

Generador de problemas de ecuaciones no lineales con gamificación

Hugo Pablo Leyva¹, Rafaela Blanca Silva López²

¹ Universidad Autónoma Metropolitana Azcapotzalco,
México

² Universidad Autónoma Metropolitana Lerma,
México

hpl@correo.azc.uam.mx, r.silva@correo.ler.uam.mx

Resumen. Uno de los problemas más laboriosos y de alguna manera tediosos, de cada periodo escolar de un curso de métodos numéricos, es la generación de problemas para cada uno de los temas. Se requieren diferentes problemas para utilizarse como ejemplos de clase, tareas o exámenes. Por lo tanto, se requiere contar con una amplia gama de problemas distintos para cada periodo escolar para evitar que los estudiantes tengan las respuestas. Los problemas generados deben contar con su solución, además de poderlos resolver en un tiempo equivalente al que se da para resolver el examen. El objetivo del presente trabajo es presentar un generador de problemas para el tema de raíces de ecuaciones no lineales. El conjunto de programas elaborados permite la generación de una gran cantidad de problemas con su solución, que un estudiante puede resolver en el tiempo que se da para realizar los exámenes. Uno de los enfoques con los cuales se puede dar un curso es el de gamificación, que esencialmente trata de usar un ambiente de juego en ámbitos tan diversos como la educación, o la salud, entre otros. Los problemas que se generan en el presente trabajo se enfocan en ecuaciones cuyas raíces tienen una interpretación en un ambiente gamificado. Se presentan los algoritmos para la elaboración de los programas, la puesta en marcha y resultados en cuanto a la generación de problemas usados en el tema de raíces de ecuaciones no lineales para el curso de Métodos Numéricos en Ingeniería de la UAM Azcapotzalco.

Palabras clave: E-learning, gamificación, aprendizaje basado en problemas, métodos numéricos, ecuaciones no lineales, algoritmos inteligentes.

Problem Generator of Nonlinear Equations with Gamification

Abstract. One of the most laborious and somehow tedious problems, of each school period of a course of numerical methods, is the generation of problems for each of the topics. Different problems are required to be used as examples of class, assignments, or exams. Therefore, a wide range of different problems are required for each school term to prevent students from having the answers. The problems generated must have their solution, in addition to being able to solve them in a time equivalent to that given to solve the exam. The objective of this work is to present a problem generator for the root theme of nonlinear equations.

The set of elaborate programs allows the generation of a lot of problems with their solution, which a student can solve in the time given to take the exams. One of the approaches with which a course can be given is gamification, which essentially tries to use a gaming environment in areas as diverse as education, or health, among others. The problems generated in this work focus on equations whose roots have an interpretation in a gamified environment. Algorithms for program development, commissioning and results are presented in terms of the generation of problems used in the topic of roots of nonlinear equations for the course of Numerical Methods in Engineering of the UAM Azcapotzalco.

Keywords: E-learning, gamification, problem-based learning, numerical methods, nonlinear equations, intelligent algorithms.

1. Introducción

Una de las labores docentes que demandan más tiempo es la generación de problemas para un curso como el de Métodos Numéricos. Hay que cuidar detalles como la complejidad que podría llevar a un tiempo largo de solución por parte de los estudiantes. Es necesario, contar con la solución del problema para que dado el caso se proporcione retroalimentación al estudiante si la solicitase. Se debe contemplar la elaboración de problemas que se utilizarán como ejemplos en clase, otros distintos para usarse en las tareas y algunos más para los exámenes. Esto evita que los estudiantes puedan copiarse con facilidad.

El aprendizaje basado en problemas puede ser usado en cursos de ingeniería y ciencias como lo es el de Métodos numéricos con excelentes resultados. Precisamente para este curso, se requiere una cantidad enorme de problemas. Otra manera de hacer más llevadera la clase de un curso como el de métodos numéricos es la gamificación [1]. Al darle un aire de juego a las actividades de la clase permite atrapar el interés del alumnado [2].

El e-learning, se ha convertido en una modalidad de conducción del proceso de enseñanza y aprendizaje vital en los tiempos actuales de la pandemia. Este permite atender a un mayor número de estudiantes empleando un LMS (Learning Management System). Este tipo de herramientas, tienen la virtud de contar con facilidades para gestionar una batería de problemas, tal que se pueden aplicar muchas versiones de un examen sin que estos se repitan. Para esto requieren de igual manera una cantidad considerable de problemas.

