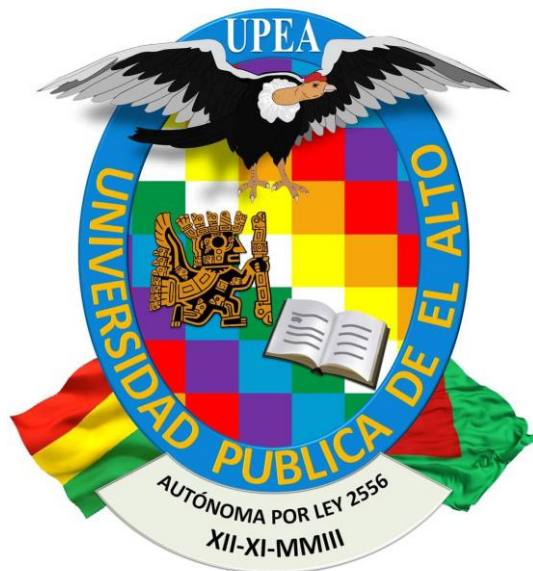


# UNIVERSIDAD PÚBLICA DE EL ALTO

## CARRERA INGENIERÍA DE SISTEMAS



### PROYECTO DE GRADO

**“TUTOR INTELIGENTE MÓVIL PARA LA ENSEÑANZA DE LA ESTÁTICA Y DINÁMICA EN ESTUDIANTES DE QUINTO DE SECUNDARIA”**

**CASO: UNIDAD EDUCATIVA DIONICIO MORALES**

**Para Optar al Título de Licenciatura en Ingeniería de Sistemas  
MENCIÓN: INFORMÁTICA Y COMUNICACIONES**

**Postulante:** Justiniano Mendoza Apaza

**Tutor Metodológico:** Ing. Marisol Arguedas Balladares

**Tutor Especialista:** Lic. Katya Maricela Pérez Martínez

**Tutor Revisor:** Lic. Adrian Eusebio Quisbert Vilela

**EL ALTO – BOLIVIA**

**2020**

## **DEDICATORIA**

A dios, por haberme dado la vida, salud, iluminación y fortaleza en los momentos más difíciles de mi vida.

A mis padres Marcelo y Susana quienes me dieron todo su amor, cariño y comprensión, y me brindaron su apoyo con sus consejos y recomendaciones a lo largo de mi vida.

A mis hermanos Blanca Eva, Heriberto, Mery Gaby, Sonia Carola, por el apoyo incondicional que me brindaron en mi formación profesional.

A mis tutores y amigos por permitirme aprender más de la vida a su lado, por su colaboración y apoyo en el desarrollo de este trabajo.

## **AGRADECIMIENTOS**

Mis sinceros agradecimientos a todas las personas que hicieron posible culminar el presente trabajo de manera satisfactoria.

A mis padres, mis hermanos, y familiares por el apoyo incondicional en mis estudios.

A mi Tutor Metodológico Ing. Marisol Arguedas Balladares, por su paciencia, sus consejos, observaciones y sus conocimientos me guio en el desarrollo del presente proyecto.

A mi Tutor Especialista Lic. Katya Maricela Pérez Martínez, por la disponibilidad de tiempo, por su apoyo incondicional y consejos por lo que pude seguir adelante con el presente proyecto.

A mi Tutor Revisor Lic. Adrian Eusebio Quisbert Vilela, por su colaboración, confianza, apoyo incondicional, y las observaciones brindadas en la revisión del proyecto.

Agradecimiento a mis docentes de la Carrera Ingeniería de Sistemas, por los conocimientos brindados durante el tiempo de mi formación.

A la Unidad Educativa “Dionicio Morales” por permitirme desarrollar el presente Proyecto de Grado.

Y finalmente a mis amigos y compañeros con las cuales compartimos muchos momentos en las aulas en los años de formación.

Muchas Gracias...

## RESUMEN

Actualmente el avance tecnológico nos permite acceder de forma libre a diferentes contenidos como la información, la comunicación y el entretenimiento. Una de estas tecnologías son los dispositivos móviles, para los cuales se han desarrollado herramientas informáticas especializadas en distintas áreas convirtiéndose en una alternativa para el proceso de construcción de conocimientos en la educación, con la cual el estudiante puede reforzar su aprendizaje o ampliar su conocimiento, ya que se tiene una facilidad de acceso.

En la Unidad Educativa “Dionicio Morales” los estudiantes de quinto de Secundaria presentan ciertas dificultades en la asimilación de contenidos, así como en la resolución de problemas en los temas de Estática y Dinámica, debido a muchas razones como ser: las limitaciones de tiempo en el avance curricular, la falta de incorporación nuevas tecnologías y medios didácticos para la enseñanza. Por esta necesidad innovar y ayudar al educando en su proceso de formación se plantea desarrollar de un Tutor Inteligente Móvil capaz de colaborar en el aprendizaje.

Para el desarrollo del prototipo de Tutor Inteligente Móvil se utilizó la Metodología de Ingeniería de Software Educativo (MEISE) combinándolas con las fases de la Metodología Mobile-D ya que sus características son orientadas para el desarrollo de software para dispositivos móviles.

Entre las herramientas, se utilizó Android Studio para su desarrollo de la aplicación móvil. Y como gestor de Base de Datos SQLite que realiza el almacenamiento de la información. Una vez finalizado el proyecto, se realizó la evaluación de la calidad de Software con la norma Internacional ISO/IEC 9126, de acuerdo con la métricas o indicadores que presenta el modelo de calidad. En cuanto a la seguridad de la información se recurrió a la ISO 17799. Y finalmente para la estimación de costos se realizó con el modelo COCOMO II basado en Kilo líneas de código.

## ÍNDICE

	Página
<b>1. MARCO PRELIMINAR</b> .....	1
1.1. Introducción .....	1
1.2. Antecedentes de la Investigación .....	2
1.3. Planteamiento del Problema .....	5
1.3.1. Problema Principal .....	5
1.3.2. Problemas Secundarios .....	6
1.4. Objetivos.....	6
1.4.1. Objetivo General .....	6
1.4.2. Objetivos Específicos .....	6
1.5. Justificación .....	7
1.5.1. Justificación Técnica .....	7
1.5.2. Justificación Económica.....	7
1.5.3. Justificación Social .....	7
1.6. Metodología.....	8
1.7. Herramientas .....	8
1.7.1. Android Studio .....	8
1.7.2. SQLite .....	8
1.7.3. DB Browser for SQLite.....	9
1.7.4. Photoshop y Xara – Visio.....	9
1.8. Límites y Alcances.....	9
1.8.1. Límites .....	9
1.8.2. Alcances .....	9
1.9. Aportes .....	10

<b>CAPITULO II</b> .....	11
<b>2. MARCO TEÓRICO</b> .....	11
<b>2.1. Introducción</b> .....	11
<b>2.1.1. Inteligencia Artificial</b> .....	11
<b>2.1.2. Inteligencia Artificial En La Educación</b> .....	12
<b>2.1.3. Definición de Tutor Inteligente.</b> .....	13
<b>2.1.4. Proceso de Enseñanza – Aprendizaje</b> .....	14
<b>2.1.4.1. Enseñanza</b> .....	15
<b>2.1.4.2. Estrategias de Enseñanza</b> .....	16
<b>2.1.4.3. Aprendizaje</b> .....	17
<b>2.1.4.4. Estilos de Aprendizaje</b> .....	17
<b>2.1.4.5. Estrategias de Aprendizaje</b> .....	18
<b>2.1.5. Dinámica de una partícula</b> .....	19
<b>2.1.5.1. Fuerza <math>F</math></b> .....	19
<b>2.1.5.2. Masa (<math>m</math>)</b> .....	19
<b>2.1.5.3. Peso <math>W</math></b> .....	20
<b>2.1.5.4. Leyes de Newton</b> .....	20
<b>2.1.6. Estática</b> .....	21
<b>2.1.6.1. Equilibrio</b> .....	21
<b>2.1.6.2. Primera condición de equilibrio</b> .....	22
<b>2.1.6.3. Segunda Condición de Equilibrio</b> .....	22
<b>2.2. Metodología</b> .....	22
<b>2.2.1. Metodología Mobile-D</b> .....	23
<b>2.2.1.1. Fase de Exploración</b> .....	24
<b>2.2.1.2. Fase de Inicialización</b> .....	24

2.2.1.3.	Fase de Producción .....	25
2.2.1.4.	Fase de Estabilización .....	25
2.2.1.5.	Fase de pruebas y reparación del sistema .....	25
2.2.2.	Metodología de Ingeniería de Software Educativo .....	25
2.2.2.1.	Fase Análisis .....	26
2.2.2.2.	Fase Diseño .....	28
2.2.2.3.	Desarrollo .....	28
2.2.2.4.	Fase Pruebas .....	28
2.3.	Herramientas .....	29
2.3.1.	Sistema Operativo Android.....	29
2.3.2.	Android Studio .....	30
2.3.3.	SQLite .....	30
2.3.4.	DB Browser for SQLite .....	31
2.3.5.	MagicDraw .....	32
2.4.	Métricas de calidad de software .....	33
2.4.1.	ISO/IEC 9126.....	33
2.4.1.1.	Funcionalidad .....	35
2.4.1.2.	Confiabilidad .....	36
2.4.1.3.	Usabilidad .....	37
2.4.1.4.	Eficiencia .....	39
2.4.1.5.	Mantenibilidad .....	40
2.4.1.6.	Portabilidad .....	41
2.5.	Seguridad de Software .....	42
2.5.1.	ISO 17799.....	42
2.6.	Métodos de Estimación de Costos de Software .....	42

2.6.1.	Modelos de Estimación .....	43
2.6.2.	Método de Estimación de Costo COCOMO II .....	44
3.	MARCO APLICATIVO.....	49
3.1.	Fases de la Metodología Mobile-D.....	50
3.1.1.	Fase de Análisis.....	50
3.1.1.1.	Requisitos de Software.....	50
3.1.1.2.	Principios pedagógicos y didácticos aplicables .....	53
3.1.1.3.	Requerimientos Funcionales y no Funcionales .....	53
3.1.1.4.	Planificación .....	54
3.1.2.	Fase de Diseño .....	55
3.1.2.1.	Tipos de Diseño .....	56
3.1.3.	Fase Desarrollo.....	57
3.1.3.1.	Modelo de Negocio. ....	57
3.1.2.2.	Iteraciones .....	62
3.1.3.	Fase de Prueba .....	65
3.1.3.2.	Prueba de campo .....	65
3.1.3.3.	Muestra poblacional.....	65
3.1.3.4.	Resultados Obtenidos. ....	67
4.	CALIDAD Y SEGURIDAD .....	69
4.1.	Calidad del Software.....	69
4.1.1.	Funcionalidad .....	69
4.1.2.	Fiabilidad .....	73
4.1.3.	Usabilidad.....	74
4.1.3.1.	Test de Usuario Final .....	74
4.1.4.	Eficiencia .....	75



4.1.5. Mantenibilidad.....	75
4.1.6. Portabilidad .....	76
4.2. Resultado total .....	77
4.3. Seguridad del tutor.....	78
4.3.1. Factor de Seguridad ISO 17799 .....	78
4.3.1.1. Confidencialidad .....	78
4.3.1.2. Integridad.....	78
4.3.1.3. Disponibilidad .....	78
<b>CAPITULO V .....</b>	<b>80</b>
<b>5. Análisis de Costo/Beneficio .....</b>	<b>80</b>
<b>5.1. Estimación de Costos de Software .....</b>	<b>80</b>
5.1.1. Costo del Software Desarrollado .....	80
5.1.2. Costo de implementación del proyecto .....	84
5.1.3. Costo de elaboración del proyecto .....	84
5.1.4. Costo total del proyecto.....	85
<b>5.2. Análisis de Beneficios .....</b>	<b>85</b>
<b>CAPITULO VI .....</b>	<b>86</b>
<b>6. CONCLUSIONES Y RECOMENDACIONES.....</b>	<b>86</b>
<b>6.1. Conclusiones. ....</b>	<b>86</b>
<b>6.2. Recomendaciones. ....</b>	<b>87</b>

## **BIBLIOGRAFÍA**

## **ANEXOS**

## ÍNDICE DE FIGURAS

<b>CAPÍTULO II</b>	<b>Página</b>
<b>Figura 2.1.</b> Equilibrio Estático.....	21
<b>Figura 2. 2.</b> Equilibrio Cinético .....	22
<b>Figura 2.3.</b> Representación Gráfica de las fases o etapas de Mobile-D.....	24
<b>Figura 2.4.</b> Ciclo de Vida de la Metodología .....	26
<b>Figura 2.5.</b> Norma de Evaluación ISO/IEC 9126 .....	34
<b>Figura 2.6.</b> Evaluación Interna, externa y Calidad de Uso ISO/IEC 9126 .....	35
<b>Figura 2.7.</b> Característica de funcionalidad .....	36
<b>Figura 2.8.</b> Característica de Confiabilidad.....	37
<b>Figura 2.9.</b> Característica de Usabilidad .....	38
<b>Figura 2.10.</b> Característica de Eficiencia.....	39
<b>Figura 2.11.</b> Característica de Mantenimiento .....	40
<b>Figura 2.12.</b> Característica de portabilidad .....	41
<b>Figura 2.13.</b> Atributos del Método COCOMO II .....	47
<b>CAPÍTULO III</b>	
<b>Figura 3.1.</b> Modelado según los requisitos .....	51
<b>Figura 3.2.</b> Diagrama de casos de uso.....	58
<b>Figura 3.3.</b> Base de datos del tutor Inteligente Móvil. ....	59
<b>Figura 3.4.</b> Diagrama de secuencias de registro de Usuarios .....	59
<b>Figura 3.5.</b> Diagrama de secuencias de Visualizador de temas .....	60
<b>Figura 3. 6.</b> Diagrama de secuencias de Evaluación del Estudiante .....	60
<b>Figura 3.7.</b> Diagrama de clase para la estructura de registro de usuario.....	62

<b>Figura 3.8.</b> Interfaz de inicio y registro de Usuario.....	62
<b>Figura 3.9.</b> Estructura de datos para el Contenido temático.....	63
<b>Figura 3.10.</b> Interfaz de Contenido.....	63
<b>Figura 3.11.</b> Estructura de datos para la evaluación.....	64
<b>Figura 3.12.</b> Interfaz de Evaluaciones .....	64

## ÍNDICE DE TABLAS

<b>CAPÍTULO II</b>	<b>Página</b>
<b>Tabla 2.1.</b> Modelo Básico del Método COCOMO II .....	45
<b>Tabla 2. 2.</b> Ecuaciones del Método COCOMO II.....	46
<b>CAPÍTULO III</b>	
<b>Tabla 3.1.</b> Metodología Mobile-D combinada con ISE. ....	49
<b>Tabla 3.2.</b> Requerimientos Funcionales y no funcionales .....	53
<b>Tabla 3.3.</b> Planificación de Actividades. ....	54
<b>Tabla 3.4.</b> Datos de entorno para el desarrollo .....	55
<b>Tabla 3.5.</b> Funciones del Prototipo .....	57
<b>Tabla 3.6.</b> Actores y Roles .....	57
<b>Tabla 3.7.</b> Especificaciones de caso de uso para la visualización del módulo tutor. .....	60
<b>Tabla 3.8.</b> Especificaciones de caso de uso para la visualización del módulo estudiante.....	61
<b>Tabla 3. 9.</b> Pruebas de campo a la población estudiantil .....	65
<b>Tabla 3.10.</b> Tiempo de Desarrollo Curricular de temas.....	68
<b>CAPÍTULO IV</b>	
<b>Tabla 4.1.</b> Valores para el ajuste de complejidad .....	70
<b>Tabla 4.2.</b> Factores de Ajuste.....	70
<b>Tabla 4.3.</b> Cálculo de puntos de función sin Ajustar.....	71
<b>Tabla 4.4.</b> Factores de ajuste usabilidad .....	74
<b>Tabla 4.5.</b> Factores de ajuste de eficiencia.....	75
<b>Tabla 4.6.</b> Factores de ajuste de mantenibilidad.....	76

<b>Tabla 4.7.</b> Factores de ajuste de portabilidad.....	76
---	----

<b>Tabla 4.8.</b> Calidad Total.....	77
--------------------------------------	----

## **CAPÍTULO V**

<b>Tabla 5.1.</b> Coeficiente de Modelo COCOMO II .....	81
---	----

<b>Tabla 5. 2.</b> Ecuaciones del Modelo COCOMO II.....	81
---	----

<b>Tabla 5.3.</b> Calculo de los atributos FAE.....	82
---	----

<b>Tabla 5.4.</b> Costo de Elaboración del Proyecto .....	84
---	----

<b>Tabla 5.5.</b> Costo Total del proyecto.....	85
---	----

# **CAPITULO I**

## **1. MARCO PRELIMINAR**

### **1.1. Introducción**

En la actualidad los dispositivos móviles han tomado mucha relevancia, porque nos brindan beneficios de acceso libre al entretenimiento, a la información y a la comunicación, es una herramienta que contribuye en la educación uno de los derechos fundamentales de los ciudadanos. El uso de esta tecnología en estos últimos tiempos ha coadyuvado en la adquisición de conocimientos en las personas, por la facilidad en el acceso a la información y la innovación constante. Se han desarrollado diferentes programas informáticos especializados para diferentes áreas orientados a la educación los cuales se han convertido como una alternativa en la educación para el desarrollo del proceso de enseñanza – aprendizaje, misma que puede despertar el interés en los estudiantes y mejorar la enseñanza viéndose reflejado en el rendimiento estudiantil.

Los estudiantes de quinto de Secundaria de la Unidad Educativa “Dionicio Morales” presentan ciertas dificultades en la comprensión de contenidos y la resolución de problemas en el área de física específicamente en los temas de estática y dinámica durante su proceso de aprendizaje y construcción de conocimientos, debido a las limitaciones de tiempo en avance curricular, la falta de incorporación nuevas tecnologías y medios didácticos para la enseñanza, la cual causa desinterés, y bajo rendimiento académico en pruebas trimestrales.

En este sentido el presente trabajo surge por la necesidad innovar y ayudar al educando en su proceso de formación, con el desarrollo de un Tutor Inteligente Móvil capaz de colaborar en el aprendizaje de temas como: la estática y dinámica, brindando un material didáctico con información de contenidos, resolución de problemas y la evaluación de conocimientos. Es una herramienta que se puede ajustar según a las necesidades del estudiante, constituyéndose en un material de apoyo que permita al estudiante construir sus propios conocimientos y de mejorar la calidad educativa.

Para su desarrollo del presente trabajo se utilizó la Metodología de Ingeniería de Software Educativo (MEISE) propuesta por Antonieta Abud, combinando con las fases de Mobile-D ya que será un sistema orientado a los Smartphones con sistema operativo Android 4.2 o superior, de esta manera nuestro sistema será portátil, novedoso, didáctico y accesible. Además, el software se desarrollará en Android Studio, con la base de datos en SQLite.

## **1.2. Antecedentes de la Investigación**

Los Tutores Inteligentes comenzaron a desarrollarse en los años ochenta con la idea de poder impartir el conocimiento usando alguna forma de inteligencia para poder asistir y guiar al estudiante en su proceso de aprendizaje. Se buscó emular el comportamiento de un tutor humano, es decir a través de un sistema que pudiera adaptarse al comportamiento del estudiante, identificando la forma en que el mismo resuelva un problema a fin de poder brindarle ayuda cuando lo requiera.

Entre los trabajos realizados para el ámbito internacional área de física se mencionan los siguientes:

En el Instituto Politécnico Nacional de la Ciudad de México:

**Título:** “LABORATORIO VIRTUAL Y AMBIENTE HIBRIDO CON SISTEMAS TUTOR INTELIGENTE PARA LA ENSEÑANZA DE LA FÍSICA”

**Autor:** Daniel Sánchez, Cesar Mora y Ricardo García

**Descripción:** Es un tutor inteligente que detecta falencias que se tiene en la teoría o técnica de como tener un conocimiento de ambiente hibrido aplicando leyes de la física, con método científico, evaluado mediante sistemas de tutor inteligente.

Así mismo mencionamos trabajos relacionados con el área de física desarrollados en el ámbito nacional de la Universidad Mayor de San Andrés en la Carrera Informática:

**Título:** “LABORATORIO VIRTUAL VÍA WEB PARA LA MATERIA DE FISICA (VECTORES)”

**Autor:** Javier Teodoro Patzi Collorana

**Año:** 2011

**Herramientas:** Las herramientas utilizados para el desarrollo de este sistema fue: Flash, PHP, Photoshop y Xara – Webstyle.

**Metodología:** Para el desarrollo del prototipo se utilizó la metodología de OOHDM u Object Oriented Hypermedia Design Methodology.

**Descripción:** El laboratorio Virtual en la Materia de Física ayuda con la parte teórica para la resolución de ejercicios de forma analítica con la aplicación de teoremas y de forma interactiva.

**Título:** “SISTEMA INTELIGENTE PARA LA ENSEÑANZA DE FÍSICA PARA PREUNIVERSITARIOS”

**Autor:** Ronald Charca Choque

**Año:** 2015

**Herramientas:** Las herramientas utilizados para el desarrollo de este sistema fue: PHP, lenguaje de programación; MySQL, gestor de Base de Datos: JavaScript. ActionScript, para animaciones Html y Css Adobe flash

**Metodología:** Para el desarrollo del prototipo se utilizó la metodología de Prometheus. Como apoyo para la documentación, se utilizó los diagramas de AUML, como el diagrama de casos de uso, de secuencia y de clases.

**Descripción:** Sistema Inteligente para la Enseñanza de Física para Preuniversitarios es una herramienta de ayuda para un ordenador que contiene la parte teórica para la resolución de ejercicios de forma analítica y la evaluación de conocimientos.



**Título:** “SIMULADOR 3D DE CINEMÁTICA PARA ESTUDIANTE DE NIVEL SECUNDARIO”.

**Autor:** Javier Teodoro Patzi Collorana

**Año:** 2016

**Herramientas:** Las herramientas utilizados para el desarrollo de este sistema fue: Unity 3D, Blender y el lenguaje de programación C#.

**Metodología:** Para el desarrollo del prototipo se utilizó la metodología MEISE, metodología de ingeniería de software educativo.

**Descripción:** Es una herramienta para el proceso de enseñanza aprendizaje que realiza simulaciones en 3D en el tema de Cinemática.

Algunos trabajos de investigación desarrollados al tutor inteligente y la educación en la carrera de Ingeniería de Sistemas de la Universidad Pública de El Alto se tienen:

**Título:** “AGENTE INTELIGENTE EN LA EDUCACIÓN PARA EL APRENDIZAJE DE PROBLEMAS MATEMATICOS”

**Autor:** Rolman Condori Mamani

**Año:** 2016

**Herramientas:** Se utilizaron para el desarrollo fueron el lenguaje GUIDE – MATLAB en el entorno de programación visual.

