

UNIVERSIDAD MIGUEL HERNÁNDEZ DE ELCHE

ESCUELA POLITÉCNICA SUPERIOR DE ELCHE

GRADO EN INGENIERÍA INFORMÁTICA EN
TECNOLOGÍAS DE LA INFORMACIÓN



Aplicación interactiva para la intervención y
monitorización psicológica de síntomas de
depresión y ansiedad en población adolescente.
TFG interdisciplinar.

TRABAJO FIN DE GRADO

Julio 2022

AUTOR: Manuel García Cotes

DIRECTOR: Manuel Quesada Martínez

RESUMEN

Se ha demostrado que los/as adolescentes experimentan una etapa llena de transformaciones en las que son más vulnerables a padecer trastornos emocionales, siendo muy comunes la depresión y la ansiedad. En respuesta a esta problemática, se ha decidido desarrollar la aplicación *Here&Now*, una aplicación web destinada a reducir síntomas de depresión y ansiedad en adolescentes. Además de intervenir en una etapa clave de la vida donde comienzan muchos problemas biopsicosociales, *Here&Now* también pretende intervenir ante síntomas para evitar que se conviertan en un futuro en trastornos emocionales más graves, a través de técnicas cognitivas-conductuales y de la técnica de Mindfulness basada en técnicas cognitivas (MBCT). Por lo tanto, en esta aplicación se presentará un espacio interactivo, guiado y acompañado por una mascota, que implementa estas dos técnicas a través de una serie de sesiones divertidas y relajantes que fomentan el autoconocimiento, autocontrol y atención plena para reducir síntomas de depresión y ansiedad y, por ende, la incidencia de adolescentes con problemas emocionales.

Se ha realizado un estudio de aplicaciones relacionadas. Este estudio nos ha dado una idea de que tipo de aplicaciones de psicología hay en el mercado y como abordar apropiadamente nuestra aplicación web. Además, la investigación del estado del arte de las tecnologías de desarrollo nos ha permitido definir una arquitectura basada en React, Three.js y Node.js usando principalmente el lenguaje de programación JavaScript, y se ha desarrollado una base de datos relacional MySQL a la que se accede a través de Node.js.

El resultado obtenido ha sido un prototipo de una aplicación web funcional que permite al usuario realizar sesiones de depresión y ansiedad previamente agendadas. Para mejorar la experiencia del usuario durante el progreso de las sesiones se ha creado un mundo 3D personalizable en el que el usuario puede sentirse más cómodo mientras realiza sus sesiones. Además, se ha proporcionado a los psicólogos una sección especial para que puedan visualizar el progreso del paciente e interactuar con él a través de mensajes si lo consideran necesario. El sistema se ha probado posteriormente en la mayoría de los navegadores web testeando que se haya cumplido todos los requisitos. Asimismo, se han identificado posibles mejoras futuras a implementar en caso de decidir comercializar la página web desarrollada.

Palabras Clave: adolescentes; depresión; ansiedad; terapia cognitivo-conductual; mindfulness; aplicación web interactiva; React; Three.js; Node.js; MySQL; javascript.

AGRADECIMIENTOS

En primer lugar, me gustaría expresar mi profundo agradecimiento a mis padres, mi pareja y toda mi familia por brindarme un apoyo incondicional a lo largo de mi trayectoria académica.

En segundo lugar, quiero extender mi gratitud a los profesores del Grado de Ingeniería Informática en Tecnologías de la Información, quienes me han proporcionado una invaluable fuente de conocimiento que he podido aplicar tanto en este proyecto como en mi desarrollo profesional. En particular, deseo destacar a mi tutor, Manuel Quesada Martínez, quien no solo ha sido un excelente docente, sino que también me ha enseñado que, aunque la programación puede parecer desafiante al principio, con dedicación y esfuerzo se puede alcanzar el nivel que he alcanzado en la actualidad.

Por último, quiero expresar mi agradecimiento a mis compañeras, Andrea y Cristina, por su constante apoyo y por brindarme claridad en mis ideas cuando más lo he necesitado. Ha sido un verdadero placer trabajar con ellas, gracias a su gran colaboración y facilidad de trabajo en equipo.



TFG INTERDISCIPLINAR

El siguiente TFG ha sido realizado en la modalidad Trabajo Fin de Grado interdisciplinar (TFGi). Este modelo de TFG ha surgido de un programa de Innovación Docente y consiste un Trabajo de Fin de Grado realizado de manera colaborativa dentro de un equipo integrado por estudiantes y tutores de diferentes titulaciones, y siendo tutorizado este trabajo por profesorado afín a la temática que se aborda.

El presente TFGi plantea el abordaje integral de un proyecto interdisciplinar, lo que implica la intervención de competencias profesionales y conocimientos transversales para su resolución, siempre sobre la base de la cooperación entre alumnado procedente de distintos grados, cuyas competencias profesionales estén vinculadas a las peculiaridades del problema a resolver y la solución a proporcionar. Por lo tanto, los contenidos de este TFG no sólo comprenden la contextualización, objetivos, metodología, resultados y conclusiones correspondientes a los contenidos y competencias propias de la titulación del estudiante que lo presenta, sino que también muestran el proceso interdisciplinar seguido por todo el equipo en el desarrollo de su proyecto, incluyendo, de forma resumida, aspectos importantes de los contenidos elaborados con el resto de titulaciones participantes que, debido a la naturaleza del trabajo, no pueden separarse del mismo para que éste mantenga su esencia.

En este TFGi han participado Andrea Montesinos Alvarado, Manuel García Cotes y Cristina Golf Toledo estudiantes de la UMH de los grados de Psicología, Ingeniería Informática y Comunicación Audiovisual, cuya temática está referida al estudio de una propuesta de intervención para reducir síntomas de depresión y ansiedad a través de la técnica de Mindfulness MBCT mediante una aplicación web interactiva donde una mascota le acompañará en las diferentes sesiones psicoterapéuticas.

Más información sobre el programa TFGi en: <http://tfgi.edu.umh.es/>

Índice de contenido

RESUMEN	2
AGRADECIMIENTOS	3
TFG INTERDISCIPLINAR	4
1.- Introducción.....	7
1.1. Definiciones: ansiedad, depresión y estrés.....	7
1.2. Aplicaciones para el tratamiento psicológico	8
1.2.1. El papel de los videojuegos y su narrativa	10
1.3. Motivación.....	11
1.4. Objetivos y trabajo interdisciplinar	12
1.4.1. Objetivos generales y específicos	12
1.4.2. Descripción breve del trabajo interdisciplinar	13
2.- Materiales y métodos	17
2.1. Estado del arte.....	17
2.1.1. Análisis de aplicaciones relacionadas	17
2.1.2. Tecnologías de desarrollo web: criterios de selección	20
2.1.3. Tecnologías para el desarrollo de aplicaciones 3D.....	21
2.1.4. Librerías utilizadas: React + ThreeJS + NodeJS.....	23
2.1.5. Elementos de React.....	26
2.1.6. Lenguajes utilizados: JavaScript	29
2.1.7. Herramientas de base de datos: MySQL.	30
2.1.8. Módulos de NodeJS para la conexión <i>frontend-backend</i>	30
2.1.9. Herramientas y modelos del mundo 3D	32
2.2. Entornos de desarrollo (IDEs) y depuración	34
2.3. Propuesta de solución	35
3.- Metodología y Resultados.....	36
3.1. Scrum como metodología ágil para la gestión del proyecto.....	36
3.2. Análisis y diseño de software.....	39
3.2.1. Requerimientos y casos de uso.	39
3.2.2. Diagrama de flujo de navegación del entorno virtual.....	40
3.2.3. Diseño de la base de datos.....	41
3.2.4. Estética y modelos de la aplicación.....	43
3.3. Otros aspectos destacables de implementación.....	45
3.3.1. React y el uso de los componentes y <i>hooks</i>	45
3.3.2. Gestión de formularios dinámicos	48

3.3.3. El uso de Three.js con React para el entorno 3D.....	50
3.3.4. Movimiento del usuario en el entorno virtual	52
3.3.5. Integración e interacción con los modelos 3D	54
3.3.6. Implementación del Backend con Node.js, Express y MySQL.....	56
3.3.7. Uso de Axios para realizar peticiones desde el <i>frontend</i>	58
3.3.8. Gestión de la sesión de usuario	59
3.4. Ejemplo de uso de la aplicación web	60
3.4.1. Login y registro de usuarios	60
3.4.2. Página principal del psicólogo.	62
3.4.3. Primer inicio de sesión del paciente.	63
3.4.4. Elementos interactivos del entorno.	64
3.4.5. Introducción a las sesiones.	66
3.4.6. Aspectos relevantes de las sesiones.	68
3.5. Vídeo con la Demo de la aplicación.....	72
3.6. Código fuente	73
4.- Conclusiones y trabajo futuro.....	74
4.1. Trabajo futuro	75
5. Bibliografía	77
ANEXO I: casos de uso desarrollados	83

1.- Introducción

La adolescencia es el período comprendido entre los 10 y los 19 años, en el que las personas perciben una serie de profundas transformaciones psicosociales, por lo que hace que sea un momento muy crítico en su vida [1].

Además de que los adolescentes experimentan cambios biopsicosociales, también es una población vulnerable para padecer problemas de salud mental, sobre todo problemas emocionales como depresión y ansiedad, según la Organización Mundial de la Salud [1].

1.1. Definiciones: ansiedad, depresión y estrés

La ansiedad se caracteriza por preocupación excesiva y difícil de controlar, lo que la lleva a ser lo suficientemente alta como para generar angustia y malestar en la vida cotidiana, siendo incluso limitante en algunas ocasiones [2].

La depresión, según la Organización Mundial de la Salud (OMS), es un trastorno común de salud mental caracterizado por la tristeza persistente, la pérdida de placer y/o interés en actividades que antes sí le agradaban, la alteración de sueño y apetito, el cansancio, la falta de concentración y la rumiación. Se calcula que, a nivel mundial, el 5% de los adultos padecen depresión y sus efectos pueden impedir que las personas actúen y vivan una vida gratificante [3].

El estrés, definido por primera vez en 1936 por Hans Selye, es considerado como un factor muy importante para la salud psicológica, fisiológica y social. Según este autor, el estrés es un conjunto coordinado de reacciones fisiológicas que surgen ante estímulos que consideramos amenazantes y fuera de nuestra capacidad adaptativa [4]. El estrés no siempre es perjudicial. Existe un tipo de estrés positivo (Eustrés) que, según Selye, las personas lo necesitan en cierto nivel para afrontar el día a día y predisponerles a la búsqueda de bienestar y realización personal. Pero, también existe el estrés negativo (Distrés), que es el más conocido y destructivo, puede generar angustia, dolor y que afecta negativamente tanto al organismo como a la salud mental [5].

Las personas más vulnerables al estrés, ansiedad y depresión, en este caso los adolescentes, disponen de esquemas cognitivos centrados en el peligro y amenaza en torno a la aceptación, competencia y la sensación de control que tienen sobre la situación estresante. También tienden a sentirse afectadas negativamente por situaciones estresantes, tendencia a experimentar emociones negativas (es decir, ansiedad, ira o afecto depresivo), timidez e impulsividad, en comparación con las personas que son más estables emocionalmente [6].

Por esta razón, se ha decidido que el presente TFGi esté destinado a población Adolescente, siguiendo la recomendación de la guía de recomendación de la Organización Mundial de la Salud [1] sobre el abordaje y prevención de casos de adolescentes con problemas de salud mental, ya que, si no se abordan, los síntomas pueden derivarse a trastornos psicológicos más graves [7]

1.2. Aplicaciones para el tratamiento psicológico

En las últimas décadas, la aplicación de las nuevas tecnologías (videojuegos, Realidad Virtual, simulaciones en ordenadores, aplicaciones para terminales móviles, *biofeedback*, etc.) se ha ido incorporando cada vez más en diversos campos de la práctica psicológica. Por ejemplo, hace 10 años que se incorporó al mercado la primera aplicación para móvil y desde entonces su uso se ha ido extendiendo cada vez más a nivel mundial, hasta suponer en 2017 más del 80% del tiempo que se dedica al uso de estos dispositivos móviles [8].

Dado que estamos en una era altamente digitalizada, es muy sencillo acceder a estos medios para recibir tratamiento. Parte del éxito de su uso recaen en los beneficios a la hora de realizar el seguimiento de un paciente. Por ejemplo, durante un tratamiento psicológico es necesario que el paciente asista regularmente a las sesiones programadas por su psicoterapeuta, además de establecer un plan de ayuda que permita al paciente trabajar diariamente en su recuperación. Es fundamental evaluar su progreso tanto durante las sesiones como en su hogar (tareas para casa) para asegurar la adhesión al tratamiento y ver los resultados obtenidos, posibilitando la modificación en caso de que no se estén obteniendo los resultados esperados. Es esencial que el paciente trabaje en ambas áreas, ya que una sesión adecuada en casa puede ser muy beneficiosa para su recuperación [9]. No obstante, en el tratamiento en el hogar surge un importante problema y es que no se puede garantizar que el paciente siga las indicaciones correctamente o tenga la motivación necesaria para llevarlas a cabo, lo que puede generar retrasos en su recuperación o problemas a largo plazo. Por lo tanto, es común recomendar un contacto más cercano con el psicoterapeuta para asegurar el progreso en el tratamiento.

Creemos que estas aplicaciones son beneficiosas y esenciales para un tratamiento. Además, los estudios apoyan nuestra postura de que el uso de aplicaciones informáticas es inevitable [10]. La adaptación de cada herramienta al paciente es un aspecto clave para maximizar sus beneficios. El uso de aplicaciones informáticas también permite que la terapia sea más accesible para personas que tienen dificultades para desplazarse a una consulta. Una de las aplicaciones que ejemplifica todo esto es Therapy Chat [11].

Therapy Chat es una de las muchas aplicaciones informáticas disponibles en la práctica de la psicología. Estas aplicaciones pueden ayudar a los pacientes a trabajar su recuperación desde casa a la vez que mantienen el contacto con su psicoterapeuta, lo que puede mejorar la adhesión al tratamiento y los resultados obtenidos. Esta aplicación brinda servicios personalizados de psicología y terapia para tratar la ansiedad, autoestima y depresión. A través de ella, se puede contactar con un psicólogo para realizar sesiones de terapia en línea de 60 minutos mediante videollamada (ver Figura 1), así como tener acceso a un chat en tiempo real para recibir ayuda en cualquier momento que lo necesites (ver Figura 2).



Figura 1: Ejemplo de la aplicación Therapy Chat [11].

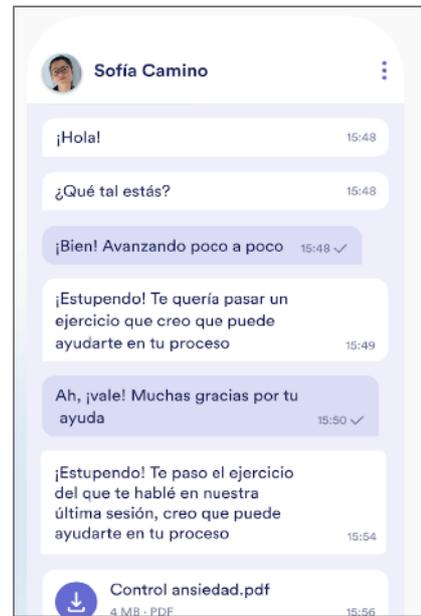


Figura 2: Ejemplo de chat con su psicólogo [11].

Otras opciones para mantener un seguimiento psicológico y emocional e incluso que te ayudan a ponerte en contacto con un psicólogo o con personas con las que hablar como Moodpath, Daylio (ver Figura 3) y Happify [12].



Figura 3: Imagen de la aplicación Daylio [12].

Además de las aplicaciones terapéuticas especializadas en el contacto con un profesional, cómo la anterior, también existen aplicaciones que ofrecen programas de tratamiento basados en técnicas específicas, como el Mindfulness, que permiten al paciente realizar tareas y hacer un seguimiento de su progreso. El Mindfulness es una de las técnicas utilizadas para tratar el estrés y la ansiedad, la cual es fácil de practicar, por lo que se han desarrollado numerosas aplicaciones informáticas para su tratamiento. Un ejemplo de este tipo de aplicaciones es Mindfulness Coach [13], cuyo objetivo es proporcionar

ejercicios guiados de meditación y atención plena, así como herramientas para establecer objetivos y hacer un seguimiento del progreso del usuario (ver Figuras 4 y 5). Además, la aplicación cuenta con una variedad de recursos y actividades que fomentan la práctica de la atención plena. En estos últimos años, ha habido un auge de aplicaciones de Mindfulness que proporcionan meditaciones guiadas que tienen una estética muy característica, como Headspace [14] o Calm [15].



Figura 4: Ejemplo de la aplicación Mindfulness Coach [13].

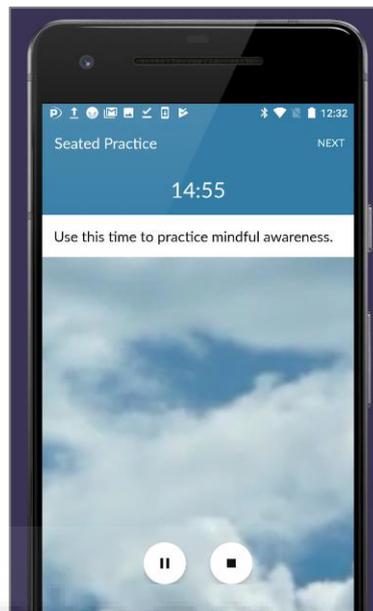


Figura 5: Imagen de Mindfulness Coach [13].

Es importante tener en cuenta que, si bien las aplicaciones informáticas pueden ser beneficiosas, no pueden reemplazar completamente a la terapia en persona. Es fundamental que el paciente reciba una evaluación y un diagnóstico precisos, y que se establezca un plan de tratamiento adecuado. Además, la terapia en persona puede ser más efectiva en algunos casos, como en situaciones de crisis o cuando se necesita un mayor nivel de apoyo emocional.

1.2.1. El papel de los videojuegos y su narrativa

El mundo de las aplicaciones y videojuegos para psicología es muy amplio. Mientras que las aplicaciones analizadas en la sección anterior suelen ser de diseño bastante básico y monótono, los videojuegos ofrecen un enfoque prometedor para el Mindfulness, ya que permiten el desarrollo de la conciencia plena y la atención en la experiencia. Además, son atractivos para los adolescentes, y hacen que se olviden del tiempo y de lo que les está pasando por su interacción absorbente [16]. Los diseñadores de los videojuegos, cada vez más, utilizan conceptos de la psicología, como la afectividad, la cognición y las emociones, aplicados tanto en los elementos referidos al diseño del propio juego, como a la narrativa de este [17].

La estética y la narrativa de la aplicación son una parte muy importante para que la experiencia del usuario sea mejor [18]. De hecho, hay estudios que

demuestran que el arte, la música y el entorno pueden tener un efecto positivo en la salud mental de las personas, provocando emociones positivas [19]. Así, el usuario no tiene que sentirse obligado a realizar las tareas que le mande la App, sino que se siente cómodo y se puede involucrar con el entorno para evitar el abandono o aburrimiento por hacer lo mismo una y otra vez [20]. Este tipo de App tendría por lo tanto un enfoque en la narrativa vista como un camino finito, con diferentes juegos interactivos con los que el usuario puede ir mejorando y avanzando tanto en el juego como en su propia historia.

Un ejemplo de un videojuego creado con propósito clínico, utilizado también como herramienta de terapia online es SPARX [21] (ver Figura 6). Este juego utiliza un enfoque de terapia cognitivo-conductual computarizada y está destinado a adolescentes entre 12 a 19 años con cuadros de ansiedad y depresión moderada [22]. También existen otros videojuegos enfocados en representar cuestiones psicológicas, como por ejemplo Celeste [23] o Gris [24]. Otros juegos pretenden resultar relajantes por sus gráficos y sus mundos como el Journey [25] (ver Figura 7) o Monument Valley [26].



Figura 6: Imagen del videojuego SPARX [21].



Figura 7: Imagen del videojuego Journey [25].

1.3. Motivación

Aunque hay ciertos estudios experimentales que han intentado crear videojuegos a través de los cuales poder dar un tratamiento a ciertos trastornos como la ansiedad, el estrés o la depresión [27], no hemos encontrado ninguna aplicación completa en los que puedas acceder a un tratamiento específico.

Nuestra motivación es contribuir a este campo desarrollando una aplicación web que permita realizar sesiones psicológicas usando técnicas de Mindfulness a través de una narrativa centrada en un entorno 3D interactivo en el que el uso del color se tendrá muy en cuenta, usando colores pastel, así como el de las formas, que serán suaves y pulidas para dotarlo de un estilo relajante y minimalista [21]. Desde el mundo, se facilitarán a los usuarios, usando tecnologías web novedosas, itinerarios interactivos que permitirán un seguimiento atractivo de las sesiones clínicas.

1.4. Objetivos y trabajo interdisciplinar

En esta sección se recogen los principales objetivos y un resumen breve del trabajo interdisciplinar realizado en este trabajo.

1.4.1. Objetivos generales y específicos

El desafío por resolver surgió durante las Jornadas de Formación organizadas como parte del programa de innovación docente de TFG interdisciplinares. Esto permitió establecer un objetivo interdisciplinar común a nivel global: *crear y planificar sesiones psicológicas guiadas por una narrativa integrada en un entorno virtual*.

En el contexto del Grado de Ingeniería Informática en Tecnologías de la Información en el que se desarrolla el trabajo descrito en este documento, se centrarán los esfuerzos en lograr los siguientes **objetivos generales**:

- Mejorar síntomas de depresión, ansiedad y estrés en adolescentes mediante el fomento del autocuidado y la técnica de Mindfulness a través de una aplicación interactiva.
- Implementar una aplicación innovadora 3D en un entorno Web que ayude a los profesionales de la salud mental a monitorizar y contactar con sus pacientes.
- Integrar estética y narrativa que generen efectos positivos en los adolescentes a nivel de salud mental.

A su vez, se definen una serie de **objetivos específicos** concretos y relacionados con la elaboración de la aplicación web, que indirectamente nos permitiría alcanzar los objetivos de más alto nivel anteriormente descritos:

1. **Dirigir la parte de IT del equipo:** liderar las reuniones con las compañeras de Psicología y Comunicación Audiovisual ayudándolas a orientar los conocimientos de sus disciplinas a un desarrollo software. Realizar un proceso de extracción de requisitos y análisis de diseño para su posterior implementación.
2. **Análisis del estado del arte y tecnologías relacionadas:** realizar un análisis exhaustivo del estado actual de la cuestión para identificar aplicaciones relacionadas, determinar la tecnología de desarrollo y los *frameworks* más adecuados para implementar la aplicación requerida en este caso específico.
3. **Implementación de la aplicación web:** implementar un prototipo funcional accesible y amigable utilizando una metodología de desarrollo de software que cumpla en la mayor medida con los requisitos y los requerimientos extraídos en los dos puntos anteriores. Este primer prototipo deberá ser extensible en un futuro, personalizable por el usuario y permitir la comunicación entre el psicólogo y su paciente.

Por último, cabe destacar que la elección de este TFG se vio motivada por los siguientes **objetivos personales**:

- **Poner en práctica conocimientos:** poner en práctica lo aprendido durante el grado sobre desarrollo web, gestión de base de datos e interfaces de usuario. Aceptar el desafío de trabajar con tecnologías novedosas de las que tengo un conocimiento limitado o nulo, con el objetivo de ampliar mi experiencia profesional en dichas áreas.
- **Formar un equipo de trabajo:** gestionar un equipo de trabajo lo más parecido al mundo laboral real, el cual requiera afrontar varios desafíos de implementación y diseño lo más parecido a un proyecto real.
- **Ventajas de un TFGi:** aprovechar la oportunidad de realizar un TFG interdisciplinar, pudiendo trabajar con personas de distintos Grados y ver múltiples perspectivas sobre este proyecto común.

1.4.2. Descripción breve del trabajo interdisciplinar

En la Figura 8 se muestra la metodología seguida por el equipo interdisciplinar (Informática, Psicología y Comunicación Audiovisual), tanto en las partes individuales como en las grupales. En las etapas iniciales del desarrollo, los tres alumnos generamos una idea inicial durante las jornadas presenciales de los TFGi. Posteriormente, esta idea se refinó a través de videollamadas, en las cuales se proponían y descartaban ideas hasta llegar a la idea final que buscábamos. Dado que a veces resultaba difícil reunirnos, también se extrajeron varios requisitos específicos mediante aplicaciones de mensajería.

Como punto de partida de este proyecto, se optó por elaborar un resumen individual que explicara cómo cada uno de nosotros concebía el desarrollo de nuestra parte individual. Luego había una puesta en común donde debatíamos qué partes podían ser interesantes y cuáles no. Esto nos permitió mantener a todo el equipo en un mismo contexto y con una idea compartida, como se refleja en la Figura 9. Se decidió mantener esta dinámica durante la fase inicial del proyecto, debido a la abundancia de ideas que surgieron. Sin embargo, muchas de ellas fueron descartadas, ya que era necesario contar con un prototipo funcional dentro del plazo establecido. Aquellas ideas que no se llevaron a cabo se tendrán en cuenta en el Capítulo 4 de Conclusiones y Trabajo futuro.

En las Figuras 9, 10 y 11 se muestran algunos ejemplos representativos de nuestras reuniones de trabajo. Durante estas sesiones, utilizamos documentos colaborativos en la nube para compartir ideas. Estas capturas son ejemplos representativos que muestran la evolución de los requisitos. Sirven como referencia para visualizar cómo el equipo fue progresando paso a paso y el impacto que cada miembro tuvo en la parte descrita.

Mapa mental Here&Now

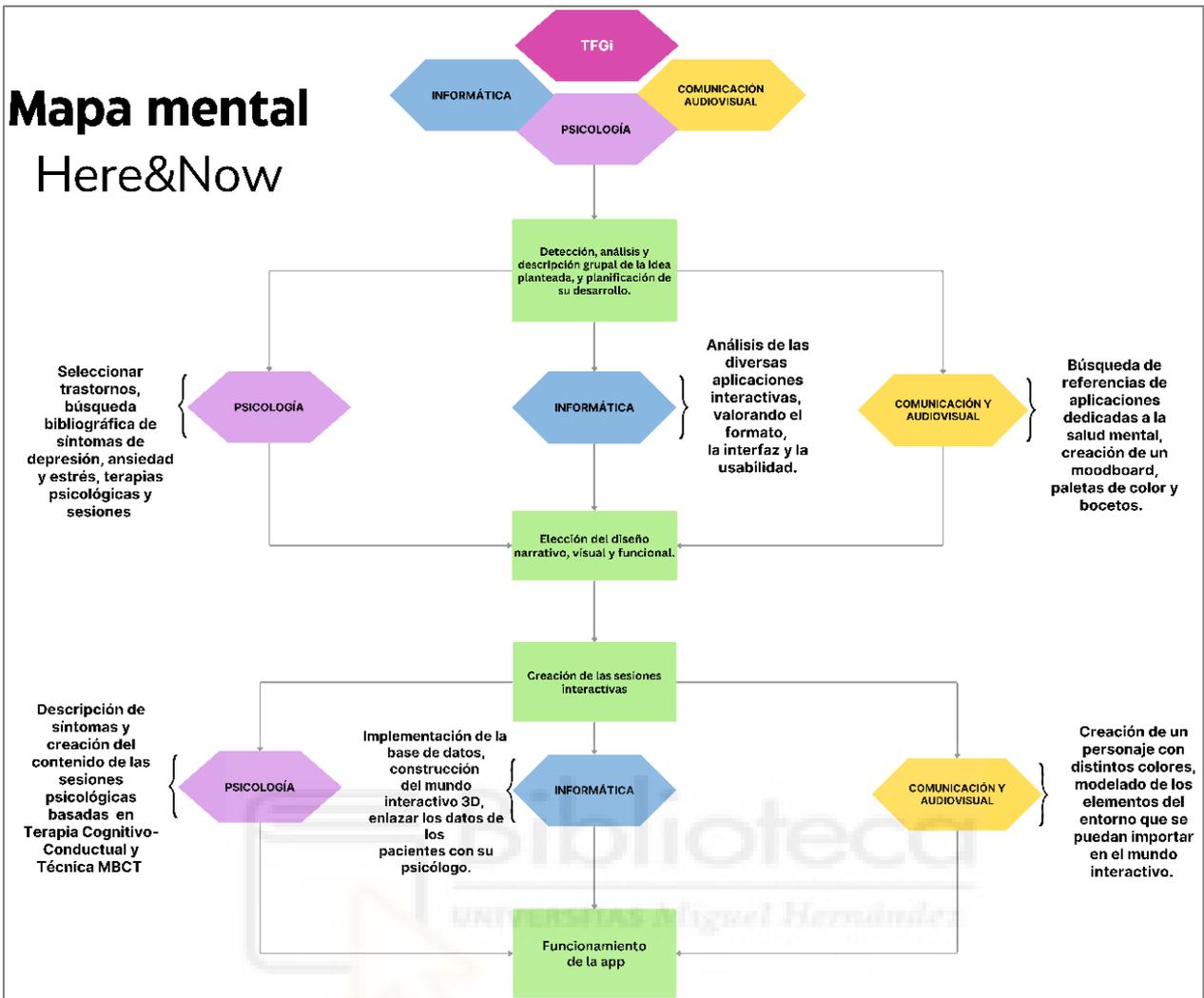


Figura 8: Mapa mental de la interacción entre las diversas titulaciones.