Uno de los temas más relevantes de un curso de Métodos numéricos es el de raíces de ecuaciones no lineales. Este tema se encuentra con mucha frecuencia en otros cursos de ingeniería en las Instituciones de Educación Superior (IES). Para contar con el número suficiente de problemas, se podrían copiar de algún libro o página web, sin embargo, sería difícil encontrar suficientes problemas para cambiarlos en cada ciclo escolar.

Otra dificultad radica en que si se descubre cual es la fuente de la que se tomaron los problemas, los estudiantes podrán tener acceso a ellos de un curso a otro y perderán su validez como instrumentos de evaluación.

Por lo anterior, se proponen el desarrollo de algoritmos inteligentes y el desarrollo de programas para la generación de problemas tales que: se cuente con un gran número

de problemas (miles) para evitar que se compartan por los estudiantes, se tenga la solución para dar retroalimentación a los alumnos, se generen en poco tiempo, se puedan usar para ejercicios, tareas o exámenes, así como para alimentar la batería de problemas que requiere un LMS para aplicar exámenes sin repetición de los problemas.

Las características de los programas contemplan:

- Generar problemas para el tema de raíces de ecuaciones no lineales de la asignatura de Métodos Numéricos en Ingeniería, junto con su solución.
- La cantidad de problemas es variable.
- Los parámetros de solución (máximo de iteraciones, cifras significativas, etc.) se pueden configurar.
- Sólo se generan problemas que se resuelvan en pocas iteraciones.
- Se apliquen filtros adicionales para generar problemas no tan laboriosos para los estudiantes.
- Las soluciones de los problemas se pueden interpretar en el contexto de un problema gamificado.
- Se genera también si se desea, un archivo que corresponde a una batería de problemas que tiene el formato para ser usada en un LMS.

Para cumplir con estas características y restricciones se diseña una serie de algoritmos inteligentes que satisfacen todas las necesidades planteadas.

2. Trabajos relacionados

Los autores que abordan el concepto de gamificación, consideran que es una táctica utilizada para motivar a las personas en actividades ordinarias [3], usar el juego en un contexto no relacionado con el mismo para cambiar el comportamiento de las personas [4]. La gamificación [5] es un motivador en el aprendizaje del alumno, de hecho aumenta el interés y la motivación para abordar un tema específico [2]. En [6] se preparan ejercicios en Excel para usarlos en un enfoque de aula invertida.

Los autores se enfocan más en cómo implementar su curso que del desarrollo de ejercicios. En [7] preparan problemas a partir de ejercicios de libros de texto que tienen un enfoque de aprendizaje adaptativo orientado a la trayectoria de presentación del material de un curso de métodos numéricos.

En [8] se proponen ejercicios diseñados por expertos que son validados con software numérico para desarrollar material de enseñanza. Los problemas son resueltos con Maple o Matlab. Se enfocan más en el desarrollo del material y su uso para la enseñanza. En [9] diseñaron ejercicios usando el software de Mathematica con objetos de aprendizaje para el curso de Métodos Numéricos.

En [10] usan GeoGebra que es un sitio web dónde se cuentan con recursos para enseñanza de las matemáticas y ciencias. Los ejercicios se emplean en un curso de métodos numéricos montado en el LMS Moodle. En [11] usan la herramienta web e-status para generar problemas numéricos. Los problemas generados son de estadística, más que de métodos numéricos.

En [12] en lugar de emplear un software que genere los problemas, se van a un enfoque más simple que es usar de un repositorio los problemas ya hechos. Los problemas son de ingeniería química, son una colección de 10 problemas numéricos.

Se trata de problemas típicos de los ingenieros químicos, cuya solución es numérica. Se analiza la solución de estos con varios softwares.

En [13] se usa OptiA, este software se emplea para resolver problemas de programación matemática, que son problemas de optimización. No se pretende obtener un generador de problemas. No se identifica algún trabajo en el que se propongan algoritmos inteligentes ni se desarrolle una aplicación que genere los problemas con sus soluciones satisfaciendo las características y restricciones mencionadas. Los programas propuestos en este trabajo cubren estas deficiencias de manera efectiva.

3. Algoritmos inteligentes y programas para la generación de problemas

Se diseñaron algoritmos que satisfacen las restricciones establecidas para la generación de problemas de raíces de ecuaciones no lineales. Contemplando la estrategia didáctica de gamificación en la enseñanza de las ecuaciones generadas. Se analiza un caso de uso gamificado en el que se utilizan ecuaciones no lineales.