**Metodología:** Para el desarrollo del prototipo se utilizó las metodologías de ICONIX y metodología de desarrollo orientado a agentes Prometheus.

**Descripción:** El agente inteligente se desarrolló para el aprendizaje de resolución de problemas en el área de matemáticas, orientado a estudiantes del nivel primario.

**Título:** “SOFTWARE EDUCATIVO MULTIMEDIA PARA LA ASIGNATURA DE FÍSICA – 3ro DE SECUNDARIA”

**Autor:** Hernán Quispe Conurana

**Año:** 2016

**Metodología:** Utilizando la metodología Ingeniería de Software Educativo por Galvis. Las herramientas que se utilizaron para el desarrollo fueron el lenguaje PHP, **CodeIgniter**.

**Descripción:** El software educativo es una herramienta que ha sido desarrollado para el apoyo en el aprendizaje de los diferentes contenidos en el área de física para los estudiantes de 3ro de secundaria en el marco de la ley 070.

### **1.3. Planteamiento del Problema**

#### **1.3.1. Problema Principal**

Los estudiantes de quinto de secundaria de la Unidad Educativa “Dionicio Morales” ubicada en la zona Bautista Saavedra de la ciudad de El Alto del departamento de La Paz, presentan dificultades en el proceso de aprendizaje de contenidos de estática y dinámica, debido a que las clases impartidas por el docente son insuficientes por el tiempo limitado en el avance curricular de contenidos, y la falta de implementación de nuevas tecnologías y estrategias de enseñanza. Estos factores conllevan a que los estudiantes se queden con muchas dudas, las cuales no son aclaradas en su debido momento, teniendo así poca comprensión de los conceptos y definiciones de estos temas, la cual causa que el estudiante no pueda resolver con facilidad los problemas planteados, por estas razones los estudiantes demuestran un bajo rendimiento académico en las pruebas trimestrales.

Por lo tanto, se plantea el siguiente Problema de Investigación:

¿De qué manera el tutor Inteligente coadyuvara en el Proceso de Aprendizaje de Estática y Dinámica en los estudiantes de quinto de secundaria en la Unidad educativa Dionicio Morales para que sea optimo?

### **1.3.2. Problemas Secundarios**

- El proceso de enseñanza es monótono, por lo que causa desinterés en los estudiantes por aprender los contenidos de estática y dinámica.
- El tiempo limitado en el desarrollo curricular, ocasiona que los estudiantes se quedan con las dudas que no son aclaradas.
- La asimilación de conocimientos insuficiente en estática y dinámica, provoca desmotivación por seguir aprendiendo.
- Los docentes no cuentan con un medio de apoyo de programa informático para impartir sus clases.
- Escaso uso de equipos computacionales para un buen aprendizaje, genera el desperdicio de recursos tecnológicos.

## **1.4. Objetivos**

### **1.4.1. Objetivo General**

Desarrollar un Tutor Inteligente Móvil para la Enseñanza de Estática y Dinámica en estudiantes de quinto de Secundaria en la Unidad Educativa “Dionicio Morales”, que permitirá optimizar la asimilación de conocimientos en un tiempo mínimo y de forma dinámica.

### **1.4.2. Objetivos Específicos**

- Diseñar un material didáctico para la parte teórica conceptual y resolución de ejercicios que contendrá conocimientos de estática y dinámica.
- Implementar una base de datos para el registro de usuarios.
- Desarrollar un módulo evaluador que permita medir el nivel de aprendizaje de los estudiantes, en base a preguntas planteadas.

- Facilitar a los estudiantes una herramienta educativa, para apoyar al docente en el desarrollo curricular de temas y la búsqueda de un buen aprendizaje de sus estudiantes.
- Despertar el interés de los estudiantes dentro y fuera de la clase sobre el avance del desarrollo curricular.

## **1.5. Justificación**

### **1.5.1. Justificación Técnica**

El desarrollo del presente trabajo se justifica técnicamente porque conlleva el uso de tecnología informática, ya cada estudiante cuenta con un dispositivo móvil la cual no es utilizada de manera eficiente como una herramienta complementaria que apoye en su formación de aprendizaje. Por lo que el Tutor Inteligente Móvil ayudara a los estudiantes de quinto de secundaria que cuente con un dispositivo móvil y que tengan conocimientos previos de estos temas, aprovechando las potencialidades de los dispositivos móviles en cuanto a imagen, video, audio, texto, entre otros que motivan al proceso de enseñanza – aprendizaje de los estudiantes de quinto de secundaria.

### **1.5.2. Justificación Económica**

Económicamente, el desarrollo del tutor inteligente móvil se justifica porque pretende ayudar a estudiantes de unidad educativa, que cuenten con un dispositivo móvil, tendrá la posibilidad de acceder a un aprendizaje sin realizar gastos adicionales y mejorar su rendimiento académico en sus exámenes. Asimismo, se convierte en un material de apoyo, no tendrá gastos económicos en los libros referidos a la materia teniendo así minimizar recursos económicos.

### **1.5.3. Justificación Social**

Socialmente, el desarrollo del tutor inteligente móvil se justifica porque se genera una herramienta en la educación que ayuda principalmente docente para el desarrollo de aprendizaje de los estudiantes que cursan quinto de secundaria de la

Unidad educativa Dionicio Morales y público en general que desee aprender y reforzar sus conocimientos en los temas de estática y dinámica.

## **1.6. Metodología**

Para el desarrollo del modelo y prototipo de Tutor Inteligente Móvil se utilizará la Metodología Mobile-D, orientada a la creación de aplicaciones móviles combinándolas con las cinco etapas de la Metodología de Ingeniería de Software Educativo MeISE de Alvaro Galvis (1994), de tal forma que se enfoca a atender los aspectos técnicos y pedagógicos del producto. Como apoyo para la documentación, se utilizará los diagramas de UML como el diagrama de casos de uso, de secuencia y de clases.

## **1.7. Herramientas**

Las herramientas que se utilizar para el desarrollo del sistema tutor inteligente móvil son las que se a conocer a continuación:

### **1.7.1. Android Studio**

Android Studio es el entorno de desarrollo integrado (IDE) oficial para el desarrollo de aplicaciones para Android. Posee un potente editor de códigos y herramientas para desarrolladores, Android Studio ofrece aún más funciones que aumentan tu productividad durante la compilación de apps para Android

### **1.7.2. SQLite**

Base de datos Open Source, es muy popular en muchos dispositivos móviles pequeños, como Android. El gestor de base de datos por defecto de Android es Lite. SQLite es una base de datos transaccional ligera que ocupa una cantidad muy pequeña de espacio en disco y memoria, de manera que es la elección perfecta para crear bases de datos en sistemas operativos para móviles como Android o IOS.

### **1.7.3. DB Browser for SQLite**

DB Browser for SQLite es una aplicación **gratuita y de código abierto** diseñada para facilitar la creación y administración de las bases de datos con SQLite. Mientras que para poder trabajar con estas bases de datos es necesario aprenderse una gran cantidad de comandos SQL, aumentando la probabilidad de que algo salga mal y hagamos que nuestra base de datos deje de funcionar correctamente.

### **1.7.4. Photoshop y Xara – Visio.**

Esta herramienta nos ayudara para el diseño de imágenes y mostrar de forma clara algunos ejemplos y mejor aspecto a la página.

## **1.8. Límites y Alcances**

### **1.8.1. Limites**

El presente trabajo está limitado por los siguientes aspectos:

- Este sistema no captura totalmente la pedagogía de un experto, como el caso del docente además de poseer conocimiento, años de experiencia y habilidades pedagógicas, lo que hace que el tutor inteligente sea una herramienta de apoyo en el aprendizaje.
- El sistema trabajará solo en el idioma español, por lo que es limitado en cuanto al idioma.
- El tutor inteligente móvil no podrá ser accedido desde cualquier dispositivo, ya que solamente ésta desarrollada para el sistema Android.

### **1.8.2. Alcances**

Entre los alcances del presente trabajo de investigación, podemos mencionar las siguientes:

- ✓ **El módulo tutor**, contiene información relevante de las definiciones, conceptos, formulas y resolución de ejercicios, con un diseño didáctico de contenidos de la estática y dinámica.
- ✓ **El módulo estudiante**, almacena la información cognitiva del estudiante sobre cuanto conoce las definiciones, resolución de ejercicios de los temas.
- ✓ **El módulo interface**, permitirá captar el interés del estudiante, en el desarrollo de adquisición de conocimientos, conformada por una interfaz amigable, dinámico y didáctico.
- ✓ **El Módulo Evaluación**, realizara el seguimiento en el proceso de evaluación al estudiante, mediante un conteo de la cantidad de respuestas correctas e incorrectas marcadas por el estudiante.

### **1.9. Aportes**

El desarrollo del prototipo del Tutor Inteligente Móvil será un aporte muy importante para los estudiantes de quinto de Secundaria de la Unidad Educativa “Dionicio Morales” ya que contribuirá en el proceso de aprendizaje de manera fácil y didáctica, optimizando el acceso a conocimientos de manera real y coherente. El estudiante tendrá el material de las lecciones de estática y dinámica, evitando así la pérdida de tiempo ya que se podrá acceder en cualquier momento y cualquier lugar desde un dispositivo móvil Android, es único y responde a una necesidad de los estudiantes para mejorar su aprendizaje.

## **CAPITULO II**

### **2. MARCO TEÓRICO**

#### **2.1. Introducción**

La educación en la actualidad demanda la búsqueda de nuevas estrategias, técnicas, herramientas, y métodos de enseñanza, para obtener resultados esperados en el aprendizaje de conocimientos de los estudiantes del nivel secundaria. Se debe formar estudiantes que tengan la capacidad de poder construir sus conocimientos propios, aplicarlos a diferentes contextos de nuestra vida real, y de esta manera mejorar la calidad educativa.

La incursión de la tecnología en el ámbito educativo es muy importante debido a que estas se han convertido en medios alternativos que ayudan al estudiante en su proceso de formación académica. Una de las tecnologías que ha sido implementado en el proceso de enseñanza aprendizaje son los tutores Inteligentes en diferentes áreas, estos son programas informáticos para equipos computacionales y/o dispositivos móviles. El uso de la tecnología ha demostrado mejoras en los estudiantes en diferentes áreas de conocimiento, aprendizaje cooperativo, pensamiento crítico, pensamiento creativo, desarrollo de la psicomotricidad en los niños, resolución de problemas y una buena comprensión de contenidos, obteniendo buenos resultados de desempeño y rendimiento, a través de la evaluación de aprendizajes.

A continuación, abarcaremos la base teórica, donde se detallarán algunos postulados científicos sobre aquella que es la teoría del conocimiento mismo que servirá para sustentar la investigación, en ese sentido se procederá a seleccionar los aspectos más relevantes, importantes.

##### **2.1.1. Inteligencia Artificial**

La IA es la rama de la ciencia que se encarga del estudio de la inteligencia en elementos artificiales y, desde el punto de vista de la ingeniería, propone la creación



de elementos que posean un comportamiento inteligente. Dicho de otra forma, la IA pretende construir sistemas y máquinas que presenten un comportamiento que, si fuera llevado a cabo por una persona, se diría que es inteligente. El aprendizaje, la capacidad de adaptación a entornos cambiantes, la creatividad, etc., son facetas que usualmente se relacionan con el comportamiento inteligente. Además, la IA es muy interdisciplinar, y en ella intervienen disciplinas tan variadas como la Neurociencia, la Psicología, las Tecnologías de la Información, la Ciencia Cognitiva, la Física, las Matemáticas, etc. (Gómez García, 2007)

La Inteligencia Artificial conforma una pieza más de los planes de estudios de las titulaciones de informática. Dentro de esta parcela, Técnicas de Inteligencia Artificial pretende ser una asignatura que vaya más allá de la introducción en las técnicas de la IA y que lleve a los estudiantes a un estudio más profundo del tema, sin olvidar que ha de ser lo suficientemente dinámica para adaptarse a los nuevos enfoques en el área y que dote a los estudiantes de una visión amplia sobre el conjunto de la IA de manera que se convierta en punto de partida para estudios más específicos en el área como la visión artificial, el aprendizaje, la robótica, el lenguaje natural, etc. (Manuel Pérez & Viejo, 2008)

La Inteligencia artificial son los procesos de simulación de la inteligencia natural mediante algunas máquinas que piensen como humanos, especialmente sistemas informáticos, es decir, un software que replica las capacidades humanas, diseñadas para tareas específicas o también tareas amplias. Estas incluyen el aprendizaje, el razonamiento y la autocorrección.

### **2.1.2. Inteligencia Artificial En La Educación**

La aplicación de la Inteligencia Artificial (IA) en la Educación, constituye actualmente un campo de creciente interés donde se tratan, fundamentalmente, de aplicar las técnicas de la IA al desarrollo de sistemas de enseñanza asistida por ordenador con el propósito de construir sistemas de enseñanza inteligentes. (Gonzales, 2004)

La educación es una de las áreas de aplicación de la IA. Se plantea que el uso de las técnicas de IA en la elaboración de software educativos permite que los sistemas:

- Se adapten mejor a las características de los estudiantes teniendo en cuenta el historial de actuaciones del estudiante y no a una respuesta aislada.
- Generen problemas, soluciones y diagnósticos cómo y cuándo se necesite durante una sesión de aprendizaje.

Una de áreas donde la Inteligencia Artificial toma mayor relevancia es en la educación, a través de softwares educativos con aprendizaje interactivo, adaptativo y personalizado, estas colaboran fundamentalmente en las dificultades de acceso al aprendizaje. Los equipos computacionales y/o dispositivos móviles las cuales han facilitado el proceso de enseñanza de contenidos

### **2.1.3. Definición de Tutor Inteligente.**

Los Sistemas Tutores Inteligentes han evolucionado desde sus inicios y son un ejemplo de sistemas educativos adaptables. La adaptabilidad es una importante característica de estos sistemas, pues en lugar de presentar información estática, el conocimiento es presentado de manera personalizada y dinámica al estudiante de acuerdo a su propio comportamiento. De esta forma, cada estudiante recibe retroalimentación de acuerdo a su interacción con el sistema. Estos sistemas pueden trabajar con reglas que permiten encontrar habilidades dominadas por los estudiantes y reforzar aquellas en donde se detectan carencias en su comprensión (Andrade, 2017).

Los Tutores Inteligentes pueden resultar muy efectivos en mejorar el aprendizaje de los estudiantes. Sin embargo, en la actualidad su uso es aún limitado. Mientras que los sistemas de e-learning han proliferado recientemente, pocos realmente utilizan tutores inteligentes. Una de las principales razones de esta falta de interés radica en la dificultad para crear un Tutor Inteligente aún por personas experimentadas. Por lo anterior se han creado herramientas que facilitan su desarrollo y que permiten

de manera automática o a través de marcos de trabajo, la creación de estos sistemas en un tiempo mucho menor.

Un Tutor Inteligente es un tutor particular de un estudiante, asistida desde un computador o algún dispositivo electrónico que brinda servicio de enseñanza. Pretende poseer los conocimientos y habilidades de un experto en pedagogía, de que enseñar, como enseñar y cuando enseñar de forma autónoma, a la vez permitiendo al estudiante realizar una retroalimentación instantánea de los contenidos desarrollados por el docente.

#### **2.1.4. Proceso de Enseñanza – Aprendizaje**

La enseñanza se define como un proceso instructivo orientado a educar a terceros, facilitando su aprendizaje. Por su parte, el aprendizaje se conceptúa como un proceso que produce un cambio personal en el modo de pensar, sentir y comportarse, respondiendo a los tradicionales saberes de: saber (conjunto de conocimientos), saber hacer (conjunto de habilidades y destrezas) y saber estar/ser (capacidad de integración) (Bricall, 2000)

El proceso de enseñanza-aprendizaje conforma una unidad que tiene como propósito y fin contribuir a la formación integral de la personalidad del futuro profesional, aunque lo sigue dirigiendo el docente, para favorecer el aprendizaje de los diferentes saberes: conocimiento, habilidades y valores; el tipo de intervención que este tenga está sujeta al paradigma con el que se identifica. (Alvarado, 2018)

El proceso de enseñanza aprendizaje es la formación integral de la personalidad del educando, constituyéndose una vía principal para la obtención de conocimientos, patrones de conducta, valores, procedimientos y estrategias de aprendizaje. En éste proceso el estudiante debe apropiarse de las leyes, definiciones, conceptos y teorías de las diferentes áreas que forman parte de un plan de estudios y al mismo tiempo al interactuar con el docente y sus compañeros se van dotando de procedimientos y estrategias de aprendizaje.

#### **2.1.4.1. Enseñanza**

Enseñar viene de la palabra latina “insignare”, cuyo significado es señalar y como se sabe, la sociedad ha dispuesto de múltiples y variados causas, estrategias y recursos para mostrar o señalar. Enseñar en su sentido más amplio significa instruir, educar, preparar para la vida y el trabajo. Desde el punto de vista escolar, dominar científicamente el proceso de enseñanza y dirigido conscientemente de manera de preparar al individuo para asumir la responsabilidad de su autoformación en el cambiante contexto científico tecnológico, el desarrollo de habilidades y competencias a lo largo de la vida, se trata pues de una educación constante que tenga cuenta promover el desarrollo biológico, cognitivo y social del individuo (Castellanos, 2012).

La enseñanza es una actividad realizada donde intervienen 3 elementos: un docente, uno o varios estudiantes y el objeto de conocimiento. El docente es quien se encarga de transmitir sus conocimientos a los estudiantes a través de diversos medios, técnicas y herramienta de apoyo, siendo la fuente de conocimiento, y el estudiante un simple receptor imitando el mismo. A la vez el docente actúa como facilitador, guía, nexos entre los conocimientos y los estudiantes, logrando un proceso de interacción, basado en la iniciativa y el afán de saber de los estudiantes; haciendo el proceso un constante, un ciclo e individualizando de algún modo la educación.

Existen medios utilizados para el proceso de enseñanza, mismos que están basados en la percepción, es decir: pueden ser orales y escritos. Las técnicas que derivan de ellos van desde la exposición, el apoyo en otros textos, técnicas de participación y dinámicas de grupos. Las herramientas más habituales con las cuales se impartía la enseñanza eran tiza, la pizarra, el lápiz, el papel y los libros de texto, las que con el avance científico de nuestros días han evolucionado hasta desarrollar distintos canales para llegar al estudiante. (Ortiz, 2009)

La enseñanza es un conjunto de actividades organizadas y planificadas por el docente, utilizando diferentes medios y estrategias metodológicas apropiados, con

el fin de poder transmitir de una mejor manera determinados contenidos de un plan de estudios, y así tener una buena asimilación y aprendizaje de forma consciente y productiva, estos contenidos son conocimientos, habilidades y actitudes esenciales que un estudiante debe dominar para lograr una educación para la vida, transmitiendo de una manera más efectiva posible.

#### **2.1.4.2. Estrategias de Enseñanza**

Las estrategias de enseñanza se conciben como los procedimientos utilizados por el docente para promover aprendizajes significativos, implican actividades conscientes y orientadas a un fin. (Parra Pineda, 2003) El adecuado y consciente uso de las estrategias, conllevan a una “instrucción estratégica interactiva y de alta calidad” las estrategias utilizadas deben reunir las siguientes características:

- Deberán ser funcionales y significativas, que lleven a incrementar el rendimiento en las tareas previstas con una calidad razonable de tiempo y esfuerzo.
- La instrucción debe demostrar que estrategias pueden ser utilizadas, como pueden aplicarse y cuándo y por qué son útiles y necesarias.
- Los estudiantes deben creer que las estrategias son útiles y necesarias.
- Deben hacer una conexión entre la estrategia enseñada y las percepciones del estudiante sobre el contexto de la tarea.
- Una instrucción eficaz y con éxito genera confianza y creencias de autosuficiencia.
- La instrucción debe ser directa, informativa y explicativa.
- La responsabilidad para generar, aplicar y controlar estrategias eficaces es transferida del instructor al estudiante.
- Los materiales instruccionales deben ser claros, bien elaborados y agradables.

#### **2.1.4.3. Aprendizaje**

El aprendizaje es el proceso a través del cual se adquieren o modifican habilidades, destrezas, conocimientos, conductas o valores como resultado del estudio, la experiencia, la instrucción, el razonamiento y la observación. Este proceso puede ser analizado desde distintas perspectivas, por lo que existen distintas teorías del aprendizaje. El aprendizaje es una de las funciones mentales más importantes en humanos, animales y sistemas artificiales.

El aprendizaje humano está relacionado con la educación y el desarrollo personal. Debe estar orientado adecuadamente y es favorecido cuando el individuo está motivado. El estudio acerca de cómo aprender interesa a la neuropsicología, la psicología educacional y la pedagogía.

El aprendizaje como establecimiento de nuevas relaciones temporales entre un ser y su medio ambiental ha sido objeto de diversos estudios empíricos, realizados tanto en animales como en el hombre. Midiendo los progresos conseguidos en cierto tiempo se obtienen las curvas de aprendizaje, que muestran la importancia de la repetición de algunas predisposiciones fisiológicas, de “los ensayos y errores”, de los períodos de reposo tras los cuales se aceleran los progresos, etc. Muestran también la última relación del aprendizaje con los reflejos condicionados. (Espitia, 2008)

Es un proceso activo, participativo y organizado para una apropiación de conocimientos habilidades, destrezas y la formación de valores a través de la implementación de una serie de medios, estrategias cognitivas con las cuales se puedan lograr una buena la asimilación de contenidos.

#### **2.1.4.4. Estilos de Aprendizaje**

El estilo de aprendizaje es el conjunto de características psicológicas que suelen expresarse conjuntamente cuando una persona debe enfrentar una situación de aprendizaje; en otras palabras, las distintas maneras en que un individuo puede aprender.

Se cree que una mayoría de personas emplea un método particular de interacción, aceptación y procesado de estímulos e información. Las características sobre estilo de aprendizaje suelen formar parte de cualquier informe psicopedagógico que se elabore de un estudiante y pretende dar pistas sobre las estrategias didácticas y refuerzos que son más adecuados para el niño. No hay estilos puros, del mismo modo que no hay estilos de personalidad puros: todas las personas utilizan diversos estilos de aprendizaje, aunque uno de ellos suele ser el predominante. (Espitia, 2008)

#### **2.1.4.5. Estrategias de Aprendizaje**

Las estrategias de aprendizaje constituyen se constituyen en actividades consciente e intencionales que guían las acciones a seguir para alcanzar determinadas metas de aprendizaje por el estudiante. Son procedimientos que se aplican de un modo intencional y deliberado de una tarea y que no pueden reducirse a rutinas automatizadas, es decir, son más simples secuencias o aglomeraciones de habilidades. (Parra Pineda, 2003)

También se pueden definir como conductas y pensamientos que un aprendiz utiliza durante el aprendizaje con la intención influir en su proceso de codificación. (Baggini 2008) las define como secuencias integradas de procedimientos o actividades que se eligen con el propósito de facilitar la adquisición, almacenamiento y/o utilización de la información.

