



**UNIVERSIDAD CATÓLICA DE  
SANTIAGO DE GUAYAQUIL**

**FACULTAD DE INGENIERÍA  
CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES**

**TEMA:  
ANÁLISIS DE LAS APLICACIONES *GAME BASED-LEARNING* QUE  
CONFLUYEN COMO DESARROLLO DE LAS TIC EN EL ÁREA DE LA  
PROGRAMACIÓN MEDIANTE LA NORMA ISO 25000.**

**AUTORA:**  
Segovia Castro, Gabriela Nathaly

**Trabajo de titulación previo a la obtención del título de INGENIERO EN  
SISTEMAS COMPUTACIONALES**

**TUTOR:**  
Ing. Roberto García Sánchez

**Guayaquil, Ecuador**

14 de septiembre del 2020



**UNIVERSIDAD CATÓLICA DE  
SANTIAGO DE GUAYAQUIL**

**FACULTAD DE INGENIERÍA  
CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES**

**CERTIFICACIÓN**

Certificamos que el presente trabajo de titulación fue realizado en su totalidad por **Segovia Castro Gabriela Nathaly**, como requerimiento para la obtención del título de **Ingeniero en Sistemas Computacionales**.

TUTOR

f.

Ing. García Sánchez, Roberto

**DIRECTOR DE LA CARRERA**

**Guayaquil, a los 14 días del mes de septiembre del año 2020**



**UNIVERSIDAD CATÓLICA DE SANTIAGO DE  
GUAYAQUIL**

**FACULTAD DE INGENIERÍA  
CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES**

**DECLARACIÓN DE RESPONSABILIDAD**

Yo, por **Segovia Castro Gabriela Nathaly**

**DECLARO QUE:**

El Trabajo de Titulación, **Análisis de las aplicaciones *game based-learning* que confluyen como desarrollo de las TIC en el área de la programación mediante la norma ISO 25000** previo a la obtención del título de **Ingeniero en Sistemas Computacionales**, ha sido desarrollado respetando derechos intelectuales de terceros conforme las citas que constan en el documento, cuyas fuentes se incorporan en las referencias o bibliografías. Consecuentemente este trabajo es de mi total autoría.

En virtud de esta declaración, me responsabilizo del contenido, veracidad y alcance del Trabajo de Titulación referido.

Guayaquil, a los 14 días del mes de septiembre del año 2020

EL AUTOR

f. Gabriela Segovia  
Segovia Castro Gabriela Nathaly



**UNIVERSIDAD CATÓLICA DE  
SANTIAGO DE GUAYAQUIL**

**FACULTAD DE INGENIERÍA  
CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES**

**AUTORIZACIÓN**

**Yo, Segovia Castro Gabriela Nathaly**

Autorizo a la Universidad Católica de Santiago de Guayaquil a la **publicación** en la biblioteca de la institución del Trabajo de Titulación, **Análisis de las aplicaciones *game based-learning* que confluyen como desarrollo de las TIC en el área de la programación mediante la norma ISO 25000**, cuyo contenido, ideas y criterios son de mi exclusiva responsabilidad y total autoría.

Guayaquil, a los 14 días del mes de septiembre del año 2020

**EL AUTOR:**

f. GABRIELA Segovia  
Segovia Castro Gabriela Nathaly



# UNIVERSIDAD CATÓLICA DE SANTIAGO DE GUAYAQUIL

FACULTAD DE INGENIERÍA  
CARRERA DE INGENIERÍA EN SISTEMAS  
COMPUTACIONALES

## REPORTE URKUND

URKUND

<b>Documento</b>	<a href="#">TESIS - SEGOVIA.docx</a> (D78173305)
<b>Presentado</b>	2020-08-25 19:42 (-05:00)
<b>Presentado por</b>	gabriela.segovia@cu.ucsg.edu.ec
<b>Recibido</b>	roberto.garcia02.ucsg@analysis.arkund.com
<b>Mensaje</b>	Tesis Segovia Castro <a href="#">Mostrar el mensaje completo</a> 4% de estas 51 páginas, se componen de texto presente en 11 fuentes.

**TUTOR**

f. \_\_\_\_\_

Ing. Garcia Sanchez Roberto

## **AGRADECIMIENTO**

Doy gracias a Dios por siempre guiarme en todos los pasos que doy y por permitirme culminar con éxito esta experiencia dentro de la universidad, por ello agradezco a todos los docentes que formaron parte de este proceso y me impulsaron a cumplir esta meta, por brindarme sus conocimientos y experiencias para afianzarme en un futuro laboral y convertirme en una excelente profesional; de igual forma agradezco infinitamente a mi tutor por saberme guiar y apoyar en el desarrollo del presente trabajo e impulsarme a continuar en el proceso de aprendizaje. Por último, agradezco a mis amigos incondicionales que fueron parte fundamental en todos los logros obtenidos, por saberme apoyar en cada paso y hacer el camino más fructífero.

Gabriela Segovia Castro

## **DEDICATORIA**

Lograr esta meta, hace realidad el esfuerzo de muchos. Por tanto, dedico el presente trabajo y toda la educación que tengo y tendré a mi Madre Jenny por saberme apoyar, guiar y aconsejar y sacar todo lo mejor de mí, por ser mi ejemplo en nunca desistir; a mi Abuelita Paquita por ser una guía en mi educación y por enseñarme de que se pueden lograr muchas cosas con esfuerzo y dedicación, a mi Papá Richard por impulsarme a seguir creciendo y ser mejor persona, a mis hermanos que son mi centro y por quienes quiero ser mejor cada día para enseñarles a través del ejemplo; a mi primos Karol, Ariana y Gustavo que son parte fundamental de lo que soy y lo que quiero ser a futuro, a Abigail por ser esa luz que ilumina con su alegría, locura y amor cada uno de mis logros obtenidos. Para culminar les dedico con todo mi corazón todo lo logrado y lo que soy a todos los angelitos que hoy ya no me acompañan y no pueden celebrar conmigo en persona este triunfo, pero me cuidaron, apoyaron y guiaron en todo lo que me he propuesto y por siempre confiar en mí.

Gabriela Segovia Castro

# ÍNDICE GENERAL

<b>RESUMEN</b> .....	XIII
<b>ABSTRACT</b> .....	XIV
<b>INTRODUCCIÓN</b> .....	2
<b>CAPÍTULO I</b> .....	4
<b>1. EL PROBLEMA</b> .....	4
1.1. Ubicación del problema en un contexto.....	4
1.2. Planteamiento del Problema .....	4
1.3. Situación Conflicto .....	5
1.4. Delimitación del problema .....	6
1.5. Formulación del problema.....	6
1.6. Evaluación del problema .....	6
1.7. Objetivos de la investigación .....	7
1.7.1. Objetivo General .....	7
1.7.2. Objetivos Específicos.....	7
1.8. Alcance del problema.....	8
1.9. Justificación e importancia de la investigación .....	9
1.10. Pregunta de Investigación.....	10
1.11. Variables de la Investigación.....	10
<b>CAPÍTULO II</b> .....	11
<b>2. MARCO TEÓRICO</b> .....	11
2.1. Tecnologías de la Información y la Comunicación.....	11
2.2. Modelos de Aprendizaje.....	13
2.3. Modelo <i>Game Based-Learning</i> como tendencia de aprendizaje.....	17
2.4. Plataformas <i>Game based-learning</i> .....	19
2.4.1. <i>Game based-learning</i> alineado a la programación .....	20
2.5. Modelo de Calidad de Software .....	25
2.5.1. Evolución de los Modelos de Calidad a Nivel de Producto .....	27
2.5.2. Norma ISO 25000 .....	28
2.5.2.1. Estructura del Modelo de la ISO 25000 .....	31
<b>CAPÍTULO III</b> .....	36
<b>3. MARCO METODOLÓGICO</b> .....	36
3.1. Tipo de investigación.....	36
3.2. Identificación de la población y muestra .....	37
3.2.1. Población .....	37
3.2.2. Muestra.....	38

3.3.	Técnicas e instrumentos para el levantamiento de información.....	39
3.4.	Tratamiento de la Información.....	40
3.5.	Análisis de resultados de las encuestas.....	40
3.6.	Análisis de resultados de las entrevistas.....	61
<b>CAPÍTULO IV.....</b>		<b>67</b>
<b>4. PROPUESTA.....</b>		<b>67</b>
<b>MODELO DE EVALUACIÓN PARA PLATAFORMAS GAME BASED-LEARNING</b>		<b>67</b>
4.1.	Campo de Aplicación.....	67
4.2.	Características del Modelo de Evaluación.....	67
4.3.	Normas Referenciales.....	68
4.4.	Estructura del Modelo de Evaluación.....	69
4.5.	Características para la aplicación del modelo.....	69
4.6.	Criterios de Inclusión y Exclusión de las plataformas.....	71
4.7.	Criterios para la determinación de las fórmulas.....	71
4.8.	Definición del Nivel de Importancia de las características.....	72
4.9.	Métricas y Fórmulas de Evaluación.....	73
4.10.	Pasos para aplicar el modelo de evaluación propuesto.....	89
4.10.1.	Definir requisitos de evaluación.....	89
4.10.2.	Definir el nivel de importancia por cada característica.....	89
4.10.3.	Establecer valores constantes para algunas fórmulas.....	89
4.10.4.	Llenar el modelo con los datos Registrar los Resultados.....	89
4.11.	Análisis del Modelo de Evaluación Propuesto.....	90
<b>5. CONCLUSIONES Y RECOMENDACIONES.....</b>		<b>94</b>
5.1.	Conclusiones.....	94
5.2.	Recomendaciones.....	95
<b>6. BIBLIOGRAFÍAS.....</b>		<b>96</b>
<b>ANEXOS.....</b>		<b>101</b>

# ÍNDICE DE FIGURAS

Figura 1. Modelos de aprendizaje.....	14
Figura 2. Actores del modelo tradicional .....	15
Figura 3. Ventajas del modelo game based-learning.....	18
Figura 4. Entorno de Netbeans .....	22
Figura 5. Interfaz de programación de blockly.....	23
Figura 6. Interfaz de programación en texto de CODEDARENA.....	23
Figura 7. Interfaz de salida de CODEDARENA.....	24
Figura 8. Modelo de Calidad en uso según la ISO 25000 .....	26
Figura 9. Modelos de Calidad a Nivel de Producto .....	27
Figura 10. Evolución de la norma de calidad de software .....	28
Figura 11. Familia de la Norma ISO 25000 .....	29
Figura 12. Estructura de la ISO 25000 .....	31
Figura 13. Modelo de calidad de software según la ISO 25010 .....	33
Figura 14. Nivel de conocimientos previos al ingreso a la universidad .....	41
Figura 15. Cumplimiento de expectativas de las herramientas de programación.....	42
Figura 16. Nivel de dificultad al cursar la primera materia de programación....	44
Figura 17. Nivel de dominio después de cursar la primera materia de programación.....	45
Figura 18. Dificultades al aprender programación.....	46
Figura 19. Aprender a través de juegos ayuda al dominio de la programación	47
Figura 20. Plataformas game based-learning.....	49
Figura 21. Herramientas de programación.....	51
Figura 22. Eficiencia de las herramientas para fomentar el aprendizaje activo	52
Figura 23. Nivel de motivación que incentivan las herramientas en la actualidad .....	54
Figura 24. Habilidades que deben fomentar las plataformas para programar..	55
Figura 25. Características que deberían tener las plataformas .....	57
Figura 26. Nivel de conocimiento de la existencia de normas de calidad de software .....	58
Figura 27. Importancia de la certificación de las herramientas bajo una norma de calidad .....	59

## ÍNDICE DE TABLAS

Tabla 1. Tendencias tecnológicas de enseñanza-aprendizaje .....	16
Tabla 2. Clasificación de los tipos de lenguaje de programación .....	21
Tabla 3. Población de la investigación .....	37
Tabla 4. Nivel de Conocimiento en Programación previo al ingreso la Universidad.....	41
Tabla 5. Cumplimiento de expectativas de las herramientas de programación.....	42
Tabla 6. Nivel de dificultad al cursar la primera materia de programación .....	43
Tabla 7. Nivel de dominio después de cursar la primera materia de programación.....	45
Tabla 8. Dificultades al aprender programación .....	46
Tabla 9. Aceptación del dominio de la programación a través de juegos.....	47
Tabla 10. Plataformas game based-learning.....	49
Tabla 11. Herramientas de programación .....	50
Tabla 12. Eficiencia de las herramientas para fomentar el aprendizaje activo .	52
Tabla 13. Nivel de motivación que incentivan las herramientas en la actualidad .....	53
Tabla 14. Habilidades que deben fomentar las plataformas para programar ...	55
Tabla 15. Características que deberían tener las plataformas de programación.....	56
Tabla 16. Nivel de conocimiento de la existencia de normas de calidad de software.....	58
Tabla 17. Nivel de conocimiento de la existencia de normas de calidad de software .....	59
Tabla 18. Características del modelo propuesto .....	70
Tabla 19. Plataformas game based-learning de programación de texto .....	71
Tabla 20. Rangos de importancia de características del modelo propuesto ....	72
Tabla 21. Definición de métricas y fórmulas de la Adecuación Funcional .....	74
Tabla 22. Definición de métricas y fórmulas de la Eficiencia de desempeño ...	75
Tabla 23. Definición de métricas y fórmulas de la Compatibilidad .....	77
Tabla 24. Ponderación de cumplimiento de características .....	77
Tabla 25. Definición de métricas y fórmulas de la Usabilidad .....	78

Tabla 26. Principios del modelo heurístico de la Interfaz de usuario.....	80
Tabla 27. Sistemas Operativos más usados en Ecuador.....	81
Tabla 28. Definición de métricas y fórmulas de la Fiabilidad.....	81
Tabla 29. Definición de métricas y fórmulas de la Seguridad.....	83
Tabla 30. Formas de Recuperación de datos.....	84
Tabla 31. Formas de Autenticación de usuario .....	85
Tabla 32. Definición de métricas y fórmulas de la Mantenibilidad .....	86
Tabla 33. Definición de métricas y fórmulas de la Portabilidad .....	88
Tabla 34. Niveles de aceptación de las plataformas game based-learning.....	90
Tabla 35. Clasificación de las plataformas evaluadas según nivel de aceptación .....	91
Tabla 36. Resumen de Evaluación de las plataformas game based-learning..	92

## RESUMEN

Las TIC han contribuido de forma significativa en la evolución de las herramientas que apoyan las tendencias de aprendizaje, por ello la presente investigación se centra en la evaluación de plataformas alineadas a la enseñanza de la programación en la educación superior que aplican la metodología *game based-learning* para la construcción de un modelo de calidad a nivel de producto inspirado en la norma ISO 25000 que permita evaluar la calidad interna y externa. Mediante una metodología de enfoque mixto, a través de instrumentos de recolección de datos aplicados a estudiantes y docentes de la carrera de Ingeniería, se analizan plataformas de programación basadas en textos, y contrastando la información analizada se definieron las características técnicas que debe de contar una plataforma que se aplique en procesos pedagógicos. De esta base parte el diseño del modelo de calidad a nivel de producto para evaluar plataformas *game based-learning* basado en diversas características internas como externas, las mismas que fueron evaluadas por medio de fórmulas claras y justas para definir por medio de un resultado final la efectividad de las mismas; lo que permitió definir la óptima para aplicarla en procesos de enseñanza de la educación superior y por ende fomentar habilidades que logren aumentar el nivel competitivo del estudiante.

**Palabras Clave:** Game Based-Learning, ISO 25000, Programación, TIC.

## ABSTRACT

ICT has contributed significantly to the evolution of tools that support learning trends. This research therefore focuses on the evaluation of platforms aligned to the teaching of programming in higher education that apply the game based-learning methodology for the construction of a product-level quality model inspired by the ISO 25000 standard that allows evaluation internal and external quality. Using a mixed approach methodology, through data collection tools applied to students and teachers in the engineering career, text-based programming platforms are analyzed, and by contrast the analyzed information, the technical characteristics that a platform that is applied in pedagogical processes must be defined. From this basis is the design of the quality model at the product level to evaluate game based-learning platforms based on various internal and external characteristics, which were evaluated by means of clear and fair formulas to define by a final result the effectiveness of them; this allowed the best to be defined for application in higher education teaching processes and thus to foster skills that will increase the competitive level of the student.

**Key words:** Game Based-Learning, ISO 25000, Programming, ICT.

# INTRODUCCIÓN

Las Tecnologías de la Información y la Comunicación (TIC) han evolucionado con el paso de los años dando apertura a la creación de múltiples plataformas alineadas a distintos objetivos sociales, contribuyendo a transformar la educación del siglo XXI, que como menciona la UNESCO (2019) en el objetivo 6 del Informe de Seguimiento de la EPT en el Mundo 2019, referente a la calidad de la educación: “es indispensable mejorar todos los aspectos cualitativos de la educación para garantizar parámetros más elevados, para conseguir resultados de aprendizaje reconocidos y mensurables” (p.114).

Resulta complicado aplicar nuevos modelos de aprendizaje si se desconoce la calidad de la enseñanza que brinda cada una de las plataformas, que en su mayoría están basadas en el modelo e-learning que solo facilita una experiencia de capacitación a distancia a través de una realidad tecnológica. Es ahí donde nace un nuevo modelo denominado game based-learning que permite aplicar las habilidades en un entorno simulado que resulta en una mejor interiorización del conocimiento a través de la gamificación que como menciona Deterding (2012) en el artículo *Gamification: Designing for Motivation*, se basa en el uso de elementos del diseño de videojuegos en contextos que no son de juego, para hacer que un producto, servicio o aplicación sea más divertido, atractivo y motivador; aplicando la filosofía del “aprender haciendo” (en inglés, *learning by doing*), puesto que se he documentado bien la eficacia del aprendizaje práctico.

Por tal motivo la presente investigación desarrolló la problemática referente al desconocimiento que se tiene sobre los parámetros que validan la calidad de las plataformas en estudio; causas y efectos que determinan la necesidad de ampliar el presente estudio.

Es por esto necesario respaldar de manera teórica los modelos de aprendizaje con sus definiciones, características y cualidades para así proceder al uso de la norma ISO 25000, de tal manera que se convierta en una normativa que garantice la calidad de las plataformas game based-learning en un conjunto

estructurado de propiedades tales como funcionalidad, fiabilidad, usabilidad, eficiencia, mantenibilidad y portabilidad; apuntando a la eficiencia en el ámbito educativo.

Es importante destacar que la metodología aplicada permitió triangular los resultados para comprender la efectividad de las herramientas que se emplean en la actualidad y cuáles son las características que consideran necesarias los estudiantes como los docentes para la aplicación de nuevas plataformas.

Es por ello que se presenta el diseño de un modelo de calidad de software para evaluar plataformas game based-learning basado en características alineadas a dicha tendencia de aprendizaje, y de esta manera de acuerdo a los resultados obtenidos aplicarlas en procesos de educación superior.

# CAPÍTULO I

## 1. EL PROBLEMA

### 1.1. Ubicación del problema en un contexto

Es habitual que muchos educadores de nivel superior desconozcan sobre la calidad de las plataformas bajo el modelo *game based-learning* que sirven como apoyo a la enseñanza relacionada a la programación. Las herramientas tecnológicas que son empleadas en la educación son limitadas por muchos factores entre ellos sociales y económicos, ocasionado un retroceso en el área de innovación dentro de las estrategias de enseñanza empleadas por los docentes, dando apertura a múltiples razones del escaso uso de estas nuevas tecnologías, como la dificultad para conseguir ediciones gratuitas, escasa disponibilidad de recursos de hardware, desconocimiento de herramientas, poca preparación en el uso de las mismas, desinterés por aplicar nuevas formas de enseñanza, entre otros. Haciendo hincapié lo que menciona Estrada & Ovide (2011), que son sólo herramientas y, como tales, hay que considerarlas, sin olvidar que su función es apoyar, como cualquier otro medio, a cumplir objetivos pedagógicos.

### 1.2. Planteamiento del Problema

En la actualidad, con la evolución de las TIC a nivel mundial han surgido un sin número de plataformas inspiradas en el modelo *game based-learning*, debido a la creciente expansión de la gamificación dentro del ámbito educacional; como mencionan Lozada Ávila y Betancur Gómez (2017), el juego ha pasado a convertirse en un activador en la atención y surge como alternativa para complementar los esquemas de enseñanza tradicional.

La problemática en la que basa la presente investigación surge a partir del desconocimiento que se tiene sobre los parámetros que validan la calidad de las plataformas en estudio, dificultando su potencial uso e importancia; considerando mucho más el avance tecnológico, científico y social.

Es notorio destacar que estas plataformas en estudio son empleadas para agilizar el aprendizaje relacionado al área de la programación y a fomentar el desarrollo de habilidades del pensamiento computacional, por lo tanto, se requiere determinar e identificar cuáles son más prácticas, sin restar importancia al modelo de aprendizaje tradicional que ya ha logrado llegar a su caducidad.

### **1.3. Situación Conflicto**

El aprendizaje tradicional ha evolucionado con el uso de las plataformas de enseñanza autónoma enfocadas en el área de la programación, pero las mismas no han sido evaluadas bajo parámetros de calidad, ocasionando que no sean incorporadas como herramientas de apoyo en la educación actual.

Existen plataformas que no cuentan con una buena adecuación funcional es decir que no proporcionan las funciones requeridas y que por ende no cumplen con los requerimientos pedagógicos relacionados a la programación, disminuyendo su valor de aceptación en el mercado educativo al no cumplir con adecuados objetivos de aprendizaje; esta circunstancia va ligada a la poca usabilidad que brindan puesto a la deficiencia para ser entendidas o al poseer una interfaz poco atractiva dando como resultado deficiencias para operarlas o controlarlas.

Así también estas plataformas no evidencian una óptima eficiencia en el desempeño, siendo evaluadas bajo varios parámetros, tales como los tiempos de respuesta y la cantidad de recursos que emplean al momento de realizar algún proceso bajo lineamientos determinados. Abriendo paso a otras aristas como la fiabilidad puesto que no están diseñadas para responder bajo escenarios de fallos y recuperación de datos; disminuyendo el nivel de seguridad y la capacidad de proteger la información y los datos de los usuarios, que en su gran mayoría no cuentan con un control de autenticación.

Otro de los puntos es la incompatibilidad para coexistir y compartir recursos en el mismo entorno, dando como resultado la imposibilidad del

intercambio de información. Como también la incapacidad de ser adaptadas de forma eficiente a distintos entornos, sean estos de hardware o software como la facilidad de ser instaladas o desinstaladas, representando la mantenibilidad de las plataformas a través de actualizaciones sean estas por acciones correctivas o de mejoras.

Todas estas observaciones que han sido declaradas con anterioridad están basadas en los estándares de calidad de la ISO 25000, las cuales representan el punto de inflexión en la aplicabilidad de las plataformas *game based-learning* en el ámbito educativo.

