

**Universidad Autónoma Metropolitana**

**División de Ciencias Básicas e Ingeniería**

**Licenciatura en Ingeniería en Computación**

Reporte de Proyecto Tecnológico

**“Tutor móvil como estrategia para desarrollo de la habilidad de resolver problemas de cinemática elemental en el nivel superior desarrollada para Android.”**

Oscar Osciel Perea Godínez

210 206 121

bio\_osc@hotmail.com

Asesor:

Gustavo Mauricio Bastián Montoya

Profesor Titular del Departamento de Ciencias Básicas

mbastien@correo.azc.uam.mx

**Trimestre 2017 – Primavera    17/Julio/2017**



**Resumen:**

El presente documento describe el desarrollo y construcción de una aplicación móvil tutor, así como las diferentes tecnologías que se utilizaron en el proceso.

El objetivo principal para realizar el proyecto es proporcionar a los estudiantes una herramienta más para desarrollar habilidades de resolución de problemas, que brinde portabilidad y disponibilidad al alumno, así como también facilite la mecanización de los procesos de aprendizaje que se manejan de manera común en los cursos de nivel superior del área de Ciencias Básicas e Ingeniería.

En la sección 'Apéndice' se incluyen secciones de código relevantes incluidos dentro de la aplicación de Android (formato .apk), algunas de estas secciones de código se describirán de manera general más adelante en el documento.

## Índice General

1. Introducción.....	6
2. Antecedentes.....	7
2.1. Tesis.....	7
2.2. Proyectos Terminales e Integración.....	7
2.3. Artículos.....	8
3. Justificación .....	9
4. Objetivos.....	10
5. Marco Teórico.....	10
5.1. Aplicación Móvil.....	10
5.2. Android .....	11
5.3. XML .....	12
5.3.1 Etiquetas.....	12
5.4. Java.. ..	12
5.5. Android Studio .....	13
5.5.1. Activity .....	15
5.5.2. Manifest .....	15
5.5.3. Toast .....	16
5.6. Cinemática.....	16
6. Desarrollo.....	17
6.1. Diseño Y Análisis De Diseño .....	17
6.2. Implementación.....	18
6.2.1. Menú Y Aspectos Generales Del Código .....	18
6.2.2. Modulo Casos.....	21
6.2.2.1. Entrada.....	21
6.2.2.2. Actividades Y Casos .....	22
6.2.3. Modulo Estrategias.....	23
6.2.3.1 Actividades.....	24
6.3. Navegación.....	25
7. Resultados .....	27
7.1. Instalación de la Aplicación.....	27
7.2. Prueba del Módulo Casos .....	29
7.3. Prueba del Módulo Estrategias.....	30
8. Análisis y Discusión de Resultados .....	32
9. Conclusiones.....	34
10. Referencias Bibliograficas.....	35
11. Apendice.....	36
11.1. AndroidManifest.xml.....	36
11.2. Código Java.....	38

11.3. Código XML.....	74
11.4 strings.xml.....	85
12. Entregables.....	88

## Índice de Figuras

Figura 1 Archivos del Proyecto en la vista de Android Studio.....	14
Figura 2 Estructura de un Archivo 'Manifest' (XML) para Aplicaciones Android.....	16
Figura 3 Estructura General de la Aplicación.....	17
Figura 4 Definición de un elemento Lógico.....	18
Figura 5 Enlace de Elementos Lógicos con Elementos Gráficos.....	19
Figura 6 Comparación Vista Texto y Vista Diseño de archivos XML en Android Studio.....	19
Figura 7 Integración del Método 'onClick' a un Elemento Lógico.....	20
Figura 8 Metodo 'onClick'.....	20
Figura 9 Estructura del Módulo Casos.....	21
Figura 10 Implementación de la Validación para la Entrada de Datos.....	22
Figura 11 Distribución de Elementos Gráficos en 'Activities' Casos 1.....	23
Figura 12 Distribución de Elementos Gráficos en 'Activities' Casos 2.....	23
Figura 13 Estructura y Funcionalidad Conceptual del Módulo Estrategias.....	23
Figura 14 Toma de Decisión en el Módulo Estrategias.....	24
Figura 15 Introducción a la aplicación Cinemática UAM.....	26
Figura 16 Eliminación de la Memoria cache de una Aplicación en Android.....	26
Figura 17 Archivo .apk en un Directorio Android.....	27
Figura 18 Instalación de un Archivo .apk en Android.....	28
Figura 19 Aplicación Instalada Correctamente en Dispositivos Android.....	28
Figura 20 Navegación a través de las Interfaces del módulo Casos.....	29
Figura 21 Error de Exceso de Datos en la Entrada de Datos y Mensaje de Notificación <i>Toast</i> .....	30
Figura 22 Teclado de Entrada Estándar Restringido para Caracteres Numéricos.....	30

Figura 23 Navegación a través de las Interfaces del Modulo Estrategias.....	31
Figura 24 Ejemplo de Retroalimentación en el modulo Estrategias.....	31
Figura 25 Resultado Erróneo debido a un cociente 0 despliega NaN.....	32

## **Índice de Tablas**

Tabla 1 Elementos de Desarrollo de la Aplicación definidos y utilizados durante la Construcción del Proyecto.....	17
---	----

## **1. Introducción:**

La evaluación en las materias que cursan los alumnos de ingeniería le da prioridad a la resolución de problemas (RP) y no a la memorización de la información, por ello es importante que un alumno desarrolle su habilidad para resolver problemas, cabe señalar que esta habilidad no es innata y se puede mejorar a través de la práctica guiada. En particular el proceso de RP en física elemental universitaria, en las carreras de ingeniería en la UAM-A ha sido estudiado y se ha planteado un modelo de resolución que representa el camino seguido por un estudiante promedio.

Para seguir este modelo, el alumno requiere conocimientos no muy complicados acerca de algunos aspectos como ecuaciones y traducción de variables físicas a matemáticas y para aplicarlos exitosamente se requiere, además, de comprensión de los enunciados y aplicación de leyes y ecuaciones en diversos contextos. Hasta ahora la única manera que tienen los alumnos de desarrollar la habilidad de RP es observando la forma en que un profesor expone en el salón de clase.

Por lo anterior se optó por una herramienta computacional para que los alumnos mejoren este proceso sin asistencia directa del docente. Debido a las características de los alumnos de CBI de la UAM-A, se ha propuesto que la herramienta sea una aplicación móvil (App) ya que desde hace algunos años la telefonía celular dio un paso importante, pues con los teléfonos inteligentes o Smartphones se desarrollaron herramientas al alcance de la mano para un gran número de tareas, entre ellas las educativas, brindando a los usuarios cierta inmediatez en cuanto al acceso y uso de información.

## **2. Antecedentes:**

Para el proyecto a realizar se revisaron los siguientes materiales como apoyo y referencias:

### **2.1. Tesis**

*Desarrollo y aplicación de una metodología de resolución de problemas de física elemental universitaria para enseñanza combinada.[1]*

Este trabajo explica la metodología utilizada en los sistemas tutor de la plataforma Moodle, detallando gran parte del proceso al desarrollar metodologías de enseñanza, la segunda mitad del trabajo trata la aplicación de dicha metodología, así como el análisis de datos obtenidos, en el actual desarrollo no se pretende evaluar el desempeño de la plataforma bajo los criterios o metodologías desarrolladas en este trabajo, pero si estructurar el contenido de la aplicación a partir de las conclusiones y experiencia que se ha reportado en anteriores plataformas tutor.

### **2.2. Proyectos Terminales e Integración**

*Aplicación móvil para la sincronización de archivos.[2]*

Este proyecto nos presenta una aplicación móvil implementada en el lenguaje Java, si bien esta aplicación cuenta con un servicio web, para el proyecto no se tiene contemplado hacer uso de la arquitectura cliente-servidor, además de que el propósito de la aplicación también difiere del de la actual propuesta, aunque de igual forma se trate de una App.

*Aplicación para el apoyo a la enseñanza de la UEA Métodos Numéricos.[3]*

En relación a este trabajo tenemos que el tipo de aplicación cumple con el mismo propósito, un sistema tutor el cual en este caso fue diseñado e implementado con aplicaciones bajo contexto de ejecución (Applets) y bibliotecas graficas de Java (Swing), su diseño e implementación está basado en otra Unidad de Enseñanza-Aprendizaje (UEA) de las Ingenierías en UAM, uno de los módulos trabajados además incluye cierto nivel de retroalimentación al alumno.



*Sistema tutor web para el aprendizaje de programación orientada a objetos.[4]*

En este caso el concepto de sistema tutor es enfocado a una UEA específica de Ingeniería en Computación, se diseñó como tutor WEB en vez de una App para dispositivos móviles, mediante Netbeans 6.8 (Plataforma de desarrollo Java), el sistema que se desarrolló servirá como herramienta para la resolución de problemas y no para ejemplificar los temas de un curso específico como en el proyecto citado.

### 2.3. Artículos

*Presenta Upaep Concurso de APPs Innovation 2017.[5]*

El artículo hace énfasis en la incursión del desarrollo de apps por parte de estudiantes de nivel medio superior y superior, abriendo los campos de desarrollo a otra áreas además de la académica, su relación con este proyecto es el enfoque de desarrollo que tiene la convocatoria, por otro lado este proyecto no pretende ser comercializado por el hecho de que se tratara de propiedad intelectual de la Universidad Autónoma Metropolitana.

*Apps que te facilitarán estudiar y pasar exámenes.[6]*

En el artículo se enlistan diferentes aplicaciones para celular, que pretenden apoyar a estudiantes haciendo de estas apps hábitos de estudio, incluso se hace mención de una app diseñada para cursos y contenido de licenciaturas de la UNAM, la mayoría no son tutores como la aplicación que se desarrolló, pero si se relacionan por lo que se pretende lograr, que es apoyar a los estudiantes en sus hábitos de estudio.

### **3. Justificación:**

“Una de las dificultades que se les presentan a los docentes al utilizar enseñanza combinada es la forma en la que se puede desarrollar la habilidad de resolver problemas” [7].

Bajo esta idea, el desarrollo de herramientas que apoyen tanto a alumnos en el aspecto del aprendizaje de un tema o curso, como a profesores al diseñar el contenido de dichos cursos, ayuda a aminorar esta brecha cognitiva que es el aprender o enseñar de manera correcta a resolver problemas, de hecho, a medida que las generaciones pasan los modelos de aprendizaje dirigido se deben modificar de tal manera que estas sean adecuadas a los alumnos, es decir que sean actuales, esto último suma a la tecnología como punto importante al crear estas herramientas.

La App que se pretende desarrollar sirve para este fin, mezclando la tecnología y la metodología de enseñanza para que los alumnos de Ciencias Básicas e Ingeniería en la UAM Azcapotzalco desarrollen estrategias para resolver problemas, esta App estará basada en el SO Android para Smartphones dado que la plataforma es la más usada por los usuarios de Smartphones desde hace algunos años. Cabe mencionar que en 2016 fue el primer año en que se vendieron más equipos móviles que computadoras de escritorio.

## **4. Objetivos:**

### **4.1. Objetivo General:**

- Diseñar e implementar una aplicación móvil que sirva de apoyo para alumnos de CBI UAM Azcapotzalco para mejorar sus habilidades de resolución de problemas de Cinemática.

### **4.2. Objetivos Específicos:**

- Diseñar un Prototipo de interfaz utilizando Prototyping in Paper (POP).
- Diseñar e implementar un módulo que presente casos de problemas de cinemática (Módulo Casos).
- Diseñar e implementar un módulo interactivo que ayude al alumno a practicar y mejorar sus estrategias de resolución de problemas paso a paso (Módulo Estrategias).
- Implementar la interfaz de la aplicación.

## **5. Marco Teórico:**

### **5.1 Aplicación Móvil**

Una de las características principales de una aplicación móvil es la necesidad de descargar e instalar ésta en un sistema móvil, ciertas características de la aplicación la hacen apta o no para un sistema móvil, una vez que la aplicación está terminada se compila o empaqueta de manera que parezca una aplicación nativa.

Las llamadas 'Apps' actualmente están enfocadas a mejorar la productividad personal, esto significa que proporcionan inmediatez en el uso y acceso a datos o bien ayudan a reducir los tiempos en algún proceso o tarea intermedia involucrada.

Existen diferentes sistemas operativos donde utilizar aplicaciones móviles, sin embargo, anteriormente se ha mencionado a Android como el sistema preferente no solo para este proyecto, si no para la comunidad consumidora de tecnología móvil en general, por lo anterior y dado que el proyecto está diseñado para este, solo se abordara Android en el presente documento.

## 5.2 Android

Jesús Tomás de Girones en su libro “El gran libro de Android”, menciona una serie de características de lo que Android ofrece hoy en día:

- La plataforma se puede usar y customizar sin pagar regalías, lo que la hace una plataforma realmente abierta.
- Es adaptable a cualquier tipo de hardware, en los últimos años se ha visto a la plataforma llegar a otros dispositivos inteligentes además de telefonía móvil o tabletas, como pantallas *Smart tv* y *Smartwatch*.
- Las aplicaciones presentan alto nivel de portabilidad pues el desarrollar en Java y su máquina virtual aseguran la ejecución en cualquier tipo de CPU.
- La arquitectura basa sus componentes en Internet, las interfaces de usuario se diseñan en XML, esto permite la ejecución en TV como se mencionó anteriormente.
- Presenta una filosofía ‘Dispositivo Siempre Conectado’.
- Cuenta con un gran número de servicios incorporados.
- Los programas están aislados entre sí por lo que el nivel de seguridad es aceptable.
- El sistema esta optimizado para baja potencia y poca memoria.

Está basado en Linux, su sencillez es la causa de que existan muchas aplicaciones disponibles que extienden la funcionalidad de los dispositivos y mejoran la experiencia del usuario [8].

Android ofrece una forma sencilla y novedosa de implementar potentes aplicaciones para diferentes tipos de dispositivos [9].

Comúnmente las aplicaciones para Android, se descargan desde ‘Google Play’ bajo la extensión ‘.apk’, los dispositivos cuentan con opciones de depuración y herramientas para los programadores en la mayoría de los casos.

### 5.3 XML

En informática se le conoce como metalenguaje al conjunto de reglas que definen a otros lenguajes, XML es entonces un metalenguaje que define las reglas, por ejemplo para HTML, eso significa que XML solo define la forma en la que se deben crear o utilizar etiquetas y no la semántica que se involucra en un lenguaje como tal.

#### 5.3.1 Etiqueta

Las etiquetas constituyen la base de XML, determinan el alcance de cualquier elemento definido en el XML, cada elemento cuenta con una etiqueta de inicio, una etiqueta de cierre y el contenido del elemento [10].

Las etiquetas hacen distinción entre mayúsculas y minúsculas, además deben de cerrarse en orden apropiado bajo modelos FILO (Firs In Last Out) es decir una etiqueta abierta dentro de otro elemento debe estar cerrada antes de que el elemento exterior cierre.

### 5.4 Java

Java presenta una sintaxis amigable, opciones orientadas a objetos, manejo de memoria y la firme idea de portabilidad [11].

La idea de programar una sola vez y ejecutar en cualquier parte es bastante atractiva si se busca un lenguaje portable, una vez creado el archivo fuente el compilador escanea el documento para encontrar errores sin dejar que termine la compilación hasta que todo se ejecute de manera correcta, el compilador genera un archivo Java *bytecode* el cual java puede interpretar y ejecutar de manera independiente de la plataforma, a esto último se le conoce como máquina virtual [12].

Java basa y divide el código en clases alojadas en archivos fuente con extensión '.class', muchas de estas clases están disponibles desde la plataforma de la que se haga uso, cada clase cuenta con sus propios métodos, sin embargo java permite crear nuestras propias clases y métodos y referenciarlas de manera que la única diferencia es la importación de estos en el archivo fuente. Todo el código que se trabaja en Java se encuentra dentro de una clase, en realidad al ejecutar

una clase, la máquina virtual ejecuta los archivos '.class' contenidos en el programa.

Todo el lenguaje Java está basado en el lenguaje C de sistemas operativos (Linux principalmente), esto es que todo lo que se implementa en java puede ser implementado en C, requiriendo generar estructuras y métodos predefinidos en java desde cero en algunos casos, Java se ha vuelto uno de los lenguajes más populares en los últimos años por el hecho de que la mayoría de implementaciones ya están predefinidas en bibliotecas basadas en el lenguaje, lo que facilita muchas de las operaciones dentro de una implementación.

## 5.5 Android Studio

Es el entorno de desarrollo integrado (IDE) que proporciona Google de manera oficial para el desarrollo de aplicaciones Android. Además del potente editor de códigos y las herramientas para desarrolladores de IntelliJ, Android Studio ofrece aún más funciones que aumentan la productividad de un proyecto durante la compilación de apps para Android, como las siguientes [13]:

- Un sistema de compilación basado en Gradle flexible.
- Un emulador rápido con varias funciones.
- Un entorno unificado en el que puedes realizar desarrollos para todos los dispositivos Android.
- Instant Run para aplicar cambios mientras tu app se ejecuta sin la necesidad de compilar un nuevo APK.
- Integración de plantillas de código y GitHub para ayudarte a compilar funciones comunes de las apps e importar ejemplos de código.
- Gran cantidad de herramientas y frameworks de prueba.
- Herramientas Lint para detectar problemas de rendimiento, usabilidad, compatibilidad de versión, etc.
- Compatibilidad con C++ y NDK.

La estructura básica de un proyecto en Android Studio incluye 2 directorios, el directorio App que contiene los archivos propios de las funcionalidades y aspectos gráficos y técnicos del proyecto, y un directorio Gradle el cual incluye las importaciones de bibliotecas utilizadas en los archivos java o información sobre la

versión SDK, el Gradle funciona como configuración para la compilación de la aplicación.

La plataforma también cuenta con un emulador de dispositivos como herramienta de depuración o presentación de las APK que se generan dentro del entorno. De manera predeterminada se muestran los archivos del proyecto del lado izquierdo de la interfaz (Figura 1) la cual está organizada en módulos.

Todos los archivos de compilación son visibles en el nivel superior de secuencias de comando de Gradle y cada módulo de la aplicación contiene una carpeta 'manifest', una carpeta con archivos java y una carpeta de recursos (res), la plataforma elimina automáticamente los recursos sin usar del paquete de la app y de las dependencias de bibliotecas del APK compilado [13].

El renderizado de los recursos gráficos se realiza en tiempo real, y se implementa un pre compilado para detectar errores mediante el Gradle.

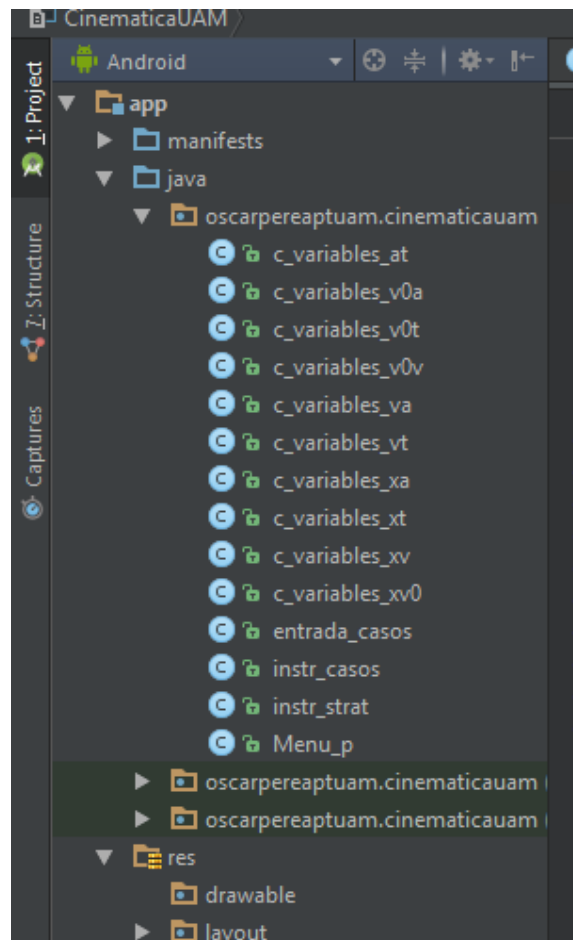


Figura 1 Archivos del Proyecto en la vista de Android Studio.

### 5.5.1 Activity

En el caso de Android contamos con actividades o activities que representan las ventanas con las que cuenta la aplicación, es decir si nuestra aplicación cuenta con 3 pantallas en total se puede traducir a utilizar 3 actividades. Todas las actividades están conformadas por un archivo XML y un archivo Java, el archivo XML corresponde al ambiente grafico mientras que el archivo Java seria la lógica de programa, al crear una nueva actividad el Gradle se encarga de enlazar los archivos desde su definición teniendo asi archivos con código para la definición de la unión de ambas partes.

Cada archivo se puede trabajar de manera independiente sin necesidad de tener ambos en la pantalla de edición de la plataforma, los archivos XML se encuentran en el directorio Res (Recursos) del directorio App, mientras que los archivos Java se encuentran en un directorio con el mismo nombre dentro del mismo App.