Características de las estrategias de aprendizaje:

- Su aplicación no es automática sino controlada.
- Implican un uso selectivo de los propios recursos y capacidades disponibles.
- Las estrategias están constituidas de otros elementos más simples, que son las didácticas de aprendizaje, las destrezas o habilidades.

### 2.1.5. Dinámica de una partícula.

La dinámica es parte de Mecánica, estudia las relaciones entre el movimiento de una partícula y la fuerza que lo produce. Si el movimiento es rectilíneo, se denomina dinámica lineal y si la trayectoria es una circunferencia se llama dinámica circunferencial. (Pérez Terrel, 2015)

La Dinámica es una de las ramas de la Mecánica que estudia el movimiento de los cuerpos tomando en cuenta las causas que lo producen (Quisbert Callisaya, 2019).

La dinámica es una parte de la mecánica que estudia al movimiento de los cuerpos acelerados, considerando las causas que lo originan dicho movimiento.

#### 2.1.5.1. Fuerza ( $\vec{F}$ ).

Es una magnitud vectorial que modifica el estado de reposo o estado de movimiento de un cuerpo. Toda fuerza aparece como resultado de la interacción de dos cuerpos.

#### 2.1.5.2. Masa ( $m$ ).

La masa es una propiedad inherente de un cuerpo en particular. La masa es una magnitud física escalar, se mide en (kg). En dinámica se define como la relación entre la fuerza y la aceleración que adquiere. (Pérez Terrel, 2015)

La masa es una magnitud escalar, que representa la cantidad de materia que posee un cuerpo.

$$m = \frac{\vec{F}}{\vec{a}}$$

**Donde:**

$m$ : masa de un cuerpo ( $kg, g, lb$ )

$a$ : aceleración de la masa ( $m/s^2, pies/s^2$ )



$\vec{F}$ : Fuerza (*Newton, dinas*)

### 2.1.5.3. **Peso ( $\vec{W}$ )**

El peso de un cuerpo se define como la fuerza con que la tierra lo atrae. Se representa mediante un vector que indica en todo instante el centro de la tierra. (Pérez Terrel, 2015).

$$\vec{W} = m \cdot g$$

**Donde:**

$\vec{W}$ : peso del cuerpo (*Newton, dinas*)

$m$ : masa de un cuerpo (*kg, g, lb*)

$g$ : aceleración de la masa (*m/s<sup>2</sup>, pies/s<sup>2</sup>*)

### 2.1.5.4. **Leyes de Newton**

#### 2.1.5.4.1. **1ra Ley de Newton (Principio de la Inercia).**

Todo cuerpo permanece en estado de reposo o de movimiento rectilíneo con una velocidad constante, mientras no actúe una fuerza externa que modifique dicho estado.

#### 2.1.5.4.2. **2da Ley de Newton (Ley de la Fuerza).**

La aceleración de un objeto es directamente proporcional a la fuerza neta que actúa sobre él e inversamente proporcional a su masa.

#### 2.1.5.4.3. **3ra Ley de Newton. (Ley de Acción y Reacción).**

Si sobre un cuerpo se le aplica una fuerza a otro (acción), entonces él otro le aplica una fuerza igual y en sentido contrario al primero (reacción).

### 2.1.6. Estática

La Estática es la parte de la mecánica que estudia el sistema de fuerzas, que actúan sobre un cuerpo, para que este se encuentre en equilibrio (Quisbert Callisaya, 2019).

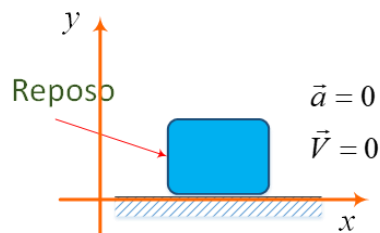
La Estática es una parte de la mecánica que estudia las condiciones de equilibrio que deben cumplir las fuerzas que actúan sobre un cuerpo para que se mantenga en equilibrio.

#### 2.1.6.1. Equilibrio

Un cuerpo se encuentra en equilibrio cuando carece de todo tipo de aceleración ( $a = 0$ ). Existen dos tipos de equilibrio:

##### 2.1.6.1.1. Equilibrio Estático

Es cuando un cuerpo no se mueve ( $V = 0$  y  $a = 0$ ).

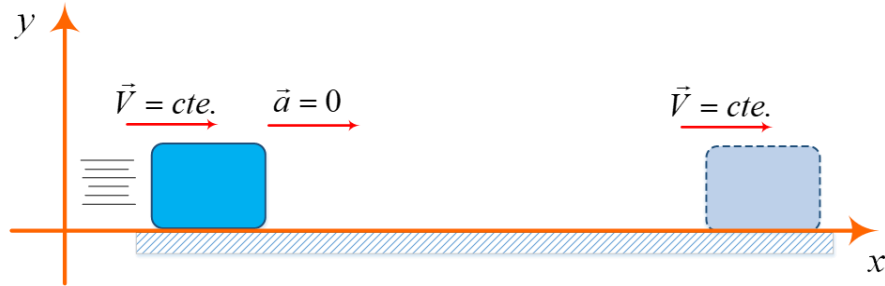


**Figura 2.1.** Equilibrio Estático

**Fuente:** (Elaboración propia)

##### 2.1.6.1.2. Equilibrio Cinético.

Es cuando un cuerpo se mueve en línea recta a velocidad constante.



**Figura 2. 2.** Equilibrio Cinético

**Fuente:** Elaboración propia

### 2.1.6.2. Primera condición de equilibrio.

Si la suma de todas las fuerzas ejercidas sobre un cuerpo es igual a cero. Entonces dicho cuerpo se encuentra en reposo o con movimiento rectilíneo uniforme.

$$\vec{F}_R = \vec{F}_1 + \vec{F}_2 + \vec{F}_3 + \dots + \vec{F}_n = 0 \quad \Rightarrow \quad \sum_{i=1}^n \vec{F}_i = 0$$

### 2.1.6.3. Segunda Condición de Equilibrio

si la suma de momentos de todas las fuerzas que actúan sobre un cuerpo es igual a cero, entonces dicho cuerpo se encuentra en equilibrio.

$$\vec{M}_R = \vec{F}_1 \times d_1 + \vec{F}_2 \times d_2 + \vec{F}_3 \times d_3 + \dots + \vec{F}_n \times d_n = 0$$

$$\vec{M}_R = \vec{M}_1 + \vec{M}_2 + \vec{M}_3 + \dots + \vec{M}_n = 0 \quad \Rightarrow \quad \sum_{i=1}^n \vec{M}_i = 0$$

## 2.2. Metodología

La metodología de desarrollo de software es un conjunto de procesos estructurados, utilizada para planificar y controlar el procedimiento de creación de un sistema de información especializada, así como para disponer de documentación necesaria para posibles cambios de software.

En el desarrollo del presente prototipo Tutor Inteligente Móvil se utilizará la Metodología Mobile-D, orientada a la creación de aplicaciones móviles y la Metodología de Ingeniería de Software Educativo MeISE de Alvaro Galvis (1994), la cual se enfoca a atender los aspectos técnicos y pedagógicos del producto.

### **2.2.1. Metodología Mobile-D**

La metodología Mobile-D fue una creación de un grupo de investigadores del VTT, con la participación importante de las empresas de TI en Finlandia, como parte del proyecto ICAROS, allá por el año 2004. Su diseño se debe de otras metodologías como la metodología XP (eXtreme Programming), RUP (Rational Unified Process) y Crystal methodologies. “Los principios de programación extrema se han reutilizado en lo que se refiere a las prácticas de desarrollo, las metodologías Crystal proporcionaron un input muy valiosos en términos de la escalabilidad de los métodos y el RUP es la base para el diseño completo del ciclo de vida” (UPM, 2009).

El ciclo del proyecto se divide de cinco fases: exploración, inicialización, producción, estabilización y pruebas del sistema figura 2.2. En general, todas las fases (con la excepción de la fase exploratoria) contienen tres días de desarrollo distintos: planificación, trabajo y liberación. Se añadirán días para acciones adicionales en casos particulares (se necesitarán días para la preparación del proyecto en la fase de inicialización, por ejemplo)” (Blanco & Camarero, 2009).



**Figura 2.3.** Representación Gráfica de las fases o etapas de Mobile-D

**Fuente:** (VTT Electronics, 2004)

### 2.2.1.1. Fase de Exploración

El equipo de desarrollo debe generar un plan y establecer las características del proyecto. Esto se realiza en tres etapas: establecimiento de actores, definición del alcance y el establecimiento de proyectos. Las tareas asociadas a esta fase incluyen el establecimiento del cliente (los clientes que toman parte activa en el proceso de desarrollo), la planificación inicial del proyecto y los requisitos de recogida, y el establecimiento de procesos

### 2.2.1.2. Fase de Inicialización

Los desarrolladores preparan e identifican todos los recursos necesarios. Se preparan los planes para las siguientes fases y se establece el entorno técnico como los recursos físicos, tecnológicos y de comunicaciones (incluyendo el entrenamiento del equipo de desarrollo). Esta fase se divide en cuatro etapas: la puesta en marcha del proyecto, la planificación inicial, el día de prueba y día de salida.

### **2.2.1.3. Fase de Producción**

Se repite la programación de tres días (planificación, trabajo, liberación) se repite iterativamente hasta implementar todas las funcionalidades. Primero se planifica la iteración de trabajo en términos de requisitos y tareas a realizar. Se preparan las pruebas de la iteración de antemano. Las tareas se llevarán a cabo durante el día de trabajo, desarrollando e integrando el código con los repositorios existentes. Durante el último día se lleva a cabo la integración del sistema (en caso de que estuvieran trabajando varios equipos de forma independiente) seguida de las pruebas de aceptación.

### **2.2.1.4. Fase de Estabilización**

Se llevan a cabo las últimas acciones de integración para asegurar que el sistema completo funciona correctamente. Esta será la fase más importante en los proyectos multi-equipo con diferentes subsistemas desarrollados por equipos distintos. En esta fase, los desarrolladores realizarán tareas similares a las que debían desplegar en la fase de “producción”, aunque en este caso todo el esfuerzo se dirige a la integración del sistema. Adicionalmente se puede considerar en esta fase la producción de documentación.

### **2.2.1.5. Fase de pruebas y reparación del sistema**

Tiene como meta la disponibilidad de una versión estable y plenamente funcional del sistema. El producto terminado e integrado se prueba con los requisitos de cliente y se eliminan todos los defectos encontrados.

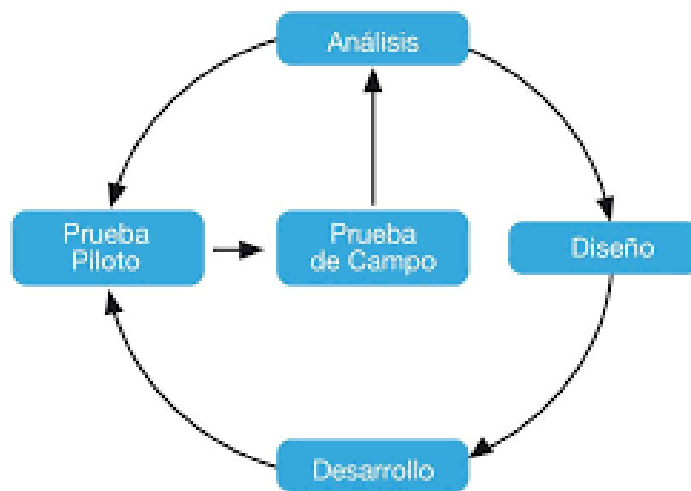
## **2.2.2. Metodología de Ingeniería de Software Educativo**

La metodología de desarrollo de software en el ámbito educativo es la Metodología de Ingeniería de Software Educativo propuesta por Álvaro Galvis. Es una referencia bastante completa y es una buena guía para el desarrollo del software.

En esencia se conservan los grandes pasos o etapas de un proceso sistemático para desarrollo de materiales (análisis, diseño, desarrollo, prueba y ajuste,

implementación). Sin embargo, en este caso se da particular énfasis a los siguientes aspectos: la solidez del análisis, como punto de partida; el dominio de teorías sustantivas sobre el aprendizaje y la comunicación humana, como fundamento para el diseño de los ambientes educativos computarizados; la evaluación permanente y bajo criterios predefinidos, a lo largo de todas las etapas del proceso, como medio de perfeccionamiento continuo del material; la documentación adecuada y suficiente de lo que se realiza en cada etapa, como base para el mantenimiento que requerirá el material a lo largo de su vida útil.

Las etapas de la metodología se ilustran en la figura 2.2.



**Figura 2.4.** Ciclo de Vida de la Metodología

**Fuente:** (Galvis, 1994)

### **2.2.2.1. Fase Análisis**

El análisis nos permite detectar qué situaciones existen, sus causas y posibles soluciones, de esta manera determinar cuáles de éstas son aplicables y podrían generar mejores resultados educativos.

Es por lo anterior que esta metodología se centra en:

- **Características de la población objetivo:** edad (física y mental), sexo, características físicas y mentales (si son relevantes), experiencias previas, expectativas, actitudes, aptitudes, intereses o motivadores por aprender.
- **Conducta de entrada y campo vital:** nivel escolar, desarrollo mental, físico o psicológico, entorno familiar y escolar, etc.
- **Problema o necesidad a atender:** Para establecer la necesidad se puede recurrir a los mecanismos de análisis de necesidades educativas en. Estos mecanismos usan entrevistas, análisis de resultados académicos, etc. para detectar los problemas o posibles necesidades que deben ser atendidas. El problema o necesidad no tiene que estar necesariamente relacionado con el sistema educativo formal, pueden ser necesidades sentidas, económicas, sociales, normativas, etc.
- **Principios pedagógicos y didácticos aplicables:** se debe analizar cómo se ha llevado a cabo el proceso de enseñanza-aprendizaje para establecer cómo debe enfocarse el ambiente, qué factores tomar en cuenta, qué objetivos debe cumplir.
- **Justificación de uso de los medios interactivos:** Para cada problema o necesidad encontrada se debe establecer una estrategia de solución contemplando diferentes posibilidades. El apoyo informático debe ser tomado en cuenta siempre y cuando no exista un mecanismo mejor para resolver el problema: soluciones administrativas, ver si el problema se soluciona al tomar decisiones de tipo administrativo; soluciones académicas, cambios en metodologías de clase; mejoras a los medios y materiales de enseñanza contemplando el uso de medios informáticos. Una vez que se han analizado todas las alternativas se puede decir por qué el uso de medios informáticos es una buena solución. La justificación se puede basar en la no existencia de otro medio mejor y en la relación costo-beneficio para la institución pues puede ser que exista una mejor solución pero que demande mayor tiempo y esfuerzo o un mayor costo económico, etc.
- **Diagramas de interacción:** Permiten ver secuencias de interacción entre el usuario y la aplicación, representando lo que se espera del diálogo y dando



más detalle a la descripción textual de la descripción de la aplicación. Los diagramas de interacción son un formalismo que permite ver la secuencia de acciones entre diferentes partes de la aplicación involucrada en llevar a cabo determinada actividad. Es importante ver la secuencia de acciones para cada escenario de interacción. Con base en estos diagramas se pueden ver cuáles pueden ser las necesidades de información en cada escenario de interacción y se puede ir pensando en cuáles pueden ser los algoritmos que serán usados.

#### **2.2.2.2. Fase Diseño**

- **Educativo** (este debe resolver las interrogantes que se refieren al alcance, contenido y tratamiento que debe ser capaz de apoyar el Sistema Educativo).
- **Comunicacional** (es donde se maneja la interacción entre usuario y máquina, se denomina interfaz).
- **Computacional** (con base a las necesidades se establece qué funciones es deseable que cumpla el Sistemas Educativo en apoyo de sus usuarios, el docente y los estudiantes).

#### **2.2.2.3. Desarrollo**

En esta fase se implementa la aplicación usando la información obtenida anteriormente. Tomando en cuenta las restricciones que se tengan.

#### **2.2.2.4. Fase Pruebas**

##### **2.2.2.4.1. Prueba Piloto**

En esta etapa se pretende ayudar a la depuración del Sistema Educativo a partir de su utilización por una muestra representativa de los tipos de destinatarios para los que se hizo y la consiguiente evaluación formativa. Es imprescindible realizar ciertas validaciones (efectuadas por expertos) de los prototipos durante las etapas de

diseño y prueba en uno a uno de los módulos desarrollados, a medida que estos están funcionales.

#### **2.2.2.4.2. Prueba de Campo**

La prueba de campo de un Sistema Educativo es mucho más que usarlo con toda la población objeto. Si se exige, pero no se limita a esto. Es importante que dentro del ciclo de desarrollo hay que buscar la oportunidad de comprobar, en la vida real, que aquello que a nivel experimental parecía tener sentido, lo sigue teniendo, es decir, si efectivamente la aplicación satisface las necesidades y cumple la funcionalidad requerida.

### **2.3. Herramientas**

Las herramientas que se utilizará para el desarrollo del sistema tutor inteligente móvil son las que se da conocer a continuación:

#### **2.3.1. Sistema Operativo Android**

Fue desarrollado por Android Inc., empresa que en 2005 fue comprada por Google, aunque no fue hasta 2008 cuando se popularizó, gracias a la unión al proyecto de Open Handset Alliance, un consorcio formado por 48 empresas de desarrollo hardware, software y telecomunicaciones, las cuales llegaron a un acuerdo para promocionar los estándares de códigos abiertos para dispositivos móviles.

Google sin embargo, ha sido quien ha publicado la mayoría del código fuente de Android bajo la licencia de Software Apache, una licencia de software libre y de código abierto a cualquier desarrollador.

Dado que Android está basado en el núcleo de Linux, tiene acceso a sus recursos, pudiendo gestionarlo, gracias a que se encuentra en una capa por encima del Kernel, accediendo así a recursos como los controladores de pantalla, cámara, memoria flash.

### **2.3.2. Android Studio**

Es el entorno de desarrollo integrado (IDE) oficial para el desarrollo de apps para Android, basado en IntelliJ IDEA. Además del potente editor de códigos y las herramientas para desarrolladores de IntelliJ, Android Studio ofrece incluso más funciones que aumentan tu productividad cuando desarrollas apps para Android, como las siguientes:

- Un sistema de compilación flexible basado en Gradle. Un emulador rápido y cargado de funciones.
- Un entorno unificado donde puedes desarrollar para todos los dispositivos Android.
- Aplicación de cambios para insertar cambios de códigos y recursos a la aplicación en ejecución sin reiniciar la aplicación.
- Integración con GitHub y plantillas de código para ayudarte a compilar funciones de apps comunes y también importar código de muestra.
- Variedad de marcos de trabajo y herramientas de prueba. Herramientas de Lint para identificar problemas de rendimiento, usabilidad y compatibilidad de la versión, entre otros. Compatibilidad con C++ y NDK.
- Compatibilidad integrada con Google Cloud Platform, que facilita la integración con Google Cloud Messaging y App Engine

### **2.3.3. SQLite**

Es una biblioteca escrita en lenguaje C que implementa un Sistema de gestión de bases de datos transaccionales SQL auto-contenido, sin servidor y sin configuración. El código de SQLite es de dominio público y libre para cualquier uso, ya sea comercial o privado. Actualmente es utilizado en gran cantidad de aplicaciones incluyendo algunas desarrolladas como proyectos de alto nivel.

- SQLite soporta múltiples tablas, índices, triggers y vistas.
- Lee y escribe directamente sobre archivos que se encuentran en el disco duro.

- El formato de la base de datos es multiplataforma y se puede utilizar el mismo archivo en un sistema de 32 y 64 bits.
- Utiliza el espacio en disco que es realmente necesario en cada momento ya que emplea registros de tamaño variable.
- SQL realiza operaciones de manera eficiente y es más rápido que MySQL y PostgreSQL.
- Cuenta con diversas interfaces API, lo que permite trabajar con C++, PHP, Python, Groovy, etc.
- Es totalmente auto contenida es decir que no tiene dependencias externas.
- Cuenta con librerías de acceso para muchos lenguajes de programación.
- Soporta funciones SQL definidas por el usuario (UDF).
- El código fuente es de dominio público y se encuentra muy bien documentado

#### 2.3.4. DB Browser for SQLite

DB Browser for SQLite es una aplicación **gratuita y de código abierto** diseñada para facilitar la creación y administración de las bases de datos con SQLite. Mientras que para poder trabajar con estas bases de datos es necesario aprenderse una gran cantidad de comandos SQL, aumentando la probabilidad de que algo salga mal y hagamos que nuestra base de datos deje de funcionar correctamente.

Esta aplicación (llamada inicialmente Database Browser for SQLite) nació como una alternativa al software Arca Database Xtra, una herramienta comercial para facilitar la creación y edición de las bases de datos SQL. Esta herramienta nació en 2012 preparada para trabajar con las bases de datos SQLite 2.x, sin embargo, a medida que ha ido pasando el tiempo se ha convertido en una de las herramientas imprescindibles cuando trabajamos con bases de datos.

Algunas de las **características que nos ofrece DB Browser for SQLite** son:

- Nos permite crear archivos de bases de datos y compactar archivos ya creados con SQLite.
- Permite crear, definir y eliminar tablas.

- Permite crear, definir y eliminar índices.
- Nos permite buscar, editar, añadir o eliminar entradas.
- Cuenta con un potente buscador de entradas.
- Importa y exporta entradas en modo texto.
- Importa y exporta tablas en ficheros CSV.
- Importa y exporta bases de datos en volcados SQL.
- Nos permite examinar los logs SQL

### **2.3.5. MagicDraw**

Es una herramienta de modelado de procesos de negocio creada por NoMagic Inc., diseñada para analistas de negocio, analistas de software, programadores y trabajadores de control de calidad entre otros. (Pinto, 2017)

**Descripción.** Es una herramienta de desarrollo dinámico lo que facilita el diseño y el análisis de sistemas orientados a objetos y bases de datos. Es usada para modelar UML 2.5, con el fin de visualizar, especificar y documentar cada una de las partes que comprende el desarrollo de software.

MagicDraw proporciona un rápido aprendizaje del interfaz de la herramienta, mediante un sencillo acceso a las operaciones más comunes. El menú de acceso es personalizable, teniendo la capacidad de poner a un solo clic las funciones que el usuario desee. Así mismo, la creación de diagramas UML también es rápida, con completado automático de atributos, operaciones y parámetros, además de corrección y verificación de errores en tiempo real. Se permiten múltiples vistas de un mismo modelo, MagicDraw es capaz de generar un diagrama de jerarquía en dos segundos partiendo de un modelo dado. Con los hipervínculos de MagicDraw, se pueden vincular a cualquier elemento del modelo, a elementos de otros diagramas, a diferentes modelos o a otros archivos y documentos fuera del modelo.