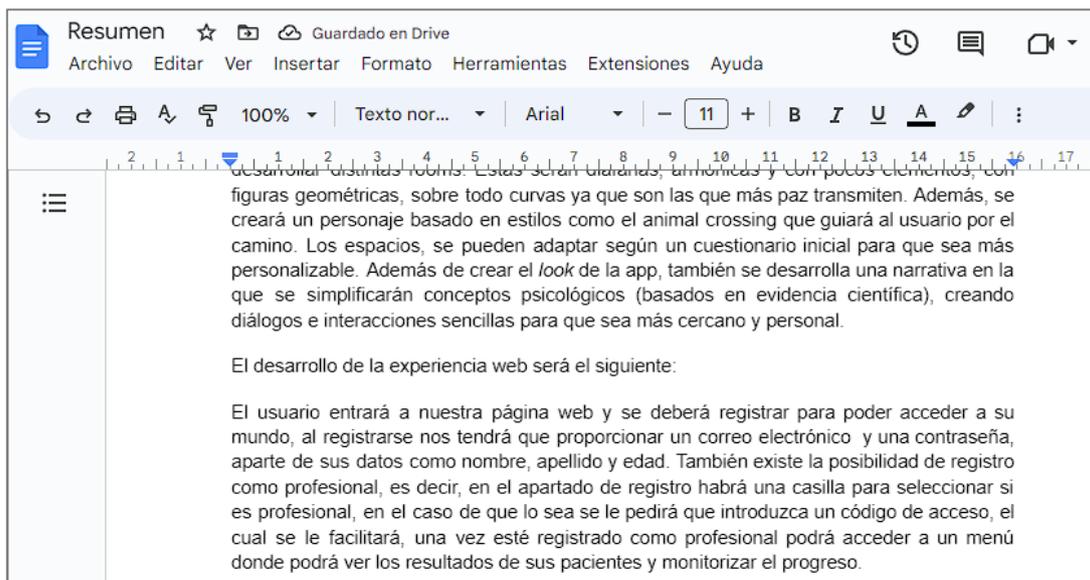


Figura 9: Ejemplo de la puesta en común del resumen individual.

En la Figura 10 se presenta un ejemplo ilustrativo de cómo abordamos individualmente los puntos críticos de la aplicación. Durante estas discusiones, compartíamos nuestras perspectivas sobre qué aspectos debíamos trabajar y una vez definidos, se creaban los recursos necesarios para aplicarlos al proyecto web. Por ejemplo, en la Figura 10, nuestra compañera de psicología determinó los usuarios de la página web, mientras que la compañera de diseño se encargó de definir las paletas de color y la tipografía que acompañarían la página.

The image shows a screenshot of a collaborative document editor. The document content includes the following text and table:

diferente manera.

1 EJEMPLO DE SESIÓN de DEPRESIÓN

I. REGISTRO EN LA APLICACIÓN

Psicología	Comunicación	Informática
- Diseña los diferentes tipos de usuarios: pacientes y psicólogos.	- Paletas de color según elección del usuario (4). - Elección de la tipografía.	- Implementación de la web con los formularios de registro.

Ejemplo de recurso generado:

UNIVERSITAS Miguel Hernández

¿Qué te gustaría trabajar hoy?

DEPRESIÓN ANSIEDAD

Figura 10: Ejemplo de análisis de los puntos críticos del desarrollo de la página web.

Una vez definida la lógica del inicio de sesión del usuario y el cuestionario de personalización de la página web, fue necesario abordar la definición de las sesiones que realizaría el usuario en nuestra aplicación. En la Figura 11, se puede observar que seguimos la misma metodología, pero en este caso, nuestra compañera de psicología hizo hincapié en la importancia de determinar con precisión qué se debía decir durante cada sesión para que fuera lo más realista posible. Este proceso se llevó a cabo para cada una de las 11 sesiones relacionadas con la depresión.

Documento colaborativo ☆ 📁 🌐

Archivo Editar Ver Insertar Formato Herramientas Extensiones Ayuda

100% | Texto nor... | Times ... | 12 | B I U A | ↺ | 📄 | 🗨️ | ⋮ | Compartir

2 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18

☰

2.2.1 SESIONES DE PSICOEDUCACIÓN

- Depresión:
 - SESIÓN 1: en primer lugar se realizará una breve sesión de psicoeducación sobre la depresión:

Bien, has elegido trabajar el área de depresión. Antes de nada, me gustaría hablarte un poco qué es la depresión:

"La depresión es una enfermedad mental que puede afectar el estado de ánimo, los pensamientos, la conducta y el cuerpo. Puedes sentirte triste, sin energía, sin esperanza y sin interés en actividades que antes disfrutabas. Es importante que comprendas que la depresión no es tu culpa, y que no es una debilidad personal o un defecto de carácter.

Es importante que sepas que la depresión no es una condición permanente, y que la mayoría de las personas que reciben tratamiento pueden recuperarse y volver a disfrutar de una vida plena y satisfactoria. Sin embargo, es fundamental que sigas las recomendaciones del tratamiento, asistas a las sesiones de terapia y tomes participación activa en tu recuperación, intentando hacer cambios positivos en tu estilo de vida, como hacer ejercicio, dormir lo suficiente y seguir una dieta saludable."

Psicología	Comunicación	Informática
- Redacción de la información que se le muestra al usuario.	- Diseño de GIF de la mascota. - Diseño de los cuadros de texto en CSS.	- Implementación de la vista donde la mascota se comunica con el usuario.

Andrea Montesin...
20:12 16 may

Lo que dice la mascota tal cual

Andrea Montesin...
21:54 28 abr

Esto es lo que dirá la mascota cuando explique la depresión (copiar y pegar tal cual en la App)

Figura 11: Ejemplo del trabajo colaborativo de una sesión de depresión.

En los siguientes apartados de la memoria enfocaremos nuestro esfuerzo en describir los aspectos específicos del análisis, diseño e implementación de la aplicación web, que es la parte más relacionada con el Grado en Ingeniería Informática en Tecnologías de la Información en el que se defiende esta parte específica del trabajo.

2.- Materiales y métodos

En este capítulo continuaremos el análisis iniciado en la introducción dirigiendo nuestro enfoque a profundizar en el análisis sobre las aplicaciones y tecnologías relacionadas.

2.1. Estado del arte

En este apartado analizaremos el estado del arte identificando desde, aspectos comunes en aplicaciones relacionadas que sean interesantes de incluir en nuestra solución, hasta las tecnologías necesarias para desarrollar nuestra aplicación web interactiva 3D.

2.1.1. Análisis de aplicaciones relacionadas

Parte del análisis realizado y explicado en la sección 1.2, consistió en que cada miembro del equipo llevó a cabo una búsqueda exhaustiva de aplicaciones móviles y web relacionadas con el tratamiento de la depresión y la ansiedad que incorporaran mundos virtuales en 3D.

Como ya hemos comentado, encontramos diversas aplicaciones que permitan a los usuarios contactar de forma virtual con psicólogos y realizar sesiones en línea y también descubrimos varios juegos móviles que mostraban una fuerte influencia del Mindfulness. Sin embargo, tras analizar todas las aplicaciones, no encontramos ninguna que lograra combinar ambas ideas de manera satisfactoria.

Se llevó a cabo una nueva búsqueda exhaustiva de aplicaciones destinadas a abordar la ansiedad y la depresión, tanto en su versión web como en aplicaciones móviles. Es importante destacar que la mayoría de las aplicaciones encontradas estaban diseñadas específicamente para dispositivos móviles.

En la búsqueda realizada en el contexto web los resultados que obtuvimos fueron poco concluyentes, ya que los resultados reflejaban que no había ninguna aplicación web que se acercara a la idea que planteábamos e incluso de menor interés que las aplicaciones móviles. La Figura 12 muestra la aplicación web que más se ajusta a las especificaciones de nuestro proyecto, ya que trata los temas de ansiedad y depresión. Sin embargo, esta es la única similitud que se puede encontrar, ya que nuestro proyecto se enfoca en un entorno 3D que permita al usuario interactuar con el ambiente mientras recibe ayuda.

Aunque hay muchas páginas web similares a esta, se decidió utilizar el ejemplo anterior para resaltar que nuestra idea de proyecto es diferente a lo que se ofrece actualmente en el Internet.



Figura 12: Ejemplo de página web para tratar la ansiedad y depresión [70].

También nos han llamado la atención dos aplicaciones móviles altamente relevantes para nuestro proyecto. La primera, llamada Yana [28], es una aplicación de autoayuda que cuenta con una función única: un chatbot que está disponible para hablar contigo siempre que lo necesites. Esto proporciona al usuario un apoyo emocional constante mientras utiliza la aplicación, como se puede observar en la Figura 13. Además, Yana también tiene un apartado de lugar seguro donde el usuario puede indicar su registro emocional o su espacio de relajación, tal como se puede ver en la Figura 14. Por último, otra característica a destacar es una sección donde el usuario puede expresar cómo se siente ese día y buscar terapia en caso de necesitarlo, como se muestra en la Figura 15.

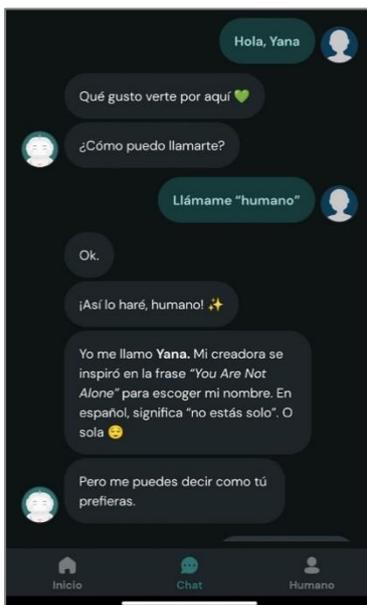


Figura 13: Chatbot de Yana [28].



Figura 14: Espacio del usuario [28].



Figura 15: Diario del usuario y opción de buscar una terapeuta [28].

La otra aplicación que nos ha llamado la atención es My Oasis [29]. En este caso, se acerca un poco más a la aplicación interactiva 3D que buscamos en el proyecto. En la Figura 16, podemos ver cómo el usuario ha ido construyendo poco a poco un oasis interactuando con la aplicación. La idea principal de la aplicación es proporcionar al usuario un espacio relajante y satisfactorio. Aunque esta aplicación no cuenta con la ayuda psicológica para la depresión y la ansiedad, la parte del mundo interactivo es un excelente ejemplo de lo que buscamos.

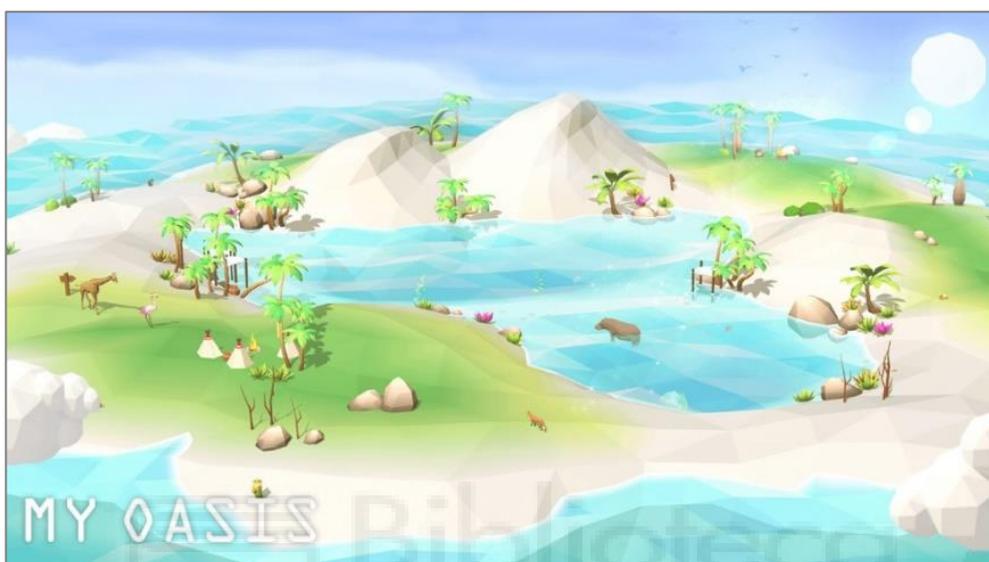


Figura 16: Imagen de la aplicación de My Oasis [29].

Dado que el objetivo de este proyecto es combinar las características de las distintas aplicaciones analizadas para crear un mundo interactivo 3D que sea relajante y satisfactorio para el usuario, y a su vez ofrecerle la disponibilidad de tener lugar seguro, un espacio de relajación y la posibilidad de contactar con profesionales en caso de ser necesario. Este enfoque permitirá al usuario disfrutar del mundo 3D mientras se enfrenta a la ansiedad y la depresión de una manera más efectiva. En resumen, se busca una aplicación que ofrezca una experiencia agradable y positiva al mismo tiempo que proporciona herramientas útiles para abordar los problemas emocionales.

En la Tabla 1 se recoge un resumen de las aplicaciones y aspectos analizados y mencionados en los párrafos anteriores.

Aplicaciones	Web o móvil	Chat con psicólogo	Cuestionarios psicológicos	Mundo 3D
Web de autoayuda	Web	No	Si	No
Yana	Móvil	Si	Si	No
My oasis	Móvil	No	No	Si
Monument Valley	Móvil	No	No	Si

Tabla 1: Tabla comparativa de las aplicaciones mencionadas.

A modo de resumen, el análisis de las páginas web relacionadas reveló que la mayoría de ellas ofrecían opciones para contactar con un psicólogo o para realizar una prueba que permitiera determinar el estado emocional del usuario. No obstante, ninguna de las alternativas evaluadas contaba con una aplicación interactiva 3D que satisficiera las necesidades específicas de nuestro proyecto. Algunas páginas ofrecían recomendaciones sobre ciertos juegos de consola, que supuestamente podrían contribuir a reducir los niveles de ansiedad y estrés, aunque estas sugerencias no contaban con validación psicológica. En relación con las aplicaciones móviles, se logró identificar varios ejemplos que se aproximaban significativamente a los requisitos establecidos. Se identificaron aplicaciones que abarcaban desde la terapia y el control emocional, hasta un juego interactivo, que se considera un excelente ejemplo para ilustrar lo que se persigue en el marco de nuestro proyecto.

Además de lo anteriores, durante el proceso de análisis, se han considerado varios factores fundamentales, entre los cuales destacan la usabilidad y la interactividad de la aplicación. Esto se debe a que nuestro objetivo principal es proporcionar al usuario una experiencia satisfactoria, especialmente considerando que nuestros usuarios potenciales padecen problemas como la ansiedad, depresión y estrés. En consecuencia, se persigue crear un ambiente lo más pacífico y placentero posible para ellos.

- **Usabilidad:** la disciplina de usabilidad en el desarrollo web se enfoca en crear sitios web que permitan a los usuarios interactuar de manera fácil, cómoda e intuitiva. El diseño centrado en el usuario es la mejor manera de lograr esto, en lugar de enfocarse en la tecnología, creatividad u originalidad. La usabilidad es esencial en la creación de sitios web y sistemas computacionales, ya que ayuda a alcanzar altos niveles de eficiencia, eficacia y satisfacción del producto para satisfacer los objetivos del usuario [30].
- **Interactividad:** la interactividad se refiere a la relación comunicativa que se establece entre un usuario/actor y un sistema, como un ordenador o un dispositivo de video. El nivel de interactividad que presenta un producto está determinado por la existencia de recursos que permiten al usuario interactuar de manera participativa y comunicativa con los materiales presentados [31].

La interactividad y la usabilidad son sumamente importantes para el desarrollo de una aplicación web centrada en el usuario, por ello, para más información al respecto se pueden leer los artículos de las referencias [30] y [31].

2.1.2. Tecnologías de desarrollo web: criterios de selección

Después de identificar los requisitos clave de la aplicación, se dedicaron los esfuerzos a mejorar los conocimientos en desarrollo web que habían sido adquiridos en la asignatura de Tecnologías Web y Desarrollo de Aplicaciones en Internet. Esta etapa es crucial en cualquier proyecto informático, ya que elegir

las herramientas adecuadas desde el principio evita tener que validar nuevamente las tecnologías seleccionadas, lo que implicaría invertir más tiempo y esfuerzo en rehacer todo el proyecto. Dado que el tiempo es limitado, es necesario invertir el tiempo necesario en elegir las herramientas correctas.

Con el fin de escoger las tecnologías adecuadas, se realizó un análisis de los siguientes puntos clave en el desarrollo de nuestro proyecto:

1. Tipo de lenguaje.
2. SDK (Kit de entorno de desarrollo).
3. *Frameworks*.
4. Librerías y *plugins*.
5. Bases de datos.

Durante este análisis también se tuvieron en cuenta los siguientes **criterios de inclusión y exclusión**:

- Criterios de inclusión:
 - Tecnología actual.
 - Amplia comunidad de soporte.
 - Flexibilidad para integrar *plugins*.
 - Buena interoperabilidad con otras herramientas.
 - Experiencia previa con el lenguaje de programación.
 - Facilidad de acceso.
- Criterios de exclusión:
 - Documentación limitada y falta de apoyo por parte del proveedor del lenguaje.
 - Escasez de ejemplos prácticos como referencia.
 - Se descartó la idea de un desarrollo móvil debido a que eso podría implicar problemas como menos accesibilidad y disponibilidad que un portal web y una interfaz de usuario 3D menos consistente.

2.1.3. Tecnologías para el desarrollo de aplicaciones 3D

A la hora de comenzar el desarrollo de algún tipo de aplicación 3D se han tenido en cuenta dos opciones: desarrollo web o desarrollo móvil.

- **Desarrollo web:** una aplicación web es un software diseñado para poder ser utilizado a través de un navegador web, sin necesidad de descargarlo ni instalarlo. El navegador permite al software utilizar las funcionalidades del dispositivo del usuario para poder darle una experiencia más personalizada y adaptada a las capacidades de su dispositivo. En las aplicaciones web suelen distinguirse tres niveles: el nivel superior que interacciona con el usuario (el cliente web, normalmente un navegador), el nivel inferior que proporciona los datos (la base de datos) y el nivel intermedio que procesa los datos (el servidor web) [32].
- **Desarrollo móvil:** lo que nos ofrece el desarrollo móvil hoy en día es que todo el mundo tiene acceso a un dispositivo móvil, lo que conlleva que sea

más accesible. Además, nos encontramos con una gran variedad de dispositivos, aunque la mayoría de ellos están bajo un sistema operativo Android o iOS. Cada dispositivo tiene configuraciones hardware distintas, lo que hace que a la hora de desarrollar una aplicación móvil pueda ser costoso centrarse en un público en concreto.

Además, uno de los problemas potenciales es que el uso de una aplicación móvil requiere la instalación de esta, por lo que algunos usuarios podrían ser reticentes a probarla debido a que tienen que instalarla. Según [33] **los lenguajes que predominan en aplicaciones móviles** para Android son Java y Kotlin, mientras que en iOS los más utilizados son Objective-C y Swift. Sin embargo, aunque estos lenguajes tienen librerías de gráficos 3D, no suelen estar centrados en hacer entornos 3D sino más bien en el uso de algún elemento 3D básico.

El desarrollo web nos ofrece una gran cantidad de posibilidades donde disponemos de diversos lenguajes de programación y muchas librerías para enriquecer el desarrollo. Según [34] **los principales lenguajes de programación web** son JavaScript, Python, Java, TypeScript y C#. El lenguaje que predomina es JavaScript, y esto es una gran ventaja ya que es el lenguaje base para la mayoría de *frameworks*, lo que nos permitirá que la integración de librerías externas basadas en JavaScript sea lo mejor posible. Además, si investigamos un poco más acerca de las librerías 3D, podemos ver que hay muchas opciones que nos permiten hacer este entorno 3D más avanzado sin necesidad de requerir mucho rendimiento.

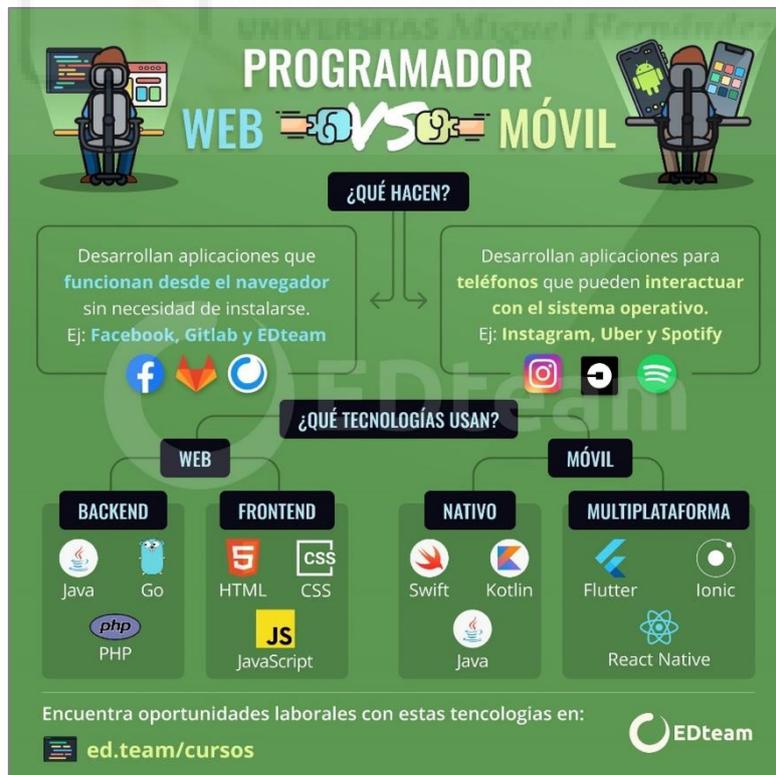


Figura 17: Diferencias entre un programador web y un programador móvil [71].

Aunque inicialmente se consideró un desarrollo móvil, finalmente nos decantamos por el desarrollo web dada la ausencia de aplicaciones centradas en el tratamiento de la depresión, ansiedad y estrés en un entorno accesible a través del navegador. Otro punto a favor del desarrollo de una aplicación web es que cualquier persona que pueda acceder a un navegador web tendrá acceso a la misma. Además, con el desarrollo web podemos resolver el problema de diferentes sistemas operativos o distinto hardware, ya que el navegador se encargará de llevar esa función a cabo y no sería necesario la instalación de la aplicación para probarla.

Una vez analizadas las ventajas e inconvenientes de los distintos tipos de desarrollos, resumimos a continuación las razones que nos han hecho decantarnos por nuestra elección:

- En este proyecto se ha decidido implementar un desarrollo web, ya que consideramos que sería más accesible de cara al público y, así, evitaríamos el problema de que el usuario tuviera que instalar algún tipo de aplicación.
- Teniendo en cuenta la parte de programación, consideramos que sería más completo desarrollar en web porque existen varios lenguajes de programación muy potentes, con una gran comunidad detrás que brinda mucho apoyo al desarrollo de proyectos mediante *frameworks* y librerías.
- Otro de los puntos críticos es que al querer centrarnos en un entorno 3D, observamos que las posibilidades para móvil eran limitadas frente al uso de tecnologías web.

Por enfatizar un poco más esta discusión, elegimos el desarrollo web debido a la amplia libertad que nos brindaba para crear un mundo 3D. Al optar por esta opción, logramos crear una experiencia inmersiva accesible desde cualquier dispositivo con conexión a Internet y un navegador web. Los usuarios podían disfrutar de esta experiencia sin necesidad de descargar o instalar ninguna aplicación adicional, lo cual facilitaba su participación y garantizaba un uso continuo.

Una vez decidido el tipo de desarrollo, en el siguiente apartado se van a explicar detalladamente las tecnologías y *frameworks* para el desarrollo de una aplicación web que han sido escogidos en este trabajo.

2.1.4. Librerías utilizadas: React + ThreeJS + NodeJS

En un proyecto web existen dos partes: el *frontend* y el *backend*. El *frontend* de una aplicación web es la parte visual que el usuario experimenta. En este proyecto nuestro *frontend* está compuesto por React y ThreeJS, tecnologías que nos permiten crear una interfaz interactiva y atractiva para el usuario. El *backend* se encarga de gestionar toda la lógica que se encuentra detrás del *frontend*. En nuestro proyecto el *backend* es responsable de manejar las interacciones del

usuario, como la interacción con elementos del mundo 3D o la realización de sesiones, asegurando una experiencia fluida sin comprometer el aspecto visual.

En nuestro proyecto el *backend* utiliza NodeJS para llevar a cabo estas tareas. Un *framework* es una estructura o marco de trabajo que proporciona una base organizativa para desarrollar proyectos con objetivos específicos. Se puede considerar como una plantilla que sirve como punto de partida para la organización y desarrollo de software [35]. Nos hemos decantado por el uso de las tecnologías React, ThreeJS y NodeJS y su justificación se resumen al final de este apartado.

En la Figura 18 se muestran los logotipos de las diferentes herramientas seleccionadas. La decisión de prescindir de un *framework* para el *frontend* se basa en la elección de aprovechar las capacidades y flexibilidad de React, evitando la complejidad innecesaria y permitiendo un desarrollo más eficiente. Con esta decisión se busca mantener la simplicidad y coherencia en el proyecto.

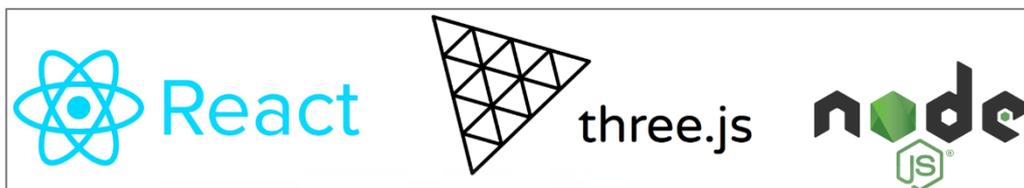


Figura 18: Logotipos de las diversas tecnologías usadas durante el desarrollo web.

Respecto a la parte *frontend*, es un error común pensar que React es un *framework*, pues realmente está catalogado como una librería de JavaScript. Esto es debido a que React depende mucho de librerías externas, lo que conlleva que no sea un marco de trabajo completo en sí, por lo tanto, no se considera un *framework*. En el caso de ThreeJS, sí podemos afirmar que es una librería, ya que su función principal es dotar a nuestro desarrollo web con gráficos 3D. Es decir, se trata de una herramienta complementaria a React. Otro error común es pensar que NodeJS es un *framework* de *backend*. Sin embargo, NodeJS tampoco es considerado un *framework*, ya que un *framework* es un marco de trabajo y NodeJS es un entorno de ejecución. En lo que respecta al *backend*, es verdad que forma parte de este, pero no es algo exclusivo de él. NodeJS se puede utilizar tanto en *frontend* como en *backend*.

- **React** [36]: es una librería de JavaScript que se conoce comúnmente como React o ReactJS. Es una tecnología de código abierto diseñada principalmente para crear interfaces de usuario. React fue desarrollado con el objetivo de simplificar la creación de aplicaciones de una sola página (SPAs) mientras se logra un alto rendimiento. Su enfoque en el desarrollo se asemeja más a la creación de videojuegos que al desarrollo tradicional de aplicaciones. La librería de React es mantenida activamente por Facebook, Instagram y una gran comunidad de desarrolladores independientes y corporaciones [37]. La principales ventajas que nos proporciona React son las siguientes: (1) función de recarga activa (*Hot Reloading*), esto permite al desarrollador poder implementar cambios en el código y ver el efecto en tiempo real, (2) DOM virtual, React utiliza su

propio DOM virtual para actualizar el DOM del navegador de manera eficiente, lo que mejora significativamente el rendimiento de la aplicación, (3) reutilización de los componentes, React permite la creación de componentes reutilizables, lo que acelera el proceso de desarrollo y mejora la organización del código, (4) flexibilidad, React es altamente adaptable y se puede utilizar con otras librerías y *frameworks*, lo que lo hace ideal para aplicaciones de cualquier tamaño y complejidad.