#### **1.4. Delimitación del problema**

Las plataformas educativas *game based-learning* son en su gran mayoría *freeware*, es decir, su distribución no tiene costo con su funcionalidad completa y sin limitaciones temporales, pero las mismas no son empleados en ámbitos educativos superiores en el área de la programación, puesto a su nula evaluación y certificación mediante estándares de calidad. Ocasionado la escasa aplicabilidad de nuevas herramientas de aprendizaje remoto para reforzar u acompañar el aprendizaje basado en el modelo tradicional de enseñanza.

#### **1.5. Formulación del problema**

¿Cuáles de las plataformas basadas en el modelo *game based-learning* cumplen con la norma ISO 25000 para aplicarla en el ámbito educativo de la programación y lograr resultados mensurables?

#### **1.6. Evaluación del problema**

El problema actual que se ha detectado y el sustento para la realización del presente trabajo de investigación se centra en aspectos que darán la apertura a una óptima evaluación de la problemática. Iniciando con la relevancia que tiene la aplicabilidad de plataformas certificadas bajo estándares de calidad en la educación superior. Las cuales han sido delimitadas de acuerdo al criterio del área de conocimiento para las que fueron

diseñadas, tal como es la programación; un proceso que consiste en escribir instrucciones que son interpretadas para realizar un software determinado, dándole un enfoque de utilidad para estudiantes de la educación superior o entusiastas que deseen conocer sobre la rama.

Origina entonces una nueva problemática y por ende original, puesto que en el modelo de enseñanza tradicional es escasa la aplicación e innovación de plataformas *game based-learning* dado al desconocimiento con respecto a la utilidad y eficiencia de las mismas enfocadas al área de la programación y al desarrollo del pensamiento crítico, en donde se deja de lado el hecho del uso de las inteligencias múltiples que como lo menciona Gardner (2005) representa el modo en el que la gente desarrolla nuevas capacidades y habilidades para aplicarlos en su vida diaria.

Se evidencia un serio problema al referirse a la deficiencia de conocimientos o facilidad para comprender el arte de la programación por parte de los estudiantes. Es por tanto factible, lograr resultados a través de la aplicación del trabajo propuesto, considerando que los recursos a emplearse serán de tipo tecnológicos, siendo estos de libre acceso.

## **1.7. Objetivos de la investigación**

### **1.7.1. Objetivo General**

Analizar las plataformas de aprendizaje *game based-learning* alineadas al área de la programación a través de los estándares de calidad de la norma ISO 25000 para aplicarlas en el ámbito educativo.

### **1.7.2. Objetivos Específicos**

- Evaluar el modelo *game based-learning* para determinar su efectividad en el aprendizaje autónomo en el área de la programación.
- Identificar y caracterizar plataformas *game based-learning* para la aplicación en el proceso de enseñanza -aprendizaje alineados a la programación.

- Diseñar un modelo de evaluación basado en la ISO 25000 para plataformas *game-based learning*.
- Analizar las plataformas *game based-learning* según los lineamientos de la norma ISO 25000, para determinar qué plataformas son adecuadas para el desarrollo de procesos de aprendizaje.

### **1.8. Alcance del problema**

El alcance de la investigación está dividido en dos instancias por ende se emplea un enfoque cualitativo y cuantitativo, es decir será de tipo exploratorio debido a que se buscarán las plataformas *game based-learning* enfocadas a la programación que no han sido validadas mediante un proceso estandarizado, y de esta manera poder familiarizarse con estas nuevas herramientas para así lograr información que posibilite obtener datos que permitan investigaciones más completas y poder presentar teorías, conceptos o premisas a futuro, con las consignas que fortalezca el uso de estas plataformas en la educación.

La investigación también utiliza el alcance de tipo descriptivo, puesto que se describirán las características para especificar con precisión su funcionalidad, uso y aplicación en beneficio de la educación del siglo XXI; siendo de suma importancia el detalle de cada una de las plataformas para llevar los datos cualitativos a cuantitativos al momento de ingresar los datos al modelo.

Es así como tendrá un alcance aplicado, debido a que se evaluarán las plataformas elegidas para la validación de calidad de las mismas y a su vez determinar la efectividad del modelo con respecto a características internas y externas de plataforma *game based-learning*, permitiendo relacionar las dos variables de investigación como lo son las plataformas *game based-learning* y la norma ISO 25000 para entender la efectividad dentro de los procesos educativos de enseñanza-aprendizaje.

## 1.9. Justificación e importancia de la investigación

El enfoque de la investigación está centrado en la aplicación de los estándares de calidad de software con el apoyo de la norma ISO 25000 para evaluar en ocho características las plataformas *game based-learning* enfocadas al aprendizaje de la programación, con la finalidad de conocer y discernir las plataformas que cumplen con los objetivos descritos en la norma. Para de esta manera aplicarlas según su porcentaje de calidad, de acuerdo con los resultados obtenidos, dentro del proceso enseñanza-aprendizaje relacionado al área mencionada. Dado a que en actualidad y con la evolución de nuevas modalidades de enseñanza es imprescindible proporcionar nuevas herramientas tecnológicas de calidad y excelencia para que se logre el “*engagement*” esperado con esta área.

Se abre paso a que se empleen estas plataformas tanto dentro de la educación básica como la superior, reflejando innovación en el proceso educativo, y como consecuencia los estudiantes como tal logren incrementar su interés para la adquisición de nuevos conocimientos sin limitaciones; enfatizando la premisa que *play is the best way to learn* (jugar es la mejor manera de aprender), puesto que como lo menciona Vattimo (2006) de esta forma se “incita y excita la capacidad de superación, y pone a prueba el poder y potencial, o traza una suerte de horizonte en referencia al cual se expone la libertad” (p.175).

Desde el punto de vista científico esta investigación entregará un aporte que sirva como base para un posterior análisis y evaluación de calidad de plataformas alineadas a los demás modelos de aprendizaje que han surgido con el desarrollo de las TIC, siendo de vital importancia para que demás investigadores comparen los resultados de la presente investigación y el modelo de calidad propuesto con algún otro modelo para definir las métricas aplicables de forma universal en otras plataformas.

Por tanto, la evaluación de acuerdo con las métricas del modelo propuesto inspirado en la norma de calidad ISO 25000 servirá de sustento

para una posible aplicabilidad de las plataformas *game based-learning* en procesos educativos enfocados a la programación.

### **1.10. Pregunta de Investigación**

El análisis de las plataformas educativas basadas en el modelo *game-based learning* mediante la norma ISO 25000 permitirá conocer su utilidad y resultados en el ámbito educativo de la programación.

### **1.11. Variables de la Investigación**

Como variable dependiente de la investigación se declaran las plataformas *game based-learning* que consisten en aprender a través del juego, alineadas a objetivos claros de aprendizaje. Las cuales son vistas como herramientas no fiables o poco certeras que apoyen el proceso de aprendizaje, siendo catalogadas como juegos de ocio.

Y como variable independiente se presenta la norma ISO 25000 que permitirá construir de acuerdo con las métricas, la calidad del software; ayudando a determinar su aplicabilidad en el ámbito educativo y por sobre todo para lograr resultados mensurables.

Es así como surge una nueva variable interviniente puesto que los procesos educativos alteran la relación entre las plataformas *game based-learning* y la ISO 25000, es decir la variable independiente y dependiente; puesto que depende de la aceptación o inclusión de las mismas dentro de estos procesos así estén certificadas por el estándar de calidad.

## CAPÍTULO II

### 2. MARCO TEÓRICO

#### 2.1. Tecnologías de la Información y la Comunicación

La tecnología se originó desde la época del paleolítico cuando el hombre comenzó a formar tecnología, al convertir los recursos naturales en herramientas simples (Fraoli et al. 1999). A inicios del siglo VI a.C los filósofos Platón y Protágoras coincidieron que la utilidad no es saber sino más bien es arte que en griego es *techné*, y que la única forma de aprender era con la práctica y la experiencia, combinado con el vocablo griego *logos* que como lo define Heráclito es la inteligencia que dirige, ordena y da armonía al devenir de los cambios; formándose así la palabra tecnología.

Tal como Migdely et al. (2007) mencionan: “La tecnología es el conjunto de conocimientos científicos y empíricos, habilidades, experiencias y organización requeridos para producir, distribuir, comercializar y utilizar bienes y servicios”(p.2). Sintetizando que la tecnología es el arte de ordenar y asimilar conocimientos a través de la práctica, resaltando que al ser de forma virtual se busca a través de la evolución de las Tecnologías de la Información y Comunicación (TIC), nuevas herramientas que faciliten la realización de procesos educativos. Así como lo referencian de distintas formas algunos autores:

Conjunto de herramientas, soportes y canales desarrollados y sustentados por las tecnologías que permiten la adquisición, producción, almacenamiento, tratamiento, comunicación, registro y presentación de informaciones, en forma de voz, imágenes y datos, contenidos en señales de naturaleza acústica, óptica o electromagnética a fin de mejorar la calidad de vida de las personas. (Ávila, 2013, p.222)

Es así como la aparición de nuevas metodologías de enseñanza-aprendizaje van de la mano con las herramientas que permitan el tratamiento de la información, y que por ende ayuden al desempeño de la sociedad en

cualquier área. Así como también una definición más específica la realizó el Programa de las Naciones Unidas para el Desarrollo (2002) donde se menciona que:

“Las TIC se conciben como el universo de dos conjuntos, representados por las tradicionales Tecnologías de la Comunicación (TC) –constituidas principalmente por la radio, la televisión y la telefonía convencional– y por las Tecnologías de la Información (TI) caracterizadas por la digitalización de las tecnologías de registros de contenidos (informática, de las comunicaciones, telemática y de las interfaces)”. (p.10)

Al resumir las estas definiciones, las TIC contemplan dos aristas, las herramientas que permiten tratar la información a través de modernas tecnologías y el medio tecnológico en el que se distribuye dicha información.

Para comienzos del siglo XXI, la tecnología ha impulsado la transformación social para que el ser humano goce de la bondad de aprender y producir conocimientos con la incorporación de las TIC en el ámbito de la educación, aperturando la brecha tecnológica para la aplicación y automatización de procesos educativos, evidenciando una gran implicancia en la enseñanza tradicional como lo ratifica Dolly Cristina & Rafael G. (2005) en que las TIC en relación con lo educativo hacen parte de cambios de gran calado en la escuela y en la sociedad en general, dado a que según Tello (2007) contemplan toda forma de tecnología usada para crear, almacenar, intercambiar y procesar información en sus varias formas, tales como datos, conversaciones de voz, imágenes fijas o en movimiento, presentaciones multimedia y otras formas, incluyendo aquéllas aún no concebidas.

Es casi común decir que las TIC han causado un profundo impacto en todo el mundo, en la ciencia, en la tecnología, en la educación y por consiguiente en la sociedad (Jacovkis, 2011). Siendo en la actualidad no concebible una normalidad sin internet, sin siquiera mencionar los dispositivos

informáticos, softwares educativos, plataformas empresariales y demás que han transformado la nueva normalidad.

En donde las herramientas tecnológicas representan el motivo de la evolución y modificación del proceso enseñanza-aprendizaje, de tal manera que la sociedad de la información se vuelva dependiente de los recursos tecnológicos de aprendizaje, representado un cambio radical en la enseñanza tradicional y los modelos de aprendizaje.

## **2.2. Modelos de Aprendizaje**

Para la sociedad un modelo “es la imagen o representación del conjunto de relaciones que definen un fenómeno con miras a un mejor entendimiento”, tal como lo menciona Ortíz Ocaña (2013, p.63). En tal virtud los modelos revelan diferentes funciones que van conectadas en base a la interpretación, el diseño y el ajuste revelando una base constructivista y transformadora. Por tanto, existen distintas tipologías de los modelos que han surgido hasta la actualidad relacionados con la actitud del docente en el aula, el perfil, los resultados que se otorgan, los insumos, productos y procesos que se obtienen, la calidad, la práctica reflexiva y la didáctica.

Es así como el modelo educativo de aprendizaje incluye modelos pedagógicos y didácticos, considerando que un modelo pedagógico involucra procesos organizados que fortalecen la formación de la personalidad y su relación en la sociedad entre el educador y el educando.

Dentro de los diferentes modelos de aprendizaje tal como se representan en la Figura 1, originados en las escuelas tradicionales antiguas y del pensamiento contribuyen a la formación del ser humano revelando su objetividad y su base científica representadas en los paradigmas de los cuales se originaron.

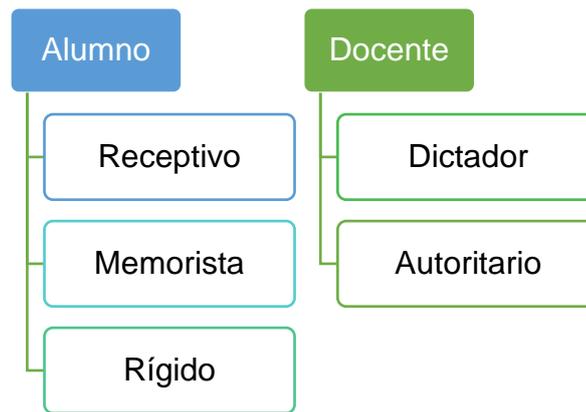


*Figura 1. Modelos de aprendizaje.*

En el contexto de la educación del Ecuador se ha evidenciado la aplicación del modelo de aprendizaje tradicional ligada a la educación pública, donde el sujeto o educando es formado según el deseo del maestro que reproduce el conocimiento que ha considerado valioso; siendo autoritario, rígido, nada creativo provocando inseguridad, poco interés y escasa participación personal.

Se detalla en la Figura 2 el modelo pedagógico tradicional, en el que es necesario identificar a los actores involucrados tanto directa como indirectamente en el modelo enseñanza-aprendizaje.

El alumno es el principal involucrado en el modelo descrito, puesto que es el receptor de la información que ha sido facilitada por el docente, convirtiéndolo, en un estudiante receptivo, memorista y rígido, existiendo una relación vertical con la posición del docente que se convierte en un dictador de clases autoritario, mecánico en donde prima el proceso de enseñanza sobre el de aprendizaje.



*Figura 2. Actores del modelo tradicional*

En la actualidad se evidencia escasa aplicabilidad puesto que los resultados son poco eficaces, nada pertinentes ni viables dado a la presencia de una currícula lineal.

Schank (2007), presidente de Socratic Arts menciona que, el sistema de enseñanza tradicional se podría resumir en donde el docente da la clase y enseña teoría y los alumnos, como mucho, toman apuntes. Y como no recuerda lo que se les dijo, los evalúan. Pero poco después de pasar esos exámenes, olvidan todo. Por lo tanto, es notorio que el modelo pedagógico tradicional está quedando caduco frente a nuevos modelos que se relacionan con herramientas tecnológicas.

Es así como el modelo tradicional se ha visto envuelto en un giro muy significativo con la contribución de las TIC, a causa del sin número de herramientas tecnológicas que han surgido y por ende han dado paso a nuevos modelos basados en las necesidades de los involucrados en el proceso enseñanza-aprendizaje.

Tal es el caso del modelo tecnológico, que tiene como característica principal ser muy estructurado, para lograr resultados de aprendizaje bien definidos. Pero es necesario discernir que este nuevo modelo convierte al estudiante en poco creativo y sistemático, en el que no se contempla la participación ni la iniciativa del alumno. Conllevando a un desnivel de las habilidades y aptitudes que van adquiriéndose durante el proceso de aprendizaje.

De tal forma, han surgido nuevas tendencias que cubren las necesidades que el docente y el estudiante requieren, para que el proceso de enseñanza-aprendizaje cumpla con todas las áreas del conocimiento.

Siendo la característica general que reúnen estas tendencias la aplicación de la tecnología como herramienta didáctica durante el proceso enseñanza-aprendizaje. Se describe en la Tabla 1 el enfoque que cada una de estas tendencias posee.

*Tabla 1. Tendencias tecnológicas de enseñanza-aprendizaje*

<b>Tendencia</b>	<b>Característica</b>
Electronic-learning	Basado en la teleformación o aprendizaje en línea, en donde se emplea Internet y las TIC como herramientas para comunicarse e interactuar en un proceso de formación.
Mobile-learning	Se emplea dispositivos móviles como herramientas siendo la evolución del e-learning dando paso a la utilización de smartphones o tablets.
Ubiquitous -learning	Contempla la unión de las 2 tendencias descritas anteriormente en la que permite a un alumno aprender en cualquier momento y en cualquier lugar.
Blended-learning	Se trata de la combinación de entornos virtuales y físicos en el proceso de aprendizaje.
Game based-learning	Fundamentada en una técnica de aprendizaje en la cual los contenidos teóricos son presentados por medio de un videojuegos.
Transformative-learning	Aprendizaje que se lleva a cabo de forma presencial o en las plataformas online de aprendizaje, dotando contenidos para dar la sensación de que se aprende por sí mismo, y no está siendo guiado.
Micro-learning	Basada en una formación comprimida en pequeñas cápsulas digitales que permite adquirir conocimientos de forma muy rápida.

Elaboración propia tomada de varios autores

### **2.3. Modelo *Game Based-Learning* como tendencia de aprendizaje**

Los juegos se han utilizado como herramienta de aprendizaje durante siglos, tal es el caso del ajedrez que se usó para enseñar pensamiento estratégico desde la Edad Media. Siendo la forma en la que se pueden interiorizar los conocimientos puesto que como lo referencia Dichev & Dicheva (2017) permiten aumentar la motivación y el compromiso de los alumnos al incorporar elementos de diseño de juegos en entornos educativos.

Ratificando el hecho de que aprender a través del juego aumenta los niveles de interés y desarrollo de habilidades alineadas a todas las áreas del conocimiento, dado a que las actividades tienen un juego en su núcleo, ya sea como la principal actividad o como estímulo para otras actividades relacionadas, y tener como resultado un aprendizaje deseado o incidental (Denham et al., 2016).

Es así como el modelo *game based-learning* se sustenta en la aplicación de herramientas gamificadas de aprendizaje puesto que según estudios y teorías se ha evidencia bien la eficacia del aprendizaje práctico que como menciona Knutas et al. (2014) los objetivos principales son mejorar ciertas habilidades, introducir objetivos que le den un propósito al aprendizaje, involucrar a los estudiantes, optimizar el aprendizaje, apoyar el cambio de comportamiento y socializar.

Por tanto, a continuación, se ejemplifica en la Figura 3 las ventajas que la diferencian de las demás tendencias de aprendizaje en donde se emplean materiales tecnológicos para su aplicabilidad.



*Figura 3. Ventajas del modelo game based-learning.*

Es así como este modelo basa sus pilares en el engagement que se puede lograr con los estudiantes puesto que está orientado a ayudarlos a razonar a través de logros que se consiguen en los juegos, tal como (Boeker et al. (2013) menciona que la activación de los alumnos apoya el proceso de aprendizaje no sólo en el dominio cognitivo, sino también en los dominios afectivos y psicomotores permitiendo de tal manera un aprendizaje activo y por ende darles el control de su aprendizaje puesto a la libertad que tienen para acceder a plataformas interactivas basadas en el juego dando apertura a la brecha de la alfabetización digital.

Proporcionando al facilitador materiales didácticos para que se apliquen los conocimientos impartidos, potenciando la creatividad y la imaginación del estudiante, permitiendo al mismo tiempo que se desarrollen habilidades relacionadas a otras áreas de las inteligencias múltiples tal como las relaciones sociales.

## **2.4. Plataformas *Game based-learning***

La aparición de plataformas diseñadas bajo el modelo game based-learning han aumentado de manera exponencial durante los últimos años en donde la aplicación de herramientas tecnológicas dentro del proceso enseñanza aprendizaje ha evolucionado y ha tomado más fuerza debido a la innovación educativa. Siendo empleados puesto a que refieren a la utilización de juegos como herramientas conductoras y de apoyo al aprendizaje, la asimilación y evaluación de conocimientos. Siendo las características de las plataformas game based-learning las que se referencian a continuación.

- Atractivo
- Evolutivo
- Motivador
- Inspirador
- Reflexivo
- Práctico
- Colaborativo

Justificando la importancia de que las plataformas cubran las características anteriormente mencionadas para que sirvan como pilares para el desarrollo de habilidades como para la interiorización y puesta en práctica de los contenidos impartidos por un facilitador de información.

Es así como hay disponible un sin número de plataformas enfocadas a distintas áreas de la ciencia, en donde dependiendo de los objetivos de aprendizajes han sido diseñadas bajo distintos enfoques. Es decir que dependiendo del público objetivo y del contenido a enseñar se han empleado estrategias alineadas al diseño de dichas plataformas para su posterior gamificación y para que de tal manera cumplan con el objetivo de su implementación.

#### **2.4.1. *Game based-learning* alineado a la programación**

Enfocarse en un área del conocimiento en específico para conocer las plataformas que facilitan la enseñanza y aprendizaje es conocer el enfoque que deben tener las mismas. Por lo cual se referencia la definición de programación para tener claro a donde deberían apuntar las plataformas en estudio.

Según Mathieu (2014) la programación es un conjunto de tareas que se deben cumplir, a fin de que el código que se escribió resulte correcto y robusto, y cumpla con el objetivo o los objetivos para los que fue creado. Es decir, refiere a la acción de crear software a través de código fuente, el cual se basa en instrucciones para que el ordenador ejecute algún programa, que como menciona Ureña (2011), es una secuencia finita de dígitos y caracteres, directamente legible por las personas y por el ordenador.

Es así como crear un software o programa es el resultado de la acción de programar que deriva de la programación, la misma que apunta a indicarle al ordenador lo que tiene que hacer, por medio de secuencias de instrucciones. Pero es de vital importancia definir que programar involucra el conocimiento de lenguajes de programación puesto que no se escribe de cualquier manera, sino más bien a través de “idiomas” para facilitar la creación de programas, referenciando a (Ureña, 2011) que menciona los lenguajes de programación son un conjunto de reglas o normas que permiten asociar a cada programa correcto un cálculo que será llevado a cabo por un ordenador (sin ambigüedades). Por tanto, es un lenguaje formal que emplea símbolos, números y palabras que permiten que el programador escriba una serie de instrucciones en forma de algoritmos.

Pero es necesario mencionar que existen varios lenguajes de programación que atienden distintos criterios, así como se lo describe en la Tabla 2.

Tabla 2. Clasificación de los tipos de lenguaje de programación

Criterio	Tipos
Proximidad al Lenguaje de Máquina	Bajo Nivel
	Medio Nivel
	Alto nivel
Propósito del Lenguaje	General
	Específico
Orientación	Procedimiento
	Problema
	Objeto

Elaboración propia tomada de varios autores.

