

## PROPUESTA PARA TRABAJO DE GRADO

**TÍTULO**

**Sistema para apoyar el aprendizaje de la  
Programación utilizando *Gamification***

**MODALIDAD**

**Aplicación práctica**

**OBJETIVO GENERAL**

Crear un sistema que utilice técnicas de *Gamification* al proceso de aprendizaje, para ser usado en cursos introductorios de programación.

**ESTUDIANTE(S)**

**Daniel Felipe Arias Ramírez** \_\_\_\_\_

Documento	Celular	Teléfono fijo	Correo Javeriano
cc. 1015441117	310 8796812	4855897	<a href="mailto:daniel-arias@javeriana.edu.co">daniel-arias@javeriana.edu.co</a>

**Daniel Serrano Reyes** \_\_\_\_\_

Documento	Celular	Teléfono fijo	Correo Javeriano
cc. 1020772075	3166960472	2565123	<a href="mailto:daniel-serrano@javeriana.edu.co">daniel-serrano@javeriana.edu.co</a>

**DIRECTOR**

**Ing. Fabio Antonio Avellaneda Pachón** \_\_\_\_\_

## Contenido

<b>1 OPORTUNIDAD O PROBLEMÁTICA .....</b>	<b>3</b>
1.1 DESCRIPCIÓN DE LA OPORTUNIDAD O PROBLEMÁTICA .....	3
1.2 FORMULACIÓN .....	4
1.3 JUSTIFICACIÓN .....	5
1.4 OBJETIVO GENERAL .....	6
1.5 OBJETIVOS ESPECÍFICOS .....	6
1.6 FASE METODOLÓGICA .....	6
1.6.1 Modalidad .....	6
1.6.2 Fases.....	6
1.6.3 Metodología de desarrollo. ....	10
<b>2 GESTIÓN DEL PROYECTO .....</b>	<b>11</b>
2.1 CALENDARIZACIÓN .....	11
2.2 PRESUPUESTO .....	12
2.3 ANÁLISIS DE RIESGOS .....	12
2.4 DERECHOS PATRIMONIALES.....	13
<b>3 MARCO TEÓRICO / ESTADO DEL ARTE.....</b>	<b>14</b>
3.1 FUNDAMENTOS .....	14
3.2 TRABAJOS IMPORTANTES EN EL ÁREA: .....	20
<b>4 REFERENCIAS BIBLIOGRÁFICAS.....</b>	<b>23</b>

## 1 Oportunidad o Problemática

### 1.1 Descripción de la Oportunidad o Problemática

La programación de computadores, consiste en definir procesos, actividades y tareas, identificando la serie de pasos lógicos para realizar dicha función, al igual que modelar objetos del mundo real, y definir cómo interactúan entre sí [1]. Esta es una actividad de alta importancia hoy en día por varias razones. Una de estas es que el mundo cada vez está más gobernado por la tecnología, y lo continuará siendo cada vez más, por lo que es importante cómo funciona el mundo en el que estamos, y entender la forma en la que son creadas las herramientas usadas. Otra razón es quizás la más obvia, poder desarrollar y diseñar estas aplicaciones tecnológicas. Pero la razón más importante no es para enfocar a todos a ser ingenieros de sistemas, sino para que aprendan a pensar algorítmicamente, una habilidad que puede ser usada en cualquier campo del conocimiento [2].

La metodología que se ha venido desarrollando para la enseñanza de esta ciencia tiene algunas deficiencias, debido a que está basada en replicar los conocimientos teóricos que se acaban de aprender, lo cual no motiva a los estudiantes ni los impulsa a relacionar los conceptos y buscar una solución por sus propios medios [3].

Algunas de estas deficiencias son: Los estudiantes se frustran fácilmente, no hay suficiente motivación y la falsa percepción respecto a la dificultad de los problemas [4]. Debido a esto se genera una alta deserción de estudiantes en los cursos introductorios y temor a inscribir estas asignaturas por parte de otros estudiantes [4] [5]. Esto es un problema especialmente en esta área del conocimiento dada la escasez de profesionales a nivel mundial. [6]

Otra de las principales razones que causan el retiro de estas asignaturas es la falta de seguimiento individual del profesor a los estudiantes [5], causando una mala retroalimentación, haciendo difícil el crecimiento de los estudiantes.

Se observa que la deserción en carreras tecnológicas en el país es grande, aun presentando unos de los salarios más altos. Según la revista Forbes [6], los trabajos relacionados con las ciencias de la computación ocupan el segundo puesto en la lista de mejores salarios, para recién egresados. En contraste, según el Ministerio de Educación de Colombia, estas mismas carreras, presentan los índices de deserción más altos en el país [7]. Esto demuestra una clara desmotivación por parte de los estudiantes.

Una solución posible para este problema es el concepto de *Gamification*.

