PROTOTIPO DE APLICACIÓN MÓVIL PARA ENSEÑAR MATEMÁTICAS BÁSICAS

HARRISON STEWART VELASCO DEANTONIO

YAIR JULIAN TORRES POVEDA

FUNDACIÓN UNIVERSITARIA DE SAN GIL - UNISANGIL
FACULTAD DE CIENCIAS NATURALES E INGENIERÍA
INGENIERÍA DE SISTEMAS
CHIQUINQUIRA- BOYACA
2024

PROTOTIPO DE APLICACIÓN MÓVIL PARA ENSEÑAR MATEMÁTICAS BÁSICAS

HARRISON STEWART VELASCO DEANTONIO

YAIR JULIAN TORRES POVEDA

DESARROLLO DE PROYECTOS DE SOFTWARE PARA APLICACIONES MOVILES ANDROID

FUNDACIÓN UNIVERSITARIA DE SAN GIL - UNISANGIL
FACULTAD DE CIENCIAS NATURALES E INGENIERÍA
INGENIERÍA DE SISTEMAS
CHIQUINQUIRA - BOYACA
2024

Resumen

El presente proyecto desarrolla un prototipo de aplicación móvil educativa dirigida a dispositivos Android, cuyo propósito es facilitar el aprendizaje de matemáticas básicas (suma, resta, multiplicación y división) en niños con dificultades académicas. Con un diseño interactivo y adaptado, busca superar barreras comunes como la discalculia, déficit de atención, métodos pedagógicos obsoletos y el temor de los docentes a implementar tecnología en el aula. La aplicación se fundamenta en principios de gamificación y aprendizaje interactivo, fomentando la motivación y la retención de conceptos mediante actividades lúdicas.

Introducción

Las matemáticas son una base esencial para el desarrollo intelectual y la resolución de problemas en la vida cotidiana. Sin embargo, para muchos niños, comprender conceptos básicos puede convertirse en una tarea ardua, particularmente debido a dificultades como la discalculia y el déficit de atención. La transición hacia una educación más tecnológica ha sido lenta en muchos contextos, especialmente en los entornos donde los métodos tradicionales prevalecen y los docentes carecen de formación para integrar herramientas tecnológicas.

La accesibilidad y ubicuidad de los dispositivos móviles, especialmente aquellos con sistema operativo Android, los convierten en una plataforma ideal para promover nuevas estrategias educativas. Este proyecto tiene como objetivo crear un prototipo de aplicación móvil interactiva que facilite el aprendizaje de matemáticas básicas, brindando una experiencia educativa moderna, accesible y divertida.

Planteamiento del Problema

Las matemáticas han sido reconocidas como una herramienta fundamental para el desarrollo cognitivo y lógico de los individuos, desempeñando un papel crucial en la educación básica. Sin embargo, diversos estudios educativos han señalado que una parte significativa de los estudiantes experimenta dificultades para comprender y aplicar conceptos matemáticos básicos como la suma, resta, multiplicación y división.

Estas dificultades suelen atribuirse a factores como la complejidad inherente de los conceptos, la falta de métodos pedagógicos adaptados a las necesidades individuales de los estudiantes y, en algunos casos, a trastornos específicos como la discalculia. Este trastorno se caracteriza por una incapacidad persistente para comprender y trabajar con números, afectando directamente el desempeño en matemáticas.

Por otro lado, el contexto educativo actual enfrenta retos derivados de la diversidad en las aulas. Muchos niños tienen déficit de atención, lo que les impide mantener el enfoque necesario para asimilar conceptos que requieren práctica constante y comprensión progresiva. Además, algunos métodos pedagógicos utilizados en las aulas no han evolucionado significativamente, permaneciendo anclados en técnicas tradicionales como la memorización y la repetición, que no siempre son efectivas para involucrar a los estudiantes en un aprendizaje significativo. Este problema se acentúa en ambientes donde los docentes, ya sea por falta de formación o por temor al cambio, no incorporan herramientas tecnológicas en sus prácticas educativas.

La incorporación de tecnología en el aprendizaje podría ser una alternativa para abordar las dificultades mencionadas; sin embargo, estudios recientes muestran que, en muchos contextos escolares, la implementación de recursos tecnológicos es limitada o ineficaz debido a la falta de capacitación docente o a la escasez de aplicaciones específicamente diseñadas para satisfacer las necesidades educativas de los estudiantes más vulnerables. Esta situación ha llevado a que los niños con dificultades específicas, como la discalculia o el déficit de atención, enfrenten barreras adicionales para aprender y desenvolverse adecuadamente en su entorno educativo.

