PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR

FACULTAD DE INGENIERÍA

CARRERA DE INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN



Trabajo de Titulación

Tema: Desarrollo de una aplicación móvil sobre la estimulación temprana del pensamiento de los niños ecuatorianos entre 2 a 4 años. Caso de estudio, Machachi.

AUTOR:

María Gabriela Gómez Yánez

QUITO DM, DICIEMBRE DE 2022

DEDICATORIA

El presente proyecto de titulación se lo dedico a Dios y a la virgen María Auxiliadora quienes me ha dado la fortaleza, entendimiento, conocimiento, amor y protección para concluir con mi carrera, su divina presencia ha sido mi guía y motivación en los momentos de dificultad.

Se la dedico a toda mi familia en especial a mis padres y hermana quienes estuvieron a mi lado en cada paso y me dieron las herramientas necesarias para terminar mi carrera en Ingeniería en Tecnologías de la Información, sin su amor incondicional, apoyo, dedicación y sacrificio este gran logro no habría sido posible. A ellos les atribuyo todo el éxito de mi vida personal y académica.

Finalmente, quiero dedicarle este trabajo con amor a todos mis abuelitos, en especial a mi abuelita Glorinda Durán que con su bendición y consejos me ha dado fuerza para no rendirme a lo largo de la carrera. A mi abuelito Juan Yánez, quién fue una gran inspiración por su fortaleza e inigualable creatividad, aunque ya no se encuentre físicamente con nosotros sé que su espíritu innovador siempre me acompañará y bendecirá en mi vida profesional y personal.

AGRADECIMIENTO

En primer lugar, quiero agradecer a Dios y a la virgen María Auxiliadora quienes me han acompañado en la culminación de otra gran etapa académica, sin su amor, bendición y guía no hubiera podido cumplir este sueño.

Gracias Ingeniero Nelson Ivan Herrera Herrera, quién supo brindarme su tiempo, paciencia, y atención en el desarrollo de este proyecto de titulación, debido a su valiosa orientación y consejos pude concretar con éxito mi proyecto.

Extiendo mi más sincero agradecimiento a todos los docentes que fueron parte de mi desarrollo académico en la universidad, llevo conmigo cada uno de sus consejos, experiencias y enseñanzas que serán fundamentales para mi desarrollo como profesional en el área de tecnologías de la información.

A mis padres y en especial a mi hermana que siempre estuvieron para mí para darme ánimos y esperanza en los días más difíciles de la universidad, gracias por creer en mí y acompañarme en todo este proceso académico.

RESUMEN

La ejecución de este proyecto de titulación se enfocó en desarrollar una aplicación sobre la estimulación temprana del pensamiento en un consultorio de estimulación del desarrollo infantil de Machachi, manejado por una especialista en el manejo de dificultades del aprendizaje. Para el desarrollo del proyecto, se tuvo en cuenta varios fundamentos teóricos que ayudan a la comprensión del mismo.

La metodología que se utilizó para el desarrollo fue el método de investigación cualitativa y la metodología scrum qué ayuda a desarrollar el software de una manera ágil, iterativa e incremental.

Para el diseño de la aplicación, se tuvo que levantar los requerimientos funcionales y no funcionales del sistema, de acuerdo a las especificaciones del usuario, luego de ello, se realizó el diseño del software y los prototipos de baja y alta fidelidad, a continuación, se definieron los ejercicios de estimulación del pensamiento que irían dentro de la aplicación. Para la implementación se seleccionaron las herramientas de Ionic, API rest y el administrador de la base de datos Microsoft SQL Server. Se realizó la construcción del software, la base de datos y la comunicación entre el API rest, la aplicación en Ionic y el Microsoft SQL Server. Finalmente, se realizaron las pruebas funcionales del sistema una evaluación de la usabilidad que lo realizó la especialista y se mostró el resultado final de todo el proyecto de titulación.

Palabras clave: prototipo, ionic, api rest, base de datos, usabilidad

ÍNDICE

ÍNDICE DE FIGURAS Y GRÁFICOS III
ÍNDICE DE TABLAS V
CAPÍTULO I: INTRODUCCIÓN 6
1. MARCO DE REFERENCIA 6
1.1. Justificación 6
1.2. Planteamiento del problema 7
1.3. Objetivo General 8
1.4. Objetivos Específicos 8
1.5. Antecedentes 8
1.6. Alcance 9
CAPÍTULO II: FUNDAMENTOS TEÓRICOS 10
2. Marco Teórico 10
2.1. Aplicación móvil 10
2.2. Diseño centrado en el usuario 11
2.3. Prototipo 11
2.4. Usabilidad 11
2.5. Ingeniería de requerimientos 12
2.6. Lenguaje de programación 12
2.7. Ionic 13
2.8. Base de datos 13
2.9. Levantamiento de requerimientos 13
2.10. Metodología Scrum 14
CAPÍTULO III: METODOLOGÍA 15
3. Metodología 15
3.1. Tipo de investigación 15
3.2. Metodología de Desarrollo de Software 15
CAPÍTULO IV: DESARROLLO DE LA INVESTIGACIÓN 20
4. Desarrollo de la aplicación 20
4.1. Descripción del software 20
4.2. Levantamiento de requerimientos 21
4.3. Levantamiento de requerimientos funcionales 22
4.4. Levantamiento de requerimientos no funcionales 24

4.5. Especificación de las interfaces del sistema 25
4.6. Diseño del software 27
4.7. Prototipo de Baja Fidelidad 37
4.8. Prototipo de Alta Fidelidad 44
4.9. Definición de los ejercicios de estimulación del pensamiento 52
CAPÍTULO V: IMPLEMENTACIÓN 54
5. Implementación de la aplicación 54
5.1. Selección de herramientas para el desarrollo de la aplicación 54
5.2. Construcción del software 55
5.3. Construcción de la base de datos 75
5.4. Construcción y comunicación del api rest con la aplicación y la base de datos 76
5.5. Pruebas funcionales del sistema 80
5.6. Evaluación de la usabilidad 86
5.7. Resultado final de la aplicación 88
CONCLUSIONES Y RECOMENDACIONES 95
BIBLIOGRAFÍA 97
ANEXOS 102

ÍNDICE DE FIGURAS Y GRÁFICOS

Gráfico 1 Pila del Sprint 1 18
Gráfico 2 Pila de Sprint 2, 3 y 4 18
Gráfico 3 Pila de Sprint 5 y 6 18
Gráfico 4 Pila de Sprint 7 y 8 19
Gráfico 5 Diagrama de caso de uso del flujo principal 28
Gráfico 6 Diagrama de caso de uso del requerimiento funcional 1 29
Gráfico 7 Diagrama de caso de uso del requerimiento funcional 2 30
Gráfico 8 Diagrama de caso de uso del requerimiento funcional 3 31
Gráfico 9 Diagrama de caso de uso del requerimiento funcional 4 32
Gráfico 10 Diagrama de caso de uso del requerimiento funcional 5 33
Gráfico 11 Diagrama de caso de uso del requerimiento funcional 6 34
Gráfico 12 Diagrama de caso de uso del requerimiento funcional 7 35
Gráfico 13 Diagrama de caso de uso del requerimiento funcional 8 36
Gráfico 14 Diagrama arquitectónico del sistema desarrollado 36
Gráfico 15 Prototipo de baja fidelidad interfaz principal externa 37
Gráfico 16 Prototipo de baja fidelidad interfaz registro de usuario 38
Gráfico 17 Prototipo de baja fidelidad interfaces informativas 38
Gráfico 18 Prototipo de baja fidelidad interfaz menú externo 39
Gráfico 19 Prototipo de baja fidelidad interfaz menú principal interno 39
Gráfico 20 Prototipo de baja fidelidad interfaz de perfil 40
Gráfico 21 Prototipo de baja fidelidad interfaz de fichas 41
Gráfico 22 Prototipo de baja fidelidad interfaz de evaluados 41
Gráfico 23 Prototipo de baja fidelidad interfaz para agregar a un evaluado 42
Gráfico 24 Prototipo de baja fidelidad interfaz de ejercicios 43
Gráfico 25 Prototipo de baja fidelidad interfaz de actividades 43
Gráfico 26 Prototipo de baja fidelidad interfaz de menú interno 44
Gráfico 27 Prototipo de alta fidelidad interfaz principal 45
Gráfico 28 Prototipo de alta fidelidad interfaz de registro 45
Gráfico 29 Prototipo de alta fidelidad interfaces informativas 46
Gráfico 30 Prototipo de alta fidelidad menú externo 46
Gráfico 31 Prototipo de alta fidelidad interfaz principal 47
Gráfico 32 Prototipo de alta fidelidad interfaz fichas 48
Gráfico 33 Prototipo de alta fidelidad interfaz de evaluados 48
Gráfico 34 Prototipo de alta fidelidad interfaz para ingresar a un evaluado 49
Gráfico 35 Prototipo de alta fidelidad interfaz de ejercicios 50
Gráfico 36 Prototipo de alta fidelidad interfaz de actividades 50
Gráfico 37 Prototipo de alta fidelidad interfaz menú interno 51
Gráfico 38 Prototipo de alta fidelidad interfaz de perfil 51
Gráfico 39 Información de la instalación de Ionic 56
Gráfico 40 Creación de carpeta del proyecto 56
Gráfico 41 Plantilla de ionic con menú deslizante 57
Gráfico 42 Ejemplo de la creación de páginas en ionic por medio del comando ionic generate page Nombredepágina 58
Gráfico 43 Estructura y diseño barra de herramientas de las interfaces externas 59

Gráfico 44 Estructura y diseño de página login 60
Gráfico 45 Funcionalidad botón Ingresar y Registrarse 61
Gráfico 46 Estructura y diseño de página registro 62
Gráfico 47 Funcionalidad botón CREAR CUENTA y VOLVER AL INICIO 63
Gráfico 48 Estructura y diseño de páginas informativas 64
Gráfico 49 Estructura y diseño página ayuda y apoyo 64
Gráfico 50 Estructura y funcionalidad del menú externo 65
Gráfico 51 Estructura y diseño barra de herramientas de las interfaces internas 66
Gráfico 52 Estructura y diseño de la página de inicio 67
Gráfico 53 Estructura y diseño de la página de perfil 68
Gráfico 54 Funcionalidad botón Editar en página perfil 69
Gráfico 55 Estructura y diseño de la página de fichas 70
Gráfico 56 Estructura y diseño de la página de información de un evaluado 71
Gráfico 57 Estructura y diseño de la página para agregar a un nuevo evaluado 72
Gráfico 58 Estructura y diseño de la página de ejercicios 73
Gráfico 59 Estructura y diseño de la página de actividades 74
Gráfico 60 Estructura del menú interno de la aplicación 75
Gráfico 61 Diagrama de la base de datos relacional 76
Gráfico 62 Diagrama de comunicación entre aplicación, api rest y base de datos 77
Gráfico 63 Creación de Api Rest 77
Gráfico 64 Configuración con la aplicacion de Ionic en environment.ts 78
Gráfico 65 Configuración con la aplicación de Ionic en environment.prod.ts 79
Gráfico 66 Configuración de la conexión entre el api rest y la base de datos 80
Gráfico 67 Resultado final interfaz de inicio de sesión 88
Gráfico 68 Resultado final interfaz Registrarse 89
Gráfico 69 Resultado final interfaz Acuerdo del usuario 89
Gráfico 70 Resultado final interfaz política de privacidad 90
Gráfico 71 Resultado final interfaz Ayuda y apoyo 90
Gráfico 72 Resultado final de menú externo 91
Gráfico 73 Resultado final de interfaz inicio 91
Gráfico 74 Resultado final interfaz de perfil 92
Gráfico 75 Resultado final interfaz fichas 92
Gráfico 76 Resultado final de interfaz para ingresar un nuevo evaluado 93
Gráfico 77 Resultado final de la interfaz de Ejercicios 93
Gráfico 78 Resultado de la interfaz tipo de usuario 94
Gráfico 79 Resultado final de la interfaz de menú interno 94

ÍNDICE DE TABLAS

Tabla 1 Pila del producto 17
Tabla 2 Identificación de las necesidades del usuario 21
Tabla 3 Requerimientos funcionales para el desarrollo de la aplicación móvil sobre la estimulación temprana del pensamiento. 22
Tabla 4 Requerimientos no funcionales para el desarrollo de la aplicación móvil sobre la estimulación temprana del pensamiento 24
Tabla 5 Pruebas del requerimiento funcional 1 80
Tabla 6 Pruebas del requerimiento funcional 2 81
Tabla 7 Pruebas del requerimiento funcional 3 82
Tabla 8 Pruebas del requerimiento funcional 4 82
Tabla 9 Pruebas del requerimiento funcional 5 83
Tabla 10 Pruebas del requerimiento funcional 6 84
Tabla 11 Pruebas del requerimiento funcional 7 85
Tabla 12 Pruebas del requerimiento funcional 8 85

CAPÍTULO I: INTRODUCCIÓN

1. MARCO DE REFERENCIA

1.1. Justificación

El fondo de las Naciones Unidas para la Infancia (UNICEF) dentro de su estudio de conocimientos, aptitudes y prácticas sobre la percepción de la primera infancia desarrollado en 2022, ha determinado que la comunidad ecuatoriana desconoce sobre la estimulación temprana en el futuro de sus hijos causando que no ingresen al sistema escolar de manera adecuada y tengan dificultades en su aprendizaje. Es por esta razón, que se motiva a los cuidadores de los niños a llevarlos a un centro psicológico especializado en el desarrollo del pensamiento con el objetivo de que puedan tener un acceso directo a ejercicios de estimulación del pensamiento en edad temprana (UNICEF, 2022).

Trabajar la estimulación temprana del pensamiento en los niños de 2 a 4 años reforzará los vínculos neuronales del cerebro, además, de sus capacidades motrices, perceptivas, sociales, lingüística y cognitivas, en otras palabras, estimulan las áreas afectivas, intelectuales y psicomotriz de los niños (Ibañez, Mudarra, & Ibañez, 2004). Abordar este problema tendrá múltiples beneficios en el consultorio de estimulación del desarrollo infantil que se encuentra en Machachi, en el cantón Mejía, dónde se pondrá en práctica este proyecto de titulación, algunas de las ventajas son las siguientes: accesibilidad fácil a los ejercicios de estimulación del pensamiento para los niños, mayor facilidad de la recopilación de datos de los niños y sus cuidadores, eficiencia al momento de aplicar los ejercicios al niño evaluado y mejora del seguimiento de cada niño que va al consultorio a mejorar la estimulación temprana del pensamiento.

Este proyecto de titulación implementará una aplicación móvil para la estimulación temprana del pensamiento de los niños ecuatorianos entre 2 a 4 años.

1.2. Planteamiento del problema

La estimulación temprana del pensamiento es importante en los niños de 2 a 4 años debido a que ayuda a que tengan un mejor desarrollo psicomotor, evita las dificultades en el aprendizaje, aislamiento o mala socialización en sus primeros años de escolaridad (Morillo, Aguavil, Garcia, & Coello, 2022). Actualmente, el fondo de las Naciones Unidas para la Infancia (UNICEF) en el "Estudio de conocimientos, aptitudes y prácticas sobre la percepción de la primera infancia" desarrollado en el 2022, ha identificado que la comunidad ecuatoriana desconoce y valora poco la relevancia que tiene la estimulación temprana en el futuro de sus hijos y por tanto no les permite ingresar al sistema escolar de manera adecuada, es por esta razón, que se motiva a los responsables de los niños que se los lleve a un centro especializado en tratamientos psicopedagógicos que les brinde una herramienta para que tengan fácil acceso a los ejercicios de estimulación temprana del pensamiento (UNICEF, 2022). En vista de esta necesidad, este trabajo se enfocará en realizar una aplicación móvil que ayude a los padres o responsables de los niños a tener un acceso directo a ejercicios de estimulación del pensamiento en edad temprana.

En función de la problemática planteada se propone la siguiente pregunta principal:

- ¿Cómo se puede integrar la tecnología en un consultorio de estimulación del desarrollo infantil?

Y las siguientes preguntas secundarias:

- ¿Cuáles son las ventajas que ofrece el uso de una aplicación móvil con ejercicios para el desarrollo del pensamiento en niños de 2 a 4 años?
- ¿Cuáles son los ejercicios del desarrollo del pensamiento que tienen más prioridad y llaman la atención de los niños?

1.3. Objetivo General

Desarrollar una aplicación móvil sobre la estimulación temprana del pensamiento de los niños ecuatorianos entre 2 a 4 años. Caso de estudio: Machachi.

1.4. Objetivos Específicos

- Determinar los ejercicios que ayudan a la estimulación temprana del pensamiento para niños de 2 a 4 años.
- Analizar la mejor herramienta para el desarrollo de la aplicación móvil.
- Identificar los requerimientos funcionales y no funcionales de la aplicación móvil.
- Desarrollar la aplicación móvil para la estimulación temprana del pensamiento.

1.5. Antecedentes

La estimulación temprana de los niños no se reconoce como un aspecto crucial en la sociedad ecuatoriana para el desarrollo de los mismos, sin embargo, se ha demostrado que una buena estimulación puede ayudarles cognitiva, emocional y socialmente, es por esta razón, que se motiva a los responsables de los niños a llevarlos a centros especializados en el desarrollo infantil para que puedan brindarles herramientas y que tengan acceso directo a ejercicios de estimulación del pensamiento en edad temprana.

En Ecuador, se han realizado algunos trabajos similares sobre el desarrollo de aplicaciones móviles para la estimulación temprana a los niños, estos han tenido buenos resultados como el conocimiento de los padres y maestros de los juegos y ejercicios que se pueden realizar para estimular el pensamiento de los niños de forma divertida considerando las necesidades de los niños (Arciniegas Sigüenza & Tálbot, 2009). Además, de lograr que los responsables de los niños sigan un esquema semanal de algunos ejercicios o conceptos para que el niño no tenga un cansancio mental ni físico (Espinosa, 2014).

El desarrollar una aplicación sobre la estimulación temprana de los niños de 2 a 4 años en el consultorio de estimulación del desarrollo infantil de Machachi, ayudará a la especialista a brindar una nueva herramienta para que los responsables de los niños tengan conocimiento y un acceso directo sobre los ejercicios que deben realizar y mejorar la estimulación de los niños.

1.6. Alcance

Este proyecto de titulación está dirigido a un consultorio de estimulación del desarrollo infantil de Machachi, manejado por una especialista en el manejo de dificultades del aprendizaje, tiene como alcance desarrollar una aplicación móvil sobre el desarrollo del pensamiento de los niños ecuatorianos entre 2 a 4 años, el programa contendrá una sección de inicio de sesión y registro para que el evaluador o responsable del niño ingrese sus datos y cree una cuenta, una sección de perfil dónde el usuario registrado puede ver sus datos personales, una sección de fichas, dónde el especialista puede registrar la información de los niños que evalúa y una sección de ejercicios en dónde se puede encontrar toso tipo de ejercicios de estimulación del pensamiento.

CAPÍTULO II: FUNDAMENTOS TEÓRICOS

2. Marco Teórico

El desarrollo de una aplicación móvil para la estimulación temprana del pensamiento de los niños se basa en entender cómo funciona el cerebro humano y cómo se desarrolla en la infancia. Hay que comprender que el cerebro de los niños es como un plástico que puede ser moldeado mediante una estimulación adecuada, el realizar estas actividades en los niños puede mejorar la capacidad cognitiva, la creatividad, la memoria y la resolución de problemas. Esta aplicación estará diseñada para poder enfocarse en la estimulación de habilidades específicas como la atención, la memoria, percepción visual y resolución de problemas, además, de tener una interfaz gráfica atractiva para los niños con colores brillantes.

Para comprender un poco más el desarrollo de este proyecto de titulación se definen algunos conceptos, como los siguientes:

2.1. Aplicación móvil

Una aplicación móvil, también se la conoce como app en inglés es un software diseñado para ser utilizado en dispositivos móviles como celulares inteligentes o tablets, que brinda una extensa diversidad de funciones y servicios, desde herramientas de comunicación hasta ventas de productos y servicios, las aplicaciones móviles, se caracterizan por ser interactivas, automáticas y se encuentran diseñadas para brindar a los usuarios experiencias de alta calidad en dispositivos móviles (Benítez, 2016). En esta ocasión la aplicación móvil realizada se dividirá en algunas secciones: la de inicio de sesión y registro para que el evaluador ingrese al niño que será estimulado, la sección de juegos de estimulación con relación a la memoria, para reforzar el lenguaje de los niños y una sección de manualidades para hacer en casa.

2.2. Diseño centrado en el usuario

El diseño centrado en el usuario o DCU es una técnica de diseño iterativo que tiene como principal objetivo concebir las necesidades y particularidades del interesado a lo largo de todo el proceso de diseño, para poder generar interfaces gráficas y productos utilizables, accesibles y efectivos (Galeano, 2008). Este concepto es importante porque ayuda a mejorar la experiencia del usuario, en este caso el del evaluador y de los niños, aumenta la adopción y retención de los usuarios, esto quiere decir, que el diseño del software será más sencillo de utilizar y comprender, de esta manera el usuario podrá utilizarlo frecuentemente, genera fidelidad de los usuarios, debido a que satisface sus necesidad y expectativas. Finalmente, promueve la innovación en la aplicación, de esta manera su diseño estará enfocado en satisfacer las necesidades del usuario siempre.

2.3. Prototipo

Un prototipo es una representación tangible de un producto o servicio que sirve para probar y testear antes de lanzarlo al mercado o presentarlo como una versión mejorada de algo que ya existe. Algunos usuarios perciben un prototipo como un ejemplo real o ideal de su categoría, el objetivo del prototipo es permitir que los usuarios experimenten con el producto o servicio y determinar su viabilidad, haciendo mejoras si es necesario (Universidad Europea, 2022). El prototipo que se utilizará en el proyecto es el de baja fidelidad y alta fidelidad. El primero será utilizado con el objetivo de probar algunas ideas y conceptos del diseño de la aplicación, en cambio, el prototipo de alta fidelidad será utilizado para mostrarle al cliente una representación de la aplicación más detallada y representar como sería el funcionamiento de la misma.

2.4. Usabilidad

La usabilidad se describe como lo sencillo que es utilizar un elemento, como una herramienta o dispositivo electrónico, en el contexto del desarrollo de una aplicación de software, la usabilidad tiene como objetivo enfocarse en la sencillez con que los usuarios logran

llegar a un objetivo específico a través de su uso. La usabilidad puede ser difícil de evaluar porque hay que tomar en cuenta varios factores, por lo tanto, es importante establecer algunas metodologías que permitan calcular la usabilidad que tienen las aplicaciones (Enriquez & Casas, 2013). La usabilidad ayudará a garantizar que los usuarios puedan utilizar a la aplicación de manera satisfactoria y que el evaluador no presente ningún inconveniente en la aplicación al momento de realizar los ejercicios con los niños, se debe verificar que los usuarios pueden utilizar la aplicación de manera intuitiva, la interfaz gráfica debe contener colores y diseño coherentes, los usuarios deben poder utilizar la aplicación de manera rápida y debe contener una retroalimentación clara en caso de que se presenten errores.

2.5. Ingeniería de requerimientos

La ingeniería de requerimientos se la puede definir como la disciplina que se orienta en descubrir, documentar y mantener los requerimientos de un sistema o producto determinado. Su objetivo es verificar que todos los requisitos del cliente se encuentren completos, sean coherentes y factibles. En otras palabras, es el proceso por el cual se pueden intercambiar varios puntos de vista para poder recopilar y modelar lo que un sistema debe realizar (Pérez, 2005). La ingeniería de requerimientos permitirá definir y documentar claramente lo que el sistema debe hacer, cómo debe realizarlo, las restricciones y limitaciones que debe tener, es importante tener en cuenta los requerimientos del usuario, los requisitos del sistema y sobre todo se debe asegurar la calidad que va a tener el software.

2.6. Lenguaje de programación

Al lenguaje de programación se lo define como la agrupación de símbolos y normas por el cual las personas logran comunicar una serie de instrucciones y algoritmos a una computadora. Este permite que una computadora logre procesar rápidamente grandes y complejas cantidades de información, existen varios lenguajes de programación y cada uno puede ser utilizado en varios contextos diferentes (López, 2020). Para realizar esta aplicación móvil los lenguajes de

programación que se utilizar serán HTML, que se lo utiliza para estructurar el contenido y diseño de la aplicación, CSS para definir la apariencia visual de la aplicación como puede ser el estilo de las letras, colores y diseño. Por último, se utilizará JavaScript, para programar la lógica de la aplicación, como la interacción con el usuario y la manipulación de datos.

2.7. Ionic

Se lo define como un framework de código abierto, es utilizado para desarrollar software como aplicaciones móviles, estas se caracterizan principalmente por ser híbridas multiplataforma, esto quiere decir, que son compatibles en el sistema operativo de Android, iPhone o Windows, utiliza los lenguajes de programación HTML5, CSS y JavaScript para desarrollar un software (Irontec, 2022).

2.8. Base de datos

A las bases de datos se las define como una recopilación de datos estructurados, estos pueden ser almacenados dentro de un sistema informático como una un servidor de bases de datos. Las bases de datos suelen estar controladas por un DBMS, también conocido como un sistema de gestión de bases de datos, este puede cumplir con algunas acciones como recuperar, modificar, agregar o eliminar de forma rápida y eficiente los datos que se encuentran dentro de una base de datos (Oracle, 2023).

2.9. Levantamiento de requerimientos

Al momento de desarrollar un software, es necesario realizar el levantamiento de requerimientos, esta es una etapa dónde se debe tener la mayor comunicación con el cliente, en este caso es la especialista en el manejo de dificultades del aprendizaje, para poder identificar el origen de los requerimientos y de lo que hace un analista para poder recolectarlos (Aristizábal, 2009). Para realizar el levantamiento de requerimientos en este proyecto, se realizó una entrevista a la especialista en el manejo de dificultades del aprendizaje, dónde nos

indicó cuál era su problema y cómo quería que migraran el material de aprendizaje que maneja con los niños a una aplicación móvil, especificó algunas características específicas que deseaba que tenga la aplicación como un cuestionario inicial, el formulario a llenar de los niños, los colores y las imágenes. Teniendo todo esto en consideración se identifica las necesidades del usuario, los requerimientos funcionales y los requerimientos no funcionales.

2.10. Metodología Scrum

La metodología Scrum es de desarrollo ágil le permite al grupo de trabajo desarrollar proyectos complejos en un entorno dinámico y flexible, está basado en entregas parciales y también regulares. Esta metodología está basada en roles definidos como, por ejemplo: el product owner quién es responsable de maximizar el valor del producto, el scrum master quién asegura la comprensión al equipo y les ayuda a seguir la teoría de Scrum y el scrum team que es el equipo que se encarga de desarrollar el producto y entregarlo. En otras palabras, la metodología scrum se centra en el trabajo y colaboración en equipo, adaptación al cambio y a la mejora continua (Sáez, 2021).