### 5.5.2 Android Manifest

Todas las aplicaciones cuentan con un archivo 'AndroidManifest.xml' dentro del directorio App, el cual proporciona información para poder ejecutar el código de la aplicación, el nombre del paquete Java, descripciones de los componentes de la aplicación, todas las actividades y servicios que integren en el código, también se declaran los permisos que debe o no tener la aplicación para acceder a recursos de hardware y componentes de la aplicación, así como la versión de la API (Application Package Index).

El archivo Manifest puede considerarse un mapa de navegación de la aplicación pues guarda jerarquías entre actividades, de hecho, en el archivo se indica cual de todas servirá como *launcher* de la aplicación (Activity de arranque) y en algunos casos guarda relaciones entre Actividades padre y Actividades hijos, esto último depende de la versión del ambiente con la que se trabaje (Figura 2).



```
manifest application activity
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
package="com.carperespruan.cinematicasuan">

<application
android:allowBackup="true"
android:icon="@mipmap/ic_launcher"
android:label="@string/app_name"
android:supportsRtl="true"
android:theme="@style/AppTheme">
<activity android:name=".MainActivity">
<intent-filter>
<action android:name="android.intent.action.MAIN" />

<category android:name="android.intent.category.LAUNCHER" />
</intent-filter>
</activity>
<activity android:name=".Instr_Casos" />
<activity android:name=".Instr_Ejerc" />
<activity android:name=".Entrada_Casos" />
<activity
android:name=".c_variables_yt"
android:label="@string/Resolver_Velocidad_y_Tiempo"
android:theme="@style/AppTheme.NoActionBar" />
</activity>
</application>
</manifest>
```

Figura 2 Estructura de un Archivo 'Manifest' (XML) para Aplicaciones Android.

### 5.5.3 Toast

Son mensajes emergentes dentro de una aplicación Android, los cuales desaparecen después de un tiempo corto, estos generalmente sirven de *feedback* tras alguna acción realizada.

### 5.6 Cinemática

La cinemática se ocupa de las variaciones de aceleración, velocidad y posición de un cuerpo, hay numerosos ejemplos en la naturaleza donde la aceleración de un cuerpo es esencialmente constante durante largos periodos de tiempo, desde núcleos atómicos hasta movimientos planetarios [14].

El Movimiento Rectilíneo Uniformemente Acelerado comprende a todo movimiento descrito por un cuerpo en el que se describa una trayectoria recta estando sometido a aceleraciones constantes y es un caso particular del movimiento acelerado [14].

## 6. Desarrollo:

El desarrollo de la aplicación se realizó mediante 2 módulos, como se muestra en la Figura 3, estos módulos se basan en la funcionalidad y objetivos determinados con anterioridad para la aplicación.

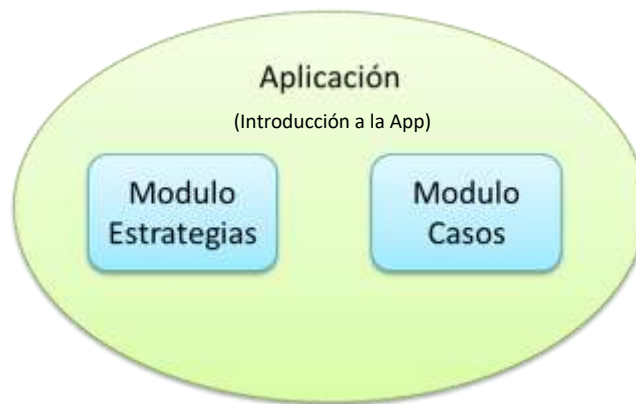


Figura 3 Estructura General de la Aplicación.

La aplicación cuenta con una explicación general de la misma en la primera pantalla que despliega, la cual solo aparece una vez por la sencillez del esquema de la aplicación y el hecho de que familiarizarse con la aplicación no es difícil.

### 6.1 Diseño y Análisis de Diseño

Un primer enfoque al iniciar el desarrollo incluía el resolver problemas de Movimiento Rectilíneo Uniformemente Acelerado (MRUA) desde la aplicación, por lo que se incluyó en los objetivos desde la definición del proyecto.

Posteriormente se definieron los medios de desarrollo, como se muestra a continuación en la Tabla 1:

	Actual	Información
Plataforma	Android 4.1 o Posterior	A la fecha el 95.2% de los dispositivos móviles Android cuenta con esta API 16 (JellyBean) como mínimo.
Lenguaje	Java	Lenguaje Orientado a Objetos trabajado bajo el paradigma de programación Orientado a Eventos.
Entorno de desarrollo	Android Studio	Versión 2.3.3

Tabla 1: Elementos de Desarrollo de la Aplicación definidos y utilizados durante la Construcción del Proyecto.

Antes de comenzar la programación de los módulos o la interfaz se trabajó el diseño de esta, utilizando la plataforma web 'Draw.io' se generaron los diseños preliminares de las diferentes pantallas con las que contaría la app, dándonos una idea de lo que esperábamos obtener como el resultado final estético del proyecto.

Posteriormente se trabajó en la plataforma web 'marvelapp' con el fin de trabajar en un prototipo funcional basado en los diseños preliminares, la plataforma permite enlazar elementos gráficos con cierto nivel de interacción simulada, los diseños preliminares no se incluyen en el documento por no ser relevantes pero se puede consultar el prototipo como referencia en la siguiente dirección, la cual incluye los diseños, la política de la plataforma no permite la descarga del prototipo. (<https://marvelapp.com/38b0e73>) [15].

Una vez generados los anteriores entregables y contando con el entorno de desarrollo, se trabaja con los módulos y el menú general que tendrá la app, Android Studio basa la aplicación en actividades o pantallas de manera que genera los archivos con un esqueleto en código, posteriormente el esqueleto se programa con código de acuerdo con la funcionalidad de la app, a continuación se describe el código utilizado para cada una de las diferentes actividades que se incluyen en la app. Algunos de los diseños no se incluyeron en la construcción final por funcionalidad en la implementación.

## 6.2. Implementación

### 6.2.1. Menú y Aspectos Generales del Código

El menú principal de la aplicación solo cuenta con 2 botones, cada uno redirige a uno de los 2 módulos de la app. Los elementos lógicos de los dos botones se definen dentro de la clase al principio de la misma, de esta manera todos los elementos lógicos posteriores se definirán igualmente (Figura 4).

```
public class Menu_p extends AppCompatActivity implements View.OnClickListener{  
  
    Button casos;  
    Button strat;
```

Figura 4 Definición de un elemento Lógico.

Dentro del método de creación de la clase se enlazan los elementos xml, la referencia mostrada se debe a que Android Studio guarda todos los elementos gráficos en un directorio 'Resources' (R) el cual incluye un archivo de Cadenas del cual se hablara más adelante, de esta manera se enlazaran elementos posteriores igualmente (Figura 5).

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_menu_p);

    casos = (Button) findViewById(R.id.b_casos); //enlaza con el
    strat = (Button) findViewById(R.id.b_strat); //elemento xml
```

Figura 5 Enlace de Elementos Lógicos con Elementos Gráficos.

Android Studio permite manipular los archivos XML de diseño de manera gráfica o lógica según se necesite, en este caso la definición de elementos se hizo de manera gráfica mientras que la distribución y propiedades de los elementos de manera lógica (Figura 6).

De esta manera podemos observar los id's relacionados en el código así como las referencias al archivo de cadenas 'strings.xml' anteriormente mencionado.

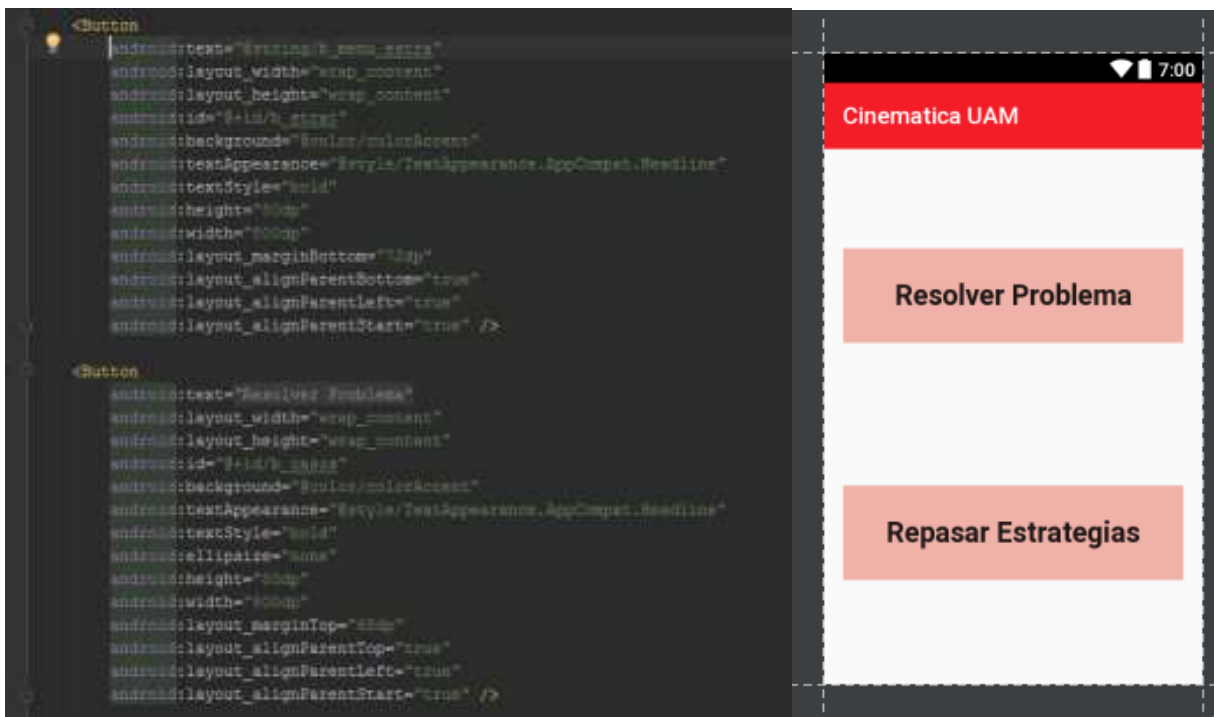


Figura 6 Comparación Vista Texto y Vista Diseño de archivos XML en Android Studio.

Retomando el código en Java, finalmente se deben habilitar los elementos lógicos para activar una acción, esto bajo el paradigma de programación orientado a eventos, para posteriores elementos botón se trabajara de igual manera (Figura 7).

```
casos.setOnClickListener(this); //habilita el elemento
strat.setOnClickListener(this); //xml para accionar al click
}
```

Figura 7 Integración del Método 'onClick' a un Elemento Lógico.

El método utilizado por la clase se define como se muestra en la Figura 8, dentro del método se define un 'INTENT' que es la manera de invocar a las diferentes actividades incluidas la app, así como la forma de lanzar a ejecución la misma.

```
@Override
public void onClick(View v){
    if (v.getId() == R.id.b_casos) {
        Intent m_casos = new Intent(this, instr_casos.class);
        startActivity(m_casos);
    }
    else if (v.getId() == R.id.b_strat){
        Intent m_strat = new Intent(this, instr_strat.class);
        startActivity(m_strat);
    }
}
```

Figura 8 Metodo 'onClick'.

Todos los ambientes gráficos XML se diseñaron para el tamaño de pantalla mínimo que cuenta la API 16 de Android (4.0, 480 x 800 hdpi Nexus 5), por esta razón la distribución o el tamaño de elementos gráficos puede parecer pequeña en comparación con equipos móviles que cuentan con resoluciones más grandes en su pantalla, esto de ninguna manera afecta la funcionalidad de la aplicación.

## 6.2.2. Modulo Casos

El modulo cuenta con una estructura entrada salida como se muestra en la Figura 9, se implementaron 10 casos a resolver dado que son los más comunes en problemas de MRUA.

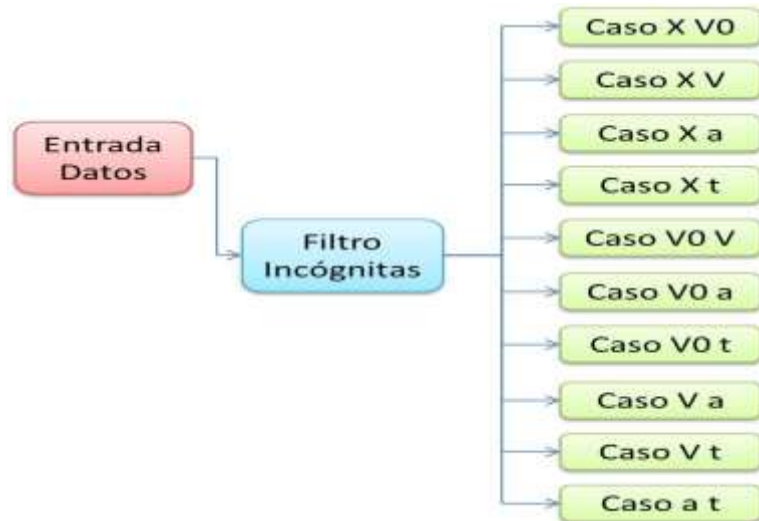


Figura 9 Estructura del Módulo Casos.

### 6.2.2.1. Entrada

La entrada de datos se realiza desde la pantalla principal del módulo casos, se cuenta con 6 campos de captura de datos (Figura 9) así como de un botón que se encarga de activar el filtro y guardar las entradas como datos para la carga de cualquiera de los 'INTENT' posteriores con los resultados del problema.

Esta estructura se encuentra en el método 'onClick' de la clase correspondiente a la entrada de datos como se muestra en la Figura 10, esta figura además muestra cómo se implementan los elementos TOAST correspondientes a errores de exceso o falta de datos.

La entrada estándar del teclado sumada a la configuración de los elementos gráficos solo permite manejar números reales signados en los campos y no así letras o caracteres especiales en la aplicación.

```

entrada_base.setOnClickListener()
entrada_x = findViewById(R.id.ed_x);
entrada_t = findViewById(R.id.ed_t);

