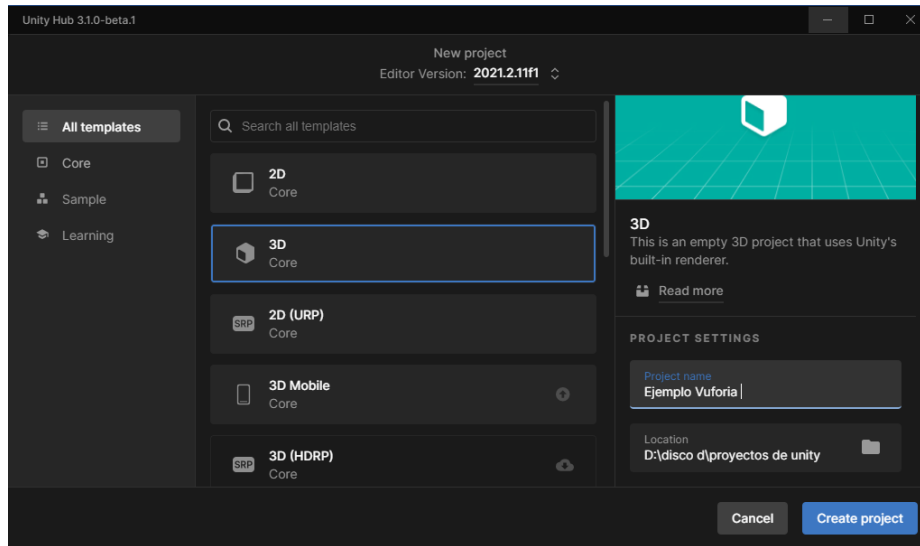


Figura 7. Creación de nuevo proyecto en Unity

Ya creado el proyecto se cambia de plataforma de desarrollo, como se ve en la figura 8, se va a la pestaña File y luego a Build Settings.

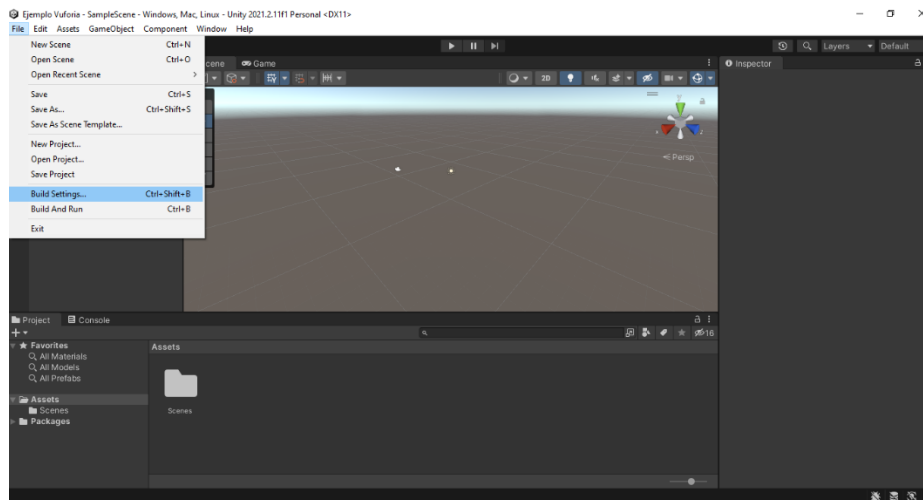


Figura 8. Selección de Builds Settings

Ahora en la ventana de Build settings, en este caso se cambia de plataforma a Android, seleccionando la plataforma y luego dando en el botón switch Plataforma como se ve en la figura 9.

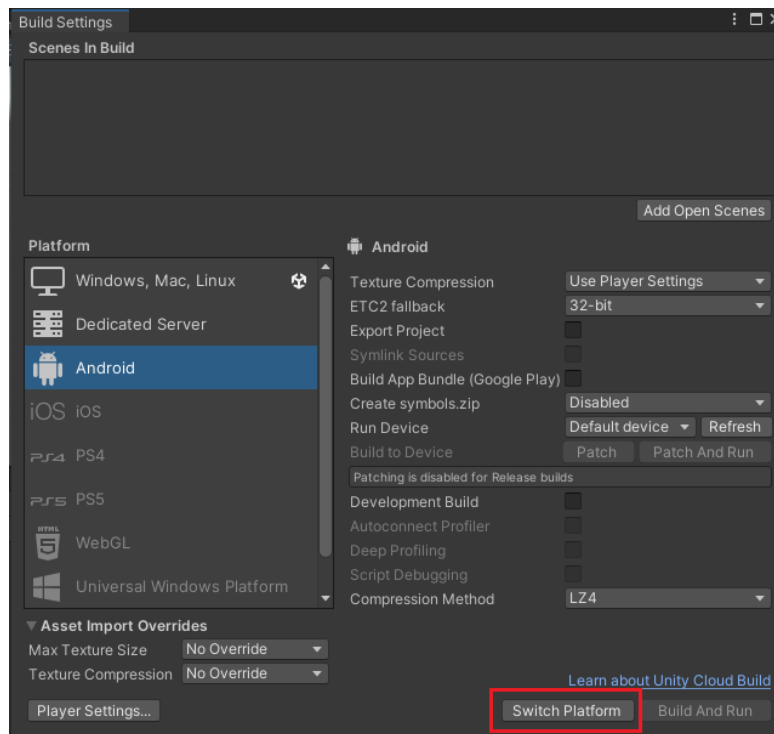


Figura 9. Cambio de plataforma

Luego se debe agregar el paquete de Vuforia para Unity descargado anteriormente, como se ve en la figura 10, nos dirigimos a la pestaña de Assets, luego a Import Package y luego a Custom Package.

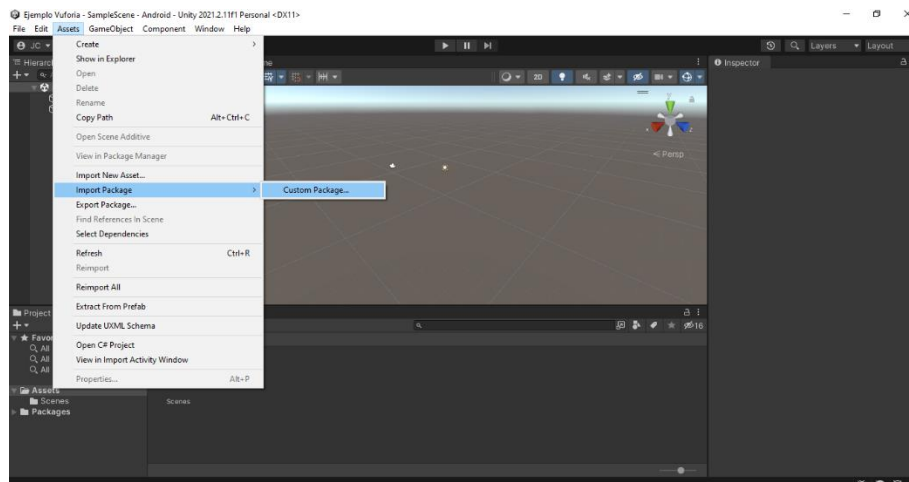


Figura 10. Ruta para agregar paquetes a Vuforia

Se abre una ventana como la de la figura 11, en donde nos dirigiremos a la ruta donde se guardo el SDK de vuforia lo seleccionaremos y le daremos en abrir.

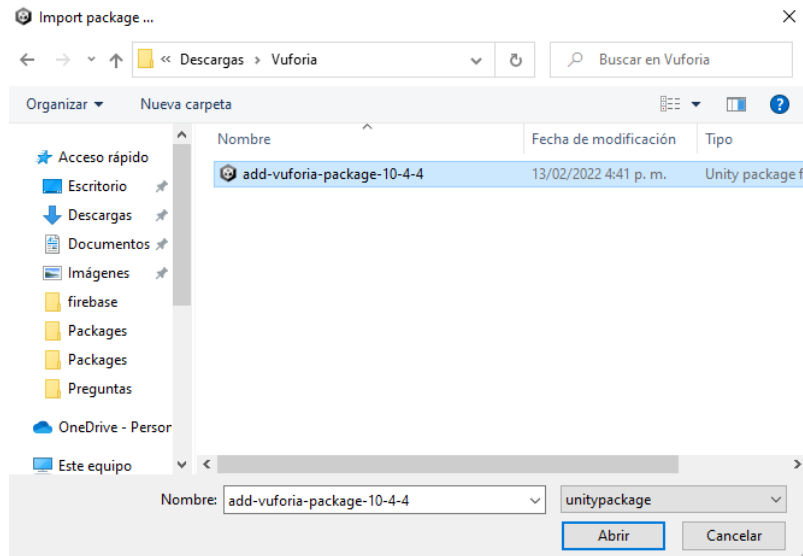


Figura 11. Abrir SDK de Vuforia

Una vez hecho esto, saldrá la ventana de la figura 12, donde aparecerán los archivos a importar y solo debemos dar en el botón import para que empiece a agregar los archivos.

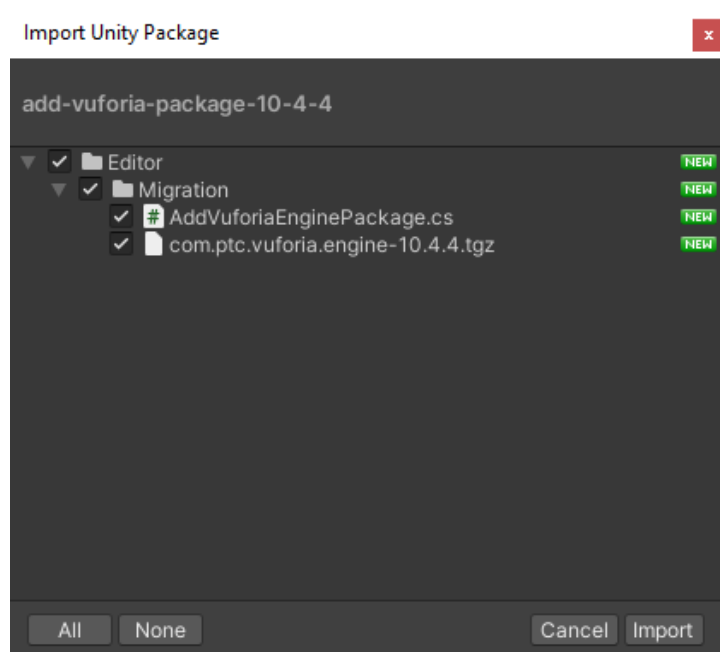


Figura 12. Importar archivos del SDK de vuforia

Cuando le demos importar, como el de la figura 12, nos saldrá un mensaje para agregar Vuforia Engine Package y le daremos en el botón Update.

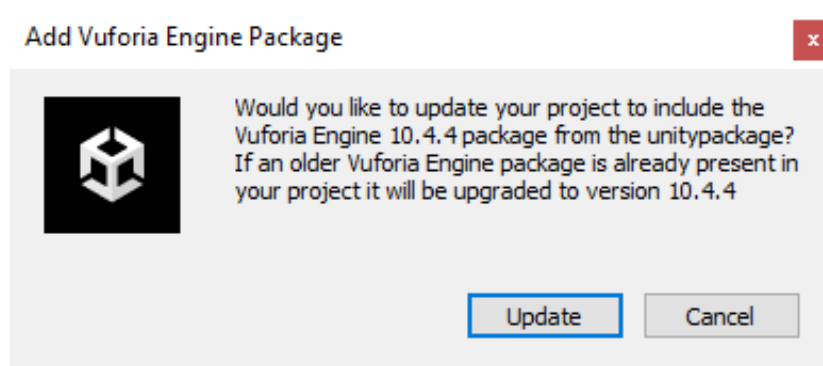


Figura 13. Mensaje para añadir Vuforia Engine Package

Una vez Agregado el SDK de vuforia ya aparecerán las opciones y herramientas de vuforia en el proyecto de Unity, como se muestra en la figura 14.

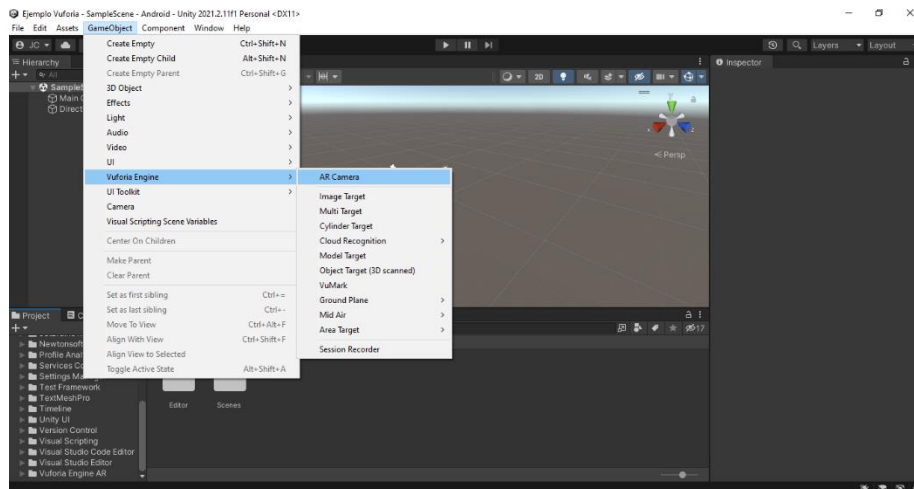


Figura 14. Herramientas de Vuforia Engine

Ahora se deberá agregar la licencia de Vuforia, vamos a la pagina oficial de vuforia y nos dirigimos a la pestaña de Develop y luego a License Manager (Figura 15.)

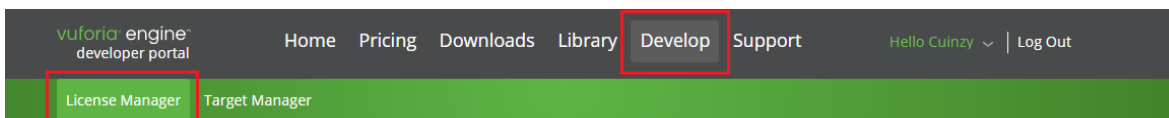


Figura 15. Pestaña de Develop

Ahora crearemos una licencia básica para nuestros proyectos dándole al botón Get Basic (Figura 16).

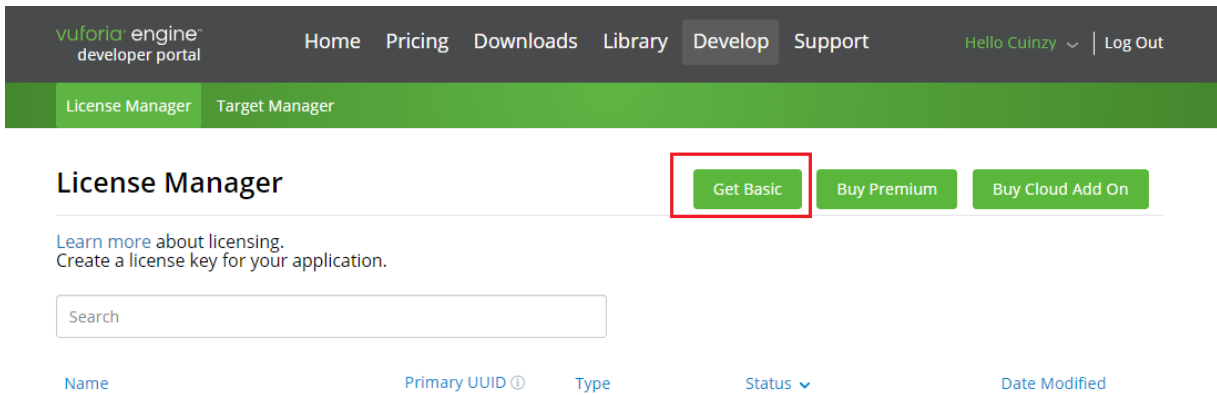


Figura 16. Crear licencia básica

Añadimos un nombre a la licencia, aceptamos los términos y condiciones y le damos en confirmar, como se ve en la figura 17.

Add a license key to your Basic plan

License Name *
Licencia Vuforia

You can change this later

License Key

Basic

Price: No Charge

Reco Usage: 1,000 per month

Cloud Targets: 1,000

VuMark Templates: 1,000 Active

VuMarks: Unlimited

By checking this box, I acknowledge that this license key is subject to the terms and conditions of the [Vuforia Developer Agreement](#).

Cancel

Confirm

Figura 17. Nombre de licencia y confirmación

En la figura 18, se ve la licencia ya creada en la cuenta e ingresamos en ella.

License Manager Get Basic Buy Premium Buy Cloud Add On

[Learn more](#) about licensing.
Create a license key for your application.

Name	Primary UUID ^①	Type	Status [▼]	Date Modified
Licencia Vuforia	N/A	Basic	Active	Feb 15, 2022

Figura 18. Licencia creada

Copiamos la llave de licencia, la cual necesitamos para nuestro proyecto de Unity, la llave de licencia es la mostrada en la figura 19.

Licencia Vuforia [Edit Name](#) [Delete License Key](#)

[License Key](#) [Usage](#)

Please copy the license key below into your app

```
AXkFCVz/////AAAEmdoVU211mUz0jTSTRYWKb0pPbmYADxM6czLnCv51g14+e57k9wNocXIJSMyYXChCBIM/coV/AILLzthX1Y+9A  
s56hUPg+V0wG3eR7nc1NbKxnHWJyhouzQgtbArsutkauMYkQFo560QcbWPrVpeF4nNupFIul8SyOcPbOnv2W1vd/Ve1W3QmfU4Y89c  
G/5BOux2A2Peeg0bLk1PdJpL2NgfHq9NQpw3eezYES78oC6AYVwH1x3mNZx7L6QdQLZqW64pkPMPp2vDE+tShIfXXUBs00MNX16qrtq  
Q99BUSZT6eXqIdSOKfBhSapFfeGzPgGhu7CxRTto/cH5L2HBthOPZoh1U94m8xm1Wf0bn/RT
```

Plan Type: Basic
Status: Active
Created: Feb 15, 2022 12:35
License UUID: 9324bb50998345239403f48365f462c7

Permissions:

- Advanced Camera
- External Camera
- Model & Area Targets
- Watermark

History:
License Created - Today 12:35

Figura 19. Llave de licencia.

Nos dirigimos nuevamente al proyecto de Unity, ahora iremos a la pestaña de Windows, luego a la opción con el nombre de Vuforia Configuration, se nos abrirá en el lado derecho una ventana con las opciones de configuración de vuforia, vamos al apartado de App License Kay y pegamos la llave de licencia que obtuvimos de la página de Vuforia (figura 20), una vez agregada la licencia a nuestro proyecto podemos empezar a utilizar las diferentes herramientas que brinda el SDK de vuforia para Unity

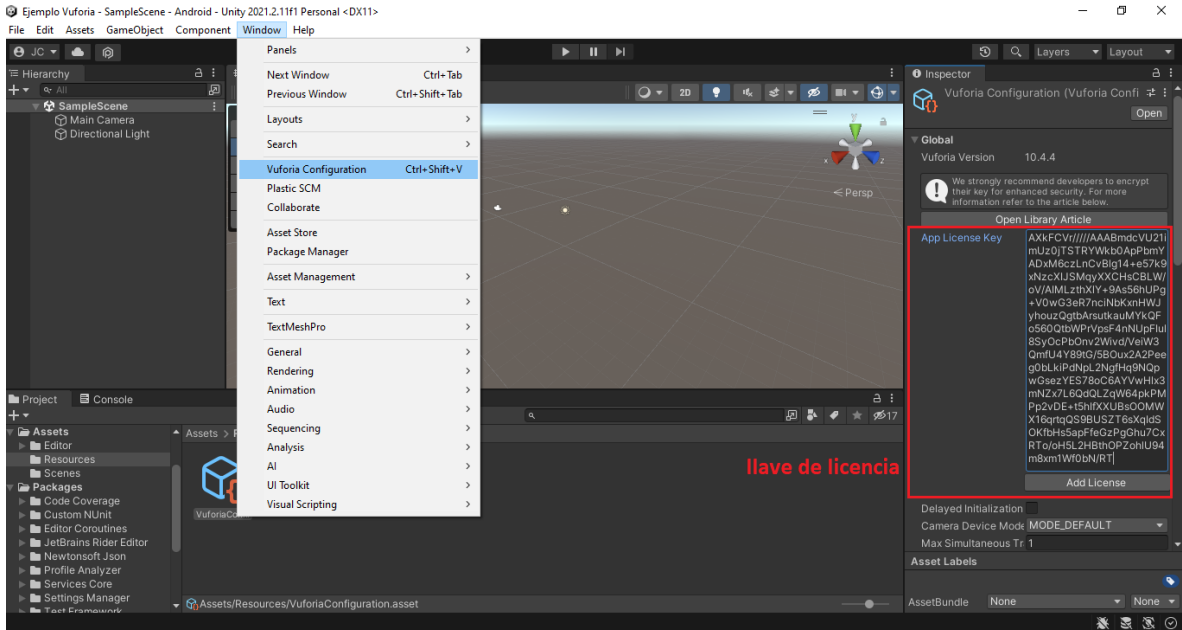


Figura 20. Agregar llave de licencia a Unity

ANEXO D

Lógica principal modulo de configuración

Interfaz inicial

La interfaz inicial como muestra la figura 1, tiene dos cuadros de ingreso de texto de tipo *"TextMeshPro - Input field"* esta clase de campos de ingreso de texto se usan de manera amplia en las demás interfaces del modulo, ya que permiten llenar el texto de manera manual o llenar el texto por medio de código vía script, además de que cuenta con la característica de convertirlo en un objeto no interactuable, así se puede configurar para no dejar que sea llenado manualmente; con esta función se guía al usuario en la navegación del modulo haciéndolo secuencialmente. Por otro lado también se cuenta con cuatro botones de tipo *"Button"* los cuales se les puede asignar una acción preestablecida como activar o desactivar una interfaz o una acción más compleja por medio de código en el script asociado a este.



Figura 1: Interfaz inicial

El primer campo de texto es para ingresar el correo y el segundo la contraseña asociado a este, dando por sentado que ya se realizó antes el registro en la herramienta. El botón de *ACCEDER* es el que permite validar esta información dentro del modulo, esta validación se hace con respecto a la base de datos que es la que almacena los datos del terapeuta, del paciente y del ejercicio. El código asociado se encuentra dentro del script *FirestoreManager.cs* que actúa como un main o punto de arranque para todo el modulo.

Dentro de este script la función publica de *LoginButton()* inicia la rutina de autenticación con firebase llamada *Login* tomando como datos de entrada los que se extraen de los campos de texto de correo y contraseña que previamente fueron asignados al GameObject que contiene el script main, para este modulo el GameObject se nombro *FirestoreManager*, igual que el script.


```

//Funcion para el boton de login
public void LoginButton()
{
    //Llama la corrutina de login para verificar si el correo y la
    //contrasena son correctas
    StartCoroutine(Login(emailLoginField.text , passwordLoginField.text))
    ;
}

```

```

//corrutina de Login
private IEnumerator Login(string _email , string _password)
{
    //Llama la autentificacion de firebase
    var LoginTask = auth.SignInWithEmailAndPasswordAsync(_email ,
    _password);
    //Espera a que las tareas esten completas
    yield return new WaitUntil(predicate: () => LoginTask.IsCompleted);
    //Una vez la tarea es completada se verifica
    if (LoginTask.Exception != null)
    {
        //La informacion fue incorrecta y se indica
        //cual de los campos fue incorrecto
    }
    else
    {
        //La informacion fue correcta y se concede el acceso
    }
}

```

El botón de olvide mi contraseña funciona de la misma manera que el botón de login pero utilizando otra función dentro del mismo script main que ya se mencionó llamado *FirestoreManager.cs*, la función pública que permite restablecer la contraseña se llama *RestablecerButton()* la cual extrae el correo del campo de texto para usarlo en una función privada dentro del main llamada *RestablecerContraseña* la cual con el correo extraído del campo de texto correo, permite enviar al mismo un enlace que genera en una pestaña del navegador un pequeño formulario en donde podrá asignar una nueva contraseña para ese correo electrónico y así no perder la información que ya se encuentra almacenada. Una vez puesta la nueva contraseña ya podrá acceder al módulo con la nueva.

```

//Funcion publica para el boton de olvide mi contrase a
public void RestablecerButton(){
    RestablecerContrase a(emailLoginField.text);
}

```

```

//Funcion privada que usando el correo (string) extraido env a el
//link del formulario para asignar una nueva contrase a al correo
private void RestablecerContrase a(string email){
    //auth.SendPasswordResetEmailAsync() permite el restablecimiento
    auth.SendPasswordResetEmailAsync(email).ContinueWith(task => {

```

```

        if (task.IsCanceled){
            Debug.LogError("El restablecimiento fue cancelado");
            return;
        }
        if (task.IsFaulted){
            Debug.LogError("El restablecimiento tuvo un error");
            return;
        }

        Debug.LogError("Restablecimiento completado");

    });
}

```

El botón de registro, permite acceder a la otra interfaz por medio del script *UiManager.cs* en donde se añaden los 3 canvas contenedores de las interfaces, uno contiene el de inicio, otro contiene el de registro y el ultimo contiene todas las interfaces relacionadas con la configuración una vez se ha autenticado. En este caso el botón de registro, usa la función *RegisterScreen()* la cual permite ir al canvas con la interfaz de registro.

```

//Funcion para activar la interfaz de registro y desactivar las otras
public void RegisterScreen ()
{
    //Desactiva los canvas
    ClearScreen ();
    //Activa el de registro
    registerUI.SetActive(true);
}

```

Por ultimo el botón con la equis blanca encerrado en el círculo rojo, es un script asociado a la cámara virtual que muestra las interfaces, este script es *ControlSalida.cs* que simplemente ejecuta la única función *exit()* la cual es una instrucción de salir de la aplicación cerrándola en su totalidad.

```

//Cierra la aplicacion
public void Exit ()
{
    Application.Quit ();
    Debug.Log("Se ha salido del juego");
}

```

Interfaz de registro

La interfaz de registro es un formulario con dos botones como muestra la figura 2 dado el caso que alguna de esta información suministrada en los campos de texto de tipo *TextMesh-Pro - input field* no sea valida emergerá un pequeño aviso arriba del botón *REGISTRAR* con letras rojas indicando cual es la falla.

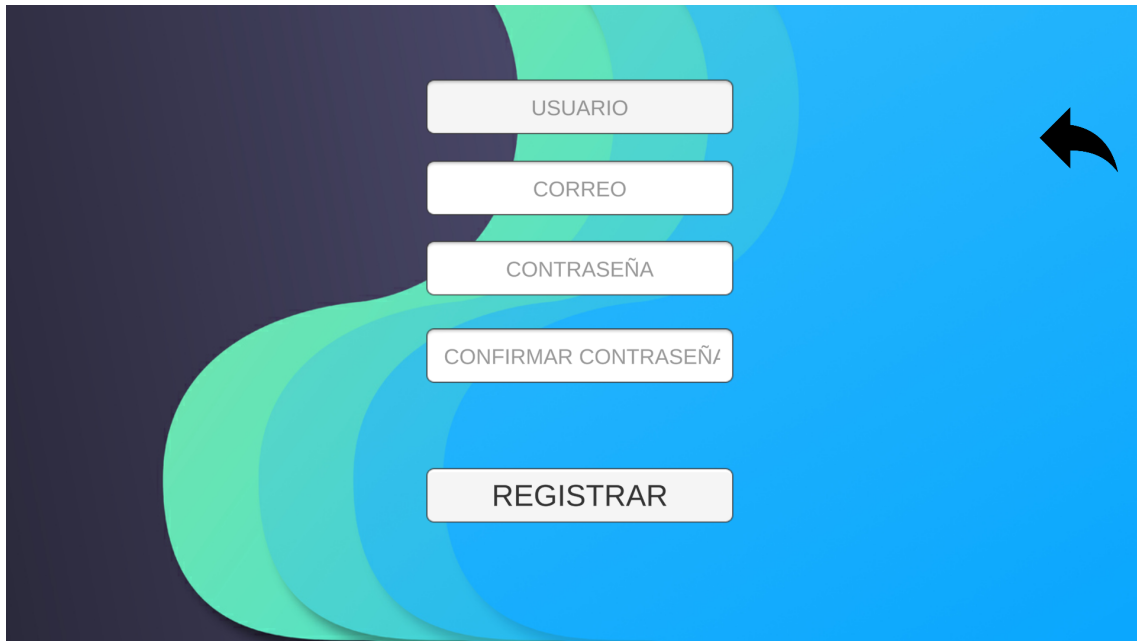


Figura 2: Interfaz de registro

El código asociado al botón de registrar se encuentra dentro del script main que es *FirebaseManager.cs*, la funcion encargada de llamar la corrutina es *RegisterButton()* quien llama la corrutina *Register* que usando los datos obtenidos de los 4 campo de textos hace la validación o como se menciono anteriormente imprimirá un aviso en pantalla de cual dato es incorrecto.

```
//Funcion que llama la corrutina de registro pasandole correo, contrase  
y username  
public void RegisterButton()  
{  
    StartCoroutine(Register(emailRegisterField.text,  
        passwordRegisterField.text, usernameRegisterField.text));  
}
```

```
//Corrutina de registro que recibe correo, contrase a y nombre de  
usuario  
private IEnumerator Register(string _email, string _password, string  
_username)  
{  
    ...  
  
    else
```

```

    {
        //En esta seccion firebase permite la creacion del usuario con
        //los datos entregados
        var RegisterTask = auth.CreateUserWithEmailAndPasswordAsync(
            _email, _password);

        yield return new WaitUntil(predicate: () => RegisterTask.
            IsCompleted);

        // Si el registro falla se informa cual es
        if (RegisterTask.Exception != null)
        {
            ...
        }
        // Si el registro es correcto, se almacena en la base de datos
        // y se regresa a la pantalla de inicio
        else
        {
            ...
        }
    }
}

```

El botón volver usando el *UiManager.cs* tiene la función de *LoginScreen* que cierra la interfaz de registro y presenta de nuevo la interfaz de inicio.

```

public void LoginScreen() //Boton de volver
{
    ClearScreen();
    loginUI.SetActive(true);
}

```

Interfaz de bienvenida

La interfaz de bienvenida como se muestra en la figura ?? posee un botón de salir igual al de la interfaz inicial y un botón llamado *IR A PACIENTES* el cual es encargado de cargar los datos directamente inferiores a la jerarquía del terapeuta en la base de datos, estos datos son el numero de pacientes, y sus información (usuario, nombre completo y documento).

La función asociada al botón *IR A PACIENTES* se encuentra en el mismo main de *FirestoreManager.cs*, la función mencionada se nombro como *DobleMostrar()* que ejecuta otras dos funciones y una corrutina, la primera función es la encargada de arrancar una corrutina que extrae los datos de la base de datos, la otra crea los botones para que posteriormente sean llenados con los datos de cada paciente y por ultimo una corrutina de espera para que los datos se carguen correctamente.

```

public void DobleMostrar()
{
    //Funcion que ejecuta la corrutina que extrae los datos de los
    //pacientes
    MostrarUsuarios();
}

```

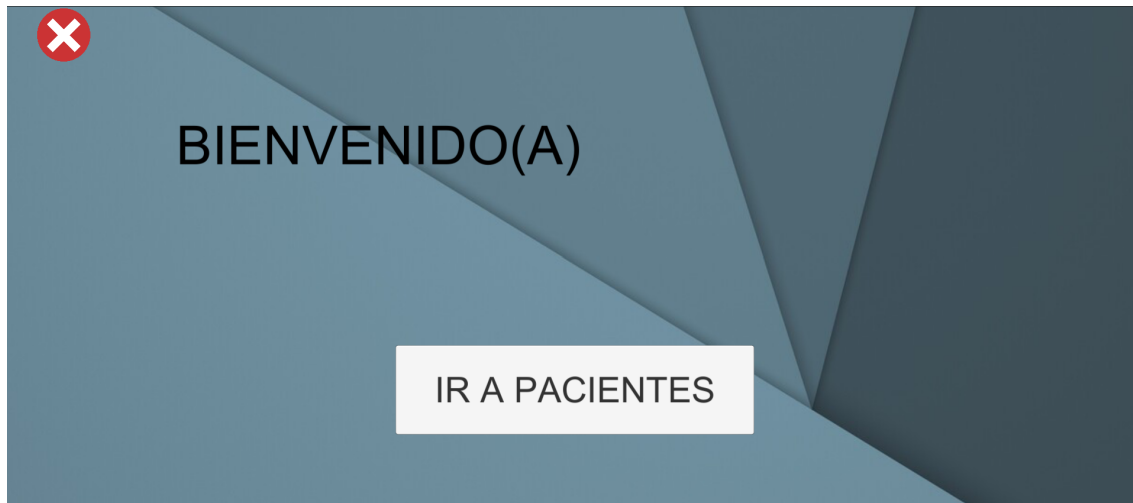


Figura 3: Interfaz de bienvenida

```
//Funcion que efectua la creacion de botones en la interfaz de  
//pacientes  
establecerNum2 ();  
  
//Funcion que efectua la corrutina de espera para que  
//los datos se carguen bien en la interfaz  
StartCoroutine (Esperar ());  
}
```

Interfaz de pacientes

La lógica aplicada en esta interfaz de pacientes vista en la figura 4 depende primero de la carga de datos hecha en la interfaz de bienvenida, esta permite el conteo de los pacientes para dependiendo esta cantidad generar botones en el lado de visualización y a cada uno de estos asignarle el nombre para que después al presionarlo en el lado de gestión muestre la información de ese paciente, la función asociada al botón del paciente es *editarInteraccion* dentro del script *FirestoreManager.cs* la cual habilita los campos de texto y el botón .OK.^{el} cual valida los cambios en la base de datos.

```
// Funcion asociada al boton de editar  
public void editarInteraccion () {  
    nomUsuario.interactable = true;  
    docUsuario.interactable = true;  
    ok.interactable = true;  
}
```

El botón de OK es el encargado de validar la información, el tiene la función de *ActualizarUSusario* el cual inicia dos corrutinas una que actualiza el nombre del paciente, otra que actualiza su documento para acto seguido desactive los campos de texto y actualice en el lado de visualización.



Figura 4: Interfaz de pacientes

```

public void ActualizarUsuario () {

    //Corrutina de actualizacion de nombre
    StartCoroutine(ActualizarNombre());

    //Corrutina de actualizacion de documento
    StartCoroutine(ActualizarDocumento());

    //Hacer no interactuable los campos de texto y el boton de OK
    nomUsuario.interactable = false;
    docUsuario.interactable = false;
    ok.interactable = false;

    //Actualizacion del lado de visualizacion
    StartCoroutine(lapso());

}

```

El botón de eliminar ejecuta la función *EliminarUsuario* dentro del mismo script el cual actualiza los botones de pacientes con uno menos, reduciendo también el numero de pacientes en el indicador y ejecutando la corrutina que elimina los datos del paciente de la base de datos.

```

public void EliminarUsuario () {

    // Actualiza los botones de paciente
    controlScroll.refresh();
    // Elimina el paciente de la base de datos
    StartCoroutine(BorrarEsteUser());

}

```

```

// Limpia los datos de la interfaz con los nuevos valores
StartCoroutine(lapso3());
}

```

El botón de ir cuando ya se ha seleccionado un paciente (Que se vean sus datos en el lado de gestion) lo valida y lo lleva a la interfaz de ejercicio donde podrá crear y configurar los ejercicios. La función asociada a este botón solo cambia la interfaz y extrae el nombre para ponerlo en el campo de texto de la interfaz siguiente de ejercicio.

```

public void IrAPacienteEstante() {

//Setea el nombre en el campo de texto
nomUsuarioEstante.text = nomUsuario.text;
NombreEjercicioActual.text = "";

//Cambia la interfaz
UsuarioScreen.SetActive(false);
EjercicioInicio.SetActive(true);
Saber.SetActive(true);
}

```

El botón de + como se menciona, lleva a la interfaz de registro, limpia los botones a la espera del nuevo paciente que se vaya a generar. En dado caso que no se cree, seguirá contando con la misma cantidad y se volverá a actualizar automáticamente.

Interfaz de registro de pacientes

La interfaz de registro de pacientes como se ve en la figura 5 es un formulario con 3 campos de texto y dos botones, verificar, guardar y volver. La función *ComprobarExistenciaPaciente* asociada al botón de *VERIFICAR* se encuentra en el script main, la cual ejecuta la corrutina que comprueba el texto en el campo de usuario con los usuarios de otros pacientes ya registrados en el modulo, eso incluye los pacientes de otros terapeutas.

```

//Ejecuta la corrutina de verificaci n
public void ComprobarExistenciaPaciente() {
StartCoroutine(ExtractorTerap());
}

```

El botón *GUARDAR* ejecuta las funciones *GuardarNuevoUsuario* y *CargandoCreacionUsuario*, la primera es la que ejecuta la corrutina de actualización en la base de datos de un nuevo paciente y sus datos, la segunda simplemente genera un aviso en pantalla de que esta cargando los datos por lo que hay que esperar hasta que vuelva a la interfaz de paciente de manera automática.

```

// Ejecuta la corrutina que registra el nuevo paciente
public void GuardarNuevoUsuario() {
StartCoroutine(Esperar2());
}

```

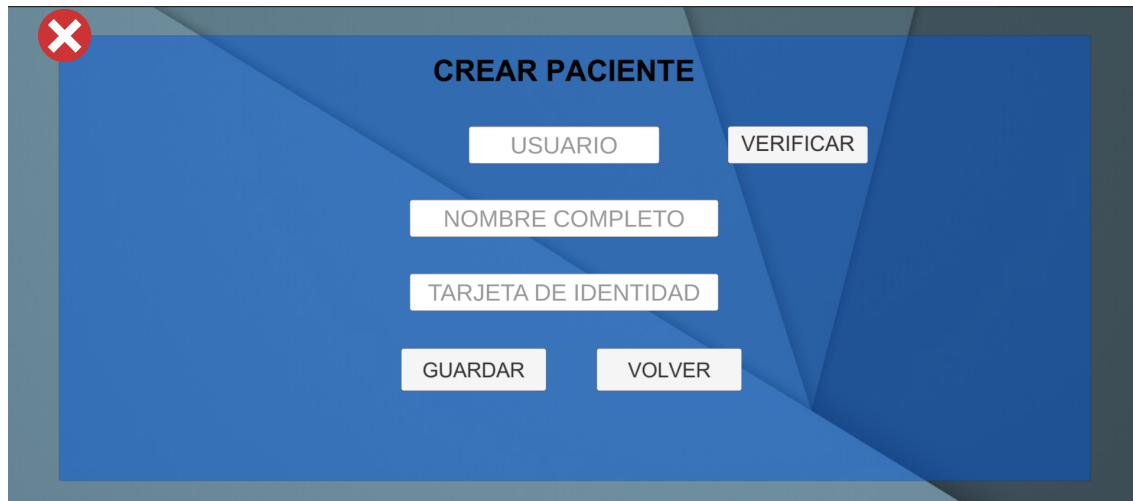


Figura 5: Interfaz de registro de pacientes

```
// Muestra en pantalla un aviso de espera
public void CargandoCreacionUsuario () {
    Cargando.text = "Cargando , _por _favor _espere _un _momento .";
    aviso.text = "";
    aviso2.text = "";
}
```

Por ultimo el botón volver hace cambio de escena y actualiza la información con los que se encuentran en la base de datos.

Interfaz de ejercicio

La interfaz de ejercicio se ve en la figura 6, presenta 4 botones y dos campos de texto no interactivos los cuales muestran el paciente actual y el ejercicio actual a configurar. El botón de *VER EJERCICIOS* lleva a la interfaz de visualización de ejercicios, el botón de *CREAR EJERCICIO* lleva a la interfaz de crear ejercicio. Por ultimo cuando ya se tenga en la parte superior en los campos de texto no interactivos el nombre del paciente y el nombre del ejercicio que se va a configurar podrá presionar el botón de *CONFIGURAR* el cual lleva a la interfaz de selección de configuración.