De esta forma se podrá personalizar la navegación por el modelo a las necesidades del usuario.

## **2.4. Métricas de calidad de software**

La gestión de la calidad de software es una actividad protectora que incorpora tanto control como aseguramiento en la calidad que se aplica a cada paso en el proceso del software. Para medir la calidad del software nos basaremos en la Norma internacional de ISO/IEC 9126 la cual nos permitirá evaluar la calidad del Tutor Inteligente Móvil.

### **2.4.1. ISO/IEC 9126**

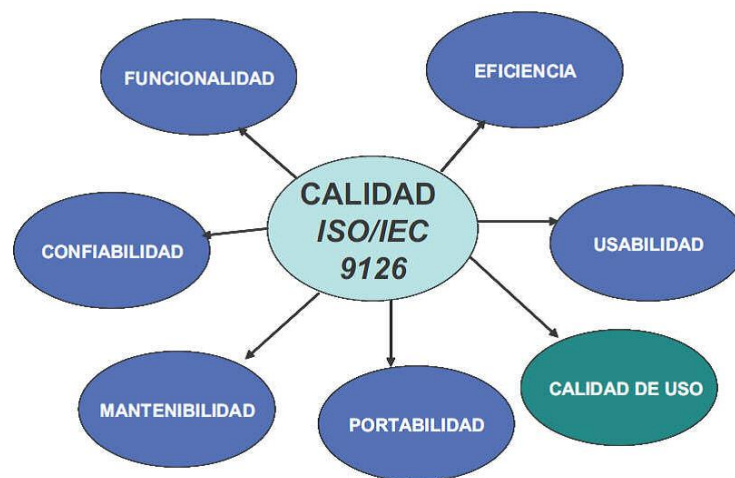
Esta norma Internacional fue publicada en 1992, la cual es usada para la evaluación de la calidad de software, llamado “Information technology-Software product evaluation-Quality characteristics and guidelines for their use”; o también conocido como ISO 9126 (o ISO/IEC 9126). Este estándar describe 6 características generales: Funcionalidad, Confiabilidad, Usabilidad, Eficiencia, Mantenibilidad, y Portabilidad.

La norma ISO/IEC 9126 permite especificar y evaluar la calidad del software desde diferentes criterios asociados con adquisición, requerimientos, desarrollo, uso, evaluación, soporte, mantenimiento, aseguramiento de la calidad y auditoria de software. Los modelos de calidad para el software se describen así:

**Calidad interna y externa:** Especifica 6 características para calidad interna y externa, las cuales, están subdivididas. Estas divisiones se manifiestan externamente cuando el software es usado como parte de un sistema Informático, y son el resultado de atributos internos de software.

**Calidad en uso:** Calidad en uso es el efecto combinado para el usuario final de las 6 características de la calidad interna y externa del software. Especifica 4 características para la calidad en uso.

Al unir la calidad interna y externa con la calidad en uso se define un modelo de evaluación más completo, se puede pensar que la usabilidad del modelo de calidad externa e interna pueda ser igual al modelo de calidad en uso, pero no, la usabilidad es la forma como los profesionales interpretan o asimilan la funcionalidad del software y la calidad en uso se puede asumir como la forma que lo asimila o maneja el usuario final. Si se unen los dos modelos, se puede definir que los seis indicadores del primer modelo tienen sus atributos y el modelo de calidad en uso sus 4 indicadores pasarían hacer sus atributos, mirándolo gráficamente quedaría así:



**Figura 2.5.** Norma de Evaluación ISO/IEC 9126

**Fuente:** (ISO/IEC, 2007)

Se establecen categorías para las cualidades de la calidad externa e interna y calidad en uso del software, teniendo en cuenta estos 7 indicadores (funcionalidad, confiabilidad, utilidad, eficiencia, capacidad de mantenimiento, portabilidad y calidad en uso), que se subdividen a su vez en varios indicadores; estas se pueden medir por métrica interna o externa.



**Figura 2.6.** Evaluación Interna, externa y Calidad de Uso ISO/IEC 9126

**Fuente:** (ISO/IEC, 2007)

Las definiciones se dan para cada característica y sub-característica de calidad del software que influye en la calidad. Para cada característica y subcaracterística, la capacidad del software es determinada por un conjunto de atributos internos que pueden ser medidos. Las características y subcaracterísticas se pueden medir externamente por la capacidad del sistema que contiene el software.

#### 2.4.1.1. Funcionalidad

Funcionalidad es la capacidad del software de cumplir y proveer las funciones para satisfacer las necesidades explícitas e implícitas cuando es utilizado en condiciones específicas. A continuación, se muestra la característica de Funcionalidad y las sub-características que cubre:





## **Figura 2.7.** Característica de funcionalidad

**Fuente:** (ISO/IEC, 2007)

La funcionalidad se divide en 5 criterios:

**Adecuación:** La capacidad del software para proveer un adecuado conjunto de funciones que cumplan las tareas y objetivos especificados por el usuario.

**Exactitud:** La capacidad del software para hacer procesos y entregar los resultados solicitados con precisión o de forma esperada.

**Interoperabilidad:** La capacidad del software de interactuar con uno o más sistemas específicos.

**Seguridad:** La capacidad del software para proteger la información y los datos de manera que los usuarios o los sistemas no autorizados no puedan acceder a ellos para realizar operaciones, y la capacidad de aceptar el acceso a los datos de los usuarios o sistemas autorizados

**Conformidad de la funcionalidad:** La capacidad del software de cumplir los estándares referentes a la funcionalidad.

### **2.4.1.2. Confiabilidad**

La confiabilidad es la capacidad del software para asegurar un nivel de funcionamiento adecuado cuando es utilizando en condiciones específicas. En este caso la confiabilidad se amplía sostener un nivel especificado de funcionamiento y no una función requerida.



**Figura 2.8.** Característica de Confiabilidad

**Fuente:** (ISO/IEC, 2007)

La confiabilidad se divide en 4 criterios:

**Madurez:** La capacidad que tiene el software para evitar fallas cuando encuentra errores. Ejemplo, la forma como el software advierte al usuario cuando realiza operaciones en la unidad de diskett vacía, o cuando no encuentra espacio suficiente en el disco duro donde está almacenando los datos.

**Tolerancia a errores:** La capacidad que tiene el software para mantener un nivel de funcionamiento en caso de errores.

**Recuperabilidad:** La capacidad que tiene el software para restablecer su funcionamiento adecuado y recuperar los datos afectados en el caso de una falla.

**Conformidad de la fiabilidad:** La capacidad del software de cumplir a los estándares o normas relacionadas a la fiabilidad.

#### **2.4.1.3. Usabilidad**

La usabilidad es la capacidad del software de ser entendido, aprendido, y usado en forma fácil y atractiva. Algunos criterios de funcionalidad, fiabilidad y eficiencia afectan la usabilidad, pero para los propósitos de la ISO/IEC 9126 ellos no clasifican como usabilidad. La usabilidad está determinada por los usuarios finales y los usuarios indirectos del software, dirigidos a todos los ambientes, a la preparación del uso y el resultado obtenido.



**Figura 2.9.** Característica de Usabilidad

**Fuente:** (ISO/IEC, 2007)

La usabilidad se divide en 5 criterios:

**Entendimiento:** La capacidad que tiene el software para permitir al usuario entender si es adecuado, y de una manera fácil como ser utilizado para las tareas y las condiciones particulares de la aplicación. En este criterio se debe tener en cuenta la documentación y de las ayudas que el software entrega.

**Aprendizaje:** La forma como el software permite al usuario aprender su uso. También es importante considerar la documentación.

**Operabilidad:** La manera como el software permite al usuario operarlo y controlarlo.

**Atracción:** La presentación del software debe ser atractiva al usuario. Esto se refiere a las cualidades del software para hacer más agradable al usuario, ejemplo, el diseño gráfico.

**Conformidad de uso:** La capacidad del software de cumplir los estándares o normas relacionadas a su usabilidad.

#### 2.4.1.4. Eficiencia

La eficiencia del software es la forma del desempeño adecuado, de acuerdo a al número recursos utilizados según las condiciones planteadas. Se debe tener en cuenta otros aspectos como la configuración de hardware, el sistema operativo, entre otros.



**Figura 2.10.** Característica de Eficiencia

**Fuente:** (ISO/IEC, 2007)

La eficiencia se divide en 3 criterios:

**Comportamiento de tiempos:** Los tiempos adecuados de respuesta y procesamiento, el rendimiento cuando realiza su función en condiciones específicas. Ejemplo, ejecutar el procedimiento más complejo del software y esperar su tiempo de respuesta, realizar la misma función, pero con más cantidad de registros.

**Utilización de recursos:** La capacidad del software para utilizar cantidades y tipos adecuados de recursos cuando este funciona bajo requerimientos o condiciones establecidas. Ejemplo, los recursos humanos, el hardware, dispositivos externos.

**Conformidad de eficiencia:** La capacidad que tiene el software para cumplir con los estándares o convenciones relacionados a la eficiencia.

### 2.4.1.5. Mantenibilidad

La capacidad de mantenimiento es la cualidad que tiene el software para ser modificado. Incluyendo correcciones o mejoras del software, a cambios en el entorno, y especificaciones de requerimientos funcionales.



**Figura 2.11.** Característica de Mantenimiento

**Fuente:** (ISO/IEC, 2007)

El mantenimiento se divide en 5 criterios:

**Capacidad de ser analizado:** La forma como el software permite diagnósticos de deficiencias o causas de fallas, o la identificación de partes modificadas.

**Cambiabilidad:** La capacidad del software para que la implementación de una modificación se pueda realizar, incluye también codificación, diseño y documentación de cambios.

**Estabilidad:** La forma como el software evita efectos inesperados para modificaciones del mismo.

**Facilidad de prueba:** La forma como el software permite realizar pruebas a las modificaciones sin poner el riesgo los datos.

**Conformidad de facilidad de mantenimiento:** La capacidad que tiene el software para cumplir con los estándares de facilidad de mantenimiento.

#### 2.4.1.6. Portabilidad

La capacidad que tiene el software para ser trasladado de un entorno a otro.



**Figura 2.12.** Característica de portabilidad

**Fuente:** (ISO/IEC, 2007)

La usabilidad se divide en 5 criterios:

**Adaptabilidad:** Es como el software se adapta a diferentes entornos especificados (hardware o sistemas operativos) sin que implique reacciones negativas ante el cambio. Incluye la escalabilidad de capacidad interna (Ejemplo: Campos en pantalla, tablas, volúmenes de transacciones, formatos de reporte, etc.).

**Facilidad de instalación:** La facilidad del software para ser instalado en un entorno específico o por el usuario final.

**Coexistencia:** La capacidad que tiene el software para coexistir con otro o varios software, la forma de compartir recursos comunes con otro software o dispositivo.

**Reemplazabilidad:** La capacidad que tiene el software para ser reemplazado por otro software del mismo tipo, y para el mismo objetivo. Ejemplo, la reemplazabilidad de una nueva versión es importante para el usuario, la propiedad de poder migrar los datos a otro software de diferente proveedor.

**Conformidad de portabilidad:** La capacidad que tiene el software para cumplir con los estándares relacionados a la portabilidad.

## **2.5. Seguridad de Software**

### **2.5.1. ISO 17799**

En toda organización que haga uso de las tecnologías de información se recomienda implementar buenas prácticas de seguridad, pues en muchas ocasiones el no seguir un proceso de implementación adecuado como el que establece el ISO 17799 puede generar huecos por la misma complejidad de las organizaciones, en ese sentido, aumenta la posibilidad de riesgos en la información.

Este estándar internacional de alto nivel para la administración de la seguridad de la información, fue publicado por la ISO en diciembre de 2000 con el objeto de desarrollar un marco de seguridad sobre el cual trabajen las organizaciones.

El ISO 17799, al definirse como una guía en la implementación del sistema de administración de la seguridad de la información, se orienta a preservar los siguientes principios de la seguridad informática:

**Confidencialidad.** Asegurar que únicamente personal autorizado tenga acceso a la información.

**Integridad.** Garantizar que la información no será alterada, eliminada o destruida por entidades no autorizadas.

**Disponibilidad.** Asegurar que los usuarios autorizados tendrán acceso a la información cuando la requieran.

## **2.6. Métodos de Estimación de Costos de Software**

“El modelo de estimación es la predicción más optimista con una probabilidad distinta de cero de ser cierta. Es una predicción que tiene la misma probabilidad de estar por encima o por debajo del valor actual que se encuentra en la jerarquía de modelos de estimación de software con el nombre de COCOMO, por Constructive

Cost Model (Modelo Constructivo de Coste)". (Métodos de estimación de costos, 2018)

- Los objetivos de la estimación de proyectos son reducir los costes e incrementar los niveles de servicio y de calidad.
- Midiendo determinados aspectos del proceso de software se puede tener una visión de alto nivel de lo que sucederá durante el desarrollo.
  - Las mediciones de procesos anteriores permiten realizar predicciones sobre los actuales.
  - Las mediciones de atributos de proceso en fases iniciales del desarrollo permiten realizar predicciones sobre fases posteriores.
  
- Las predicciones de proceso conducen la toma de decisiones antes del comienzo del desarrollo, durante el proceso de desarrollo, durante la transición del producto al cliente y a lo largo de la fase de mantenimiento.

### 2.6.1. Modelos de Estimación

En la estimación del tamaño de Software COCOMO II utiliza técnicas:

- **Líneas de Códigos Fuente** El objetivo es medir la cantidad de trabajo intelectual puesto en el desarrollo de un programa.  
Definir una línea de código es difícil debido a que existen diferencias conceptuales cuando se cuentan sentencias ejecutables y de declaraciones de datos en lenguajes diferentes.  
A los efectos de COCOMO II, se eliminan las categorías de software que consumen poco esfuerzo. Así no están incluidas librerías de soporte, sistemas operativos, librerías comerciales, ni tampoco el código generado con generadores de código fuente.
- **Conversión de Puntos Función a Líneas de Código Fuente** Para determinar el esfuerzo nominal en el modelo COCOMO II los puntos función



no ajustados tienen que ser convertidos a líneas de código fuente considerando el lenguaje de implementación.” (LuisMi Gracia, 2012, p.19)

### 2.6.2. Método de Estimación de Costo COCOMO II

“Modelo COCOMO II, modelo de estimación que se encuentra en la jerarquía de modelos de estimación de software con el nombre de COCOMO, por Constructive Cost Model (Modelo Constructivo de Coste). El modelo COCOMO original se ha convertido en uno de los modelos de estimación de coste del software más utilizados” (COCOMO II, 2018)

El Modelo Constructivo de Costos (o COCOMO, por su acrónimo del inglés Constructive Cost Model) es un modelo matemático de base empírica utilizado para estimación de costos de software. Incluye tres submodelos, cada uno ofrece un nivel de detalle y aproximación, cada vez mayor, a medida que avanza el proceso de desarrollo del software: básico, intermedio y detallado.

A la vez, cada submodelo también se divide en modos que representan el tipo de proyecto, y puede ser:

- **modo orgánico:** un pequeño grupo de programadores experimentados desarrollan software en un entorno familiar. El tamaño del software varía desde unos pocos miles de líneas (tamaño pequeño) a unas decenas de miles (medio).
- **modo semilibre o semiacoplado:** corresponde a un esquema intermedio entre el orgánico y el rígido; el grupo de desarrollo puede incluir una mezcla de personas experimentadas y no experimentadas.
- **modo rígido o empotrado:** el proyecto tiene fuertes restricciones, que pueden estar relacionadas con la funcionalidad y/o pueden ser técnicas. El problema a resolver es único y es difícil basarse en la experiencia, puesto que puede no haberla.

**Tabla 2.1.** Modelo Básico del Método COCOMO II

<b>MODO</b>	<b>A</b>	<b>b</b>	<b>c</b>	<b>d</b>
<b>Orgánico</b>	2.40	1.05	2.50	0.38
<b>Semi – orgánico</b>	3.00	1.12	2.50	0.35
<b>Empotrado</b>	3.60	1.20	2.50	0.33

**Fuente:** (COCOMO, 2018)

### **Características**

Pertenece a la categoría de modelos estimadores basados en estimaciones matemáticas. Está orientado a la magnitud del producto final, midiendo el "tamaño" del proyecto, en función de la cantidad de líneas de código, principalmente.

### **Modelos de Estimación de Costos**

Existen diferentes modelos que define COCOMO:

- **Modelo básico:** Se basa exclusivamente en el tamaño expresado en LDC y se utiliza para obtener una primera aproximación rápida del esfuerzo.
- **Modelo intermedio:** Este añade al modelo básico quince modificadores opcionales para tener en cuenta en el entorno de trabajo, incrementando así la precisión de la estimación además del tamaño del programa incluye un conjunto de medidas subjetivas llamadas conductores de costes.
- **Modelo avanzado o detallado:** Incluye todo lo del modelo intermedio además del impacto de cada conductor de coste en las distintas fases de desarrollo. Para nuestro caso el modelo intermedio será el que usaremos, dado que realiza las estimaciones con bastante precisión.

Presenta principalmente dos mejoras respecto a las anteriores:

- Los factores correspondientes a los atributos son sensibles o dependientes de la fase sobre la que se realizan las estimaciones. Aspectos tales como

la experiencia en la aplicación, utilización de herramientas de software, etc., tienen mayor influencia en unas fases que en otras, y además van variando de una etapa a otra.

- Establece una jerarquía de tres niveles de productos, de forma que los aspectos que representan gran variación a bajo nivel, se consideran a nivel módulo, los que representan pocas variaciones, a nivel de subsistema; y los restantes son considerados a nivel sistema.

Para realización del COCOMO previamente necesitaremos conocer el número de líneas de código, posteriormente para poder realizar los cálculos del método de estimación usaremos las siguientes ecuaciones:

**Tabla 2. 2.** Ecuaciones del Método COCOMO II

<b>Variable</b>	<b>Educación</b>	<b>Tipo de Unidad</b>
<b>Esfuerzo requerido por el proyecto</b>	$E = a * (KLDC)^b * FAE$	Personas por mes
<b>Tiempo Requerido por el proyecto</b>	$T = c * (E)^d$	Meses
<b>Número de personas requeridas para el proyecto</b>	$P = \frac{E}{T}$	Personas
<b>Costo Total</b>	$CT = sueldo\ Mes * P * T$	\$us

**Fuente:** (Prentice – Hall, 2005)

**Dónde:**

- E es el esfuerzo requerido por el proyecto, en persona-mes
- T es el tiempo requerido por el proyecto, en meses
- P es el número de personas requerido por el proyecto
- a, b, c y d son constantes con valores definidos en una tabla, según cada submodelo
- KLDC es la cantidad de líneas de código.

- FAE es un multiplicador que depende de 15 atributos.

Por otro lado, también debemos de hallar la variable FAE, la cual se obtiene mediante la multiplicación de los valores evaluados en los diferentes 15 conductores de coste que se observan en la siguiente **Figura 2.13**.

Atributos	Valor					
	Muy bajo	Bajo	Nominal	Alto	Muy alto	Extra alto
<b>Atributos de software</b>						
Fiabilidad	0,75	0,88	1,00	1,15	1,40	
Tamaño de Base de datos		0,94	1,00	1,08	1,16	
Complejidad	0,70	0,85	1,00	1,15	1,30	1,65
<b>Atributos de hardware</b>						
Restricciones de tiempo de ejecución			1,00	1,11	1,30	1,66
Restricciones de memoria virtual			1,00	1,06	1,21	1,56
Volatilidad de la máquina virtual		0,87	1,00	1,15	1,30	
Tiempo de respuesta		0,87	1,00	1,07	1,15	
<b>Atributos de personal</b>						
Capacidad de análisis	1,46	1,19	1,00	0,86	0,71	
Experiencia en la aplicación	1,29	1,13	1,00	0,91	0,82	
Calidad de los programadores	1,42	1,17	1,00	0,86	0,70	
Experiencia en la máquina virtual	1,21	1,10	1,00	0,90		
Experiencia en el lenguaje	1,14	1,07	1,00	0,95		
<b>Atributos del proyecto</b>						
Técnicas actualizadas de programación	1,24	1,10	1,00	0,91	0,82	
Utilización de herramientas de software	1,24	1,10	1,00	0,91	0,83	
Restricciones de tiempo de desarrollo	1,22	1,08	1,00	1,04	1,10	

**Figura 2.13.** Atributos del Método COCOMO II

**Fuente:** (Roger S.Pressman, 2010)

### Atributos de los Costes

Cada atributo se cuantifica para un entorno de proyecto. La escala es muy bajo - bajo - nominal - alto - muy alto - extremadamente alto. Dependiendo de la calificación de cada atributo, se asigna un valor para usar de multiplicador en la fórmula (por ejemplo, si para un proyecto el atributo DATA es calificado como muy alto, el resultado de la fórmula debe ser multiplicado por 1000).

El significado de los atributos es el siguiente, según su tipo:

### **Atributos de software**

- **RELY:** garantía de funcionamiento requerida al software. Indica las posibles consecuencias para el usuario en el caso que existan defectos en el producto. Va desde la sola inconveniencia de corregir un fallo (muy bajo) hasta la posible pérdida de vidas humanas (extremadamente alto, software de alta criticidad).
- **DATA:** tamaño de la base de datos en relación con el tamaño del programa. El valor del modificador se define por la relación:  $D/K$ , donde D corresponde al tamaño de la base de datos en bytes y K es el tamaño del programa en cantidad de líneas de código.
- **CPLX:** representa la complejidad del producto.

### **Atributos de hardware**

- **TIME:** limitaciones en el porcentaje del uso de la CPU.
- **STOR:** limitaciones en el porcentaje del uso de la memoria.
- **VIRT:** volatilidad de la máquina virtual.
- **TURN:** tiempo de respuesta requerido.

### **Atributos del personal**

- **ACAP:** calificación de los analistas.
- **AEXP:** experiencia del personal en aplicaciones similares.
- **PCAP:** calificación de los programadores.
- **VEXP:** experiencia del personal en la máquina virtual.
- **LEXP:** experiencia en el lenguaje de programación a usar.

### **Atributos del proyecto**

- **MODP:** uso de prácticas modernas de programación.
- **TOOL:** uso de herramientas de desarrollo de software.
- **SCED:** limitaciones en el cumplimiento de la planificación.

## CAPITULO III

### 3. MARCO APLICATIVO.

El desarrollo del Tutor Inteligente Móvil para la enseñanza de la Estática y Dinámica es un sistema de software educativo, una alternativa para el proceso aprendizaje de nuevos conocimientos, que interactúa con los estudiantes de quinto de secundaria de la Unidad Educativa “Dionicio Morales” de forma dinámica y activa. La mejora en el aprendizaje se refleja en el desarrollo de un nuevo conocimiento y la comprensión de los contenidos, ya que el estudiante puede acceder en momento oportuno y necesario.

