

## Aplicación de Técnicas de Gamificación para la Enseñanza de Programación a Alumnos de Primer Año de Ingeniería

Pablo M. Vera, Edgardo J. Moreno, Rocío A. Rodríguez, M. Carina Vázquez, Federico E. Valles

GIDFIS (Grupo de Investigación, Desarrollo y Formación en Innovación de Software)

Departamento de Ingeniería e Investigaciones Tecnológicas

Universidad Nacional de La Matanza

{pvera, emoreno, rocio.rodriguez, mcvazquez, fvalles} @unlam.edu.ar

### Resumen

Este artículo presenta el diseño de una plataforma que permite aplicar técnicas de gamificación a la enseñanza de programación a alumnos universitarios de los primeros años de ingeniería. El docente prepara enunciados junto con un lote de casos de prueba que se utilizarán para corroborar si el código desarrollado por el alumno es correcto. Mediante un portal web los alumnos envían su código fuente y este es compilado y verificado automáticamente según los casos de prueba planteados. A cada ejercicio se le asigna una cierta cantidad de puntos por su correcta resolución. Dichos puntos se van decrementando según los errores detectados y tiempo empleado. La asignación de puntos permite aplicar distintas técnicas de gamificación, como ser rankings, premios, desafíos, insignias, etc. Esta plataforma servirá para incentivar a los alumnos a realizar prácticas fuera del ámbito universitario, buscando una superación personal y un reconocimiento social al poder comparar su puntaje con el resto de sus compañeros.

**Palabras clave:** Gamificación, Programación, Lenguaje C

### Motivación del Trabajo

Uno de los principales problemas con el que nos enfrentamos hoy en día con los alumnos que recién comienzan sus primeros pasos en

la universidad es la falta de dedicación a las materias fuera del horario de cursada. Una considerable parte del alumnado no posee la disciplina necesaria para realizar ejercitación, investigar por su cuenta y comprometerse realmente con la materia. Una encuesta realizada en la Universidad Nacional de La Matanza en los cursos de programación de primer año de ingeniería durante el segundo cuatrimestre del 2015 (cabe destacar que es una materia cuatrimestral con una carga horaria de 8 hs reloj semanales); indica que el 45% de los alumnos no dedica horas de estudio para la materia sino que comienza a estudiar cuando se aproxima la fecha de examen (una semana antes ó incluso con menor antelación). Un 13% del alumnado indica que le bastan las horas de clase para prepararse para el examen no usando horas adicionales. La figura 1 muestra con más profundidad el análisis resultante de la encuesta por medio de la cantidad de horas que dedican al estudio de la materia.

Programación es una materia que requiere mucha práctica para poder afianzar los conocimientos. La capacidad de elaborar estrategias para la resolución de los problemas (algoritmos) es una capacidad que se va forjando lentamente a medida que se incorporan nuevas herramientas que brindan los lenguajes de programación. La experiencia obtenida por el grupo de trabajo en el dictado de la materia, pone en evidencia la falta de dedicación de tiempo por parte de los alumnos al no realizar ejercitación por su cuenta.