El botón *CREAR EJERCICIO* tiene asociado una función simple de cambio de interfaz en el main scrpit, esta función es *IrACrearEjercicio*, la cual desactiva la interfaz de ejercicio y activando la de creación de ejercicios.

```
// Cambio de interfaz
public void IrACrearEjercicio () {
    EjercicioInicio.SetActive(false);
    CrearEjercicioScreen.SetActive(true);
}
```

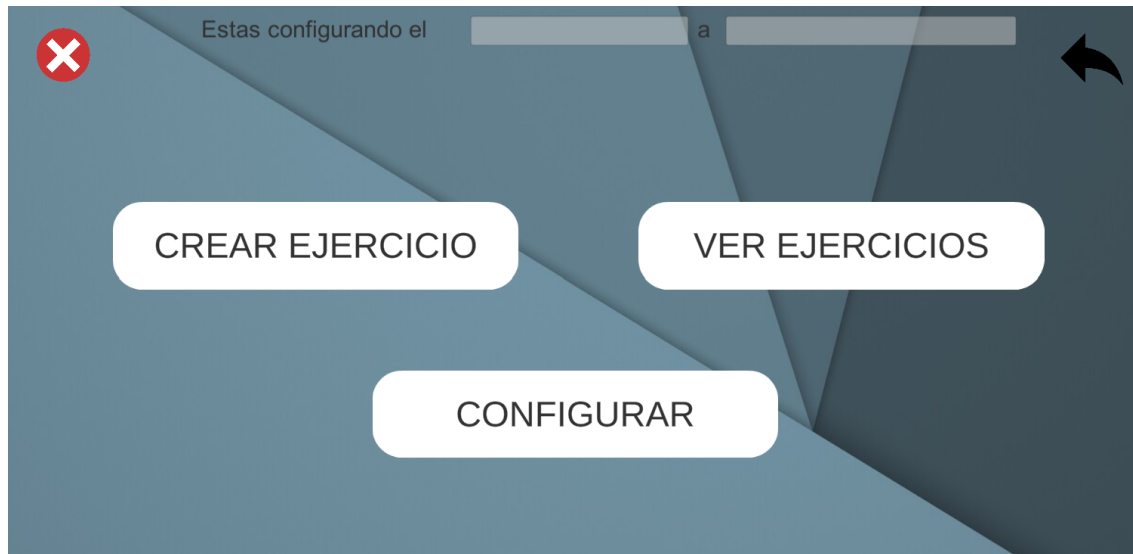



Figura 6: Interfaz de ejercicio

El botón de *VER EJERCICIOS* lleva a la interfaz de visualización de ejercicios, con la función *OprimirVerEjercicio* en el mismo script main el cual cambia la interfaz y además ejecuta una corrutina que carga los datos de los ejercicios para mostrar.

```
public void OprimirVerEjercicios () {
    StartCoroutine (MostarEjercicios ());
    VerEjercicio .SetActive (true);
    EjercicioInicio .SetActive (false);
}
```

El botón de *CONFIGURAR* una vez se tenga en la parte superior los datos de nombre y de ejercicio, permite extraer la información de la base de datos que tenga el ejercicio para presentarla en las interfaces de reto y configuración del estante en ese ejercicio, acto seguido cambia la interfaz cerrando la de ejercicio.

```
public void IrAConfig () {
    StartCoroutine (EsperarPaciente ());
}
```

```
private IEnumerator EsperarPaciente () {
    StartCoroutine (LoadPacienteData ());
    yield return new WaitForSeconds (5);
    seccionBotonesEstante .SetActive (true);
    EjercicioInicio .SetActive (false);
    Saber .SetActive (true);
}
```

Interfaz de creación de ejercicios

La interfaz de creación de ejercicio como muestra la figura 7 permite crear un ejercicio estableciendo como primera instancia el nombre de este y una fecha de inicio y de fin que debe llevar un formato de tipo DD/MM/AAAA. Con un único botón llamado *CREAR* que valida y carga la información en la base de datos.



Figura 7: Interfaz de creación de ejercicio

El botón crear tiene la función de *CrearEjercicioBoton* en el mismo main, la cual ejecuta dos corrutinas que se encargan de almacenar la información de la fecha de inicio y la fecha de fin del ejercicio dentro del mismo en la jerarquia. Esto quiere decir que primero se creara el ejercicio dentro de los datos del paciente y los datos de fecha de inicio y fecha de fin dentro del ejercicio.

```
public void CrearEjercicioBoton () {  
    //Si falta alg n dato por llenar no lo validara  
    ...  
    else {  
        StartCoroutine ( CrearEjercicioInicio ( NombreDelEjercicio . text ,  
            FechaInicioEjercicio . text ) );  
        StartCoroutine ( CrearEjercicioFinal ( NombreDelEjercicio . text ,  
            FechaFinEjercicio . text ) );  
        StartCoroutine ( esperarEjer ( ) );  
    }  
}
```

Al validarse la información pasara automáticamente a la interfaz de ejercicio para poder dirigirse a *VER EJERCICIOS* si se quiere modificar o eliminar un ejercicio o ir *CONFIGURAR* para asignarle una configuración de objetivo y ubicación de objetos en el estante del minisuper.

Interfaz de visualización de ejercicios

La interfaz de ver ejercicios como indica la figura 8 permite cargar todos los ejercicios que se encuentren hechos para este paciente.

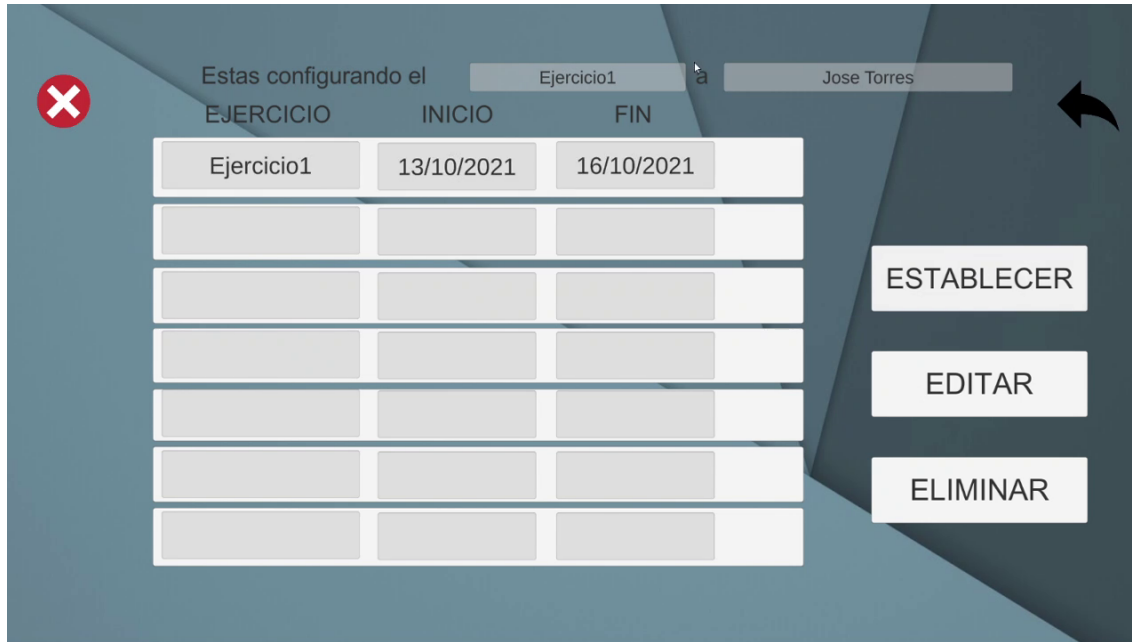


Figura 8: Interfaz de visualización de ejercicios

Se debe presionar la barra blanca que contiene el nombre del ejercicio y sus fechas para poder usar los botones de la derecha. El botón *ESTABLECER* tiene asignada la función *OprimirSeleccionarEjer* y *LimpiarEjercicios* del main, la primera función cambia la interfaz a la de ejercicio y establece el ejercicio en la parte superior de la pantalla para configurarlo la segunda actualiza los ejercicios, el botón *EDITAR* tiene asignada la función de *OprimirEditarEjer* y también la de *LimpiarEjercicios*, la primera es la que se encarga de ejecutar las corrutinas que cambian los datos de las fechas del ejercicio en la base de datos; por ultimo el de eliminar tiene asignada la función de *OprimirEliminarEjer* la cual ejecuta la corrutina de eliminación en la base de datos y cambia la interfaz a la de ejercicio.

```

// Asigna el ejercicio y cambia la interfaz
public void OprimirSeleccionarEjer() {
    NombreEjercicioActual.text = NombresEjer[ejerciciosControl.dato].
        text;
    VerEjercicio.SetActive(false);
    EjercicioInicio.SetActive(true);
}

```

```

//Edita los datos de fecha del ejercicio y cambia de interfaz
public void OprimirEditarEjer() {
    StartCoroutine(EditarEjercicioI(NombresEjer[ejerciciosControl.dato].
        text, FechasIniEjer[ejerciciosControl.dato].text));
    StartCoroutine(EditarEjercicioF(NombresEjer[ejerciciosControl.dato].
        text, FechasFiniEjer[ejerciciosControl.dato].text));
    VerEjercicio.SetActive(false);
    EjercicioInicio.SetActive(true);
}

```

```

//Elimina el ejercicio y cambia de interfaz
public void OprimirEliminarEjer() {
    StartCoroutine(BorrarEjercicio(NombresEjer[ejerciciosControl.dato].
        text));
    VerEjercicio.SetActive(false);
    EjercicioInicio.SetActive(true);
}

```

Interfaz de selección de configuración

La interfaz de selección de configuración vista en ?? cuenta con dos botones uno que es *IR A ESTANTE* en donde se cambia la interfaz a ala de selección de lado de estante y el otro *CONFIGURAR RETO* la cual lleva a la interfaz de configuración de reto.

Los dos botones forman parte de la logica del script *ManagerEstante.cs* el cual funciona como el *UIManager.cs* en las interfaces principales solo que con las correspondientes a la configuración del estante y del reto. El código de cada uno de estos dos botones se encuentran en el Start del script.



Figura 14: Interfaz de seleccion de configuracion

```

void Start ()
    {
        // Ir al estante
        irBotonesEstante.GetComponent<Button>().onClick.AddListener(()=>{
            ScreenBotonesInicio.SetActive(false);ScreenBotones.SetActive(
                true);
        });
        ...
        // Ir a a configurar el reto
        Reto.GetComponent<Button>().onClick.AddListener(()=>{
            ScreenBotonesInicio.SetActive(false);ScreenReto.SetActive(true)
                ;});
        ...
    }

```

Interfaz de configuración de reto

La interfaz de configuración de reto permite establecer el objetivo para el ejercicio, en la figura 15 se ve la interfaz y a manera de ejemplo se realiza la configuración del objetivo del Ejercicio1, se presionara el objeto de todos los disponibles en el recuadro izquierdo, seguido de la cantidad en el recuadro derecho siendo un máximo de 12, se puede confirmar viendo en los campos de texto no interactivables el objeto y la cantidad; acto seguido debera presionar el botón de guardar para cargar esta informacion a la base de datos, una vez hecho podrá ver si se realizo el guardado al entrar a la interfaz de ver reto con el botón de *VER RETOS*, se puede colocar un máximo de objetivos de 15 y cada uno con un máximo de búsqueda de 12 pero esta cantidad seria superior a los espacios que ofrece el estante de objetos.



Figura 15: Interfaz de configuración de reto

El botón guardar tiene asociada la función *SaveDataReto*, la cual ejecuta una corrutina que crea el reto dentro del ejercicio que tiene el paciente. Actualizándose cada que el terapeuta cree un nuevo reto y oprima el botón. En dado caso de que sea un mismo objeto, este cambiara la cantidad por la nueva guardada.

```

\\Actualiza el reto del ejercicio en la base de datos
public void SaveDataReto() {
    StartCoroutine(UpdateReto());
}

```

El botón de ver retos, tiene la función *MostrarRetos* la cual ejecuta un cambio de interfaz y una corrutina que se encarga de cargar los retos y mostrarlos en la interfaz de visualización de reto.

```
// Cambio de interfaz y carga los datos de reto
public void MostrarRetos () {
    StartCoroutine (MostarElReto ());
    Reto.SetActive (false);
    VerReto.SetActive (true);
}
```

Interfaz de visualización de reto

La interfaz de visualización de reto permite borrar todos los retos presionando el botón *BORRAR* o simplemente visualizar los retos asociados a ese ejercicio. En este caso la figura 16 muestra que en el ejercicio1 se deben buscar 2 bananas y 4 uvasVariantes



Figura 16: Interfaz visualización de reto

El botón borrar tiene la función de *DelateReto* la cual ejecuta una corrutina que borra los retos del ejercicio en la base de datos y actualiza los que se ven en la interfaz dejando ninguno y volviendo a la interfaz de configuración de reto.

```
public void DelateReto () {
    StartCoroutine (BorrarReto ());
    LimpiarRetos ();
    Reto.SetActive (true);
    VerReto.SetActive (false);
}
```

Interfaz de selección de lado de estante

El presionar el boton *IR AL ESTANTE* en la interfaz de selección de configuración, llevara a esta interfaz como se ve en la figura 17 la cual nos presenta los 4 lados configurables del estante. En cada uno de estos lados se puede seleccionar el objeto que tendrá cada espacio, el frontal y el posterior los cuales cuentan con un numero especifico de espacios y los laterales con menor, los 4 funcionando de la misma manera por lo que al presionar un lado llevara a la interfaz de configuración de estante adaptado para ese lado en especifico.

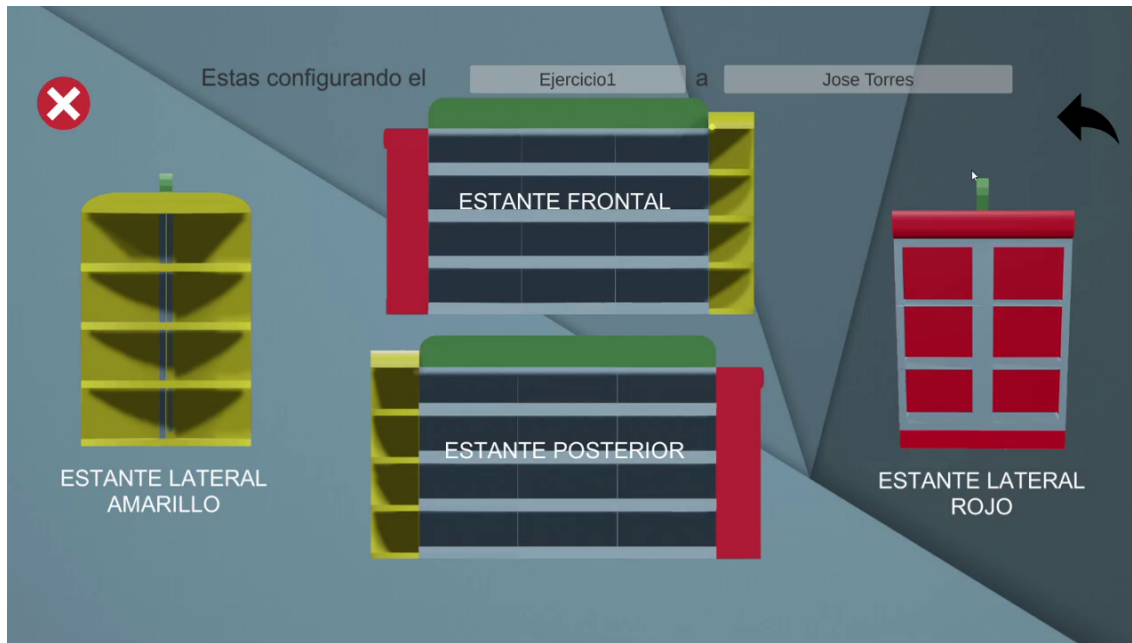


Figura 17: Interfaz selección de lado de estante

El script *ManagerEstante.cs* es el encargado de hacer que cada uno de estos botones con los lados lleve a la interfaz de configuración de estante adaptada para cada lado. La funcion especifica de este script que hace el cambio es *controlScreenBotones* y *cambiarPantalla(int x)* la cual recibe el numero del botón presionado que arroja *controlScreenBotones* y lleva a la interfaz.

```
//Cambio de interfaz
public void controlScreenBotones () {
    Frontal.GetComponent<Button>().onClick.AddListener(()=>
        cambiarPantalla(0));
    Posterior.GetComponent<Button>().onClick.AddListener(()=>
        cambiarPantalla(1));
    LateralAmarillo.GetComponent<Button>().onClick.AddListener(()=>
        cambiarPantalla(2));
    LateralRojo.GetComponent<Button>().onClick.AddListener(()=>
        cambiarPantalla(3));
    Back.GetComponent<Button>().onClick.AddListener(()=>cambiarPantalla
        (4));
}
```


Interfaz de configuración de estante

La interfaz de configuración de estante se presenta de la misma manera como se ve la figura 18, solamente que varia el estante a configurar en la mitad de la interfaz dependiendo cual se selecciono en la interfaz de selección de lado, cuenta con unos botones en cada espacio configurable del lado que este actualmente activo, en donde no hay objeto se presenta como una X pero si ya cuenta con un objeto este se mostrara en el espacio, al presionar un espacio con o sin objeto se desplegara un cuadro a la izquierda con los objetos los cuales pueden cambiar a sus formas variantes que tienen un cambio de color presionando el botón variante que solo estará habilitado cuando el cuadro de objetos lo este. El lado derecho permite configurar donde aparecerá el personaje extra a buscar y el tiempo de búsqueda para todo el ejercicio. Además cuenta con 4 botones más los cuales son guardar, cerrar sesión, botón mágico y uno con la forma del estante el cual permitirá volver a la interfaz de selección de lado.

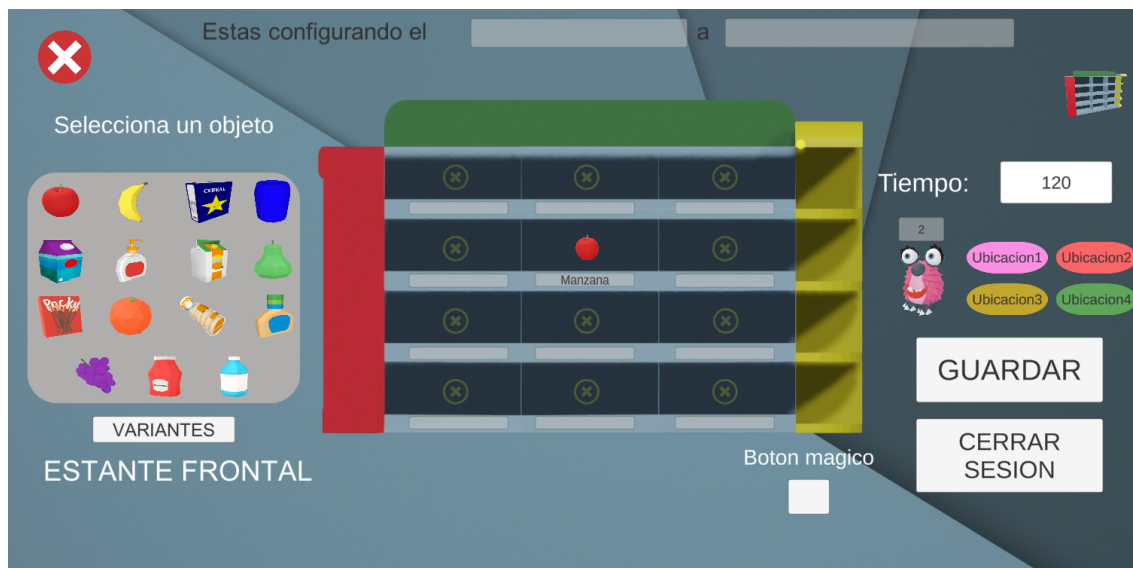


Figura 18: Interfaz configuración de estante

Cada lado cuenta con su propio script eso quiere decir que hay 4 los cuales son *Control-Frontal.cs*, *ControlPosterior.cs*, *ControlLateralRojo.cs*, *ControlLateralAmarillo.cs* los cuales permiten la visualización del lado del estante en la mitad de la interfaz y la interacción con los objetos que se presionan para que se almacenen en el espacio, también de que se encargan cada uno de su botón mágico respectivo el cual permite la múltiple selección de espacios para un llenado más rápido. El botón guardar al igual que el de reto con la función *SaveDataPaciente*, almacena todos los datos de ubicación de objetos y tiempo en el ejercicio del paciente. Cabe mencionar que el main *FirestoreManager.cs* es quien posee la función de guardar y cerrar sesión *SignOutButton*.

```

public void SaveDataPaciente () {

    StartCoroutine (UpdateSandIPaciente ("Tiempo" , int . Parse ( tiempoField .
        text )));
    StartCoroutine (UpdateSandIPaciente ("Monster" , int . Parse ( monsterField .
        text )));
    StartCoroutine (UpdateEstante (controlFrontal . entradasF [0] . text , "
        estanteFrontal1" ));
    StartCoroutine (UpdateEstante (controlFrontal . entradasF [1] . text , "
        estanteFrontal2" ));
    StartCoroutine (UpdateEstante (controlFrontal . entradasF [2] . text ,
        ...
    StartCoroutine (UpdateEstante (controlLateralRojo . entradasR [3] . text , "
        estantelateralRojo4" ));
    StartCoroutine (UpdateEstante (controlLateralRojo . entradasR [4] . text , "
        estantelateralRojo5" ));
    StartCoroutine (UpdateEstante (controlLateralRojo . entradasR [5] . text , "
        estantelateralRojo6" ));

}

```

```

public void SignOutButton ()
{
    auth . SignOut () ;
    UIManager . instance . LoginScreen () ;
    LimpiarBotones () ;
    nomUsuario . text = "" ;
    usUsuario . text = "" ;
    ...
    ClearRegisterFeilds () ;
    ClearLoginFeilds () ;
}

```

ANEXO E

Lógica principal modulo de juego serio

Inicio de sesión

En la figura 1, se observa la interfaz de inicio de sesión el cual cuenta con un botón(Volver) el cual tiene asociada la acción para volver a la escena anterior. Se le agregó un cuadro de texto con el mensaje para ingresar el nombre de usuario, el cual lo registra y genera en la interfaz del terapeuta, y otro cuadro de texto vacío para ingresar dicho usuario requerido, además se cuenta con otro botón (Login), el cual verificara si el usuario es correcto o incorrecto.



Figura 1: Interfaz de inicio de sesión

Una vez presionado el botón de login, se ejecuta una función para obtener los usuarios registrados en la base de datos y validar si se ingreso el usuario correcto, una corrutina para realizar el inicio de sesión correcto de los datos con el nombre utilizado, otra corrutina con un retraso para darle tiempo al programa de obtener los datos desde firebase, otra función para obtener la fecha necesaria los datos de los ejercicios.

```
sacarUsuarios ();  
StartCoroutine (Login (emailLoginField , passwordLoginField));  
StartCoroutine (Saltos1 ());  
otenerFecha ();
```

La función sacarUsuarios guarda los datos obtenidos de la base de datos, también guarda los valores de conteo para saber cuantos doctores hay registrados en la base de datos y su posición en orden de registro. Recorre todo el vector de datos obtenidos de la base de datos, en este caso de las terapeutas, y se guardan dichas posiciones en un vector. También se realiza otro recorrido para obtener los datos un peldaño mas abajo de la jerarquía de la base de datos, en cual corresponde a los usuarios registrados para cada terapeuta, para poder sacarlos y hacer la respectiva comparación.

Usando un condicional para saber si el usuario ingresado es correcto y posteriormente guardarlo y hacer la verificación. El usuario ingresado en la interfaz se compara con los usuarios registrados en la base de datos para realizar la verificación de usuario, además el terapeuta

al que esta registrado, igualmente en caso que el usuario no se encuentre registrado con ningún terapeuta, se muestra un mensaje de usuario invalido. Dentro de la corrutina de login, se obtiene la información de usuario registrada en la base de datos, además se obtienen los datos de nombre y documento, mediante los comando de lectura de la librería de firebase mostrados a continuación.

```
//Se obtiene de firebase el nombre y la identificacion del usuario
Userss=snapshot . Child (OnHit . nicks) . Child ("Nombre_Completo") . Value . ToString ()
;
Userssd=snapshot . Child (OnHit . nicks) . Child ("Documento") . Value . ToString () ;
```

En el código de inicio de sesión, Se obtienen los ejercicios asignados al usuario ,mediante el nombre ingresado en la interfaz de inicio se sesión ,también se obtiene los datos de las posiciones de los estantes del modelo 3D, el reto programado en la interfaz de creación de ejercicios, la fecha asignada para la realización de cada ejercicio y datos de numero de ejercicios por paciente e intentos por ejercicio.

Código para inicio de sesión

```
//Se ejecuta la corrutina para obtener el nombre de usuario
StartCoroutine (Loaduserss (Token));
//Se ejecutan todas las corrutinas para obtener los datos completos de
    usuario
//pero sin antes verificar que haya un nombre de usuario y un token(id de
    terapeuta encargado) para aviar errores de variables vac as
if (Token != null && OnHit.nicks != null)
{
    StartCoroutine (SacarEjercicio (Token, OnHit.nicks));
    StartCoroutine (LoadUserData (OnHit.nicks, Ejercicio, Token));
    StartCoroutine (loadReto (OnHit.nicks, Ejercicio, Token));
    StartCoroutine (Loadintentos (OnHit.nicks, Ejercicio, Token));
    StartCoroutine (LoadFecha (OnHit.nicks, Ejercicio, Token));
}
```

Para la obtención los ejercicios asignados al usuario, en el código de obtención de ejercicios, se llaman los datos en la jerarquía de la base de firebase correspondientes a los ejercicios y se guarda la cantidad de ejercicios y el nombre de cada ejercicio.

Código de obtención de ejercicios

```
//Se guarda el n umeor de ejercicios del usuario
numeroE = int . Parse (snapshot . ChildrenCount . ToString ());
Eje = new string [numeroE];
//Foreach para sacar el nombre de los ejercicios
foreach (var us in snapshot.Children)
{
    Eje [i] = us . Key . ToString ();
    Debug . Log (" Ejercicios :_" + Eje [i]);
    i++;
}
```

la corrutina loadUserData , llama los datos en la jerarquía de la base de firebase correspondientes a las posiciones de los estantes, y en el código de asignación de objetos a posiciones, se guardan los datos obtenidos de firebase sobre la posición del monster, el tiempo limite dependiendo del ejercicio, y el contenido frontal del estante, de esta misma forma para la parte trasera del estante y los laterales.

Código de asignación de objetos a posiciones

```
//SE obtiene de firebase la posicon del monster
Monster = int.Parse(snapshot.Child("Monster").Value.ToString());
//Se obtiene de firebase el tiempo limite par acompletar la mision
tiempoL = snapshot.Child("Tiempo").Value.ToString();
//Se obtiene de firebase los nombres de los objetos en cada posicion del
    estante
espacioF1 = snapshot.Child("estanteFrontal1").Value.ToString();
espacioF2 = snapshot.Child("estanteFrontal2").Value.ToString();
espacioF3 = snapshot.Child("estanteFrontal3").Value.ToString();
espacioF4 = snapshot.Child("estanteFrontal4").Value.ToString();
espacioF5 = snapshot.Child("estanteFrontal5").Value.ToString();
espacioF6 = snapshot.Child("estanteFrontal6").Value.ToString();
espacioF7 = snapshot.Child("estanteFrontal7").Value.ToString();
espacioF8 = snapshot.Child("estanteFrontal8").Value.ToString();
espacioF9 = snapshot.Child("estanteFrontal9").Value.ToString();
espacioF10 = snapshot.Child("estanteFrontal10").Value.ToString();
espacioF11 = snapshot.Child("estanteFrontal11").Value.ToString();
espacioF12 = snapshot.Child("estanteFrontal12").Value.ToString();
```

De la misma forma en la corrutina CargarReto se llaman los datos correspondientes de la jerarquía en firebase para los retos por ejercicio. Para la corrutina de CargarFecha se ubico en la parte de de la jerarquía correspondiente a los usuarios, una vez ubicados en la parte correcta se procede a obtener los datos, lo cual se hace en el código de obtención de datos de ejercicios, datos como el numero de ejercicios, numero de intentos por ejercicio y el intervalo de fechas en las que el paciente podrá realizar el ejercicio.

Código de obtención de datos de ejercicios

```
//Se obtiene de firebase la cantidad de ejercicios asigandos al usuario
NumEjercicio = int.Parse(snapshot.Child(paciente).Child("Ejercicio").
    ChildrenCount.ToString());
//SE obtiene de firebase el numeor d eintentos realizado por ejercicio
intentos = int.Parse(snapshot.Child(paciente).Child("Errores").Child
    (ejercicio).ChildrenCount.ToString());

//Se obtiene de firebase el intervalo de fecha en la que se podra realizar
    el ejercicio
FF = snapshot.Child("FechaFin").Value.ToString();
FI = snapshot.Child("FechaInicio").Value.ToString();
```

Perfil de Usuario

La interfaz del perfil de usuario cuanta con dos secciones, como se ve en la figura 2, al lado izquierdo se muestran los datos de usuario como lo son el nombre y el documento, los

cuales de obtienen previamente al iniciar sesión con el usuario, al lado derecho se aprecian los ejercicios asignados al usuario previamente cargados, los cuales se podrán realizar de acuerdo al intervalo de tiempo definido previamente cuando se generó el ejercicio, además también se asigna un botón(salir) con la acción para volver a la pantalla de inicio, para acceder a los ejercicios se debe presionar el botón correspondiente el cual posee el nombre del ejercicio creado, por ejemplo en el caso de la figura 3, el botón tiene asignado el nombre del ejercicio Peras, y solo se debe presionar para dirigirlo Asia la siguiente escena la cual mostrara las condiciones del reto para ese ejercicio.



Figura 2: Interfaz de perfil de usuario

Pantalla de objetivos Una vez presionado el botón del ejercicio, se llevara a la escena siguiente mostrada en la figura 3, la cual tiene la información del reto/misión asignada al ejercicio, para que el usuario lo realice, esta infracción se obtienen de la base de datos al realizar el inicio de sesión . Con la información del intervalo de fechas por ejercicio obtenidas anteriormente, se agregaron dos botones, uno para volver a la escena anterior, el cual también tiene asignado la función de login nuevamente, esto para que vuelva a cargar los datos del usuario y evitar que la información se sobre escriba y de errores, también un botón para avanzar ala siguiente escena, en caso de que el ejercicio este en las fechas correctas se podrá presionar el botón siguiente, sin embargo si el ejerció no se encuentra en el intervalo indicado, se desactivara el botón "Siguiente".

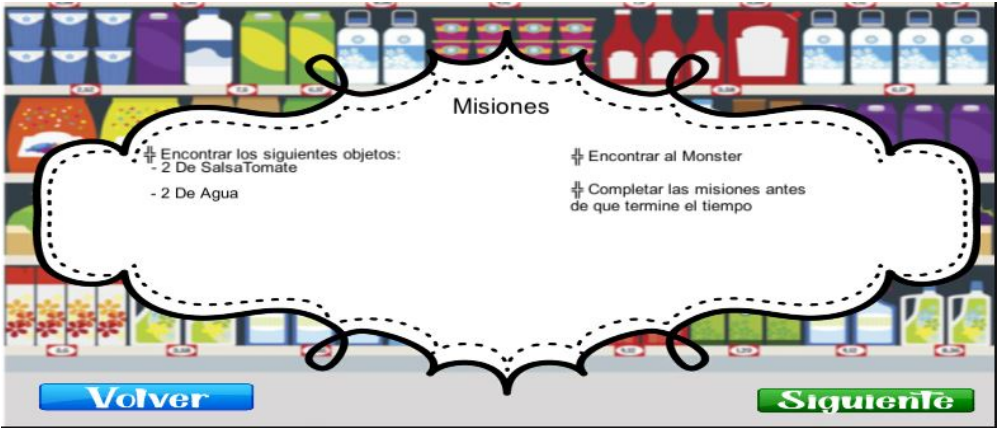


Figura 3: Interfaz de misiones

Escena de juego

Para la realización de la escena de juego en realidad aumentada, se utilizó la herramienta de Vuforia para la detección de planos que ya viene incorporada con el paquete de instalación, empezando se eliminan la cámara que trae la escena por defecto y se agrega una cámara como se observa en la figura 4. Para la detección de planos de Vuforia se agregan dos herramientas: *Ground plane stage* y *Plane Finder*, como se ve en la figura 5, el *Ground plane stage* actúa como una superficie virtual en la cual se va a asociar el plano detectado, y se puede posicionar objetos en ese espacio, *Plane Finder* es el buscador de planos en el entorno real.

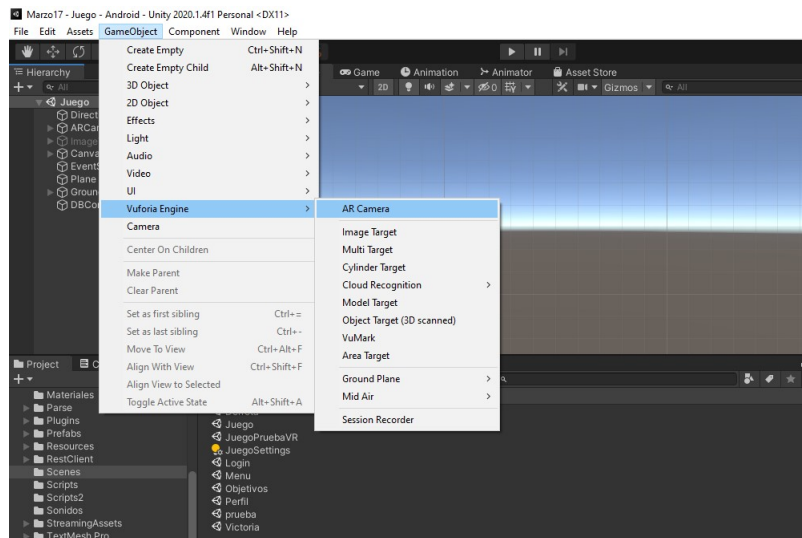


Figura 4: Agregar cámara en AR

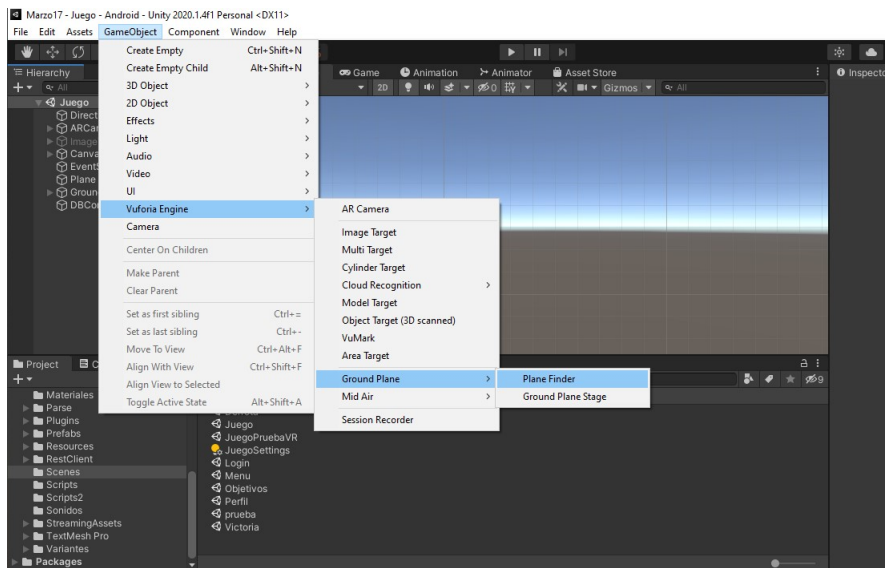


Figura 5: Agregar herramientas de para detección de planos

En el inspector de *Plane Finder* se debe colocar en la parte de *Content Positioning Behaviour* y dentro de *Anchor Stage* arrastrar el *Grund Plane Stage*, como se muestra en la

figura 6, cuando el buscador de planos detecte un *click*, posicionara la superficie de *Ground plane stage* en el plano detectado en el entorno real, por ende los objetos asociados a este se proyectaran también en el entorno real, debido a eso, el objeto que se requiere proyectar en la realidad aumentada debe estar dentro de la jerarquía de *Ground Plane Stage*, como se ve en la figura 7

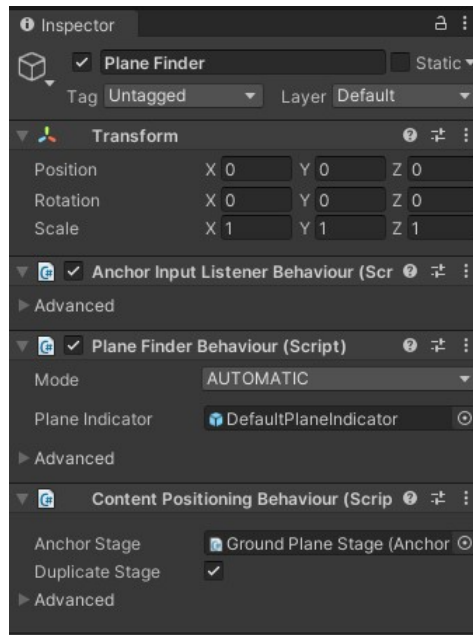


Figura 6: Inspector de *Plane Finder*

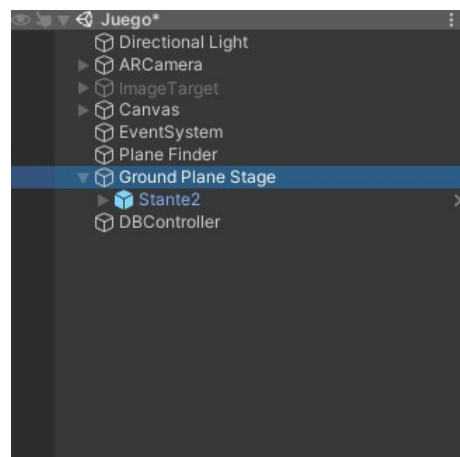


Figura 7: Jerarquía de *Ground Plane Stage*

Con esto realizado se puede proyectar el objeto en la realidad aumentada mediante un simple *click*, sin embargo por defecto se puede reubicar con cada *click* o activando la opción de duplicado se puede crear tantos objetos como se desee, para nuestro caso solo era necesario proyectar el objeto una vez y luego interactuar con el, por esto en el Código para desactivar *Plane Finder*, se realiza un *script* para anular *Plane finder*, para que al

instanciar la primera vez el objeto, se desactiva *Plane finder* para que no busque mas superficies y pueda proyectar mas objetos o repositonar el que esta colocado.

Código para desactivar *Plane Finder*

```
public void OnMouseDown()  
{  
    //Desactivar Plane Finder  
    plano.SetActive(false);  
}
```

Con esto realizado, cuando se ejecute la aplicación se mostrara un indicador para saber donde proyectar la el objeto (figura 8), en este caso se proyectara el estante con los objetos correspondientes , datos los cuales se obtienen al momento de realizar el inicio de sesión.



Figura 8: Puntero para colocar objeto

Todos los objetos se encuentran ocultos y desactivamos, como se aprecia en la figura 9. Teniendo en cuenta la jerarquía de los objetos, en el código ejemplo de activación de objetos, mediante un *switch case* y con los datos de cada posición de los objetos en el estante recibidas de la base de datos, se activaran de manera correcta para cada ejercicio.

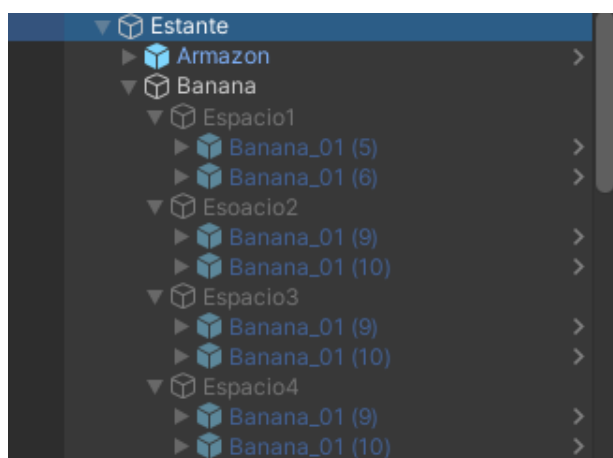


Figura 9: Jerarquía de objetos desactivados

Código ejemplo de activación de objetos

```
switch (SeleccionDeOjeto.Estante1[i])
{
    case "Banana" :
        T_Banana = E_Banana.transform.GetChild(i).gameObject;
        T_Banana.SetActive(true);
    break;
}
```

Debido a que el monster es un objeto especial para motivar al usuario a buscar mas a fondo, se le asignaron 4 posiciones especificas para su aparición, las cuales se encuentran desactivadas como se muestra en la figura 10. En el código de activación de Monster, se realiza un **switch case** independiente para su activación dependiendo del valor de posición tomado de la base de datos.