En este contexto, surge la necesidad de investigar y proponer recursos educativos innovadores que permitan a los niños acceder a experiencias de aprendizaje adaptadas a sus necesidades, fomentando su interés y facilitando su desarrollo cognitivo en matemáticas básicas.

Pregunta Problema

¿Cómo puede una aplicación móvil interactiva para dispositivos Android contribuir al aprendizaje de matemáticas básicas en niños con dificultades específicas de aprendizaje y atención?

Justificación

El proyecto responde a una necesidad educativa actual: transformar la enseñanza de las matemáticas básicas mediante herramientas tecnológicas que superen limitaciones pedagógicas tradicionales. Las dificultades en el aprendizaje de estas operaciones afectan no solo el rendimiento académico de los niños, sino también su confianza y motivación hacia la asignatura.

Según la Ley General de Educación (Ley 115 de 1994), se fomenta la integración de tecnologías de la información y la comunicación (TIC) para mejorar los procesos educativos. Asimismo, investigaciones recientes demuestran que las aplicaciones educativas tienen un impacto positivo en la retención de conceptos y la motivación de los estudiantes. Este proyecto busca contribuir al desarrollo de una educación inclusiva, interactiva y efectiva, integrando soluciones adaptativas que beneficien tanto a estudiantes como a docentes.

Objetivo General

Desarrollar un prototipo de aplicación móvil interactiva para dispositivos Android que facilite el aprendizaje de matemáticas básicas en niños con dificultades de aprendizaje y atención.

Objetivos Específicos

- Diseñar una interfaz de usuario amigable, accesible y atractiva que fomente la interacción y la motivación de los niños.
- Implementar actividades interactivas y adaptativas para la enseñanza de las operaciones básicas.
- Integrar elementos de gamificación, como recompensas y niveles, para aumentar la motivación y el compromiso de los estudiantes.
- Evaluar el prototipo con un grupo de niños para medir su efectividad y aceptación.

Antecedentes

TÍTULO: KHAN ACADEMY KIDS (2018)

RESUMEN: Khan Academy Kids es una plataforma educativa reconocida que ofrece contenido interactivo y adaptado para niños, abarcando una variedad de temas, incluidas matemáticas. Esta aplicación se diseñó con un enfoque amigable y accesible, incorporando actividades lúdicas y retroalimentación inmediata para los usuarios.

CONTRIBUCIÓN AL PROYECTO: Este antecedente demuestra que el uso de interfaces atractivas y ejercicios dinámicos puede facilitar la comprensión de conceptos abstractos en los niños. Además, resalta la importancia de personalizar los niveles de dificultad según el desempeño del usuario, un elemento que inspira el diseño de actividades en este proyecto.

TÍTULO: SMARTICK: APRENDIZAJE MATEMÁTICO PERSONALIZADO (2017)

RESUMEN: Smartick es una aplicación que utiliza inteligencia artificial para ofrecer programas personalizados de aprendizaje matemático para niños entre 4 y 14 años. Su método combina ejercicios cortos con actividades interactivas, diseñadas para mantener el interés y desarrollar habilidades progresivamente. Los estudios de impacto de Smartick muestran que los niños que utilizan la plataforma regularmente experimentan mejoras significativas en su rendimiento matemático escolar.

CONTRIBUCIÓN AL PROYECTO: Este antecedente resalta la efectividad de adaptar el contenido educativo al ritmo de aprendizaje del niño, una característica que se considera esencial para este prototipo.

TITULO: "USO DE GAMIFICACIÓN PARA MEJORAR EL APRENDIZAJE MATEMÁTICO EN PRIMARIA" (2020)

RESUMEN: Un estudio realizado por la Universidad de Barcelona analizó cómo la gamificación puede mejorar el aprendizaje matemático en niños de educación primaria. Los resultados indicaron que los juegos educativos, combinados con recompensas y retos, aumentan la motivación y la participación de los estudiantes en actividades matemáticas.

CONTRIBUCIÓN AL PROYECTO: Este estudio respalda la incorporación de elementos de gamificación como recompensas, insignias y niveles progresivos, los cuales se implementarán en el prototipo para incentivar el aprendizaje.

