



# INNOVACIÓN y SOFTWARE





**Vol. 1 N° 2 2020 Septiembre - Febrero**

ISSN N°: 2708-0935

DOI: 10.48168/innosoft.s2

ARK: ark:/42411/s2

PURL: 42411/s2

Depósito Legal: 2023-08884

Periodicidad: Semestral

Publicado: 30/09/2020

## **Editado por:**

Universidad La Salle

RUC: 20456344004

Av. Alfonso Ugarte N° 517, Cercado, Arequipa

## **COMITÉ EDITORIAL**

Editor jefe:

Dr. Yasiel Pérez Vera

Editores asociados:

MSc. Anié Bermudez Peña

MSc. Percy Oscar Huertas Niquén

Miembros del Consejo Editorial

Dr. José Manuel Patricio Quintanilla Paulet

Hno. Jacobo Meza Rodríguez

Dr.C José Javier Zavala Fernández

Dr.C Cristian José López del Álamo

Dr.C Álvaro Rodolfo Fernández del Carpio

MSc. Paul Mauricio Mendoza del Carpio

Corrección de estilos

MSc. Orlando Alonso Mazeyra Guillén

Maquetación

Diego Joseph Añazco Bolívar



## EDITORIAL

Los avances de las investigaciones y el desarrollo tecnológico en la Ingeniería de Software y las Ciencias de la Computación

**p. 4 - 5**

## ARTÍCULOS CORTOS

Gestión de la configuración y el cambio en proyectos de servicios TI

Autores: Gretel Danitza Canasa Condori, Fernando Omar Cayo Riveros

**p. 6 - 11**

## ARTÍCULOS ORIGINALES

MPI vs OpenMP: Un caso de estudio sobre la generación del conjunto de Mandelbrot

Autor: Ernesto Soto Gómez

**p. 12 - 26**

Resolución paralela de sistemas triangulares

Autores: Ihosvany Rodríguez González, Anié Bermudez Peña

**p. 27 - 39**

Design Thinking en la planificación de pruebas de software

Autores: Yasiel Pérez Vera, Juan José Gallegos Valdivia, Sandra María Zapata Quentasi, Doris Marcela Ccama Yana, Rosa Elvira Choque Apaza

**p. 40 - 51**

Sistema para la evaluación integral estudiantil

Autores: Duriet Aguilera Álvarez, Alcides Rodríguez Rodríguez

**p. 52 - 61**

Aplicación de software gamificada en la asignatura teleinformática

Autores: Dargel Veloz Morales, Laritza González Marrero

**p. 62 - 74**



La Revista Innovación y Software de la Facultad de Ingeniería, en la Universidad La Salle, se complace en presentar este segundo número que tiene como objetivo el promover investigaciones, los cambios y usos de nuevos elementos tecnológicos y su interrelación con la Ingeniería de Software y la Ciencia de la Computación.

Estas tecnologías, como puede notarse, han venido a revolucionar el mundo en todos los ámbitos de desarrollo, sea este social, económico, político y educativo, por ello la importancia de la educación en esta área es cada vez mayor. En el presente número, se concentran estudios de los avances de las investigaciones y desarrollo tecnológico sobre diferentes áreas, generados por los investigadores, académicos y estudiantes; presentándose avances que resuelven problemas reales basados en diferentes principios.

Ante el actual escenario de pandemia mundial se está poniendo de manifiesto factores claves como el talento y la tecnología sustentado en actividades de investigación orientados a encontrar soluciones adecuadas para estos problemas. Despierta el intelecto y permite buscar soluciones prácticas a problemas que sustentan criterios sobre el uso de modelos, técnicas y metodologías de la Ingeniería de Software y la Ciencia de la Computación.

Por tanto, esta tecnología está resultando ser un factor decisivo en las mejores prácticas a la hora de buscar soluciones. Soluciones orientadas a buscar criterios que permitan encontrar juicios en problemas de la gestión y configuración y el cambio, programación paralela, la resolución de sistemas triangulares, el uso del *Design Thinking* en las pruebas de software, la reducción del tiempo en el procesamiento de la información académica, así como la aplicación de la gamificación en asignaturas de teleinformática.

Estos pocos ejemplos son posibles gracias a diferentes servicios de computación en medios óptimos; los mismos que permiten el teletrabajo sin depender de la infraestructura física, así como de tecnologías complementarias como las de *big data* y *machine learning* que permiten trabajar sobre criterios autónomos y de permanente interacción.

Estas tecnologías habilitadoras muy reconocidas como la cuarta revolución industrial son las que de manera eficaz ayudarán a superar la crisis vivida, junto al talento de las personas que las han creado y el talento y profesionalidad de investigadores que apuestan por soluciones que otorgan flexibilidad de uso a profesionales cuando se enfrentan a problemas de ingeniería en la vida real.

En el futuro tenemos que enfocarnos en la importancia de la formación y la educación de las diferentes áreas, sin descuidar aquellas investigaciones que dan sustento a modelos de soporte a la Ingeniería de Software y la Ciencia de la Computación. Formación orientada a la interrelación de disciplinas que pueden sustentar el soporte de nuevas de ellas.



El primer artículo muestra como el uso de las tecnologías de la información depende de la estrategia empleada por los administradores y del modelo escogido relacionado con la gestión de la configuración y el cambio. Hace notar que los cambios son actividades recurrentes por lo que recomienda que debemos estar preparados para enfrentarnos a ellos.

En el segundo artículo se lleva a cabo una comparación de herramientas relacionadas con el paso de mensajes y el multiprocesamiento abierto debido a los diferentes enfoques en la comunicación de tareas. Para ello analizan el caso de estudio de la computación paralela del conjunto de Mandelbrot concluyendo la existencia de un mejor desempeño en el multiprocesamiento abierto.

El tercer artículo muestra la investigación realizada sobre la implementación y comparación de un conjunto de algoritmos paralelos con respecto a un algoritmo secuencial que resuelve sistemas de ecuaciones triangulares; se toman en cuenta las métricas de tiempo aritmético, tiempo de comunicaciones, aceleración y eficiencia máxima.

El cuarto artículo se enfoca en la utilización del *Design Thinking* orientada en la planificación de las pruebas de software. Las pruebas pueden definirse desde etapas tempranas de la construcción de software y definidas en la fase de educación de requisitos, pero deben ser planificadas para llevar a cabo respuestas del usuario que permitan definir la verificación de los requisitos de software.

En el artículo quinto se muestran los resultados de la implementación de un Sistema para la Evaluación Integral Estudiantil con el objetivo que se centra en el proceso de integridad y caracterización de los estudiantes por medio de expedientes digitales. Dicho sistema fue desarrollado en Python y emplea PostgreSQL como servidor de base de datos y empleando la metodología AUP-UCI.

Finalmente, el sexto artículo trata de la aplicación de la gamificación para la enseñanza de asignaturas de Teleinformática. El tema, sumamente interesante, trata de resolver las dificultades sobre el cumplimiento de objetivos en este tipo de asignaturas y propone una guía para el diseño y aplicación de un instrumento evaluativo.

**Comité Editorial**

# Gestión de la configuración y el cambio en proyectos de servicios de TI

## *Configuration and change management in IT services projects*



### **Gretel Danitza Canasa Condori**

Universidad La Salle

@ [gcanasac@ulasalle.edu.pe](mailto:gcanasac@ulasalle.edu.pe)

id <https://orcid.org/0000-0003-2580-2459>

### **Fernando Omar Cayo Riveros**

Universidad La Salle

@ [fcayorr@gmail.com](mailto:fcayorr@gmail.com)

id <https://orcid.org/0000-0003-2921-8153>

RECIBIDO 20/09/2020 • ACEPTADO 21/09/2020 • PUBLICADO 30/09/2020

## RESUMEN

En la actualidad se nota un creciente uso de las tecnologías de la información, contribuyendo al desarrollo de las organizaciones. La diferencia entre el éxito y fracaso del uso de las tecnologías de la información no depende mucho de las características de estas, sino de la habilidad de los administradores en usarla estratégicamente, por lo que se han desarrollado modelos de la gestión de servicios de las tecnologías de la información con la finalidad de las organizaciones puedan ofrecer servicios de manera eficaz y con mayor nivel de calidad. Estos modelos ven varios aspectos, uno de estos aspectos es la gestión de la configuración. Los cambios van a ocurrir en cualquier momento por lo que debemos estar preparados. Se hizo una comparación de tres modelos de gestión de servicios de las tecnologías de la información enfocándose en la gestión de la configuración y el cambio. Finalmente se puede concluir que los modelos de gestión de servicios de tecnologías de información, pueden ser combinados utilizando los procesos necesarios que se adecuen a un proyecto.

**Palabras claves:** Gestión de la configuración, gestión de servicios, CMMI-SVC, ITIL, COBIT.

## ABSTRACT

*Nowadays, there is a growing use of information technologies, contributing to the development of organizations. The difference between the success and failure of the use of information technologies does not depend much on the characteristics of this, Rather the ability of managers to use it strategically. Therefore, models of Information Technology Service Management have been developed so that organizations can offer services effectively and with a higher level of quality. These models see several aspects, one of which is configuration and change management. Changes will occur at any time so we must be prepared. A comparison of three IT service management models was made focusing on configuration management. IT Service Management models can be combined using the necessary processes and adapting them to the project.*

**Keywords:** *IT Service Management, configuration management, CMMI-SVC, ITIL, COBIT.*



## INTRODUCCIÓN

Las Tecnologías de la Información (TI) es la utilización de herramientas computacionales para la manipulación de datos, contribuyen al desarrollo de las organizaciones, convirtiéndose en muchas de ellas en un factor clave para el logro de las ventajas competitivas. El mercado de la TI representa USD 2,23 billones en todo el mundo. De estos, aproximadamente el 50% se refiere a software y servicios de TI. La productividad y la calidad son fundamentales para el crecimiento de este sector [1]. Las TI están disponibles para cualquier compañía. Por lo tanto, la diferencia entre el éxito y fracaso del uso de las TI no depende mucho de las características de estas, sino de la habilidad de los administradores en usarla estratégicamente [2].

Un servicio es una manera de proveer valor a los clientes facilitando resultados que quieren obtener evitando asumir costos y riesgos específicos. La gestión de servicios se ha convertido en la actualidad en una de las ventajas competitivas más poderosas que una empresa puede tener, por el simple hecho de que un buen servicio puede acercar, adquirir y retener clientes fácilmente. Un Servicio de TI se compone de una combinación de personas, procesos y tecnología y debería estar definido en un acuerdo de Nivel de servicio [3].

Gestión de Servicios de las Tecnologías de la Información (GSTI) es la capacidad organizativa mediante procesos, métodos, funciones, roles y actividades, que abarca desde la primera idea, hasta que se ofrece y entrega un servicio a los clientes. Es importante decir que una adecuada gestión proporciona múltiples beneficios como aumento de la calidad del servicio, reducción de costos, uso productivo de las habilidades del personal y atención a clientes oportunamente [3].

Gestión de la configuración del software (GCS) es una disciplina para controlar la evolución de los sistemas de software. En la actualidad con la introducción de las metodologías ágiles el software tiene la posibilidad de cambiar de acuerdo a las nuevas necesidades del cliente. Por lo tanto, los cambios en el software pueden ocurrir en cualquier momento por lo que debemos estar preparados [4]. Las actividades de GCS sirven para:

- Identificar el cambio de nuestro software.
- Controlar ese cambio.
- Garantizar que el cambio quede bien implantado.
- Informar el cambio [4].

Los modelos de GSTI son una colección de buenas prácticas que sus creadores han recogido como experiencias obtenidas en las organizaciones. Estos modelos intentan ayudar a las organizaciones sobre cómo pueden gestionar, de manera eficaz, los servicios de TI que brindan o que utilizan [5].



El objetivo del trabajo es contrastar los modelos de GSTI y analizar cuál tiene un mejor conjunto de mejores prácticas para la gestión de la configuración y el cambio. A continuación, se describen algunos de los modelos más usados que son: CMMI-SVC, ITIL y COBIT. Luego se hará una comparación de los 3 modelos enfocándose en la gestión de la configuración y el cambio.

## MODELOS DE GESTIÓN DE SERVICIOS DE LAS TECNOLOGÍAS DE LA INFORMACIÓN

Existen muchos modelos que se dedican a la gestión de los servicios de tecnologías de la información. En el presente trabajo se consideraron analizar CMMI-SVC, ITIL y COBIT.

### CMMI

Integración de sistemas modelos de madurez de capacidades (CMMI) es un modelo de buenas prácticas desarrollado con el objetivo de ayudar a las organizaciones en el manejo de sus procesos [5]. CMMI está conformado por 22 áreas que abarca el desarrollo de software e ingeniería de sistemas. CMMI de entre los beneficios que ofrece su utilización podemos encontrar los siguientes:

- Mejora la calidad del producto.
- Mayor rapidez de respuesta.
- Mejora la productividad.
- Menor costo de desarrollo.
- Logra satisfacer al cliente [6].

### CMMI para servicios

CMMI para servicios o CMMI-SVC, es una colección de buenas prácticas dirigidas a las organizaciones que se especializan en el préstamo de servicios. CMMI-SVC es el conjunto de buenas prácticas sobre servicios de gobierno y la industria, generado de la Marco de CMMI, se encuentra dirigido a proveedores de servicio, que pueden estar relacionados con los conceptos de capacidad y madurez o que busca mejorar sus procesos de servicios [7].



## ITIL

La biblioteca de Infraestructura de Tecnología de Información (ITIL o Information Technology Infrastructure Library) se concibe como un conjunto de buenas prácticas dirigidas a alcanzar una correcta gestión de los servicios TI.

En ella, se describen detalladamente procedimientos de gestión que servirán para:

- Aumentar la eficiencia de las organizaciones
- Lograr una gestión de la calidad adecuada
- Disminuir los riesgos relacionados con las TI
- Desarrollar conjuntamente los procesos de negocio y la infraestructura de las TI [8].

ITIL ofrece una estrategia que se encamina hacia el sentido de una estrategia de carácter empresarial, pero a esto agrega el componente de TI, buscando con esto integrar la tecnología con el negocio de las organizaciones. Lo idea que trata de plantear ITIL es que los servicios tecnológicos sean implementados y ofertados para apoyar el negocio, como también plantear objetivos teniendo en consideración el futuro y considerando las posibles mejoras en la infraestructura de TI en las cuales invertir [9].

## COBIT

COBIT es un marco de trabajo para la gobernabilidad y la gestión de las TI en una organización. Está compuesto de prácticas y principios referidos a la gestión y gobierno de las TI de una empresa. Integra prácticas de ITIL y normas ISO relacionadas a su objetivo [10].

## COMPARACIÓN DE LOS MODELOS SGTI

Los modelos de gestión CMMI, COBIT e ITIL; son marcos de trabajo con el objetivo de ofrecer buenas prácticas a la hora de ofrecer servicios de TI por parte de las empresas, si bien TI por parte de las empresas, si bien tienen un mismo objetivo cada uno cuenta con proceso o conjunto de tareas que podrían ser iguales, como también diferentes como se muestra en la Tabla 1.



Tabla 1. Comparativa de las características entre CMMI, COBIT e ITIL [10].

Características	CMMI	COBIT	ITIL
Modelo enfocado a Procesos en la Operación del Servicio	Sí	Sí	Sí
Modelo enfocado a Proyectos	Sí	Sí	No Aplica
Modelo enfocado a Gestión de Procesos	Sí	Sí	Sí
Modelo enfocado al Desarrollo	Sí	Sí	No Aplica
Modelo enfocado a la Infraestructura	No Aplica	No Aplica	Sí
El modelo abarca todo el ciclo del producto	Sí	Sí	Sí
El modelo abarca la Gestión de Cambios	Sí	Sí	Sí
El modelo abarca la Gestión de Incidencias	Sí	Sí	Sí
Se definen métricas en procesos	Sí	Sí	Sí
Se define operativa y actividades en los procesos	No	No Aplica	Sí
El modelo permite el seguimiento de las actividades	Sí	Sí	Sí
El modelo tiene entre sus objetivos la mejora continua	Sí	Sí	Sí
El modelo permite la certificación de la organización	No Aplica	No Aplica	No Aplica
El modelo es compatible con la ISO 9001	Sí	Sí	Sí
El modelo es compatible con la ISO 20000	Sí	Sí	Sí

## CONCLUSIONES

La utilización de modelos de GSTI como ITIL, COBIT Y CMMI-SVC son buenas prácticas para la gestión de servicios de tecnología de la información. En la literatura y documentación de empresas que han utilizado estos modelos no han utilizado todo el proceso en general para sus distintas áreas, sino que han utilizado solo algunos procesos que se adapten a la empresa o las han modificado el proceso de acuerdo a las necesidades del proyecto a realizar. Se pueden combinar dichos modelos y el más utilizado y conocido es ITIL. Existen aplicaciones de software para la gestión de servicios de TI que se basan en el modelo ITIL como Opsgenie, Freshservice, Invgate. Es posible cambiar de COBIT a ITIL, COBIT describe los principios que soportan a una organización orientada a las necesidades corporativas mientras ITIL se enfoca en la generación de valor para el negocio. En el caso de CMMI con ITIL estos modelos son complementarios. CMMI se centra en garantizar la calidad en el desarrollo de software mientras ITIL garantiza la explotación del producto de software.



## REFERENCIAS

- [1] S. Reinehr and A. Malucelli, "Maturity in IT Service Management: A Longitudinal Study," in *Systems, Software and Services Process Improvement*, Cham, 2019, pp. 99–110, doi: 10.1007/978-3-030-28005-5\_8.
- [2] D. A. A. Wazzan, "A Framework for the Strategic Management of Information Technology," *AJBM*, vol. 7, no. 1, Art. no. 1, Apr. 2018, doi: 10.11634/216796061706918.
- [3] A. D. la Cruz and D. Mauricio, "Una Revisión de la Gestión de Servicios de Tecnologías de Información," *Revista de investigación de Sistemas e Informática*, vol. 4, no. 1, Art. no. 1, 2007, Accessed: Jul. 29, 2020. [Online]. Available: <https://revistasinvestigacion.unmsm.edu.pe/index.php/sistem/article/view/5076>.
- [4] S. Paredes, C. Hinojosa and J. Ruiz, "La importancia de la Gestión de la Configuración del Software, en una Empresa de Desarrollo," *GEEKS DECC-REPORTS*, vol. 1, no. 3, 2011.
- [5] K. A. Melendez-Llave and A. E. Dávila-Ramón, "Problemas en la adopción de modelos de gestión de servicios de tecnologías de información. Una revisión sistemática de la literatura," *DYNA*, vol. 85, no. 204, Art. no. 204, Jan. 2018, doi: 10.15446/dyna.v85n204.57076.
- [6] Equipo del Producto, C. M. M. I. "CMMI® para Servicios, Versión 1.3", Software Engineering Institute, Tech. Rep. CMMI Institute-2013-TR-001, 2013.
- [7] O. Puello, "Modelo de Verificación y Validación Basado en CMMI," *INNOVACIONING*, vol. 1, no. 1, Art. no. 1, Jan. 2013, doi: 10.17081/invinno.1.1.2068.
- [8] E. C. Tejada, *Gestión de servicios en el sistema informático*. IFCT0509. IC Editorial, 2015.
- [9] C. Ward, V. Aggarwal, M. Buco, E. Olsson, and S. Weinberger, "Integrated change and configuration management," *IBM systems Journal*, vol. 46, no. 3, pp. 459–478, 2007.
- [10] A. Hochstein, R. Zarnekow, and W. Brenner, "ITIL as common practice reference model for IT service management: formal assessment and implications for practice," in *2005 IEEE International Conference on e-Technology, e-Commerce and e-Service*, 2005, pp. 704–710.
- [11] A. Cater-Steel, M. Toleman, and W.-G. Tan, "Transforming IT service management-the ITIL impact," 2006.