CAPÍTULO III: METODOLOGÍA

3. Metodología

3.1. Tipo de investigación

Investigación Cualitativa

El tipo de investigación para el siguiente proyecto de titulación es de carácter cualitativo, por lo tanto, se utilizarán técnicas como entrevistas y grupos de discusión con la especialista en el manejo de dificultades del aprendizaje, con estas técnicas se podrán recopilar datos sobre la experiencia de la especialista en el consultorio del desarrollo infantil, la manera en cómo se realizan las terapias, la influencia de los padres en los niños y los desafíos que ha tenido en su trabajo.

Algunas de las técnicas de recuperación documental que se utilizarán para poder obtener la información cualitativa son entrevistas, libros dónde se observarán las técnicas que utiliza la doctora para realizar los respectivos tratamientos a cada niño y documentación de la especialista que utiliza cuando trata a un nuevo paciente.

3.2. Metodología de Desarrollo de Software

Para elaborar este proyecto de titulación se utilizará la metodología ágil SCRUM, esta se enfocará en la comunicación constante con el cliente que es la especialista en el manejo de dificultades del aprendizaje, el objetivo de esta metodología es ir desarrollando el software de manera iterativa e incremental para adaptarse rápidamente a los cambios y requerimientos del cliente (Sáez, 2021).

3.2.1. Propósito de la metodología de desarrollo de software

El propósito de esta metodología de desarrollo de software es proporcionar información de manera más accesible a las personas involucradas en el desarrollo de la aplicación “Centro de diagnóstico y tratamiento psicopedagógico Niño Jesús de Praga”.

3.2.2. Alcance y cargos de la metodología del desarrollo de software

Para el desarrollo de este software se determinó un período específico de tiempo desde el 1 de marzo de 2023 hasta el 1 de junio de 2023. Se definió los cargos que compone la metodología de Scrum, entre ellos se encuentra:

Scrum Master, desarrollado por Gabriela Gómez, quién organiza y orienta los objetivos para facilitar la práctica e implementación correcta de la metodología Scrum dentro del proyecto.

Product Owner, desarrollado por Gabriela Gómez, quien determina y prioriza los componentes que debe tener del software y garantiza la calidad de producto desarrollado.

Equipo de desarrollo, por esta ocasión al ser un proyecto de titulación, el equipo se compone únicamente por Gabriela Gómez fue quién fue diseñador, programador, analista y documentador del proyecto.

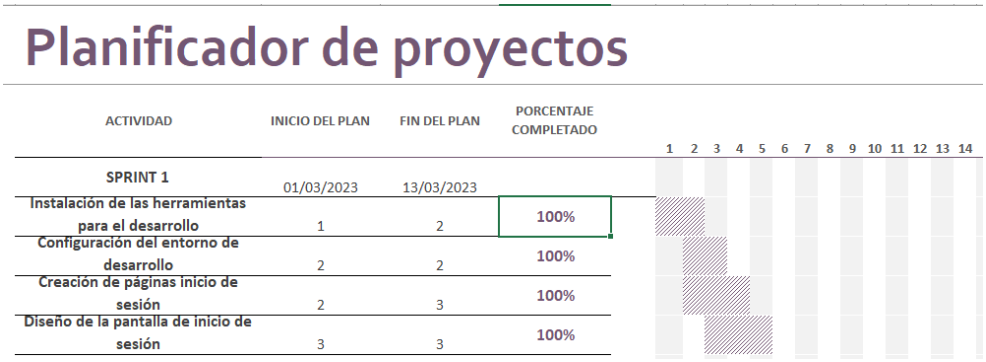
3.2.3. Artefactos

Se realizará la entrega de la pila del producto y la pila del sprint, además, hay que tener en cuenta la comunicación que hubo a lo largo del proyecto entre el usuario y el equipo de desarrollo.

Tabla 1 *Pila del producto*

ID	PRIORIDAD	DESCRIPCIÓN	ESTIMACIÓN DE ESFUERZO
1	MUY ALTA	Inicio de sesión a la aplicación	100
2	MUY ALTA	Registro de usuario en la aplicación	100
3	MEDIA	Desarrollo de menú externo	70
4	ALTA	Navegación correcta en las interfaces externas	90
5	ALTA	Desarrollo del diseño de interfaces internas	90
6	MEDIA	Desarrollo de menú interno	70
7	ALTA	Navegación correcta en las interfaces internas	90
8	MUY ALTA	Desarrollo de la base de datos	100
9	ALTA	Interacción en tiempo real con la base de datos	100
10	MUY ALTA	Desarrollo de CRUD para ingreso de nuevo paciente	100
11	MEDIA	Desarrollo de barra de búsqueda dentro de la aplicación	80
12	MEDIA	Desarrollo de portabilidad	100
13	ALTA	Eficiencia del software	100
14	MEDIA	Fiabilidad en el diseño	100
15	ALTA	Disponibilidad del software	100
16	ALTA	Documentación de la aplicación	100

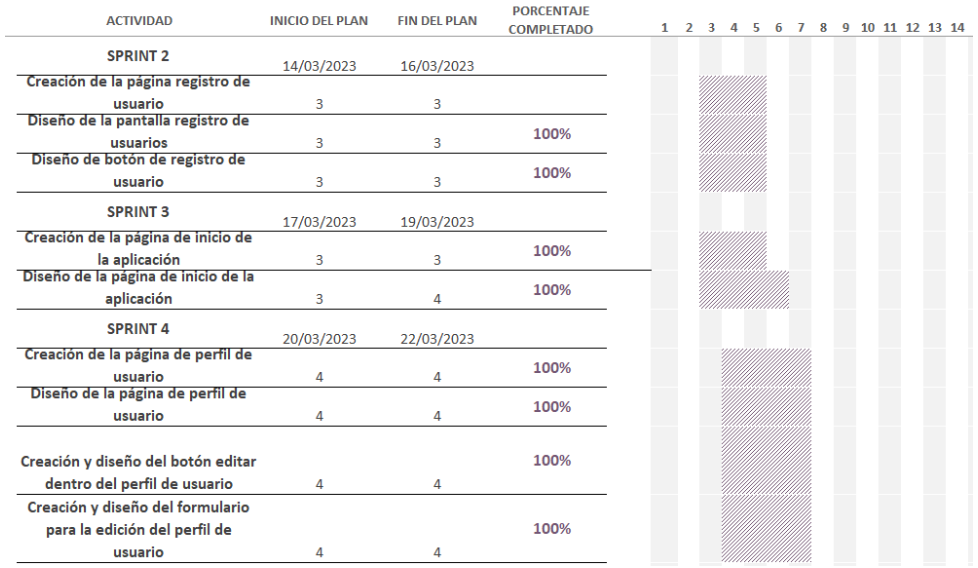
Gráfico 1 *Pila del Sprint 1*



Planificador de proyectos

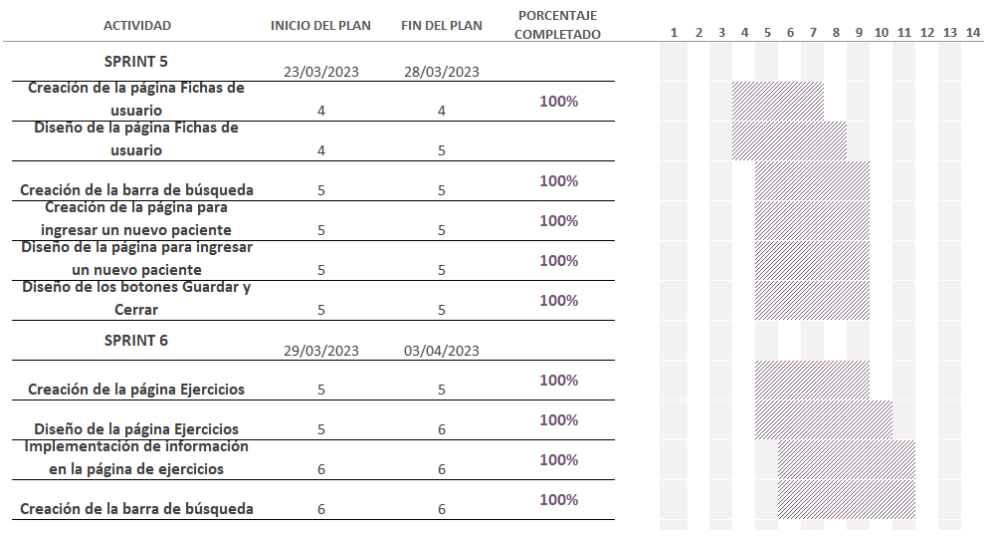
ACTIVIDAD	INICIO DEL PLAN	FIN DEL PLAN	PORCENTAJE COMPLETADO
SPRINT 1	01/03/2023	13/03/2023	
Instalación de las herramientas para el desarrollo	1	2	100%
Configuración del entorno de desarrollo	2	2	100%
Creación de páginas inicio de sesión	2	3	100%
Diseño de la pantalla de inicio de sesión	3	3	100%

Gráfico 2 *Pila de Sprint 2, 3 y 4*



ACTIVIDAD	INICIO DEL PLAN	FIN DEL PLAN	PORCENTAJE COMPLETADO
SPRINT 2	14/03/2023	16/03/2023	
Creación de la página registro de usuario	3	3	
Diseño de la pantalla registro de usuarios	3	3	100%
Diseño de botón de registro de usuario	3	3	100%
SPRINT 3	17/03/2023	19/03/2023	
Creación de la página de inicio de la aplicación	3	3	100%
Diseño de la página de inicio de la aplicación	3	4	100%
SPRINT 4	20/03/2023	22/03/2023	
Creación de la página de perfil de usuario	4	4	100%
Diseño de la página de perfil de usuario	4	4	100%
Creación y diseño del botón editar dentro del perfil de usuario	4	4	100%
Creación y diseño del formulario para la edición del perfil de usuario	4	4	100%

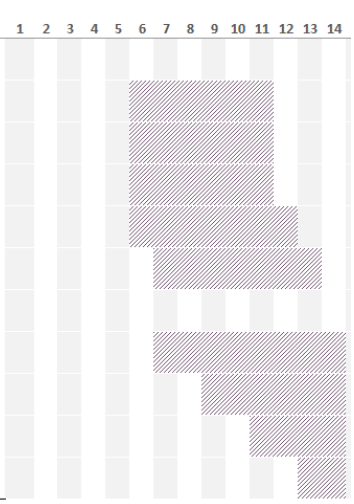
Gráfico 3 *Pila de Sprint 5 y 6*



ACTIVIDAD	INICIO DEL PLAN	FIN DEL PLAN	PORCENTAJE COMPLETADO
SPRINT 5	23/03/2023	28/03/2023	
Creación de la página Fichas de usuario	4	4	100%
Diseño de la página Fichas de usuario	4	5	
Creación de la barra de búsqueda	5	5	100%
Creación de la página para ingresar un nuevo paciente	5	5	100%
Diseño de la página para ingresar un nuevo paciente	5	5	100%
Diseño de los botones Guardar y Cerrar	5	5	100%
SPRINT 6	29/03/2023	03/04/2023	
Creación de la página Ejercicios	5	5	100%
Diseño de la página Ejercicios	5	6	100%
Implementación de información en la página de ejercicios	6	6	100%
Creación de la barra de búsqueda	6	6	100%

Gráfico 4 *Pila de Sprint 7 y 8*

ACTIVIDAD	INICIO DEL PLAN	FIN DEL PLAN	PORCENTAJE COMPLETADO
SPRINT 7	04/04/2023	11/04/2023	
Creación de Menú Interno	6	6	100%
Funcionalidad de Navegación del menú interno	6	6	100%
Creación de Menú Externo	6	6	100%
Funcionalidad de Navegación del menú externo	6	7	100%
Funcionalidad de Navegación entre pantallas	7	7	100%
SPRINT 8	12/04/2023	28/05/2023	
Creación de la base de datos	7	9	100%
Conexión entre aplicación, API REST y base de datos	9	11	100%
Funcionalidad de botones en todas las pantallas	11	13	100%
Realización de pruebas en la aplicación	13	14	100%



CAPÍTULO IV: DESARROLLO DE LA INVESTIGACIÓN

4. Desarrollo de la aplicación

4.1.Descripción del software

Esta aplicación móvil sobre la estimulación temprana del pensamiento de los niños ecuatorianos entre 2 a 4 años es un software independiente para el uso exclusivo de la especialista en el manejo de dificultades del aprendizaje y los responsables de sus pacientes con el objetivo de darles una herramienta para que los evaluados puedan mejorar la estimulación del desarrollo del pensamiento y no tengan problemas del aprendizaje en el futuro.

Funcionalidad del producto

La aplicación debe permitir al evaluador ingresar los datos del niño a estimular, luego debe permitir realizar una ficha médica del niño, además, es necesario que la aplicación cuente con un test brindado por la especialista para poder evaluar las áreas debilitadas y que faltan y que hay que desarrollar en el niño, a continuación, debe presentarse la sección de ejercicios de estimulación donde los niños reforzarán el área necesaria, estas pueden estar relacionadas con su lenguaje y una sección de manualidades, entre otros.

Características del usuario

Tipo de usuario: Para esta aplicación se necesita que se encuentren presentes dos tipos de usuarios, el evaluador que debe conocer exactamente de lo que trata la aplicación y el cuidador del niño evaluado.

Formación: No es requerido que ningún usuario cuente con alguna formación previa

Habilidades: Conocer las funcionalidades básicas de un teléfono.

Actividades: Referidas del profesional.

Restricciones

- La aplicación debe ser fácil de utilizar y comprensible para los niños.
- El software debe ser desarrollado para teléfonos con el sistema operativo iOS y Android.

Suposiciones y dependencias

La aplicación se podrá ejecutar en los equipos que tengan el sistema operativo iOS y Android.

4.2.Levantamiento de requerimientos

Para el levantamiento de requerimientos se tuvo que identificar las necesidades del usuario, que en este caso es la especialista en el manejo de dificultades del aprendizaje, todos los datos que se mostrarán a continuación, se obtuvieron a partir de entrevistas y grupos de discusión que hubo a lo largo del desarrollo de este proyecto de titulación para la ejecución del software.

Tabla 2 *Identificación de las necesidades del usuario*

No.	Categoría	Característica
1	Interfaz de usuario	La aplicación ser sencilla e intuitiva, puede contener colores verde, amarillo o blanco y puede existir una mezcla entre estos colores.
2	Inicio de sesión y registro	La aplicación debe contener una interfaz de inicio de sesión y registro para el ingreso a la información.
3	Perfil de usuario	La aplicación debe contener una sección que muestre el nombre, apellido, e-mail y descripción del usuario.
4	Contacto del centro	La aplicación debe mostrar el lugar dónde se encuentra el centro y su contacto.
5	Ficha médica	La aplicación debe poder crear una nueva ficha médica para crear un nuevo historial del evaluado
6	Datos del evaluado	Dentro de la ficha médica, debe incluir datos como: fecha de ingreso del evaluado, nombres, apellidos, fecha de nacimiento, edad, sexo, información de los

		padres como sus nombres, apellidos, nacimiento y trabajo.
7	Ejercicios de fortalecimiento	La aplicación debe contener una sección dónde se muestren los tipos de ejercicios que hay para fortalecer para las áreas debilitadas del evaluado.
8	Menú de la aplicación	La aplicación debe contener un menú que le permita al usuario navegar fácilmente.
9	Idioma	Toda la aplicación debe ser en español
10	Tipo y tamaño de letra	La aplicación debe tener un tipo y tamaño de letra legible para cualquier persona.

4.3.Levantamiento de requerimientos funcionales

Los requerimientos funcionales son los requisitos del sistema sobre la funcionalidad e interacción que tiene el programa con su entorno (Universidad de Granada DECSAI, s.f.). La tabla que se presenta a continuación muestra los requerimientos funcionales que se han identificado luego de las entrevistas y grupos de discusión que se tuvo con el usuario.

Tabla 3 *Requerimientos funcionales para el desarrollo de la aplicación móvil sobre la estimulación temprana del pensamiento.*

Referencia	Nombre	Descripción	Prioridad
RF1	Inicio de sesión y registro	El sistema debe permitir al usuario registrar sus datos y acceder a la aplicación con un usuario y contraseña	ALTA
FR2	Desarrollo de interfaces externas	El sistema debe contar con cinco pantallas externas (Inicio de sesión, Registro, Ayuda,	MEDIA

		Acuerdo con el usuario y política de privacidad)	
RF3	Desarrollo de interfaces internas	El sistema debe contar con 4 pantallas internas (Perfil, Fichas Inicio y Ejercicios)	ALTA
RF4	Navegación entre interfaces	El sistema debe contar con botones que permitan la correcta navegación entre todas las interfaces	ALTA
RF5	Gestión de datos	El sistema debe tener conexión con una base de datos para manejar la edición de los mismos en las interfaces que lo requieran	ALTA
RF6	Cambios en tiempo real	El usuario debe ser capaz de ver los cambios que realiza el sistema tiempo real	ALTA
RF7	Advertencias y mensajes de error	El usuario debe ser capaz de ver los errores o advertencias que devuelve el sistema	ALTA
RF8	Funcionalidad de botones	Varios de los botones tendrán las funciones de un CRUD por lo que el sistema debe funcionar correctamente	ALTA

4.4.Levantamiento de requerimientos no funcionales

Los requerimientos no funcionales son requisitos del usuario sobre cómo debe verse y ser el sistema, cómo por ejemplo el color del sistema, la disponibilidad del programa, entre otros. (Universidad de Granada DECSAI, s.f.). La tabla que se presenta a continuación muestra los requerimientos no funcionales que se han identificado luego de las entrevistas y grupos de discusión que se tuvo con el usuario.

Tabla 4 *Requerimientos no funcionales para el desarrollo de la aplicación móvil sobre la estimulación temprana del pensamiento*

Referencia	Nombre	Descripción	Prioridad
RNF1	Interfaz del sistema	El sistema debe ser intuitivo, sencillo y agradable para el usuario	ALTA
RNF2	Color del sistema	El sistema debe contener los colores verde, amarillo o blanco y puede existir una mezcla entre ellos.	ALTA
RNF3	Tipo y tamaño de letra	La aplicación debe tener un tipo y tamaño de letra legible para cualquier persona.	ALTA
RNF4	Flexibilidad	Debe ser sencillo realizar una acción anterior y debe adaptarse a la resolución de pantalla	ALTA
RNF5	Eficiencia	Las acciones realizadas dentro del software deben ejecutarse rápidamente	ALTA

RNF6	Compatibilidad con sistemas operativos	El sistema debe ser compatible con usuarios Android y usuarios iOS.	MEDIA
RNF7	Disponibilidad del software	El programa debe estar disponible para el usuario en cualquier momento	ALTA
RNF8	Documentación	El software debe contar con una documentación adecuada, clara y precisa.	MEDIA

4.5. Especificación de las interfaces del sistema

Las interfaces del usuario se dividirán en interfaces externas e interfaces internas, las cuales serán explicadas a continuación:

Interfaces externas

Las interfaces externas son aquellas que cualquier usuario no registrado en la base de datos de la aplicación puede ver. Entre ellas se encuentra se encuentran:

1. Interfaz principal: Esta consiste en un espacio para que la persona que este utilizando el software ingrese su usuario y contraseña. También cuenta con dos botones el de "Ingresar" que permite adentrarse en la aplicación y "Registrarse".
2. Interfaz de Registro: Esta interfaz le permite a la persona que utiliza el software ingresar los datos requeridos para crear una cuenta.
3. Interfaces informativas: Estas interfaces muestran información importante sobre la aplicación, consiste en la interfaz "Acuerdo del usuario", la interfaz "Política de

privacidad” y la interfaz “Ayuda y apoyo”, en esta última se podrá encontrar las preguntas frecuentes que suelen realizar los usuarios cuando utilizan un software.

4. Interfaz de menú: La interfaz de menú consistirá en mostrar todos los títulos de las interfaces externas y permitirá que el usuario pueda navegar entre ellas.

Interfaces internas

Las interfaces internas son aquellas que solo puede visualizar el usuario que se ha registrado y consta en la base de datos de la aplicación. Cuenta con varias interfaces, las cuales son:

1. Interfaz principal: Muestra los datos del centro de diagnóstico y tratamiento psicopedagógico de la especialista.
2. Interfaz de perfil: Muestra los datos del usuario registrado, esta interfaz cuenta con un botón para poder editarlos.
3. Interfaz de fichas: Esta cuenta con un buscador, un botón para poder agregar a los evaluados y el acceso a una interfaz para ver a los evaluados registrados
4. Interfaz de evaluados: En esta interfaz el usuario es capaz de ver los datos que registro sobre el evaluado, en ella se encuentran los botones editar y eliminar datos.
5. Interfaz para agregar a un evaluado: Esta interfaz le sirve usuario para agregar los datos del evaluado y cuenta con los botones de guardar y cerrar.
6. Interfaz de ejercicios: Esta consiste en un buscador para encontrar el tipo de ejercicio que el usuario desee realizar y si aplasta en uno de ellos le lleva a la interfaz de actividades
7. Interfaz de actividades: En esta interfaz se muestra la explicación del tipo de ejercicios que el usuario puede escoger.
8. Interfaz de menú: La interfaz de menú consistirá en mostrar todos los títulos de las interfaces internas y permitirá que el usuario pueda navegar entre ellas.

Interfaces de hardware

Será necesario disponer de las siguientes características:

- Memoria RAM: Mínimo 3GB
- Almacenamiento: Mínimo 8GB
- Teléfono con una pantalla que funcione correctamente.

Interfaces de software

Será necesario que se disponga el siguiente requerimiento:

- Sistema operativo: Android o superior o iOS 8 o superior

Interfaces de comunicación

El software no requiere de interfaces de comunicación adicionales.

4.6. Diseño del software

En la etapa del diseño del programa se definen las acciones que deben implementarse tanto en el software como en el hardware, en otras palabras, esta es la etapa dónde se concreta la estructura que tendrá el programa al igual que las interacciones con el usuario (Sommerville, 2005). A continuación, se mostrará los diagramas de casos de uso que se utilizaron para poder definir el diseño del sistema.

Diagramas de casos de uso

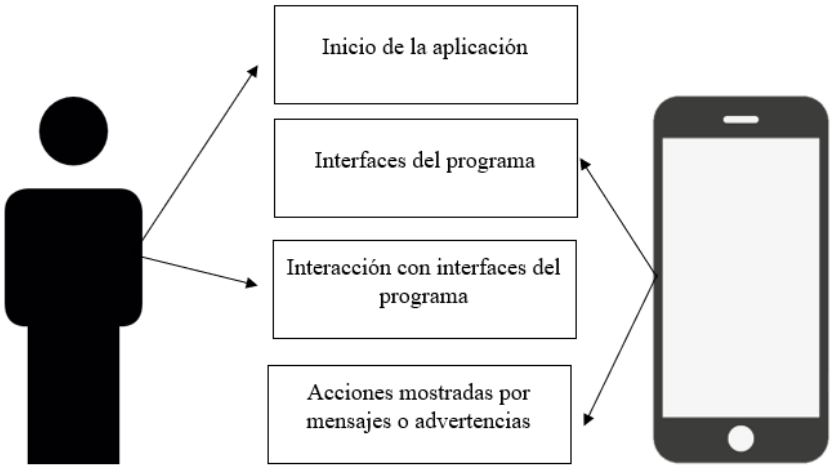
Actores: Usuario y Sistema

Flujo principal

1. El usuario inicia la aplicación
2. El sistema muestra las interfaces del programa
3. El usuario interactúa con las interfaces del programa

4. El sistema muestra las acciones del usuario por medio de mensajes o advertencias

Gráfico 5 *Diagrama de caso de uso del flujo principal*



Requerimiento Funcional (RF1) – Inicio de sesión y registro

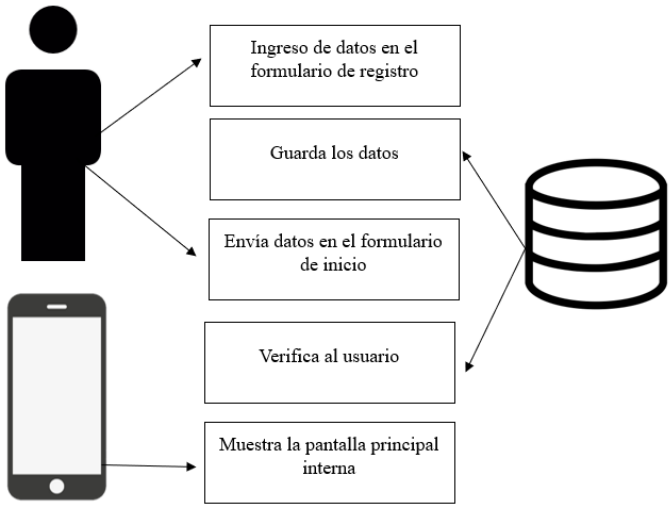
Descripción: El sistema debe permitir al usuario registrar sus datos y acceder a la aplicación con un usuario y contraseña

Actores: Usuario, Sistema y Base de Datos

Flujo general del RF1:

1. El sistema muestra la pantalla inicio y registro
2. El usuario ingresa los datos en el formulario de registro
3. El sistema se comunica con la base de datos
4. La base de datos agrega la información registrada
5. El usuario ingresa sus datos en la pantalla de inicio
6. La base de datos revisa que el usuario se encuentre en el sistema
7. El sistema le muestra la pantalla principal interna al usuario

Gráfico 6 *Diagrama de caso de uso del requerimiento funcional 1*



Requerimiento Funcional (RF2) – Desarrollo de interfaces externas

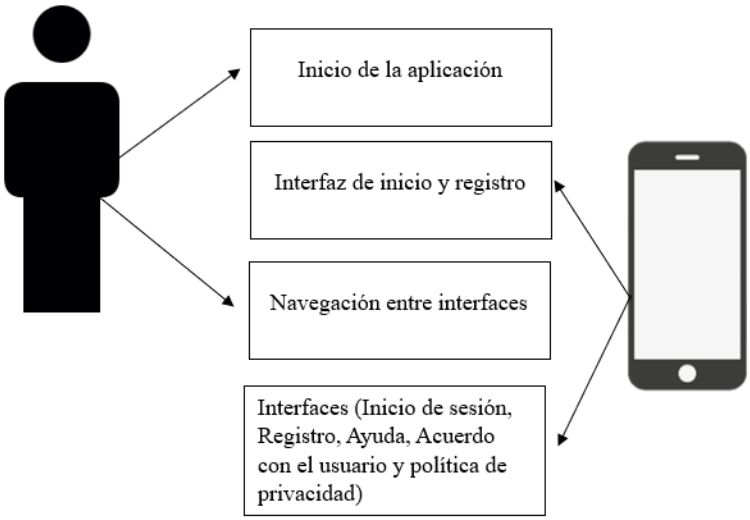
Descripción: El sistema debe contar con cinco pantallas externas (Inicio de sesión, Registro, Ayuda, Acuerdo con el usuario y política de privacidad).