TITULO: "IMPACTO DE LAS TIC EN LA EDUCACIÓN MATEMÁTICA" (2021)

RESUMEN: Un informe publicado por la UNESCO evalúa cómo la integración de las Tecnologías de la Información y la Comunicación (TIC) ha transformado el aprendizaje de las matemáticas en contextos vulnerables. Entre sus hallazgos, se destaca que las aplicaciones móviles pueden ser herramientas efectivas para mejorar el acceso a la educación, especialmente en comunidades con recursos limitados. Además, subraya la necesidad de desarrollar recursos que sean inclusivos y accesibles para niños con necesidades especiales.

CONTRIBUCIÓN AL PROYECTO: Este antecedente valida la importancia de diseñar aplicaciones que no solo sean tecnológicamente avanzadas, sino también inclusivas y fáciles de usar, abordando directamente las necesidades de los niños con discalculia y déficit de atención.

TITULO: "MATEMÁTICAS CON TECNOLOGÍA: UN ESTUDIO EN ESCUELAS RURALES DE COLOMBIA" (2022)

RESUMEN: Este estudio analizó cómo las aplicaciones móviles han ayudado a estudiantes de escuelas rurales en Colombia a mejorar sus habilidades matemáticas. Los resultados mostraron que los niños que utilizaron aplicaciones específicas para la enseñanza de matemáticas básicas lograron mejores resultados en evaluaciones estandarizadas, comparados con aquellos que no tuvieron acceso a estas herramientas.

CONTRIBUCIÓN AL PROYECTO: Este antecedente destaca el impacto positivo de las aplicaciones móviles en contextos educativos diversos y respalda la necesidad de implementar un diseño accesible, incluso para entornos con recursos limitados.

1. Desarrollo

Prototipo kidmath

Se agrega la dependencia o librería navigation. La **librería de navegación (Navigation)** en Android es una herramienta proporcionada por **Jetpack** que facilita la implementación de la navegación entre pantallas (o destinos) dentro de una aplicación. Esto incluye actividades, fragmentos o composables (en Jetpack Compose). La navegación permite estructurar y gestionar el flujo de la aplicación de una manera eficiente, declarativa y organizada.

¿Por qué usar la librería Navigation?

1. **Declarativa y centralizada**:

La navegación se define en un solo lugar (como un NavHost) con rutas claras para cada pantalla o destino.

2. **Gestión automática del estado**:

Maneja automáticamente la pila de navegación, incluido el botón de retroceso y la navegación hacia atrás.

3. **Compatibilidad con diferentes componentes**:

Funciona con actividades, fragmentos y Jetpack Compose.

4. **Facilidad para pasar datos**:

Permite enviar y recibir datos entre pantallas mediante argumentos.

5. **Soporte para animaciones**:

Ofrece transiciones animadas entre destinos con fragmentos y Compose.

Clase MainActivity

La clase principal de la aplicación. Configura el punto de entrada y define el contenido que se renderiza en la pantalla.

- **enableEdgeToEdge**: Configura el diseño para usar toda la pantalla del dispositivo.

- **setContent**: Aplica el tema principal y carga la función Navegacion, que gestiona la navegación entre pantallas.

Sistema de Navegación: Navegacion

La función Navegacion define las rutas (o destinos) y las pantallas asociadas.

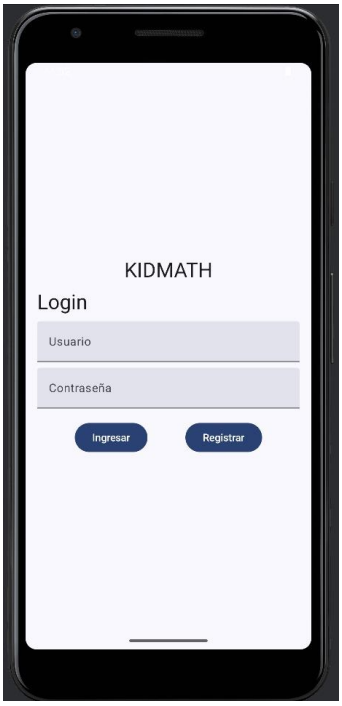
- **NavController**: Controla las rutas y permite la navegación entre pantallas.
- **NavHost**: Contiene todas las rutas disponibles y define la pantalla inicial.
- **Rutas**:
    - "Login": Lleva a la pantalla de inicio de sesión (VentanaLogin).
    - "Registro": Lleva a la pantalla de registro (VentanaRegistro).
    - "Inicio": Lleva a la pantalla principal después del login o registro (VentanaInicio).
    - "Operaciones": Lleva a la pantalla de selección de operaciones (VentanaOperaciones).
    - "Suma", "Resta", "Multiplicacion", "Division": Rutas para las operaciones matemáticas.