- **ThreeJS** [38]: es una librería de código abierto que existe desde 2010. Permite crear fácilmente escenas 3D con un buen aspecto dentro del navegador. ThreeJS utiliza WebGL para renderizar las animaciones, aunque también puede utilizar HTML5 y SVG. Generar escenas 3D utilizando directamente WebGL o mediante un Canvas resulta una tarea sumamente compleja. Afortunadamente, ThreeJS ofrece una API de uso sencillo que permite crear y manipular objetos y escenarios en 3D sin requerir un conocimiento profundo de WebGL o fórmulas matemáticas complejas [39]. Lo más destacable de la librería es que al usar ThreeJS no se requiere un ordenador de gama alta para poder mostrar gráficos 3D fotorrealistas. De igual forma, no es necesario descargar ninguna aplicación especial para que funcione. Ahora, cualquier persona puede experimentar de gráficos 3D detallados con tan solo usar un navegador web [40].
- **NodeJS** [41]: es un entorno de tiempo de ejecución de JavaScript basado en eventos y asíncrono que está diseñado para la construcción de aplicaciones de red escalables. NodeJS permite mantener muchas conexiones que pueden ser manejadas simultáneamente, en el caso de que NodeJS no tenga ningún proceso pendiente entra en modo suspensión. Este modelo de concurrencia es diferente al modelo de concurrencia más común de la actualidad, el cual se basa en hilos de Sistema Operativo (OS *Threads*). Este enfoque de conexión de redes basado en hilos es relativamente ineficiente y muy difícil de utilizar. Además, los usuarios de NodeJS no necesitan preocuparse por la posibilidad de bloqueo del proceso, ya que no se emplean bloqueos en este entorno. En NodeJS, prácticamente ninguna función realiza operaciones de entrada/salida (I/O) de manera directa, por lo que el proceso nunca se bloquea, excepto cuando se usan los métodos síncronos de la librería estándar de NodeJS para realizar operaciones de I/O. Debido a la inexistencia de bloqueos, el desarrollo de sistemas escalables resulta muy factible en NodeJS. Una de las principales ventajas a destacar es la gran integración con módulos externos, ya que se pueden instalar módulos creados por la comunidad afines a NodeJS, los cuales nos permiten extender la funcionalidad de nuestro proyecto.

Se ha decidido escoger estas tecnologías porque son ideales para nuestro proyecto. Para el *frontend*, tenemos React y ThreeJS, donde usaremos React para la base y ThreeJS para todo el contenido 3D. El uso de React es debido a lo mencionado anteriormente, funciona muy bien en los navegadores y es muy

eficiente. Cabe destacar que tiene el apoyo de una gran comunidad y de Facebook, lo que significa que va a estar recibiendo actualizaciones constantemente y no se va a quedar obsoleto. En este proyecto estaremos utilizando React 18, que fue la última versión lanzada en junio de 2022. Otra parte fundamental es ThreeJS, ya que toda la parte interactiva recaerá en esta librería. Decidimos usarla por encima de las otras ya que, al igual que pasa con React, funciona muy bien en el navegador y es justo lo que buscamos: poder realizar un entorno 3D interactivo fotorrealista accesible para todo el mundo sin necesidad de instalar ninguna aplicación. La librería ThreeJS también nos permite poder integrar modelos 3D de Blender, Cinema 4D o Houdini, entre muchos otros. Esta librería nos ofrece una gran libertad para poder desarrollar el proyecto. Por ello creemos que estas dos herramientas funcionan muy bien de forma conjunta. La versión de ThreeJS que vamos a utilizar es la r150, lanzada en febrero de 2023. Por último, para este proyecto utilizaremos NodeJS como *backend*. NodeJS nos da la opción de poder instalarle módulos complementarios para mejorar la integración del proyecto, gracias a estos módulos podemos conectar el *backend* de nuestro proyecto con nuestra base de datos y poder gestionar la transferencia de datos entre el *frontend* y el *backend* de forma sencilla, algunos de estos módulos pueden ser *frameworks* que nos ayuden a realizar esta tarea. La versión de NodeJS que estamos utilizando para el proyecto es la 18.12.1 lanzada en noviembre de 2022.

2.1.5. Elementos de React

Para poder realizar una implementación correcta del código es necesario conocer la funcionalidad de las herramientas que nos brinda el lenguaje. Primero analizaremos el uso de React. En el caso de React todos los componentes heredan de la clase de React y tienen asociado un ciclo de vida que se divide en diferentes fases, que son fundamentales para comprender el proceso de montaje, actualización y desmontaje de los componentes, tal y como podemos ver en la Figura 19.

A continuación, dividiremos las fases por puntos para explicar la importancia de cada una de ellas:

1. **Fase de montaje:** en esta fase, el componente de React se crea y se inserta en el DOM, aquí es donde se ejecutan los siguientes métodos. El método **constructor** es el encargado de inicializar el estado del componente, el **renderer** es el encargado de renderizar el componente en un elemento de React y por último el método **componentDidMount** se llama justo después de que el componente se haya renderizado en el DOM, este sirve para establecer suscripciones y llamadas API.
2. **Fase de actualización:** esta fase es la encargada de activarse siempre y cuando un componente experimente algún tipo de cambio, justo en ese momento se llama al método **shouldComponentUpdate**, este método nos permite controlar si el componente debe renderizarse de nuevo o no. En el caso de que haya habido algún tipo de cambio se llama directamente

al método **componentDidUpdate** después de haber aplicado los cambios en el DOM, este sirve para actualizar el estado del componente o incluir nuevas llamadas API.

3. **Fase de desmontaje:** Esta fase se produce cuando un componente se elimina del DOM. Una vez se haya eliminado del componente se llama al método **componentWillUnmount**, este se encargará de cancelar las solicitudes API y realizar limpieza de las suscripciones.

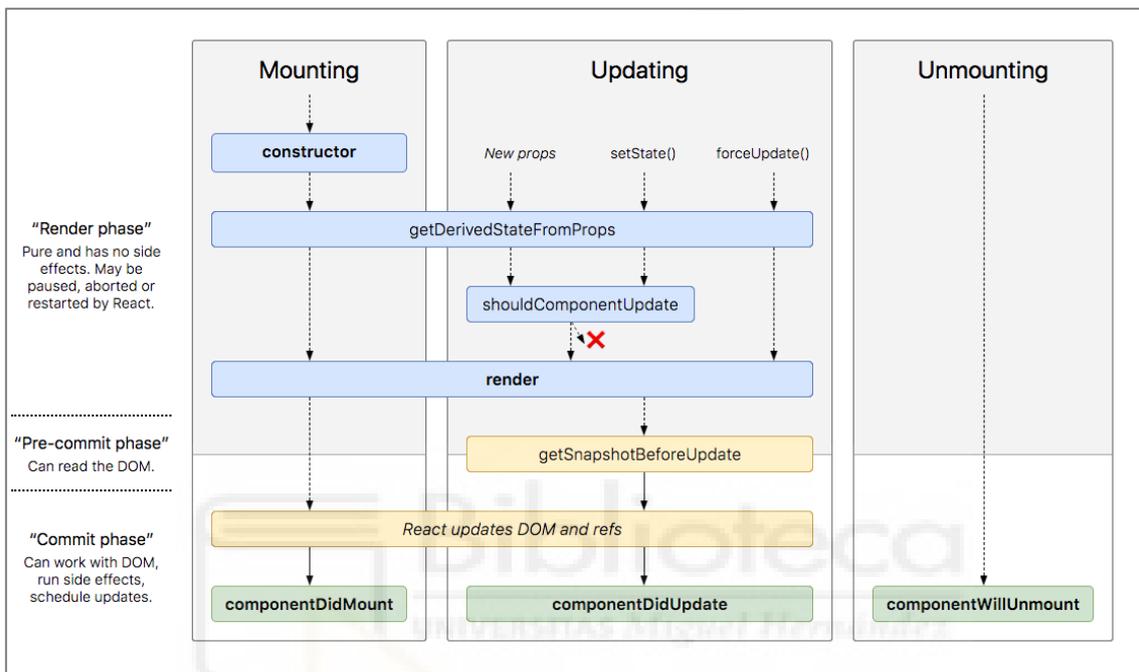


Figura 19: Ciclo de vida de los componentes de React v16.4. [42].

Estas fases y métodos las podemos ver en la Figura 19, nos permiten poder controlar el ciclo de vida de los componentes en React y realizar acciones específicas en cada una de las fases. Es importante comprender este funcionamiento ya que nos puede librar de futuros problemas al montar componentes.

Tras haber explicado el ciclo de vida de un componente en React para poder comprender el funcionamiento básico hay que hablar de los *hooks*. Los *hooks* son una característica que fue añadida tras el lanzamiento de la versión 18.6 [43] que nos da la posibilidad de aprovechar el potencial de los componentes de React sin necesidad de crear clases. Los *hooks* son funciones especiales que se llaman dentro de los componentes de React para agregarles funcionalidad, podemos ver algunos de los *hooks* más comunes en la Figura 20.

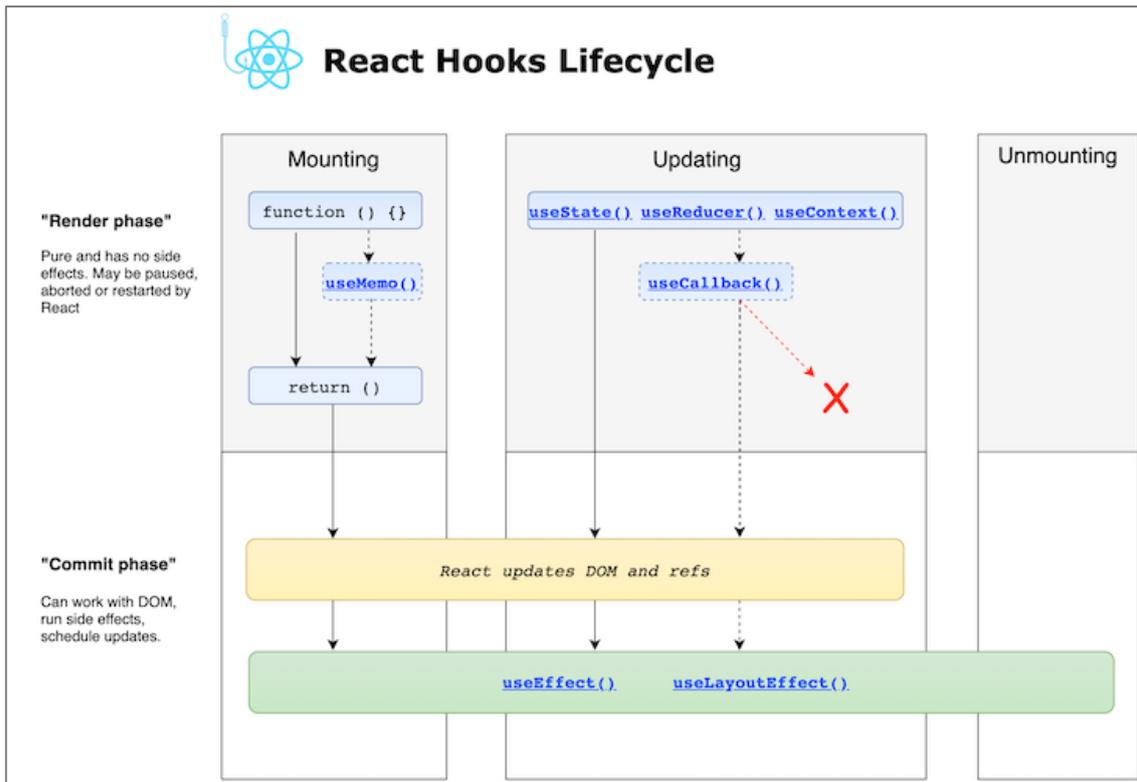


Figura 20: Ciclo de vida de los hooks de React [72].

1. **useState:** este *hook* nos proporciona una forma de declarar y reutilizar variables dentro del estado del componente.
2. **useEffect:** este *hook* nos permite realizar llamadas API, suscribirnos a eventos y manipular el DOM. UseEffect se ejecuta una vez renderizado o actualizado el componente.
3. **useContext:** este *hook* nos proporciona una forma de pasar datos globalmente a través del árbol de componentes.

Estos son solo algunos ejemplos de los hooks que nos proporciona React, pero hay muchas más como por ejemplo **useCallback**, **useReducer**, **useMemo**, entre otros.

Por último y antes de mostrar como he usado dichos métodos en nuestro código queremos hablar brevemente de los *props*. Los *props* en React son mecanismos que nos permiten pasar datos y funcionalidades de un componente padre a un componente hijo. Los *props* mejoran la cohesión del código estableciendo una comunicación clara y estructurada entre los componentes.

Tras comentar algunas de las funcionalidades de React, podemos concluir que es la librería más apropiada para este proyecto. La forma en que React se encarga de los estados y proporciona funciones para su control resulta fundamental para nosotros.

Si lo comparamos con una aplicación web genérica que utilice HTML5, CSS y JavaScript, vemos que React sigue siendo claramente el ganador en este sentido. Una aplicación web genérica que utilice HTML5, CSS y JavaScript puede funcionar correctamente para ciertos proyectos más simples y estáticos. Sin embargo, cuando se trata de proyectos más complejos que involucran un manejo dinámico de estados y una interfaz de usuario interactiva, React ofrece ventajas significativas sobre una web genérica como hemos visto a lo largo de este apartado.

2.1.6. Lenguajes utilizados: JavaScript

A continuación, describiré cuales es el lenguaje de programación que se ha utilizado en este proyecto. La elección del lenguaje ha sido realmente clara, ya que se ha visto influenciadas por las tecnologías descritas anteriormente. Al escoger las librerías de React y ThreeJS solo podíamos usar JavaScript, ya que están escritos en dicho lenguaje. Lo mismo ocurre con NodeJS, tanto el propio entorno como sus módulos están escritos en JavaScript, por ende, era la única opción disponible para cumplir nuestros requisitos.

JavaScript [44] es un lenguaje de programación interpretado que se caracteriza por ser orientado a objetos y permitir la implementación de funciones complejas. Además, se ha tomado la decisión de utilizar la extensión JSX. La sintaxis que podemos ver en la Figura 21 es JSX [45], que es una extensión de JavaScript sugerida de ser utilizada en conjunto con React para definir la apariencia y funcionalidad de la interfaz de usuario. Aunque la sintaxis pueda ser parecida a XML, JSX presenta todas las funcionalidades de JavaScript.

React reconoce que la lógica de representación gráfica está íntimamente ligada con la lógica de la interfaz de usuario. Esto implica cómo se gestionan los eventos, cómo evoluciona el estado con el tiempo y cómo se procesan los datos para ser mostrados. Aunque no es obligatorio utilizar JSX en React, la mayoría de las personas lo encuentran beneficioso como herramienta visual para trabajar con la interfaz de usuario en el código JavaScript. Además, esto permite que React muestre mensajes de error o advertencia más informativos [46].

Otra ventaja que tiene JSX con respecto a JavaScript son los *hooks* explicados anteriormente. Para que haya una mejor claridad en el proyecto, todos los archivos de React llevarán extensión JSX y el resto de los archivos como los de NodeJS tendrán extensión JavaScript.

```
const element = <h1>Hello, world!</h1>;
```

Figura 21: Ejemplo de código de JSX [45].

2.1.7. Herramientas de base de datos: MySQL.

A la hora de elegir la base de datos tuvimos varias opciones, hacer una base de datos relacional (SQL) o una base de datos no relacional (NoSQL). En el caso de que la base de datos fuera no relacional pensamos en utilizar mongo DB, ya que es una de las bases de datos más utilizadas para el NoSQL. Por otra parte, para una base de datos relacional escogimos MySQL [47].

Tras haber estudiado detalladamente cada opción, llegamos a la conclusión de que era mejor utilizar una base de datos relacional. La principal ventaja del uso de mongo DB, es el análisis de datos en tiempo real, ya que manejan de forma eficiente la información no estructurada como documentos, correos electrónicos, multimedia y las redes sociales [48]. Todas esas características de Mongo DB no iban a ser aprovechadas en nuestro proyecto, por lo que decidimos usar una base de datos relacional.

Decidimos usar MySQL porque vimos que se podía integrar fácilmente en nuestro proyecto y las herramientas de trabajo que nos ofrecía eran muy sencillas, como MySQL Workbench y el asistente de diagramas E/R. Como ya he comentado anteriormente, al escoger MySQL tenemos que implementar un diseño de una base de datos relacional, eso se consigue usando SQL. El SQL [49] es un lenguaje que se utiliza para definir, manipular y controlar bases de datos relacionales y es un estándar ANSI/ISO. A diferencia de los lenguajes procedimentales, el SQL es un lenguaje declarativo, lo que significa que sólo se tiene que indicar lo que se desea hacer y no cómo hacerlo. El SQL es muy similar al inglés y es altamente expresivo. Es un lenguaje estándar, lo que permite acceder a todos los sistemas relacionales comerciales. Al ser un lenguaje estándar facilitaría el proceso de compartir datos estadísticos para posteriores estudios psicológicos, siempre y cuando el usuario este de acuerdo con ello.

2.1.8. Módulos de NodeJS para la conexión *frontend-backend*

Tras haber comentado cuales son las tecnologías que vamos a utilizar para el *frontend* y el *backend*, a continuación, describiremos como comunicar ambas partes para que los datos que el usuario nos facilite en el entorno 3D se pueden guardar en la base de datos.

Como se ha comentado todo el *backend* estará gestionado por NodeJS y sus módulos, los módulos fundamentales para el *backend* son: Express, MySQL, Cors y Body-Parser. Comentamos cada uno de ellos a continuación:

- **Express** [50]: es un *framework* para aplicaciones web de NodeJS que es simple, adaptable y ofrece una amplia variedad de middleware que facilita la tarea de gestionar solicitudes y respuestas HTTP. Entre sus múltiples ventajas, podemos encontrar las siguientes: (1) permite crear aplicaciones web de manera rápida y eficiente, (2) es altamente adaptable y escalable, y (3) es compatible con una gran cantidad de paquetes y herramientas de NodeJS.

- **MySQL** [51]: el paquete MySQL para NodeJS es un controlador escrito en JavaScript que no requiere compilación y tiene una licencia MIT. Su objetivo es conectar la base de datos con el framework Express, permitiendo que el *backend* tenga acceso a la información almacenada en la base de datos.
- **Cors** [52]: es uno de los *middlewares* que ofrece el *framework* de Express para el correcto funcionamiento de nuestro proyecto. Un *middleware* es un componente intermedio entre el *frontend* y el *backend*, que agrega funcionalidades a las solicitudes y respuestas HTTP en una aplicación web para mejorar su comportamiento y rendimiento. Cors establece las reglas de acceso y comunicación entre diferentes dominios y recursos del servidor, lo que garantiza que la aplicación sea segura y confiable. Además, permite la transferencia de *cookies* y mantiene las sesiones activas, lo que significa que una vez que el usuario inicie sesión, su sesión se mantendrá activa mientras el *token* de autenticación sea válido.
- **Body-parser** [53]: al igual que cors, body-parser es un *middleware* de Express que permite extraer los datos de una solicitud enviada por el cliente en el cuerpo de la petición. Una vez que se extraen los datos, body-parser los convierte en un objeto JavaScript fácilmente manipulable por la parte del servidor. Este objeto JavaScript se puede utilizar para realizar cualquier tipo de operación o manipulación de datos necesarios en la aplicación web.

Para garantizar una comunicación eficaz entre el *frontend* y el *backend* de una aplicación web, es necesario instalar una librería en la parte *frontend*. Una de las librerías más populares y ampliamente utilizadas es Axios. Esta librería proporciona una forma fácil de realizar solicitudes HTTP desde el navegador y se integra perfectamente con el *framework* de Express.

- **Axios** [54]: es una biblioteca de JavaScript que permite realizar solicitudes HTTP desde el navegador del cliente al servidor en NodeJS. Esto hace posible que los datos se compartan y se manipulen fácilmente entre el cliente y el servidor, dando como resultado que nuestra aplicación web sea más dinámica e interactiva y mejore la experiencia del usuario.

En la Figura 22 se muestra un ejemplo de cómo se conectaría nuestro sistema. Si un usuario desea iniciar sesión, debe ingresar sus credenciales en nuestra aplicación web con React. Luego, cuando envíe el formulario, la solicitud se enviará a nuestro servidor NodeJS a través de la librería Axios, la cual está configurada con la URL de la API y los parámetros necesarios (nombre de usuario y contraseña). En el servidor NodeJS, Express recibirá la solicitud y utilizará Cors para garantizar el origen de la conexión y permitir el acceso solo a los recursos permitidos. Body-parser extraerá los parámetros de la solicitud de inicio de sesión y convertirá los datos en un objeto JavaScript. A continuación, Express comprobará los datos en nuestra base de datos utilizando el *driver*

MySQL. Si las credenciales son correctas, se generará un *token* de autenticación y Express lo recibirá. Luego, el servidor NodeJS responderá al usuario de nuestra aplicación React con el *token*. El usuario guardará el *token* de autenticación en su estado local y lo utilizará en las solicitudes futuras para verificar la autenticación. La sesión se mantendrá abierta mientras el *token* sea válido. En caso de que el *token* caduque o se vuelva inválido, se requerirá que el usuario inicie sesión nuevamente.

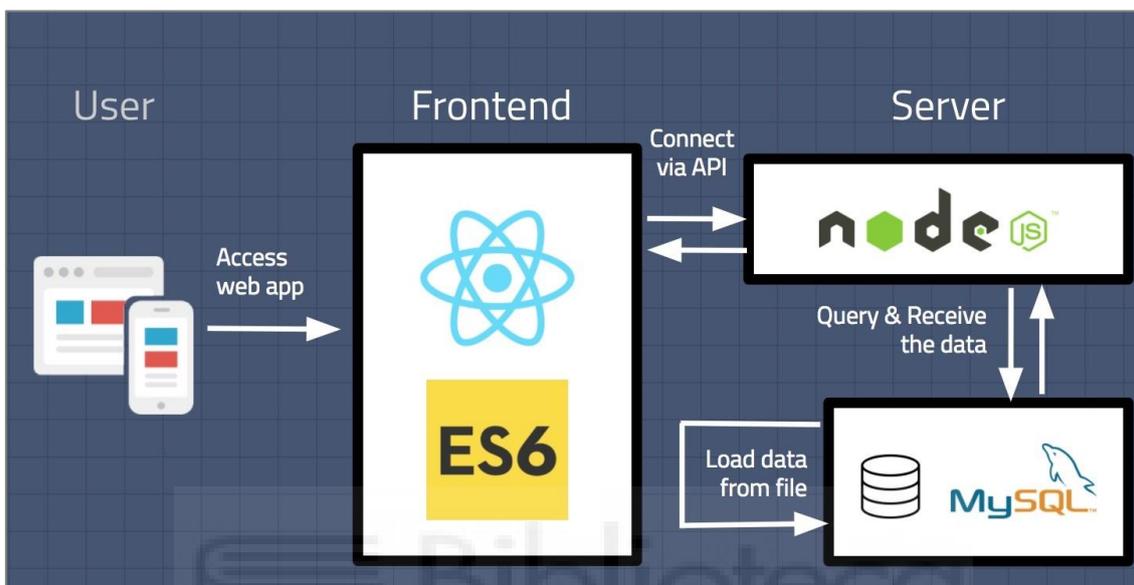


Figura 22: Ejemplo de comunicación entre el frontend y el backend [55].

2.1.9. Herramientas y modelos del mundo 3D

Three.js nos permite integrar modelos 3D en nuestro entorno virtual, lo cual es fundamental para nuestro proyecto web. Esto nos brinda la capacidad de diseñar un entorno visualmente atractivo al proporcionar una amplia variedad de elementos en el mundo virtual. Para facilitar esta tarea, ThreeJS ofrece diversas funciones, siendo la más utilizada y la que hemos empleado en este proyecto la **GLTFLoader**. Esta función nos permite cargar modelos con formato de transmisión GL, es decir, archivos con extensión .glTF o .glb. En la documentación de Three.js se sugiere utilizar este formato debido a que glTF se enfoca en la entrega eficiente de recursos en tiempo de ejecución, con transmisión compacta y carga rápida. Incluye información sobre mallas, materiales, texturas, esqueletos, animaciones, luces y cámaras. De esta forma, tenemos la capacidad de manipular los modelos de acuerdo con nuestras necesidades. Podemos aplicar mayor iluminación, cambiar los colores de las mallas e incluso añadir animaciones si así lo deseamos.

El uso del formato GL no solo es fundamental para cargar modelos en nuestro mundo virtual, sino que también es ampliamente compatible con la mayoría de los editores de modelos 3D, como Blender, Houdini, Cinema 4D, entre otros [56]. Podemos encontrar una lista completa de las herramientas compatibles con este formato en la página del glTF Project Explorer [57]. Sin embargo, en caso de no

querer utilizar el formato GL, ThreeJS nos brinda la opción de utilizar otros cargadores de modelos, como FBXLoader o OBJLoader. Aunque estos cargadores no están desarrollados específicamente como GLTFLoader, aún se pueden utilizar para cargar modelos 3D en la aplicación.

Para el desarrollo de este proyecto hemos empleado modelos 3D tanto de páginas gratuitas como modelos 3D creados por nosotros. Para la creación de estos modelos 3D se ha utilizado la herramienta Blender [58].

Blender es un software de modelado, animación y renderización en 3D gratuito de código abierto. Se utiliza en la industria del cine, los videojuegos y la animación para crear contenido digital. Con Blender, se pueden crear modelos 3D, animar objetos y personajes, simular efectos físicos, aplicar texturas y renderizar imágenes y animaciones finales. Gracias a esta herramienta, pudimos crear los modelos necesarios para nuestra aplicación y utilizar su funcionalidad de exportación en formato GL para integrar directamente los modelos en nuestro entorno virtual. Esto nos ha brindado la capacidad de trabajar de manera eficiente y sin problemas en la incorporación de los modelos a la aplicación.

Para los demás modelos del entorno virtual, hemos recurrido a la página web Poly.pizza [59]. Esta plataforma nos ha permitido descargar modelos 3D con estética *low poly* de forma gratuita. Como se puede apreciar en la Figura 23, la página ofrece tanto paquetes de modelos como modelos individuales. La mayoría de los modelos han sido creados por la comunidad, aunque también se pueden encontrar modelos 3D creados por Google. Una de las principales ventajas de utilizar esta plataforma es que los modelos son gratuitos y vienen en formato GL, lo cual los hace ideales para integrarlos en nuestro entorno virtual.

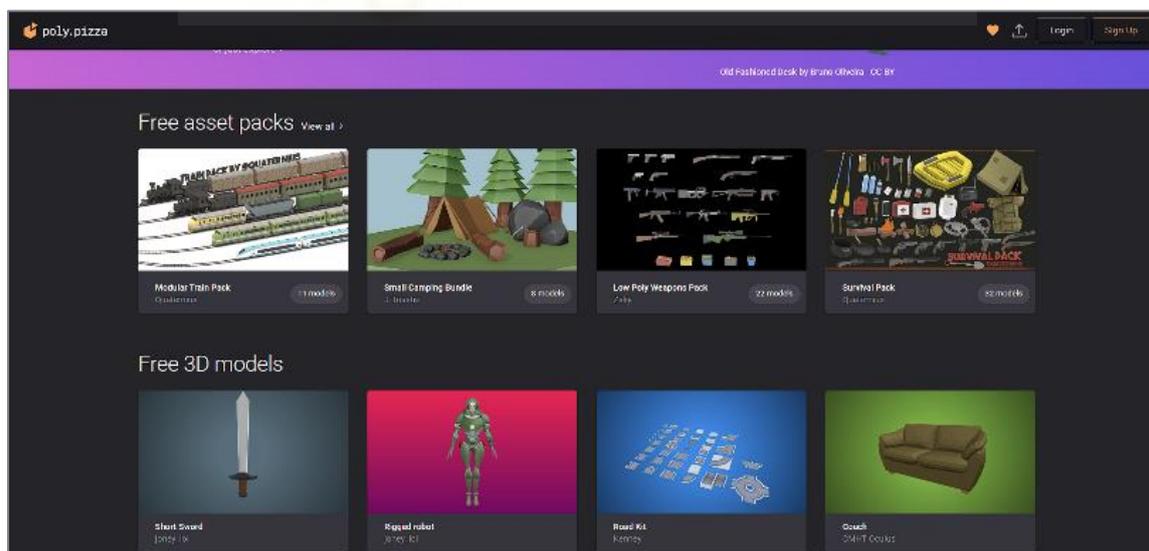


Figura 23: Modelos de la página web Poly.pizza [59].

Es importante tener en cuenta que en un entorno 3D es necesario utilizar un motor de físicas y de colisiones para garantizar consistencia y realismo. Esto es crucial para evitar que los modelos se perciban como huecos y brinden una experiencia de realidad falsa cuando los usuarios interactúen con ellos. Nuestro

objetivo es lograr que los usuarios se sientan cómodos en un mundo virtual que puedan explorar e investigar. Al implementar modelos con físicas adecuadas, aseguramos que el entorno se comporte de manera realista y evitamos generar desconfianza en nuestros usuarios.