filtro_sol.setOnClickListener(this);
}

@Override
public void onClick(View v) {
    if (v.getId() == R.id.b_resolver) {
        cad_u0 = entrada_u0.getText().toString();
        cad_x = entrada_x.getText().toString();
        cad_v0 = entrada_v0.getText().toString();
        cad_y = entrada_y.getText().toString();
        cad_a = entrada_a.getText().toString();
        cad_t = entrada_t.getText().toString();
        // string de validacion para la entrada "Incongnita"
        if (cad_u0.isEmpty()) {
            mensaje.setText(getApplicationContext().getString(R.string.mensaje_incongnita));
        }
        else {
            if (cad_x.isEmpty()) {
                if (cad_v0.isEmpty()) {
                    if (cad_a.isEmpty() || cad_t.isEmpty()) {
                        mensaje.setText(getApplicationContext().getString(R.string.mensaje_incongnita));
                    }
                    else { // caso variables x y v0
                        u0 = Float.parseFloat(cad_u0);
                        v0 = Float.parseFloat(cad_v0);
                        a = Float.parseFloat(cad_a);
                        t = Float.parseFloat(cad_t);
                        Intent select_c = new Intent(this, c_variables_v0.class);
                        select_c.putExtra("pos_sol", u0);
                        select_c.putExtra("vel", v0);
                        select_c.putExtra("acel", a);
                        select_c.putExtra("tiempo", t);
                        startActivity(select_c);
                    }
                }
            }
            else if (cad_v0.isEmpty()) {
                if (cad_x.isEmpty() || cad_a.isEmpty() || cad_t.isEmpty()) {
                    mensaje.setText(getApplicationContext().getString(R.string.mensaje_incongnita));
                }
                else { // caso variables x y v0
                    u0 = Float.parseFloat(cad_u0);
                    v0 = Float.parseFloat(cad_v0);
                    a = Float.parseFloat(cad_a);
                    t = Float.parseFloat(cad_t);
                    Intent select_c = new Intent(this, c_variables_v0.class);
                    select_c.putExtra("pos_sol", u0);
                    select_c.putExtra("vel", v0);
                    select_c.putExtra("acel", a);
                    select_c.putExtra("tiempo", t);
                    startActivity(select_c);
                }
            }
        }
    }
}
}

```

Figura 10 Implementación de la Validación para la Entrada de Datos.

### 6.2.2.2 Actividades y Casos

Se implementaron 10 de los 15 casos posibles a resolver, tomando en cuenta que cada caso corresponde a 2 variables incógnita de las 6 variables a considerar en un problema de MRUA, se descartaron los casos correspondientes a X0 como incógnita por el hecho de que no son casos comunes de problemas prácticos de cinemática elemental. Por otra parte una vez que se comenzó a programar el modulo la implementación de estos aumenta el tiempo de procesamiento del Gradle, el cual no es muy significativo en tiempo de ejecución pero si en tiempo de compilación.

Cada *Activity* en el módulo está diseñada y programada para que despliegue un ejemplo de problema con los datos de la entrada, un procedimiento simple con el cual se puede resolver el problema y la solución correspondiente a la entrada (Figura 11). La interfaz también cuenta con botones para salir al menú o bien para volver a introducir datos si es que se desea resolver otro problema con diferentes valores de entrada (Figura 12).

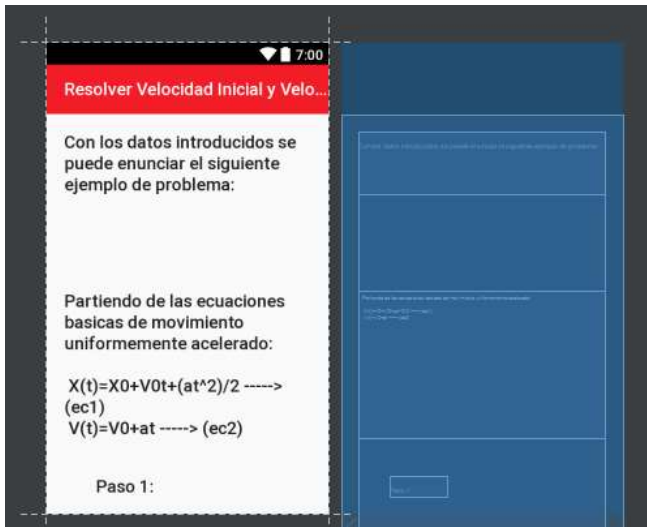


Figura 11 Distribución de Elementos Gráficos en 'Activities' Casos 1.

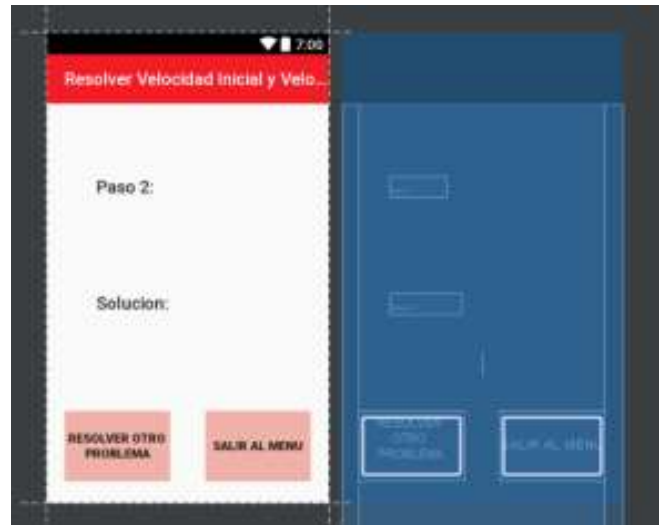


Figura 12 Distribución de Elementos Gráficos en 'Activities' Casos 2.

Los textos se generan de manera dinámica para incluir los datos de entrada.

### 6.2.3 Modulo Estrategias

El modulo basa su construcción en la toma de decisiones y posible análisis que puede hacer un alumno al enfrentarse con resolución de problemas de cinemática, por lo que se implementó como un árbol de decisión como se muestra en la Figura 13.

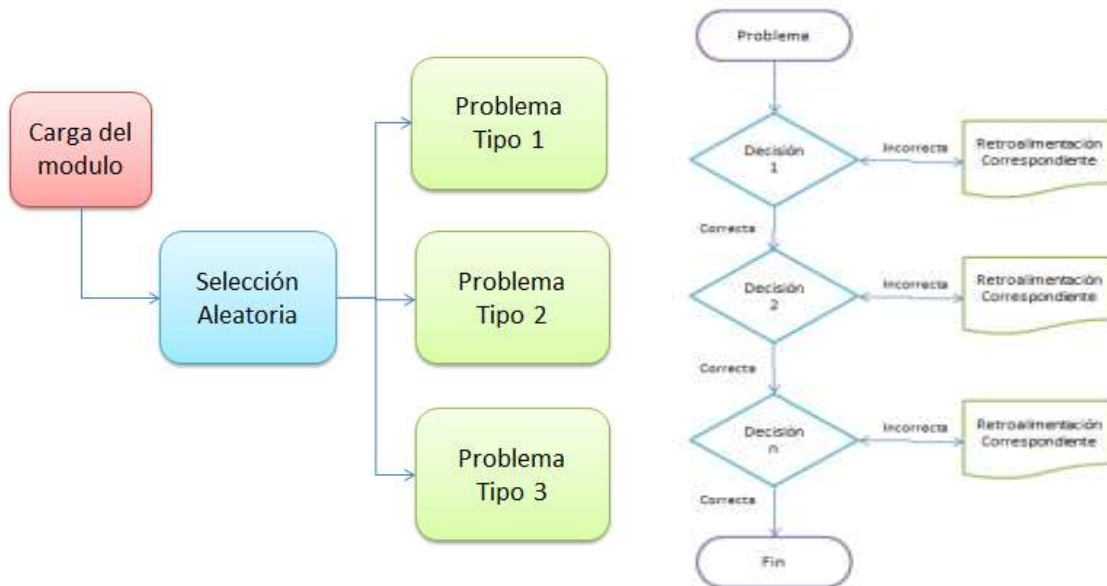


Figura 13 Estructura y Funcionalidad Conceptual del Módulo Estrategias.



### 6.2.3.1 Actividades

Se implementaron 3 problemas de análisis en la aplicación, los cuales incluyen de 4 a 5 decisiones conceptuales enfocadas en la comprensión y análisis conceptual, análisis de datos, análisis de expresiones algebraicas y análisis de soluciones, las clases java correspondientes a los diferentes problemas se invocan de manera semi-aleatoria.

Cada toma de decisión va acompañada de 2, 3 o 4 posibles respuestas o selecciones (Figura 14) acompañadas de un botón del lado derecho, el cual se encarga de desplegar la retroalimentación necesaria al usuario.



Figura 14 Toma de Decisión en el Módulo Estrategias.

Al terminar un problema se cuenta con botones para salir al menú o bien para comenzar otro problema, de manera similar a las Actividades del módulo Casos.

Toda la información referente a la retroalimentación se seleccionó basada en la experiencia de Mauricio Bastián como docente de la Universidad Autónoma Metropolitana unidad Azcapotzalco, así como también de lo mencionado en su trabajo de 2010 “Desarrollo y aplicación de una metodología de resolución de problemas de física elemental universitaria para enseñanza combinada” [1].

### 6.3 Navegación

La navegación de toda la aplicación se basa en eventos, los cuales principalmente se activan al momento de hacer *click* en uno de los botones programados y contenidos en la aplicación.

El ‘mapa’ digital de la aplicación está alojado en el archivo `AndroidManifest.xml`, por defecto se invoca a la misma instancia de las *Activities* en lugar de crear nuevas instancias de la misma cuando se hace la correspondiente invocación. Esto evita que existan elementos en memoria repetidos, dado que el funcionamiento de la aplicación no lo requiere.

La aplicación no cuenta con menús desplegables pues no existen opciones más allá de las mostradas en el menú principal, por otro lado al resolver cualquier problema o bien al terminar un problema de decisión se cuenta con re direccionamientos hacia la interfaz principal de la aplicación mediante botones destinados a ello, esto permite también moverse de un módulo a otro sin dejar de darle seguimiento a datos de salida o retroalimentación según sea el caso.

La pantalla de introducción solo se muestra la primera vez que se ejecuta la aplicación en el sistema Android (Figura 15), o bien si se borra la cache de la aplicación desde el menú de aplicaciones del sistema (Figura 16), implementar la aplicación de esta manera asegura que se tendrá una pequeña explicación de la misma y que una vez que el alumno se familiarice con su funcionamiento esta explicación pueda ser descartada.



Figura 16 Introducción a la aplicación Cinemática UAM.

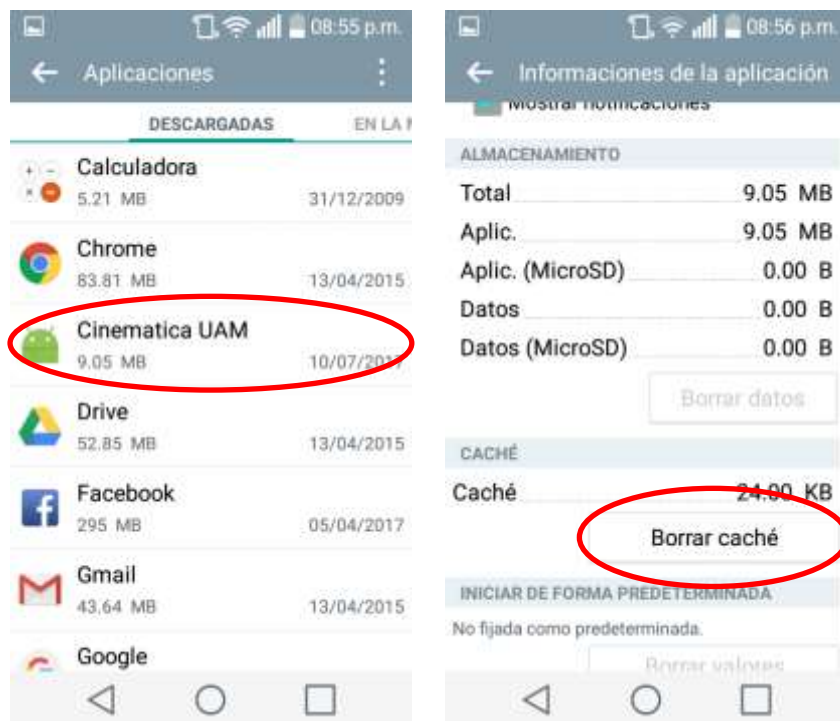


Figura 15 Eliminación de la Memoria cache de una Aplicación en Android.

## 7. Resultados:

Al finalizar el proyecto se genera el archivo 'CinematicaUAM.apk' como paquete de instalación para el sistema móvil Android versión 4.1 o cualquiera de sus versiones posteriores, el archivo contiene los archivos java, los archivos XML de interfaces gráficas y los archivos XML de recursos adicionales (p. ej. El archivo 'strings.xml' contiene todos los textos estáticos desplegados) en su versión compilada.

### 7.1. Instalación de la aplicación

Se deberá acceder al archivo '.apk' ingresando al directorio correspondiente desde un sistema móvil Android, en la Figura 17 se muestra un ejemplo, que para este caso el archivo se encuentra en el directorio raíz de nuestro sistema.

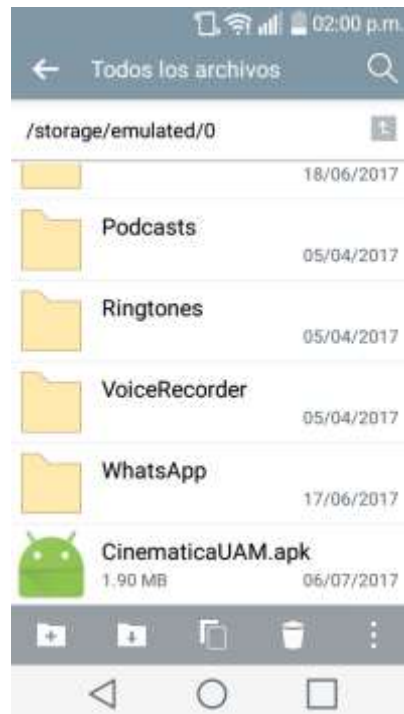


Figura 17 Archivo .apk en un Directorio Android.

Posteriormente al hacer *click* en el archivo, Android reconoce el archivo como un paquete de instalación, y nos mostrara una pantalla en la que se muestran los permisos de hardware con los que cuenta la aplicación (Figura 18), la aplicación que se desarrolló no necesita acceso a ningún hardware adicional a la

pantalla *touch screen* del teléfono o a la entrada estándar del teclado incluida, por lo que se indica que no se necesitan permisos adicionales.

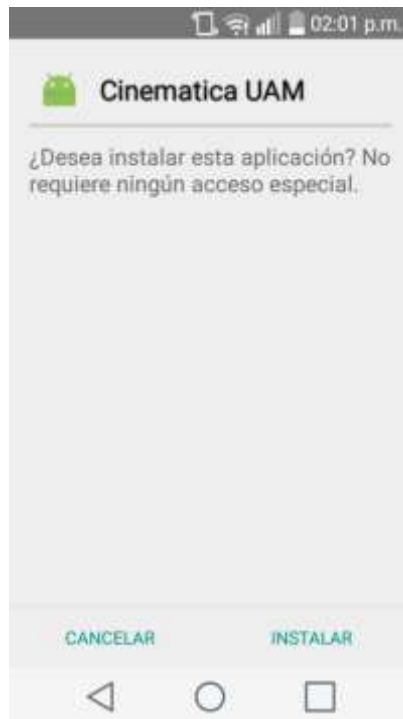


Figura 18 Instalación de un Archivo .apk en Android.

Si la aplicación se ha instalado correctamente en nuestro sistema Android, o bien si ha ocurrido un error en esta, el sistema operativo se encarga de desplegar una notificación correspondiente al usuario, como se observa en la Figura 19.

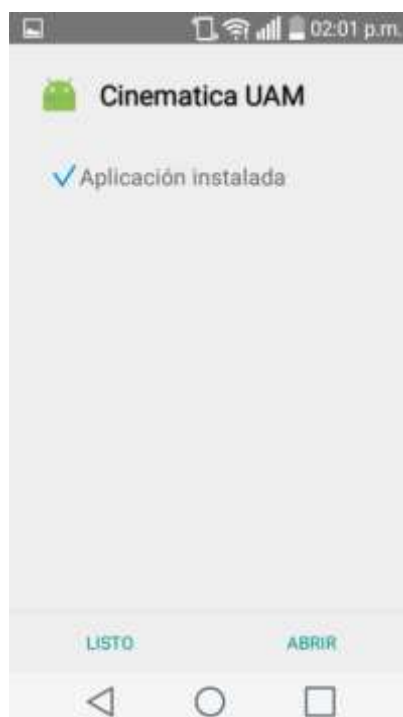


Figura 19 Aplicación Instalada Correctamente en Dispositivos Android.

El ejecutable de la aplicación se encontrara en el dock de aplicaciones del dispositivo.

## 7.2. Prueba del Módulo ‘Casos’

El menú principal de la aplicación muestra 2 botones, el botón izquierdo/superior (según si la pantalla de introducción se muestra o no) corresponde al módulo casos, al hacer *click* en él se muestran las instrucciones del módulo y se podrá continuar hacia la entrada de datos (Figura 20).

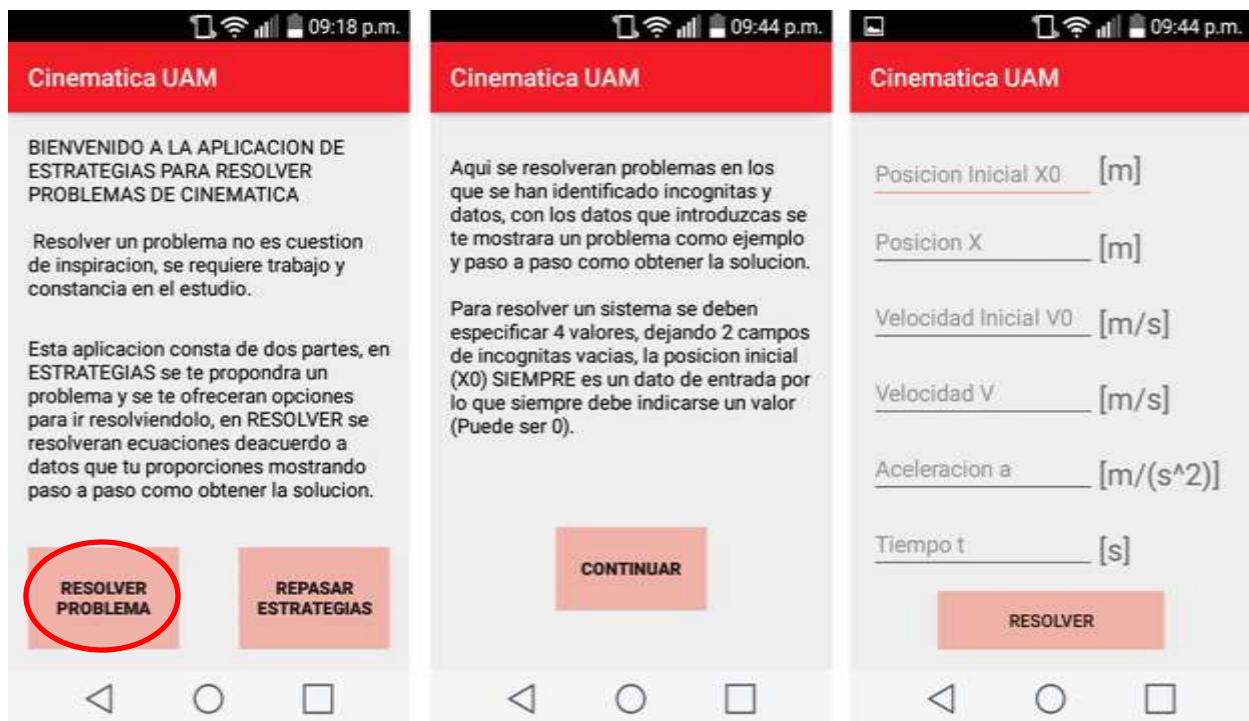


Figura 20 Navegación a través de las Interfaces del módulo Casos.

Una vez en esta interfaz el usuario debe introducir 4 de los 6 campos, cada campo cuenta con un texto fantasma que sirve de *Hint* para el usuario así como unidades del lado derecho para ayudar al usuario a introducir cantidades de manera correcta, mediante la entrada del teclado Android, para posteriormente dar *click* en el botón con el texto ‘RESOLVER’.

En caso de que exista algún error en la entrada, el árbol de validación nos ayudara a identificarla, como se muestra en la Figura 21, además la entrada estándar del teclado de Android solo permite introducir caracteres numéricos, punto decimal y de ser necesario signo negativo (Figura 22).

Finalmente la aplicación desplegará un método para encontrar la solución apoyada de sistemas de ecuaciones, así como también la solución del sistema.

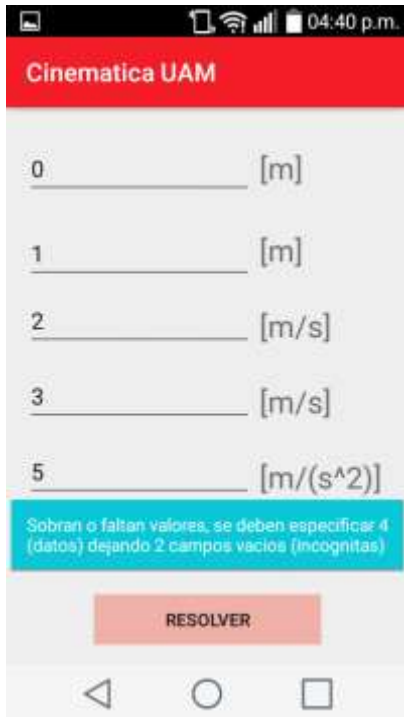


Figura 21 Error de Exceso de Datos en la Entrada de Datos y Mensaje de Notificación *Toast*.



Figura 22 Teclado de Entrada Estándar Restringido para Caracteres Numéricos.

### 7.3 Prueba del Módulo Estrategias

El menú principal de la aplicación muestra 2 botones, el botón derecho/inferior (según si la pantalla de introducción se muestra o no) corresponde al módulo estrategias, al hacer *click* en él se muestran las instrucciones del módulo y se podrá continuar hacia cualquiera de los problemas incluidos en la aplicación (Figura 23).

El módulo está diseñado para que el usuario reciba retroalimentación en cada paso si su respuesta es incorrecta (Figura 24), a medida que el usuario avanza se marca un razonamiento correcto para la resolución del problema.



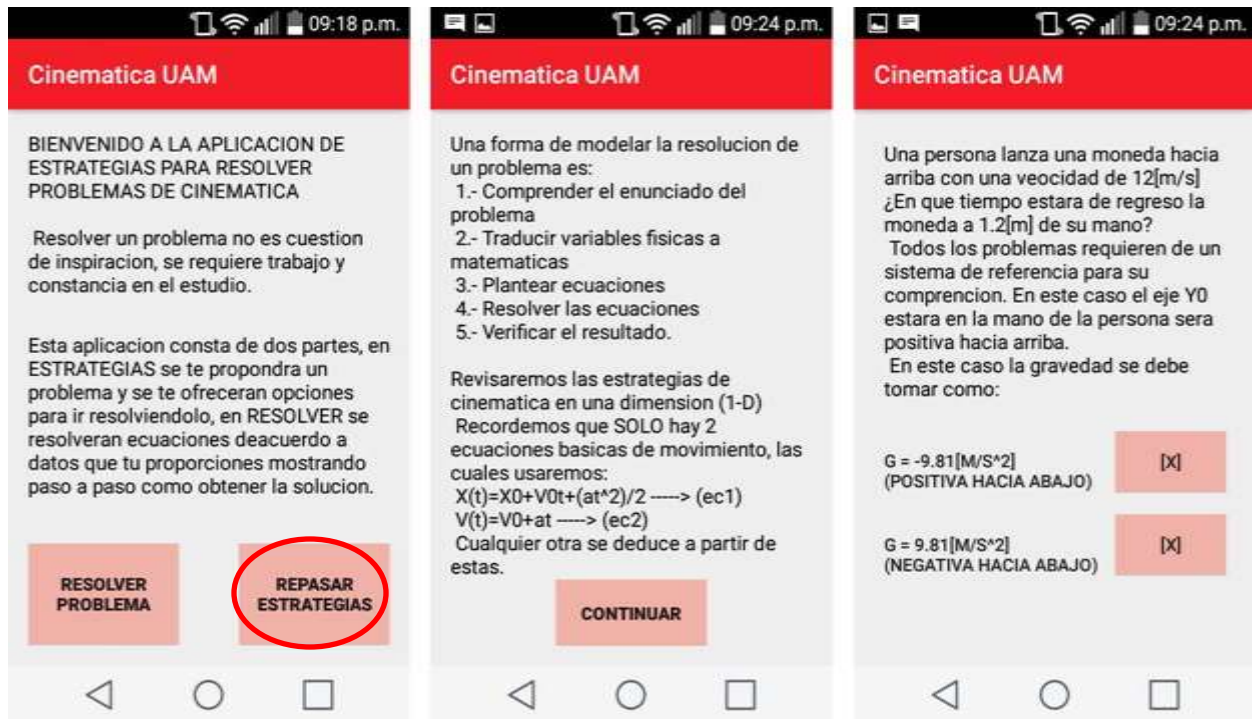


Figura 21 Navegación a través de las Interfaces del Modulo Estrategias.



Figura 22 Ejemplo de Retroalimentación en el modulo Estrategias.



## 8. Análisis y Discusión de Resultados:

Tanto el modulo estrategias como el modulo casos trabajan correctamente, el modulo estrategias despliega adecuadamente las pantallas asignadas para cada retroalimentación, por su parte el modulo casos solo resuelve sistemas en los cuales solo se tienen 2 incógnitas, sin desplegar erróneamente interfaces que no correspondan con la solicitud de la entrada.

El modulo casos sin embargo presenta irregularidades cuando al dar solución al sistema se tienen cosas como cocientes iguales a 0, por ejemplo si se introducen ceros tanto en la posición como la posición inicial al resolver aceleración y tiempo se desplegara la interfaz correspondiente, pero los valores de salida se mostraran en pantalla como NaN (Not A Number) como se muestra en la Figura 25, esto puede interpretarse de dos maneras:

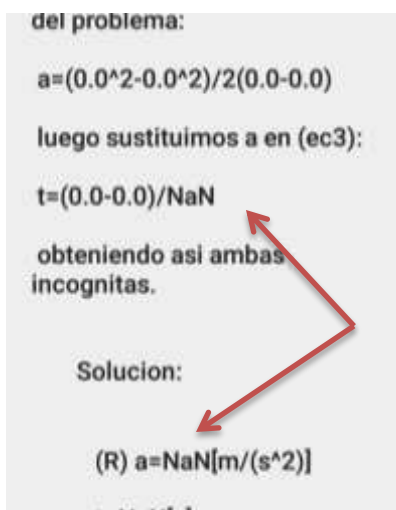


Figura 23 Resultado Erróneo debido a un cociente 0 despliega NaN.

Desde el punto de vista matemático, las divisiones por cero no están definidas ni permitidas pues se consideran paradojas matemáticas, por lo que el procesador no puede realizar la operación.

Desde el punto de vista físico, si la distancia del objeto final e inicial es la misma (no existe desplazamiento o bien es 0), o bien si la velocidad inicial y final es la misma (no existe aceleración o bien es 0), significa que el móvil está en reposo a su vez que no ha transcurrido ningún instante de tiempo por lo que el problema no tiene sentido, correspondiendo al resultado erróneo.

Este tipo de casos sin embargo solo se presentan al forzar los datos de cierta manera, los datos que normalmente se trabajan en libros con problemas de cinemática o bien durante una clase o curso de cinemática jamás presentaran este tipo de datos forzados a cero o forzados a datos en los que se resten de tal manera que generen la indeterminación matemática, por lo que, si bien el error puede presentarse no se corrigió por las siguientes razones:

- No afecta la funcionalidad de la aplicación.
- En el peor de los casos simplemente se deben corregir los datos de la entrada para generar una solución nueva.
- De ningún modo se sobrecarga la memoria principal si se llega a presentar el caso en el que los valores generen NaN.
- El caso no se presentara a menos que se fueren los valores para generar el error.
- No todos los casos del módulo presentan cocientes para su resolución, lo que hace aún más raro que el error se presente en las variables.

De hecho al probar la aplicación e introducir únicamente 0 al tratar de brincar la interfaz fue que se encontró la posibilidad de que las variables contengan NaN en vez de un número de punto flotante como se definió la variable en Java, por ultimo cabe mencionar que al introducir valores negativos no se tienen este tipo de errores en las variables, siempre y cuando no sean forzados para generar el error nuevamente.

La parte grafica por otra parte se presenta de manera simple y no presenta mayor problema para los usuarios, se eligieron los siguientes colores RGB para el aspecto grafico de la aplicación:

```
<color name="colorPrimary">#F31D28</color>  
<color name="colorPrimaryDark">#000000</color>  
<color name="colorAccent">#F0B2A8</color>
```

## 9. Conclusiones:

La aplicación no es muy robusta si se trata de la cantidad de código utilizado, su diseño sencillo permite la convivencia de esta con sistemas Android ligeros y de poca memoria, su punto fuerte radica precisamente en el hecho de que con poco código se ha creado una herramienta altamente útil para el estudiante promedio de ingeniería, dentro y fuera de la Universidad Autónoma Metropolitana.

Sumado a esto, la estructura de la aplicación permite que fácilmente se integren nuevos módulos a futuro, si es que se quisiera ampliar su funcionalidad a otros temas de ingeniería, dado que es en esta área de las ciencias de nivel superior en donde más se presentan enfoques a la resolución de problemas.

En cuanto al error mencionado en la sección 8. Análisis y Discusión de Resultados, si bien es no es un error en la lógica de la aplicación, este podría considerarse una limitante al momento de resolver ciertos problemas por algunos usuarios, esto de ninguna manera hace a la aplicación 0% funcional, dado que es una herramienta sencilla en su estructura, pero fuerte si se trata de abarcar gran parte de lo que se presenta en un entorno real de problemas de cinemática.

Por otro lado construir la aplicación en un ambiente diseñado en principio para aplicaciones Android reduce el tiempo de desarrollo, dado que cuenta con herramientas de virtualización y compilación además permite generar aplicaciones con firma digital sin mayor problema, su interfaz de trabajo no es difícil de manejar y cuenta con una documentación extensa por parte de Google y Android.

Ambos módulos cumplen con lo establecido en la sección 4.2. Objetivos Específicos, bajo las especificaciones del Dr. Mauricio Bastián en el proceso de desarrollo, y a lo largo del proceso mismo no se presentaron inconvenientes que detuvieran la creación de la aplicación o alguno de sus componentes.

El proyecto tutor móvil se realiza de manera exitosa cumpliendo así con la sección 4.1. Objetivo General, la aplicación presenta un ambiente sencillo, fácil de manipular, el cual cumple ampliamente con las expectativas y condiciones de los involucrados en su conceptualización y desarrollo.

## 10. Referencias Bibliográficas

- [1]. Instituto Politécnico Nacional, “Desarrollo y aplicación de una metodología de resolución de problemas de física elemental universitaria para enseñanza combinada”, tesis doctoral, <http://tesis.ipn.mx/jspui/handle/123456789/9550?mode=full>, 2010. También se encuentra en Bastián M. G. M. . “Desarrollo y aplicación de una metodología de resolución de problemas de física elemental universitaria para enseñanza combinada”. Tesis doctoral. México D. F.: Centro de Investigación en Ciencia Aplicada y tecnología Avanzada, IPN, 2010.
- [2]. R. Ramirez Reyes, "Aplicación móvil para la sincronización de archivos.", proyecto terminal, División de CBI, Universidad Autónoma Metropolitana Azcapotzalco, México, 2010.
- [3]. M. Tapia Tellez, “Aplicación para el apoyo a la enseñanza de la UEA Métodos Numéricos.”, proyecto terminal, División de CBI, Universidad Autónoma Metropolitana Azcapotzalco, México, 2010.
- [4]. J. J. Perea Sanchez, “Sistema tutor web para el aprendizaje de programación orientada a objetos.”, proyecto terminal, División de CBI, Universidad Autónoma Metropolitana Azcapotzalco, México, 2011.
- [5]. Periódico Gratuito ‘Puntual’, “Presenta Upaep Concurso de App’s Innovation 2017”, <https://www.diariopuntual.com/universitarios/2017/02/13/45081>, 2017.
- [6]. Merca2.0, “Estas apps te facilitarán estudiar y pasar exámenes”, <http://www.merca20.com/estas-apps-te-facilitaran-estudiar-pasar-examenes/>, 2017.
- [7]. Google Docs, “Desarrollo de un Software Basado en Investigación”, <https://docs.google.com/document/d/1IZDjvST1XDA2dGjCnCNceRM4IgbYxjdShNZUkmj3qCU/edit?usp=sharing>, 2012.
- [8]. C. Robledo, D. Robledo, “Programación en Android”, Aula Mentor, 2012, p. 3.
- [9]. J. T. Girones, “El gran libro de Android”, Marcombo, Tercera edición, 2013, p. 5-6.
- [10]. M. A. Acera Garcia, “XML Manual Imprescindible”, Segunda Edición, 2011.
- [11]. K. Sierra, B. Bates, “Head First Java”, Segunda Edición, 2005, p. 1.
- [12]. K. Sierra, B. Bates, “Head First Java”, Segunda Edición, 2005, p. 2.
- [13]. Android Studio Users Guide, “Conoce Android Studio”, <https://developer.android.com/studio/intro/index.html>, 2014.
- [14]. H. E. White, “Física Moderna Volumen 1”, Primera Edición, 2007, p. 59-62.
- [15]. Prototyping on Paper, “Proto Cinematica UAM”, <https://marvelapp.com/38b0e73>, 2017.

## 11. Apéndice:

Los siguientes apartados corresponden al código utilizado para desarrollar la aplicación, el nombre de las secciones corresponde con el nombre de los archivos utilizados en la compilación de la aplicación, los cuales se incluyen anexos al medio óptico en el que se entrega el presente reporte, el directorio específico se anexa posterior al título de la sección.

### 11.1. AndroidManifest.xml - CinematicaUAM\app\src\main

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="oscarpereaptuam.cinematicauam">

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".introduccion">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity android:name=".Menu_p"/>
        <activity android:name=".instr_casos" />
        <activity android:name=".instr_strat" />
        <activity android:name=".entrada_casos" />
        <activity
            android:name=".c_variables_vt"
            android:label="@string/title_activity_c_variables_vt"
            android:theme="@style/AppTheme.NoActionBar" />
        <activity
            android:name=".c_variables_xa"
            android:label="@string/title_activity_c_variables_xa"
            android:theme="@style/AppTheme.NoActionBar" />
        <activity
            android:name=".c_variables_va"
            android:label="@string/title_activity_c_variables_va"
            android:theme="@style/AppTheme.NoActionBar" />
        <activity
            android:name=".c_variables_xv"
            android:label="@string/title_activity_c_variables_xv"
            android:theme="@style/AppTheme.NoActionBar" />
        <activity
            android:name=".c_variables_at"
            android:label="@string/title_activity_c_variables_at"
            android:theme="@style/AppTheme.NoActionBar" />
        <activity
            android:name=".c_variables_xt"
            android:label="@string/title_activity_c_variables_xt"
            android:theme="@style/AppTheme.NoActionBar" />
        <activity
            android:name=".c_variables_v0v"
            android:label="@string/title_activity_c_variables_v0v"
            android:theme="@style/AppTheme.NoActionBar" />
        <activity
```

```

        android:name=".c_variables_xv0"
        android:label="@string/title_activity_c_variables_xv0"
        android:theme="@style/AppTheme.NoActionBar" />
    <activity
        android:name=".c_variables_v0a"
        android:label="@string/title_activity_c_variables_v0a"
        android:theme="@style/AppTheme.NoActionBar" />
    <activity
        android:name=".c_variables_v0t"
        android:label="@string/title_activity_c_variables_v0t"
        android:theme="@style/AppTheme.NoActionBar" />
    <activity android:name=".strat_1" />
    <activity android:name=".strat_2" />
    <activity android:name=".strat_rB" />
    <activity android:name=".strat_rA" />
    <activity android:name=".strat_rC" />
    <activity android:name=".strat_3" />
    <activity android:name=".strat_rD" />
    <activity android:name=".strat_rE" />
    <activity android:name=".strat_4" />
    <activity android:name=".strat_rF" />
    <activity android:name=".strat_5" />
    <activity android:name=".strat_6" />
    <activity android:name=".strat_rG" />
    <activity android:name=".strat_7" />
    <activity android:name=".strat_rH" />
    <activity android:name=".strat_8" />
    <activity android:name=".strat_rI" />
    <activity android:name=".strat_9" />
    <activity android:name=".strat_rJ" />
    <activity android:name=".strat_10" />
    <activity android:name=".strat_11" />
    <activity android:name=".strat_rK" />
    <activity android:name=".strat_12" />
    <activity android:name=".strat_rL" />
    <activity android:name=".strat_13" />
    <activity android:name=".strat_rM" />
    <activity android:name=".strat_rN" />
    <activity android:name=".strat_rO" />
    <activity android:name=".strat_14" />
</application>
</manifest>

```

## 11.2. Código Java - CinematicaUAM\app\src\main\java\oscarpereaptuam\cinematicauam

Se generaron las siguientes clases Java dentro de la aplicación con el siguiente código.

### 11.2.1. introducción.java

```
public class introduccion extends AppCompatActivity implements View.OnClickListener{

    Button casos;
    Button strat;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_introduccion);

        casos = (Button) findViewById(R.id.b_icasos);
        strat = (Button) findViewById(R.id.b_istrat);

        casos.setOnClickListener(this);
        strat.setOnClickListener(this);
    }

    @Override
    public void onClick(View v) {
        if (v.getId() == R.id.b_icasos) {
            Intent m_casos = new Intent(this,instr_casos.class);
            startActivity(m_casos);
        }
        else if (v.getId() == R.id.b_istrat){
            Intent m_strat = new Intent(this,instr_strat.class);
            startActivity(m_strat);
        }
        else{}
    }
}
```

### 11.2.2. Menu\_p.java

```
public class Menu_p extends AppCompatActivity implements View.OnClickListener{

    Button casos;
    Button strat;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_menu_p);

        casos = (Button) findViewById(R.id.b_casos);
        strat = (Button) findViewById(R.id.b_strat);

        casos.setOnClickListener(this);
        strat.setOnClickListener(this);
    }

    @Override
    public void onClick(View v) {
        if (v.getId() == R.id.b_casos) {
            Intent m_casos = new Intent(this,instr_casos.class);
            startActivity(m_casos);
        }
    }
}
```

```

    }
    else if (v.getId() == R.id.b_strat){
        Intent m_strat = new Intent(this,instr_strat.class);
        startActivity(m_strat);
    }
    else{}
}
}

```

### 11.2.3. instr\_casos.java

```

public class instr_casos extends AppCompatActivity implements View.OnClickListener{

    Button continua;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_instr_casos);

        continua = (Button) findViewById(R.id.b_cont_entra_c);

        continua.setOnClickListener(this);
    }

    @Override
    public void onClick(View v){
        Intent m_c_entrada = new Intent(this,entrada_casos.class);
        startActivity(m_c_entrada);
    }
}

```

### 11.2.4. entrada\_casos.java

```

public class entrada_casos extends AppCompatActivity implements View.OnClickListener{

    Button filtro_sol;
    EditText entrada_x0;
    EditText entrada_x;
    EditText entrada_v0;
    EditText entrada_v;
    EditText entrada_a;
    EditText entrada_t;
    String cad_x0;
    String cad_x;
    String cad_v0;
    String cad_v;
    String cad_a;
    String cad_t;
    Float x0;
    Float x;
    Float v0;
    Float ve;
    Float a;
    Float t;
    Toast en_inval;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_entrada_casos);

        filtro_sol = (Button) findViewById(R.id.b_resolver);
        entrada_x0 = (EditText) findViewById(R.id.en_X0);
        entrada_x = (EditText) findViewById(R.id.en_X);
    }
}

```



```

    entrada_v0 = (EditText) findViewById(R.id.en_v0);
    entrada_v = (EditText) findViewById(R.id.en_v);
    entrada_a = (EditText) findViewById(R.id.en_a);
    entrada_t = (EditText) findViewById(R.id.en_t);

    filtro_sol.setOnClickListener(this);
}

@Override
public void onClick(View v){
    if (v.getId() == R.id.b_resolver) {
        cad_x0 = entrada_x0.getText().toString();
        cad_x = entrada_x.getText().toString();
        cad_v0 = entrada_v0.getText().toString();
        cad_v = entrada_v.getText().toString();
        cad_a = entrada_a.getText().toString();
        cad_t = entrada_t.getText().toString();
        // albol de validacion para la entrada (Incognitas)
        if(cad_x0.isEmpty()){
            en_inval.makeText(getApplicationContext(),"Sobran o faltan valores, se
deben especificar 4 (datos) dejando 2 campos vacios (incognitas) X0 siempre es
dato",Toast.LENGTH_LONG).show();
        }
        else{
            if(cad_x.isEmpty()){
                if(cad_v0.isEmpty()){
                    if(cad_v.isEmpty() || cad_a.isEmpty() || cad_t.isEmpty()){
                        en_inval.makeText(getApplicationContext(),"Sobran o faltan
valores, se deben especificar 4 (datos) dejando 2 campos vacios
(incognitas)",Toast.LENGTH_LONG).show();
                    }
                    else{//caso variables x y v0
                        x0 = Float.parseFloat(cad_x0);
                        ve = Float.parseFloat(cad_v);
                        a = Float.parseFloat(cad_a);
                        t = Float.parseFloat(cad_t);
                        Intent select_c = new Intent(this,c_variables_xv0.class);
                        select_c.putExtra("pos_ini",x0);
                        select_c.putExtra("vel",ve);
                        select_c.putExtra("acel",a);
                        select_c.putExtra("tiempo",t);
                        startActivity(select_c);
                    }
                }
                else if(cad_v.isEmpty()){
                    if(cad_a.isEmpty() || cad_t.isEmpty()){
                        en_inval.makeText(getApplicationContext(),"Sobran o faltan
valores, se deben especificar 4 (datos) dejando 2 campos vacios
(incognitas)",Toast.LENGTH_LONG).show();
                    }
                    else{//caso variables x y v
                        x0 = Float.parseFloat(cad_x0);
                        v0 = Float.parseFloat(cad_v0);
                        a = Float.parseFloat(cad_a);
                        t = Float.parseFloat(cad_t);
                        Intent select_c = new Intent(this,c_variables_xv.class);
                        select_c.putExtra("pos_ini",x0);
                        select_c.putExtra("vel_ini",v0);
                        select_c.putExtra("acel",a);
                        select_c.putExtra("tiempo",t);
                        this.startActivity(select_c);
                    }
                }
            }
            else if(cad_a.isEmpty()){
                if(cad_t.isEmpty()){
                    en_inval.makeText(getApplicationContext(),"Sobran o faltan

```

```

valores, se deben especificar 4 (datos) dejando 2 campos vacios
(incognitas)", Toast.LENGTH_LONG).show();
    }
    else{//caso variables x y a
        x0 = Float.parseFloat(cad_x0);
        v0 = Float.parseFloat(cad_v0);
        ve = Float.parseFloat(cad_v);
        t = Float.parseFloat(cad_t);
        Intent select_c = new Intent(this, c_variables_xa.class);
        select_c.putExtra("pos_ini", x0);
        select_c.putExtra("vel_ini", v0);
        select_c.putExtra("vel", ve);
        select_c.putExtra("tiempo", t);
        startActivity(select_c);
    }
}
else if(cad_t.isEmpty()){//caso variables x y t
    x0 = Float.parseFloat(cad_x0);
    v0 = Float.parseFloat(cad_v0);
    ve = Float.parseFloat(cad_v);
    a = Float.parseFloat(cad_a);
    Intent select_c = new Intent(this, c_variables_xt.class);
    select_c.putExtra("pos_ini", x0);
    select_c.putExtra("vel_ini", v0);
    select_c.putExtra("vel", ve);
    select_c.putExtra("acel", a);
    startActivity(select_c);
}
else{
    en_inval.makeText(getApplicationContext(), "Sobran o faltan
valores, se deben especificar 4 (datos) dejando 2 campos vacios
(incognitas)", Toast.LENGTH_LONG).show();
}
}
else if(cad_v0.isEmpty()){
    if(cad_v.isEmpty()){
        if(cad_a.isEmpty() || cad_t.isEmpty()){
            en_inval.makeText(getApplicationContext(), "Sobran o faltan
valores, se deben especificar 4 (datos) dejando 2 campos vacios
(incognitas)", Toast.LENGTH_LONG).show();
        }
        else{//caso variables v0 y v
            x0 = Float.parseFloat(cad_x0);
            x = Float.parseFloat(cad_x);
            a = Float.parseFloat(cad_a);
            t = Float.parseFloat(cad_t);
            Intent select_c = new Intent(this, c_variables_v0v.class);
            select_c.putExtra("pos_ini", x0);
            select_c.putExtra("pos", x);
            select_c.putExtra("acel", a);
            select_c.putExtra("tiempo", t);
            startActivity(select_c);
        }
    }
}
else if(cad_a.isEmpty()){
    if(cad_t.isEmpty()){
        en_inval.makeText(getApplicationContext(), "Sobran o faltan
valores, se deben especificar 4 (datos) dejando 2 campos vacios
(incognitas)", Toast.LENGTH_LONG).show();
    }
    else{//caso variables v0 y a
        x0 = Float.parseFloat(cad_x0);
        x = Float.parseFloat(cad_x);
        ve = Float.parseFloat(cad_v);
        t = Float.parseFloat(cad_t);
        Intent select_c = new Intent(this, c_variables_v0a.class);

```

```

        select_c.putExtra("pos_ini",x0);
        select_c.putExtra("pos",x);
        select_c.putExtra("vel",ve);
        select_c.putExtra("tiempo",t);
        startActivity(select_c);
    }
}
else if(cad_t.isEmpty()){//caso variables v0 y t
    x0 = Float.parseFloat(cad_x0);
    x = Float.parseFloat(cad_x);
    ve = Float.parseFloat(cad_v);
    a = Float.parseFloat(cad_a);
    Intent select_c = new Intent(this,c_variables_v0t.class);
    select_c.putExtra("pos_ini",x0);
    select_c.putExtra("pos",x);
    select_c.putExtra("vel",ve);
    select_c.putExtra("acel",a);
    startActivity(select_c);
}
else{
    en_inval.makeText(getApplicationContext(),"Sobran o faltan
valores, se deben especificar 4 (datos) dejando 2 campos vacios
(incognitas)",Toast.LENGTH_LONG).show();
}
}
else if(cad_v.isEmpty()){
    if(cad_a.isEmpty()){
        if(cad_t.isEmpty()){
            en_inval.makeText(getApplicationContext(),"Sobran o faltan
valores, se deben especificar 4 (datos) dejando 2 campos vacios
(incognitas)",Toast.LENGTH_LONG).show();
        }
        else{//caso variables v y a
            x0 = Float.parseFloat(cad_x0);
            x = Float.parseFloat(cad_x);
            v0 = Float.parseFloat(cad_v0);
            t = Float.parseFloat(cad_t);
            Intent select_c = new Intent(this,c_variables_va.class);
            select_c.putExtra("pos_ini",x0);
            select_c.putExtra("pos",x);
            select_c.putExtra("vel_ini",v0);
            select_c.putExtra("tiempo",t);
            startActivity(select_c);
        }
    }
}
else if(cad_t.isEmpty()){//caso variables v y t
    x0 = Float.parseFloat(cad_x0);
    x = Float.parseFloat(cad_x);
    v0 = Float.parseFloat(cad_v0);
    a = Float.parseFloat(cad_a);
    Intent select_c = new Intent(this,c_variables_vt.class);
    select_c.putExtra("pos_ini",x0);
    select_c.putExtra("pos",x);
    select_c.putExtra("vel_ini",v0);
    select_c.putExtra("acel",a);
    startActivity(select_c);
}
else{
    en_inval.makeText(getApplicationContext(),"Sobran o faltan
valores, se deben especificar 4 (datos) dejando 2 campos vacios
(incognitas)",Toast.LENGTH_LONG).show();
}
}
else if(cad_a.isEmpty()){
    if(cad_t.isEmpty()){//caso variables a y t
        x0 = Float.parseFloat(cad_x0);

```



```

        if(valores!=null){
            x0 = (Float) valores.get("pos_ini");
            ve = (Float) valores.get("vel");
            a = (Float) valores.get("acel");
            t = (Float) valores.get("tiempo");
        }

        v0 = ve-(a*t);
        x = x0+(v0*t)+((a*t*t)/2);

        despliega = "Un movil se encuentra en
X0="+x0.toString()+getResources().getString(R.string.uni_p)+" como posicion inicial y
se comienza a mover con una velocidad inicial de
"+v0.toString()+getResources().getString(R.string.uni_v)+" y una aceleracion constante
de "+a.toString()+getResources().getString(R.string.uni_a)+" , despues de
"+t.toString()+" segundos lleva una velocidad de
"+ve.toString()+getResources().getString(R.string.uni_v)+" . Determina la velocidad con
la que partio de X0 y su posicion despues de "+t.toString()+" segundos.";
        prob.setText(despliega);
        despliega = "Para este tipo de problemas comenzaremos despejando V0 de (ec2):
\n\n V0=V-(a*t) \n y posteriormente sustituiremos: \n\n V0="+ve.toString()+"-
("+a.toString()+"*"+t.toString()+") \n obteniendo asi la primera incognita del
problema.";
        sust_paso1.setText(despliega);
        despliega = "Ahora podemos sustituir directamente en (ec1): \n\n
X="+x0.toString()+"+"+v0.toString()+"*"+t.toString()+"+"+"a.toString()+"*"+t.toString()
+"^2)/2 \n\n obteniendo asi nuestra segunda incognita X="+x.toString()+" y resolviendo
el problema.";
        sust_paso2.setText(despliega);
        despliega = " (R) X="+x.toString()+getResources().getString(R.string.uni_p)+"
\n\n V0="+v0.toString()+getResources().getString(R.string.uni_v);
        sust_sol.setText(despliega);

        ret_menu.setOnClickListener(this);
        ret_entrada.setOnClickListener(this);
    }

    @Override
    public void onClick(View v){
        if (v.getId() == R.id.b_ret_entr) {
            Intent r_ent = new Intent(this, entrada_casos.class);
            startActivity(r_ent);
        }
        else if(v.getId() == R.id.b_ret_menu){
            Intent r_men = new Intent(this,Menu_p.class);
            startActivity(r_men);
        }
        else{}
    }
}