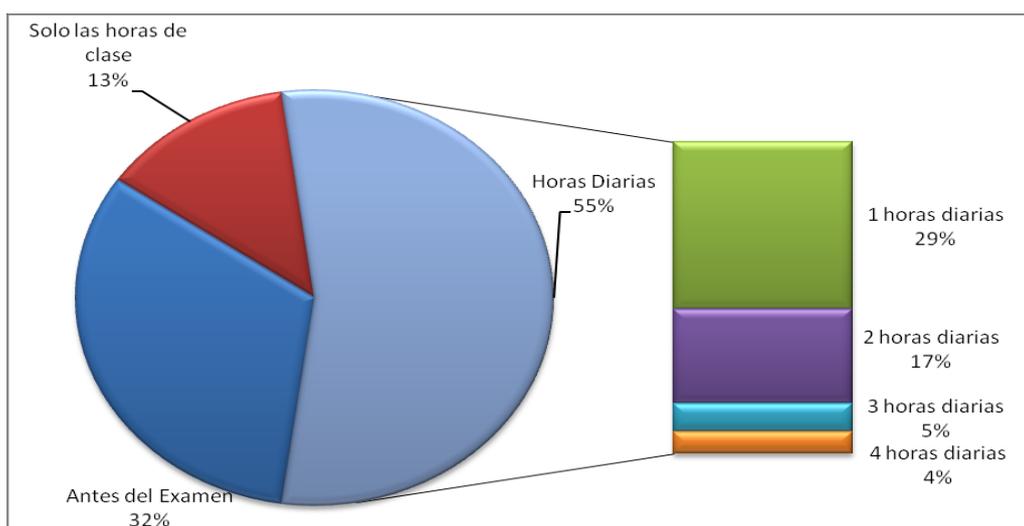


Figura 1. Horas de Dedicadas al Estudio

Se busca entonces una forma de motivar a los alumnos sin ningún mecanismo de coerción, ya que sería fácil por ejemplo hacer obligatoria la entrega de trabajos para aprobar o promocionar la materia, pero se busca lograr la participación por un interés personal. “Para mantener el interés del educando, se debe aprovechar la energía natural del alumno para sentirse capaz y orientado hacia el logro de metas. Es necesario que realice de manera voluntaria, lo que se espera que haga; y que desarrolle sus aptitudes para que alcance la meta deseada. Así, cuanto más capaz se sienta un alumno de desarrollar una actividad, más motivado estará para persistir en ella, a su vez, le dará la sensación de éxito o de mejoría y le ayudará a mantenerse motivado”. [1]

Este objetivo puede alcanzarse utilizando distintas técnicas de Gamificación.

“El objetivo general de la gamificación es influir y motivar a los usuarios para conseguir adquirir hábitos y alcanzar objetivos” [2].

## Gamificación

El concepto de Gamificación puede definirse como:

- “el proceso de usar el pensamiento y las mecánicas de juegos en contextos distintos a

estos con el fin de resolver problemas y comprometer a los usuarios” [3].

- “intentar aprovechar el poder motivacional de los juegos y aplicarlo al mundo real” [4].

De esta manera no solo se estaría afirmando la autonomía e independencia del estudiante en cuanto a las decisiones que desea tomar, sino también respecto de los criterios seleccionados y el acercamiento a situaciones de la realidad que se puede plantear desde el desarrollo de la técnica. La gamificación plantea una serie de elementos [5] que se reconocen como esenciales:

1. Dinámicas de juegos: Consiste en aspectos generales de orientación del sistema que le otorgan sentido a la actividad. Algunas dinámicas más habituales que puede mencionarse [6]:
  - Restricciones
  - Emociones
  - Narrativa
  - Progresión
  - Relaciones
2. Mecánica de juegos: Refiere a las reglas que se consideran en la técnica para avanzar, como:
  - Retos
  - Competencia

- Cooperación
- Retroalimentación
- Recompensas
- Estados

3. Componentes de juegos: Consiste en aquellos elementos necesarios para la implementación de las mecánicas y dinámicas, entre los que se pueden mencionar:

- Insignias
- Ranking
- Niveles
- Puntos
- Avatar

A través de la gamificación se puede trabajar los procesos cognitivos de manera más creativa, atractiva y favoreciendo el aprendizaje.

“...la gamificación no es convertir las asignaturas en un juego, ni simplemente poner una puntuación a cada tarea. El proceso de gamificar, al igual que el proceso de virtualizar una asignatura, supone un esfuerzo de planificación y seguimiento, diseño y realimentación constantes y adaptados a cada asignatura, grupo y curso” [7].