La clasificación descrita con anterioridad permite tener claro que las plataformas *game based-learning* que se evaluarán en la presente investigación corresponden a la enseñanza-aprendizaje de lenguajes de alto nivel puesto a su sencillez, simplicidad y legibilidad permitiendo una comprensión de la codificación. Tenido como características los lenguajes de alto nivel por seguir un conjunto de reglas sintácticas que se rigen mediante un léxico previamente definido facilitando que el ordenador interprete las líneas de código.

Por tanto, es imprescindible que el programador tenga una experticia con las normas para una correcta escritura de código, como se mencionó se debe de conocer la sintaxis, estructura, terminologías reservadas y demás componentes que conforman un lenguaje de programación. Siendo necesario que se cuente con entornos de programación para el desarrollo de un programa, los mismos que como tareas básicas permiten la edición de código, compilación, ejecución y depuración.

Pero los entornos no son amigables ni cuentan con una interfaz visual atractiva para las personas que están iniciando en el mundo de la programación, siendo estos rústicos y de difícil comprensión, imposibilitando que como estudiantes logren los resultados esperados. Tal como se ejemplifica en la Figura 4.

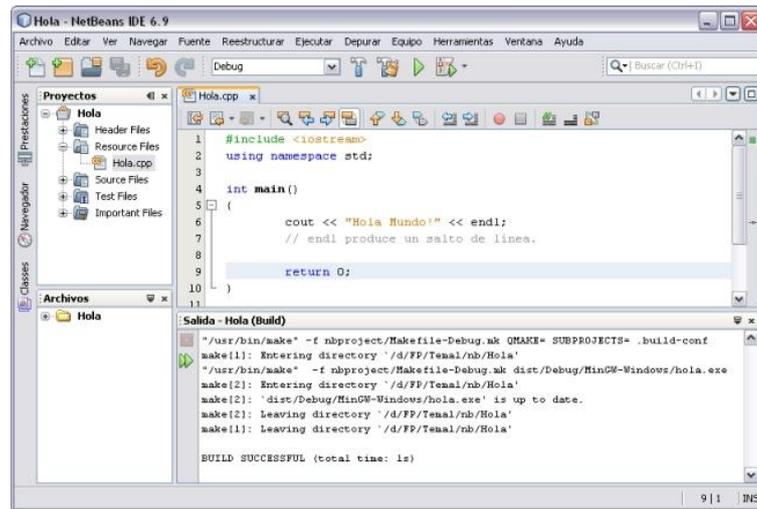


Figura 4. Entorno de Netbeans.

Es por ello que gracias a la contribución de las TIC han surgido nuevas plataformas *game based-learning* que simulan los entornos de desarrollo tradicionales resultando más interactivos para los principiantes permitiendo que de una forma divertida se aprenda un lenguaje de programación específico y sobretodo lograr resultados eficaces con la aplicación de dichas plataformas. Por tanto, en el Anexo 1, se listan las plataformas alineadas a la enseñanza-aprendizaje de la programación.

Estas plataformas están orientadas a la enseñanza de distintos lenguajes de programación, así como también están diseñadas con distintos niveles de complejidad dependiendo de la edad del público objetivo. Tal es el caso de que las plataformas están diseñadas bajo dos tipologías para la programación, como lo es a través de bloques y de texto.

**Programación de bloques:** se refiere a una programación visual en donde se colocan una serie de bloques, los mismos que cumplen una instrucción definida internamente, permitiendo que al construir un conjunto de bloque de instrucciones se ejecuten de forma secuencial. Por lo general esta tipología es empleada en las plataformas para niños que están iniciando con la programación para el desarrollo del pensamiento computacional. Tal como se muestra en la Figura 5 de una plataforma basada en la tipología descrita anteriormente.

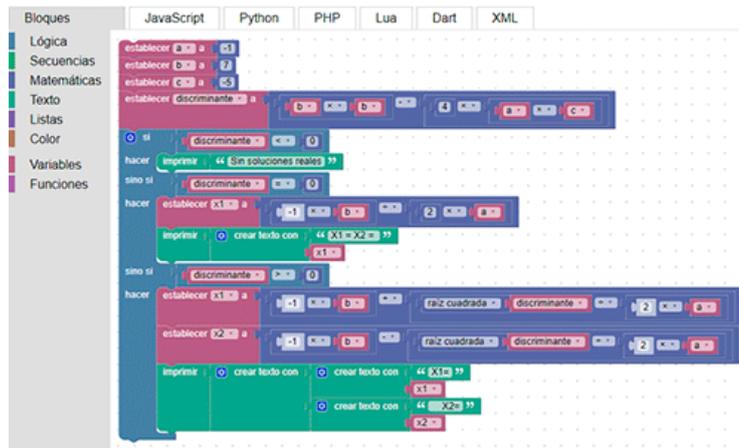


Figura 5. Interfaz de programación de blockly

**Programación de texto:** es la forma tradicional en la cual se van colocando las líneas código cumpliendo con las reglas sintácticas de acuerdo con el lenguaje de programación, tal como se lo demuestra con la Figura 6.



Figura 6. Interfaz de programación en texto de CODEDARENA

Y difiere de los entornos tradicionales, puesto a que la ejecución de las líneas de código se ven reflejadas en personajes que de acuerdo al tema tienen movimiento, dependiendo del enfoque de las plataformas, así como se muestra en la Figura 7.



Figura 7. Interfaz de salida de CODEDARENA

Por tanto, para la presente investigación se emplearán las plataformas que utilicen como medio de programación de texto, puesto que van dirigidas y son aplicables para estudiantes de secundaria en adelante.

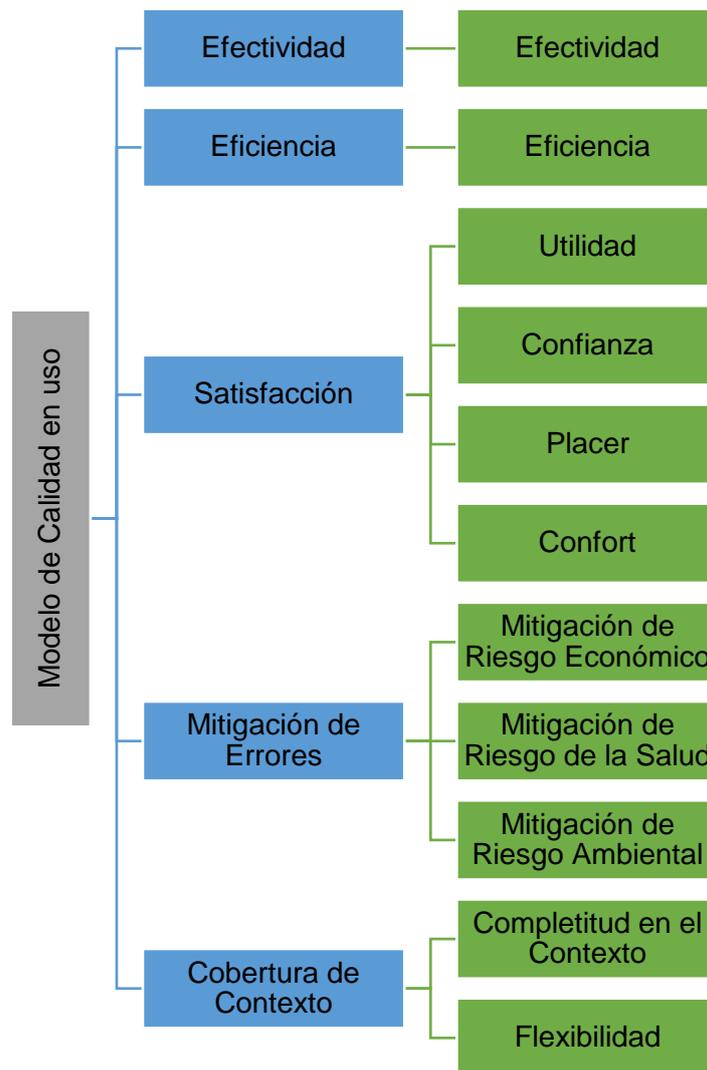
Es necesario recalcar que por lo general estas plataformas *game based-learning* no son empleadas dentro de la currícula en la educación del Ecuador puesto a su escasa fiabilidad de que logren objetivos mensurables en el proceso de enseñanza-aprendizaje. Y esto se debe a que no se encuentran certificadas bajo ningún estándar de calidad que ratifica su utilidad en procesos educativos.

## 2.5. Modelo de Calidad de Software

El término calidad de software se refiere al grado de desempeño de las principales características con las que debe cumplir un sistema computacional durante su ciclo de vida (Callejas et al., 2017). Es así como a través de objetivos y características se puede medir su calidad en varios aspectos. Así como también Carrizo & Alfaro (2018) mencionan que en el instituto de Ingenieros Eléctricos y Electrónicos (IEEE) lo referencian como el “grado con el que un sistema, componente o proceso cumple los requerimientos especificados y las necesidades o expectativas del cliente o usuario” (p.118).

Es así como los modelos de calidad permiten ratificar la eficacia de su desarrollo, así como también le da un valor que certifica que el software cumple con todas las características necesarias para calificar como un software de calidad, y de esta manera poder estar al frente de demás software alineados a un área en semejanza. Así mismo, los modelos de calidad de software se clasifican de acuerdo con el enfoque de evaluación, ya sea a nivel de proceso, producto o calidad en uso.

En donde a nivel de proceso establece que la evaluación del software se realiza durante su desarrollo, haciendo referencia a Callejas et al. (2017), que menciona que por cada etapa del proceso de desarrollo se debe llevar a cabo el control y seguimiento de los aspectos de calidad para la reducción de riesgos y errores en la implementación. Por otro lado, a nivel de producto se centra en especificar y evaluar la calidad a través de medidas internas de las propiedades del software y externas indirectas del comportamiento frente a los usuarios. Siendo finalmente a nivel de calidad de uso, tal como se muestra en Figura 8, el conjunto de atributos relacionados con la aceptación por parte del usuario final, que considera varias características para su posterior evaluación.

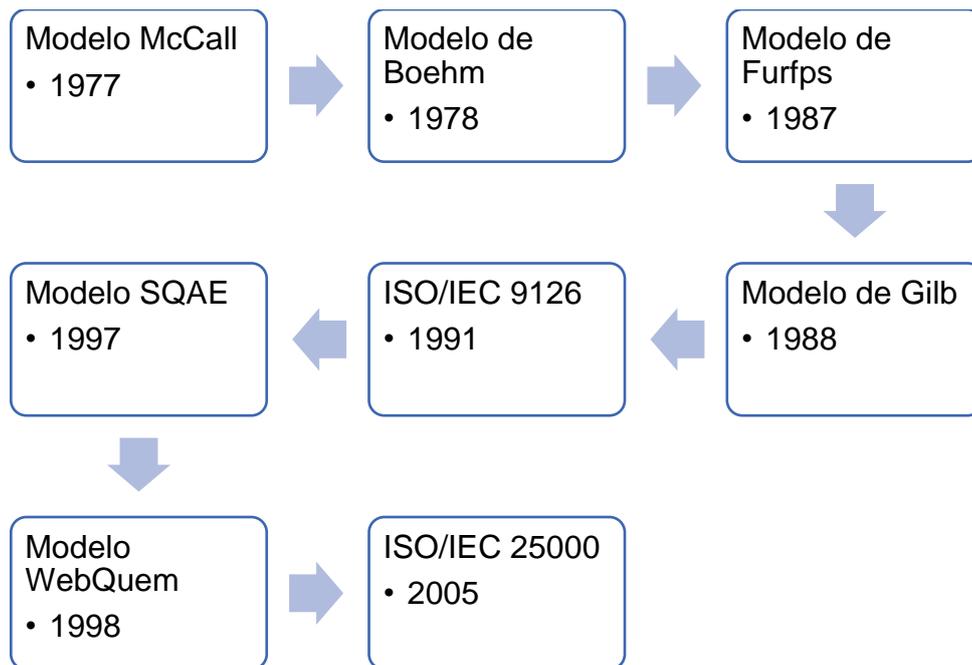


*Figura 8. Modelo de Calidad en uso según la ISO 25000*

Por tanto, conociendo el enfoque de las normas y el alcance de sus características en la presente investigación se evaluarán los softwares a nivel de producto puesto que abarca características y parámetros de evaluación necesarios para medir la calidad del producto. Dentro del enfoque de evaluación a nivel de producto, existen varios modelos de calidad que presentan distintas métricas de medición, características, subcaracterísticas, rangos de prioridad y demás ítems que permiten reducir la objetividad en la asignación de un valor que represente la buena calidad del software sea esta manera interna o externa. Tomando en consideración que los modelos presentan algunas características para la evaluación de la calidad externa y a su vez ayudan a validar la calidad interna.

### 2.5.1. Evolución de los Modelos de Calidad a Nivel de Producto

La Norma ISO ha evolucionado con el paso del tiempo, tal como se representa en la Figura 9 con un diagrama de los modelos de calidad que han surgido hasta la actualidad de manera cronológica referente a la calidad a nivel del producto volviendo un producto cada vez más competitivo dentro del mercado.



*Figura 9. Modelos de Calidad a Nivel de Producto*

Dentro de los modelos anteriores se encuentra el primero modelo diseñado por la ISO/IEC (Organización Internacional de Normalización) e IEC (Comisión Electrotécnica Internacional) en el año 1991 que evolucionó debido a inconsistencias detectadas con las versiones anteriores, dando como resultado la ISO/IEC 25000, que abarca nuevas características que han permitido que la norma se reestructure debido a nuevas necesidades de los usuarios. Tal como lo mencionan Calero et al., (2010) que “esta serie de normas interpretan la calidad de un sistema software como el grado en el que el sistema satisface las necesidades implícitas y explícitas de sus diferentes usuarios” (p.55), es por tanto que en la Figura 10, se estructura la evolución de la ISO 9126 hasta la actualidad.



Figura 10. Evolución de la norma de calidad de software

En el Anexo 2 se presenta una comparación de los modelos a nivel de producto, que permitirá identificar cual modelo abarca las mejores características de calidad requerida.

## 2.5.2. Norma ISO 25000

La Norma ISO/IEC 25000, conocida como SQuaRE (*System and Software Quality Requirements and Evaluation*), es una familia de normas que tiene por objetivo la creación de un marco de trabajo común para evaluar la calidad del producto software (Iso25000, 2018). Permitiendo la evaluación del software a través de métricas que disminuyan la subjetividad al momento de asignarle un porcentaje de calidad de acuerdo con los parámetros evaluados.

La familia ISO/IEC 25000 es el resultado de la evolución de otras normas anteriores, tal como se referencia en la Figura 10, especialmente de la norma ISO/IEC 9126, que describe las particularidades de un modelo de calidad del producto software, y la ISO/IEC 14598, que aborda el proceso de evaluación de productos software. La serie de estándares SQuaRE consta de las siguientes divisiones bajo el título general División para la Gestión de Calidad (SQuaRE) cómo se lo define en la Figura 11.

Cada una de estas divisiones que conforman la ISO 25000 permiten segmentar la gestión de calidad de un software, para que de esta manera los evaluadores tengan un contexto específico de cada una de las partes que se contemplan al momento de evaluar un producto de software.



*Figura 11. Familia de la Norma ISO 25000*

La ISO 2501n presenta dos modelos de calidad que incluyen características para la calidad interna y externa, siendo el primero la ISO 25010 que permite evaluar el producto y la calidad en uso a través de subcaracterísticas y la ISO 25012 que permiten evaluar la calidad de los datos que forman parte del sistema de información. Dando apertura a la ISO 2502n por la necesidad de definir parámetros o métricas que permitan evaluar la calidad del producto, por tanto, se encuentra subdividida en 5 normas.

- ISO/IEC 25020: incluye consideraciones para seleccionar medidas de calidad y funciones de medición normalizadas para medidas de calidad (ISO/IEC, 2019).
- ISO/IEC 25021: proporciona guías para especificar los Elementos de medida de calidad (QME) y el conjunto inicial de QME, es decir una medida definida en términos de una propiedad y el método de medición para cuantificarla, para la transformación por una función matemática (ISO/IEC, 2020).
- ISO/IEC 25022: define medidas de calidad en uso para las características definidas en ISO / IEC 25010 que permiten evaluar el software (ISO/IEC, 2011a).
- ISO/IEC 25023: define medidas de calidad para evaluar cuantitativamente la calidad del sistema y del producto de software en términos de la ISO / IEC 25010 a través de sus características y subcaracterísticas (International Organization for Standardization & International Electrotechnical Commission, 2014).

- ISO/IEC 25024: define medidas de calidad de datos para medir cuantitativamente la calidad de datos en términos de la ISO / IEC 25012 a través de sus características (ISO/IEC, 2015).

Definidos los objetivos de cada una de las divisiones de la ISO 2502n es necesario entender que es indispensable especificar requisitos para la calidad que pueda ser utilizados como entradas para el proceso evaluación del software, siendo esta la ISO 2503n que se compone de la ISO 25030 que proporciona el marco para los requisitos de calidad para productos de software, que incluye el concepto de los requisitos de calidad y los requisitos y recomendaciones para los procesos y métodos para obtenerlos, definirlos, usarlos y gobernarlos (ISO/IEC, 2019).

Siendo por último la ISO 2504n indispensable puesto a que conlleva requisitos, recomendaciones y guías necesarias para llevar a cabo el proceso de evaluación de un software. Es así como esta división se encuentra conformado por las siguientes normas:

- ISO/IEC 25040: proporciona una descripción del proceso para evaluar la calidad del producto de software y establece los requisitos para la aplicación de este proceso (ISO/IEC, 2011).
- ISO/IEC 25041: proporciona requisitos, recomendaciones y pautas para la evaluación de la calidad del producto del sistema y software, para la aplicación de ISO / IEC 25040 (ISO/IEC, 2012).
- ISO/IEC 25042: define la estructura y el contenido de la documentación que se utilizará para describir un módulo de evaluación (ISO/IEC, 2011).
- ISO/IEC 25045: utiliza una metodología que involucra dos tipos de evaluación para la recuperabilidad (ISO/IEC, 2010).

Una vez entendido los modelos que forman parte de la familia ISO/IEC 25000 que contemplan todos los objetivos para la evaluación total de un software, es necesario delimitar cual es la estructura de evaluación, para de esta manera entender cómo se compone para la evaluación de un producto de software.

### 2.5.2.1. Estructura del Modelo de la ISO 25000

Para la evaluación de un producto de software y siguiendo los estándares de calidad de la ISO/IEC 25000 y todas las normas que trae consigo, se lleva una estructura tal como se lo representa en la Figura 12, en donde según características y subcaracterísticas previamente definidas permiten evaluar de manera más detallada un aspecto general, determinando métricas y parámetros de evaluación por cada una de las mencionadas con antelación para lograr un valor que represente la calidad del producto.

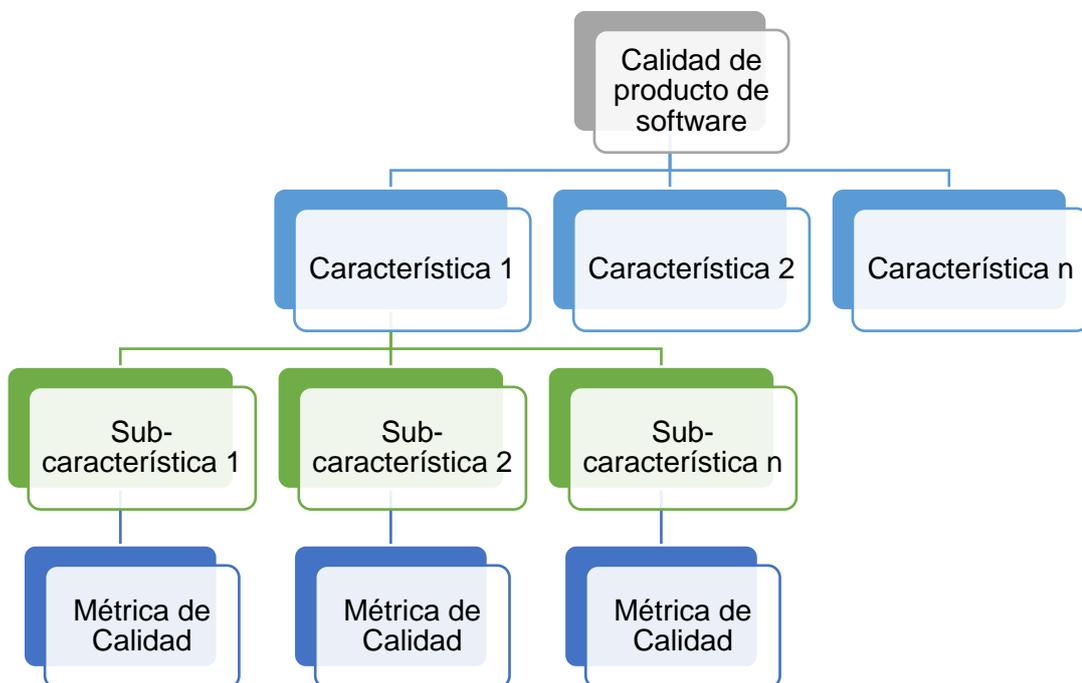


Figura 12. Estructura de la ISO 25000

Las métricas que se definen en la ISO 25000 permiten determinar un parámetro de evaluación por cada característica, para una posterior medición, que como lo menciona Fernández et al. (2015) representa un proceso por el cual se asigna un número a atributos o características del mundo real, por tanto por cada una de ellas se definen reglas claras, bien definidas y consistentes de asignación de números.

Conociendo cual es la estructura que se llevará para la evaluación de las plataformas *game based-learning* alineadas a la programación, se han definido las características y subcaracterísticas necesarias para la evaluación según la ISO 25000, tal como se muestra en la Figura 13. Análogamente se

debe tener en cuenta que para la presente investigación se las adecuarán según el enfoque de las plataformas a evaluar, puesto a la necesidad de incorporar nuevas medidas que se deben contemplar para una óptima y eficaz evaluación, logrando según los resultados la aplicación eficiente de las mismas dentro del ámbito educativo.