Actores: Sistema y usuario

Flujo general del RF2:

1. El usuario inicia la aplicación
2. El sistema muestra la interfaz de inicio y registro
3. El usuario navega y escoge otra interfaz
4. El sistema muestra la interfaz (Inicio de sesión, Registro, Ayuda, Acuerdo con el usuario y política de privacidad)

Gráfico 7 *Diagrama de caso de uso del requerimiento funcional 2*



Requerimiento Funcional (RF3) – Desarrollo de interfaces internas

Descripción El sistema debe contar con 4 pantallas internas (Perfil, Fichas, Inicio y Ejercicios).

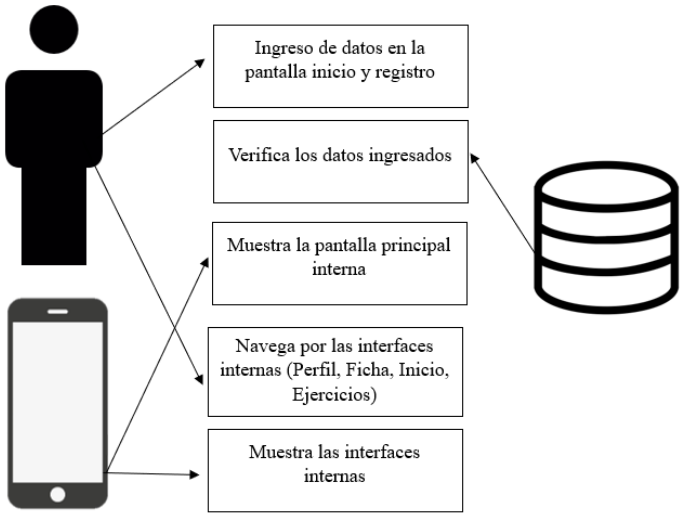
Actores: Usuario, sistema y base de datos

Flujo general del RF3:

1. Usuario ingresa sus datos en la interfaz inicio y registro
2. La base de datos verifica los datos del usuario
3. Sistema muestra la pantalla principal interna
4. El usuario navega entre las interfaces internas
5. El sistema muestra las interfaces internas

Excepción: Si el usuario no está registrado en la base de datos entonces no podrá navegar por las interfaces internas.

Gráfico 8 *Diagrama de caso de uso del requerimiento funcional 3*



Requerimiento Funcional (RF4) – Navegación entre interfaces

Descripción: El sistema debe contar con botones que permitan la correcta navegación entre todas las interfaces

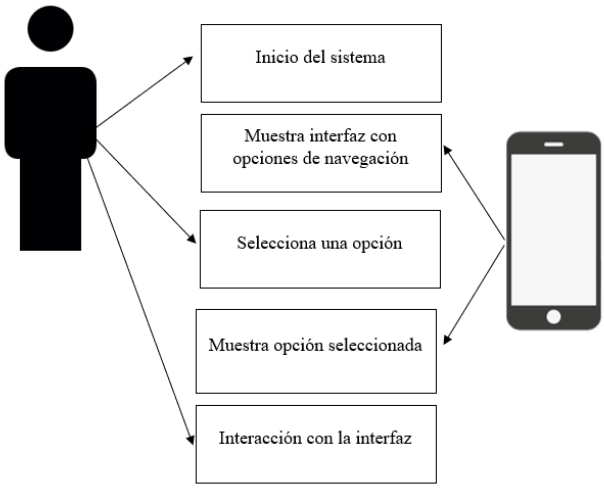
Actores: Usuario y Sistema

Flujo general del RF4:

1. El usuario inicia el sistema
2. El sistema muestra una interfaz con opciones de navegación
3. El usuario selecciona una opción para navegar
4. El sistema muestra la interfaz seleccionada
5. El usuario interactúa con la interfaz

Excepción 1: Si el usuario no se ha registrado o ha iniciado sesión solo interactúa con las interfaces externas.

Gráfico 9 *Diagrama de caso de uso del requerimiento funcional 4*



Requerimiento Funcional (RF5) – Gestión de datos

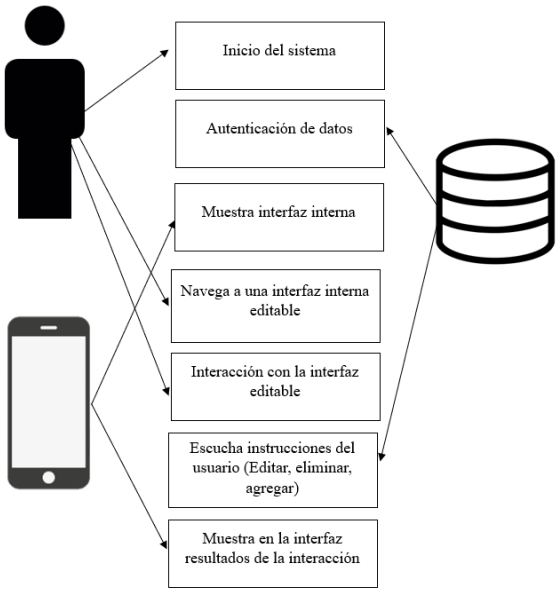
Descripción: El sistema debe tener conexión con una base de datos para manejar la edición de los mismos en las interfaces que lo requieran

Actores: Usuario, Sistema y Base de Datos

Flujo general del RF5:

1. El usuario inicia sesión en el sistema
2. La base de datos verifica y autentica los datos
3. El sistema muestra la pantalla principal interna
4. El usuario navega a una interfaz editable como Perfil y Fichas
5. El usuario interactúa (agrega, elimina, edita) con la interfaz
6. La base de datos escucha la instrucción del usuario
7. El sistema muestra los resultados de la interacción

Gráfico 10 *Diagrama de caso de uso del requerimiento funcional 5*



Requerimiento Funcional (RF6) – Cambios en tiempo real

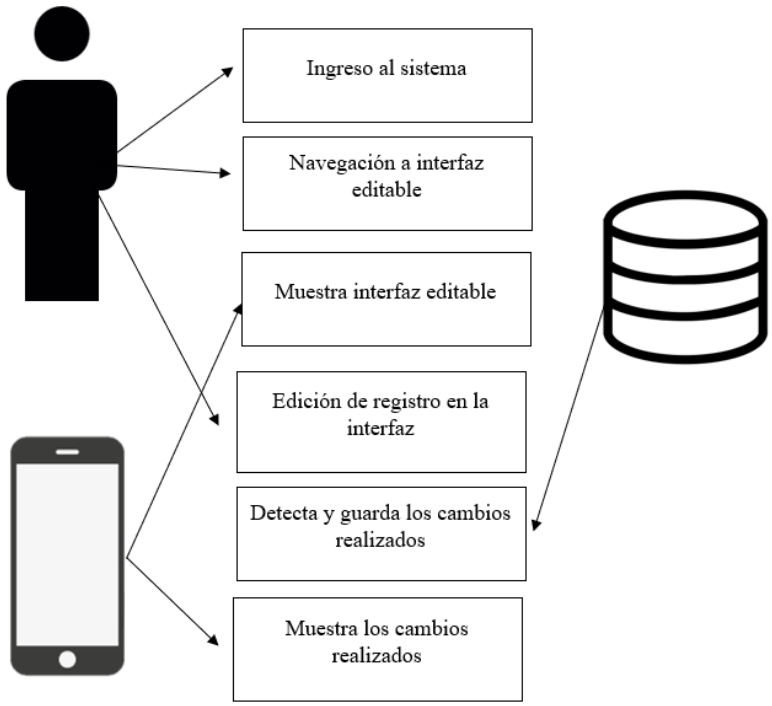
Descripción: El usuario debe ser capaz de ver los cambios que realiza el sistema tiempo real

Actores: Usuario, Sistema y Base de Datos

Flujo general del RF6:

1. El usuario ingresa al sistema
2. El usuario navega a una interfaz editable
3. El sistema muestra la interfaz
4. El usuario edita un registro de la interfaz
5. La base de datos detecta y guarda los cambios
6. El sistema muestra los cambios realizados

Gráfico 11 *Diagrama de caso de uso del requerimiento funcional 6*



Requerimiento Funcional (RF7) – Advertencias y mensajes de error

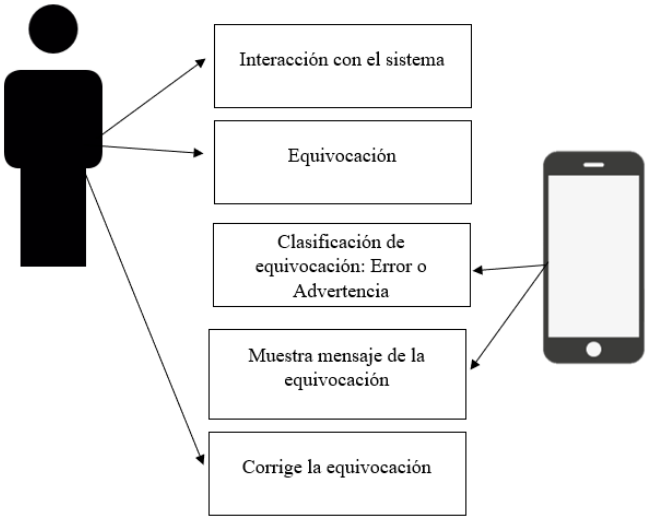
Descripción: El usuario debe ser capaz de ver los errores o advertencias que devuelve el sistema.

Actores: Usuario y Sistema

Flujo general del RF7:

1. El usuario interactúa con el sistema
2. El usuario comete una equivocación
3. El sistema clasifica a la equivocación como error o advertencia
4. El sistema muestra un mensaje dependiendo del tipo de equivocación
5. El usuario corrige su equivocación

Gráfico 12 *Diagrama de caso de uso del requerimiento funcional 7*



Requerimiento Funcional (RF8) – Funcionalidad de botones

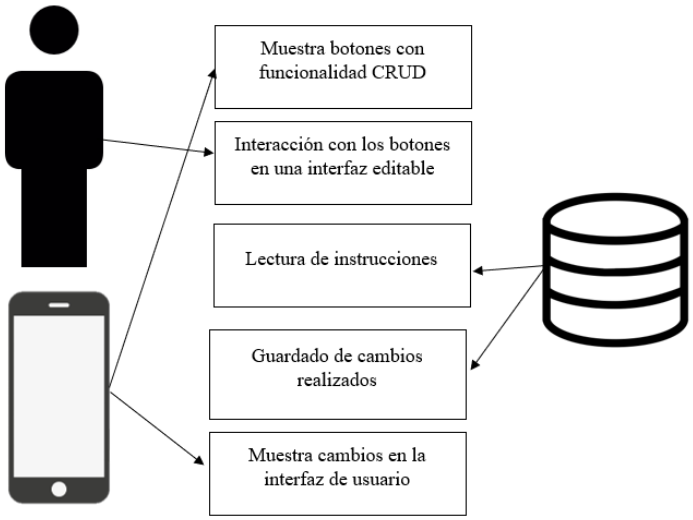
Descripción: Varios de los botones tendrán las funciones de un CRUD por lo que el sistema debe funcionar correctamente

Actores: Usuario, Sistema y Base de Datos

Flujo general del RF8:

1. El sistema muestra botones con funcionalidades de CRUD
2. El usuario interactúa con el botones agregar, editar, borrar en una interfaz editable
3. La base de datos lee las instrucciones del usuario
4. La base de datos guarda los cambios realizados
5. El sistema muestra los cambios en la interfaz del sistema

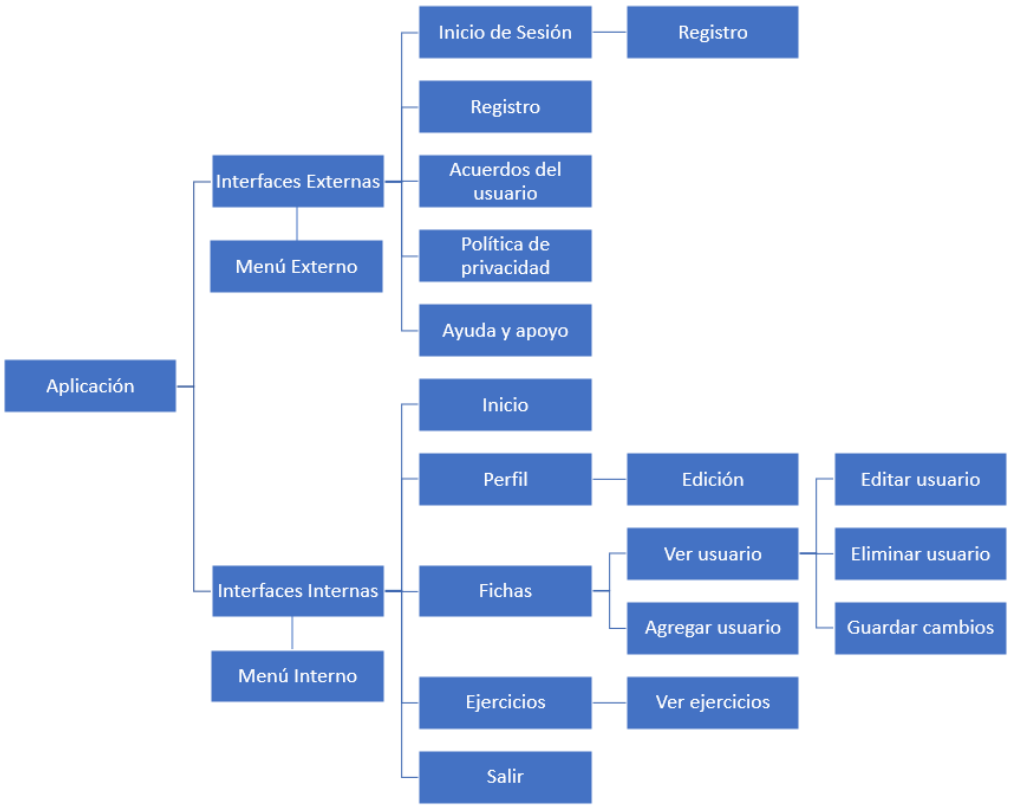
Gráfico 13 *Diagrama de caso de uso del requerimiento funcional 8*



Diseño arquitectónico del sistema

Para poder desarrollar la aplicación de este proyecto de titulación se utilizó la siguiente arquitectura de repositorio.

Gráfico 14 *Diagrama arquitectónico del sistema desarrollado*



4.7.Prototipo de Baja Fidelidad

El prototipo de baja fidelidad representa la visión inicial de cómo será realizado el software, se lo puede desarrollar de varias formas, en este caso, se eligió dibujar a mano el prototipo, en cada una de las imágenes se podrá observar el detalle de cada interfaz, los botones que tendrá, dónde están ubicados, entre otros requerimientos del usuario.

Interfaces externas

Interfaz principal

En esta interfaz se observa dónde el usuario colocará su nombre de usuario, contraseña y tiene las opciones del botón de ingresar a la aplicación o el botón de registrarse en la misma.

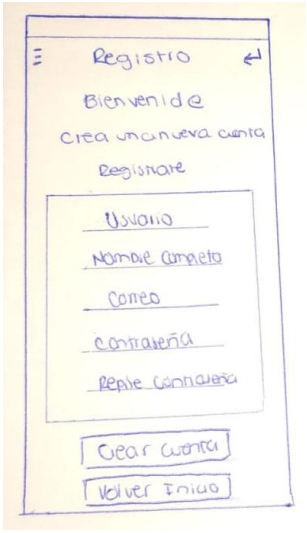
Gráfico 15 *Prototipo de baja fidelidad interfaz principal externa*



Interfaz de Registro

En esta interfaz el usuario será capaz de agregar su información para registrarse en la aplicación. Esta sección obtendrá el usuario, nombre completo del usuario que quiere registrarse, correo, contraseña y comprobar la contraseña antes escrita.

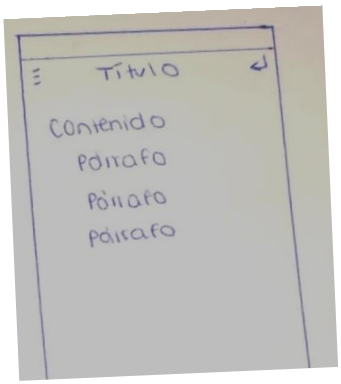
Gráfico 16 *Prototipo de baja fidelidad interfaz registro de usuario*



Interfaces informativas

Estas interfaces muestran información importante sobre la aplicación, consiste en la interfaz “Acuerdo del usuario”, la interfaz “Política de privacidad” y la interfaz “Ayuda y apoyo”, todas contarán con la misma estructura del prototipo.

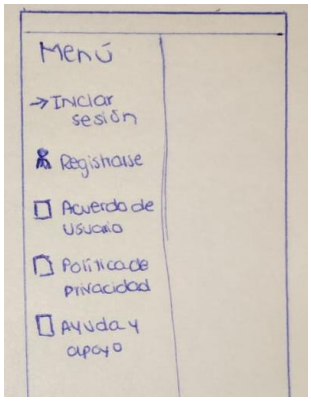
Gráfico 17 *Prototipo de baja fidelidad interfaces informativas*



Interfaz menú externo

La interfaz de menú muestra todos los títulos de las interfaces externas y permitirá que el usuario pueda navegar entre ellas sin ningún problema.

Gráfico 18 *Prototipo de baja fidelidad interfaz menú externo*



Interfaces internas

Interfaz principal

Para esta interfaz se tomará en cuenta la información del centro de diagnóstico y tratamiento psicopedagógico de la especialista. Además, se tomará en cuenta la atención y ubicación del centro.

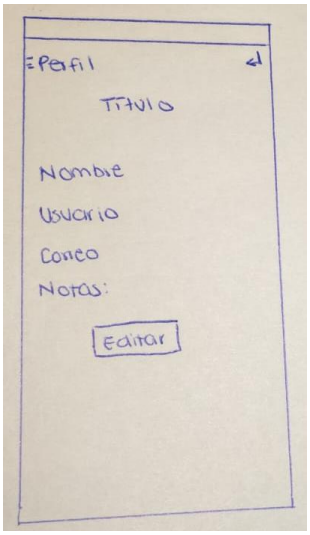
Gráfico 19 *Prototipo de baja fidelidad interfaz menú principal interno*



Interfaz de perfil

El perfil consta de la información que el usuario registro al momento de crear una nueva cuenta en la aplicación, además, cuenta con un botón de editar para corregir errores o agregar notas

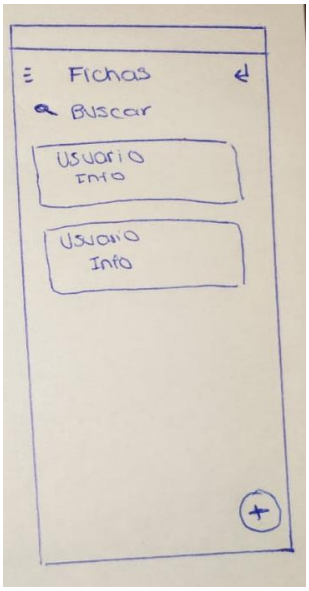
Gráfico 20 *Prototipo de baja fidelidad interfaz de perfil*



Interfaz de fichas

La interfaz de las fichas consiste en poder visualizar todas las fichas que han sido agregadas por el evaluador, cuenta con un botón de agregar para ingresar los datos de otros niños que requieran ser evaluados. Adicionalmente, esta interfaz tiene un pequeño buscador, este ayudará a la búsqueda de cualquier evaluado en el caso de tener muchos niños ingresados.

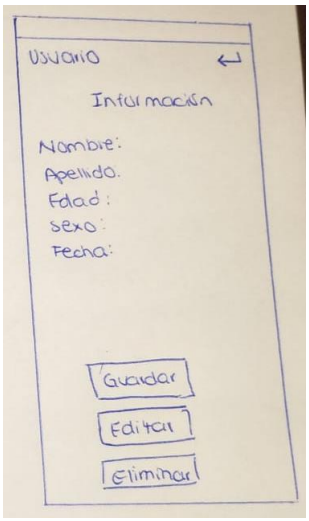
Gráfico 21 *Prototipo de baja fidelidad interfaz de fichas*



Interfaz de evaluados

Cuando se selecciona a un evaluado es posible visualizar, editar y eliminar toda la información que se ingresó al momento de crear un nuevo paciente.

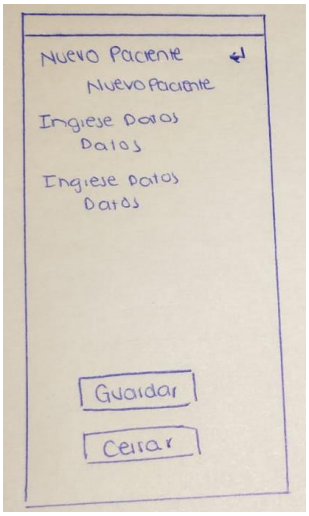
Gráfico 22 *Prototipo de baja fidelidad interfaz de evaluados*



Interfaz para agregar a un evaluado

Al momento de colocar una nueva ficha se ingresa a una nueva interfaz la cual, nos dejará ingresar todos los datos del niño como su nombre, apellido, edad, fecha de ingreso al consultorio, sexo, información acerca de los padres, entre otros. Contará con un botón de guardar para poder almacenar todos los datos dentro de la base de datos y cerrar en el caso de no querer agregar un nuevo paciente

Gráfico 23 *Prototipo de baja fidelidad interfaz para agregar a un evaluado*



Interfaz de ejercicios

Esta interfaz es simple de utilizar debido a que aquí se encontraran los tipos de ejercicios que se pueden aplicar a los niños evaluados, estos tipos pueden ser: lenguaje receptivo, lenguaje expresivo, coordinación corporal, entre otros.

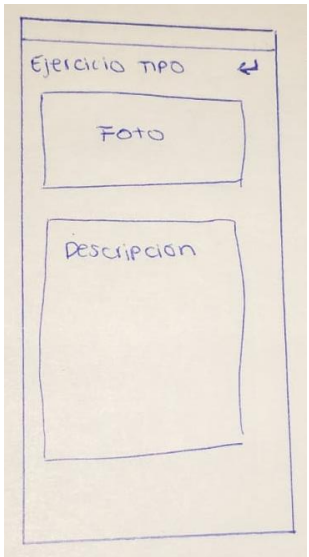
Gráfico 24 *Prototipo de baja fidelidad interfaz de ejercicios*



Interfaz de actividades

Una vez elegido el tipo de ejercicio que se va a realizar, se abrirá una nueva interfaz que mostrará los detalles de ese tipo de ejercicio, lo que se debe realizar y cómo se debe realizar.

Gráfico 25 *Prototipo de baja fidelidad interfaz de actividades*



Interfaz menú interno

La interfaz de menú muestra todos los títulos de las interfaces internas y permitirá que el usuario pueda navegar entre ellas sin ningún problema.

Gráfico 26 *Prototipo de baja fidelidad interfaz de menú interno*



4.8.Prototipo de Alta Fidelidad

El prototipo de alta fidelidad representa una visión más clara que el prototipo de baja fidelidad, en ella el usuario podrá tener una interacción más real con la aplicación y verificar si cumple con todos los requerimientos solicitados o si desea modificar algo. El prototipo de alta fidelidad se realizó en la herramienta Mockittapp, una herramienta para poder realizar prototipos de aplicaciones. A continuación, se mostrarán capturas de pantalla del prototipo de alta fidelidad que se realizó.

Interfaces Externas

Interfaz principal

En esta interfaz se observa dónde el usuario colocará su nombre de usuario, contraseña y tiene las opciones del botón de ingresar a la aplicación o el botón de registrarse en la misma.

Gráfico 27 *Prototipo de alta fidelidad interfaz principal*



Interfaz de Registro

En esta interfaz el usuario será capaz de agregar su información para registrarse en la aplicación. Esta sección obtendrá el usuario, nombre completo del usuario que quiere registrarse, correo, contraseña y comprobar la contraseña antes escrita.

Gráfico 28 *Prototipo de alta fidelidad interfaz de registro*



Interfaces informativas

Estas interfaces muestran información importante sobre la aplicación, consiste en la interfaz “Acuerdo del usuario”, la interfaz “Política de privacidad” y la interfaz “Ayuda y apoyo”, todas contarán con la misma estructura del prototipo.

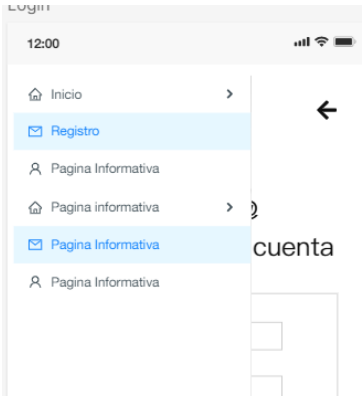
Gráfico 29 *Prototipo de alta fidelidad interfaces informativas*



Interfaz menú externo

La interfaz de menú muestra todos los títulos de las interfaces externas y permitirá que el usuario pueda navegar entre ellas sin ningún problema.

Gráfico 30 *Prototipo de alta fidelidad menú externo*

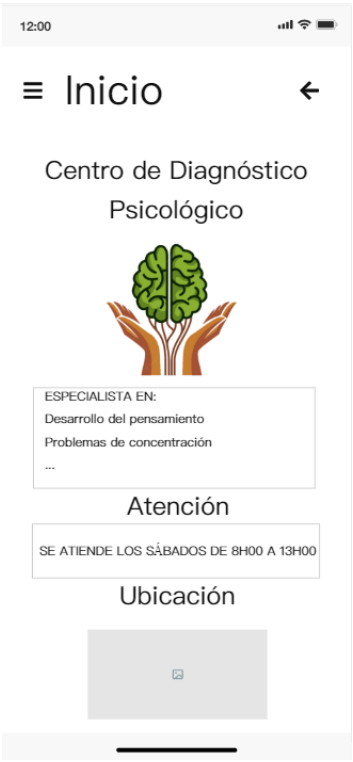


Interfaces externas

Interfaz principal

Para esta interfaz se tomará en cuenta la información del centro de diagnóstico y tratamiento psicopedagógico de la especialista. Además, se tomará en cuenta la atención y ubicación del centro.

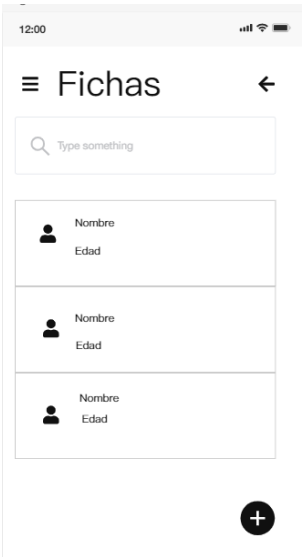
Gráfico 31 *Prototipo de alta fidelidad interfaz principal*



Interfaz de fichas

La interfaz de las fichas consiste en poder visualizar todas las fichas que han sido agregadas por el evaluador, cuenta con un botón de agregar para ingresar los datos de otros niños que requieran ser evaluados. Adicionalmente, esta interfaz tiene un pequeño buscador, este ayudará a la búsqueda de cualquier evaluado en el caso de tener muchos niños ingresados.

Gráfico 32 *Prototipo de alta fidelidad interfaz fichas*



Interfaz de evaluados

Cuando se selecciona a un evaluado es posible visualizar, editar y eliminar toda la información que se ingresó al momento de crear un nuevo paciente.