```

## 11.2.6. c\_variables\_xv.java

```
public class c_variables_xv extends AppCompatActivity implements View.OnClickListener{

    String despliega;
    TextView prob;
    TextView sust_pasol;
    TextView sust_paso2;
    TextView sust_sol;
    Button ret_entrada;
    Button ret_menu;
    Float x0;
    Float x;
    Float v0;
    Float ve;
    Float a;
    Float t;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_c_variables_xv);
        Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);

        prob = (TextView) findViewById(R.id.textv_s1);
        sust_pasol = (TextView) findViewById(R.id.textv_s2);
        sust_paso2 = (TextView) findViewById(R.id.textv_s3);
        sust_sol = (TextView) findViewById(R.id.textv_sol_v);
        ret_entrada = (Button) findViewById(R.id.b_ret_entr);
        ret_menu = (Button) findViewById(R.id.b_ret_menu);

        Intent entrada = getIntent();
        Bundle valores = entrada.getExtras();
        if(valores!=null){
            x0 = valores.getFloat("pos_ini");
            v0 = valores.getFloat("vel_ini");
            a = valores.getFloat("acel");
            t = valores.getFloat("tiempo");
        }

        x = x0+(v0*t)+((a*t*t)/2);
        ve = v0+(a*t);

        despliega = "Un movil tiene una velocidad inicial de "+v0.toString()+getResources().getString(R.string.uni_v)+" y parte de la posicion X0="+x0.toString()+getResources().getString(R.string.uni_p)+" , moviendose con una aceleracion de "+a.toString()+getResources().getString(R.string.uni_a)+" . Determina su velocidad y posicion al tiempo t="+t.toString()+getResources().getString(R.string.uni_t)+" .";
        prob.setText(despliega);
        despliega = "Date cuenta que ambas ecuaciones basicas son funciones dependientes de t, esto es que con (ec1) podemos obtener la posicion en cualquiera tiempo y que con (ec2) podemos obtener la velocidad del sistema para cualquiera tiempo, solo es necesario sustituir de manera corecta los valores dados en el enunciado del problema.";
        sust_pasol.setText(despliega);
        despliega = "Al sustituir en (ec1) tenemos que: \n\n X="+x0.toString()+"+"+v0.toString()+"*"+t.toString()+"+"+"+a.toString()+"*"+t.toString()+"^2)/2 \n\n y al sustituir en (ec2)tenemos que: \n\n V="+v0.toString()+"+"+a.toString()+"*"+t.toString()+" \n\n obteniendo asi la posicion y velocidad en t="+t.toString()+getResources().getString(R.string.uni_t)+" .";
        sust_paso2.setText(despliega);
        despliega = " (R) X="+x.toString()+getResources().getString(R.string.uni_p)+" \n\n V="+ve.toString()+getResources().getString(R.string.uni_v);
```

```

sust_sol.setText(despliega);

ret_menu.setOnClickListener(this);
ret_entrada.setOnClickListener(this);
}

@Override
public void onClick(View v){
    if (v.getId() == R.id.b_ret_entr) {
        Intent r_ent = new Intent(this, entrada_casos.class);
        startActivity(r_ent);
    }
    else if(v.getId() == R.id.b_ret_menu){
        Intent r_men = new Intent(this,Menu_p.class);
        startActivity(r_men);
    }
    else{}
}
}
}

```

### 11.2.7. c\_variables\_xa.java

```

public class c_variables_xa extends AppCompatActivity implements View.OnClickListener{

    String despliega;
    TextView prob;
    TextView sust_pasol;
    TextView sust_paso2;
    TextView sust_sol;
    Button ret_entrada;
    Button ret_menu;
    Float x0;
    Float x;
    Float v0;
    Float ve;
    Float a;
    Float t;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_c_variables_xa);
        Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);

        prob = (TextView) findViewById(R.id.textv_s1);
        sust_pasol = (TextView) findViewById(R.id.textv_s2);
        sust_paso2 = (TextView) findViewById(R.id.textv_s3);
        sust_sol = (TextView) findViewById(R.id.textv_sol_v);
        ret_entrada = (Button) findViewById(R.id.b_ret_entr);
        ret_menu = (Button) findViewById(R.id.b_ret_menu);

        Intent entrada = getIntent();
        Bundle valores = entrada.getExtras();
        if(valores!=null){
            x0 = (Float) valores.get("pos_ini");
            v0 = (Float) valores.get("vel_ini");
            ve = (Float) valores.get("vel");
            t = (Float) valores.get("tiempo");
        }

        if(t==0){
            a = (float) 0;
        }
        else{
            a = (ve-v0)/t;
        }
    }
}

```

```

    }
    x = x0+(v0*t)+(a*t*t)/2;

    despliega = "Un movil parte de X0 =
"+x0.toString()+getResources().getString(R.string.uni_p)+" con una velocidad de
"+v0.toString()+getResources().getString(R.string.uni_v)+". Despues de "+t.toString()+"
segundos su velocidad cambia a
"+ve.toString()+getResources().getString(R.string.uni_v)+". Si se sabe que se movio con
aceleracion conctante determina su aceleracion y posicion en t
="+t.toString()+getResources().getString(R.string.uni_t)+".";
    prob.setText(despliega);
    despliega = "Se sustituyen los valores en (ec2) : \n\n
"+ve.toString()+"="+v0.toString()+"a"+t.toString()+" \n\n y despejamos para obtener a:
\n\n a="+ve.toString()+"-"+v0.toString()+"/"+t.toString()+" \n\n resolviendo se tiene
que a="+a.toString()+".";
    sust_pasol.setText(despliega);
    despliega = "Lo siguiente es tomar el resultado y sustutuirlo en (ec1) : \n\n
X="+x0.toString()+"="+v0.toString()+"*"+t.toString()+(" "+a.toString()+"*"+t.toString()
+"^2)/2 \n\n obteniendo directamente la posicion en el instante t="+t.toString()+".";
    sust_paso2.setText(despliega);
    despliega = " (R) X="+x.toString()+getResources().getString(R.string.uni_p)+"
\n\n a="+a.toString()+getResources().getString(R.string.uni_a);
    sust_sol.setText(despliega);

    ret_menu.setOnClickListener(this);
    ret_entrada.setOnClickListener(this);
}

@Override
public void onClick(View v){
    if (v.getId() == R.id.b_ret_entr) {
        Intent r_ent = new Intent(this, entrada_casos.class);
        startActivity(r_ent);
    }
    else if(v.getId() == R.id.b_ret_menu){
        Intent r_men = new Intent(this,Menu_p.class);
        startActivity(r_men);
    }
    else{}
}
}

```

### 11.2.8. c\_variables\_xt.java

```

public class c_variables_xt extends AppCompatActivity implements View.OnClickListener{

    String despliega;
    TextView prob;
    TextView sust_pasol;
    TextView sust_paso2;
    TextView sust_sol;
    Button ret_entrada;
    Button ret_menu;
    Float x0;
    Float x;
    Float v0;
    Float ve;
    Float a;
    Float t;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
    }
}

```



```

setContentView(R.layout.activity_c_variables_xt);
Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
setSupportActionBar(toolbar);

prob = (TextView) findViewById(R.id.textv_s1);
sust_paso1 = (TextView) findViewById(R.id.textv_s2);
sust_paso2 = (TextView) findViewById(R.id.textv_s3);
sust_sol = (TextView) findViewById(R.id.textv_sol_v);
ret_entrada = (Button) findViewById(R.id.b_ret_entr);
ret_menu = (Button) findViewById(R.id.b_ret_menu);

Intent entrada = getIntent();
Bundle valores = entrada.getExtras();
if(valores!=null){
    x0 = (Float) valores.get("pos_ini");
    v0 = (Float) valores.get("vel_ini");
    ve = (Float) valores.get("vel");
    a = (Float) valores.get("acel");
}

t = (ve-v0)/a;
x = x0+(v0*t)+((a*t*t)/2);

despliega = "Un movil se encuentra en
X0="+x0.toString()+getResources().getString(R.string.uni_p)+" con una velocidad inicial
de "+v0.toString()+getResources().getString(R.string.uni_v)+"", comienza a moverse con
aceleracion constante de "+a.toString()+getResources().getString(R.string.uni_a)".
Determina el tiempo en el que alcanza una velocidad de
"+ve.toString()+getResources().getString(R.string.uni_v)+" asi como su posicion en ese
instante.";
prob.setText(despliega);
despliega = "Despejar t de (ec2): \n\n t=(V-V0)/a \n\n y sustituir para obtener
t: \n\n t="+ve.toString()+"-"+v0.toString()+"/"+a.toString()+" \n\n obtenido asi
t="+t.toString()+".";
sust_paso1.setText(despliega);
despliega = "Al sustituir en (ec1) tenemos que: \n\n
X="+x0.toString()+"+"+v0.toString()+"*"+t.toString()+(" "+a.toString()+"*"+t.toString()
+"^2)/2 \n\n dado que esta ecuacion representa posicion para cualquier tiempo: \n\n
X="+x.toString()+" \n\n obteniendo asi la posicion al tiempo t="+t.toString()+",
resolviendo ambas incognitas.";
sust_paso2.setText(despliega);
despliega = " (R) X="+x.toString()+getResources().getString(R.string.uni_p)+"
\n\n t="+t.toString()+getResources().getString(R.string.uni_t);
sust_sol.setText(despliega);

ret_menu.setOnClickListener(this);
ret_entrada.setOnClickListener(this);
}

@Override
public void onClick(View v){
    if (v.getId() == R.id.b_ret_entr) {
        Intent r_ent = new Intent(this, entrada_casos.class);
        startActivity(r_ent);
    }
    else if(v.getId() == R.id.b_ret_menu){
        Intent r_men = new Intent(this,Menu_p.class);
        startActivity(r_men);
    }
    else{}
}
}

```

## 11.2.9. c\_variables\_v0v.java

```
public class c_variables_v0v extends AppCompatActivity implements View.OnClickListener{

    String despliega;
    TextView prob;
    TextView sust_pasol;
    TextView sust_paso2;
    TextView sust_sol;
    Button ret_entrada;
    Button ret_menu;
    Float x0;
    Float x;
    Float v0;
    Float ve;
    Float a;
    Float t;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_c_variables_v0v);
        Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);

        prob = (TextView) findViewById(R.id.textv_s1);
        sust_pasol = (TextView) findViewById(R.id.textv_s2);
        sust_paso2 = (TextView) findViewById(R.id.textv_s3);
        sust_sol = (TextView) findViewById(R.id.textv_sol_v);
        ret_entrada = (Button) findViewById(R.id.b_ret_entr);
        ret_menu = (Button) findViewById(R.id.b_ret_menu);

        Intent entrada = getIntent();
        Bundle valores = entrada.getExtras();
        if(valores!=null){
            x0 = (Float) valores.get("pos_ini");
            x = (Float) valores.get("pos");
            a = (Float) valores.get("acel");
            t = (Float) valores.get("tiempo");
        }

        v0 = (x/t)-(x0/t)-((a*t)/2);
        ve = v0+(a*t);

        despliega = "Un movil se encuentra inicialmente en
X0="+x0.toString()+getResources().getString(R.string.uni_p)+" se comienza a mover con
una aceleracion constante de "+a.toString()+getResources().getString(R.string.uni_a)+"
despues de "+t.toString()+" segundos se encuentra en la posicion
X="+x.toString()+getResources().getString(R.string.uni_p)+". Determina la velocidad con
la que partio de X0 y la velocidad final despues de "+t.toString()+" segundos.";
        prob.setText(despliega);
        despliega = "Despejar V0 de (ec1): \n\n V0=(X-X0-(a*t^2)*2)/t \n\n asi podemos
sustituir los datos del problema: \n\n V0="+x.toString()+"/"+t.toString()+")-
("+x0.toString()+"/"+t.toString()+")-("+a.toString()+"*"+t.toString()+"/2) \n\n
obteniendo asi el valor de V0.";
        sust_pasol.setText(despliega);
        despliega = "Luego podemos utilizar (ec2) para obtener V: \n\n
V="+v0.toString()+""+a.toString()+""+t.toString()+"" \n\n dado que (ec2) representa
velocidad para todo t, una vez obtenido el valor de V se resuelve el problema.";
        sust_paso2.setText(despliega);
        despliega = " (R) V0="+v0.toString()+getResources().getString(R.string.uni_v)+"
\n\n V="+ve.toString()+getResources().getString(R.string.uni_v);
        sust_sol.setText(despliega);

        ret_menu.setOnClickListener(this);
    }
}
```

```

        ret_entrada.setOnClickListener(this);
    }

    @Override
    public void onClick(View v) {
        if (v.getId() == R.id.b_ret_entr) {
            Intent r_ent = new Intent(this, entrada_casos.class);
            startActivity(r_ent);
        }
        else if (v.getId() == R.id.b_ret_menu) {
            Intent r_men = new Intent(this, Menu_p.class);
            startActivity(r_men);
        }
        else {}
    }
}

```

### 11.2.10. c\_variables\_v0a.java

```

public class c_variables_v0a extends AppCompatActivity implements View.OnClickListener{

    String despliega;
    TextView prob;
    TextView sust_pasol;
    TextView sust_paso2;
    TextView sust_sol;
    Button ret_entrada;
    Button ret_menu;
    Float x0;
    Float x;
    Float v0;
    Float ve;
    Float a;
    Float t;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_c_variables_v0a);
        Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);

        prob = (TextView) findViewById(R.id.textv_s1);
        sust_pasol = (TextView) findViewById(R.id.textv_s2);
        sust_paso2 = (TextView) findViewById(R.id.textv_s3);
        sust_sol = (TextView) findViewById(R.id.textv_sol_v);
        ret_entrada = (Button) findViewById(R.id.b_ret_entr);
        ret_menu = (Button) findViewById(R.id.b_ret_menu);

        Intent entrada = getIntent();
        Bundle valores = entrada.getExtras();
        if(valores!=null){
            x0 = (Float) valores.get("pos_ini");
            x = (Float) valores.get("pos");
            ve = (Float) valores.get("vel");
            t = (Float) valores.get("tiempo");
        }

        if(t!=0){
            v0 = (2*(x-x0-(ve*t)/2))/t;
            a = (ve-v0)/t;
        }
        else{
            v0 = ve;
            a = (float) 0;
        }
    }
}

```

```

        despliega = "Un movil se encuentra en su posicion inicial
X0="+x0.toString()+getResources().getString(R.string.uni_p)+" , despues de
"+t.toString()+" segundos adquiere una velocidad de
"+ve.toString()+getResources().getString(R.string.uni_v)+" alcanzando la posicion de
X="+x.toString()+getResources().getString(R.string.uni_p)". Determina su velocidad
inicial asi como su aceleracion teniendo en cuenta que esta es constante.";
        prob.setText(despliega);
        despliega = "Podemos despejar a de (ec2): \n\n a=(V-V0)/t \n\n la cual podemos
llamar (ec3) y sustituir en (ec1): \n\n X=X0+V0*t+((V-V0)/t)*t^2/2 \n\n desarrollando
el tercer termino tenemos que: \n\n X=X0+V0*t+(V*t)/2)-(V0*t)/2 \n\n agrupando los
terminos que contienen a V0 y despejando el termino: \n\n (V0*t)/2=X-X0-(V*t)/2 \n\n
despejando V0 podremos sustituir: \n\n V0=2*( "+x.toString()+"-"+x0.toString()+"-
("+ve.toString()+"*"+t.toString()+")/2)/t \n\n obteniendo V0="+v0.toString()+".";
        sust_pasol.setText(despliega);
        despliega = "Apartir de (ec2) o bien de (ec3) si no queremos despejar
nuevamente: \n\n a=( "+ve.toString()+"-"+v0.toString()+")/"+t.toString()+" \n\n
resolviendo ambas incognitas en el problema.";
        sust_paso2.setText(despliega);
        despliega = " (R) V0="+v0.toString()+getResources().getString(R.string.uni_v)+"
\n\n a="+a.toString()+getResources().getString(R.string.uni_a);
        sust_sol.setText(despliega);

        ret_menu.setOnClickListener(this);
        ret_entrada.setOnClickListener(this);
    }

    @Override
    public void onClick(View v){
        if (v.getId() == R.id.b_ret_entr) {
            Intent r_ent = new Intent(this, entrada_casos.class);
            startActivity(r_ent);
        }
        else if(v.getId() == R.id.b_ret_menu){
            Intent r_men = new Intent(this,Menu_p.class);
            startActivity(r_men);
        }
        else{}
    }
}

```

### 11.2.11. c\_variables\_v0t.java

```

public class c_variables_v0t extends AppCompatActivity implements View.OnClickListener{

    String despliega;
    TextView prob;
    TextView sust_pasol;
    TextView sust_paso2;
    TextView sust_sol;
    Button ret_entrada;
    Button ret_menu;
    Float x0;
    Float x;
    Float v0_1;
    Float v0_2;
    Float ve;
    Float a;
    Float t1;
    Float t2;
    Toast negativo;
}

```

```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_c_variables_v0t);
    Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
    setSupportActionBar(toolbar);

    prob = (TextView) findViewById(R.id.textv_s1);
    sust_paso1 = (TextView) findViewById(R.id.textv_s2);
    sust_paso2 = (TextView) findViewById(R.id.textv_s3);
    sust_sol = (TextView) findViewById(R.id.textv_sol_v);
    ret_entrada = (Button) findViewById(R.id.b_ret_entr);
    ret_menu = (Button) findViewById(R.id.b_ret_menu);

    Intent entrada = getIntent(); //se obtienen datos en
    Bundle valores = entrada.getExtras(); //entrada por referencia
    if(valores!=null){
        x0 = (Float) valores.get("pos_ini");
        x = (Float) valores.get("pos");
        ve = (Float) valores.get("vel");
        a = (Float) valores.get("acel");
    }

    Float A = -a/2;
    Float B = ve;
    Float C = x0-x;
    Float det = (B*B)-(4*A*C);
    Double raiz = Math.sqrt(det);

    if(det == 0){
        t1 = -(B)/(2*A);
        t2 = -(B)/(2*A);
    }
    else if(det > 0){
        t1 = (float) (-(B)+raiz)/(2*A);
        t2 = (float) (-(B)-raiz)/(2*A);
    }
    else{
        t1 = (float) 0;
        t2 = (float) 0;
        negativo.makeText(getApplicationContext(),"El determinante de la ecuacion
de segundo grado asociada al tiempo es negativo. Esto significa que el problema nunca se
presenta en la naturaleza pues los tiempos son numeros complejos: (a+bi y a-bi) donde i
es raiz cuadrada de -1.",Toast.LENGTH_LONG).show();
    }

    v0_1 = ve - (a*t1);
    v0_2 = ve - (a*t2);

    despliega = "Un movil se encuentra en la posicion
X0="+x0.toString()+getResources().getString(R.string.uni_p)+" y se mueve con una
aceleracion de "+a.toString()+getResources().getString(R.string.uni_a)+" , pasado un
tiempo cuando pasa por la posicion
X="+x.toString()+getResources().getString(R.string.uni_p)+" lleva una velocidad de
"+ve.toString()+getResources().getString(R.string.uni_v)+". Determina la velocidad
inicial con la que partio y el tiempo que tardo en llegar a
X="+x.toString()+getResources().getString(R.string.uni_p)+".";
    prob.setText(despliega);
    despliega = "Despejar V0 de (ec2) como se muestra a continuacion: \n\n V0=V-a*t
\n\n sustituirla en (ec1) de la siguiente manera: \n\n X=X0+(V-(a*t))*t+(a*t^2)/2 \n\n
se realiza el algebra antes de sustituir valores: \n\n X=X0+(V*t)-(a*t^2)+(a*t^2)/2
\n\n separaremos los terminos asociandolos a t y sus potencias: \n\n t^2*(-
a/2)+t*(V)+X0-X=0 \n\n la ecuacion de segundo grado sobre t se sustituye: \n\n t^2*(-
"+a.toString()+")/2)+t*( "+ve.toString()+")+"+x0.toString()+"-"+x.toString()+ "=0 \n\n
resolviendo la ecuacion para t tenemos que t1="+t1.toString()+ " y
t2="+t2.toString()+".";