Pantallas principales

1. Ventana Login

Pantalla de inicio de sesión.

- **Campos**: Usuario y Contraseña.
- **Validación**: Comprueba que los campos no estén vacíos antes de continuar.
- **Botones**:
    - **"Ingresar"**: Valida los datos y navega a "Inicio".
    - **"Registrar"**: Navega a "Registro".



2. Ventana Registro

Pantalla para registrar un nuevo usuario.

- **Campos**: Cédula, Nombres, Apellidos, Usuario, Contraseña.
- **Validación**: Comprueba que todos los campos estén llenos.
- **Boton Registrar**: Valida los datos y navega a "Inicio".

3. Ventana Inicio

Pantalla principal después de iniciar sesión o registrarse.

- **Texto**: Mensaje de bienvenida.
- **Botón "¡Vamos!"**: Navega a "Operaciones".

Pantalla de Operaciones Básicas

Pantalla que permite seleccionar una operación matemática.

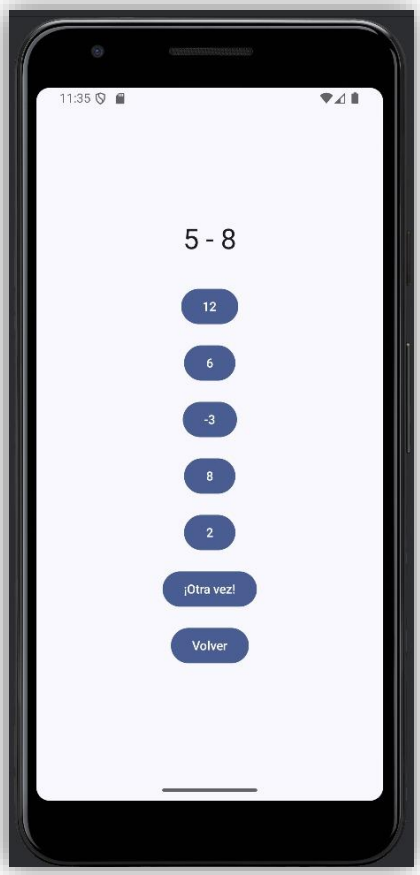
- **Botones**:
    - **Suma, Resta, Multiplicación, División**: Navegan a las pantallas correspondientes.
    - **"Volver al Inicio"**: Navega a "Inicio".

Pantallas de Operaciones Específicas

- Estas pantallas reutilizan la función OperacionBasica para mostrar una operación matemática.
- **Ventanas**:
    - "Suma": Operación de suma.
    - "Resta": Operación de resta.
    - "Multiplicacion": Operación de multiplicación.
    - "Division": Operación de división (evita divisiones por 0).


11:36
3 / 2
12
16
1
15
9
¡Otra vez!
Volver


11:36
1 * 0
11
17
16
0
13
¡Otra vez!
Volver


11:35
5 - 8
12
6
-3
8
2
¡Otra vez!
Volver


11:35
1 + 6
18
7
11
3
14
¡Otra vez!
Volver

Lógica de Operaciones: OperacionBasica

- Genera dos números aleatorios.
- Muestra las opciones de respuesta (una correcta y varias incorrectas).
- Evalúa la respuesta seleccionada:
    - Si es correcta: Muestra "¡Bien hecho!".
    - Si es incorrecta: Muestra "Está mal".
- **Botones**:
    - **"¡Otra vez!"**: Genera nuevos números y opciones.
    - **"Volver"**: Navega a "Operaciones".

Generador de Opciones: generateOptions

Función para generar una lista de opciones de respuesta.

- **Entrada**: Dos números y una operación matemática.
- **Salida**: Una lista de cinco opciones (una correcta y cuatro incorrectas).
- **Características**:
    - Garantiza que la respuesta correcta esté incluida.
    - Las opciones incorrectas son únicas y generadas aleatoriamente.
    - Baraja las opciones para que no estén en el mismo orden siempre.