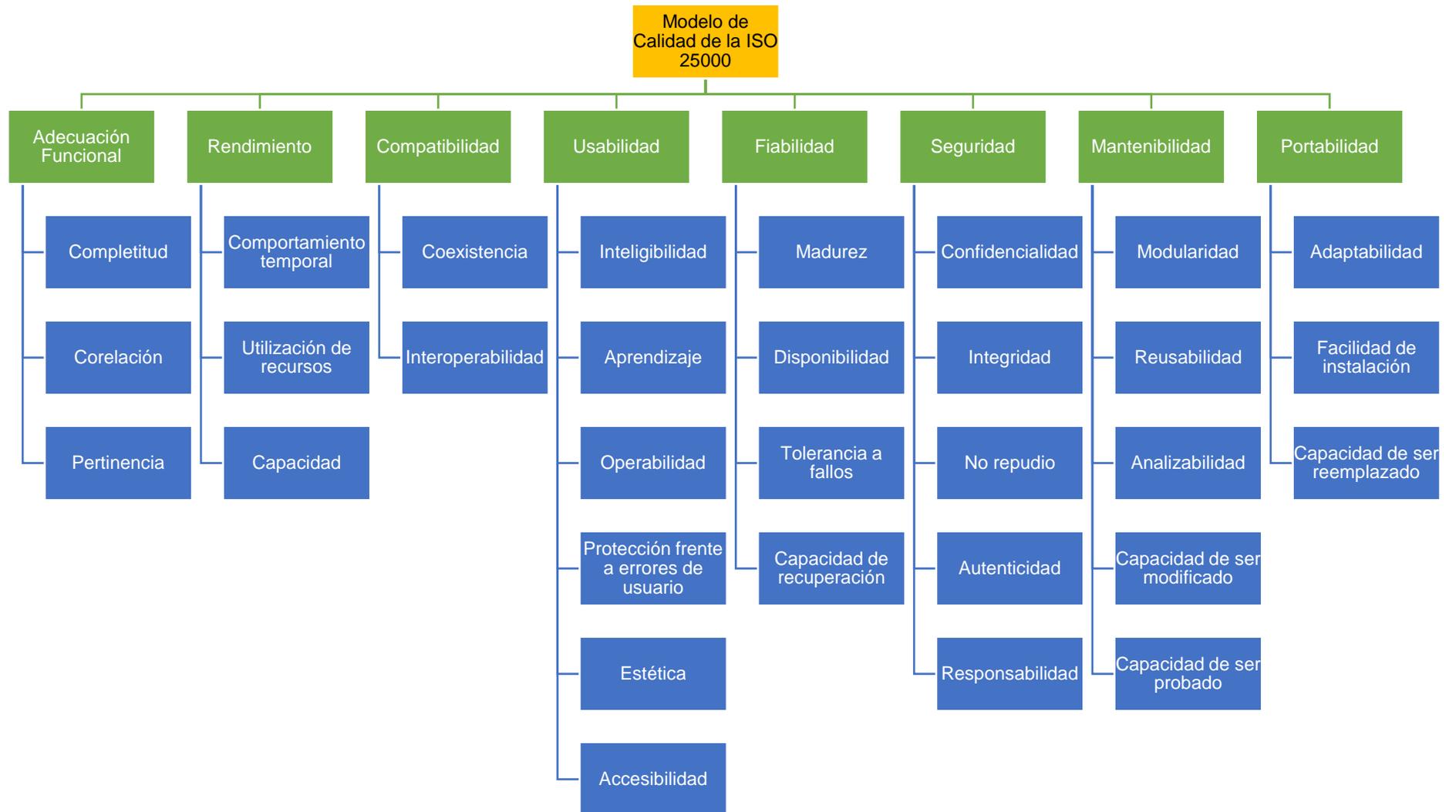


Figura 13. Modelo de calidad de software según la ISO 25010

Una vez definido el modelo de calidad, a continuación, se detallan cada una de las características para un entendimiento de cada una de las mismas tal como lo interpreta Rahman (2020).

- **Adecuación Funcional**

Grado en el que un producto o sistema proporciona funciones que satisfacen las necesidades establecidas e implícitas cuando se usa en condiciones específicas. Idoneidad y adecuación de las herramientas de funcionalidad que cubren varios propósitos: comunicación, monitoreo, compartir, seguridad, interoperabilidad etc.

- **Rendimiento**

El rendimiento relativo a la cantidad de recursos utilizados en las condiciones establecidas. La eficiencia del rendimiento evalúa el uso del sistema y la cantidad óptima de recursos.

- **Compatibilidad**

Grado en el que un producto, sistema o componente puede intercambiar información con otros productos, sistemas o componentes, y / o realizar sus funciones requeridas, mientras comparte el mismo hardware o software medio ambiente. La compatibilidad evalúa la intercambio y uso de información e impacto en otros productos que comparten la misma plataforma.

- **Usabilidad**

Grado en el que un producto o sistema puede ser utilizado por usuarios específicos para lograr objetivos específicos con efectividad, eficiencia y satisfacción en un contexto de uso específico. La usabilidad analiza la calidad del aprendizaje, la enseñanza, el entorno de aprendizaje y la interacción.

- **Fiabilidad**

Grado en el que un sistema, producto o componente realiza funciones específicas en condiciones específicas durante un período de tiempo específico. La confiabilidad es la continuidad de un servicio correcto y de qué tan bien el sistema cumple con los requisitos de los usuarios. Es la evaluación de qué tan bien los usuarios piensan que el sistema proporciona el servicio que necesitan.

- **Seguridad**

Grado en el que un producto o sistema protege la información y los datos para que las personas u otros productos o sistemas tengan el grado de acceso a los datos adecuado a sus tipos y niveles de autorización. La seguridad del sistema cubre aspectos de privacidad y seguridad.

- **Mantenibilidad**

Grado de efectividad y eficiencia con el cual un producto o sistema puede ser modificado por los programadores previstos. La capacidad de mantenimiento es la capacidad del sistema para identificar y corregir una falla dentro del sistema y los componentes del software.

- **Portabilidad**

Grado de efectividad y eficiencia con el cual el sistema, producto o componente puede transferirse de un hardware, software u otro entorno operativo o de uso a otro. La portabilidad se refiere a qué tan bien el sistema puede adoptar los cambios en su entorno o en otro

## CAPÍTULO III

### 3. MARCO METODOLÓGICO

#### 3.1. Tipo de investigación

El presente proyecto cuenta con un enfoque mixto, es decir cualitativo y cuantitativo para el cumplimiento de los objetivos específicos y por consecuente el objetivo general. Por ello se lo ha dividido en dos momentos puesto a que está enfocado en el diseño de un modelo y la aplicación del mismo. Aplicándose en primera instancia una metodología de investigación descriptiva, la misma que como indica Arias (2006) consiste en la caracterización de un hecho, fenómeno, grupo o individuo con el fin de establecer su estructura y comportamiento; como también una investigación explicativa que se encarga de buscar el porqué de los hechos mediante el establecimiento de relaciones causa-efecto.

Permitiendo identificar mediante las búsquedas e indagaciones las razones de la aplicación de plataformas *game based-learning* alineadas a la programación en el proceso enseñanza-aprendizaje. De igual forma es de tipo exploratoria puesto que se analizarán cada una de las plataformas definidas para recopilar la información de sus características para su previo contraste con la ISO 25000. Por ello se aplicó el instrumento de la encuesta y entrevista para la validación de los componentes del modelo y las características incluidas de acuerdo a los requerimientos de los involucrados en el proceso de enseñanza-aprendizaje como la situación actual de las herramientas que se emplean en la Facultad de Ingeniería, Carrera de Ingeniería en Sistemas Computacionales y Ciencias de la Computación de la Universidad Católica de Santiago de Guayaquil.

Una vez construido el modelo de evaluación en segunda instancia se emplea una investigación aplicada la cual se basa en evaluar los aspectos teóricos de las plataformas en estudio para generar un resultado cuantitativo y definir las como aplicables a procesos de educación superior. De esta forma se pondrá a prueba el modelo para determinar su efectividad en la evaluación interna como externa de un producto software y la completitud de las características del mismo.

### 3.2. Identificación de la población y muestra

#### 3.2.1. Población

La presente investigación tiene como finalidad proporcionar de acuerdo a los resultados de la evaluación de las plataformas bajo el modelo de calidad propuesto inspirado en la norma ISO 2500, herramientas certificadas bajo lineamientos que permitan innovar el proceso enseñanza aprendizaje. Por tanto, resulta indispensable indagar sobre la situación actual de los estudiantes y docentes, es decir cómo se desarrolla el proceso en mención, es por ello que se indispensable definir la población objetivo que como lo menciona López (2004) es el conjunto de personas u objetos de los que se desea conocer algo en una investigación.

El primer instrumento se aplicará en los estudiantes de la Universidad Católica de Santiago de Guayaquil de la Facultad de Ingeniería, Carrera de Ciencias de la Computación y Sistemas Computacionales, desde el segundo ciclo hasta los estudiantes que se encuentran cursando el proceso de titulación. Es por tanto que a continuación se define la población a emplear.

*Tabla 3. Población de la investigación*

<b>Población de:</b>	<b>Cantidad</b>
Estudiantes de Sistemas Computacionales	52
Estudiantes de Ciencias Computacionales	67
<b>TOTAL</b>	<b>119</b>

Elaboración propia.

Siendo el segundo instrumento, las entrevistas; que se realizarán a dos docentes elegidos de forma aleatoria que imparten materias relacionadas a la programación a estudiantes de la Universidad Católica de Santiago de Guayaquil de la Facultad de Ingeniería, Carrera de Ciencias de la Computación y Sistemas Computacionales.

### 3.2.2. Muestra

Una vez definida la población, la obtención de la muestra se hizo en base al universo total de estudiantes entre la Carrera de Ciencias de la Computación y Sistemas Computacionales, para determinar un subconjunto o parte del universo en el que se llevará a cabo la investigación. Hay distintos procedimientos para obtener la cantidad de los componentes de la muestra y determinar la parte representativa de la población.

Con base a este universo se utilizó la fórmula del “muestreo aleatorio simple para poblaciones finitas”, tomando un margen de error del 6%. La fórmula que se aplicó fue la siguiente:

$$n = \frac{m}{e^2 (m - 1) + 1}$$

Detallando las variables utilizadas se tiene:

m = Tamaño de la población (119)

e = Error de estimación (6%)

Una vez definido los valores a emplear, se aplicó la fórmula.

$$n = \frac{m}{e^2 (m - 1) + 1}$$

$$n = \frac{119}{(0.06)^2 (119 - 1) + 1}$$

$$n = \frac{119}{1.4248}$$

$$n = 83$$

Se obtuvo una muestra de 83 estudiantes.

### 3.3. Técnicas e instrumentos para el levantamiento de información

La presente investigación requirió el uso de técnicas que permitieron la obtención de información para el respectivo análisis y contraste de lo empírico frente a lo teórico; las encuestas, la entrevista, y la observación, cumplen con objetivos determinados que facilitan el sustento de la presente investigación, a continuación, se declara la finalidad de cada uno de los instrumentos empleados.

La encuesta es un instrumento que permite obtener y elaborar datos de modo rápido y eficaz (Casas et al., 2003), para lo cual utiliza un conjunto de procedimientos estandarizados mediante los cuales se recoge y analiza datos de una muestra de casos representativos de una población o universo, del que se pretende explorar, describir, predecir y/o explicar una serie de características (García et al., 2016). De esta forma permitirá recopilar la información necesaria para tener un contexto de la situación actual de los estudiantes con respecto a los conocimientos aprendidos sobre programación con las herramientas que se emplean en la actualidad en la Universidad.

Se emplearán entrevistas semiestructuradas, cuya facilidad permitirá entablar una conversación con el entrevistado a través de preguntas abiertas que den paso a la opinión, e indagar sobre la innovación en el uso de nuevas herramientas que contribuyan a la enseñanza de la programación de manera pedagógica. Es indispensable contar con la percepción del otro involucrado dentro del proceso enseñanza-aprendizaje, siendo este el docente, referente a tres segmentos, la factibilidad, usabilidad y aplicabilidad del uso de aplicaciones *game based-learning* relacionadas a la programación, por tanto como lo referencia Carrillo et al., (2020) “es una técnica de gran utilidad en la investigación cualitativa para recabar datos; que se define como una conversación que se propone un fin determinado distinto al simple hecho de conversar”.

Por último, para una óptima evaluación de las plataformas según los parámetros de la ISO 25000, se estructuraron bitácoras de análisis, como lo indica (Hernández Sampieri et al., 2010) “tienen la función de documentar el procedimiento de análisis y las propias reacciones del investigador al

proceso”, a través de un conjunto de categorías. Es así como este instrumento se aplicará a las plataformas en estudio para evaluarlas de acuerdo a las métricas referentes a las subcaracterísticas de la norma.

Por tanto, la aplicación de estos instrumentos permitió, triangular mediante el análisis de la información recopilada, la aplicabilidad de las plataformas *game based-learning* alineadas a la programación y comprender la situación actual del proceso educativo.

### **3.4. Tratamiento de la Información**

Una vez obtenida la información por medio de los instrumentos aplicados, se procedió al análisis de los mismos. Es así como, para las encuestas se realizó la tabulación a las respuestas obtenidas y se demostró por medio de gráficos estadísticos el resumen de estas, con la finalidad de visualmente evidenciar cuales fueron los resultados y como es la situación actual desde la perspectiva del estudiante sobre el proceso enseñanza-aprendizaje.

Por otra parte, la entrevista al ser un instrumento cualitativo permite sintetizar las respuestas bajo el criterio personalizado de cada uno de los docentes, de esta manera se logró construir una conclusión referente a la opinión de los entrevistados sobre la importancia y necesidad de implementar nuevas plataformas educativas, y que características deberían tener para ser aplicadas en el proceso de enseñanza, para lograr buenos resultados.

### **3.5. Análisis de resultados de las encuestas**

La realización de las encuestas a través de un cuestionario cerrado ha permitido evidenciar la situación actual del nivel de los conocimientos adquiridos por parte de los estudiantes. Es por tanto que a continuación se detallan los resultados obtenidos, para de esta manera lograr un entendimiento general de lo que se debe cubrir para cumplir la totalidad los objetivos trazados para llevar de una forma correcta el proceso enseñanza-aprendizaje.

## 1. ¿Cómo considera usted su nivel de conocimientos de programación antes de ingresar a la Universidad?

Tabla 4. Nivel de Conocimiento en Programación previo al ingreso la Universidad

ITEM DE RESPUESTA	CANTIDAD	PORCENTAJE
Nada (1)	53	63,9%
Bajo (2)	22	26,5%
Medio (3)	6	7,2%
Suficiente (4)	1	1,2%
Alto (5)	1	1,2%
TOTAL	83	100%

Elaboración propia.

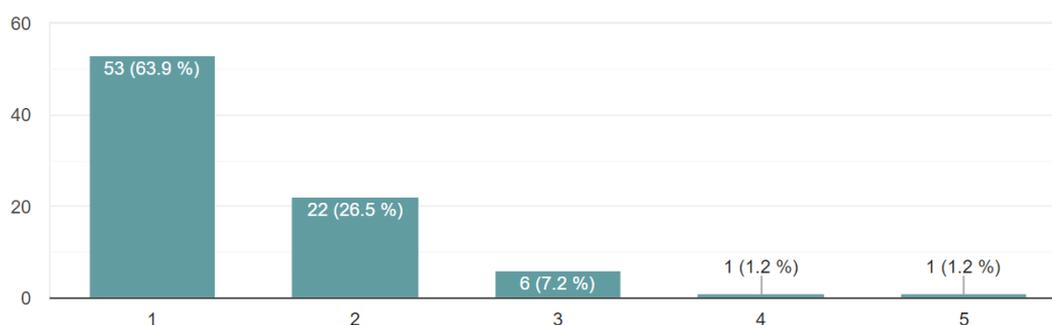


Figura 14. Nivel de conocimientos previos al ingreso a la universidad

De las encuestas realizadas, y con respecto a la pregunta ¿Cómo considera usted su nivel de conocimientos de programación antes de ingresar a la Universidad?; se obtuvieron los siguientes resultados: el 63,9% de los estudiantes encuestados ingresaron a la Universidad con nulos conocimientos referentes a programación, puesto a la deficiente enseñanza del área en mención en la educación secundaria, el 28,5% ingresó con conocimientos limitados, el 7,2% ingresó con conocimientos medios, el 1,2% ingresó con conocimientos suficientes, razón por la que seguramente en el colegio apuntó a una carrera de este tipo y por último el 1,2 % ingresó con un alto nivel de conocimientos, cabe mencionar que representa a 1

estudiante, que probablemente tomó cursos antes de ingreso a la universidad.

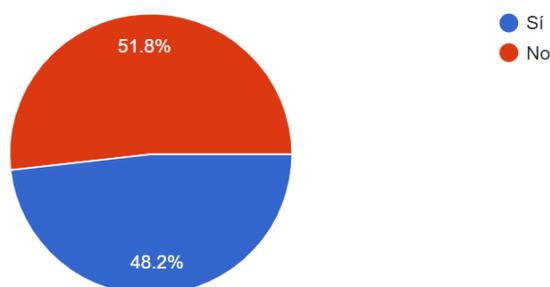
Permitiendo concluir que la mayoría de los estudiantes ingresaron a la Universidad con nulos conocimientos referentes a programación, siendo un punto de partida importante para determinar la eficacia y eficiencia de la experiencia cursando la primera materia relacionada a programación. Puesto que es necesario que las herramientas que se emplean para el proceso enseñanza-aprendizaje de Introducción a la Programación, sean las más didácticas para que los estudiantes comprendan y asocien los conceptos que formarán parte de la carrera y futuro profesional.

**2. ¿Considera usted que las herramientas utilizadas en clases cubren las expectativas básicas de enseñanza de la programación?**

*Tabla 5. Cumplimiento de expectativas de las herramientas de programación*

ITEM DE RESPUESTA	CANTIDAD	PORCENTAJE
SI	40	48,2%
NO	43	51,8%
TOTAL	83	100%

Elaboración propia.



*Figura 15. Cumplimiento de expectativas de las herramientas de programación*

De las encuestas realizadas, y con respecto a la pregunta: ¿Considera usted que las herramientas utilizadas en clases cubren las expectativas básicas de enseñanza de la programación?; se obtuvieron los siguientes resultados: el 51,8% de los estudiantes mencionaron que las herramientas empleadas en la actualidad NO cubren las expectativas básicas de programación.

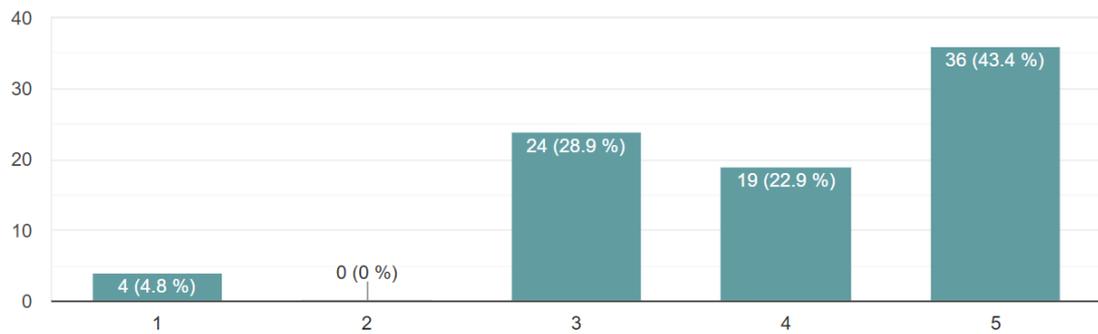
Concluyendo que la aplicación de las mismas no permite la comprensión del área en mención o que son de difícil usabilidad, es por tanto que, en otra de las preguntas planteadas en el presente formulario se tabularán las dificultades que han presentado los estudiantes al emplear dichas herramientas. Y el 48,2% de los estudiantes encuestados, consideran que las herramientas empleadas son lo suficiente para cubrir áreas básicas de programación.

**3. ¿Qué tanta dificultad presentó usted al cursar la primera materia relacionada a la programación?**

*Tabla 6. Nivel de dificultad al cursar la primera materia de programación*

ITEM DE RESPUESTA	CANTIDAD	PORCENTAJE
Nada (1)	4	4,8%
Poca (2)	0	0%
Bastante (3)	24	28,9%
Mucha (4)	19	22,9%
Demasiada (5)	36	43,4%
TOTAL	83	100%

Elaboración propia.



*Figura 16. Nivel de dificultad al cursar la primera materia de programación*

De las encuestas realizadas, y con respecto a la pregunta: ¿Qué tanta dificultad presentó usted al cursar la primera materia relacionada a la programación?; se obtuvieron los siguientes resultados: el 43,4% indicó que presentó demasiada dificultad para cursar su primera relacionada a la programación, el 22,9% presentó mucha dificultad, el 28,9% presentó bastante dificultad, el 0% presentó poca dificultad y por último el 4,8% presentó nada de dificultad.

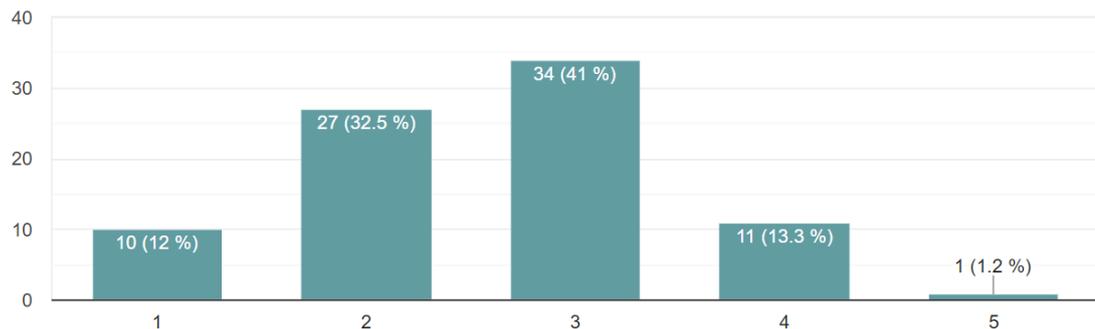
Es así como se concluyó que más del 80% de los estudiantes presentaron dificultad al cursar su primera materia relacionada a programación, puesto a que como se lo menciona en la pregunta 1, los mismos ingresaron a la universidad son nulos conocimientos. Es decir que se deben de buscar nuevas herramientas que faciliten la comprensión de los conocimientos de una forma divertida para el desarrollo de nuevas habilidades para disminuir la dificultad que presentan los estudiantes que no han ingresado con conocimientos previos.

**4. ¿Cuánto considera usted que fue su nivel de dominio adquirido después de cursar la primera materia de programación en la Universidad?**

*Tabla 7. Nivel de dominio después de cursar la primera materia de programación*

ITEM DE RESPUESTA	CANTIDAD	PORCENTAJE
Nada (1)	10	12%
Poco (2)	27	32,5%
Medio (3)	34	41%
Mucho (4)	11	13,3%
Demasiado (5)	1	1,2%
TOTAL	83	100%

Elaboración propia.



*Figura 17. Nivel de dominio después de cursar la primera materia de programación*

De las encuestas realizadas, y con respecto a la pregunta: ¿Cuánto considera usted que fue su nivel de dominio adquirido después de cursar la primera materia de programación en la Universidad?; se obtuvieron los siguientes resultados: el 41% de los estudiantes encuestados manifestaron que después de cursar su primera materia relacionada a programa obtuvieron un dominio medio del área, el 32,5% obtuvo poco dominio de programación, el 13,3% obtuvo mucho dominio de la programación, el 12% no obtuvo ningún dominio sobre el área en mención y el 1,2% manifestó que obtuvo demasiado dominio de la programación.