Caso de uso: Para cobrar la herencia.

A tu difunto tío, le encantaban las matemáticas y los acertijos. Como dudaba de que todos sus sobrinos fueran merecedores de su gran fortuna, les dejó un acertijo en su testamento. Al momento de leerlo el albacea menciona la condición que debe cubrir el heredero que se quedará con toda la fortuna. "Mi fortuna será para aquel de mis sobrinos que se presente a reclamar la herencia en la fecha y hora señaladas, en las oficinas del albacea. Solo aquel que llame a la puerta el día y a la hora correcta podrá entrar. Se descalificará a cualquiera que pretenda quedarse a esperar ante la puerta". La fecha y la hora están dadas por las soluciones del siguiente par de ecuaciones:

$$a_2f(x) + a_1x^n + a_0 = 0.$$

$$b_4x^4 + b_3x^3 + b_2x^2 + b_1x + b_0 = 0.$$

"El que se presente a reclamar la herencia deberá de entregar por escrito, la solución junto el razonamiento que empleó para hallarla. En el caso de que 2 o más herederos se presenten a la hora y fecha correctas, el ganador será aquel que tenga la solución más precisa acorde a los requerimientos de los valores a encontrar y con el menor número posible de pasos. En el supuesto de empate, el albacea les proporcionara más problemas a resolver en ese momento. Ganará el que resuelva todos en las mismas condiciones que el problema que los llevo hasta ahí".

Si tú fueras uno de los sobrinos ¿Cuál debe ser la metodología más adecuada para resolver la ecuación y así poder reclamar la herencia?

En este problema $f(x)$ es una ecuación no lineal que puede ser alguna de las siguientes: $\sin(x)$, $\cos(x)$, $\sinh(x)$, $\cosh(x)$, $\tanh(x)$, $\operatorname{ctgh}(x)$, $\operatorname{sech}(x)$, $\operatorname{csch}(x)$, e^x , e^{-x} , e^{x^2} , e^{-x^2} , $\ln(x)$, $\log(x)$, \sqrt{x} , $\sqrt[3]{x}$, $\frac{1}{x}$.

Las constantes son: $a_2, a_1, a_0, n, b_4, b_3, b_2, b_1, b_0$.

El grado del término polinomial n varía de 1 a 4.

Los coeficientes de la ecuación no lineal y el grado del término polinomial se generan aleatoriamente.

Los coeficientes del polinomio se obtienen al aproximar por la serie de Taylor a un polinomio de grado 4 la ecuación no lineal. Las raíces reales de este polinomio se usan para generar aproximaciones a las raíces de la ecuación no lineal. El polinomio se resuelve analíticamente con la solución general de la ecuación de tercer o cuarto grado según se requiera [14]. Solo se consideran ecuaciones no lineales cuyo polinomio de Taylor tengan raíces reales.

Las raíces tanto del polinomio como de la ecuación no lineal se deben poderse interpretar ya sea como fechas u horas. Para esto se toman las primeras seis cifras de las raíces considerando el punto decimal y se interpretan de la siguiente manera:

- Si las 2 primeras cifras delante del punto decimal están entre 1 y 23 se puede tratar ya sea de una fecha o una hora.
- Si estas cifras están entre 24 y 31 se puede tratar de una fecha.
- Si son 0 se puede tratar de una hora.
- Si las segundas 2 cifras después del punto decimal están entre 01 y 12 se puede tratar ya sea de una fecha o una hora.
- Si estas cifras están entre 13 y 59 se puede tratar de una hora.
- Si son 00 se puede tratar de una hora.
- Si las terceras 2 cifras están entre 00 y 59 se puede tratar ya sea de una fecha o una hora.
- Si estas cifras están entre 60 y 99 se puede tratar de una fecha.
- Las fechas tienen el formato día/mes/año.
- Los días varían de 1 a 31.
- Los meses varían de 01 a 12.
- Los años que se manejan con 2 cifras varían de 00 a 99.
- Las horas tienen el formato HH:MM:SS.
- Las horas varían de 0 a 23.
- Los minutos varían de 00 a 59.
- Los segundos varían de 00 a 59.

Al procesar las 6 primeras cifras de cada raíz se deben de tener combinaciones validas conforme a las reglas anteriores. También se deben considerar el número de días que tiene cada mes, inclusive en año bisiesto.

Se descartan las raíces negativas y complejas.

Es posible que una raíz se pueda interpretar tanto como fecha u hora.