Figura 10: Jerarquía del Monster desactivado

Código de activación de Monster

```
switch (SeleccionDeOjeto.Monster)
{
    case 1:
        T_Monster = E_Monster.transform.GetChild(0).gameObject;
        T_Monster.SetActive(true);
    break;
    case 2:
        T_Monster = E_Monster.transform.GetChild(1).gameObject;
        T_Monster.SetActive(true);
    break;
    case 3:
        T_Monster = E_Monster.transform.GetChild(2).gameObject;
        T_Monster.SetActive(true);
    break;
    case 4:
        T_Monster = E_Monster.transform.GetChild(3).gameObject;
        T_Monster.SetActive(true);
    break;
    default:
        Debug.Log("No recibio ningun dato");
    break;
}
```

Interacción de objetos

Todos los objetos tiene la misma interacción, dependiendo de el reto programado por el ejercicio el objeto desaparecerá a si es necesario encontrarlo, añadiendo un contador de aciertos o en caso contrario de que no sea necesario encontrarlo y se seleccione simplemente emitir a una señal de error y aumentara el conteo de fallos. Como se aprecia en la figura 11, cada objeto tiene asignado un tag para poder identificarlo, utilizando esos tags en el código ejemplo para identificación de objeto y conteo, se realiza un **switch case** para identificar cual objeto esta siendo pulsado, donde tipoObjeto es una variable la cual contiene el tag del objeto, una vez identificado el objeto se procede a hacer un conteo del objeto para mas adelante identificar la condición de victoria.

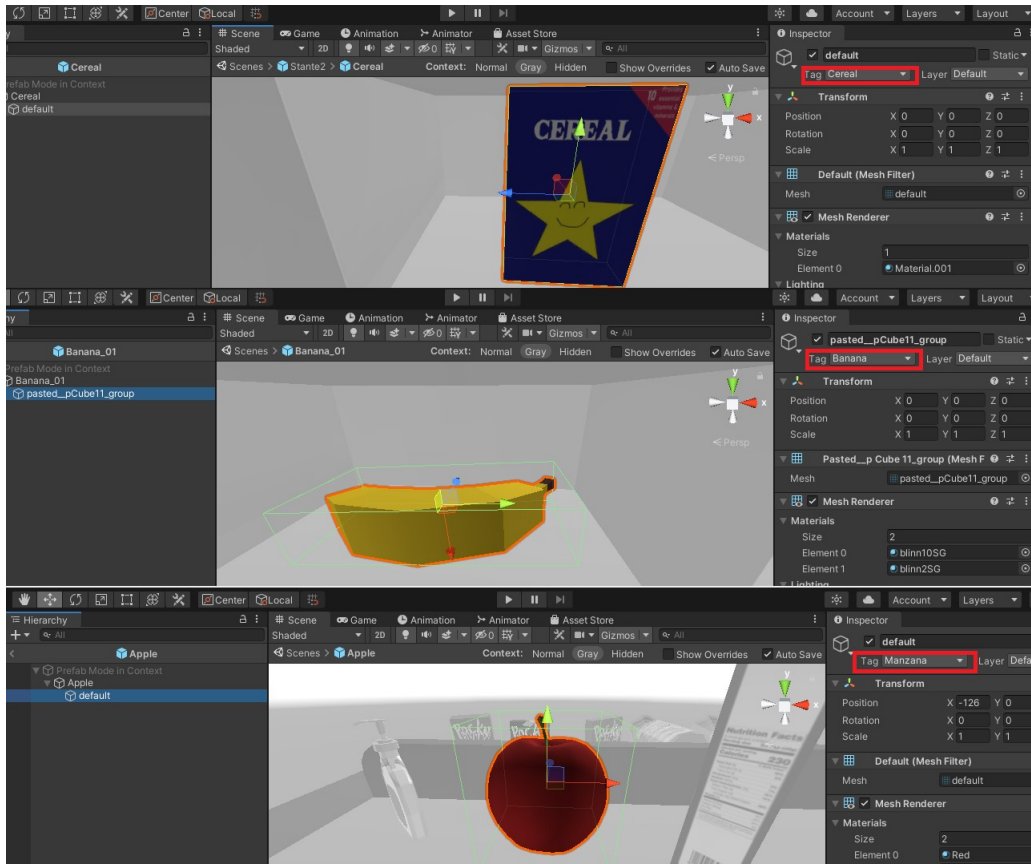


Figura 11: Asignación de tag

Código ejemplo para identificación de objeto y conteo

```
switch (tipoObjeto)
{
    case "Banana":
        Timer.Banana = Timer.Banana + 1;
        break;
}
```

Una vez identificado el objeto se hace la comparación con los valores requeridos por el reto, para saber si se debe destruir el objeto y considerar como acierto, o por el contrario mandar

la señal de error y marcar como fallo, en el código para selección de objeto correcto, se muestra la lógica usada, en el cual se realiza una secuencia de ciclos que tendrá una duración igual al numero objetos por cada misiones , para hacer el recorrido completo, se asegura que solo se destruya el objeto correcto mediante la condición de que el tag del objeto sea igual a nombre del objeto puesto en la misión , si se cumple esta condición se incrementa el numero de aciertos y se destruye a el objeto para evitar que vuelva a usarlo como acierto, de lo contrario se contara como fallo, se realiza la sumatoria al conteo de errores y para el envío de fallas se divide entre la cantidad de veces que el ciclo alcanzo a realizar, para evitar que se cuenten mas errores .

Código para selección de objeto correcto

```

for (int i = 0; i < SeleccionDeOjeto.mNombres.Length; i++)
{
    //Compara el tag del objeto con los nombres de los objetos de la mision
    //para saber si selecciono el objeto correcto
    if ((tipoObjeto == SeleccionDeOjeto.mNombres[i] || tipoObjeto == "
    Monster"))
    {
        //Llama a la fucnion de ocndicion de victoria y aumenta el numero de
        //aciertos
        SeleccionDeOjeto.condicionVictoria();
        Timer.acierto += 1;

        //Llama al sonido de interacion con el objeto y los destruye para
        //que no siga sumando aciertos ocn ese objeto
        Instantiate(sonidoSelect);
        Destroy(item.gameObject);
        //COMo es unacierto, le resta la cantidad de veces que va el ciclo
        //for a los errores para evitar falsos conteos,
        //ademas de romper el ciclo ya que el objeto esta dentro de los que
        //se deben buscar
        Timer.conteoErrores = Timer.conteoErrores - i;
        break;
    }
    //COndicion cuando el objeto no es el correcto
    else if (tipoObjeto !=SeleccionDeOjeto.mNombres[i] && tipoObjeto!= "
    Monster")
    {
        //Aumenta en 1 el conteo de errores e instancia el objeto de sonido
        //par ala interaccion con el objeto
        Timer.conteoErrores += 1;
        Instantiate(sonidoBoton);
    }
}

//Guarda el conteo de erroes en la varaible fallas del script de
//SeleccionDeOjeto
//y se la divide entre el n umeor de objetos de la mision para descontar los
//conteos extra del ciclo for
SeleccionDeOjeto.fallas = Timer.conteoErrores / SeleccionDeOjeto.mNombres.
Length;

```

Condición de Victoria y envío de datos

Una vez empezado el juego solo se puede llegar a dos escenas siguientes, escena de victoria(figura 12), y escena de derrota(figura 13), esto depende de la condición de victoria la cual esta definía por la el reto asignado al ejercicio, como se ven en el código de condición de victoria, para que la condición de victoria sea valida se compara el conteo de los objetos pulsados correctamente con el objeto y el numero que esta establecido en el reto, en caso de que sean pulsados los objetos correctos en el tiempo indicado cambiara a la escena de victoria y mandara los datos a firebase, sin embargo para la pantalla de derrota solo es necesario que se termine el tiempo.



Figura 12: Pantalla de Victoria



Figura 13: Pantalla de derrota

Código de condición de victoria

```
for (int i = 0; i < mNombres.Length; i++)
{
    Timer.val[i] = Timer.conObjetos[posiciones[i]];
}

for (int i = 0; i < mNombres.Length; i++)
{
    //SE hace la evaluacion si el conteo de objetos es igual al numero de
    //objetos requerido por la mision
    if (Timer.val[i] == misiones[i])
    {
        //Se aumenta en 1 el valor de la variable conteo para poder contar
        //las misiones que se vna completando
        //Ademas de cambiar el nombre del objeto de la mision a "Completado"
        //y
        //el numero de objetos a buscar a "-1" para que no sigan contando los
        //aciertos de ese objeto
        Timer.conteo++;
        mNombres[i] = "Completado";
        misiones[i] = -1;
    }
}
```

Inmediatamente cambia a la escena de victoria o derrota (figura 12 y 13 respectivamente), se enviarán los datos a la base de datos de firebase, los cuales se utilizarán para los análisis en el módulo de reportes, los datos que se envían son el tiempo utilizado, el número de fallas, y el número de aciertos, en una posición ya definida dentro de la jerarquía de la base de datos.

```
public void SignOutButton()
{
    auth.SignOut();
    UIManager.instance.LoginScreen();
    LimpiarBotones();
    nomUsuario.text = "";
    usUsuario.text = "";
    ...
    ClearRegisterFields();
    ClearLoginFields();
}
```

ANEXO F

Lógica principal modulo de reportes

Para el del modulo de reportes se parte de la implementación de las librerías y paquetes ya mencionados en la seccion 2.3.3 de librerías utilizadas, más específicamente en las librerías utilizadas para el modulo de reportes, algunas de estas se importan teniendo en cuenta la previa instalación de la misma. Generalmente la instalación se realiza utilizando el comando en terminal *"pip install"* seguido de la librería deseada. Todo el modulo se desarrollo en un solo script llamado *main.py*.

```
import sys
from PyQt5 import QtWidgets
from PyQt5.QtWidgets import QDialog, QApplication
from PyQt5.uic import loadUi
import pyrebase
import matplotlib.pyplot as plt
import numpy as np
from reportlab.lib.pagesizes import A4
from reportlab.pdfgen import canvas
from reportlab.lib.units import inch, mm
from reportlab.lib.utils import ImageReader
import random, math
import pathlib
from datetime import datetime
```

Una vez se ha importado se realiza la conexión con la base de datos utilizando *pyrebase*, esta conexión es única para el proyecto por lo que los parámetros de acceso de la base de datos son obtenidos en el mismo proyecto generado. Estos parámetros contienen la configuración de la base de datos que son atribuidos a un diccionario denominado *firebaseconfig*.

```
firebaseConfig = {
    "apiKey": "AIzaSyCStKY5gKZ28i9hvx9tez9byonQgHoojm4",
    "authDomain": "appterapeuta.firebaseio.com",
    "databaseURL": "https://appterapeuta-default-rtdb.firebaseio.com",
    "projectId": "appterapeuta",
    "storageBucket": "appterapeuta.appspot.com",
    "messagingSenderId": "836459726857",
    "appId": "1:836459726857:web:058a18d22934756c0190d8"
}
```

Después de establecer los parámetros de acceso y asignarlos a *firebaseConfig*, se hace la inicialización de la base de datos con esta información para que la conexión se establezca, se usan las funciones de *pyrebase* para asignar la base de datos a *db* y el sistema de autenticación de usuario a *auth*, esto debido a la cantidad de terapeutas que pueden registrarse en la herramienta; por lo que usando las credenciales de registro en el modulo de configuración se puede excluir la información de pacientes que no forman parte de su diseño de ejercicios.

```
firebase = pyrebase.initialize_app(firebaseConfig)
db = firebase.database()
auth = firebase.auth()
id = 'default'
```

Se establecen variables globales que se usaran para almacenar datos usados en la lógica de programación y la información extraída de la base de datos, entre esa información están los datos del paciente, los resultados de la realización del ejercicio y los parámetros configurados en este.

```
intentos = []
fallas = []
tiempoSobrante = []
tiempoSobranteE = []
ejercicios = []
ejerciciosE = []
nombresEjercicios = []
nombresErrores = []
configEjercicios = []
num2 = []
nombreDatos = []
pacienteN = ''
pacienteS = ''
numeroEjer = 0
nombreCompleto = ''
documentoPaciente = ''
userTerapeuta = ''
```

Interfaz inicial

Para el diseño de la interfaz principal usando *PyQt5* se hizo el diseño en la herramienta de *designer.py* que se instala al momento de contar con la librería de *PyQt5*, el diseño de la interfaz principal cuanto con un total de 3 objetos interactuables, 2 campos de texto uno para correo y otro para la contraseña y un botón de validación o para acceder. Si la información es correcta podrá acceder a la interfaz principal, de lo contrario se visualizara un aviso en texto indicando que alguno de los dos campos de texto contiene información incorrecta. La interfaz se puede apreciar en la figura 1 y fue guardada como *login.ui*, para esta interfaz se estableció una clase denominada login con dos funciones, la función *__init__* que llama al login.ui que es la interfaz diseñada con el designer y establece la relación con los botones y los campos de texto, la función *loginfunction* que establece la conexión con la base de datos permitiendo autenticarse accediendo a otra clase que es la de la interfaz principal y en caso de no poder autenticarse, mandar el aviso para volverlo a intentar.

```
class Login(QDialog):
    def __init__(self):
        super(Login, self).__init__()
        loadUi("login.ui", self)
        self.loginbutton.clicked.connect(self.loginfunction)
        self.password.setEchoMode(QtWidgets.QLineEdit.Password)
        self.invalid.setVisible(False)

    def loginfunction(self):
        email=self.email.text()
        password=self.password.text()
```

```
try:
    user = auth.sign_in_with_email_and_password(email, password)
    global id
    id = user['localId']
    createacc=CreateAcc()
    widget.addWidget(createacc)
    widget.setCurrentIndex(widget.currentIndex()+1)
except:
    self.invalid.setVisible(True)
```



Figura 1: Interfaz acceso/login modulo de reportes

Interfaz principal

La interfaz principal diseñada en el *designer.py* y guardada como *createacc.ui*, fue hecha de tal manera que requiera colocar el nombre identificativo o usuario del paciente. Un botón para verificar si este cuenta con datos, denominado *CARGAR DATOS* visible en la figura 2, en caso de ser correcto y contar con información se habilitara un botón oculto divisable en la figura 3 llamado *GENERAR REPORTE* el cual generara un documento en pdf en la carpeta donde se encuentre el ejecutable de esta aplicación.



Figura 2: Interfaz de extracción de datos paciente



Figura 3: Interfaz de generación de reporte y reutilización

La lógica usada para esta interfaz principal en el mismo script *main.py*, se definió de la siguiente manera. Una nueva clase para empaquetar toda la estructura, la primera parte es la de cargar la interfaz, ya que si los datos de acceso fueron correctos se llama a está siendo una función de tipo *__init__*, en donde los botones de generar reporte se oculta hasta

cuando se pueda confirmar si la información de usuario suministrada es correcta y contiene datos.

```
class CreateAcc(QDialog):
    def __init__(self):
        super(CreateAcc, self).__init__()
        loadUi("createacc.ui", self)
        self.BtnGenerar.setVisible(False)
        self.BtnGenerar.clicked.connect(self.generarPDF)
        self.generarid.clicked.connect(self.mostrarDatos)
        self.BtnLimpiar.clicked.connect(self.LimpiarDatos)
        self.confirm.setVisible(False)
        self.datoconfi.setVisible(False)
        self.datofall.setVisible(False)
        self.BtnLimpiar.setVisible(True)
```

la siguiente función es la de *LimpiarDatos*, esta función permite que todas las variables globales y locales usadas en esta clase, borren sus datos o se establezcan por defecto para poder hacer una nueva búsqueda de usuario y generar un nuevo reporte. De no hacerlo, los datos del usuario antes buscado con el nuevo se juntarían generando un reporte impreciso.

```
def LimpiarDatos(self):

    self.paciente.clear()

    self.confirm.setVisible(False)
    self.datoconfi.setVisible(False)
    self.datofall.setVisible(False)
    self.generarid.setVisible(True)
    self.BtnGenerar.setVisible(False)
    self.BtnLimpiar.setVisible(False)

    global nombresEjercicios
    global nombresErrores
    global configEjercicios
    global pacienteDatos
    global nombreDatos
    ...

    infoIntentoT = []
    infoIntentoAN = []
    infoIntentoFN = []
    infoIntentoTN = []
    infoIntentoANsobreTotal = []
    infoIntentoTNsobreTotal = []
```

La tercera función es la más importante, denominada *mostrarDatos* es la que se encarga de verificar si primeramente el usuario que se coloco en el campo de texto sea correcto y cuente con información ya que toda la función esta determinada en un *try except*, se intenta y de no ser correcto sale el aviso para que se ponga un usuario correcto o un usuario del que se tengan datos (Que haya realizado al menos una vez el ejercicio asignado) en contra parte si es correcto el usuario y cuenta con datos, la función se encarga de almacenar en variables globales y locales de esta clase información relevante para generar el informe o

reporte, como el nombre de los ejercicios, el numero de intentos, de fallas, de aciertos, toda esta información se almacena en las variables.

```
def mostrarDatos(self):
    try:
        ...

        self.datoconfi.setVisible(True)
        self.datofall.setVisible(False)
        self.BtnGenerar.setVisible(True)
        self.generarid.setVisible(False)
        self.BtnLimpiar.setVisible(True)

        global paciente
        paciente = self.paciente.text()
        global pacienteN
        pacienteN = 'Reporte_' + str(paciente) + '_' + fechastr + '.pdf'
        global pacienteS
        pacienteS = str(paciente)
        print(id)
        global nombreDatos
        global nombreCompleto
        ...

        dicci = dict()
        docci = []
        vi = dict()
        ki = dict()
        ji = dict()

        users = db.child("users").child(id).child("Usuarios").child(
            paciente).get()
        nombreCP = db.child("users").child(id).child("Usuarios").child(
            paciente).child('Nombre_Completo').get()
        nombreCompleto = nombreCP.val()
        documentoCP = db.child("users").child(id).child("Usuarios").
            child(paciente).child('Documento').get()
        documentoPaciente = documentoCP.val()
        userT = db.child("users").child(id).child("username").get()
        userTerapeuta = userT.val()

        ...

    except:
        self.datoconfi.setVisible(False)
        self.datofall.setVisible(True)
        self.BtnGenerar.setVisible(False)
```

La ultima función pero no menos importante es la función de *generarPDF* que principalmente usa la librería de *reportlab* y de *matplotlib*, la primera para asignar la información a una estructura establecida para presentarla en un pdf y posteriormente exportarlo, entre la información presentada esta el numero de ejercicios, el nombre completo del paciente y su

documento, la ubicación de los objetos del ejercicio entre otra información que se mostrara más adelante en la estructura del pdf, y la librería de matplotlib que es la que con los datos seleccionados permite generar gráficos para un mejor análisis de los mismos. Aparte de esto en esta función también se usa la librería de *numpy* para la media y la varianza del tiempo utilizado, siendo uno de los objetivos específicos del proyecto y que se muestra en el siguiente fragmento de código de esta función.

```
mediaTU = np.mean(infoIntentoTN)
mediaTS = np.mean(infoIntentoTNsobreTotal)
elcanvas.drawString(10 * mm, 90 * mm, "La media del tiempo _
    utilizado es:_" + str(round(mediaTU,1)) + "_segundos")

varianzaTU = np.var(infoIntentoTN)
varianzaTS = np.var(infoIntentoTNsobreTotal)
elcanvas.drawString(10 * mm, 70 * mm, "La varianza del tiempo _
    utilizado es:_" + str(round(varianzaTU,2)) + "_segundos")
```

ANEXO G

MANUAL DE USUARIO MINISÚPER AR



Rosmer Hasan Yepes Hoyos – Jhoan Sebastian Hurtado Campo

UNICAUCA rosmerh@unicauca.edu.co – jhoanhurtado@unicauca.edu.co

Contenido

ACCIONES PREVIAS.....	3
MÓDULO DE CONFIGURACIÓN	7
Acceder al módulo de configuración	7
Pantalla inicial	8
Registro en la aplicación.....	9
Iniciar sesión	11
Pantalla de bienvenida	12
Interfaz de administración de pacientes.....	13
Creación de paciente:.....	14
Modificación información de un paciente	18
Eliminación de un paciente de la base de datos	19
Visualización de un paciente	20
Interfaz de ejercicios	20
Creación de un ejercicio	22
Ver ejercicios	25
Establecer ejercicio.....	27
Modificar fechas ejercicio.....	28
Eliminar ejercicio	29
Interfaz de configuración de ejercicio	30
Configurar reto u objetivo	32
Eliminar reto y cantidades máximas	35
Configurar estante o entorno AR.....	36
Moverse entre las caras.....	38
Establecer objeto en un espacio	38
Botón mágico o de auto llenado	40

Ocultando el monster	42
Ubicaciones del monster	44
Establecer tiempo para realizar la búsqueda	46
Guardar configuración.....	46
APLICACIÓN MÓVIL	48
Instalar aplicación móvil en el dispositivo	48
Entrar por primera vez a la app	66
Iniciar sesión en la app	67
Interfaz principal	69
Acceder al ejercicio	70
Realizar ejercicio	71
Ver objetivo	71
Proyectar entorno AR	72
Interacción con los objetos.....	73
Extra, proyección en el suelo y movimiento alrededor	78
MÓDULO DE REPORTE.....	83
Extraer módulo de reportes	83
Ejecutar módulo de reportes.....	85
Iniciar sesión	86
Interfaz principal	86
Generar reporte	87
Contenido reporte.....	89

ACCIONES PREVIAS

Se debe acceder al siguiente enlace:

https://drive.google.com/drive/folders/1rYJ9YOu3dvWjtMllcKcv70UvNH_sk-6y?usp=sharing

Dentro de ese enlace se puede visualizar una carpeta denominada “MiniSuperARFull”, se procederá a hacer clic derecho sobre esta y seleccionar la opción de descarga como se muestra en la figura 1.

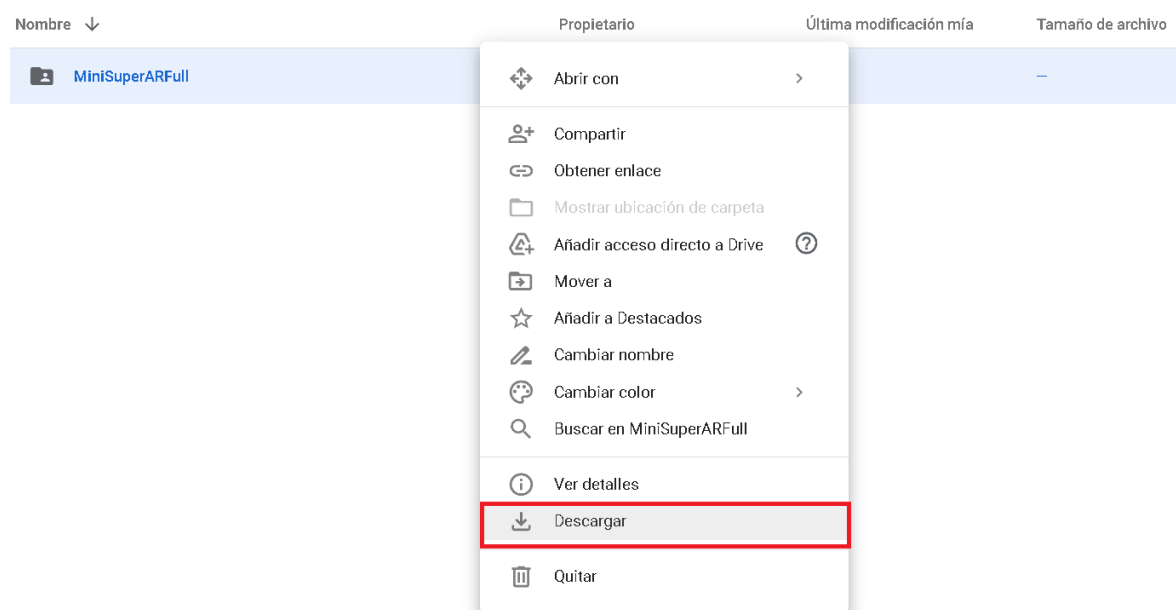


Figura 1.

Luego de seleccionar descargar, en la esquina inferior derecha se mostrará una ventana emergente que empezará a comprimir dicha carpeta (Figura 2) para su posterior almacenamiento en el lugar deseado (Figura 3). El tiempo que tarde esta descarga dependerá de la velocidad de conexión a internet que se posea.

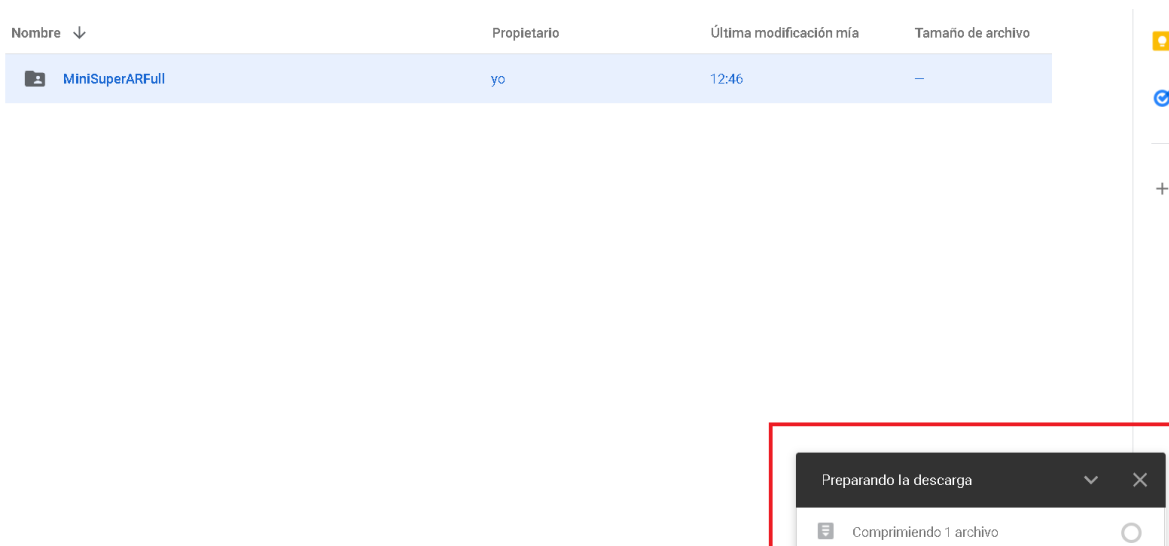


Figura 2.

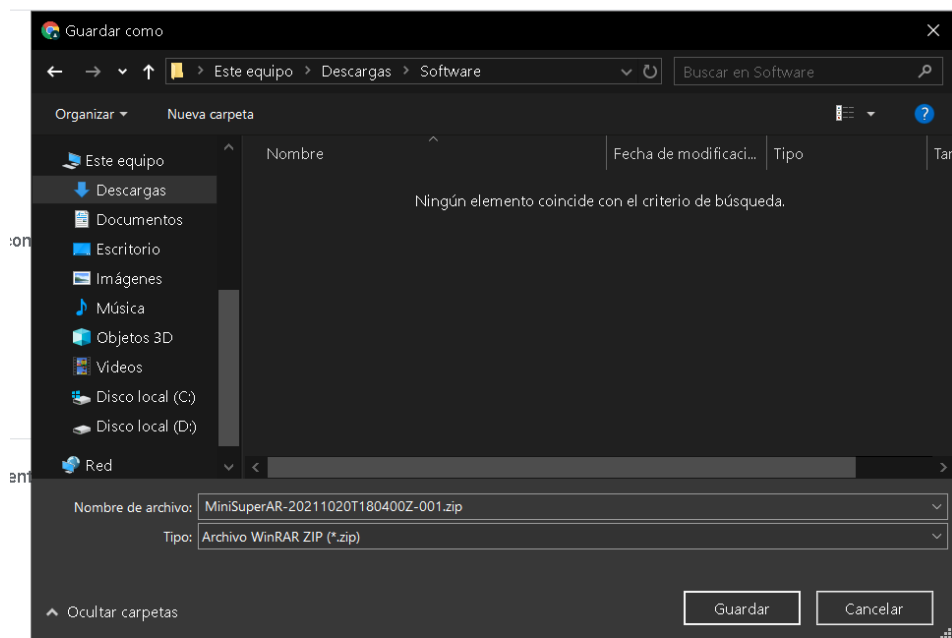


Figura 3.

Una vez guardado el archivo comprimido en el lugar de elección, se deberá acceder a este dando clic derecho y utilizando un software para extraer archivos comprimidos como winrar o 7zip, extraerá el archivo en la ubicación deseada esto se puede ver en la figura 4.

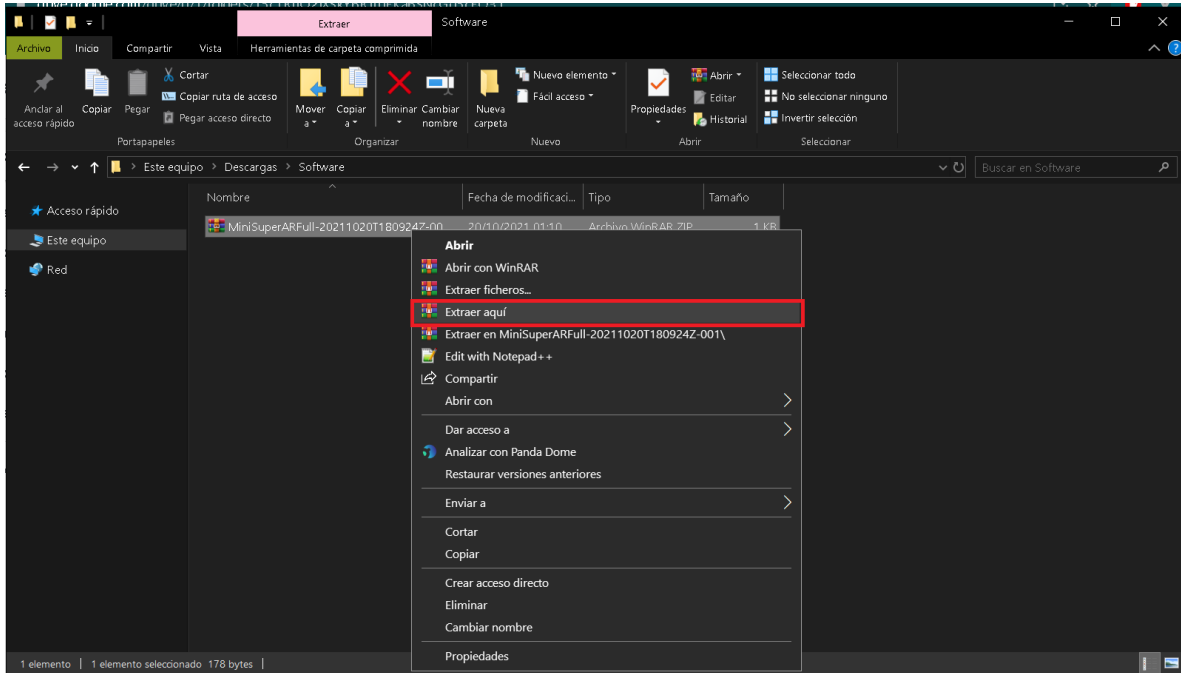


Figura 4.

Al extraerlo se generará una carpeta con el mismo nombre que en el enlace “MiniSuperARFull” (Figura 5) que contendrá todos los módulos, los videos y esta guía en pdf. Con fines explicativos, solo se tendrá las carpetas con los módulos y se omitirá de la visualización los videos y la guía (Figura 6).

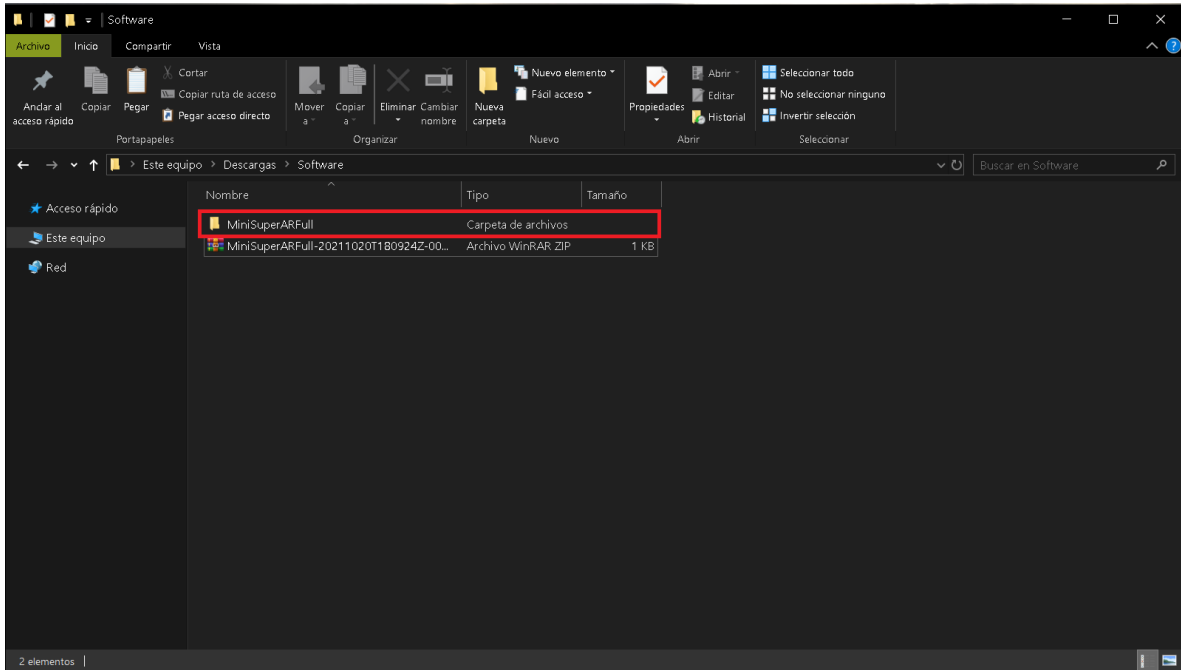


Figura 5.

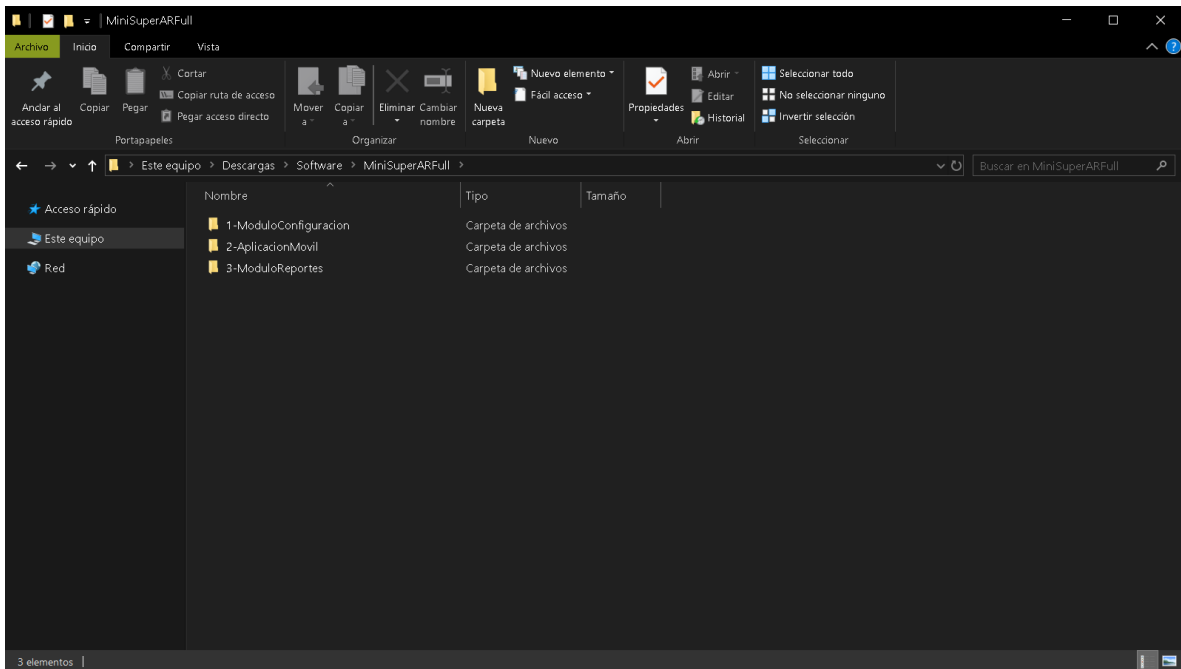


Figura 6.

Si se encuentra en este punto, ha terminado de realizar las acciones previas.

MÓDULO DE CONFIGURACIÓN

El módulo de configuración es la aplicación de escritorio en el que el terapeuta podrá administrar los pacientes y configurar en su totalidad los ejercicios para estos.

Acceder al módulo de configuración

Para acceder al módulo de configuración se debe entrar a la carpeta “1-ModuloConfiguracion”, luego acceder a la carpeta “ModuloConfiguracion” y dar doble clic al archivo “MiniSuperAR.Maestro.exe” con el cual podremos ejecutar la aplicación (Figura 7)

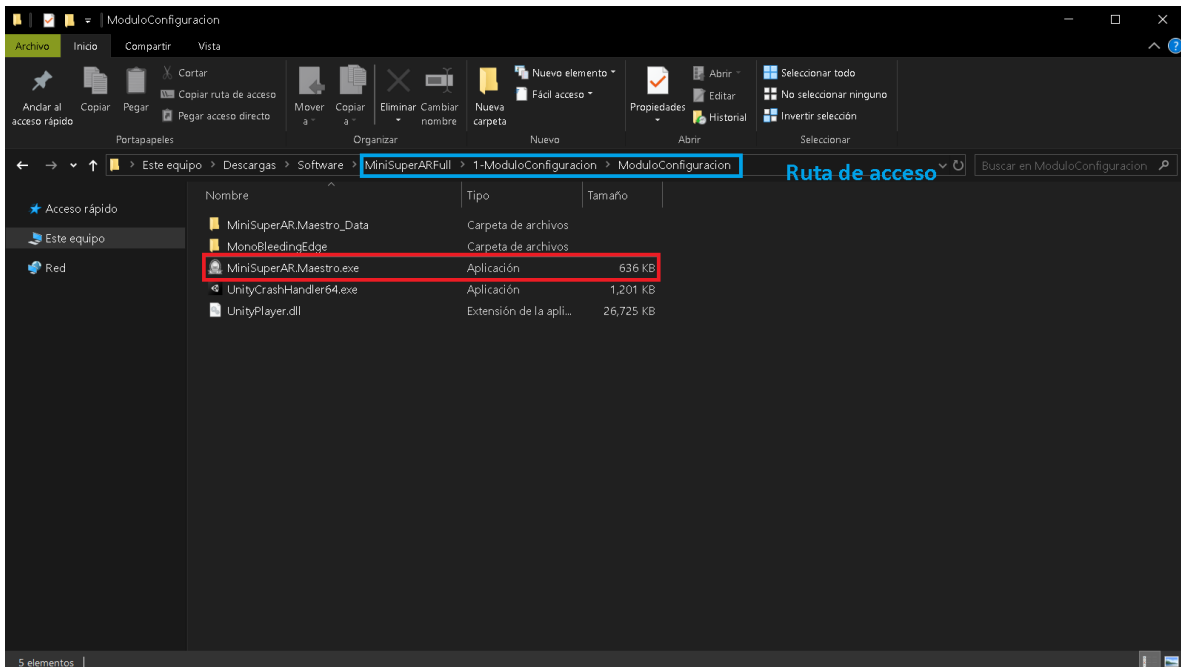


Figura 7

Pantalla inicial

En la pantalla inicial se cuenta con 4 botones, y dos campos de ingreso de texto como se ve en la figura 8.