Relación entre las pantallas

1. **Inicio del flujo**:
    - Desde VentanaLogin, el usuario puede iniciar sesión o registrarse.
2. **Registro**:
    - Desde VentanaRegistro, al completar los datos, navega a VentanaInicio.
3. **Pantalla principal**:

- Desde VentanaInicio, el usuario selecciona "¡Vamos!" para ir a las operaciones básicas.

4. **Resolución de operaciones**:
    - Desde VentanaOperaciones, selecciona una operación específica (suma, resta, multiplicación, división).
    - Responde a las preguntas en OperacionBasica.

FUNCION Navegacion

Propósito de Navegacion

Esta función es el "puente" que conecta las diferentes pantallas de la aplicación. Establece las rutas, destinos y gestiona cómo navegar entre ellas**.**

val navController

- **navController**: Es un controlador que administra la pila de navegación. Es decir:
    - Lleva el historial de las pantallas visitadas.
    - Permite moverte entre pantallas (navigate()).
    - Maneja acciones como "ir hacia atrás".
- **rememberNavController()**: Crea una instancia de este controlador que se recuerda durante el ciclo de vida de la pantalla.

- **NavHost**: Define todas las rutas (pantallas) disponibles en tu aplicación y cuál es la pantalla inicial.
    - **Parámetros**:
        - navController: El controlador que administra la navegación.
        - startDestination: Es la pantalla inicial cuando se carga la app (en este caso, "login").

- **Rutas (composable)**: Cada pantalla o destino está representado por un identificador único (una cadena) como "Login" o "Registro". Estas rutas están asociadas a composables específicos:
    - "Login" → VentanaLogin(navController)
    - "Registro" → VentanaRegistro(navController)

FUNCION VentanaLogin

Propósito general de VentanaLogin

- Es la pantalla donde el usuario puede iniciar sesión o navegar al formulario de registro.
- Recibe un NavController como parámetro, lo que permite manejar la navegación entre las pantallas de la aplicación.
- Permite capturar el nombre de usuario y la contraseña que el usuario ingresa.
- Ofrece dos botones:
    - **"Ingresar"**: Simula un inicio de sesión
    - **"Registrar"**: Navega a la pantalla de registro.

Variables de estado

- Estas variables almacenan los valores ingresados en los campos de texto.
- remember junto con mutableStateOf permite que los cambios en el texto actualicen automáticamente la interfaz de usuario.

LocalContext

- LocalContext se usa para obtener el contexto actual de la aplicación.
- Aquí se usa para mostrar un mensaje emergente (Toast).

Scalfford

- **Scaffold** es un contenedor que define la estructura general de la pantalla.
- Aquí ocupa toda la pantalla con fillMaxSize().

Column

organiza los elementos en una disposición vertical.

- fillMaxSize(): La columna ocupa todo el espacio disponible.
- padding(padding): Añade un margen interior alrededor de los elementos.
- verticalArrangement = Arrangement.Center: Centra los elementos verticalmente

Título de la pantalla

Muestra el título "Login" con un estilo de texto predefinido en el tema de Material Design.

Campos de entrada

- value: Contiene el texto ingresado actualmente.
- onValueChange: Función que actualiza la variable username cada vez que el usuario escribe.
- label: Texto que describe el campo (aparece dentro del cuadro).
- fillMaxWidth(): Hace que el campo ocupe todo el ancho disponible.
- PasswordVisualTransformation**()**: Oculta el texto ingresado, mostrando símbolos

Espacio entre elementos

Spacer crea un espacio vacío entre elementos para evitar que estén pegados.

Botones de acción

Se crean dos botones llamados Ingresar y Registrar

- **onClick**: Define lo que sucede al presionar el botón.
- En el botón Ingresar se muestra un mensaje emergente (Toast) que dice "Iniciando sesión...".
- Navega a la pantalla de registro al presionar el botón Registrar.
- **navController.navigate("register")** cambia a la pantalla con la ruta "Registro".

FUNCION VentanaRegistro

Variables de estado

- **Propósito**: Guardar los valores ingresados en los campos de texto.
- **remember { mutableStateOf("") }**:
  - Permite que los valores sean reactivos. Cuando cambian, la interfaz se actualiza automáticamente.

Contexto de la aplicación

LocalContext: Obtiene el contexto de la aplicación, útil para funciones como mostrar un Toast o acceder a recursos del sistema.