En la figura 1 se tiene el diagrama de flujo del algoritmo que genera las ecuaciones no lineales y polinomios.

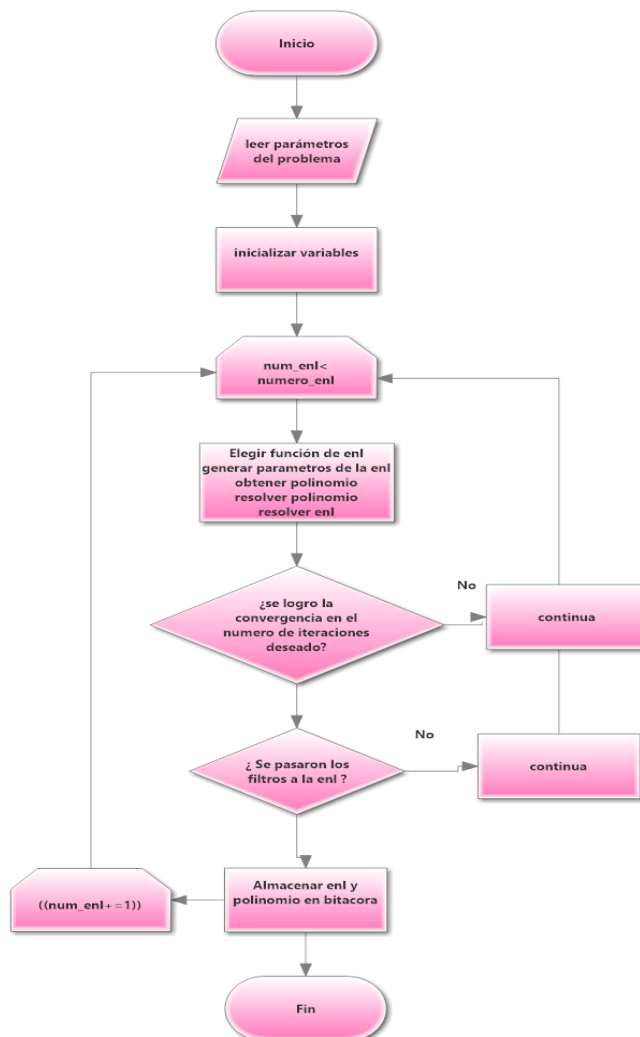


Fig. 1. Algoritmo para generar ecuaciones no lineales y polinomios.

La ecuación no lineal se resuelve usando el método de Newton Rapshon modificado [15]. Se prefiere este método por su velocidad de convergencia y por ser inmune a problemas de convergencia que se pueden presentar en caso de tener raíces dobles.

Entre los parámetros que se proporcionan están:

- El número de problemas a generar.
- El nombre del directorio donde se almacenan los problemas junto con su solución y las bitácoras con el listado final de problemas generados.
- El límite del rango de valores para el coeficiente a_2 que se generara aleatoriamente.