## 5. ¿Qué dificultades usted ha presentado para aprender programación?

Tabla 8. Dificultades al aprender programación

ITEM DE RESPUESTA	CANTIDAD	PORCENTAJE
Los programas son de difícil entendimiento y operabilidad	46	55,4%
No comprender la explicación de las clases	60	72,3%
Softwares limitados	39	47%
Escasos conocimientos previo al ingreso a la universidad	55	66,3%
Pocas clases prácticas	12	14,5%
Otros	3	3,6%
<b>TOTAL</b>	<b>83</b>	<b>100%</b>

Elaboración propia.

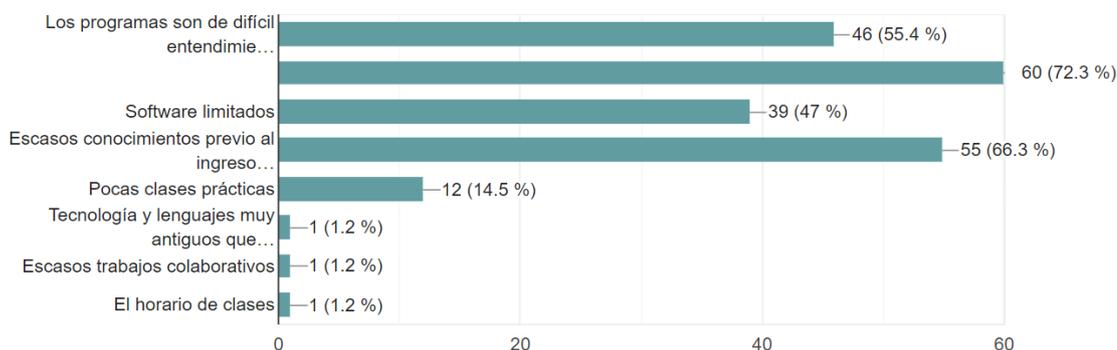


Figura 18. Dificultades al aprender programación

De las encuestas realizadas, y con respecto a la pregunta: ¿Qué dificultades usted ha presentado para aprender programación?; se obtuvieron los siguientes resultados: el 72,3% de los estudiantes han presentado facultades puesto a que no comprenden la explicación realizada durante las clases, el 66,3% de los estudiantes han presentado facultades puesto a que ingresaron a la universidad con escasos conocimiento referentes a programación; el 55,4% de los encuestados manifestaron que se les dificultó aprender puesto a que los programas que se emplean son de difícil entendimiento y operabilidad, el 47% de las respuestas indican que se

les dificultó puesto a que los software empleados son limitados, el 14,5% de los estudiantes indicaron que se llevan a cabo pocas clases prácticas y por último el 3,6% de los encuestados manifestaron otras razones del porque la dificultad al cursar la primera materia relacionada a programación.

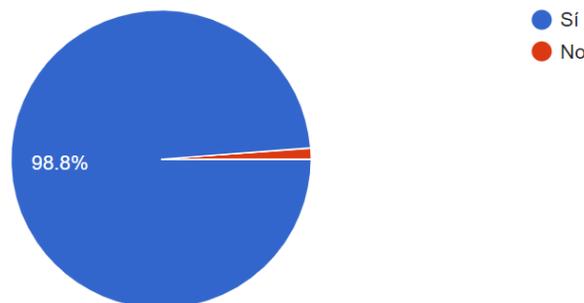
Concluyendo que la mayoría de los estudiantes concuerdan que las tres razones principales, por las cuáles se les dificultó cursar su primera materia relacionada a programación, fue porque el proceso de enseñanza no era óptimo y por ende no se completaba el proceso de aprendizaje, puesto a que como también lo indican, los softwares empleados son de difícil operabilidad, puesto a que los mismos no son didácticos ni están orientados a un aprendizaje a través de juegos.

**6. ¿Considera usted que aprender programación a través de juegos ayudaría al dominio de la programación?**

*Tabla 9. Aceptación del dominio de la programación a través de juegos*

ITEM DE RESPUESTA	CANTIDAD	PORCENTAJE
SI	82	98.8%
NO	3	1,2%
TOTAL	83	100%

Elaboración propia.



*Figura 19. Aprender a través de juegos ayuda al dominio de la programación*

De las encuestas realizadas, y con respecto a la pregunta: ¿Considera usted que aprender programación a través de juegos ayudaría al dominio de la programación?; se obtuvieron los siguientes resultados: el 98,8% de los estudiantes encuestados manifestaron que aprender a través de juegos ayudaría al dominio de la programación, y el 1,2% manifestó que no sería de utilidad. Basándose en que un porcentaje máximo menciono que sería de utilidad.

Se puede concluir que los estudiantes que según la pregunta 2 presentaron demasiada dificultad al cursar su primera materia relacionada a programación, aplicando plataformas game based-learning alineadas al área del conocimiento en mención, disminuirían los niveles de dificultad puesto que, según las investigaciones realizadas, emplear herramientas inspiradas en este modelo facilita y pone por delante el proceso de aprendizaje.

7. ¿De las plataformas *Game based-learning* descritas a continuación cuál o cuáles usted ha empleado para el autoaprendizaje de la programación?

Tabla 10. Plataformas *game based-learning*

ITEM DE RESPUESTA	CANTIDAD	PORCENTAJE
CodedArena	4	4,8%
Codemurai	9	10,8%
Py Learn to Game	9	10,8%
CodinGame	6	7,2%
Mimo	6	7,2%
Colobot	7	8,4%
Robocode	4	4,8%
Ninguna	60	72,3%
Kahoot	1	1,2%
Java, Python	2	2,4%
TOTAL	83	100%

Elaboración propia.

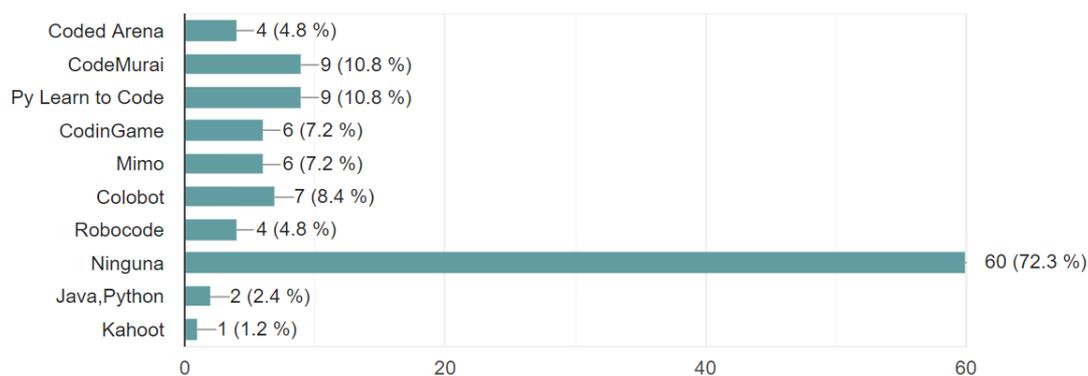


Figura 20. Plataformas *game based-learning*

De las encuestas realizadas, y con respecto a la pregunta: ¿De las plataformas game based-learning descritas a continuación cuál o cuáles usted ha empleado para el autoaprendizaje de la programación?; se obtuvieron los siguientes resultados: el 72,3% de los encuestas no ha empleado ninguna de las plataformas game based-learning que se listaron, el 10,8% ha empleado la plataforma Codemurai, el 10,8% ha empleado la plataforma Py Learn To Game, el 8,4 ha empleado la plataforma Colobot, el 7,2% ha empleado la plataforma CodinGame, el 7,2% ha empleado la plataforma Mimo, el 4,8% ha empleado la plataforma CodedArena, el 4,8% ha empleado la plataforma Robocode y el 3,6% ha empleado otro tipo de plataformas game based-learning.

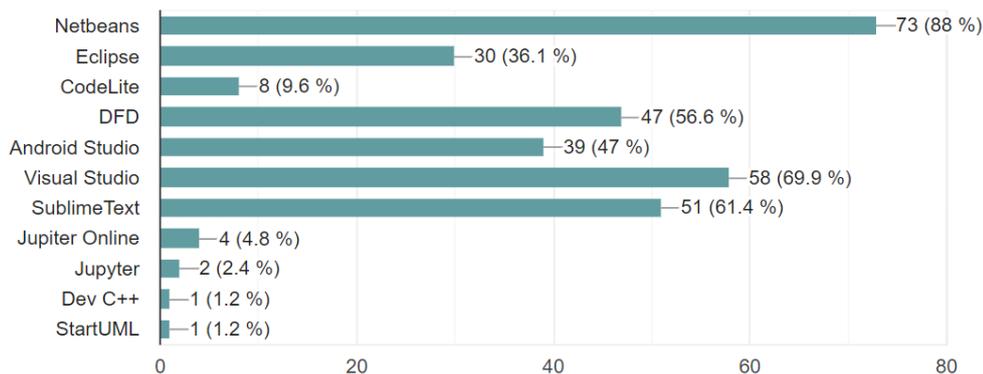
Concluyendo que la mayoría de los estudiantes encuestados no han empleado ninguna plataforma dentro del proceso enseñanza-aprendizaje como herramientas de apoyo, con la finalidad de poner en práctica la teoría impartida durante las clases relacionadas a la programación y por ende reforzar los conocimientos.

**8. ¿De las herramientas descritas cuál o cuáles usted ha empleado desde que ingresó a la universidad?**

*Tabla 11. Herramientas de programación*

ITEM DE RESPUESTA	CANTIDAD	PORCENTAJE
Netbeans	73	88%
Eclipse	30	36,1%
CodeLite	8	9,6%
DFD	47	56,6%
Android Studio	39	47%
Visual Studio	58	69,9%
SublimeText	51	61,4%
Otras	1	9,6%
<b>TOTAL</b>	<b>83</b>	<b>100%</b>

Elaboración propia.



*Figura 21. Herramientas de programación*

De las encuestas realizadas, y con respecto a la pregunta: ¿De las herramientas descritas cuál o cuáles usted ha empleado desde que ingresó a la universidad?; se obtuvieron los siguientes resultados: el 88% de los estudiantes han empleado Netbeans, el 69,9% ha empleado VisualStudio, el 61,4% ha empleado SublimeText, el 56,6% ha empleado DFD, el 47% ha empleado Android Studio, el 36,1% ha empleado Eclipse, el 9,6% ha empleado CodeLite y por último el 9,6% ha empleado otras herramientas.

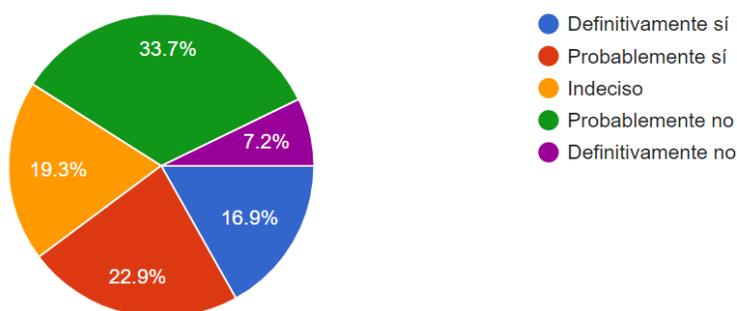
Analizando los resultados, puesto a la relevancia de conocer en que ciclos son más empleadas dichas herramientas, se concluye que Netbeans y Eclipse son las herramientas más empleadas por los estudiantes del segundo ciclo. De la misma forma para el análisis de la pregunta en cuestión se revisó si los mismos estudiantes de segundo ciclo presentaron dificultades al cursar su primera materia relacionada a programación, evidenciando que la mayoría de los estudiantes del ciclo en mención presentaron dificultades; por ende, es necesario la aplicabilidad de otras herramientas que apoyen a la interiorización de los contenidos impartidos.

9. **¿Considera usted que las herramientas de enseñanza actual referente al área de la programación son las más eficientes para fomentar el aprendizaje activo?**

*Tabla 12. Eficiencia de las herramientas para fomentar el aprendizaje activo*

ITEM DE RESPUESTA	CANTIDAD	PORCENTAJE
Definitivamente SI	14	16,9%
Probablemente SI	19	22,9%
Indeciso	16	19,3%
Probablemente NO	28	33,7%
Definitivamente NO	6	7,2%
TOTAL	83	100%

Elaboración propia.



*Figura 22. Eficiencia de las herramientas para fomentar el aprendizaje activo*

De las encuestas realizadas, y con respecto a la pregunta: ¿Considera usted que las herramientas de enseñanza actual referente al área de la programación son las más eficientes para fomentar el aprendizaje activo?; se obtuvieron los siguientes resultados: el 33,7% de los estudiantes manifestaron que probablemente las herramientas empleadas en la actualidad NO sean eficientes para fomentar el aprendizaje activo, el 22,9% de los encuestados manifestaron que probablemente SI son eficientes para promover el aprendizaje activo, el 19,3% están indecisos en que si las

herramientas empleadas en la actualidad fomentan o no el aprendizaje activo, el 16,9% de los estudiantes indicaron que definitivamente las herramientas fomentan el aprendizaje activo, y por último el 7,2% indico que definitivamente NO fomentan el aprendizaje activo.

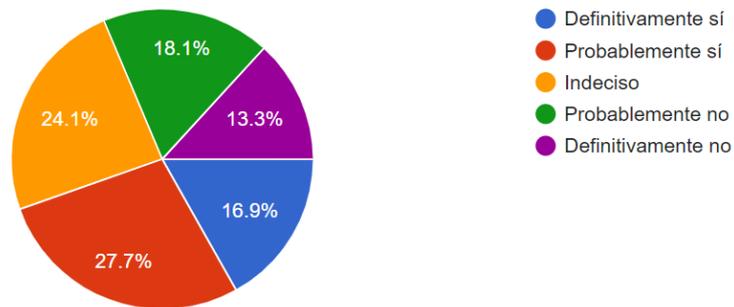
Permitiendo concluir que la mayoría de los estudiantes consideran las herramientas empleadas para el aprendizaje de programación empleadas en la actualidad no fomenta el aprendizaje activo, dando paso a entender por qué el bajo dominio del área en mención puesto a que las herramientas que se están utilizando no fomenta el interés por seguir practicando a través de la práctica.

**10. ¿Considera usted que las herramientas empleadas en la actualidad lo motivan a seguir aprendiendo?**

*Tabla 13. Nivel de motivación que incentivan las herramientas en la actualidad*

ITEM DE RESPUESTA	CANTIDAD	PORCENTAJE
Definitivamente SI	14	16,9%
Probablemente SI	23	27,7%
Indeciso	20	24,1%
Probablemente NO	15	18,1%
Definitivamente NO	11	13,3%
TOTAL	83	100%

Elaboración propia.



*Figura 23. Nivel de motivación que incentivan las herramientas en la actualidad*

De las encuestas realizadas, y con respecto a la pregunta: ¿Considera usted que las herramientas empleadas en la actualidad lo motivan a seguir aprendiendo?; se obtuvieron los siguientes resultados: el 27,7% menciona que las herramientas empleadas probablemente sí los motivan a seguir aprendiendo, el 24,1% de los estudiantes están indecisos sobre si las herramientas los motivan o no a seguir a aprendiendo, el 18,1% considera que probablemente no los motiva a seguir aprendiendo, el 16,9% considera que definitivamente las herramientas sí los motivan a seguir aprendiendo, y por último el 13,3% de los estudiantes consideran que las herramientas definitivamente no los motiva a seguir aprendiendo.

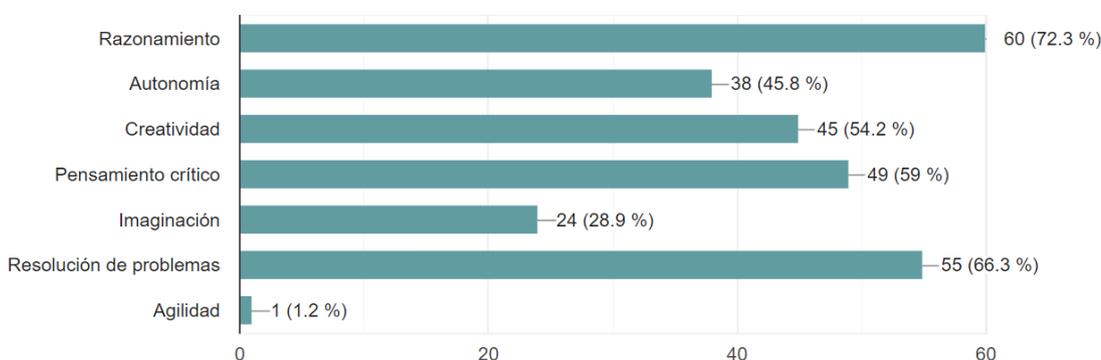
Concluyendo que el 31,4% de los estudiantes encuestados consideran que las herramientas empleadas en la actualidad no los motiva a seguir aprendiendo, conllevando a la deficiencia de los conocimientos puesto a que es indispensable reforzar los contenidos a través de la práctica.

**11. De las siguientes habilidades, ¿cuál o cuáles cree usted que las plataformas para programar deben incentivar?**

*Tabla 14. Habilidades que deben fomentar las plataformas para programar*

ITEM DE RESPUESTA	CANTIDAD	PORCENTAJE
Razonamiento	60	72,3%
Autonomía	38	45,8%
Creatividad	45	54,2%
Pensamiento Crítico	49	59%
Imaginación	24	28,9%
Resolución de problemas	55	66,3%
Agilidad	1	1,2%
TOTAL	83	100%

Elaboración propia.



*Figura 24. Habilidades que deben fomentar las plataformas para programar*

De las encuestas realizadas, y con respecto a la pregunta. De las siguientes habilidades, ¿cuál o cuáles cree usted que las plataformas para programar deben incentivar?; se obtuvieron los siguientes resultados: el 72,3% de los encuestados manifestaron que la habilidad principal que deben incentivar los programas para aprender programación es el razonamiento, el 66,3% de los estudiantes indicaron que la habilidad que debería de incentivar es la resolución de problemas, el 59% de los encuestados manifestaron que

las herramientas deberían incentivar el pensamiento crítico, el 54% de los estudiantes mencionaron que la creatividad debería incentivarse al emplear los entornos para programar, el 45% indicaron que se debería incentivar la autonomía, el 28,9 manifestaron que la imaginación debería incentivarse al emplear este tipo de programas y por último el 1,2% manifestaron otras habilidades que se deberían de incentivar.

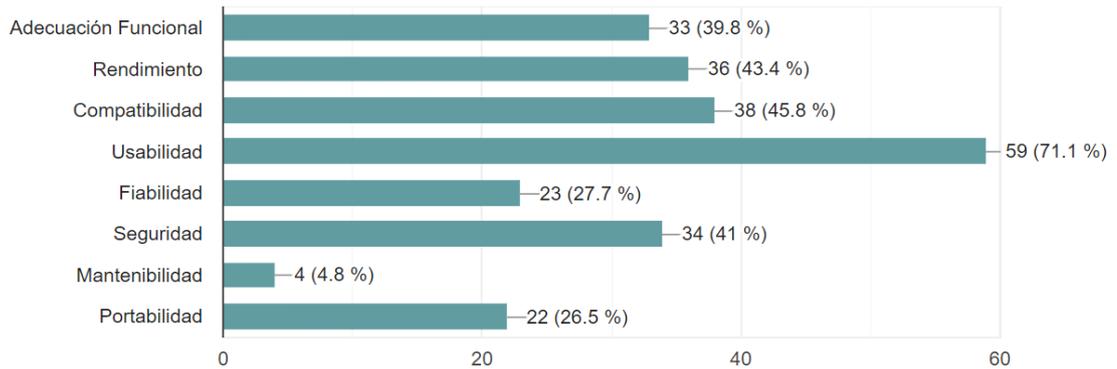
Concluyendo que es indispensable que las plataformas que apoyen el proceso de enseñanza-aprendizaje de la programación deban incentivar 3 habilidades principales, las cuales son; razonamiento, resolución de problemas y pensamiento críticos, sintetizando son las habilidades que todo programador debe de contar puesto a que la rama apunta a la r5esolución de problemas por medio de la automatización.

**12. Según su criterio seleccione las 3 características que considera más importantes que deben cumplir las herramientas para programar empleadas en la carrera.**

*Tabla 15. Características que deberían tener las plataformas de programación*

ITEM DE RESPUESTA	CANTIDAD	PORCENTAJE
Adecuación Funcional	33	39,8%
Rendimiento	36	43,4%
Compatibilidad	38	45,8%
Usabilidad	59	71,1%
Fiabilidad	23	27,7%
Seguridad	34	41%
Mantenibilidad	4	4,8%
Portabilidad	22	26,5%
TOTAL	83	100%

Elaboración propia.



*Figura 25. Características que deberían tener las plataformas*

De las encuestas realizadas, y con respecto a la pregunta. Según su criterio seleccione las 3 características que considera más importantes que deben cumplir las herramientas para programar empleadas en la carrera.; se obtuvieron los siguientes resultados: el 71,1% de los estudiantes indicaron que la usabilidad es la característica principal que debe de contar una plataforma, el 45,8% mencionaron que la compatibilidad debería considerarse como la segunda características que debe de contemplar un programa, el 43,4% de los encuestados consideran que el rendimiento es la tercera característica que debería contar, el 41% de los estudiantes consideran que la seguridad es la cuarta característica que debe de evidenciar cualquier software de desarrollo, el 39,8% consideran que la adecuación funcional es la quinta característica que debe de validar, el 27,7% indican que la fiabilidad es la sexta característica que debe de contemplar cualquier plataforma para aprender programación, el 26,5% de los encuestados indican que la portabilidad deberá ser la séptima característica que debe de tener un software y por último el 4,8% de los estudiantes encuestados manifestaron que la mantenibilidad debe de ser la última característica de este tipo de programas.

Concluyendo que según los resultados las tres características que los estudiantes consideran importante y por ende las que deben primar en un software son, usabilidad, compatibilidad y rendimiento; evidenciando la importancia para los estudiantes de plataformas que cumplan con la totalidad de las características anteriormente mencionadas.

### 13. ¿Tiene usted conocimiento de la existencia de normas que permiten evaluar la calidad de un software?

Tabla 16. Nivel de conocimiento de la existencia de normas de calidad de software

ITEM DE RESPUESTA	CANTIDAD	PORCENTAJE
SI	54	65,1%
NO	29	34,9%
TOTAL	83	100%

Elaboración propia.

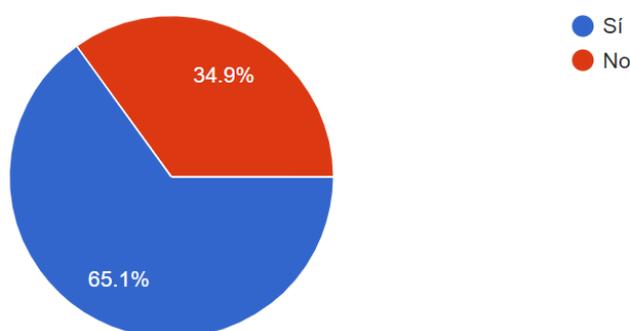


Figura 26. Nivel de conocimiento de la existencia de normas de calidad de software

De las encuestas realizadas, y con respecto a la pregunta: ¿Tiene usted conocimiento de la existencia de normas que permiten evaluar la calidad de un software?; se obtuvieron los siguientes resultados: el 65,1% conoce de la existencia de normas de calidad que permitan evaluar un software, y el 34,9% desconocen.