```

```

        sust_pasol.setText(despliega);
        despliega = "Sustituir t en (ec2): \n\n V=V0+a*t ----> V0=V-a*t \n\n al tener
dos valores de t se deben sustituir ambos en la ecuacion para generar dos pares de
soluciones: \n\n V0(t1)="+ve.toString()+"-"+a.toString()+"*"+t1.toString()+" y
V0(t2)="+ve.toString()+"-"+a.toString()+"*"+t2.toString()+" \n\n obteniendo asi la
solucion del problema.";
        sust_paso2.setText(despliega);
        despliega = " (1)
V0="+v0_1.toString()+getResources().getString(R.string.uni_v)+" \t
t="+t1.toString()+getResources().getString(R.string.uni_t)+" \n\n (2)
V0="+v0_2.toString()+getResources().getString(R.string.uni_v)+" \t
t="+t2.toString()+getResources().getString(R.string.uni_t);
        sust_sol.setText(despliega);

        ret_menu.setOnClickListener(this);
        ret_entrada.setOnClickListener(this);
    }

    @Override
    public void onClick(View v){
        if (v.getId() == R.id.b_ret_entr) {
            Intent r_ent = new Intent(this, entrada_casos.class);
            startActivity(r_ent);
        }
        else if(v.getId() == R.id.b_ret_menu){
            Intent r_men = new Intent(this,Menu_p.class);
            startActivity(r_men);
        }
        else{}
    }
}

```

### 11.2.12. c\_variables\_va.java

```

public class c_variables_va extends AppCompatActivity implements View.OnClickListener{

    String despliega;
    TextView prob;
    TextView sust_pasol;
    TextView sust_paso2;
    TextView sust_sol;
    Button ret_entrada;
    Button ret_menu;
    Float x0;
    Float x;
    Float v0;
    Float ve;
    Float a;
    Float t;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_c_variables_va);
        Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);

        prob = (TextView) findViewById(R.id.textv_s1);
        sust_pasol = (TextView) findViewById(R.id.textv_s2);
        sust_paso2 = (TextView) findViewById(R.id.textv_s3);
        sust_sol = (TextView) findViewById(R.id.textv_sol_v);
        ret_entrada = (Button) findViewById(R.id.b_ret_entr);
        ret_menu = (Button) findViewById(R.id.b_ret_menu);
    }
}

```

```

Intent entrada = getIntent();
Bundle valores = entrada.getExtras();
if(valores!=null){
    x0 = (Float) valores.get("pos_ini");
    x = (Float) valores.get("pos");
    v0 = (Float) valores.get("vel_ini");
    t = (Float) valores.get("tiempo");
}

if(t==0){
    a = (float) 0;
}
else{
    a = (x-x0-(v0*t))/(t*t);
}
ve = v0+(a*t);

despliega = "Un movil parte de
X0="+x0.toString()+getResources().getString(R.string.uni_p)+" con una velocidad de
"+v0.toString()+getResources().getString(R.string.uni_v)+" y se desplaza durante
"+t.toString()+" segundos hasta alcanzar la posicion
X="+x.toString()+getResources().getString(R.string.uni_p)+". si se sabe que se desplaza
con aceleracion constante determina su velocidad en ese instante y la aceleracion que
lleva en todo momento.";
prob.setText(despliega);
despliega = "Comenzamos despejando a de (ec1): \n\n a=2(X-X0-V0t)/t^2 \n\n
para despues sustituir los datos del problema: \n\n a="+x.toString()+"-
"+x0.toString()+"-"+v0.toString()+"*"+t.toString()+"/"+t.toString()+"^2 \n\n
obteniendo directamente la aceleracion.";
sust_pasol.setText(despliega);
despliega = "Sustituir a en (ec2) junto con V0 y t para obtener V: \n\n
V="+v0.toString()+"+"+a.toString()+"*"+t.toString()+" \n\n resolviendo el problema.";
sust_paso2.setText(despliega);
despliega = " (R) V="+ve.toString()+getResources().getString(R.string.uni_v)+"
\n\n a="+a.toString()+getResources().getString(R.string.uni_a);
sust_sol.setText(despliega);

ret_menu.setOnClickListener(this);
ret_entrada.setOnClickListener(this);
}

@Override
public void onClick(View v){
    if (v.getId() == R.id.b_ret_entr) {
        Intent r_ent = new Intent(this, entrada_casos.class);
        startActivity(r_ent);
    }
    else if(v.getId() == R.id.b_ret_menu){
        Intent r_men = new Intent(this,Menu_p.class);
        startActivity(r_men);
    }
    else{}
}
}

```

### 11.2.13. c\_variables\_vt.java

```
public class c_variables_vt extends AppCompatActivity implements View.OnClickListener{

    String despliega;
    TextView prob;
    TextView sust_pasol;
    TextView sust_paso2;
    TextView sust_sol;
    Button ret_entrada;
    Button ret_menu;
    Float x0;
    Float x;
    Float v0;
    Float v1;
    Float v2;
    Float a;
    Float t1;
    Float t2;
    Toast negativo;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_c_variables_vt);
        Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);

        prob = (TextView) findViewById(R.id.textv_s1);
        sust_pasol = (TextView) findViewById(R.id.textv_s2);
        sust_paso2 = (TextView) findViewById(R.id.textv_s3);
        sust_sol = (TextView) findViewById(R.id.textv_sol_v);
        ret_entrada = (Button) findViewById(R.id.b_ret_entr);
        ret_menu = (Button) findViewById(R.id.b_ret_menu);

        Intent entrada = getIntent();
        Bundle valores = entrada.getExtras();
        if(valores!=null){
            x0 = (Float) valores.get("pos_ini");
            x = (Float) valores.get("pos");
            v0 = (Float) valores.get("vel_ini");
            a = (Float) valores.get("acel");
        }

        Float A = a/2;
        Float B = v0;
        Float C = x0-x;
        Float det = (B*B)-4*A*C;
        Double raiz = Math.sqrt(det);

        if(det == 0){
            t1 = -(B)/(2*A);
            t2 = -(B)/(2*A);
        }
        else if(det > 0){
            t1 = (float) (-(B)+raiz)/(2*A);
            t2 = (float) (-(B)-raiz)/(2*A);
        }
        else{
            t1 = (float) 0;
            t2 = (float) 0;
            negativo.makeText(getApplicationContext(),"El determinante de la ecuacion
de segundo grado asociada al tiempo es negativo. Esto significa que el problema nunca se
presenta en la naturaleza pues los tiempos son numeros complejos: (a+bi y a-bi) donde i
es raiz cuadrada de -1.",Toast.LENGTH_LONG).show();
        }
    }
}
```



```

    }

    v1 = v0 + (a*t1);
    v2 = v0 + (a*t2);

    despliega = "Un movil tiene una velocidad de
"+v0.toString()+getResources().getString(R.string.uni_v)+" y se encuentra en X0 =
"+x0.toString()+getResources().getString(R.string.uni_p)+" cuando se comienza a mover
con una aceleracion de "+a.toString()+getResources().getString(R.string.uni_a)".
Determina el tiempo que tarda en llegar a X =
"+x.toString()+getResources().getString(R.string.uni_p)+" y la velocidad que tiene en
este punto.";
    prob.setText(despliega);
    despliega = "Se sustituyen los valores en (ec1) : \n\n
"+x.toString()+"="+x0.toString()+"+"+v0.toString()+"t+("+a.toString()+"t^2)/2 \n\n para
obtener los valores de t resolvemos la ecuacion de 2do grado: \n\n
"+A.toString()+"t^2"+B.toString()+"t"+C.toString()+"=0 \n\n obteniendo
t1="+t1.toString()+" y t2="+t2.toString()+".";
    sust_paso1.setText(despliega);
    despliega = "Lo siguiente es tomar estos valores y sustutuirlos en (ec2) esto
genera 2 pares de soluciones: \n\n
V1="+v0.toString()+"+"+a.toString()+"*"+t1.toString()+"\n
V2="+v0.toString()+"+"+a.toString()+"*"+t2.toString()+" \n\n siendo estas las
soluciones del problema.";
    sust_paso2.setText(despliega);
    despliega = " (1) V="+v1.toString()+getResources().getString(R.string.uni_v)+"
\t t="+t1.toString()+getResources().getString(R.string.uni_t)+" \n\n (2)
V="+v1.toString()+getResources().getString(R.string.uni_v)+" \t
t="+t2.toString()+getResources().getString(R.string.uni_t);
    sust_sol.setText(despliega);

    ret_menu.setOnClickListener(this);
    ret_entrada.setOnClickListener(this);
}

@Override
public void onClick(View v){
    if (v.getId() == R.id.b_ret_entr) {
        Intent r_ent = new Intent(this, entrada_casos.class);
        startActivity(r_ent);
    }
    else if(v.getId() == R.id.b_ret_menu){
        Intent r_men = new Intent(this,Menu_p.class);
        startActivity(r_men);
    }
    else{}
}
}
}

```

## 11.2.14. c\_variables\_at.java

```
public class c_variables_at extends AppCompatActivity implements View.OnClickListener{

    String despliega;
    TextView prob;
    TextView sust_pasol;
    TextView sust_paso2;
    TextView sust_sol;
    Button ret_entrada;
    Button ret_menu;
    Float x0;
    Float x;
    Float v0;
    Float ve;
    Float a;
    Float t;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_c_variables_at);
        Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);

        prob = (TextView) findViewById(R.id.textv_s1);
        sust_pasol = (TextView) findViewById(R.id.textv_s2);
        sust_paso2 = (TextView) findViewById(R.id.textv_s3);
        sust_sol = (TextView) findViewById(R.id.textv_sol_v);
        ret_entrada = (Button) findViewById(R.id.b_ret_entr);
        ret_menu = (Button) findViewById(R.id.b_ret_menu);

        Intent entrada = getIntent();
        Bundle valores = entrada.getExtras();
        if(valores!=null){
            x0 = (Float) valores.get("pos_ini");
            x = (Float) valores.get("pos");
            v0 = (Float) valores.get("vel_ini");
            ve = (Float) valores.get("vel");
        }

        a = ((ve*ve)-(v0*v0))/(2*(x-x0));
        t = (ve-v0)/a;

        despliega = "Un movil parte de un punto
X0="+x0.toString()+getResources().getString(R.string.uni_p)+" con una velocidad inicial
de "+v0.toString()+getResources().getString(R.string.uni_v)+" cuando alcanza la
posicion X="+x.toString()+getResources().getString(R.string.uni_p)+" tiene una
velocidad de "+ve.toString()+getResources().getString(R.string.uni_v)". Si se sabe que
el movil lleva siempre aceleracion constante determina el tiempo que le toma ir de X0 a
X asi como su aceleracion.";
        prob.setText(despliega);
        despliega = "Despejar t de (ec2): \n\n t=(V-V0)/a \n\n la cual llamaremos (ec3)
y substituiremos en (ec1): \n\n X=X0+(V0(V-V0)/a)+((a*(V-V0)^2)/a^2 \n\n simplificar:
\n\n X=X0+(2(V*V0))/2a+(V^2+2(V*V0)+V0^2)/2a - los terminos 2 y 3 tienen el mismo
denominador \n desarrollando terminos y restando: \n\n X=X0+(V^2 - V0^2)/2a \n\n
despejar a (ec4): \n\n a=(V^2-V0^2)/2(X-X0).";
        sust_pasol.setText(despliega);
        despliega = "Substituir (ec4) con los valores del problema: \n\n
a="+ve.toString()+"^2-"+v0.toString()+"^2)/2("+x.toString()+"-"+x0.toString()+") \n\n
luego substituímos a en (ec3): \n\n t=("+ve.toString()+"-
"+v0.toString()+")/"+a.toString()+") \n\n obteniendo así ambas incógnitas.";
        sust_paso2.setText(despliega);
        despliega = " (R) a="+a.toString()+getResources().getString(R.string.uni_a)+"
\n\n t="+t.toString()+getResources().getString(R.string.uni_t);
```

```

sust_sol.setText(despliega);

ret_menu.setOnClickListener(this);
ret_entrada.setOnClickListener(this);
}

@Override
public void onClick(View v){
    if (v.getId() == R.id.b_ret_entr) {
        Intent r_ent = new Intent(this, entrada_casos.class);
        startActivity(r_ent);
    }
    else if(v.getId() == R.id.b_ret_menu){
        Intent r_men = new Intent(this,Menu_p.class);
        startActivity(r_men);
    }
    else{}
}
}
}

```

### 11.2.15. instr\_strat.java

```

public class instr_strat extends AppCompatActivity implements View.OnClickListener{

    Button continua;
    Random rand_select;
    int tipo;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_instr_strat);

        continua = (Button) findViewById(R.id.b_cont_estrat);
        rand_select = new Random(System.currentTimeMillis());
        tipo = rand_select.nextInt(3);
        continua.setOnClickListener(this);
        rand_select.setSeed(System.currentTimeMillis());
    }

    @Override
    public void onClick(View v){
        if(tipo==0){
            Intent m_c_entrada = new Intent(this,strat_1.class);
            startActivity(m_c_entrada);
        }
        else if(tipo==1){
            Intent m_c_entrada = new Intent(this,strat_6.class);
            startActivity(m_c_entrada);
        }
        else if(tipo==2){
            Intent m_c_entrada = new Intent(this,strat_11.class);
            startActivity(m_c_entrada);
        }
        else{}
    }
}
}

```

## 11.2.16. strat\_1.java

```
public class strat_1 extends AppCompatActivity implements View.OnClickListener{

    Button avanza;
    Button incorrecta;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_strat_1);

        incorrecta = (Button) findViewById(R.id.b1_opc_1);
        avanza = (Button) findViewById(R.id.b1_opc_2);

        avanza.setOnClickListener(this);
        incorrecta.setOnClickListener(this);
    }

    @Override
    public void onClick(View v){
        if (v.getId() == R.id.b1_opc_1){
            Intent retro = new Intent(this, strat_rA.class);
            startActivity(retro);
        }
        else if (v.getId() == R.id.b1_opc_2) {
            Intent siguiente = new Intent(this, strat_2.class);
            startActivity(siguiente);
        }
        else{}
    }
}
```

## 11.2.17. strat\_2.java

```
public class strat_2 extends AppCompatActivity implements View.OnClickListener{

    Button avanza;
    Button incorrecta_1;
    Button incorrecta_2;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_strat_2);

        incorrecta_1 = (Button) findViewById(R.id.b2_opc_1);
        avanza = (Button) findViewById(R.id.b2_opc_2);
        incorrecta_2 = (Button) findViewById(R.id.b2_opc_3);

        avanza.setOnClickListener(this);
        incorrecta_1.setOnClickListener(this);
        incorrecta_2.setOnClickListener(this);
    }

    @Override
    public void onClick(View v){
        if (v.getId() == R.id.b2_opc_1) {
            Intent siguiente = new Intent(this, strat_rB.class);
            startActivity(siguiente);
        }
        else if (v.getId() == R.id.b2_opc_2){
```

```

        Intent retro = new Intent(this, strat_3.class);
        startActivity(retro);
    }
    else if(v.getId() == R.id.b2_opc_3){
        Intent retro = new Intent(this, strat_rC.class);
        startActivity(retro);
    }
    else{}
}
}

```

### 11.2.18. strat\_3.java

```

public class strat_3 extends AppCompatActivity implements View.OnClickListener{

    Button avanza;
    Button incorrecta_1;
    Button incorrecta_2;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_strat_3);

        incorrecta_1 = (Button) findViewById(R.id.b3_opc_1);
        avanza = (Button) findViewById(R.id.b3_opc_2);
        incorrecta_2 = (Button) findViewById(R.id.b3_opc_3);

        avanza.setOnClickListener(this);
        incorrecta_1.setOnClickListener(this);
        incorrecta_2.setOnClickListener(this);
    }

    @Override
    public void onClick(View v){
        if (v.getId() == R.id.b3_opc_1) {
            Intent siguiente = new Intent(this, strat_rD.class);
            startActivity(siguiente);
        }
        else if (v.getId() == R.id.b3_opc_2){
            Intent retro = new Intent(this, strat_4.class);
            startActivity(retro);
        }
        else if(v.getId() == R.id.b3_opc_3){
            Intent retro = new Intent(this, strat_rE.class);
            startActivity(retro);
        }
        else{}
    }
}
}

```

### 11.2.19. strat\_4.java

```
public class strat_4 extends AppCompatActivity implements View.OnClickListener{

    Button avanza;
    Button incorrecta;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_strat_4);

        incorrecta = (Button) findViewById(R.id.b4_opc_1);
        avanza = (Button) findViewById(R.id.b4_opc_2);

        avanza.setOnClickListener(this);
        incorrecta.setOnClickListener(this);
    }

    @Override
    public void onClick(View v){
        if (v.getId() == R.id.b4_opc_1) {
            Intent siguiente = new Intent(this, strat_rF.class);
            startActivity(siguiente);
        }
        else if (v.getId() == R.id.b4_opc_2){
            Intent retro = new Intent(this, strat_5.class);
            startActivity(retro);
        }
        else{}
    }
}
```

### 11.2.20. strat\_5.java

```
public class strat_5 extends AppCompatActivity implements View.OnClickListener{

    Button otrop;
    Button menu;
    Random rand_select;
    int tipo;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_strat_5);

        otrop = (Button) findViewById(R.id.b_otro_p);
        menu = (Button) findViewById(R.id.b_ret_menu);
        rand_select = new Random(System.currentTimeMillis());
        tipo = rand_select.nextInt(2);
        otrop.setOnClickListener(this);
        menu.setOnClickListener(this);
        rand_select.setSeed(System.currentTimeMillis());
    }

    @Override
    public void onClick(View v){
        if (v.getId() == R.id.b_otro_p){
            if(tipo==0){
                Intent problema = new Intent(this, strat_6.class);
                startActivity(problema);
            }
        }
    }
}
```

```

    }
    if(tipo==1){
        Intent problema = new Intent(this, strat_11.class);
        startActivity(problema);
    }
}
else if (v.getId() == R.id.b_ret_menu){
    Intent sale_m = new Intent(this,Menu_p.class);
    startActivity(sale_m);
}
else{}
}
}
}

```

### 11.2.21. strat\_6.java

```

public class strat_6 extends AppCompatActivity implements View.OnClickListener{

    Button avanza;
    Button incorrecta;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_strat_6);

        incorrecta = (Button) findViewById(R.id.b6_opc_1);
        avanza = (Button) findViewById(R.id.b6_opc_2);

        avanza.setOnClickListener(this);
        incorrecta.setOnClickListener(this);
    }

    @Override
    public void onClick(View v){
        if (v.getId() == R.id.b6_opc_1) {
            Intent retro = new Intent(this,strat_rG.class);
            startActivity(retro);
        }
        else if (v.getId() == R.id.b6_opc_2){
            Intent siguiente = new Intent(this,strat_7.class);
            startActivity(siguiente);
        }
        else{}
    }
}
}

```

## 11.2.22. strat\_7.java

```
public class strat_7 extends AppCompatActivity implements View.OnClickListener{

    Button avanza;
    Button incorrecta;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_strat_7);

        avanza = (Button) findViewById(R.id.b7_opc_1);
        incorrecta = (Button) findViewById(R.id.b7_opc_2);

        avanza.setOnClickListener(this);
        incorrecta.setOnClickListener(this);
    }

    @Override
    public void onClick(View v){
        if (v.getId() == R.id.b7_opc_1) {
            Intent siguiente = new Intent(this, strat_8.class);
            startActivity(siguiente);
        }
        else if (v.getId() == R.id.b7_opc_2){
            Intent retro = new Intent(this, strat_rH.class);
            startActivity(retro);
        }
        else{}
    }
}
```

## 11.2.23. strat\_8.java

```
public class strat_8 extends AppCompatActivity implements View.OnClickListener{

    Button avanza;
    Button incorrecta_1;
    Button incorrecta_2;
    Button incorrecta_3;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_strat_8);

        incorrecta_1 = (Button) findViewById(R.id.b8_opc_1);
        incorrecta_2 = (Button) findViewById(R.id.b8_opc_2);
        avanza = (Button) findViewById(R.id.b8_opc_3);
        incorrecta_3 = (Button) findViewById(R.id.b8_opc_4);

        avanza.setOnClickListener(this);
        incorrecta_1.setOnClickListener(this);
        incorrecta_2.setOnClickListener(this);
        incorrecta_3.setOnClickListener(this);
    }

    @Override
    public void onClick(View v){
        if (v.getId() == R.id.b8_opc_1) {
            Intent retro = new Intent(this, strat_rI.class);
        }
    }
}
```



```

        startActivity(retro);
    }
    else if (v.getId() == R.id.b8_opc_2){
        Intent retro = new Intent(this, strat_rI.class);
        startActivity(retro);
    }
    else if (v.getId() == R.id.b8_opc_3){
        Intent siguiente = new Intent(this, strat_9.class);
        startActivity(siguiente);
    }
    else if (v.getId() == R.id.b8_opc_4){
        Intent retro = new Intent(this, strat_rI.class);
        startActivity(retro);
    }
    else{}
}
}
}

```

### 11.2.24. strat\_9.java

```

public class strat_9 extends AppCompatActivity implements View.OnClickListener{

    Button avanza;
    Button incorrecta;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_strat_9);

        incorrecta = (Button) findViewById(R.id.b9_opc_1);
        avanza = (Button) findViewById(R.id.b9_opc_2);

        avanza.setOnClickListener(this);
        incorrecta.setOnClickListener(this);
    }

    @Override
    public void onClick(View v){
        if (v.getId() == R.id.b9_opc_1) {
            Intent retro = new Intent(this, strat_rJ.class);
            startActivity(retro);
        }
        else if (v.getId() == R.id.b9_opc_2){
            Intent siguiente = new Intent(this, strat_10.class);
            startActivity(siguiente);
        }
        else{}
    }
}
}

