



Verano Científico TECNM 2023



Aplicación Móvil: Frac - Helper

App de fracciones para niños

Nombre: Laura Johana Eugenio Rodríguez

Asesor: Dr. Roberto Ángel Meléndez Armenta

Proyecto: Computo en la Nube para
Principiantes



EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO
NACIONAL DE MÉXICO

Instituto Tecnológico Superior
de Misantla
Dirección General
Subdirección de Vinculación
Departamento de Vinculación

Misantla, Ver., **25/julio/2023**
Asunto: **Carta de Terminación**

RAMIRO RICO MARTÍNEZ
SUBDIRECTOR ACADÉMICO
INSTITUTO TECNOLÓGICO DE QUERÉTARO
PRESENTE

En atención y cumplimiento a la convocatoria del "2º Verano Nacional de Investigación en Ciencia y Tecnología del Tecnológico Nacional de México 2023", se hace constar que la C. **LAURA JOHANA EUGENIO RODRÍGUEZ**, con numero de control 18141058, de la carrera de Ingeniería en Sistemas Computacionales del IT de Querétaro, ha cumplido satisfactoriamente su estancia de investigación en el Instituto Tecnológico Superior de Misantla con el proyecto "**Cómputo en la Nube para Principiantes**", bajo el asesoramiento del Dr. Roberto Ángel Meléndez Armenta, durante el periodo del 19 de junio al 25 de julio de 2023.

Sin otro particular, reciba un cordial saludo.

ATENTAMENTE

Excelencia en Educación Tecnológica®
"Excelencia Académica para un Futuro de Calidad"

FEDERICO PERERA ZAPATA
DEPARTAMENTO DE VINCULACIÓN



C.c.p. Dr. Roberto Ángel Meléndez Armenta. – Docente investigador del ITS de Misantla. - Para su conocimiento.
C.c.p. Archivo
LHD/FPZ*



Km. 1.8 Carretera a Loma del Cojolite, Misantla, Veracruz, C.P. 93821 Tels. 235 323 1545
Ext. 130, e-mail: vin_dmisantla@tecnm.mx | vinculacion@itsm.edu.mx



2023
Francisco
VILLA

Introducción

En la actualidad existe una aplicación para casi cualquier necesidad, en este caso el equipo de SystemSquad se encargará del desarrollo de una aplicación móvil que ayude a los niños que cursan la primaria a entender mejor el tema de las fracciones.

Diseñada especialmente para niños en edad escolar, Frac - Helper presenta una interfaz de usuario amigable y cautivadora que estimula la curiosidad y la creatividad. A través de coloridas animaciones, juegos interactivos y desafíos educativos, los niños podrán sumergirse en el mundo mágico de las fracciones, comprendiendo conceptos fundamentales con facilidad y diversión.

Desde las fracciones más sencillas hasta los desafíos matemáticos más complejos, Frac - Helper ofrece una amplia gama de actividades adaptadas a distintos niveles de habilidad. Los niños podrán aprender a sumar, restar, multiplicar y dividir fracciones de manera intuitiva, consolidando su conocimiento en cada paso del camino.