# MPI vs OpenMP: Un caso de estudio sobre la generación del conjunto de Mandelbrot

*MPI vs OpenMP: A case study on parallel generation of Mandelbrot set*

**Ernesto Soto Gómez**

Universidad de las Ciencias Informáticas

 [esoto@uci.cu](mailto:esoto@uci.cu)

 <https://orcid.org/0000-0001-6521-2221>

RECIBIDO 22/09/2020 • ACEPTADO 23/09/2020 • PUBLICADO 30/09/2020

## RESUMEN

Algunas de las herramientas más populares hoy en día para la programación paralela son Interfaz de Paso de Mensajes y Multiprocesamiento Abierto. Es de interés comparar estas herramientas en la resolución de los mismos tipos de problemas, debido a la utilización de diferentes enfoques en la comunicación entre tareas. Este trabajo tiene como objetivo contribuir a este empeño al ejecutar pruebas en una arquitectura de memoria compartida y centralizada en el caso de problemas con una solución completamente paralela. El caso de estudio seleccionado fue la computación paralela del conjunto de Mandelbrot. Las pruebas se realizaron para diferentes límites de iteración, cantidad de procesadores y variantes de implementación en C++. Los resultados muestran un mejor desempeño en el caso de Multiprocesamiento Abierto.

**Palabras claves:** C++, computación paralela, conjunto de Mandelbrot, MPI, OpenMP.

## ABSTRACT

*Nowadays, some of the most popular tools for parallel programming are Message Passing Interface and Open Multi-Processing. It is of interest to compare these tools in solving the same kind of problems, because of the use of different approaches to inter-task communication. This work attempts to contribute to this goal by running trials in a centralized shared memory architecture in the case of problems with an entirely parallel solution. The selected case study was the parallel computation of Mandelbrot set. Trials were conducted for different iteration limits, processors amount, and C++ implementation variants. The results show better performance in the case of Open Multi-Processing.*

**Keywords:** C++, Mandelbrot set, MPI, OpenMP, parallel computing.



## INTRODUCTION

There are diverse tools for parallel programming. Some of the most popular nowadays are Message Passing Interface (MPI)<sup>1</sup> and Open Multi-Processing (OpenMP)<sup>2</sup>. Both tools are essentially dissimilar because of the use of different approaches to inter-task communication: OpenMP uses shared-memory (tasks are realized by using threads in the same operating system process) [1,2] but MPI uses message-passing (tasks are realized by using a different operating system processes) [3,4]. For this reason, it is of interest to compare these tools in solving the same kind of problems. That is, which is the best in computing the same kind of solution for the same kind of problems taking into account that Do the concerned tools use different inter-task communication mechanisms? This article attempts to contribute to the answer of this question in a centralized shared memory architecture [5] in the case of problems with an entirely parallel solution, that is, a solution with the absolute absence of the need for synchronization –except for the gathering of the partial solutions from several subtasks in order to construct one final solution–.

To accomplish this goal, the parallel generation of the Mandelbrot set has been chosen as an example. This case has been studied in the parallel computing context, usually as a didactic example [1,6,7] because it can be generated from a simple mathematical expression. Also, the Mandelbrot set is a fractal: a figure that possesses a detailed structure in a wide range of scales. Fractal geometrical relations are found in several natural structures, thereby fractals are of great interest to science [8]. This last point adds to the motivation of the study of this example.

The parallel computing of the Mandelbrot set has been already studied in the case of MPI and OpenMP independently from each other [1, 9, 10]. The current work makes comparisons between the straightforward sequential implementation and corresponding parallel versions implemented in MPI and OpenMP with different schedule strategies. C++ has been used as the programming language and the comparisons were made for different iteration limits and number of processors. All generated data as well as all used code may be found in <https://github.com/ESTog/mandelbrotc-/tree/0.1>.

The current document is structured in the following manner. First, the fundamental theoretical elements, the proposed sequential algorithm, and the corresponding parallel versions are exposed. Second, the characteristics of the experiment and the obtained results are described. Last, final remarks are made

---

<sup>1</sup> <https://www.open-mpi.org/>

<sup>2</sup> <https://www.openmp.org/>



## METHODS AND MATERIALS

### Sequential implementation

The Mandelbrot set is the set of all  $c \in \mathbb{C}$  for which the recurrence relation (Equation 1):

$$z_n = z_{n-1}^2 + c \tag{1}$$

does not diverge with  $z_n \in \mathbb{C}$  and  $z_0 = 0$ .

It is known [6,7] that such sequence does not diverge when (Equation 2):

$$|z_n| \leq 2 \tag{2}$$

for all  $z_n \in \mathbb{C}$  where<sup>3</sup> (Equation 3):

$$|z_n| = \sqrt{\Re(z_n)^2 + \Im(z_n)^2} \tag{3}$$

As a way of visualization, the values of  $c$  that are members of the Mandelbrot set may be drawn in the complex plane. Figure 1 shows images of the Mandelbrot set. The images were generated by using the C++ solution developed for this research.

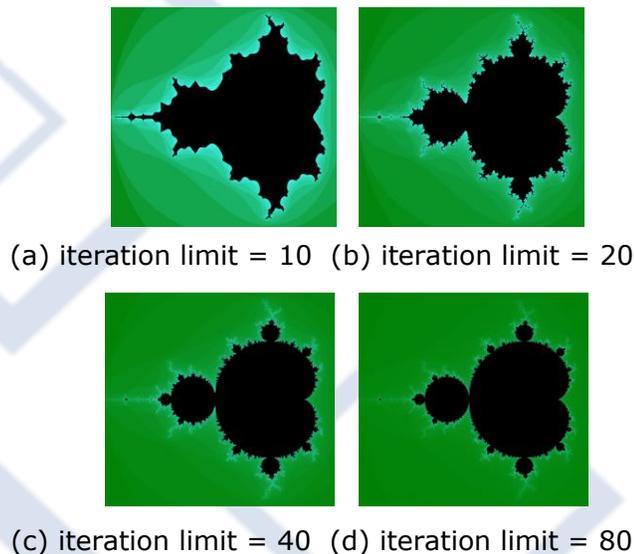


Figure 1. Representation of the Mandelbrot set in the complex plane.

<sup>3</sup>  $\Re(z)$  and  $\Im(z)$  stands for real and imaginary parts, respectively.



A straightforward algorithm that gives an approximation of the Mandelbrot set is to move along a subset of a discrete version of the domain of  $c$  and verify that  $z_n$  does not diverge by using (Equation 2). The generation of the sequence determined by (Equation 1) is made while a given iteration limit is not exceeded [6,7]. A sequential C++ implementation of the mentioned algorithm is given in (Listing 1).

**Listing 1.** Sequential C++ implementation of the straightforward algorithm to approximately compute the Mandelbrot set.

```
1 void compute_mandelbrot_subset(int* result, int iter_limit, int x_resolution,
2                               int y_resolution, double x_begin, double y_begin) {
3     double x_step = (x_end-x_begin) / (x_resolution-1);
4     double y_step = (y_end-y_begin) / (y_resolution-1);
5     int i, j;
6     complex<double> c, z;
7     for (i = 0; i < x_resolution * y_resolution; i++) {
8         c = complex<double>(x_begin + (i % x_resolution) * x_step,
9                             y_begin + (i / x_resolution) * y_step );
10        z = 0; j = 0;
11        while (norm(z) <= 4 && j < iter_limit) { z = z*z + c; j++; }
12        result[i-start] = j;
13    }
14 }
```

Procedure *compute\_mandelbrot\_set* in (Listing 1) receives an array *result* where the computed set will be stored. Although it represents the complex plane, *result* is a unidimensional array. This will allow the implementation of similar parallel versions for MPI and OpenMP even though, in the moment of the visualization of the set in a two-dimensional space, some transformations must be done. The mandelbrot set and its complement are given as an array of integers. Each of these values is the number of iterations before 2 is found true. This is useful when visualizing the mandelbrot set. Figure 1 shows some examples. The images were generated for different iteration limits with a similar procedure<sup>4</sup> to those described in [10] and [7, pp. 103–108]. The iteration limit is given by parameter *iter\_limit*. Parameters *x\_resolution* and *y\_resolution* stand for how big the computed set is, that is, the amount of computed detail. In this case, the length of *result* is the product of *x\_resolution* and *y\_resolution*. Parameters *x\_begin*, *x\_end*, *y\_begin* and *y\_end*

<sup>4</sup> See procedure *print\_result* in [https://github.com/ESTog/mandelbrotc-/tree/0.1/code/common/print\\_result.cpp](https://github.com/ESTog/mandelbrotc-/tree/0.1/code/common/print_result.cpp)



denote the domain of real and imaginary dimensions, respectively. That is, if<sup>5</sup>  $c = x + yi$  then  $x \in [x\_begin, x\_end]$  and  $y \in [y\_begin, y\_end]$ . When visualizing the set, the ranges are usually around  $x \in [-2.5, 1]$  and  $y \in [-1, 1]$ . Variables  $x\_step$  and  $y\_step$  determine the level of discretization of the plane, that is, the width of the steps taken in each dimension. The full C++ sequential implementation may be found in folder `code/mandelbrot_sequential`<sup>6</sup>.

## Parallel implementation

The parallel computing of  $z_n$  may be difficult due to the nonlinear character of (Equation 1). Moreover, if (Equation 1) is expanded the following relations hold (Equation 4 and 5):

$$\text{Re}(z_n) = \Re(z_{n-1})^2 - \Im(z_{n-1})^2 + \Re(c) \quad (4)$$

$$\text{Im}(z_n) = 2\Re(z_{n-1})\Im(z_{n-1}) + \Im(c) \quad (5)$$

May be observed that (Equation 4) and (Equation 5) reference to each other recursively, making more difficult the problem of the parallel computing of  $z_n$ . For these reasons, normally, the parallel computing of the Mandelbrot set is realized by making parallel computations of the iterations. In this case, the plane is divided into parts. In the proposed sequential procedure, *result* a unidimensional array, which means that only one loop must be parallelized.

Implementation in OpenMP is straightforward by using directive *omp for* [1, pp. 53–78]. The fact that a unidimensional array has been chosen to store the solution simplifies the division of its range, making it possible to use the same procedure code in the sequential version as well as in the OpenMP implementation and in each subtask of the MPI implementation. In both parallel versions, because each part is independent among each other, it is not necessary to synchronize the execution of the tasks. The C++ code for this procedure is shown in (Listing 2). Its implementation may be found in file `code/common/compute_mandelbrot_subset.cpp`<sup>7</sup>.

In this case, parameters start and end mark the beginning and the ending of the corresponding part. This will allow using the procedure in each subtask in the MPI implementation. In the sequential version, the procedure is called with *start=0* and *end=x\_resolution\*y\_resolution*. When the procedure is used by the sequential and MPI variants the directives of OpenMP have not effect because the compiler flags for OpenMP are no used. Also, in this general procedure, all the other referenced variables (*x\_resolution*, *x\_begin*, *y\_begin*, *x\_step*, and *y\_step*) are defined as global constants because their values will not change while the execution of the programs. The full C++ implementation using OpenMP may be found in folder `code/mandelbrot_openmp`<sup>8</sup>.

<sup>5</sup>  $i$  denotes the imaginary unit. That is,  $i^2 = -1$ .

<sup>6</sup> [https://github.com/ESTog/mandelbrotc/tree/0.1/code/mandelbrot\\_sequential](https://github.com/ESTog/mandelbrotc/tree/0.1/code/mandelbrot_sequential)

<sup>7</sup> [https://github.com/ESTog/mandelbrotc/tree/0.1/code/common/compute\\_mandelbrot\\_subset.cpp](https://github.com/ESTog/mandelbrotc/tree/0.1/code/common/compute_mandelbrot_subset.cpp)

<sup>8</sup> [https://github.com/ESTog/mandelbrotc/tree/0.1/code/mandelbrot\\_openmp](https://github.com/ESTog/mandelbrotc/tree/0.1/code/mandelbrot_openmp)



---

**Listing 2.** C++ procedure used to compute the Mandelbrot set in the sequential implementation as well as in the parallel ones.

---

```
1 void compute_mandelbrot_subset(int* result, int iter_limit, int start, int end) {
2     int i, j;
3     complex<double> c, z;
4     # pragma omp parallel shared(result, iter_limit, start, end) private(i, j, c, z)
5     # pragma omp for schedule(runtime)
6     complex<double> c, z;
7     for (i = start; i < end; i++) {
8         c = complex<double>(x_begin + (i % x_resolution) * x_step,
9                             y_begin + (i / x_resolution) * y_step );
10        z = 0; j = 0;
11        while (norm(z) <= 4 && j < iter_limit) { z = z*z + c; j++; }
12        result[i-start] = j;
13    }
14 }
```

---

In the case of MPI, partition of the loop has to be done by hand. That is, to follow the master-slave procedure [11]:

1. Divide the range of the array into  $p$  parts, approximately of the same size, where  $p$  is the number of available processors.
2. Compute the  $i$ -th part by using processor  $i$ .
3. Group the result of each partial computation together into one array.

In MPI, two variants may be considered to realize this procedure. One of the variants is to use *MPI\_Send* and *MPI\_Recv* functions to send and receive messages directly between the processors [12,13]. One of the processors, the master, distribute the tasks between the others and group the results together into one array. That processor also computes a part of the whole solution. The C++ code for this processor is shown in (Listing 3). The other processors, the slaves, only receive the indexes that define a part to be computed. After generated, they send the part to the master. The C++ code for these processors are shown in (Listing 4). In the two cases (master and slaves)  $part\_width=result\_size/processors\_amount$ . The full C++ implementation using MPI



with `MPI_Send` and `MPI_Recv` functions may be found in folder `code/mandelbrot_mpi_send_recv`<sup>9</sup>.

The other variant in MPI is to use `MPI_Gather` function which allows gathering the partial computations of each slave into one array [12]. Its use, in this case, is very concise as can be seen in (Listing 5). After space has been reserved for arrays `result` and `partial_result`, only remains to compute the part in each processor –including the master– and then gather this result by using `MPI_gather` function. In this case `part_width=result_size/processors_amount` and `start=current_processor*part_width`. The complete C++ implementation may be found in folder `code/mandelbrot_mpi_gather`<sup>10</sup>.

---

**Listing 3.** C++ code executed by the master in one of the MPI implementation variants.

---

```
1 // distribute tasks
2 int start, end = 0;
3 for (int i = 1; i < processors_amount; i++) {
4     start = end; end += part_width;
5     int message[2] = {start, end};
6     MPI_Send(message, 2, MPI_INT, i, 0, MPI_COMM_WORLD);
7 }
8 compute_mandelbrot_subset(current+end, iter_limit, end, result_size);
9 // join pieces together
10 for (int i = 1; i < processors_amount; i++) {
11     MPI_Recv(current, part_width, MPI_INT, i, 0, MPI_COMM_WORLD, MPI_STATUS_IGNORE);
12     current += part_width;
13 }
```

---

**Listing 4.** C++ code executed by the slaves in the MPI implementation.

---

```
1 int message[2];
2 MPI_Recv(message, 2, MPI_INT, 0, 0, MPI_COMM_WORLD, MPI_STATUS_IGNORE);
3 int* partial_result = new int[part_width];
4 compute_mandelbrot_subset(partial_result, iter_limit, message[0], message[1]);
5 MPI_Send(partial_result, part_width, MPI_INT, 0, 0, MPI_COMM_WORLD);
6 delete[] partial_result;
```

---

<sup>9</sup> [https://github.com/ESTog/mandelbrotc-/tree/0.1/code/mandelbrot\\_mpi\\_send\\_rec](https://github.com/ESTog/mandelbrotc-/tree/0.1/code/mandelbrot_mpi_send_rec)

<sup>10</sup> [https://github.com/ESTog/mandelbrotc-/tree/0.1/code/mandelbrot\\_mpi\\_gather](https://github.com/ESTog/mandelbrotc-/tree/0.1/code/mandelbrot_mpi_gather)



---

**Listing 5.** C++ code for MPI using MPI\_gather function.

---

```
1 compute_mandelbrot_subset(partial_result, iter_limit, start, start+part_width);  
2 MPI_Gather(partial_result, part_width, MPI_INT, result, part_width, MPI_INT, 0, MPI_COMM_WORLD);
```

---

## RESULTS AND DISCUSSION

### Execution environment

The trials consisted in running each implementation for iteration limits 100, 1000, 10000, and 100000 and with one, two, four, and eight processors. In the case of OpenMP the schedule strategies *static*, *dynamic*, and *guided* were considered. The scheduling strategy and the number of processors were passed to the program through environment variables *OMP\_SCHEDULE* and *OMP\_NUM\_THREADS* [12]. Each combination of program, iteration limit, and the number of processors was executed three times and the average of the results were studied by using high-performance computing metrics. Each program was executed in random order with respect to each other, each iteration limit and number of processors in a machine dedicated solely to the running of the trials<sup>11</sup>. Also, the considered resolution –that is, the size of the computed set– was 1024x1024.

The running machine was a computer model *HP Notebook - 15-db0069wm*<sup>12</sup>. In Tables 1 and 2 it is shown relevant information about the running machine and operating system as well as programming and execution tools and libraries, respectively. In file *data/info.txt*<sup>13</sup> may be found information that was automatically recorded at the beginning of the whole experiment by using program *inxi*<sup>14</sup> in *root* mode<sup>15</sup>.

A Python 3 script was developed for the purpose of automatically recording the results to a *csv* file called *data/run\_data.csv*<sup>16</sup> and plotting the data by using Python libraries *pandas* [14] and

---

<sup>11</sup> X graphics and other services like *AppArmor* were deactivated.

<sup>12</sup> <https://support.hp.com/us-en/product/hp-15-db0000-laptop-pc/20395843/model/24094114/document/c06125323>

<sup>13</sup> <https://github.com/ESTog/mandelbrotc-/tree/0.1/data/info.txt>

<sup>14</sup> See Linux man page by using command *man inxi*.

<sup>15</sup> The used command line was *sudo inxi -Fmxxx -t c20 -z -! 31*.

<sup>16</sup> [https://github.com/ESTog/mandelbrotc-/tree/0.1/data/run\\_data.csv](https://github.com/ESTog/mandelbrotc-/tree/0.1/data/run_data.csv)



*seaborn* [14], respectively. These results may be found in folder *data*<sup>17</sup>. The whole Python program may be found in folder *trials\_runner*<sup>18</sup>.

Table 1. Characteristics of the running machine and operating system.

<b>Model</b>	HP Notebook - 15-db0069wm
<b>Microprocessor</b>	AMD Ryzen™ 5 2500U Quad-Core
<b>RAM</b>	8 GB DDR4-2400 SDRAM
<b>Operating System</b>	OpenSUSE Leap 15.1
<b>Type</b>	64 bits
<b>Kernel</b>	4.12.14-lp151.28.40-default

Table 2. Development and execution tools and libraries.