En este capítulo realizaremos el análisis y diseño del prototipo de Tutor inteligente Móvil, utilizando las cinco etapas de la Metodología Mobile-D orientada a la creación de aplicaciones móviles, combinándolas con las etapas de la Metodología de Ingeniería de Software Educativo MEISE propuesta por Maria Antonieta Abud.en 2009.

**Tabla 3.1.** Metodología Mobile-D combinada con ISE.

FASES	METODOLOGÍA MOBILE-D	METODOLOGÍA ISE
FASE 1	Exploración e Inicialización (Mobile-D)	Análisis
FASE 2		Diseño
FASE 3	Productización (Mobile-D)	Desarrollo
FASE 4	Estabilización y Prueba (Mobile-D)	Prueba Piloto
FASE 5		Prueba de Campo

**Fuente:** (Elaboración propia).

### **3.1. Fases de la Metodología Mobile-D**

#### **3.1.1. Fase de Análisis**

En esta fase se realizará una investigación de los estudiantes de quinto de secundaria de la Unidad Educativa “Dionicio Morales”, para la obtención de los requerimientos de software que debe cumplir el prototipo.

##### **3.1.1.1. Requisitos de Software**

###### **3.1.1.1.1. Reconocimiento del problema.**

Al iniciar la recolección de la información en la Unidad Educativa se han identificado características, necesidades y problemas por los que atraviesan los estudiantes en su formación escolar. Entre estos problemas que se han identificado y se pretende solucionar son:

- El proceso de enseñanza es monótono, por lo que causa desinterés en los estudiantes por aprender los contenidos de estática y dinámica.
- El tiempo limitado en el desarrollo curricular, ocasiona que los estudiantes se quedan con las dudas que no son aclaradas.
- La asimilación de conocimientos insuficiente en estática y dinámica, provoca desmotivación por seguir aprendiendo.
- Los docentes no cuentan con un medio de apoyo de programa informático para impartir sus clases.
- Escaso uso de equipos computacionales para un buen aprendizaje, genera el desperdicio de recursos tecnológicos.

###### **3.1.1.1.2. Evaluación y síntesis**

Con la información recabada en la fase anterior se tomarán los aspectos más relevantes para el desarrollo del prototipo.

- Necesidad de implementar una herramienta tecnológica móvil que sea didáctica y pedagógica que despierte el interés del estudiante.

- Esta herramienta debe coadyuvar al aprendizaje de los temas propuestos para dicha enseñanza.
- El desarrollo de esta herramienta tecnológica debe estar enfocado al aprendizaje de temas de Estática y Dinámica.

### 3.1.1.1.3. Modelado del Tutor Inteligente.

En la figura 3.3, tenemos el modelado que se puede extraer del punto anterior, según los requisitos del software.



**Figura 3.1.** Modelado según los requisitos

**Fuente:** Elaboración Propia

### 3.1.1.1.4. Especificaciones

La plataforma adecuada para el desarrollo de la aplicación móvil es Android Studio, ya que nos permite desarrollar un producto novedoso y didáctico, además a la aplicación podrá accederse desde un dispositivo móvil, siempre y cuando sean de Sistema Operativo Android.

### 3.1.1.1.5. Análisis del contexto

El Tutor Inteligente Móvil busca atender las diferentes necesidades de proceso de enseñanza aprendizaje de los contenidos de Estática y Dinámica.

#### a) Características de la Población Objetivo



En esta etapa inicialmente se establece la población a la cual se pretende llegar con desarrollo del prototipo Tutor Inteligente Móvil para la enseñanza de la Estática y Dinámica, se delimita de la siguiente manera:

- Realizaran uso los estudiantes que cursan el nivel quinto de secundaria.
- También pueden utilizar personas con conocimiento de estos temas y estudiantes que estén en la etapa de transición a la Educación Superior.

#### **b) Conducta de entrada**

Para los datos de entrada se pudo obtener de siguientes medios:

- Información acerca de Estática y Dinámica obtenida de libros de Física de Quinto de Secundaria y libros Universitarios que abarcan el tema.
- Datos extraídos de la web, textos educativos, herramientas multimedia que puedan ayudar a una mejor comprensión.

#### **c) Problemas o Necesidad a entender.**

Se definen los siguientes problemas que deben ser considerados a la hora de realizar el desarrollo del prototipo del tutor inteligente móvil para la enseñanza de Estática y Dinámica.

- En la actualidad los estudiantes no cuentan con herramientas tecnológicas que les permita realizar la retroalimentación en estos temas en específico.
- La falta de información necesaria o poco conocimiento sobre los temas de estudio.

La necesidad que se busca atender es mejorar el proceso de enseñanza – aprendizaje de los temas de Estática y Dinámica que permita promover el estudio de estos temas y así tener un aprendizaje significativo.

### 3.1.1.2. Principios pedagógicos y didácticos aplicables

En la institución educativa una de las necesidades importantes es la de generar aprendizajes significados, la cual despierte el interés del estudiante por el aprendizaje de los temas. El Tutor Inteligente Móvil es un software educativo orientado a la enseñanza de contenidos como la Estática y Dinámica, con una información concreta y necesaria respecto a los contenidos, donde el estudiante asimilara los contenidos a medida que vaya utilizando la aplicación.

El uso de la tecnología en la educación es de mucha importancia, porque la mayoría de los estudiantes cuenta con un dispositivo móvil con recursos potenciales que no son aprovechadas de forma eficiente. La implementación de medios didácticos en el proceso de enseñanza aprendizaje, mejorará el aprendizaje de contenidos, así mismo será un medio de apoyo para el docente. El Tutor Inteligente Móvil será una herramienta tecnológica informática, que presentara una interfaz amigable, didáctica y simple, además es un prototipo móvil accesible por su portabilidad.

### 3.1.1.3. Requerimientos Funcionales y no Funcionales

Para los requerimientos funcionales está basado en la arquitectura de los sistemas tutores inteligentes STI, que esta arquitectura cuenta con los siguientes módulos, como se puede observar en la Tabla 3.1.

**Tabla 3.2.** Requerimientos Funcionales y no funcionales

REQUERIMIENTOS	DESCRIPCIÓN
<b>FUNCIONALES</b>	<ul style="list-style-type: none"><li>✓ <b>El módulo tutor</b>, contiene información relevante de las definiciones, conceptos, formulas y resolución de ejercicios, con un diseño didáctico de contenidos de la estática y dinámica.</li><li>✓ <b>El módulo estudiante</b>, almacena la información cognitiva del estudiante sobre cuanto conoce las definiciones, resolución de ejercicios de los temas.</li></ul>

- ✓ **El módulo interface**, permitirá captar en interés del estudiante, en el desarrollo de adquisición de conocimientos, conformada por una interfaz amigable, dinámico y didáctico.
- ✓ **El Módulo Evaluación**, realizara el seguimiento en el proceso de evaluación al estudiante, mediante un conteo de la cantidad de respuestas correctas e incorrectas marcadas por el estudiante.

---

**NO FUNCIONALES** El sistema debe poder ejecutarse en cualquier dispositivo que cuente con el sistema operativo Android, a partir de la version 4.0 en adelante.

---

**Fuente:** (Elaboración Propia)

#### 3.1.1.4. Planificación

**Planificación de Actividades:** Se establece una planificación de actividades para el desarrollo del prototipo Tutor Inteligente Móvil para la enseñanza de Estática y Dinámica detallando en la siguiente tabla.

**Tabla 3.3.** Planificación de Actividades.

NUMERO DE ACTIVIDAD	NOMBRE DE LA ACTIVIDAD	DESCRIPCIÓN	TAREA
1	Administrador de datos	Ingreso y registro de Usuarios	<ul style="list-style-type: none"> <li>▪ Diseñar la estructura de almacenamiento de datos al usuario.</li> <li>▪ Diseñar la interfaz para el ingreso y registro de usuario.</li> </ul>

2	Modulo Tutor	Contenido Temático	<ul style="list-style-type: none"> <li>▪ Diseñar la estructura del contenido temático que será visualizado.</li> <li>▪ Diseñar la interfaz para la presentación del contenido temático.</li> </ul>
3	Modulo estudiante	Estudio y Evaluación del contenido temático	<ul style="list-style-type: none"> <li>▪ Diseñar la estructura de los temas seleccionados a estudiar y su posterior evaluación.</li> <li>▪ Diseñar la interfaz para la visualización de los resultados de la evaluación.</li> </ul>

**Fuente:** (Elaboración Propia)

### 3.1.2. Fase de Diseño

En función a la información obtenida en la fase de análisis se empieza el desarrollo del Tutor Inteligente Móvil concentrándose en el estudio de estudiantes de quinto de secundaria de acuerdo al siguiente detalle:

**Tabla 3.4.** Datos de entorno para el desarrollo

DATOS	CARACTERÍSTICAS
<b>Destinatarios</b>	El Tutor Inteligente Móvil está orientada a estudiantes que cursan quinto de secundaria.
<b>Área de contenido</b>	<p>La información con la que cuenta el Tutor inteligente Móvil son las siguientes:</p> <ul style="list-style-type: none"> <li>▪ Definiciones y conceptos de temas.</li> <li>▪ Gráficos.</li> <li>▪ Ecuaciones y Formulas.</li> <li>▪ Evaluaciones.</li> <li>▪ Resultados Finales.</li> </ul>

<b>Necesidad educativa</b>	<ul style="list-style-type: none"> <li>▪ La complejidad de los temas.</li> <li>▪ La cantidad de información en cada tema.</li> </ul>
<b>Limitaciones</b>	El contenido de avance solo debe abarcar a promover la Estática y Dinámica para quinto de secundaria.
<b>Recursos para los Usuarios, hardware y Software.</b>	<p><b>Hardware:</b></p> <ul style="list-style-type: none"> <li>▪ Un dispositivo Móvil (celulares, Smartphone, Tablets) con SO Android.</li> </ul> <p><b>Software:</b></p> <ul style="list-style-type: none"> <li>▪ SO Android 4.0 o superior instalado en el equipo.</li> <li>▪ ADT de Android con librerías de Android Studio.</li> <li>▪ Emulador de Java Android.</li> </ul>

**Fuente:** (Elaboración Propia)

### 3.1.2.1. Tipos de Diseño

A continuación, detallaremos 3 tipos de diseño que son: Diseño Educativo, Diseño Comunicacional y Diseño Computacional.

#### a) Diseño Educativo

El uso frecuente del prototipo de TimE&D, mejorará la enseñanza de contenidos educativos, del mismo modo se logrará mejorar la asimilación de contenidos, identificando y familiarizándose con las definiciones, leyes y formulas, aprendiendo así de manera más didáctica. Se realizará un diseño con imágenes la cual hará que la enseñanza sea más didáctica.

#### b) Diseño Comunicacional.

La comunicación entre el usuario y el prototipo está representada por el bosquejo de interfaces, es por eso que deberá ser de forma didáctica y sencilla, amigable y atrayente para captar el interés de los estudiantes.

#### c) Diseño Computacional.

Con la información obtenida en las anteriores fases y en base a las necesidades que deben ser atendidas se establece las funcionalidades que deberá tener el prototipo.

**Tabla 3.5.** Funciones del Prototipo

<b>APOYO DOCENTE</b>	<b>AL</b>	<ul style="list-style-type: none"> <li>▪ TimE&amp;D, facilitara la labor del maestro, puesto que en el módulo del tutor se encuentran almacenados los temas a las que el estudiante podrá acceder en cualquier momento mediante un dispositivo móvil Android.</li> <li>▪ El maestro a través de las evaluaciones en el módulo de estudiante podrá verificar el rendimiento escolar del estudiante.</li> </ul>
<b>APOYO ESTUDIANTE</b>	<b>AL</b>	<ul style="list-style-type: none"> <li>▪ TimE&amp;D facilita al estudiante el contenido de las lecciones, así este puede repasar las lecciones según lo vea conveniente y necesario.</li> <li>▪ TimE&amp;D atrae el interés del estudiante por lo didáctico y educativo que es el prototipo.</li> <li>▪ Las lecciones al ser repasadas constantemente por el estudiante, provocan la retención de información.</li> </ul>

**Fuente:** (Elaboración Propia)

### 3.1.3. Fase Desarrollo

#### 3.1.3.1. Modelo de Negocio.

##### a) Identificación de actores y escenarios

Se identifican a los actores y sus roles, que interactúan con el prototipo educativo móvil.

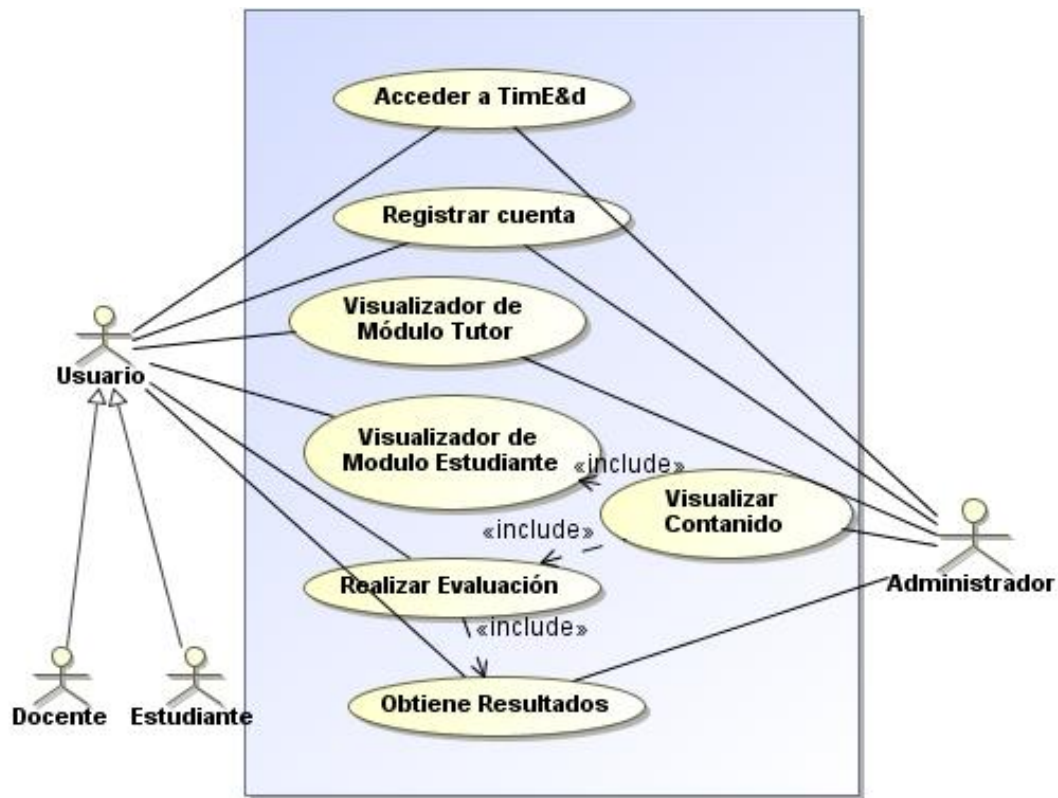
**Tabla 3.6.** Actores y Roles

<b>ACTORES</b>	<b>ROLES</b>
<b>Docente</b>	Accede al contenido del módulo tutor y módulo estudiante

	(lecciones, resultados de evaluaciones de cada estudiante)
<b>Estudiante</b>	Accede a los módulo turo y módulo estudiante.
<b>Usuario</b>	Este actor debe tener conocimientos básicos de los compuestos binarios, de esta manera poder acceder e interpretar el contenido del módulo tutor y módulo estudiante.

**Fuente:** (Elaboración Propia)

**b) Diagramas de caso de uso**

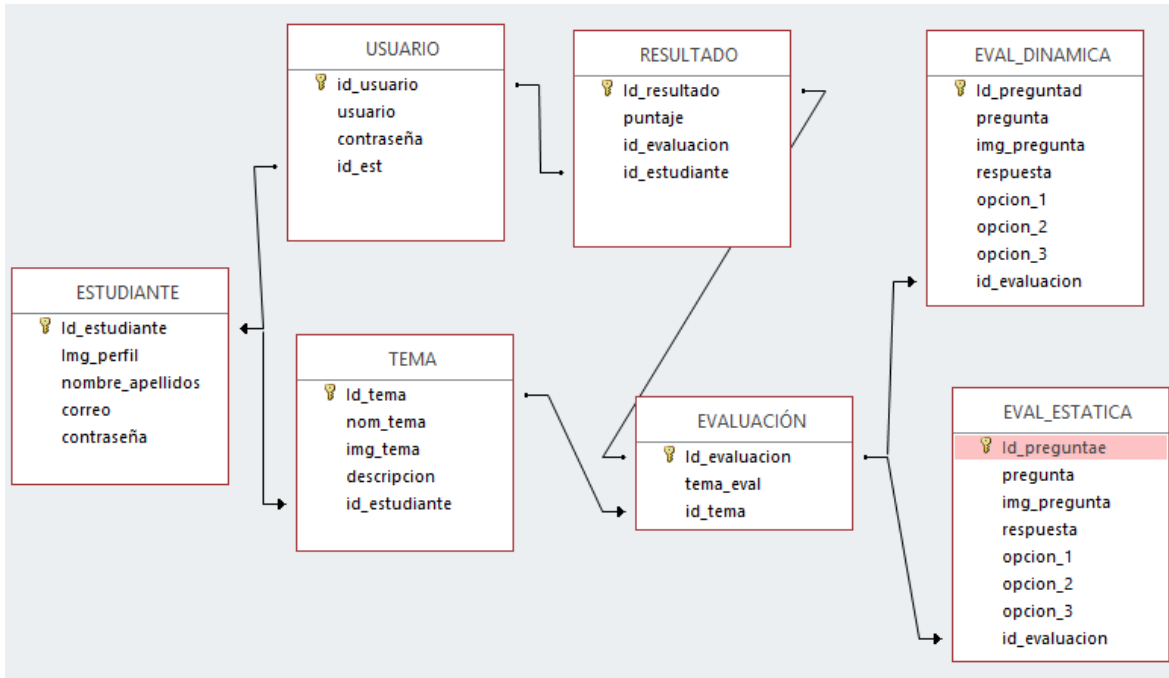


**Figura 3.2.** Diagrama de casos de uso.

**Fuente:** (Elaboración Propia)

**c) Diagrama del modelo de datos**

Muestra la representación gráfica de la base de datos del Tutor Inteligente Móvil.

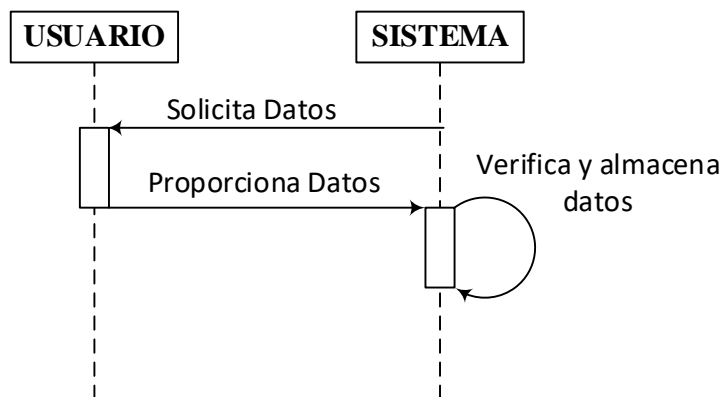


**Figura 3.3.** Base de datos del tutor Inteligente Móvil.

**Fuente:** (Elaboración propia)

**d) Diagrama de secuencias**

- **Registro de estudiantes.** A continuación, ilustraremos el diagrama de secuencias del registro de estudiantes.

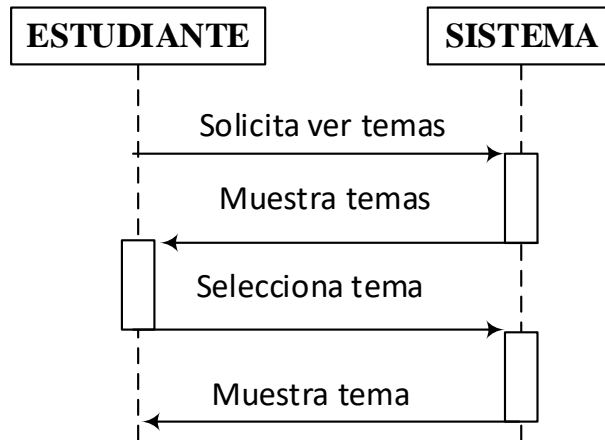


**Figura 3.4.** Diagrama de secuencias de registro de Usuarios

**Fuente:** (Elaboración Propia)

- **VISUALIZAR TEMAS.** El diagrama de secuencias de la visualización de temas se muestra en la siguiente figura.

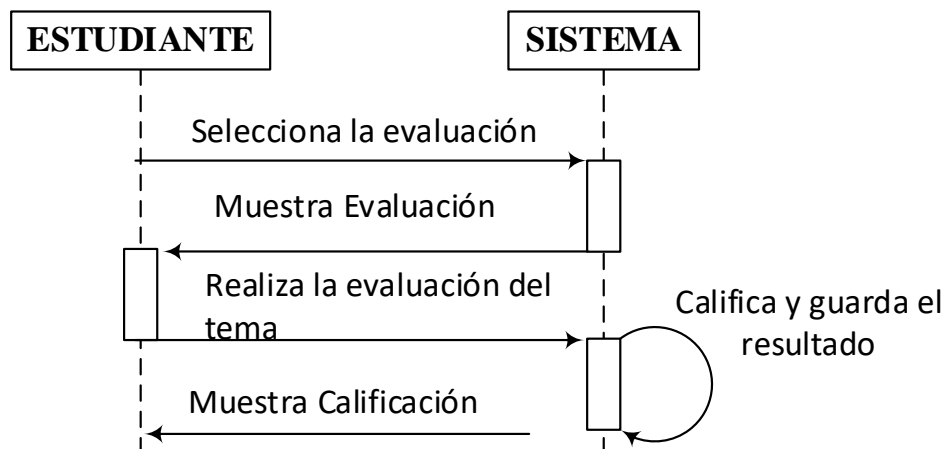




**Figura 3.5.** Diagrama de secuencias de Visualizador de temas

**Fuente:** (Elaboración Propia)

- **EVALUACIÓN DEL ESTUDIANTE.** En la siguiente figura observaremos el diagrama de secuencia de la evaluación del estudiante.