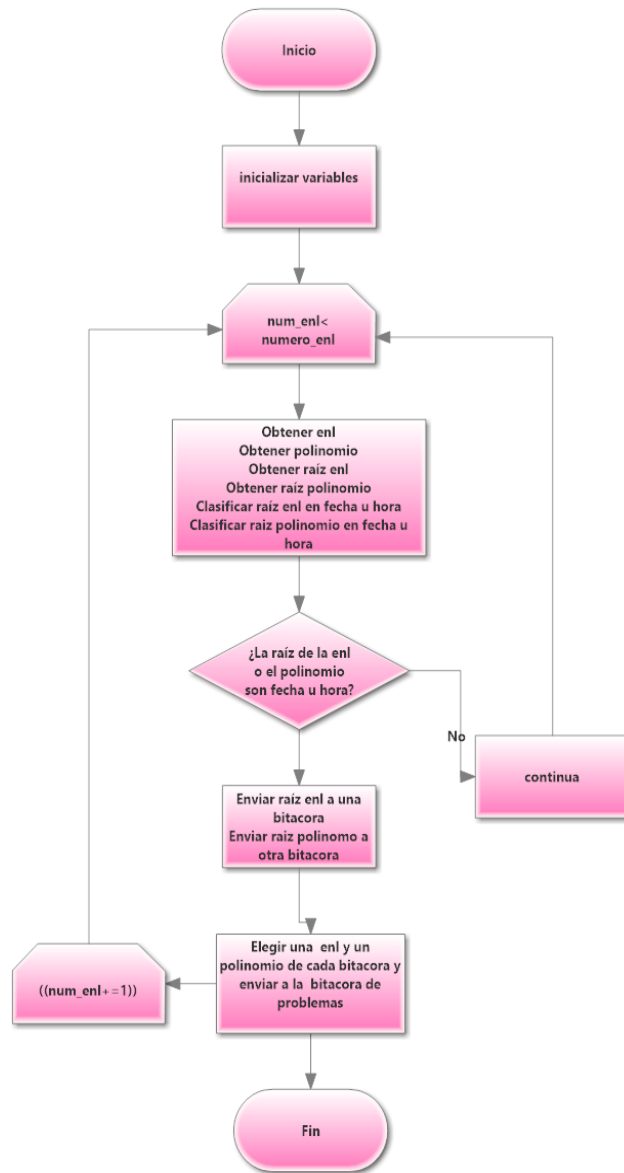


Fig. 2. Algoritmo para generar parejas de ecuaciones no lineales y polinomios.

- El límite del rango de valores para el coeficiente a_1 que se generara aleatoriamente.
- El límite del rango de valores para el coeficiente a_0 que se generara aleatoriamente.

- El número de cifras significativas con el que se desea encontrar las raíces.
- El número máximo de iteraciones que se usara para el método de Newton Rapshon modificado.
- El número de cifras que se usaran para hacer los cálculos.

Los filtros que se mencionan son:

- Los coeficientes a_2, a_1, a_0 deben de ser distintos a cero para que se pueda considerar una ecuación no lineal con el formato que se usa.
- El polinomio de Taylor asociado a la ecuación no lineal debe tener al menos una raíz real.

En la figura 2 se tiene el diagrama de flujo del algoritmo para obtener las parejas de ecuaciones no lineales y polinomios cuyas raíces se pueden interpretar como fechas u horas, que se requieren para el problema gamificado.

El algoritmo de la figura 2 contempla las siguientes restricciones:

- Se descartan las raíces negativas. Se consideran solo las primeras 6 cifras de la raíz considerando el punto decimal.
- Se verifica que las seis cifras cumplan el formato de fechas u horas mencionado antes.
- Se clasifica en raíces que se interpretan solo como fechas, solo como horas o ambiguas.
- Se forman parejas de una ecuación no lineal y un polinomio que tengan una raíz que se interpreta como fecha y otra como hora. También se considera que solo una raíz se defina como fecha u hora y la otra que pueda ser ambigua.

Los programas se elaboraron en los lenguajes Korn Shell, C, Latex, HTML. Como se usó el sistema operativo Linux se emplearon Shell scripts en Korn Shell para la generación de las ecuaciones y manipulación de los archivos generados con la lista de problemas, así como con su solución. La solución de los polinomios y las ecuaciones no lineales se programaron en lenguaje C. El programa genera páginas Web con la lista de problemas y otra con las soluciones. Estas páginas están en HTML.

Para generar el listado de las ecuaciones en formato algebraico se usó Latex. Se utilizan librerías de Linux para obtener las ecuaciones como imágenes en una página web. Para realizar los cálculos en los Shell scripts de Korn Shell se usó la herramienta BC, la cual es una calculadora que viene con el sistema operativo. En esta se especifica la cantidad de cifras que se emplean al hacer cálculos. Las funciones que generan valores aleatorios se hicieron con facilidades que proporciona Korn Shell.

3.1. Pruebas

Se hicieron 10 corridas con los siguientes parámetros cada una:

- Numero de ecuaciones a generar 1000.
- Directorio donde almacenan los problemas generados T21I_# donde el # vario de 0 a 9.
- El límite del rango de valores para el coeficiente a_2 es 100.
- El límite del rango de valores para el coeficiente a_1 es 10.

```

< > ↻ Archivo | C:/Users/Profesor/Downloads/T21I_0/T21I_0_ecs_raiz/index.html
1.  $-66.0 \operatorname{sech}(x) + 1.0x^4 - 83.9836120605 = 0$ 
2.  $x_0 = +3.080371472894498641$ 
3.  $x_1 = -3.080371472894498641$ 
4.  $-5.1699523925 \operatorname{senh}(x) + 8.0x + 33.0 = 0$ 
5.  $x_0 = +3.110747110645122326$ 

```

Fig. 3. Listado de ecuaciones no lineales generadas con su solución.

```

< > ↻ Archivo | C:/Users/Profesor/Downloads/T21I_0/T21I_0_pol_raiz/index.html
1.  $-149.98361206 + 33x^2 + 1x^4 = 0$ 
2.  $x_0 = +2.012044687887661443$ 
3.  $x_1 = -2.012044687887661443$ 

```

Fig. 4. Listado de polinomios generados con su solución.