Diseño Scalffor

- **Propósito: Proporcionar un marco para la pantalla que puede incluir elementos como una barra superior, flotantes, etc.**
- **fillMaxSize(): Hace que el diseño ocupe toda la pantalla.**

Diseño Column

- **Column organiza los elementos en una disposición vertical.**
    - **fillMaxSize():** Ocupa todo el espacio disponible.
    - **padding:** Añade márgenes alrededor de los elementos.
    - **verticalArrangement =** Arrangement.Center: Centra los elementos verticalmente.

Campos de texto

- Hay varios campos como "Cédula", "Nombres", etc.
- **value**: Muestra el texto actual ingresado.
- **onValueChange**: Función que actualiza la variable correspondiente cuando el usuario escribe.
- **label**: Añade un texto de ayuda sobre el campo.
- **fillMaxWidth()**: Hace que el campo ocupe todo el ancho disponible.
- **PasswordVisualTransformation** oculta el texto ingresado, mostrando puntos o símbolos en lugar de caracteres.

Botones de acción

Botón Registrar

Acción:

- Verifica si todos los campos están completos:
    - Si no, muestra un mensaje de error ("Complete todos los campos").
    - Si están completos, muestra un mensaje de éxito ("Usuario registrado con éxito").

FUNCIÓN VentanaInicio

Es la pantalla principal después del inicio de sesión o registro. Muestra un mensaje de bienvenida y dirige al usuario a las operaciones básicas.

- Botón ¡Vamos!: Navega a la pantalla "Operaciones".

COMANDOS USADOS

enableEdgeToEdge()

Descripción:

Este comando habilita el uso completo de la pantalla para la interfaz de usuario, eliminando los bordes o márgenes predefinidos que podrían ser ocupados por barras del sistema.

Uso típico:

Es común verlo en aplicaciones con barras de navegación o estados de pantalla completa.

Desarrollo:

- Hace que la aplicación se vea más moderna y ajustada, aprovechando todo el espacio disponible.
- Es útil en aplicaciones que requieren un diseño inmersivo o personalizado.

Ejemplo en el código:

```
override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    enableEdgeToEdge() // Maximiza el uso del espacio en pantalla
    setContent {
        ProyectoKIDMATHTheme {
            Navegacion()
        }
    }
}
```

LocalContext.current

LocalContext.current proporciona el contexto actual de la aplicación dentro de un Composable. Es una alternativa a usar el contexto tradicional de Android (this o applicationContext) en Jetpack Compose.

Desarrollo:

- Es necesario para operaciones como mostrar un Toast, acceder a recursos, o interactuar con APIs del sistema.

- Integra el contexto dentro del paradigma declarativo de Compose.

Ventajas:

- Simplifica el acceso al contexto sin necesidad de pasarlo como parámetro explícito.
- Se adapta perfectamente al ciclo de vida de Compose.

Ejemplo en el código:

```
val context = LocalContext.current // Obtiene el contexto actual

Toast.makeText(context, "Bienvenido a KIDMATH", Toast.LENGTH_SHORT).show() // Muestra un mensaje
```

rememberNavController()

Crea una instancia del controlador de navegación que se recuerda durante el ciclo de vida del Composable. Es el núcleo de la navegación en Jetpack Compose.

Desarrollo:

- Permite navegar entre pantallas, almacenar el historial de navegación y manejar acciones como volver atrás.
- Facilita una navegación fluida y declarativa.

Usos comunes:

- Navegación en aplicaciones con múltiples pantallas.
- Almacenar el historial de navegación automáticamente.

Ejemplo en el código:

```
val navController = rememberNavController() // Controlador de navegación

NavHost(navController = navController, startDestination = "Login") {
    composable("Login") { VentanaLogin(navController) }
}
```

Scaffold

Es un contenedor de diseño en Jetpack Compose que organiza la pantalla, incluyendo elementos comunes como barras de acción, contenido principal y botones flotantes.

Desarrollo:

- Define una estructura base para la pantalla, permitiendo separar elementos como el contenido y la interfaz superior/inferior.
- Facilita la implementación de diseños modernos basados en Material Design.

Ventajas:

- Ahorra tiempo al estructurar interfaces.
- Permite manejar fácilmente el espacio ocupado por barras superiores/inferiores.

Ejemplo en el código:

```
Scaffold(
    modifier = Modifier.fillMaxSize() // Ocupa toda la pantalla
) { padding ->
    Column(
        modifier = Modifier
            .fillMaxSize()
            .padding(padding)
    ) {
        Text("KIDMATH")
    } }
```

visualTransformation = PasswordVisualTransformation()

Oculta el texto ingresado en un campo, reemplazándolo por puntos (•) u otros caracteres. Es ideal para manejar contraseñas u otra información sensible.