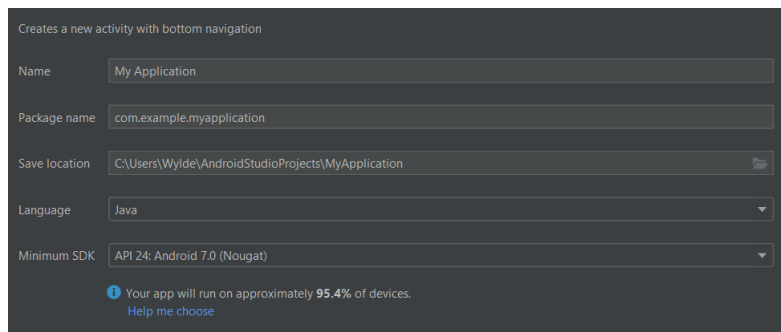


Ilustración 1 - Logo de la App

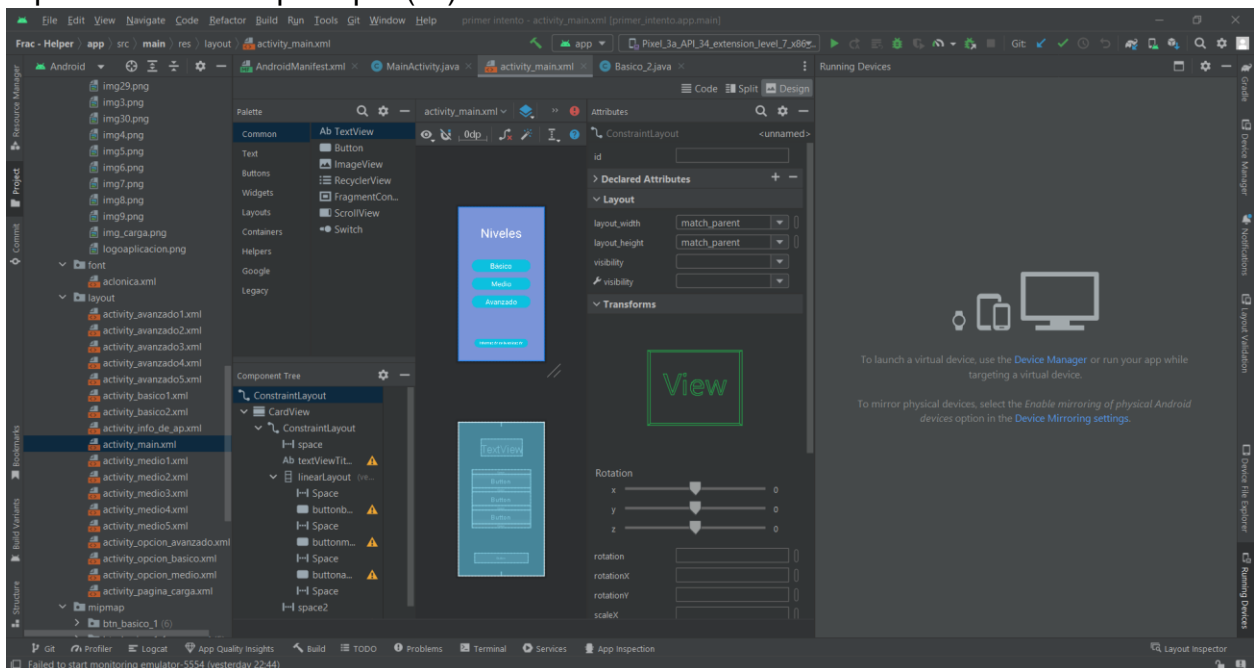
Proceso de Programación

El proceso de programación de la aplicación en un principio se realizó en Unity, pero la idea fue descartada debido a que falta la implementación de la retroalimentación solicitada en la aplicación. Por tal motivo, se migro hacia Android Studio donde continuamos con el proceso de codificación.