## **Gamificación aplicada a la enseñanza de programación**

El desafío surge en cómo aplicar la gamificación en un curso universitario. La base de la gamificación para poder aplicar distintas mecánicas y dinámicas es la asignación de puntos. Hay herramientas en las que un docente mediante un teléfono celular puede en el aula ir asignando puntos a los alumnos a medida que realizan determinadas acciones, como ser participar en clase, ayudar a un compañero, proponer un punto de vista diferente, etc. Estas herramientas como por ejemplo EdMODO [8] ó Schoology [9], no están enfocadas al ámbito universitario ya que la cantidad de alumnos, principalmente en los

primeros años es demasiado numeroso y requieren de un conocimiento más personalizado de los alumnos por parte del docente.

Por ello se busca una forma de asignar puntos mediante la resolución de distintos ejercicios por parte de los alumnos, generando una plataforma que permita la corrección automática de dichos ejercicios y asigne puntos al evaluar los resultados, la velocidad de realización y la constancia de los alumnos en el uso de la herramienta.

## **Diseño de la Plataforma de Gamificación**

A fin de lograr los objetivos planteados se diseña una plataforma que permite evaluar en forma automática el código fuente escrito por un alumno. Para lograr la corrección automática del código fuente es necesario saber exactamente el resultado esperado, prever errores posibles y pensar lotes de prueba que abarque las distintas partes del programa. Para lograr este objetivo se plantea el desarrollo del código de una función, donde el alumno sabe exactamente qué datos recibe (parámetros de entrada o parámetros formales) y se le especifica que valor debe retornar en cada caso. Este enfoque no tiene en cuenta el procedimiento para llegar al resultado sino que los resultados sean correctos.

La preparación de cada uno de los ejercicios a resolver conlleva una serie de pasos y un trabajo exhaustivo por parte del docente. Pero una vez establecidos, la corrección ya será automática.

Los pasos a seguir son los siguientes:

1. El docente redacta el enunciado de la función a desarrollar, especificando su prototipo, el nombre de los parámetros formales y los valores esperados de retorno si corresponde (hay casos en que la función no retorna un valor pero puede modificar un dato pasado por referencia, por ejemplo un vector).

2. Para poder automatizar la prueba, el docente prepara un programa que incluye el prototipo de la función y la utiliza, pero no tiene el desarrollo de la misma. Es decir el docente codifica en lenguaje C el programa invocando a la función que deben codificar los alumnos.
3. El programa deberá enviar parámetros a la función y comprobar que el resultado sea correcto, armando así uno o más casos de prueba. Si algún caso de prueba falla, entonces el programa creado por el docente retornará distintos códigos de error para poder identificar los mismos mostrando un mensaje adecuado al alumno en cada caso.
4. El docente carga mediante la plataforma el desafío, subiendo el archivo creado que incluye los casos de prueba.
5. Por cada código de error que puede retornar el programa, se realiza una descripción sencilla para que sea informado al alumno en el caso que ocurra.
6. El alumno ingresa a la plataforma y comienza a realizar el desafío planteado registrando el tiempo exacto de inicio.
7. Una vez que la codificación esté terminada el alumno envía el código para ser evaluado.
8. Al archivo de código fuente generado por el docente, se le concatena debajo el código escrito por el alumno.
9. Usando un compilador de C estándar se compila el archivo, y en caso de que no tenga errores se linkea y se genera el ejecutable.
10. Una vez obtenido el ejecutable se ejecuta el programa y se analizan el código de error retornado. Si la ejecución es exitosa retornará el valor 0, pero en el caso de que no pase alguno de los casos de prueba planteados por el docente retornará el código de error establecido en el punto

5. Este código permitirá mostrarle al alumno en mensaje aclaratorio del error cometido pudiendo corregir el código y volver a enviarlo tantas veces como sea necesario hasta lograr cumplir con todos los casos de prueba planteados.