- El límite del rango de valores para el coeficiente a_0 es 100.
- El número de cifras significativas para encontrar las raíces es 8.
- El número máximo de iteraciones es 10.
- El número de cifras para hacer los cálculos es 10.

Para realizar las corridas se hizo con el siguiente comando en una terminal de Linux:

`$ for ((i=0; ((i<10)); ((i+=1)))) do nohup time generaecuacionesaleatorias.ksh 1000 T21I_0$i 100 10 100 8 10 10 & done` El ciclo for del comando anterior se ejecutará un total de 10 veces. El nohup junto con el carácter & en el comando anterior permiten que el programa se quede en ejecución aun si la terminal llegara a cerrarse. El time devolverá el tiempo de ejecución de cada corrida del programa. Las 10 corridas se hacen concurrentemente.

El primer argumento son el número de problemas a generar en este caso 1000. El segundo argumento es el nombre de cada directorio donde se almacenarán los problemas generados. En este caso serán T21I_0, T21I_1, ... T21I_9. El tercer argumento es límite del rango de valores aleatorios que se generan para el término que corresponde a alguna de las funciones que se emplean en este programa. En este caso 100. El cuarto argumento es límite del rango de valores aleatorios que se generan para el termino polinomial.

En este caso 10. El quinto argumento es límite del rango de valores aleatorios que se generan para el termino independiente. En este caso 100. El sexto argumento son las cifras significativas con las que encontrarán las raíces de las ecuaciones no lineales. En este caso 8. El séptimo argumento es el máximo de iteraciones con las que se realizara el método numérico que resuelve la ecuación no lineal. En este caso 10. El octavo argumento son las cifras con las que el comando BC realizara los cálculos para generar las ecuaciones no lineales o polinomios. En este caso 10. Se muestra a continuación parte de las páginas web generadas con la lista de problemas y sus soluciones.

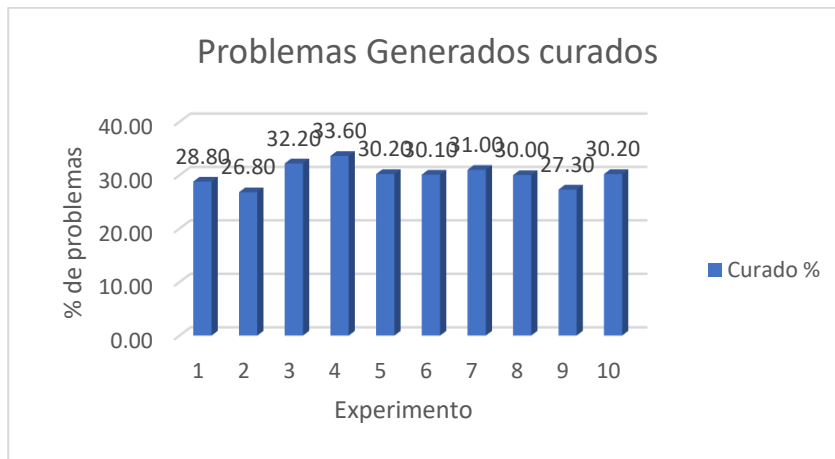


Fig. 5. Porcentaje de problemas generados curados.

Tabla 1. Características del servidor y tiempo de corrida.

Sistema operativo	CPU	Memoria RAM	Disco duro	Tiempo de corrida
Linux CentOS 6.7	24 Intel	268 Gb	7 Tb	30 min
Linux CentOS 7.8	2 Intel	3 Gb	214 Gb	3 hr 30 min

De los 1000 problemas generados, sólo el 30% aproximadamente, son interpretados adecuadamente en el contexto de gamificación del caso propuesto.

Se entiende por problemas curados aquellos cuya solución se puede interpretar como fechas u hora. En la Tabla 1 se muestran las características de 2 servidores donde se hicieron las pruebas con su tiempo de ejecución. Un ejemplo de un problema generado por el programa es: Para cobrar la herencia.