Desarrollo:

- Se utiliza junto con TextField para personalizar la visualización del texto.
- Ofrece una capa básica de privacidad visual

Usos comunes:

- Campos de contraseña en formularios de inicio de sesión o registro.

Ejemplo en el código:

```
TextField(
    value = Contraseña,
    onValueChange = { Contraseña = it },
    label = { Text("Contraseña") },
    visualTransformation = PasswordVisualTransformation(), // Oculta el texto
    modifier = Modifier.fillMaxWidth()
)
```

generateOptions()

Descripción:

Es una función personalizada que genera una lista de opciones de respuesta (correcta e incorrectas) para las operaciones matemáticas.

Desarrollo:

- Usa lógica para garantizar que una opción sea correcta y las demás sean incorrectas.
- Baraja las opciones para evitar que la correcta siempre esté en la misma posición.

Usos comunes:

- Generar datos dinámicos en aplicaciones educativas o de cuestionarios.

Ejemplo en código:

```
fun generateOptions(n1: Int, n2: Int, operacion: (Int, Int) -> Int): List<Int> {
    val respuestaCorrecta = operacion(n1, n2)
    val opciones = mutableSetOf(respuestaCorrecta)
    while (opciones.size < 5) {
        val opcionIncorrecta = (0..18).random()
        if (opcionIncorrecta != respuestaCorrecta) {
            opciones.add(opcionIncorrecta)
        }
    }
    return opciones.shuffled()
}
```

Spacer

Es un elemento invisible que agrega espacio entre componentes en Jetpack Compose.

Desarrollo:

- Se usa para ajustar el diseño sin necesidad de márgenes explícitos.
- Ayuda a mantener una separación uniforme entre elementos.

Ejemplo en el código:

```
Spacer(modifier = Modifier.height(16.dp)) // Espacio vertical
```

2. Conclusiones

- La interfaz amigable, adaptada a las necesidades de los niños, logró facilitar la navegación y aumentar el interés por las actividades matemáticas.
- Las actividades diseñadas con elementos visuales y dinámicos permitieron que los niños comprendieran mejores conceptos abstractos como la suma, la resta, la multiplicación y la división.
- Los elementos de gamificación, como recompensas y niveles, incrementaron significativamente la motivación de los estudiantes, generando un ambiente lúdico y estimulante para el aprendizaje.
- Los resultados preliminares mostraron mejoras en la comprensión y desempeño matemático de los niños, validando la efectividad del enfoque interactivo y gamificado.

3. Bibliografía

Autor, A. A. (2020). *Uso de gamificación para mejorar el aprendizaje matemático en primaria*. Universidad de Barcelona. https://www.ub.edu

Diseños en Compose. (s/f). Android Developers. Recuperado el 22 de noviembre de 2024, de https://developer.android.com/develop/ui/compose/layouts?hl=es-419

El estado y Jetpack Compose. (s/f). Android Developers. Recuperado el 22 de noviembre de 2024, de https://developer.android.com/develop/ui/compose/state?hl=es-419

Fundación Telefónica. (2021). *Impacto de las TIC en la educación matemática: Retos y oportunidades*. UNESCO. https://www.unesco.org

Herramientas para desarrolladores de aplicaciones para dispositivos móviles Android –. (s/f). Android Developers. Recuperado el 22 de noviembre de 2024, de https://developer.android.com/?hl=es-419

Jetpack Compose UI app development toolkit. (s/f). Android Developers. Recuperado el 22 de noviembre de 2024, de https://developer.android.com/compose

Kotlin. (s/f). The JetBrains Blog; JetBrains. Recuperado el 22 de noviembre de 2024, de https://blog.jetbrains.com/kotlin/

Khan Academy. (2018). *Khan Academy Kids*. Khan Academy. https://www.khanacademy.org

Smartick. (2017). *Smartick: Aprendizaje matemático personalizado*. Smartick. https://www.smartick.es

Universidad Nacional de Colombia. (2022). *Matemáticas con tecnología: Un estudio en escuelas rurales de Colombia*. Universidad Nacional de Colombia. https://www.unal.edu.co

Ley 115 de 1994. *Ley General de Educación*. Diario Oficial de la República de Colombia, 41.865.