Gráfico 33 *Prototipo de alta fidelidad interfaz de evaluados*



Interfaz para agregar a un evaluado

Al momento de colocar una nueva ficha se ingresa a una nueva interfaz la cual, nos dejará ingresar todos los datos del niño como su nombre, apellido, edad, fecha de ingreso al consultorio, sexo, información acerca de los padres, entre otros. Contará con un botón de guardar para poder almacenar todos los datos dentro de la base de datos y cerrar en el caso de no querer agregar un nuevo paciente.

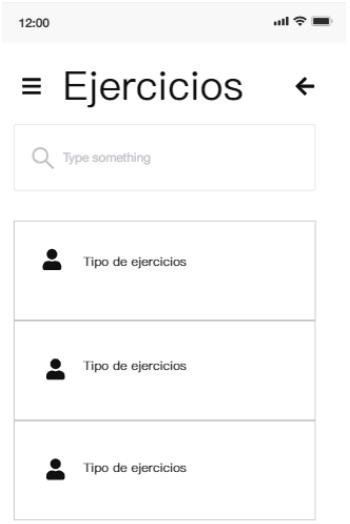
Gráfico 34 *Prototipo de alta fidelidad interfaz para ingresar a un evaluado*



Interfaz de ejercicios

Esta interfaz es simple de utilizar debido a que aquí se encontraran los tipos de ejercicios que se pueden aplicar a los niños evaluados, estos tipos pueden ser: lenguaje receptivo, lenguaje expresivo, coordinación corporal, entre otros.

Gráfico 35 *Prototipo de alta fidelidad interfaz de ejercicios*



Interfaz de actividades

Una vez elegido el tipo de ejercicio que se va a realizar, se abrirá una nueva interfaz que mostrará los detalles de ese tipo de ejercicio, lo que se debe realizar y cómo se debe realizar.

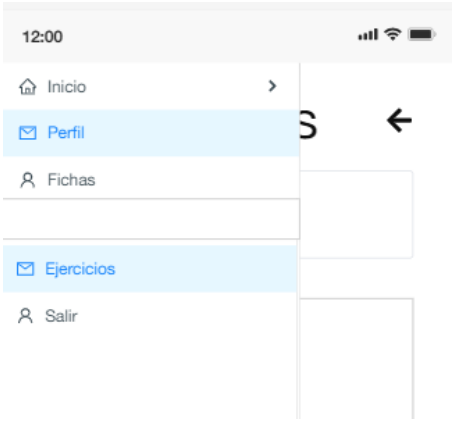
Gráfico 36 *Prototipo de alta fidelidad interfaz de actividades*



Interfaz menú interno

La interfaz de menú muestra todos los títulos de las interfaces internas y permitirá que el usuario pueda navegar entre ellas sin ningún problema.

Gráfico 37 *Prototipo de alta fidelidad interfaz menú interno*



Interfaz de perfil

El perfil consta de la información que el usuario registro al momento de crear una nueva cuenta en la aplicación, además, cuenta con un botón de editar para corregir errores o agregar notas

Gráfico 38 *Prototipo de alta fidelidad interfaz de perfil*



4.9.Definición de los ejercicios de estimulación del pensamiento

Después de haber realizado una entrevista con la especialista en el manejo de dificultades del aprendizaje, se determinó los ejercicios más importantes y más comunes para los niños que se encuentran en la edad de 2 a 4 años. Algunos de los que se platicó en la entrevista son los siguientes:

1. Ejercicios de autoconcepto
    a. Mostrar expresiones de propiedad y posesión
    b. Utilizar pronombres o nombres para referirse a sí mismo
2. Ejercicios de atención
    a. Mire y señale un dibujo
    b. Preste atención a un objeto durante 3 segundos o más
3. Ejercicios de coordinación corporal
    a. Patear una pelota
    b. Avanzar dos o más pasos en una línea
    c. Mantenerse sobre un pie
4. Ejercicios de locomoción
    a. Correr una distancia de 3 metros sin caerse
    b. Saltar con los pies juntos
    c. Bajar escaleras alternando los pies
5. Ejercicios de motricidad fina
    a. Copiar una línea vertical, un círculo y una cruz
    b. Dibujar una persona con seis elementos
6. Ejercicios de lenguaje receptivo
    a. Hacer preguntas para que el niño responda sí o no adecuadamente
    b. Utilizar frases de tres palabras

7. Ejercicios de discriminación perceptiva
    a. Emparejar formas geométricas sencillas
    b. Emparejar circulo, cuadrado y triángulo
8. Ejercicios de memoria
    a. Repetir secuencias de dos dígitos
    b. Elegir la mano en la que se esconde un juguete
    c. Recordar objetos familiares
9. Ejercicios de desarrollo conceptual
    a. Identificar tamaños grande y pequeño
    b. Identificar objetos familiares por su uso.

CAPÍTULO V: IMPLEMENTACIÓN

5. Implementación de la aplicación

El proceso de la implementación de la aplicación consiste en la instalación, configuración y actualización del software que se está desarrollando, en esta sección se demostrará el desarrollo del software y todas las fases que se realizaron para llegar al producto fina (VmWare, 2023). Para poder observar el código de la aplicación se debe acceder a los repositorios de Github, por medio de la siguiente URL: https://github.com/Mggomezy/centro-ninos.git

5.1. Selección de herramientas para el desarrollo de la aplicación

Visual Studio Code

Esta herramienta es un editor de texto de código abierto desarrollado por Microsoft, cuenta con varias herramientas que facilitan el desarrollo de software en varios lenguajes de programación (Flores, 2022). En esta ocasión se utilizará el framework ionic para desarrollar la aplicación, los lenguajes de programación utilizados son HTML, SCSS, JavaScript y TyperScript.

Node.js

Node.js es un ambiente de ejecución de JavaScript de código abierto que permite a los usuarios crear aplicaciones escalables de servidor y red, proporciona un modelo E/S sin bloqueo lo que permite que todas las aplicaciones desarrolladas sean escalables y puedan manejar una gran cantidad de conexiones al mismo tiempo (Vargas, 2023). Esta aplicación permitirá instalar ionic y todos los paquetes necesarios para poder ejecutarlo. Para poder instalarlo hay que ingresar a su página oficinal (https://nodejs.org/es/download) y seguir los pasos de instalación de la aplicación.

Ionic

Como se mencionó anteriormente Ionic es un framework de código abierto que se lo utiliza para desarrollar aplicaciones móviles híbridas multiplataforma. Para poder instalar la aplicación en la computadora, se deben seguir los pasos de la página oficial de ionic que es la siguiente: https://ionicframework.com/

Microsoft SQL Server Management Studio

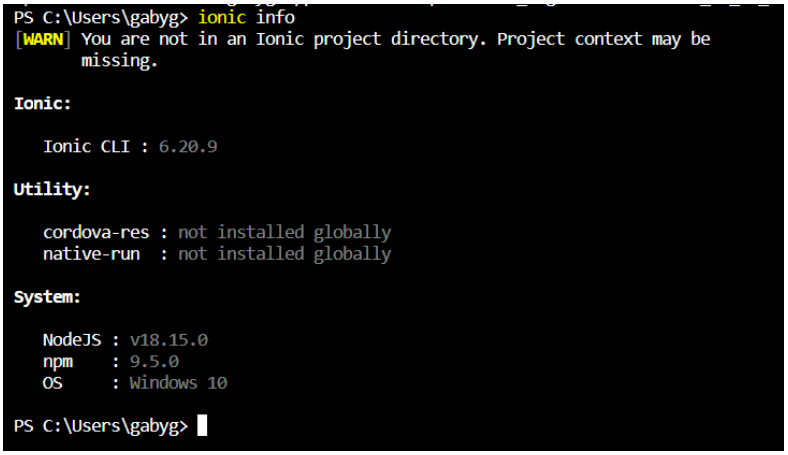
Para el desarrollo de la base de datos de este proyecto de titulación se utilizó la SQL Server 2014 Management Studio, a este se lo define como un entorno integrado que tiene la capacidad de administrar todo tipo de infraestructura de SQL, con esta herramienta se logra configurar, gestionar y administrar los componentes de la base de datos (Microsoft, 2023).

5.2. Construcción del software

Para iniciar con la construcción del software, primero hay que asegurarse de tener instalado correctamente node.js debido a que este permitirá la ejecución de algunos elementos que utiliza ionic en el desarrollo de aplicaciones móviles híbridas. A continuación, dentro del editor de código Visual Studio Code que se instaló previamente, es necesario abrir una nueva terminal para ejecutar el comando **npm install -g @ionic/cli** e instalar Ionic.

Cuando el comando haya terminado de ejecutarse es necesario verificar que se tenga la versión más actualizada de ionic en el sistema, para ello se debe ejecutar el comando **ionic info** que permite visualizar la versión instalada de ionic.

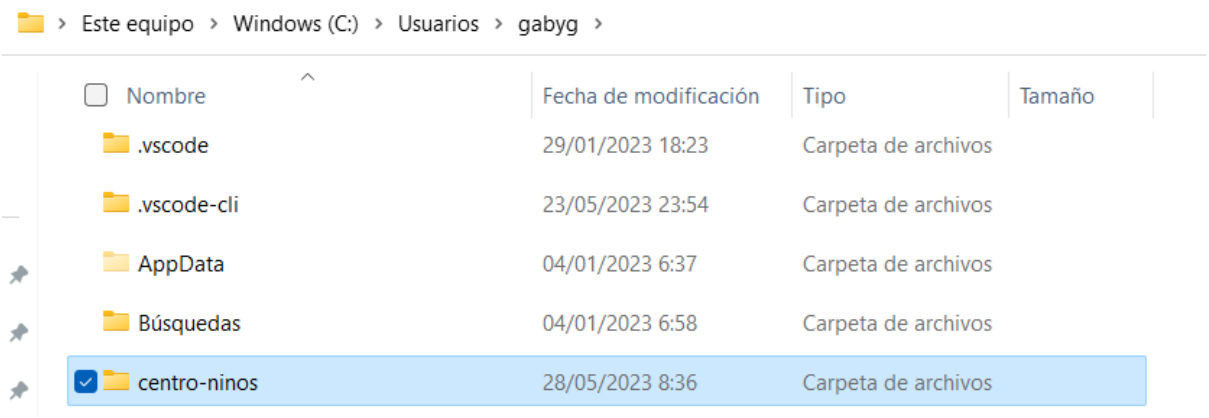
Gráfico 39 *Información de la instalación de Ionic*



Creación del entorno de desarrollo de la aplicación

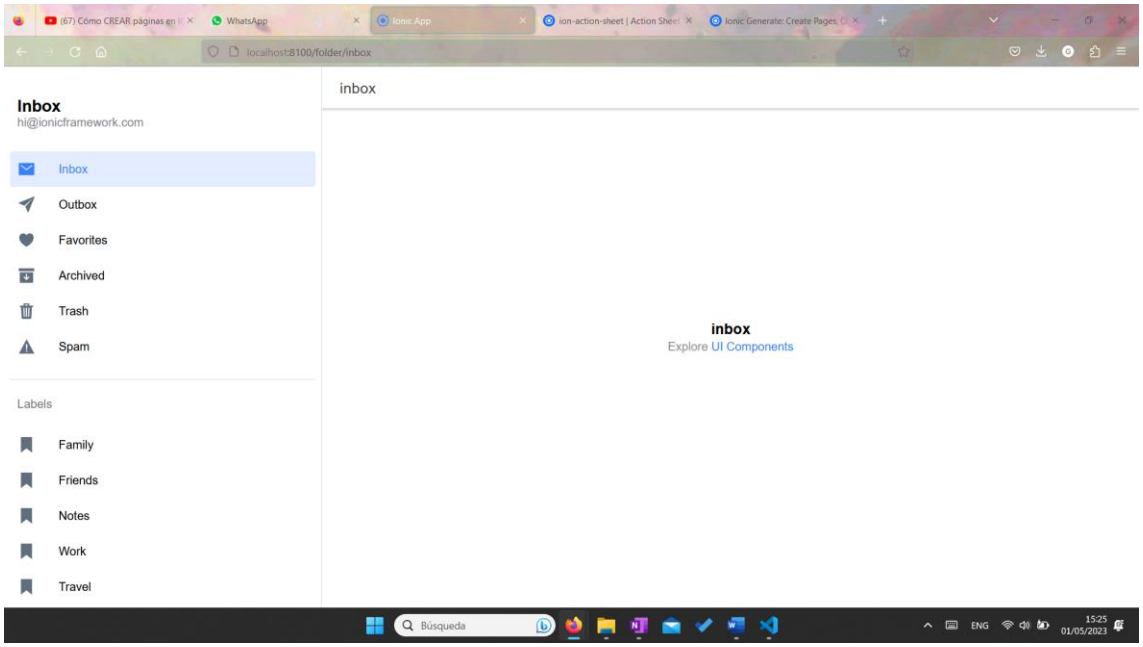
Para la creación del entorno de desarrollo de la aplicación primero se debe crear una carpeta con el nombre que tendrá el proyecto en el lugar de la computadora que se desee almacenar, en este caso el proyecto se llama centro-ninos.

Gráfico 40 *Creación de carpeta del proyecto*



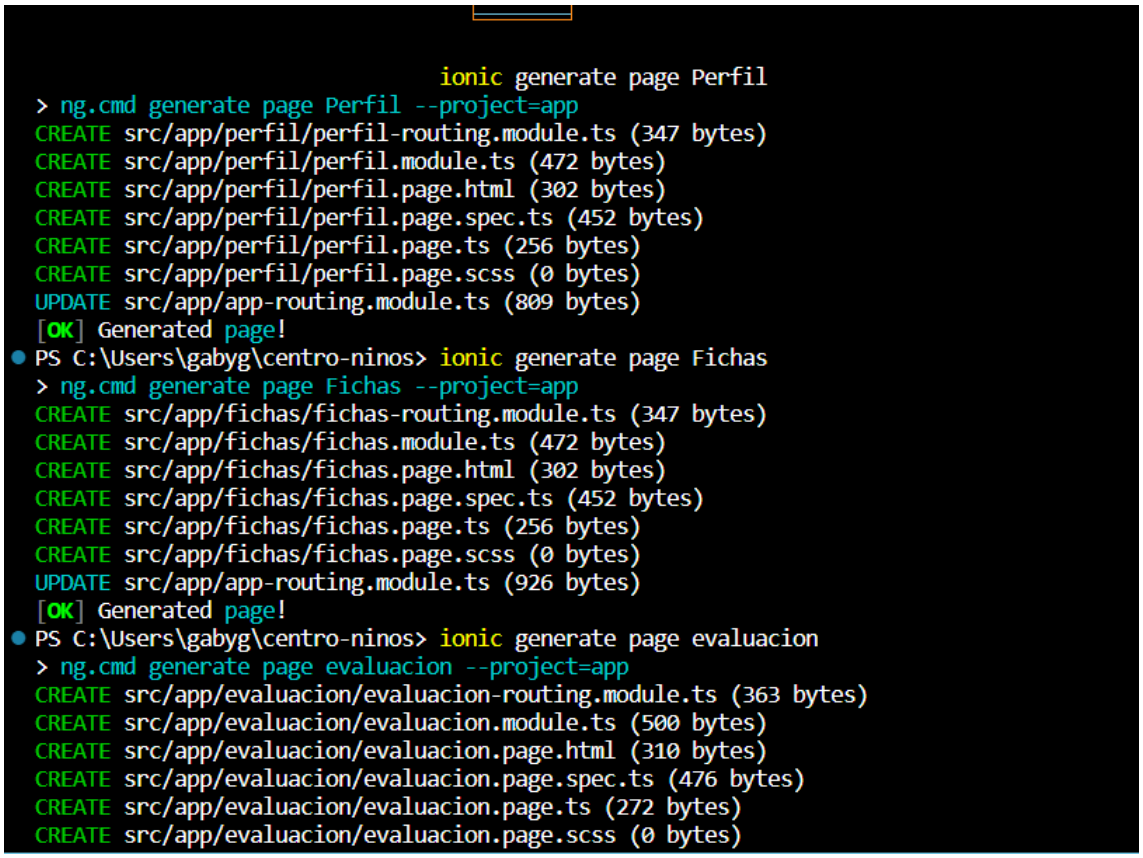
Una vez creada la carpeta es necesario dirigirse al editor de código Visual Studio, dentro de este programa, se debe arrastrar la carpeta centro-ninos y el usuario debe abrir una terminal para ejecutar el comando **ionic start centro-ninos sidemenu.** Este comando permitirá crear una plantilla de menú deslizante hacia la derecha en Ionic, contará con una estructura y base funcional para poder iniciar con el desarrollo de una aplicación que tenga el menú deslizante. A continuación, se muestra la plantilla con la que se inicio a desarrollar esta aplicación.

Gráfico 41 *Plantilla de ionic con menú deslizante*



Al crear la plantilla del proyecto se genera una serie de archivos y directorios fundamentales para el funcionamiento de la aplicación, sin embargo, no todos son importantes ni se utilizarán para el desarrollo de la misma en este proyecto de titulación, es por esta razón, que se eliminan todas las páginas de la plantilla creada y se utiliza el comando **ionic generate page Nombredelapagina** para generar las páginas planeadas para la creación de la aplicación. En esta ocasión las páginas que se crearon y que serán descritas a detalle más adelante son: actividades, acuerdo, ayuda, ejercicios, evaluados, fichas, home, login, nuevo, perfil, privacidad y registro.

Gráfico 42 *Ejemplo de la creación de páginas en ionic por medio del comando ionic generate page Nombredepágina*



A continuación, se detalla el diseño, código y funcionamiento de cada interfaz para la creación de la aplicación, tomando en cuenta los requisitos funcionales y no funcionales que habían sido identificados anteriormente luego de las entrevistas y reuniones que se tuvo con el usuario. Hay que tener en cuenta que para crear cada página se utilizan tres archivos principales: el archivo .html que se lo utiliza para la estructura y diseño de la página, el archivo .ts para la lógica y funcionamiento de la misma y el archivo. scss para desarrollar los estilos que tendrá cada una. Para una mejor organización se explicará la aplicación dividiéndola en interfaces externas e interfaces internas y cada sección de cada interfaz se explicará con su código HTML, TS y SCSS.

Desarrollo de las interfaces externas

Todas las interfaces externas cuentan con una barra de herramientas que se encuentra en la parte superior de cada interfaz, utiliza la misma estructura HTML y SCSS que será mostrada a continuación, la única diferencia que se encontrará en ellas es el nombre del título y el botón en la parte superior izquierda que redirecciona a las páginas actuales a la página de /login.

Gráfico 43 *Estructura y diseño barra de herramientas de las interfaces externas*

CÓDIGO

.HTML

```
<ion-header [translucent]="true">
  <ion-toolbar color="mycolor"> <!--Color de la barra de herramientas-->
    <!--Titulo de la pagina-->
    <ion-title>Bienvenid@s</ion-title>
    <!--Llama a la funcion para que el menu funcione-->
    <ion-buttons slot="start"> <!--Se coloca en la parte izquierda de la barra de herramientas-->
      <ion-button id="" (click)="openMenu()">
        <ion-icon name="menu"></ion-icon>
      </ion-button>
    </ion-buttons>
  </ion-toolbar>
</ion-header>
```

.SCSS

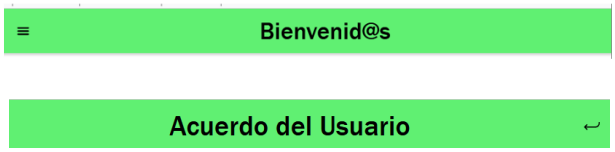
login.page.html | login.page.scss

centro-ninos > src > app > login > login.page.scss > h1

```
ion-toolbar{
    background-color: var(--ion-color-mycolor);
}

ion-title{
    text-align: center;
    font-size: 30px;
    font-family: 'Franklin Gothic Medium', 'Arial Narrow', Arial, sans-serif;
}
```

RESULTADO



Página login

Como se ha ido mencionado la página de login permitirá a la persona interesada ingresar con un usuario y contraseña al sistema. Para iniciar, la estructura de la interfaz cuenta con una barra de herramientas que fue explicada anteriormente, a continuación, cuenta con un título, una imagen, el espacio para que el cliente ingrese su usuario y contraseña y los botones que permitirán realizar todas las funciones. El siguiente gráfico muestra solamente la estructura de la página en html y su diseño en scss.

Gráfico 44 *Estructura y diseño de página login*

CÓDIGO

.HTML

```
<!--Contenido-->
<ion-content class="contenido">
  <h1>Consultorio psicológico</h1>
<!--Imagen del contenido-->
  <div class="contenedor_imagen">
    <img src="../../assets/imagenes evaluados/consultorio.jpg.png" width="250px" height="250px">
  </div>
<!--Parrafo del contenido-->
  <h3 class="slogan">La aplicación ideal para psicólogos, fortaleciendo el bienestar infantil</h3>

<!--Espacio para ingreso de datos del usuario y su contrasena-->
  <div class="info_sesion">
    <ion-input type="text" placeholder="Usuario" [(ngModel)]="usuario"></ion-input>
    <ion-input type="password" placeholder="Contraseña" [(ngModel)]="contrasena"></ion-input>
  </div>

 <!--Boton para iniciar sesion-->
  <ion-button class="btn" (click)="login()">
    Ingresar
  </ion-button>

 <h3 class="Pregunta">¿No tienes una cuenta?</h3>

 <!--Boton para registrar usuario-->
 <ion-button class="btn" (click)="registro()">
   Registrarse
 </ion-button>
</ion-content>
```

.SCSS

```
h1{
    font-family:'Franklin Gothic Medium', 'Arial Narrow', Arial, sans-serif;
    text-align: center;
}

.Pregunta{
    text-align: center;
    font-size: 16px;
    font-family:Arial, Helvetica, sans-serif;
}

.contenedor_imagen{
    display: flex;
    justify-content: center;
    align-items: center;
}
.slogan{
    text-align: center;
    font-size: 15px;
    font-family:'Franklin Gothic Medium', 'Arial Narrow', Arial, sans-serif;
}

ion-button.btn{
    border-radius: 10px;
    margin-top: 5px;
    text-align: center;
    //align-items: center;
    display: flex;
    justify-content: center;
}

.info_sesion{
    justify-content: center;
    align-items: center;
}

ion-input{
    font-family:'Franklin Gothic Medium', 'Arial Narrow', Arial, sans-serif;
    text-align: center;
    color: black;
}
```

RESULTADO



Para la funcionalidad de cada uno de los botones de la página login, se utilizó el archivo login.ts. Para el botón INGRESAR es necesaria la declaración async que define a una función como asincrónica, esto quiere decir, que la tarea que realice se ejecutará en un segundo plano. En esta ocasión la tarea que debe realizar es verificar que el usuario que la persona ingresa existe en la base de datos, esto lo realiza con la con la función “verificarUsuarioExistente”, es importante mencionar, que esta función está siendo llamada del servicio “miServerService”, el cual explicaremos más adelante. Async Login utiliza el operador await, el cual espera a que se verifique que el usuario existe en la base de datos para obtener la variable “existeUsuario” en el caso de que esta variable exista, se ejecuta la función “obtenerDatosUsuario” igualmente del servicio miServer, este hará que la base de datos busque los datos del usuario y los colocará en una variable llamada “datosUsuario”, a continuación, se verificará que el usuario y la contraseña coincida con los datos almacenados, si las credenciales coincide, el botón lo dejará

ingresar a la interfaz /home, caso contrario soltará un aviso que le indique al usuario que debe registrar una cuenta.

Para el botón registrar solo tiene implementado que al momento de hacer click lo lleve a la página de registro (/registro) para que el usuario pueda crear una nueva cuenta.

Gráfico 45 *Funcionalidad botón Ingresar y Registrarse*

<table>
<tr><th>CÓDIGO</th><th>RESULTADO</th></tr>
<tr><td>.TS BOTÓN INGRESAR

```
//Para el inicio de sesion
async login(){
  try {
    // Se verifica si existe el usuario en la base de datos
    const existeUsuario = await this.miServerService.verificarUsuarioExistente(this.usuario).toPromise();
    if (existeUsuario) {
      // Se obtienen los datos del usuario
      this.datosUsuario = await this.miServerService.obtenerDatosUsuario(this.usuario).toPromise();

      if (this.datosUsuario && this.datosUsuario.Contrasena === this.contrasena) {
        // Cuando las credenciales son válidas, se redirige a la página de inicio
        this.router.navigate(['/home'], { state: { usuario: this.usuario, datosUsuario: this.datosUsuario } });
      } else {
        this.presentToast('Contraseña incorrecta');
      }
    } else {
      this.presentToast('Usuario no encontrado');
    }
  } catch (error) {
    console.error('Error al verificar usuario', error);
    this.presentToast('Error al verificar usuario');
  }
}
```

.TS BOTÓN REGISTRARSE

```
//Para el registro de un nuevo usuario
registro(){
  this.router.navigate(['/registro']);
}
```

</td><td rowspan="2">
</td></tr>
</table>

Página de Registro

Esta página le permitirá al usuario ingresar datos importantes para el registro de su nueva cuenta, algunos de esos datos son: el usuario, el nombre completo, correo, contraseña y que se repita la contraseña, además, va a tener dos botones importantes que son el de CREAR CUENTA y el botón VOLVER AL INICIO. A continuación, se muestra la estructura HTML de la página de registro junto con su código SCSS del diseño.

Gráfico 46 *Estructura y diseño de página registro*

CÓDIGO

.HTML

```
<ion-card>
  <h2>Registrate</h2>
  <ion-card>
    <ion-input type="text" placeholder="Usuario" [(ngModel)]="Usuario"></ion-input>
  </ion-card>
  <ion-card>
    <ion-input type="text" placeholder="Nombre Completo" [(ngModel)]="Nombre_Completo"></ion-input>
  </ion-card>
  <ion-card>
    <ion-input type="email" placeholder="Correo" [(ngModel)]="Correo"></ion-input>
  </ion-card>
  <ion-card>
    <ion-input type="password" placeholder="Contraseña" [(ngModel)]="Contrasena"></ion-input>
  </ion-card>
  <ion-card>
    <ion-input type="password" placeholder="Repite la contraseña" [(ngModel)]="Repetir"></ion-input>
  </ion-card>
</ion-card>

<ion-button class="btn" (click)="registrarUsuario()">
  Crear Cuenta
</ion-button>

<ion-button class="btn" (click)="inicio()">
  Volver al inicio
</ion-button>
</ion-content>
```

.SCSS

```
h1{
    font-family:'Franklin Gothic Medium', 'Arial Narrow', Arial, sans-serif;
    text-align: center;
}
h2{
    margin: 10px;
    font-family:'Franklin Gothic Medium', 'Arial Narrow', Arial, sans-serif;
    font-size: 18px;
    color: ■black;
    text-align: center;
}

ion-input{
    font-family:'Franklin Gothic Medium', 'Arial Narrow', Arial, sans-serif;
    text-align: center;
    color: ■black;
}

.btn{
    border-radius: 10px;
    text-align: center;
    align-items: center;
    display: flex;
    justify-content: center;
}
```

RESULTADO



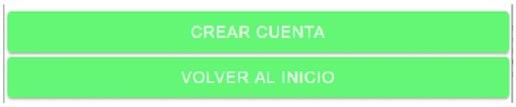
Para la funcionalidad de cada uno de los botones se utiliza el archivo registro.ts. Para el botón CREAR CUENTA se utiliza la función registrarUsuario (), esta se encarga de registrar una nueva cuenta en la base de datos del programa, luego se recopilan los datos en un objeto que se llama datosRegistro y se verifica que todos los datos estén completos, en el caso de que uno no esté completo se llama a la función presentToast4() que muestra el mensaje “Su cuenta no ha sido registrada. Debe completar todos los campos”. Existe otro caso, dónde las contraseñas no coincidan, aquí también se muestra un mensaje de error que dice “Su cuenta no ha sido registrada. Las contraseñas no coinciden”. Estas advertencias y errores le ayudan al usuario a verificar dónde debe corregir su error y tras corregirlo, ahora la aplicación podrá llamar a la función registrarUsuario() que se encuentra en el servicio miServer, si el usuario se registra exitosamente se muestra un mensaje de aviso que dice “Cuenta creada exitosamente. Para continuar vuelva al inicio” y en el caso de que el usuario o correo ya existan en la base de datos se muestra un mensaje de error que dice “Su cuenta no ha sido registrada. Las contraseñas

no coinciden". Con estas validaciones se asegura que en la base de datos solamente exista un usuario único, lo que facilitará la autenticación de cuentas.