```

## 11.2.25. strat\_10.java

```
public class strat_10 extends AppCompatActivity implements View.OnClickListener{

    Button otro_p;
    Button menu;
    Random rand_select;
    int tipo;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_strat_10);

        otro_p = (Button) findViewById(R.id.b_otro_p);
        menu = (Button) findViewById(R.id.b_ret_menu);
        rand_select = new Random(System.currentTimeMillis());
        tipo = rand_select.nextInt(2);
        otro_p.setOnClickListener(this);
        menu.setOnClickListener(this);
        rand_select.setSeed(System.currentTimeMillis());
    }

    @Override
    public void onClick(View v){
        if (v.getId() == R.id.b_otro_p){
            if(tipo==0){
                Intent problema = new Intent(this, strat_1.class);
                startActivity(problema);
            }
            if(tipo==1){
                Intent problema = new Intent(this, strat_11.class);
                startActivity(problema);
            }
        }
        else if (v.getId() == R.id.b_ret_menu){
            Intent sale_m = new Intent(this, Menu_p.class);
            startActivity(sale_m);
        }
        else{}
    }
}
```

## 11.2.26. strat\_11.java

```
public class strat_11 extends AppCompatActivity implements View.OnClickListener{

    Button avanza;
    Button incorrecta;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_strat_11);

        incorrecta = (Button) findViewById(R.id.b11_opc_1);
        avanza = (Button) findViewById(R.id.b11_opc_2);

        avanza.setOnClickListener(this);
        incorrecta.setOnClickListener(this);
    }
}
```

```

@Override
public void onClick(View v) {
    if (v.getId() == R.id.b11_opc_1) {
        Intent retro = new Intent(this, strat_rK.class);
        startActivity(retro);
    }
    else if (v.getId() == R.id.b11_opc_2) {
        Intent siguiente = new Intent(this, strat_12.class);
        startActivity(siguiente);
    }
    else {}
}
}

```

### 11.2.27. strat\_12.java

```

public class strat_12 extends AppCompatActivity implements View.OnClickListener{

    Button avanza;
    Button incorrecta;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_strat_12);

        avanza = (Button) findViewById(R.id.b12_opc_1);
        incorrecta = (Button) findViewById(R.id.b12_opc_2);

        avanza.setOnClickListener(this);
        incorrecta.setOnClickListener(this);
    }

    @Override
    public void onClick(View v) {
        if (v.getId() == R.id.b12_opc_1) {
            Intent siguiente = new Intent(this, strat_13.class);
            startActivity(siguiente);
        }
        else if (v.getId() == R.id.b12_opc_2) {
            Intent retro = new Intent(this, strat_rL.class);
            startActivity(retro);
        }
        else {}
    }
}

```

## 11.2.28. strat\_13.java

```
public class strat_13 extends AppCompatActivity implements View.OnClickListener{

    Button avanza;
    Button incorrecta_1;
    Button incorrecta_2;
    Button incorrecta_3;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_strat_13);

        incorrecta_1 = (Button) findViewById(R.id.b13_opc_1);
        avanza = (Button) findViewById(R.id.b13_opc_2);
        incorrecta_2 = (Button) findViewById(R.id.b13_opc_3);
        incorrecta_3 = (Button) findViewById(R.id.b13_opc_4);

        avanza.setOnClickListener(this);
        incorrecta_1.setOnClickListener(this);
        incorrecta_2.setOnClickListener(this);
        incorrecta_3.setOnClickListener(this);
    }

    @Override
    public void onClick(View v){
        if (v.getId() == R.id.b13_opc_1) {
            Intent retro = new Intent(this, strat_rM.class);
            startActivity(retro);
        }
        else if (v.getId() == R.id.b13_opc_2){
            Intent siguiente = new Intent(this, strat_14.class);
            startActivity(siguiente);
        }
        else if (v.getId() == R.id.b13_opc_3){
            Intent retro = new Intent(this, strat_rN.class);
            startActivity(retro);
        }
        else if (v.getId() == R.id.b13_opc_4){
            Intent retro = new Intent(this, strat_rO.class);
            startActivity(retro);
        }
        else{}
    }
}
```

### 11.2.29. strat\_14.java

```
public class strat_14 extends AppCompatActivity implements View.OnClickListener{

    Button otrop;
    Button menu;
    Random rand_select;
    int tipo;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_strat_14);

        otrop = (Button) findViewById(R.id.b_otro_p);
        menu = (Button) findViewById(R.id.b_ret_menu);
        rand_select = new Random(System.currentTimeMillis());
        tipo = rand_select.nextInt(2);
        otrop.setOnClickListener(this);
        menu.setOnClickListener(this);
        rand_select.setSeed(System.currentTimeMillis());
    }

    @Override
    public void onClick(View v){
        if (v.getId() == R.id.b_otro_p){
            if(tipo==0){
                Intent problema = new Intent(this, strat_1.class);
                startActivity(problema);
            }
            if(tipo==1){
                Intent problema = new Intent(this, strat_6.class);
                startActivity(problema);
            }
        }
        else if (v.getId() == R.id.b_ret_menu){
            Intent sale_m = new Intent(this, Menu_p.class);
            startActivity(sale_m);
        }
        else{}
    }
}
```

### 11.2.30. strat\_rA.java

```
public class strat_rA extends AppCompatActivity implements View.OnClickListener{

    Button ret;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_strat_r_a);

        ret = (Button) findViewById(R.id.b_ret);
        ret.setOnClickListener(this);
    }

    @Override
    public void onClick(View v){
        Intent otro = new Intent(this, strat_1.class);
        startActivity(otro);
    }
}
```

### 11.2.31. strat\_rB.java

```
public class strat_rB extends AppCompatActivity implements View.OnClickListener{

    Button ret;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_strat_r_b);

        ret = (Button) findViewById(R.id.b_ret);
        ret.setOnClickListener(this);
    }

    @Override
    public void onClick(View v) {
        Intent otro = new Intent(this, strat_2.class);
        startActivity(otro);
    }
}
```

### 11.2.32. strat\_rC.java

```
public class strat_rC extends AppCompatActivity implements View.OnClickListener{

    Button ret;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_strat_r_c);

        ret = (Button) findViewById(R.id.b_ret);
        ret.setOnClickListener(this);
    }

    @Override
    public void onClick(View v) {
        Intent otro = new Intent(this, strat_2.class);
        startActivity(otro);
    }
}
```

### 11.2.33. strat\_rD.java

```
public class strat_rD extends AppCompatActivity implements View.OnClickListener{

    Button ret;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_strat_r_d);

        ret = (Button) findViewById(R.id.b_ret);
        ret.setOnClickListener(this);
    }

    @Override
    public void onClick(View v) {
        Intent otro = new Intent(this, strat_3.class);
        startActivity(otro);
    }
}
```

```
}  
}
```

### 11.2.34. strat\_rE.java

```
public class strat_rE extends AppCompatActivity implements View.OnClickListener{  
  
    Button ret;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_strat_r_e);  
  
        ret = (Button) findViewById(R.id.b_ret);  
        ret.setOnClickListener(this);  
    }  
  
    @Override  
    public void onClick(View v) {  
        Intent otro = new Intent(this, strat_3.class);  
        startActivity(otro);  
    }  
}
```

### 11.2.35. strat\_rF.java

```
public class strat_rF extends AppCompatActivity implements View.OnClickListener{  
  
    Button ret;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_strat_r_f);  
  
        ret = (Button) findViewById(R.id.b_ret);  
        ret.setOnClickListener(this);  
    }  
  
    @Override  
    public void onClick(View v) {  
        Intent otro = new Intent(this, strat_4.class);  
        startActivity(otro);  
    }  
}
```

### 11.2.36. strat\_rG.java

```
public class strat_rG extends AppCompatActivity implements View.OnClickListener{  
  
    Button ret;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_strat_r_g);  
  
        ret = (Button) findViewById(R.id.b_ret);  
        ret.setOnClickListener(this);  
    }  
  
    @Override
```

```

public void onClick(View v) {
    Intent otro = new Intent(this, strat_6.class);
    startActivity(otro);
}
}

```

### 11.2.37. strat\_rH.java

```

public class strat_rH extends AppCompatActivity implements View.OnClickListener{

    Button ret;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_strat_r_h);

        ret = (Button) findViewById(R.id.b_ret);
        ret.setOnClickListener(this);
    }

    @Override
    public void onClick(View v) {
        Intent otro = new Intent(this, strat_7.class);
        startActivity(otro);
    }
}

```

### 11.2.38. strat\_rI.java

```

public class strat_rI extends AppCompatActivity implements View.OnClickListener{

    Button ret;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_strat_r_i);

        ret = (Button) findViewById(R.id.b_ret);
        ret.setOnClickListener(this);
    }

    @Override
    public void onClick(View v) {
        Intent otro = new Intent(this, strat_8.class);
        startActivity(otro);
    }
}

```

### 11.2.39. strat\_rJ.java

```

public class strat_rJ extends AppCompatActivity implements View.OnClickListener{

    Button ret;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_strat_r_j);

        ret = (Button) findViewById(R.id.b_ret);
        ret.setOnClickListener(this);
    }
}

```



```

}

@Override
public void onClick(View v){
    Intent otro = new Intent(this, strat_9.class);
    startActivity(otro);
}
}

```

#### 11.2.40. strat\_rK.java

```

public class strat_rK extends AppCompatActivity implements View.OnClickListener{

    Button ret;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_strat_r_k);

        ret = (Button) findViewById(R.id.b_ret);
        ret.setOnClickListener(this);
    }

    @Override
    public void onClick(View v){
        Intent otro = new Intent(this, strat_11.class);
        startActivity(otro);
    }
}

```

#### 11.2.41. strat\_rL.java

```

public class strat_rL extends AppCompatActivity implements View.OnClickListener{

    Button ret;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_strat_r_l);

        ret = (Button) findViewById(R.id.b_ret);
        ret.setOnClickListener(this);
    }

    @Override
    public void onClick(View v){
        Intent otro = new Intent(this, strat_12.class);
        startActivity(otro);
    }
}

```

#### 11.2.42. strat\_rM.java

```

public class strat_rM extends AppCompatActivity implements View.OnClickListener{

    Button ret;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_strat_r_m);

        ret = (Button) findViewById(R.id.b_ret);
        ret.setOnClickListener(this);
    }
}

```

```

}

@Override
public void onClick(View v){
    Intent otro = new Intent(this, strat_13.class);
    startActivity(otro);
}
}

```

### 11.2.43. strat\_rN.java

```

public class strat_rN extends AppCompatActivity implements View.OnClickListener{

    Button ret;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_strat_r_n);

        ret = (Button) findViewById(R.id.b_ret);
        ret.setOnClickListener(this);
    }

    @Override
    public void onClick(View v){
        Intent otro = new Intent(this, strat_13.class);
        startActivity(otro);
    }
}

```

### 11.2.44. strat\_rO.java

```

public class strat_rO extends AppCompatActivity implements View.OnClickListener{

    Button ret;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_strat_r_o);

        ret = (Button) findViewById(R.id.b_ret);
        ret.setOnClickListener(this);
    }

    @Override
    public void onClick(View v){
        Intent otro = new Intent(this, strat_13.class);
        startActivity(otro);
    }
}

```

### 11.3. Código XML - CinematicaUAM\app\src\main\res\layout

Se generaron 54 archivos XML para los *layouts* de la parte gráfica de la aplicación, en este apartado no se colocan todos los archivos, dado que algunos de ellos repiten su estructura, pero se coloca de manera general el cuerpo indicando a cuales archivos pertenece.

#### 11.3.1. activity\_introduccion.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="oscarpereaptuam.cinematicauam.introduccion">

    <TextView
        android:id="@+id/tv_1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
        android:layout_alignParentTop="true"
        android:text="@string/intro_1"
        android:textAppearance="@android:style/TextAppearance.Widget.TextView" />

    <TextView
        android:id="@+id/tv_2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/intro_2"
        android:textAppearance="@android:style/TextAppearance.Widget.TextView"
        android:layout_marginTop="26dp"
        android:layout_below="@+id/tv_1"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true" />

    <Button
        android:text="@string/b_menu_estra"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:height="80dp"
        android:width="120dp"
        android:id="@+id/b_istrat"
        android:background="@color/colorAccent"
        tools:textStyle="bold"
        android:textStyle="bold"
        android:layout_below="@+id/tv_2"
        android:layout_marginTop="35dp"
        android:layout_alignParentRight="true"
        android:layout_alignParentEnd="true" />

    <Button
        android:text="@string/b_menu_casos"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
```

```

    android:height="80dp"
    android:width="120dp"
    android:id="@+id/b_icasos"
    android:background="@color/colorAccent"
    tools:textStyle="bold"
    android:textStyle="bold"
    android:layout_below="@+id/tv_2"
    android:layout_marginTop="35dp"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true"/>

```

```
</RelativeLayout>
```

### 11.3.2. activity\_menu\_p.xml

```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/activity_menu_p"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="oscarpereaptuam.cinematicauam.Menu_p">

    <Button
        android:text="@string/b_menu_estra"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/b_strat"
        android:background="@color/colorAccent"
        android:textAppearance="@style/TextAppearance.AppCompat.Headline"
        android:textStyle="bold"
        android:height="80dp"
        android:width="800dp"
        android:layout_marginBottom="72dp"
        android:layout_alignParentBottom="true"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true" />

    <Button
        android:text="@string/b_menu_casos"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/b_casos"
        android:background="@color/colorAccent"
        android:textAppearance="@style/TextAppearance.AppCompat.Headline"
        android:textStyle="bold"
        android:ellipsize="none"
        android:height="80dp"
        android:width="800dp"
        android:layout_marginTop="68dp"
        android:layout_alignParentTop="true"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true" />

</RelativeLayout>

```

### 11.3.3. activity\_instr\_\*.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/activity_instr_strat"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="oscarpereaptuam.cinematicauam.instr_strat">

    <TextView
        android:id="@+id/text_inst_e"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:text="@string/strat_instr"
        android:textAppearance="@android:style/TextAppearance.Widget.TextView"
        android:layout_alignParentRight="true"
        android:layout_alignParentEnd="true" />

    <Button
        android:text="@string/cont"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:height="65dp"
        android:width="120dp"
        android:id="@+id/b_cont_estrat"
        android:background="@color/colorAccent"
        tools:textStyle="bold"
        android:textStyle="bold"
        android:layout_below="@+id/text_inst_e"
        android:layout_centerHorizontal="true" />

</RelativeLayout>
```

### 11.3.4. actividad\_entrada\_casos.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/actividad_entrada_casos"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="oscarpereaptuam.cinematicauam.entrada_casos">

    <EditText
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/en_X0"
        android:height="60dp"
        android:width="180dp"
        android:layout_marginTop="10dp"
        android:hint="Posicion Inicial X0"
        android:text=""
        android:inputType="numberSigned|numberDecimal"
        android:maxLines="1"
        android:layout_alignParentTop="true" />
```

```

        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true" />

<TextView
    android:text="@string/uni_p"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/textv_x0"
    android:height="50dp"
    android:width="150dp"
    android:textSize="24sp"
    android:layout_margin="2dp"
    android:layout_alignTop="@+id/en_X0"
    android:layout_toRightOf="@+id/en_X0"
    android:layout_toEndOf="@+id/en_X0" />

<EditText
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/en_X"
    android:height="60dp"
    android:width="180dp"
    android:hint="Posicion X"
    android:text=""
    android:inputType="numberSigned|numberDecimal"
    android:maxLines="1"
    android:layout_below="@+id/textv_x0"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true" />

<TextView
    android:text="@string/uni_p"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:height="50dp"
    android:width="150dp"
    android:textSize="24sp"
    android:layout_marginTop="10dp"
    android:id="@+id/textv_x"
    android:layout_below="@+id/textv_x0"
    android:layout_alignLeft="@+id/textv_x0"
    android:layout_alignStart="@+id/textv_x0" />

<EditText
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/en_V0"
    android:height="60dp"
    android:width="180dp"
    android:hint="Velocidad Inicial V0"
    android:text=""
    android:inputType="numberSigned|numberDecimal"
    android:maxLines="1"
    android:layout_below="@+id/textv_x"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true" />

<TextView
    android:text="@string/uni_v"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:height="50dp"
    android:width="150dp"
    android:textSize="24sp"
    android:layout_marginTop="10dp"
    android:id="@+id/textv_v0"

```

```

        android:layout_alignTop="@+id/en_V0"
        android:layout_alignLeft="@+id/textv_x"
        android:layout_alignStart="@+id/textv_x" />

<EditText
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/en_v"
    android:height="60dp"
    android:width="180dp"
    android:hint="Velocidad v"
    android:text=""
    android:inputType="numberSigned|numberDecimal"
    android:maxLines="1"
    android:layout_below="@+id/textv_v0"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true" />

<TextView
    android:text="@string/uni_v"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:height="50dp"
    android:width="150dp"
    android:textSize="24sp"
    android:layout_marginTop="10dp"
    android:id="@+id/textv_v"
    android:layout_alignTop="@+id/en_v"
    android:layout_alignLeft="@+id/textv_v0"
    android:layout_alignStart="@+id/textv_v0" />

<EditText
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/en_a"
    android:height="60dp"
    android:width="180dp"
    android:hint="Aceleracion a"
    android:text=""
    android:inputType="numberSigned|numberDecimal"
    android:maxLines="1"
    android:layout_below="@+id/textv_v"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true" />

<TextView
    android:text="@string/uni_a"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:height="50dp"
    android:width="150dp"
    android:textSize="24sp"
    android:layout_marginTop="10dp"
    android:id="@+id/textv_a"
    android:layout_alignTop="@+id/en_a"
    android:layout_alignLeft="@+id/textv_v"
    android:layout_alignStart="@+id/textv_v" />

<EditText
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/en_t"
    android:height="60dp"
    android:width="180dp"
    android:hint="Tiempo t"
    android:text=""

```

```

        android:inputType="numberSigned|numberDecimal"
        android:maxLines="1"
        android:layout_below="@+id/textv_a"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true" />

```

```

<TextView
    android:text="@string/unit"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:height="50dp"
    android:width="150dp"
    android:textSize="24sp"
    android:layout_marginTop="10dp"
    android:id="@+id/textv_t"
    android:layout_alignTop="@+id/en_t"
    android:layout_alignLeft="@+id/textv_a"
    android:layout_alignStart="@+id/textv_a"
    android:layout_alignRight="@+id/textv_a"
    android:layout_alignEnd="@+id/textv_a" />

```

```

<Button
    android:text="@string/resolver_c"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/b_resolver"
    android:background="@color/colorAccent"
    android:height="60dp"
    android:width="180dp"
    android:layout_below="@+id/textv_t"
    android:layout_centerHorizontal="true" />

```

```
</RelativeLayout>
```

### 11.3.5. activity\_c\_variables\_\*.xml

```

<?xml version="1.0" encoding="utf-8"?>
<android.support.design.widget.CoordinatorLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:fitsSystemWindows="true"
    tools:context="oscarpereaptuam.cinematicauam.c_variables_at">

    <android.support.design.widget.AppBarLayout
        android:id="@+id/app_bar"
        android:layout_width="match_parent"
        android:layout_height="@dimen/app_bar_height"
        android:fitsSystemWindows="true"
        android:theme="@style/AppTheme.AppBarOverlay">

        <android.support.design.widget.CollapsingToolbarLayout
            android:id="@+id/toolbar_layout"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:fitsSystemWindows="true"
            app:contentScrim="?attr/colorPrimary"
            app:layout_scrollFlags="scroll|exitUntilCollapsed">

            <android.support.v7.widget.Toolbar
                android:id="@+id/toolbar"
                android:layout_width="match_parent"
                android:layout_height="?attr/actionBarSize"
                app:layout_collapseMode="pin"

```



```

        app:popupTheme="@style/AppTheme.PopupOverlay" />

    </android.support.design.widget.CollapsingToolbarLayout>
</android.support.design.widget.AppBarLayout>

<include layout="@layout/content_c_variables_at" />
</android.support.design.widget.CoordinatorLayout>

```

### 11.3.6. content\_c\_variables\_\*.xml

```

<?xml version="1.0" encoding="utf-8"?>
<android.support.v4.widget.NestedScrollView
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
app:layout_behavior="@string/appbar_scrolling_view_behavior"
tools:context="oscarpereaptuam.cinematicauam.c_variables_vt"
tools:showIn="@layout/activity_c_variables_vt">

    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_margin="20dp">

        <TextView
            android:text="@string/problema"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_alignParentTop="true"
            android:layout_alignParentLeft="true"
            android:layout_alignParentStart="true"
            android:id="@+id/textv_1"
            android:textAppearance="@style/TextAppearance.AppCompat.Title" />

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text=""
            android:id="@+id/textv_s1"
            android:textAppearance="@style/TextAppearance.AppCompat.Title"
            android:layout_marginTop="42dp"
            android:layout_below="@+id/textv_1"
            android:layout_alignParentLeft="true"
            android:layout_alignParentStart="true" />