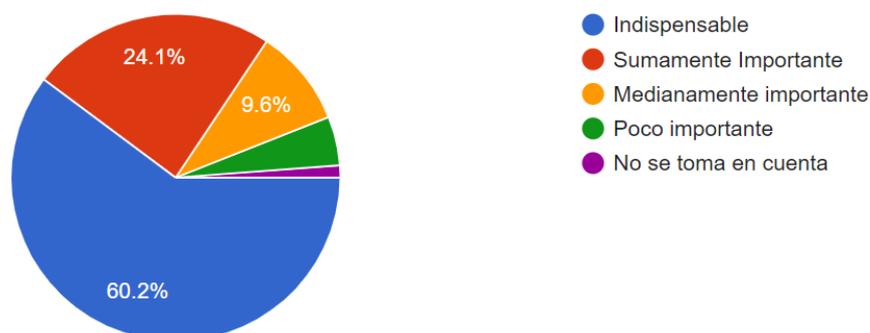
Permitiendo concluir que el desarrollo de un modelo para evaluar software inspirados en el modelo game based-learning será de utilidad, para la elección de las mejores plataformas que serán de apoyo al proceso de aprendizaje.

**14. ¿Considera usted importante que las herramientas que se empleen en la carrera cumplan con estándares de calidad ISO?**

*Tabla 17. Nivel de conocimiento de la existencia de normas de calidad de software*

ITEM DE RESPUESTA	CANTIDAD	PORCENTAJE
Indispensable	50	60,2%
Sumamente importante	20	24,1%
Medianamente importante	8	9,6%
Poco importante	4	4,8
No se toma en cuenta	1	1,2%
<b>TOTAL</b>	<b>83</b>	<b>100%</b>

Elaboración propia.



*Figura 27. Importancia de la certificación de las herramientas bajo una norma de calidad*

De las encuestas realizadas, y con respecto a la pregunta: ¿Considera usted importante que las herramientas que se empleen en la carrera cumplan con estándares de calidad ISO?; se obtuvieron los siguientes resultados: el 60,2% de los estudiantes considera indispensable que las plataformas que se emplean durante la carrera estén certificadas con estándares de calidad, el 24,1% consideran que es sumamente importantes, el 9,6% considera medianamente importante, el 4,8% considera que es poco importante y por último el 1,2% considera que no se toma en cuenta si las plataformas están avaladas bajo estándares de calidad.

Concluyendo, que la mayoría de los estudiantes consideran indispensable que todas las herramientas a emplear cumplan con normas de calidad de software, puesto que de esta manera se conocerán las más óptimas y las que cubren con todas las características que necesita un estudiante que está iniciando su carrera universitaria afín al desarrollo del software.

### 3.6. Análisis de resultados de las entrevistas

Una vez realizadas las entrevistas a los docentes referente a la aplicabilidad de nuevas plataformas en el proceso enseñanza-aprendizaje de la programación, se llegó a las siguientes conclusiones:

Según la experiencia en la docencia impartiendo materias relacionados a la programación, consideran que las herramientas que se emplean en el proceso enseñanza-aprendizaje ofrecen una amplia gama de IDEs para desarrollo de aplicaciones. Lo cual favorece y apoya el proceso de enseñanza buscando desarrollar habilidades de programación en los estudiantes. Dando apertura a conocer que las herramientas que se emplean permiten convertir al alumno en resolucionador de problemas a través de la experimentación y la prueba-error, direccionando los esfuerzos del estudiante a la solución de problemas más que a la sintaxis del lenguaje de programación empleado.

Siendo las herramientas empleadas el recurso principal para la aplicabilidad de los conocimientos impartidos, por tanto, como lo referencian los docentes en la carrera se tienen seleccionadas algunas herramientas, sin embargo, se les otorga a los estudiantes la flexibilidad de que aborde nuevas herramientas que le permitan de mejor manera acercarse a la solución de un problema de programación planteado. Es así como se puede concluir de acuerdo a lo mencionado, los estudiantes tienen abierta la posibilidad de emplear cualquier otra plataforma que apoye a la interiorización de los tópicos impartidos; es así como también si los docentes consideran necesario emplear otra herramienta que no haya sido definida por la carrera no presentan limitaciones al momento de incluirla en el proceso de enseñanza-aprendizaje.

Por tanto, resulta importante indagar sobre si las herramientas externas que se llegasen en algún momento a emplear cumplen con el objetivo para lo que fueron incluidas. Es por tanto que según lo comentado por dos docentes mencionan que *game-based learning* es una técnica en la cual los contenidos teóricos son presentados por medio de un videojuego. Pero les resulta un poco conflictivo el hecho de no conocerlas más a fondo y comprender sus ventajas y

beneficios a la enseñanza de la programación, pero no descartan el hecho de que aplicarlas les resulte útil como apoyo en la aplicación de los conocimientos.

Considerando que emplear este tipo de plataformas mejoran la retención de conocimientos por parte del estudiante, pero como lo menciona un docente es necesaria una elección apropiada de escenarios que resulten significativos y tomados de contextos de un perfil laboral. Es así como también indican que para emplearlas se tendría que evaluar el impacto técnico y pedagógico al incluir esta nueva técnica. Lo que va ligado con la validación de las mismas a través de un modelo de calidad que garantice que cumpla con características internas y externas para aplicarlas al proceso de enseñanza.

Concluyendo que la aplicabilidad de nuevas herramientas según los entrevistados está tomando más fuerza puesto que con la contribución y evolución de las tecnologías de la información han sido adecuadas para que los estudiantes fomenten diversas habilidades inmersas en la educación remota, aparte de la absorción y refuerzo de los conocimientos impartidos durante las clases. De esta forma han surgido diversos métodos de aprendizaje, tal como el *game based-learning* que apoya y mejora el aprendizaje, y sobre todo al ser herramientas free son accesibles para todos los involucrados en el proceso de enseñanza-aprendizaje; pero es necesario que estén certificadas bajo normas de calidad para asegurar su efectividad interna como externa cumpliendo con los objetivos para lo que fue diseñada.

### 3.7. Análisis del modelo *game based-learning*

El modelo *game based-learning* nace de la evolución de las tendencias de aprendizaje tradicional, que combina el aprendizaje teórico como el experimental, a través de la práctica; pero va más allá, esto se debe al hecho de que el aprendizaje práctico tradicional está quedando obsoleto a causa del surgimiento de diversas herramientas que permiten que los usuarios aprendan de forma divertida por medio de la incorporación del juego como didáctica de aprendizaje.

Es así como se recogieron las mejores características de todos los modelos de aprendizaje que han surgido hasta la actualidad, logrando un modelo fortalecido, que potencia todas las áreas del conocimiento. A continuación, se presenta el análisis de sus características.

- **Atractivo:** A este modelo se lo ha considerado atractivo en donde prima la presencia de escenarios con una interfaz amigable para el estudiante, cuya finalidad ha sido diferenciarse de los programas tradicionales.
- **Evolutivo:** La información con el paso del tiempo va evolucionando por ende las plataformas están en constante actualización de acuerdo a las últimas tendencias relacionadas a la trascendencia de la currícula implementada en la misma, la cual recoge mejoras para la superación de diferentes incógnitas que puedan surgir durante la implementación.
- **Motivador:** Emplear plataformas bajo este modelo permite crear una experiencia de aprendizaje positiva, incentivando el querer seguir aprendiendo puesto a que se potencializa el esfuerzo y la narrativa a través de los elementos gamificados.
- **Inspirador:** Los escenarios aplicados son significativos para la sociedad, en donde se ejemplifican situaciones de la vida real para que los usuarios entren en contexto de habitual recurrencia.
- **Reflexivo:** Ayuda a la asociación de los problemas presentados y por ende se fomentan habilidades del pensamiento crítico y la resolución de problemas, convirtiéndolos en reflexivos ante situaciones de toma de decisiones.

- Práctico: La relación de los personajes permite una interacción más real dando la alusión de ser parte del juego.
- Colaborativo: El trabajo colaborativo que se fomenta en el modelo da paso al desarrollo de habilidades del trabajo en equipo, en donde ayudan al usuario a identificar sus fortalezas para perfeccionarlas.

Es por tales características que este modelo de aprendizaje se ha convertido en la nueva modalidad de las herramientas para interiorizar los conocimientos, puesto a la presencia de propiedades didácticas. Por ello aplicarlo trae consigo un sin número de ventajas que según contraste con diversas investigaciones, mejoran el proceso de aprendizaje, es así como a continuación se definen las mismas.

#### Motiva al alumno

Tal como se lo definió como una de las características, el modelo a través de una interfaz amigable fomenta diversas habilidades, entre las cuales están:

-Autocontrol: la capacidad para controlar la motivación y emociones mientras se establecía la meta de terminar una actividad específica.

-Autoeficacia: permite que los estudiantes resuelvan sus problemas, lo que va de la mano con la autonomía.

#### Ayuda a razonar y ser autónomo

La reflexión conlleva una parte importante en el desarrollo de habilidades en donde el alumno debe reflexionar y tomar decisiones y resolver fallas, ligado con el desarrollo de capacidades cognitivas a través del pensamiento crítico, el análisis del entorno y la resolución de problemas.

#### Permite el aprendizaje activo

Da la apertura para ejercitar y poner en práctica los conocimientos impartidos de forma teórica, puesto que en ese momento tiene la posibilidad de realizar pruebas y si es el caso corregirlas, en donde

establece relaciones entre los conocimientos previos y nuevos para tomar decisiones que mejoren el resultado final.

#### Da al alumno el control de su aprendizaje

Mediante el juego el estudiante logra una retrospectiva de lo enseñado durante el proceso de aprendizaje sobre una temática, permitiéndole ser consciente de su grado de adquisición de lo aprendido y poder retroalimentarse en los conocimientos débiles.

#### Proporciona información útil al profesor

La realización de actividades o problemas didácticos que deban de resolver en la plataforma, arrojan una puntuación final, es así como le permite al docente detectar fortalezas y debilidades respecto a un tema específico o para comprobar el nivel de comprensión de los conocimientos. Lo que resulta relevante para que de acuerdo a ello el docente tenga un acercamiento más profundo al alumno, en cuanto a su capacidad de razonar y toma de decisiones.

#### Potencia la creatividad y la imaginación

La aplicación de un juego permite que los estudiantes a su vez tengan el control de poner en práctica lo aprendido, por ende, les da apertura a generar nuevas ideas, imaginar cosas que no existen, transformar ideas existentes en ideas nuevas y hacer asociaciones, lo que contribuye a abrir la mente del estudiante y su percepción de la realidad.

#### Fomenta las habilidades sociales

Tal como se lo describe en una de las características, fomenta el aprendizaje colaborativo por ende el estudiante tiene la oportunidad de interactuar y trabajar la educación emocional, el diálogo y la capacidad de líder y la colaboración por un objetivo en común.

#### Contribuye a la alfabetización digital

La aplicación de plataformas online aumenta los beneficios de la aplicación de las TIC en el aula, en donde de forma implícita los

estudiantes refuerzan los conocimientos sobre la temática del juego y mejoran su manejo de las nuevas tecnologías.

Dando como conclusión que el modelo en estudio es aplicable en el proceso enseñanza-aprendizaje puesto a su completitud de acuerdo a las características descritas con anterioridad. De igual forma las ventajas que presenta frente a los demás modelos de enseñanza lo posiciona como un modelo completo que engloba las nuevas habilidades que los estudiantes deben de desarrollar con la aparición de herramientas tecnológicas que siguen surgiendo como contribución de las TIC. Por tanto, resulta factible aplicar dicho modelo puesto que los estudiantes están en constante manipulación con la tecnológica y resultaría sencillo incorporarlo dentro de un proceso educativo.

## CAPÍTULO IV

### 4. PROPUESTA

#### MODELO DE EVALUACIÓN PARA PLATAFORMAS *GAME BASED-LEARNING*

##### 4.1. Campo de Aplicación

El modelo de evaluación de calidad de software propuesto en el presente trabajo de investigación basado en un conjunto de métricas externas e internas surge a partir de la investigación de los modelos de calidad de software que han surgido hasta la actualidad que tienen como objetivo lograr valores que determinan la calidad de un producto de software.

Las métricas declaradas en el presente modelo han sido planteadas tomando como referencia la norma ISO 25000 y sus divisiones, tal como se presenta en el apartado 4.3, permitiendo determinar las que se adecuan mejor y permitan evaluar plataformas *game based-learning* para el aprendizaje de la programación.

Siendo relevante mencionar que, es posible que cualquier persona que emplee el modelo aplique o no una métrica de acuerdo al campo de estudio del producto de software siendo este un juego, puesto que el modelo cubre áreas que otros productos de software no tienen, por tanto, es necesario que, al excluir alguna métrica, el modelo cumpla la totalidad del puntaje de importancia de las métricas para lograr una valoración correcta.

##### 4.2. Características del Modelo de Evaluación

El desarrollo del modelo de evaluación propuesto para plataformas *game based-learning* alineadas al aprendizaje de la programación cuenta con propiedades para su definición y creación.

- Las características de calidad externa e interna están basadas en las normas de la familia ISO 25000.
- Las subcaracterísticas están adaptadas según el campo de aplicación del modelo.

- El modelo cuenta con métricas que evalúan aspectos internos como externos, pero dependiendo del caso es posible descartar alguna característica.
- La sumatoria de todas las subcaracterísticas deberá estar basada en un 100% para un correcto uso del modelo.
- Se determinaron niveles de importancia para la categorización de cada subcaracterísticas.
- Se han definido valores constantes para algunas de las métricas que son sujetas a cambios dependiendo de un parámetro específico de evaluación.
- La capacidad que tiene una plataforma sobre una característica ha sido determinada mediante el cumplimiento o no, asignándole un valor representativo.

### 4.3. Normas Referenciales

Las normas ISO que permitieron definir y construir el modelo de evaluación propuesto, y por ende se emplearon como referencia para el desarrollo de las métricas para evaluar plataformas *game based-learning* alineadas al aprendizaje de la programación.

- ISO/IEC 9126 Calidad del Producto-Modelo de Calidad
- ISO/IEC 9126 Calidad del Producto-Métricas Externas
- ISO/IEC 9126 Calidad del Producto-Métricas de Calidad en Uso
- ISO 25000 Requerimientos y Evaluación de Calidad del producto de software (Square)
- ISO 25010 Requerimientos y Evaluación de Calidad del producto de software (Square) - División para el Modelo de Calidad
- ISO 25020 Requerimientos y Evaluación de Calidad del producto de software (Square) - División para la Medición de Calidad
- ISO 25030 Requerimientos y Evaluación de Calidad del producto de software (Square) - División para los Requisitos de Calidad
- ISO 25040 Requerimientos y Evaluación de Calidad del producto de software (Square) - División para la Evaluación de Calidad

La aplicación de cada una de estas normas dio paso al entendimiento del objetivo de evaluación de cada característica, para de esta manera contemplar todas las aristas necesarias para que el resultado de la evaluación sea el más preciso para determinar la calidad de un producto de software.

#### **4.4. Estructura del Modelo de Evaluación**

El modelo se encuentra dividido como se lo referencia en la Figura 12. Estructura de la ISO 25000, permitiendo de una forma ordenada conocer cómo se compone el modelo, en el que por cada característica existen varias subcaracterísticas que permiten de forma detallada evaluar un aspecto específico del software. A su vez, cada una de ellas cuentan con una métrica que permite evaluar un parámetro de acuerdo al objetivo de la misma.

Siendo necesario recalcar que el resultado que se obtiene en cada métrica, está sujeto a variaciones puesto a que dependen de la cantidad de un parámetro específico de acuerdo a la fórmula, por ello en las métricas sujetas a variaciones se han definidos datos constantes para que la evaluación sea bajo los mismos valores con todos los productos de software y el resultado sea el más justo.

#### **4.5. Características para la aplicación del modelo**

El modelo propuesto, tal como se lo definió este compuesto por 8 características y 30 subcaracterísticas que permiten evaluar la calidad interna como externa de las plataformas. Por tanto, en la Tabla 18 se referencia como se compone la estructura de evaluación según el enfoque de aprendizaje, en donde se alinea al área de la programación.

Tabla 18. Características del modelo propuesto

<b>Características</b>	<b>Subcaracterísticas</b>
<b>Adecuación funcional</b>	Compleitud funcional
	Correlación funcional
	Pertinencia funcional
<b>Eficiencia de desempeño</b>	Comportamiento temporal
	Utilización de recursos
	Capacidad
<b>Compatibilidad</b>	Coexistencia
	Interoperabilidad
<b>Usabilidad</b>	Inteligibilidad
	Aprendizaje
	Operabilidad
	Protección frente a errores de usuario
	Estética
	Accesibilidad
<b>Fiabilidad</b>	Madurez
	Disponibilidad
	Tolerancia a fallos
	Capacidad de recuperación
<b>Seguridad</b>	Confidencialidad
	Integridad
	Autenticidad
	Responsabilidad
<b>Mantenibilidad</b>	Modularidad
	Reusabilidad
	Analizabilidad
	Capacidad de ser modificado
	Capacidad de ser probado
<b>Portabilidad</b>	Adaptabilidad
	Facilidad de instalación
	Capacidad de ser reemplazado

Elaboración propia

#### 4.6. Criterios de Inclusión y Exclusión de las plataformas

El modelo de evaluación propuesto está contemplado para evaluar plataformas *game based-learning* enfocadas a la enseñanza del área de la programación, por ello se realizó una recopilación de los productos software basados en juegos alineados a lo mencionado con anterioridad, tal como se lo describe en el Anexo 1. Plataformas *Game Based-Learning* de programación, en donde se definieron 46 aplicaciones basadas en dos modelos de programación.

Es así como para la evaluación del modelo de calidad propuesto se incluyeron las 9 plataformas para ordenador con base de programación de texto como se lo resumen en la Tabla 19, puesto a que su enfoque está dirigido a estudiantes de la educación superior puesto a la semejanza de los programas que se emplean en un ambiente laboral. De acuerdo a ello en el Anexo 3 de adjuntan las características de las plataformas evaluadas.

Tabla 19. Plataformas *game based-learning* de programación de texto

N°	NOMBRE
1	Code combat
2	Code monkey
3	Colobot
4	Robocode
5	Codingame
6	Tynker
7	Code avengers
8	Check io
9	Coded Arena

Elaboración propia

#### 4.7. Criterios para la determinación de las fórmulas

Las métricas definidas permitirán evaluar un parámetro de acuerdo a una subcaracterísticas, la misma que representará la variable a encontrar para darle una puntuación cuantitativa. Es para ello que por cada característica y por ende

cada métrica se han determinado fórmulas, para que según los datos se obtenga siempre la completitud de la misma. Por tanto, a continuación, se listan los criterios que fueron tomados para la definición de las fórmulas.

- Se contempló que, la formula evalué la calidad de una característica, más no la cantidad de un parámetro.
- Se determinó que, en la fórmula, todo será dividido para la cantidad total de la variable a evaluar, para que de esta manera el resultado no varíe por el aumento o disminución de la misma.
- Se determinó que la fórmula será dividida para 10, puesto que dicho valor representa el máximo es decir la excelencia.
- Para evaluar algunas características se definieron valores constantes dentro de la fórmula para la estandarización de parámetros de evaluación.
- Para fórmulas de las métricas que son evaluadas bajo un valor específico, se aplicó una regla de tres para de una forma más simple y eficiente lograr un resultado justo.

#### 4.8. Definición del Nivel de Importancia de las características

Las características que forman parte del modelo propuesto están definidas para evaluar cualquier tipo de plataforma *game based-learning* alineada a la enseñanza de la programación. Es por tanto que, según los resultados obtenidos en las encuestas y entrevistas, y de acuerdo a la relevancia de cada una de ellas según la investigación; se han definido niveles de importancia para asignarles a cada una de las características, para que de esta manera se pueda evaluar cada plataforma bajo el 100%.

*Tabla 20. Rangos de importancia de características del modelo.*

Nivel de importancia	Ponderación
ALTO	5
MEDIO	3
BAJO	1
NO APLICA	0

Elaboración propia

Siendo ALTO el nivel más importante, refiriendo a que esa característica es de vital importancia y que es indispensable que se cumple al 100%. El nivel de importancia MEDIO refiere al hecho de que no es tan imprescindible el cumplimiento de esas características, el nivel BAJO significa que la ausencia de esa característica no influye en la calidad de la plataforma, y por último NO APLICA se adjudica a características que no son evaluadas por parte de los usuarios finales o auditores.

#### **4.9. Métricas y Fórmulas de Evaluación**

El desarrollo del modelo propuesto está basado en características y subcaracterísticas que permiten evaluar las plataformas desde la perspectiva interna y externa. En donde la aplicabilidad del mismo está orientado a los usuarios finales de la plataforma. Por tanto, a continuación, se definen cada una de las métricas de evaluación.

##### **Adecuación funcional**

La plataforma tiene la capacidad de ser funcionalmente correcto y, por ende, satisface las necesidades de los usuarios, es decir que cumple con los requisitos para completar las tareas para lo que fue diseñado. Es así como en la Tabla 21 se divide en 3 subcaracterísticas para evaluar aspectos más detallados de la adecuación funcional, como sus métricas y fórmulas.

Tabla 21. Definición de métricas y fórmulas de la Adecuación Funcional

CARACTERÍSTICA	SUBCARACTERÍSTICAS	MÉTRICA	FÓRMULA
ADECUACIÓN FUNCIONAL	COMPLETITUD FUNCIONAL	Complejidad de la implementación funcional	$CF = \frac{((x - y) * 100)}{x}$ /10 x= Número total de funcionalidades y= Número de funcionalidades incorrectas
	CORRELACIÓN FUNCIONAL	Correlación entre las funcionalidades	$CF = \frac{((x - y) * 100)}{x}$ /10 x= Número total de funcionalidades y= Número total de funcionalidades no correlacionadas.
	PERTINENCIA FUNCIONAL	Complejidad de la Idoneidad Funcional	$PF = \frac{((x - y) * 100)}{x}$ /10 x= Número total de funcionalidades y= Número de funcionalidades no pertinentes

Elaboración propia.

- Complejidad funcional

Grado en el cual todas las funcionalidades cubren correctamente las tareas para lo que fueron diseñadas.

- Correlación funcional

Capacidad de la plataforma para relacionar sus funcionalidades y por ende exista un común denominador entre ellas.