Figura 8.

- (1) Botón Login: Este botón permite acceder con las credenciales de los campos de texto, correo (5) y contraseña (6) en caso de que este mal, aparecerá un aviso en rojo indicando cual es el campo de texto que presenta la falencia.
- (2) Botón Registro: Este botón nos permite entrar a la interfaz de registro para tener una cuenta en la aplicación y así poder acceder a las funcionalidades.

- (3) Botón Olvide mi contraseña: Este botón nos permite recuperar la contraseña si la hemos olvidado, se deberá colocar el correo con el que se accedía o con el que se registró en el campo de texto correo (5) y presionar el botón olvide mi contraseña, para que a ese correo llegue un formulario simple de restablecimiento y pueda acceder con la nueva contraseña.
- (4) Botón Salir: Este botón se presentará a lo largo de las interfaces o pantallas, permite cerrar la aplicación su totalidad.
- (5) Campo de texto correo: En este campo de texto se deberá colocar el correo de acceso con el que se registró en la aplicación.
- (6) Campo de texto contraseña: En este campo de texto se deberá colocar la contraseña asociada al correo (5) con la cual se hizo el registro.

Registro en la aplicación

Para hacer el registro en la aplicación se debe presionar el botón de Registro, el cual llevará a una interfaz de tipo formulario como se ve en la figura 9, se deberá llenar cada campo de texto. Lea la información detallada de cada campo de texto si tiene dudas de que debe ir.

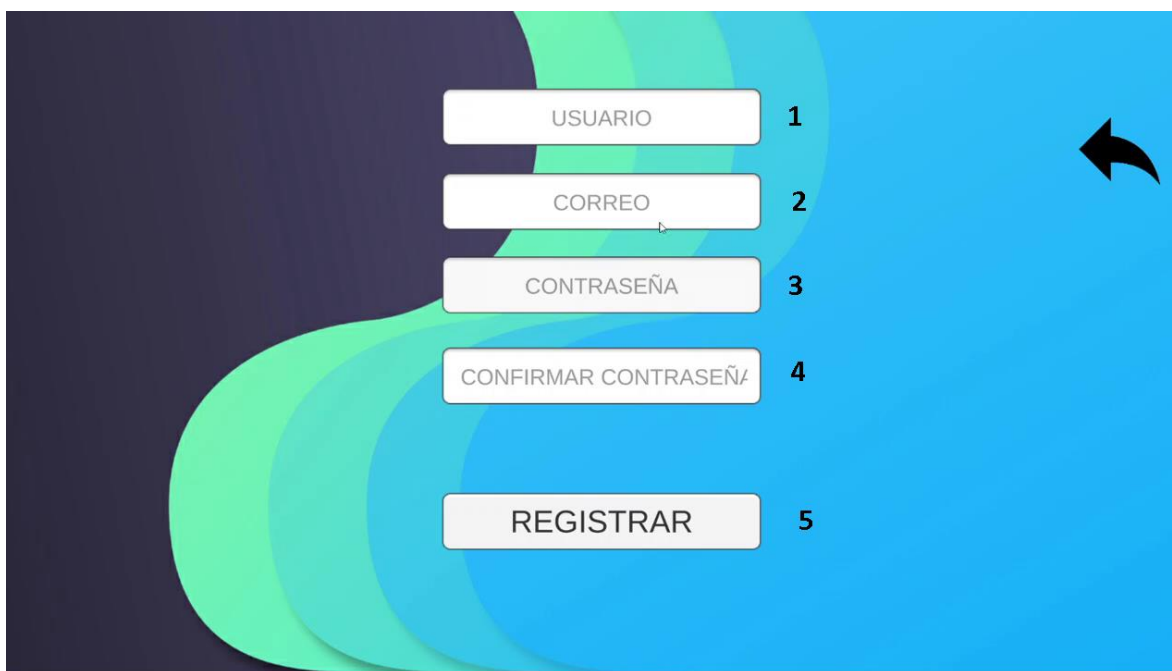


Figura 9.

- (1) Campo de texto usuario: Se debe colocar un nombre de usuario, un distintivo con el cual podrá identificarse en el sistema. No tiene límite mínimo de letras, este usuario no podrá cambiarse y en caso de existir alguno con este mismo, aparecerá un aviso que indicará que debe intentar con otro nombre. Ejemplo de usuario: DocLuis.
- (2) Campo de texto correo: Se deberá colocar el correo, este correo debe ser con formato de correo, o sea poseer un @mail.com como puede ser gmail.com o outlook.com. Se recomienda que este correo sea el personal pues en caso de restablecer la contraseña se enviara el formulario de restablecimiento a ese correo. Ejemplo de correo: luisd@gmail.com
- (3) Campo de texto contraseña: En este campo de texto se pondrá la contraseña, esta puede tener caracteres alfanuméricos o símbolos, se recomienda una contraseña fácil de recordar además de que debe tener como mínimo 6 caracteres. Una cantidad menor, no la aceptara la

aplicación. Ejemplo de contraseña: 123456

- (4) Campo de texto confirmar contraseña: Se debe colocar la misma contraseña que en el campo (3) para verificar la seguridad de que esa contraseña es la que se desea establecer para el correo (2)
- (5) Botón Registrar: Una vez llenados correctamente los campos al presionar este botón, llevara otra vez a la pantalla inicial para poder acceder con los datos con los cuales se registró (Correo y contraseña) en caso de que un dato este mal colocado, aparecerá un dialogo en rojo que mencionara el campo donde está la falencia.

Iniciar sesión

Para iniciar sesión se debe colocar la información de acceso en la pantalla de inicio, el correo y la contraseña para después presionar el botón de login, en dado caso que algún campo este mal aparecerá un dialogo en rojo indicando la falencia. Si todo es correcto, aparecerá en verde “Logeado” y accederá a la aplicación (Figura 10).



Figura 10.

Pantalla de bienvenida

En la pantalla de bienvenida (Figura 11) se da un saludo al usuario que accedió, en donde están dos botones. El botón de la equis blanca que como se mencionó, aparece en las demás interfaces y permite cerrar en su totalidad la aplicación y un botón de ir a pacientes en donde está la primera interfaz de configuración por lo que se debe presionar para ingresar.

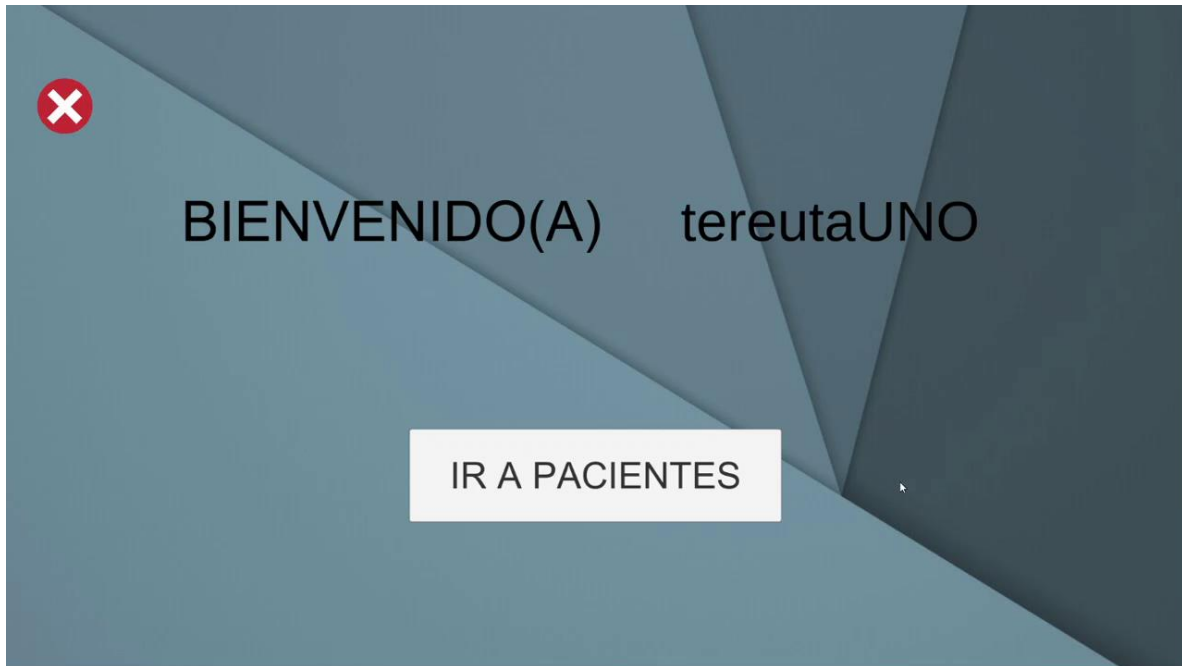


Figura 11.

Interfaz de administración de pacientes

Luego de oprimir el botón de “Ir a pacientes” se presenta la interfaz de pacientes (Figura 12) que cuenta con dos secciones la sección izquierda donde se pueden ver los nombres de los pacientes dentro de un botón cada uno (Para desplazarse entre los pacientes se debe presionar afuera de los botones de pacientes y moverse arriba o abajo) además abajo también está la cantidad de pacientes que se tienen registrados; y la sección derecha que contiene los botones para administrar el paciente y ver información adicional de este, para desplegar la información de un paciente solo se debe presionar el botón con su nombre y en la sección derecha aparecerá la información.



Figura 12.

Para ver el proceso de una mejor manera se hará el procedimiento para la creación de un paciente, la edición, la eliminación, la visualización y la selección.

Creación de paciente:

Si no se poseen pacientes la interfaz que se visualizara será como la figura 13, se debe presionar en el botón con un símbolo de más “+” que se resalta en rojo la figura 13, adicionalmente el botón “Cerrar sesión” resaltado en azul, vuelve a la pantalla inicial de iniciar sesión por lo que no se usara para la demostración.



Figura 13.

Una vez presionado el botón con el más “+”, saldrá la interfaz de creación de usuarios Figura 14. Aparecerá el botón de “Guardar” invalido (Que no se puede presionar), pues es de suma importancia verificar (Usando el botón “Verificar”) si el usuario que se le vaya a asignar al paciente no se encuentra ya registrado en la base de datos; cabe añadir que este usuario es lo más importante en la creación del paciente, pues con este usuario es que el infante podrá ingresar a la aplicación móvil para realizar los ejercicios por lo que se recomienda que el nombre de usuario sea fácil de recordar para él.

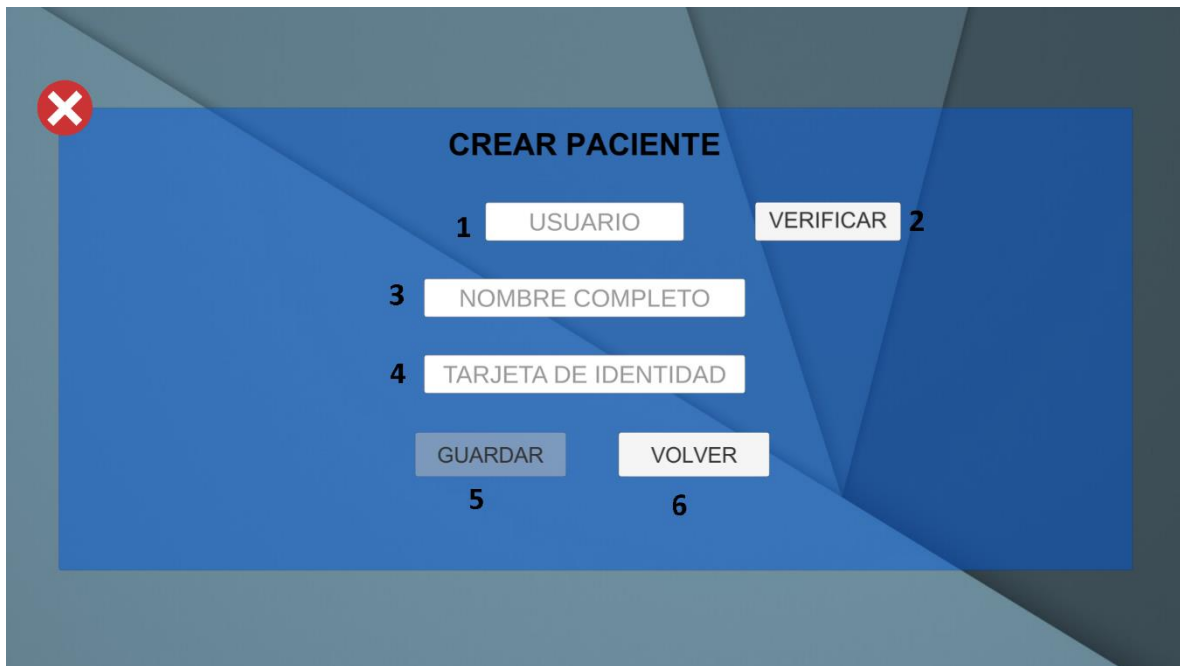


Figura 14.

- (1) Campo de texto usuario: Es la sección más importante de la interfaz de creación de usuarios, en este campo va a ir un nombre que puede poseer caracteres alfanuméricos y símbolos, pero que será la credencial única con la que el paciente podrá acceder a la aplicación móvil donde desarrollará el ejercicio cognitivo. Ejemplo de usuario: Punpun1
- (2) Botón Verificar: Ya que se trata de una credencial única el usuario, se deberá verificar que algún paciente no posea dicho usuario, por lo que una vez colocado el usuario en el campo (1) deberá de presionar este botón. Aparecerá un texto en rojo diciendo que ya existe en dado caso y en verde si es lo contrario. Además, si es en verde el botón guardar (5) será interactuable (Se podrá presionar) para guardar junto con los datos de (3) y (4)
- (3) Campo de texto nombre: Aquí podrá colocar el nombre completo del paciente, incluya apellidos.
- (4) Campo de texto tarjeta de identidad: Aquí podrá colocar el número de documento del paciente.

- (5) Botón Guardar: Este botón se habilitará una vez se haya verificado que el usuario que se le va asignar al paciente es único, cuando se presione aparecerá un mensaje que indicara que cargando los datos para posteriormente volver a la interfaz de usuarios automáticamente con el usuario ya almacenado y visible como se ve en la figura 15 donde ya se hizo el almacenamiento de un paciente.
- (6) Botón Volver: Al presionar el botón, se mantendrá la información que se encuentra ya en los campos (1) (3) (4) para retomarlos después. Aparecerá un mensaje que indicará que está cargando los datos para posteriormente volver a la interfaz de usuarios automáticamente con la información que se tenía.

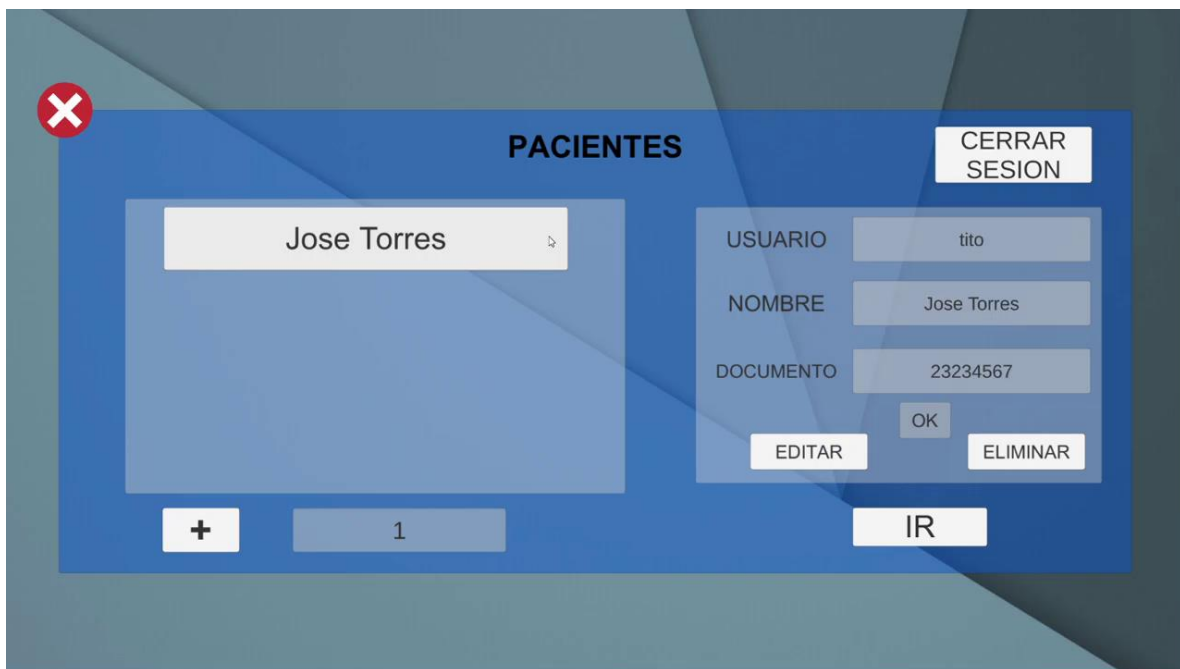


Figura 15.

Al crear un nuevo paciente el conteo de pacientes al lado del botón de más "+" aumentara.

Modificación información de un paciente

Para modificar la información de un paciente, se debe seleccionar el paciente presionando el botón con su nombre (Que su información salga en la sección derecha, usuario, nombre y documento) Una vez presionado se debe presionar también el botón “Editar” que se encuentra subrayado en el cuadro rojo en la figura 16.



Figura 16.

Cuando se presione el botón de “Editar” se habilitarán los campos de texto “Nombre” y “Documento” Figura 17, para que se pueda modificar la información, al igual que el botón “OK” que esta resaltado en un rectángulo azul en la figura 17. Una vez modificada la información de los campos ya habilitados, se debe presionar el botón “OK” y esperar un momento a que la información se actualice y vuelva a salir la interfaz normalmente con los botones de pacientes en la sección izquierda.



Figura 17.

Eliminación de un paciente de la base de datos

Igual que con la edición se debe seleccionar el paciente presionando el botón con el nombre para que aparezca su información en la sección de la derecha (Usuario, nombre y documento) luego de eso se debe presionar el botón "Eliminar" que se destaca en rojo en la Figura 18. Posteriormente se debe esperar un momento a que se cargue la nueva información reduciéndose el conteo de pacientes en el cuadro al lado del botón más "+" y saliendo los demás pacientes de nuevo.



Figura 18.

Visualización de un paciente

Para visualizar la información de un paciente ya creado, se debe oprimir el botón con su nombre y la información como el usuario, el nombre completo y el documento se verá en la sección derecha de la interfaz como se puede apreciar en la figura 18.

Interfaz de ejercicios

Para acceder a la interfaz de ejercicios se debe establecer un paciente para luego darle en el botón "Ir" que se encuentra resaltado en rojo en la figura 19.



Figura 19.

Al darle en el botón “Ir” aparecerá la interfaz de la figura 20, para saber si se está trabajando con el paciente adecuado debe salir su nombre en el cuadro rojo de la figura 20. Con eso se sabrá que se están haciendo las configuraciones pertinentes en el paciente a tratar.

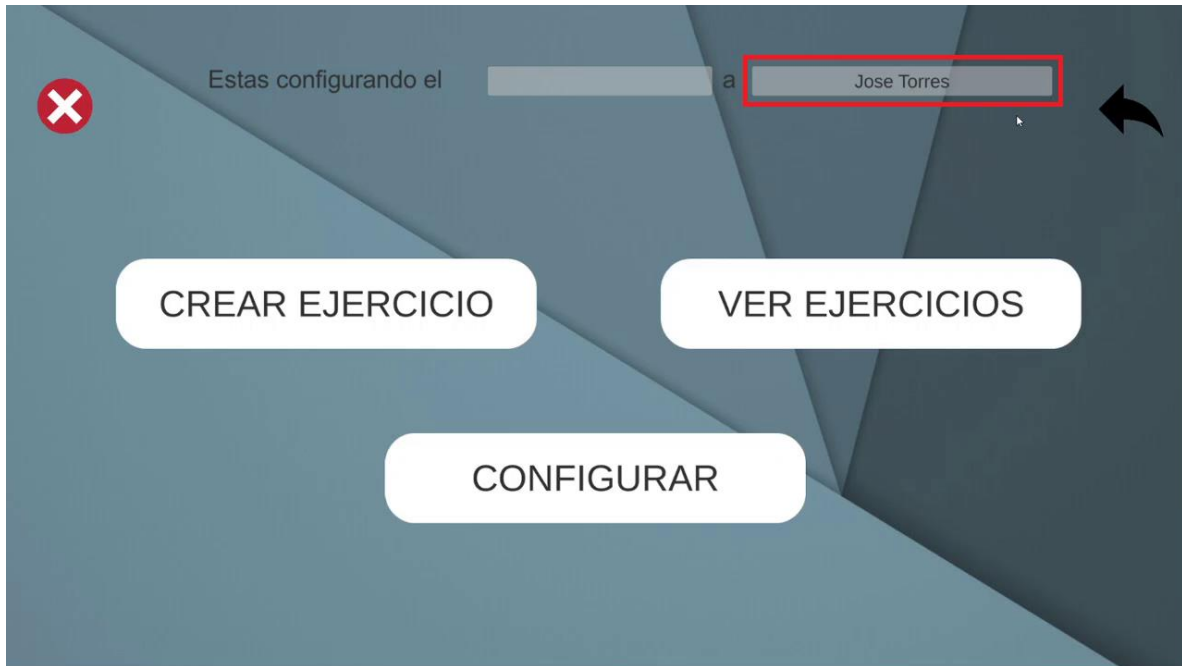


Figura 20.

Se puede observar también que hay un cuadro en blanco junto al nombre del paciente establecido, en ese cuadro debe estar el ejercicio a configurar. Debido a que no se tiene uno se hará el proceso de creación y establecimiento de un ejercicio para el paciente del ejemplo “Jose Torres”.

Creación de un ejercicio

Para la creación de un ejercicio se deberá presionar el botón “Crear ejercicio” resaltado en la figura 21, una vez presionado llevará a la interfaz de creación de ejercicios mostrada en la figura 22.

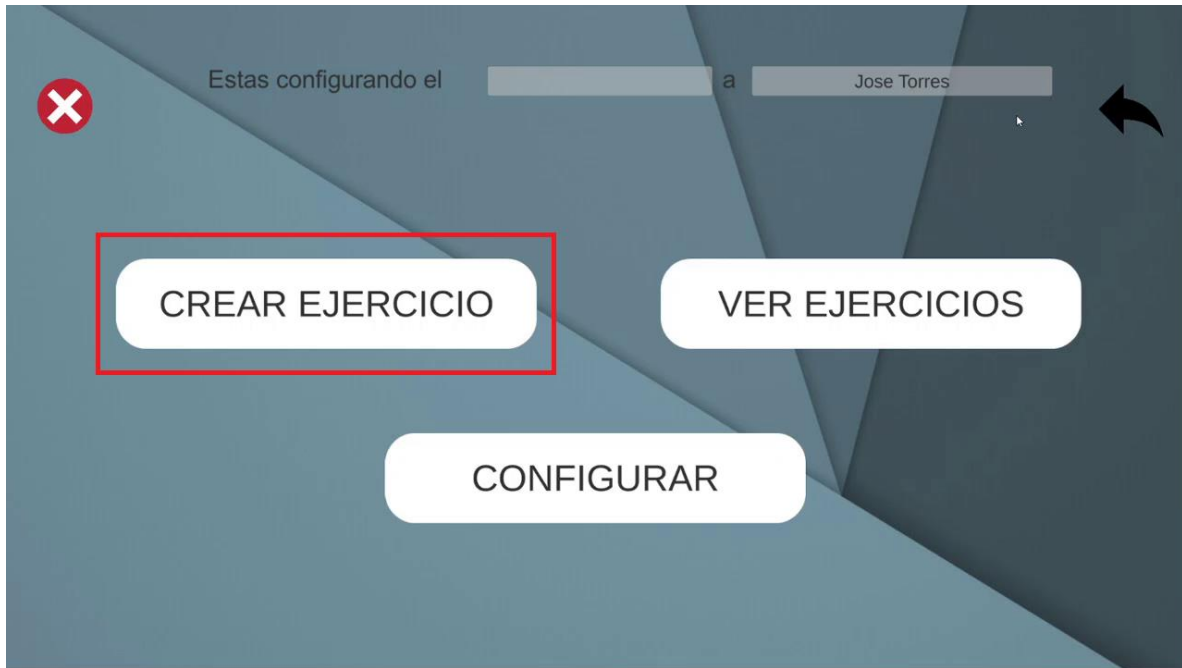


Figura 21.



Figura 22.

- (1) Campo de texto nombre ejercicio: Se puede establecer un nombre característico para el ejercicio a configurar, puede contener números letras y símbolos. Ejemplo: Ejercicio1.
- (2) Campo de texto fecha inicio: En este campo debe ir la fecha de inicio para que el paciente pueda efectuar el ejercicio, el formato es Día/Mes/Año. Ejemplo: 01/11/2021
- (3) Campo de texto fecha fin: En este campo debe ir la fecha de finalización del ejercicio, hasta cuando tiene el límite para hacer el ejercicio. Ejemplo: 05/11/2021
- (4) Botón crear: Una vez establecida la información de manera correcta en los campos (1)(2) y (3) se debe presionar este botón para crear el ejercicio, se confirmara si se creó de forma exitosa si aparece en el campo (5) y volverá automáticamente a la interfaz de ejercicios.
- (5) Campo de visualización ejercicio actual: En este espacio se ve el ejercicio actual asignado y configurándose para el paciente en el campo (6)
- (6) Campo de visualización paciente actual: En este espacio se ve el paciente actual asignado
- (7) Botón Volver: Este botón permite regresar a la interfaz anterior.

Una vez creado el ejercicio el campo de ejercicio actual, resaltada en color rojo en la figura 23 aparecerá, si se quieren generar más se debe hacer el mismo proceso. En dado caso que se desee trabajar con otro ejercicio y no con el que se muestra actualmente deberá ir a la sección “Ver ejercicios” y a la subsección “Establecer ejercicio” de este manual.

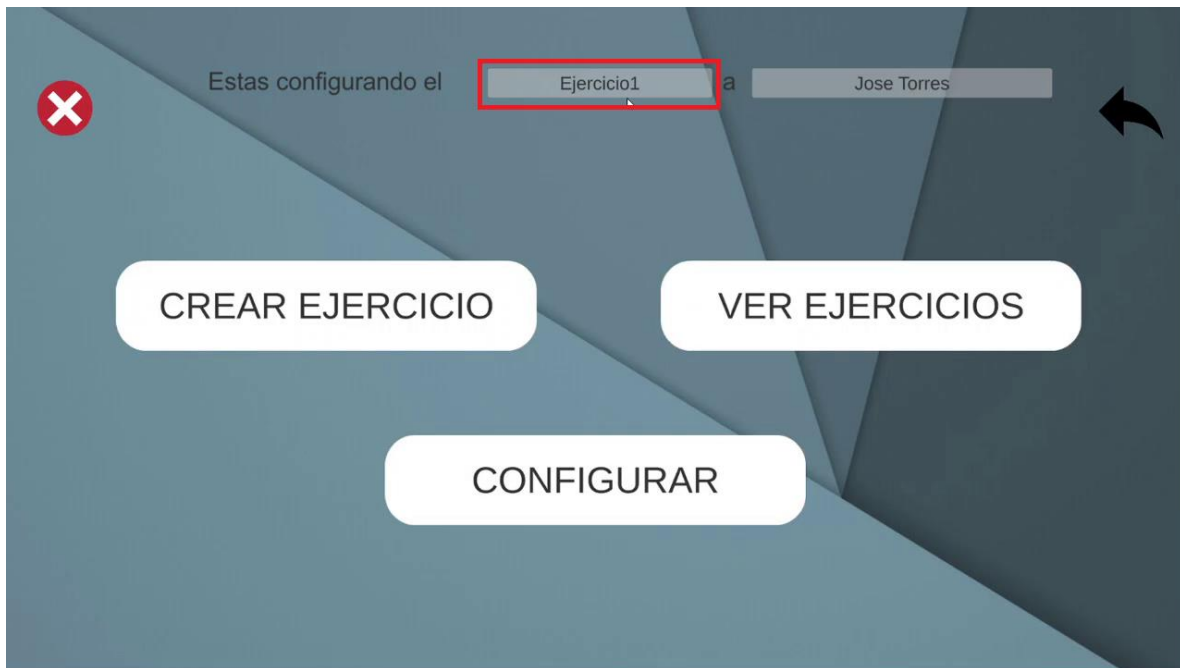


Figura 23.

NOTA: Para tener en cuenta el límite de ejercicios que se pueden crear en simultaneo es 7, eso quiere decir que por paciente existirá un máximo de 7 ejercicios, estos pueden contar con rangos de fechas para su ejecución variables como se verá a continuación. Debido a esto, lo que se recomienda es hacer uso de los 7 ejercicios que se pueden crear para establecer un cronograma de ejercicios y una vez terminado los 7 en el lapso de tiempo del primer al último ejercicio, generar el reporte en el módulo de reportes para tener la información de estos y así poder ya crear otros 7 eliminando y creando o sobreponiendo en los que ya hay.

Ver ejercicios

Para ver los ejercicios se debe presionar el botón “Ver ejercicios” resaltado en la figura 24. Se llevará a la interfaz de visualización de ejercicios figura 25, donde se podrá establecer para configurarlo como el ejercicio actual o también se podrá modificar las fechas de estos o eliminarlos.

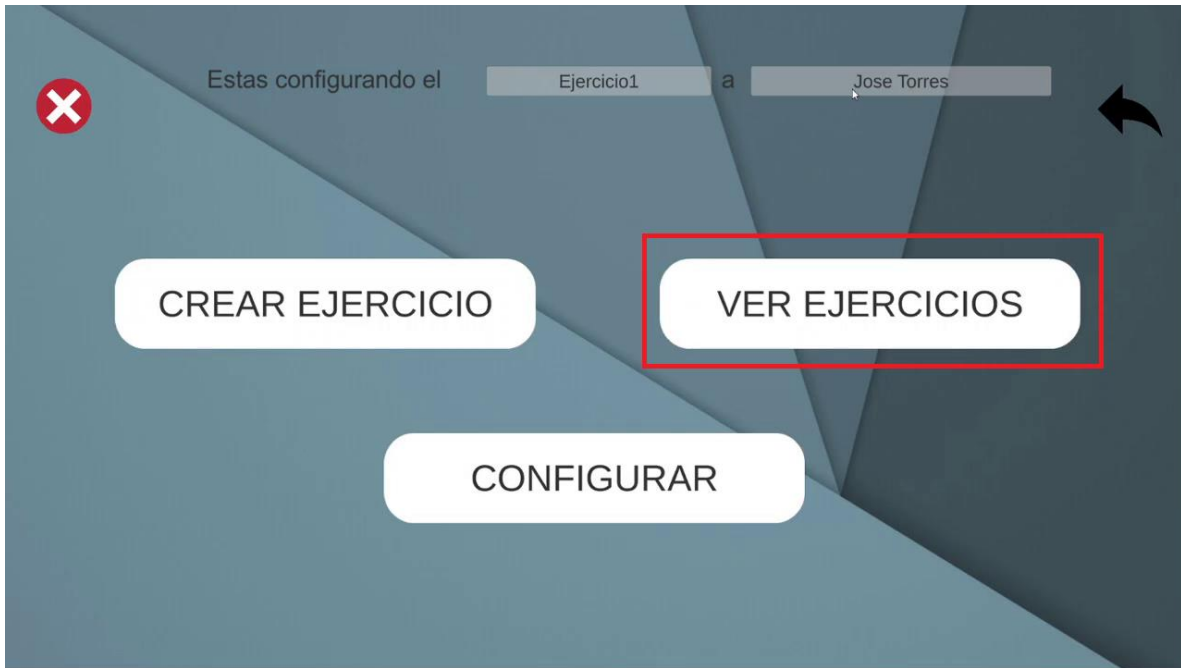


Figura 24.



Figura 25.

Establecer ejercicio

Para establecer un ejercicio como el ejercicio actual a configurar se debe seleccionar el ejercicio, si ha sido seleccionado este aparecerá resaltado de un color anaranjado como se ve en la figura 26, después oprimir el botón a la derecha llamado establecer resaltado en la figura 26 de color rojo. Al hacer esto aparecerá en la parte superior el nombre del ejercicio establecido figura 27, y se habrá retornado a la interfaz de ejercicio.

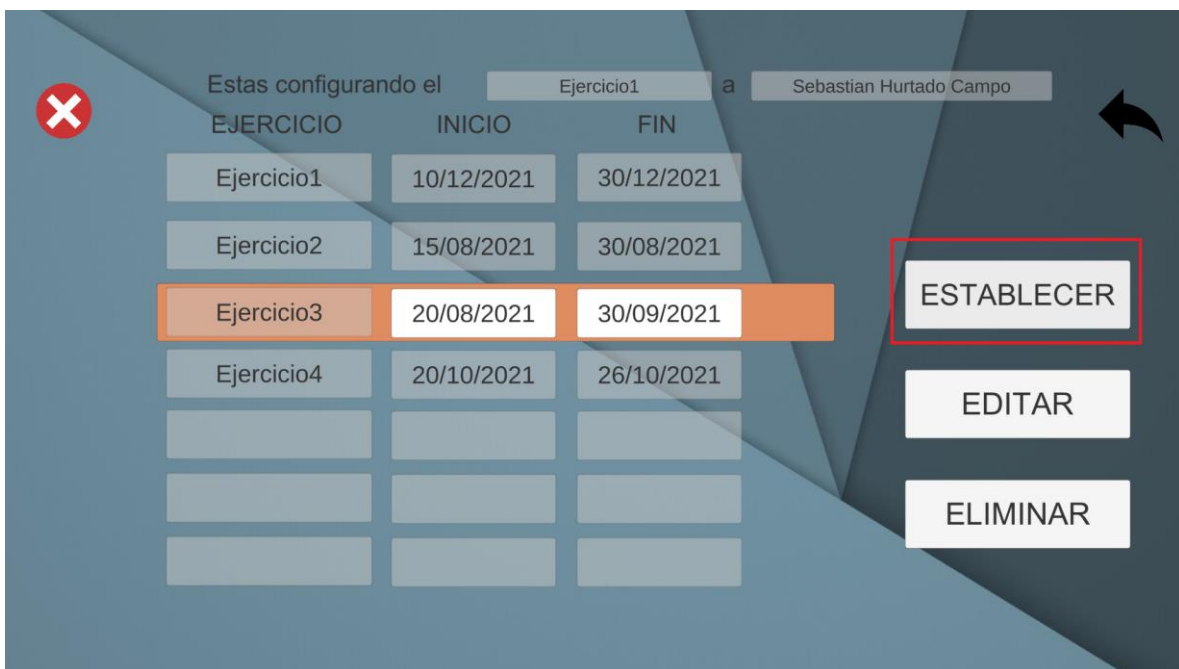


Figura 26.

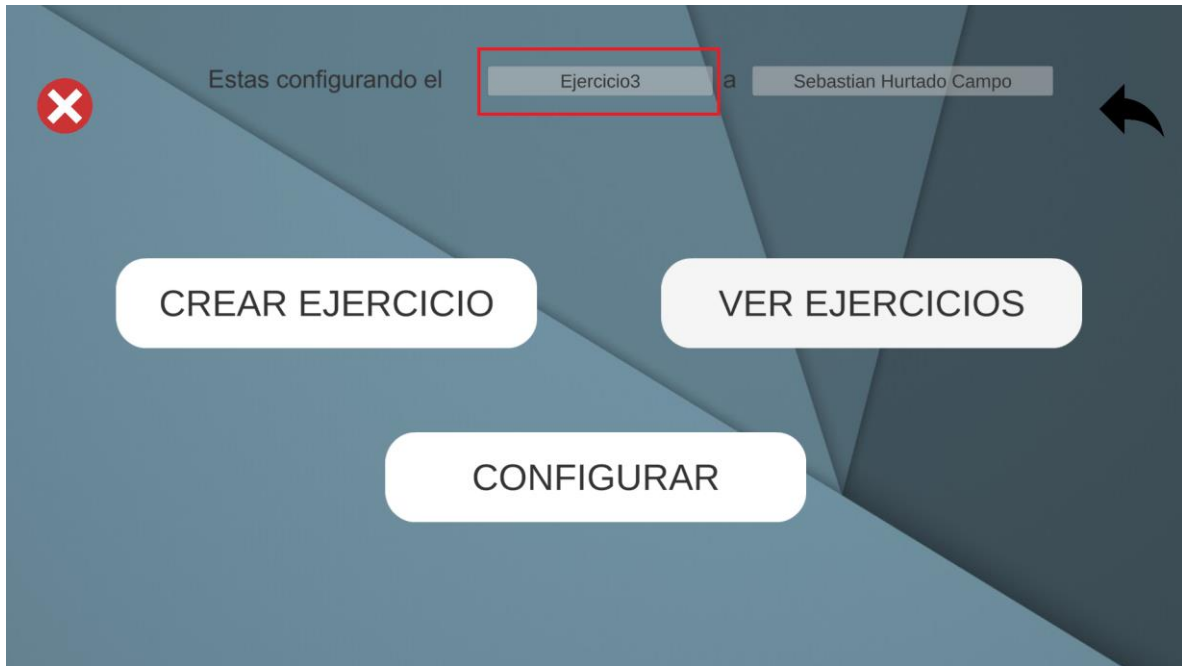


Figura 27.

Modificar fechas ejercicio

Para modificar las fechas de un ejercicio se debe estar en la interfaz de visualización de ejercicios, luego presionar el ejercicio que queremos modificarle la fecha y que salga resaltado en un color anaranjado figura 28. Después los campos de fecha inicio y fecha fin aparecerán habilitados para su modificación por lo que se podrá hacer el cambio de fecha pertinente como se ve en la figura 28. Una vez hecho el cambio se presiona el botón "Editar" a la derecha y este ya se habrá modificado volviendo de nuevo a la interfaz de ejercicios automáticamente.

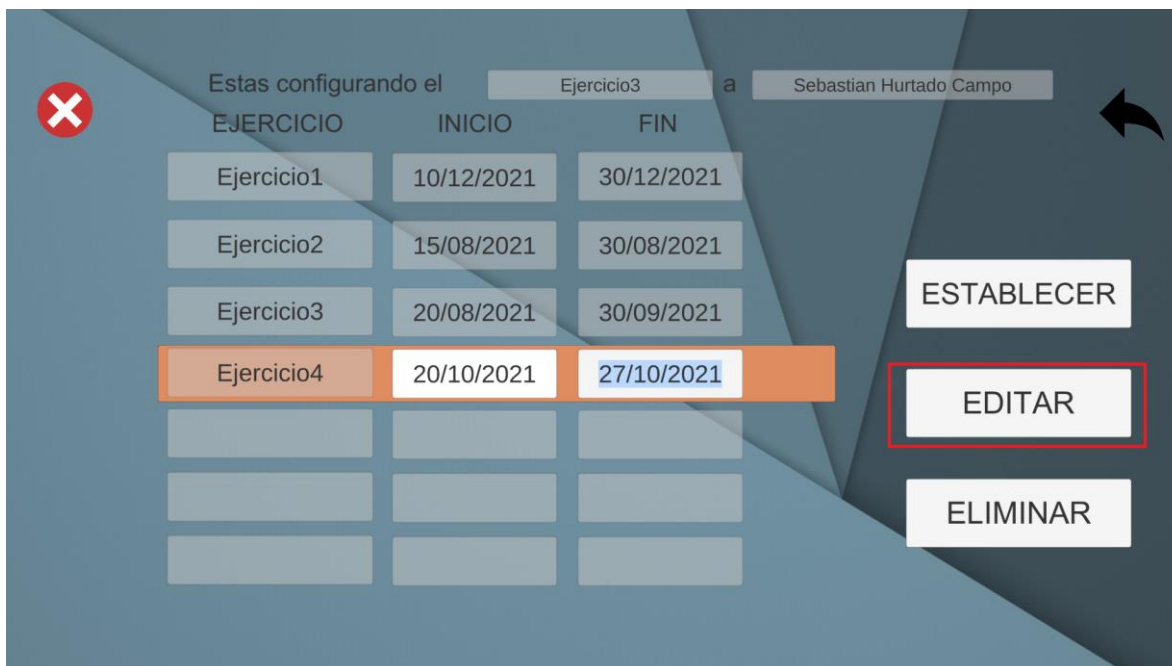


Figura 28.