“*Gamification* es un término informal para el uso de elementos de videojuegos en sistemas que no sean juegos, para aumentar la experiencia de usuario, así como su compromiso” [8]. Esta es una técnica ampliamente usada desde hace varios años, con resultados exitosos. El objetivo de este documento es plantear cómo puede ser utilizado también para enseñar y atraer estudiantes al área de la programación.

## 1.2 Formulación

¿Cómo aplicar la teoría de *Gamification* para mejorar el proceso de aprendizaje de programación en universidades y colegios?

### 1.3 Justificación

Otro elemento que se implementa al tiempo con *Gamification* es el de evaluación automática [9]. Este consiste en dar retroalimentación automática al código de los estudiantes por medio de un software. Con esto se espera aportar ventajas significativas al estudiante como: mayor rapidez en recibir una respuesta, posibilidad de múltiples envíos, mejor retroalimentación, calificación más objetiva, detección de copia, entre otros. La corrección automática se basa en un software en el que los estudiantes envían su código o el resultado de haber ejecutado un set de entrada en el ejercicio, recibiendo la retroalimentación inmediatamente. Esta retroalimentación puede ser tan sencilla como decir si estuvo bien o mal o analizar por ejemplo el tiempo de ejecución, memoria utilizada, estructuras de datos usadas, etc., para cursos más complejos como estructuras de datos o programación de computadores.

Esta técnica es muy usada en ambientes competitivos como la ICPC de ACM o el CodeJam de Google por ejemplo.

Estos resultados se pueden observar en la Universidad de Helsinki [10], en donde subió el número de estudiantes con calificación mayor a 90% de 334 a 718, utilizando un sistema de calificación automática.

Varias páginas de educación virtual, han utilizado conceptos similares a *Gamification*, para dictar cursos de diferentes áreas. Esta iniciativa ha sido liderada por algunas de las mejores universidades del mundo, dando su material de forma gratuita a personas ajenas a la universidad. Algunos ejemplos destacados de estas páginas son Coursera [11] con más de 3 millones de usuarios y Khan Academy con más de 10 millones [10]. Estas páginas funcionan bien, cuando el estudiante quiere aprender por su propia cuenta, aunque consideramos que hace falta el componente de *Gamification* en cursos de programación un poco más avanzados que los introductorios. Lo que este tipo de páginas no abordan es el ser una herramienta de apoyo a una institución educativa, en donde hay un profesor que enseña los materiales ne-

cesarios, dejando la herramienta como un material de apoyo para el trabajo individual.

## 1.4 Objetivo general

Crear un sistema que utilice técnicas de *Gamification* al proceso de aprendizaje, para ser usado en cursos introductorios de programación.

## 1.5 Objetivos Específicos

- Seleccionar los factores que dificultan el aprendizaje de la programación que se resolverán y a partir de estos especificar los requerimientos funcionales y no funcionales.
- Seleccionar las técnicas de *Gamification* con las que será desarrollado el sistema basado en los requerimientos.
- Desarrollar un sistema que cumpla con los requerimientos propuestos.
- Validar el funcionamiento del sistema en estudiantes de cursos introductorios.

## 1.6 Fase Metodológica

### 1.6.1 Modalidad

La modalidad del trabajo es una aplicación práctica, basada en el trabajo de grado titulado “Modelo de la programación utilizando *Gamification*” [12], el cual se desarrolló con una metodología investigativa.

### 1.6.2 Fases

El sistema se desarrollará en tres fases, donde se abordará tanto el objetivo general como los objetivos específicos, estas son:

1. Búsqueda de información y establecimiento de requerimientos: esta fase consiste en la investigación de información acerca de *Gamification*, sus usos, técnicas y resultados. Búsqueda de información con respecto a cómo lograr una mejora en la motivación de los estudiantes y las metodologías que se han venido desarrollando para la enseñanza de la programación. Toda esta información con el fin de recopilarla y transformarla en los requerimientos en los cuales se desarrollará el sistema.
2. Diseño e implementación del sistema: luego de tener los requerimientos priorizados se procederá a realizar los diseños del sistema, los cuales permitirán una implementación del sistema ordenada y estructurada para dar una versión de mejor calidad al producto.
3. Validación: en esta fase se pondrá a prueba el funcionamiento del sistema en estudiantes de cursos introductorios de programación, en diferentes ambientes donde se puedan evaluar las funcionalidades del sistema y poder obtener un buen conjunto de resultados para poder analizarlos y realizar mejoras y posibles trabajos a futuro.

En la siguiente tabla se mostrarán los objetivos y actividades asociadas a estas fases y los resultados que se esperan obtener de cada una.