**Figura 3. 6.** Diagrama de secuencias de Evaluación del Estudiante

**Fuente:** (Elaboración Propia)

### e) ESPECIFICACIONES DE CASO DE USO

**Tabla 3.7.** Especificaciones de caso de uso para la visualización del módulo tutor.

<b>Descripción:</b>	Se visualizara las interfaces relacionadas con la enseñanza al estudiante (Lecciones, Ejercicios y contenido complementario).
<b>Pre-condición:</b>	El estudiante de introducir su usuario.
<b>Post-condiciones:</b>	
<b>Actores:</b>	Estudiantes y docente.
<b>Episodios:</b>	<ol style="list-style-type: none"> <li>1. Ingresa al prototipo.</li> <li>2. Introduce usuario.</li> <li>3. Selecciona la opción de lecciones</li> <li>4. Comienza las lecciones.</li> </ol>

**Fuente:** (Elaboración propia)

**Tabla 3.8.** Especificaciones de caso de uso para la visualización del módulo estudiante.

<b>Descripción:</b>	Se podrá realizar las evaluaciones adaptativas y dependiendo del resultado se presentara el contenido complementario.
<b>Pre-condición:</b>	El estudiante de introducir su usuario, y haber ingresado y desglosado las lecciones.
<b>Post-condiciones:</b>	
<b>Actores:</b>	Estudiantes y docente.
<b>Episodios:</b>	<ol style="list-style-type: none"> <li>1. Ingresa al prototipo.</li> <li>2. Introduce usuario.</li> <li>3. Se hace una introducción general de los temas.</li> <li>4. Comienza las lecciones.</li> <li>5. Selecciona la opción de evaluación.</li> <li>6. Visualiza el puntaje.</li> </ol>

**Fuente:** (Elaboración propia)

### 3.1.2.2. Iteraciones

#### a. Primera Iteración.

La primera pantalla que visualiza el administrador y usuario(estudiante) es la pantalla de inicio del prototipo TimE&D.

##### i. Registro de usuarios.

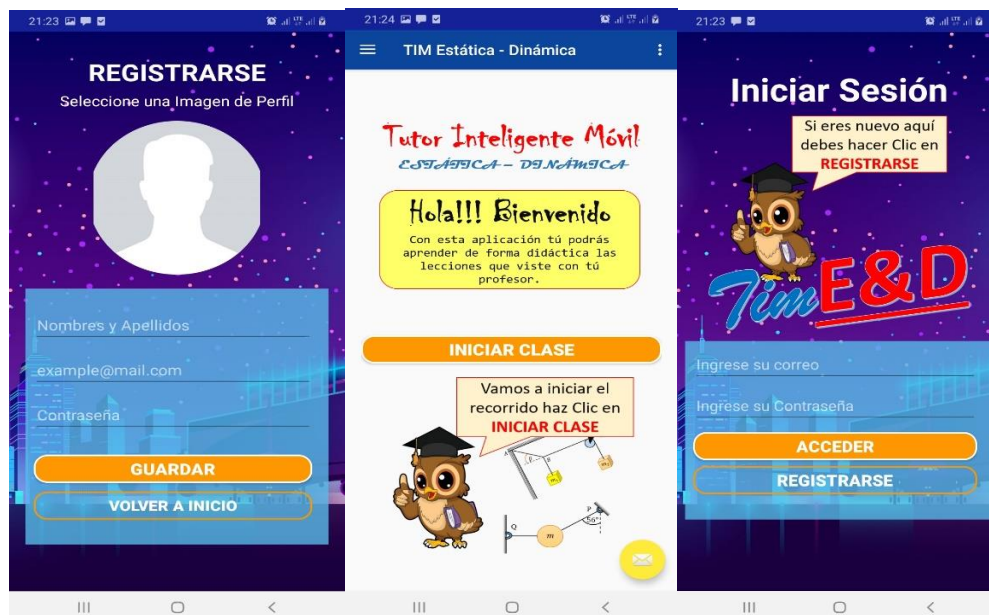
**TAREA 1:** Diseño de la estructura de almacenamiento de datos del Usuario.



**Figura 3.7.** Diagrama de clase para la estructura de registro de usuario

**Fuente:** (Elaboración propia)

**TAREA 2:** Diseño de la interfaz para el registro de usuario.



**Figura 3.8.** Interfaz de inicio y registro de Usuario.

**Fuente:** (Elaboración propia)

## b. Segunda iteración: modulo tutor

### I. Lecciones

Tarea 1: Diseñar la estructura del contenido a ser presentado.



Figura 3.9. Estructura de datos para el Contenido temático.

Fuente: (Elaboración propia)

TAREA 2: Diseño de interfaz para la presentación del contenido.

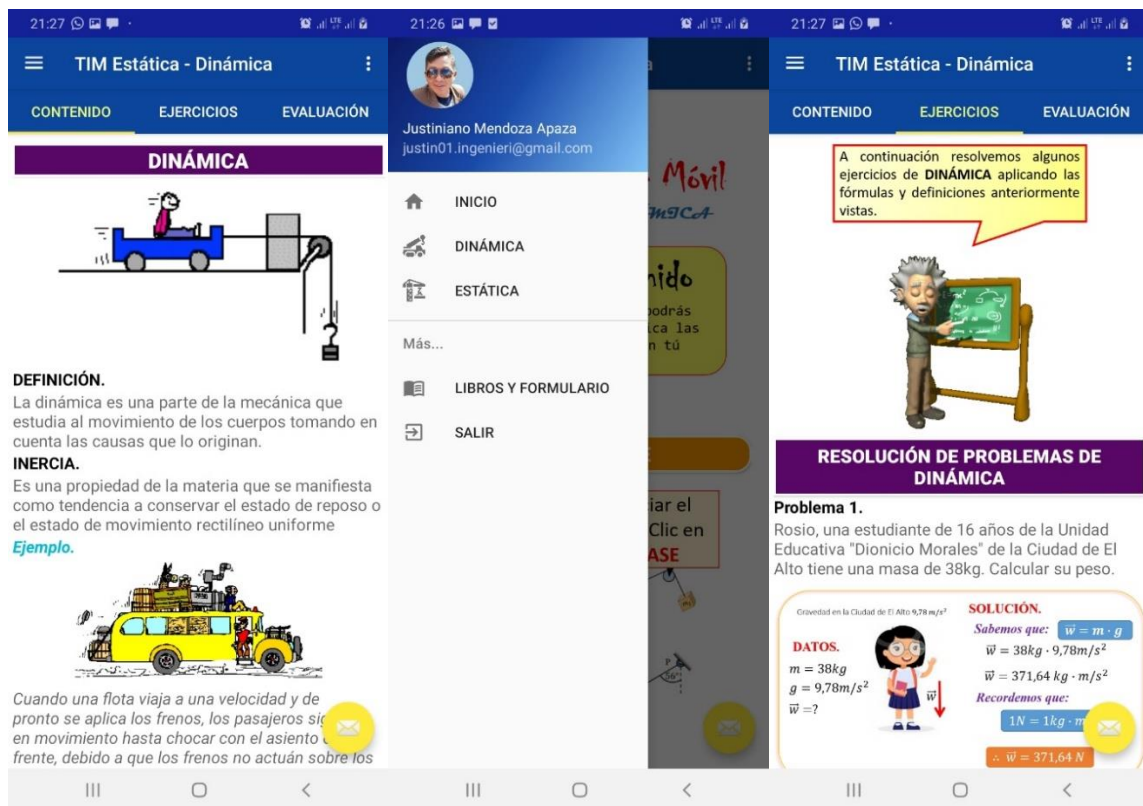


Figura 3.10. Interfaz de Contenido

Fuente: (Elaboración propia)

## ii. Evaluación

**TAREA 1:** Diseño de estructura de datos para realizar las evaluaciones.



**Figura 3.11.** Estructura de datos para la evaluación.

**Fuente:** (Elaboración propia)

**TAREA 2:** Diseño de interfaz para mostrar resultados.



**Figura 3.12.** Interfaz de Evaluaciones

**Fuente:** (Elaboración propia)

### 3.1.3. Fase de Prueba

#### 3.1.3.2. Prueba de campo

En esta etapa se realizarán las pruebas del sistema con la población real de estudiantes de quinto de secundaria de la Unidad Educativa “Dionicio Morales”, además el proyecto contó desde un principio con apoyo externo proveniente del docente de la especialidad de Física de dicha Unidad Educativa. Durante desarrollo de este proyecto, se llevaron a cabo reuniones frecuentes con el docente de la unidad educativa con el fin de ir mostrando prototipos sobre los cuales se determinaban posibles mejoras a elementos ya programados. Esta retroalimentación fue constante, y de esta manera se tuvo la certeza de que el objetivo general y los objetivos específicos se fueran cumpliendo paulatinamente.

#### 3.1.3.3. Muestra poblacional

Para llevar a cabo las pruebas de campo se realizaron con una muestra poblacional de un grupo conformado por estudiantes, considerando que estas son lo usuarios que interactuarán con la herramienta de software educativo propuesto.

Los resultados de las pruebas de campo con la población estudiantil se presentan en la siguiente tabla.

**Tabla 3. 9.** Pruebas de campo a la población estudiantil

<b>Nombre del Estudiante</b>	<b>Experiencia previa</b>	<b>Resultados</b>	<b>Comentarios</b>
Aduviri Layme Jhianet	Media	Bueno	- Le gusta el entorno del software - Le gusta el contenido de las lecciones
Añaya Poma Lizet	Si	Muy bueno	- Le agrada la resolución de ejercicios - Le gusta el entorno del software
Laime Mamani Carlos Daniel	Si	Muy bueno	- Le agrada el interfaz de las evaluaciones

					- Le agrada el contenido de las lecciones.
Blanco Alexander	Chuquimia	Si		Muy bueno	- Le gusta el interfaz de las evaluaciones - Le gusta el interfaz del software
Quispe Ronaldo	Prado	Si		Muy Bueno	- Le gusta el interfaz de las evaluaciones - Le llama la atención el formato de presentación de contenidos.
Laura Coronel	Jhonny	Si		Muy bueno	- Le agrada el interfaz del software - Le agrada el interfaz de la resolución de ejercicios
Coaquira Roger Ivan	Coronel	Media		Bueno	- Le gusta el entorno Grafico
Torrez Pillco	Jhenny	Si		Bueno	- Le agrada la presentación de las lecciones. - Le llama la atención la aplicación
Velasquez Wilmer Wilfredo	Aruquipa	Si		Muy Bueno	- Le llama la atención la aplicación - Le agrada el interfaz de la resolución de ejercicios
Huanca Javier	Chura Alan	Si		Muy Bueno	- Le llama la atención la aplicación - Le agrada el interfaz de la resolución de ejercicios
Kapa Daniel	Mamani Yordi	Media		Bueno	- Le agrada el interfaz de las evaluaciones. - Le gusta la presentación de los contenidos

Avircata Damaris Thalya	Chino	Media	Bueno	<ul style="list-style-type: none"> <li>- Le agrada el contenido de las lecciones.</li> <li>- Le gusta el entorno del software</li> </ul>	
Chura Marcos Joel	Paucara	Si	Muy Bueno	<ul style="list-style-type: none"> <li>- Le llama la atención la aplicación</li> <li>- Le gusta el entorno del software</li> </ul>	
Paño Leandro	Acho	José	Si	Muy Bueno	<ul style="list-style-type: none"> <li>- Le gusta el interfaz de la resolución de ejercicios</li> <li>- Le llama la atención el formato de presentación de contenidos</li> </ul>

**Fuente:** (Elaboración Propia)

Estas pruebas determinaron resultados favorables y positivos, teniendo una aceptación muy buena con respecto a la herramienta de software que se les presentó. Además, los resultados obtenidos de las pruebas a la población de estudiantes fueron tomados en cuenta para realizar las modificaciones a la interfaz o apariencia de la herramienta educativa en base a los comentarios vertidos que los estudiantes hicieron.

La muestra poblacional, es lo suficientemente valido para mostrar las tendencias de los estudiantes. En general todos se sintieron muy contentos con la herramienta educativa, realizaron comentarios favorables con respecto a la interfaz del usuario. En las pruebas realizadas el 95% de la población con los que se trabajó en la muestra han sido capaz de resolver las pruebas sin mayor problema.

#### **3.1.3.4. Resultados Obtenidos.**

Los estudiantes de quinto de Secundaria hicieron uso de la aplicación la cual ayudo a asimilar los contenidos de Estática y Dinámica en un tiempo mínimo. Para



determinar estos los resultados, se realizó una tabla comparativa en la cual refleja los resultados un antes y un después del uso de la aplicación.

**Tabla 3.10.** Tiempo de Desarrollo Curricular de temas

<b>Tema</b>	<b>Tiempo de Desarrollo Sin Tutor</b>	<b>Tiempo de desarrollo con el tutor</b>
Estática	3 meses	2 meses
Dinámica	3 meses	2 meses

**Fuente:** (Elaboración Propia)

## CAPITULO IV

### 4. CALIDAD Y SEGURIDAD

#### 4.1. Calidad del Software

(Pressman R. , 2002), define a la Calidad de Software como: “la concordancia con los requisitos funcionales y de rendimientos explícitamente establecidos, estándares de desarrollo explícitamente documentados y características implícitas que se espera de todo software desarrollado profesionalmente”.

Podemos decir que la calidad del Software comprende distintos aspectos como: la estética (debe ser agradable para la vista), funcionalidad (debe ser fácil de usar), eficiencia (debe ejecutarse con rapidez y precisión), y otros. Para poder evaluar las propiedades de este Tutor Inteligente Móvil se tomaron las características de calidad del modelo de la NORMA ISO/IEC 9126.

La calidad según esta norma ISO 9126 define 6 características las cuales son:

- Funcionalidad
- Fiabilidad
- Usabilidad
- Eficiencia
- Mantenibilidad
- Portabilidad

##### 4.1.1. Funcionalidad

Funcionalidad es la capacidad del software de cumplir y proveer las funciones para satisfacer las necesidades explícitas e implícitas cuando es utilizado en condiciones específicas. Se obtiene mediante el Punto de Función, tomaremos como base la Tabla de ajustes de complejidad, ver **Tabla 4.1**

**Tabla 4.1.** Valores para el ajuste de complejidad

<b>DATOS DE AJUSTE</b>	<b>VALOR</b>
Sin influencia	0
Incidental	1
Moderado	2
Medio	3
Significativo	4
Esencial	5

**Fuente:** (Pressman R. , 2002)

Estos valores de ajuste de complejidad se las da la ponderación según las respuestas a las siguientes preguntas, los cuales se aprecian en la siguiente tabla, ver **Tabla 4.2:**

**Tabla 4.2.** Factores de Ajuste

<b>FACTORES DE AJUSTE</b>	<b>VALOR</b>
1. ¿Requiere el sistema de copias de seguridad y de recuperación fiables?	3
2. ¿Se requiere comunicación de datos?	5
3. ¿Existen funciones de procesamiento distribuido?	3
4. ¿Es crítico el rendimiento?	3
5. ¿Se ejecutará el sistema en un entorno operativo existente y fuertemente utilizado?	4
6. ¿Requiere el sistema entrada de datos interactiva?	5
7. ¿Requiere la entrada de datos interactiva que las transacciones de entrada se lleven a cabo sobre múltiples pantallas u operaciones?	4
8. ¿Se actualizan los archivos maestros de forma interactiva?	4

9. ¿Son complejas las entradas, las salidas, los archivos o las peticiones?	3
10. ¿Es complejo el procesamiento interno?	4
11. ¿Se ha diseñado el código para ser reutilizable?	4
12. ¿Están incluidas en el diseño la conversión y la instalación?	3
13. ¿Se ha diseñado el sistema para soportar múltiples instalaciones en diferentes organizaciones?	4
14. ¿Se ha diseñado la aplicación para facilitar los cambios y para ser fácilmente utilizada por el usuario?	5
<b>Total <math>\sum(Fi)</math></b>	<b>54</b>

**Fuente:** (Pressman R. , 2002)

Por consiguiente, utilizaremos la **Tabla 4.3** que contiene los factores de ponderación utilizados:

**Tabla 4.3.** Cálculo de puntos de función sin Ajustar

<b>PARAMETROS DE CUENTA MEDICIÓN</b>	<b>CUENTA</b>	<b>SIMPLE</b>	<b>MEDIO</b>	<b>COMPLEJO</b>	<b>VALOR OBTENIDO</b>
Número de entradas de usuario	11	8x3=24	2x4=8	2x6=12	<b>44</b>
Número de salidas de usuario	5	4x4=16	1x5=5	2x7=14	<b>35</b>
Número de peticiones del usuario	6	6x3=18	2x3=6	0x3=0	<b>24</b>
Número de archivos	10	5x7=35	0x10=0	2x15=30	<b>65</b>
Número de interfaces externas	1	1x5=5	1x7=7	1x10=10	<b>22</b>
<b>Cuenta Total</b>					<b>190</b>

**Fuente:** (Pressman R. , 2002)

A continuación, realizaremos el cálculo de Punto de función, para lo cual se debe utilizar la fórmula:

$$PF = Cuenta\ Total * (Grado\ de\ confiabilidad + Tasa\ de\ Error * \Sigma(Fi))$$

**Donde:**

**Cuenta Total:** Total de puntos función sin ajustar.

**Grado de Confiabilidad:** Valor de 0.65, (pie de corrección).

**Tasa de Error:** Valor 0.01, (Error de confiabilidad de la aplicación).

**$\Sigma(Fi)$ :** Valor Total de la complejidad de la aplicación la cual se la obtiene de la Tabla.

Reemplazando los valores tenemos:

$$PF = 190 \times [0.65 + (0.01 \times 54)]$$

$$PF = 226.1$$

Para hallar el punto de Función Ideal, calculamos con el valor máximo de  $\Sigma(Fi) = 70$ .

$$PF_{ideal} = 190 \times [0.65 + (0.01 \times 70)]$$

$$PF_{ideal} = 256.5$$

Ahora calculamos el % de funcionalidad deseada:

$$Funcionalidad = \frac{PF}{PF_{ideal}} = \frac{226.1}{256.5} = 0.8814814815$$

$$Funcionalidad = 0.88148 \times 100 = 88.15\%$$

**FUNCIONALIDAD = 88.15%**

Por lo tanto, se obtuvo una funcionalidad de 88.15%, lo cual significa que la aplicación móvil cumple con los requisitos funcionales de manera satisfactoria a la Adecuación, Exactitud, Interoperabilidad, Seguridad y Conformidad.

#### 4.1.2. Fiabilidad

Para hallar un valor numérico aproximado de la fiabilidad del prototipo funcional debemos realizar los cálculos de “El tiempo medio entre fallas” y “La medida de disponibilidad de la aplicación”.

- ✓ Tiempo medio entre fallas.

$$TMEF = TMDF + TMDR$$

**Donde:**

**TMEF:** Tiempo medio entre fallos.

**TMDF:** Tiempo medio de fallos

**TMDR:** Tiempo medio de reparación.

Remplazando los valores se tiene:

$$TMEF = 7horas + 0.5horas$$

$$TMEF = 7.5horas$$

- ✓ Medida de disponibilidad de la aplicación.

$$Disponibilidad = \frac{TMDF}{TMDF + TMDR} = \frac{7 horas}{7 horas + 0.5 horas} = 0,933333333$$

$$Fiabilidad = 0,9333 \times 100 = 93.3\%$$

$$FIABILIDAD = 93.3\%$$

Por lo tanto, se tiene que en un lapso de 7.5 horas se tiene un 93.3% de fiabilidad.

### 4.1.3. Usabilidad

Se entiende por usabilidad el esfuerzo que se requiere por el usuario para utilizar el producto satisfactoriamente. Para comprobar la usabilidad de la aplicación móvil, se considera el método propuestos por Jacob Nielsen:

- Test de Usuario.

#### 4.1.3.1. Test de Usuario Final

El test de usuario final consiste en realizar una evaluación escrita después de la realización de las pruebas en los usuarios finales con una ponderación de un valor mínimo de 0 y un valor máximo de 100. Para lo cual se utilizó las siguientes preguntas y de la cual se obtuvo el siguiente puntaje de parte de los usuarios, detallado en la siguiente **Tabla 4.4.**

**Tabla 4.4.** Factores de ajuste usabilidad

<b>FACTOR DE AJUSTE</b>	<b>VALOR</b>
¿La aplicación es comprensible?	96
¿La aplicación puede ser utilizado fácilmente?	97
¿La aplicación es agradable para la vista?	91
¿Es útil para mi formación escolar y para determinadas tareas?	93
¿Es fácil aprender a manejar la aplicación?	96
¿La aplicación satisface las necesidades que usted requiere?	90
<b>PORCENTAJE TOTAL DE USABILIDAD</b>	<b>94%</b>

**Fuente:** (Pressman R. , 2002)

***USABILIDAD = 94%***

Podemos interpretar el resultado obtenido de usabilidad % como la facilidad en el acceso a interfaces. Es decir que de 100 usuarios pudieron entender y operar con facilidad.

#### 4.1.4. Eficiencia

Otra de las características que se debe medir es la eficiencia, la cual se define como “El grado en que el software emplea en forma óptima los recursos del sistema” (Pressman R. , 2002)

En la siguiente tabla detallaremos los factores de ajuste de eficiencia:

**Tabla 4.5.** Factores de ajuste de eficiencia

<b>FACTOR DE AJUSTE</b>	<b>VALOR</b>
Funciona con rapidez cuando realiza sus funciones	97
Tiene un rendimiento adecuado con los factores que utiliza	92
Utiliza solo cantidades y recursos necesarios.	91
<b>PORCENTAJE TOTAL DE EFICIENCIA</b>	<b>93.3%</b>

**Fuente:** (Pressman R. , 2002)

El valor obtenido para la eficiencia es:

$$EFICIENCIA = 93.3\%$$

La eficiencia de la aplicación móvil es de 93.3%, por lo que podemos decir que hace un buen uso de los recursos.

#### 4.1.5. Mantenibilidad

La mantenibilidad se refiere a los atributos que permiten medir el esfuerzo necesario para adaptarse a las nuevas especificaciones y requisitos del software. Para obtener la puntuación de la mantenibilidad del sistema, se desarrollaron las siguientes



preguntas, estas son contestadas por el desarrollador del sistema y no por los usuarios detallada en la **Tabla 4.6**.

**Tabla 4.6.** Factores de ajuste de mantenibilidad

<b>FACTOR AJUSTE</b>	<b>VALOR</b>
Puede ser modificado el sistema	94
Deja identificar las partes que deben ser modificadas	91
Permitir implementar una modificación específica.	83
No presenta efectos inesperados en posibles errores	92
<b>PORCENTAJE TOTAL DE MANTENIBILIDAD</b>	<b>90%</b>

**Fuente:** (Pressman R. , 2002)

Entonces obtenemos el valor porcentual para la mantenibilidad que es:

$$MANTENIBILIDAD = 90\%$$

El valor obtenido de 90% en mantenibilidad, se puede interpretar que el producto software contempla una cantidad de esfuerzo mínimo para cualquier, cambio o modificación que requiera el sistema.