2.2. Entornos de desarrollo (IDEs) y depuración

En este apartado explicaremos los elementos necesarios para realizar las tareas de desarrollo, programación y prueba de la aplicación. Para el desarrollo de código y gestión de la base de datos hemos utilizado:

- **Visual Studio Code** [60]: es un programa para editar código fuente que tiene la particularidad de ser ligero, pero con gran capacidad. Este software puede ser instalado en nuestro escritorio y está disponible para sistemas operativos como Windows, macOS y Linux. Además, cuenta con funcionalidades predefinidas para trabajar con lenguajes de programación como JavaScript, TypeScript y Node.js. También, ofrece una amplia variedad de extensiones que pueden ser utilizadas para trabajar con otros lenguajes y entornos de programación, como C++, C#, Java, Python, PHP, Go y .NET.
- **MySQL Workbench** [61]: como se ha mencionado previamente, MySQL Workbench es un software que ofrece una interfaz gráfica unificada para facilitar el trabajo de arquitectos de bases de datos, desarrolladores y administradores de bases de datos. Este programa proporciona herramientas para modelar datos, desarrollar código SQL y llevar a cabo tareas de administración de manera detallada, como la configuración del servidor, la gestión de usuarios, la realización de copias de seguridad, entre otras. Además, es posible utilizar MySQL Workbench en sistemas operativos como Windows, Linux y Mac OS X.
- **Bitbucket** [62]: es una plataforma en la nube para alojar y gestionar repositorios de código fuente. Compatible con Git y Mercurial, Bitbucket ofrece características como repositorios privados y públicos, integración con herramientas de desarrollo, revisión de código, integración con servicios en la nube, y control de acceso y permisos. Es una herramienta versátil y poderosa para colaborar en proyectos de desarrollo de software

Aunque estas tecnologías son necesarias para el desarrollo de nuestro proyecto, no son las óptimas para depurar posibles fallos en el código. Si ocurre un error durante la ejecución de la aplicación, en el navegador se mostrará una alerta en el mismo, pero a menudo esta alerta no proporciona información suficiente para entender el problema. Por esta razón, es recomendable instalar una extensión para Google Chrome llamada React Developer Tools [63], tal y como podemos ver en la Figura 24, para poder inspeccionar y cambiar directamente los componentes que se están renderizando en pantalla, de esta forma se puede identificar el componente que está fallando. Además, es importante probar nuestra aplicación en la mayoría de los navegadores web para garantizar que

funcione correctamente en todos ellos y que cualquier usuario pueda acceder a nuestro proyecto.



Figura 24: Herramienta para depurar componentes de React en la web.

2.3. Propuesta de solución

Tras exponer las tecnologías a usar y los dispositivos disponibles, se muestra en la Figura 25 un esquema de la relación entre los dispositivos y tecnologías usadas.

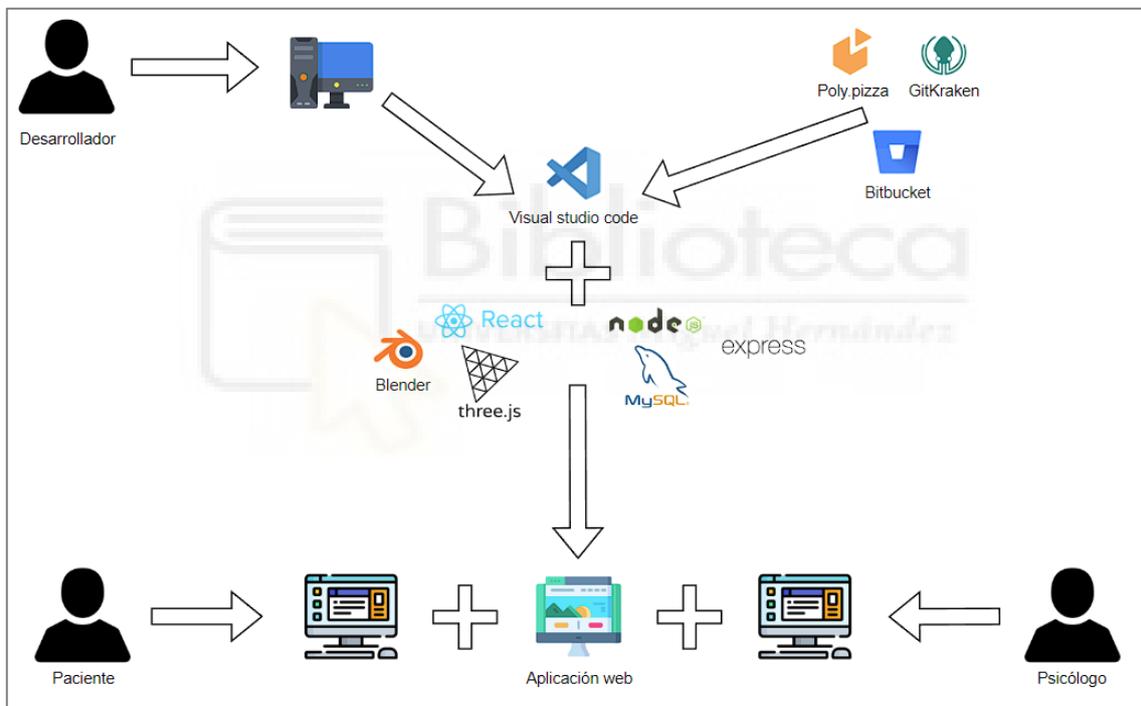


Figura 25: Diagrama con la propuesta de hardware y software de nuestra solución.

3.- Metodología y Resultados

En este capítulo, se van a tratar aspectos técnicos relacionados con la implementación del proyecto, desde su planificación y metodología hasta algunos aspectos relevantes de ingeniería de software y de la aplicación web implementada.

3.1. Scrum como metodología ágil para la gestión del proyecto

Para el desarrollo de este proyecto se utilizó la metodología ágil Scrum [64]. Scrum es ideal para proyectos web debido a su flexibilidad para adaptarse a cambios frecuentes, su enfoque en la entrega continua de valor y su énfasis en la colaboración y comunicación entre los miembros del equipo. Permite responder rápidamente a las necesidades del usuario y garantizar una experiencia en línea satisfactoria.

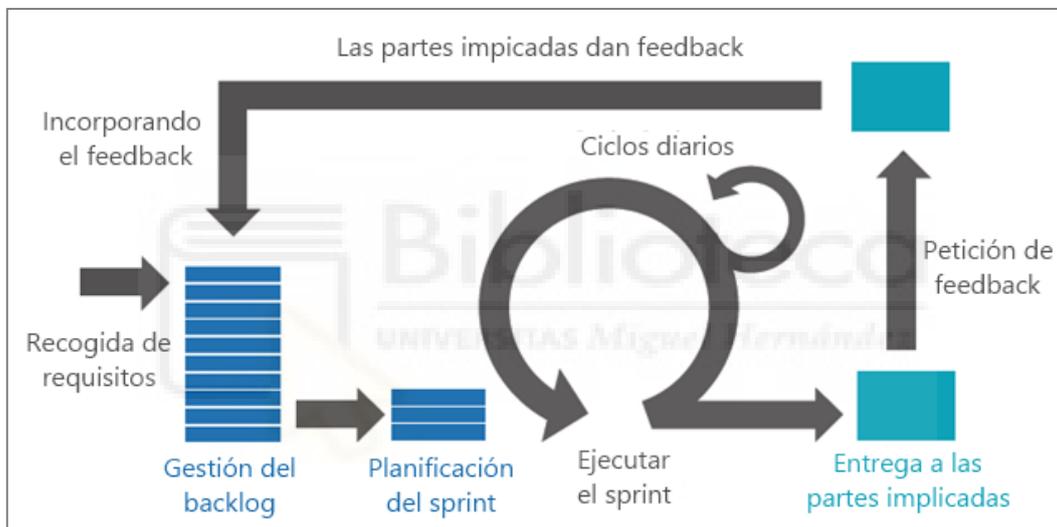


Figura 26: Diagrama de metodología Scrum.

Scrum es una metodología ágil que organiza el trabajo en iteraciones o *sprints*. En la Figura 26 se muestran las fases que conforman esta metodología. A continuación, explicaremos cada una de ellas:

1. **Recogida de requisitos:** se lleva a cabo una reunión para identificar las nuevas características que se pueden implementar en el desarrollo de la aplicación web.
2. **Planificación del sprint:** con base en los requisitos recopilados, se planifica el *sprint* para asegurar que los requisitos se cumplan de manera efectiva.
3. **Ciclos diarios:** aunque no se siguieron ciclos diarios estrictos, se mantuvo una comunicación casi diaria para recibir *feedback* sobre el progreso individual de cada miembro del equipo.

4. **Ejecución del *sprint*:** mediante los ciclos diarios nos asegurábamos de que la progresión del equipo fuese la correcta y que ningún integrante se quedase muy atrasado con sus objetivos.
5. **Entrega de las partes implicadas y *feedback*:** al finalizar cada *sprint*, se realizaba una reunión para revisar los objetivos del proyecto. Luego, se definían los objetivos para el siguiente *sprint*, priorizándolos según su importancia y duración.

Como ya he comentado anteriormente, aunque no tuvimos *feedback* diario, los tres estudiantes del equipo interdisciplinar manteníamos una comunicación casi diaria a través de aplicaciones de mensajería debido a la dificultad de reunirnos todos los días. Después de recopilar los requisitos, continuamos teniendo estas reuniones a medida que íbamos entregando gradualmente las partes del trabajo. La compañera de psicología me proporcionaba las instrucciones necesarias para llevar a cabo las sesiones, incluyendo elementos multimedia como videos, imágenes y audio. Con la compañera de comunicación y diseño, manteníamos conversaciones para discutir el diseño del mundo 3D, donde ella me proporcionaba algunos de los modelos utilizados y hacíamos hincapié en los elementos personalizables, como colores y tipografía. Puedo afirmar que esta metodología basada en Scrum se mantuvo hasta las últimas etapas del desarrollo de la aplicación web.

En la Tabla 2 se muestra un cronograma específico de las tareas relacionadas con el desarrollo del software. En él se puede ver de manera aproximada los diferentes *sprints* y su duración. A continuación, mostrare algunos de los bloques más importantes llevados a cabo durante este proyecto:

- **Bloque 1:** Análisis del estado del arte
 - 1.1. Análisis de contenido del entorno virtual.
 - 1.2. Elección de las actividades del entorno virtual.
 - 1.3. Investigación y elección de las librerías y tecnologías a usar.
- **Bloque 2:** Diseño del entorno virtual
 - 2.1. Instalación y conexión del *backend* con el *frontend*.
 - 2.2. Creación de la interfaz del psicólogo.
 - 2.3. Creación del entorno virtual vacío y controles primera persona con cámara.
 - 2.4. Gestión de los datos del usuario y cuestionario de personalización.
 - 2.5. Integración de modelos 3D en el entorno virtual.
 - 2.6. implementación de físicas.
- **Bloque 3:** Implementación de las sesiones
 - 3.1. Obtención de recursos digitales para las sesiones.
 - 3.2. Análisis de la interactividad de los modelos con el usuario.
 - 3.3. Creación de las sesiones y gestión de sus datos.

- **Bloque 4:** Interactividad entre psicólogo y paciente
 - **4.1.** Creación de la interfaz de comunicación entre paciente y el psicólogo.
 - **4.2.** Gestión de mensajes entre usuarios en la base de datos.
 - **4.3.** Implementación del seguimiento de las actividades en la interfaz del psicólogo.

- **Bloque 5:** Documentación y memoria del proyecto
 - **5.1.** Escribir la memoria del TFG.
 - **5.2.** Documentar gráficamente la aplicación y grabación de la demo.

Tareas	S 1	S 2	S 3	S 4	S 5	S 6	S 7	S 8	S 9	S 10	S 11	S 12	S 13	S 14	S 15	S 16	S 17	S 18	S 19	S 20	S 21	S 22		
Bloque 1																								
1.1.																								
1.2.																								
1.3.																								
Bloque 2																								
2.1.																								
2.2.																								
2.3.																								
2.4.																								
2.5.																								
2.6.																								
Bloque 3																								
3.1.																								
3.2.																								
3.3.																								
Bloque 4																								
4.1.																								
4.2.																								
4.3.																								
Bloque 5																								
5.1.																								
5.2.																								

Tabla 2: Cronograma de las distintas fases desarrolladas.

3.2. Análisis y diseño de software.

A continuación, vamos a adentrarnos en los casos de uso, la base de datos y la estética y modelos de nuestra aplicación. Estos aspectos son fundamentales para el desarrollo y funcionamiento adecuado del proyecto. A lo largo de esta sección, exploraremos cada uno de estos elementos, destacando su importancia y cómo contribuyen al éxito de nuestra aplicación.

3.2.1. Requerimientos y casos de uso.

En esta sección, se presentará un diagrama de los casos de uso más relevantes para el funcionamiento de la aplicación. El objetivo es representar las necesidades y requerimientos identificados a lo largo del proyecto de manera resumida.

En la Figura 27 y en la Figura 28, se presentan una selección de los 25 casos de uso que han sido desarrollados y detallados en el Anexo 1. En esta selección se pretende resumir, con el objetivo de no extender demasiado este apartado, la interacción general del paciente con la aplicación, se han omitido por ejemplo los casos de uso relacionados con las actividades y sesiones que sí se pueden encontrar en el anexo.

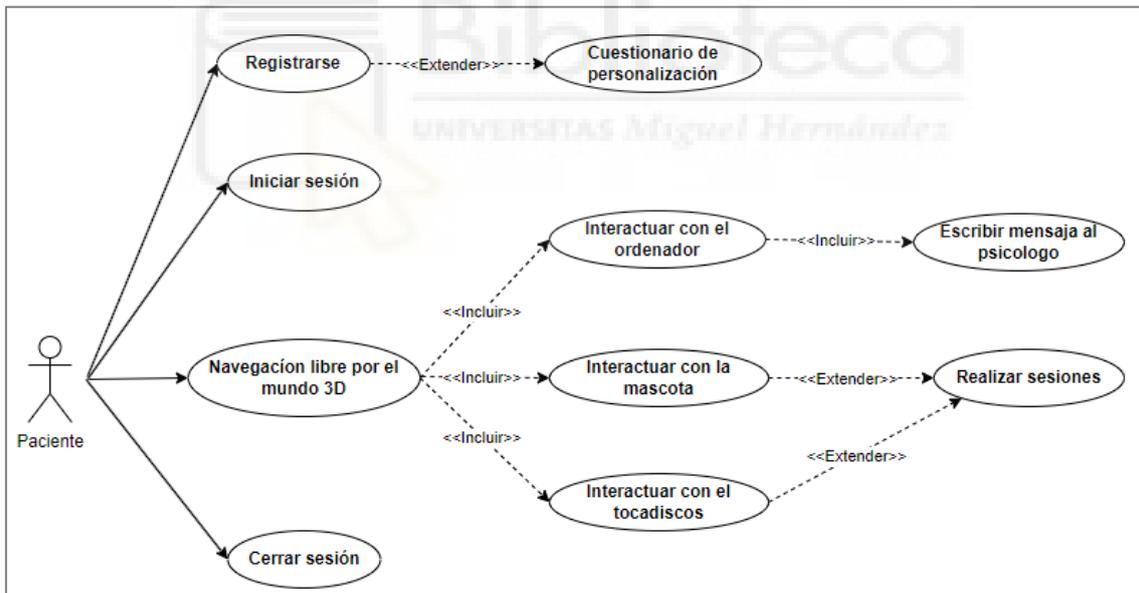


Figura 27: Diagrama de casos de uso del paciente.

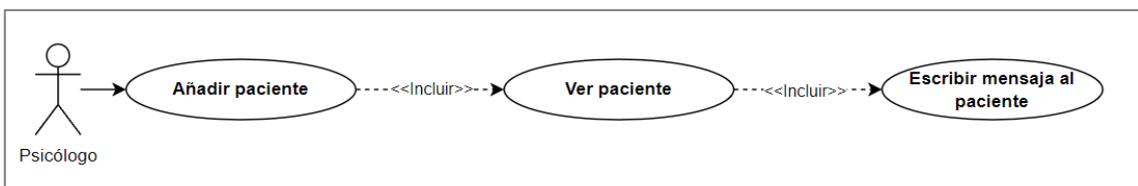


Figura 28: Diagrama de casos de uso del psicólogo.

3.2.2. Diagrama de flujo de navegación del entorno virtual

En la Figura 30 se muestra el proceso de realización de una sesión en el entorno virtual. Al iniciar sesión en la aplicación web, si es la primera vez que el usuario accede al entorno virtual, se le solicitará que complete un cuestionario de personalización. Una vez que el usuario haya completado el cuestionario, podrá acceder al entorno virtual. Una vez dentro, al interactuar con la mascota, se activará la sesión correspondiente que el usuario tenga pendiente.

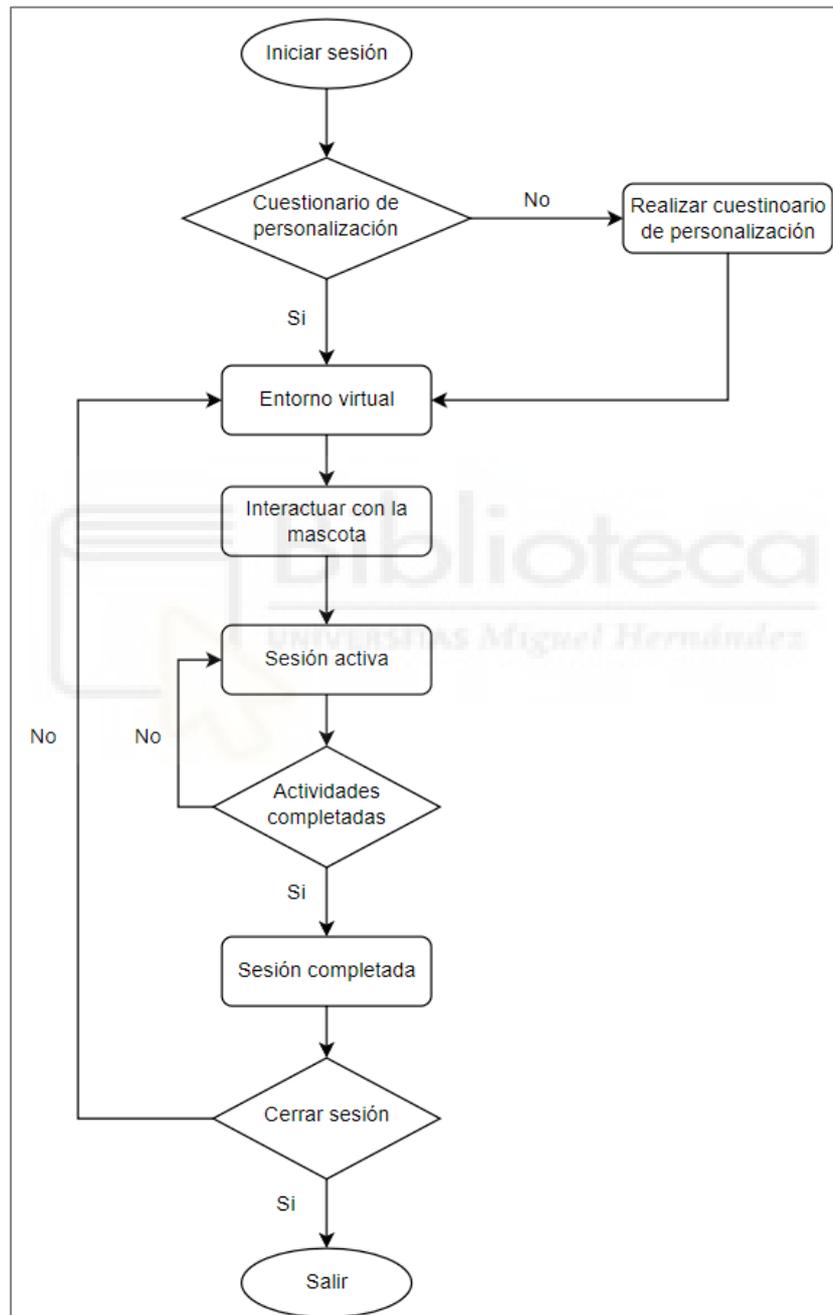


Figura 30: Diagrama de flujo de una sesión en el entorno virtual.

Las sesiones han sido diseñadas completamente por la compañera de psicología, y las hemos integrado en el entorno de nuestra aplicación web. Es relevante tener en cuenta que algunas sesiones incluyen varias actividades, y

algunas de estas actividades son compartidas por varias sesiones. A continuación, se resumen brevemente en qué consiste una sesión de depresión. En la sección 3.4 se proporcionarán más detalles sobre las sesiones en sí.

- Una vez que el paciente haya seleccionado realizar una sesión de depresión, la próxima vez que interactúe con una mascota se iniciará automáticamente la primera sesión de depresión.
- Al comienzo de cada sesión, se proporcionará al paciente una breve introducción que incluirá instrucciones y se abordarán abiertamente temas relacionados con la depresión. Una vez que el paciente haya leído la explicación, comenzará a participar en las actividades específicas de la sesión.
- En algunas sesiones se llevarán a cabo meditaciones guiadas mediante audio o video. En otras ocasiones, se realizarán actividades como cuestionarios, selección de actividades gratificantes y asociación de sonidos con imágenes, entre otras opciones.
- Una vez que el paciente haya completado una actividad, se le brindará un breve resumen de lo que ha logrado durante esa actividad, junto con consejos para continuar trabajando en su tratamiento y bienestar.

La sesión se considerará completa una vez que se hayan realizado todas las actividades establecidas. En caso de que queden actividades pendientes, el usuario deberá realizarlas para poder completar la sesión, lo que mantendrá la sesión activa. Después de completar la sesión, si el usuario decide no cerrar sesión, podrá seguir dentro del entorno virtual y realizar tantas sesiones como desee. Si el usuario opta por cerrar sesión, saldrá del entorno virtual y su progreso se guardará para la próxima vez que acceda.

Es importante destacar que no se han incluido las interacciones con los elementos de la sesión en el esquema para evitar una visualización demasiado compleja. Se ha tomado la misma decisión para los elementos interactivos fuera de las sesiones, como la posibilidad de interactuar con el ordenador y enviar mensajes al psicólogo.

3.2.3. Diseño de la base de datos

En la Figura 31 se muestra el modelo entidad-relación de la base de datos utilizada para la aplicación. En este diagrama, se presentan las relaciones entre las tablas y la lógica presente en la base de datos.

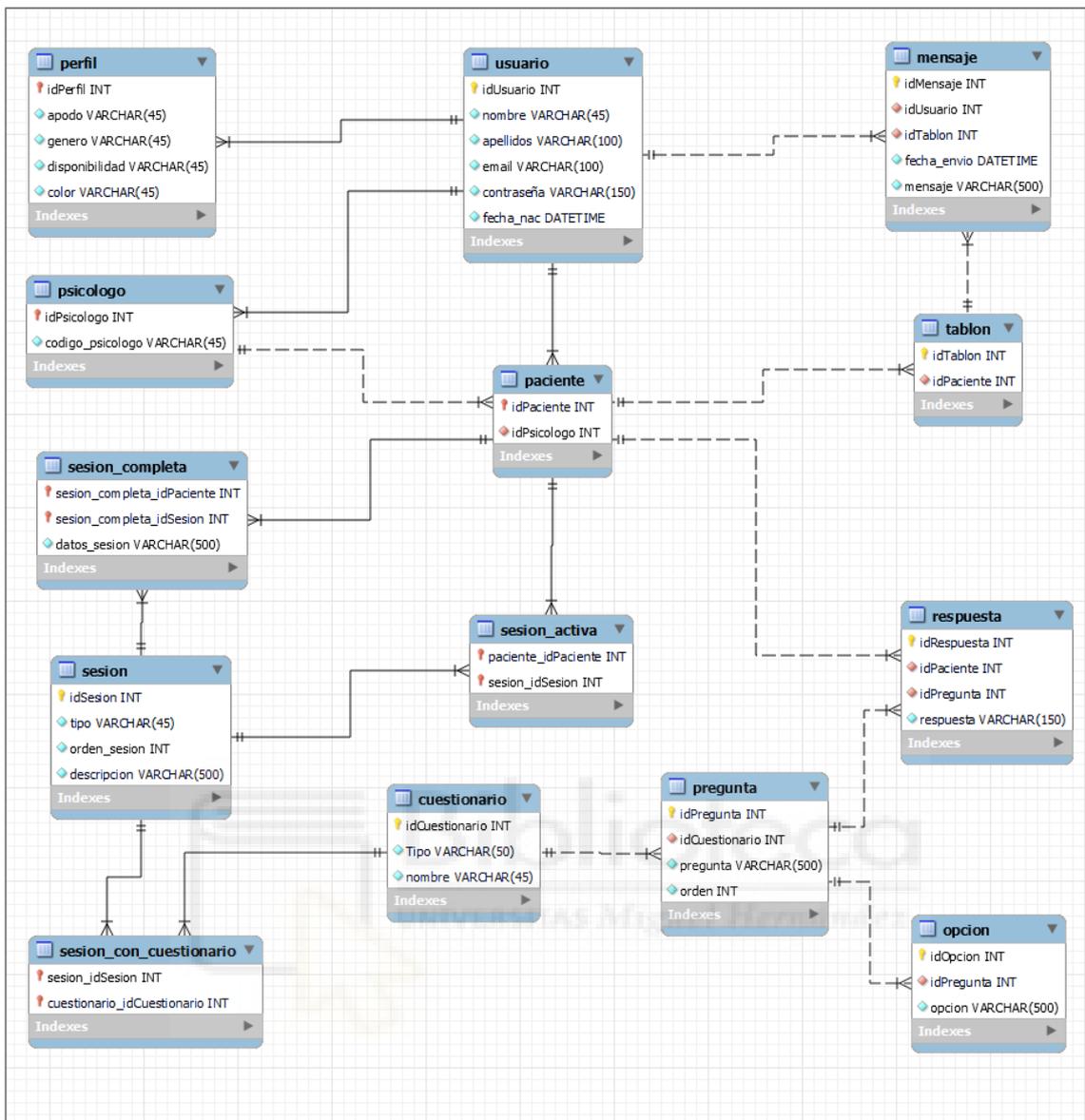


Figura 31: Esquema entidad-relación de la base de datos.

En la Figura 31, podemos observar tanto líneas continuas como líneas discontinuas. Las líneas continuas indican relaciones obligatorias, por ejemplo, un usuario necesariamente debe ser un paciente o un psicólogo. Por otro lado, las líneas discontinuas representan relaciones opcionales, como el caso de un psicólogo que puede tener o no pacientes asociados, ya que la existencia de un psicólogo no depende de tener pacientes.

A continuación, se presenta el esquema de tablas relacionales de la base de datos utilizada para la aplicación y se describen las tablas y sus relaciones:

- La tabla "Usuario" se considerada como la tabla central del sistema, ya que establece conexiones con las tablas "Psicólogo" y "Paciente". Sin embargo, es importante destacar que la tabla "Paciente" tiene una mayor

importancia debido a que constituye una gran parte de la aplicación, tal y como se busca en este proyecto, que todo gire en torno al paciente.

- Dentro de las tablas relacionadas con las sesiones, encontramos la tabla "Sesión Activa" y la tabla "Sesión Completa". La tabla "Sesión Activa" se encarga de registrar la sesión actual del usuario, mientras que la tabla "Sesión Completa" almacena los datos de las sesiones que han finalizado. Estas tablas permiten al psicólogo realizar un seguimiento del progreso del paciente.
- La tabla "Tablón" desempeña un papel importante en la comunicación entre el psicólogo y el paciente. Actúa como un chat que facilita la interacción y el intercambio de mensajes entre ambas partes.
- En relación con las sesiones, existe la tabla "Sesión" que almacena información sobre las sesiones en general, tanto aquellas relacionadas con la ansiedad como con la depresión. Además, se establece la tabla "Sesión con Cuestionario" para indicar si una sesión específica está asociada a un cuestionario en particular.
- Además de las tablas mencionadas, también existen tablas específicas para las preguntas, opciones y respuestas de los cuestionarios utilizados en la aplicación. Estas tablas se encargan de almacenar la información de las preguntas, las distintas opciones de respuesta disponibles y las respuestas registradas por los pacientes. El psicólogo tiene acceso a estas tablas a través de los datos de sus pacientes, lo que le permite evaluar el progreso y los resultados obtenidos en los cuestionarios.

En la sección 3.3.2 del proyecto se proporcionará una explicación detallada sobre cómo se pueden incorporar diversos modelos de cuestionarios en el sistema, en caso de que sea necesario en el futuro.

3.2.4. Estética y modelos de la aplicación

La estética y los modelos utilizados en esta aplicación es uno de los aspectos más importantes de este proyecto. Con el objetivo de proporcionar la mejor experiencia posible al usuario, se han creado:

- Modelos 3D específicos para brindar a los usuarios una experiencia personalizada y agradable.
- Se ha trabajado en los esquemas de color utilizados en la aplicación, así como en la presentación del contenido.
- Elección de una tipografía atractiva.

Como se puede observar en la Figura 32, la compañera del Grado de Comunicación Audiovisual nos proporcionó una selección de colores de diferentes tonos, además de algunas indicaciones específicas. Gracias a esto, en el caso de que fuera necesario podíamos escoger entre los múltiples colores

disponibles para crear un mejor diseño estético si comprometer la coherencia de este. La razón para ofrecer diferentes configuraciones de color es permitir al usuario crear un entorno personalizado y acogedor que se ajuste a sus gustos y preferencias, brindándole una sensación de comodidad y bienestar durante las sesiones.