### **Asignación de Puntos**

Cada ejercicio terminado correctamente asignará puntos al alumno, existiendo un puntaje máximo para el caso de que se entregue en tiempo y sin errores. Luego se le descontarán puntos por:

- Errores de compilación.
- Errores lógicos detectados por los casos de prueba.
- Errores por Time-Out (el programa nunca termina por ejemplo porque queda en un loop infinito).
- Errores inesperados (por ejemplo una división por cero).
- Tiempo de resolución superior al propuesto (para cada ejercicio se establece un tiempo de referencia en minutos).

El sistema fue diseñado para ser flexible en cuanto a la asignación de puntajes y tiempos, es decir que para cada ejercicio se configuran los puntos que se podrán obtener y la cantidad de puntos que se restan por errores cometidos. Esto será ajustado por el docente tomando en cuenta el nivel de dificultad del ejercicio planteado.

### **Mecanismos de Gamificación a Emplear**

Los ejercicios los irá habilitando el docente de cada curso según el grado de avance del dictado de la materia y del tema que se haya finalizado. Los ejercicios se agrupan por niveles pudiendo establecerlos por ejemplo por cada tema de la materia.

Se utilizan distintas técnicas de gamificación que se basan en la resolución de los ejercicios planteados. Generándose rankings en base a los puntos obtenidos:

- Los 10 puntajes más altos
- Los 10 puntajes más altos por nivel.

Además se crearán diferentes insignias:

- Una insignia por cada nivel que el alumno supere (al completar todos los ejercicios de un tema dado)
- Una insignia por velocidad de resolución (al resolver un ejercicio correctamente en un tiempo inferior al propuesto)
- Una insignia por constancia, al entrar en forma periódica a la plataforma e iniciar 5 desafíos a lo sumo al día siguiente en que el docente lo habilita
- Una insignia final al terminar con todos los temas de la materia

También se incorporaran herramientas sociales como:

- Desafíos grupales (el alumno puede vincularse a un grupo y se mostraran puntos acumulados por cada grupo)
- Posibilidad de vinculación con redes sociales, publicando sus logros e insignias en facebook o twitter
- Torneos, donde mediante inscripción previa se arme una llave de competencia por eliminación directa entre dos alumnos que tendrán asignado el mismo ejercicio, el que obtenga más puntos pasará a la fase siguiente y así sucesivamente hasta llegar a un ganador

Al completar cada ejercicio (dentro del tiempo estimado) se le dará al alumno un premio al azar que consistirá en tarjetas con las siguientes ventajas:

- Un minuto adicional para la resolución de un ejercicio
- Cinco minutos adicionales para la resolución de un ejercicio
- Puntos extra (esta tarjeta la podrá utilizar antes de enviar su código por primera vez y permitirá incrementar en un 10% los puntos obtenidos en el ejercicio que está realizando)

### **Desarrollo de la Plataforma**

La plataforma fue desarrollada para ser accedida mediante un entorno web adaptativo, para que pueda ser visualizada correctamente desde distintos dispositivos. El backend de la aplicación fue desarrollado en C# con ASP.Net MVC 5 [10] y el frontend utilizando HTML 5 y el Framework materializecss [11].

De esta forma el alumno podrá ingresar y realizar la ejercitación desde cualquier navegador compatible con HTML5. En la figura 2 se muestra una captura de la pantalla donde el alumno visualiza el enunciado a desarrollar, escribe el código fuente y lo envía para su corrección.