A tu difunto tío, le encantaban las matemáticas y los acertijos. Como dudaba de que todos sus sobrinos fueran merecedores de su gran fortuna, les dejó un acertijo en su testamento. Al momento de leerlo el albacea menciona la condición que debe cubrir el heredero que se quedará con toda la fortuna. "Mi fortuna será para aquel de mis sobrinos que se presente a reclamar la herencia en la fecha y hora señaladas, en las oficinas del albacea. Solo aquel que llame a la puerta el día y a la hora correcta podrá entrar. Se descalificará a cualquiera que pretenda quedarse a esperar ante la puerta". La fecha y la hora están dadas por las soluciones del siguiente par de ecuaciones:

$$33e^x + 9x^4 - 34.4176330566 = 0$$

$$-1.2916666666x^4 + 0x^3 + 22.4110412597x^2 + 0x - 97 = 0$$

"El que se presente a reclamar la herencia deberá de entregar por escrito, la solución junto el razonamiento que empleó para hallarla. En el caso de que 2 o más herederos se presenten a la hora y fecha correctas, el ganador será aquel que tenga la solución más precisa acorde a los requerimientos de los valores a encontrar y con el menor número

posible de pasos. En el supuesto de empate, el albacea les proporcionara más problemas a resolver en ese momento. Ganará el que resuelva todos en las mismas condiciones que el problema que los llevo hasta ahí".

Si tú fueras uno de los sobrinos ¿Cuál debe ser la metodología más adecuada para resolver la ecuación y así poder reclamar la herencia?

La ecuación no lineal tiene 2 raíces:

$$x_0=+0.04206064203937895374$$

$$x_1=-1.295971852465515717$$

Solo se considera la raíz positiva. Tomando las primeras cifras tenemos 0.0420. La primera cifra es un cero por lo cual es una hora. Serían las 0:04:20, o sea las cero horas con cuatro minutos y veinte segundos.

El polinomio tiene 4 raíces:

$$x_0=+2.876013313714593789$$

$$x_1=-2.876013313714593789$$

$$x_2=+3.013143039440491222$$

$$x_3=-3.013143039440491222$$

Solo se consideran las raíces positivas. Tomando las primeras cifras tenemos 2.8760 y 3.0131. La primera raíz se descarta porque tenemos 87 como las segundas cifras, esta no se puede considerar ni una fecha, ni una hora. De la segunda puede ser una fecha o una hora. Como hora sería 03:01:31 que sería las tres horas con un minuto y treinta y un segundos. Como fecha sería 03/01/31 que sería el tres de enero de 2031.

Como la raíz de la ecuación no lineal se interpreta como una hora, la raíz del polinomio se interpretará como una fecha. Por todo lo anterior hay que tocar la puerta del albacea el 03/01/31 a las 0:04:20.

4. Conclusiones y trabajo a futuro

Se observa que se genera una gran cantidad de problemas (miles) en poco tiempo. De estos sólo un porcentaje pequeño alrededor del 30 % son válidos para el contexto de gamificación del caso de uso utilizado. Aun así, la cantidad de problemas son suficientes para usarlos como ejercicios en clase, tareas y exámenes, sin que se repitan y puedan ser compartidos por los estudiantes.

Dependiendo de la metodología para impartir el curso es suficiente con un solo caso de uso gamificado, ya que se usan los problemas generados en un solo tema del curso, y se puede dejar como tarea un solo problema. De esta forma el alumno no se verá abrumado por tener que resolver todos los problemas generados.

Los demás problemas pueden sugerirse que los resuelva como practica para que se prepare para el examen. El tiempo de solución de los problemas, cumple con la restricción del tiempo asignado para realizar el examen, ya que a lo mucho se tomará 10 iteraciones para ser resueltos por un método numérico como es el de Newton-Raphson modificado.

Si no se considera un contexto de gamificación, se cuenta entonces con una gran cantidad problemas para el curso (100% de problemas generados). El tiempo de ejecución para generar unos 1000 problemas es razonable, aunque no se cuenten con muchos recursos de cómputo por parte del servidor.

El algoritmo que se usó para generar los problemas para un uso gamificado, redujo el total a un 30%, ya que deben de cumplir los requerimientos del caso de uso. Como trabajo futuro se pretende adaptar el algoritmo para generar las raíces que ya cumplan con los requisitos del problema gamificado, y con estas generar los problemas con las ecuaciones requeridas.

Es probable que la dificultad que se puede tener con este enfoque es que las ecuaciones resultantes tarden más de las iteraciones requeridas para que el problema se pueda resolver en el tiempo destinado a un examen.