Fases	Objetivos Es- pecíficos	Actividades	Resultados
1. <b>Búsqueda de información y establecimiento de requerimientos.</b>	<p>Seleccionar los factores que dificultan el aprendizaje de la programación que se resolverán y a partir de estos especificar los requerimientos funcionales y no funcionales.</p> <p>Seleccionar las técnicas de Gamification con las que será desarrollado el sistema basado en los requerimientos.</p>	<p><b>1.1</b> Identificar los principales problemas de la enseñanza de la programación. Basado en los resultados de la investigación del trabajo de grado “titulado Modelo de la programación utilizando <i>Gamification</i>” [12].</p> <p><b>1.2</b> Identificar y priorizar los requerimientos funcionales y no funcionales.</p> <p><b>1.3</b> Identificar y escoger las técnicas de <i>Gamification</i> que serán utilizadas.</p>	<ul style="list-style-type: none"> <li>• Requerimientos funcionales y no funcionales.</li> <li>• Priorización de requerimientos.</li> <li>• Documento de investigación con los principales problemas de la enseñanza de la programación.</li> <li>• Documento con descripción de las técnicas de gamification en general, y cuales de estas serán aplicadas en el proyecto.</li> </ul>
2. <b>Diseño e implemen-</b>	Desarrollar un sistema que cum-	<b>2.1</b> Diseño de Arqui-	<ul style="list-style-type: none"> <li>• Diagramas:               <ul style="list-style-type: none"> <li>○ Clases</li> </ul> </li> </ul>



<b>tación del sistema</b>	pla con los requerimientos propuestos.	<p>tectura del sistema.</p> <p><b>2.2</b> Diseño de Base de Datos.</p> <p><b>2.3</b> Diseño de Diagrama de Clases.</p> <p><b>2.4</b> Selección de Tecnología que se va a implementar.</p> <p><b>2.5</b> Desarrollo del sistema.</p> <p><b>2.6</b> Pruebas unitarias.</p> <p><b>2.7</b> Pruebas de Integración.</p>	<ul style="list-style-type: none"> <li>○ Arquitectura</li> <li>○ Relacional</li> <li>• Conjunto de pruebas unitarias y de integración.</li> <li>• Sistema desplegado en un servidor.</li> </ul>
<b>3. Validación</b>	Validar el funcionamiento del sistema en estudiantes de cursos introductorios.	<p><b>3.1</b> Elegir el protocolo para realizar la validación del sistema.</p> <p><b>3.2</b> Escoger una muestra significativa con la que será validado el sistema.</p> <p><b>3.3</b> Analizar los resultados obtenidos de la validación llegando a una conclusión respecto a la efectividad del sistema.</p>	<ul style="list-style-type: none"> <li>• Documento del protocolo de la realización de las pruebas.</li> <li>• Documento de análisis de resultados de las pruebas.</li> <li>• Propuesta de posibles pruebas para realizar a futuro.</li> </ul>

### **1.6.3 Metodología de desarrollo.**

Para la realización del proyecto se escogerá la metodología iterativa incremental [13], dada que nos permite aprender de las primeras iteraciones teniendo en cuenta el restrictivo del tiempo que se tiene para entregar el sistema en su totalidad. Esta ofrece la libertad de añadir o refinar requerimientos y elementos de diseño durante el desarrollo, y un historial de hitos y versiones de software estables que nos permite darnos cuentas en etapas tempranas y posibles mejoras hacia futuro.

## 2 Gestión del Proyecto

### 2.1 Calendarización

Teniendo en cuenta la duración del ciclo lectivo 1503, son 16 semanas en las cuales se desarrollará el sistema y todos sus documentos. De acuerdo a la metodología propuesta en la sección anterior se realizó un cronograma para establecer la distribución de las fases a desarrollar.

	Duración(Semanas)															
Actividades	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
<b>Fase 1</b>																
1.1																
1.2																
1.3																
<b>Fase 2</b>																
2.1																
2.2																
2.3																
2.4																
2.5																
2.6																
2.7																
<b>Fase 3</b>																
3.1																
3.2																
3.3																

## 2.2 Presupuesto

Motivo	Descripción	Costo al mes	Costo Total
<b>Sueldo Daniel Arias</b>	\$20.000/hora, 8 horas semanales	\$640.000	\$10.240.000
<b>Sueldo Daniel Serrano</b>	\$20.000/hora, 8 horas semanales	\$640.000	\$10.240.000
<b>Papelería</b>	Impresiones	\$10.000	\$180.000
<b>Servicios</b>	Internet	\$40.000	\$640.000
<b>Dirección de trabajo de grado</b>			\$5.000.000
<b>Total</b>			\$26.300.000

## 2.3 Análisis de Riesgos

Riesgo	Forma de mitigarlo
<b>Mal diseño inicial del sistema</b>	Validar el diseño con gente que tenga experiencia en el área.
<b>Investigación insuficiente</b>	Comenzar el proceso de investigación en la elaboración de la propuesta, con

	referencias suficientes.
<b>Mala planificación del cronograma</b>	Verificaciones periódicas, de elementos completados a la fecha.
<b>Lenguaje de programación inadecuado</b>	Suficiente investigación antes de comenzar etapa de desarrollo.
<b>Herramienta existente que solucione el problema</b>	Investigación y validación con expertos en el tema. Demostrar una ventaja competitiva respecto a la herramienta.
<b>Tiempo y recursos insuficientes para la validación del sistema</b>	Comenzar en etapas tempranas el diseño de la fase de validación.