<b>Tools and libraries</b>	<b>Name</b>	<b>Version</b>	<b>OpenSUSE package</b>
C++ compiler	GNU Compiler for C/C++	7.5.0	gcc 7-lp151.3.5
Building system	CMake	3.10.2	cmake 3.10.2-lp151.4.1
MPI development and programming environment	Local Area Multicomputer (LAM)	7.1.4	lam 7.1.4-lp151.2.38
MPI library	OpenMPI	1.10.7	openmpi 1.10.7-lp151.11.4
OpenMP library	GNU Offloading and Multi Processing Runtime Library	8.2.1	libgomp1 8.2.1+r264010-lp151.1.33

<sup>17</sup> <https://github.com/ESTog/mandelbrotc-/tree/0.1/data>

<sup>18</sup> [https://github.com/ESTog/mandelbrotc-/tree/0.1/trials\\_runner](https://github.com/ESTog/mandelbrotc-/tree/0.1/trials_runner)



## Execution time

Although OpenMP and MPI provide specialized functions to measure the execution time of a program [12,15], the execution time was measured by using a method that is valid to all the considered implementations. The function `clock_gettime` and the clock `CLOCK_MONOTONIC_RAW`<sup>19</sup> were used to obtain a monotonic raw hardware-based real-time that cannot be disturbed by system calls. This allowed having a normalized and non-biased way of measuring time. Only the master and main thread execution time were measured in the case of MPI and OpenMP, respectively. Also, in all variants, only the code involved in computing the mandelbrot set was measured. That is, initialization and finalization code, including the allocation and deallocation of `result` array, was not measured. In (Listing 6) is shown the function used to obtain the current time. This function may be found in file `code/common/now.cpp`<sup>20</sup>.

**Listing 6.** C++ function used to obtain current time.

```
1  #include <ctime>
2  #include <cstdlib>
3
4  double now() {
5      struct timespec tp;
6      if (clock_gettime(CLOCK_MONOTONIC_RAW, &tp) != 0) exit(1);
7      return tp.tv_sec + tp.tv_nsec / (double)1000000000;
8  }
```

The obtained execution time is showed in Figure 2. The graphics show how MPI variants have the worst execution time while OpenMP implementation is best when using a dynamic schedule. Also, it is important to notice that the three OpenMP schedules variants behave with different performances. Moreover, in spite of the fact that both use the same basic strategy, OpenMP with a static schedule has better results than the MPI implementations in these trials. This may be due to the fact that each slave has to allocate memory to store the computed part –and deallocate it at the end– and later sent it to the master. This may cause an overhead that is not seen in the OpenMP variants. Finally, it is observed that MPI variant with `MPI_Send` and `MPI_Recv` functions obtained better results than the variant with `MPI_gather` function. This suggests that, in some cases, it is better to use low-level functions than high-level functions to build a concrete solution in order to manifest better performance.

<sup>19</sup> See Linux man page for `clock_gettime(3)` by using command `man 3 clock_gettime`.

<sup>20</sup> <https://github.com/ESTog/mandelbrotc-tree/0.1/code/common/now.cpp>

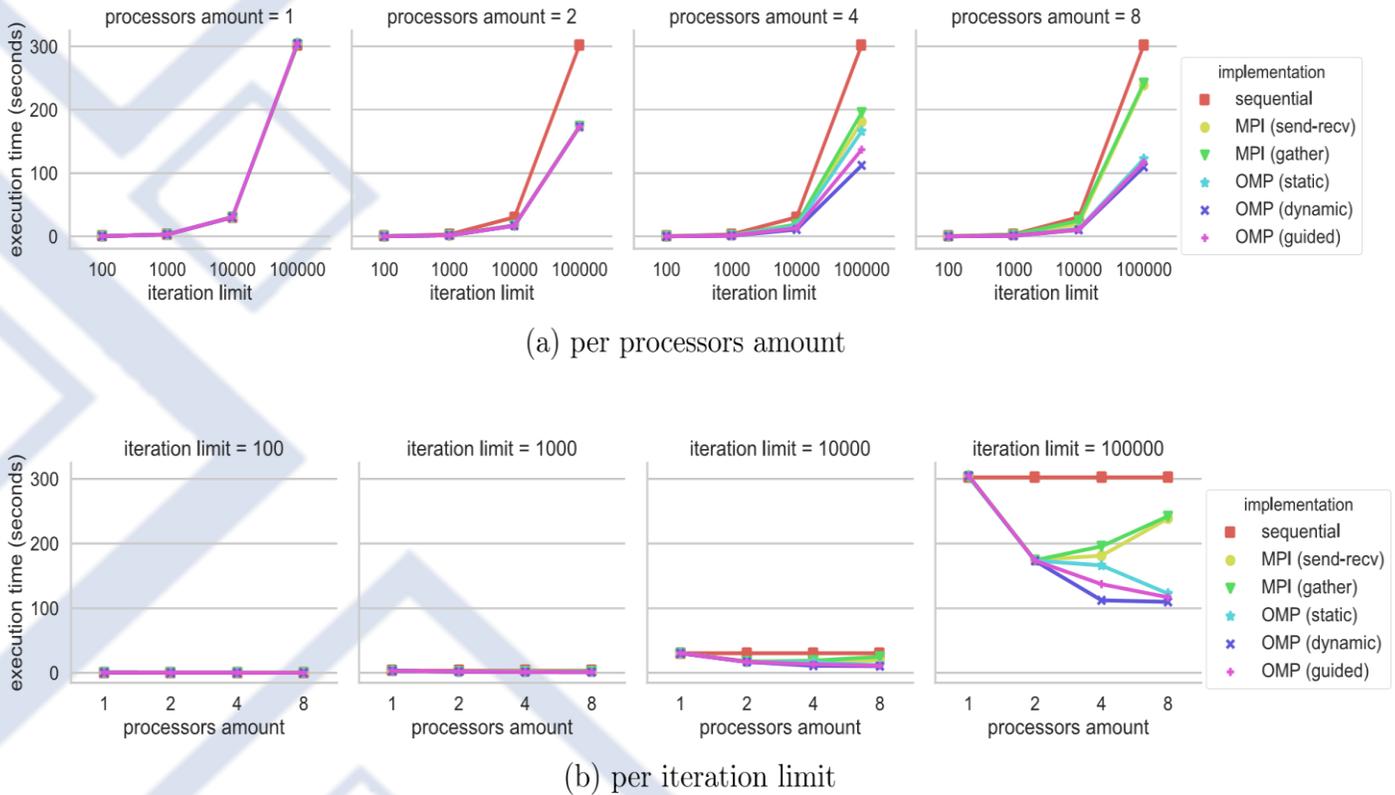


Figure 2. Execution time in seconds. Lower is better.

## Speedup

Speedup is a high-performance computing metric that gives an idea of how much the parallel execution time is better than the sequential execution time. The obtained value is better while closer to the number of available processors. The speedup for  $p$  processors is (Equation 6):

$$S(p) = \frac{t(1)}{t(p)} \tag{6}$$

Here  $t(1)$  is the sequential execution time and  $t(p)$  is the execution time when  $p$  processors are available in the considered parallel alternative [16–18].

The obtained results for speedup are shown in Figure 3. The graphics show in a better manner the performance difference between the variants. Also, it is noticed that the speedup for four and



eight processors do not come near to these values. This suggests that an increase of processors amount will not bring much more improvement to performance in the case of the considered resolution (1024x1024).

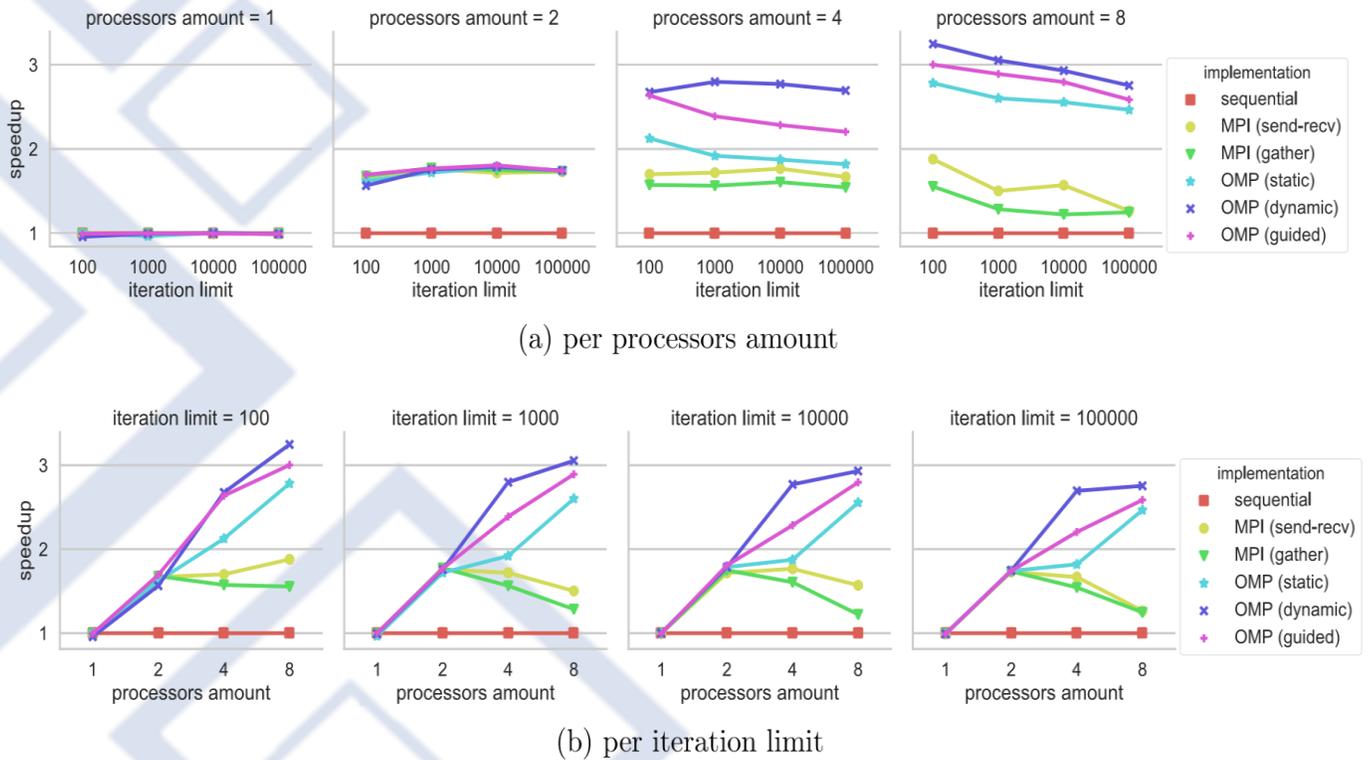


Figure 3. Speedup. Higher is better.

## Parallel efficiency

Parallel efficiency is a high-performance computing metric that gives an idea of how much the speedup is close to the number of available processors, that is, how well the parallel program had used the available computational resources (processors in this case). The best-case scenery is when the speedup equals the number of available processors, meaning that the parallel program had maximum exploitation of the available processing units.

The parallel efficiency for  $p$  processors is (Equation 7) [16–18]:

$$E(p) = \frac{S(p)}{p} \tag{7}$$



The obtained results are shown in Figure 4. The graphics show the decrease of parallel efficiency with the increment of processors amount. The results are consistent in each iteration limit. This reaffirm the idea that an increase of processors amount will not bring better performance, which is more obvious in the case of MPI. In this case, the decrease in efficiency may be due to the fact that the resolution has been taken constant in these trials, and there will be a moment when the parts to compute become too small. This may bring as a consequence that little gain in performance is obtained by computing the parts in a parallel manner because the time that takes to transmit a message is almost the same as the time to compute apart.

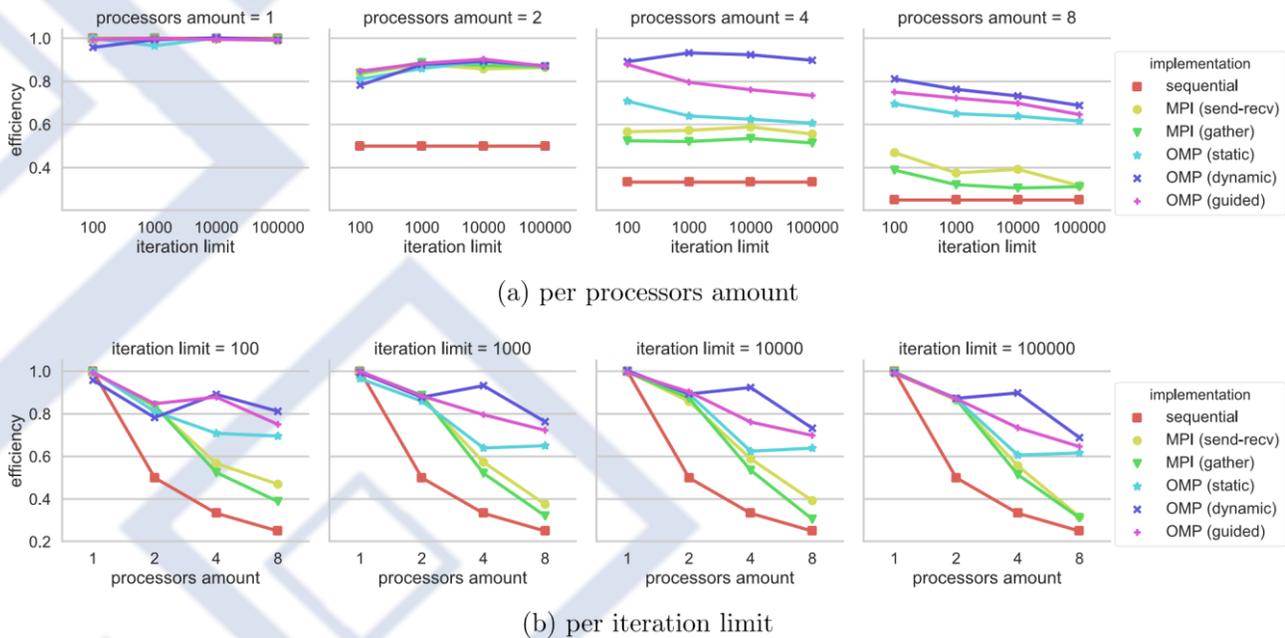


Figure 4. Parallel efficiency. Higher is better.

## CONCLUSIONS

In the present work, a comparison of the parallel generation of Mandelbrot set by using OpenMP and MPI has been conducted. The trials were executed for different iteration limits, the number of processors, and C++ implementation variants. In this case, and in general, OpenMP obtained better performance results than the MPI implementations. It is worth to notice that, although the present work is a case study and for that reason, results should not be taken as conclusive, the conducted trials may contribute to further research and study. Also, running scripts, images, as well as C++ source code is provided to allow reproduction and enhancing of the experiments. Moreover, the current work may be used as a didactic example to the study of the performance of parallel programs.



## REFERENCES

- [1] R. Trobec, B. Slivnik, P. Bulić, and B. Robič, "Programming Multi-core and Shared Memory Multiprocessors Using OpenMP," in *Introduction to Parallel Computing: From Algorithms to Programming on State-of-the-Art Platforms*, ser. Undergraduate Topics in Computer Science, R. Trobec, B. Slivnik, P. Bulić, and B. Robič, Eds. Cham: Springer International Publishing, 2018, pp. 47–86.
- [2] M. J. Quinn, "Shared-Memory Programming," in *Parallel Programming in C with MPI and OpenMP*. McGraw-Hill Education, 2003, pp. 404–435.
- [3] R. Trobec, B. Slivnik, P. Bulić, and B. Robič, "MPI Processes and Messaging," in *Introduction to Parallel Computing: From Algorithms to Programming on State-of-the-Art Platforms*, ser. Undergraduate Topics in Computer Science, R. Trobec, B. Slivnik, P. Bulić, and B. Robič, Eds. Cham: Springer International Publishing, 2018, pp. 87–132.
- [4] M. J. Quinn, "Message-Passing Programming," in *Parallel Programming in C with MPI and OpenMP*. McGraw-Hill Education, 2003, pp. 93–114.
- [5] P. Czarnul, "Generic Taxonomy of Parallel Computing Systems," in *Parallel Programming for Modern High Performance Computing Systems*. Chapman & Hall/CRC, 2018, pp. 11–12.
- [6] M. McCool, J. Reinders, and A. Robison, "Mandelbrot," in *Structured Parallel Programming: Patterns for Efficient Computation*. Morgan Kaufmann, Jun. 2012, pp. 131–143.
- [7] J. M. Stewart, "Two-Dimensional Graphics," in *Python for Scientists*, 2nd ed. Cambridge University Press, 2017, pp. 82–108.
- [8] I. Stewart and A. C. Clarke, "The Nature of Fractal Geometry," in *The Colours of Infinity: The Beauty and Power of Fractals*. Clear Press Ltd, 2004, pp. 2–23.
- [9] M. Tracoli, "Parallel generation of a Mandelbrot set," *VIRT&L-COMM*, Apr. 2016. [Online]. Available: <http://services.chm.unipg.it/ojs/index.php/virtlcomm/article/view/112>
- [10] John Burkardt, "MANDEBRTOT - ASCII Portable Pixel Map (PPM) Image of the Mandelbrot Set," Mar. 2020. [Online]. Available: [https://people.sc.fsu.edu/~jburkardt/cpp\\_src/mandelbrot\\_openmp/mandelbrot\\_openmp.html](https://people.sc.fsu.edu/~jburkardt/cpp_src/mandelbrot_openmp/mandelbrot_openmp.html)
- [11] P. Czarnul, "Master-Slave," in *Parallel Programming for Modern High Performance Computing Systems*. Chapman & Hall/CRC, 2018, pp. 35–39.
- [12] —, "Message Passing Interface (MPI)," in *Parallel Programming for Modern High Performance Computing Systems*. Chapman & Hall/CRC, 2018, pp. 74–102.



[13] M. J. Quinn, "Floyd's Algorithm," in *Parallel Programming in C with MPI and OpenMP*. McGraw-Hill Education, 2003, pp. 137–158.

[14] W. McKinney, "Plotting and Visualization," in *Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython*, 2nd ed. O'Reilly Media, Inc., 2017, pp. 250–283.

[15] P. Czarnul, "OpenMP," in *Parallel Programming for Modern High Performance Computing Systems*. Chapman & Hall/CRC, 2018, pp. 102–118.



# Resolución paralela de sistemas triangulares

27

## *Parallel resolution of triangular systems*

**Ihosvany Rodríguez González**

Centro Nacional de Biopreparados

@ [ihosvany@biocen.cu](mailto:ihosvany@biocen.cu)

id <https://orcid.org/0000-0003-0212-9556>

**Anié Bermudez Peña**

Universidad de las Ciencias Informáticas

@ [abp@uci.cu](mailto:abp@uci.cu)

id <https://orcid.org/0000-0002-1387-7472>

RECIBIDO 04/09/2020 • ACEPTADO 10/09/2020 • PUBLICADO 30/09/2020

### RESUMEN

La resolución de sistemas triangulares es un núcleo computacional ampliamente utilizado en diversas aplicaciones científicas. Esta investigación realiza la implementación y comparación de varios algoritmos paralelos frente a un algoritmo secuencial eficiente para la resolución de sistemas triangulares. Los algoritmos se distinguen por la forma de particionado de la matriz y la asignación a los procesadores. Se realiza el análisis del comportamiento de los algoritmos en la solución de sistemas de ecuaciones lineales triangulares superiores en un clúster de computadoras. Para ello se tienen en cuenta las métricas de tiempo aritmético, tiempo de comunicaciones, aceleración y eficiencia máxima. Se realizaron experimentos para cada algoritmo con distintos tamaños de matrices sobre varios procesadores. El algoritmo con mejores resultados fue el que divide por bloques las filas de la matriz y aplica una distribución cíclica en el clúster.