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@string/ecuaciones"
            android:textAppearance="@style/TextAppearance.AppCompat.Title"
            android:layout_marginTop="42dp"
            android:id="@+id/textv_2"
            android:layout_below="@+id/textv_s1"
            android:layout_alignParentLeft="true"
            android:layout_alignParentStart="true" />

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@string/pasol"
            android:textAppearance="@style/TextAppearance.AppCompat.Title"
            android:layout_marginTop="42dp"
            android:layout_marginStart="36dp"

```

```

        android:layout_marginLeft="36dp"
        android:id="@+id/textv_3"
        android:layout_below="@+id/textv_2"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true" />

<TextView
    android:text=""
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/textv_s2"
    android:textAppearance="@style/TextAppearance.AppCompat.Title"
    android:layout_marginTop="42dp"
    android:layout_below="@+id/textv_3"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true" />

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/paso2"
    android:textAppearance="@style/TextAppearance.AppCompat.Title"
    android:layout_marginTop="42dp"
    android:layout_marginStart="36dp"
    android:layout_marginLeft="36dp"
    android:id="@+id/textv_4"
    android:layout_below="@+id/textv_s2"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true" />

<TextView
    android:text=""
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/textv_s3"
    android:textAppearance="@style/TextAppearance.AppCompat.Title"
    android:layout_marginTop="42dp"
    android:layout_below="@+id/textv_4"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true" />

<TextView
    android:text="@string/sol_plu"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true"
    android:textAppearance="@style/TextAppearance.AppCompat.Title"
    android:layout_marginLeft="36dp"
    android:layout_marginStart="36dp"
    android:layout_below="@+id/textv_s3"
    android:layout_marginTop="42dp"
    android:id="@+id/textv_sol_t" />

<TextView
    android:text=""
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/textv_sol_v"
    android:textAppearance="@style/TextAppearance.AppCompat.Title"
    android:layout_marginTop="44dp"
    android:layout_below="@+id/textv_sol_t"
    android:layout_centerHorizontal="true" />

<Button
    android:text="@string/sal_menu"

```

```

        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:height="80dp"
        android:width="120dp"
        android:id="@+id/b_ret_menu"
        android:background="@color/colorAccent"
        tools:textStyle="bold"
        android:textStyle="bold"
        android:layout_below="@+id/textv_sol_v"
        android:layout_marginTop="42dp"
        android:layout_alignParentRight="true"
        android:layout_alignParentEnd="true"
        android:layout_marginBottom="100dp"/>

<Button
    android:text="@string/ent_casos"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:height="80dp"
    android:width="120dp"
    android:id="@+id/b_ret_entr"
    android:background="@color/colorAccent"
    tools:textStyle="bold"
    android:textStyle="bold"
    android:layout_below="@+id/textv_sol_v"
    android:layout_marginTop="42dp"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true"
    android:layout_marginBottom="100dp"/>

</RelativeLayout>

</android.support.v4.widget.NestedScrollView>

```

### 11.3.7. activity\_strat\_\*.xml

```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/activity_strat_8"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="oscarpereaptuam.cinematicauam.strat_8">

    <TextView
        android:id="@+id/tv_1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_margin="8dp"
        android:text="@string/strat_8"
        android:textAppearance="@android:style/TextAppearance.Widget.TextView"
        android:layout_alignParentTop="true"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true" />

    <TextView
        android:id="@+id/tv_opc_1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignLeft="@+id/tv_1"
        android:layout_alignStart="@+id/tv_1"
        android:layout_alignTop="@+id/b8_opc_1"

```

```

        android:layout_marginTop="13dp"
        android:text="@string/strat_8_opc_1"
        android:textAppearance="@style/TextAppearance.AppCompat.Widget.Button" />

<Button
    android:id="@+id/b8_opc_1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/sel_opc"
    android:height="10dp"
    android:width="10dp"
    android:background="@color/colorAccent"
    android:layout_marginTop="15dp"
    android:layout_below="@+id/tv_1"
    android:layout_alignRight="@+id/tv_1"
    android:layout_alignEnd="@+id/tv_1" />

<TextView
    android:id="@+id/tv_opc_2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/strat_8_opc_2"
    android:textAppearance="@style/TextAppearance.AppCompat.Widget.Button"
    android:layout_alignBaseline="@+id/b8_opc_2"
    android:layout_alignBottom="@+id/b8_opc_2"
    android:layout_alignLeft="@+id/tv_opc_3"
    android:layout_alignStart="@+id/tv_opc_3" />

<Button
    android:id="@+id/b8_opc_2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/sel_opc"
    android:height="10dp"
    android:width="10dp"
    android:background="@color/colorAccent"
    android:layout_marginTop="18dp"
    android:layout_below="@+id/b8_opc_1"
    android:layout_alignLeft="@+id/b8_opc_1"
    android:layout_alignStart="@+id/b8_opc_1" />

<TextView
    android:id="@+id/tv_opc_3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/strat_8_opc_3"
    android:textAppearance="@style/TextAppearance.AppCompat.Widget.Button"
    android:layout_alignBaseline="@+id/b8_opc_3"
    android:layout_alignBottom="@+id/b8_opc_3"
    android:layout_alignLeft="@+id/tv_opc_1"
    android:layout_alignStart="@+id/tv_opc_1" />

<Button
    android:id="@+id/b8_opc_3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/sel_opc"
    android:height="10dp"
    android:width="10dp"
    android:background="@color/colorAccent"
    android:layout_marginTop="18dp"
    android:layout_below="@+id/b8_opc_2"
    android:layout_alignLeft="@+id/b8_opc_2"
    android:layout_alignStart="@+id/b8_opc_2" />

<TextView

```

```

    android:id="@+id/tv_opc_4"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/strat_8_opc_4"
    android:textAppearance="@style/TextAppearance.AppCompat.Widget.Button"
    android:layout_alignBaseline="@+id/b8_opc_4"
    android:layout_alignBottom="@+id/b8_opc_4"
    android:layout_alignLeft="@+id/tv_opc_1"
    android:layout_alignStart="@+id/tv_opc_1" />

```

```

<Button
    android:id="@+id/b8_opc_4"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/sel_opc"
    android:height="10dp"
    android:width="10dp"
    android:background="@color/colorAccent"
    android:layout_marginTop="18dp"
    android:layout_below="@+id/b8_opc_3"
    android:layout_alignLeft="@+id/b8_opc_3"
    android:layout_alignStart="@+id/b8_opc_3" />

```

```
</RelativeLayout>
```

### 11.3.8. activity\_strat\_r\_\*.xml

```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/activity_strat_r_i"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="oscarpereaptuam.cinematicauam.strat_rI">

```

```

    <TextView
        android:id="@+id/tv_1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/retro_I"
        android:textAppearance="@style/TextAppearance.AppCompat.Title"
        android:layout_alignParentTop="true"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true" />

```

```

    <Button
        android:id="@+id/b_ret"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/reg"
        android:height="20dp"
        android:width="50dp"
        android:background="@color/colorAccent"
        android:layout_centerVertical="true"
        android:layout_centerHorizontal="true" />

```

```
</RelativeLayout>
```

## 11.4. strings.xml - CinematicaUAM\app\src\main\res\values

```
<resources>
  <string name="app_name">Cinematica UAM</string>
  <string name="action_settings">Configuración</string>
  <string name="intro_1">BIENVENIDO A LA APLICACION DE ESTRATEGIAS PARA RESOLVER
  PROBLEMAS DE CINEMATICA \n\n Resolver un problema no es cuestión de inspiración, se
  requiere trabajo y constancia en el estudio.</string>
  <string name="intro_2">Esta aplicación consta de dos partes, en ESTRATEGIAS se te
  propondrá un problema y se te ofrecerán opciones para ir resolviéndolo, en RESOLVER
  PROBLEMA se resolverán ecuaciones de acuerdo a datos que tu proporciones mostrando paso
  a paso como obtener la solución.</string>
  <string name="b_menu_estra">Estrategias</string>
  <string name="b_menu_casos">Resolver Problema</string>
  <string name="cont">Continuar</string>
  <string name="resolver_c">Resolver</string>
  <string name="reg">Regresar</string>
  <string name="otro_p_estrategias">Continuar con Otro Problema</string>
  <string name="casos_instr">Aquí se resolverán problemas en los que se han
  identificado incógnitas y datos, con los datos que introduzcas se resolverá un problema
  como ejemplo y te mostrará paso a paso como obtener la solución.\n\nPara resolver un
  sistema se deben especificar 4 valores, dejando 2 campos de incógnitas vacías, la
  posición inicial (X0) SIEMPRE es un dato de entrada por lo que siempre debe indicarse
  un valor (Puede ser 0).</string>
  <string name="strat_instr">Una forma de modelar la resolución de un problema es: \n
  1.- Comprender el enunciado del problema \n 2.- Traducir variables físicas a
  matemáticas \n 3.- Plantear ecuaciones \n 4.- Resolver las ecuaciones \n 5.- Verificar
  el resultado.\nRevisaremos las estrategias de cinemática en una dimensión (1-D) \n
  Recordemos que SOLO hay 2 ecuaciones básicas de movimiento, las cuales usaremos: \n
  X(t)=X0+V0t+(at^2)/2 -----> (ec1) \n V(t)=V0+at -----> (ec2) \n
  Cualquier otra se deduce a partir de éstas. A continuación resolverás un problema.</string>
  <string name="uni_p">[m]</string>
  <string name="uni_v">[m/s]</string>
  <string name="uni_a">[m/(s^2)]</string>
  <string name="uni_t">[s]</string>
  <string name="problema">Con los datos introducidos se puede enunciar el siguiente
  ejemplo de problema: </string>
  <string name="ecuaciones">Partiendo de las ecuaciones básicas de movimiento
  uniformemente acelerado: \n\n X(t)=X0+V0t+(at^2)/2 -----> (ec1) \n V(t)=V0+at ----->
  (ec2)</string>
  <string name="pasos">Paso 1:</string>
  <string name="pasos2">Paso 2:</string>
  <string name="sol_sin">Solución:</string>
  <string name="sol_plu">Solución(es):</string>
  <string name="large_text">Bio-Osc</string>
  <string name="title_activity_c_variables_vt">Resolver Velocidad y Tiempo</string>
  <string name="title_activity_c_variables_xa">Resolver Posición y
  Aceleración</string>
  <string name="title_activity_c_variables_va">Resolver Velocidad y
  Aceleración</string>
  <string name="title_activity_c_variables_xv">Resolver Posición y Velocidad</string>
  <string name="title_activity_c_variables_at">Resolver Aceleración y Tiempo</string>
  <string name="title_activity_c_variables_xt">Resolver Posición y Tiempo</string>
  <string name="title_activity_c_variables_v0v">Resolver Velocidad Inicial y
  Velocidad</string>
  <string name="title_activity_c_variables_xv0">Resolver Posición y Velocidad
  Inicial</string>
  <string name="title_activity_c_variables_v0a">Resolver Velocidad Inicial y
  Aceleración</string>
  <string name="title_activity_c_variables_v0t">Resolver Velocidad Inicial y
  Tiempo</string>
  <string name="sal_menu">Salir al Menú</string>
  <string name="ent_casos">Resolver Otro Problema</string>
  <string name="cont_estr">Continuar</string>
  <string name="sel_opc">[X]</string>

```

<string name="strat\_1">Un automóvil lleva una velocidad de 15[m/s] y frena con  $a=2.5[m/s^2]$ . ¿En qué tiempo se detendrá? ¿En qué posición se encuentra? En primer lugar debes aprender a traducir frases a ecuaciones o condiciones.\n\n Por ejemplo la frase: \n "Lleva una velocidad de 15[m/s]" significa: </string>

<string name="strat\_1\_opc\_1">Su velocidad final es \n 15 [m/s]</string>

<string name="strat\_1\_opc\_2">Su velocidad inicial es \n 15 [m/s]</string>

<string name="retro\_A">Considera tu respuesta, el problema indica que se lleva una velocidad antes de que el automóvil frene, velocidad inicial es lo correcto en este caso.</string>

<string name="strat\_2">Efectivamente, corresponde a su velocidad inicial \n Esto puede expresarse como:</string>

<string name="strat\_2\_opc\_1"> $a_0$  (aceleración inicial)</string>

<string name="strat\_2\_opc\_2"> $V_0$  (velocidad inicial)</string>

<string name="strat\_2\_opc\_3"> $t_0$  (tiempo inicial)</string>

<string name="retro\_B">La aceleración inicial no corresponde con velocidad inicial que es lo que tratamos de identificar.</string>

<string name="retro\_C">El tiempo inicial no corresponde con velocidad inicial que es lo que tratamos de identificar.</string>

<string name="strat\_3">Claro es  $V_0$  (velocidad inicial), ahora la frase \n "¿En qué tiempo se detendrá?" \n significa:</string>

<string name="strat\_3\_opc\_1">En que tiempo  $a=0$ </string>

<string name="strat\_3\_opc\_2">En que tiempo  $V=0$ </string>

<string name="strat\_3\_opc\_3">En que tiempo  $t=0$ </string>

<string name="retro\_D">Buscamos que el auto se detenga, por lo tanto que su velocidad sea 0, no  $a=0$ , si su aceleración es 0 significa que su velocidad es constante.</string>

<string name="retro\_E">Buscamos que el auto se detenga, por lo tanto que su velocidad sea 0, no  $t=0$ </string>

<string name="strat\_4">Es correcto, entonces si buscamos que  $V=0$  podemos ocupar (ec2) de la que conocemos todo excepto t: \n  $V=V_0+a*t$  \n  $t=(V-V_0)/a$  \n los datos del problema nos dicen que  $V=0[m/s]$ ,  $V_0=15[m/s]$  y que:</string>

<string name="strat\_4\_opc\_1"> $a = 2.5[m/s^2]$  (positiva)</string>

<string name="strat\_4\_opc\_2"> $a = -2.5[m/s^2]$  (negativa)</string>

<string name="retro\_F">No puede ser positiva pues el problema menciona que el auto "frena", es decir desacelera y por lo tanto el valor de la aceleración debe ser negativo.</string>

<string name="strat\_5">Correcto, sustituyendo tenemos que: \n  $t=(-15/-2.5)$  \n  $t = 6[s]$  \n Con este tiempo podemos encontrar en que posición se encuentra.\n Sustituyendo en (ec1) tenemos que \n  $X=X_0+V_0t+(at^2)/2$  \n  $X=0+15*6+((-2.5)*36)/2$  \n  $X=90-45=45[m]$  \n esta no es la única forma de resolver el problema, intenta hacerlo de otra manera.</string>

<string name="strat\_6">Una persona lanza una moneda hacia arriba con una velocidad de 12[m/s] ¿En qué tiempo estará de regreso la moneda a 1.2[m] de su mano? \n Todos los problemas requieren de un sistema de referencia para su comprensión. Vamos a tomar el origen del eje Y en la mano de la persona y positivo hacia arriba.\n En este caso la gravedad se debe tomar como:</string>

<string name="strat\_6\_opc\_1"> $g = 9.81[m/s^2]$  \n(positiva hacia abajo)</string>

<string name="strat\_6\_opc\_2"> $g = -9.81[m/s^2]$  \n(negativa hacia abajo)</string>

<string name="retro\_G">Incorrecto, si el eje Y es positivo hacia arriba la aceleración debida a la gravedad, al estar dirigida hacia abajo es negativa.</string>

<string name="strat\_7">Si, una vez acordado el sistema de referencia sabemos que la moneda sube y pasa de nuevo por su posición original. El problema nos presenta datos e incógnitas en cuanto a posición y tiempo, por lo tanto la ecuación que se debe ocupar en este caso es:</string>

<string name="strat\_7\_opc\_1"> $Y(t)=Y_0+V_0t+(gt^2)/2$  \n (ec1)</string>

<string name="strat\_7\_opc\_2"> $V(t)=V_0+gt$  \n (ec2)</string>

<string name="retro\_H">Esta ecuación no contiene la variable posición por lo que es mejor utilizar la otra (ec1).</string>

<string name="strat\_8">Efectivamente (ec1) es la que debemos utilizar. ¿Si sustituimos los datos cual es la ecuación resultante?</string>

<string name="strat\_8\_opc\_1"> $Y = -15t-(9.8*t^2)/2$ </string>

<string name="strat\_8\_opc\_2"> $Y = 15t+(9.8*t^2)/2$ </string>

<string name="strat\_8\_opc\_3"> $Y = 15t-(9.8*t^2)/2$ </string>

<string name="strat\_8\_opc\_4"> $Y = -15t+(9.8*t^2)/2$ </string>

<string name="retro\_I">Dado que tenemos \n  $V_0 = 15[m/s]$  \n  $g = -9.8[m/s^2]$  \n esta no es la respuesta correcta. Recuerda como se determinaron los ejes de referencia del



```

problema.</string>
  <string name="strat_9">Es correcto, es decir se tiene  $0=15t-(9.8*t^2)/2$  \n ya
que la posición final es igual a la inicial. Resolviendo esta ecuación se tiene que \n
 $t_1=0[s]$  \n  $t_2=3.06[s]$  \n Cual resultado es el mas adecuado para solucionar el
problema?</string>
  <string name="strat_9_opc_1">t=0[s]</string>
  <string name="strat_9_opc_2">t=3.06[s]</string>
  <string name="retro_J">A pesar de ser una solución correcta para  $Y=0$ ,  $t=0$  indica
que la moneda aun no sale de la mano del lanzador, el problema nos pide el tiempo en el
que la moneda regresa a la mano del lanzador.</string>
  <string name="strat_10">Así es  $t=3.06[s]$  es el tiempo que le toma a la moneda
llegar de vuelta a la mano del lanzador.</string>
  <string name="strat_11">Un ciclista alcanza una velocidad máxima de  $20[m/s]$  y la
aceleración más grande que puede alcanzar es de  $2.30[m/s^2]$ . Si el ciclista comienza a
moverse desde el reposo y pedalea al máximo durante  $22[s]$  ¿Que distancia recorrió en
ese tiempo? \n\n ¿Si sustituimos la ecuación  $X=X_0+V_0t+(at^2)/2$  \n  $X=0+0+(2.3*22^2)/2$ 
\n es correcto?</string>
  <string name="strat_11_opc_1">Si es correcto</string>
  <string name="strat_11_opc_2">No es correcto</string>
  <string name="retro_K">No! En este problema no se puede sustituir directamente de
esta forma porque la aceleración del problema no se aplica durante los 22 segundos, es
decir el problema no nos da una aceleración constante sino una aceleración en un
intervalo de tiempo.</string>
  <string name="strat_12">No podemos utilizar esta ecuación por que no actúa durante
los 22 segundos del recorrido, una manera de comprobarlo es usando  $V=V_0+at$  ---->
 $20=0+2.3t$  \n  $t=20/2.3=8.7[s]$  \n Esto significa que la aceleración actúa 8.7 segundos,
después aunque se intente acelerar no se podrá hacerlo. Si utilizamos la ecuación  $X=X_0+V_0t+(at^2)/2$ 
\n  $X=0+0+(2.3*8.7^2)/2$  \n  $X=87.04[m]$ \n ¿Hemos terminado el
problema?</string>
  <string name="strat_12_opc_1">No</string>
  <string name="strat_12_opc_2">Si</string>
  <string name="retro_L">No aun no, el problema nos pide obtener la distancia después
de  $22[s]$ , ya hemos obtenido la distancia recorrida en  $8.7[s]$  por lo que falta calcular
la distancia del resto del tiempo en este caso  $11.3[s]$ .</string>
  <string name="strat_13">No, claro que no, aún nos falta determinar la distancia
recorrida en los restantes  $11.3[s]$ . ¿Cuál ecuación es la adecuada para esto?</string>
  <string name="strat_13_opc_1"> $X=0+20*22+(2.3*22^2)/2$ </string>
  <string name="strat_13_opc_2"> $X=87.04+20*11.3$ </string>
  <string name="strat_13_opc_3"> $X=0+20*11.3$ </string>
  <string name="strat_13_opc_4"> $X=0+20*22+(2.3*11.3^2)/2$ </string>
  <string name="retro_M">No, porque la aceleración no actúa el mismo tiempo que la
velocidad de  $20[m/s]$ .</string>
  <string name="retro_N">No, porque en la formula no se incluye la distancia
recorrida cuando si actúa la aceleración, es decir  $87.04[m]$ .</string>
  <string name="retro_O">No, porque la velocidad no la lleva durante los  $22[s]$  del
recorrido, así como tampoco lleva la aceleración durante los  $11.3[s]$ 
restantes.</string>
  <string name="strat_14">Exacto la ecuación  $X=87.04+20*11.3$  \n  $X=313.04$  \n
incluye la distancia recorrida acelerando durante  $8.7[s]$  e incluye la distancia
recorrida con velocidad constante durante los otros  $11.3[s]$  del recorrido.</string>
</resources>

```



## **12. Entregables:**

La siguiente documentación fue generada durante la realización del proyecto, con el motivo de cumplir con los lineamientos establecidos para la entrega del mismo:

- Diseños preliminares (Imágenes en formato JPG)
- Prototipo en POP (<https://marvelapp.com/38b0e73>)
- Modulo Casos (Código Fuente Java)
- Modulo Estrategias (Código Fuente Java)
- Interfaz (Documentos XML para Java)
- Aplicación formato APK
- El presente Reporte