Eliminar ejercicio

Para eliminar un ejercicio se debe estar en la interfaz de “Ver ejercicios”, seleccionarlo y que quede resaltado de anaranjado como en la figura 29, después oprimir el botón a la derecha de “Eliminar” resaltado en rojo en la misma figura, esto hará que se elimine el ejercicio y se vuelva a la interfaz de ejercicios automáticamente.

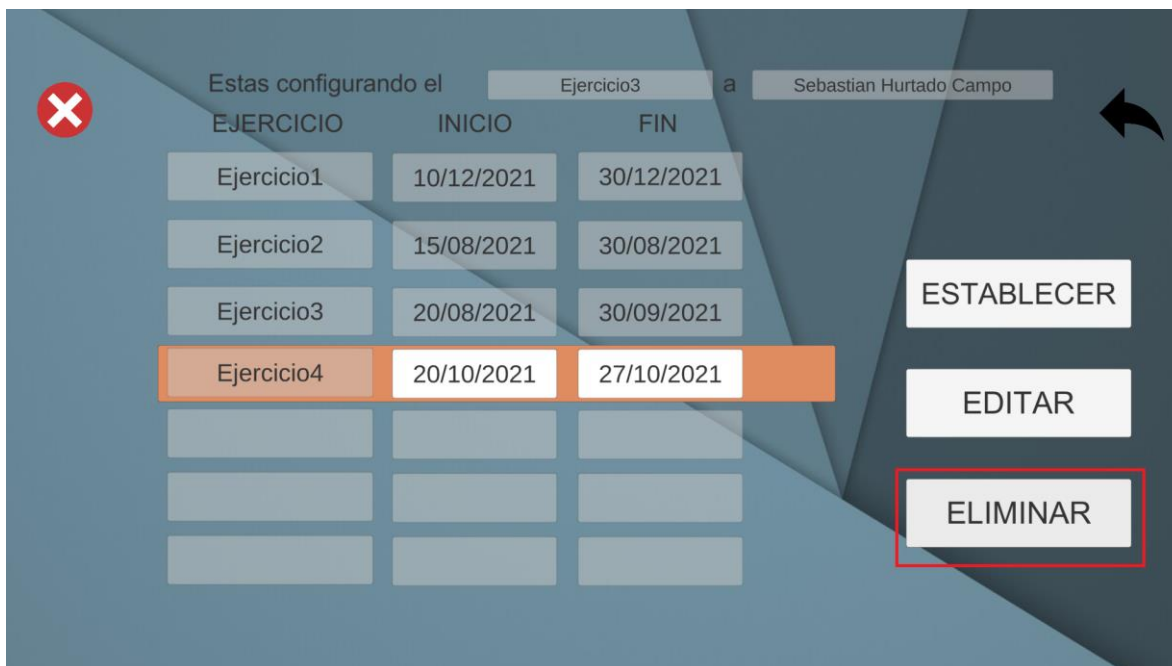


Figura 29.

Interfaz de configuración de ejercicio

Para acceder a la interfaz de configuración se debe asegurar que en los campos de visualización de ejercicio actual y paciente actual se encuentren los datos que desea en la figura 30 se marcan estos dos campos con un recuadro rojo. Si es correcto deberá presionar el botón de configurar marcado en verde en la figura 30 que lo llevará a la interfaz de configuración figura 31.

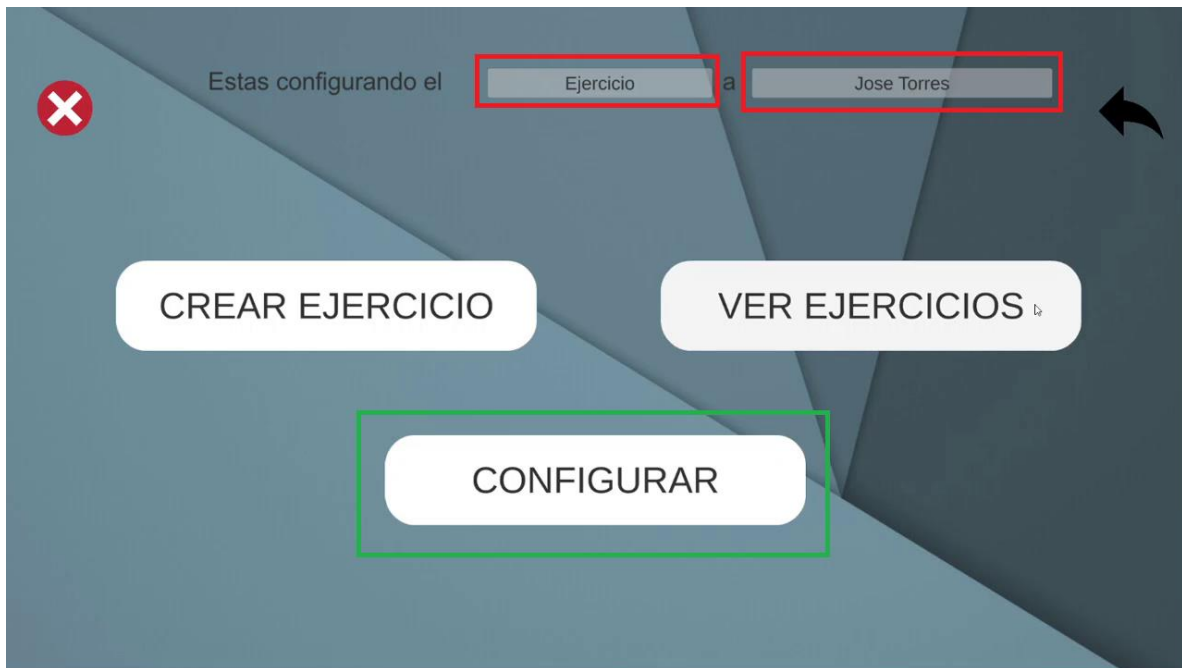


Figura 30.



Figura 31.

En la interfaz de configuración hay dos botones, el botón "Ir a estante" y el botón "Configurar reto", se recomienda que para la configuración del ejercicio

en este caso “Ejercicio1” para “Jose Torres” en el ejemplo que se presenta en Figura 31, se establezca el reto o la misión antes de configurar el espacio en realidad aumentada para su búsqueda, aunque no existe problema si se hace, al contrario. Se recalca que un ejercicio está compuesto por la parte de configuración del entorno que se encuentra en la sección “Ir a estante” y la parte de establecimiento de objetivo en “Configurar reto”

Configurar reto u objetivo

En la interfaz de configuración de ejercicio, se debe presionar el botón configurar ejercicio que se encuentra marcado en un recuadro rojo en la figura 32, después de presionarlo lo llevara a la interfaz de configuración de reto en la figura 33.



Figura 32.



Figura 33.

- (1) Campo de visualización objeto: En este campo aparecerá el nombre del objeto que desea establecer para buscar presionando uno de las imágenes del banco de objetos en (3) un ejemplo de llenado está en la figura 34
- (2) Campo de visualización cantidad: En este campo aparecerá el numero/cantidad de objetos del tipo (1) que se deberían buscar presionando uno de los números de la cantidad en (4) con un máximo de 12 por objeto, un ejemplo de llenado está en la figura 34
- (3) Banco de objetos: Estas imágenes reflejan cada objeto que puede estar en el estante, al presionar una se establece en (1) el nombre del objeto para asignarle una cantidad a buscar
- (4) Banco de cantidad: Estos números permiten establecer la cantidad del objeto (1) que se haya escogido, el máximo valor es 12 y el mínimo es 1.
- (5) Botón guardar: Para establecer los datos asignados en (1) y (2) como el primer reto, se debe oprimir este botón. Después de presionarlo,

puede cambiar los datos de (1) y (2) y asignar otro reto, todo esto para el ejercicio actual. Para visualizar los retos o búsquedas guardadas debe dar en el botón de ver retos (6)

(6) Botón ver retos: Al presionarlo permite ver los retos o búsquedas asignadas al ejercicio actual, cambiando la interfaz a la de la figura 35.

(7) Botón volver: Al igual que los demás botones de volver, permite regresar a la interfaz anterior.



Figura 34.

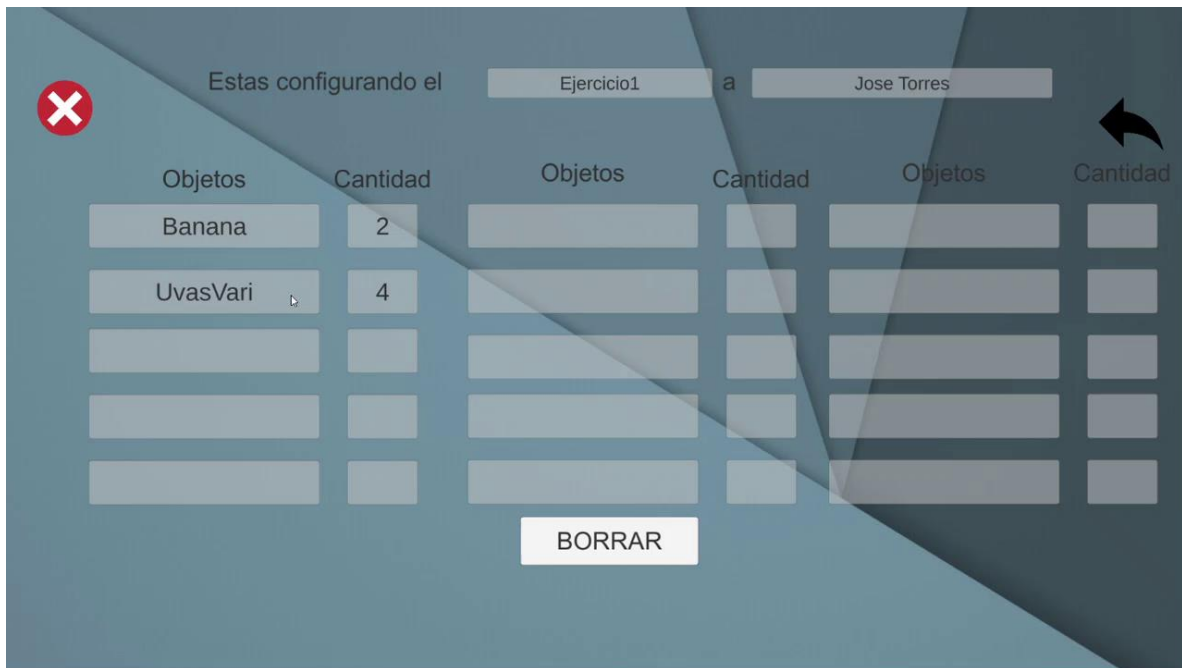


Figura 35.

En la figura 35, se ven los retos configurados para el “Ejercicio1” se deben encontrar 2 Bananas y 3 UvasVari (Siendo las variantes de las uvas moradas, o sea las verdes), dado el caso que estén mal y se quieran eliminar debe hacer lo de la siguiente subsección “Eliminar reto y cantidades máximas”

Eliminar reto y cantidades máximas

Para eliminar el reto, se debe oprimir el botón “Borrar” que se encuentra en la figura 35, eso hará que los retos establecidos sean eliminados y se pueda generar de nuevo las búsquedas. El máximo de búsquedas que se puede establecer por ejercicio es de 15 pero se advierte que si se generan búsquedas de objetos con cantidades que no se encuentran en el estante no podrá cumplirlas con éxito.

Configurar estante o entorno AR

Para configurar el estante o entorno AR, en la interfaz de configuración de ejercicios se debe presionar el botón “Ir a estante” que esta resaltado en un cuadro rojo en la figura 36, esto llevara a la interfaz de configuración del estante en la figura 37. en donde esta cada uno de los lados del estante que se pueden configurar. El proceso es el mismo para cada uno de los lados, simplemente que los lados laterales que son el rojo y el amarillo cuentan con 6 y 8 espacios respectivamente mientras que el frontal y posterior cuentan con 12.

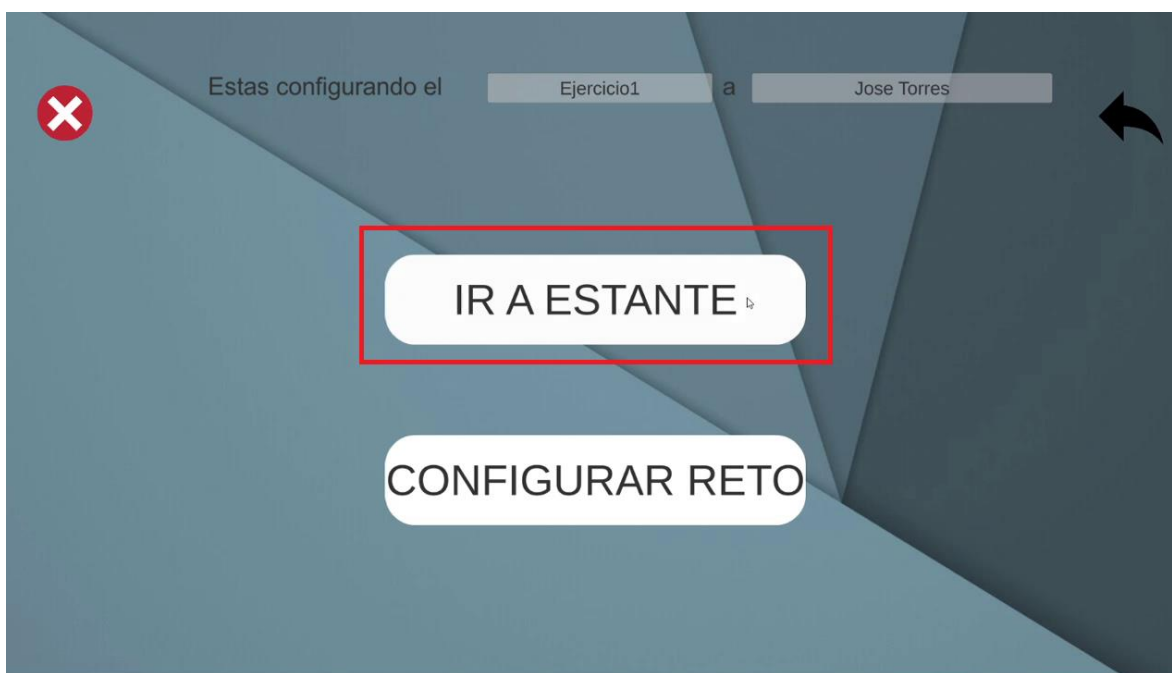


Figura 36.

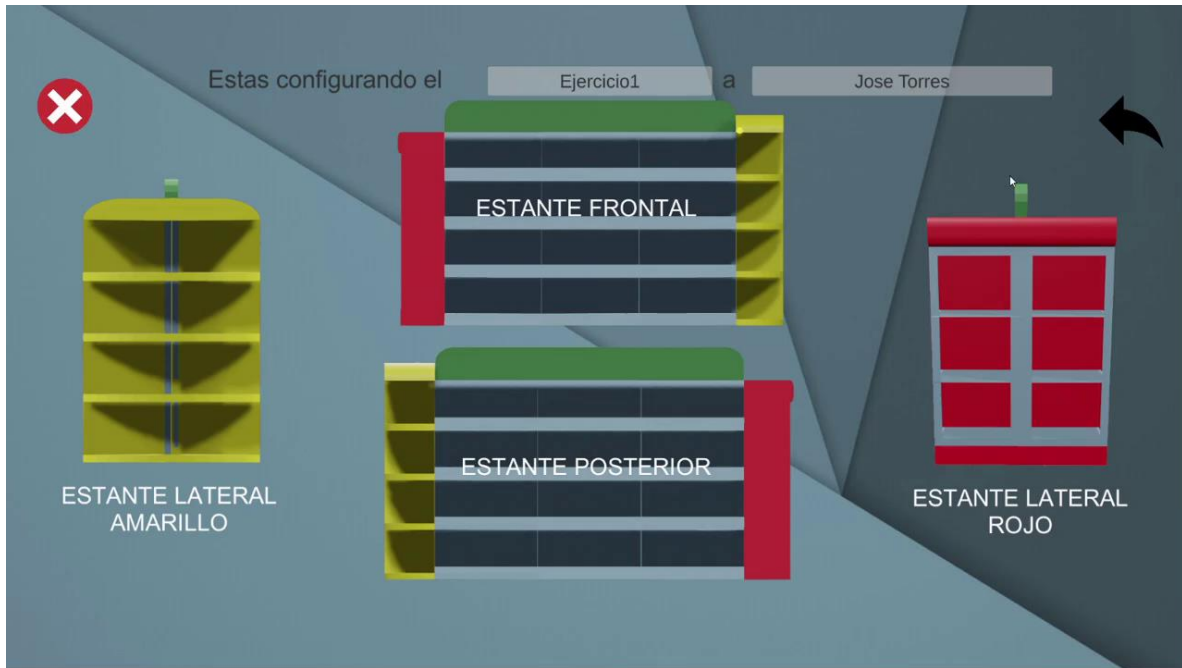


Figura 37.

Una vez estemos en la interfaz del estante, deberemos seleccionar una de las caras o lados de este. Para el ejemplo se hará con la frontal (Es lo mismo que con las demás solo que el lateral amarillo y el lateral rojo cuenta con menos espacio) por lo que se presiona el lado y llevara a la interfaz de ese lado específico para su configuración figura 38.

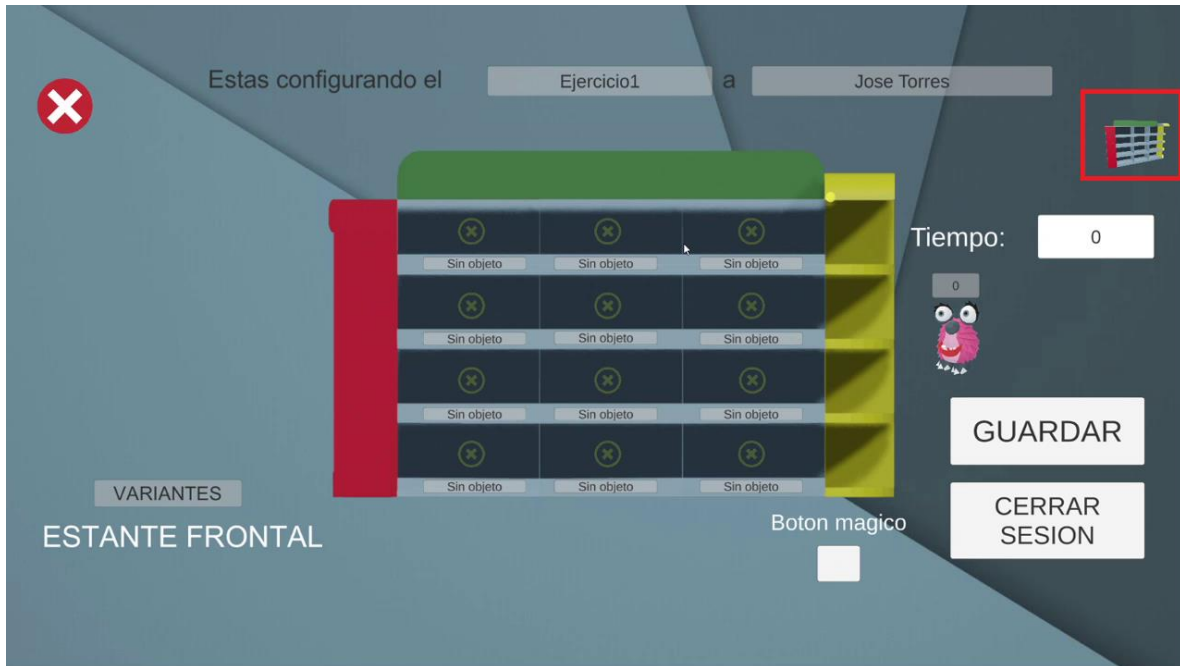


Figura 38.

Moverse entre las caras

Para moverse entre las caras del estante, se debe presionar en la miniatura del estante ubicada en la esquina superior derecha, con la cual podremos acceder a la interfaz del estante y acceder a otro lado. Este icono esta resaltado en la figura 38, de color rojo.

Establecer objeto en un espacio

Para establecer un objeto en un espacio del estante se deberá presionar el espacio, puede ser cualquiera de los 12 (Se presiona en la equis verde o si ya cuenta con un objeto, en el objeto que se desea cambiar) y esto desplegara un cuadro con los objetos que se pueden poner figura 39, si se desea poner un objeto variante o sea uno que tenga una variación del mismo, se debe oprimir el botón "Variantes" resaltado en rojo en la misma figura, mostrando los objetos variantes como la figura 40 para asignarlo al espacio presionado

anteriormente, si se quiere volver a los objetos normales solo volver a presionar el botón “Variantes”.

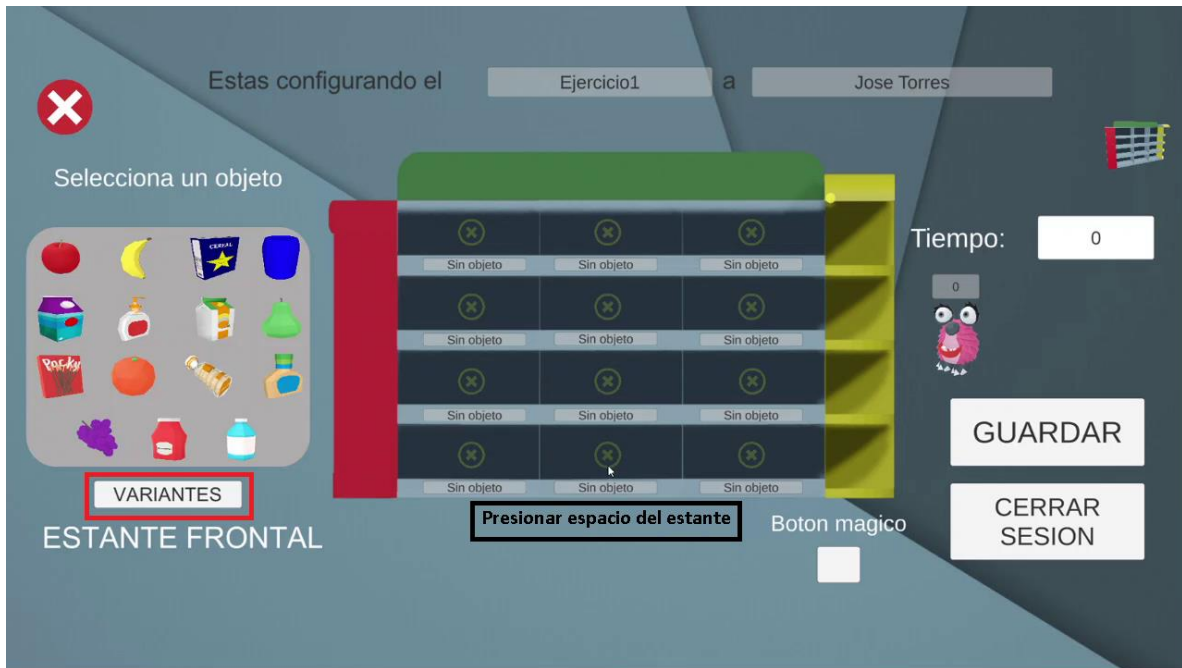


Figura 39.

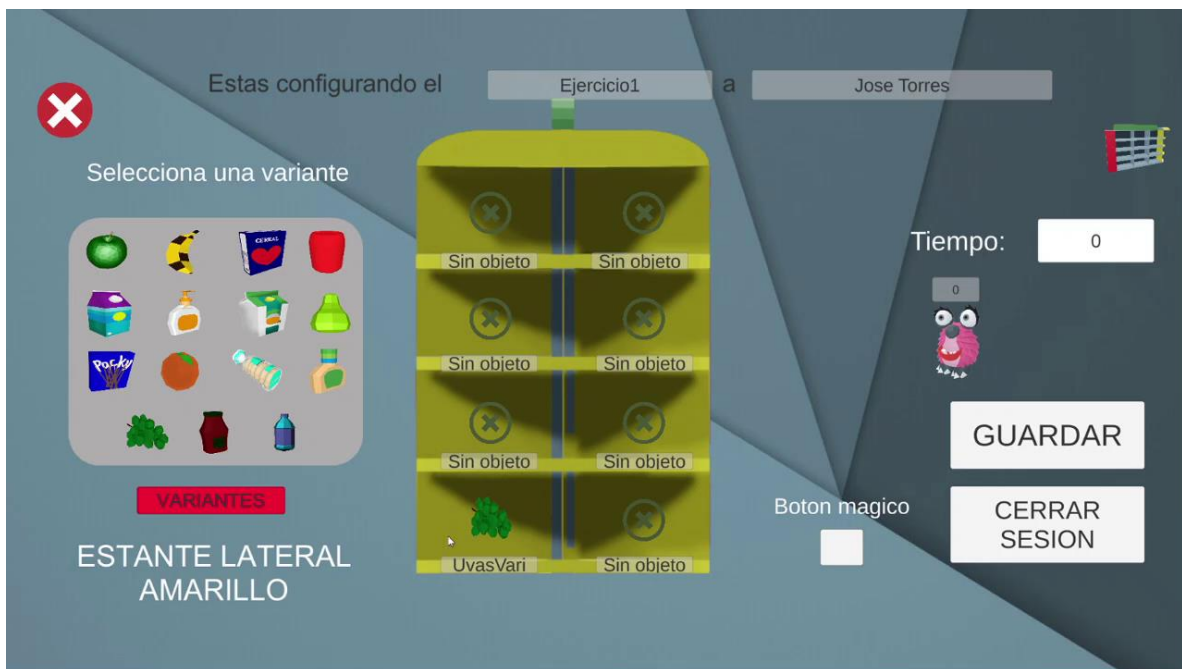


Figura 40.

Botón mágico o de auto llenado

En cada uno de las caras o lados del estante se cuenta con un “Botón mágico” al presionarlo se encenderá de un color respectivamente a el lado en el que se encuentre. Tener este botón encendido implica que se pueden presionar varios espacios para establecer un mismo objeto. En resumen, lo que se debe hacer es, encender el botón mágico figura 41, presionar cada espacio que se desee llenar con un mismo objeto figura 42, y para finalizar presionar el objeto para que se llene con ese objeto figura 43, quedando así todo lleno como en la figura 44.

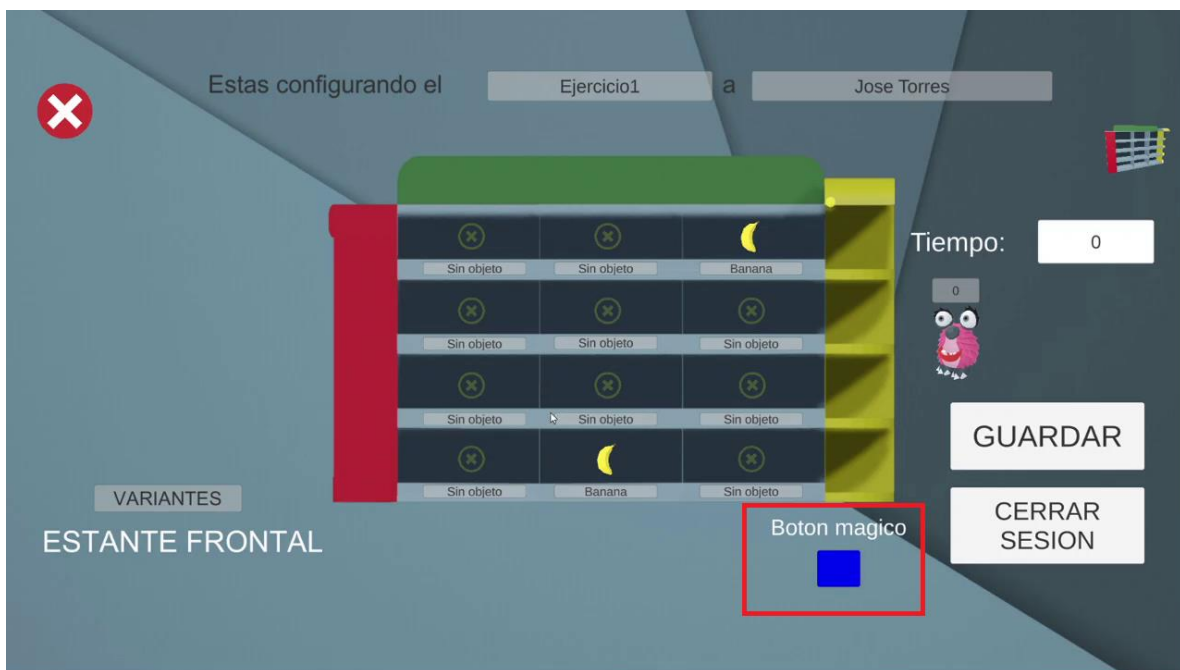


Figura 41.

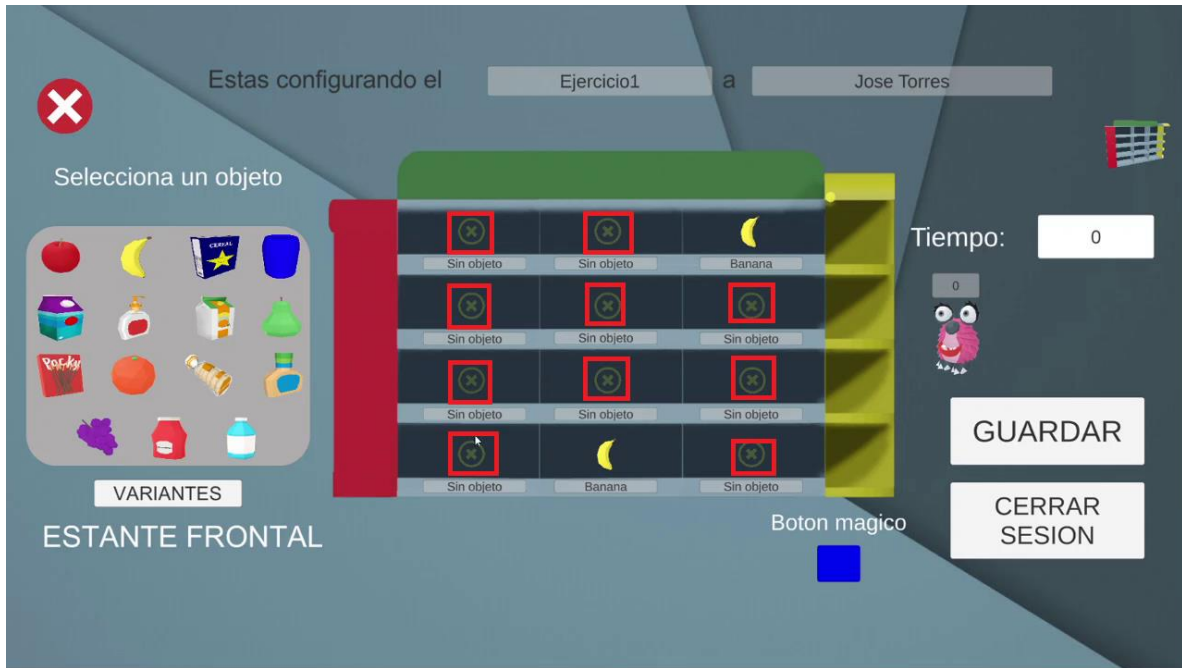


Figura 42.

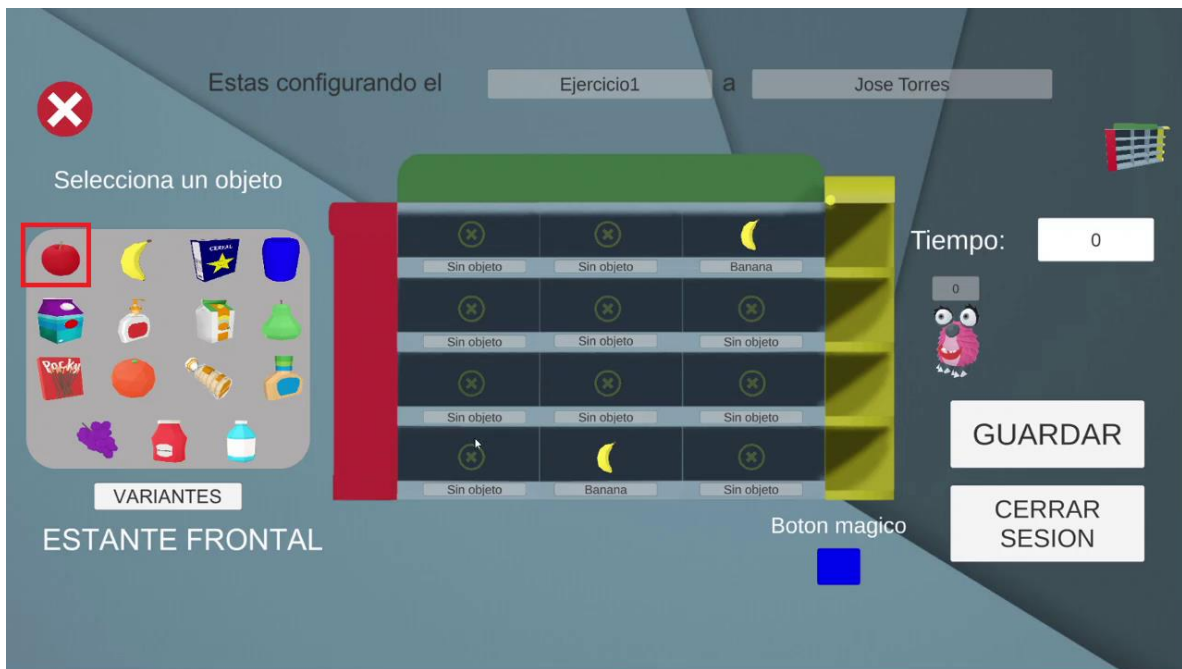


Figura 43.

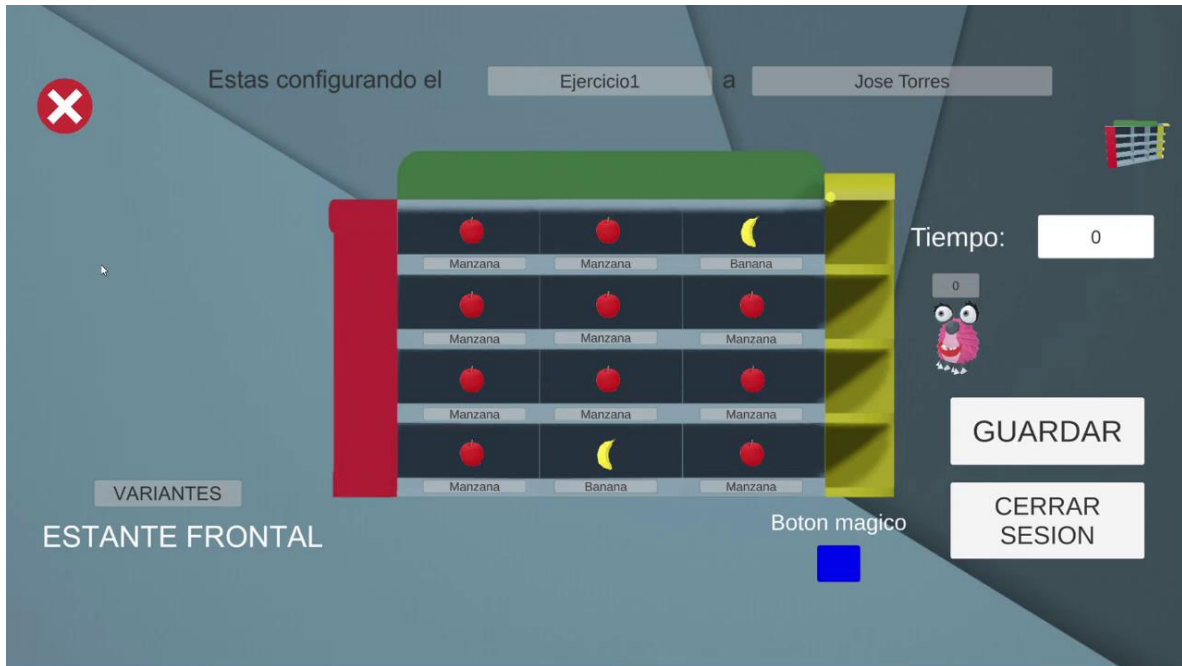


Figura 44.

Ocultando el monster

El monster es un personaje que puede tener 4 ubicaciones según se elija, este siempre se va a incluir en el reto para su cumplimiento, por defecto aparece en la ubicación 1 pero si esto se quiere cambiar se debe presionar el monster que se resalta en la figura 45, desplegando las otras opciones de ubicación figura 46, para su selección.

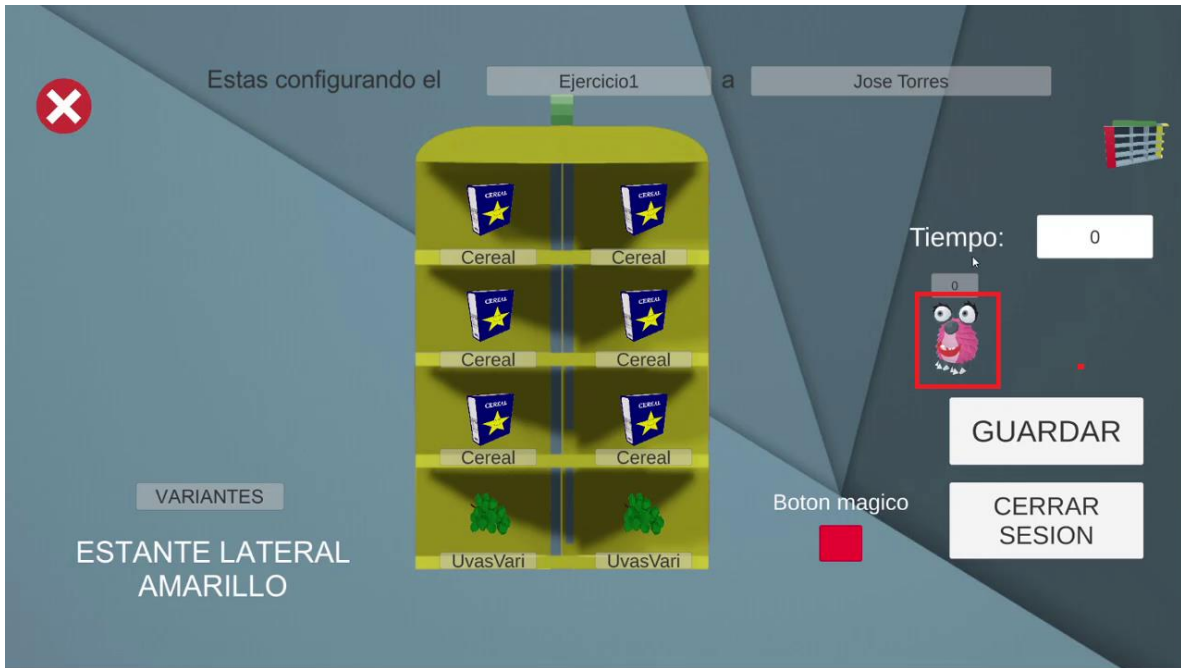


Figura 45.