El botón VOLVER AL INICIO solamente tiene la función de volver a la página de login para que el usuario pueda ingresar su usuario creado y contraseña, de esta forma podrá ingresar a la aplicación.

Gráfico 47 *Funcionalidad botón CREAR CUENTA y VOLVER AL INICIO*

CÓDIGO	RESULTADO

BOTÓN CREAR CUENTA
`if (!this.Usuario || !this.Nombre_Completo || !this.Correo || !this.Contrasena || !this.Repetir) {`
`  console.log('Por favor, complete todos los campos');`
`  this.presentToast4();`
`  return;`
`}`
`//si las contrasenas no coinciden entonces no podra crear la cuenta`
`if (this.Contrasena !== this.Repetir) {`
`  console.log('Las contraseñas no coinciden');`
`  this.presentToast2();`
`  return;`
`}`
`//Verificacion de que el usuario ingreso correctamente`
`  await this.miServerService.registrarUsuario(datosRegistro).toPromise();`
`  console.log('Usuario registrado exitosamente');`
`  this.presentToast1();`
`  //Limpiar datos luego de haber registrarlo al usuario`
`  this.Usuario = '';`
`  this.Nombre_Completo = '';`
`  this.Correo = '';`
`  this.Contrasena = '';`
`  this.Repetir = '';`
`} catch (error) {`
`  console.error('Error al registrar usuario', error);`
`  this.presentToast2();`
`}`
BOTÓN VOLVER AL INICIO
`inicio(){`
`  this.router.navigate(["/login"]);`
`}`
	

Páginas informativas

Las páginas informativas de esta aplicación solamente se encargan de mostrar la información al usuario, por lo tanto, cuentan con la misma estructura .html y scss. Las páginas que se encuentran en esta sección son: la página de "Acuerdo del usuario" que está definida como página /acuerdo. La página "Política de Privacidad" definida en el proyecto como /privacidad y la página de "Ayuda y apoyo" que se la localiza en el proyecto como /ayuda. Esta última es la única que tendrá una estructura HTML distinta a las demás. En el siguiente gráfico

se mostrará la estructura HTML de las dos primeras páginas informativas al igual que su archivo. scss.

Gráfico 48 *Estructura y diseño de páginas informativas*

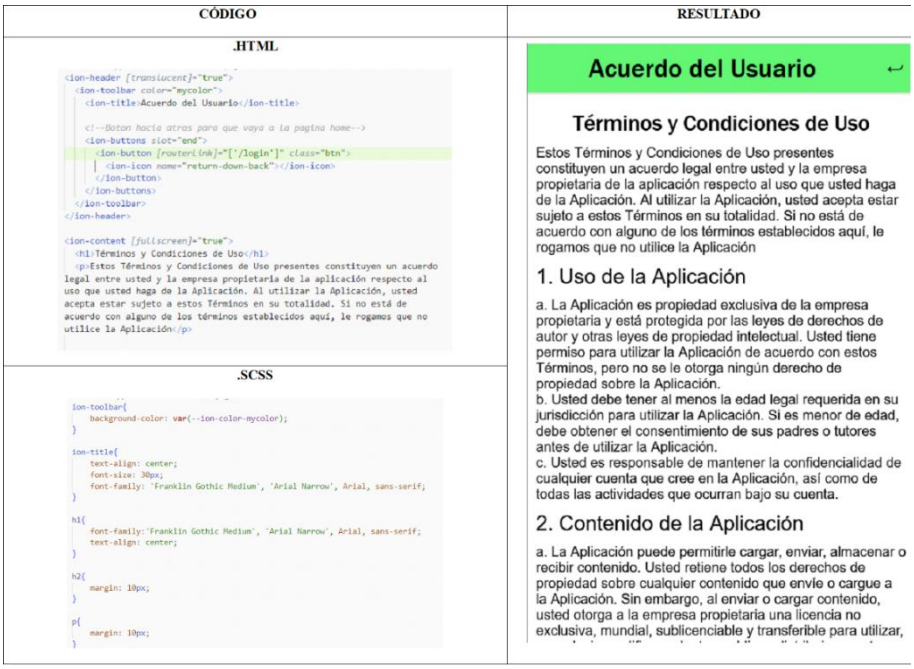


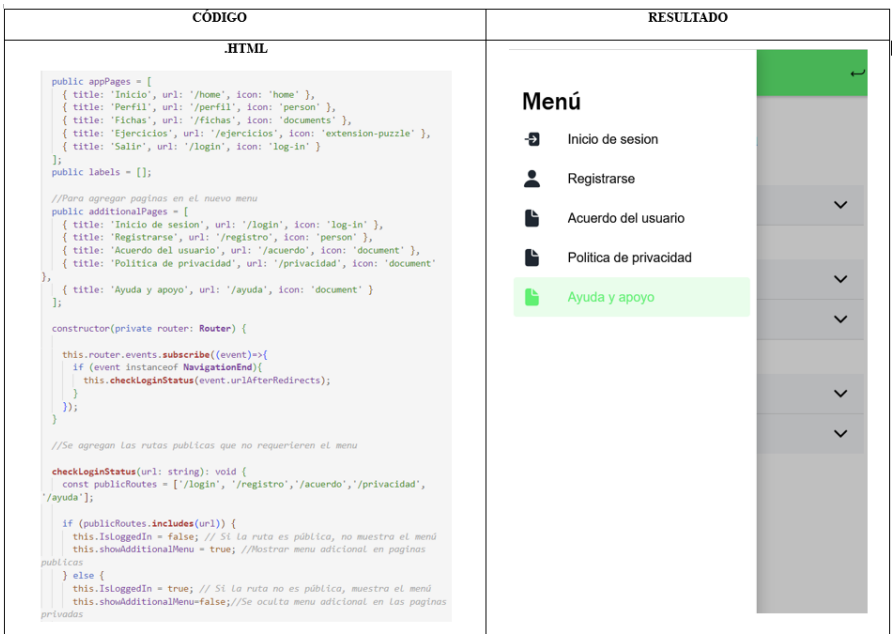
Gráfico 49 *Estructura y diseño página ayuda y apoyo*



Menú externo de la aplicación

El menú externo se creó con el objetivo de separar la información interna de la externa. En este menú se puede encontrar todas las páginas que hemos mencionado anteriormente como la de inicio de sesión, registrarse, acuerdos del usuario, política de privacidad y por último la página de ayuda y apoyo. Este menú lo pueden visualizar usuarios que no estén registrados en la base de datos de la aplicación, estos no podrán ver nada del menú interno a menos que se registren y luego inicien sesión. A continuación, se muestra la estructura de cómo se realizó la programación para que este menú externo se muestre únicamente cuando los usuarios no están registrados en la aplicación. Es importante mencionar, que toda la programación de este menú se realizó en app.components.ts. Para agregar las páginas en el menú externo se tuvo que declarar un arreglo llamado additionalPages. En el constructor se llama a la función checkLoginStatus, esta función compara la URL a la página que hace referencia, si esta se encuentra en publicRoutes, puede ser mostrada cuando el usuario no este ingresado a la sesión. Sin embargo, si no se encuentra dentro publicRoutes, entonces no será mostrada, a menos de que el usuario ingrese sesión.

Gráfico 50 *Estructura y funcionalidad del menú externo*

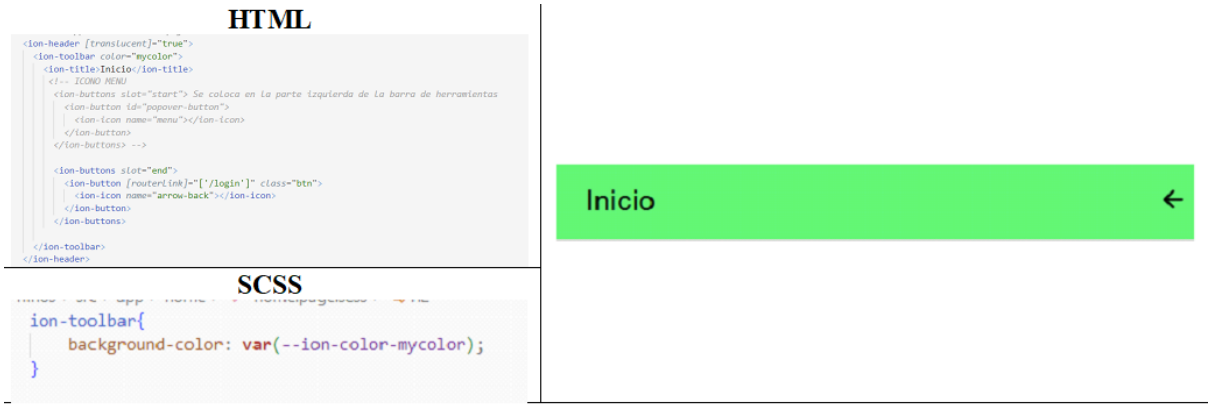


Desarrollo de las interfaces internas

Todas las interfaces internas cuentan con una barra de herramientas que se encuentra en la parte superior de cada interfaz, utiliza la misma estructura HTML y SCSS que será mostrada a continuación, la única diferencia que se encontrará en ellas es el nombre del título de cada página.

La barra de herramientas se crea con el objetivo de identificar la página en donde se encuentra el usuario, esta interfaz cuenta con un título y también con un icono hacia atrás que le ayuda al usuario a volver a la página de inicio o la página principal de donde se encontré.

Gráfico 51 *Estructura y diseño barra de herramientas de las interfaces internas*

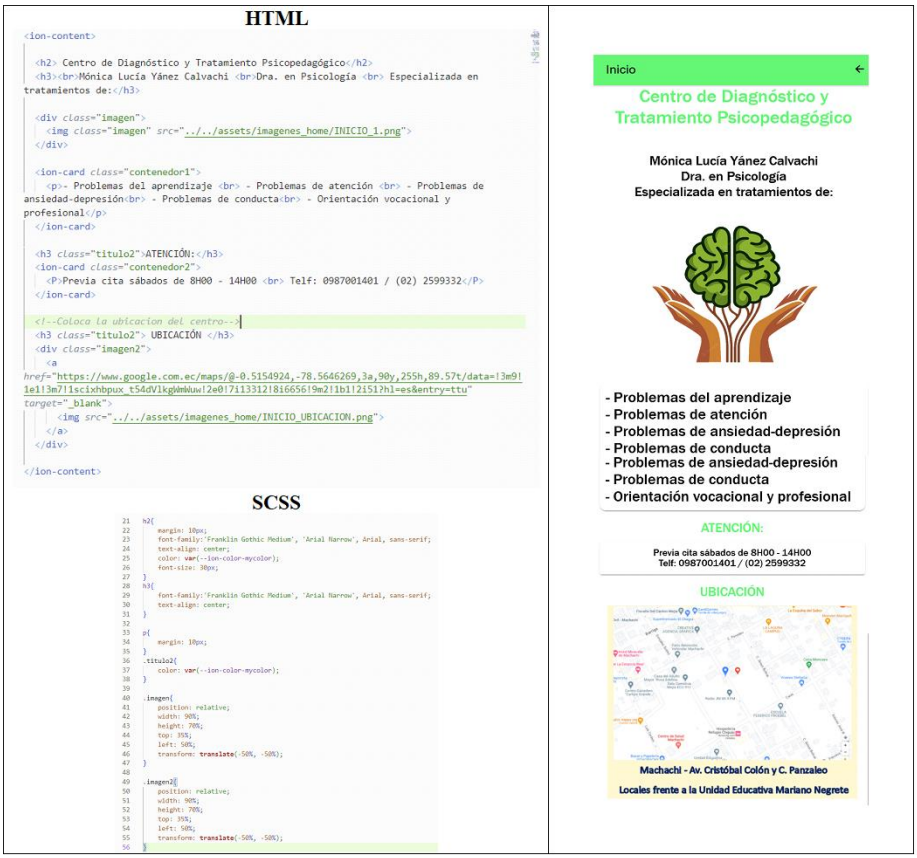


Página de inicio

La página de inicio se crea con el objetivo de hacer que el usuario entiendo de lo que se trata la aplicación, en este caso la aplicación es sobre un centro de diagnóstico psicológico que lo maneja una especialista y se expone todos los tratamientos ofrecidos. El diseño estructural de esta página se enfoca en el HTML, aquí se puede ver que están definidos los títulos y las imágenes, además, se crea una imagen que al momento de dar clic le lleva al usuario a la ubicación en Google Maps del consultorio. Es importante mencionar que en el código se muestra información sobre el centro de diagnóstico y nombre de la profesional, su horario,

tratamientos y su ubicación. A continuación, se muestra hoy el código HTMLY el código SCSS para el diseño estructural y el diseño gráfico de la página.

Gráfico 52 *Estructura y diseño de la página de inicio*

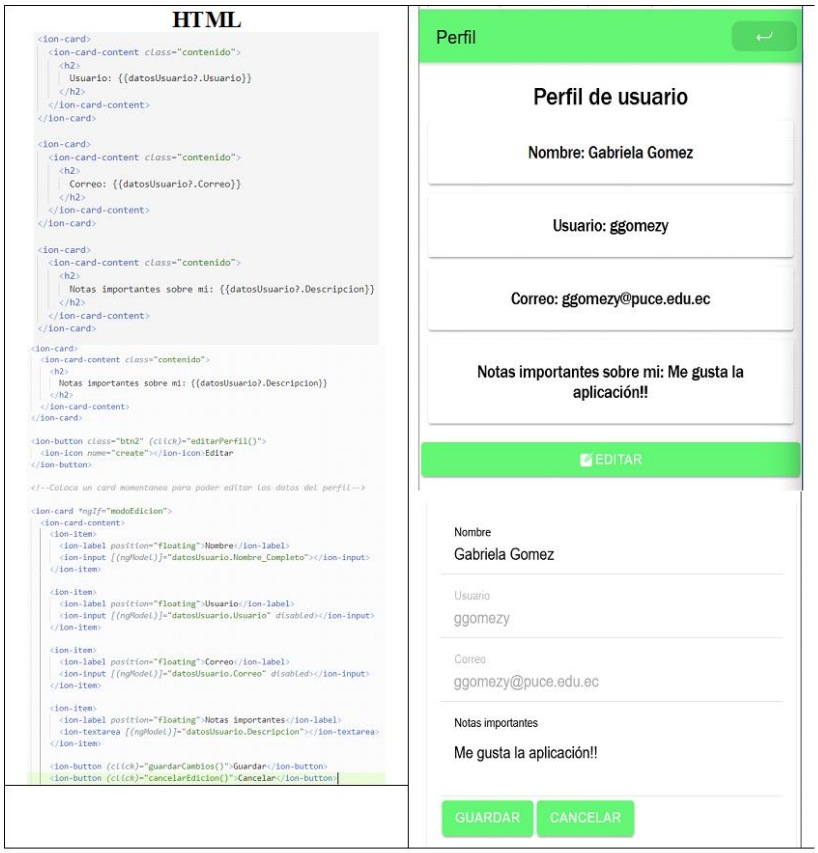


Página de perfil

La página de perfil se la realiza con el objetivo de mostrar todos los datos que el usuario ingresó en la página de registro, aquí se incluye el nombre de la persona que está haciendo uso de la aplicación, el nombre de usuario, el correo y una sección de notas importantes sobre el usuario. Cuenta con un botón para poder editar los datos que se encuentra en el perfil de usuario, a excepción del usuario y el correo, los demás datos son totalmente editables. El usuario y el correo no se editan debido a que hay una regla en la base de datos de que estos datos sean totalmente únicos y que si se repiten no pueden ser creados. A continuación se muestra el HTML, dentro de este archivo se observa que con la expresión {{datosUsuario?.Nombre_Completo}} se pueden llamar a los datos del usuario desde la base

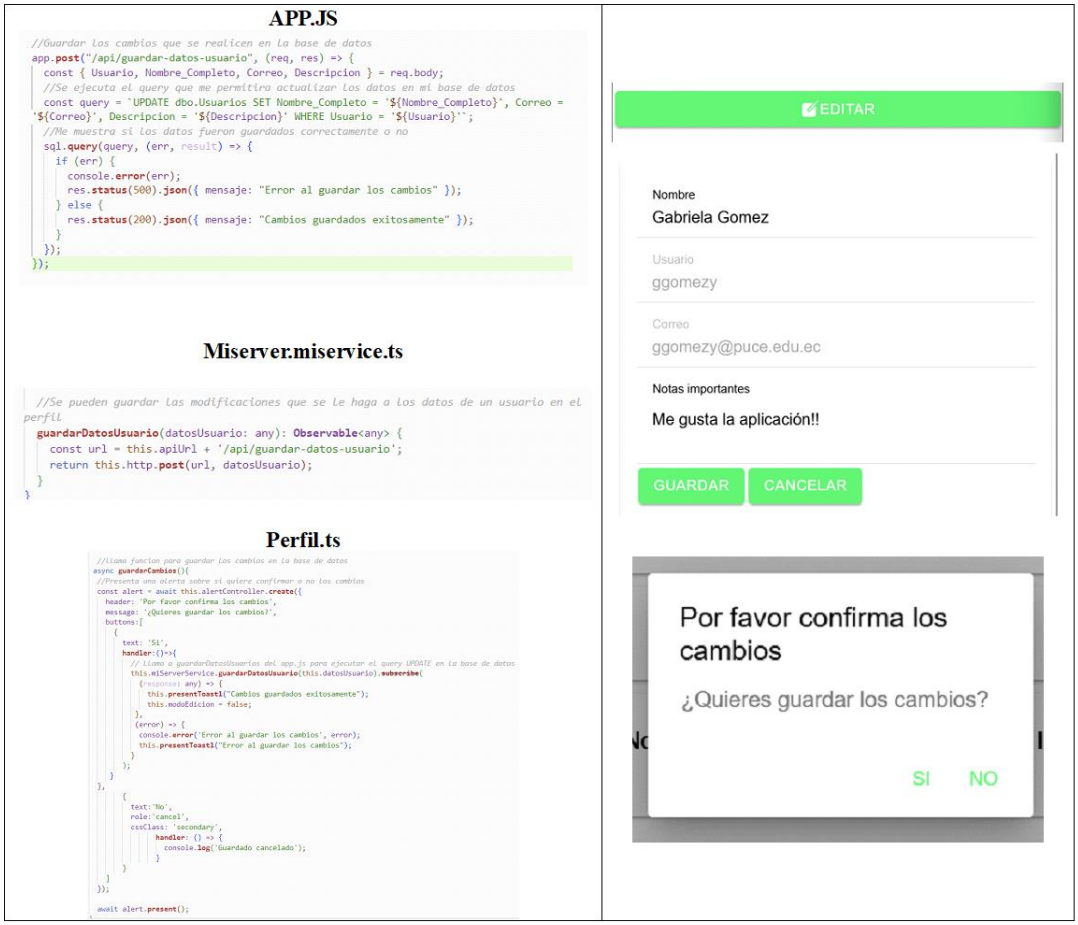
de datos a la interfaz visual del perfil. Es importante mencionar que cuando el usuario hace clic en el botón de editar este despliega otra carta debajo de los datos donde se pueden escribir nuevos datos, una vez modificados, el usuario puede hacer click en guardar los datos y el sistema desplegará una ventana para verificar si desea guardar los cambios realizados, caso contrario solo podrá colocar el botón cancelar.

Gráfico 53 *Estructura y diseño de la página de perfil*



Para la funcionalidad del botón de editar es necesario crear una nueva función en el archivo app.js que es el documento que tiene conexión con la base de datos, dentro de la función la aplicación manda un comando query a la base de datos actualizando las modificaciones que se hayan realizado. A continuación, desde el archivo miserver.service.ts se llama a la función que fue realizada en el archivo app.js y esta le brindará la funcionalidad al botó de Guardar.

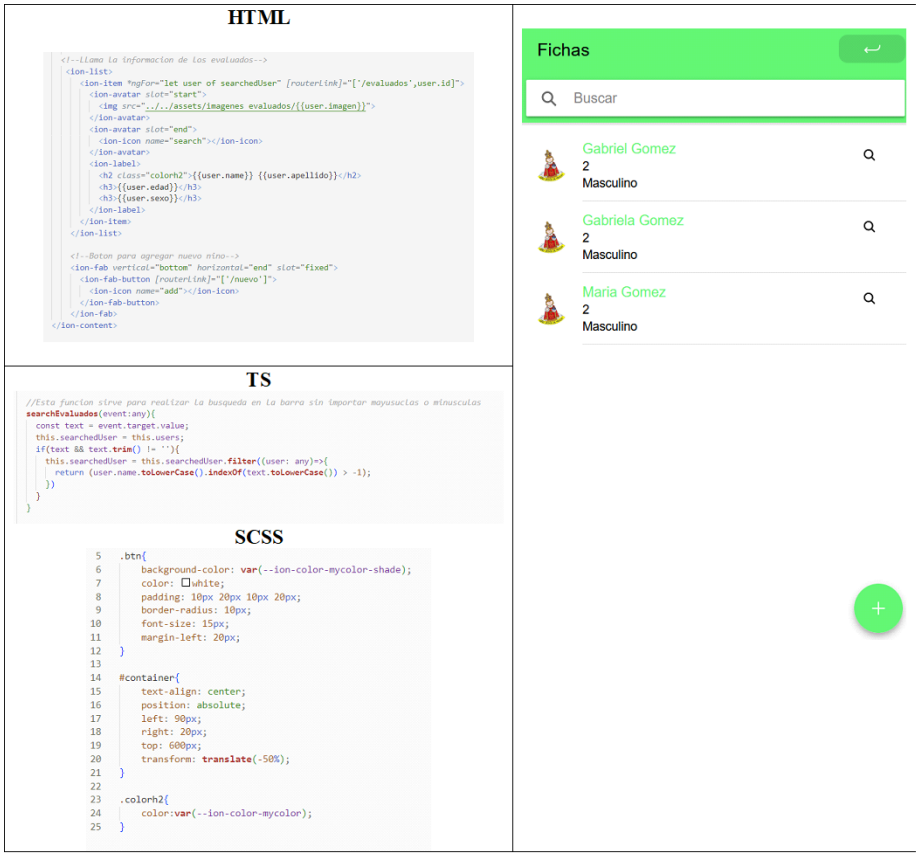
Gráfico 54 *Funcionalidad botón Editar en página perfil*



Página de fichas

La página de fichas se utiliza con el objetivo de observar todas las fichas creadas por el evaluador o por el usuario, en cada una de las fichas se podrá observar el nombre del evaluado, el sexo y la edad. La página cuenta con un buscador, este se utiliza en el caso de tener muchos evaluados, facilitará la búsqueda del evaluado que esté tratando de encontrar la persona que está utilizando la aplicación, la página también cumple con un botón para poder agregar nuevos evaluados y el acceso a una interfaz para ver a los evaluados ingresados. El siguiente código presentado mostrará la estructura y el diseño de la página de fichas y la llamada de algunas funciones para que cada uno de los botones utilizados dentro de esta interfaz funcionen correctamente.

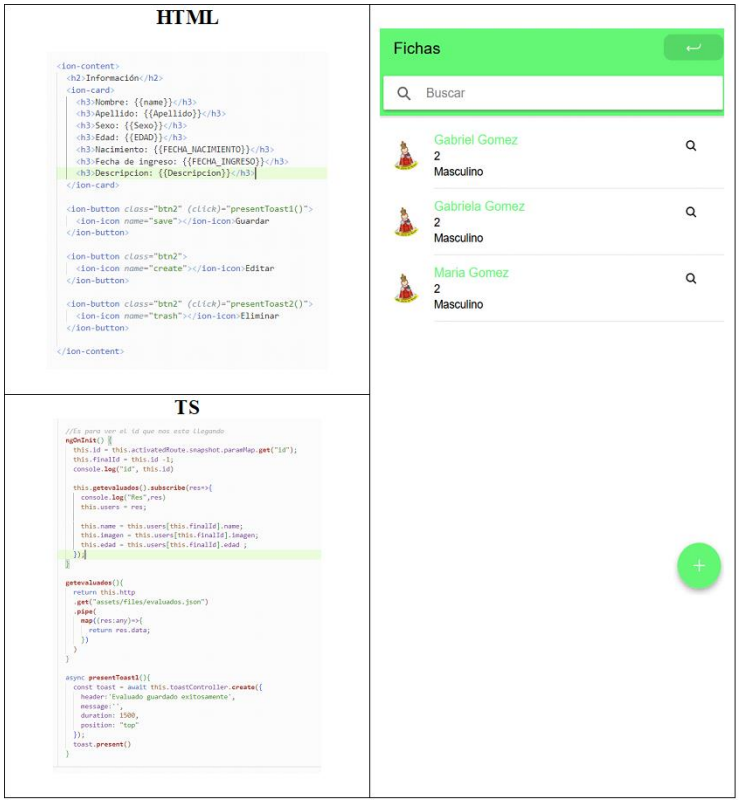
Gráfico 55 *Estructura y diseño de la página de fichas*



Página de información de un evaluado

Dentro de la página de información de un nuevo evaluado se muestran todos los datos que fueron registrados cuando se ingresó a un nuevo niño evaluado, dentro de esta página se podrá encontrar todos los datos del niño, la información del padre y la información de la madre, también se va a mostrar el test aplicado luego de la evaluación que realizó el especialista, el diagnóstico, el resultado, las recomendaciones y el plan de tratamiento que se seguirá con este niño evaluado. Es importante de mencionar que la página cuenta con los botones de guardar, editar y cerrar, estos le permitirán a la especialista interactuar al con la información de los niños evaluados en el caso de que exista algún percance o nueva información necesaria de registrar. A continuación, se muestra el código html y ts que se utilizó para realizar la página y que funcione correctamente.