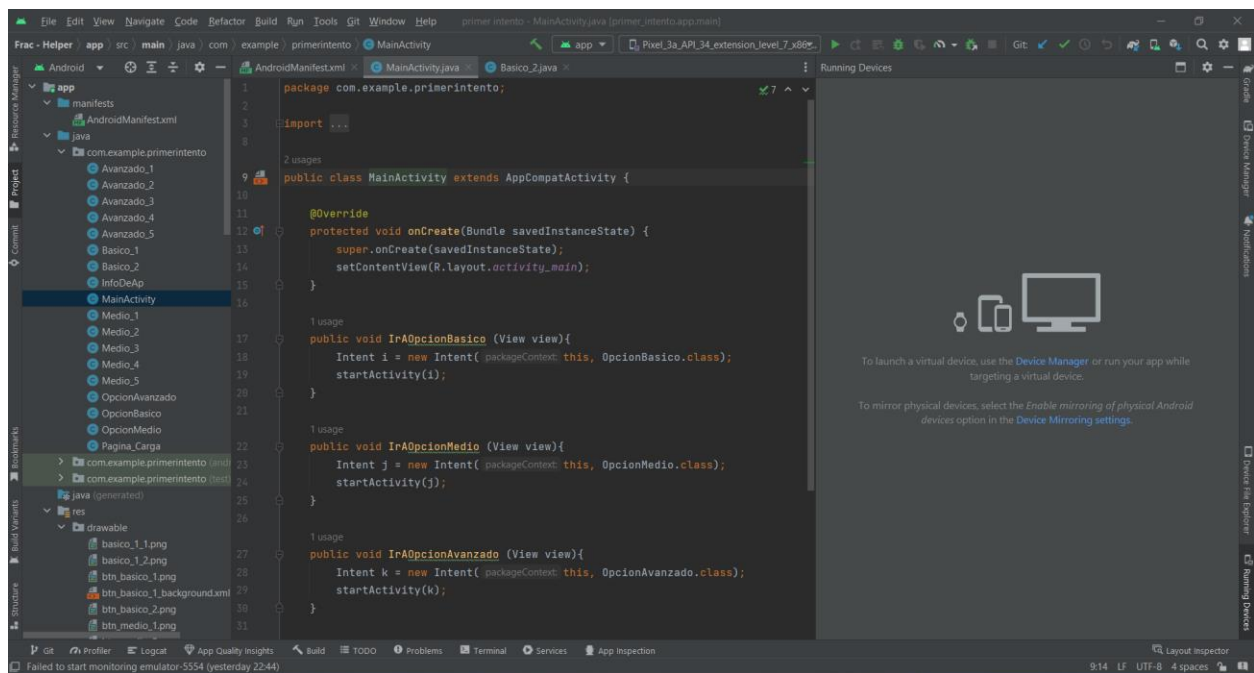
1. **Creación de un nuevo proyecto con:** "Start a new Android Studio project" o selecciona "File" > "New" > "New Project".
2. Elige la opción "**Empty Activity**" o cualquier otra plantilla según las necesidades de la app.



3. **Abrir el archivo activity_main.xml:** Cuando creas un nuevo proyecto en Android Studio, se genera automáticamente un archivo XML llamado "activity_main.xml" (u otro nombre si especificaste uno diferente) en la carpeta "res/layout". Este archivo representa la vista principal (UI) de tu actividad.