## 2.4 Derechos Patrimoniales

Los derechos patrimoniales de este trabajo pertenecerán a los estudiantes Daniel Arias y Daniel Serrano. El código del proyecto será publicado libremente bajo la licencia GNU GPL V2.0, que permitirá la distribución y modificación del programa y su código, atribuyendo derechos a los creadores, y distribuyéndolo bajo los mismos términos de licencia. [14]

## 3 Marco Teórico / Estado del Arte

### 3.1 Fundamentos

Actualmente no existe un consenso, de qué temas y de qué manera se deberían enseñar en los cursos introductorios de programación [15]. Se mostrará a continuación qué técnicas, temas y problemáticas existen en estos cursos alrededor del mundo. Para esto, se utilizarán como base artículos en lo que se realizaron encuestas a profesores de cursos de programación introductorios en universidades y colegios.

Existen varias metodologías para enseñar la programación, a continuación se presentan cuatro tipos de metodología mencionadas por Essi Lahtinen que resultan ser las más recurrentes en las instituciones [16].

- **Ambiente libre de sintaxis:** En esta metodología, se busca separar lo que es escribir código con programar, es decir se intenta que el estudiante entienda los conceptos de resolución de problemas, antes de pasar a programar soluciones. Este tipo de metodología se puede hacer incluso sin medios electrónicos [17]. Una vez el estudiante ha entendido los conceptos básicos se procede al uso de un pseudocódigo para resolver problemas sencillos. [18]
- **Literacy:** Es una metodología similar a la libre de sintaxis, en el sentido que primero se enseñan conceptos separados que serán útiles a la hora de programar. Se diferencia de la metodología anterior en que se utiliza un lenguaje de programación para que los estudiantes puedan aplicar los conceptos aprendidos; por lo tanto, es una metodología más práctica, pero también con un componente fuertemente teórico. [18]
- **Solución de problemas:** Se enfoca en enseñar a los estudiantes el análisis y diseño en el área de resolución de problemas. Se asume que esto es una habilidad extrapolable que se puede aplicar en otros campos una vez sea dominada. Se utiliza usualmente con un solo lenguaje de programación para poner en práctica los conoci-

mientos, pero puede ser aplicada con más de un lenguaje durante varios cursos, o sin el uso de ningún lenguaje. [18]

- **Computación como interacción:** Este paradigma es un poco diferente a los anteriores. La idea es aprovechar que la mayoría de estudiantes estén familiarizados con gran cantidad de interfaces y programas existentes. La forma tradicional se basa en enseñar a programar con un único hilo de ejecución, y con muy poca interacción con el usuario [18]. Este método busca enseñar desde el primer curso una programación más real en donde existan varios hilos de ejecución y una interfaz mucho más clara con el usuario [19].

Vale la pena aclarar que aunque estos son los modelos descritos, la mayor cantidad de profesores e instituciones, utilizan una combinación de estos [16].

Partiendo de los artículos en donde se realizaron encuestas a estudiantes y profesores, se encontraron los siguientes resultados que serán útiles para analizar la problemática en la enseñanza de la programación.

- En la gran mayoría de clases que no utilizan un ambiente libre de sintaxis, se utilizan lenguajes, usados en la industria, es decir aunque es importante conocer estos lenguajes, son lenguajes con una gran cantidad de funcionalidades y por lo tanto difíciles de aprender. Según la encuesta los lenguajes más usados son, C++ 73% y Java 17% [16].
- Los estudiantes tienden a preferir diferentes métodos para aprender, y consideran que aprenden más de alguna forma específica. Según la encuesta, los estudiantes percibían que estaban aprendiendo significativamente (en orden de mayor a menor aprendizaje) en los siguientes métodos: trabajo individual de tareas, estudio individual, en sesiones prácticas, sesiones en clase de trabajo en grupo, en clases magistrales. Por otro lado, los profesores encuestados encuentran las mejores prácticas en el siguiente orden, de mayor aprendizaje a menor: sesiones prácticas, trabajo in-

dividual de tareas, sesiones en clase de trabajo en grupo, estudio individual, clases magistrales. [16]

- No hay un currículo estándar y los temas que se enseñan varían mucho entre instituciones, pero los temas que más dificultad presentan los estudiantes son: recursión, eficiencia de algoritmos, genéricos y estructuras de datos. [15]

A parte de analizar el estado actual de la enseñanza de la programación, también es necesario evaluar cuales son los principales problemas en la forma en la que actualmente se enseña. Los principales elementos problemáticos encontrados, fueron:

- La enseñanza se basa más en problemas específicos de programación al igual que específicamente en un lenguaje, más que en la habilidad de resolver problemas mediante la programación [15]. Aunque muchos profesores saben esto, es difícil dirigir la clase hacia algo totalmente abstracto, dado que los estudiantes pierden motivación y no es fácil para ellos entender los conceptos sin algún tipo de aplicación práctica [18]. La mayor dificultad que se tiene es la complejidad de los lenguajes usados, dado que en su mayoría son lenguajes de alta complejidad, que no es sencillo entender la sintaxis [16].
- Los estudiantes son capaces de implementar los programas que los profesores piden para las tareas o ejercicios, pero no tienen el modelo mental de lo que está pasando en la parte de memoria del computador. Conceptos incorrectos de este tipo llevan a problemas en temas más complicados, como los mencionados anteriormente como los más problemáticos. [15]
- Los estudiantes, obtienen conocimientos muy superficiales, es decir, dado que la clase usualmente se basa en la sintaxis del lenguaje, más que en entender que está pasando más profundamente, los estudiantes entienden línea por línea del código, pero no pueden extrapolar los conocimientos aprendidos a situaciones ligeramente diferentes a los aprendidos en clase, ni pueden avanzar fácilmente a temas más avanzados [16].
- Hay una gran diferencia en los conocimientos previamente adquiridos, por lo que no es fácil encontrar un punto medio, en el que todos estén aprendiendo, y sea retador



al mismo tiempo. En una encuesta realizada a más de 500 estudiantes, 58% de los estudiantes, tenían conocimientos sobre programar. [16]

En los últimos años la enseñanza virtual se ha convertido en una de las soluciones para apoyar la educación en masa, resolviendo necesidades que la educación tradicional ha sido incapaz de resolver, como las personas distantes de las instituciones educativas o con escasos recursos para hacer pagos de matrículas completas, problemas de tiempo y horarios que puedan abarcar gran cantidad de estudiantes como profesores, o la baja demanda la cual no justifica la apertura de estos cursos [20].

La educación a distancia resuelve estos problemas, sin embargo también tiene fallos en cuanto a la poca interactividad entre un profesor y su alumno, lo cual dificulta la rectificación de errores al estudiante [21].

E-learning es una de las tantas técnicas de aprendizaje el cual está basada en la enseñanza de cursos o temas a distancia apoyándose en los avances de las telecomunicaciones [], esta técnica puede utilizarse como herramienta de apoyo a un curso donde los estudiantes pueden entrar libremente y resolver o repasar los temas vistos en clase o como el curso completo en donde la mayoría transmiten los conocimientos por medios multimedia como los videos [22].

Para poder desarrollar un buen entorno virtual para el estudiante, se debe tener en cuenta que todos los seres humanos aprendemos de maneras distintas, lo importante es poder estimular de la mejor forma posible estos métodos de aprendizaje en los estudiantes, ofreciendo una mejora en cuanto a la personalización del aprendizaje de estos, lo cual sería muy complejo para la enseñanza tradicional ofrecer un profesor o un curso distinto que se acomode a los métodos de aprendizaje por cada uno de los integrantes presenciales del curso [23]. Los pasos a tener en cuenta al momento de desarrollar un curso de enseñanza virtual es la descripción de los contenidos y las técnicas pedagógicas que se usarán para este adaptándose a las diferentes formas de aprendizaje de los estudiantes, la correlación de los temas y la planificación de las rutas que puede tomar el estudiante para realizar el curso [24].

Esta serie de pasos a tener en el desarrollo del curso virtual proveen beneficios en el aprendizaje del estudiante como la retroalimentación temprana y oportuna, la fácil actualización de los contenidos, el uso de contenido multimedia para mejorar la experiencia con el usuario y el desarrollo del curso por parte del estudiante a su propio ritmo [25].

Se debe tener en cuenta que no es la realización de una clase tradicional plasmada en un video, se deben aprovechar las ventajas que ofrecen las tecnologías de hoy en día para

*Gamification* es un término que se ha convertido en tendencia. Como resultado este ha llegado a una definición estándar en donde se entiende como un proceso que integra características de juegos en ambientes que no son juegos [26]. Estas características se usan de acuerdo al contexto en donde se quieran aplicar, ya que no se ha llegado a una metodología o un modelo específico para realizar con éxito este proceso [27].

Estas características se usan para atraer clientes y mejorar la motivación de estos al hacer uso de sus servicios. Se debe tener en cuenta que el uso de *Gamification* no debe pasar ciertos lineamientos en los que pueda convertirse en un juego [26]. El uso de *Gamification* ha demostrado buenos resultados, aunque se evidencian casos en los que no se tuvo éxito o no se lograron los resultados esperados debido a un mal diseño y al descuido hacia el público al que este iba a ser presentado; en respuesta esto llevó a una mala comprensión por parte de los participantes y a una colaboración nula en términos de motivación [28].