**Palabras claves:** particionado de matrices, programación paralela, sistema triangular.

### ABSTRACT

*The resolution of triangular systems is a computational nucleus widely used in various scientific applications. This research performs the implementation and comparison of several parallel algorithms against an efficient sequential algorithm for solving triangular systems. The algorithms are distinguished by the way of partitioning the matrix and the allocation to the processors. The analysis of the behavior of the algorithms is performed in the solution of systems of linear superior triangular equations in a cluster of computers. For this, the arithmetic time, communication time, speed-up, and maximum efficiency metrics are taken into account. Experiments were performed for each algorithm with different matrix sizes on various processors. The algorithm with the best results was the one that blocks the rows of the matrix and applies a cyclical distribution in the cluster.*

**Keywords:** matrix partitioning, parallel programming, triangular system.



## INTRODUCCIÓN

La resolución de sistemas triangulares es un núcleo computacional ampliamente utilizado en diversas aplicaciones científicas [1]. Esto se debe a la relación existente entre este tipo de sistemas y el desarrollo de sistemas de ecuaciones genéricos. Generalmente, la resolución de un sistema de ecuaciones genérico se traduce en la resolución de dos sistemas triangulares una vez realizada la triangularización de la matriz [2].

La computación paralela se ha convertido en uno de los pilares más potentes de las computadoras. Para muchos problemas que requieren computación intensiva basta con disponer de un conjunto de computadoras conectadas mediante una red. De esta forma se han difundido los clústeres de procesadores como herramienta típica para la programación paralela [3].

El problema que se aborda en esta investigación trata de la resolución de sistemas triangulares. Para ello se realiza una comparación detallada de varios algoritmos paralelos frente a un determinado algoritmo secuencial eficiente. Los algoritmos paralelos son: Escalar, TRB (Triangular Row Block), TRS (Triangular Row Scatter) y TRBS (Triangular Row Block Scatter). Se realiza el análisis del comportamiento del algoritmo secuencial y de dichos algoritmos paralelos para la solución de sistemas de ecuaciones lineales triangulares superiores en el clúster de computadoras con las siguientes características: 4 HP Proliant DL180 G6, con dos procesadores Quad Core Intel Xeon serie E5520, 2.26 GHz., memoria RAM 24 GB, DDR3 1066 MHz., 2 puertos Gigabit Ethernet, almacenamiento local 60 GB SATA2.

## MARCO TEÓRICO

En un primer momento es necesario mostrar el algoritmo secuencial eficiente que soluciona los sistemas triangulares. Dada la matriz  $U \in \mathbb{R}^{n \times n}$ ,  $u_{ij}=0$  si  $i>j$ , y  $b \in \mathbb{R}^n$ , el algoritmo siguiente resuelve el sistema de ecuaciones  $Ux=b$ .

```
Para  $j=n-1, n-2, \dots, 0$   
     $x_j = b_j/u_{jj}$  //Resolver  
    Para  $i=0, 1, \dots, j-1$   
         $b_i = b_i - u_{ij}x_j$  //Actualizar  
    Fin  
Fin
```

Los diferentes algoritmos paralelos para la resolución de sistemas triangulares superiores se pueden clasificar en función de las distribuciones de datos realizadas [4]. El primer factor a tener en cuenta para realizar la distribución de una matriz (en este caso triangular superior,  $U$ ) es si esta se realiza por filas o por columnas. Es decir, a los diferentes procesadores se les asignará un conjunto de filas de la matriz  $U$ , o bien un conjunto de columnas de esta. En el desarrollo de



este caso de estudio se aplican solo distribuciones por filas, de esta forma se reduce el número de las posibles divisiones de la matriz U.

## Distribuciones de datos

Existen varias formas en que se puede dividir la matriz U por filas, así como asignaciones de dichas particiones a los diferentes procesadores. De la adecuada elección del particionado y la asignación (distribución de datos) de la matriz U dependerá la eficiencia del algoritmo paralelo resultante [5]. Dicha elección debe hacerse atendiendo a dos factores: equilibrado de la carga y reducción del coste de comunicación.

A continuación, se introducen las formas de particionado de la matriz U y posteriormente se enumeran las formas en las que se puede realizar la asignación sobre los diferentes procesadores de la aplicación paralela.

## Particionado de la matriz [6]

El particionado por filas de la matriz U puede ser Escalar o por Bloques. En el particionado por bloques se divide la matriz de n filas en bloques de filas de tamaño s (con  $s \geq 1$ ), mientras que en el particionado escalar se consideran tantas divisiones como filas posee la matriz. Por tanto, el particionado escalar es el particionado por bloques cuando  $s=1$ . En la Figura 1 se muestran los dos tipos de particionado de la matriz U[7].

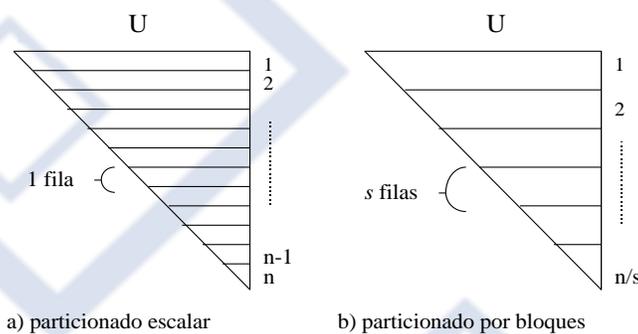


Figura 1. Particionado Escalar y por Bloques de la matriz triangular superior U.

## Asignación a los diferentes procesadores

La asignación de las particiones a los procesadores p, se denomina por Bloques si el número de particiones coincide con el número de procesadores. Mientras que, si el número de particiones es mayor que p, la asignación se realiza de forma Cíclica.

Cuando  $p \ll n$  el número de distribuciones de datos posibles se reduce a tres. Las características de estas tres distribuciones se muestran en la Tabla 1 y la Figura 2.



Tabla 1. Diferentes distribuciones de la matriz triangular.

Algoritmo	Distribución	Particionado	Asignación	Notas
TRB y Escalar	Bloques de filas	Escalar	Bloques	$s=n/p$
TRS	Cíclica por filas	Escalar	Cíclica	$s=1$
TRBS	Cíclica por bloques de filas	Bloques	Cíclica	$1 < s < n/p$

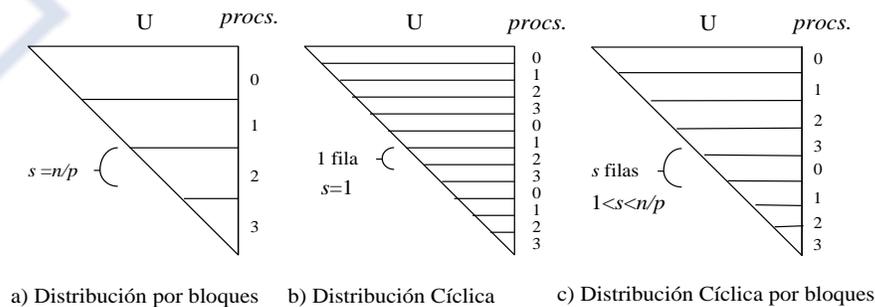


Figura 2. Diferentes distribuciones de la matriz triangular  $U$  sobre cuatro procesadores.

En la Figura 3 se brinda una visión global de un ejemplo de distribución por bloques de los datos en 4 procesadores.

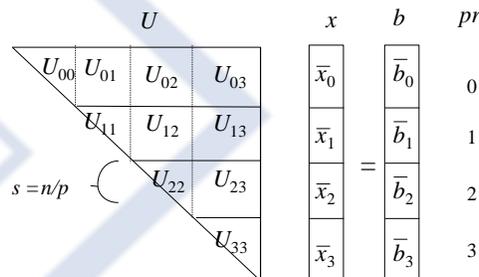


Figura 3. Distribución por bloques de filas del sistema lineal  $Ux=b$  entre  $p=4$  procesadores.

Para completar la distribución de datos del problema de resolver el sistema  $Ux=b$  es necesario comentar que los vectores  $b$  y  $x$  quedan distribuidos de igual forma que  $U$ .

### Algoritmos paralelos implementados

En esta sección se realiza una descripción de los cuatro algoritmos paralelos para la resolución de sistemas triangulares. Al algoritmo paralelo que utiliza la distribución por bloques se le denomina TRB (*Triangular Row Block*), y si la implementación se hace orientada a elementos sin tener en cuenta los bloques se le denomina TRB Escalar. Al algoritmo paralelo que utiliza la



distribución Cíclica se le denomina TRS (*Triangular Row Scatter*) y al que utiliza la distribución Cíclica por Bloques se le denomina TRBS (*Triangular Block Scatter*).

A continuación, se describen los algoritmos atendiendo a una visión global de las variables del problema en los procesadores  $P_{pr}$ .

## Algoritmo Escalar

```
Para  $j=n-1, n-2, \dots, 0$ 
  Si la fila  $j \in P_{pr}$  ( $pr=jDIVs$ )
    Entonces
       $x_j = b_j/u_{jj}$ 
      Difunde  $x_j$  a  $P_0 P_1 \dots P_{pr-1}$ 
    Sino
      Si  $pr < (jDIVs)$  Entonces Recibe  $x_j$ 
  Para  $i=0, 1, \dots, j-1$ 
    Si la fila  $i \in P_{pr}$  ( $pr=iDIVs$ ) Entonces  $b_i = b_i - u_{ij}x_j$ 
  Fin
Fin
```

## Algoritmo TRB

```
Para  $j=p-1, p-2, \dots, 0$ 
  Si el bloque  $j \in P_{pr}$  ( $pr=j$ )
    Entonces
      RESOLVER  $U_{jj}\bar{x}_j = \bar{b}_j$ 
      Difunde  $\bar{x}_j$  a  $P_0 P_1 \dots P_{pr-1}$ 
    Sino
      Si  $pr < j$  Entonces Recibe  $\bar{x}_j$ 
  Para  $i=0, 1, \dots, j-1$ 
    Si el bloque  $i \in P_{pr}$  ( $pr=i$ ) Entonces Actualizar  $\bar{b}_i = \bar{b}_i - U_{ij}\bar{x}_j$ 
  Fin
Fin
```



## Algoritmo TRS

```
Para  $j=n-1, n-2, \dots, 0$   
  Si la fila  $j \in Ppr$  ( $pr=j \text{MOD} p$ )  
    Entonces  
       $x_j = b_j / u_{jj}$   
      Difunde  $x_j$  a  $P_k$  con  $k \neq pr$   
    Sino  
      Recibe  $x_j$   
    Para  $i=0, 1, \dots, j-1$   
      Si la fila  $i \in Ppr$  ( $pr=i \text{MOD} p$ ) Entonces  $b_i = b_i - u_{ij} * x_j$   
    Fin  
  Fin
```

## Algoritmo TRBS

```
Para  $j=n/s-1, n/s-2, \dots, 0$   
  Si el bloque  $j \in Ppr$  ( $pr=j \text{MOD} p$ )  
    Entonces  
      Resolver  $U_{ij} \bar{x}_j = \bar{b}_j$   
      Difunde  $\bar{x}_j$  a  $P_k$  con  $k \neq pr$   
    Sino  
      Recibe  $\bar{x}_j$   
    Para  $i=0, 1, \dots, j-1$   
      Si el bloque  $i \in Ppr$  ( $pr=i \text{MOD} p$ ) Entonces actualizar  $\bar{b}_i = \bar{b}_i - U_{ij} \bar{x}_j$   
    Fin  
  Fin
```

La distribución de datos escogida caracteriza el algoritmo paralelo implementado y, por tanto, las prestaciones del mismo. Todas las distribuciones tienen sus ventajas e inconvenientes; la distribución por Bloques supone un mal equilibrio de carga, mientras que la distribución Cíclica supone un aumento en el tiempo de comunicación [8]. Utilizando la distribución Cíclica por Bloques se persigue equilibrar estos dos factores.



A continuación, en la Tabla 2, se muestra un resumen de las prestaciones de dichos algoritmos, atendiendo al tiempo aritmético ( $t_A$ ), tiempo de comunicaciones ( $t_c$ ), *speed-up* ( $S_p$  o aceleración) y eficiencia máxima ( $E_p$ ).

Tabla 2. Prestaciones comparativas de los diferentes algoritmos paralelos.

Algoritmo	$t_A$	$t_c$	$S_p$	$E_p$
<i>Escalar</i>	$\frac{2n^2 F}{p}$	$n\tau + n\beta$	$\frac{p}{2}$	$\frac{1}{2}$
<i>TRB</i>	$\frac{3n^2 F}{p}$	$n\tau + p\beta$	$\frac{p}{3}$	$\frac{1}{3}$
<i>TRS</i>	$\frac{n^2 F}{p}$	$n\tau + n\beta$	$p$	1
<i>TRBS</i>	$\frac{n^2 F}{p} + nsF$	$n\tau + \frac{n}{s}\beta$	$\frac{p}{1 + \frac{sp}{n}}$	$\frac{1}{1 + \frac{sp}{n}}$

Se resalta el desequilibrio de carga presente en el TRB y el alto costo de las comunicaciones en el TRS y Escalar.

Esto viene dado porque como cada procesador almacena un bloque de tamaño  $n/p$  el de más alto índice calcula su bloque envía su información y no hace nada más y así va pasando con el resto. No sucede lo mismo en el TRS al estar las filas repartidas cíclicamente. Aquí el equilibrio de carga es bueno, pero el tiempo de comunicación es en extremo elevado, ya que es proporcional al tamaño del problema.

## EVALUACIÓN Y RESULTADOS

Se utilizó el lenguaje C para programar los algoritmos. Se implementaron las rutinas para la carga de las matrices, las cuales son guardadas en un archivo con sus respectivos vectores filas separados por tabulaciones. Para calcular el tiempo estimado consumido por cada algoritmo en el cálculo del sistema de ecuaciones se utilizó la función *clock* del lenguaje C definida en la librería *time.h*. Esta función devuelve la cantidad de tiempo de procesamiento que un programa ha utilizado. Es bueno aclarar que en este tiempo no se incluye la carga de los datos, ni el resultado que se muestra por pantalla.

Se utiliza MPI (*Message Passing Interface*) como API que define la sintaxis y semántica de una librería que provee rutinas básicas para construir programas paralelos siguiendo el paradigma de paso de mensaje y comunicación por memoria distribuida. Al ser una API, MPI puede usarse desde cualquier lenguaje de programación (Fortran, C, C++, Java, Python, etc.). Existen bindings



de MPI para varios lenguajes. Su diseño se realizó teniendo en cuenta: eficiencia, potencia de comunicación, portabilidad y uso sencillo de topologías. MPI permite estandarizar: rutinas globales de comunicación (ej. *broadcasting*) y rutinas para realizar cálculos globales (ej. *reduction*).

## Evaluación de la red de interconexión del clúster de computadoras

Se trabajó con un algoritmo que calcula la latencia de la red [9], basado en la técnica *ping-pong* que consiste en la trasmisión de información desde el *front-end* a los nodos y una vez en estos, se trasmite la información desde los nodos al *front-end*.

Para obtener la latencia de red, es decir, el retardo provocado por el medio físico se transmitieron tramas sin información, de forma que al realizar un control de tiempos (tiempo de inicio - tiempo de llegada); se obtenga el retardo provocado por el medio de transmisión utilizado[10].

Después se utilizó un algoritmo que calcula el ancho de banda de la red, basado en la técnica *ping-pong*, pero con tramas de tamaño de información muy grande de forma que se obtenga el ancho de banda absoluto y no relativo. El objetivo en enviar grandes tramas de información sin llegar a sobrepasar la tasa máxima de transferencia es obtener el ancho de banda real del medio de transmisión.

Se realizaron pruebas utilizando un algoritmo que calcula el producto vector-matriz. La finalidad de este algoritmo es observar la ganancia de tiempo en la resolución de dicho algoritmo según se vayan añadiendo nuevos nodos al clúster. Este algoritmo consiste en realizar la multiplicación de un vector por una matriz, de forma que cada nodo del clúster se encargue de realizar la multiplicación del vector por una parte de la matriz. El método que se ha llevado a cabo para partir la matriz ha sido dividir el número de filas entre el número de procesos que se van a ejecutar.

Se demostró que a medida que se utilizan matrices y vectores grandes, el proceso paralelo resulta más conveniente que la resolución de dicho algoritmo por el método secuencial.

La Ecuación 1 muestra el resultado por mínimos cuadrados de la curva obtenida.

$$Y = 8.10^{-8} + 0.0002 \quad (1)$$

De donde se obtienen los valores de latencia ( $\beta$ ) y costo por paquetes ( $t$ ) para la red del clúster utilizado:  $\beta = 2.10^{-4}$  sg,  $t = 8.10^{-8}$  sg.

Se realizaron pruebas con 8 procesadores para tener una mejor respuesta de la paralelización, se trabajó con *cblas* y el *clapack* [11], [12].



## Evaluación de los algoritmos y análisis de las prestaciones

En las pruebas se utilizaron 24 sistemas de ecuaciones lineales con matrices triangulares superiores de la forma  $Ux=b$ , estas matrices cuadradas y los vectores incógnitas "x" fueron generados con números reales y aleatorios entre 1 y 5000 con dimensiones entre 64 y 2048. Además, se ejecutó un programa para calcular los vectores independientes  $b$ . Estas matrices fueron de orden 64, 128, 256, 512, 1024 y 2048. Además, para cada dimensión se realizaron 4 sistemas de ecuaciones.

Se probaron los algoritmos en todas las matrices de pruebas y los tiempos medidos con cada grupo de matrices en cada una de las dimensiones fueron promediados y graficados respectivamente. Se tuvo en cuenta hacer las corridas con 1 procesador para el algoritmo secuencial, mientras que para los algoritmos paralelos se utilizaron 2, 4 y 8 procesadores.

## Análisis del Speed-Up

El análisis teórico del *Speed-Up* o aceleración del algoritmo Escalar muestra que este debe comportarse entre 1 y  $p/2$ , mientras que para el algoritmo TRS es entre 1 y  $p$ . En la Figura 4 se observa que cuando se mantiene constante el número de procesadores y se hace crecer el tamaño de la matriz hay un intervalo en el cual el *Speed-Up* es prácticamente 1, lo que indica que los algoritmos paralelos no tienen ventajas sobre el algoritmo secuencial para matrices de dimensión menor que 256.

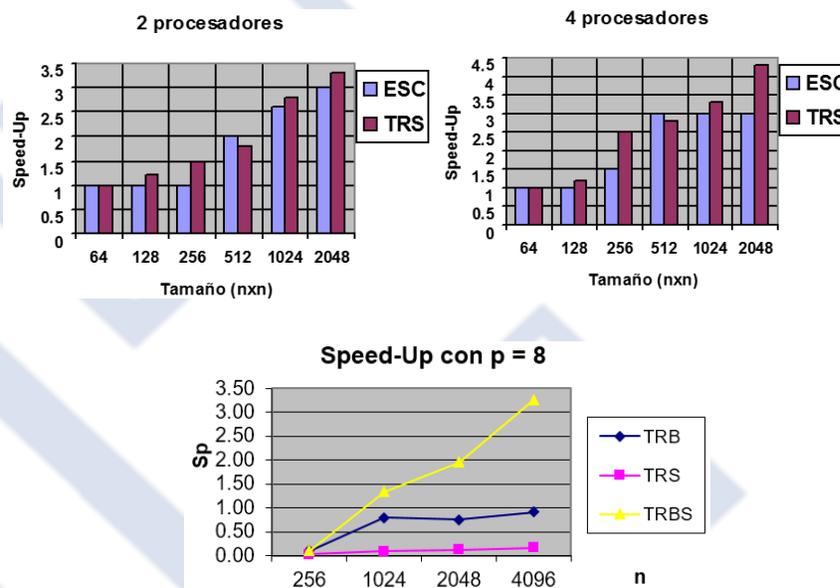


Figura 4. Comportamiento del parámetro Speed-Up para los algoritmos con distinta cantidad de procesadores.