Figura 32: Paletas de color de la aplicación.

Como se mencionó anteriormente, los modelos utilizados en el proyecto siguen una estética consistente. En la Figura 33, se pueden apreciar las cuatro versiones del modelo de la mascota. Este enfoque también se aplica a los árboles presentes en el entorno virtual. Con el fin de agregar variedad, se han creado dos versiones de árboles del mismo tipo y color. Esto evita que el bosque luzca monótono, como se puede observar en la Figura 34. Este enfoque garantiza que el usuario pueda personalizar y controlar su propio mundo, al permitirle seleccionar el color que más le agrade. Los modelos presentados en la Figura 33 y Figura 34 fueron creados por nosotros desde cero. Nosotros mismos nos encargamos de su diseño y desarrollo, garantizando así que se ajusten perfectamente a nuestras necesidades y requisitos para la aplicación.



Figura 33: Mascota en varios colores.



Figura 34: Ejemplo de los distintos arboles con sus colores.

Como se comentó previamente, cabe destacar que algunos de los modelos presentes en el mundo virtual fueron obtenidos de la página web Poly.pizza [59]. Estos modelos son gratuitos y siguen la estética *Low poly* [57]. Además, se realizaron ajustes en el código de algunos de estos modelos para adaptar su esquema de color a la estética del entorno virtual, como se muestra en la Figura 35.



Figura 35: Modelos de la página Poly.pizza [73] modificados.

3.3. Otros aspectos destacables de implementación

En este apartado se resumirán los aspectos relacionados con la implementación del código y que supusieron un desafío durante el proceso de desarrollo. Se debe tener en cuenta que en esta sección se describirán los aspectos más relevantes del código, con el fin de describir el uso de las tecnologías React, ThreeJS y NodeJS, con el objetivo de que sirva de documentación útil en un futuro para alumnos que deban de enfrentarse un proyecto similar.

3.3.1. React y el uso de los componentes y *hooks*

Comenzaremos con un ejemplo de `useState` por ser el método más utilizado al ser una buena forma de almacenar las variables de los componentes, tal y como podemos ver en la Figura 36. La principal ventaja que tiene es que se le puede asignar cualquier tipo de variable como enteros, *strings*, *arrays*, *true* o *false*... Uno de los principales problemas que tuve al utilizar el método fue que, al actualizar el estado, como el cambio no era inmediato muchas veces al realizar comprobaciones de variables daba resultados que no eran reales, es por eso por lo que a veces era necesario utilizar las variables más convencionales de JavaScript como `const` o `let`, tal y como se ve en la Figura 37.

```
const [PlayerApi, setPlayerApi] = useState(initialState: null);
const [nextSong, setNextSong] = useState(initialState: 0);
const [sesionActiva, setSesionActiva] = useState(initialState: '');
const [posicionPizarra] = useState(initialState: [-5, 1.8, -0.5]);
```

Figura 36: Ejemplo de uso de `useState`.

```

setShowInfo(value: (prevIndex) => {
  const newIndex = prevIndex + 1;

  if (newIndex === 2) {
    setTextoBoton(value: "¡Vamos allá!");
  } else if (newIndex > 2) {
    navigate(to: '/pilotoAutomatico');
  }
  return newIndex;
});

```

Figura 37: Ejemplo de la asignación de `useState` y uso de una variable convencional.

El uso del `useEffect` también era requerido por mi aplicación, debido a que se necesita un flujo de datos rápido para poder montarle al paciente el entorno virtual de la forma más rápida posible, tal y como se puede ver en la Figura 38.

```

useEffect(effect: () => {
  Axios.get(url: 'http://localhost:3001/loginUsr').then(onfulfilled: (response) => {
    if (response.data.loggedIn === true && response.data.user.Psicologo === false) {
      navigate(to: '/logeadoUser');
    } else if (response.data.loggedIn === true && response.data.user.Psicologo === true) {
      navigate(to: '/logeadoPsico');
    }
  });
});

```

Figura 38: Ejemplo del `useEffect`.

Como ya he explicado el `useEffect` se ejecuta cada vez que hay un cambio en el componente y al ser un mundo virtual en constante cambio se lanzaba muchas veces, esto puede ser positivo y negativo a la vez, ya que se pueden hacer comprobaciones constantes de si el usuario está autenticado como en la Figura 38, y en el caso de no estarlo redirigirlo al inicio, de esa forma no puede entrar al entorno sin que haya iniciado sesión, pero a la vez puede ser muy problemático ya que se pueden crear bucles infinitos, por lo tanto, hay que tener mucho cuidado cuando se usa.

Gracias a React hay una forma de controlar cuantas veces se realiza dicha actualización del `useEffect`, como podemos observar en la Figura 39, al final del `useEffect` se le pueden indicar unas dependencias. Si las dependencias están vacías significa que el `useEffect` solo se debe lanzar una única vez, y es cuando se carga el componente. En el caso de que las dependencias no estén vacías y tengan algún tipo de variable, esto indicará que el `useEffect` se debe volver a ejecutar cada vez que dicha variable cambia de contenido. Esto nos proporciona un control un gran control sobre la actualización de nuestro componente.

```

useEffect(effect: () => {
  if (showAudio) {
    const handleAudioEnded = () => {
      changeInfo();
      setTextoBoton(value: '¡Has');
      setShowAudio(value: false);
    }

    audioSesion.addListener('ended', handleAudioEnded);

    return () => {
      audioSesion.removeListener('ended', handleAudioEnded);
    };
  }
}, deps: [audioSesion, showAudio]);

useEffect(effect: () => {
  Axios.get(url: 'http://localhost:3000/api/usuarios/login', {
    headers: { 'Content-Type': 'application/json' },
    data: { email: email, password: password },
  })
  .then(response => {
    if (response.data.logged) {
      setUsuarioPsicologo(response.data.usuario);
      setEmailPsicologo(response.data.email);
      setIdPsicologo(response.data.id);
      setCargarPaciente(response.data.paciente);
    } else {
      navigate(to: '/');
    }
  })
  .catch(error => {
    console.log(error);
  });
}, deps: []);

```

Figura 39: Ejemplo del uso de dependencias en el método `useEffect`.

El último ejemplo sobre las funciones de React es el método `useContext`, este método sirvió para poder tener una variable global que todos los componentes pudieran compartir, en el caso de nuestro código es el color elegido por el usuario y su apodo. Tal y como comentaba en la sección 3.2.4, el color que elige del usuario determina toda la tonalidad presente en el entorno, por eso es necesario tenerlo presente en todo momento. Si nos fijamos en la Figura 40 podemos ver como se define el `useContext` para toda la aplicación y en la Figura 41 podemos ver un ejemplo de su uso.

```

const ProfileContext = createContext();
const ProfileProvider = ({ children }) => {
  // Obtener el color y el apodo de la base de datos
  const [color, setColor] = useState(initialState: 'Rosa');
  const [apodo, setApodo] = useState(initialState: '');

  // Proporcionamos el valor del contexto a los componentes envueltos
  return (
    <ProfileContext.Provider value={{ color, setColor, apodo, setApodo }}>
      {children}
    </ProfileContext.Provider>
  );
};

```

Figura 40: Ejemplo de creación del contexto para la aplicación.

```

const { color, setColor } = useContext(context: ProfileContext);
const { setApodo } = useContext(context: ProfileContext);

```

Figura 41: Ejemplo de uso de obtención del color por medio del `useContext`.

3.3.2. Gestión de formularios dinámicos

En este proyecto, se pueden identificar dos aspectos destacables relacionados con el uso de React. El primero es la capacidad de obtener cuestionarios dinámicos, mientras que el segundo es la generación de componentes pacientes. Como se mencionó previamente en la sección 3.2.3, tanto las preguntas como las opciones de los cuestionarios se almacenan en una base de datos. Actualmente, existen dos métodos para almacenar los cuestionarios en dicha base de datos. El primero corresponde a los cuestionarios estándar, que constan de una pregunta y cuatro respuestas en formato de texto. El segundo método se utiliza para los cuestionarios de escala tipo Likert. En la Figura 42 se muestra cómo se almacenan los registros en la base de datos.

idPregunta	idCuestionario	pregunta	orden
32	3	Caso 1: Pablo...	1
33	3	Caso 2: Laur...	2
34	3	Caso 3: Juan ...	3
35	4	¿Estoy confu...	1
36	4	¿Estoy llegan...	2
37	4	¿Este pensam...	3
38	4	¿Este pensam...	4
39	4	¿Me estoy ce...	5
40	4	¿Creo que n...	6
NULL	NULL	NULL	NULL

idOpcion	idPregunta	opcion
32	32	'Sobregeneralización', 'Pensamiento polarizado', 'Falacia de justicia'
33	33	'Sobregeneralización', 'Pensamiento catastrófico', 'Pensamiento polarizado'
34	34	'Pensamiento polarizado', 'Minimización', 'Falacia de justicia'
35	35	'No', 'Creo que no', 'No lo sé', 'Puede ser', 'Sí'
36	36	'No', 'Creo que no', 'No lo sé', 'Puede ser', 'Sí'
37	37	'No', 'Creo que no', 'No lo sé', 'Puede ser', 'Sí'
38	38	'No', 'Creo que no', 'No lo sé', 'Puede ser', 'Sí'
39	39	'No', 'Creo que no', 'No lo sé', 'Puede ser', 'Sí'
40	40	'No', 'Creo que no', 'No lo sé', 'Puede ser', 'Sí'
NULL	NULL	NULL

Figura 42: La primera imagen contiene las preguntas y la segunda las opciones.

Con el fin de evitar una excesiva extensión en esta sección, se procederá a explicar detalladamente la creación de un cuestionario de tipo normal. En la Figura 43 adjunta, se presenta una representación visual de cómo se reciben las opciones desde la base de datos y se realiza el mapeo de las preguntas y las respectivas opciones, para luego mostrarlas al usuario. Ahora bien, si nos enfocamos en la segunda imagen de la Figura 43, podemos observar que se utiliza la función `map` con el propósito de crear tantas preguntas como se hayan recibido desde la base de datos.

A continuación, se emplea el componente `FormControl` de la biblioteca `mui material`, el cual nos permite crear un campo de formulario con una estructura adecuada. Dentro del componente `FormControl`, se utiliza el componente `FormLabel` para presentar la pregunta correspondiente. Siguiendo con la

ejecución del código, se puede notar que para cada pregunta se realiza un mapeo de las opciones disponibles. Dichas opciones se presentan mediante el componente `RadioGroup`, permitiendo así al usuario seleccionar una respuesta para cada pregunta. Para mostrar el texto asociado a cada opción, se emplea el componente `FormControlLabel`. De esta manera, se logra generar dinámicamente un cuestionario proveniente de la base de datos, lo que facilita la integración de cualquier cuestionario con una estructura similar sin necesidad de realizar modificaciones sustanciales en el código existente.

```
const question = response.data.Preguntas.map((question, index) => {
  const options = question.opcion.replace(/'/g, '').split(', ');
  idQuestion.push(...items: question.idPregunta);
  return {
    question: `${question.pregunta}`,
    options: options
  };
});
setQuestions(value: question);
setValues(value: Array(...items: question.length).fill(value: ''));
setIdQuestions(value: idQuestion);

{questions.map(callbackfn: (question, index) => (
  <QuestionWrapper key={index}>
    <FormControl component="fieldset">
      <FormLabel component="legend" sx={{ fontSize: '1.5rem' }}>{question.question}</FormLabel>
      <RadioGroup
        name={`question${index + 1}`}
        value={values[index]}
        onChange={(event) => handleChange(event, index)}
      >
        {question.options.map((option, index) => (
          <FormControlLabel key={index} value={option} control={<Radio />} label={<Typography variant='h5'>{option}
        ))}
      </RadioGroup>
    </FormControl>
  </QuestionWrapper>
))}
```

Figura 43: La primera imagen muestra cómo se mapean los datos recibidos por la base de datos y en la segunda imagen como se genera el cuestionario de forma dinámica a raíz de los datos obtenidos.

El segundo aspecto destacable es la generación de múltiples componentes dinámicos sobre una misma interfaz, en la Figura 44 podemos ver como se les muestran sus pacientes a un psicólogo que accede a la web. Cada uno de esos pacientes mostrados es un componente dinámico que se genera con los datos que hay almacenados en nuestra base de datos.



Figura 44: Ejemplo de cómo se muestran los pacientes en la UI del psicólogo.

Cada uno de los pacientes es un componente paciente, que se generan como se puede ver en la Figura 45. Se obtienen los datos necesarios y se pasan como un *prop* al componente paciente para que los monte. A continuación, el componente paciente utiliza los datos proporcionados para crear la interfaz de usuario que refleja la información relevante de cada paciente desde la perspectiva del psicólogo (ver vista en la Figura 44).

```
Axios.post(url: 'http://localhost:3001/getPatient', data: {
  idPsicologo: idPsicologo
}).then(onfulfilled: (response) => {
  if (response.data.get === 'OK') {
    const pacientes = response.data.pacientes;
    const pacientesComponente = pacientes.map((paciente) => {
      const fecha = new Date(value: paciente.fecha_nac);
      return (
        <Paciente
          key={paciente.idUsuario}
          idUsuario={paciente.idUsuario}
          Nombre={paciente.nombre}
          Apellidos={paciente.apellidos}
          Email={paciente.email}
          Fecha_nac={fecha.getDate() + "/" + (fecha.getMonth() + 1) + "/" + fecha.getFullYear()}
          idPsicologo={idPsicologo} />
      );
    });
    setComponente(value: [...componente, pacientesComponente]);
  }
});
```

Figura 45: Creación de los componentes pacientes.

3.3.3. El uso de Three.js con React para el entorno 3D

Antes de comenzar con esta sección, es importante recordar que se ha decidido utilizar la librería React Three Fiber [65] para este proyecto. Esta librería se basa en Three.js pero ha sido adaptada para aprovechar los componentes y estados de React, lo que permite renderizar objetos 3D en una escena de manera eficiente. Inicialmente, se consideró utilizar la librería Three.js nativa, que está escrita en JavaScript, lo que no presentaría problemas al usarla junto con React. Sin embargo, justo antes de iniciar el desarrollo del mundo 3D, nos encontramos con React Three Fiber. A pesar de ser una librería relativamente nueva, decidimos elegirla para este proyecto ya que se ajustaba perfectamente a las necesidades de desarrollo y estaba altamente enfocada en React.

Como mencioné anteriormente, dado que React Three Fiber se basa en Three.js, el estudio previo realizado resultó beneficioso, ya que la forma de crear elementos seguía siendo similar. Sin embargo, al utilizar React Three Fiber, la creación de elementos se simplificó aún más al trabajar en un enfoque basado en componentes. Esto permitió una implementación más sencilla y eficiente en comparación con el enfoque tradicional de Three.js. Al observar la Figura 46, se pueden apreciar claramente las diferencias entre la creación de una escena utilizando Three.js nativo y React Three Fiber. Como se puede ver, la versión de React Three Fiber es considerablemente más sencilla e intuitiva en comparación con la versión original.

```
const scene = new THREE.Scene()
const camera = new THREE.PerspectiveCamera(75, width / height, 0.1, 1000)

const renderer = new THREE.WebGLRenderer()
renderer.setSize(width, height)
document.querySelector('#canvas-container').appendChild(renderer.domElement)

const mesh = new THREE.Mesh()
mesh.geometry = new THREE.BoxGeometry()
mesh.material = new THREE.MeshStandardMaterial()

scene.add(mesh)

function animate() {
  requestAnimationFrame(animate)
  renderer.render(scene, camera)
}

animate()
```

```
<Canvas>
  <mesh>
    <boxGeometry />
    <meshStandardMaterial />
  </mesh>
```

Figura 46: En la primera imagen podemos ver un ejemplo de Three.js nativo y en la segunda figura un ejemplo de React Three Fiber [65].

Una vez explicado de manera concisa el funcionamiento de React Three Fiber, al observar la implementación del código de la Figura 47, se aprecia una estructura similar. Dado que Three.js es una librería extensa, se ha dividido en varias bibliotecas, siendo React Three Fiber la principal, como se mencionó anteriormente, y React Drei, que se encarga del control de cámaras, elementos estéticos como video, audio y textos en 3D, así como una herramienta para importar modelos. En la Figura 47, también se pueden apreciar algunos componentes estéticos, como *Stars*, *Sky*, *Environment*, y la cámara en primera persona de *PointerLockControls*. Como su nombre indica, estos componentes suelen ser utilizados dentro del entorno 3D y no para construirlo, como es el caso de React Three Fiber.

```
<Canvas>
  <Suspense fallback=<Loader />>
    <Sky sunPosition={[100, 80, 10]} />
    <Stars radius={100} depth={50} count={100} factor={7} saturation={0} fade speed={0.1} />
    <Environment preset="sunset" background />
    <PointerLockControls onLock={handleLock} onUnlock={handleUnlock} />
```

Figura 47: Ejemplo de uso de React Three Fiber con elementos de React Drei.

Los componentes de React Drei no solo son utilizados al crear el mundo, sino que en las diversas sesiones se utilizan estos elementos también, en la Figura

48 se puede apreciar cómo se hace uso del componente Text para mostrar información en el mundo 3D.

```
<Text position={[-4.95, 2.3, -0.5]} fontSize={0.2} color={'#000000'} maxWidth={5} rotation={[0, Math.PI / 2, 0]}
  textAlign="justify"
>
```

Figura 48: Ejemplo del uso del componente text.

3.3.4. Movimiento del usuario en el entorno virtual

Otro de los elementos de suma importancia para nuestro entorno virtual es brindarle al usuario la libertad de movimiento dentro del escenario. Para lograr esto, es necesario realizar varias implementaciones. En primer lugar, nos enfocaremos en capturar los eventos del teclado del usuario y registrarlos. De esta manera, cada vez que se presione una tecla, como "w", "a", "s" o "d", el personaje se moverá de acuerdo con la tecla pulsada.

La segunda parte de esta implementación consiste en transmitir estos eventos del teclado al entorno 3D para permitir que el usuario se desplace. Para lograr esto, requerimos de un motor de físicas, y es aquí donde entra en juego la librería React Cannon. Esta librería nos proporciona *hooks* basados en físicas para React, lo que nos permite crear objetos rígidos e incluso objetos sujetos a la gravedad.

Una vez que ambas implementaciones se hayan realizado, obtendremos un resultado similar al mostrado en la Figura 49. Al analizar dicha figura, observamos que estamos capturando los eventos del teclado mediante una función previamente creada llamada `useKeyboard`. A continuación, creamos una esfera con físicas que servirá como el elemento con el cual el usuario podrá desplazarse por el plano. Por último, creamos la cámara. Luego, debemos suscribir los eventos del teclado a la esfera mediante el uso de `useEffect`. De esta manera, la esfera se moverá en respuesta a los eventos del teclado.

Para lograr que la cámara siga el movimiento de la esfera, es necesario crear vectores tridimensionales de Three.js. Estos vectores nos permitirán obtener el movimiento del usuario y, de esta manera, podremos mantener la cámara alineada con la dirección del usuario en todo momento. Este proceso se puede apreciar en la Figura 50.

Por último, para proporcionar al usuario un movimiento coherente y permitir que la esfera se desplace en la dirección en la que está mirando, es necesario normalizar constantemente el vector. Esto garantiza que el usuario pueda moverse en todas las direcciones con una cámara consistente.

```

const {
  moveForward, moveBackward, moveLeft, moveRight,
} = useKeyboard();

const { camera } = useThree();

const [SphereRef, SphereApi] = useSphere(fn: () => ({
  mass: 1,
  type: "Dynamic",
  position: [0, 0.4, 2],
  args: [0.48]
})));

const posicion = useRef(initialValue: [0, 0, 0]);
useEffect(effect: () => {
  SphereApi.position.subscribe(callback: (pos) => {
    posicion.current = pos;
  });
  handlePlayerApi(SphereApi);
}, deps: [SphereApi.position, handlePlayerApi, SphereApi]);

const velocidad = useRef(initialValue: [0, 0, 0]);
useEffect(effect: () => {
  SphereApi.velocity.subscribe(callback: (vel) => {
    velocidad.current = vel;
  });
}, deps: [SphereApi.velocity]);

```

Figura 49: Creación de las físicas de movimiento del usuario y suscripción de los eventos del teclado a la esfera.

```

useFrame(callback: () => {
  if (controlsEnabled === true) {
    camera.position.copy(
      v: new THREE.Vector3(
        x: posicion.current[0],
        y: posicion.current[1] + 0.9,
        z: posicion.current[2]
      )
    );

    const direction = new THREE.Vector3();
    const frontVector = new THREE.Vector3(
      x: 0,
      y: 0,
      z: (moveBackward ? 1 : 0) - (moveForward ? 1 : 0) /
    );
    const sideVector = new THREE.Vector3(
      x: (moveLeft ? 1 : 0) - (moveRight ? 1 : 0),
      y: 0,
      z: 0
    );
  }
});

```

Figura 50: Copia de la cámara en cada momento que el usuario se mueve por el plano.

3.3.5. Integración e interacción con los modelos 3D

Aquí se abordará cómo se introdujeron los modelos 3D en el entorno virtual. Para poder cargarlos haremos uso de la librería de Three.js, React Three Drei. En concreto, utilizaremos el método `useGLTF`. Por lo general, la importación de algunos modelos puede ser muy compleja, ya que depende de la cantidad de polígonos que tenga y puede requerir mapear más o menos mallas. Sin embargo, React Three Fiber facilita esto, ya que su documentación en el apartado "Loading models" no proporciona esta página [66]. En esta página, debemos introducir nuestros modelos 3D. Como podemos ver en la Figura 51, esta página mapea todo el modelo 3D y proporciona el código del modelo para que podamos introducirlo de forma sencilla en nuestro código. De esta manera, se soluciona el problema de crear la malla para cada polígono. Tal y como se mencionó en la sección 3.2.4, gracias a esta página pude identificar cuál era la malla que le daba color al tocadiscos o al portátil y modificar el color según la tonalidad elegida por el usuario, proporcionando así una mayor sensación de personalización.



Figura 51: Ejemplo de uso del cargador de modelos de React Three Fiber (React Drei) [66].

A continuación, en la Figura 52 se encuentra el código del modelo del tocadiscos, se ha escogido este en particular ya que se puede apreciar que se le han aplicado físicas, el color elegido según la preferencia del usuario y el cambio de música de ambiente que hay en el entorno virtual. Se entrará más en detalle de este último punto en la sección 3.4. Si nos fijamos en la Figura 52 podemos ver varios detalles importantes, el primero es la propia importación del modelo que es exactamente igual como decía en la Figura 51, pero en este caso si nos fijamos en la primera malla podemos ver que el contenido ha sido cambiado por `colorMaterial`, el contenido de esta variable lo recibe como `prop` y es el color seleccionado por el usuario. El segundo detalle importante son las físicas del

modelo, como podemos ver, se ha llamado a la librería React Cannon para importar el método `useBox`, ahora en sus propiedades le pasamos la posición del tocadiscos y en los `args`, cuanto se tiene que expandir la caja alrededor del modelo del tocadiscos, una vez estén definidos si nos fijamos en la etiqueta `group` podemos ver cómo le pasamos la referencia de la caja con físicas. De esta forma la caja con físicas estará fijada con el conjunto de mallas del modelo.

```
import React, { useRef } from "react";
import { useGLTF } from "@react-three/drei";
import { useBox } from "@react-three/cannon";
import * as THREE from "three";

export function Tocadiscos({ color, controlMusica, PlayerApi, controlNextSong }) {
  const { nodes, materials } = useGLTF(path: "./Modelos/Tocadiscos.glb");
  const colorMaterial = new THREE.MeshBasicMaterial(parameters: { color: color });
  const posicionTocadiscos = [3, 0.81, -3.95];

  const playerPosition = useRef(initialValue: [0, 0, 0]);

  const [tocadiscos] = useBox(fn: () => ({
    mass: 1,
    type: "Static",
    position: posicionTocadiscos,
    rotation: [0, Math.PI / 2, 0],
    args: [2, 2, 3.43],
  }));

  return (
    <group dispose={null} ref={tocadiscos} scale={0.008} onClick={handleChangeSong}>
      <mesh
        castShadow
        receiveShadow
        geometry={nodes.ChamferCyl009_1_4.geometry}
        material={colorMaterial}
      />
    </group>
  );
}
```

Figura 52: Importación del modelo, creación de físicas y cambio de color.

Si ahora miramos la Figura 53 podemos ver la función `handleChangeSong`, esta función se ejecuta cuando se hace clic al tocadiscos, pero tiene algo especial y es que solo se lanzará si el usuario está lo suficientemente cerca, esto es necesario hacerlo, ya que de lo contrario desde cualquier parte del entorno virtual si se le hace *clic* se llamara a la función, independientemente de si está a una distancia demasiado grande o si hay otros modelos por medio. Esto al principio fue un gran problema ya que el usuario podía ejecutar esta función sin querer dándole *clic* al tocadiscos a través de algún modelo del entorno, lo que podría provocar confusión. Para solucionar eso lo que se hizo fue tener en cuenta la posición del personaje, como ya se comenté anteriormente al copiar la posición de la cámara constantemente en la posición del usuario, se decidió pasar ese parámetro a todos los modelos que requiriesen alguna interacción, de esa forma se comparaba la posición actual del modelo con la posición del usuario y si la distancia estaba dentro del umbral de acción, la interacción se llevaba a

cabos, en el caso de que la distancia fuera mayor no funcionaría, de esta forma conseguimos solucionar el problema de interacciones indeseadas.

```
const playerPosition = useRef(initialValue: [0, 0, 0]);

//Se ejecuta al hacer click al tocadiscos
const handleChangeSong = () => {

  const umbral = 4;

  //Obtenemos la posición del jugador
  PlayerApi.position.subscribe((pos) => {
    playerPosition.current = pos;
  });

  //Calculamos la distancia entre el jugador y el tocadiscos
  const distancia = Math.sqrt(
    Math.pow(x: playerPosition.current[0] - posicionTocadiscos[0], y: 2) +
    Math.pow(x: playerPosition.current[1] - posicionTocadiscos[1], y: 2) +
    Math.pow(x: playerPosition.current[2] - posicionTocadiscos[2], y: 2)
  );

  //Si la distancia es menor que el umbral, se ejecuta la función de pasar a la siguiente canción
  if (distancia < umbral) {
    if (controlMusica=== true) {
      controlNextSong(true)
    }
  }
};
```

Figura 53: Función de cambio de canción si el usuario está dentro del umbral de acción.

3.3.6. Implementación del Backend con Node.js, Express y MySQL

En este apartado, se detallará cómo funciona la conexión del *backend* del proyecto utilizando Node.js y el *framework* Express. Se explicará cómo se utiliza Express para crear y manejar el servidor web y las rutas de la aplicación, así como para conectarse a la base de datos MySQL y manejar las solicitudes HTTP que se reciben desde el *frontend*. También se hablará sobre cómo se utiliza Axios en el *frontend* para hacer solicitudes al *backend* y cómo se configura CORS para permitir el intercambio de recursos entre el *frontend* y el *backend*.

Una de las partes más importantes de nuestro proyecto es el uso de sesiones que nos proporciona Express. Gracias a esto, se puede mantener la sesión del usuario activa, evitando momentos incómodos de cierres de sesión al cerrar el navegador. Además, mejora la seguridad de la aplicación al requerir que se haya iniciado sesión antes de acceder a cualquier parte del entorno. Otro aspecto importante para tener en cuenta es el uso de Cors para gestionar las políticas de seguridad de intercambio de recursos entre el *frontend* y el *backend*.

Si observamos la Figura 54, podemos identificar varias cosas interesantes. En primer lugar, se encuentra la definición de las librerías que se utilizarán. Como ya se mencionó anteriormente, se encuentran Express y Cors. Además, se utiliza

MySQL para establecer la conexión con la base de datos. También se incluye `cookieParser`, el cual se utiliza en conjunto con Express para analizar las cookies que se envían junto con las llamadas HTTP. El `bodyParser` se utiliza para leer el cuerpo de las llamadas HTTP en diversos formatos, como `urlencoded`. Por otro lado, se encuentra la librería `bcrypt`, que se utiliza para cifrar datos y proporcionar seguridad adicional. Por último, se menciona el módulo `Express-session`, el cual se utiliza para crear y gestionar la sesión del usuario en la aplicación.

```
const express = require('express');
const mysql = require('mysql');
const cors = require('cors');
const bcrypt = require('bcrypt');
const bodyParser = require('body-parser');
const cookieParser = require('cookie-parser');
const session = require('express-session');
```

Figura 54: Librerías usadas para el backend del proyecto.

Si continuamos leyendo el código del *backend*, tal como se muestra en la Figura 55, podemos observar el uso de Express en todas las interacciones. Se crea una variable llamada `app` que copia el contenido del método Express, lo que significa que cualquier ejecución de código pasará a través de Express.