#### **4.1.6. Portabilidad**

Se define como la capacidad del producto o componente de ser transferido de un entorno a otro de forma efectiva y eficiente. Para la puntuación de la portabilidad, se desarrollaron las siguientes preguntas (Ver **Tabla 4.7**), Las cuales están contestadas por el desarrollador del sistema y no por los usuarios.

**Tabla 4.7.** Factores de ajuste de portabilidad

<b>FACTOR DE AJUSTE</b>	<b>VALOR</b>
Puede ser transferido de un entorno a otro	80
Se puede adaptar a otros ambientes con facilidad.	98
Es fácil de instalar	99
Es capaz de reemplazar a un software similar	91

---

<b>PORCENTAJE TOTAL DE PORTABILIDAD</b>	<b>92%</b>
---	------------

---

**Fuente:** (Pressman R. , 2002)

***PORTABILIDAD = 92%***

Este resultado significa que en un 92% tiene la capacidad de ejecutarse en diferentes dispositivos móviles especialmente S.O. Android.

#### **4.2. Resultado total**

Según a los resultados obtenidos se puede obtener la calidad total del sistema, en base a parámetros medidos y expresadas en porcentaje. En la **Tabla 4.8** muestra la calidad total del sistema, que es relacionada con el grado de satisfacción con el usuario que ingresa al sistema.

**Tabla 4.8.** Calidad Total

<b>FACTOR DE AJUSTE</b>	<b>VALOR</b>
Funcionalidad	88.15%
Fiabilidad	93.3%
Usabilidad	94%
Eficiencia	93.3%
Mantenibilidad	90%
Portabilidad	92%
<b>PORCENTAJE TOTAL</b>	<b>91.79%</b>

**Fuente:** (Elaboración Propia)

Entonces el resultado Total de la calidad del software desarrollado es:

***EVALUACIÓN DE CALIDAD TOTAL = 91.79%***

Este resultado se puede interpretar de que el **91.79%** de los 30 estudiantes, consideran, al Tutor Inteligente Móvil como una herramienta útil y de buena calidad.

### **4.3. Seguridad del tutor.**

La seguridad de la tecnología o ciberseguridad es la capacidad de protección de la información y los datos de manera que las personas no autorizadas puedan leerlas y/o modificarlas.

#### **4.3.1. Factor de Seguridad ISO 17799**

La ISO 17799 es una norma internacional, que se define como una guía protocolar de un conjunto de normas a llevar a cabo en la implementación del sistema de administración de la seguridad de la información, y se orienta a preservar los siguientes principios:

##### **4.3.1.1. Confidencialidad**

La aplicación TimE&D tiene una seguridad mínima ya que la información guardada en el sistema no puede ser utilizada en contra de nadie. Por lo tanto, la seguridad que tiene el acceso al sistema es mediante el módulo de inicio de sesión de usuarios que restringe el acceso a usuario no autorizado. El usuario realiza su autenticación de ingreso al sistema por medio de usuario (correo) y contraseña (clave alfanumérica) donde estos están encriptados por un cifrado.

##### **4.3.1.2. Integridad**

Garantizar que la información contenida en el sistema no será alterada, eliminada o destruida por entidades no autorizadas; preservando así la exactitud y la completitud de la misma y de los métodos de su procesamiento.

El tutor inteligente Móvil solamente puede ser modificado por el que creo dicha aplicación, ya que el software lleva un único instalador.

##### **4.3.1.3. Disponibilidad**

Los contenidos de los temas de la aplicación estarán disponibles para todos los usuarios autorizados en el momento oportuno y necesario para complementar su aprendizaje.

La protección de la información constituye un mecanismo básico en cualquier organización, sean instituciones de gobierno, educativas, no obstante, dependiendo de la naturaleza y metas de las estructuras organizacionales, éstas mostrarán especial énfasis en algún dominio o área del estándar ISO 17799.

## **CAPITULO V**

### **5. Análisis de Costo/Beneficio**

El análisis de costo beneficio se considera en una de las tareas de mayor importancia dentro de una planificación de proyectos de software, consiste en determinar, con cierto grado de certeza, los recursos necesarios para el desarrollo del proyecto como ser: hardware y software, costo, tiempo y esfuerzo.

En Ingeniería de Software, la estimación de costos básicamente se refiere a estimar la cantidad de personas necesarias para desarrollar el proyecto.

#### **5.1. Estimación de Costos de Software**

La estimación de costos de software relaciona conceptos generales y técnicas de análisis económico en el mundo particular de la Ingeniería de Software. Para la estimación de costos de desarrollo de software existen distintos métodos, las cuales son una relación matemática entre el esfuerzo y el tiempo de desarrollo.

Para determinar el Costo Total del proyecto se tomará en cuenta los siguientes costos:

- Costo del software desarrollado.
- Costo de implementación del proyecto.
- Costo de la Elaboración del Proyecto.

##### **5.1.1. Costo del Software Desarrollado**

Para determinar el costo de la aplicación se usa el modelo de costos COCOMO II, en la cual la estimación del sistema ha sido desarrollada bajo las KLDC (Kilo – Líneas de Código).

El presente proyecto TimE&D se implementó con 8596 Líneas de código en el lenguaje Java:

$$KLDC = LDC/1000$$

$$KLDC = 8596/1000$$

$$KLDC = (8,596) KLDC$$

Por lo que la evaluación del sistema ha sido considerada en, **8.596 KLDC**.

Los coeficientes que se usarán serán los valores que se detallan en la siguiente tabla:

**Tabla 5.1.** Coeficiente de Modelo COCOMO II

MODO	a	B	c	D
Orgánico	2.40	1.05	2.50	0.38
Semi – orgánico	3.00	1.12	2.50	0.35
Empotrado	3.60	1.20	2.50	0.33

**Fuente:** (Pressman R. , 2002)

Ecuaciones para calcular el costo de Software:

**Tabla 5. 2.** Ecuaciones del Modelo COCOMO II

Variable	Educación	Tipo de Unidad
Esfuerzo requerido por el proyecto	$E = a * (KLDC)^b * FAE$	Personas por mes
Tiempo Requerido por el proyecto	$T = c * (E)^d$	Meses
Número de personas requeridas para el proyecto	$P = \frac{E}{T}$	Personas
Costo Total	$CT = sueldo\ Mes * P * T$	\$us

**Fuente** (Prentice- Hall, 1981)

Para hallar los valores de FAE se utiliza la **Tabla 5.3**.

**Tabla 5.3.** Calculo de los atributos FAE

ATRIBUTOS	Valor					
	Muy bajo	Bajo	Nominal	Alto	Muy alto	Extra alto
<b>Atributos de software</b>						
Fiabilidad del software	0,75	0,88	<b>1,00</b>	1,15	1,40	
Tamaño de base de datos		<b>0,94</b>	1,00	1,08	1,16	
Complejidad del producto	0,70	0,85	<b>1,00</b>	1,15	1,30	1,65
<b>Atributos de hardware</b>						
Restricciones de tiempo de ejecución			<b>1,00</b>	1,11	1,30	1,66
Restricciones de almacenamiento principal			<b>1,00</b>	1,06	1,21	1,56
Volatilidad de la máquina virtual		<b>0,87</b>	1,00	1,15	1,30	
Tiempo de respuesta		0,87	<b>1,00</b>	1,07	1,15	
<b>Atributos de personal</b>						
Capacidad de análisis	1,46	1,19	<b>1,00</b>	0,86	0,71	
Experiencia en la aplicación	1,29	1,13	<b>1,00</b>	0,91	0,82	
Calidad de los programadores	1,42	1,17	<b>1,00</b>	0,86	0,70	
Experiencia de S.O. usado	1,21	1,10	1,00	<b>0,90</b>		
Experiencia en el lenguaje	1,14	1,07	<b>1,00</b>	0,95		
<b>Atributo de Proyecto</b>						
Uso de técnicas actualizadas de programación	1,24	1,10	1,00	<b>0,91</b>	0,82	
Utilización de Herramientas de software	1,24	1,10	1,00	<b>0,91</b>	0,83	
Restricciones de tiempo de desarrollo	1,22	1,08	<b>1,00</b>	1,04	1,10	
<b>TOTAL FAE = 0,609498</b>						

**Fuente:** (Elaboración Propia)

Aplicando las ecuaciones (detalladas en la **Tabla 5. 2.**) así como los coeficientes a y c y los exponentes b y d que en nuestro caso el tipo orgánico será el más apropiado ya que el número de líneas de código no supera los 50 KLDC, (descritos en la **Tabla 5. 1**) y el cálculo de los atributos FAE (descrito en la **Tabla 5. 2**).

Se tiene:

**Calculando el Esfuerzo:**

$$E = a * (KLDC)^b * FAE \text{ (Personas/Mes)}$$

$$E = 2,4 * (8,596)^{1,05} * 0,609498 \text{ (Personas/Mes)}$$

$$E = 14,0022 \text{ (Personas/Mes)} \cong 14 \text{ (Personas/Mes)}$$

**Calculando el Tiempo de desarrollo:**

$$T = c * (E)^d \text{ (Meses)}$$

$$T = 2,5 * (14,0022)^{0,38} \text{ (Meses)}$$

$$T = 6,81545 \text{ (Meses)} \cong 7 \text{ (Meses)}$$

**Calculando la Productividad:**

$$PR = LCD/E \text{ (LDC/Personas Mes)}$$

$$PR = 8596/14,0022 \text{ (LDC/Personas Mes)}$$

$$PR = 613,9035 \text{ (LDC/Personas Mes)}$$

**Calculando del Personal Promedio:**

$$P = \frac{E}{T} \text{ (Personas)}$$

$$P = \frac{14,0022}{7,00} \text{ (Personas)}$$

$$P = 2,00 \text{ Equivalente a 2 Personas}$$

**Cálculo de Costo** Personas mes (*Salario promedio = 300 \$ o 2090 Bs*)

$$\text{Costo Mes} = \text{Persona} * \text{Salario promedio entre programadores}$$



$$\text{Costo Mes} = 2 * 300 = 600\$$$

Calculo de Costo del software

$$\text{Costo Total} = \text{Costo Mes} * \text{Tiempo}$$

$$\text{Costo Total} = 600 * 7,00 = 4200 \$$$

En conclusión, se requiere un estimado de 2 personas trabajando alrededor de 7 meses con un costo total de 4200 \$us.

### 5.1.2. Costo de implementación del proyecto

La distribución de la aplicación a los estudiantes de la Unidad Educativa “Dionicio Morales” será de distribución libre, por lo que el costo de la implementación es cero para la aplicación.

### 5.1.3. Costo de elaboración del proyecto

Los costos de elaboración del proyecto, referidos a los costos del estudio del sistema, en la etapa de recopilación y análisis, se detallan en la siguiente tabla:

**Tabla 5.4.** Costo de Elaboración del Proyecto

<b>Descripción</b>	<b>Costo Total (\$us)</b>
Análisis y Diseño del Proyecto	40
Material de escritorio	20
Investigación en Internet	60
Bibliografía	40
Otros	35
<b>TOTAL</b>	<b>195</b>

**Fuente:** (Elaboración Propia)

#### 5.1.4. Costo total del proyecto

Para el costo total del proyecto se realiza la sumatoria del costo de la aplicación desarrollada, costo de implementación de la aplicación y costo de elaboración del proyecto, como se muestra en la siguiente tabla:

**Tabla 5.5.** Costo Total del proyecto

<b>Descripción</b>	<b>Costo Total(\$us)</b>
Costo del Software desarrollado	4200
Costo de Implementación de la aplicación	0
Costo de Elaboración del Proyecto	195
<b>TOTAL</b>	<b>4395</b>

**Fuente:** (Elaboración Propia)

#### 5.2. Análisis de Beneficios

Los beneficios que presentará el proyecto de grado para la unidad educativa “Dionicio Morales” son:

- El beneficio del software desarrollado se trata en términos de uso de la aplicación, que brindará indirectamente un beneficio al aprendizaje de Estática y Dinámica de los estudiantes.
- El costo total del desarrollo de la aplicación definida anteriormente.

## **CAPITULO VI**

### **6. CONCLUSIONES Y RECOMENDACIONES.**

#### **6.1. Conclusiones.**

Al termino del presente proyecto “TUTOR INTELIGENTE MÓVIL PARA LA ENSEÑANZA DE LA ESTÁTICA Y DINÁMICA EN ESTUDIANTES DE QUINTO DE SECUNDARIA CASO UNIDAD EDUCATIVA DIONICIO MORALES”, tomando en cuenta los objetivos planteados se llega a las siguientes conclusiones:

- Se diseñó un material didáctico que logro apoyar en la parte teórica conceptual y resolución de ejercicios a los estudiantes, con los conocimientos de estática y dinámica.
- Se realizó la implementación de una base de datos para el registro de estudiantes de quinto de secundaria.
- Se implementó un módulo evaluador que permite medir el nivel de aprendizaje de los estudiantes, en base a preguntas planteadas y de forma aleatoria.
- Se facilitó a los estudiantes una herramienta educativa, que a su vez apoya al docente en el desarrollo curricular de temas y la búsqueda de un buen aprendizaje de sus estudiantes.
- Estudiantes motivados y con interés de aprender los contenidos temáticos de Estática y Dinámica, dentro y fuera de la clase sobre el avance del desarrollo curricular.

De esta manera se logró alcanzar el objetivo general, ya que el Tutor Inteligente Móvil permitió mejorar la asimilación de conocimientos en un tiempo mínimo y forma dinámica, en los estudiantes de quinto de Secundaria de la Unidad Educativa “Dionicio Morales”

## **6.2. Recomendaciones.**

A partir de este presente trabajo es necesario que se tomen en cuenta las siguientes recomendaciones con el fin de buscar mejorar este trabajo o futuras creaciones algún tipo de relación con este sistema.

- Se recomienda analizar la situación inicial de los estudiantes con respecto de conocimientos previos de los estudiantes en los temas para el cual se desarrolló el proyecto.
- Se recomienda promover el uso de las nuevas tecnologías como herramientas de enseñanza y aprendizaje, porque se debe dar el buen uso a estas.
- Contar con un dispositivo móvil que soporte el sistema operativo Android con la versión mínima de 4.2.
- Desarrollar el sistema en el entorno web, para que no sea solamente utilizado en dispositivos móviles, sino pueda ser utilizado en cualquier dispositivo.
- Ampliar el contenido en el módulo tutor, para que también pueda ser una herramienta que sirva a los estudiantes universitarios.

## BIBLIOGRAFÍA

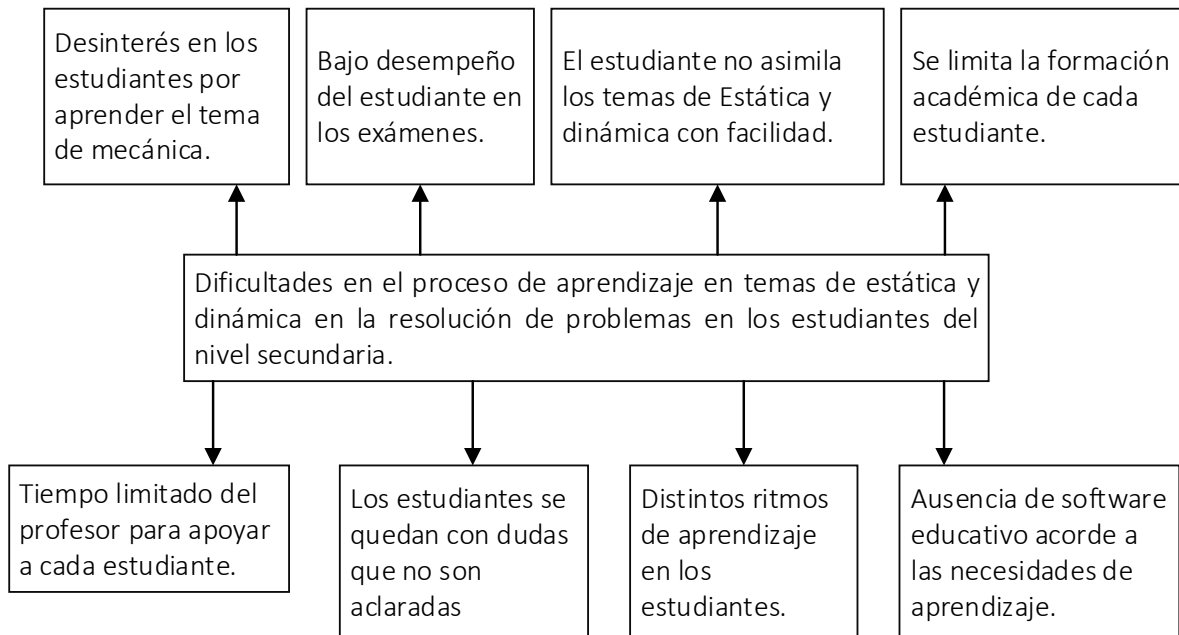
- Alvarado, A. (2018). El proceso de enseñanza-aprendizaje de los Estudios Lingüísticos: su impacto en la motivación hacia el estudio de la lengua . *Mendive*.
- Andrade, G. H. (2017). Modelo de un Sistema Tutor Inteligente para el desarrollo del pensamiento computacional.
- Castellanos, D. (2012). *Aprender y Enseñar en la Escuela*. La Habana - Cuba: Pueblo y Educación.
- Cataldi, Z. (2004). Metodología para el diseño de sistemas tutores Inteligentes. *Plan de tesis Tesis doctoral (en elaboracion) Facultad de Informática*.
- Cataldi, Z. (2009). *SISTEMAS TUTORES INTELIGENTES ORIENTADOS A LA ENSEÑANZA PARA LA COMPRENSIÓN*. Buenos Aires: Revista Electronica de Tecnologia Educativa.
- Espitia, C. (2008). *Programa de Psicología Aprendizaje Autonomo*. UNIMINUTO.
- Galvis, A. (1994). Ingeniería de Software Educativo. *Revista Internacional de Educación en Ingeniería*, Volumen 2.
- Gómez Garcia, Á. (2007). *INTELIGENCIA ARTIFICIAL Y COMPUTACIÓN AVANZADA*. <https://cdv.dei.uc.pt/wp-content/uploads/2014/03/ms07.pdf>: Colección Informática.
- Gómez Martín, M. A. (2002). Agente pedagógico para enseñar la estructura de la JVM. *Universidad Complutense Madrid*, Madrid España.
- Gonzáles Soledad, C. (2004). ISTEMAS INTELIGENTES EN LA EDUCACIÓN: UNA REVISIÓN DE LAS LÍNEAS DE INVESTIGACIÓN Y APLICACIONES ACTUALES. *Revista ELección de Investigación y EValuación Educativa*.
- Gonzales, C. S. (2004). SISTEMAS INTELIGENTES EN LA EDUCACIÓN: UNA REVISIÓN DE LAS LÍNEAS DE INVESTIGACIÓN Y APLICACIONES ACTUALES. *Revista ELección de Investigación y Evaluación Educativa*.

- Il, C. (2018). *Recuperado de Unpocodejava*:. Obtenido de <https://unpocodejava.com/2012/02/07/modelos-de-estimacion-un-poco-sobrecocomo-ii/>
- ISO/IEC. (2007).
- Manuel Pérez, J., & Viejo, D. (2008). Una concepción moderna de Técnicas de Inteligencia Artificial. *Dpto. de Ciencia de la Computación e I.A.* , Apartado 99 03080 Alicante .
- Parra Pineda, D. M. (2003). *MANUAL DE ESTRATEGIAS DE ENSEÑANZA/APRENDIZAJE*. Colombia.
- Pérez Terrel, W. (2015). *FÍSICA Teoría y Práctica*. Perú: SAN MARCOS.
- Pressman, R. (2002). *Ingeniería de Software Un enfoque práctico 5° Edición*. Madrid - España: McGraw-Hill.
- Pressman, R. S. (2010). *Ingeniería del software Un enfoque práctico*. Mexico: McGrawHill.
- Quisbert Callisaya, L. P. (2019). *Física - Química Educación Secundaria Comunitaria Productiva*. La Paz: Abya Yala.