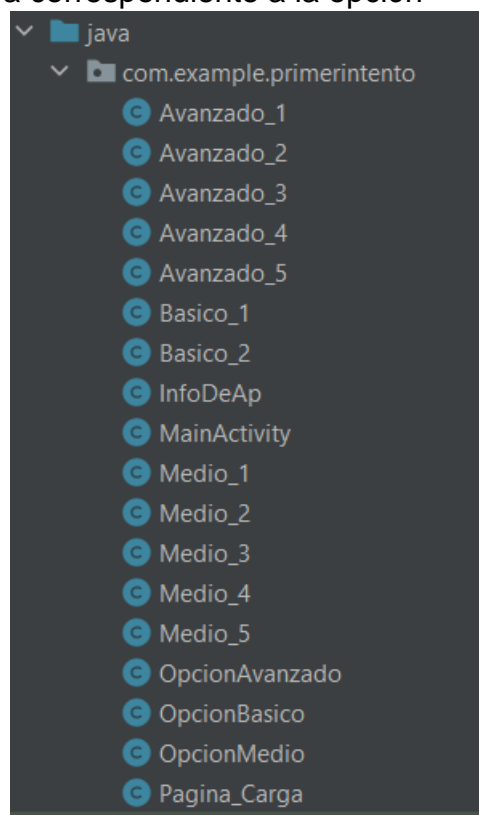


4. **Diseñar la interfaz gráfica utilizando el Editor de Diseño:** En el editor "Layout Editor" nos permitirá diseñar la interfaz de usuario arrastrando y soltando componentes en la pantalla. Arrastrando los componentes que necesitas desde la paleta hasta la pantalla de diseño.
5. **Ajustar las propiedades de los componentes:** Después de colocar los componentes en el diseño, es posible que necesites ajustar sus propiedades para que se vean y funcionen como deseas. Puedes hacerlo en la sección de "Atributos" del Inspector de Propiedades:
 - Cambiar el texto o las imágenes de los componentes.
 - Configurar el tamaño y el color de fuente del texto.
 - Establecer márgenes y rellenos para controlar el espacio entre los componentes.
 - Definir identificadores únicos (IDs) para los componentes que deseas acceder desde el código.
6. **Alternar entre la vista de diseño y el código XML:** Se alterno entre la vista de diseño y el código XML haciendo clic en las pestañas correspondientes en la parte inferior del editor.
7. **Añadir interacción a los componentes:** Para que la interfaz gráfica sea interactiva, es necesario agregar eventos a los componentes. De ahí la asignación.
8. **Previsualizar en diferentes pantallas:** Fue crucial asegurarnos de que el diseño se adapte correctamente a diferentes tamaños de pantalla y orientaciones (Responsividad para dispositivos Android).
9. **Abrir el archivo MainActivity.java:** Cuando creas un nuevo proyecto en Android Studio, se crea automáticamente un archivo llamado MainActivity.java (o con otro nombre si has especificado otro). Este archivo es donde se encuentra la clase principal de la actividad que se inicia al ejecutar la aplicación.
10. **Definir variables para acceder a componentes de la interfaz gráfica:** Para interactuar con los componentes de la interfaz gráfica definidos en el archivo XML de diseño (activity_main.xml), necesitas crear referencias a ellos en el archivo MainActivity.java. Para ello, debes declarar variables del mismo tipo que los componentes y asignarles el valor utilizando el método findViewById en el método onCreate.
11. **Implementar la lógica de la app:** La lógica de la aplicación se implementa dentro de los métodos apropiados de la clase MainActivity.java.



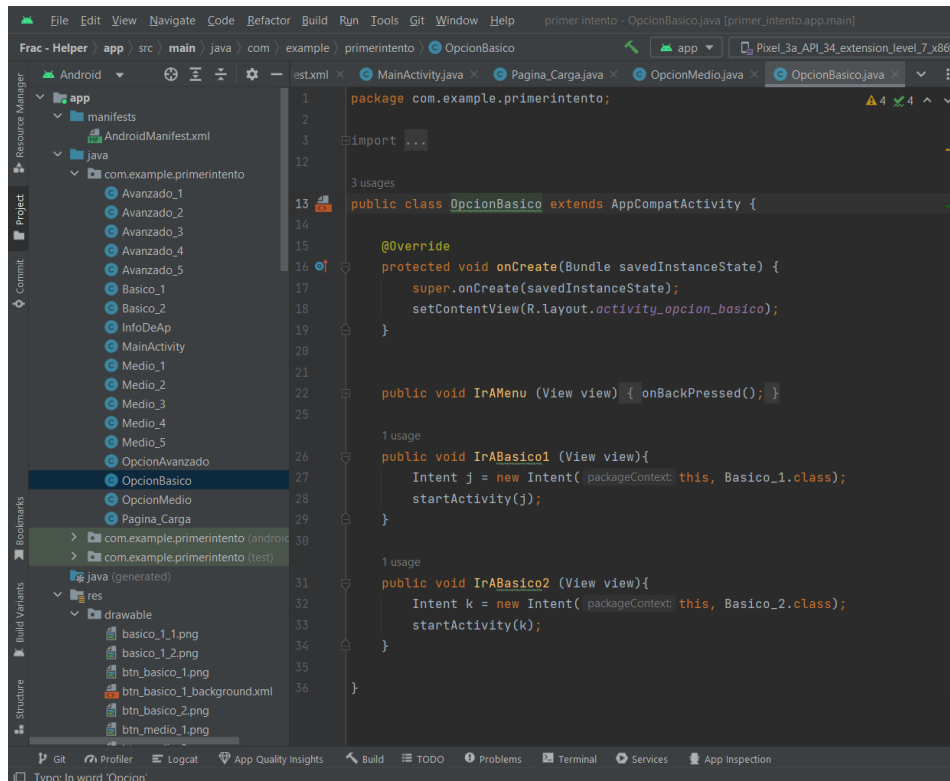
Aquí está el menú principal con las opciones de los niveles de dificultad.