*Gamification* es un proceso por el cual el usuario resolviendo o desarrollando actividades del mundo real, puede sentirse en la presencia de un ambiente de juego [29]. El uso de técnicas de *Gamification* se ha desarrollado con éxito en diversos temas como ambientes de oficina, e-learning, sector educativo, sector bancario, deportes e idiomas por mencionar algunos [30]. El tema en el cual se va a desarrollar el proceso de *Gamification* es en el sector educa-

tivo, más específicamente en el apoyo del aprendizaje de la programación en cursos introductorios.

Se ha implementado con éxito *Gamification* mediante la integración de insignias, puntos, y tablas de clasificación, más no son las únicas técnicas con las que se puede hacer uso de *Gamification* [31]. Estas técnicas se efectúan con el fin de mejorar la motivación en el usuario, la idea no es colocar un sistema de puntos y permitir que el usuario gane puntos sin sentido.

Con el uso de *Gamification* se quiere apoyar la motivación del aprendizaje en el estudiante ya que se ha encontrado que la falta de motivación en los estudiantes conlleva a bajas notas [16], el retiro de la materia o deserción en las carreras. Algunas causas de estos problemas se relacionan con las deficiencias en la metodología que se ha venido desarrollando para la enseñanza de esta ciencia, deficiencias como la fácil frustración de los estudiantes, la falta de motivación y la falsa percepción respecto a la dificultad de los problemas [4].

En el proceso de *Gamification* se pretende aumentar la motivación intrínseca y extrínseca del usuario. La primera puede ser entendida como una motivación interior, por lograr una satisfacción personal y aumentar su autoestima. Por otro lado, la motivación extrínseca es el estímulo que se le brinda a una persona para realizar determinada acción [32]. En el proceso de aprendizaje se ha observado que sin motivación los estudiantes presentan fallas en la calidad del aprendizaje [33]. Por lo que obligar a los estudiantes a realizar ciertas actividades o tareas por la obtención de una nota disminuye la motivación intrínseca del estudiante, la cual es la que se pretende aumentar con el uso de *Gamification*.

Si se suministra un ambiente controlado para el aprendizaje, pero se deja que el estudiante tenga *libertad* en el desarrollo de las actividades del sistema se puede mejorar la motivación interna, es decir la motivación intrínseca. Lo anterior permite que el estudiante compita consigo mismo por la autosatisfacción de lograr dar solución a un problema, conseguir más puntos o poder avanzar más en el mapa para desafiar al sistema y los nuevos retos [34]. Estos resultados se pueden lograr debido a los beneficios que se obtienen del uso de *Gami-*

*fication* que son: la retroalimentación adecuada y oportuna, la tolerancia al fracaso y una mejor estructura de los objetivos de la clase y de los problemas a resolver [35].

Para mejorar la motivación extrínseca se usan técnicas de *Gamification* como la competencia visualizada en las tablas de clasificación por la obtención de puntos e insignias, y el uso de una comunidad donde el usuario puede mostrar los resultados obtenidos, dado que el ser humano es un ser social y lo que suceda en la sociedad afecta a este [36], la motivación extrínseca aumenta en el estudiante en el momento de publicar, demostrar sus logros obtenidos y formar competencia con los otros estudiantes de su curso, incluso la competencia entre cursos u otras entidades. Aunque hay que manejar bien esta técnica, debido a que la comparación entre usuarios de niveles muy disparejos puede resultar en una colaboración negativa en la motivación del estudiante, dado que los estudiantes que se encuentren en niveles inferiores al ser comparados con otros estudiantes de niveles muy superiores se vean frustrados al intentar alcanzarlos.

Se deberán estructurar muy bien las técnicas a ser utilizadas, para que el sistema pueda dar una motivación duradera y suficiente, debido a que los cursos introductorios de programación tienen un periodo de duración de 16 semanas, se espera que el estudiante haga uso del sistema durante la completitud del periodo académico o como mínimo el 60% de su totalidad, donde se puedan observar resultados de mejoras en el aprendizaje de la programación.

### **3.2 Trabajos importantes en el área:**

Actualmente existen varios trabajos que intentan hacer uso de las ventajas de *Gamification*, en distintas áreas. Uno de los campos en el que más se ha intentado motivar a los usuarios, es en los ejercicios personales. Aplicaciones como Nike+, que es usada por más de 11 millones de usuarios, usa un sistema de puntos, retos y tablas de posiciones para motivar a sus usuarios [35]. Otro ejemplo en el área del ejercicio es Fitocracy, que usa el sistema de medallas con un componente social para hacer que sus usuarios estén en forma [37]. También se

utiliza en tareas cotidianas, en donde lo que se busca es que el usuario mejore su productividad en alguna forma, un ejemplo es CARROT, una aplicación de tareas por hacer, donde se cuenta una historia a partir de un personaje para que la persona cumpla a tiempo sus tareas [35].