# **ANEXOS**

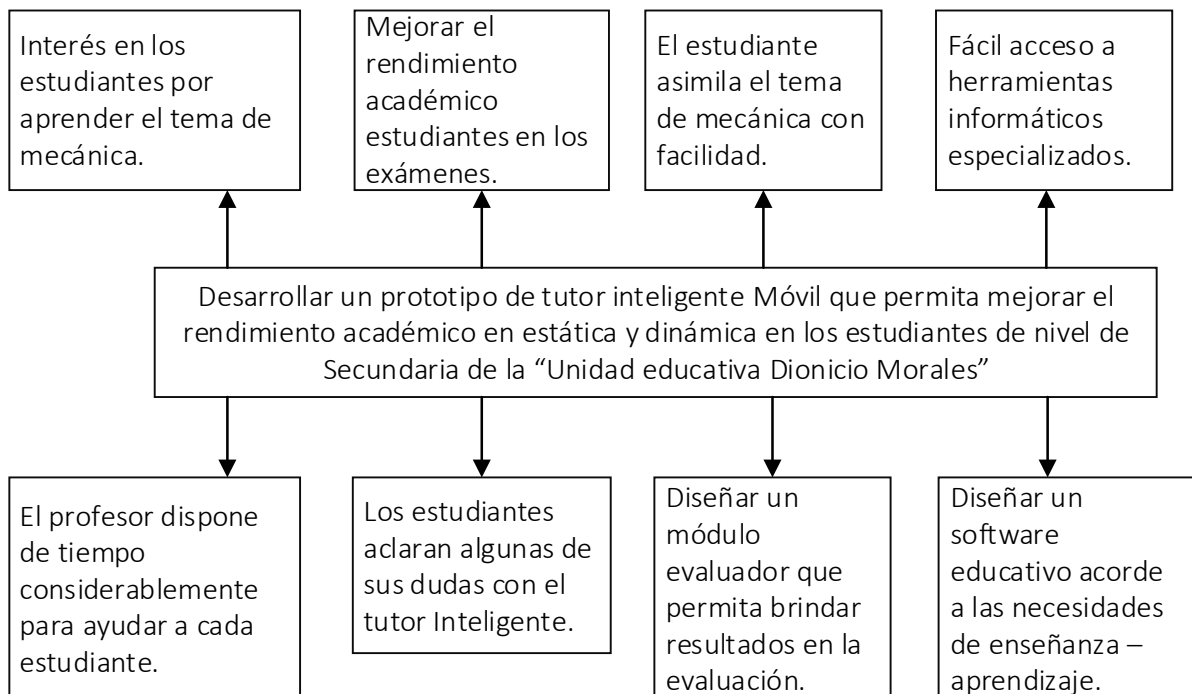
---

## ANEXO A. ARBOL DE PROBLEMAS.



## ANEXO B.

### ARBOL DE OBJETIVOS.

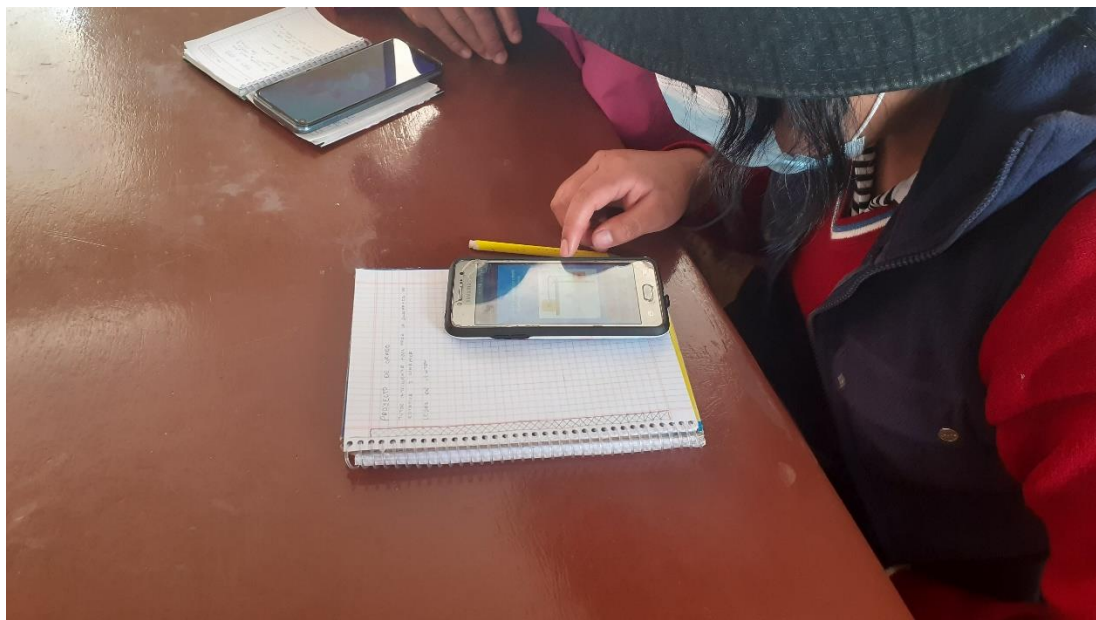




## ANEXO C. ESTUDIANTES DE QUINTO DE SECUNDARIA



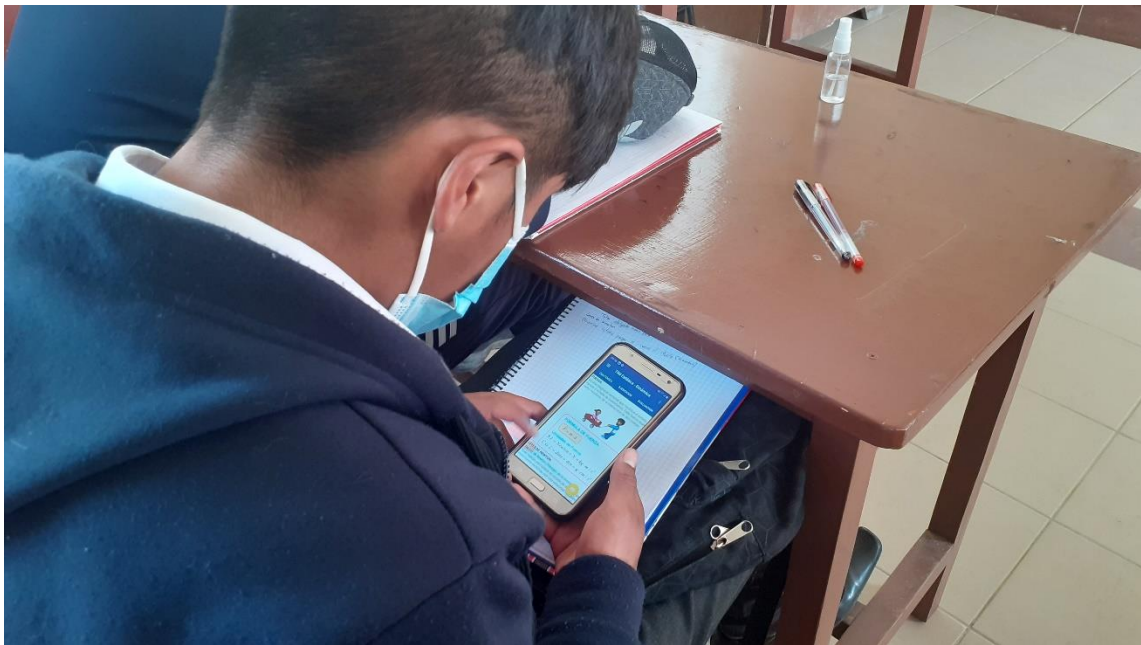
**Estudiantes realizando pruebas de la aplicación**



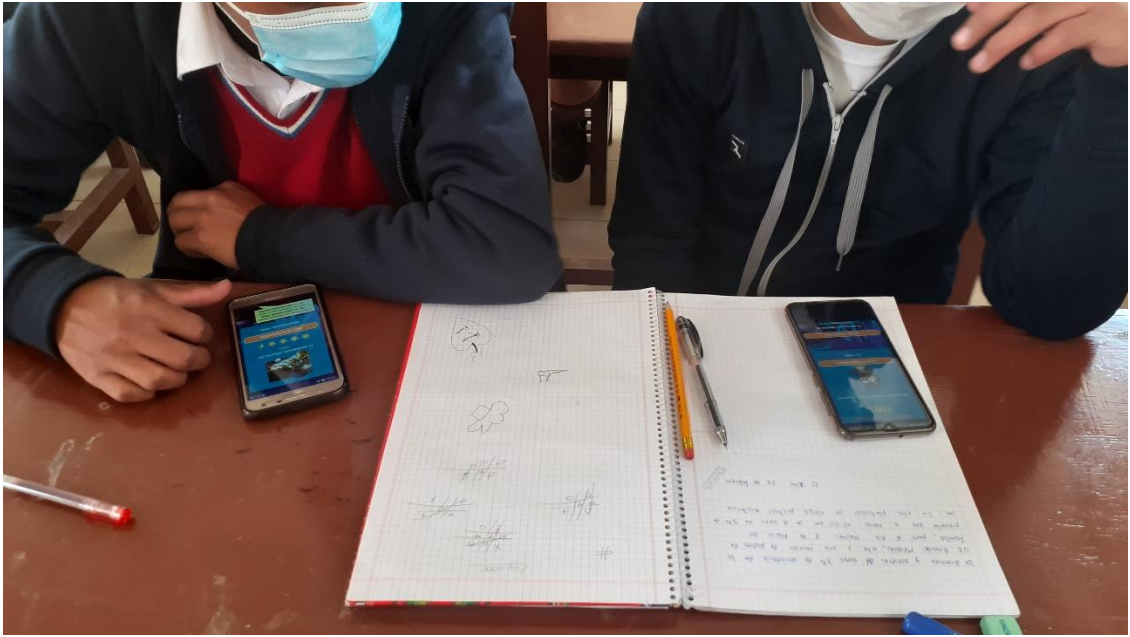
**Estudiantes repasando las lecciones**



**Estudiantes Utilizando la aplicación para reforzar sus conocimientos**







**Estudiantes realizando la evaluación con la aplicación**



MANUAL DE USUARIO APLICACIÓN MOVIL

**TUTOR INTELIGENTE MÓVIL PARA LA  
ENSEÑANZA DE LA ESTÁTICA Y DINÁMICA EN  
ESTUDIANTES DE QUINTO DE SECUNDARIA**



**JUSTINIANO MENDOZA APAZA**

**Versión 1.0**

La Paz – Bolivia

2020

## 1. INTRODUCCIÓN

El Tutor Inteligente Móvil es una herramienta educativa, que se puede utilizar como una alternativa que ayuda en el aprendizaje de conocimientos de los temas Estática y Dinámica para los estudiantes de quinto de Secundaria de la Unidad Educativa “Dionicio Morales”

## 2. Requerimiento Mínimo de Hardware

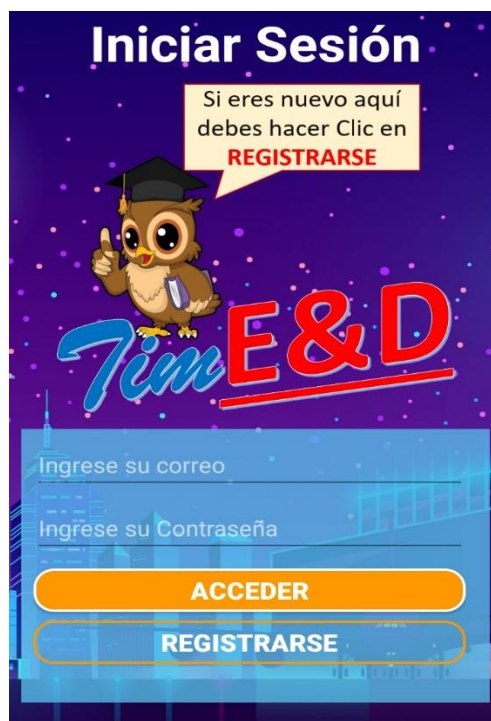
- Un dispositivo Móvil (Celular SmartPhone)
- Memoria RAM 1GB
- Procesador 1,6 GHz

## 3. Requerimiento Mínimo de Hardware

- Android versión 4.4

## 4. INGRESO AL SISTEMA

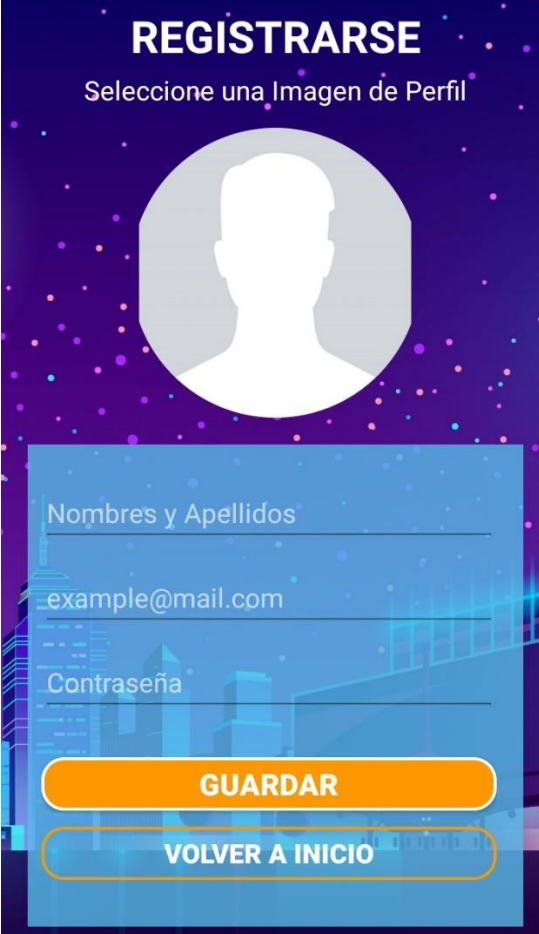
Para ingresar a sistema primeramente se muestra el interfaz de inicio de Sesión donde se autentican los usuarios previo registro para el Uso de la aplicación. Se debe ingresar con el Correo y Contraseña registrada.



## 5. REGISTRO DE USUARIO

Para registrarse, el Usuario (Estudiante) debe ingresar los siguientes datos:

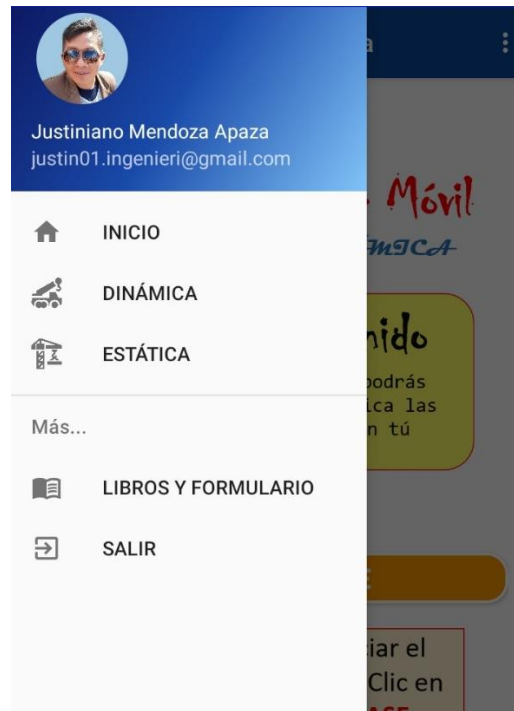
- Seleccionar una imagen de Perfil de Usuario
- Ingresar su nombre completo
- Correo Electrónico del usuario.
- Ingresar una contraseña Alfanumérico de 6 caracteres mínimamente.



The image shows a registration form with a dark blue background and a starry pattern. At the top, the word "REGISTRARSE" is written in white. Below it, the text "Seleccione una Imagen de Perfil" is displayed. A circular placeholder for a profile picture is shown. Below the profile picture, there are three input fields: "Nombres y Apellidos" with the text "Nombres y Apellidos" inside, "example@mail.com" in the email field, and "Contraseña" in the password field. At the bottom, there are two buttons: "GUARDAR" (orange) and "VOLVER A INICIO" (white with orange border).

## 6. MENU DE CONTENIDOS

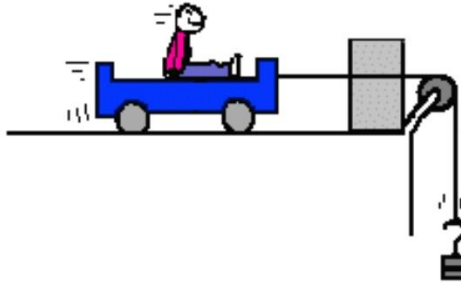
Donde el usuario realiza la selección de tema en el cual necesita apoyo haciendo Click en la Opcion.



## 7. SECCION TEMA.

Tiene tres opciones de selección CONTENIDO, EJERCICIOS Y EVALUACION. Las cuales pueden ser accedidas desde el menú. Y haciendo deslizar pantalla a la izquierda.

**DINÁMICA**



**DEFINICIÓN.**

La dinámica es una parte de la mecánica que estudia al movimiento de los cuerpos tomando en cuenta las causas que lo originan.

**INERCIA.**

Es una propiedad de la materia que se manifiesta como tendencia a conservar el estado de reposo o el estado de movimiento rectilíneo uniforme

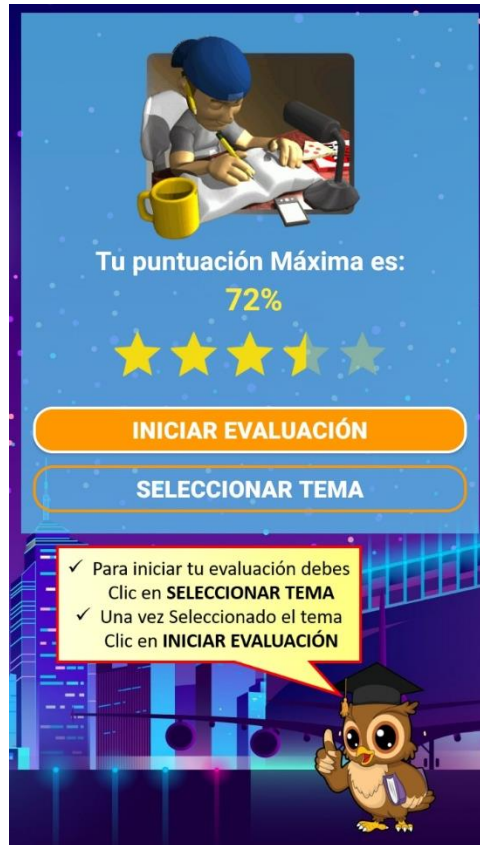
*Ejemplo.*



**8. ESTUDIANTE**

Es el interfaz del estudiante donde se refleja, el nivel de simulación en los temas de estática y Dinámica.





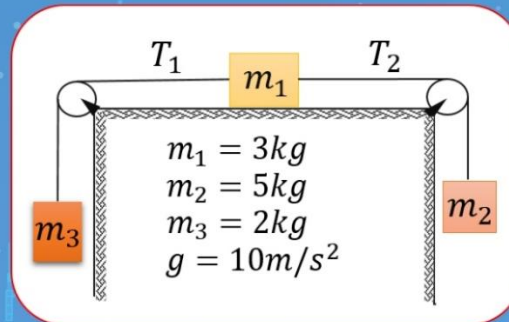
## 9. INTERFAZ EVALUACIÓN.

Muestra las preguntas y las opciones de respuestas en forma aleatoria, realizando el conteo de la cantidad de respuestas marcadas de forma correcta e incorrecta.

Correctos: 9

Incorrectos: 2

Calcular la aceleración de los bloques en el sistema físico mostrado en la figura. Sabiendo que no existe rozamiento.



$6\text{m/s}^2$

$4\text{m/s}^2$

Ninguna de las anteriores.

$3\text{m/s}^2$

El Alto, 22 de noviembre de 2020

Señor:

**Ing. David Carlos Mamani Quispe**

**DIRECTOR CARRERA INGENIERÍA DE SISTEMAS – UPEA**

Presente. -

**REF.: AVAL DE CONFORMIDAD**

Distinguido Ingeniero:

Tengo a bien dirigirme a su persona para comunicarle mi conformidad del trabajo final del Proyecto de Grado, titulado **“TUTOR INTELIGENTE MÓVIL PARA LA ENSEÑANZA DE LA ESTÁTICA Y DINÁMICA EN ESTUDIANTES DE QUINTO DE SECUNDARIA CASO: UNIDAD EDUCATIVA DIONICIO MORALES”**, elaborado por el universitario: **JUSTINIANO MENDOZA APAZA**, con Cedula de Identidad: **6132294 L.P.** y Registro Universitario: **12007386**, para su defensa pública y evaluación correspondiente a la materia de **Taller de Licenciatura II**, de acuerdo al reglamento vigente de la Carrera de Ingeniería de Sistemas de la Universidad Pública de El Alto.

Sin otro particular, me despido de usted.

Atentamente.



.....  
Ing. Marisol Arguedas Balladares  
**TUTOR METODOLÓGICO**

El Alto, 22 de noviembre de 2020

Señor:

**Ing. David Carlos Mamani Quispe**

**DIRECTOR CARRERA INGENIERÍA DE SISTEMAS – UPEA**

Presente. -

**REF.: AVAL DE CONFORMIDAD**

Distinguido Ingeniero:

Tengo a bien dirigirme a su persona para comunicarle mi conformidad del trabajo final del Proyecto de Grado, titulado **“TUTOR INTELIGENTE MÓVIL PARA LA ENSEÑANZA DE LA ESTÁTICA Y DINÁMICA EN ESTUDIANTES DE QUINTO DE SECUNDARIA CASO: UNIDAD EDUCATIVA DIONICIO MORALES”**, elaborado por el universitario: **JUSTINIANO MENDOZA APAZA**, con Cedula de Identidad: **6132294 L.P.** y Registro Universitario: **12007386**, para su defensa pública y evaluación correspondiente a la materia de **Taller de Licenciatura II**, de acuerdo al reglamento vigente de la Carrera de Ingeniería de Sistemas de la Universidad Pública de El Alto.

Sin otro particular, me despido de usted.

Atentamente.



.....  
Lic. Katya Maricela Pérez Martínez  
**TUTOR ESPECIALISTA**

El Alto, 22 de noviembre de 2020

Señor:

**Ing. David Carlos Mamani Quispe**

**DIRECTOR CARRERA INGENIERÍA DE SISTEMAS – UPEA**

Presente. -

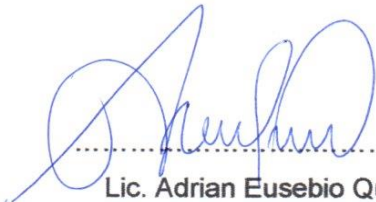
**REF.: AVAL DE CONFORMIDAD**

Distinguido Ingeniero:

Tengo a bien dirigirme a su persona para comunicarle mi conformidad del trabajo final del Proyecto de Grado, titulado **“TUTOR INTELIGENTE MÓVIL PARA LA ENSEÑANZA DE LA ESTÁTICA Y DINÁMICA EN ESTUDIANTES DE QUINTO DE SECUNDARIA CASO: UNIDAD EDUCATIVA DIONICIO MORALES”**, elaborado por el universitario: **JUSTINIANO MENDOZA APAZA**, con Cedula de Identidad: **6132294 L.P.** y Registro Universitario: **12007386**, para su defensa pública y evaluación correspondiente a la materia de **Taller de Licenciatura II**, de acuerdo al reglamento vigente de la Carrera de Ingeniería de Sistemas de la Universidad Pública de El Alto.

Sin otro particular, me despido de usted.

Atentamente.

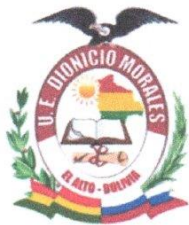


.....

Lic. Adrian Eusebio Quisbert Vilela

**TUTOR REVISOR**





# UNIDAD EDUCATIVA "DIONICIO MORALES"

DISTRITO EDUCATIVO EL ALTO - 1  
NIVELES INICIAL, PRIMARIA Y SECUNDARIA COD. SIE - 40730514  
Fundado el 18 de Febrero de 2008  
EL ALTO - LA PAZ



El Alto, noviembre de 2020

Señor:

Justiniano Mendoza Apaza

De: Lic. Huber Guarachi Quispe

**DIRECTOR DE LA UNIDAD EDUCATIVA "DIONICIO MORALES"**

La Unidad Educativa "Dionicio Morales" se complace en extender su **AVAL DE CONFORMIDAD** a la solicitud del Sr. **Justiniano Mendoza Apaza C.I. 6132294 L.P.** que realizó el proyecto: **"TUTOR INTELIGENTE MÓVIL PARA LA ENSEÑANZA DE LA ESTÁTICA Y DINÁMICA EN ESTUDIANTE DE QUINTO DE SECUNDARIA CASO: UNIDAD EDUCATIVA DIONICIO MORALES"**, en fechas 06 de julio al 13 de noviembre del 2020. Cumpliendo satisfactoriamente el proceso de obtención de la información referente a la situación académica de los estudiantes e implementación del prototipo del proyecto, cabe resaltar que cubrió las expectativas de los estudiantes de Quinto de Secundaria.

Brindamos este certificado a pedido del interesado, deseando éxitos en los proyectos que realice posteriormente, nuestra institución está a disposición si se quiere actualizar o contribuir con otro proyecto en bien de la comunidad estudiantil.

Atentamente.

cc. Arch.



Lic. Huber Guarachi Quispe  
**DIRECTOR**  
U.E. "DIONICIO MORALES"

Zona Bautista Saavedra Unidad Vecinal "H" – Sub Distrito Municipal 14 Av. Simón Bolívar – Calle Carlos Palenque S/N