- Pertinencia funcional

Grado en el cual todas las funcionalidades cubren los objetivos de enseñanza relacionado a la programación.

### **Eficiencia de desempeño**

La plataforma funciona de acuerdo a una cantidad específica de recursos bajo distintos escenarios sean de hardware o software, es decir como se desempeña al momento de realizar alguna acción. Es así como en la Tabla 22 se divide en 3 subcaracterísticas para evaluar aspectos más detallados de la eficiencia de desempeño, como sus métricas y fórmulas.

*Tabla 22. Definición de métricas y fórmulas de la Eficiencia de desempeño*

CARACTERÍSTICA	SUBCARACTERÍSTICAS	MÉTRICA	FÓRMULA
<b>EFICENCIA DE DESEMPEÑO</b>	COMPORTAMIENTO TEMPORAL	Tiempo de respuesta	$TPR = \frac{x_1 + x_2 + x_3}{3}$ $x = \text{Tiempo de respuesta en 1 minuto}$ $CT = (10 * TP) - [11 * (TP - 1)]$
	UTILIZACIÓN DE RECURSOS	Utilización de CPU	$UR = [x - (11 * (x/10 - 1))]$ $x = \text{Porcentaje del uso del procesador}$
	CAPACIDAD	Tamaño de memoria usada por el proceso	$C = (x / 10) - [11 * ((x/100) - 1)]$ $x = \text{Tamaño de la memoria usada por el proceso}$

Elaboración propia.

- Comportamiento temporal

Tiempos de respuesta cuando la plataforma lleva a cabo una función. En dicha característica se establecen tres tiempos promedios de respuesta en 1 minuto.

**Variables:** x1= Tiempo 1 de respuesta en 1 minuto

x2= Tiempo 2 de respuesta en 1 minuto

x3= Tiempo 3 de respuesta en 1 minuto

- Utilización de recursos

Cantidad de recursos utilizados cuando la plataforma lleva a cabo una función bajo condiciones determinadas. En dicha característica se contempla como variable el porcentaje de uso del procesador.

- Capacidad

Cantidad de memoria que emplea por el proceso que se está ejecutando de la plataforma.

## **Compatibilidad**

Capacidad que tienen dos o más plataformas para intercambiar información y/o llevar a cabo sus funciones cuando comparten el mismo entorno, sea este de hardware o software. Es así como en la Tabla 23 se divide en 2 subcaracterísticas para evaluar aspectos más detallados de la compatibilidad, como sus métricas y fórmulas.

Tabla 23. Definición de métricas y fórmulas de la Compatibilidad

CARACTERÍSTICA	SUBCARACTERÍSTICAS	MÉTRICA	FÓRMULA
COMPATIBILIDAD	COEXISTENCIA	Número de programas causantes de conflictos	$C = (10 * x) - [(10 - x) - (10 * x)]$ x= Número de programas causantes de conflictos
	INTEROPERABILIDAD	Capacidad de intercambiar información con otras plataformas o programas	I= Capacidad de intercambiar información con otras plataformas o programas

Elaboración propia.

- Coexistencia

Grado en el que la plataforma puede coexistir con otro software independiente, sea o no compartiendo recursos.

- Interoperabilidad

Grado en el que dos o más plataformas tienen la capacidad de intercambiar información y emplear la información que fue compartida. En la presente característica se evalúa si cumple o no con la misma, por ello se establece la ponderación de acuerdo a la tabla 24 descrita a continuación.

Tabla 24. Ponderación de cumplimiento de características

Cumplimiento	Ponderación
Afirmativo	10
Negativo	0

Elaboración propia.

## Usabilidad

Capacidad de la plataforma de ser comprendida, entendida, usada y resulte didáctica por parte del usuario final. Es así como en la Tabla 25 se divide en 6 subcaracterísticas para evaluar aspectos más detallados de la usabilidad, como sus métricas y fórmulas.

Tabla 25. Definición de métricas y fórmulas de la Usabilidad

CAR.	SUBCARACTERÍSTICAS	MÉTRICA	FÓRMULA
USABILIDAD	INTELIGIBILIDAD	Capacidad del software de detectar vacíos conceptuales	$I = \text{Capacidad del software de detectar vacíos conceptuales}$
	APRENDIZAJE	Número de características <i>game based-learning</i> que cumple una plataforma	$A = (10 * x) / 7$ x= Número de características <i>game based-learning</i> que cumple.
	OPERABILIDAD	Claridad en los mensajes	$O = [ ((x - y) * 100) / x ] / 10$ x= Número total de mensajes y= Número de mensajes no entendibles
	PROTECCIÓN FRENTE A ERRORES DE USUARIO	Verificación de entradas validadas	$PEU = [ ((x - y) * 100) / x ] / 10$ x= Número total de entradas y= Número de entradas sin validación
	ESTÉTICA	Número de principios de la evaluación heurística de la interfaz de usuario	$E = (10 * x) / 12$ x= Número de principios de estética que cumple
	ACCESIBILIDAD	Número de Sistemas operativos compatibles	$A = (10 * x) / 4$ x= Número total de sistemas operativos compatibles

Elaboración propia.

- Inteligibilidad

Capacidad de la plataforma para detectar de acuerdo a los resultados obtenido por el usuario la deficiencia de conceptos. En la presente característica se evalúa si cumple o no con la misma, por ello se establece la ponderación de acuerdo a la Tabla 23. Ponderación de cumplimiento de características.

- Aprendizaje

Capacidad de la plataforma de cumplir con las características que potencialicen las habilidades de la tendencia *game based-learning*. Las características se encuentran listadas en la Figura 2. Características de las plataformas *game based-learning*.

- Operabilidad

Capacidad de la plataforma para ser de fácil uso a través de mensajes para la aclaración de cualquier punto que pueda ser sujeto a dudas.

- Protección frente a errores de usuario

Capacidad de la plataforma para proteger a los usuarios a través de validaciones y evitar posibles errores.

- Estética

Grado en el que la plataforma cumple con los principios heurísticos de la estética en la interfaz de usuario. En la Tabla 26 se encuentran listados.

Tabla 26. Principios del modelo heurístico de la Interfaz de usuario

**Principios del modelo heurístico de la Interfaz de usuario**

Posibilidad de atajos

Feedback informativo

Diseñar diálogos de finalización de procesos

Permitir la recursividad de acciones

Apoyar los controles internos

Usar un diálogo simple y natural

Buscar la consistencia en la terminología y acciones requeridas

Mantener al usuario informado sobre lo que el sistema está haciendo

Proporcionar mecanismos de salida de emergencia

Dar mensajes de error inequívocos

Prevenir errores mediante una disposición de opciones clara y fácil

Proporcionar ayuda online, instrucciones y documentación de forma clara y concisa

---

Elaboración propia.

- Accesibilidad

Capacidad de la plataforma de ser accesible a distintos entornos de software. En dicha característica se contemplan la cantidad de sistemas operativos, es por tanto que en el Anexo 2, se listan todos los sistemas operativos, pero es necesario mencionar que solo se incluyeron como se lo lista en la Tabla 27 los más empleados en el Ecuador según las estadísticas del portal StatCounter GlobalStats.

Tabla 27. Sistemas Operativos más usados en Ecuador

Sistemas operativos para PC
Windows
Mac OS
Unix
Chrome OS

Elaboración propia.

### Fiabilidad

Capacidad de la plataforma para desempeñar una función específica, cuando se usa bajo unas condiciones o periodos de tiempo determinados. Es así como en la Tabla 28 se divide en 4 subcaracterísticas para evaluar aspectos más detallados de la fiabilidad, como sus métricas y fórmulas.

Tabla 28. Definición de métricas y fórmulas de la Fiabilidad

CAR.	SUBCARACTERÍSTICAS	MÉTRICA	FÓRMULA
FIABILIDAD	MADUREZ	Número de fracasos en casos de prueba	$M = [ ((x - y) * 100) / x ] / 10$ x= Número total de pruebas realizadas y= Número total de fallas detectadas
	DISPONIBILIDAD	Tiempo de operación de servicios	$D = [ ((x - y) * 100) / x ] / 10$ x= Tiempo de uso en la plataforma en segundos y= Tiempo de no disponibilidad de la plataforma
	TOLERANCIA A FALLOS	Tiempo pérdida por fallas	$TF = (x - y) * NI$ x= Tiempo total de uso de la plataforma en minutos y= Sumatoria de tiempos de duración fallos
	CAPACIDAD DE RECUPERACIÓN	Tiempo promedio de recuperación	$TPR = (x_1 + x_2 + x_3) / 3$ x = Tiempo de respuesta en 1 minuto $CR = (10 * TP) - [11 * (TP - 1)]$

Elaboración propia.

- Madurez

Capacidad de la plataforma para detectar fallas que determinen la completitud de la misma a través de pruebas de validación.

- Disponibilidad

Capacidad de la plataforma o componente de estar operativo para su uso cuando se requiere en un periodo de tiempo específico.

- Tolerancia a fallos

Capacidad de la plataforma para operar según lo previsto en presencia de fallos, sean estos de hardware o software en un periodo de tiempo específico.

- Capacidad de recuperación

Capacidad de la plataforma de recuperarse ante un fallo inesperado y por ende reestablecerse en un período de tiempo específico. En dicha característica se establecen tres tiempos promedios de respuesta en 1 minuto.

**Variables:** x1= Tiempo 1 de respuesta en 1 minuto

x2= Tiempo 2 de respuesta en 1 minuto

x3= Tiempo 3 de respuesta en 1 minuto

## **Seguridad**

Capacidad de la plataforma para proteger la información y los datos y que por ende personas no autorizados no puedan leerlos o modificarlos. Es así como en la Tabla 29 se divide en 4 subcaracterísticas para evaluar aspectos más detallados de la seguridad, como sus métricas y fórmulas.

Tabla 29. Definición de métricas y fórmulas de la Seguridad

CARACTERÍSTICA	SUBCARACTERÍSTICAS	MÉTRICA	FÓRMULA
<b>SEGURIDAD</b>	CONFIDENCIALIDAD	Número de métodos de recuperación y validación de datos	$C=(x - y) * NI$ x= Número total de formas para recuperación o validación de datos y= Número de formas de recuperación o validación de datos no válidas en la plataforma
	INTEGRIDAD	Número de solicitudes vs mensajes de alerta recibidos	x= Número de solicitudes de cambio y= Número de mensajes de alerta recibidos
	AUTENTICIDAD	Número de formas de autenticación	$A= (x - y)*NI$ x= Número de formas posibles de autenticación y= Número de formas que no permite autenticación
	RESPONSABILIDAD	Capacidad de la plataforma de rastrear ingresos maliciosos	x= Número de ingresos malicioso y= Número de rastreos exitosos

Elaboración propia.

- Confidencialidad

Formas de protección contra el acceso de datos e información por personas externas no autorizadas. En la Tabla 30 se definen las formas de recuperación de datos, variable en dicha característica.

Tabla 30. Formas de Recuperación de datos

Formas de Recuperación de datos
Pregunta de Seguridad
Código mediante SMS
Código mediante SMTP
Imagen de Seguridad
A través de una URL vía SMS
A través de una URL vía SMTP
Pin de Seguridad

Elaboración propia.

- Integridad

Capacidad de la plataforma para notificar accesos o modificaciones por personas externas no autorizadas. En dicha subcaracterística se valida si una variable es menor o igual a otra definida, por tanto, de acuerdo a ello se establece dos comparaciones:

-Si se cumple que  $x=y$  es afirmativo

-Si se cumple  $x<y$  es negativo

Por ende, la ponderación es de acuerdo la Tabla 24. Ponderación de cumplimiento de características.

- Autenticidad

Formas que tiene la plataforma para demostrar la identidad correcta de un usuario. En la Tabla 31 se definen las formas de recuperación de datos, variable en dicha característica.

Tabla 31. Formas de Autenticación de usuario

Formas de Autenticación
Inicio de sesión tradicional
Identificador en línea
Mediante SMS OTP
A través de una URL vía SMS
A través de una URL vía SMTP
Alias o Nombre de Usuario
Código de Clase

Elaboración propia.

- Responsabilidad

Capacidad de la plataforma para rastrear de forma precisa las acciones maliciosas de una persona externa no autorizada. En dicha subcaracterística se valida si una variable es menor o igual a otra definida, por tanto, de acuerdo a ello se establece dos comparaciones:

-Si se cumple que  $x=y$  es afirmativo

-Si se cumple  $x<y$  es negativo

Por ende, la ponderación es de acuerdo la Tabla 24. Ponderación de cumplimiento de características.

## Mantenibilidad

Capacidad de la plataforma para proteger la información y los datos y que por ende personas no autorizados no puedan leerlos o modificarlos. Es así como en la Tabla 32 se divide en 5 subcaracterísticas para evaluar aspectos más detallados de la mantenibilidad, como sus métricas y fórmulas.

Tabla 32. Definición de métricas y fórmulas de la Mantenibilidad

CARACTERÍSTICA	SUBCARACTERÍSTICAS	MÉTRICA	FÓRMULA
<b>MANTENIBILIDAD</b>	MODULARIDAD	Número de módulos concatenados	$M = [ ((x - y) * 100) / x ] / 10$ x= Número total de módulos y= Número de módulos no concatenados
	REUSABILIDAD	Número de atributos y funciones heredadas	$R = (x - y) * NI$ x= Suma de atributos y funciones de todos los objetos del software y= Suma de atributos y funciones heredadas de todos los objetos
	ANALIZABILIDAD	Fallas diagnosticadas	$A = [ ((x - y) * 100) / x ] / 10$ x= número total de fallas encontradas y=número total de fallas no diagnosticadas
	CAPACIDAD DE SER MODIFICADO	Complejidad de la modificación	$CM = x / y$ x= Número de modificaciones y= Tiempo que le toma al desarrollador hacer las modificaciones en minutos
	CAPACIDAD DE SER PROBADO	Número de reinicios para pruebas	$CP = [ ((x - y) * 100) / x ] / 10$ x= número de total de reinicios y= número de reinicios fallidos

Elaboración propia.

- Modularidad

Grado en el que la plataforma tiene separados todos sus componentes para que la afección en alguno tenga un impacto mínimo en los demás.

- Reusabilidad

Grado en el que el código de la plataforma puede ser utilizado en más de un sistema software o en la construcción de otros.

- Analizabilidad

Grado en el que la plataforma tiene la facilidad de encontrar fallas en el mismo y diagnosticar las causas de los fallos.

- Capacidad de ser modificado

Grado en el que la plataforma permite que sea modificado de forma eficiente sin afectar su desempeño.

- Capacidad de ser probado

Grado en el que la plataforma tiene la facilidad de reiniciar sus funciones y reestablecerse después de las pruebas realizadas.

## **Portabilidad**

Capacidad de la plataforma para ser transferida exitosamente a cualquier entorno, sea este de hardware o software. Es así como en la Tabla 33 se divide en 3 subcaracterísticas para evaluar aspectos más detallados de la mantenibilidad, como sus métricas y fórmulas.

Tabla 33. Definición de métricas y fórmulas de la Portabilidad

CARACTERÍSTICA	SUBCARACTERÍSTICA	MÉTRICA	FÓRMULA
PORTABILIDAD	ADAPTABILIDAD	Capacidad de adaptarse a otros entornos	A= Capacidad de adaptarse a otros entornos Tabla de cumplimiento
	FACILIDAD DE INSTALACIÓN	Número de pasos previo a la de Instalación	$FI = (10 * x) - [11 * (x - 1)]$ x= Número de pasos para la instalación
	CAPACIDAD DE SER REEMPLAZADO	Capacidad de ser reemplazado	CR=Capacidad de ser reemplazado Tabla de Cumplimiento

Elaboración propia.

- Adaptabilidad

Capacidad del producto para ser adaptado de forma exitosa a diferentes entornos determinados de hardware o software.

- Facilidad de instalación

Grado en el que la plataforma tiene la facilidad para ser instalado de forma exitosa.

- Capacidad de ser reemplazado

Capacidad de la plataforma para que pueda ser utilizado en lugar de otro software con el mismo propósito.

#### **4.10. Pasos para aplicar el modelo de evaluación propuesto**

##### **4.10.1. Definir requisitos de evaluación**

El presente modelo de evaluación está orientado para evaluar plataformas *game based-learning* alineadas a la programación, por lo tanto, está compuesto por características externas, internas y de operación para determinar la calidad de las mismas. Por ende, previo a la aplicación del modelo es indispensable definir los requisitos de evaluación, sean estos de tipo interno, externo o de calidad en uso; para de acuerdo a ello definir las características que se alinean a dichos requisitos.

##### **4.10.2. Definir el nivel de importancia por cada característica**

Definidos los requisitos y por ende las características que formarán parte de la evaluación, se deben definir niveles de importancia por cada una de ellas, de acuerdo a la Tabla 20. Rangos de importancia de las características del modelo propuesto; en donde la sumatoria de dichos valores deben sumar un total de 100 para evaluar la plataforma de forma eficiente en toda su completitud.

##### **4.10.3. Establecer valores constantes para algunas fórmulas**

Las fórmulas definidas en el modelo están contempladas para evaluar la calidad de un aspecto relacionado a la plataforma, por ello en algunos casos se han definido valores constantes para tener una equidad en los resultados, y que estos no varían de acuerdo a variables externas o parámetros de evaluación. Por tanto, se deben definir los mismos valores constantes en las métricas en donde se requiere establecer un intervalo de tiempo para la evaluación de una característica.

##### **4.10.4. Llenar el modelo con los datos Registrar los Resultados**

Una vez que se han contemplado todas las premisas en el modelo, se debe de seleccionar el tipo de plataforma sea esta de tipo

texto o bloque, y de acuerdo a ello seleccionar que plataforma se va a evaluar. Luego rellenar todos los campos editables de cada característica, para luego dar clic sobre el botón “Registrar”. Es así como de forma automática se realizarán los cálculos y el resumen se colocará en la pestaña de Resumen de Evaluación. De esta forma al término de la evaluación de todas las plataformas en estudio se visualizará la puntuación final de cada una de ellas para de acuerdo al mismo determinar su calidad de acuerdo a la ponderación.

#### 4.11. Análisis del Modelo de Evaluación Propuesto

El modelo propuesto trae consigo una estructura para de acuerdo a una puntuación final, determinar la completitud externa e interna de las plataformas *game based-learning* en estudio alineadas a la programación. En la cual tal como se lo detalla en el apartado de la Propuesta está compuesto por ocho características y treinta subcaracterísticas, cada una con su respectiva métrica, fórmula y nivel de importancia, lo que permitió evaluarlas bajo una puntuación final de 1000.

Es así como se han definido niveles de aceptación para clasificar las plataformas, tal como se lo describe a continuación en la Tabla 34.

*Tabla 34. Niveles de aceptación de las plataformas game based-learning*

Nivel de aceptación	Rango
Excelente	1000 - 800
Buena	799 - 600
Regular	< 599

Elaboración propia.

Una vez obtenidos los resultados parciales por cada una de las características, se procede a sumar todos los valores, para determinar la

valoración final; es así como se realizó el cálculo de todas y cada una de las plataformas en estudio como se lo aprecia en la Tabla 36.

De acuerdo a los resultados obtenidos y los niveles de aceptación se clasificaron las plataformas, para determinar las aplicables a procesos educativos de educación superior, tal como se lo evidencia en la Tabla 35.

*Tabla 35. Clasificación de las plataformas evaluadas según el modelo de calidad propuesto*

Nivel de aceptación	Plataformas
Excelente	Codingame Tynker Code Avengers Code Monkey
Buena	Coded Arena Check IO Colobot Code Combat
Regular	Robocode

Elaboración propia.

Tabla 36. Resumen de Evaluación de las plataformas game based-learning

	NOMBRE	Coded Arena	Check IO	Code Avengers	Tynker	Codingame	Robocode	Colobot	Code Monkey	Code Combat
<b>ADECUACIÓN FUNCIONAL</b>	<b>Complejidad funcional</b>	50	50	50	50	50	50	50	50	50
	<b>Correlación funcional</b>	30	30	30	30	30	30	30	30	30
	<b>Pertinencia funcional</b>	50	50	50	50	50	50	50	50	50
<b>EFICENCIA DE DESEMPEÑO</b>	<b>Comportamiento temporal</b>	36.66	20	41.66	40	41.66	20	40	43.33	36.66
	<b>Utilización de recursos</b>	22.1	39	40.65	24.6	43.8	16.6	42.7	21.9	22
	<b>Capacidad</b>	29.68	39.54	39.65	33.12	30.37	17.28	42.87	32.58	29.34
<b>COMPATIBILIDAD</b>	<b>Coexistencia</b>	30	30	30	30	30	24	30	30	30
	<b>Interoperabilidad</b>	0	0	0	0	0	50	0	50	0
<b>USABILIDAD</b>	<b>Inteligibilidad</b>	0	0	30	0	30	0	30	30	0
	<b>Aprendizaje</b>	35.71	21.42	50	50	50	14.28	28.57	50	35.71
	<b>Operabilidad</b>	30	21.42	26.25	30	30	23.33	23.47	30	26.11
	<b>Protección frente a errores de usuario</b>	50	43.75	50	50	50	50	50	50	50
	<b>Estética</b>	33.33	33.33	50	50	45.83	29.16	29.16	37.5	37.5
	<b>Accesibilidad</b>	50	50	50	50	50	37.5	37.5	50	50
<b>FIABILIDAD</b>	<b>Madurez</b>	30	20	30	30	30	10	20	20	30
	<b>Disponibilidad</b>	42	39	45	45	45	42	42	45	45
	<b>Tolerancia a fallos</b>	40.2	41.1	42.6	45	45	42.6	43.2	41.1	43.2
	<b>Capacidad de recuperación</b>	3	20	22	20	23	8	20	25	18
<b>SEGURIDAD</b>	<b>Confidencialidad</b>	5	5	25	15	5	0	0	5	10

	<b>Integridad</b>	0	0	0	30	30	0	0	0	30
	<b>Autenticidad</b>	5	30	10	15	30	5	5	20	20
	<b>Responsabilidad</b>	0	0	0	30	0	0	0	0	30
<b>MANTENIBILIDAD</b>	<b>Modularidad</b>	0	0	0	0	0	0	0	0	0
	<b>Reusabilidad</b>	0	0	0	0	0	0	0	0	0
	<b>Analizabilidad</b>	0	0	0	0	0	0	0	0	0
	<b>Capacidad de ser modificado</b>	0	0	0	0	0	0	0	0	0
	<b>Capacidad de ser probado</b>	6.66	6.66	10	10	10	6.66	10	10	10
<b>PORTABILIDAD</b>	<b>Adaptabilidad</b>	30	30	30	30	30	21	30	30	30
	<b>Facilidad de instalación</b>	30	30	30	30	30	18	27	30	30
	<b>Capacidad de ser reemplazado</b>	30	30	30	30	30	30	30	30	30
<b>TOTAL</b>		669.36	680.24	812.82	817.72	839.67	595.43	711.49	811.41	773.53

Elaboración propia.