Para problemas de tamaños entre 256 y 2048, el comportamiento de los algoritmos difiere en los casos de 2 y 4 procesadores. En el caso de 2 procesadores, los algoritmos se comportan de la forma esperada, mostrando un incremento del *Speed-Up* al crecer el tamaño del problema de forma tal que los problemas de mayor tamaño son resueltos en menos tiempo por cualquiera de los algoritmos paralelos que por el secuencial. Es bueno señalar que el algoritmo TRS se comporta ligeramente mejor con valores de *Speed-Up* algo superiores al algoritmo Escalar.

Para problemas de tamaño 512 y mayores, el algoritmo Escalar ha alcanzado su mejor valor para el caso de 4 procesadores y supera al caso de 2 procesadores, pero para problemas de 1024 y 2048 las diferencias de *Speed-Up* de este algoritmo son poco significativas cuando se comparan los casos de 2 y 4 procesadores.

Para 4 procesadores, el *Speed-Up* del algoritmo TRS crece a medida que crece el problema, mientras que el algoritmo Escalar a partir de un tamaño se mantiene constante. En la comparación de los algoritmos, para tamaños de problemas creciente con el número de procesadores constantes, el *Speed-Up* se mantiene en la velocidad 1 para tamaños de problema con dimensiones de 64 y 128, tanto con 2 y 4 procesadores. Para tamaño 256, el *Speed-Up* es marcadamente mejor en el algoritmo TRS con 4 procesadores que para el mismo algoritmo con 2 procesadores y que el algoritmo Escalar con 2 y 4 procesadores. El algoritmo TRS para problemas de tamaño 512 y mayores se comporta mejor para el caso de 4 procesadores que para 2.

## Análisis de la eficiencia

Otro aspecto de interés a la hora de evaluar un algoritmo paralelo es su eficiencia [13]. Esta nos aporta el grado de utilización del sistema. El análisis teórico de los algoritmos muestra que la eficiencia del algoritmo Escalar debe estar comprendida entre  $1/p$  y  $1/2$  y para el algoritmo TRS entre  $1/p$  y 1. En la Figura 5 se observa el comportamiento de la eficiencia de los algoritmos, al variar el tamaño del problema con 2 y 4 procesadores. Se parece mucho al análisis del comportamiento del *Speed-Up*[14]. Es notable que los valores de eficiencia calculados para el caso de 4 procesadores se ajustan más a los valores teóricos que los obtenidos con 2 procesadores, porque estos últimos son superiores a los valores esperados.

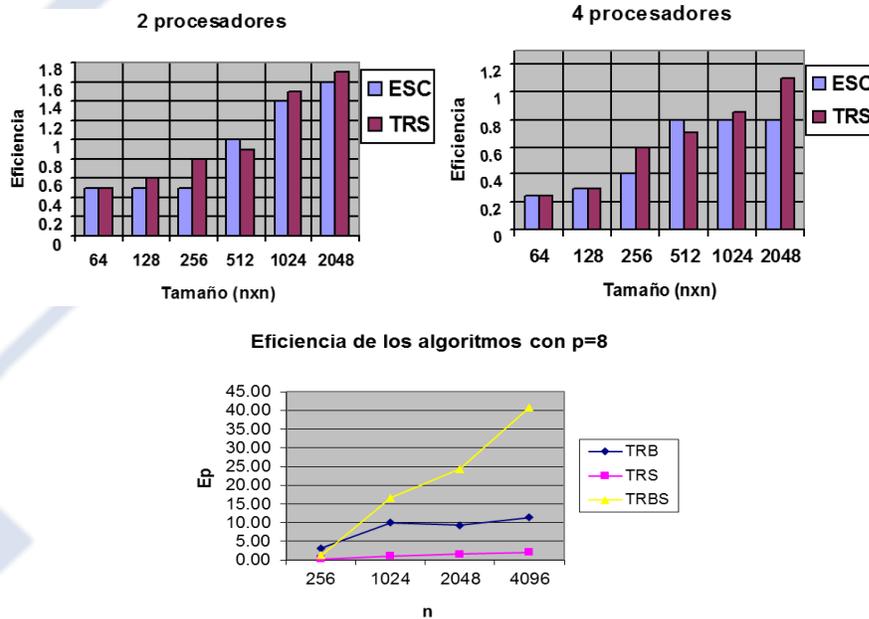


Figura 4. Comportamiento de la eficiencia de los algoritmos.

Al igual que con el  $Sp$ , es el algoritmo TRBS [15] el que alcanza una mayor eficiencia. Análogamente como sucedía al analizar el desempeño de los algoritmos respecto al  $Sp$ , para valores pequeños de  $n$  la eficiencia de estos es baja. Al aumentar el tamaño del problema, el TRBS logra aprovechar mejor los recursos del clúster, no así los otros que se ven lastrados por el alto costo de las comunicaciones.

Finalmente, resalta el desequilibrio de carga presente en el TRB y el alto costo de las comunicaciones en el TRS. Esto viene dado porque como cada procesador almacena un bloque de tamaño  $n/p$  el de más alto índice calcula su bloque, envía su información. No sucede lo mismo con TRS, al estar las filas repartidas cíclicamente. El equilibrio de carga es bueno pero el tiempo de comunicación es en extremo elevado, ya que es proporcional al tamaño del problema.

El TRBS trata de mejorar tanto el desequilibrio como el tiempo de comunicación. En este, el parámetro  $\beta$  está multiplicado por  $n/s$  en lugar de  $n$  como en el TRS y por  $p$  en el TRB. Es de esperar entonces que para valores altos de  $n$  el  $Sp$  tienda a  $p$  sin un aumento tan marcado del tiempo de comunicación, el cual también va a depender del tamaño del bloque. TRBS toma lo mejor del TRS y el TRB, logrando alcanzar un mejor desempeño al mantener la carga balanceada y no muy alto el costo de las comunicaciones.



## CONCLUSIONES

El resolver de forma eficiente un sistema de ecuaciones, utilizando algoritmos paralelos sobre múltiples procesadores, puede aportar grandes ventajas al reducir drásticamente el tiempo necesario para computar la solución.

Como se ha visto con la distribución cíclica por filas y el algoritmo TRS asociado a esta distribución los resultados no son nada halagüeños, mucho menos en un clúster con una red de interconexión como la empleada para realizar los experimentos con un elevado valor de  $\beta$ .

El TRB logra disminuir el costo de las comunicaciones ya que  $n$  no multiplica a  $\beta$ , esto está dado porque cada procesador almacena un bloque de la matriz, pero no es bueno el equilibrio de la carga.

Se aprecia que el algoritmo TRBS, con el aumento de las dimensiones del problema, las comunicaciones no se convierten en un obstáculo y la carga entre los distintos procesadores del sistema se mantenga equilibrada.

Con este trabajo se evidencia la relación existente entre la distribución de los datos y la influencia de esta, tanto en la aceleración como en la eficiencia que logran los algoritmos, así como el equilibrio de carga y reducción de las comunicaciones máximas para lograr el mejor rendimiento.

## REFERENCIAS

- [1] R. Marichal, E. Dufrechou, and P. Ezzatti, "Assessing the solution of one sparse triangular linear system on multi-many core platforms," 2019.
- [2] V. Sonzogni, P. Sanchez, and M. Storti, "Resolución de grandes sistemas de ecuaciones en un cluster de computadoras," *Mecánica Computacional*, vol. 23, pp. 3211–3227, 2004.
- [3] L. Chuquiguanca, E. Malla, F. Ajila, and R. Guamán, "Arquitectura Clúster de Alto Rendimiento Utilizando Herramientas de Software Libre High Performance Cluster Architecture Using Free Software Tools," vol. 2, no. 1, 2015.
- [4] J. L. Bolaño Herazo, "Estudio de rendimiento para la solución de ecuaciones lineales usando computación en paralelo," 2015.
- [5] J. D. Jaramillo, A. M. V. Maciá, and F. J. C. Zabala, "Métodos directos para la solución de sistemas de ecuaciones lineales simétricos, indefinidos, dispersos y de gran dimensión," Universidad Eafit, 2006.
- [6] C. Baeza Sanz, "Explotación de una política de partición de datos para aplicaciones paralelas," 2015.



- [7] S. Piña, "El impacto de shocks contractivos de política monetaria en un modelo DSGE estimado con métodos bayesianos para Chile," 2016.
- [8] H. L. Bodlaender and T. Hagerup, "Parallel algorithms with optimal speedup for bounded treewidth," *SIAM Journal on Computing*, vol. 27, no. 6, pp. 1725–1746, 1998.
- [9] C. Gómez Crespo, "Diseño y evaluación de un cluster HPC: aplicaciones," thesis, Universitat Politècnica de Catalunya, 2014.
- [10] P. E. Leibovich, F. Issouribehere, and J. C. Barbero, "Ensayo y comparación de métodos de transmisión de sincrofasores sobre redes Ethernet," in XVIII Encontro Regional Ibero-Americano do CIGRE (ERAC 2019)(Foz do Iguaçu, Brasil, 19 a 23 de maio de 2019), 2019.
- [11] G. H. A. Salinas and E. M. A. Salinas, "Utilización de CLAPACK para resolver sistemas de ecuaciones lineales mediante paralelismo y optimización de memoria," *UNACIENCIA*, vol. 2, no. 3, pp. 7–7, 2009.
- [12] P. Ezzatti, E. S. Quintana-Ortí, and A. Remón, "Resolución de sistemas triangulares en tarjetas Gráficas (gpu)," *Mecánica Computacional*, vol. 29, no. 30, pp. 3053–3061, 2010.
- [13] M. Hernández, A. A. Del Barrio, and G. Botella, "Clúster de Computación Científica de Bajo Coste y Consumo," 2018.
- [14] J. B. B. Darmawan and S. Mungkasi, "Parallel computations using a cluster of workstations to simulate elasticity problems," in *Journal of Physics: Conference Series*, 2016, vol. 776, p. 012081.
- [15] I. S. Silva, L. O. Luz, R. Nepomuceno, and J. C. dos Santos, "Programação de processadores multi-core: Uma experiência educacional utilizando plataformas didáticas embarcadas em fpga," *International Journal in Computer Architecture Education (IJCAE)*, vol. 3, no. 1, pp. 9–12, 2014.



# Design Thinking en la Planificación de Pruebas de Software

40

## *Design Thinking in Software Testing Planning*

### **Yasiel Pérez Vera**

Universidad Nacional de San Agustín

@ [yperezv@unsa.edu.pe](mailto:yperezv@unsa.edu.pe)

<https://orcid.org/0000-0001-9421-9529>

### **Juan José Gallegos Valdivia**

Universidad Nacional de San Agustín

@ [jgallegosv@unsa.edu.pe](mailto:jgallegosv@unsa.edu.pe)

### **Sandra María Zapata Quentasi**

Universidad Nacional de San Agustín

@ [szapataq@unsa.edu.pe](mailto:szapataq@unsa.edu.pe)

### **Doris Marcela Ccama Yana**

Universidad Nacional de San Agustín

@ [dccamay@unsa.edu.pe](mailto:dccamay@unsa.edu.pe)

<https://orcid.org/0000-0002-4994-4884>

### **Rosa Elvira Choque Apaza**

Universidad Nacional de San Agustín

@ [rchoqueapaz@unsa.edu.pe](mailto:rchoqueapaz@unsa.edu.pe)

RECIBIDO 15/08/2020 • ACEPTADO 04/09/2020 • PUBLICADO 30/09/2020

## RESUMEN

Este artículo presenta la posibilidad de trabajar en el desarrollo de software utilizando no solo metodologías ágiles para los procesos, sino también el uso de *Design Thinking* para la planificación del Plan de Pruebas del Software. Ya que durante el proceso de gestión del software existe la dificultad de comprenderlo, para una de las etapas de la gestión de software, en este caso la planificación de las pruebas del software, el que ejecuta las pruebas necesita saber lo que el usuario requiere, debido a que el producto de software es intangible y difícil de medir, el que realiza las pruebas debe escuchar al usuario final y realizar las pruebas de acuerdo a los requerimientos que se solicita. Para mejorar el proceso de comprensión de lo que el usuario realmente requiere, se utiliza sus pilares como: empatía, colaboración y creación de prototipos para una mayor interpretación, la cual permitirá obtener una actitud asertiva al equipo de pruebas. Por otro lado, tenemos que realizar el plan de pruebas del software, en base a esto se realizó un estudio sobre la posibilidad de utilizar *Design Thinking* para ayudar al proceso de planificación de pruebas de software.

**Palabras claves:** *Design Thinking*, Planificación, Pruebas de Software.



## ABSTRACT

*This article presents the possibility of working on software development using not only agile methodologies for processes but also the use of Design Thinking for the planning of the Software Test Plan. Since during the software management process there is the difficulty of understanding the software, for one of the stages of the software management, in this case, the planning of the software tests, the tester needs to know what the user requires, due to Because the software product is intangible and difficult to measure, the tester must listen to the end-user and perform the tests according to the requirements requested. To improve the process of understanding what the user really requires, its pillars are used as empathy, collaboration, and prototyping for a better interpretation, which will allow obtaining an assertive attitude to the tester. On the other hand, we have to carry out the software test plan, based on this a study was carried out on the possibility of using Design Thinking to help the software test planning process.*

**Keywords:** Design Thinking, Planning, Software Testing.

---

## INTRODUCCIÓN

En los proyectos de desarrollo de software, es común que surjan situaciones que comprometan el éxito o la continuación de los proyectos debido a diferentes factores. Usualmente los problemas surgen debido a demoras, sobrecostos no previstos y mala calidad del producto final. Es responsabilidad de los jefes de proyecto anticiparse a estas situaciones a fin de reducir la probabilidad de ocurrencia, y minimizar el impacto en sus proyectos [1].

*Design Thinking* permite concebir proyectos informáticos para dar soluciones a los problemas del mundo real. Se centra en las personas, considerando multidisciplinariedad, colaboración y concreción de pensamientos y procesos, los cuales son caminos que llevan a soluciones innovadoras su base en la resolución de problemas, desde el punto de vista del usuario y con el apoyo de prototipos [2]. Estudios sugieren que este enfoque utiliza métodos basados en soluciones para explorar valores centrados en el ser humano a lo largo del proceso de diseño de ingeniería. Estos hallazgos se reflejan en muchas aplicaciones del pensamiento de diseño: la creación de prototipos, un método basado en soluciones, a menudo se cita como una forma útil de fomentar la inspiración, la ideación y el aprendizaje organizacional, todos valores centrados en el ser humano [3].

En este artículo se presenta una aplicación de *Design Thinking* en el área de planificación de pruebas de software, ya que este enfoque aplicado a pruebas de software significa desarrollar métodos de prueba innovadores, tipos de prueba y estrategias de prueba, tanto como las pruebas. Esto se puede lograr mediante un cambio fundamental en el pensamiento y la acción. Todos los integrantes del equipo de pruebas de software, desarrolladores y gerentes deben participar en este proceso de cambio ya que práctica tan innovadora requiere empleados de



mente abierta, gerentes y la propia empresa [4]. El objetivo de las pruebas es mejorar y mejorar la calidad del software. Por lo tanto, las pruebas siempre están estrechamente vinculadas a la búsqueda de ideas creativas e innovadoras.

## MATERIALES Y MÉTODOS

### Metodología ágil

Las metodologías ágiles son un formato más contemporáneo para el desarrollo de software donde funciona el compromiso activo del cliente, la comunicación recurrente entre el equipo de desarrollo y la entrega rápida del valor del cliente, de acuerdo con el Manifiesto Ágil [5]. Los métodos ágiles permiten un trabajo de *sprint* corto, que pueden ser características lanzadas en siete o catorce días, por ejemplo, entregar valor al cliente que puede probar durante el período de desarrollo. Sin embargo, debido a que tiene *sprints* cortos, el software falla en algunos puntos, como la falta de documentación o la documentación deficiente del software o software desarrollado que se lanza con cada *sprint*.

Al tener un modelo de comunicación interna y externa muy fácil, ya que las reuniones de lanzamiento y planificación se llevan a cabo al comienzo de cada *sprint*, el cliente siempre tiene fácil acceso al prototipo desarrollado y le permite iterar entre lo que se está haciendo. desarrollado y el usuario final. Por otro lado, el usuario tiene acceso solo cuando se finaliza el prototipo, sin ser escuchado antes de que comience el *sprint*, lo que puede hacer que las fallas, que son predecibles durante el período de desarrollo, sean más frecuentes de lo esperado.

### Design Thinking

Es un modelo mental de desarrollo empresarial innovador que permite enfocar todo el proceso de producción en cómo las personas pueden llegar a lo que se está desarrollando. Aunque es un modelo para el desarrollo empresarial y las ideas de diseño, *Design Thinking* se puede aplicar de muchas maneras dentro de otros procesos, como la gestión de proyectos de ingeniería de software.

El proceso de *Design Thinking* se basa en tres pilares importantes [6]: empatía, colaboración y experimentación. El pilar de la empatía busca colocar a todas las personas que participan en el proceso de desarrollo en lugar de las personas que se verán afectadas por el desarrollo de este proyecto. De esa manera, los participantes del proyecto deben descifrar todo lo que la gente necesita para comprender el proceso, quién debe ser el público objetivo de una solución determinada. El pilar de la colaboración busca comprender cómo cada una de las personas que participan en el proyecto puede ayudar en cualquier momento dado, facilitando el proceso de desarrollo y trabajando eficazmente en equipos para obtener el mejor resultado posible. El pilar de la experimentación propone que cuando se comprende un estudio del requisito y las necesidades de un usuario, los participantes del proyecto deben crear un prototipo que minimice



o resuelva el requisito del usuario, lo ponga en uso y brinde retroalimentación, generando aprendizaje sobre el proceso.

Al usarlo como método de desarrollo, también parte de la premisa de que lo que se desarrollará debe ser innovador, pero más que eso debe ser: deseable, económicamente viable y técnicamente posible [6].

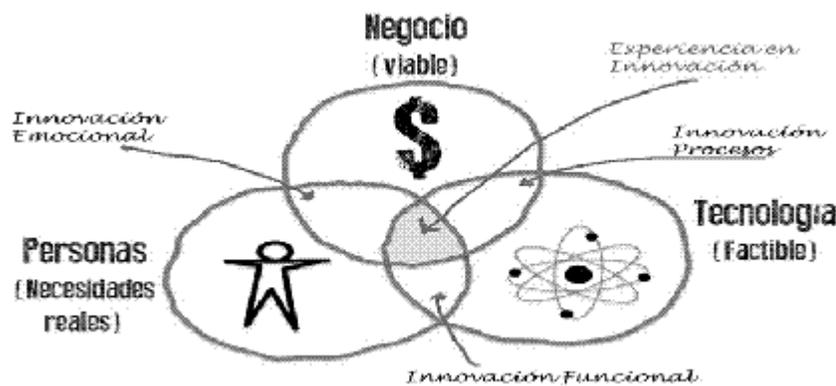


Figura 1. *Design Thinking* [6].

*Design Thinking* funciona con cinco fases básicas para que un proyecto tenga éxito, siendo ellas:

1. **Empatía:** donde la empatía es el punto más importante del proceso. Los actores del proceso deben comprender y diferir en cómo los usuarios sufren un problema particular que debe abordarse. En este momento, se recopilan todos los posibles problemas que el usuario pueda tener con respecto al problema en cuestión.
2. **Definición:** momento de conversión de ideas donde los actores del proceso se ponen a agregar a lo recogido algunas ideas y así entienden cuál debe ser la necesidad del usuario. En esta etapa, también se piensan formas de interpretar la empatía causada por el proceso de descubrimiento y vinculación con la información sobre los requisitos de las personas.
3. **Ideación:** a partir de ese momento, las necesidades de las personas están pensadas para resolver su requerimiento. Es un momento de divergencia en el que todos tienen la oportunidad de hacer observaciones y pensar en cómo podrían ser las soluciones a estos problemas planteados.
4. **Prototipado:** este es un momento para crear un prototipo de la solución y entregarla al usuario para que la pruebe y reciba comentarios sobre sus impresiones de la solución. En este momento no se desarrollan soluciones terminadas, solo prototipos que dan la idea de lo que la gente podría tener como solución a sus problemas en el desafío propuesto. Es tiempo de escuchar e interpretar al usuario, como en la segunda fase.