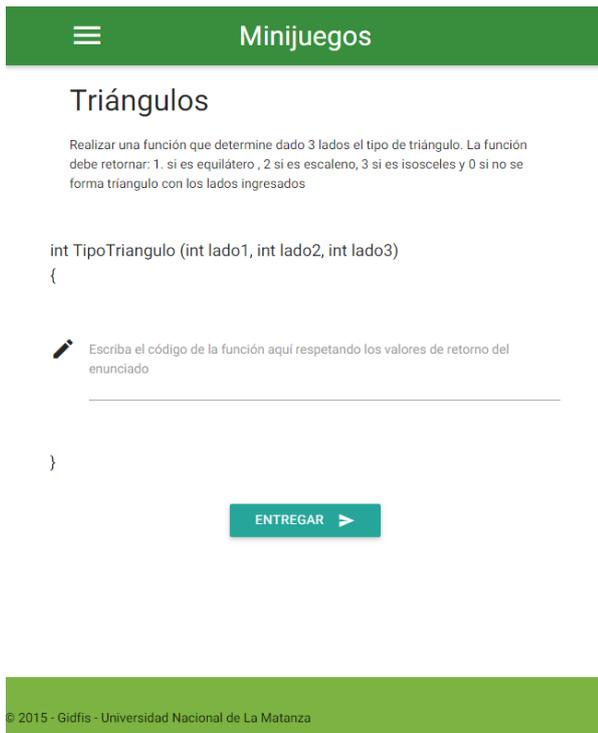


Figura 2. Pantalla con el desafío de código

Adicionalmente para la asignación de premios como el alumno deberá elegir entre una serie de cartas y dar vuelta una para obtener su premio se utiliza el plugin jquery flip [12] para agregar una animación aumentando el atractivo visual.

### Definición de Casos de Prueba

A fin de clarificar la metodología para la elaboración de ejercicios se muestra un ejemplo completo con sus casos de prueba.

#### Enunciado:

Realizar una función que determine dado 3 lados el tipo de triángulo. La función debe retornar: 1. si es equilátero, 2 si es escaleno, 3 si es isósceles, 0 si no se forma triángulo con los lados ingresados.

#### Prototipo de la función:

```
int TipoTriangulo (int lado1, int lado2, int lado3)
```

#### Errores posibles:

La tabla 1 muestra los errores que son tenidos en cuenta por los casos de prueba planteados y su código de error correspondiente.

Tabla 1. Errores posibles para el ejercicio de triángulos

Código	Descripción
100	Error al intentar determinar si un triángulo es equilátero
200	Error al intentar determinar si un triángulo es escaleno
300	Error al intentar determinar si un triángulo es isósceles
400	No se está verificando correctamente si los lados ingresados forman o no un triángulo
500	Se debe validar que los lados ingresados sean mayor que 0

#### Código fuente de los casos de prueba:

```
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
#define ERROR_EQUILATERO 100
#define ERROR_ESCALENO 200
#define ERROR_ISOSCELES 300
#define NO_TRIANGULO 400
#define LADOS_INVALIDOS 500

int TipoTriangulo (int lado1, int lado2, int lado3);
int main ()
{
    if (TipoTriangulo(-40,-40,-40)!=0) //lados negativos no forman triángulo
        exit(LADOS_INVALIDOS);

    if (TipoTriangulo(-40,-30,-40)!=0) //lados negativos no forman triángulo
        exit(LADOS_INVALIDOS);

    if (TipoTriangulo(-40,-30,-20)!=0) //lados negativos no forman triángulo
        exit(LADOS_INVALIDOS);

    if (TipoTriangulo(0,0,0)!=0) //lados cero no forman triángulo
        exit(LADOS_INVALIDOS);

    if (TipoTriangulo(0,40,40)!=0) //lados cero no forman triángulo
        exit(LADOS_INVALIDOS);

    if (TipoTriangulo(40,0,40)!=0) //lados cero no forman triángulo
        exit(LADOS_INVALIDOS);
}
```

```

    if (TipoTriangulo(40,40,0)!=0)
//lados cero no forman triángulo
    exit(LADOS_INVALIDOS);

    if (TipoTriangulo(1,5,40)!=0)
//no forma triángulo
    exit(NO_TRIANGULO);

    if (TipoTriangulo(20,20,20)!=1)
//equilatero
    exit(ERROR_EQUILATERO);

    if (TipoTriangulo(20,30,40)!=2)
//escaleno
    exit(ERROR_ESCALENO);

    if (TipoTriangulo(20,20,40)!=3)
//isosceles
    exit(ERROR_ISOSCELES);

    return 0;
}