A continuación, se utiliza Cors. En este punto, se debe indicar qué origen tiene permitido realizar solicitudes al *backend*. En este caso, se han habilitado los métodos GET y POST, y también se permite el uso de *cookies*. Después, se activa el método de `cookieParser` y `bodyParser`. Estos son utilizados para analizar las cookies que se envían junto con las llamadas HTTP y para leer el cuerpo de las solicitudes en formato `urlencoded`, respectivamente. Más adelante en el código, se utiliza el módulo de `express-session` para crear la sesión del usuario. Aquí se especifica que la *cookie* tenga una edad máxima de un día, lo que implica que la sesión del usuario se mantendrá activa durante ese período de tiempo.

Por último, se establece la conexión con la base de datos MySQL. Con todas estas configuraciones, podemos garantizar una conexión estable y segura tanto para el *backend* como para el *frontend* de la aplicación.

```

const app = express();

app.use(...handlers: express.json());

app.use(...handlers: cors(options: {
  origin: ["http://localhost:3000"],
  methods: ["GET", "POST"],
  credentials: true
}));

app.use(...handlers: cookieParser());
app.use(...handlers: bodyParser.urlencoded(options: { extended: true }));

app.use(...handlers: session(options: {
  key: "activerUsr",
  secret: "secreto",
  resave: false,
  saveUninitialized: false,
  cookie: {
    maxAge: 24 * 60 * 60 * 1000,
  },
}));

const db = mysql.createConnection(connectionUri: {
  user: "root",
  host: "localhost",
  password: "admin1234",
  database: "database"
});

```

Figura 55: Implementación de las herramientas necesarias para el correcto funcionamiento del backend.

3.3.7. Uso de Axios para realizar peticiones desde el *frontend*

Se explicará cómo se realizaría una petición desde el *frontend* utilizando Axios y cómo se relaciona con los puntos mencionados anteriormente. Como se muestra en la Figura 56, esta es la forma de utilizar Axios para hacer una llamada al *backend* desde el *frontend*. Una vez que se realiza esta llamada, el *backend* la recibe. Aquí es donde Cors verifica si el origen de la petición es aceptado. Si Cors aprueba la petición, Express se encargará de manejar los datos y realizar la consulta en la base de datos (ver Figura 57). Una vez que la BBDD haya procesado la consulta, devolverá los datos solicitados a Express. Por ejemplo, en el caso de la Figura 57 la consulta es del tipo `SELECT`. Estos datos se enviarán al *frontend* para que Axios los reciba y pueda trabajar con ellos.

```

Axios.post(url: 'http://localhost:3001/getDataPsico', data: {
  idUsuario: response.data.user.Id
}).then(onfulfilled: (responsePsico) => {
  setIdPsicologo(value: responsePsico.data.idPsicologo);
  setNombrePsicologo(value: responsePsico.data.Nombre);
  setApellidosPsicologo(value: responsePsico.data.Apellidos);
  setEmailPsicologo(value: responsePsico.data.Email);
});

```

Figura 56: Llamada Axios desde el frontend del proyecto.

```

app.post(path: '/getDataPsico', ...handlers: (req, res) => {

  const idUsuario = req.body.idUsuario;

  db.query(options: "SELECT idPsicologo FROM paciente WHERE idPaciente = ?", values: idUsuario,
  callback: (err, resultIdPsico) => {
    if (err) {
      console.log(message: err);
    } else {

      db.query(options: "SELECT nombre, apellidos, email FROM usuario WHERE idUsuario = ?", values: resultIdPsico[0].idPsicologo,
      callback: (error, resultDatosPsico) => {
        if (error) {
          console.log(message: error);
        } else {
          let datosPsicologo = {
            idPsicologo: resultIdPsico[0].idPsicologo,
            Nombre: resultDatosPsico[0].nombre,
            Apellidos: resultDatosPsico[0].apellidos,
            Email: resultDatosPsico[0].email
          }
          res.send(body: datosPsicologo);
        }
      }
    }
  }
}

```

Figura 57: Llamada recibida en el backend, consulta a la base de datos y respuesta con los datos necesarios.

3.3.8. Gestión de la sesión de usuario

La sesión del usuario se crea una vez que el usuario haya iniciado sesión. Si observamos la Figura 58, podemos ver que una vez que el usuario ha sido autenticado correctamente y nuestra base de datos haya confirmado que el usuario existe, obtendremos sus datos y crearemos la variable `datosUsr`.

Tras ello, asignaremos esta variable a la variable de sesión `user`, que posteriormente se puede utilizar para obtener los datos del usuario en el *frontend* si es necesario. De esta manera, se establece y mantiene la sesión del usuario durante su interacción con el entorno virtual.

```

if (resultPaciente[0].Paciente === 1) {
  let datosUsr = {
    Nombre: resultUsr[0].nombre,
    Apellidos: resultUsr[0].apellidos,
    Email: resultUsr[0].email,
    Id: resultUsr[0].idUsuario,
    Psicologo: false
  };
  req.session.user = datosUsr;
  res.send(body: datosUsr);
}

```

Figura 58: Almacenamiento de datos de usuario en su sesión.

En la primera imagen de la Figura 59, se muestra que cuando se realiza una llamada a `loginUsr`, se verifica si el usuario ha iniciado sesión o no. En caso de que haya iniciado sesión, se envían los datos al *frontend*. Si observamos la segunda imagen, podemos ver que mediante el uso del método `useEffect`, se realiza constantemente una verificación para determinar si el usuario ha iniciado sesión. En caso de que no haya iniciado sesión, se redirigirá al usuario a la página de inicio de sesión, como se mencionó anteriormente en este apartado. Esta verificación garantiza que solo los usuarios autorizados tengan acceso a las funciones y contenido deseado en la aplicación.

```
app.get(path: '/loginUsr', ..handlers: (req, res) => {
  if (req.session.user) {
    res.send(body: { loggedIn: true, user: req.session.user });
  } else {
    res.send(body: { loggedIn: false });
  }
})

useEffect(effect: () => {
  Axios.get(url: 'http://localhost:3001/loginUsr').then(onfulfilled: (response) => {
    if (response.data.loggedIn !== true) {
      navigate(to: '/');
    } else if (response.data.loggedIn === true && response.data.user.Psicologo === true) {
      navigate(to: '/logeadoPsico');
    }
    setIdUsuario(value: response.data.user.Id);
  });
});
```

Figura 59: Ejemplo del uso de sesión del usuario.

3.4. Ejemplo de uso de la aplicación web

En esta sección, se describirá un ejemplo de uso de la aplicación web desarrollada para nuestro proyecto. Aunque esta versión podría considerarse como un prototipo de lo que podría llegar a ser, es una muy buena base para tener una aplicación web completamente funcional que se pueda lanzar al público.

3.4.1. Login y registro de usuarios

En la Figura 60 se muestra la interfaz que el usuario encontraría al iniciar sesión o registrarse. En la imagen de la derecha, se solicitan datos como el nombre, apellidos, email, contraseña y fecha de nacimiento para el registro. Además, el usuario deberá aceptar el consentimiento informado para acceder. En caso de que el usuario complete incorrectamente un campo o no acepte el consentimiento informado, se le mostrará un mensaje de error indicándole cómo solucionar el problema (ver Figura 61).

Iniciar Sesión

EMAIL

PASSWORD

Soy psicólogo:

HE OLVIDADO MI CONTRASEÑA

REGISTRARME

INICIAR SESIÓN

Registrarme

Nombre:
NOMBRE

Apellidos:
APELLIDOS

Email:
EMAIL

Contraseña:
PASSWORD

Fecha de nacimiento:
dd/mm/aaaa

Consentimiento informado:

Soy psicólogo:

REGISTRARME

Figura 60: Ejemplo de login y registro.



Figura 61: Ejemplo de mensaje de error al registrarse.

Para aceptar el consentimiento informado, el usuario deberá marcar la casilla correspondiente y se le mostrará un desplegable con el texto completo del consentimiento. Después de leerlo y aceptarlo, podrá registrarse como usuario en la web. Una vez registrado, el usuario deberá esperar a que su psicólogo lo agregue como paciente para poder acceder al entorno virtual. Si un usuario se quiere registrar como usuario psicólogo deberá pulsar la casilla “Soy psicólogo” e introducir un numero de colegiado valido, en el caso de que no sea válido no podrá registrarse en la aplicación web.

3.4.2. Página principal del psicólogo.

Cuando un psicólogo inicie sesión, podrá acceder a una interfaz similar a la mostrada en la Figura 62. En la cabecera de la pantalla, se mostrará el nombre y correo electrónico del psicólogo. En el centro de la pantalla, se presentará una lista de los pacientes asignados al psicólogo, donde se proporcionará una breve descripción de cada paciente, incluyendo su nombre y apellidos, correo electrónico y fecha de nacimiento.

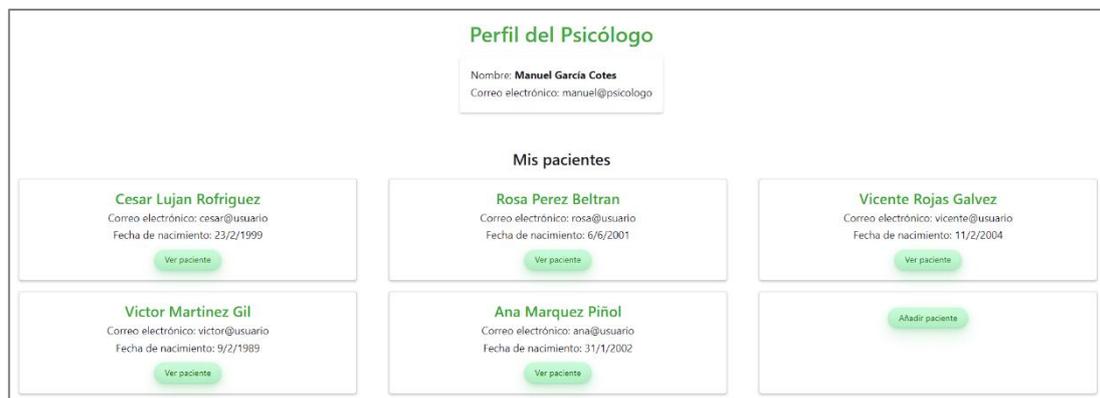


Figura 62: Ejemplo de la interfaz gráfica del psicólogo.

Si el psicólogo decide pulsar el botón de "Añadir paciente", se abrirá un cuadro modal donde podrá introducir el correo electrónico del usuario registrado, como se muestra en la Figura 63. En caso de que el correo introducido no sea válido o el usuario no esté registrado, se mostrará un mensaje de error indicando que el correo no es válido datos. También es importante tener en cuenta que, si el usuario ya está asociado a otro psicólogo, se mostrará un mensaje de error informando que el paciente ya tiene un psicólogo asignado.

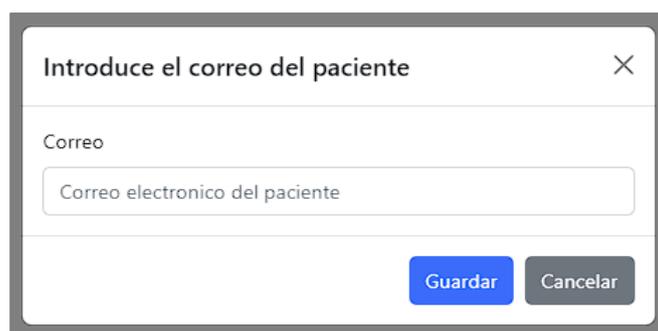


Figura 63: Ejemplo de asignación de un usuario como paciente.

El psicólogo podrá acceder al perfil individual de cada paciente pulsando el botón "Ver paciente", tal y como se puede ver en la Figura 62, donde encontrará varias funcionalidades disponibles. Si observamos la Figura 64, podremos identificar varios elementos de importancia en el perfil del paciente. En primer lugar, el psicólogo tendrá acceso a un chat que le permitirá comunicarse con el paciente en cualquier momento que lo desee. Además, en el perfil del paciente, en la parte

derecha de la pantalla, el psicólogo podrá visualizar el progreso que el paciente ha realizado en las sesiones. Se puede observar que las sesiones se presentan en forma de desplegable, donde cada sesión cuenta con información asociada. En el caso de que una sesión incluya un cuestionario relacionado, también se proporcionará información adicional sobre el mismo en el desplegable correspondiente a esa sesión en particular. Es importante destacar que, si una sesión aún no se ha llevado a cabo, no aparecerá en el desplegable.

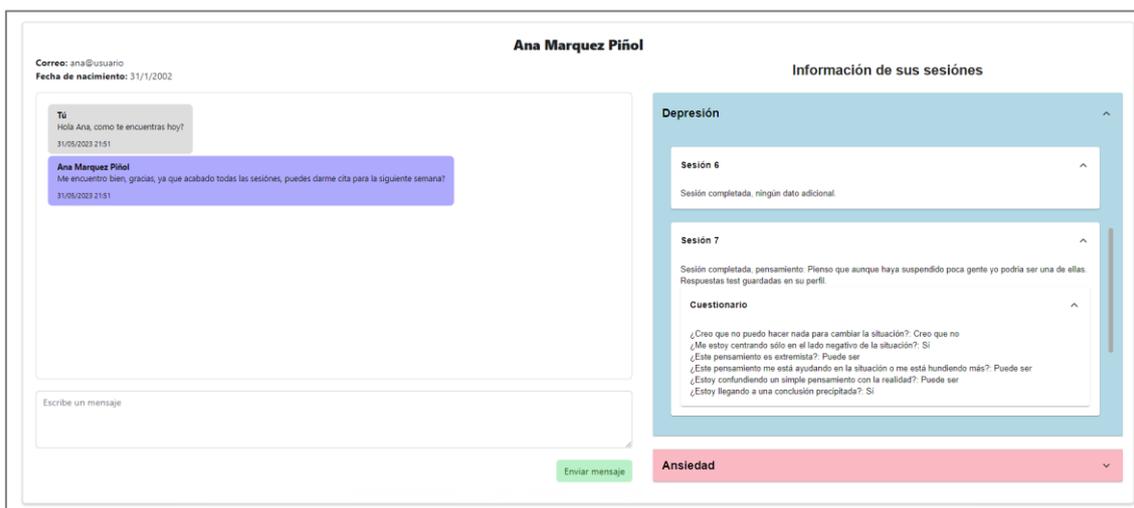


Figura 64: Ejemplo del perfil de un paciente con sesiones realizadas.

3.4.3. Primer inicio de sesión del paciente.

Una vez que el psicólogo haya registrado al usuario como paciente, este último podrá acceder al entorno virtual. Sin embargo, antes de comenzar, se le pedirá al paciente que complete un cuestionario de personalización. En el cuestionario, se le solicitará al paciente que elija un apodo con el que se sienta como, cuál es su sexualidad, que disponibilidad tiene y que tonos de color prefiere (Ver Figura 65).

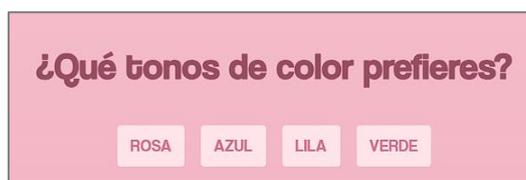


Figura 65: Ejemplo selección de color.

Una vez que el usuario haya completado el cuestionario de personalización, podrá ingresar al entorno virtual. En la Figura 66 se muestra una representación visual de cómo se vería el entorno virtual después de que el usuario haya

seleccionado tonalidades de color azul. Además, tras entrar sonará una música relajante de fondo.



Figura 66: Ejemplo del entorno virtual con tonalidad azul seleccionada.

3.4.4. Elementos interactivos del entorno.

En este apartado explicare cuales son las posibles interacciones previas que se pueden realizar fuera de las sesiones.

- **Cambiar la música de ambiente:** si el usuario desea cambiar la música de ambiente puede acercarse al tocadiscos que se puede ver en la Figura 67 y dándole *click* podrá pasar a la siguiente canción. Una vez entre a realizar una sesión la música se detendrá y la acción se bloqueará hasta que termine la sesión activa una vez finalizada se volverá a reactivar.



Figura 67: Ejemplo del tocadiscos y del portátil en el entorno virtual.

- **Comunicación con el psicólogo:** se ha implementado la opción de comunicarse con su psicólogo sin necesidad de abandonar esta plataforma. Para ello deberá acercarse al portátil que se puede ver en la Figura 67 y darle doble *click*. Al hacerlo, se abrirá la interfaz del portátil, como se muestra en la Figura 68. En la parte superior de la interfaz se

visualiza el nombre completo del paciente, seguido del nombre del psicólogo asociado y su dirección de correo electrónico. Además, se ha integrado un chat de mensajes interno que permite una comunicación fluida y segura entre el paciente y su psicólogo. Este chat se puede utilizar tanto dentro como fuera de las sesiones programadas, brindando al paciente la libertad de establecer contacto en cualquier momento.

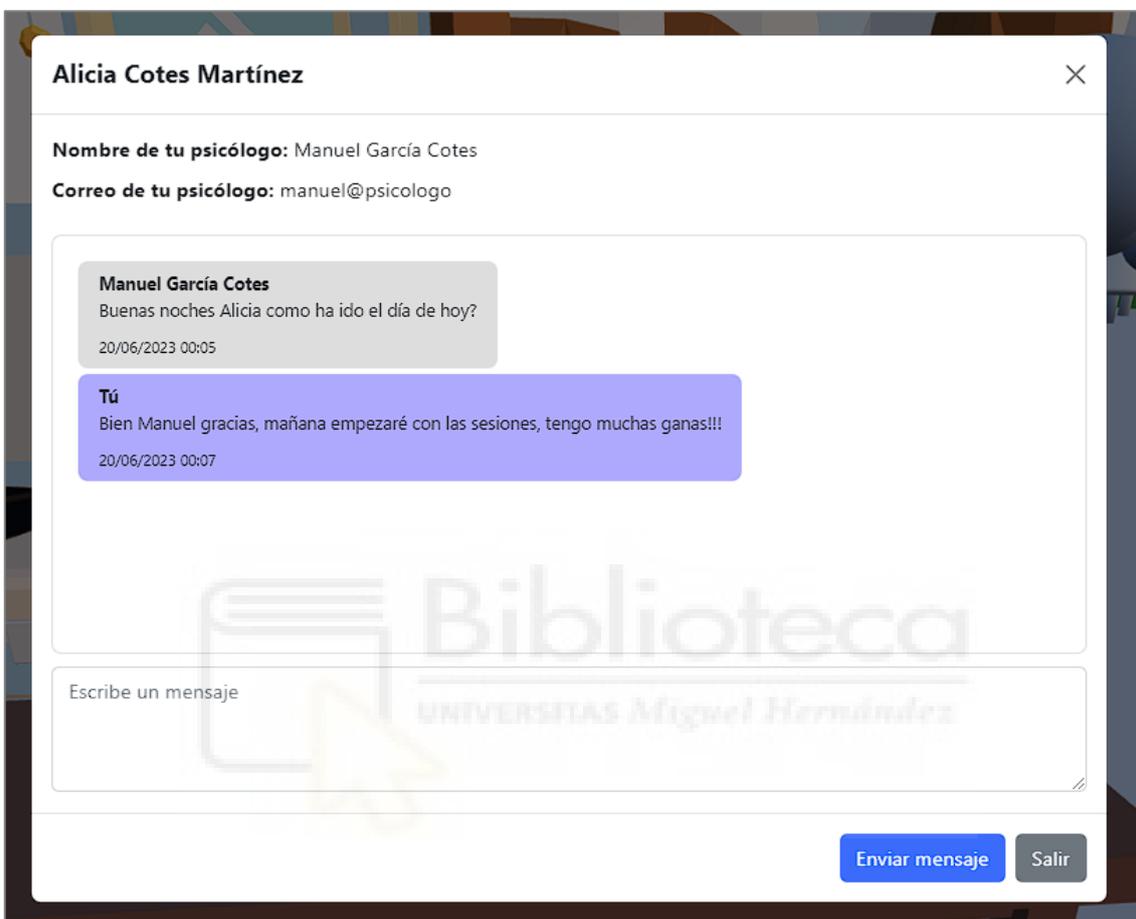


Figura 68: Ejemplo de la interfaz del portátil dentro del mundo virtual.

- **Cierre de sesión:** se ha implementado la opción para que el usuario pueda cerrar sesión en cualquier momento. Para realizar esta acción, el usuario deberá dirigirse a la casa de la mascota que se encuentra detrás de el tras aparecer entorno virtual, tal como se muestra en la Figura 69. Al hacer doble clic en la casa de la mascota, aparecerá un cuadro modal que le solicitará confirmar si desea cerrar sesión. Es importante tener en cuenta que, si el usuario decide cerrar sesión durante una sesión activa, deberá reiniciar dicha sesión desde el principio, ya que el progreso solo se guarda una vez que la sesión se haya completado.

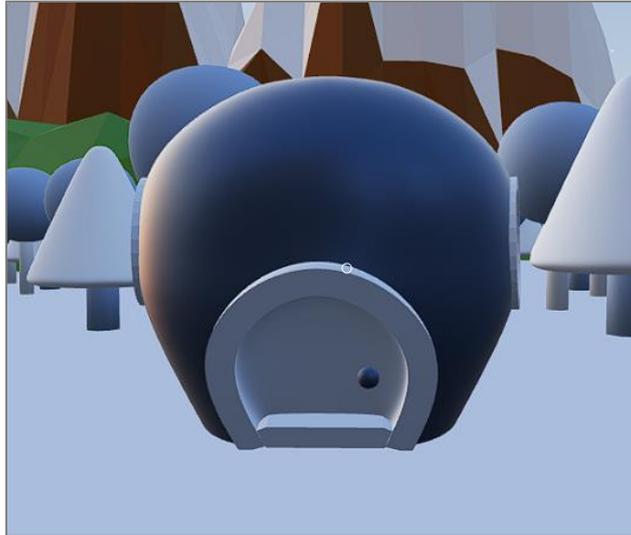


Figura 69: Ejemplo de la casa de la mascota que tiene como función cerrar sesión.

3.4.5. Introducción a las sesiones.

En este apartado se explicará cómo el usuario puede comenzar a realizar una sesión y cómo progresar a través de ellas. También se abordarán los aspectos más importantes de algunas de estas sesiones. Para iniciar, es necesario que el usuario interactúe con la mascota. Como se muestra en la Figura 70, al hacer clic en la mascota, se abrirá la opción de seleccionar entre las diferentes sesiones disponibles (ver Figura 71). A modo de ejemplo, se describirá una sesión de depresión.



Figura 70: Ejemplo de interacción con la mascota para iniciar una sesión.

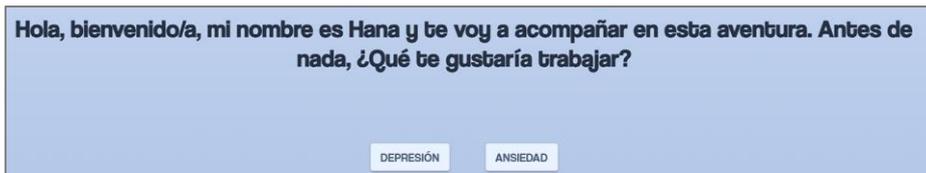


Figura 71: Ejemplo de selección de una sesión.

Una vez que el usuario haya seleccionado la sesión deseada, en este caso la sesión de depresión regresará al entorno virtual y el mensaje de la mascota se habrá actualizado. Cada sesión cuenta con un breve mensaje que indica la sesión en curso, su número y las instrucciones para continuar. Puede consultarse la Figura 72 para visualizar este mensaje actualizado.



Figura 72: Mensaje de ejemplo de la sesión 1 de depresión.

Conforme indica el mensaje de la mascota, si el usuario interactúa con ella haciendo clic, se mostrará un texto en la pizarra junto con un botón interactivo que dice "Siguiente". Este elemento se encuentra ilustrado en la Figura 73. Es importante destacar que esta primera sesión de depresión se enfoca exclusivamente en la psicoeducación. Por lo tanto, si el usuario lee el contenido de la pizarra y continúa haciendo clic en el botón "Siguiente", completará la sesión.

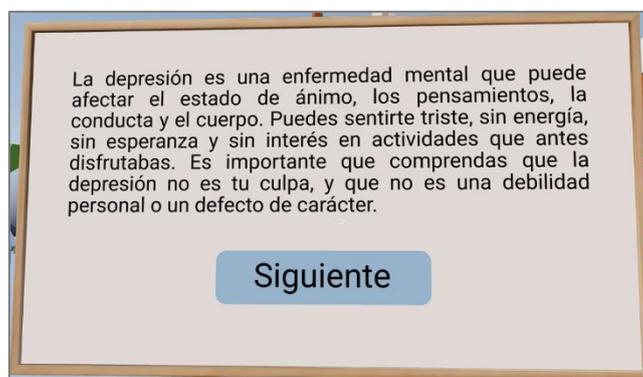


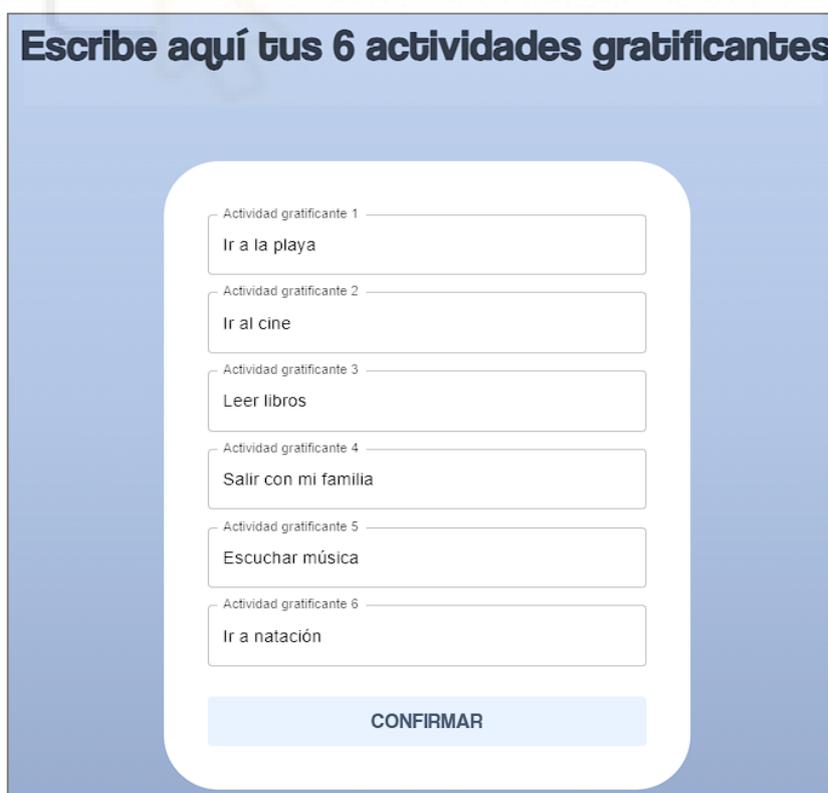
Figura 73: Ejemplo de la información mostrada en la pizarra de la sesión 1 de depresión.

Todas las sesiones siguen un patrón similar en su inicio. El usuario verá un breve mensaje explicativo por parte de la mascota. Después de interactuar con la mascota, se mostrará la información relevante en la pizarra. Una vez que el usuario haya leído la información, dará comienzo a la actividad correspondiente. Es importante destacar que la mayoría de las sesiones de depresión contienen diferentes tipos de actividades.

A continuación, se detallarán los elementos destacados de cada una de estas sesiones de manera individual. En caso de que alguna sesión en particular no se mencione, se debe a que se trata de sesiones informativas o de repaso, como es el caso de la Sesión 1 de Depresión.

3.4.6. Aspectos relevantes de las sesiones.

En este apartado se abordarán individualmente los aspectos más importantes de cada sesión. En la Figura 74, se puede observar al usuario realizando la actividad de "Actividades gratificantes". Esta actividad pertenece exclusivamente a la Sesión 2 de Depresión. En esta actividad, se le solicita al usuario que escriba 6 actividades gratificantes y posteriormente elija sus dos favoritas entre ellas, como se muestra en la Figura 75. El usuario no podrá avanzar hasta que seleccione únicamente dos actividades. Las actividades gratificantes seleccionadas por el usuario se guardarán y podrán ser visualizadas por su psicólogo asociado accediendo a la Sesión 2 en el perfil del usuario. Una vez completada la actividad, el usuario regresará al entorno virtual habiendo finalizado la segunda sesión.



Escribe aquí tus 6 actividades gratificantes

Actividad gratificante 1
Ir a la playa

Actividad gratificante 2
Ir al cine

Actividad gratificante 3
Leer libros

Actividad gratificante 4
Salir con mi familia

Actividad gratificante 5
Escuchar música

Actividad gratificante 6
Ir a natación

CONFIRMAR

Figura 74: Ejemplo de la actividad "Actividades gratificantes".