En el ámbito de mejorar la productividad, también están los sistemas que capacitan al usuario en algún software o servicio. Compañías como Adobe y AutoCAD, han invertido en programas de este tipo, creando cursos que motivan al usuario a continuar aprendiendo y a desarrollar los talleres propuestos en programas como AutoCAD y Photoshop. [37]

En el ámbito en el que *Gamification* más ha probado su utilidad es en el de la educación, funcionando tanto en pequeños softwares usados en colegios y universidades como sistemas a gran escala utilizados por millones de usuarios. Ejemplos de pequeños programas usados en colegios, se pueden encontrar para la enseñanza de materias como biología, matemáticas, etc. Encontrando una mejora significativa en las calificaciones de los estudiantes, como ejemplo se puede tomar el software McClean y Moreno que enseña biología y programación por medio de un videojuego, obteniendo un incremento en las notas de 30% y 19% respectivamente [37].

Probablemente los mayores exponentes de la educación por medio de gamification son, *Code School*, un sistema que enseña a programar virtualmente, haciendo uso de elementos como puntos y medallas para medir el progreso [37] y *Khan Academy*, “un sistema que gamifico la educación en áreas incluyendo matemáticas, ciencias, historia y economía” [35], usando elementos como medallas y misiones para mantener motivado al usuario [35]. Una de las grandes ventajas de *Khan Academy* es que, dado que funciona con videos, es totalmente personalizado y los estudiantes pueden seguir su propio ritmo, algo que ha cambiado la forma de enseñar tanto dentro como fuera del salón [38], y ha atraído más de 10 millones de visitantes mensualmente [10]. Aunque estas páginas son similares al proyecto descrito, tienen unas diferencias importantes. KhanAcademy no tiene un componente fuerte en programación, solo tiene cursos básicos. CodeSchool, es pago. La mayor diferencia es que estos sistemas funcionan con auto aprendizaje, es decir el estudiante debe aprender sin ninguna

ayuda externa al sistema. El sistema descrito en este documento, pretende ser una ayuda a las clases de programación, dictadas por instituciones.

## 4 Referencias bibliográficas

- [1] A. F. Blackwell, «What is Programming?,» University of Cambridge Computer Laboratory , 2014.
- [2] B. Isong, «A Methodology for Teaching Computer,» *Modern Education and Computer Science*, pp. 15-21, 2014.
- [3] D. Crow, «Why every child should learn to code,» *The Guardian*, 2014.
- [4] K. F. C. S. Rebecca Vivian, «Can everybody learn to code?: computer science community perceptions about learning the fundamentals of programming,» *Koli Calling '14 Proceedings of the 14th Koli Calling International Conference on Computing Education Research*, pp. 41-50, 2014.
- [5] L. M. Päivi Kinnunen, «Why students drop out CS1 course?,» *ICER '06 Proceedings of the second international workshop on Computing education research*, pp. 97-108, 2006.
- [6] S. Adams, «The College Degrees With The Highest Starting Salaries in 2015,» *Forbes*, 2014.
- [7] G. B. Mantilla, «Jaque a la desercion,» *Educacion superior*, n° 14, 2010.
- [8] M. S. N. K. O. D. D. Sebastian Deterding, «Gamification. using game-design elements in non-gaming contexts,» *CHI EA '11 CHI '11 Extended Abstracts on Human Factors in Computing System*, pp. 2425-2428, 2011.
- [9] A. K. R. S. Lauri Malmi, «Experiences in automatic assessment on mass courses and issues for designing virtual courses,» *ITiCSE '02 Proceedings of the 7th annual conference on Innovation and technology in computer science education*, pp. 55-59, 2002.
- [10] G. H. L.-D. S. W. Bhatt, «Case Study, Khan Academy,» 2014.

- [11] «Coursera,» [En línea]. Available: <http://coursera.org>. [Último acceso: 25 Mayo 2015].
- [12] R. J. A. París, «Programación, Modelo para la Motivación del Aprendizaje de la,» Pontificia Universidad Javeriana, Bogota, 2014.
- [13] C. J. S. A. A. C. Roberth G. Figueroa, «METODOLOGÍAS TRADICIONALES VS. METODOLOGÍAS ÁGILES».
- [14] GNU, «GNU General Public License v2.0,» GNU, Junio 1991. [En línea]. Available: <http://choosealicense.com/licenses/gpl-2.0/>. [Último acceso: 25 Mayo 2015].
- [15] J. B. Carsten Schulte, «What do teachers teach in introductory,» 2006.
- [16] K. A.-M. H.-M. J. Essi Lahtinen, «A study of the difficulties of novice programmers,» *ITiCSE '05 Proceedings of the 10th annual SIGCSE conference on Innovation and technology in computer science education*, pp. 14-18, 2005.
- [17] D. T. Colin Fidge, «Losing their marbles: syntax-free programming for assessing problem-solving skills,» *ACE '09 Proceedings of the Eleventh Australasian Conference on Computing Education - Volume 95*, pp. 75-82, 2009.
- [18] S. Fincher, «What Are We Doing When We Teach Computing in Schools?,» *Communications of the ACM, Vol. 58 No. 5*, pp. 24-26, 2015.
- [19] L. A. Stein, «Interactive programming: revolutionizing introductory computer science,» *ACM Computing Surveys (CSUR) - Special issue: position statements on strategic directions in computing research Surveys Homepage archive*, 1996.
- [20] A. G. R. y. E. M. Caro, «ESTILOS DE APRENDIZAJE Y E-LEARNING. HACIA UN MAYOR,» Cartagena, 2010.
- [21] M. A. Carlos J. Costa, «Analysis of e-learning processes,» Lisboa , 2011.
- [22] S. A. K. B. Sethu Subramanian N, «Enhancing E-learning Education with Live Interactive,» 2014.
- [23] A. G. a. L. Morales, «E-Learning and Intelligent Planning: Improving Content