5. Testeo: La evolución del proceso ocurre al final de un ciclo de empatía, colaboración y experimentación, donde el usuario fue escuchado, pensó en cómo se sentía, observó y habló, interpretó y entregó un prototipo a estos usuarios.

Después de recopilar comentarios en la fase anterior, los participantes del proceso ahora pueden pensar en cómo evolucionar el prototipo en función de lo que se ha escuchado de los usuarios y lo que se debe rediseñar.

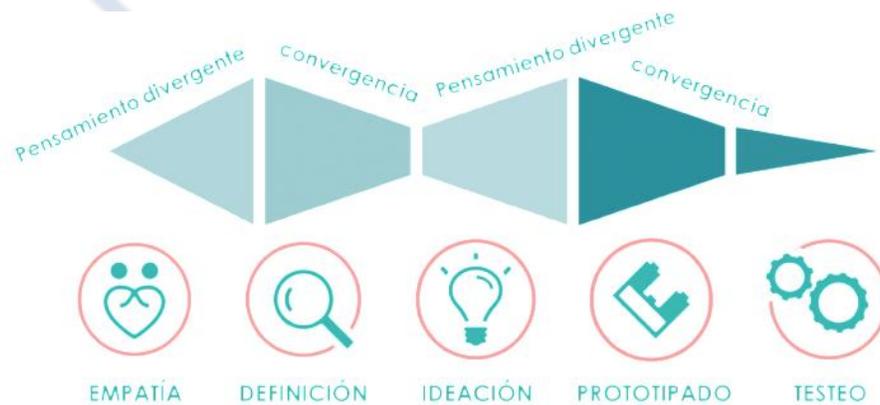


Figura 2. Fases de Design Thinking [6]

Todo el proceso debe repetirse hasta que la solución sea satisfactoria para el usuario. Es importante tener en cuenta que el proceso también puede permitir la evolución de un proyecto *ad-eternum* por el principio muy innovador que tiene *Design Thinking*.

## Métodos

El método elegido para realizar la investigación es el método análisis-sintético. Por lo tanto, se tratará de investigar cómo se desarrolla un plan de pruebas y los pasos a seguir en un *Design Thinking* para luego unificarlos y lograr un resultado óptimo en las pruebas de software. La investigación se realiza mediante la búsqueda de artículos relacionados sobre cómo realizar un plan de pruebas y las actividades que uno debe realizar para llegar a un resultado favorable, las librerías específicas utilizadas son las siguientes [7]–[11], también informándonos sobre la metodología *Design Thinking* el cual se enfoca en la interacción de los individuos de un equipo, así como también las tareas realizadas por cada uno de ellos que pueden ser resultado de un proceso de *Design Thinking*.

Para el entendimiento de la toma de requerimientos de los usuarios, se pueden seguir actividades de creación de interfaces a bajo nivel. La interacción del usuario a bajo nivel permite retroalimentar rápidamente el proceso. De esta forma, se realiza un ejercicio de prueba/falla hasta que el usuario considere que un resultado cubre con sus expectativas. Esto permite acortar el tiempo de desarrollo de interfaces a alto nivel, eliminando la curva de aprendizaje y el coste



de las herramientas necesarias para realizarlo. De esta forma, el usuario puede identificar rápidamente con qué características de diseño está conforme y cuáles no, usando solo el tiempo necesario para llegar a esta conclusión. Las entrevistas con el equipo en donde los temas a tratar serán sobre identificar las funcionalidades nuevas o ya existentes, la estrategia, los entornos y la metodología se realiza en una habitación, que el equipo puede acondicionar a su criterio, en donde se debe incentivar el pensamiento creativo. En este caso, los miembros del equipo se convierten en los usuarios sobre los que hay que aplicar el proceso de *Design Thinking* en cual nos apoyaremos de algunas herramientas como es el mapa mental, entrevista y manualidades creativas. Para la validación y verificación del plan de pruebas el scrum master debe considerar las fases de observación y evaluación.

*Design Thinking* es un proceso iterativo en donde cada paso a seguir será utilizado de manera optimizada, dentro del paso entender el objetivo fundamental es comprender por qué se usan las pruebas de software con mayor precisión y enfatizar cada uno para luego verificar si se han aumentado los riesgos y por qué es importante crear un plan de pruebas también nos sirve para darnos cuenta sobre las vulnerabilidades solucionadas y por solucionar.

En el segundo paso, llamado observar, es donde uno aborda directamente el tema de las pruebas, problemas y riesgos, en este paso se sostienen conversaciones y también se escuchan, donde cada persona explica su propio enfoque y cómo resuelven los problemas durante las pruebas dependiendo al plan desarrollado, en el cual todo tiene que estar detalladamente documentado. El tercer paso llamado punto de vista es donde los involucrados analizan que las pruebas de software necesitan soluciones creativas nuevas e innovadoras con respecto a los problemas y riesgos encontrados donde cada uno da su punto de vista para encontrar la solución más adecuada al problema y documentar correctamente.

El cuarto paso, el paso llamado idear es donde se recopila las ideas sobre las soluciones creativas sin evaluarlas y se usan herramientas en forma de lluvia de ideas, mapa mental con el fin de resolver los problemas y riesgos del software. Ejemplos de posibles estrategias creativas utilizando pensamiento visual para mejorar el resultado de las pruebas de software, como se muestra en la Tabla 1.

Tabla 1. Posibles Estrategias.

Posibles Estrategias	<ul style="list-style-type: none"><li>• Matriz de puntos de contacto</li><li>• <i>Storyboard</i></li><li>• <i>Customer journey</i></li><li>• Aprendizaje de oráculos de pruebas</li></ul>
----------------------	---

El quinto paso llamado prototipo es la posible solución encontrada para resolver el problema que al principio se investiga y se verifica relacionando las clases de pruebas con ideas nuevas, innovadoras, creativas y didácticas como se muestra en la tabla 2.



Tabla 2. Modelo prototipo para cada estrategia.

Estrategias	Ejemplos
Matriz de puntos de contacto	<i>Touchpoint Matrix</i>
<i>Storyboard</i>	<i>Shotbox</i>
<i>Customer journey</i>	<i>Customer Journey Map</i>
Aprendizaje de oráculos de prueba	Migración de software

El último paso, denominado prueba, es donde se usa y se prueba de inmediato en el caso de las pruebas de software, esto podría significar que la solución es exitosa y podría integrarse en las pruebas como una parte esencial fija, o puede verse como una forma de ampliar aún más la idea.

## RESULTADOS Y DISCUSIÓN

Dentro de los resultados se mostrará como los métodos tradicionales de planificación de las pruebas de software pueden ser mejorados poniendo en práctica *Design Thinking* mediante el uso de metodologías ágiles. En este artículo se trabajará con las 5 etapas del plan de pruebas propuestos en [12], las cuales se detallarán en la Tabla 3 con sus principales objetivos.

Tabla 3. Plan de Pruebas de Software.

<b>Plan de Pruebas de Software</b>	
Planeación de pruebas.	<ul style="list-style-type: none"><li>- Tipos de Prueba.</li><li>- Estrategia de Pruebas.</li></ul>
Diseño de pruebas.	<ul style="list-style-type: none"><li>- Análisis de la documentación.</li></ul>
Implementación de pruebas.	<ul style="list-style-type: none"><li>- Creación de datos de prueba.</li><li>- Ejecución de los casos de prueba.</li></ul>
Evaluación de criterios de salida.	<ul style="list-style-type: none"><li>- Finalización del ciclo.</li><li>- Criterios de salida.</li></ul>
Cierre del proceso.	<ul style="list-style-type: none"><li>- Cerrar las incidencias reportadas.</li></ul>



Para medir el impacto que tendrá el uso de *Design Thinking* en la planificación de pruebas se tomará las métricas según el ISO 9126 que nos permitirán obtener el grado de satisfacción de los usuarios. La Tabla 4 nos muestra las métricas que estamos tomando en cuenta y el aspecto que estamos evaluando.

Tabla 4. Métricas a evaluar.

<b>Métrica</b>	<b>Aspecto a evaluar</b>
Efectividad de la prueba.	Defectos reportados por las pruebas.
Esfuerzo.	Tiempo promedio de resolución de defectos. Cantidad de defectos reportados.
Sobre esfuerzo.	Cantidad de casos de prueba escritos. Cantidad de defectos reportados.
Cobertura de requisitos.	Requisitos funcionales y no funcionales totales.
Cobertura de riesgos.	Cantidad de riesgos detectados.
Efectividad de la prueba.	Cantidad de defectos detectados.

Para la incorporación de *Design Thinking* en la planificación de las pruebas de software se tomó en cuenta las etapas del plan de software expuestas y las fases *Design Thinking*. En la Tabla 5 se muestra las etapas del plan de pruebas utilizando las fases de este enfoque.



Tabla 5. Plan de Pruebas de Software con *Design Thinking*.

<b>Plan de Pruebas de Software con <i>Design Thinking</i></b>		
Planeación de pruebas.	<ul style="list-style-type: none"><li>- Tipos de Prueba.</li><li>- Estrategia de Pruebas.</li></ul>	<ul style="list-style-type: none"><li>- Empatía: Análisis de lo que se va a juzgar del producto.</li></ul> Ejemplo: Participación activa con los usuarios finales.
Diseño de pruebas.	<ul style="list-style-type: none"><li>- Análisis de la documentación.</li></ul>	<ul style="list-style-type: none"><li>- Empatía: Entender cómo se comporta el usuario.</li></ul> Ejemplo: Proponer laboratorios de <i>Design Thinking</i> para la discusión de requerimientos hacer probados.
Implementación de pruebas.	<ul style="list-style-type: none"><li>- Creación de datos de prueba.</li><li>- Ejecución de los casos de prueba.</li></ul>	<ul style="list-style-type: none"><li>- Definición: Se toma en cuenta las necesidades del usuario.</li></ul> Ejemplo: Los datos de prueba deben reflejar los datos de los usuarios finales. <ul style="list-style-type: none"><li>- Ideación: Solución y desarrollo de tareas.</li></ul> Ejemplo: Las soluciones esperadas deben ser el reflejo de las necesidades de los usuarios finales.
Evaluación de criterios de salida.	<ul style="list-style-type: none"><li>- Finalización del ciclo.</li><li>- Criterios de salida.</li></ul>	<ul style="list-style-type: none"><li>- Ideación: Resolver problemas.</li></ul> Ejemplo: Las necesidades de los usuarios se vuelven prioridad para ser comparadas con los resultados obtenidos. <ul style="list-style-type: none"><li>- Prototipado: Todas las suposiciones deben ser validadas.</li></ul> Ejemplo: Las salidas de las pruebas deben ser validadas con los usuarios finales.
Cierre del proceso.	<ul style="list-style-type: none"><li>- Cerrar las incidencias reportadas.</li></ul>	<ul style="list-style-type: none"><li>- Testeo: Reunión de todo el proceso. Se decide si es necesario un nuevo ciclo.</li></ul> Ejemplo. El cliente recibe los resultados de las pruebas de acuerdo a las necesidades de los usuarios finales.



Los resultados obtenidos en la aplicación de *Design Thinking* en el plan de pruebas se muestran en la Tabla 6 donde podemos observar que es importante tener una comunicación constante con los usuarios para que los resultados al finalizar el plan sean los ideales.

Tabla 6. Resultados aplicando *Design Thinking*.

<b>Resultados aplicando <i>Design Thinking</i></b>		
<b>Etapas del Plan de Pruebas</b>	<b>Métodos tradicionales</b>	<b><i>Design Thinking</i> con métricas de evaluación</b>
Planeación de pruebas.	Las pruebas son levantadas de acuerdo a las especificaciones obtenidas.	Trato directo con los usuarios finales para la planificación.
Diseño de pruebas.	Análisis de documentación de las especificaciones de los requerimientos.	Reuniones con los usuarios y el cliente para la obtención de las pruebas requeridas.
Implementación de pruebas.	Realizar casos de prueba en base a los datos de la documentación.	Obtención de datos mediante el trato directo con el usuario final.
Evaluación de criterios de salida.	Las salidas reflejan los datos de la documentación.	Las salidas reflejan las necesidades de los usuarios finales.
Cierre del proceso.	No intervienen usuarios finales.	El cliente requiere la opinión de los usuarios finales.

Los resultados demuestran que utilizando el enfoque *Design Thinking* podemos obtener mejores resultados de las pruebas debido a que los usuarios finales son nuestros actores importantes y con los que se interactúa directamente. Según las métricas que se consideran un sobre esfuerzo se reduce debido a que la interacción directa con el usuario permite tener un mejor plan de pruebas, la reducción de riesgos aumenta ya que se cuenta con la opinión del cliente, el empatizar tanto con los clientes y usuarios finales nos permitirá tener cubierto la mayor parte de los casos de uso. La posibilidad de realizar de un nuevo ciclo de plan de pruebas debido a que la primera no cubrió las necesidades del producto queda desestimada debido a que en cada momento se cuenta con la opinión de los usuarios y clientes.



Según [13], donde realiza la aplicación de *Design Thinking* en la ingeniería de requerimientos y la medida de la calidad de software indican que utilizando el enfoque tiene un impacto positivo al igual que en [1] que aplica *Design Thinking* a la gestión y desarrollo de software.

## CONCLUSIONES

Los resultados obtenidos en el artículo demuestran que las metodologías tradicionales pueden ser mejoradas para la obtención de mejores resultados, mediante la aplicación de *Design Thinking* en las etapas del plan de pruebas. Este enfoque todavía es nuevo en comparación con las metodologías de desarrollo anteriores, sin embargo, lo hace adecuado para la ingeniería de software con cierto éxito, ya que puede integrarse con los métodos existentes. El uso de *Design Thinking* se usa normalmente cuando se encuentran involucrados individuos el cual ayuda a mejorar su comprensión en los problemas. *Design Thinking* es dinámico y por lo tanto se puede adaptar y aplicar a cualquier disciplina.

## REFERENCIAS

- [1] J. C. Espinoza Vásquez and E. E. Espinoza Zapata, "Marco de trabajo en base a Design Thinking y metodologías ágiles de desarrollo de software," 2017.
- [2] R. Villarroel, H. Spencer, and R. Muñoz, "Aplicación de design thinking de manera interdisciplinaria en la asignatura de ingeniería de software," 2017.
- [3] M. Greene, "Design Thinking Vs. Systems Thinking for Engineering Design: What's the difference?[WWW Document]," in DS 87-2 Proceedings of the 21st International Conference on Engineering Design (ICED 17), 2017, vol. 2, pp. 21–25.
- [4] Y. Tomita, K. Watanabe, S. Shirasaka, and T. Maeno, "Applying Design Thinking in Systems Engineering Process as an Extended Version of DIKW Model," INCOSE International Symposium, vol. 27, no. 1, pp. 858–870, 2017, doi: 10.1002/j.2334-5837.2017.00398.x.
- [5] K. Beck et al., The agile manifesto. Feb, 2001.
- [6] T. Brown and B. Katz, Change by design: how design thinking transforms organizations and inspires innovation, vol. 20091. HarperBusiness, 2019.
- [7] IEEE, "IEEE Xplore Digital Library," 2019. <http://ieeexplore.ieee.org/Xplore/home.jsp> (accessed Mar. 20, 2017).
- [8] Springer, "Home - Springer," 2019. <http://link.springer.com/> (accessed Mar. 20, 2017).
- [9] SciELO, "SciELO - Scientific Electronic Library Online," 2019. <http://www.scielo.org/php/index.php> (accessed Mar. 20, 2017).



- [10] ScienceDirect, "ScienceDirect." <http://www.sciencedirect.com/> (accessed Jun. 08, 2016).
- [11] Elsevier, "Elsevier Home," 2019. <https://www.elsevier.com/> (accessed Mar. 20, 2017).
- [12] D. E. Soto Durán, A. X. Reyes Gamboa, and J. Jiménez Builes, "Aplicación de la Gestión de Conocimiento al proceso de pruebas de software," *Ing.USBMed*, vol. 8, no. 2, pp. 6–13, Jul. 2017, doi: 10.21500/20275846.2836.
- [13] W. Santos, C. Quarto, and L. Fonseca, "Study about software project management with Design Thinking," in *Proceedings of the Euro American Conference on Telematics and Information Systems*, 2018, pp. 1–4.



# Sistema para la evaluación integral estudiantil

## System for the Integral Student Evaluation

52

### Duriet Aguilera Álvarez

Universidad de las Ciencias Informáticas

@ [duri@uci.cu](mailto:duri@uci.cu)

id <https://orcid.org/0000-0002-6816-3581>

### Alcides Rodríguez Rodríguez

Universidad de las Ciencias Informáticas

@ [alci1109@gmail.com](mailto:alci1109@gmail.com)

id <https://orcid.org/0000-0001-9391-233X>

RECIBIDO 20/09/2020 • ACEPTADO 25/09/2020 • PUBLICADO 30/09/2020

## RESUMEN

La Federación Estudiantil Universitaria en Cuba lleva a cabo un importante número de procesos que requieren reducir el tiempo de procesamiento de la información que se genera, estandarizar la documentación que aportan las diferentes áreas en relación con la organización y elevar la persistencia de los datos. En el presente informe se muestran los resultados de la implementación de la herramienta Sistema para la Evaluación Integral Estudiantil. Esta es una aplicación informática que asegura la calidad del proceso de evaluación estudiantil. Se centra principalmente en el proceso de integralidad y caracterización de los estudiantes, registrando mediante evidencias la trayectoria de cada uno de ellos en un expediente digital. El sistema es desarrollado sobre el marco de trabajo Django, utilizando como lenguaje de programación Python y empleando PostgreSQL como servidor de base de datos. Para guiar todo el proceso de desarrollo del sistema fue empleada la metodología AUP-UCI ajustándose a la soberanía tecnológica por la cual aboga Cuba. Cuenta con diez subsistemas que se integran entre sí y permiten el correcto funcionamiento, integridad y persistencia de los datos que se introducen y procesan. Como parte de la investigación se valida la herramienta a partir de diferentes pruebas al sistema permitiendo examinar el correcto funcionamiento del mismo. Los resultados satisfactorios en la validación del sistema le otorgan validez a la presente investigación, siendo esto la base para favorecer los procesos estudiantiles que tienen lugar en las universidades o sedes del país.

**Palabras claves:** evaluación integral, gestión de información, sistema informático.

## ABSTRACT

*The University Student Federation in Cuba carries out an important number of processes that require reducing the processing time of the information that is generated, standardizing the documentation provided by the different areas in relation to the organization, and increasing the persistence of the data. This report shows the results of the implementation of the Comprehensive Student Assessment System tool. This is a computer application that ensures the quality of the student evaluation process. It focuses mainly on the process of comprehensiveness and characterization of students, recording the trajectory of each of them in a digital file through evidence. The system was developed on the Django framework, using Python as the programming language, using PostgreSQL as the database server. To guide the entire development process of the system, the AUP-UCI methodology was used, adjusting to the technological sovereignty for*



*which Cuba advocates. It has ten subsystems that are integrated with each other and allow the correct operation, integrity, and persistence of the data that is entered and processed. As part of the investigation, the tool is validated from different tests to the system which allowed to test the correct operation of the system and the satisfactory results of the same gave validity to the present investigation, this being the basis for the system to be able to promote student processes that take place in all universities or headquarters in the country.*

**Keywords:** *comprehensive evaluation, computer system, information management.*

---

## INTRODUCCIÓN

Las aceleradas transformaciones científicas y tecnológicas en los sistemas de producción, económico, político y cultural, que caracteriza a la sociedad del siglo XXI, demandan de la educación cambios en el quehacer formativo, y particularmente del nivel superior. Lo cual influye en la formación de la persona, del ciudadano y profesional capaz de adaptarse a los nuevos escenarios sociales y hacer frente a las problemáticas que estos traen consigo. Es por ello que las Instituciones de Educación Superior (IES) del presente siglo, orientan sus esfuerzos en promover la formación integral del estudiante desarrollando un pensamiento analítico, crítico, creativo y propositivo [1].