Se analizarán otros casos de uso para problemas gamificados tales que, las raíces de las ecuaciones no lineales se puedan interpretar de acuerdo con los requerimientos establecidos en el caso de uso. Por ejemplo, que las raíces sean: edades, coordenadas GPS, pesos, estaturas, entre otros.

Estos casos de uso se podrán usar para darle variedad a los problemas gamificados que se dejarán como tarea al alumno para llamar su atención y no aburrirlo con un solo ejemplo de caso de uso. También se permitirá cambiar el caso de uso de un trimestre a otro. Por último, se podrá dejar diferentes casos de uso a cada estudiante.

Referencias

1. Genç, Z., Tinmaz, H.: The perception on fundamentals of online courses a case on prospective instructional designers. *European Journal of Contemporary Education*, vol. 15, no. 1, pp. 163–172 (2016) doi: 10.13187/ejced.2016.15.163
2. Simões, J., Díaz, R., Vilas, A.: A social gamification framework for a K-6 learning platform. *Computers in Human Behavior*, vol. 29, no. 2, pp. 345–353 (2013) doi: 10.1016/j.chb.2012.06.007
3. Deterding, S., Dixon, D., Khaled, R., Nacke, L. E.: From game design elements to gamefulness: Defining gamification. In: *Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environ*, pp. 9–15 (2011) doi: 10.1145/2181037.2181040
4. Aristizábal, J. H., Colorado, H., Gutiérrez, H.: El juego como una estrategia didáctica para desarrollar el pensamiento numérico en las cuatro operaciones básicas. *Sophia*, vol. 12, no. 1, pp. 117–125 (2016)
5. Nah, F. F. H., Telaprolu, V. R., Rallapalli, S., Venkata, P. R.: Gamification of education using computer games. In: Yamamoto, S. (eds) *Human Interface and the Management of Information, Information and Interaction for Learning, Culture, Collaboration and Business, HIMI 2013, Lecture Notes in Computer Sciences*, vol. 8018, no. 3, pp. 99–107 (2013) doi: 10.1007/978-3-642-39226-9_12
6. Pramita, M., Sukmawati, R. A., Sari, D. P.: The implementation of flipped classroom assisted by learning management system for numerical method courses. vol. 274, pp. 158–162 (2018) doi: 10.2991/iccite-18.2018.36
7. Clark, R. M., Kaw, A.: Adaptive learning in a numerical methods course for engineers: Evaluation in blended and flipped classrooms. *Computer Applications in Engineering Education*, vol. 28, no. 1, pp. 62–79 (2020) doi: 10.1002/cae.22175
8. Handayani, A., Herman, D. T., Fatimah, S.: Developing teaching material software assisted for numerical methods. *Journal of Physics: Conference Series, International Conference on Mathematics and Science Education*, vol. 895, no. 1 (2017) doi: 10.1088/1742-6596/895/1/012068

9. Caligaris, M., Rodríguez, G., Favieri, A., Laugero, L.: Developing mathematical abilities using learning objects when learning numerical methods. ICERI2017 Proc., vol. 1, pp. 6018–6026 (2017) doi: 10.21125/iceri.2017.1569
10. Becerra-Romero, A., Díaz-Rodríguez, M., González-Estrada, O. A.: Development of a virtual learning environment for the subject numerical methods under Moodle. J. Journal of Physics: Conference Series, vol. 1161, no. 1 (2019) doi: 10.1088/1742-6596/1161/1/012010
11. Alastrué, G., Antonio, J.: Problemas numéricos con corrección automática: qué se puede y qué se debe hacer. In: V Jornadas para la Enseñanza y Aprendizaje de la Estadística y la Investigación Operativa, pp. 35–52 (2014)
12. Cutlip M. B.: A collection of 10 numerical problems in chemical engineering solved by various mathematical software packages. Computer Applications in Engineering Education, vol. 6, no. 3, pp. 169–180 (1998) doi: 10.1002/(SICI)1099-0542(1998)6:3<169: AID-CAE6>3.0.CO;2-B
13. Doležal, J., Fidler, J.: Numerical solution of dynamic optimization problems using parametrization and Op^hA software. vol. 78, NO. 2-3, pp. 111–121 (1996) doi: 10.1016/0096-3003(96)00002-1
14. Perry, R. H., Green, D. W., Maloney, J. O.: Perry manual del ingeniero químico. 3ra ed, México, DF: McGraw-Hill (1992)
15. Burden, R. L., Faires, D. J., Burden, A. M.: Análisis Numérico. 10a ed, México, D.F.: Cengage Learning Editores, S.A. de C.V (2017)