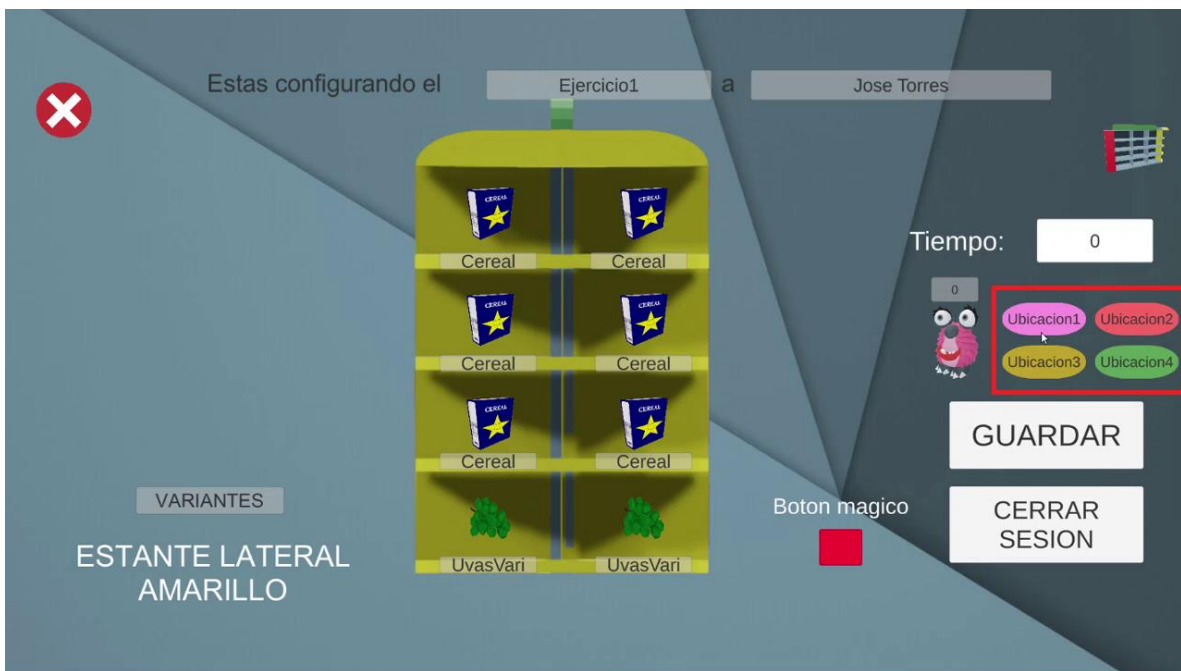


Figura 46.

Ubicaciones del monster

Ubicación 1: Frontal, Segunda fila, tercera columna

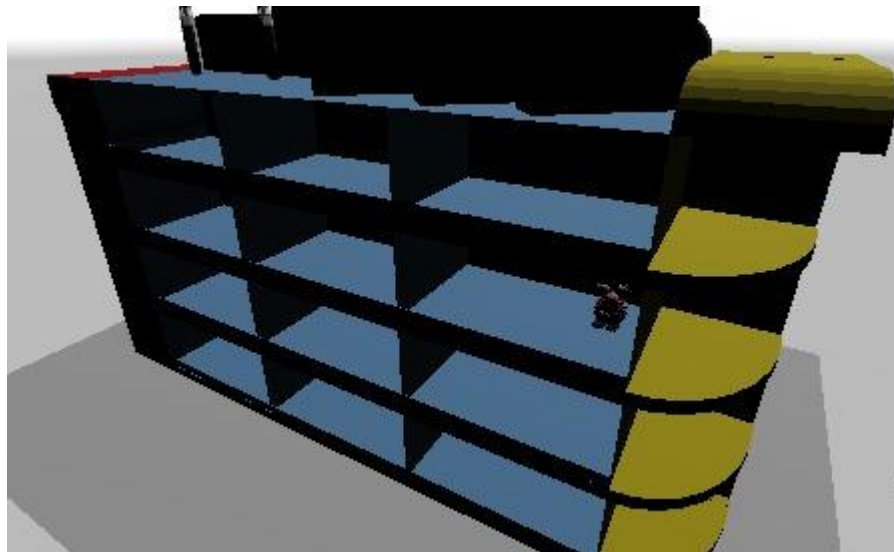


Figura 47.

Ubicación 2: Posterior, Cuarta fila, tercera columna



Figura 48.

Ubicación 3: Encima del estante



Figura 49.

Ubicación 4: Lateral rojo, segunda fila, primera columna

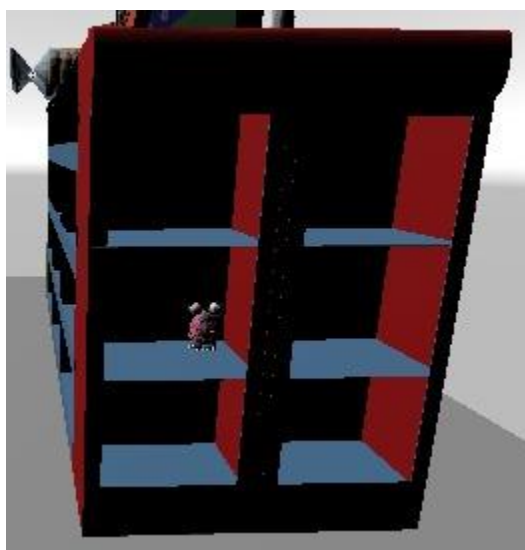


Figura 50.

Establecer tiempo para realizar la búsqueda

El tiempo para establecer la búsqueda es diferente al tiempo del ejercicio, el tiempo del ejercicio se agenda con fechas siendo el tiempo en el que el paciente podrá realizar la actividad configurada las veces que se deseen. Mientras que el tiempo de la búsqueda o la actividad es el tiempo en el que podrá realizar el reto por cada intento, este se establece en segundos en la parte de "Tiempo" que se resalta en rojo de la figura 51. Un ejemplo: 120, esto serian dos minutos para hacer la búsqueda.

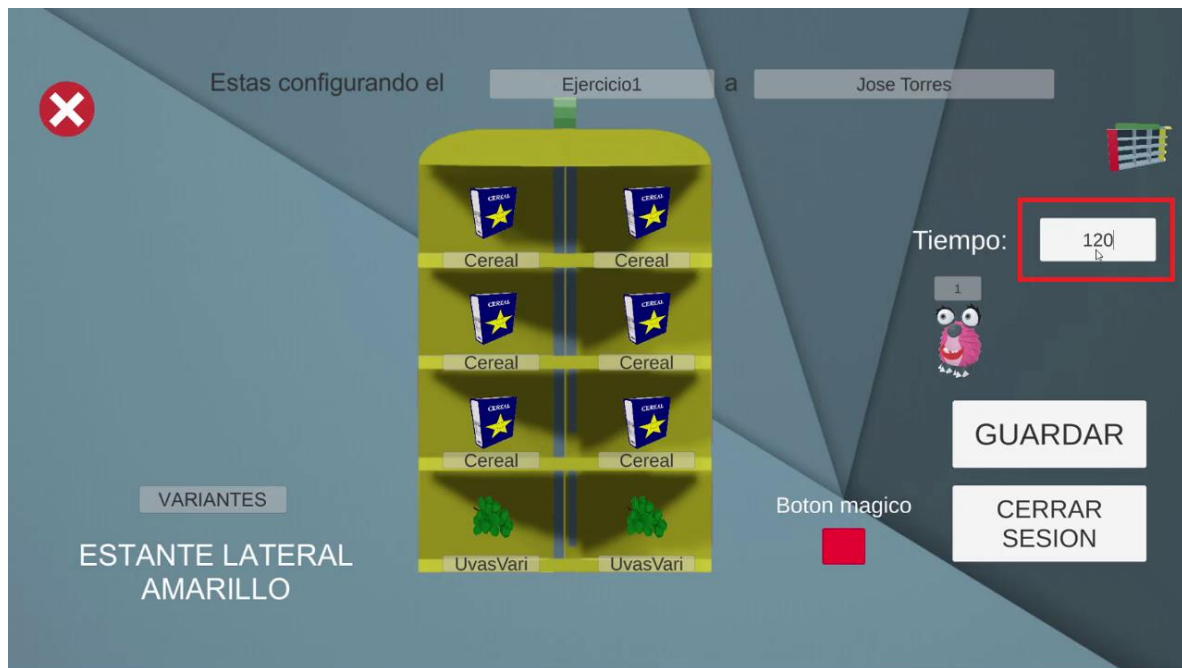


Figura 51.

Guardar configuración

Para guardar la configuración del ejercicio, solo se debe dar en el botón guardar figura 52, presentado en cualquiera de las caras. Con solo presionar uno se guarda la configuración de todas las caras, el monster y el tiempo de búsqueda. Una vez presionado se puede cerrar sesión o se puede retornar para

seleccionar otro ejercicio, cambiar el reto o salir de la aplicación con equis blanca.

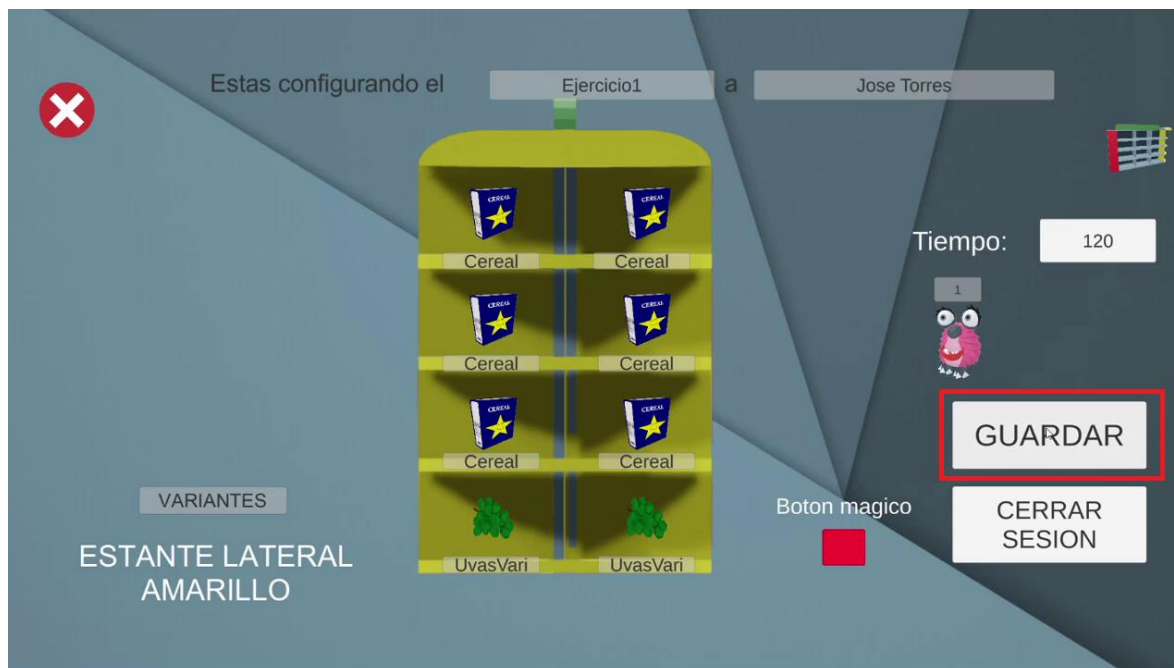


Figura 52.

APLICACIÓN MÓVIL

La aplicación móvil es la encargada de proyectar el entorno de MiniSuper en realidad aumentada, es donde podrá realizar los ejercicios de búsqueda configurados en el módulo de configuración por el terapeuta. Esta aplicación solamente funciona en dispositivos Android de gama media alta.

Instalar aplicación móvil en el dispositivo

Teniendo ya hechas las acciones previas se cuenta con una carpeta “2-AplicacionMovil” que contara con un archivo “MiniSuperAR.apk” ese archivo es la aplicación. Lo primero que hay que hacer es conectar nuestro dispositivo por cable USB al computador, en el dispositivo deberá aparecer la opción de transferir archivos Figura 53 o algo similar, presionamos esa opción y ya se deberá visualizar en el computador más específicamente en la sección de “Este equipo”, el dispositivo, en este caso es “Rosmer” y se presenta como una smartphone figura 54.



Figura 53.

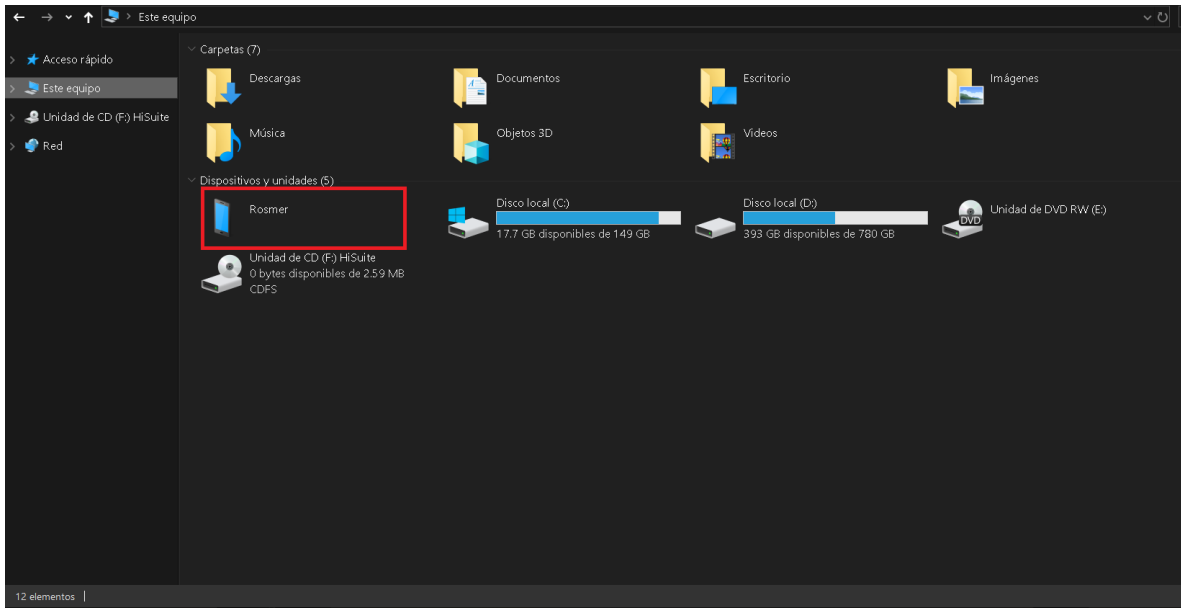


Figura 54.

Una vez este nuestro dispositivo conectado, necesitamos copiar la carpeta o el .apk al celular, en este caso será la carpeta así que se hace clic derecho, copiar en la carpeta “2-Aplicación Móvil” Figura 55, se abre el dispositivo móvil en el computador Figura 56, se escoge un lugar en este caso va a ser “Memoria interna” se da doble clic y se pega en un espacio en el que no se interactúe con otra carpeta Figura 57 se deja hasta que la carpeta se copie en su totalidad Figura 58, Figura 59.

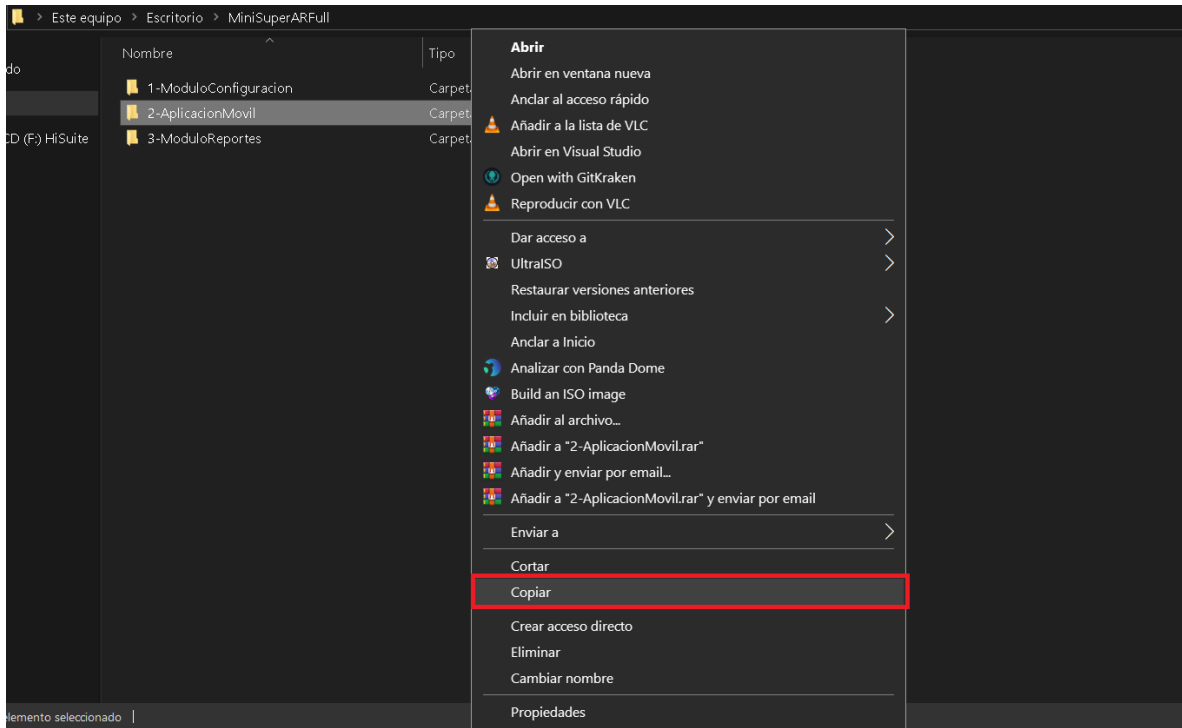


Figura 55.

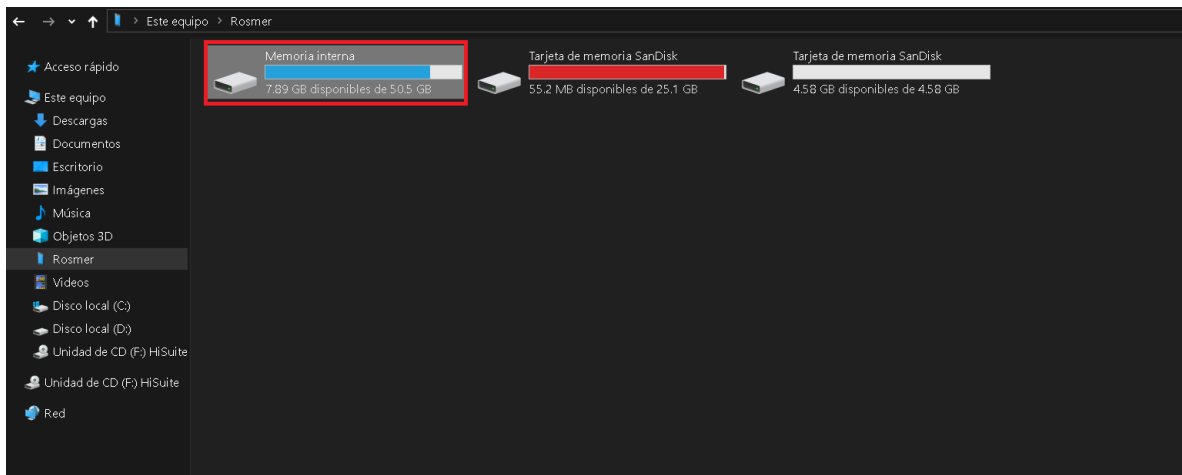


Figura 56.

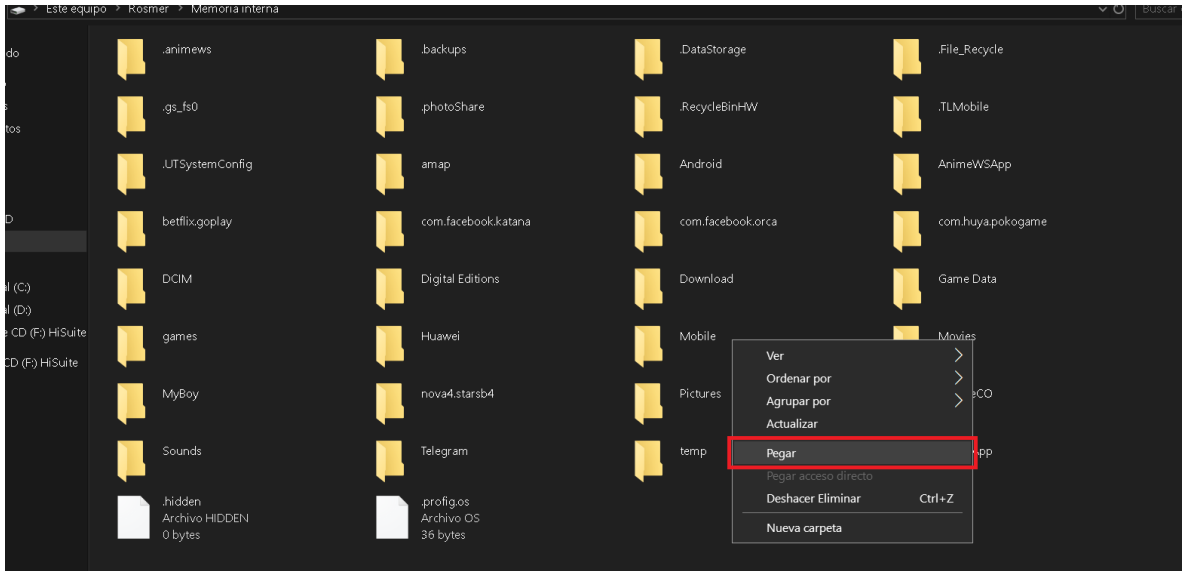


Figura 57.

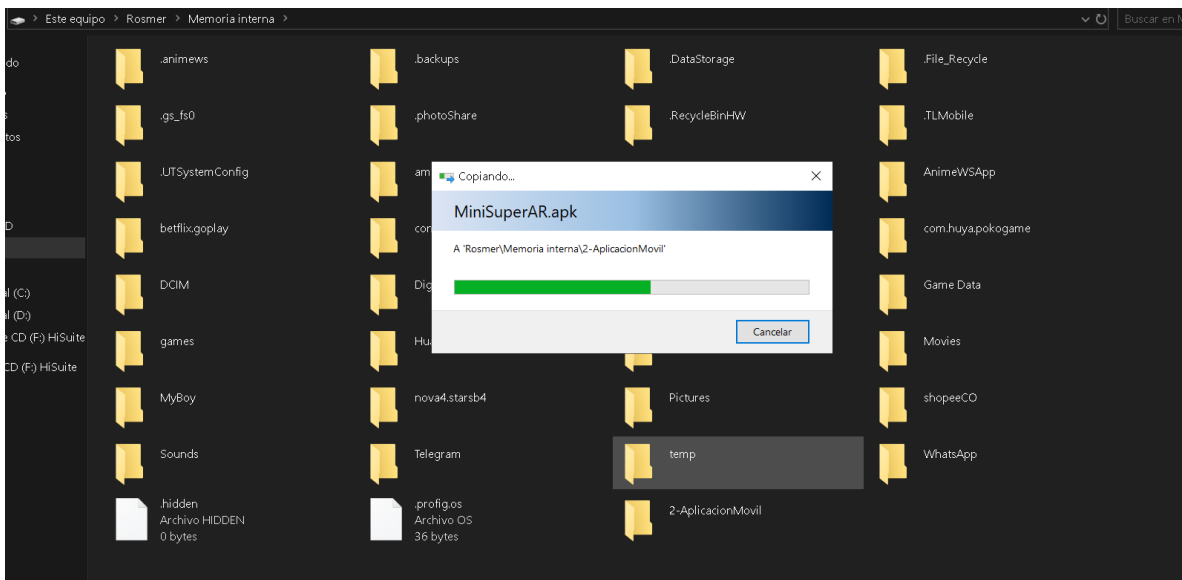


Figura 58.

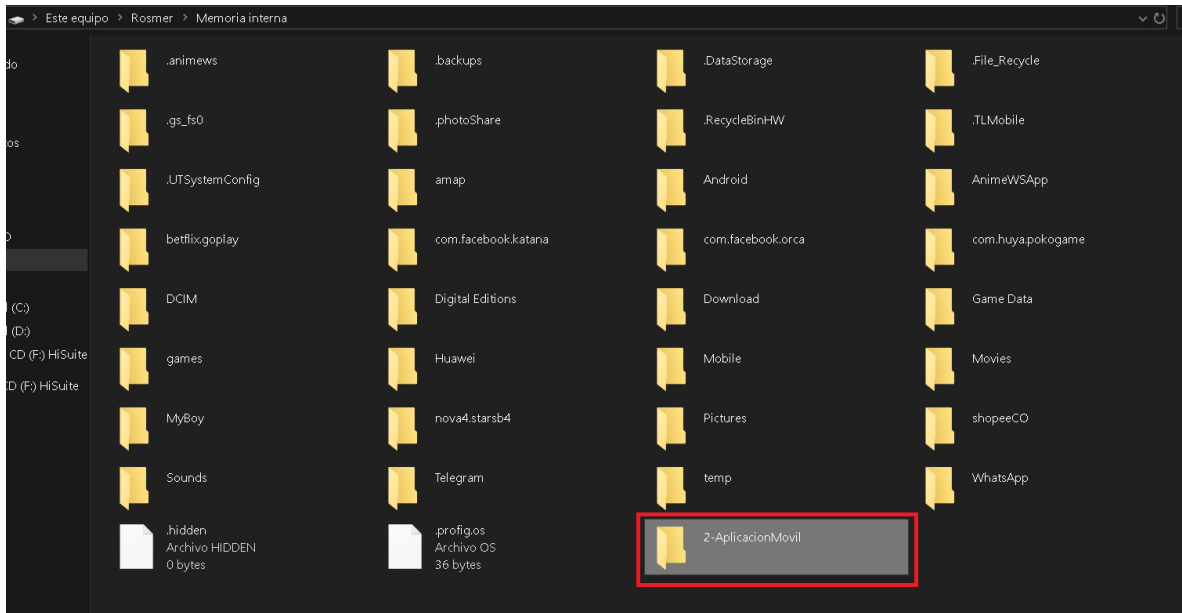


Figura 59.

Ya se puede desconectar el dispositivo del computador, debido a que se dejó en la memoria interna de este. Se dirige a la sección de archivos del dispositivo Figura 60, busca la sección de la memoria interna en este caso es “Teléfono” Figura 61, se distingue la carpeta que acabamos de copiar “2- Aplicación móvil” figura 62.



Figura 60.

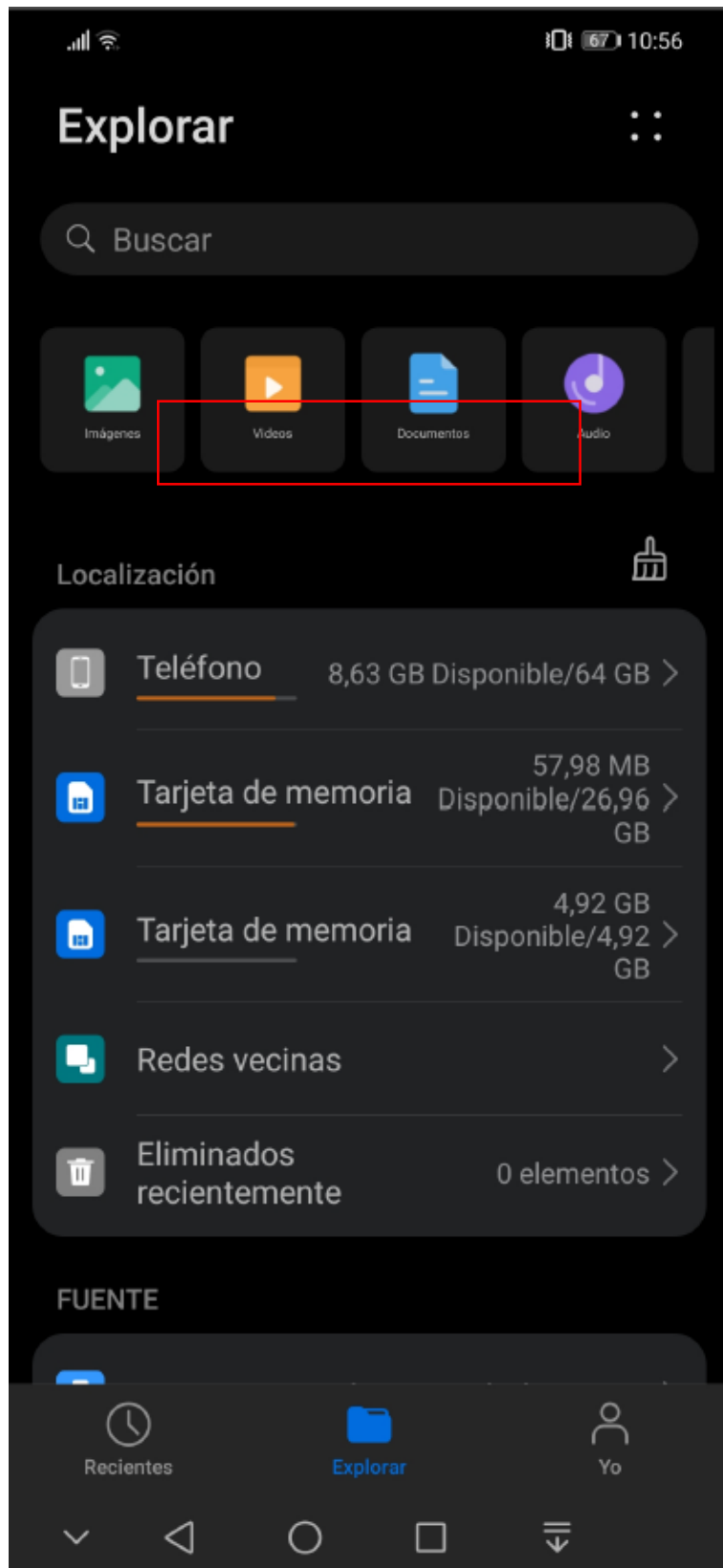


Figura 61.



Figura 62.

Abrimos la carpeta y se va a encontrar la aplicación en .apk Figura 63, la presionamos y nos va a decir si la deseamos instalar usamos el instalador de paquetes Figura 64, al no ser instalada de un tienda oficial nos va a decir que tenemos que permitir las aplicaciones de fuentes desconocidas, por lo que oprimimos en “Configuración” Figura 65.

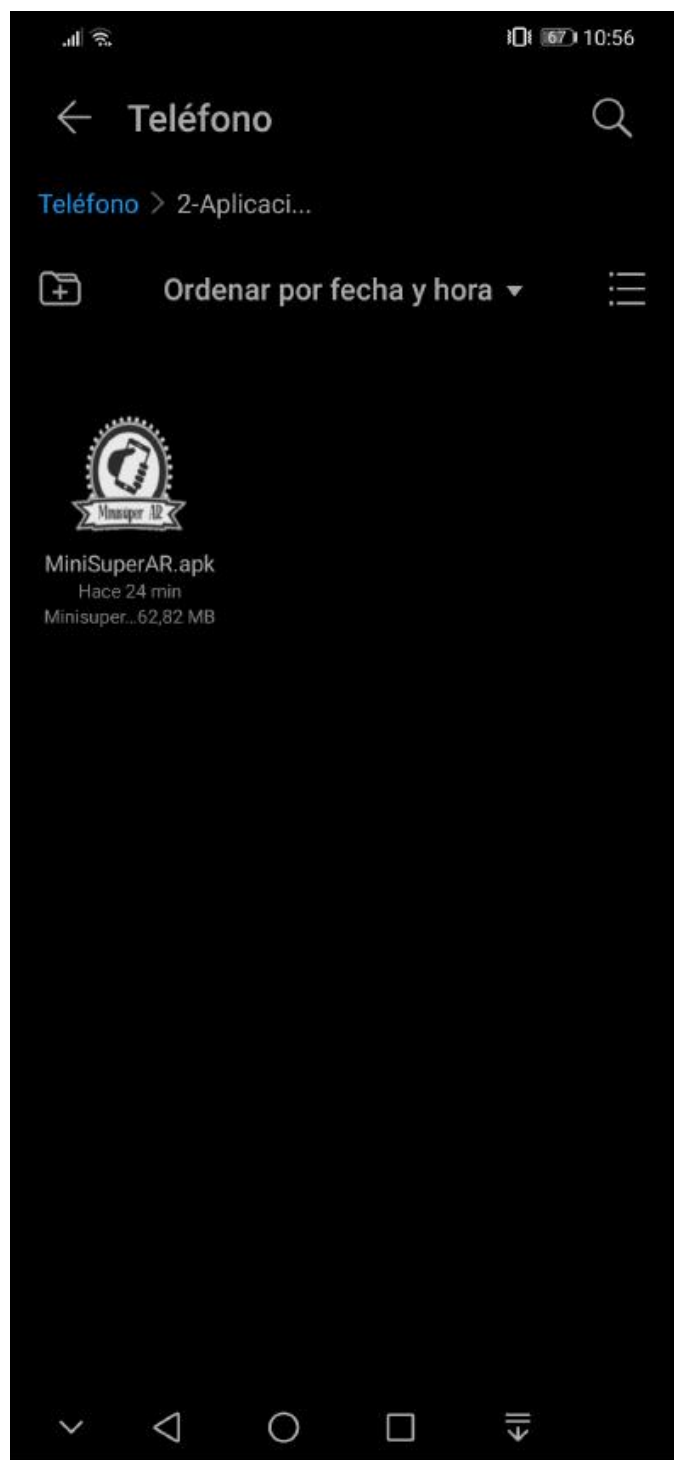


Figura 63.

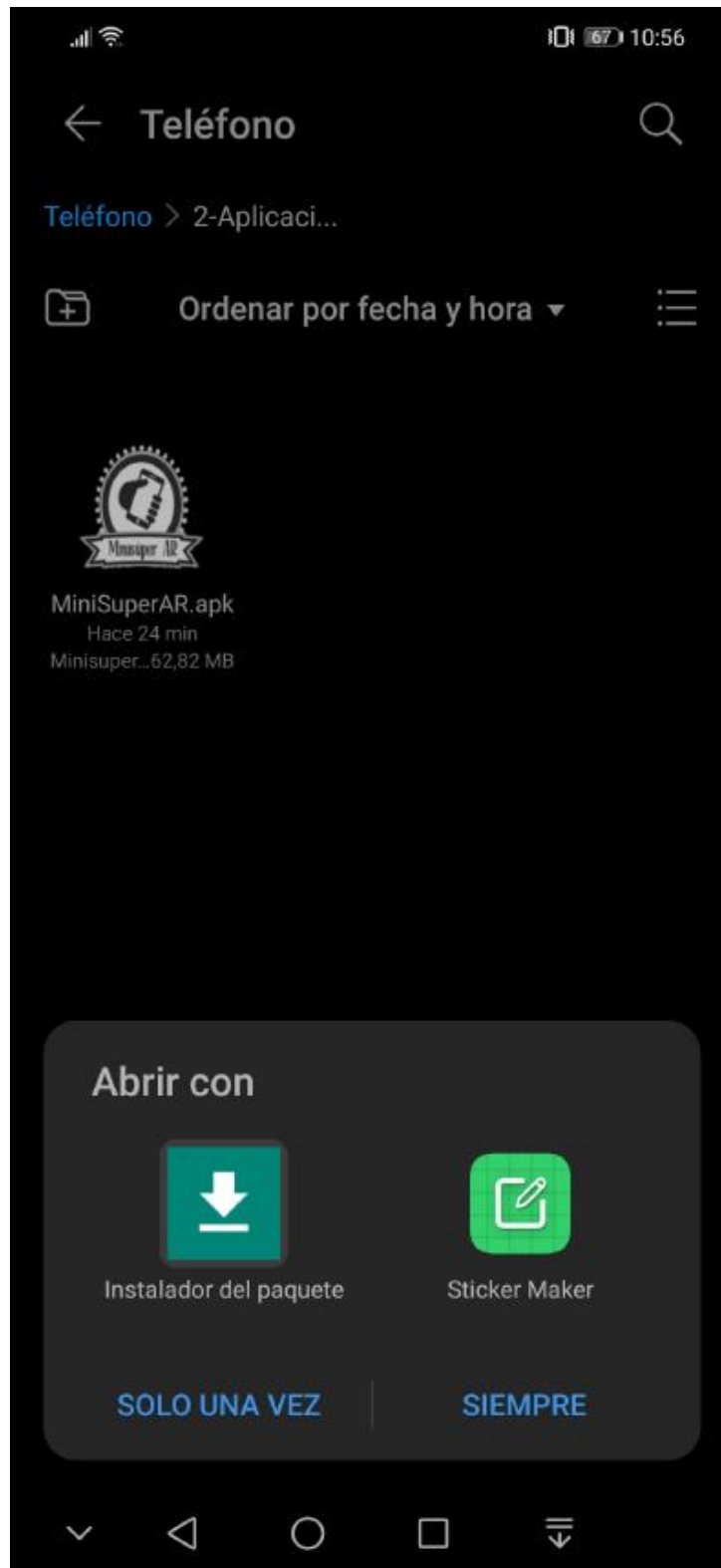


Figura 64.



Figura 65.

Una vez le demos en configuración, solo tenemos que activar la opción Figura 66, 67 y luego de hacerlo, volveremos a donde estábamos antes para continuar la instalación Figura 68.

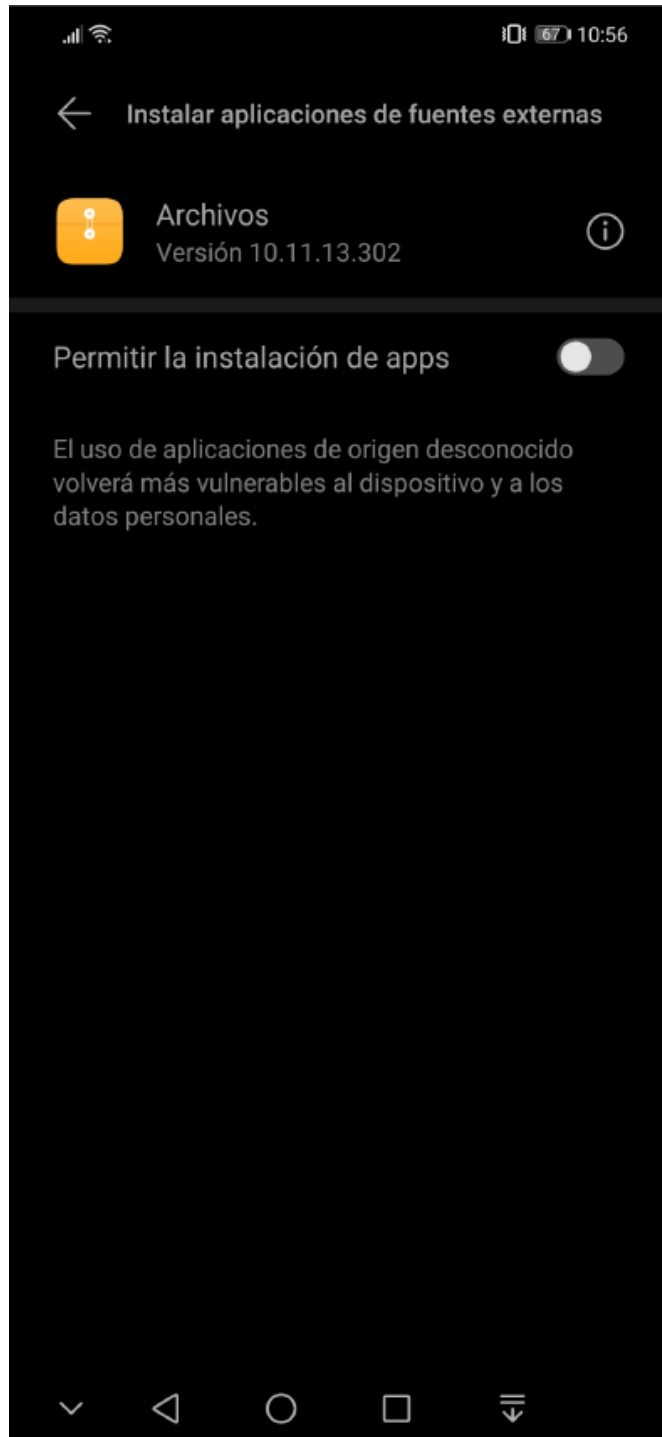


Figura 66.