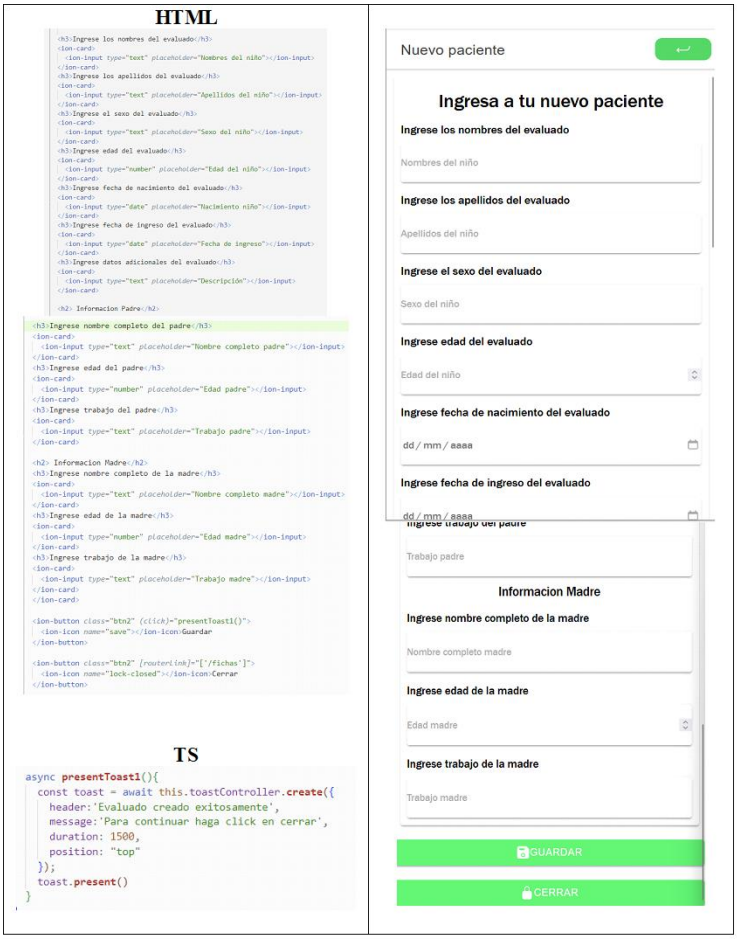
Gráfico 56 *Estructura y diseño de la página de información de un evaluado*



Página para agregar a un evaluado

La página para agregar a un nuevo evaluado será necesaria para el ingreso de la información del niño que está siendo registrado la base de datos del centro de diagnóstico, esta le permitirá a la especialista almacenar toda la información del niño, al igual que la información del padre y la madre, además, dentro de esta interfaz, la especialista puede almacenar toda la historia clínica del niño como el desarrollo que ha tenido a lo largo de su vida, el desempeño que tuvo en el test, información de los test que se han aplicado, el diagnóstico, las recomendaciones y el plan de tratamiento. Es de suma importancia tener esta historia clínica debido a que le ayuda a la psicóloga a tener un repositorio de toda la información los niños evaluados en su consultorio.

Gráfico 57 *Estructura y diseño de la página para agregar a un nuevo evaluado*

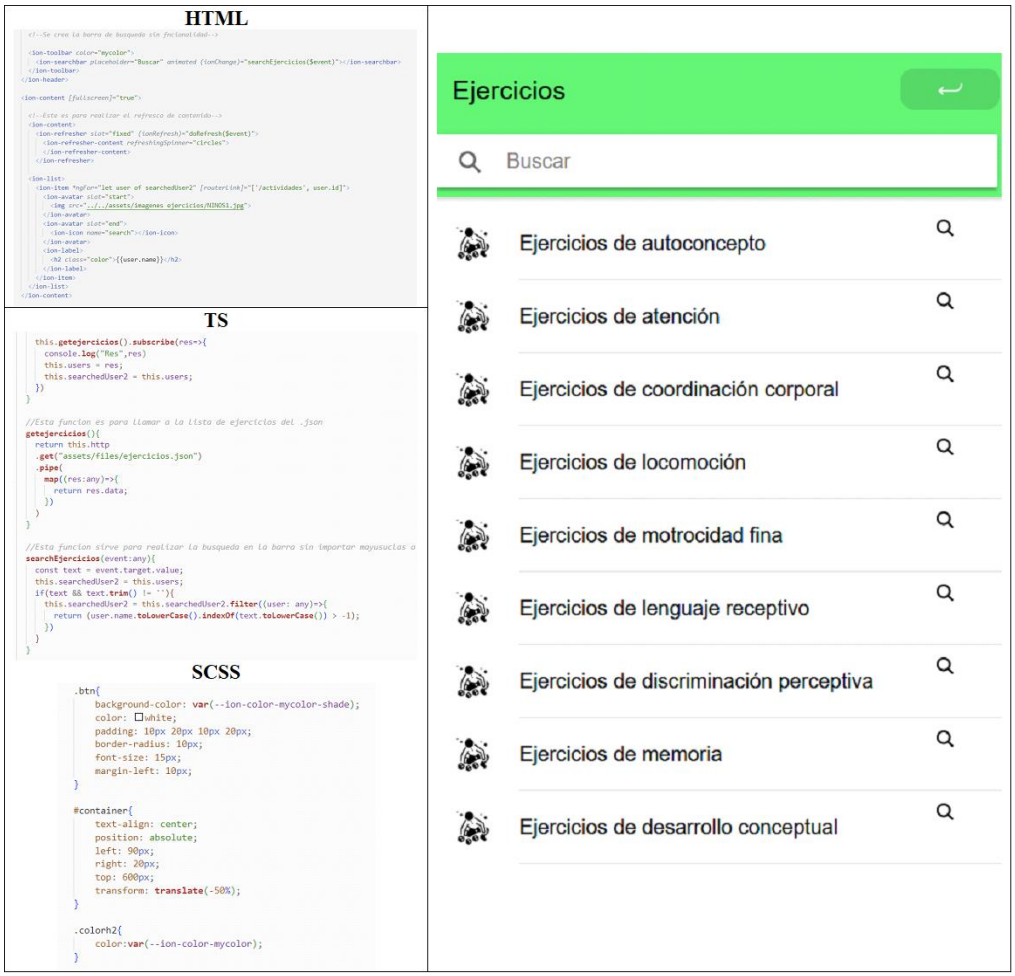


Página de ejercicios

La página de ejercicios sirve para encontrar todo tipo de ejercicios de estimulación del pensamiento para niños entre 2 a 4 años, esta tiene la misma estructura que la interfaz de fichas, con la diferencia de que va a encontrar solamente los tipos de ejercicio que el usuario desee buscar, contiene un buscador para realizar la acción de encontrar un tipo de ejercicios y se aplasta en 1 de ellos le va a llevar a una nueva interfaz donde se van a especificar las actividades de ese tipo de ejercicio. Esta página no es editable debido a que está diseñada para que el usuario solo lea y siga las instrucciones de los ejercicios que debe realizar con su niño, teniendo en cuenta el diagnóstico y resultado que le dio la especialista luego de aplicar los test necesarios. Es importante mencionar que toda la información de esta sección del programa esta almacenada en una archivo.json que la página llama para poder mostrar en la interfaz de

ejercicios. La siguiente imagen mostrará el código html de la estructura de la página al igual que el scss del diseño de la misma y el ts que ayuda a la funcionalidad de la página.

Gráfico 58 *Estructura y diseño de la página de ejercicios*

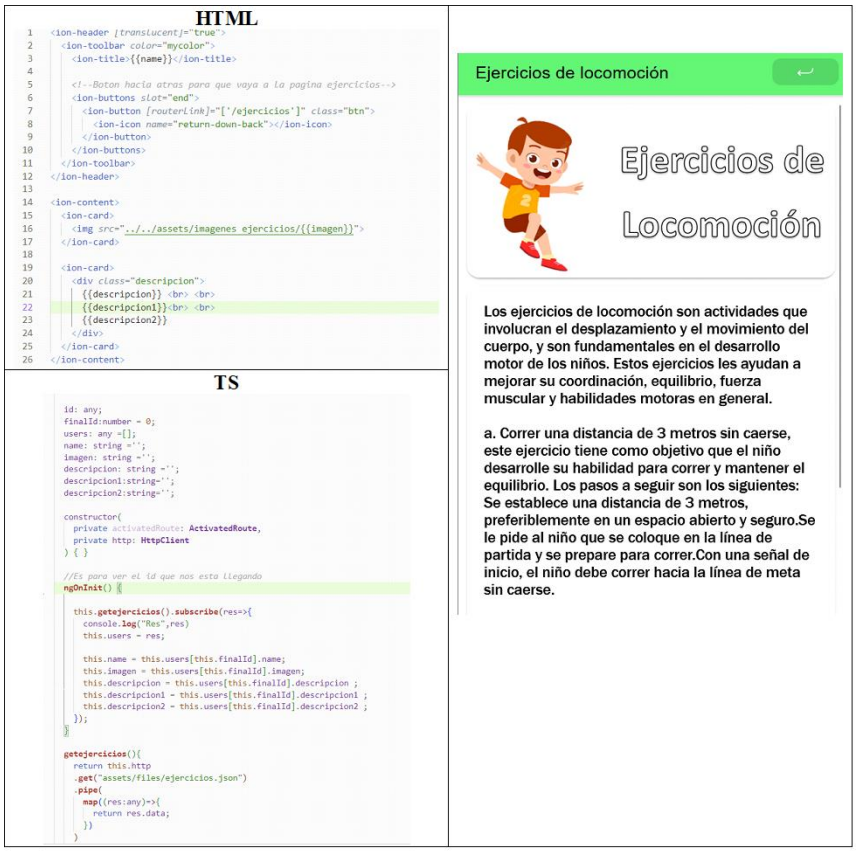


Página de actividades

La página de actividades se creó con el objetivo de mostrar todos los ejercicios que debe realizar el responsable o padre de familia del niño. Este cuenta con ejercicios brindados por la especialista para mejorar cualquier deficiencia que tenga el niño en el desarrollo del aprendizaje. La página contará con una imagen y debajo de ella con una descripción de los ejercicios, es importante mencionar que todo el contenido de la página está almacenado en un archivo. json al cual se lo llama desde el archivo. ts de la aplicación. El desarrollo de esta

página fue sencillo debido a que no se tuvieron que utilizar muchos elementos simplemente se utilizó el llamado al archivo. json que contiene toda la información necesaria.

Gráfico 59 *Estructura y diseño de la página de actividades*

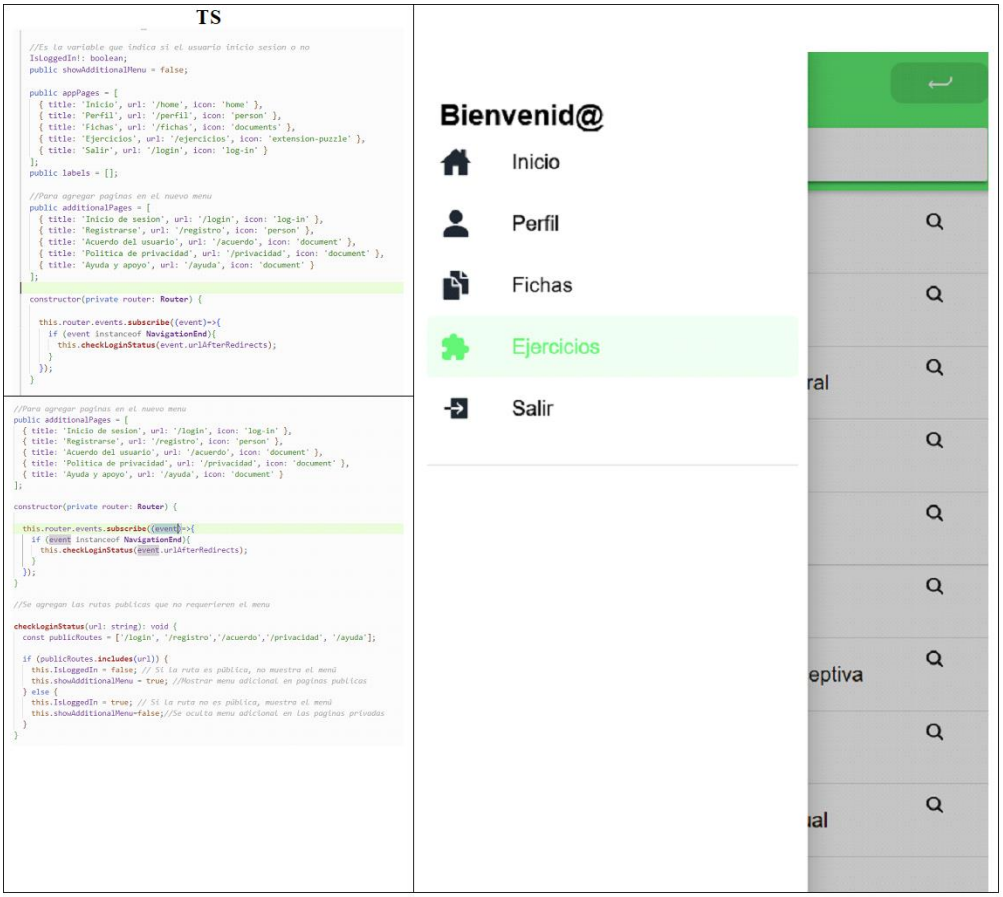


Menú interno de la aplicación

El menú interno se creó con el objetivo de separar la información interna de la externa. En este menú se puede encontrar algunas de las páginas que se ha mencionado anteriormente como la de inicio, perfil, fichas, ejercicios y un botón de salir para poder regresar a la página de inicio de sesión. Este menú lo pueden visualizar los usuarios que estén registrados en la base de datos de la aplicación. A continuación, se muestra la estructura de cómo se realizó la programación para que este menú interno no se muestre cuando los usuarios no están registrados en la aplicación. Es importante mencionar, que toda la programación de este menú se realizó en app.components.ts. Para agregar las páginas para separar las páginas externas de las internas se tuvo que declarar un arreglo llamado additionalPages. En el constructor se llama a la función

checkLoginStatus, esta función compara la URL a la página que hace referencia, si esta se encuentra en publicRoutes, puede ser mostrada cuando el usuario no este ingresado a la sesión. Sin embargo, si no se encuentra dentro publicRoutes, entonces no será mostrada, a menos de que el usuario ingrese sesión.

Gráfico 60 *Estructura del menú interno de la aplicación*

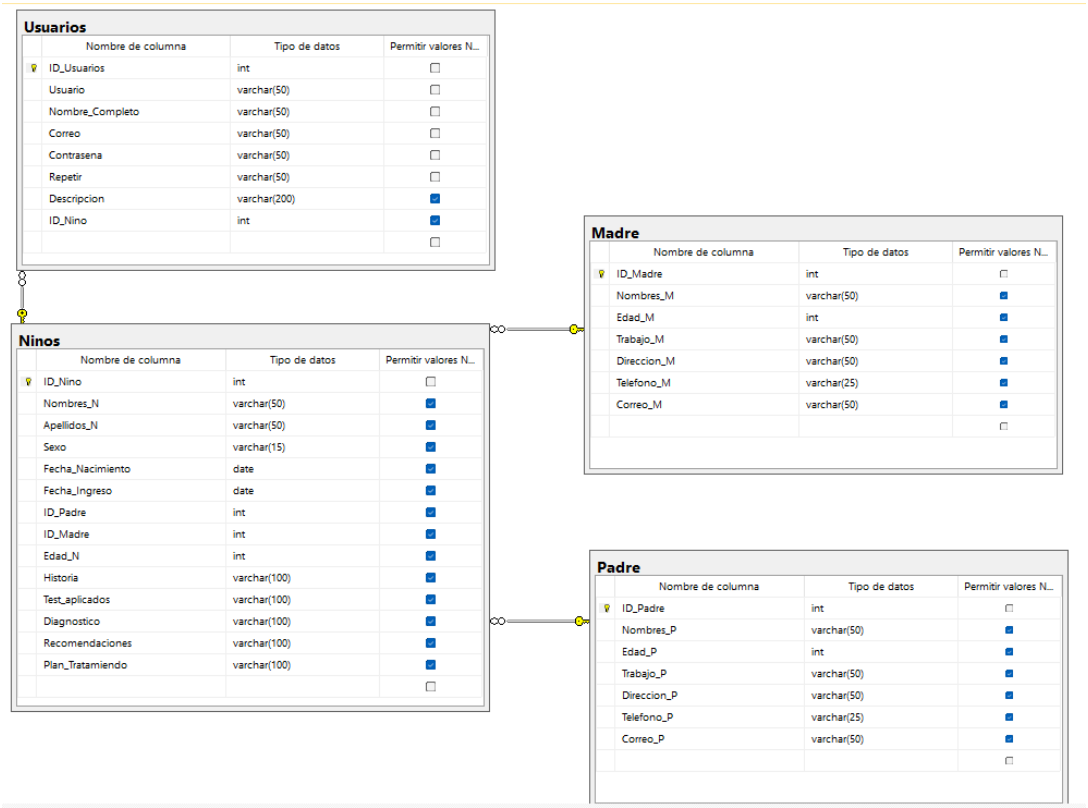


5.3.Construcción de la base de datos

La base de datos que se utilizó es SQL Server 2014 Management Studio, se eligió este administrador de base de datos debido a su interfaz gráfica fácil de utilizar que permite crear modificar y administrar tablas de bases de datos relacionales de manera sencilla. Esta herramienta además de tener múltiples funcionalidades es fácil de integrar a visual Studio y al api rest que se utilizó en este proyecto de titulación.

Para construir la base de datos de la aplicación se tuvo que tomar en cuenta todos los requerimientos que pedía el usuario. La base de datos que se creó en el administrador de bases de datos se llama DB_Centro, cuenta con cuatro tablas: la tabla de la madre, la tabla del padre, la tabla de la información de un nuevo niño registrado y la tabla de los usuarios. Cada una de las tablas de esta base de datos relacional cuenta con su respectivo nombre de tabla, columnas, dónde se encuentran los atributos de la tabla, el tipo de datos que tiene cada atributo, su clave primaria y las relaciones que tiene con otras tablas. A continuación, se mostrará el diagrama de la base de datos relacional para el desarrollo de la aplicación móvil.

Gráfico 61 *Diagrama de la base de datos relacional*

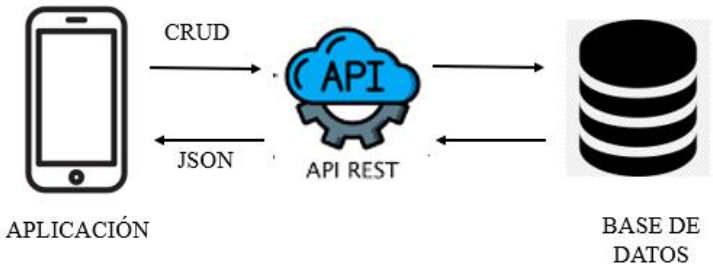


5.4. Construcción y comunicación del api rest con la aplicación y la base de datos

Un api rest es la comunicación que tienen dos sistemas para intercambiar la información de una forma segura a través de internet (AWS, 2023). En esta ocasión se utiliza esta herramienta para manejar las operaciones de la base de datos como un servicio backend, esto quiere decir, que se ejecuta en segundo plano. De esta forma, la aplicación se comunica con el

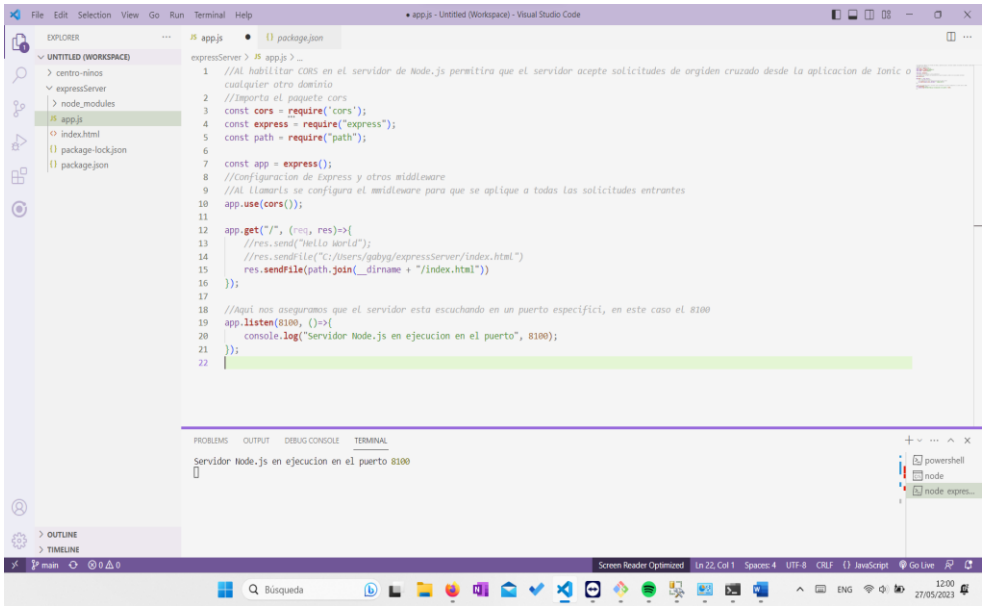
apirest para poder realizar operaciones de consulta, edición, creación o eliminación en la base de datos. A continuación, se mostrará un diagrama de cómo es la comunicación entre la aplicación, el api rest y la base de datos.

Gráfico 62 *Diagrama de comunicación entre aplicación, api rest y base de datos*



Para la conexión del api rest con la aplicación y la base de datos, primero se tuvo que crear un nuevo proyecto dónde se realizará la creación del Api rest con la aplicación y la base de datos, el archivo dónde se configurará todo esto se llama app.js. En este archivo, primero se debe habilitar el CORS en el servidor de Node.js, este permitirá que el servidor acepte solicitudes de origen cruzado, esto quiere decir que permitirá solicitudes de Ionic u otro dominio. A continuación, se verifica que el servidor este escuchando en un puerto específico, para esta ocasión se utilizó el puerto 8100.

Gráfico 63 *Creación de Api Rest*



Para poder configurar la conexión de la aplicación de Ionic con el api rest, es necesario modificar los archivos environment.prod.ts y enviroment.ts, dentro de estos archivos se colocará la URL con el puerto correcto para el servidor Node.js, que en este caso es http://localhost:8100

Gráfico 64 *Configuración con la aplicacion de Ionic en environment.ts*

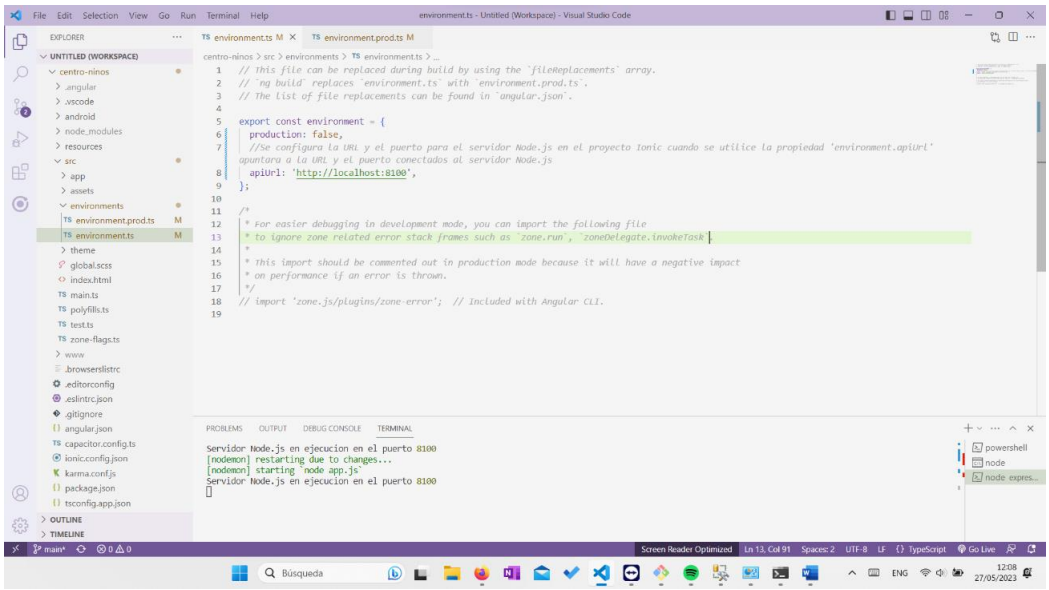
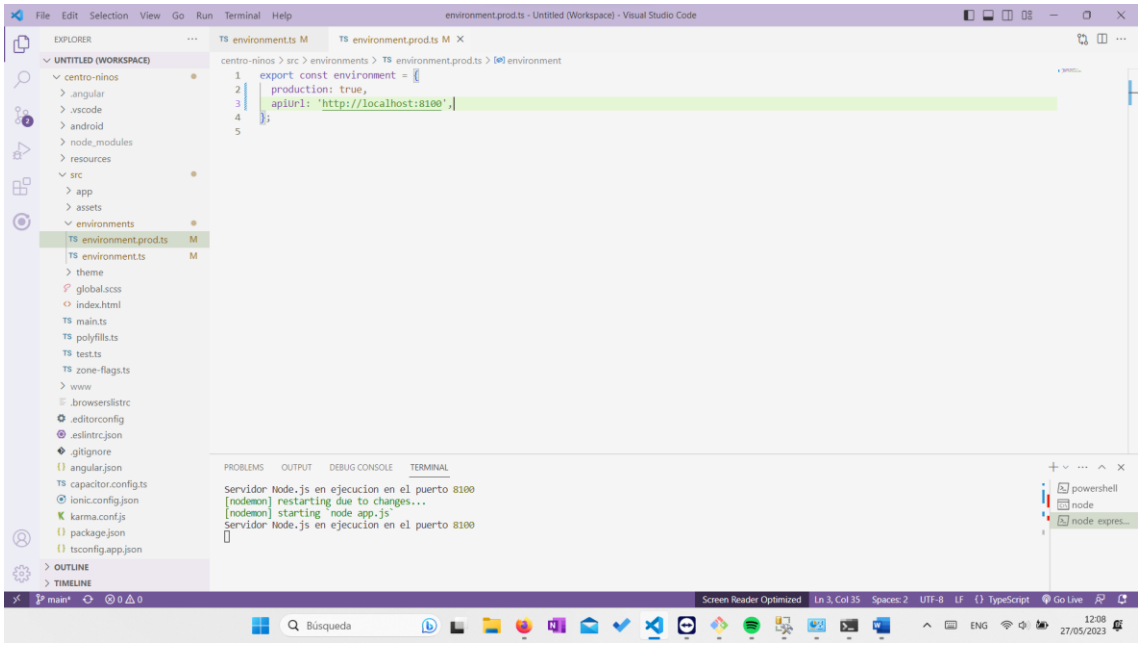
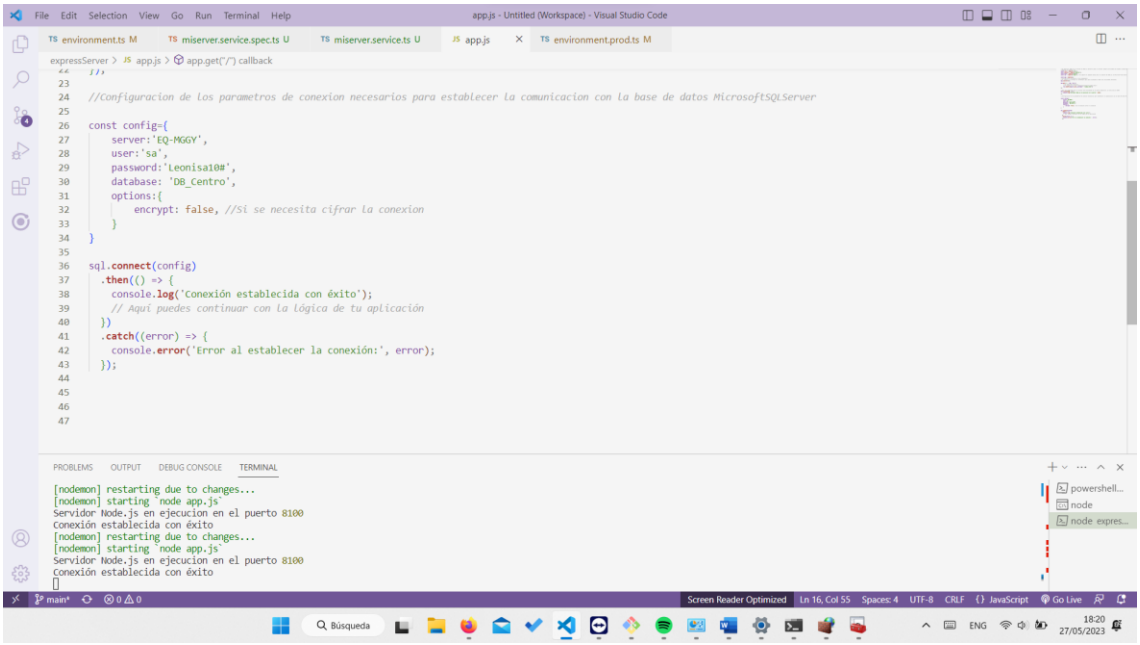


Gráfico 65 *Configuración con la aplicación de Ionic en environment.prod.ts*



Los pasos anteriores demuestran que hay conectividad entre la aplicación Ionic y el Api rest. Como paso final, se debe realizar la conexión entre el Api rest y la base de datos, para ello es necesario ejecutar el paquete de mssql que se lo realiza con el comando npm install mssql, una vez instalado, en el archivo app.js se debe escribir la configuración de la base de datos, la configuración incluye información como el nombre del servidor, usuario, contraseña y el nombre de la base de datos. Para asegurarse de que si existe conectividad se realizó una condición que muestra por consola “Conexión establecida con éxito” cuando el api rest y la base de datos tienen conexión y “Error al establecer la base de datos” cuando la conexión entre ellas ha fallado.