12. Crear cada archivo .java correspondiente a la opción

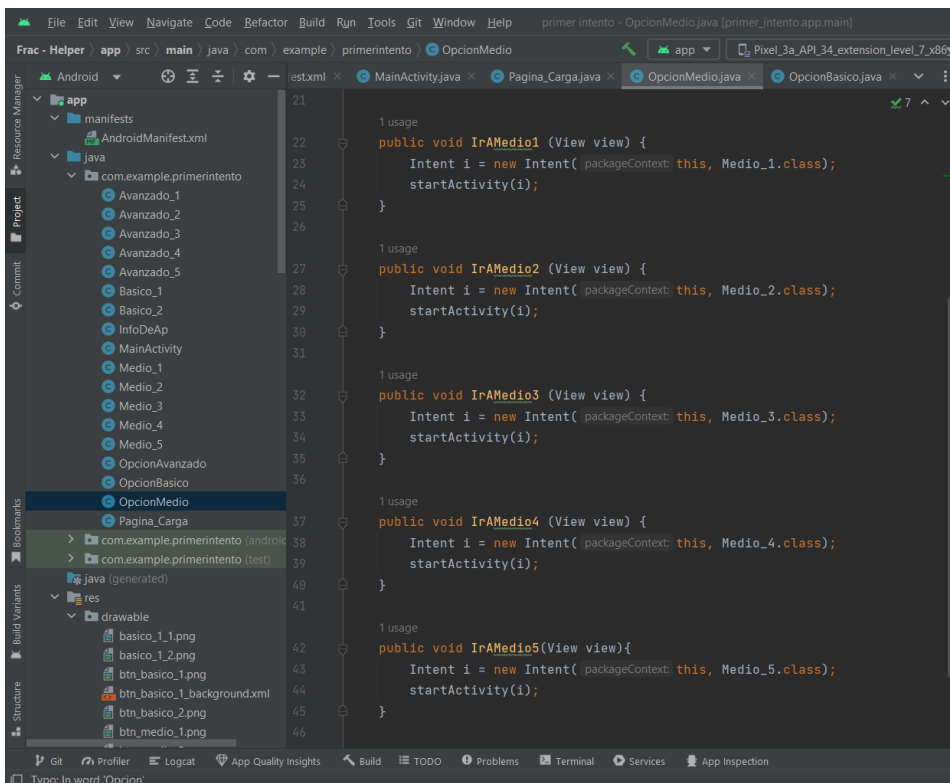


13. Crear submenús según cada nivel seleccionado:

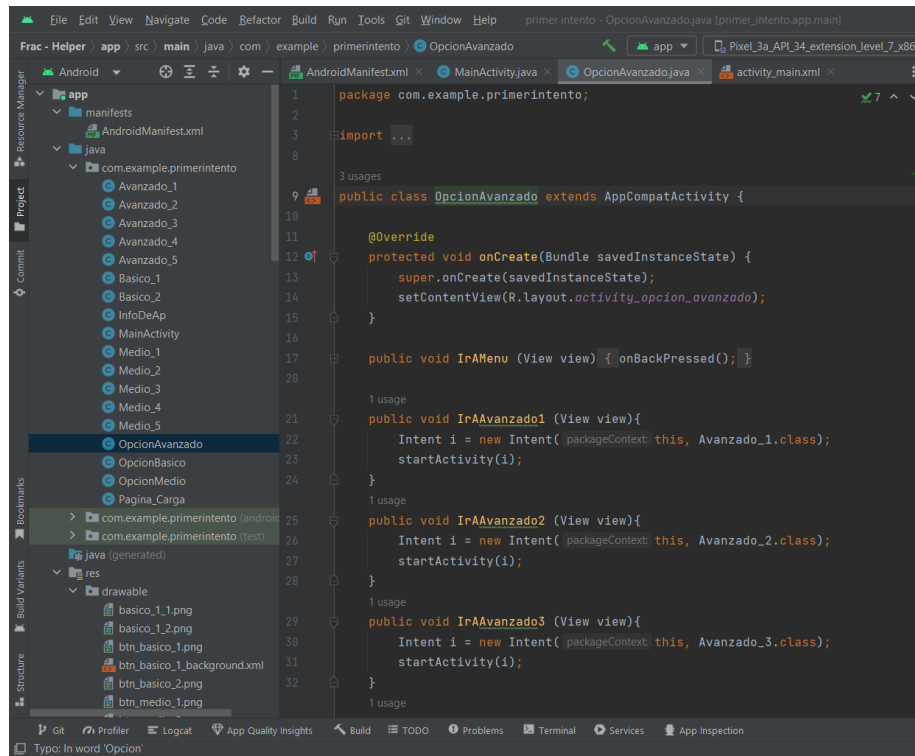
- Básico:



- Intermedio:



- Avanzado:



Pruebas de la App

Prueba 1: La aplicación no pasaba de la carga de pantalla en el simulador Device Manager.