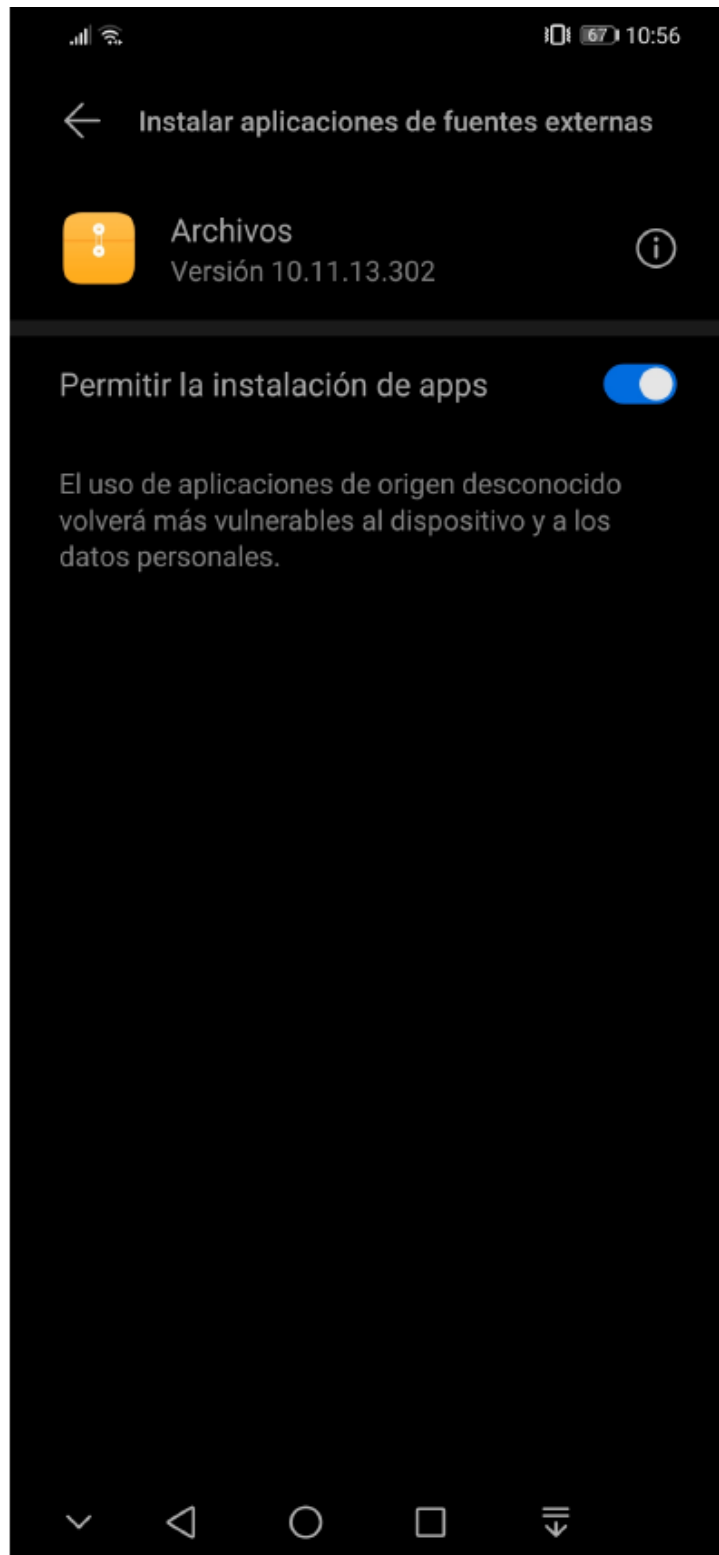


Figura 67.



Figura 68.

Ya habiéndole dado los permisos para instalar aplicaciones de fuentes desconocidas nos saldrá para instalar la aplicación Figura 68, la instalamos y esperamos a que se termine de instalar Figura 69, 70. Con esto ya poseeremos la aplicación instalada en nuestro dispositivo figura 71.



Figura 69.



Figura 70.



Figura 71.

Entrar por primera vez a la app

Al entrar por primera vez a la app, nos saldrá el logo de “Unity” Figura 72 y también saldrá una opción de primera vez en la que nos pide que aceptemos el uso de la cámara Figura 73, solo deberemos dar en el botón de aceptar y estaremos ya en la interfaz inicial de la app, Figura 74.

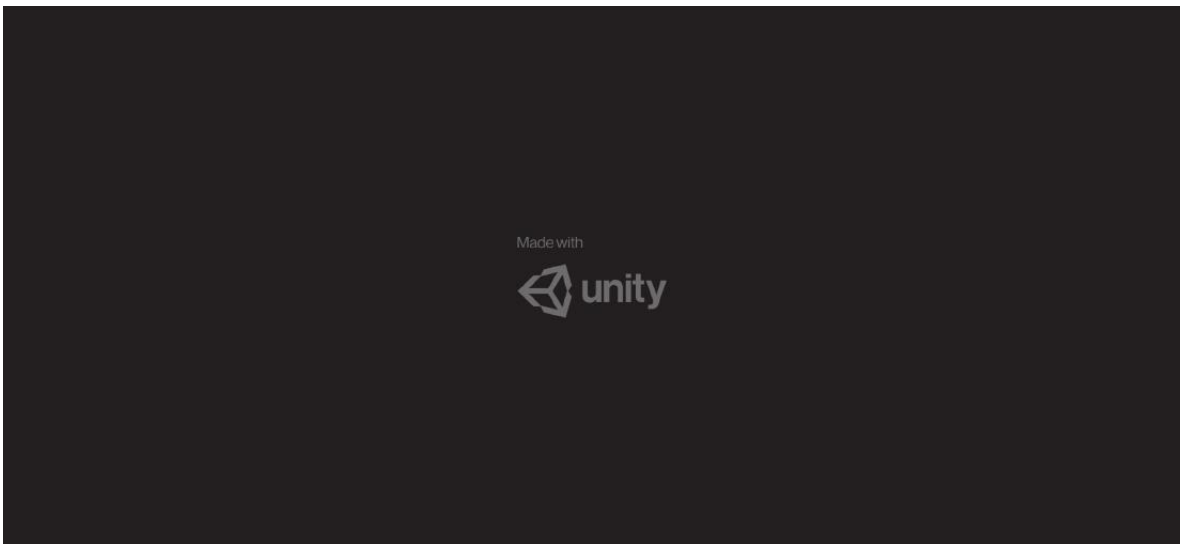


Figura 72.

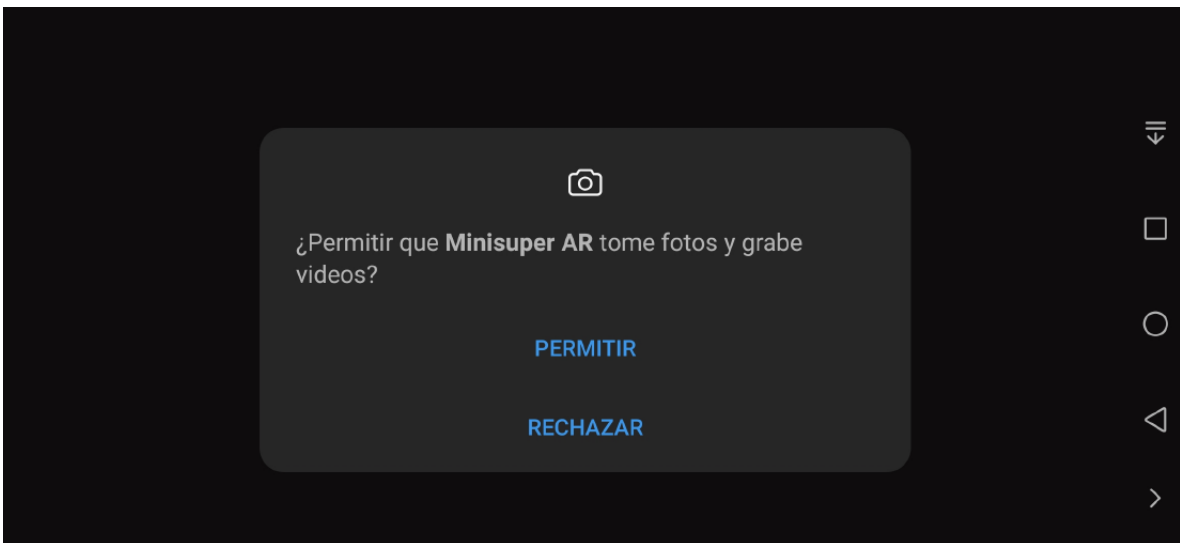


Figura 73.



Figura 74.

Iniciar sesión en la app

Para iniciar sesión en la app se presiona el botón “Jugar” de color rojo que se puede ver en la figura 74, una vez se presiona el botón nos saldrá la interfaz de inicio de sesión Figura 75.



Figura 75.

Para entrar a la cuenta, el paciente solamente deberá escribir su usuario exactamente como el terapeuta lo asigno. Ejemplo “tita” sin espacios y respetando mayúsculas y minúsculas. Al presionar el cuadro blanco con el texto de usuario, saldrá el teclado Figura 76 y podrá escribir el usuario para después oprimir el botón login Figura 77 y entrar a la interfaz principal Figura 78.

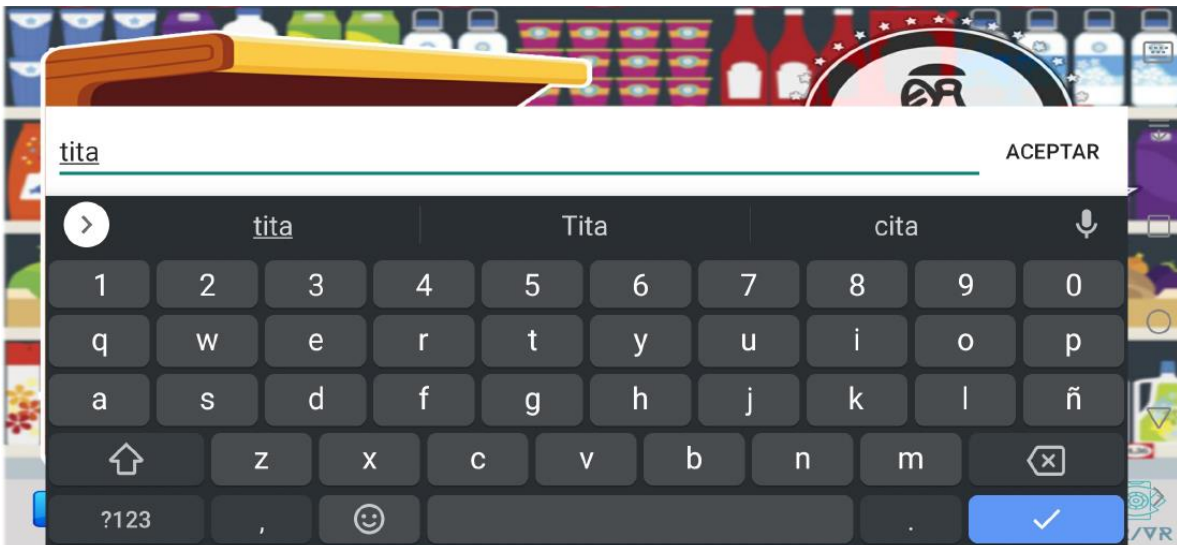


Figura 76.



Figura 77.



Figura 78.

Interfaz principal

La interfaz principal cuenta con dos secciones como se ve en la figura 79, la que esta remarcada de color rojo es la sección de información. No hay nada interactuable solo se ve el nombre y el documento, en la sección remarcada de color azul están los ejercicios (En este caso solo hay un ejercicio y se llama Peras), como en el módulo de configuración hay un máximo de 7 en sincronía, saldrán en esta sección, pero solo podrá hacer los que estén en el rango de tiempo actual, si no se ha llegado a la fecha del rango de fecha inicio y fecha fin, este no se podrá hacer pues no aparecerá el botón "Siguiente" del ejercicio al que se haya accedido como se ve en la sección del manual "Acceder al ejercicio", el botón salir nos devuelve a la pantalla de inicio.



Figura 79.

Acceder al ejercicio

Para acceder al ejercicio solo deberemos presionar el botón con su nombre Figura 78,79, al acceder aparecerá una visualización del objetivo o reto que debe realizar Figura 80. Si en la esquina inferior izquierda sale el botón “Siguiente” lo podrá realizar, si no sale este botón es porque no está habilitado pues es para una fecha posterior o una fecha que ya paso.

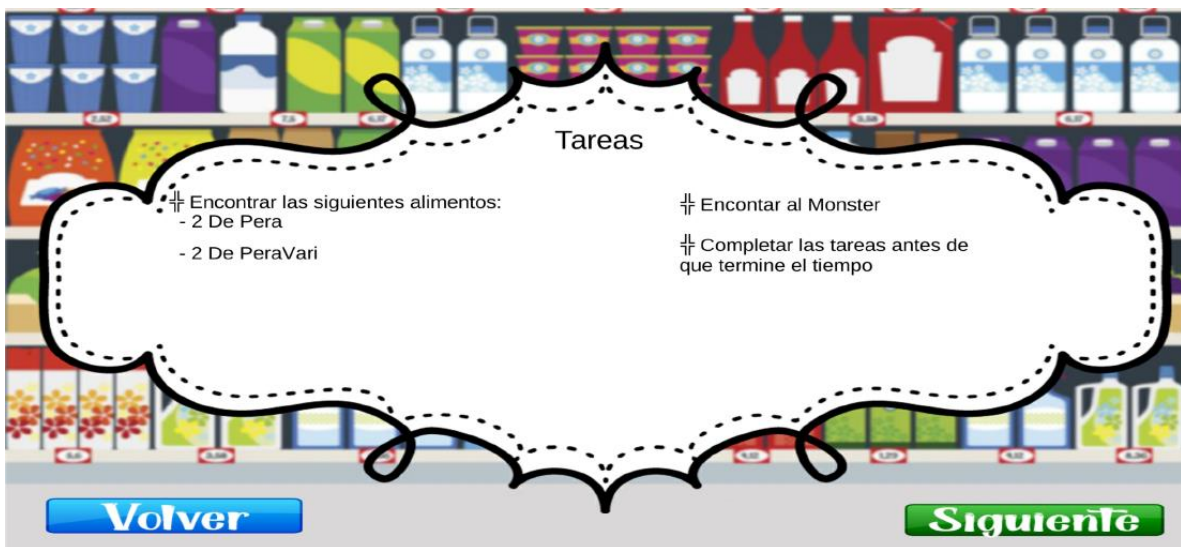


Figura 80.

En el caso del ejercicio “Peras” este si se encuentra habilitado por lo que sale el botón de siguiente para poder realizar el ejercicio.

Realizar ejercicio

Se deberá presionar el botón de siguiente como se ve en la figura 80, al presionarlo veremos que se activa la cámara (Figura 81) y ya estará casi listo para realizar la actividad propuesta (El reto).

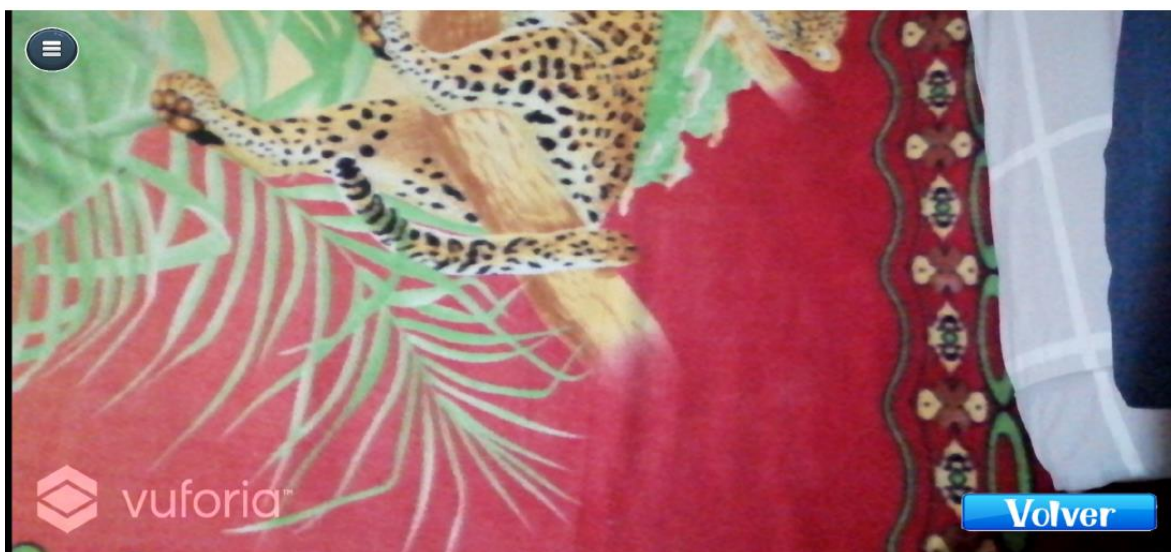


Figura 81.

Ver objetivo

En la interfaz de la figura 80, se puede leer el objetivo antes de realizar el ejercicio, pero en dado caso que necesite recordarlo en medio de la terapia podrá presionar el botón e la esquina superior izquierda marcado con rojo (Figura 82) que desplegara un cuadro con los objetos a buscar, cuantos faltan o si ya están completados para guardarla solo deberá volver a presionar el botón. Cabe aclarar que si presiona objetos que no están en esta lista contarán

como fallos que luego se verán en los reportes y si presiona una cantidad mayores de los que se piden, estos extras también contarán como fallos.

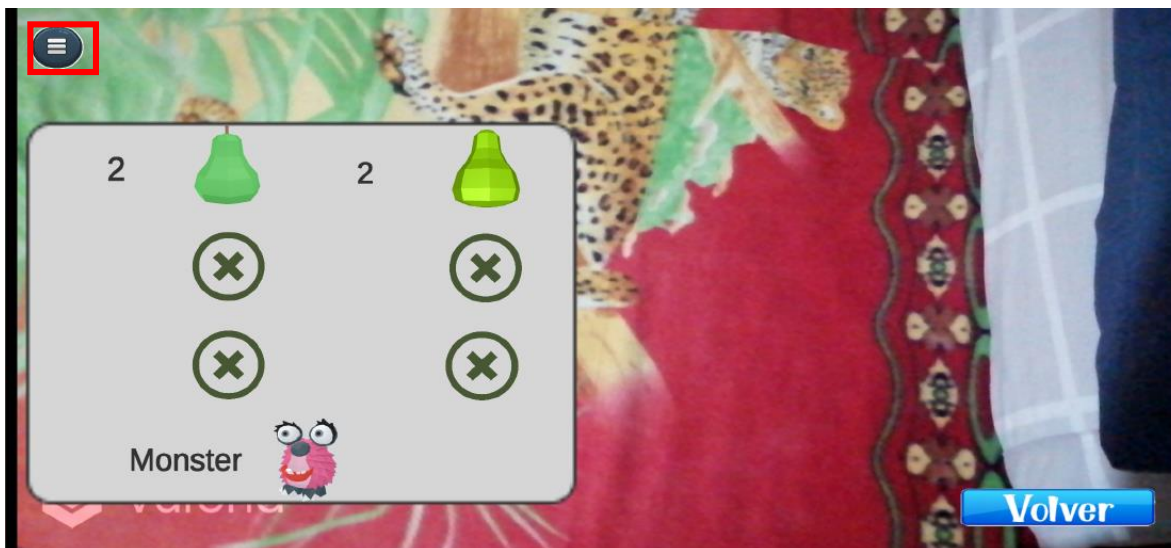


Figura 82.

Proyectar entorno AR

Para proyectar el entorno AR del estante con la configuración hecha por el terapeuta, hay que sostener el dispositivo de forma horizontal y moverlo de tal forma que se acerque y se aleje de la superficie en la que desea proyectar, dependiendo la luz y el plano esto podría demorar un poco o ser instantáneo. Como se ve en la figura 83, solo basto un pequeño movimiento de acercamiento a la superficie y salió el apuntador que también nos muestra la dirección de proyección que va a tener el objeto, la idea es tenerlo centrado para verlo centrado (Esto no afecta a la realización del ejercicio) como se aprecia en la figura 83, después solo dándole un toque a la pantalla creará el entorno en el apuntador, Figura 84. Se mantendrá en su lugar así se esté mirando con el dispositivo a otro lado, y estará ahí hasta que se complete el tiempo establecido (Perdiendo) o hasta que complete el ejercicio (Ganando).



Figura 83.



Figura 84.

Interacción con los objetos

Para realizar el ejercicio debe con un toque, presionar el objeto en la pantalla que se le está pidiendo una vez proyectado el entorno (Figura 84), para llevar el control de los objetos puede presionar el botón de la izquierda superior que desplegara los objetivos del ejercicio (Figura 82) que se está realizando y para

retirarlo volver a presionar el botón, en este caso el ejercicio de ejemplo se debe encontrar 2 peras de cada tipo que hay, siendo una más clara que la otra y el monster. Si presiona objetos que no le piden o más objetos de los pedidos, se sumaran a los fallos que se reflejaran en el reporte final y los objetos que le piden desaparecerán al toque, eso quiere decir que, si presiona 3 peras de un tipo, desaparecen las 2 primeras y la última se quedara, pero se sumara un fallo, así con cada objeto que se le pida, esto además acompañado de un sonido cuando es correcto y otro cuando no lo es. Cabe mencionar que se podrá acercar y alejar, moverse a los lados con el dispositivo apuntando al entorno para buscar los objetos como en la figura 85,86 que se acerca para presionar 2 peras y verificar el progreso en la figura 87.

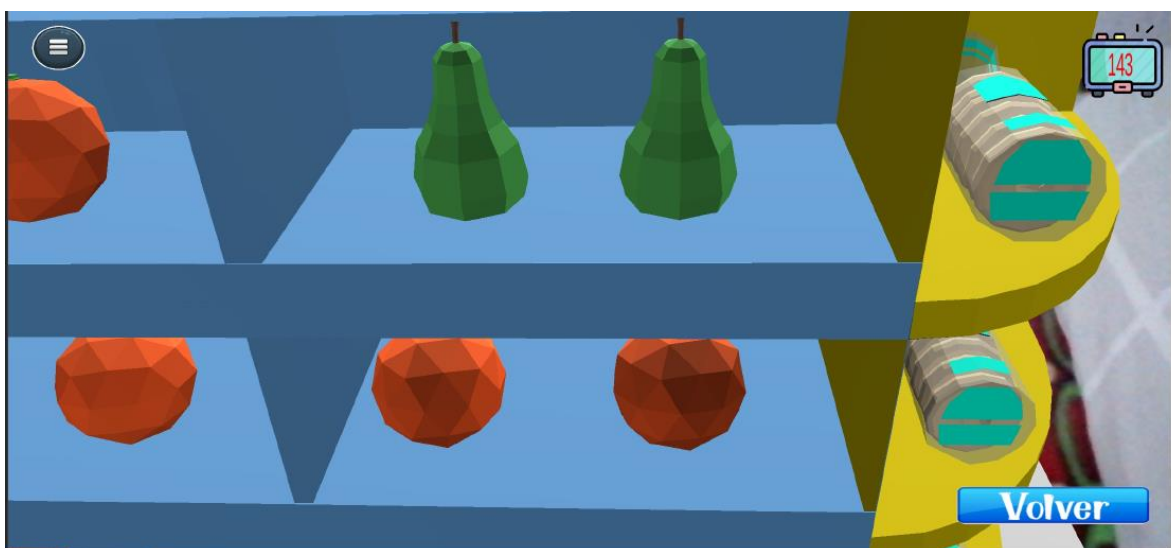


Figura 85.

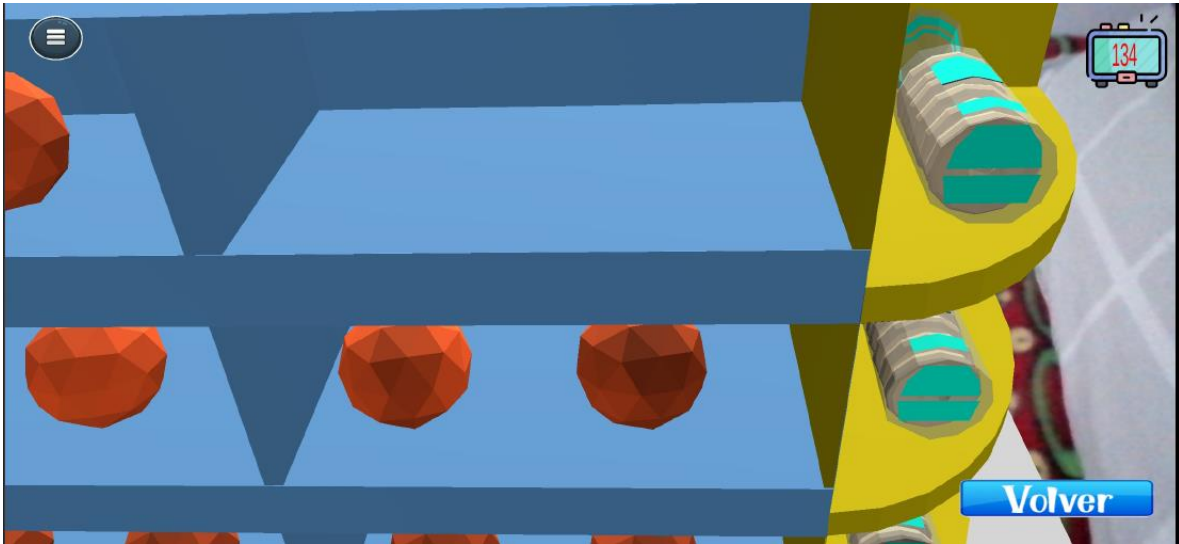


Figura 86.

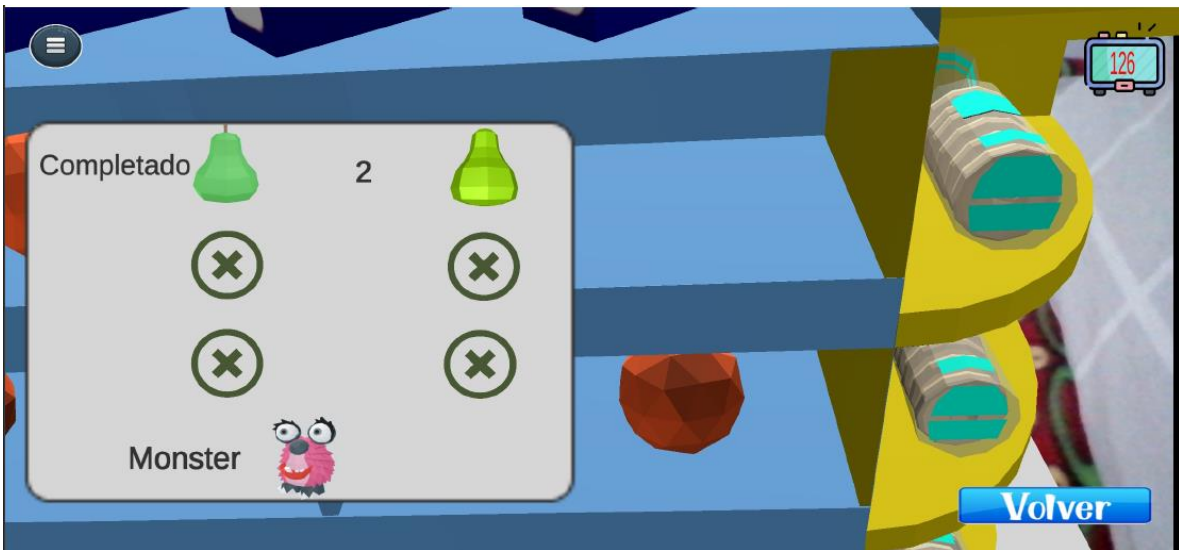


Figura 87.

Luego de hacer la búsqueda en el entorno Figura 88,89,90,91,92 completa el ejercicio antes del tiempo y sale la interfaz de victoria que lo lleva al menú principal donde podrá volverlo a hacer o si fallo, saldrá la interfaz de fallo que lo llevara igual al menú principal presionando el botón de volver, podrá realizar el ejercicio las veces que desee mientras se encuentre en las fechas establecidas. Todo esto quedara plasmado en el módulo de reportes que se puede divisar en la sección de este manual, su configuración, el número de

intentos (Veces que hizo el ejercicio), porcentaje de aciertos por intentos, fallos por intentos, tiempo utilizado por intentos. El promedio del tiempo utilizado y la varianza.

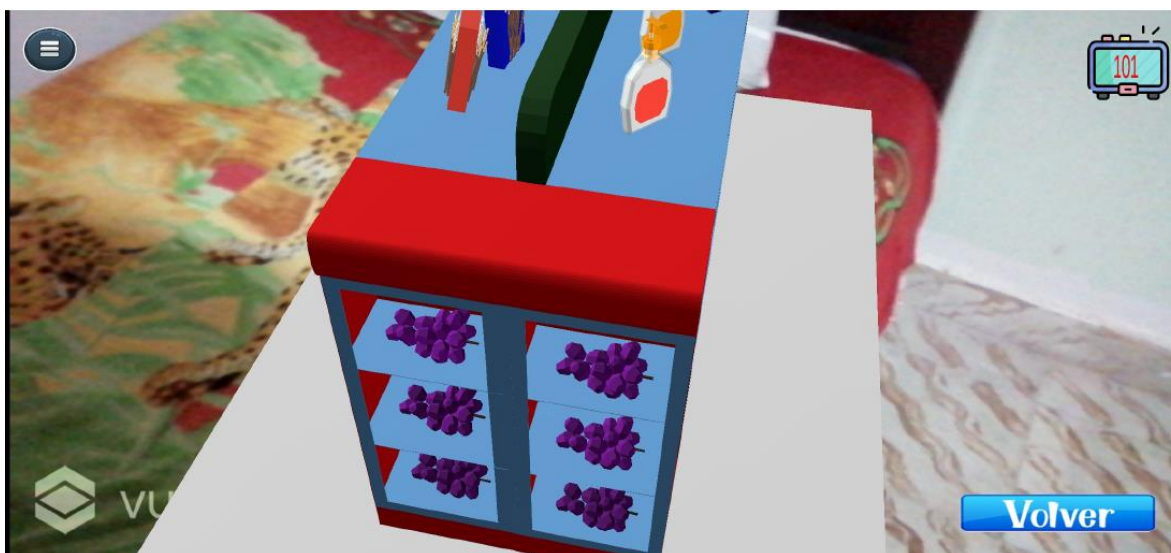


Figura 88.

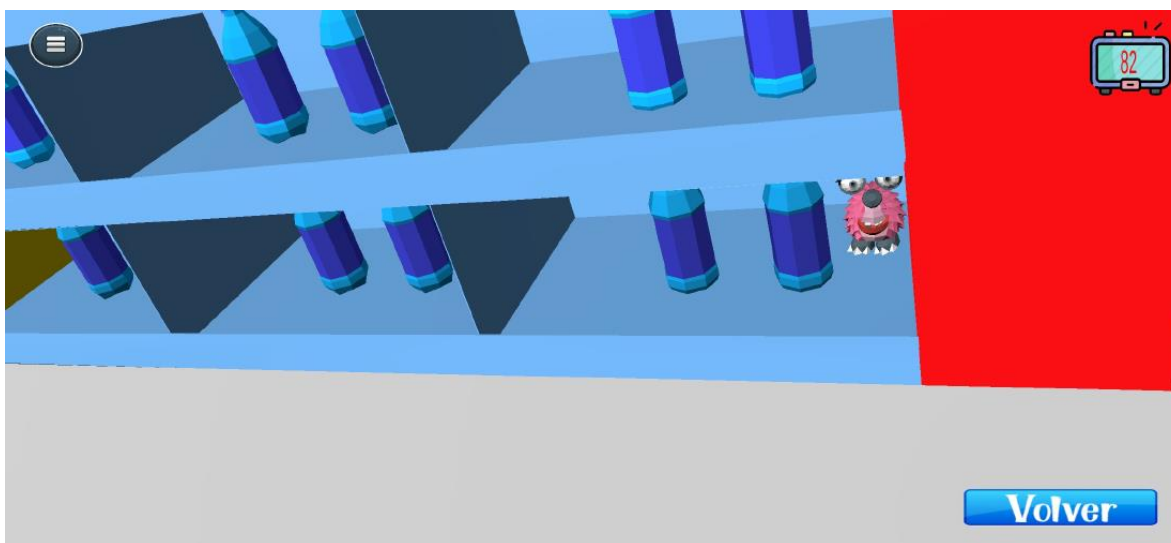


Figura 89.

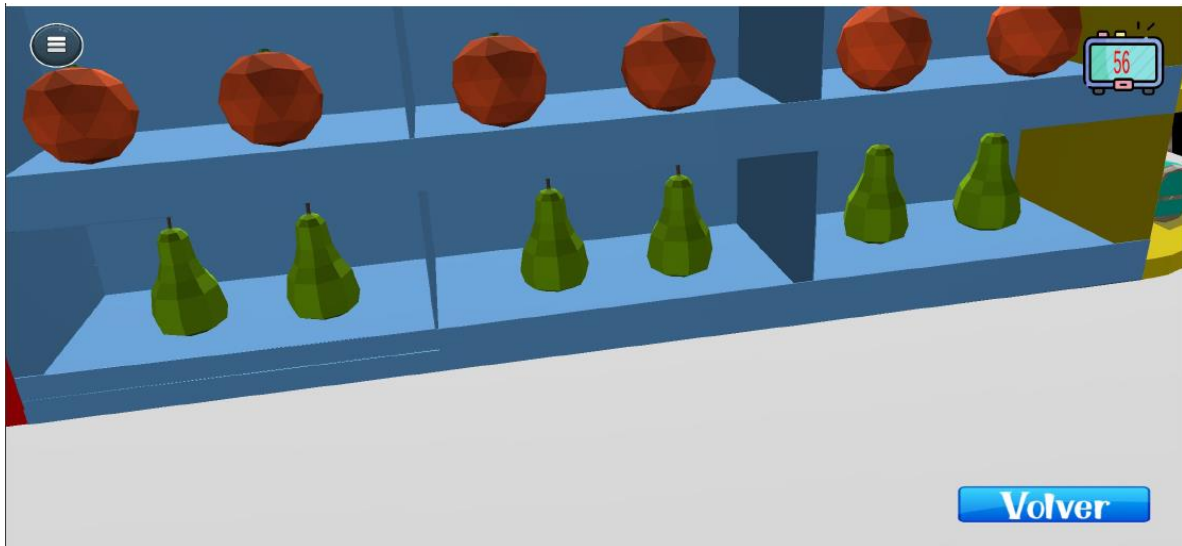


Figura 90.



Figura 91.



Figura 92.

Extra, proyección en el suelo y movimiento alrededor

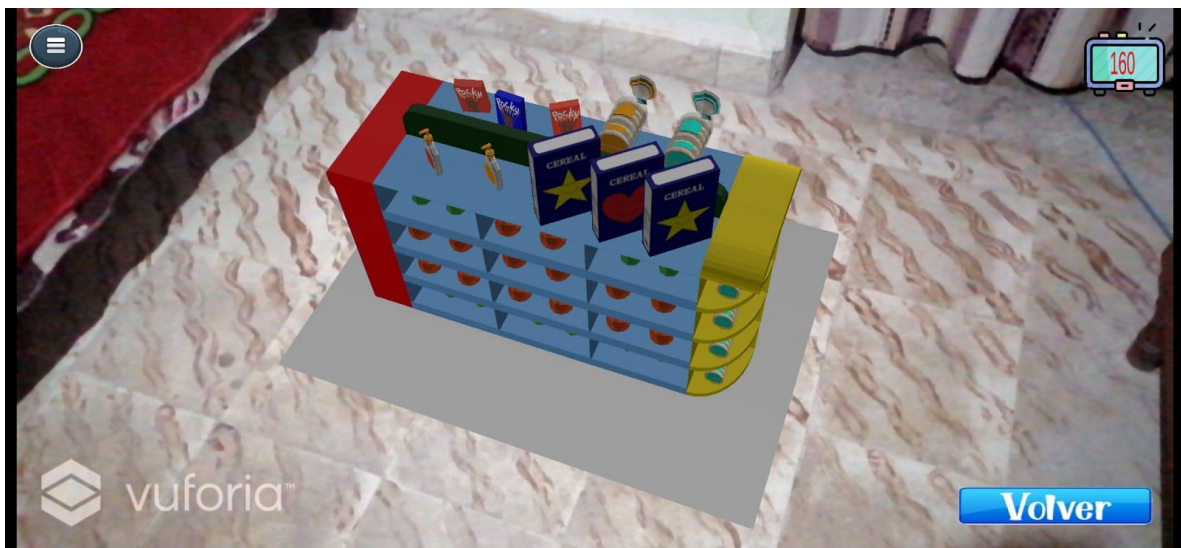


Figura 93.



Figura 94.

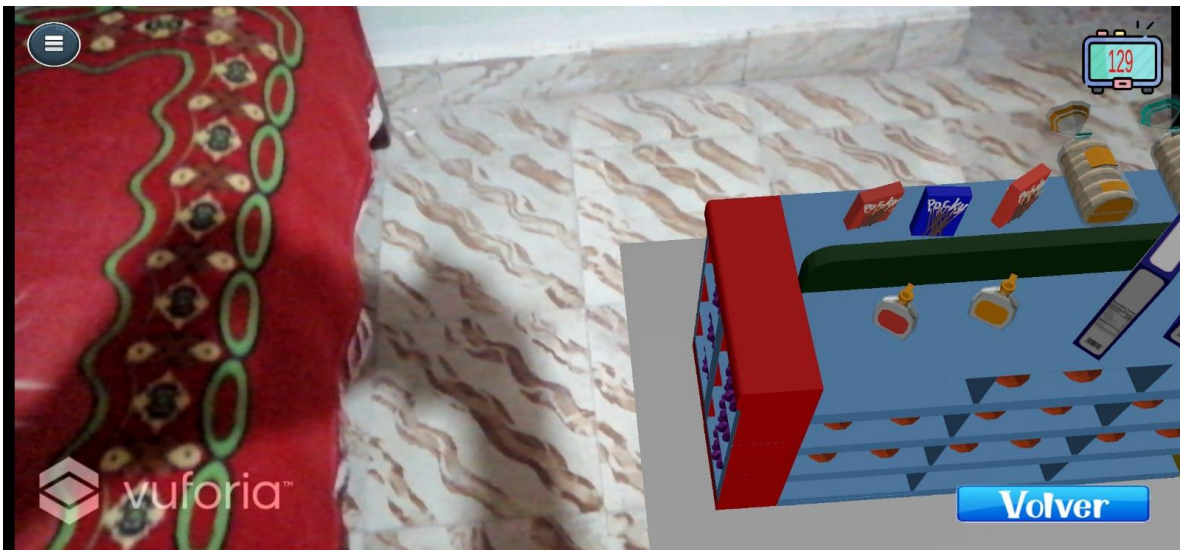


Figura 95.

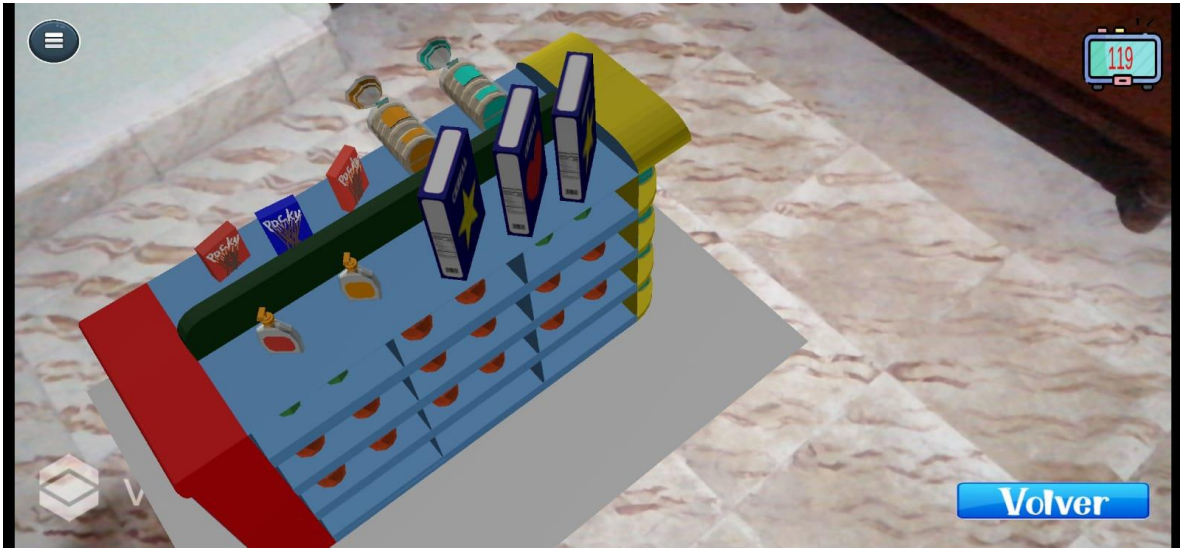


Figura 96.