Ahora de esas 6 actividades o cosas que te gustan hacer, te pido que escojas 2 para realizarlas al menos una vez al día durante una semana. Es importante que te comprometas a realizarlas para así participar activamente en tu recuperación.

- Ir a la playa
- Ir al cine
- Leer libros
- Salir con mi familia
- Escuchar música
- Ir a natación

¡LO HARÉ!

Figura 75: Ejemplo de la actividad “Actividades gratificantes”.

El siguiente aspecto relevante se encuentra en la Sesión 3 de Depresión. Como se puede apreciar en la Figura 76, en esta sesión se lleva a cabo una interacción con el televisor. Una vez que el usuario haya leído la información en la pizarra, se le indicará que se dirija al televisor y presione el botón de “Play”. Esto activará un video de una sesión guiada de relajación muscular progresiva de Jacobson, el cual se encuentra alojado en YouTube y es proporcionado por el Psicólogo David Sánchez [67]. El televisor cuenta con dos botones: uno de “Play/pause” y otro de “Mute”, en caso de que el usuario desee detener o pausar el video, o silenciar el sonido. Una vez que el video de relajación haya concluido, el usuario habrá completado la sesión y podrá continuar con la siguiente.

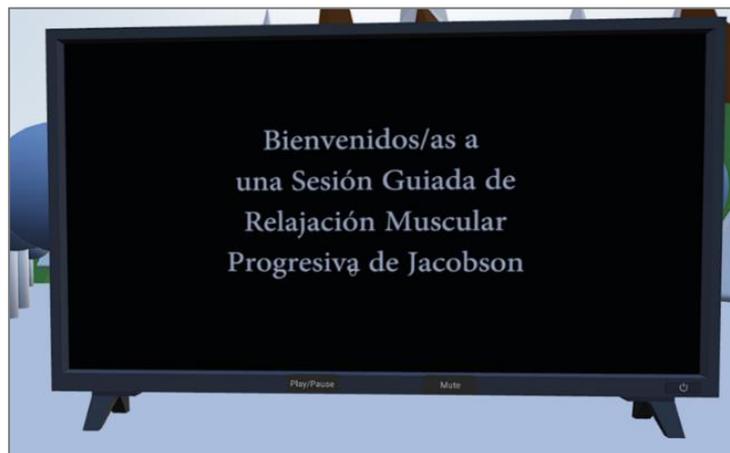


Figura 76: Ejemplo del video de relajación de la sesión 3.

Un aspecto relevante que se puede observar en la Sesión 4 es la interacción con el tocadiscos. Después de leer el contenido de la pizarra, el usuario debe dirigirse al tocadiscos y presionar el botón de “Play/pause” para reproducir el audio correspondiente a esta sesión, tal y como se muestra en la Figura 77. Los audios

del tocadiscos han sido grabados por la compañera de psicología. Esta actividad de interacción con el tocadiscos se repite en las sesiones 6, 8, 9 y 10. Algunos audios se comparten entre sesiones, específicamente las sesiones 4 y 8, así como las sesiones 6 y 9. En el caso de la sesión 10, el audio es único para esta sesión. La interacción con el tocadiscos es similar a la del televisor, ya que dispone de un botón de "Play/pause" y un botón de "Mute". Una vez que el audio haya finalizado, el usuario deberá dirigirse a la pizarra y leerá información acerca de la sesión realizada, posteriormente pulsará el botón mostrado y la sesión habrá finalizado.



Figura 77: Ejemplo de interacción con el tocadiscos.

En la Sesión 5, también se encuentra otro aspecto destacable, ya que se realiza la actividad de "Pensamiento automático". En esta sesión, se le explicará brevemente al usuario algunos conceptos relacionados con la actividad que se llevará a cabo a continuación. Una vez que el usuario haya leído toda la información, ingresará a la actividad en sí. En esta actividad, se le presentarán varios casos y deberá responder a un cuestionario, como se puede observar en la Figura 78. Una vez que haya completado el cuestionario, se le mostrará un mensaje indicando cuántos casos ha respondido correctamente y que casos ha respondido incorrectamente. Después de leer la evaluación y pulsar el botón correspondiente, la Sesión 5 se considerará finalizada.

Caso 3: Juan lleva trabajando dos años en un supermercado. Se acercan las fiestas de su pueblo y le gustaría tener un día libre para disfrutarlas pero cuando las va a pedir el jefe se las niega porque al ser un día de fiesta habrá muchos clientes y necesitan a todos los trabajadores en la tienda trabajando. En ese momento Juan pensó: "¡No puede creer que me nieguen tener un día libre cuando llevo dos años siéndole fiel a esta empresa, deberían agradecerme y permitirme ese día libre!"

Pensamiento polarizado

Minimización

Falacia de justicia

ENVIAR

Figura 78: Ejemplo de la actividad "Pensamiento automático" de la sesión 5.

El siguiente aspecto relevante se encuentra en la Sesión 7, donde se realizará la actividad de "Piloto automático". Al igual que en las demás sesiones, esta actividad se iniciará una vez que el usuario haya leído la explicación proporcionada en la pizarra. En esta actividad, se presentará una situación hipotética y el usuario deberá escribir cuál sería su pensamiento ante dicha situación, como se ilustra en la Figura 79. Posteriormente, se le presentará un cuestionario con varias preguntas relacionadas con su respuesta, como se muestra en la Figura 80. Una vez que haya completado el cuestionario, podrá leer la explicación final, de esta forma se dará por completada la sesión.

Ahora te pido que te imagines la siguiente situación

Llegas a clase y el profesor anuncia los resultados de un examen super importante del curso, y cuando reparte las notas, ves que tú has sido de los pocos alumnos de clase que ha suspendido. Escribe algunos pensamientos que te vendrían a la cabeza en ese momento:

Que uno de los alumnos que ha suspendido he sido yo.

ADELANTE

Figura 79: Ejemplo de la situación hipotética de la actividad "Piloto automático".

3. ¿Este pensamiento me está ayudando en la situación o me está hundiendo más?

No Creo que no No lo sé Puede ser SI

SIGUIENTE

Figura 80: Ejemplo de las preguntas sobre su situación hipotética de la actividad "Piloto automático".

La Sesión 9 es una de las sesiones más completas, ya que incluye tanto explicación previa en la pizarra, una actividad de relajación con audio, como la actividad principal llamada "Estoy presente cuando te hablo". Una vez que el usuario haya leído la explicación y haya realizado la actividad de relajación con el audio del tocadiscos, deberá dirigirse nuevamente a la pizarra para comenzar la actividad de "Estoy presente cuando te hablo". Esta actividad consiste en que se le presentarán una serie de imágenes. Una vez que las imágenes estén cargadas, se reproducirá un audio de fondo, como una risa, un llanto o un grito. En este momento, el usuario deberá seleccionar la imagen correspondiente al audio. Además, se le ofrece la opción de pulsar el botón "Play audio" para escuchar el audio nuevamente. Cuando el usuario elija una opción, se mostrará un borde verde o rojo alrededor de la imagen. El borde verde indicará que ha acertado la imagen correspondiente al audio, mientras que el borde rojo indicará que no lo ha logrado. Todo esto se puede apreciar en la Figura 81. Una vez que el usuario haya respondido todas las preguntas, al finalizar podrá ver un resumen

de la sesión realizada. Después de leer el resumen y pulsar el botón correspondiente, volverá al mundo virtual y la sesión se dará por completada.



Figura 81: Ejemplo de la relación de imágenes con su audio de la actividad "Estoy presente cuando te hablo".

Para concluir este apartado, es importante mencionar que no he profundizado en todas las sesiones debido a lo mencionado anteriormente. Las sesiones 6, 8 y 10 siguen el mismo formato que la sesión 4, pero con su respectiva explicación y audio. En cuanto a la Sesión 11, que es la última de las sesiones de depresión, se trata de un resumen del progreso del usuario a lo largo de las sesiones de depresión, donde simplemente se debe leer la explicación proporcionada en la pizarra. Una vez completada la Sesión 11, se habrán finalizado todas las sesiones de depresión y el usuario volverá al punto inicial, donde podrá elegir la siguiente sesión a realizar, en este caso, la sesión de ansiedad.

3.5. Vídeo con la Demo de la aplicación

Se ha elaborado un video resumen que exhibe un ejemplo práctico del funcionamiento de la aplicación. Se puede encontrar en el siguiente enlace: [68].



Figura 82: Vídeo con la demo de la aplicación web desarrollada

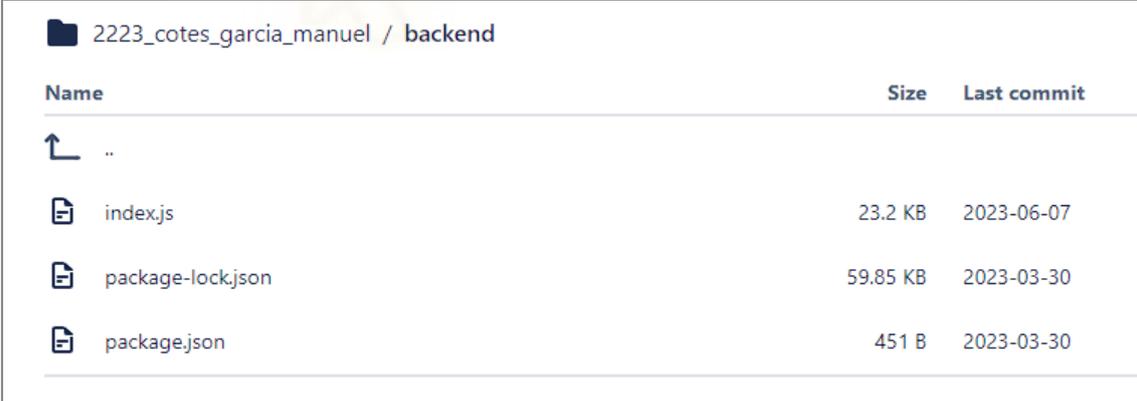
3.6. Código fuente

El código fuente de la aplicación está alojado en un repositorio privado de Bitbucket. Para acceder a este repositorio y explorar el código, se puede solicitar permisos de acceso en el siguiente enlace: [69]. En la Figura 83 podremos ver el contenido de la carpeta componentes del apartado de *frontend* y en la Figura 84 el contenido de la carpeta *backend*.



2223_cotes_garcia_manuel / frontend / src / components		
Name	Size	Last commit
..		
login		2023-06-07
three		15 minutes ago
usrPsicologo		15 minutes ago
Mensaje.jsx	818 B	2023-03-30
sharedStyles.js	25.32 KB	2023-05-29

Figura 83: Imagen donde se muestra la carpeta frontend del repositorio alojado en Bitbucket



2223_cotes_garcia_manuel / backend		
Name	Size	Last commit
..		
index.js	23.2 KB	2023-06-07
package-lock.json	59.85 KB	2023-03-30
package.json	451 B	2023-03-30

Figura 84: Imagen donde se muestra la carpeta backend del repositorio alojado en Bitbucket

4.- Conclusiones y trabajo futuro

El objetivo principal de este proyecto fue desarrollar un entorno virtual 3D personalizable e interactivo con el propósito de proporcionar tratamiento digital a personas que sufren de depresión o ansiedad, evitando así la necesidad de terapia completamente presencial. Además, se han proporcionado herramientas al psicólogo para acompañar al paciente durante este proceso, convirtiendo nuestro proyecto en una herramienta complementaria más que en una actividad externa.

Tras la finalización del proyecto, podemos concluir que hemos logrado desarrollar un prototipo funcional de una aplicación web 3D interactiva destinada al tratamiento de personas con depresión y ansiedad. Nuestra herramienta proporciona al usuario la posibilidad de llevar a cabo sesiones terapéuticas de forma digital en un entorno virtual personalizable. Además, hemos incorporado un chat interno para facilitar la comunicación con el psicólogo.

El estudio exhaustivo de aplicaciones relacionadas nos permitió comprobar que no existía ninguna aplicación web o móvil similar al proyecto que hemos desarrollado. Si bien es cierto que hay algunas aplicaciones con ciertos aspectos similares, como chats interactivos o cuestionarios de ayuda, ninguna es idéntica a la nuestra. Por tanto, podemos afirmar con total confianza que la herramienta creada en este proyecto tiene el potencial de ser de gran utilidad en el campo de la psicología, contribuyendo a una mejor digitalización de las terapias en esta era tan tecnológica.

Desde el punto de vista de la arquitectura de desarrollo de software y el análisis del estado del arte, hemos seleccionado tecnologías como React, Three.js y Node.js para la implementación de nuestra aplicación web 3D interactiva. Además, hemos utilizado una base de datos MySQL, la cual se integra de manera eficiente con el backend y el frontend de nuestra aplicación gracias al *framework* Express y librerías como Cors y Axios. Por último, hemos realizado pruebas en diferentes entornos, como los navegadores Google Chrome, Microsoft Edge, Opera GX y Firefox.

En cuanto a la funcionalidad lograda por nuestro prototipo, la aplicación web permite: visualizar el progreso de los pacientes a medida que realizan las sesiones programadas, proporcionar informes textuales detallados de cada sesión llevada a cabo por el usuario, incluyendo las respuestas de los cuestionarios asociados, en caso de que se hayan incluido, ofrecer al usuario un entorno personalizado basado en sus preferencias indicadas en el cuestionario de personalización previo, brindándole una sensación de libertad al permitirle moverse libremente por el entorno y cambiar la música de fondo para disfrutar de la melodía que más le haga sentir cómodo.

En cuanto al aspecto técnico, la implementación de una aplicación web de este tipo ha sido un gran desafío. Nos hemos enfrentado al diseño e implementación

de un sistema complejo que consta de tres tecnologías principales y sus respectivas librerías complementarias. Además, hemos tenido que abordar el diseño del entorno 3D, la integración de físicas y modelos, la comunicación interna entre los usuarios pacientes y los usuarios psicólogos, la implementación de un sistema de base de datos complejo, la personalización de elementos variables según las preferencias del usuario, así como una conexión fluida y rápida entre el *frontend* y el *backend*.

Algunos de los elementos más destacados que podemos resaltar son:

- Más de 25 funciones a las que mandar peticiones desde el *frontend*.
- Modelos personalizables hechos a medida.
- Más de 10 sesiones interactivas variables sobre el mismo entorno.
- Buena fluidez entre el entorno virtual y las actividades WEB.
- Sensación de personalización y de control en todo momento.

4.1. Trabajo futuro

A pesar de haber logrado un prototipo funcional que ofrece las funcionalidades que esperábamos inicialmente, a lo largo del desarrollo han surgido nuevas ideas que no hemos podido implementar debido a limitaciones de tiempo. Sin embargo, hemos considerado estas ideas como trabajo futuro, lo que nos permitirá continuar con el proyecto a nivel profesional y seguir mejorándolo.

- **Mejoras en los modelos:** reemplazar los modelos genéricos existentes e implementar nuevos modelos para elementos como la pizarra, el televisor, el tocadiscos, el portátil, entre otros. Nuestro objetivo es dotar a nuestro entorno virtual con una esencia única que estamos buscando y que creemos que aún se puede explotar aún más.
- **Más tonalidades de colores:** aunque actualmente ofrecemos 4 tonalidades entre las cuales el usuario puede elegir, creemos que brindarle total libertad para seleccionar cualquier color del espectro RGB sería un desafío, pero tendría un impacto significativo en la personalización estética. Esto supondría una mejora sustancial al permitir a los usuarios expresarse de manera más individualizada y adaptar el entorno virtual a sus preferencias exactas.
- **Mayor personalización del entorno virtual:** como desafío futuro, tenemos en mente permitir que el usuario acceda y personalice la casa dentro del entorno virtual. Esto implicaría la posibilidad de elegir distintos elementos estéticos como muebles o decoración, con el objetivo de brindarle una mayor sensación de hogar durante las sesiones terapéuticas. Queremos que esta herramienta no solo sea temporal, sino que se convierta en un espacio al que los usuarios puedan regresar de forma recurrente para personalizar su entorno. Esto les brindará la motivación y la actitud óptima para trabajar en las sesiones terapéuticas.

- **Mejora de la narrativa de las sesiones:** además, hemos considerado la posibilidad de ampliar las formas en las que se presenta la información durante las sesiones. En lugar de limitarse a una pizarra o a audios reproducidos en un tocadiscos, hemos ideado diferentes espacios dentro del entorno virtual. Estos espacios requerirán que el usuario se acerque con su personaje para continuar con las sesiones. De esta manera, el usuario podrá explorar el entorno mientras participa en las sesiones, lo que le permitirá establecer una conexión más profunda con el entorno virtual.
- **Mejor comunicación con el psicólogo:** a pesar de que la comunicación con el psicólogo ya está bastante avanzada, consideramos que sería de gran ayuda tener la opción de programar las sesiones dentro de la aplicación mediante videollamada. Esto permitiría al usuario realizar una videollamada directa con el psicólogo si lo necesitara. Además, también hemos contemplado la posibilidad de integrar el calendario de sesiones presenciales en la página web, de modo que el usuario pueda consultar las fechas y solicitar su siguiente sesión presencial directamente a través de la página web. Asimismo, sería beneficioso implementar un sistema de notificaciones por correo electrónico.
- **Pruebas con usuarios y puesta en práctica con pacientes:** una vez realizados los cambios, se planea enviar versiones de prueba a varios psicólogos para que lo implementen en sus sesiones y evaluar el progreso de los usuarios en su tratamiento mientras utilizan el entorno virtual.

Finalmente, la parte más ambiciosa de este proyecto es lograr una transformación estética significativa en el entorno virtual para proporcionar una experiencia altamente personalizable. Además, se busca una completa integración de la interactividad entre el psicólogo y el paciente. Esta integración incluirá la posibilidad de realizar videollamadas con el paciente, la gestión de citas desde el entorno virtual y un sistema de notificaciones externo que permitirá a ambos mantenerse informados sobre el progreso sin necesidad de estar conectados a la plataforma en ese momento.

5. Bibliografía

- [1] Organización Mundial de la Salud, “Salud mental del Adolescente.,” 2021. <https://www.who.int/es/news-room/fact-sheets/detail/adolescent-mental-health> (accessed Jun. 09, 2023).
- [2] V. Bermúdez, “Ansiedad, depresión, estrés y autoestima en la adolescencia. Relación, implicaciones y consecuencias en la educación privada,” *Cuestiones pedagógicas*, vol. 26, pp. 37–52, 2018.
- [3] Organización Mundial de la Salud, “Depresión.,” 2023. <https://www.who.int/es/news-room/fact-sheets/detail/depression> (accessed Jun. 09, 2023).
- [4] S. I. C. , y C. J. A. I. Caballo, *Manual de psicopatología y trastornos psicológicos*, 2a ed. Pirámide, 2014.
- [5] M. L. N. Pereira, “Una revisión teórica sobre el estrés y algunos aspectos relevantes de éste en el ámbito educativo,” *Revista educación*, pp. 171–190, 2009.
- [6] U. Barańczuk, “The five factor model of personality and emotion regulation: A meta-analysis,” *Pers Individ Dif*, vol. 139, pp. 217–227, 2019.
- [7] D. A. S. Corry and G. Leavey, “Adolescent trust and primary care: Help-seeking for emotional and psychological difficulties,” *J Adolesc*, vol. 54, pp. 1–8, 2017.
- [8] B. Horrillo-Álvarez, C. Marín-Martín, and M. R. Abuín, “La Adherencia al Entrenamiento en Meditación Mindfulness con Registro en Papel y en Aplicación Móvil,” *Clin Salud*, vol. 30, no. 2, pp. 99–108, Jun. 2019, doi: 10.5093/clysa2019a15.
- [9] E. Roca, “Terapia cognitiva para la depresión, según Beck.”
- [10] V. Guillén, C. Botella, and R. Baños, “PSICOLOGÍA CLÍNICA POSITIVA Y TECNOLOGÍAS POSITIVAS,” 2017. [Online]. Available: <https://www.redalyc.org/articulo.oa?>
- [11] Google Play Store, “TherapyChat en Google Play Store,” 2023. <https://play.google.com/store/apps/details?id=com.terapiachat.android&hl=es&gl=US&pli=1> (accessed Jun. 14, 2023).
- [12] S. Valera, “Best Mood Tracker Apps,” *Verywell mind*, Apr. 08, 2023. <https://www.verywellmind.com/best-mood-tracker-apps-5212922> (accessed Jun. 14, 2023).
- [13] Google Play Store, “Mindfulness Coach en Google Play Store,” 2023. <https://play.google.com/store/apps/details?id=gov.va.mobilehealth.ncptsd.mindfulnesscoach&hl=es&gl=US> (accessed Jun. 14, 2023).

- [14] Headspace, "Headspace: Meditación y Sueño," 2023. <https://www.headspace.com/es> (accessed Jul. 04, 2023).
- [15] Calm, "Calm - La aplicación #1 para meditar y dormir," 2023. <https://www.calm.com/es> (accessed Jul. 04, 2023).
- [16] J. Jerez, "Flujo y disfrute de los medios," *Teoría de la comunicación*, vol. 14, no. 4, pp. 328–347, 2004.
- [17] B. Perron and F. Schroter, *Video Games and the Mind: Essays on Cognition, Affect and Emotion*. Jefferson: McFarland, 2016.
- [18] Redacción Perú 21, "Estudiantes diseñan innovadores videojuegos para combatir la depresión y la ansiedad," *Perú 21*, Nov. 24, 2021. <https://peru21.pe/cheka/videojuegos/estudiantes-disenan-innovadores-videojuegos-para-combatir-la-depresion-y-la-ansiedad-estudiantes-tecnologia-videojuegos-noticia/> (accessed Jun. 14, 2023).
- [19] Rocky mountain forest productions, "Estética y diseño de interiores en Denver: efectos en la salud mental en general," *Rocky mountains forest productions*, Sep. 09, 2022. <https://www.rmfp.com/es/blog/aesthetics-interior-design-denver-effects-on-mental-health/> (accessed Jun. 14, 2023).
- [20] J. Cuello and J. Vittone, *Diseñando apps para móviles*, vol. 1. Catalina Duque Giraldo, 2013. Accessed: Jun. 14, 2023. [Online]. Available: https://books.google.fr/books?hl=en&lr=&id=ATiqsjH1rvwC&oi=fnd&pg=PA7&dq=por+que+los+usuarios+dejan+de+usar+aplicaciones&ots=a4en1T5tcm&sig=jaQPArkvrVJZLQyAqTIYk7vRmf4&redir_esc=y#v=onepage&q=por%20que%20los%20usuarios%20dejan%20de%20usar%20aplicaciones&f=false
- [21] The University of Auckland, "SPARX." 2021. Accessed: Jun. 30, 2023. [Online]. Available: <https://landing.sparx.org.nz/>
- [22] J. Li, Y.-L. Theng, and S. Foo, "Game-Based Digital Interventions for Depression Therapy: A Systematic Review and Meta-Analysis," *Cyberpsychol Behav Soc Netw*, no. 8, pp. 519–527, 2014.
- [23] I. Mallén, "Celeste y el potencial del videojuego para tratar la ansiedad," *Meristation as*, Sep. 17, 2018. https://as.com/meristation/2018/09/17/reportajes/1537172088_704714.html (accessed Jun. 14, 2023).
- [24] Hecate Comms, "Gris y la narrativa sobre el duelo," *Hecate Comms*, Aug. 27, 2021. <https://hecatecomms.com/gris-y-la-narrativa-sobre-el-duelo/> (accessed Jun. 14, 2023).
- [25] thatgamecompany, "Journey." Steam, 2020. Accessed: Jun. 30, 2023. [Online]. Available: <https://store.steampowered.com/app/638230/Journey/>

- [26] ustwo games, "Monument Valley." Google Play Store, 2014. Accessed: Jun. 30, 2023. [Online]. Available: <https://play.google.com/store/apps/details?id=com.ustwo.monumentvalley>
- [27] F. Nogales Urrutia, "Videojuego como herramienta para reducir el estrés AGRADECIMIENTOS," 2020.
- [28] YANA APP S.A.P.I. de C.V., "Yana: Tu acompañante emocional," *Google Play Store*, 2021. <https://play.google.com/store/apps/details?id=com.yanaapp&hl=es&gl=US> (accessed Jun. 30, 2023).
- [29] Buff Studio, "My Oasis: Relaxing, Satisfying," *Google Play Store*, 2021. <https://play.google.com/store/apps/details?id=com.buffstudio.myoasis&hl=es&gl=US> (accessed Jun. 30, 2023).
- [30] W. Sanchez, "2. La usabilidad en Ingeniería de Software- definicion y características," 2011, Accessed: Mar. 13, 2023. [Online]. Available: <http://www.redicces.org.sv/jspui/handle/10972/1937>
- [31] U. De Girona, "INTERACTIVIDAD E INTERACCIÓN MERITXELL ESTEBANELL MINGUELL," 2002. Accessed: Mar. 13, 2023. [Online]. Available: <https://mascvuex.unex.es/revistas/index.php/relatec/article/view/2>
- [32] S. Luján, *Programación de aplicaciones web: Historia. Principios básicos y Clientes web*. Editorial Club Universitario, 2002.
- [33] R. Velasco, "¿Quieres hacer apps para móviles? Lenguajes a conocer," *softzone.es*, Nov. 23, 2022. <https://www.softzone.es/programas/lenguajes/programar-apps-moviles/> (accessed Mar. 07, 2023).
- [34] Redacción KeepCoding, "Los 5 lenguajes más usados actualmente en desarrollo web," Dec. 20, 2022. <https://keepcoding.io/blog/lenguajes-desarrollo-web/> (accessed Mar. 08, 2023).
- [35] Edix, "Framework: qué es, para qué sirve y algunos ejemplos," 2022. <https://www.edix.com/es/instituto/framework/> (accessed Jun. 30, 2023).
- [36] Facebook INC, "React JS," 2023. <https://reactjs.org/> (accessed Mar. 09, 2023).
- [37] J. Jimenez, "Descubre React," 2015. Accessed: Mar. 09, 2023. [Online]. Available: <http://leanpub.com/descubre-react>
- [38] R. Cabello, "ThreeJS," *ThreeJS*, 2023. <https://threejs.org/> (accessed Mar. 09, 2023).
- [39] Jos. Dirksen, *Three.js essentials : create and animate beautiful 3D graphics with this fast-paced tutorial*. Packt Pub, 2014. Accessed: Mar. 09, 2023. [Online]. Available: <https://www.dominictran.com/pdf/ThreeJS.Essentials.PACKT.pdf>

- [40] L. Blue, "Discover three.js," *ThreeJS*, 2023. <https://discoverthreejs.com/> (accessed Mar. 10, 2023).
- [41] Open JS Foundation, "About Node.js," 2023. <https://nodejs.org/en/about/> (accessed Mar. 10, 2023).
- [42] J. Galan and M. Varela, "Ciclo de vida de los componentes," *Workshop - ReactJS*, 2019. <https://workshops-js-montevideo.gitbook.io/react/ciclo-de-vida-de-los-componentes> (accessed Jun. 17, 2023).
- [43] Facebook INC, "Presentando Hooks," 2023. <https://es.legacy.reactjs.org/docs/hooks-intro.html> (accessed Jun. 17, 2023).
- [44] J. Eguíluz Pérez, "Introducción a JavaScript," 2012. [Online]. Available: www.librosweb.es
- [45] Facebook INC, "Presentando JSX," 2023, Accessed: Mar. 13, 2023. [Online]. Available: <https://es.reactjs.org/docs/introducing-jsx.html>
- [46] Facebook INC, "Presentando JSX," *React*, 2023. <https://es.reactjs.org/docs/introducing-jsx.html> (accessed Mar. 10, 2023).
- [47] MySQL, "MySQL," 2021. <https://www.mysql.com/> (accessed Jul. 01, 2023).
- [48] C. Gyorodi, R. Gyorodi, G. Pecherle, and A. Olah, "A comparative study: MongoDB vs. MySQL," in *2015 13th International Conference on Engineering of Modern Electric Systems, EMES 2015*, Institute of Electrical and Electronics Engineers Inc., Jul. 2015. doi: 10.1109/EMES.2015.7158433.
- [49] C. Martí Escofet, "El lenguaje SQL." Accessed: Mar. 11, 2023. [Online]. Available: <https://pdfcursos.com/pdf/0018-lenguaje-sql.pdf>
- [50] Open JS Foundation, "Express," 2023. <https://expressjs.com/> (accessed Mar. 11, 2023).
- [51] npmjs, "mysql NodeJS," 2020. <https://www.npmjs.com/package/mysql> (accessed Mar. 11, 2023).
- [52] Open JS Foundation, "Cors Express," 2023. <https://expressjs.com/en/resources/middleware/cors.html> (accessed Mar. 11, 2023).
- [53] Open JS Foundation, "Body-parser express," 2023. <http://expressjs.com/en/resources/middleware/body-parser.html> (accessed Mar. 11, 2023).
- [54] M. Zabriskie, "Axios," 2023. <https://axios-http.com/es/docs/intro> (accessed Mar. 11, 2023).
- [55] designil, "Building Data Science Web Application with React, NodeJS, and MySQL," *Perth Ngarmtrakulchol*, 2018.