Gráfico 66 *Configuración de la conexión entre el api rest y la base de datos*



5.5. Pruebas funcionales del sistema

Para verificar las pruebas funcionales del sistema se realizará una serie de tablas con los requisitos funcionales que se describieron anteriormente dentro del proyecto de titulación. Esto con el objetivo de comprobar cada funcionalidad y requerimiento del usuario. Estas tablas contendrán una descripción, precondiciones condiciones de ejecución, pasos que se realizaron para comprobar el funcionamiento del requerimiento, los resultados esperados, el estado del caso de prueba, el resultado obtenido y los errores asociados.

Tabla 5 *Pruebas del requerimiento funcional 1*

RF1 Inicio de sesión y registro	
Descripción	El sistema debe permitir al usuario registrar sus datos y acceder a la aplicación con un usuario y contraseña
Precondiciones	Los módulos front end y back end deben tener conexión
Condiciones de ejecución	La ejecución se realiza en un sistema operativo Windows
Pasos para realizar la prueba	1. Se ejecuta el programa en ionic 2. Se inicia la conexión de ionic con el api rest y la base de datos 3. Se interactúa con la pantalla de login 4. Se crea un usuario

		5. Se verifica que salgan todos los mensajes correctamente 6. Se inicia sesión en la aplicación	
Resultados de la prueba El resultado de la prueba es exitoso			
Estado del caso de prueba	Exitoso (x)	Fallido ( )	Frenado( )
Errores asociados		Por el momento no se encuentra ningún error asociado	
Comentarios		El comentario emergente puede durar más tiempo para que el usuario termine de leerlo	

Tabla 6 *Pruebas del requerimiento funcional 2*

RF2 Desarrollo de interfaces externas			
Descripción		El sistema debe contar con cinco pantallas externas (Inicio de sesión, Registro, Ayuda, Acuerdo con el usuario y política de privacidad)	
Precondiciones		El diseño de la interfaz de cada una debe contener todos los requerimientos no funcionales que el usuario solicitó. Debe contener contenido claro y legible.	
Condiciones de ejecución		La ejecución se realiza en un sistema operativo Windows	
Pasos para realizar la prueba		1. Se ejecuta el programa en ionic 2. Se dirige a las interfaces externas 3. Se observan si la letra es legible 4. Se observa que el desarrollo y estructura de la página no tenga errores. 5. Se observa que se encuentren las páginas solicitadas	
Resultados de la prueba El resultado de la prueba es exitoso si se logra navegar correctamente entre las páginas. El contenido de cada una de ellas va acorde a lo solicitado por el usuario El contenido es legible y claro No es difícil de manejarlo			
Estado del caso de prueba	Exitoso (x)	Fallido ( )	Frenado( )
Errores asociados		Por el momento no se encuentra ningún error asociado	
Comentarios		No aplica	

***Tabla** 7 Pruebas del requerimiento funcional 3*

RF2 Desarrollo de interfaces internas			
Descripción		El sistema debe contar con 4 pantallas internas (Perfil, Fichas Inicio y Ejercicios)	
Precondiciones		El diseño de la interfaz de cada una debe contener todos los requerimientos no funcionales que el usuario solicitó. Debe contener contenido claro y legible.	
Condiciones de ejecución		La ejecución se realiza en un sistema operativo Windows	
Pasos para realizar la prueba		1. Se ejecuta el programa en ionic 2. Se dirige a las interfaces internas 3. Se observan si la letra es legible 4. Se observa que el desarrollo y estructura de la página no tenga errores. 5. Se observa que se encuentren las páginas solicitadas	
Resultados de la prueba El resultado de la prueba es exitoso si se logra navegar correctamente entre las páginas. El contenido de cada una de ellas va acorde a lo solicitado por el usuario El contenido es legible y claro No es difícil de manejarlo			
Estado del caso de prueba	Exitoso (x)	Fallido ( )	Frenado( )
Errores asociados		Por el momento no se encuentra ningún error asociado	
Comentarios		No aplica	

***Tabla 8** Pruebas del requerimiento funcional 4*

RF4 Navegación entre interfaces	
Descripción	El sistema debe contar con botones que permitan la correcta navegación entre todas las interfaces
Precondiciones	El diseño de cada interfaz debe ser acorde a los requerimientos del usuario. Debe existir una buena implementación de enrutamiento para la navegación entre las páginas. Los botones deben estar correctamente dirigidos para la navegación entre todas las interfaces
Condiciones de ejecución	La ejecución se realiza en un sistema operativo Windows
Pasos para realizar la prueba	1. Se ejecuta el programa en ionic 2. Se dirige al menú de la aplicación

	3. Se observan los títulos de las páginas 4. Se inicia con la navegación entre ellas. 5. Se inicia sesión en la aplicación 6. Se observa un menú diferente 7. Se observan los títulos de las páginas 8. Se inicia la navegación entre ellas.
Resultados de la prueba El resultado de la prueba es exitoso si se logra navegar correctamente entre las páginas. Siempre está presente el menú de navegación Las páginas están acordes a su contenido Existe un botón de retroceso que dependiendo de dónde se encuentre le permite volver a la página principal	

Estado del caso de prueba	Exitoso (x)	Fallido ( )	Frenado( )
Errores asociados		Por el momento no se encuentra ningún error asociado	
Comentarios		No aplica	

Tabla 9 *Pruebas del requerimiento funcional 5*

RF5 Gestión de datos	
Descripción	El sistema debe tener conexión con una base de datos para manejar la edición de los mismos en las interfaces que lo requieran
Precondiciones	Los módulos front end y back end deben tener conexión.
Condiciones de ejecución	La ejecución se realiza en un sistema operativo Windows
Pasos para realizar la prueba	1. Se ejecuta el programa en ionic 2. Se realiza la primera prueba en la página de registro 3. Se revisa que los datos sean almacenados correctamente 4. Se realiza la segunda prueba en el inicio de sesión 5. Se comprueba que se ejecute correctamente botón de ingreso (si el usuario esta registrado o no) 6. Se realiza la tercera prueba en la página de perfil. 7. Se comprueba que se puedan editar y guardar cambios. 8. Se realiza la cuarta prueba en la página de fichas.

		9. Se comprueba que se puedan agregar usuarios, editarlos, eliminarlos y verlos.	
Resultados de la prueba El resultado de la prueba es medianamente exitoso, algunos de los botones no cumplen con la función que deben o no están funcionando correctamente. Si aparece el mensaje de aviso de error en el caso de que no se registre, edite o elimine un botón.			
Estado del caso de prueba	Exitoso ( )	Fallido (x)	Frenado( )
Errores asociados		Algunos botones no cumplen con la función que deben	
Comentarios		Revisar la programación de cada uno de ellos para tener una buena funcionalidad en cada uno de ellos.	

Tabla 10 *Pruebas del requerimiento funcional 6*

RF6 Cambios en tiempo real			
Descripción		El usuario debe ser capaz de ver los cambios que realiza el sistema tiempo real	
Precondiciones		Cada vez que el usuario realice un cambio, ya sea agregar, editar o eliminar lo debe ver rápidamente reflejado en la aplicación	
Condiciones de ejecución		La ejecución se realiza en un sistema operativo Windows	
Pasos para realizar la prueba		1. Se ejecuta el programa en ionic 2. Se dirige a una de las páginas editables como Perfil 3. Se realiza la prueba de edición 4. Se comprueba que los cambios se muestren enseguida. 5. Se dirige a otra página editable 6. Se realiza la prueba de edición y eliminación 7. Se comprueba que los cambios se muestren enseguida	
Resultados de la prueba El resultado de la prueba fue medianamente éxitoso debido a que en el requerimiento anterior algunos de los botones no funcionaban correctamente no se pudo obtener un resultado claro de los cambios en tiempo real. Sin embargo, de los botones que funcionan correctamente, si se muestra los cambios en tiempo real y rápidamente.			
Estado del caso de prueba	Exitoso (x)	Fallido ( )	Frenado( )
Errores asociados		Error asociado con el requerimiento funcional 5.	
Comentarios		No aplica	

Tabla 11 *Pruebas del requerimiento funcional 7*

RF7 Advertencias y mensajes de error			
Descripción		El usuario debe ser capaz de ver los errores o advertencias que devuelve el sistema	
Precondiciones		Al momento de ejecutar un botón y que el contenido no este correctamente utilizado, debe aparecer un mensaje de error o advertencia al usuario, indicándole en qué se equivocó o lo que debe tener en consideración.	
Condiciones de ejecución		La ejecución se realiza en un sistema operativo Windows	
Pasos para realizar la prueba		1. Se ejecuta el programa en ionic 2. Navega por la aplicación 3. Se pone a prueba los botones que se encuentren dentro de la aplicación 4. Se comprueba que los botones manden mensajes de error o advertencia al usuario en el caso de que no estén aplicándolo bien.	
Resultados de la prueba El resultado de la prueba es exitoso se muestran mensajes emergentes de error y advertencia al usuario en el caso de que realice una acción mal o le falte completar algo. Los mensajes son claros, cortos y entendibles para el usuario. No se utilizan palabras técnicas.			
Estado del caso de prueba	Exitoso (x)	Fallido ( )	Frenado( )
Errores asociados		Por el momento no se encuentra ningún error asociado	
Comentarios		No aplica	

Tabla 12 *Pruebas del requerimiento funcional 8*

RF8 Funcionalidad de botones	
Descripción	Varios de los botones tendrán las funciones de un CRUD por lo que el sistema debe funcionar correctamente
Precondiciones	Todos los botones deben funcionar correctamente, dirigir a las páginas adecuadas o responder a las acciones de un CRUD en el caso de ser necesario
Condiciones de ejecución	La ejecución se realiza en un sistema operativo Windows
Pasos para realizar la prueba	1. Se ejecuta el programa en ionic 2. Navega por la aplicación 3. Se pone a prueba los botones que se encuentren dentro de la aplicación

		4. Se comprueba que los botones funcionen correctamente a lo que menciona el título.	
Resultados de la prueba El resultado de la prueba es medianamente exitoso, todos los botones están funcionando correctamente al momento de redirigir a una nueva página, los botones de edición y guardado están funcionando. Sin embargo, los botones de eliminar tienen un pequeño fallo.			
Estado del caso de prueba	Exitoso (x)	Fallido ( )	Frenado( )
Errores asociados		Error asociado con el requerimiento funcional 5	
Comentarios		No aplica	

Teniendo en cuenta las tablas pruebas de los requerimientos funcionales se procedió a realizar las correcciones necesarias para poder mostrarle al usuario el resultado final de la aplicación y poder realizar la evaluación de la usabilidad.

5.6. Evaluación de la usabilidad

La evaluación de la usabilidad se la utiliza para comprender la interacción entre el usuario y el software, se observa como este percibe a la aplicación, la interacción que tiene con el mismo y la experiencia del usuario. La evaluación que se utilizará para este proyecto de titulación será la evaluación del conductor esta implica observar y analizar la interacción del usuario con cada función del software, esto se realiza con el objetivo de encontrar problemas de usabilidad o retroalimentación para adaptar el software a las necesidades del usuario. La prueba se realizó con una entrevista a la especialista en el manejo de dificultades del aprendizaje, la cual fue grabada con el consentimiento de la misma, se generaron algunas preguntas por parte del usuario, también algunas recomendaciones y observaciones mientras navegaba en la página.

Lista de problemas de usabilidad identificados ene le sistema evaluado

1. Existen algunas faltas ortográficas en algunas de las instrucciones
2. Las ventanas emergentes deben durar un poco más de tiempo para poder leerlo

3. Faltan algunos datos dentro del registro de un nuevo paciente, esto mejorará la visualización de la ficha médica del mismo.
4. Debería existir un botón para abrir el menú por que al inicio no sabía cómo navegar.

Lista de recomendaciones de diseño

1. Debería colocarse el botón de menú en todas las interfaces para que sea más sencillo navegar.

Análisis e interpretación de los resultados

El usuario disfrutó de la aplicación le pareció muy fácil de utilizar y no tuvo problemas con el diseño de la misma, ya que cumplía con los requerimientos no funcionales que se habían obtenido en las entrevistas iniciales del desarrollo de la aplicación.

El usuario mencionó algunas recomendaciones para mejorar el diseño de la aplicación y que el usuario pueda tener una mejor experiencia.

El usuario considera que mientras la aplicación vaya avanzando se puede colocar una opción premium para qué los usuarios tengan acceso a test de la especialista, con ello pueden identificar los problemas de los niños desde casa.

Necesidades de información del usuario en el sistema evaluado

El usuario no tenía claro cómo navegar en la aplicación, por lo tanto, se tuvo que intervenir y explicarle que para ingresar al menú debe deslizar de izquierda a derecha.

Resultado del método del conductor

Este método del conductor permitió descubrir las necesidades de información y de diseño que tiene el usuario en el sistema, permitiendo analizar el modelo analítico y mental que tiene el mismo cuando utiliza el programa, además, ayudó a detectar las razones por las cuales que

se cometieron algunos errores y se lograron corregirlas, antes de entregar el producto final al usuario.

5.7. Resultado final de la aplicación

A continuación, se describe el resultado final de la aplicación con todos los errores de usabilidad corregidos y entregados al usuario. Este se describe como un manual para el usuario que le ayudará a dirigirse correctamente y navegar correctamente en la aplicación.