Personalization,» 2014.

- [24] N. Arman, «E-learning Materials Development: Implementing Software Reuse Principles and Granularity Levels in the Small Using Taxonomy Search,» Bethlehem, 2010.
- [25] E. E. T. M. M. H. Z. S. Y. Y. C. C. M.-L. T. E. E. T. M. M. H. Z. S. Y. Y. C. C. M. M.-Y. K. a. 2. C. T. Yin-Leng Theng, «An Empirical Study of Students' Perceptions on,» Singapore, 2010.
- [26] Y. D. P. A. W. Alfa Ryano Yohannis, «Defining Gamification From lexical meaning and process viewpoint towards a gameful reality,» Jakarta, 2014.
- [27] P. H. A. S. Benjamin Heilbrunn, «Tools for Gamification Analytics: A Survey,» 2014.
- [28] J. S.-d.-N. L. d.-M. L. F.-S. C. P. Adrián Domínguez, «Gamifying learning experiences: Practical implications and outcomes,» Madrid, 2013.
- [29] B. K. A. D. David Rojas, «The Missing Piece in the Gamification Puzzle,» Ontario, 2013.
- [30] P. G. Anant Vaibhav, «Gamification of MOOCs for Increasing User Engagement,» 2014.
- [31] F. B. K. O. A. N. T. Elisa D. Mekler<sup>1</sup>, «Do Points, Levels and Leaderboards Harm Intrinsic Motivation? An Empirical Analysis of Common Gamification Elements,» Ontario,, 2013.
- [32] M. Iriarte, «¿MOTIVACIÓN “INTRÍNSECA” Y “EXTRÍNSECA”?,» 2007.
- [33] F. J. G. B. y. F. D. Betoret, «Motivación, aprendizaje y rendimiento escolar,» 2002.
- [34] J. A. Huertas, «Motivacion querer aprender,» Argentina, 1997.
- [35] C. W. T. N. G. Chad Richards, «Beyond designing for motivation: the importance of context in gamification,» *CHI PLAY '14 Proceedings of the first ACM SIGCHI annual symposium on Computer-human interaction in play*, pp. 217-226, 2014.

- [36] M. D. P. C. Tatjana Welzer, «Awareness of Culture in e-learning,» Maribor, 2010.
- [37] S. G. M. J. F. D. G. Gabriel Barata, «Improving Student Creativity with Gamification and Virtual Worlds,» Ontario,, 2013.
- [38] C. THOMPSON, «How Khan Academy Is Changing the Rules of Education,» *Wired*, 2011.
- [39] A. E. N. Alok Kumar Vishwakarma, «E-learning as a Service: A New Era for Academic Cloud Approach,» 2012.
- [40] S. M. D. H. B. P. B. Richa Sharma, «An Opinion-based framework for designing socially aware e-Learning systems,» 2012.
- [41] D. I. F. Katie Seaborn, «Gamification in theory and action: A survey,» Ontario, 2014.
- [42] J. F. Michael D. Hanus, «Assessing the effects of gamification in the classroom: A longitudinal study on intrinsic motivation, social comparison, satisfaction, effort, and academic performance,» Columbus, 2014.
- [43] G. R. David Codish, «Adaptive approach for gamification optimization,» 2014.
- [44] «Gamification for Engaging Computer Science Students in Learning Activities: A Case Study,» 2014.
- [45] K. S. H. W. Satoshi Arai, «A Gamified Tool for Motivating Developers to Remove Warnings of Bug Pattern Tools,» 2014.
- [46] M. A. A. S. Philipp Herzig, «A Generic Platform for Enterprise Gamification,» 2012.
- [47] R. Rughiniş, «Gamification for Productive Interaction Reading and Working with the Gamification Debate in Education,» Bucharest, Romania.
- [48] N. S. T. S. M. A. I. Firas Layth Khaleel, «The Study of Gamification Application Architecture for Programming Language Course,» Indonesia., 2015.
- [49] J. D. a. S. C. LINDA WERNER, «Children Programming Games: A Strategy for

Measuring Computational Learning,» 2014.

[50] B. B. M. B. DiSalvo, «Khan Academy Gamifies Computer Science,» Georgia, 2014.