```

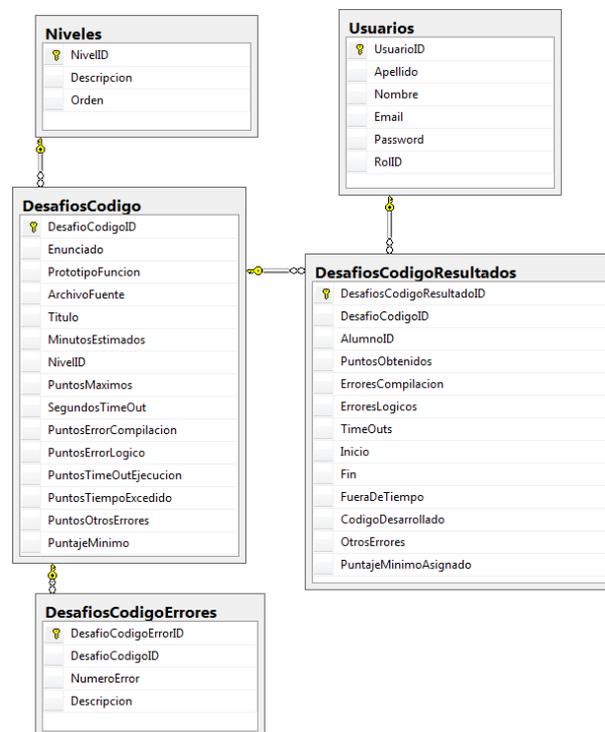
Además se configura el ejercicio con los valores de la Tabla 2.

**Tabla 2.** Parametrización del ejercicio de triángulos

Puntos Máximos	1000
Minutos Estimados	60
Segundos de time out	30
Puntos descontados por error de compilación	5
Puntos descontados por error lógico	20
Puntos descontados por error inesperado	30
Puntos descontados por time-out	30
Puntos descontados por tiempo excedido	200
Puntos mínimos por haber resuelto el desafío	100

En este ejercicio en particular hay una cantidad máxima de puntos que el alumno podrá obtener que es 1.000 y una cantidad mínima que es 100. Se grabará en la base de datos la puntuación final obtenida por el alumno así como otros parámetros que son de interés (por ejemplo: tiempo de resolución, casos de prueba que no pasó satisfactoriamente, etc.).

Toda esta parametrización es soportada por una base de datos relacional cuyo esquema puede verse en la figura 3.



**Figura 3.** Esquema de base de datos para los desafíos de código

## Estado actual del desarrollo y trabajos futuros

El desarrollo de la plataforma se encuentra en su etapa final, ultimando detalles para su lanzamiento como prueba piloto en un curso (en el segundo cuatrimestre). Para ello se está trabajando simultáneamente en generar una base considerable de ejercicios para poder asignar a los alumnos en cada tema.

## Trabajos Relacionados

La gamificación es una técnica mayormente aplicada en nivel secundario, existiendo una menor cantidad de trabajos que la apliquen en el ámbito universitario. En esta sección se describen brevemente algunos artículos de interés que muestran implementaciones en Universidades:

1. La Universidad Javeriana de Bogotá, analizó la posibilidad de implementar una plataforma para apoyar un curso tradicional de programación [13]. Para lo cual se desarrolló un prototipo y como se podrían implementar los distintos conceptos vinculados con la gamificación.
2. Un trabajo realizado en la Universidad Politécnica de Madrid presenta una propuesta para integrar técnicas de gamificación para motivar a los estudiantes con el objetivo de lograr mejores resultados en aprobación y bajar los niveles de deserción. Para lo cual han diseñado una plataforma web en la cual pueden ser cargadas actividades, realizar prácticas y observar la puntuación obtenida así como la clasificación general. [14].
3. Un trabajo realizado en la Universidad Central de Ecuador, trabajo en especial en la problemática de la enseñanza del Programación I, donde los alumnos aprenden a resolver los problemas usando algoritmos y resolución de problemas, tratando a la vez de mejorar las estadísticas de abandono y especialmente alentar la motivación a sus alumnos. Para identificar donde radica el inconveniente, efectúa un análisis completo de material, docentes, alumnos y metodologías aplicadas a proceso enseñanza aprendizaje. Planteando la idea de utilizar moodle para poder gamificar el curso como idea de trabajo futuro [15].