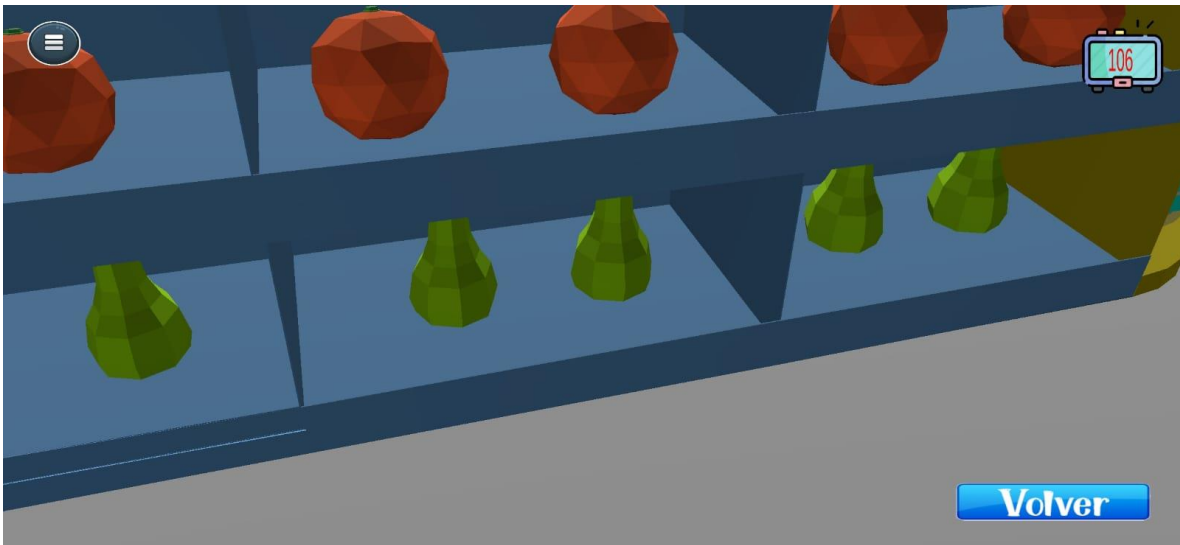


Figura 97.



Figura 98.

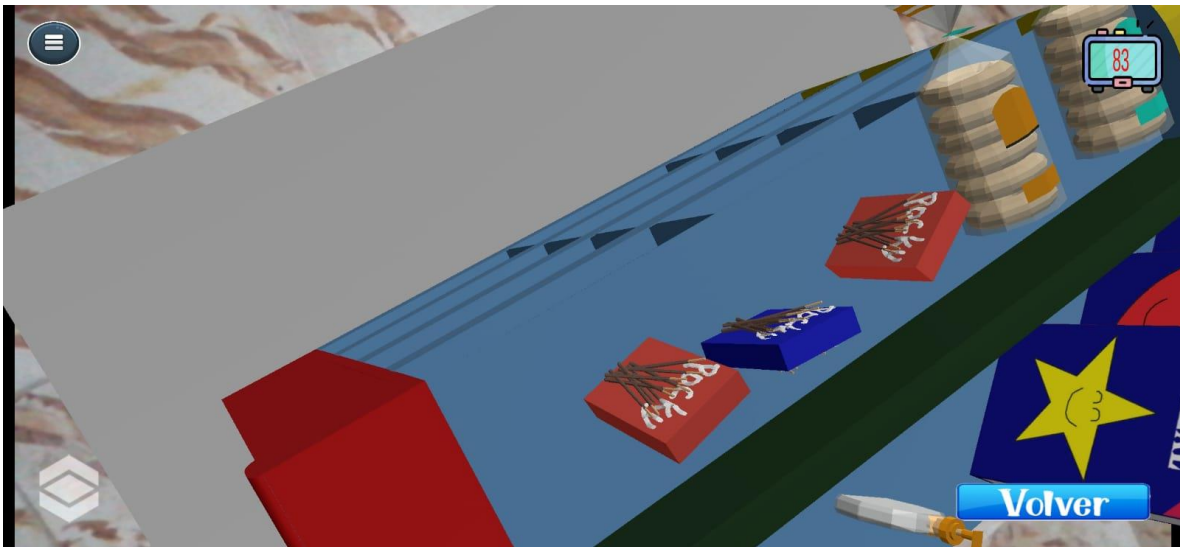


Figura 99.

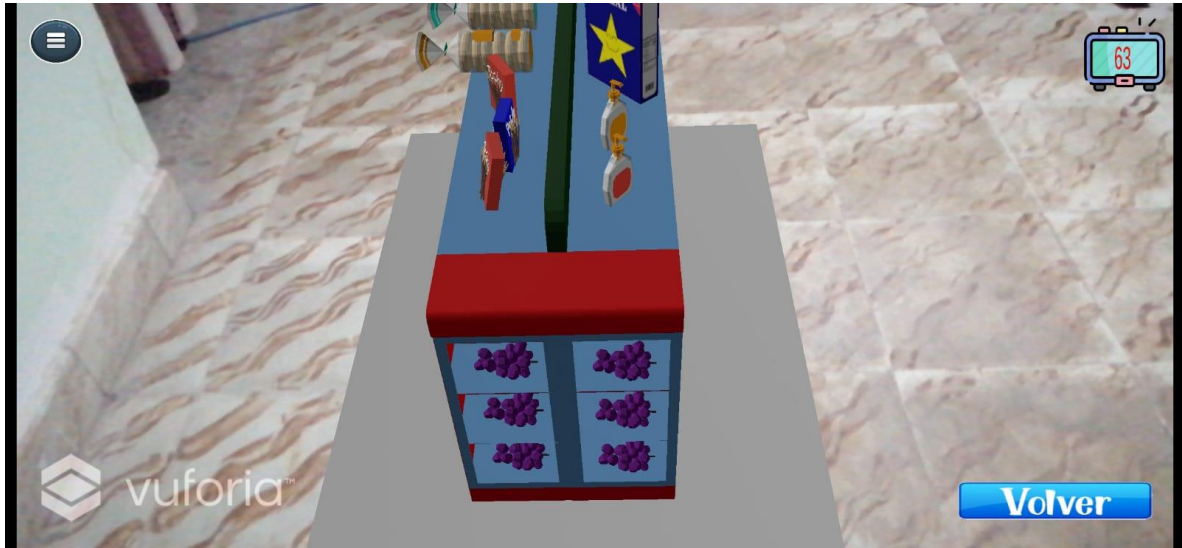


Figura 100.

MÓDULO DE REPORTES

El módulo de reportes es una herramienta extra que permite la generación de un reporte en el que se verá reflejado el desempeño del paciente en la realización de o los ejercicios, por consiguiente, si no ha realizado ningún ejercicio y se busca generar un reporte este simplemente no funcionara, al menos tuvo que realizar un intento para que pueda empezar a generar el reporte de desempeño.

Extraer módulo de reportes

Para ejecutar esta herramienta o modulo, se tienen que haber hecho las “Acciones previas” por lo que se va a poseer una carpeta llamada “3-ModuloReportes” Figura 101, que contara con un archivo .rar Figura 102, que se deberá descomprimir tal cual como se hizo en las “Acciones previas”, dando clic derecho y “Extraer aqui” y esto generara una carpeta llamada “MiniSuper ModuloReportes” como se ve en la figura 102,103

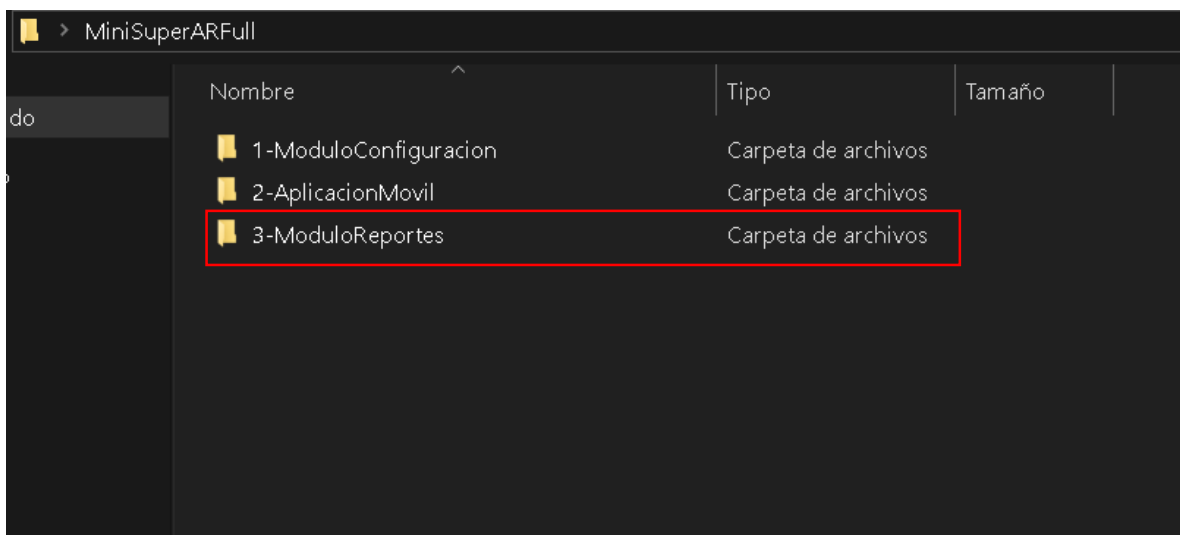


Figura 101.

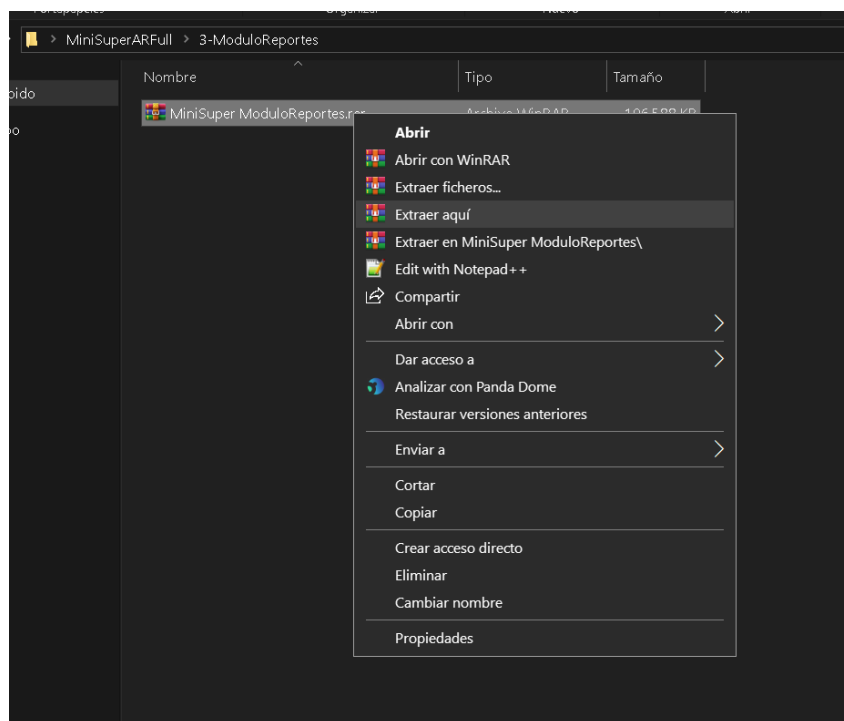


Figura 102.

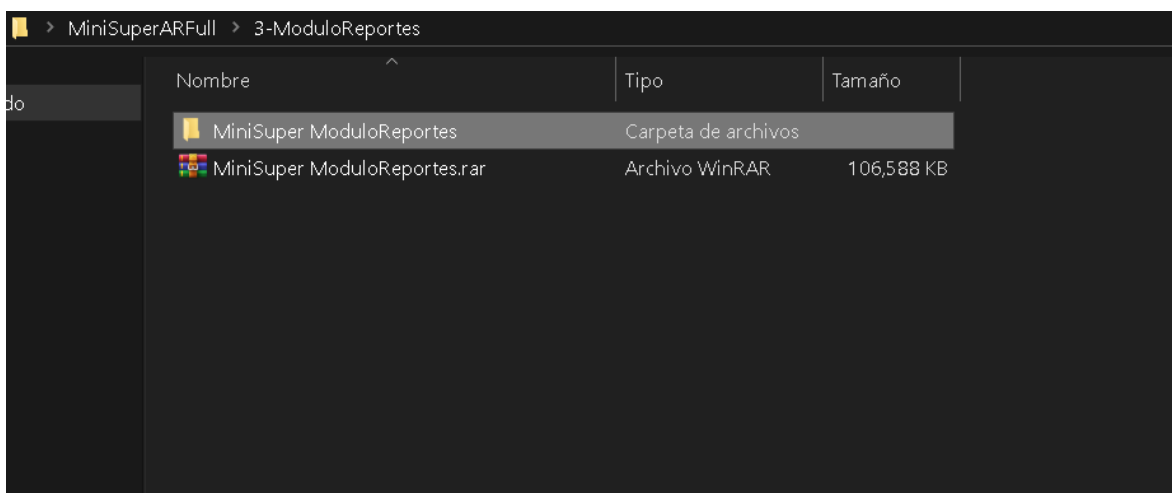


Figura 103.

Ejecutar módulo de reportes

En la carpeta “MiniSuper ModuloReportes” se encuentra un archivo llamado “ModuloReportes.exe” Figura 104, hay que darle doble clic y esperar a que se abra la interfaz inicial, una vez abierta visualizara lo que se ve en la figura 105.

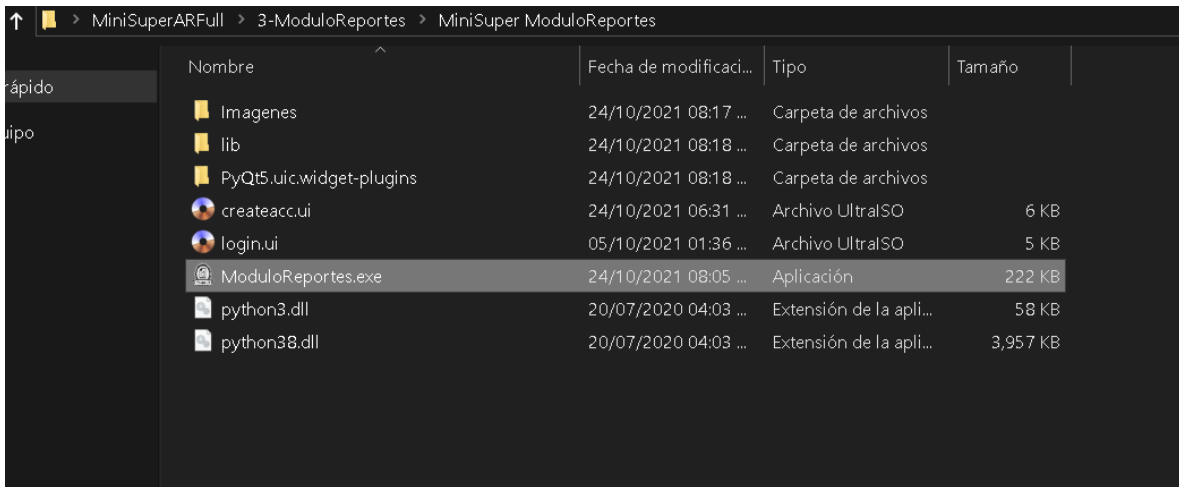


Figura 104.

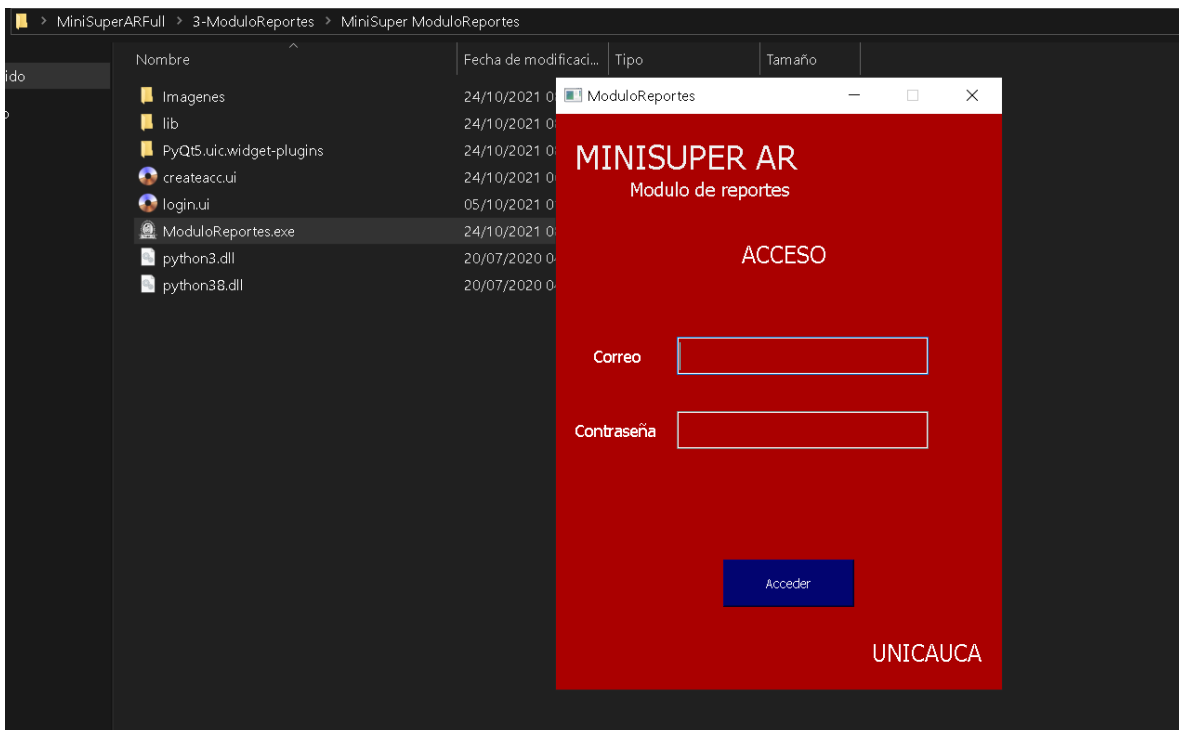


Figura 105.

Iniciar sesión

Para iniciar sesión, una vez ejecutado el modulo hay que ingresar el correo y la contraseña con el se realizó el registro en el módulo de configuración y con el que accede también a ese modulo, luego debe darle en el botón “Acceder”. Estos campos llenos se pueden ver en la figura 106.

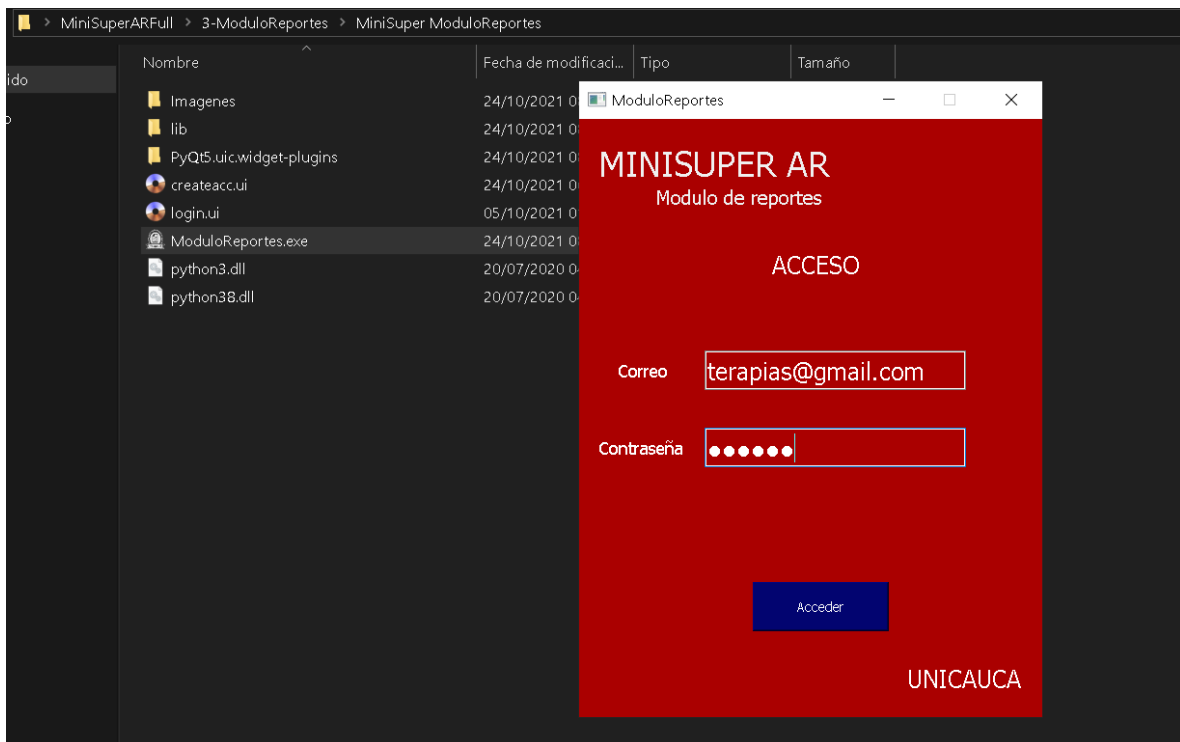


Figura 106.

Lo que lo llevara a la interfaz principal en donde generara los reportes en pdf.

Interfaz principal

La interfaz principal como se ve en la Figura 107, posee dos botones iniciales “Cargar datos” y “Limpiar”. Como dice el texto que se presenta en la parte superior se debe introducir el nombre de usuario del paciente en el campo de texto, respetando espacios, mayúsculas y minúsculas. Este nombre de usuario

es el mismo con el que ese paciente accede a la aplicación móvil, como se dijo anteriormente es el principal distintivo del paciente ya que es único, cabe aclarar que solo podrá generar el reporte de los pacientes que haya creado en el módulo de configuración, esto quiere decir que no podrá generar el reporte de un paciente de otro terapeuta.

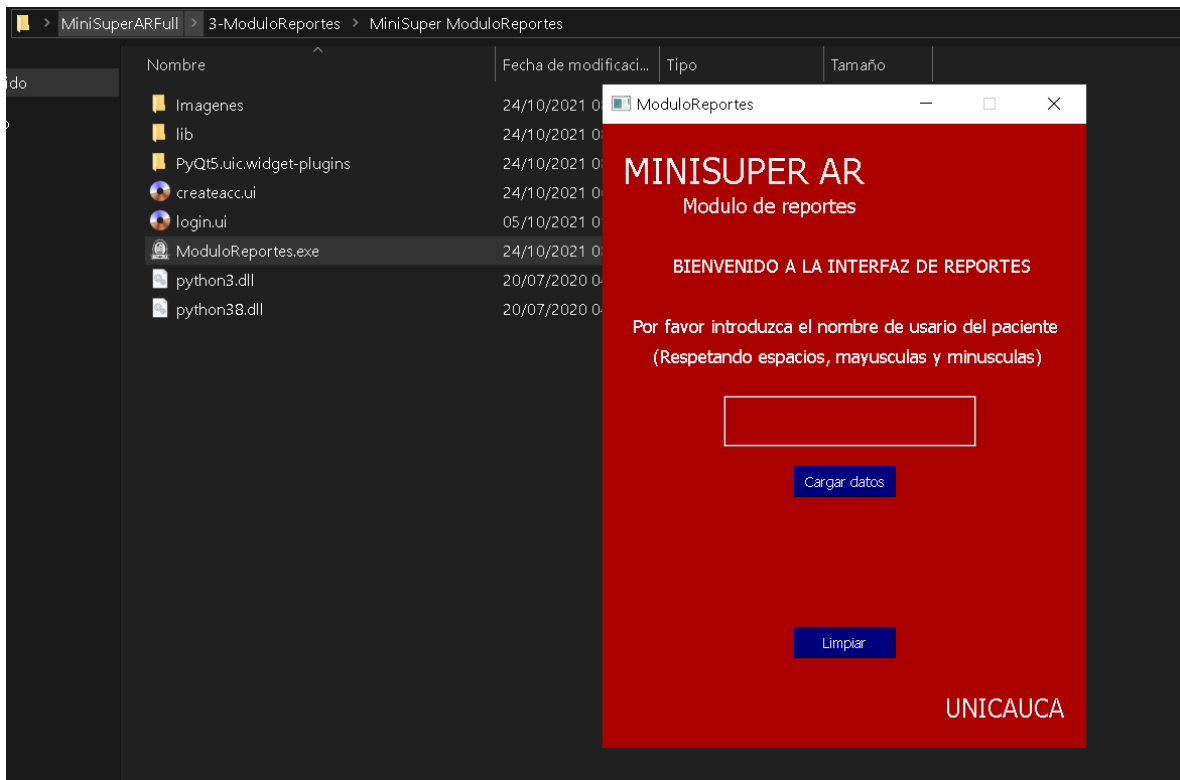


Figura 107.

Generar reporte

Para generar el reporte debe colocar en el campo de texto el nombre de usuario del paciente, respetando espacios, mayúsculas y minúsculas; para este ejemplo usaremos el nombre de usuario "tita". Una vez colocado el nombre de usuario se deberá comprobar si existen datos oprimiendo el botón "Cargar Datos", en dado caso que este no posea saldrá un aviso de error, se podrá cerrar la herramienta o darle en el botón de limpiar para colocar otro usuario de paciente, pues si no ha realizado los ejercicios no habría un reporte que

generar. Si efectivamente posee datos saldrá un aviso diciendo “Se han cargado los datos, puede generar el reporte”, desaparecerá el boton “Cargar datos” y aparecerá un botón de “Generar reporte” como se ve en la figura 108.

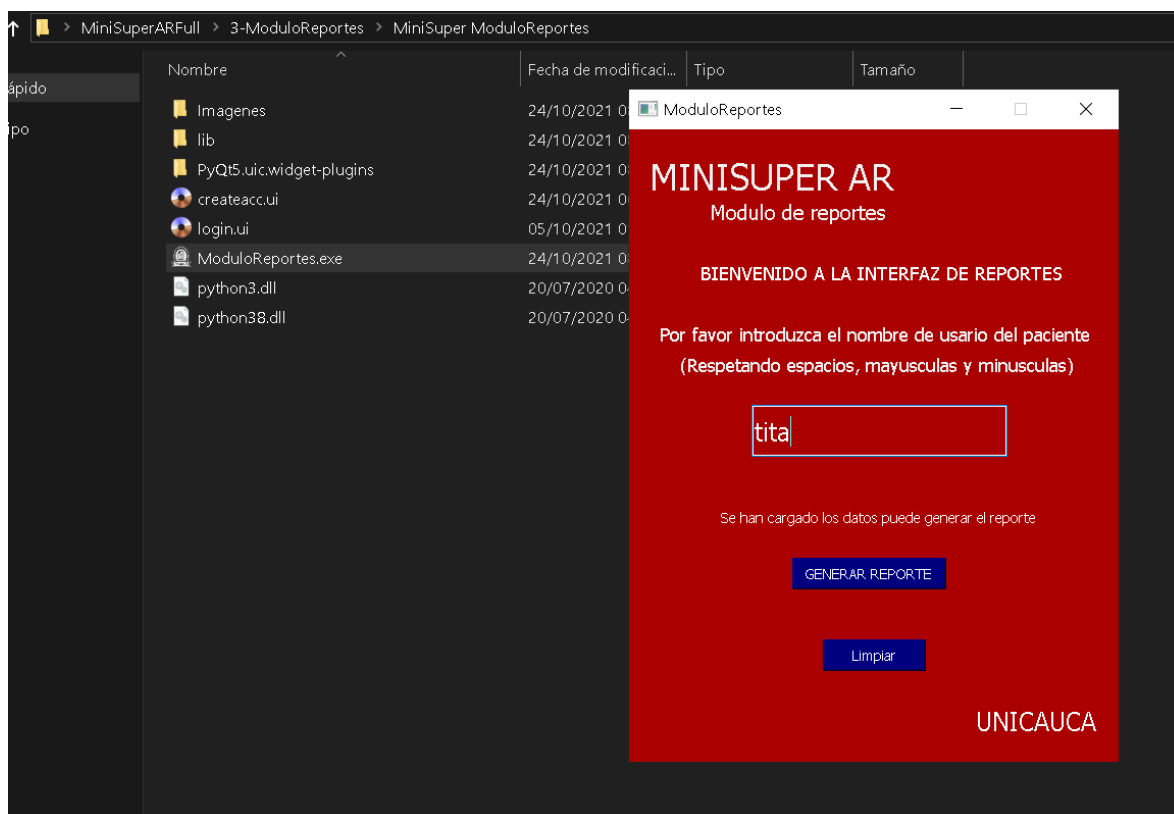


Figura 108.

Si no desea generar el reporte, puede darle en el botón limpiar y volverá al estado inicial como en la figura 107, si va a generar el reporte debe presionar el botón “Generar reporte” y en la carpeta donde se encuentra el “ModuloReportes.exe” se crearan unas imágenes que se pueden ignorar llamadas “Grafico”, sin temor a que se acumulen pues estas se sobrescriben en cada reporte que se genere y se creara el reporte del paciente en pdf, este estará nombrado como “Reporte de ____” junto a su fecha y hora de creación Figura 109, se recomienda que el reporte se traslade a otra carpeta para mantener un orden.

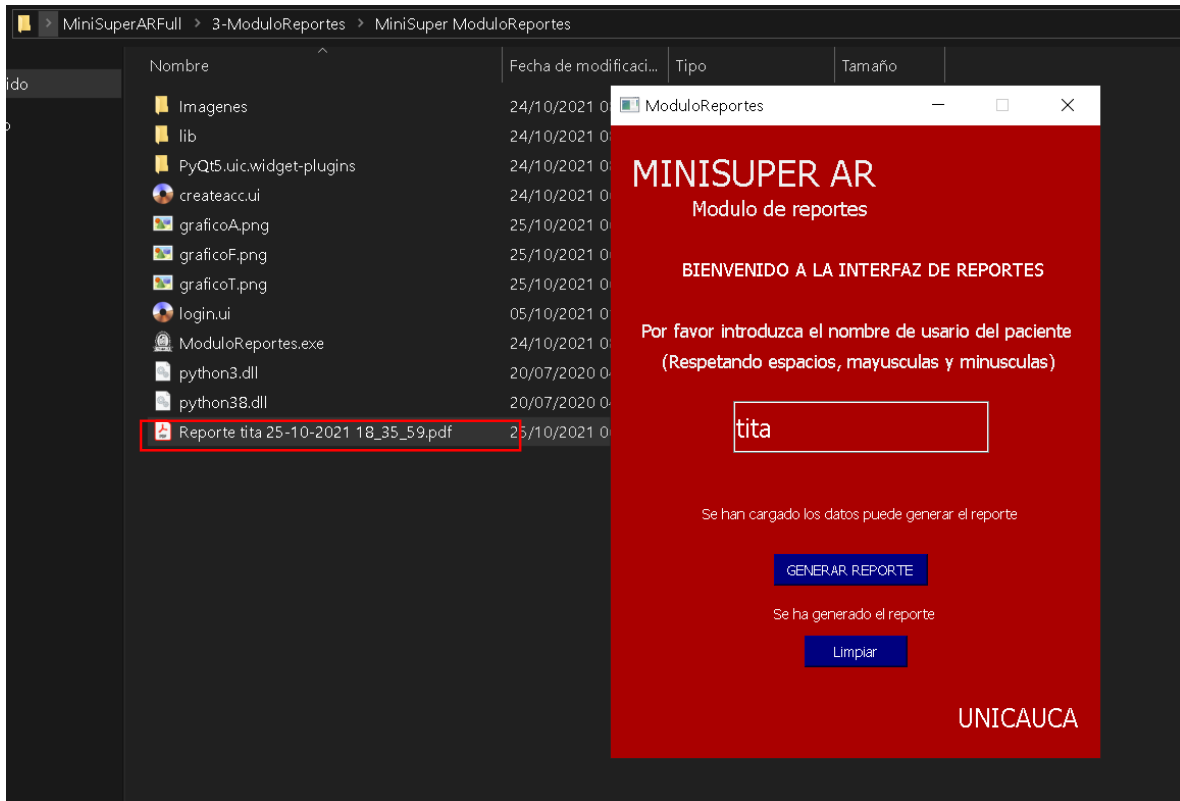


Figura 109.

Una vez generado el reporte en pdf, puede limpiar presionando el botón “Limpiar” y crear otro reporte de otro paciente o salir de la aplicación.

Contenido reporte

La primera página del reporte siempre contendrá la mismo Figura 110, el nombre completo del paciente, su documento, el usuario del terapeuta a cargo, la cantidad de ejercicios configurados y el nombre de estos ejercicios, también las caras del estante con su abreviación y los objetos con su tag o nombre, eso quiere decir que EF1 hace referencia al espacio 1 del estante frontal

Reporte de tita

Nombre completo: Laura Ortega

Documento: 32456786

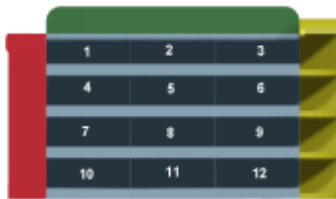
Username terapeuta asignado: tereutaUNO

La cantidad de ejercicios configurados fueron: 1

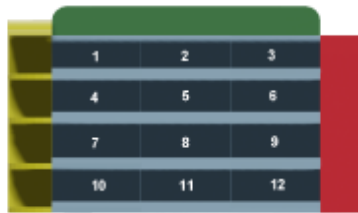
Los nombres de estos ejercicios son: Peras

Estante

Estante frontal (EF) posiciones:



Estante posterior (EP) posiciones:



Estante lateral rojo (ELR) posiciones:



Estante lateral amarillo (ELA) posiciones:



Objetos



Figura 110.

Después de la primera página empieza el reporte de los ejercicios, cada ejercicio realizado tiene dos páginas dedicadas, una página destinada a mostrar la configuración que tuvo y otra página a los resultados. En este ejemplo se cuenta solo con el ejercicio llamado “Peras” por lo que tendrá dos páginas más dando un total de 3, la página de “Peras configuración” Figura 111 siendo la 2, y la página 3 “Peras resultados” Figura 112.

Reporte de tita

Peras configuracion.

El tiempo configurado fue de: 180

La ubicacion del monster fue la numero: 2

EF1 tiene un(a): Pera	EP1 tiene un(a): AguaVari	ELA1 tiene un(a): RosquillasVari
EF2 tiene un(a): Naranja	EP2 tiene un(a): AguaVari	ELA2 tiene un(a): RosquillasVari
EF3 tiene un(a): Pera	EP3 tiene un(a): AguaVari	ELA3 tiene un(a): RosquillasVari
EF4 tiene un(a): Naranja	EP4 tiene un(a): AguaVari	ELA4 tiene un(a): RosquillasVari
EF5 tiene un(a): Naranja	EP5 tiene un(a): AguaVari	ELA5 tiene un(a): RosquillasVari
EF6 tiene un(a): Naranja	EP6 tiene un(a): AguaVari	ELA6 tiene un(a): RosquillasVari
EF7 tiene un(a): Naranja	EP7 tiene un(a): AguaVari	ELA7 tiene un(a): RosquillasVari
EF8 tiene un(a): Naranja	EP8 tiene un(a): AguaVari	ELA8 tiene un(a): RosquillasVari
EF9 tiene un(a): Naranja	EP9 tiene un(a): AguaVari	
EF10 tiene un(a): PeraVari	EP10 tiene un(a): AguaVari	ELR1 tiene un(a): RosquillasVari
EF11 tiene un(a): PeraVari	EP11 tiene un(a): AguaVari	ELR2 tiene un(a): RosquillasVari
EF12 tiene un(a): PeraVari	EP12 tiene un(a): AguaVari	ELR3 tiene un(a): RosquillasVari

El numero de retos fue de: 3 incluyendo el monster y fueron los siguientes objetos.

El objeto fue monster con una cantidad no modificable de 1

El objeto fue Pera con una cantidad de 2

El objeto fue PeraVari con una cantidad de 2

ELR4 tiene un(a): RosquillasVari

ELR5 tiene un(a): RosquillasVari

ELR6 tiene un(a): RosquillasVari

Figura 111.

En la página de configuración Figura 111, se ve el tiempo asignado, la ubicación del monster (De las 4 posibles), todas las ubicaciones de los estantes y el objeto asignado a cada ubicación y en la esquina inferior izquierda el reto o el objetivo asignado.

En la página de resultados Figura 112, se ve la fecha de inicio y de fin del ejercicio configurado, el número de intentos que hizo en ese lapso de tiempo. A su vez se ven los intentos con el número de aciertos, fallas y el tiempo utilizado. También se acompaña de 3 gráficos, el primero muestra el factor de acierto por cada intento, esto quiere decir si completo la tarea o sea si gano siendo un 100% o si perdió a cuanto porcentaje estuvo de completar la tarea. El segundo grafico muestra el número de fallos por cada intento y el tercero el tiempo utilizado en cada intento, además a la izquierda de este último grafico se ve el promedio de tiempo utilizado en la realización del ejercicio y la varianza del tiempo utilizado.

Reporte de tita

Peras resultados

Este ejercicio tuvo como fecha de inicio: 22/10/2021 y su fecha de finalizacion fue: 29/10/2021

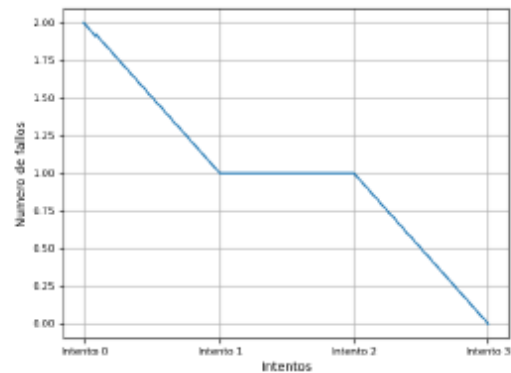
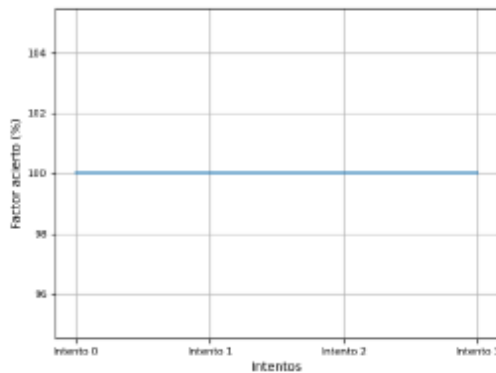
Realizo un numero de intentos de: 4

En el Intento 0: tuvo un total de 5 aciertos, 2 fallas y un tiempo utilizado de 87

En el Intento 1: tuvo un total de 5 aciertos, 1 fallas y un tiempo utilizado de 132

En el Intento 2: tuvo un total de 5 aciertos, 1 fallas y un tiempo utilizado de 84

En el Intento 3: tuvo un total de 5 aciertos, 0 fallas y un tiempo utilizado de 123



La media del tiempo utilizado es: 106.5 segundos

La varianza del tiempo utilizado es: 452.25 segundos

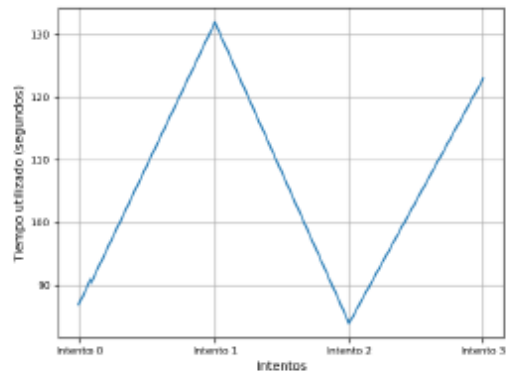


Figura 112.