## 5. CONCLUSIONES Y RECOMENDACIONES

### 5.1. Conclusiones

La tendencia *game based-learning* demuestra su efectividad al contar con herramientas con características que las diferencian de otras, puesto que son atractivas, evolutivas, motivadoras, inspiradoras, reflexivas, prácticas y colaborativas permitiendo al estudiante desarrollar diversas habilidades.

Las plataformas *game based-learning* que se emplean en el proceso de enseñanza de la programación fomentan el pensamiento crítico, resolución de problemas y pensamiento computacional para permitir al estudiante tener el control de su aprendizaje a través de la autonomía que se logra durante el proceso educativo.

Se destaca que tanto para el estudiante como para el docente una plataforma *game based-learning* debe contar con una buena adecuación funcional, compatibilidad y portabilidad puesto que debe cumplir el propósito esperado con su aplicación y ser accesible a cualquier entorno de hardware o software.

La norma ISO 25000 contribuyó al diseño de un modelo de evaluación específico para validar la efectividad de plataformas *game based-learning*.

La aplicación del modelo de calidad en las plataformas en estudio, dio a conocer según las ponderaciones que el 44% de ellas son excelentes para ser aplicadas a procesos educativos puesto a su complejidad de los contenidos propuestos los cuales se alinean a currículas de educación superior, así como también por su completitud en las demás características definidas en la evaluación.

## 5.2. Recomendaciones

Se sugiere que las autoridades competentes relacionadas con procesos educativos de nivel superior se familiaricen con nuevas herramientas basadas en la tendencia game based-learning para aplicarlas y apoyar el aprendizaje de los estudiantes.

Se espera que los estudiantes involucrados dentro del proceso de aprendizaje de la programación logren alcanzar un nivel competitivo gracias a las habilidades desarrolladas gracias al uso de dichas plataformas.

Se debería motivar a la comunidad estudiantil a que innove en el desarrollo de plataformas game based-learning considerando la adecuación funcional, compatibilidad y portabilidad.

Se debería considerar que para la aplicación del modelo propuesto es importante clasificar las plataformas de acuerdo a su disponibilidad sea de hardware o software para una óptima evaluación.

Sería importante llevar el modelo de calidad propuesto a un software que permita evaluar la calidad de plataformas game based-learning y que ofrezcan resultados gráficos estadísticos que evidencien la calidad a nivel de cada característica.

## 6. BIBLIOGRAFÍAS

- Arias, F. G. (2006). El Proyecto de Investigación - Introducción a la metodología científica. In *Journal of Chemical Information and Modeling* (Vol. 53, Issue 9). Editorial Episteme.
- Ávila, W. (2013). Hacia una reflexión histórica de las TIC. *Hallazgos*, 10(19). <https://doi.org/10.15332/s1794-3841.2013.0019.13>
- Boeker, M., Andel, P., Vach, W., & Frankenschmidt, A. (2013). *Game-Based E-Learning Is More Effective than a Conventional Instructional Method: A Randomized Controlled Trial with Third-Year Medical Students*. <https://doi.org/10.1371/journal.pone.0082328>
- Calero, C., Moraga, M. Á., & Piattini, M. (2010). Calidad Del Producto Y Proceso Software. In *DFVCL*. <https://doi.org/ISBN:8478979611> ISBN-13: 9788478979615
- Callejas, M., Alarcón, A. C., & Álvarez, A. M. (2017). *Modelos de calidad del software, un estado del arte\**. 13(1). <https://doi.org/10.18041/entramado.2017v13n1.25125>
- Carrillo, B. A., Sánchez, M., & Leenen, I. (2020). Amenazas a la validez en evaluación: implicaciones en educación médica. *Investigación En Educación Médica*, 2(34), 100–107. <https://doi.org/10.22201/facmed.20075057e.2020.34.221>
- Carrizo, D., & Alfaro, A. (2018). Método de aseguramiento de la calidad en una metodología de desarrollo de software: Un enfoque práctico. *Ingeniare*, 26(1), 114–129. <https://doi.org/10.4067/S0718-33052018000100114>
- Casas, J., Repullo Labrador, J. R., & Donado Campos, J. (2003). La encuesta como técnica de investigación. Elaboración de cuestionarios y tratamiento estadístico de los datos (I). *Atención Primaria*, 31(8), 527–538. [https://doi.org/10.1016/s0212-6567\(03\)70728-8](https://doi.org/10.1016/s0212-6567(03)70728-8)
- Denham, A. R., Mayben, R., & Boman, T. (2016). Integrating Game-Based Learning Initiative: Increasing the Usage of Game-Based Learning Within K-12 Classrooms Through Professional Learning Groups. *TechTrends*, 60(1), 70–76. <https://doi.org/10.1007/s11528-015-0019-y>
- Deterding, S. (2012). Gamification: Designing for Motivation. *Interactions*, 19(4), 14. <https://doi.org/10.1145/2212877.2212883>
- Dichev, C., & Dicheva, D. (2017). Gamifying education: what is known, what is

- believed and what remains uncertain: a critical review. In *International Journal of Educational Technology in Higher Education* (Vol. 14, Issue 1, p. 9). Springer Netherlands. <https://doi.org/10.1186/s41239-017-0042-5>
- Dolly Cristina, P., & Rafael G., H. (2005). *Nómadas (Col)*. *Nómadas (Col)*, 140–150.  
<http://www.redalyc.org/articulo.oa?id=105116598012%0Ahttp://www.redalyc.org/articulo.oa?id=105146818003%0Ahttp://www.redalyc.org/articulo.oa?id=105127475015>
- Estrada, R. B., & Ovide, E. (2011). El impacto de las nuevas tecnologías en la educación en valores del siglo XXI. *Sinética Revista Electrónica de La Educación*, 37, 1–16.  
[http://www.scielo.org.mx/scielo.php?script=sci\\_arttext&pid=S1665-109X2011000200002](http://www.scielo.org.mx/scielo.php?script=sci_arttext&pid=S1665-109X2011000200002)
- Fernández, L., Pagés, C., & Rueda, M. J. (2015). *Proceso de definición de métricas y criterios de calidad en el ciclo de vida de la educación virtual accesible*. 104–111. [www.usability.gov](http://www.usability.gov)
- Fraioli, L., Bacchin, G., Gaudenzi, G. (Giacinto), & Stalio, I. (1999). *La historia de la tecnología: el hombre crea su mundo*. Editorial Editex.  
<https://books.google.com.bo/books?id=eJxenwMIUmoC&q=historia+de+la+tecnologia&dq=historia+de+la+tecnologia&hl=es-419&sa=X&ved=0ahUKEwjJ-PztnfbhAhWm1VkKHU1rD7YQ6AEIMjAC>
- García, F., Alvira, F., Alonso, B., & Escobar, M. (2016). *El análisis de la realidad social: métodos y técnicas de investigación* (4ta ed.).  
<https://dialnet.unirioja.es/servlet/libro?codigo=574494>
- Gardner, H. (2005). *Inteligencias múltiples: La teoría en la práctica* - Howard Gardner - Google Libros.  
[https://books.google.es/books?id=l\\_ntBgAAQBAJ&dq=inteligencias+múltiples+gardner&hl=es&source=gbs\\_navlinks\\_s](https://books.google.es/books?id=l_ntBgAAQBAJ&dq=inteligencias+múltiples+gardner&hl=es&source=gbs_navlinks_s)
- Hernández Sampieri, R., Fernández, C., & Baptista, P. (2010). *Metodología de la investigación* (5ta. ed.). DF. In *DF México: McGraw Hill*. [www.FreeLibros.com](http://www.FreeLibros.com)
- International Organization for Standardization, & International Electrotechnical Commission. (2014). *ISO/IEC DIS 25023:2014- Systems and software engineering — Systems and software Quality Requirements and Evaluation*

- (SQuaRE) — *Measurement of system and software product quality.*  
<https://www.iso.org/standard/35747.html>
- ISO/IEC. (2010). *{ISO}/{IEC} 25045:2010 {Systems} and software engineering -- {Systems} and software {Quality} {Requirements} and {Evaluation} ({SQuaRE}) -- {Evaluation} module for recoverability.*  
<https://www.iso.org/standard/35770.html>
- ISO/IEC. (2011a). *ISO/IEC 25022:2012 Systems and software engineering Systems and software Quality Requirements and Evaluation. 1.3.2011, 1–44.* <https://www.iso.org/standard/35746.html>
- ISO/IEC. (2011b). *ISO/IEC 25040:2011 Systems and software engineering — Systems and software Quality Requirements and Evaluation ( SQuaRE ) — Evaluation process (SO/IEC 25040:2011).*  
<https://www.iso.org/standard/35765.html>
- ISO/IEC. (2011c). *ISO - ISO / IEC 25040: 2011 - Ingeniería de sistemas y software - Requisitos y evaluación de calidad de sistemas y software (SQuaRE) - Proceso de evaluación. ISO.*  
<https://www.iso.org/obp/ui/#iso:std:iso-iec:25040:ed-1:v1:en>
- ISO/IEC. (2012). *ISO/IEC 25041:2012 Systems and software engineering -- Systems and software Quality Requirements and Evaluation (SQuaRE) -- Evaluation guide for developers, acquirers and independent evaluators.*  
<https://www.iso.org/standard/35766.html>
- ISO/IEC. (2015). *BS ISO/IEC 25024:2015 Systems and software engineering — Systems and software Quality Requirements and Evaluation ( SQuaRE ) — Measurement of data quality. BSI Standards Publication.*  
<https://www.iso.org/standard/35749.html>
- ISO/IEC 25020:2019 *Systems and software engineering — Systems and software quality requirements and evaluation (SQuaRE) — Quality requirements framework, (2019).* <https://www.iso.org/standard/72117.html>
- ISO/IEC 25030:2019 *Systems and software engineering — Systems and software quality requirements and evaluation (SQuaRE) — Quality requirements framework, (2019).* <https://www.iso.org/standard/72116.html>
- ISO/IEC. (2020). *ISO/IEC 25021:2012 Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — Quality measure elements.* <https://www.iso.org/standard/55477.html>

- Iso25000. (2018). *NORMAS ISO 25000*. ISO 25000. <https://iso25000.com/index.php/normas-iso-25000>
- Jacovkis, P. M. (2011). *Las TIC en América Latina: historia e impacto social* (No. 1668–0030). <https://www.redalyc.org/pdf/924/92422639003.pdf>
- Knutas, A., Ikonen, J., Nikula, U., & Porras, J. (2014). Increasing collaborative communications in a programming course with gamification: A case study. *ACM International Conference Proceeding Series*, 883, 370–377. <https://doi.org/10.1145/2659532.2659620>
- López, P. L. (2004). POBLACIÓN MUESTRA Y MUESTREO. *Punto Cero*, 09(08), 69–74. [http://www.scielo.org.bo/scielo.php?script=sci\\_arttext&pid=S1815-02762004000100012](http://www.scielo.org.bo/scielo.php?script=sci_arttext&pid=S1815-02762004000100012)
- Lozada Ávila, C., & Betancur Gómez, S. (2017). La gamificación en la educación superior: una revisión sistemática. *Revista Ingenierías Universidad de Medellín*, 16(31), 97–124. <https://doi.org/10.22395/rium.v16n31a5>
- Mathieu, M. J. (2014). *Introducción a la programación*.
- Migdely, M., Ochoa Ávila, B., Mario, M., Soa, V., Yovanni, L., & Aballe, Q. (2007). *Innovación, tecnología y gestión tecnológica*. [http://bvs.sld.cu/revistas/aci/vol16\\_4\\_07/aci081007.htm](http://bvs.sld.cu/revistas/aci/vol16_4_07/aci081007.htm)
- Ortíz Ocaña, A. L. (2013). *Modelos pedagógicos y teorías del aprendizaje*. [https://books.google.com.ec/books/about/Modelos\\_pedagógicos\\_y\\_teorías\\_del\\_apre.html?id=NT0jDwAAQBAJ&printsec=frontcover&source=kp\\_read\\_button&redir\\_esc=y#v=onepage&q&f=false](https://books.google.com.ec/books/about/Modelos_pedagógicos_y_teorías_del_apre.html?id=NT0jDwAAQBAJ&printsec=frontcover&source=kp_read_button&redir_esc=y#v=onepage&q&f=false)
- Programa de las Naciones Unidas para el Desarrollo. (2002). *LAS TECNOLOGÍAS DE LA INFORMACIÓN Y LA COMUNICACIÓN ALL SERVICIO DEL DESARROLLO*. Nº3, 249.
- Rahman, A. (2020). Quality Consideration for e-Learning System Based on ISO/IEC 25000 Quality Standard. In *undefined*.
- Schank, R. (2007). «Los colegios no deberían existir». <http://www.kindsein.com/es/21/1/485/>
- Tello, E. (2007). Information and Communication Technology (ICT) and the digital gap: their impact on Mexican society. *RUSC. Universities and Knowledge Society Journal*, 4(2). <https://doi.org/10.7238/rusc.v4i2.305>
- UNESCO. (2011). *América Latina y EL CARIBE, INFORME DE SEGUIMIENTO*

DE LA EPT EN EL MUNDO 2011. In *Cuadernos de Pedagogía* (p. 412).  
<https://unesdoc.unesco.org/ark:/48223/pf0000367436/PDF/367436spa.pdf>.  
multi

Ureña, C. (2011). *Lenguajes de Programación Capítulo 1. Introducción*.  
<https://docplayer.es/27319234-Lenguajes-de-programacion-capitulo-1-introduccion.html>

Vattimo, G. (2006). *La interpretación del mundo: cuestiones para el tercer milenio*.

[https://books.google.com.ec/books?hl=es&lr=&id=\\_8aEM2CIX3oC&oi=fnd&pg=PA7&ots=IU-V9bgZpT&sig=66e8AVOzwmsIVlzpKtY\\_NoOq0TI&redir\\_esc=y#v=snippet&q=incita&f=false](https://books.google.com.ec/books?hl=es&lr=&id=_8aEM2CIX3oC&oi=fnd&pg=PA7&ots=IU-V9bgZpT&sig=66e8AVOzwmsIVlzpKtY_NoOq0TI&redir_esc=y#v=snippet&q=incita&f=false)

# **ANEXOS**

### Anexo 1. Plataformas *Game Based-Learning* de programación

N°	NOMBRE	N°	NOMBRE
1	Scratch	24	Colobot
2	Lego mindstorms ev3	25	Robocode
3	Lego spike prime	26	Machineers
4	Code org	27	Codea
5	Code combat	28	Codemancer
6	Lightbot	29	Tis 100
7	The foos	30	Codeable crafts
8	Hopscotch	31	Codingame
9	Daisy the dinosaur	32	Python challenge
10	Kodable	33	Code avengers
11	Tynker	34	Check io
12	Mover la tortuga	35	Codemurai
13	Cargo bot	36	Code karts
14	Cato's hike	37	Toca blocks
15	Bit by bit	38	Bee bot
16	Robozzle	39	Blue bot
17	Think and learn code	40	Goldie blox
18	Grasshopper	41	Nancy drew: codes & clues
19	Swift playgrounds	42	Sprite box coding
20	Mimo	43	Coded arena
21	Py learn to code	44	Code adventures
22	Space chem	45	Blockly
23	Code monkey	46	Minecraft Education

Elaboración propia.

## Anexo 2: Sistemas Operativos

### Sistemas operativos para PC

Windows  
Mac OS  
Unix  
Solaris  
FreeBSD  
OpenBSD  
Android (GNU/Linux)  
Google Chrome OS (GNU/Linux)  
Debian GNU/Linux (GNU/Linux)  
Gentoo Linux (GNU/Linux)  
SUSE Linux (GNU/Linux)  
Red Hat Enterprise Linux (GNU/Linux)  
Ubuntu Linux (GNU/Linux)  
Elementary OS (GNU/Linux)  
Fedora (GNU/Linux)  
OpenSUSE (GNU/Linux)  
Sabayon (GNU/Linux)  
Wave OS  
webOS  
Haiku (BeOS)  
Plan 9  
Freespire  
HP-UX  
ReactOS  
BeOS  
LindowsOS/Linspire

---

Elaboración propia.

### Anexo 3. Comparativa de los modelos de Calidad a Nivel de Producto

Característica	Modelo McCall	Modelo de Boehm	Modelo de Furps	Modelo de Gilb	ISO/IEC 9126	Modelo SQA	Modelo WebQuem	ISO/IEC 25000
Adecuación								
Funcional	X		X		X	X	X	X
Rendimiento		X	X	X			X	X
Compatibilidad				X				X
Usabilidad	X	X	X	X	X		X	X
Fiabilidad		X	X				X	X
Seguridad	X				X	X		X
Mantenibilidad	X	X	X		X	X		X
Operabilidad	X	X		X	X	X		X

Elaboración propia.

#### Anexo 4. Caracterización de las plataformas *game based-learning* evaluadas

N	Nombre	Tipo de programación	Edad	Sistema operativo	Lenguaje de programación	Licencia	Link
1	Code combat	Texto	9+	Web	Python, Lua, Javascript	Free	<a href="https://codecombat.com">https://codecombat.com</a>
2	Tynker	Texto	9+	Web, Android, iOS	Javascript, Python, HTML, CSS	Free	<a href="https://www.tynker.com">https://www.tynker.com</a>
3	Code monkey	Texto	8+	Website	JavaScript, Python	Plan	<a href="https://www.codemonkey.com">https://www.codemonkey.com</a>
4	Colobot	Texto	10+	Windows	C++, C# or Java.	Free	<a href="https://colobot.info">https://colobot.info</a>
5	Robocode	Texto	8+	Windows, MAC, Linux	C#, Java	Free	<a href="https://robocode.sourceforge.io">https://robocode.sourceforge.io</a>
6	Codingame	Texto	8+	Website	25 lenguajes	Free	<a href="https://www.codingame.com/start">https://www.codingame.com/start</a>
7	Code avengers	Texto	5+	Website	JavaScript o HTML	Plan	<a href="https://www.codeavengers.com">https://www.codeavengers.com</a>
8	Check io	Texto	8+	Website	Typescript, Python	Free	<a href="https://checkio.org">https://checkio.org</a>
9	Coded arena	Texto	11+	Website	Python, Java	Free	<a href="https://codedarena.com">https://codedarena.com</a>

Elaboración propia.



## DECLARACIÓN Y AUTORIZACIÓN

Yo, **Segovia Castro Gabriela Nathaly**, con C.C: # **0927300582** autor/a del trabajo de titulación: **Análisis de las aplicaciones *game based-learning* que confluyen como desarrollo de las TIC en el área de la programación mediante la norma ISO 25000.** previo a la obtención del título de **Ingeniero en Sistemas Computacionales** en la Universidad Católica de Santiago de Guayaquil.

1.- Declaro tener pleno conocimiento de la obligación que tienen las instituciones de educación superior, de conformidad con el Artículo 144 de la Ley Orgánica de Educación Superior, de entregar a la SENESCYT en formato digital una copia del referido trabajo de titulación para que sea integrado al Sistema Nacional de Información de la Educación Superior del Ecuador para su difusión pública respetando los derechos de autor.

2.- Autorizo a la SENESCYT a tener una copia del referido trabajo de titulación, con el propósito de generar un repositorio que democratice la información, respetando las políticas de propiedad intelectual vigentes.

Guayaquil, **14 de septiembre de 2020**

f. Gabriela Segovia

**Segovia Castro Gabriela Nathaly**

**C.C: 0927300582**



## REPOSITORIO NACIONAL EN CIENCIA Y TECNOLOGÍA

### FICHA DE REGISTRO DE TESIS/TRABAJO DE TITULACIÓN

<b>TEMA Y SUBTEMA:</b>	Análisis de las aplicaciones <i>game based-learning</i> que confluyen como desarrollo de las TIC en el área de la programación mediante la norma ISO 25000.		
<b>AUTOR(ES)</b>	Segovia Castro Gabriela Nathaly		
<b>REVISOR(ES)/TUTOR(ES)</b>	Ing. García Sánchez Roberto		
<b>INSTITUCIÓN:</b>	Universidad Católica de Santiago de Guayaquil		
<b>FACULTAD:</b>	Ingeniería		
<b>CARRERA:</b>	Ingeniería en Sistemas computacionales		
<b>TÍTULO OBTENIDO:</b>	Ingeniero en Sistemas computacionales		
<b>FECHA DE PUBLICACIÓN:</b>	14 de septiembre del 2020	<b>No. DE PÁGINAS:</b>	105
<b>ÁREAS TEMÁTICAS:</b>	Educación Superior, Modelo de Calidad, Plataformas educativas		
<b>PALABRAS CLAVES/ KEYWORDS:</b>	Game Based-Learning, ISO 25000, Programación, TIC.		
<b>RESUMEN/ABSTRACT:</b>	<p>Las TIC han contribuido de forma significativa en la evolución de las herramientas que apoyan las tendencias de aprendizaje, por ello la presente investigación se centra en la evaluación de plataformas alineadas a la enseñanza de la programación en la educación superior que aplican la metodología <i>game based-learning</i> para la construcción de un modelo de calidad a nivel de producto inspirado en la norma ISO 25000 que permita evaluar la calidad interna y externa. Mediante una metodología de enfoque mixto, a través de instrumentos de recolección de datos aplicados a estudiantes y docentes de la carrera de Ingeniería, se analizan plataformas de programación basadas en textos, y contrastando la información analizada se definieron las características técnicas que debe de contar una plataforma que se aplique en procesos pedagógicos. De esta base parte el diseño del modelo de calidad a nivel de producto para evaluar plataformas <i>game based-learning</i> basado en diversas características internas como externas, las mismas que fueron evaluadas por medio de fórmulas claras y justas para definir por medio de un resultado final la efectividad de las mismas; lo que permitió definir la óptima para aplicarla en procesos de enseñanza de la educación superior y por ende fomentar habilidades que logren aumentar el nivel competitivo del estudiante.</p>		
<b>ADJUNTO PDF:</b>	<input checked="" type="checkbox"/>	SI	<input type="checkbox"/> NO
<b>CONTACTO CON AUTOR/ES:</b>	<b>Teléfono:</b> +593-967-082192	<b>E-mail:</b> gabriela.segovia@cu.ucsg.edu.ec	
<b>CONTACTO CON LA INSTITUCIÓN (COORDINADOR DEL PROCESO UTE)::</b>	<b>Nombre:</b> Ing. Edison Toala Quimi, Mgs.		
	<b>Teléfono:</b> +593-990976776		
	<b>E-mail:</b> edison.toala@cu.ucsg.edu.ec		
<b>SECCIÓN PARA USO DE BIBLIOTECA</b>			
<b>Nº. DE REGISTRO (en base a datos):</b>			
<b>Nº. DE CLASIFICACIÓN:</b>			
<b>DIRECCIÓN URL (tesis en la web):</b>			