## **Conclusiones**

La metodología planteada permite la utilización de técnicas de gamificación aplicándolas a las actividades tradicionales de la materia de programación como ser la resolución de ejercicios. De esta forma la

ejercitación cobra una nueva dimensión permitiendo aplicar distintos componentes como puntos, rankings e insignias. Llevando a la aplicación de distintas dinámicas como retos, competición, retroalimentación y recompensas.

Esta plataforma permitirá trabajar sobre la dinámica de la progresión al ir habilitando ejercicios organizados por niveles, a medida que se va avanzando sobre los temas de la materia. También la incorporación de herramientas sociales, como la conformación de grupos, permitirá la utilización de la dinámica de relaciones.

Si bien la preparación de ejercicios requiere un gran esfuerzo por parte del docente al tener que realizar un programa completo con los casos de prueba, ese tiempo es compensado con la corrección automática del código de los alumnos por lo que esta plataforma también es adecuada para un cursado a distancia o semi-presencial.

Consideramos por otra parte que es importante generar recursos que permitan motivar a los alumnos en su mayoría “nativos digitales”, quienes están muy predispuestos a hacer uso de la tecnología.

## **Referencias**

- [1] HERNÁNDEZ, Ana Polanco. La motivación en los estudiantes universitarios. Revista Actualidades Investigativas en Educación, 2011, vol. 5, no 2.
- [2] Morales Artero, Juan José. "La gamificación en la universidad para mejorar los resultados académicos de los alumnos." (2013): 1-15.
- [3] S. Deterding, D. Dixon, R. Khaled, and L. Nacke, "From game design elements to

gamefulness: defining gamification,” Proc. 15th ..., pp. 9–15, 2011.

[4] K. Erenli, “The impact of gamification: A recommendation of scenarios for education,” in 2012 15th International Conference on Interactive Collaborative Learning (ICL), 2012, pp. 1–8.

[5] Herranz (2013) Gamification I Feria Informática (febrero 2013). Universidad Carlos III Madrid España.

[6] R. Arenas Paris (2014). Modelo para la Motivación del Aprendizaje de la Programación utilizando Gamification. Pontificia Universidad Javeriana. Bogotá D. C.

[7] Barragán Piña, A. J., Ceada Garrido, Y., Andújar Márquez, J. M., Irigoyen Gordo, E., Gómez Garay, V., & Artaza Fano, F. (2015). Una propuesta para la motivación del alumnado de ingeniería mediante técnicas de gamificación.

[8] Edmodo. Red educativa global (2016). <https://www.edmodo.com/>

[9] Schoology. Learning Management System (2016). <https://www.schoology.com/>

[10] Microsoft. ASP .NET MVC 5 (2016). <http://www.asp.net/mvc/mvc5>

[11] Carnegie Mellon University. Wang A, Chang A, Mark A, Louie K. (2016). Materialize - Un framework web front-end moderno y responsivo basado en Material Design. <http://materializecss.com/>

[12] Nattawat Nonsung, jQuery Flip v1.0.20. A JQuery plugin to flip content with 3D animation. <https://nnattawat.github.io/flip/>

[13] Arenas París, R. J. (2014). Modelo para la Motivación del Aprendizaje de la Programación utilizando Gamification.

[14] Morro González, S. (2015). Técnicas de gamificación para la motivación de los estudiantes.

[15] Beltrán, J., Sánchez, H., & Rico, M. (2015). Análisis cuantitativo y cualitativo del aprendizaje de Programación I en la Universidad Central del Ecuador. Revista Tecnológica-ESPOL, 28(5).