Gráfico 67 *Resultado final interfaz de inicio de sesión*

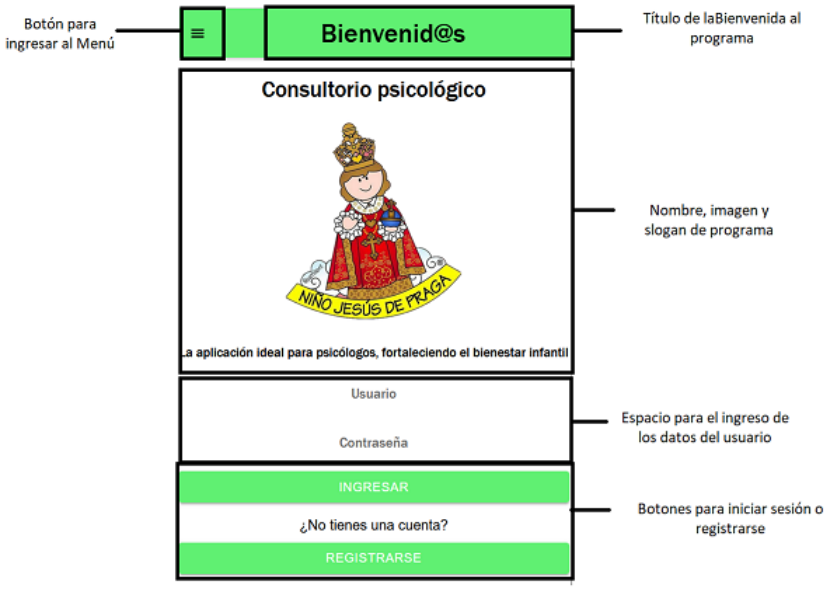


Gráfico 68 *Resultado final interfaz Registrarse*

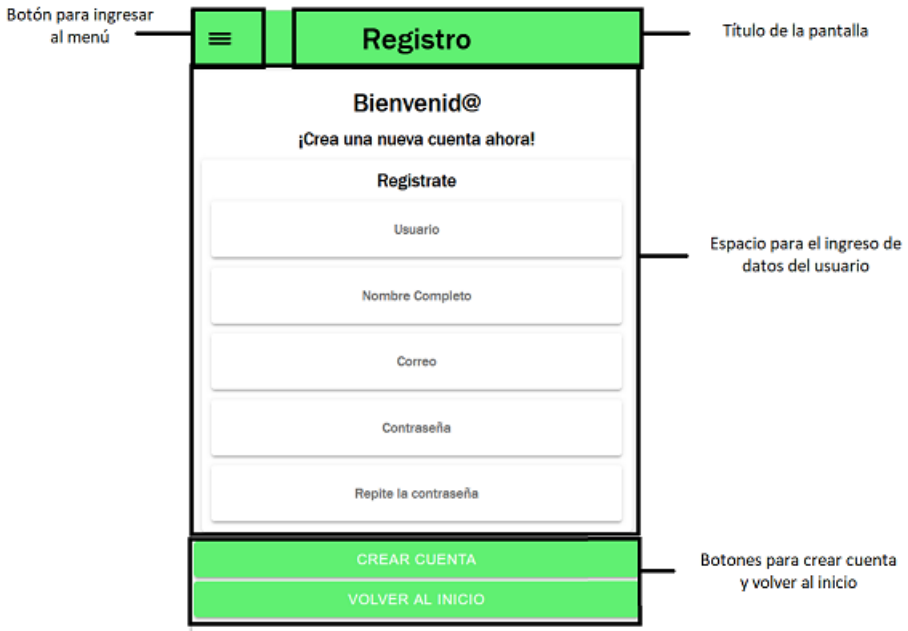


Gráfico 69 *Resultado final interfaz Acuerdo del usuario*

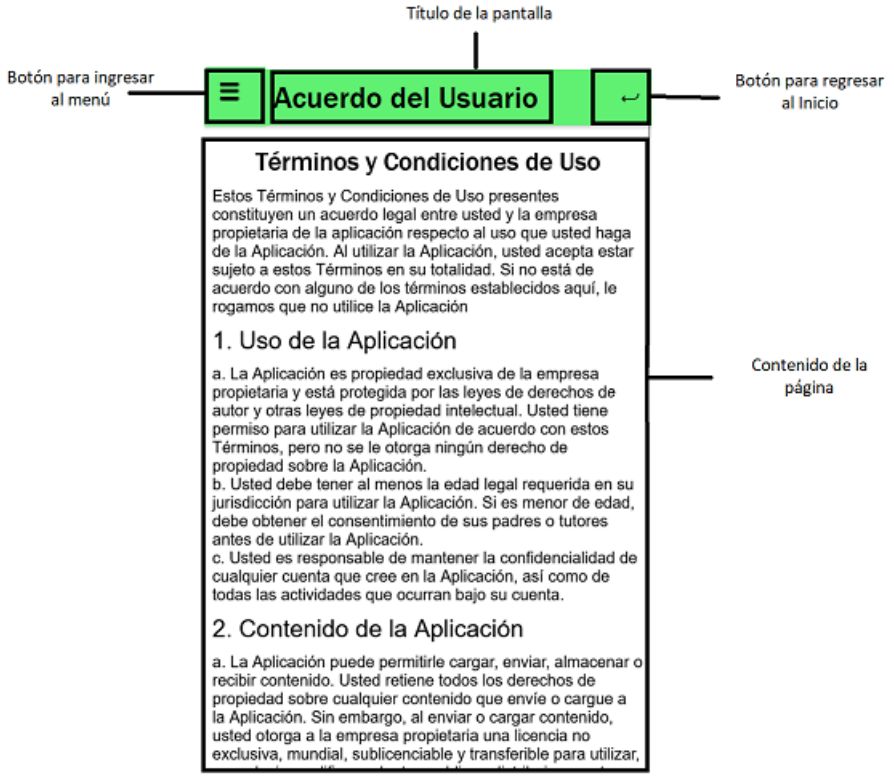


Gráfico 70 *Resultado final interfaz política de privacidad*

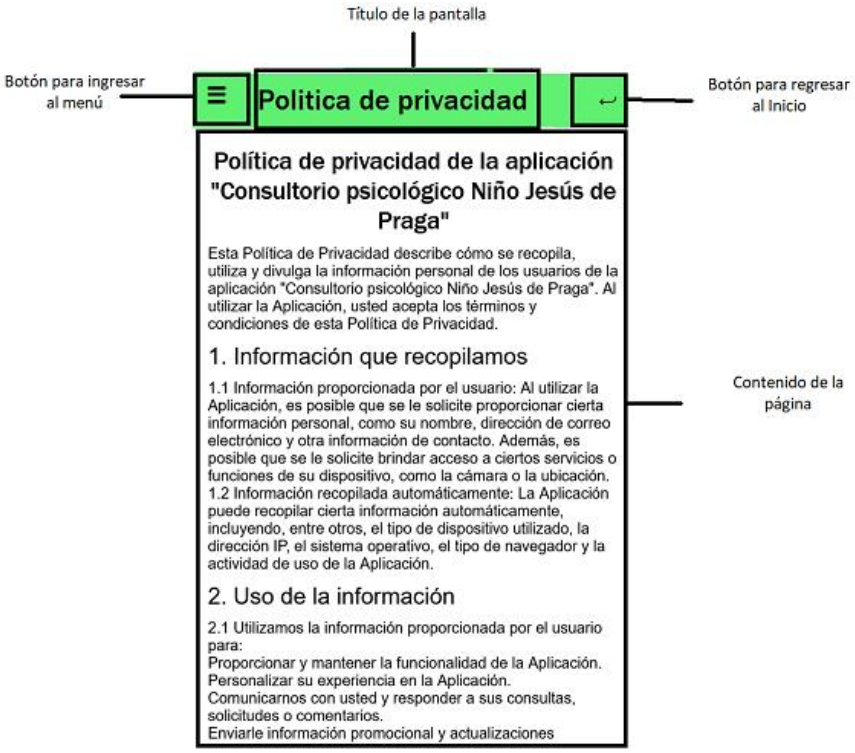


Gráfico 71 *Resultado final interfaz Ayuda y apoyo*

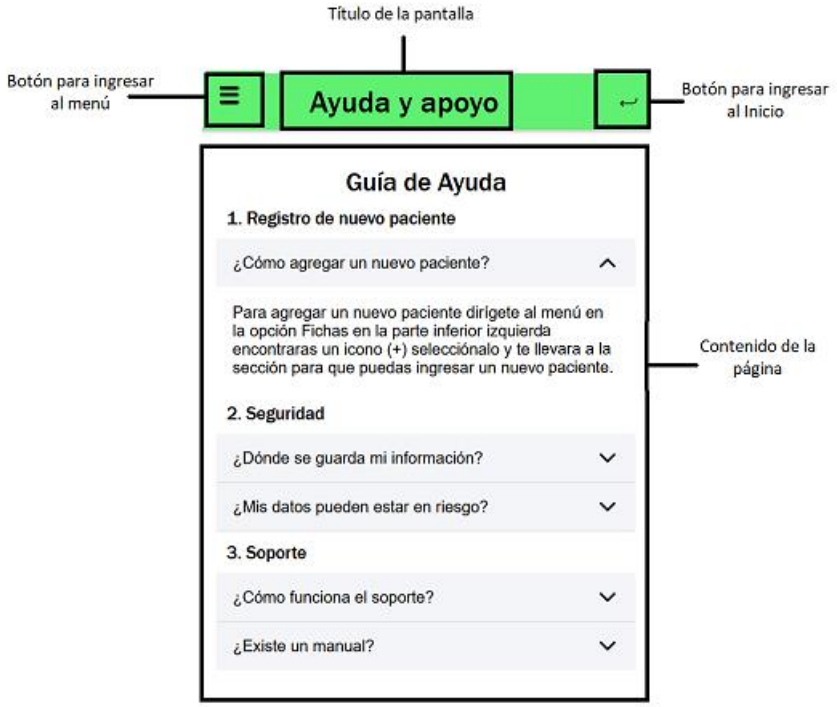


Gráfico 72 *Resultado final de menú externo*

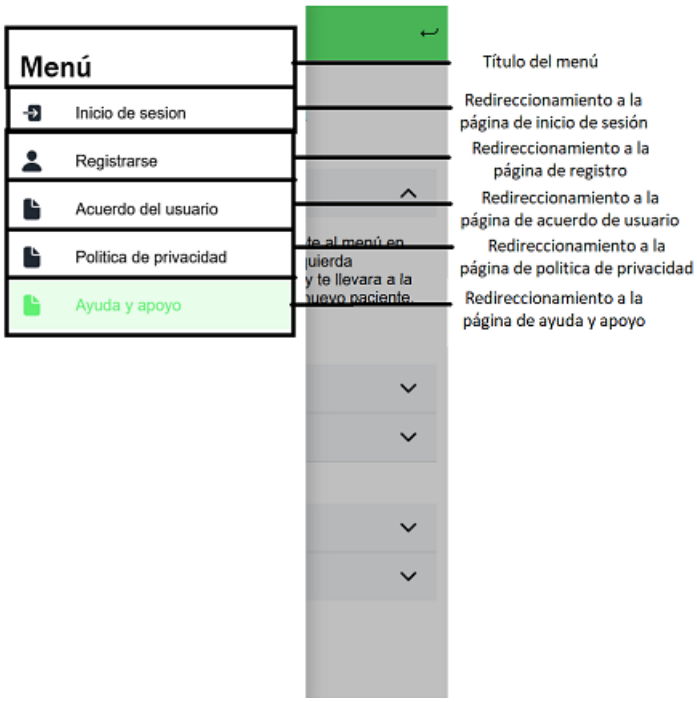


Gráfico 73 *Resultado final de interfaz inicio*

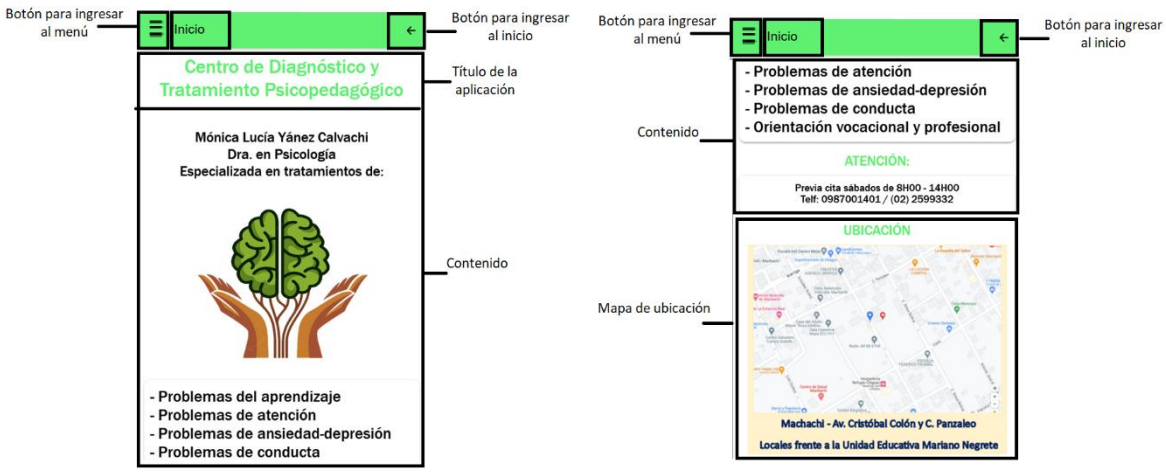


Gráfico 74 *Resultado final interfaz de perfil*

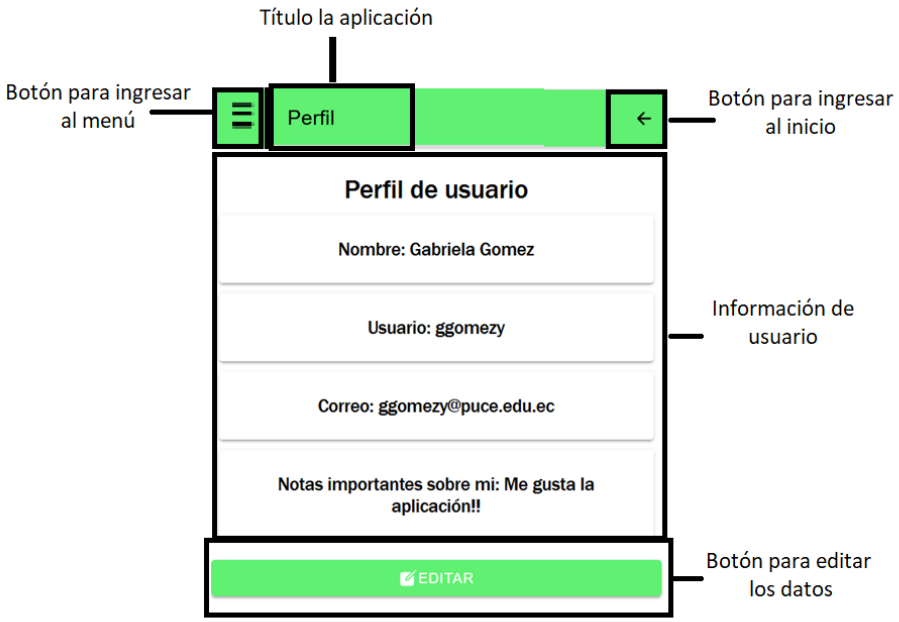


Gráfico 75 *Resultado final interfaz fichas*

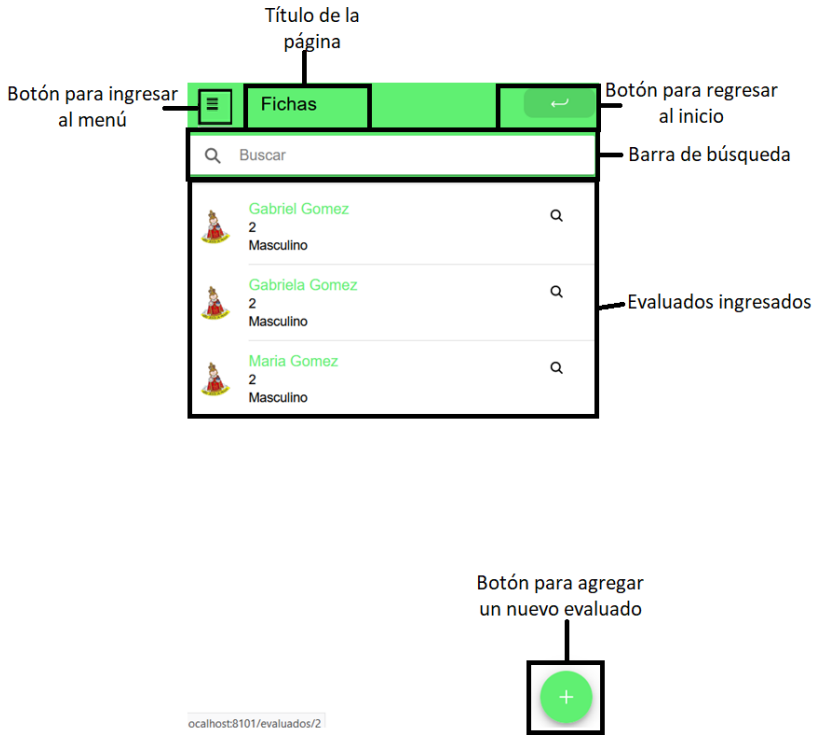


Gráfico 76 *Resultado final de interfaz para ingresar un nuevo evaluado*



Gráfico 77 *Resultado final de la interfaz de Ejercicios*

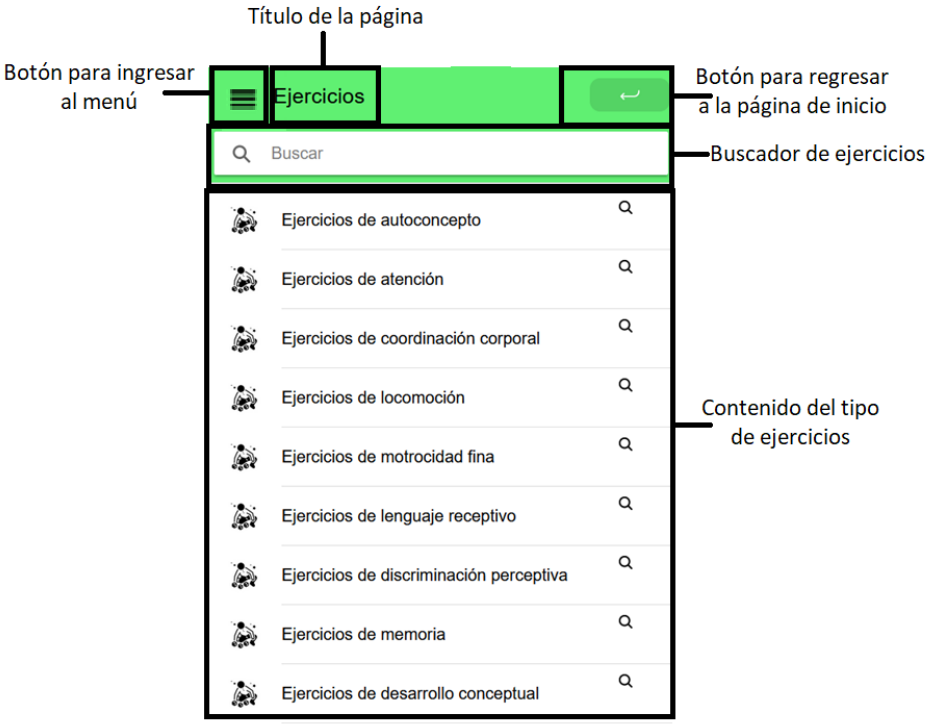


Gráfico 78 *Resultado de la interfaz tipo de usuario*

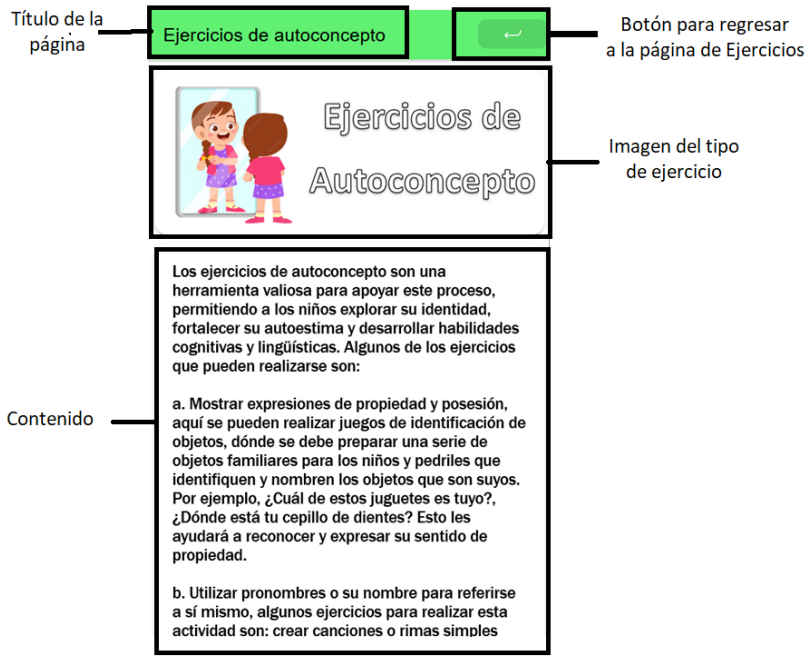
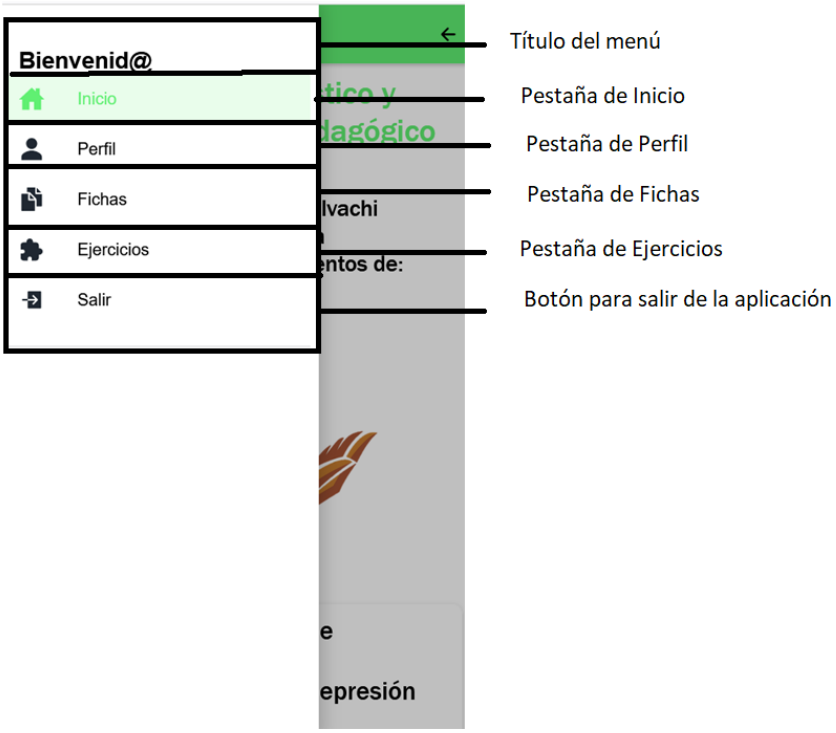


Gráfico 79 *Resultado final de la interfaz de menú interno*



CONCLUSIONES Y RECOMENDACIONES

Conclusiones

- A lo largo del desarrollo de la aplicación móvil sobre la estimulación temprana del pensamiento en niños ecuatorianos entre 2 a 4 años, se concluye que la tecnología dentro de un consultorio de estimulación del desarrollo infantil puede tener una gran influencia debido a que se lo puede utilizar como herramienta para fortalecer los ejercicios determinados en la consulta con la especialista.
- La aplicación móvil con ejercicios para el desarrollo en niños de 2 a 4 años ofrece varias ventajas como la accesibilidad y disponibilidad en cualquier momento, esto logra que los pacientes refuercen las áreas que se diagnosticaron como débiles de una manera rápida y sencilla.
- Después de analizar algunas opciones para el desarrollo de la aplicación, se concluye que Ionic, Api Rest y el administrador de la base de datos Microsoft SQL Server son una combinación efectiva debido a que la instalación y uso de las mismas fue fácil y sencillo, lo que ayudó a que la aplicación se desarrollara de manera fluida y sin muchas complicaciones.

Recomendaciones

- Se recomienda que, en el futuro, la aplicación cuente con una opción premium, dónde los que deseen adquirirla puedan observar y desarrollar los test que la especialista suele aplicar en su consultorio y que estos brinden un diagnóstico general del evaluado.
- Añadir un diagrama del proceso del paciente y que este se pueda exportar a Excel para el conocimiento de los padres de familia o responsables del niño evaluado.

- Expandir la aplicación a no solo tener ejercicios para niños de 2 a 4 años, sino para niños de otras edades y otras áreas de especialización que tiene la profesional.

BIBLIOGRAFÍA

Almagro, C. A. (2011, Diciembre). *Universidad de Granada.* Departamento de Lenguajes y Sistemas Informáticos: https://lsi.ugr.es/curena/doce/lp/tr-11-12/lp-c01-impr.pdf

angular.io. (2020). *angular.io.* https://angular.io/docs

Aristizábal, N. (2009). *Técnicas de Levantamiento de Requerimientos con.* Universidad Autónoma de Bucaramanga UNAB.

Auth0® Inc. (2013 - 2020). *https://jwt.io/.* https://jwt.io/introduction/

AWS. (2023). *¿Qué es la API RESTful?* https://aws.amazon.com/es/what-is/restful-api/

Axessnet. (2020, 12 06). *AXESSNET.* https://axessnet.com/como-funciona-el-internet-via-satelite-enlace-satelital/

Benítez, L. F. (2016). *La implementación del mobile marketing como herramienta multidisciplnar en el sector turístico y aeroportuario.* Mágala.

Bootstrap Community. (2020). *getbootstrap.com.* https://getbootstrap.com/docs

C. Xia, G. Y. (2009). Efficient Implement of ORM (Object/Relational Mapping) Use in J2EE Framework: Hibernate. *Efficient Implement of ORM (Object/Relational Mapping) Use in J2EE Framework: Hibernate*. Wuhan, Hubei, China: International Conference on Computational Intelligence and Software Engineering.

Codina, M., Parra, J., Tello, B., Rivadeneira, M., Medina, D., & Castellanos, J. (2022). *Reportes de la ENSANUT 2018 Volimen No 2. Desarrollo Infantil.* Quito: INEC.

Developers. (2023). *Introducción a Android Studio*. https://developer.android.com/studio/intro?hl=es-419

Eduardo Polo Ortega, F. J. (2015). *Servicios de red e Internet.* Madrid, España: RA-MA Editorial.

Enriquez, J., & Casas, S. (2013). USABILIDAD EN APLICACIONES MÓVILES. *Informes científicos técnicos-UNPA*, 25-47.

Enriquez, J., y Casas, S. (2013). USABILIDAD EN APLICACIONES MÓVILES. *Informes científicos técnicos-UNPA*, 25-47.

Flores, F. (2022). *Qué es Visual Studio Code y qué ventajas ofrece*. https://openwebinars.net/blog/que-es-visual-studio-code-y-que-ventajas-ofrece/

G., E. C. (2011, 08 18). *slideshare.net.* https://es.slideshare.net/edisoncoimbra/71-redes-por-satlite-sh

Galeano, R. (2008). Diseño centrado en el usuario. *Revista Q*, 6-7.

Garcia, M. (2020). *¿QUE ES XAMPP Y COMO PUEDO USARLO?* https://www.nettix.com.pe/blog/web-blog/que-es-xampp-y-como-puedo-usarlo/

GilJin Yang, B. C. (2014, 01). *Research Gate.* ww.researchgate.net/publication/298642533_Implementation_of_HTTP_live_streaming_for_an_IP_camera_using_an_open_source_multimedia_converter

Grasa, J. M. (2017, 17 10). Acceso a Internet vía satélite. En J. Mora, *Guías de Tecnología fácil* (pág. 24). Madrid: Asociación española ingenieros de telecomunicación. http://www.coitaoc.org/files/estudios/tecnologia_facil_7aba8393.pdf

Graydon, M. &. (2019, August 7). *'Connecting the unconnected': a critical assessment of US satellite Internet services.* SAGE JOURNALS: https://doi.org/10.1177/0163443719861835

Grijalva, N. (2012, 10 15). *blogspot*. http://sofware1nathalygrijalva.blogspot.com/2012/10/modelo-espiral.html

Guniganti, R. &. (2013). A Comparison of RTMP and HTTP Protocols with respect to Packet Loss and Delay Variation based on QoE. *semanticscholar.org*.

Gutiérrez, J. J. (2014, 05 12). *Qué es un framework web.* http://www. lsi. us. es/~ javierj/investigacion_ficheros/Framework

I. Fette, A. M. (2011, December). *Internet Engineering Task Force (IETF)* . https://www.hjp.at/doc/rfc/rfc6455.html

Ibañez, P., Mudarra, M., y Ibañez, A. (2004). La estimulación psicomotriz en la infancia a través del método estitsológico multisensorial de atención temprana. *Educación XXI, 7*, 111-133.

Instituto Nacional de Estadísticas y Censos. (2019). *www.ecuadorencifras.gob.ec.* https://www.ecuadorencifras.gob.ec/documentos/web-inec/Estadisticas_Sociales/TIC/2019/201912_Principales_resultados_Multiproposito_TIC.pdf

Irontec. (2022). *Curso de desarrollo de Aplicaciones móviles con Ionic Framework*. https://www.irontec.com/cursos/curso-de-desarrollo-de-aplicaciones-moviles-con-ionic-framework

Joskowicz, J. (2008, 02 10). Reglas y prácticas en eXtreme Programming. Universidad de Vigo, 22. https://d1wqtxts1xzle7.cloudfront.net/31398587/xp_-_jose_joskowicz-with-cover-page-v2.pdf?Expires=1625441610&Signature=HZfoDu6RCpoB-dKMMuLRNmZRaiz0cWUrcjbndtRyjECrK33QWDAGtlNDg1Cnw9kvQJ9Psul9gXX8CRJculpI5KBhJgUMc~bIqu72mdVo6cpvqyy3-XejGZUvukkePQRHmxpb-Ddq

Lagatree, K. (2006). Keep it Together. En K. Lagatree, *Keep It Together: 200+ Tips, Tricks, Lists, and Solution for EverydayLife* (pág. 432). Random House Reference.

Ley De Comercio Electrónico, Ley 67 (Congreso Nacional 05 17, 2002).

Ley Orgánica De Comunicación, 22 (Legislativo 06 25, 2013).

López, M. (2020). *Qué es un lenguaje de programación*. https://openwebinars.net/blog/que-es-un-lenguaje-de-programacion/

Maria. (2019, 04 24). *instalacionestk.com*. https://www.instalacionestk.com/conoce-las-ventajas-y-desventajas-del-internet-satelital/

Maza, M. Á. (2012). *javascript Certificado de profesionalidad.* Innovación Y Cualificación.

Microsoft. (2023). *What is SQL Server Management Studio (SSMS)?* https://learn.microsoft.com/en-us/sql/ssms/sql-server-management-studio-ssms?view=sql-server-ver16

Ministerio de Inclusión Económica y Social. (2022). *Desarrollo Infantil Integral*. https://www.inclusion.gob.ec/desarrollo-infantil-integral/

Morillo, A., Aguavil, J., Garcia, B., & Coello, B. (2022). *La estimulación temprana como estrategia para el desarrollo evolutivo.* Ciudad de Mexico: Ciencia Latina Revista Científica Multidisciplinar.

Morillo, A., Aguavil, J., Garcia, B., y Coello, B. (2022). *La estimulación temprana como estrategia para el desarrollo evolutivo.* Ciudad de Mexico: Ciencia Latina Revista Científica Multidisciplinar.

Muñoz, J. (2006, 07 25). *maestrosdelweb.* www.maestrosdelweb.com/intersatelite/

Oracle Corporation and/or its affiliate. (2020). *What is MySQL.* https://dev.mysql.com/doc/refman/8.0/en/what-is-mysql.html

Oracle. (2023). *Qué es una base de datos?* https://www.oracle.com/mx/database/what-is-database/

Ordoñez, J. L. (2009). *Videoconferencia.* Mexico: Alfaomega Grupo Editor.

Pérez Ibarra, S. G. (2021, 06 17). *Red de Universidades con Carreras en Informática.* http://sedici.unlp.edu.ar/handle/10915/120476

Pérez, M. (2011). *Microsoft SQL Server 2008 R2. Motor de base de datos y administración.* RC Libros.

Pérez, M. d. (2005). *Ingeniería de Requerimientos.* Pachuca: Universidad Autónoma del Estado de Hidalgo.

PRIMICIAS. (2022). *Montuvios e indígenas solo estudian siete años de educación básica*. https://www.primicias.ec/noticias/sociedad/educacion-escuelas-estudiantes-inec-ecuador/

PUCE. (2017, 07 10). *LA ESTACIÓN CIENTÍFICA YASUNÍ*. http://www.yasuni.ec

Real Academia Española. (2021). REAL ACADEMIA ESPAÑOLA.

Red5. (2020, 12 06). *Red5pro.com*. Red5.org: http://red5pro.com/

Richard, A. (2017). Can Weather Affect Satellite Internet? *Hearst Newspapers*, 1. Can Weather Affect Satellite Internet?: http://smallbusiness.chron.com/can-weather-affect-satellite-internet-26822.html

Sáez, J. (2021). *Cómo funciona la Metodología Scrum: Qué es y cómo utilizarla*. https://www.iebschool.com/blog/metodologia-scrum-agile-scrum/

Sampieri, R. F. (2014). Definiciones de los enfoques cuantitativo y cualitativo, sus similitudes y diferencias. En C. F. Roberto Hernández Sampieri. RH Sampieri, Metodología de la Investivación.

Sommerville, I. (2005). *Ingeniería de Softwre.* Pearson Educación.

Taylor Otwell. (2011-2020). *Laravel.* https://laravel.com/

The PHP Group. (2001-2020). *php.net.* https://www.php.net/manual/es/intro-whatis.php

UNICEF. (2022). *ESTUDIO DE CONOCIMIENTOS, APTITUDES Y PRÁCTICAS SOBRE LA PERCEPCIÓN DE LA PRIMERA INFANCIA.* UNICEF.

Universidad de Granada DECSAI. (s.f.). *Especificación de requerimientos Diseño de bases de datos*. https://elvex.ugr.es/idbis/db/docs/design/2-requirements.pdf

Universidad Europea. (2022). *¿Qué es un prototipo y para qué sirve?* https://universidadeuropea.com/blog/que-es-prototipo/

Vargas, D. (2023). *Qué es Node.js: Casos de uso comunes y cómo instalarlo*. https://www.hostinger.es/tutoriales/que-es-node-js

VmWare. (2023). *¿Qué es la implementación de aplicaciones?* https://www.vmware.com/es/topics/glossary/content/application-deployment.html

Yolanda Martínez, S. d. (2015, 06). *Triplemente marcadas: Desconexiones comunicativas en la Amazonia sur ecuatoriana.* Cuenca: Universidad de Cuenca. Triplemente marcadas: Desconexiones comunicativas en la Amazonia sur ecuatoriana: http://www.scielo.cl/scielo.php?script=sci_arttext&pid=S0719-367X2015000100007

ANEXOS

Anexo A: Primera entrevista con el usuario

ENTREVISTA 1

Gabriela Gómez: ¿Qué necesita en el inicio donde le ingresarán los datos al niño?

Cliente: Fecha de aplicación, nombre y apellido del niño, fecha de nacimiento, edad, nombre de la institución a la que pertenece.

Gabriela Gómez: ¿y de los padres?

Cliente: No es tan relevante, de ahí motivo de consulta, pero eso sería historia clínica, hasta ahí los datos.

Gabriela Gómez: ¿Necesita que el niño tenga una historia clínica?

Cliente: Si, dentro de la historia clínica va el motivo de consulta, historia familiar (nombre de los padres, con quién vive, las relaciones con los padres), historia personal (análisis desde el embarazo hasta la edad que tiene el niño, aquí se analiza el desarrollo, psicológico, motriz entre otras). Luego de eso viene la aplicación de los test, aquí se coloca un test de inteligencia o test de aptitudes podría ser, el resultado del test que va a salir en la aplicación debería mostrar las áreas que está debilitadas.

Gabriela Gómez: Entonces, primero vendría llenar la historia clínica del niño, luego el test y luego…

Cliente: Y luego sale el diagnostico que debe mostrar que áreas no tiene problemas y qué áreas están debilitadas y de acuerdo a las áreas debilitadas viene la aplicación de los ejercicios para fortalecer esas áreas debilitadas.

Gabriela Gómez: ¿Qué colores le gustaría que tenga su aplicación?

Cliente: Para los niños y su concentración es importante el color amarillo o el rojo

Gabriela Gómez: ¿Algún tipo de letra especial que necesite?

Cliente: No, solo que sea claro y legible

Gabriela Gómez: En la aplicación necesita algún color específico, imágenes, ¿en la hoja clínica del chico necesita alguna foto?

Cliente: En el momento que iniciamos los datos del niño estaría bien colocar una foto, en el test si habrá imágenes, podrían ser blanco, negro, amarillo, verde.

Gabriela Gómez: Para el inicio de la pantalla, ¿qué colores le gustaría tener?

Cliente: Verde lima con violeta, amarrillo, recomiéndame colores.

Gabriela Gómez: Listo en la próxima reunión le recomiendo colores. ¿Alguna cosa adicional que desee agregar?
Cliente: Por el momento no.
Gabriela Gómez: Gracias.