El Ministerio de Educación Superior (MES) en Cuba tiene entre sus mayores desafíos, el logro de la formación profesional del estudiante en el que se combine el desarrollo intelectual con el sentido de la responsabilidad ética, social, ambiental, disposición y capacidad de gestión para participar de manera activa, crítica y creativa en la etapa histórica concreta en que desarrolla su vida.

Una vía que contribuye a alcanzar ese reto es el proceso de evaluación integral del estudiante al ofrecerle elementos cualitativos y cuantitativos que revelan datos sobre la forma multifacética y armónica de cómo transcurre su desarrollo y los resultados obtenidos. En la misma medida, le aporta mecanismos para auto-reconocer (o que otros reconozcan) tanto las fallas como las potencialidades y oportunidades que lo ayudan a superarlas, al proponerse metas, compromisos, tareas, acciones en aras de incrementar su implicación en el proceso de formación [2].

La Universidad de las Ciencias Informáticas (UCI), como centro docente-productor que forma profesionales altamente calificados y comprometidos en la rama de la informática lleva a cabo el proceso de evaluación estudiantil, denominado Proceso de Integralidad para los estudiantes de primero a cuarto año y Proceso de Caracterización para los de quinto, desarrollando las capacidades, actitudes, valores y potencialidades del futuro profesional. Esto se logra durante todo el periodo de formación en el pregrado a partir del control y evaluación de los estudiantes en el ejercicio de sus actividades.



En cada curso escolar el estudiante debe entregar una autoevaluación del período señalado que es tomada como base por la brigada para otorgarle una evaluación. La forma de realizar este proceso trae consigo que la veracidad de esa información se vea comprometida, reflejando el poco control de la trayectoria estudiantil ya que las actividades no son registradas en el momento en que ocurren.

Según los registros de estos procesos en años anteriores, obtenidos en las actas de los Consejos de la Federación Estudiantil Universitaria (FEU), una cifra cercana al 60% de los estudiantes no entrega su caracterización o lo hace con mala calidad. Lo anterior obliga a los jefes de brigada a recopilar todas las evidencias necesarias para llevar a cabo esta evaluación, ralentizando el envío de la información a organismos superiores y el cierre del proceso.

Al no existir una estandarización en la documentación a almacenar, cada estudiante decide en qué formato enviar sus datos: PDF, Microsoft Office u Open Office/Libre Office, lo que torna engorroso los pasos de unificar y procesar la información. Tampoco se cuenta con un archivo histórico para la consulta de la información de años anteriores.

En estos momentos todo este proceso se realiza de forma manual lo que hace más engorroso el control de esta información, la que se almacena en el mejor de los casos en documento Word. Luego es enviada a niveles superiores a través de correo electrónico, dispositivos de almacenamiento de datos, lo que resulta compleja su manipulación y almacenamiento, afectándose así la agilidad y calidad del proceso. Esto trae consigo la existencia de duplicados o ausencia de la información, dificultades debido a la gestión manual de grandes volúmenes de información y en la confección de la caracterización de los estudiantes.

Lo anterior evidencia la poca fiabilidad en la información que se recopila, comprometiéndose así la integridad, la calidad y la autenticidad de la misma.

Para resolver los problemas antes mencionados, este trabajo propone un Sistema para la Evaluación Integral Estudiantil (SEIE), siendo esto la base para que el sistema pueda favorecer los procesos estudiantiles que tienen lugar en la UCI, así como en todas las universidades o sedes del país.

## **METODOLOGÍA COMPUTACIONAL**

Debido a que la universidad no cuenta con un sistema que le permita gestionar la evaluación integral de los estudiantes vinculados a las diferentes dimensiones educativas, se ha decidido implementar una aplicación que disponga de las funcionalidades fundamentales para garantizar una respuesta adecuada de los procesos que se llevan a cabo.



## Metodología de desarrollo, tecnologías y lenguajes

Para desarrollar un sistema que tenga la calidad y la eficiencia requerida se utilizaron las siguientes tecnologías, metodología y lenguajes para su implementación. Para guiar todo el transcurso de la aplicación se empleó como metodología de desarrollo AUP-UCI haciendo uso del escenario No. 4 [3]. Como herramienta de modelado se utilizó Visual Paradigm en su versión 8.0, como lenguaje de programación Python en su versión 3.5.1, como entorno de desarrollo Pycharm en su versión 2017.2.2, como Sistema Gestor de Bases de Datos PostgreSQL en su versión 9.4.1 y como marco de trabajo Django en su versión 1.11.8.

## Sistemas homólogos en Cuba y el mundo

Actualmente existen diversas soluciones informáticas encargadas de gestionar la información de los procesos estudiantiles que se realizan dentro de unidades educacionales que se extienden por Cuba y el mundo.

### ALEXIA

Desarrollada por Educaria, empresa dedicada a prestar asistencia y servicios informáticos ubicada en Madrid, España, es totalmente web y tiene más de 1.200 centros educativos usuarios. Aporta flexibilidad en todas las áreas de gestión: facturación, impagados, ventas, contabilidad, evaluación, programación de aula, seguimiento del alumno, comunicación, etc. Dispone de cuadro de mando para extraer indicadores a nivel de centro y agrupación. Se integra con G Suite, Office 365, Moodle, y otros aplicativos, y permite incorporar soluciones específicas de horarios, calidad y aprendizaje, con herramientas para que el centro pueda desarrollar su propia estrategia de contenidos. La usan docentes, a quienes se les facilita el acompañamiento de sus estudiantes al poder observar calificaciones con el fin de evaluar rendimientos y elaborar hipótesis sobre la ruta de sus aprendizajes. Datos que pueden cruzar, a su vez, con la asistencia y con las herramientas que le ofrece plataforma de enseñanza-aprendizaje Alexia. A través de la misma los centros escolares pueden gestionar con eficacia y rapidez las tareas administrativas de su secretaría, volcar calificaciones, controlar asistencia y comunicarse con las familias según diversos y numerosos objetivos.

### ÁGORA

Ágora es un producto de software estándar de gran calidad para la Gestión de Centros Docentes y Academias de todo tipo, desarrollado por la empresa española Kherian Soft. Se adapta a cualquier tipo de centro o formación, ya sea de tipo oficial o de carácter libre (academias de enseñanza general, de idiomas, informática, música u oposiciones entre otras), ya se trate de un pequeño centro o una gran empresa de formación con una gestión centralizada de varias sucursales distantes geográficamente. Para ello, la aplicación cuenta con varias versiones y múltiples opciones de organización docente para utilizar en cada caso.



## AKADEMOS

Sistema de Gestión Universitaria de la UCI, conocido como Akademos, es un sistema de planificación empresarial de recursos universitarios que incluye la concepción de varias líneas de desarrollo agrupadas en las áreas de procesos [4]. Fue desarrollado para mejorar la planeación, organización y dirección de los procesos universitarios facilitando su ejecución y control tanto para trabajadores como para estudiantes.

## SIGENU

El Sistema de Gestión de la Nueva Universidad (SIGENU) es un software desarrollado en la Universidad Tecnológica de la Habana "José Antonio Echeverría" (CUJAE) en el 2007 que se perfilaba, en su diseño original, como el resultado de la acción mancomunada de varios IES, en el desarrollo de módulos que automatizarían la gestión de los procesos fundamentales y de apoyo de las universidades: matrícula, proceso docente, postgrado, relaciones internacionales, investigación, etc. No obstante, los esfuerzos realizados por los desarrolladores y la dirección del MES, hasta la actualidad, sólo se ha extendido a todos las IES el módulo de matrícula y promoción [5].

## DATAFEU

Esta es una aplicación informática que asegura la calidad del proceso de evaluación estudiantil y apoya a sus principales dirigentes en el proceso de toma de decisiones. Se centra principalmente en el proceso de integralidad y caracterización de los estudiantes registrando mediante evidencias la trayectoria de cada uno de ellos en un expediente digital.

DataFEU fue desarrollado sobre los marcos de trabajo *Symfony* y *Ext JS* empleando MySQL como servidor de base de datos, ajustándose a la soberanía tecnológica por la cual aboga Cuba. Cuenta con cinco subsistemas que se integran entre sí y permiten el correcto funcionamiento, integridad y persistencia de los datos que se introducen y procesan. Brinda información a sistemas externos mediante servicios web que garantizan la interoperabilidad del mismo.

En la Tabla 1 se muestra un resumen del estudio de las soluciones informáticas para gestionar la información de los procesos estudiantiles.

Se puede concluir que las propuestas anteriores no solventan las nuevas situaciones [6-9]. No cuentan con la capacidad de llevar el control, mantenimiento y gestión de la información relacionada con la evaluación integral y el desempeño de los estudiantes en las diversas dimensiones educativas o presentan una cierta escalabilidad de roles dependiendo de permisos otorgados por niveles estudiantiles lo que resulta complicado y extenso la realización de este proceso para el estudiante. En algunas no se puede acceder al código fuente ya que es privativo



o su interfaz principal necesita ser instalada en cada computadora que se va a utilizar por ser esta una aplicación de escritorio, lo que ocasiona pérdidas de tiempo y limitaciones en su uso.

Tabla 1. Comparación y análisis de soluciones informáticas.

<b>Sistemas homólogos</b>	<b>Ventajas</b>	<b>Limitantes</b>
<b>ALEXIA</b>	Gestiona con eficacia y rapidez las tareas administrativas.	No posee un registro de información referente a la participación de los estudiantes en las diversas dimensiones educativas.
<b>ÁGORA</b>	Centralización de la base de datos.  Generación automática de la documentación.	Carácter propietario, por lo que hay que pagar para su utilización e instalación.
<b>AKADEMOS</b>	Mejora la planeación, organización y dirección de los procesos universitarios.	No gestiona sistemáticamente los datos relacionados a la trayectoria de los estudiantes en las esferas de cultura, deporte y residencia a tener en cuenta en la evaluación integral.
<b>SIGENU</b>	Desarrollo de módulos para la gestión de procesos fundamentales de un centro universitario.	Es una aplicación de escritorio.  Solo implementa los módulos de matrícula y promoción sin tener en cuenta la participación de los estudiantes en las diversas dimensiones educativas.
<b>DATAFEU</b>	Se centra en el proceso de integralidad y caracterización de los estudiantes.  Registra la trayectoria de los estudiantes en un expediente digital.	No está diseñado para la inserción de nuevas carreras y la fusión de facultades.  Presenta cierta escalabilidad en roles, provocando que el estudiante se sienta desmotivado al acceder al sistema.

## Análisis y diseño de la herramienta

Para el desarrollo de la propuesta de solución se tuvieron en cuenta las pautas que define la FEU UCI para el proceso de integralidad estudiantil. Los estudiantes son autoevaluados y se tiene en cuenta la opinión de sus profesores guías en el cumplimiento de los deberes estudiantiles que recogen los estatutos de la organización. Además, si es dirigente estudiantil debe ser evaluado por su desempeño en las funciones que desempeña. En cada uno de los aspectos se otorgarán las evaluaciones siguientes:



- Bien (B).
- Regular (R).
- Mal (M).
- Vanguardia integral por esfera.

## Descripción de la propuesta de solución

El sistema web para la gestión y control de la evaluación integral estudiantil SEIE está compuesto por 10 módulos:

- **Perfil:** permite archivar las actividades en las que el estudiante haya participado y la evaluación asignada teniendo en cuenta las pautas de evaluación que lleva cabo la FEU para emitir una valoración, así como la constancia de los méritos obtenidos en cualquier dimensión educativa, y los cargos a nivel de facultad o universidad que haya ejercido. Posteriormente se realizan los cálculos necesarios atendiendo al cumplimiento de los deberes estudiantiles que recogen los estatutos de la organización. Nunca será evaluado como integral o vanguardia por esfera un estudiante cuyo desempeño docente no sea satisfactorio. Además, SEIE debe permitir al estudiante exportar su perfil en formato PDF.
- **Deporte:** permite gestionar las actividades deportivas que existen en la universidad y en las cuales cada estudiante puede participar. Además, permite archivar los resultados obtenidos en cada actividad.
- **Residencia:** permite gestionar las actividades que se realizan en la residencia estudiantil, así como, los resultados y evaluaciones obtenidas.
- **Cultura:** permite gestionar las actividades culturales que se realizan en la universidad, así como archivar los resultados obtenidos en cada actividad.
- **Investigación:** permite gestionar las actividades investigativas realizadas por el estudiante, así como los resultados obtenidos en cada una de ellas.
- **FEU-UJC:** permite archivar datos sobre los cargos que ha ocupado el estudiante en el Consejo de la FEU o Comité Primario de la Unión de Jóvenes Comunistas (UJC) tanto en la facultad como en la universidad, así como distinciones que haya alcanzado y las evaluaciones adquiridas por cada actividad realizada.
- **Producción:** permite archivar los eventos vinculados a la producción en los que haya participado el estudiante, así como los resultados obtenidos. Además, se tiene en cuenta las cantidades de ausencias al centro al cual pertenece.
- **Trabajo socialmente útil:** permite archivar las evaluaciones obtenidas por el estudiante



durante esta actividad.

- Guardia estudiantil: permite archivar la cantidad de ausencias que ha presentado el estudiante, así como el tipo de ausencia (justificada e injustificada).
- Repositorio: permite archivar los perfiles de cada estudiante ordenados por año académico con la opción de exportar cada perfil en formato PDF.

En la Figura 1 se muestra la pantalla inicial de SEIE.



Figura 1. Interfaz principal del sistema

## RESULTADOS Y DISCUSIÓN

Para la validación del sistema propuesta se realizaron pruebas de carga y estrés. Con el objetivo de saber con un alto nivel de proximidad el soporte del SEIE, teniendo en cuenta que la matrícula de los estudiantes en la Facultad 1 oscila 1000 estudiantes, se desarrollaron las pruebas para un total de 700, 1000 y 3000 usuarios concurrentes. Las pruebas se desarrollaron con el apoyo de la herramienta Apache Meter 2.12, en la que se simuló el entorno donde debe interactuar el sistema para obtener la información más correcta acerca del comportamiento y resultados en general.

Las pruebas realizadas muestran que el sistema es capaz de responder a 1016 peticiones de 700 usuarios conectados simultáneamente en un tiempo promedio de 1433 milisegundos (1.4 segundos aproximadamente) con 0 % de error y una velocidad de carga de 1003 milisegundos (1 segundo aproximadamente), esto evidencia que el sistema puede procesar la carga esperada para esta cantidad de usuarios.



Por otra parte, se realizaron 2400 peticiones iniciadas por 1000 usuarios y en este caso el sistema respondió en 1134 milisegundos (1.1 segundos aproximadamente) como tiempo promedio y una velocidad de carga de 1500 milisegundos (1.5 segundos aproximadamente). Esto demuestra que el sistema es capaz de procesar la carga esperada para este número de usuarios, aunque falló en un 0.04% de las peticiones.

Por último, y con el objetivo de analizar el comportamiento del sistema en condiciones extremas, se realizó una prueba de estrés para un total de 2000 usuarios conectados simultáneamente. En este caso, el sistema responde a las 2900 peticiones en un tiempo promedio de 1012 milisegundos (1 segundo aproximadamente), pero falla el 0.28 % de las peticiones, con una velocidad de carga de 2100 milisegundos (2.1 segundos aproximadamente).

El sistema propuesto permite contribuir a la gestión y control del proceso de evaluación integral estudiantil que realiza la FEU en la UCI. Este es un sistema interactivo que le permite al estudiante archivar todas las actividades realizadas durante cada año académico, además de facilitar exportar en un expediente digital su perfil.

Para validar el SEIE se aplicaron una serie de encuestas anónimas, donde se realizaron preguntas a los usuarios que interactúan con el sistema SEIE. Participaron en la realización un total de 856 estudiantes de todos los años y las facultades, ver Tabla 2, lo que representa aproximadamente un 20% del total de la matrícula actual de la UCI. Las facultades que mayor número de participantes tiene en la encuesta corresponden a la FTE, la 3 y la 1 mientras que los años son 3ro, 4to y 5to.

Tabla 2. Cantidad de personas encuestadas por área y año académico en la UCI.

	<b>Primero</b>	<b>Segundo</b>	<b>Tercero</b>	<b>Cuarto</b>	<b>Quinto</b>	<b>TOTAL</b>
<b>Facultad 1</b>	15	24	46	44	20	149
<b>Facultad 2</b>	9	20	30	27	17	103
<b>Facultad 3</b>	16	26	34	56	35	167
<b>Facultad 4</b>	12	28	35	32	29	136
<b>CITEC</b>	23	25	35	21	14	118
<b>FTE</b>	21	29	32	45	56	183
<b>TOTAL</b>	96	152	212	225	171	856



Los resultados fueron favorables a la utilización del sistema, haciendo énfasis en que este logra una mejor gestión y organización de la información.

## CONCLUSIONES

La elaboración del marco teórico-metodológico que sustenta el desarrollo y utilización de un Sistema de Evaluación Integral Estudiantil ratificó la necesidad de informatizar los procesos que lleva a cabo la FEU que solventen las nuevas situaciones. Actualmente no se conoce a nivel nacional un sistema para la gestión y control de la trayectoria estudiantil en las diversas entidades universitarias del país, el sistema SEIE se encarga de garantizar la fiabilidad en la información que se recopila, así como la integridad, la calidad y la autenticidad de la misma. La aplicación permite reducir el tiempo de procesamiento de la información, estandarizar la documentación y elevar la persistencia de los datos del proceso de evaluación estudiantil.

## REFERENCIAS

- [1] F. Pérez, "La formación integral del estudiante universitario". México, 2017.
- [2] I. Acevedo, A. Acosta, M. Nazco, "La evaluación integral del estudiante universitario". 2017.
- [3] T. Sánchez, "Metodología de desarrollo para la Actividad productiva de la UCI", 2014.
- [4] I. Flores, C. Cruz, V. Miguel, "Una contribución a la gestión de la información de ciencia, tecnología e innovación". Habana, Cuba, 2016.
- [5] A. Ruiz, J. Vidal, "Las TIC en la Gestión Universitaria Cubana: barreras, principios y acciones". Habana, Cuba, 2018.
- [6] M. Campoverde, "Evaluación integral de los docentes universitarios desde varios puntos de vista", 2015.
- [7] Y. Cruz, "Gestión de la Información y del Conocimiento para la toma de decisiones organizacionales", 2015.
- [8] G. Duarte, "Gestión de la Información: precisiones conceptuales a partir de sus orígenes", 2016.
- [9] A. Laguna, "Procedimiento con enfoque multicriterio para la evaluación integralidad emulativa de la FEU", Holguín, Cuba, 2018.