- <https://www.byperth.com/2018/04/19/guide-building-data-science-web-application-with-react-nodejs-and-mysql/> (accessed Mar. 11, 2023).
- [56] G. Méndez, E. Obviedo, G. Fallas, C. Vega, and A. Méndez, “Análisis de las herramientas Unity y Blender para el desarrollo de videojuegos con un enfoque educativo,” 2014.
- [57] Khronos Group, “glTF Project Explorer,” 2023. <https://github.khronos.org/glTF-Project-Explorer/> (accessed Jul. 01, 2023).
- [58] Blender Foundation, “Blender - Home of the Blender project - Free and Open 3D Creation Software,” 2023. <https://www.blender.org/> (accessed Jul. 03, 2023).
- [59] Poly Pizza, “Poly Pizza,” 2021. <https://poly.pizza/> (accessed Jul. 01, 2023).
- [60] Microsoft, “Documentación de Visual Studio Code,” 2023. <https://code.visualstudio.com/docs> (accessed Mar. 12, 2023).
- [61] Oracle, “MySQL Workbench,” 2023. <https://www.mysql.com/products/workbench/> (accessed Mar. 12, 2023).
- [62] Atlassian, “Bitbucket ,” 2023. <https://bitbucket.org/product/> (accessed Jul. 03, 2023).
- [63] Facebook, “React Developer Tools - Chrome Web Store,” 2023. <https://chrome.google.com/webstore/detail/react-developer-tools/fmkadmapgofadopljbjfkapdkoienihi> (accessed Jul. 03, 2023).
- [64] Atlassian, “Qué es scrum y cómo empezar,” 2021. <https://www.atlassian.com/es/agile/scrum> (accessed Jun. 15, 2023).
- [65] Pmndrs.docs, “Your first scene,” 2023. <https://docs.pmnd.rs/react-three-fiber/getting-started/your-first-scene> (accessed Jun. 17, 2023).
- [66] Pmndrs.docs, “GLTF file loader,” 2023. <https://gltf.pmnd.rs/> (accessed Jun. 18, 2023).
- [67] David Sánchez Psicología, “Relajación Muscular Progresiva de Jacobson | Sesión Guiada con Instrucciones,” *Youtube*, Sep. 05, 2017. https://www.youtube.com/watch?v=eu-2iWv_fCM (accessed Jun. 20, 2023).
- [68] M. Garcia Cotes, “Vídeo con la demostración de la herramienta realizada en el Trabajo de Fin de grado titulado Aplicación interactiva para la intervención y monitorización psicológica de síntomas de depresión y ansiedad en población adolescente.” 2023. Accessed: Jul. 04, 2023. [Online]. Available: <https://drive.google.com/drive/u/2/folders/1oK3iislxFtrTnbZqwpjBGNA0midGeavE>

- [69] M. Garcia Cotes, "Repositorio con el código fuente de la aplicación web del Trabajo de Fin de grado titulado Aplicación interactiva para la intervención y monitorización psicológica de síntomas de depresión y ansiedad en población adolescente.," *Bitbucket*, 2023. https://bitbucket.org/syliconlab-tfgtfms/2223_cotes_garcia_manuel (accessed Jul. 04, 2023).
- [70] P. Koeck, "Ayuda en línea contra el estrés, la ansiedad o la depresión con un autotest gratuito," 2023. <https://www.mis15minutos.com/> (accessed Mar. 13, 2023).
- [71] Yimmerlys Lopez, "Programador web vs móvil ¿cuáles son las diferencias?," 2022. <https://ed.team/blog/programador-web-vs-movil-cuales-son-las-diferencias> (accessed Mar. 13, 2023).
- [72] Log Rocket, "useState vs. useRef: Similarities, differences, and use cases," May 24, 2021. <https://blog.logrocket.com/usestate-vs-useref/> (accessed Jun. 17, 2023).
- [73] Poly Pizza, "Poly Pizza - Modelos 3D low poly," 2023. <https://poly.pizza/> (accessed Jun. 16, 2023).



ANEXO I: casos de uso desarrollados

En las siguientes tablas se incluyen los casos de uso desarrollados en la sección de 3.2 análisis y diseño del software.

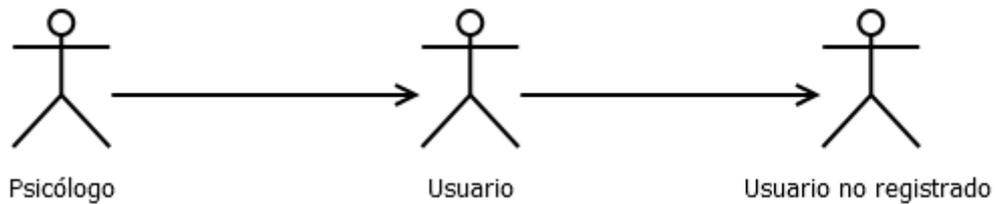


Figura 82: Diagrama de jerarquía

Actores relacionados en los casos de uso:

Actor	Psicólogo
Descripción	El psicólogo es responsable de llevar un seguimiento del progreso de su paciente y comunicarse con él a través del mundo interactivo. También puede tener acceso a todos sus datos y a toda la información proporcionada por la aplicación acerca de las sesiones.
Casos de uso relacionados	C.U.5, C.U.6, C.U.7, C.U.8

Actor	Usuario
Descripción	El usuario podrá acceder al mundo interactivo e ir realizando las sesiones programadas. También tiene la libertad de poder explorar los alrededores, cambiar la música de ambiente, elegir el color que más le guste para su mundo y hablar con su psicólogo siempre que lo necesite.
Casos de uso relacionados	C.U.4, C.U.9, C.U.10, C.U.11, C.U.12, C.U.13, C.U.14, C.U.15, C.U.16, C.U.17, C.U.18, C.U.19, C.U.20, C.U.21, C.U.22, C.U.23, C.U.24, C.U.25

Actor	Usuario no registrado
Descripción	Solo podrá visitar la página y registrarse en ella, pero no podrá acceder hasta que su psicólogo lo haya asignado como paciente.
Casos de uso relacionados	C.U.1, C.U.2, C.U.3

Aquí elaboramos un índice de los diferentes casos de uso que se elaboran a continuación:

1. Casos de uso de gestión de usuarios
 - a. C.U.1 Visitar la página WEB.
 - b. C.U.2 Registrarse como usuario en la página WEB.
 - c. C.U.3 Registrarse como psicólogo en la página WEB.
 - d. C.U.4 Iniciar sesión como usuario en la página WEB.
 - e. C.U.5 Iniciar sesión como psicólogo en la página WEB.
2. Casos de uso de gestión de pacientes
 - a. C.U.6 Ver paciente.
 - b. C.U.7 Añadir un paciente.
 - c. C.U.8 Escribir un mensaje al paciente.
3. Casos de uso de interacción con el mundo 3D
 - a. C.U.9 Cuestionario de personalización.
 - b. C.U.10 Interactuar con la mascota.
 - c. C.U.11 Interactuar con la casa de la mascota.
 - d. C.U.12 Interactuar con el tocadiscos.
 - e. C.U.13 Interactuar con el ordenador.
 - f. C.U.14 Escribir un mensaje al psicólogo
4. Casos de uso de progresión de las sesiones de depresión
 - a. C.U.15 Sesión 1 de depresión. Psicoeducación.
 - b. C.U.16 Sesión 2 de depresión. Actividades gratificantes.
 - c. C.U.17 Sesión 3 de depresión. Relajación muscular de Jacobson.
 - d. C.U.18 Sesión 4 de depresión. Consciencia del cuerpo y de la respiración.
 - e. C.U.19 Sesión 5 de depresión. ¿Pienso de manera automática?
 - f. C.U.20 Sesión 6 de depresión. Lo veo, lo acepto.
 - g. C.U.21 Sesión 7 de depresión. Botón de piloto automático.
 - h. C.U.22 Sesión 8 de depresión. Mundo lleno de estrés.
 - i. C.U.23 Sesión 9 de depresión. Estoy presente cuando te hablo.
 - j. C.U.24 Sesión 10 de depresión. Mi relación con los demás estando triste.
 - k. C.U.25 Sesión 11 de depresión. Todo lo que he aprendido.

Casos de uso de gestión de usuarios:

C.U.1	Visitar la página WEB	
Actores	Usuario no registrado.	
Descripción	Mediante el navegador cualquier usuario podrá acceder a la página WEB donde podrá usar nuestra aplicación interactiva.	
Dependencias	Ninguna.	
Precondición	Ninguna.	
Secuencia normal	Paso	Acción
	1	Buscar la página WEB.
	2	Entrar en la página WEB.
Postcondición	Ninguna.	
Excepciones	Ninguna.	
Rendimiento		
Frecuencia	Alta.	
Importancia	Alta.	
Urgencia	Alta.	
Comentarios	Ninguno.	

C.U.2	Registrarse como usuario en la página WEB.	
Actores	Usuario no registrado.	
Descripción	Una vez esté dentro de la página y quiera registrarse, el usuario tendrá que rellenar un formulario con sus datos y aceptar el consentimiento informado.	
Dependencias	C.U.1 Visitar la página WEB.	
Precondición	Ninguna.	
Secuencia normal	Paso	Acción
	1	Darle clic al botón de registrarse.
	2	Rellenar los campos necesarios para el registro.
Postcondición	El usuario se encontrará registrado en la WEB.	
Excepciones	En el caso de que algún campo sea incorrecto o no acepte el consentimiento informado no podrá registrarse en la WEB.	
Rendimiento		
Frecuencia	Alta.	
Importancia	Alta.	
Urgencia	Alta.	
Comentarios	Ninguno.	

C.U.3	Registrarse como psicólogo en la página WEB	
Actores	Usuario no registrado.	
Descripción	Una vez esté dentro de la página, tendrá que darle al botón de registrarse e introducir sus datos además de su número de colegiado.	
Dependencias	C.U.1 Visitar la página WEB.	
Precondición	El psicólogo debe introducir un numero de colegiado valido.	
Secuencia normal	Paso	Acción
	1	Darle clic al botón de registrarse.
	2	Rellenar los campos necesarios para el registro.
Postcondición	El psicólogo se encontrará registrado en la WEB.	
Excepciones	En el caso de que algún campo sea incorrecto o no acepte el consentimiento informado no podrá registrarse en la WEB.	
Rendimiento		
Frecuencia	Alta.	
Importancia	Alta.	
Urgencia	Alta.	
Comentarios	Ninguno.	

C.U.4	Iniciar sesión como usuario en la página WEB	
Actores	Usuario.	
Descripción	Un usuario previamente registrado podrá iniciar sesión en la página WEB.	
Dependencias	C.U.2 Registrarse como usuario en la página WEB.	
Precondición	Tener una cuenta registrada y que su psicólogo lo haya añadido como paciente.	
Secuencia normal	Paso	Acción
	1	Rellenar el formulario de inicio de sesión y acceder.
Postcondición	El usuario podrá acceder a la App interactiva.	
Excepciones	En el caso de que algún campo sea incorrecto o su usuario no haya sido añadido como paciente por un psicólogo no podrá acceder a la WEB.	
Rendimiento		
Frecuencia	Alta.	
Importancia	Alta.	
Urgencia	Alta.	
Comentarios	Ninguno.	

C.U.5	Iniciar sesión como psicólogo en la página WEB	
Actores	Psicólogo.	
Descripción	Una vez esté dentro de la página, tendrá que darle al botón de iniciar sesión e introducir su número de colegiado.	
Dependencias	C.U.3 Registrarse como psicólogo en la página WEB.	
Precondición	Tener una cuenta registrada como psicólogo.	
Secuencia normal	Paso	Acción
	1	Rellenar el formulario de inicio de sesión y acceder.
Postcondición	El psicólogo se encontrará registrado en la WEB.	
Excepciones	Ninguna.	
Rendimiento		
Frecuencia	Alta.	
Importancia	Alta.	
Urgencia	Alta.	
Comentarios	Ninguno.	

Casos de uso de gestión de pacientes:

C.U.6	Ver paciente	
Actores	Psicólogo.	
Descripción	Dentro del perfil del psicólogo podrá hacer clic en cualquiera de sus pacientes, de esta forma podrá ver sus datos asociados como nombre, apellidos, correo... También podrá ver los datos de las sesiones que ha realizado en la aplicación interactiva y tendrá un chat donde podrá hablar con su paciente.	
Dependencias	C.U.5 Iniciar sesión como psicólogo en la página WEB.	
Precondición	Tener una cuenta registrada como usuario psicólogo.	
Secuencia normal	Paso	Acción
	1	Iniciar sesión como psicólogo.
	2	Darle clic al botón de ver paciente.
Postcondición	Ninguna.	
Excepciones		
Rendimiento	Ninguno.	
Frecuencia	Alta.	
Importancia	Alta.	
Urgencia	Media.	
Comentarios	Ninguno.	

C.U.7	Añadir un paciente	
Actores	Psicólogo.	
Descripción	Dentro del perfil del psicólogo habrá un botón que ponga “Añadir paciente”, una vez pulsado le aparecerá un campo donde tendrá que introducir el correo del paciente. De esta forma el usuario pasará de usuario registrado a paciente.	
Dependencias	C.U.5 Iniciar sesión como psicólogo en la página WEB.	
Precondición	Que el usuario se haya registrado previamente.	
Secuencia normal	Paso	Acción
	1	Iniciar sesión como psicólogo.
	2	Darle clic al botón añadir paciente.
	3	Escribir el correo del usuario registrado y darle clic a confirmar.
Postcondición	El usuario seleccionado pasará de usuario registrado a paciente.	
Excepciones	El paciente sólo podrá ser añadido por un único psicólogo a la vez.	
Rendimiento	Ninguno.	
Frecuencia	Alta.	
Importancia	Alta.	
Urgencia	Media.	
Comentarios	Ninguno.	

C.U.8	Escribir un mensaje al paciente	
Actores	Psicólogo.	
Descripción	Dentro de los datos del paciente se podrán ver los mensajes escritos por el paciente. En el caso de que el psicólogo quiera responder tendrá un cuadro de texto donde podrá escribir al paciente, para confirmar el envío deberá darle clic al botón "Enviar mensaje".	
Dependencias	C.U.6 Ver paciente.	
Precondición	Ninguna.	
Secuencia normal	Paso	Acción
	1	Darle clic al botón "Ver paciente"
	2	Escribir el mensaje en el cuadro de texto y darle al botón enviar mensaje.
Postcondición	El paciente podrá ver el mensaje en el ordenador dentro de la aplicación interactiva.	
Excepciones		
Rendimiento	Ninguno.	
Frecuencia	Media.	
Importancia	Alta.	
Urgencia	Media.	
Comentarios	Ninguno.	

Casos de uso de interacción con el mundo 3D:

C.U.9	Cuestionario de personalización	
Actores	Usuario.	
Descripción	Cuando el usuario entre por primera vez al mundo 3D se le pasara un cuestionario de personalización para generar el mundo acorde a sus preferencias.	
Dependencias	C.U.4 Iniciar sesión como usuario en la página WEB.	
Precondición	Ninguna.	
Secuencia normal	Paso	Acción
	1	Realizar el cuestionario de personalización.
Postcondición	Se creará un mundo acorde a las opciones seleccionadas.	
Excepciones	Ninguna.	
Rendimiento		
Frecuencia	Media.	
Importancia	Alta.	
Urgencia	Media.	
Comentarios	Ninguno.	

C.U.10	Interactuar con la mascota	
Actores	Usuario.	
Descripción	Una vez el usuario esté dentro de la aplicación para seleccionar una sesión deberá darle clic a la mascota, esta le dará dos sesiones para elegir y el usuario debe seleccionar una de ellas.	
Dependencias	C.U.4 Iniciar sesión como usuario en la página WEB.	
Precondición	Ninguna.	
Secuencia normal	Paso	Acción
	1	Darle clic a la mascota.
	2	Seleccionar la sesión.
Postcondición	El usuario tendrá activa una sesión.	
Excepciones	Si el usuario selecciona una sesión tendrá que completarla antes de seleccionar otra.	
Rendimiento		
Frecuencia	Alta.	
Importancia	Alta.	
Urgencia	Alta.	
Comentarios	Ninguno.	

C.U.11	Interactuar con la casa de la mascota	
Actores	Usuario.	
Descripción	Para que el usuario pueda salir de la aplicación interactiva y cerrar sesión tendrá que darle doble clic a la casa de la mascota, una vez lo haga aparecerá un dialogo modal para preguntarle si desea cerrar sesión.	
Dependencias	C.U.4 Iniciar sesión como usuario en la página WEB.	
Precondición	Ninguna.	
Secuencia normal	Paso	Acción
	1	Darle doble clic a la casa de la mascota.
	2	Aceptar el dialogo modal para cerrar sesión.
Postcondición	El usuario habrá cerrado sesión	
Excepciones	Ninguna.	
Rendimiento		
Frecuencia	Alta.	
Importancia	Baja.	
Urgencia	Baja.	
Comentarios	Ninguno.	

C.U.12	Interactuar con el tocadiscos	
Actores	Usuario.	
Descripción	Cuando el usuario haga clic en el tocadiscos cambiará la música de ambiente.	
Dependencias	C.U.4 Iniciar sesión como usuario en la página WEB.	
Precondición	No estar con una sesión en progreso ya que durante las sesiones la música se pausará y el tocadiscos dejará de reproducir música.	
Secuencia normal	Paso	Acción
	1	Darle clic al tocadiscos.
Postcondición	La canción de ambiente habrá cambiado.	
Excepciones	Ninguna.	
Rendimiento	UNIVERSITAS Miguel Hernández	
Frecuencia	Baja.	
Importancia	Baja.	
Urgencia	Baja.	
Comentarios	Ninguno.	

C.U.13	Interactuar con el ordenador	
Actores	Usuario.	
Descripción	Cuando el usuario haga doble clic en el ordenador podrá acceder al chat que tiene con su psicólogo. También podrá ver sus datos de contacto en el caso de que los necesite.	
Dependencias	C.U.4 Iniciar sesión como usuario en la página WEB.	
Precondición	Ninguna.	
Secuencia normal	Paso	Acción
	1	Hacer doble clic en el ordenador.
Postcondición	El usuario habrá entrado en la interfaz del ordenador.	
Excepciones	Ninguna.	
Rendimiento		
Frecuencia	Media.	
Importancia	Alta.	
Urgencia	Baja.	
Comentarios	Ninguno.	

C.U.14	Escribir un mensaje al psicólogo	
Actores	Usuario.	
Descripción	Si el usuario lo desea podrá escribir a su psicólogo y leer los mensajes que le ha enviado este, para ello debe escribir el mensaje en el cuadro de texto y pulsar "Enviar Mensaje".	
Dependencias	C.U.13 Interactuar con el ordenador.	
Precondición	Ninguna.	
Secuencia normal	Paso	Acción
	1	Escribir el mensaje en el cuadro de texto y darle al botón enviar mensaje.
Postcondición	El psicólogo podrá ver el mensaje del paciente en su acceso identificado.	
Excepciones	Ninguna.	
Rendimiento		
Frecuencia	Media.	
Importancia	Alta.	
Urgencia	Media.	
Comentarios	Ninguno.	

Casos de uso de progresión de las sesiones de depresión:

C.U.15	Sesión 1 de depresión	
Actores	Usuario.	
Descripción	En esta sesión de depresión el usuario realizara una sesión de psicoeducación donde se le hablara sobre que es la depresión.	
Dependencias	C.U.10 Interactuar con la mascota.	
Precondición	Haber seleccionado la sesión de depresión.	
Secuencia normal	Paso	Acción
	1	Ir a la pizarra y leer la información mostrada.
Postcondición	El usuario habrá completado la sesión número 1 de depresión y su psicólogo podrá ver los datos asociados a ella.	
Excepciones	Ninguna.	
Rendimiento		
Frecuencia	Alta.	
Importancia	Alta.	
Urgencia	Media.	
Comentarios	Ninguno.	

C.U.16	Sesión 2 de depresión	
Actores	Usuario.	
Descripción	En esta sesión de depresión el usuario deberá leer la información de la pizarra y realizar la actividad “actividades gratificantes”.	
Dependencias	C.U.15 Sesión 1 de depresión.	
Precondición	Haber completado la sesión de depresión anterior.	
Secuencia normal	Paso	Acción
	1	Ir a la pizarra, leer la información mostrada y seguir las instrucciones.
	2	Realizar la actividad de actividades gratificantes.
Postcondición	El usuario habrá completado la sesión número 2 de depresión y su psicólogo podrá ver los datos asociados a ella.	
Excepciones	Ninguna.	
Rendimiento		
Frecuencia	Alta.	
Importancia	Alta.	
Urgencia	Media.	
Comentarios	Ninguno.	

C.U.17	Sesión 3 de depresión	
Actores	Usuario.	
Descripción	En esta sesión de depresión el usuario deberá leer la información de la pizarra e interactuar con el televisor para realizar una actividad de relajación.	
Dependencias	C.U.16 Sesión 2 de depresión.	
Precondición	Haber completado la sesión de depresión anterior.	
Secuencia normal	Paso	Acción
	1	Ir a la pizarra, leer la información mostrada y seguir las instrucciones.
	2	Darle clic al botón de "Play/Pause" para que el video comience a reproducirse.
	3	Ver el resumen de la actividad en la pizarra e interactuar con ella para finalizar la sesión.
Postcondición	El usuario habrá completado la sesión número 3 de depresión y su psicólogo podrá ver los datos asociados a ella.	
Excepciones	Ninguna.	
Rendimiento		
Frecuencia	Alta.	
Importancia	Alta.	
Urgencia	Media.	
Comentarios	El usuario podrá pausar el video y silenciarlo si lo cree necesario.	

C.U.18	Sesión 4 de depresión	
Actores	Usuario.	
Descripción	En esta sesión de depresión el usuario deberá leer la información de la pizarra e interactuar con el tocadiscos para realizar la actividad de “consciencia del cuerpo y respiración”.	
Dependencias	C.U.17 Sesión 3 de depresión.	
Precondición	Haber completado la sesión de depresión anterior.	
Secuencia normal	Paso	Acción
	1	Ir a la pizarra, leer la información mostrada y seguir las instrucciones.
	2	Darle clic al botón de “Play/Pause” para que el audio comience a reproducirse.
	3	Ver el resumen de la actividad en la pizarra e interactuar con ella para finalizar la sesión.
Postcondición	El usuario habrá completado la sesión número 4 de depresión y su psicólogo podrá ver los datos asociados a ella.	
Excepciones	Ninguna.	
Rendimiento		
Frecuencia	Alta.	
Importancia	Alta.	
Urgencia	Media.	
Comentarios	El usuario podrá pausar el audio y silenciarlo si lo cree necesario.	

C.U.19	Sesión 5 de depresión	
Actores	Usuario.	
Descripción	En esta sesión de depresión el usuario deberá leer la información de la pizarra y realizar la actividad de “¿Pienso de forma automática?”.	
Dependencias	C.U.18 Sesión 4 de depresión.	
Precondición	Haber completado la sesión de depresión anterior.	
Secuencia normal	Paso	Acción
	1	Ir a la pizarra, leer la información mostrada y seguir las instrucciones.
	2	Realizar el cuestionario de Pensamiento automático.
	3	Ver el resultado del cuestionario.
Postcondición	El usuario habrá completado la sesión número 5 de depresión y su psicólogo podrá ver los datos asociados a ella.	
Excepciones	Ninguna.	
Rendimiento		
Frecuencia	Alta.	
Importancia	Alta.	
Urgencia	Media.	
Comentarios	Ninguno.	

C.U.20	Sesión 6 de depresión	
Actores	Usuario.	
Descripción	En esta sesión de depresión el usuario deberá leer la información de la pizarra y realizar la actividad de “Lo veo, lo acepto”.	
Dependencias	C.U.19 Sesión 5 de depresión.	
Precondición	Haber completado la sesión de depresión anterior.	
Secuencia normal	Paso	Acción
	1	Ir a la pizarra, leer la información mostrada y seguir las instrucciones.
	2	Darle clic al botón de “play/pause” para que el audio comience a reproducirse.
	3	Ver el resumen de la actividad en la pizarra e interactuar con ella para finalizar la sesión.
Postcondición	El usuario habrá completado la sesión número 6 de depresión y su psicólogo podrá ver los datos asociados a ella.	
Excepciones	Ninguna.	
Rendimiento		
Frecuencia	Alta.	
Importancia	Alta.	
Urgencia	Media.	
Comentarios	El usuario podrá pausar el audio y silenciarlo si lo cree necesario.	

C.U.21	Sesión 7 de depresión	
Actores	Usuario.	
Descripción	En esta sesión de depresión el usuario deberá leer la información de la pizarra y realizar la actividad de “Botón de piloto automático”.	
Dependencias	C.U.20 Sesión 6 de depresión.	
Precondición	Haber completado la sesión de depresión anterior.	
Secuencia normal	Paso	Acción
	1	El usuario deberá dirigirse a la pizarra, leer la información mostrada y seguir las instrucciones.
	2	Realizar el cuestionario de la actividad botón de piloto automático.
Postcondición	El usuario habrá completado la sesión número 7 de depresión y su psicólogo podrá ver los datos asociados a ella.	
Excepciones	Ninguna.	
Rendimiento		
Frecuencia	Alta.	
Importancia	Alta.	
Urgencia	Media.	
Comentarios	Ninguno.	

C.U.22	Sesión 8 de depresión	
Actores	Usuario.	
Descripción	En esta sesión de depresión el usuario deberá leer la información de la pizarra y realizar la actividad de “Un mundo lleno de estrés”.	
Dependencias	C.U.21 Sesión 7 de depresión.	
Precondición	Haber completado la sesión de depresión anterior.	
Secuencia normal	Paso	Acción
	1	Ir a la pizarra, leer la información mostrada y seguir las instrucciones.
	2	Darle clic al botón de “Play/Pause” para que el audio comience a reproducirse.
	3	Ver el resumen de la actividad en la pizarra e interactuar con ella para finalizar la sesión.
Postcondición	El usuario habrá completado la sesión número 8 de depresión y su psicólogo podrá ver los datos asociados a ella.	
Excepciones	Ninguna.	
Rendimiento		
Frecuencia	Media.	
Importancia	Alta.	
Urgencia	Baja.	
Comentarios	El usuario podrá pausar el audio y silenciarlo si lo cree necesario.	

C.U.23	Sesión 9 de depresión	
Actores	Usuario.	
Descripción	En esta sesión de depresión el usuario deberá leer la información de la pizarra y realizar la actividad de “Estoy presente cuando te hablo”.	
Dependencias	C.U.22 Sesión 9 de depresión.	
Precondición	Haber completado la sesión de depresión anterior.	
Secuencia normal	Paso	Acción
	1	Ir a la pizarra, leer la información mostrada y seguir las instrucciones.
	2	Darle clic al botón de “Play/Pause” para que el audio comience a reproducirse.
	3	Ir a la pizarra para continuar con la actividad.
	4	Realizar la actividad de relacionar el audio con la imagen.
Postcondición	El usuario habrá completado la sesión número 9 de depresión y su psicólogo podrá ver los datos asociados a ella.	
Excepciones	Ninguna.	
Rendimiento		
Frecuencia	Media.	
Importancia	Alta.	
Urgencia	Baja.	
Comentarios	El usuario podrá pausar el audio y silenciarlo si lo cree necesario.	

C.U.24	Sesión 10 de depresión	
Actores	Usuario.	
Descripción	En esta sesión de depresión el usuario deberá leer la información de la pizarra y realizar la actividad de “Mi relación con los demás estando triste”.	
Dependencias	C.U.23 Sesión 9 de depresión.	
Precondición	Haber completado la sesión de depresión anterior.	
Secuencia normal	Paso	Acción
	1	Ir a la pizarra, leer la información mostrada y seguir las instrucciones.
	2	Darle clic al botón de “Play/Pause” para que el audio comience a reproducirse.
	3	Ver el resumen de la actividad en la pizarra e interactuar con ella para finalizar la sesión.
Postcondición	El usuario habrá completado la sesión número 10 de depresión y su psicólogo podrá ver los datos asociados a ella.	
Excepciones	Ninguna.	
Rendimiento		
Frecuencia	Media.	
Importancia	Alta.	
Urgencia	Baja.	
Comentarios	El usuario podrá pausar el audio y silenciarlo si lo cree necesario.	

C.U.25	Sesión 11 de depresión	
Actores	Usuario.	
Descripción	Esta será la última sesión, el usuario deberá leer la información de la pizarra en la cual habrá un resumen de su progreso por esta sesión de depresión.	
Dependencias	C.U.24 Sesión 10 de depresión.	
Precondición	Haber completado la sesión de depresión anterior.	
Secuencia normal	Paso	Acción
	1	Ira a la pizarra y leer la información mostrada.
Postcondición	El usuario habrá completado la sesión número 11 de depresión y su psicólogo podrá ver los datos asociados a ella.	
Excepciones	Ninguna.	
Rendimiento		
Frecuencia	Media.	
Importancia	Alta.	
Urgencia	Baja.	
Comentarios	Ninguno.	