Prueba 2: Las imágenes no se preservan en su lugar después de una navegación sencilla entre menús.

Prueba 3: Las preguntas en un principio eran estáticas, pero debido a que no se maneja un Login para usuarios, se optó por presentar preguntas aleatorias (obviamente, respetando el nivel de dificultad de la app).

Prueba 4: La validación de las preguntas fue lo más difícil, pues teníamos que hacer pruebas con cada pregunta para ver que respetara la asignación de ID de respuestaCorrecta.

Prueba 5: Por un tiempo de entrega reducido, se logro que marcara de color verde la respuesta correcta cuando el usuario respondia incorrectamente, al mismo tiempo que se marca de color rojo el botón que había presionado anteriormente.

Soluciones

1. Se debatieron las posibilidades basadas en el tiempo, los recursos y las características solicitadas en el documento, llegando a la conclusión de priorizar la funcionalidad de la aplicación sobre los lujos.
2. Tras varios intentos de acomodar el layout de cada activity, se encontró el estándar utilizado en cada submenú de los niveles medio y avanzado. Todos los demás detalles fueron determinados comparando con el diseño original para aproximarse lo más posible tanto en tipo de letra como en acomodo.
3. Tras revisar línea por línea de código para verificar que cada comando funcione, se encontró la falla a cada problema y se corrigió como se muestra en el código final. El problema terminó siendo un ciclo innecesario con while y la condición `respconfirmar.equals(imagen_preg[numerogenerado])` ya que en un descuido se comparaban una cadena de caracteres con un número entero. Posteriormente estas modificaciones dieron pie a eficiencias para evitar líneas de código innecesarias. El funcionamiento de este activity fue la base para el funcionamiento del resto de submenús tanto para el nivel medio como para el nivel avanzado.
4. Todas las partes del código que utilizaban imágenes para funcionar se cambiaron por funciones correspondientes a un TextView, además de adaptar el layout a preguntas con texto. Dicho código se repitió en todos los activity de nivel medio y avanzado relacionados a los ejercicios respectivamente con nuevos arreglos que tengan la información necesaria en cada uno.
5. En vez de generar otro activity, se generó un botón extra para finalizar la evaluación y mostrar 2 TextView que tuvieran el puntaje final y la leyenda correspondiente a su resultado.
6. Se tuvieron que hacer varias imágenes con diferentes resoluciones con el mismo nombre para poder visualizar los cambios generados en AndroidManifest.xml.
7. La única forma de que el celular le permitiera al programa instalar la aplicación, era conectándolo mediante cable USB a la computadora, accediendo a opciones de desarrollador, activar depuración USB y permitir instalaciones vía USB. Se hizo la prueba en 2 celulares de diferentes marcas dando como resultado las mismas funciones que en el emulador de Android Studio.

Conclusiones

- Se logro completar el desarrollo de la app móvil en el tiempo asignado por parte del cliente.
- Se migro de Unity a Android Studio por cuestiones de infraestructura de los equipos computacionales (Es importante mencionar esto pues se había el dicho la propuesta del MVP con Unity).
- Todos miembros del equipo de SystemSquad cumplió con cada uno de sus roles asignados.
- Hay algunas notas en el código que sirvieron como guía para organizar mejor el código y un comando que finalmente no fue utilizado en el programa que permaneció en algunos activities, siendo este: `ArrayList<String> respuestaArr = new ArrayList<String>(Arrays.asList(respuesta_preg));`. Si se desea limpiar el código para tener solamente líneas de código funcionales se pueden eliminar sin problema, o conservar como parte de una evidencia de administración dentro del mismo equipo.
- Algunos elementos de los layout utilizados no fueron modificados en su id ya que no era necesario para el código, sin embargo, se pueden renombrar para una mejor presentación y atención al detalle.
- La teoría de cada nivel puede mejorar, explicando el proceso de suma, resta, multiplicación y división de cada uno, sin embargo, se priorizó el funcionamiento sobre esos detalles para asegurar acabar la aplicación con el tiempo suficiente para que el cliente pueda modificarla a su gusto.