# Aplicación de software gamificada en la asignatura Teleinformática

62

## *Gamified software application in Teleinformatics class*

### **Dargel Veloz Morales**

Universidad de las Ciencias Informáticas

@ [dveloz@uci.cu](mailto:dveloz@uci.cu)

 <https://orcid.org/0000-0002-4231-5831>

### **Laritza González Marrero**

Universidad de las Ciencias Informáticas

@ [lgmarrero@uci.cu](mailto:lgmarrero@uci.cu)

 <https://orcid.org/0000-0002-6128-8496>

RECIBIDO 16/09/2020 • ACEPTADO 25/09/2020 • PUBLICADO 30/09/2020

## **RESUMEN**

Al enfrentarse ante evaluaciones en la asignatura Teleinformática es común que los estudiantes presenten dificultades demostrando el cumplimiento de los objetivos. Las evaluaciones frecuentes se convierten en una herramienta efectiva en manos de los profesores para propiciar la consolidación de conocimientos y el desarrollo de habilidades. Sin embargo, la carga docente y la tendencia al finalismo por parte de los estudiantes provocan poca sistematización en el estudio de la asignatura y por tanto malos resultados en las evaluaciones frecuentes. Teniendo en cuenta el desarrollo tecnológico existente y la vinculación constante de los estudiantes a las TIC se propone en este trabajo una guía para el diseño y aplicación de un instrumento evaluativo para evaluación frecuente combinado con un sistema informático, involucrando así, novedad y motivación en los estudiantes a través de técnicas de gamificación.

**Palabras claves:** enseñanza, evaluación, gamificación, teleinformática.

## **ABSTRACT**

*When facing evaluations in the Teleinformatic subject, it is common for students to present difficulties demonstrating the fulfillment of the objectives. Frequent evaluations become an effective tool in the hands of teachers to promote the consolidation of knowledge and the development of skills. However, the teaching load and the tendency towards finalism in students cause little systematization in the study of the subject and therefore poor results in frequent evaluations. Taking into account the existing technological development and the constant linkage of students to ICT, a guide for the design and application of an evaluative instrument for frequent evaluation combined with a computer system is proposed in this work, thus involving novelty and motivation in students through gamification techniques.*

**Keywords:** evaluation, gamification, teaching, teleinformatic.



## INTRODUCCIÓN

La enseñanza es una de las tareas más nobles de la humanidad, genera satisfacción para quien aprende algo nuevo y también para el que instruye, pero suele ser un reto en muchas ocasiones. Es noble enseñar, sí; también es indispensable. Cada oficio o función que se ejerce hoy o en el futuro depende de la instrucción que se ha recibido en el hogar, en la sociedad o en el ambiente docente.

El uso de las Tecnologías de la Información y las Comunicaciones (TIC) representa una variación notable en la sociedad, lo cual fomenta, de cierto modo, un cambio en la educación, en las relaciones interpersonales y en la forma de difundir y generar conocimientos [1].

La aparición de las innovaciones tecnológicas en el último medio siglo ha tenido un enorme impacto en las posibilidades de aprendizaje, especialmente con la convergencia entre dos prácticas innovadoras, el aprendizaje semipresencial o *blended learning* y la ludificación o *gamification* [2].

El objetivo de la gamificación es afectar directamente los comportamientos y actitudes relevantes para el aprendizaje. A su vez, se plantea la hipótesis de que estos comportamientos y actitudes afectan la relación entre el contenido de instrucción y los resultados del aprendizaje a través de la moderación o la mediación, dependiendo de la naturaleza de los comportamientos y actitudes a los que se dirige la gamificación [3].

Una implicación importante de un proceso de mediación total es que la relación causal entre el antecedente y el resultado no existiría sin el mediador. En la teoría del aprendizaje gamificado, para que los elementos del juego sean efectivos a través del proceso de mediación, (a) los elementos del juego deben causar el comportamiento objetivo y (b) el comportamiento objetivo debe aumentar el aprendizaje. Por ejemplo, si la gamificación creó con éxito una impresión de diversión en los estudiantes, pero esa diversión no afectó el aprendizaje, los elementos del juego finalmente no tendrían ningún efecto en el aprendizaje. Si la diversión afectara el aprendizaje, pero la gamificación no condujera a la diversión, los elementos del juego tampoco tendrían un efecto final en el aprendizaje. Por lo tanto, la gamificación puede no tener éxito en mejorar el aprendizaje si alguna de las dos relaciones causales dentro de la mediación no se cumple: el instructor debe asegurarse de que los elementos del juego conduzcan al comportamiento y también que el comportamiento conduzca al aprendizaje. Si alguno es falso, la gamificación no producirá los resultados esperados. Este enfoque mediacional es la aplicación más común de gamificación [4].

Cuando en términos de educación se habla, no son pocos los aportes al proceso de enseñanza que se implementan en Cuba. Sobre todo, es importante no pasar por alto que el empleo de tecnologías demanda la atención controlada del profesor, utilizando adecuadamente en el proceso las opciones que proporcionan las TIC. Experiencias al respecto hay muchas. "*La integración de*



*las TIC al proceso de formación del profesional de la educación no se reduce a una simple aplicación de los recursos y las tecnologías asociadas a ellas, sino que tiene una implicación más profunda en la concepción didáctica de la dirección de dicho proceso" [5].*

La enseñanza en la educación superior tiene sus propias particularidades. Los que la cursan ya tienen algunos años recorridos en la vida que los dota de conocimientos previos al presente, y al mismo tiempo, se encuentran en la línea definitoria de su formación profesional.

En la Universidad de las Ciencias Informáticas (UCI) se estudia, entre otras carreras, la Ingeniería en Ciencias Informáticas, portando dentro de su programa la asignatura Teleinformática. Esta es muy importante dentro de la carrera, pues introduce a los estudiantes al mundo de las redes; los cuales, a pesar de haber usado internet constantemente, no conocerán de sus bases y especificaciones hasta la llegada de esta asignatura en tercer año de sus estudios universitarios.

El contenido en Teleinformática es bastante extenso, por ende, las evaluaciones frecuentes juegan un rol fundamental en su impartición, no solo porque permiten evaluar a los estudiantes, sino porque colabora en la asimilación de los contenidos, pues su carácter breve y objetivo permite centrar la intención del docente y recibe aceptación dentro del estudiantado, sobre todo por el volumen de conocimiento que se imparte y la necesidad de evaluar cada uno de ellos. Esto último es relevante, puesto que Teleinformática cuenta además con solo dos exámenes parciales y un examen final en 64 horas clases. El uso de esas evaluaciones frecuentes es importante para la asimilación y consolidación del contenido, además de ser evidencia para el docente y el sistema educativo.

Evaluar no tiene ningún sentido si no se ha impartido ningún contenido y existen muchos métodos para ambas cosas. Con el transcurso de los años el empleo de la gamificación ha venido ganando espacio dentro de la educación superior, pues entre sus componentes incorpora la motivación a partir de otros elementos, los cuales no pertenecen a la enseñanza propiamente, pero su aporte va proporcionando resultados en dependencia de su forma de ejecución.

Otro componente que viene aportando frutos en la enseñanza es el empleo y aprovechamiento de las TIC en el Proceso Enseñanza Aprendizaje (PEA). Lo cual ha venido ocupando varias áreas en la enseñanza de la educación superior. Con múltiples casos de éxito dentro de los que se encuentra Moodle, CodecAcademic, Coursera, DuoLingo.

Algunas actividades de Moodle tienen cierta capacidad adaptativa propia, como objeto de aprendizaje independientes del resto de los recursos o actividades de un curso moodle [6].

En el caso de CodecAcademic, es un sistema con más de 5.5 millones de usuarios registrados en el 2014, y un registro de 24 millones de usuarios históricos en el mismo año [7]. CodecAcademic se ha dedicado al fortalecimiento de habilidades en el desarrollo de aplicaciones web mediante una gama de cursos en diferentes lenguajes de programación, a los cuales se puede acceder si



un usuario ingresa en el sistema (usuario registrado en CodecAcademic) y posteriormente inscribirse en los cursos que seleccione a su voluntad.

Una plataforma líder a nivel mundial en el tema de los MOOC es Coursera. Es una empresa educativa establecida en 2011, fundada y promocionada por grandes universidades. Ya en abril de 2013 contaba con el patrocinio de 62 universidades de clase mundial lideradas por Stanford [8].

Sobre DuoLingo existe una característica interesante. En cuanto a las estructuras gramaticales, se ofrecen oraciones descontextualizadas que además integran el vocabulario nuevo y el que se presupone que ya ha sido adquirido. Es decir, se utiliza el conocimiento previo que el alumno tiene para que pueda inferir su significado [9].

El objetivo de este trabajo es diseñar una estrategia para obtener mejores resultados en la asimilación de contenido en la asignatura Teleinformática, a partir de la utilización de una aplicación de software que se combina con la evaluación frecuente y elementos de gamificación.

## **MATERIALES Y MÉTODOS**

Uno de los aspectos más relevantes para que se dé el aprendizaje es la motivación, no hay duda alguna acerca de que cuando esta no existe, los estudiantes difícilmente aprenden [10].

Los juegos favorecen el aprendizaje, ya que cuando en este proceso interviene la diversión aumenta la motivación y se reduce el estrés. Es evidente que cuando se disfruta aprendiendo, se aprende mejor. Además, en los juegos el miedo a cometer errores se reduce y, por tanto, el proceso de aprendizaje se ve favorecido [11].

Los juegos más atractivos, como las grandes obras de ficción, evocan en el jugador emociones que varían en su naturaleza, valencia e intensidad. A pesar de que los practicantes reconocen desde hace mucho tiempo la importancia de las emociones en los juegos, los académicos recién han comenzado a estudiar el complejo flujo de emociones positivas y negativas en el diseño de juegos [12].

Resulta evidente la necesidad de analizar qué características tienen y en qué se basan. La gamificación fue definida como la utilización de dinámicas y elementos de juegos en entornos no lúdicos [13], de esta forma se aprovechan los aspectos motivacionales que son características comunes de juegos en ambientes asociados a temas bien serios, como es el caso de la enseñanza en la educación superior. La gamificación convierte tareas tediosas en atractivas, fomenta la participación del usuario, fortalece vínculos con la estructura social y fideliza a los usuarios [14]. La gamificación no es más que un recurso para mejorar los resultados y convertir el proceso educativo en una acción más eficiente, lo que, indudablemente es algo distinto al juego y al ocio [15].



## Elementos de gamificación

Es necesario entonces identificar cuáles son esos elementos de juegos que proporcionan la motivación que deseamos y en qué consisten; porque más adelante será relevante realizar la selección de elementos a utilizar. Cuando estos elementos de juego aparecen en un ambiente que no tiene relación con el juego se convierten en elementos de gamificación. Algunos de los elementos de gamificación que se utilizan con frecuencia se describen a continuación.

- **Límite de tiempo:** un tiempo especificado para cumplir un conjunto de tareas que tributan a una meta. En la mayoría de las ocasiones es representado como un conteo regresivo que delimita un fracaso cuando llega al valor cero.
- **Puntuación:** cada tarea a realizar debe significar algo que tenga apariencia de premio. Particularmente se representa con valores numéricos que son otorgados cuando la meta se cumple total o parcialmente. Una tarea mal resuelta o pendiente resulta en una puntuación de cero.
- **Tabla de posiciones:** definen un elemento competitivo entre los participantes, quienes hayan hecho más tareas y con mayor calidad habrán obtenido mayor cantidad de puntos. Los nombres de los participantes con mayor puntuación estarán siempre en los mejores lugares de la tabla.
- **Créditos:** se obtienen a medida que se van solucionando dificultades. Son muy parecidos a los puntos, pero a diferencia de ellos no se utilizan para definir lugares. Solamente proveen facilidades para el participante y su empleo dentro del ambiente gamificado es similar al del dinero.
- **Bono:** es un componente dentro del sistema que propicia ventajas para el participante, facilitándole ambientes complejos. Puede tratarse de aumentar puntos, créditos u otorgar habilidades especiales definidas en el sistema.
- **Trofeo:** es un componente dentro del sistema que simboliza el arribo de los participantes a logros definidos en el sistema o metas especiales.
- **Tienda:** es un concepto basado en reglas que aplica de forma particular a cada participante en la mayoría de las ocasiones. Sin embargo, siempre se basa en el lugar donde pueden adquirir bonos u otros beneficios en dependencia de la cantidad de crédito que posean.
- **Guión:** es un elemento de fantasía que se emplea para ambientar todas las tareas involucradas. Una narrativa que ayuda a digerir la temática con mayor facilidad, sin apartar al participante del objeto principal. Pueden ser muy sencillas o complejas en dependencia del tema gamificado.
- **Niveles o desafíos:** conjunto simple de acciones que involucran la resolución de una o más tareas. Permite simplificar el trabajo completo, creando metas parciales y generando



sensación de éxito en mayor número que fracasos.

- **Diálogo o retroalimentación:** permite que el participante conozca qué está aconteciéndole en cada situación relevante en la que se encuentre involucrado.
- **Ambiente interactivo:** define un apoyo al participante, ofreciendo cierta ayuda. Pueden ser recordatorios, un asistente que le ayuda a resolver las tareas, o sugerencias en momentos de estancamiento.

## Aplicación de Software

La aplicación de software empleada desempeña un papel fundamental dentro de la estrategia, pues en ella están incorporados los elementos de gamificación necesarios para el objetivo deseado. También contiene las preguntas a presentar a los estudiantes para ser evaluados en forma de evaluación frecuente.

Todas las preguntas incluidas en la aplicación han sido homologadas. Todas están diseñadas en forma de selección simple o selección múltiple, con niveles de complejidad bajo, medio y alto. Cada pregunta contiene cuatro posibles respuestas, de las cuales el estudiante debe seleccionar.

De los elementos de gamificación antes descritos, la aplicación de software incorpora algunos, los cuales se enuncian y describen a continuación.

- **Niveles:** los mismos están ordenados en mundos y niveles. Para una temática específica de Teleinformática la aplicación define un mundo, el cual contiene varios niveles. Cada nivel contiene varias preguntas para responder.
- **Límite de tiempo:** cada nivel contiene un contador decreciente de tiempo. El valor inicial de tiempo depende de la complejidad del nivel.
- **Puntuación:** cada pregunta a responder otorga puntos si se responden correctamente, acumulando así puntos en cada nivel para cada estudiante.
- **Tabla de posiciones:** se establecen en cada nivel las posiciones de los estudiantes en correspondencia con la cantidad de puntos obtenidos, y en caso de empate, se define el lugar por el que tardó menos tiempo para obtener dichos puntos.
- **Bono:** fueron diseñados en la aplicación como mecanismo de descongestión cognitiva. Durante el desarrollo de la actividad los estudiantes ganan los bonos y podrán ser usados cuando se enfrenten a una pregunta que no puedan responder. Los bonos definidos son: cambiar pregunta, menos 50%, nuevo intento y consultar.
- **Bono cambiar pregunta:** al ser utilizado, la aplicación le busca automáticamente otra pregunta que no estaba concebida antes y elimina la actual, sin penalización alguna.



- Bono menos 50%: de todas las posibles respuestas que brinda la aplicación a la pregunta en cuestión, se eliminan la mitad. Reduciendo así el riesgo de equivocarse.
- Bono nuevo intento: al ser utilizado, la aplicación permite al estudiante volver a responder una pregunta en la cual haya tenido errores, otorgando así una segunda oportunidad para recibir los puntos.
- Bono consultar: permite al estudiante llevar a votación las posibles respuestas que tiene una pregunta. De esa forma sus compañeros de aula le dan su opinión sobre la respuesta, aun así, el estudiante debe escoger su propia respuesta.

En las siguientes figuras se muestran dos pantallas del software desarrollado.

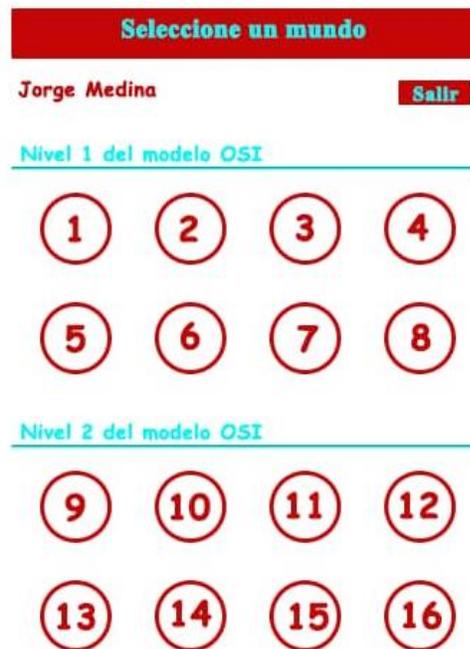


Figura 1. Vista de los mundos existentes.



Figura 2. Vista de un nivel de ejecución con una pregunta.

Se recomienda que antes de iniciar la clase sea planificada debidamente, sin perder de vista el objetivo y demás procesos del PEA. Teniendo en cuenta además el dinamismo y extensión de la clase con la incorporación de la gamificación como factor motivacional y el entorno competitivo proporcionado.

Deben ser empleados niveles del mismo mundo, de esa forma serán abordados conocimientos sobre una misma temática, permitiendo evaluar a todos los estudiantes en el mismo contenido de la asignatura.

El aula debe ser dividida en dos equipos, de esa forma se propicia un mayor número de ganadores de forma subjetiva, sin eliminar la tabla de posiciones individual, pues la evaluación al final de la clase también es individual.

Contrario a lo que pudiera aparentar, el profesor no queda obsoleto en esta actividad. Su papel orientador es fundamental para su desarrollo. Mantiene la energía y competencia en los niveles de disciplina adecuados, presencia el tiempo y reacción de sus estudiantes ante cada pregunta, evita fraudes y eleva el ánimo de los que se equivocan al responder. El profesor siempre podrá medir y analizar elementos que la aplicación de software no hará.



## RESULTADOS Y DISCUSIÓN

Para el desarrollo de la actividad docente se escoge a todo el tercer año de la facultad de Ciencias y Tecnologías Computacionales, de la Universidad de las Ciencias Informáticas. Participando finalmente 74 estudiantes, pertenecientes a tres grupos docentes. Este escenario permite ejecutar la misma clase tres veces frente a diferentes estudiantes en cada ocasión.

La clase comienza recibiendo las orientaciones organizativas del profesor. Como la aplicación de software no incorpora el trabajo en equipo dentro de sus funcionalidades, se hace necesario que el profesor incorpore otros medios educativos (como la pizarra) para suplir este déficit. Así pues, el aula se divide en dos equipos y la aplicación de software está acondicionada, desde el punto de vista académico, con un banco de preguntas pertenecientes a temáticas relacionadas a la capa Física y capa de Enlace de Datos del modelo OSI (contenido de Teleinformática). Todas las preguntas incorporan 4 posibles respuestas, en modalidades de selección simple y selección múltiple. Cada pregunta bien respondida permite acumular 1 punto.

Cada estudiante en la clase recibe por parte de la aplicación de software, lo que esta última define como un mundo de cinco niveles; cada nivel contiene una pregunta, por tanto, el valor del mundo es de cinco puntos. También tendrán habilitados los cuatro comodines de bono explicados en la sección anterior, pero cuando un estudiante en particular consuma un comodín específico, provoca que el profesor deshabilite el mismo al resto de los miembros del equipo, accediendo a las configuraciones de la aplicación.

Luego de los diez minutos iniciales comienza la competición. Los estudiantes de forma ordenada, se van enfrentando a los niveles y explicando al resto del aula el porqué de sus decisiones, permitiéndoles la libertad de declarar cuando obtuvo sus puntos por causa del azar sin penalización alguna. Cuando esto último ocurre, cualquier estudiante del equipo contrario responde con argumentos la razón que permite sea correcta la respuesta. Este comportamiento posibilita que un desconocimiento en un estudiante, se convierta en la recepción de ese conocimiento en el resto del grupo. Añadiendo además que todo queda a la expectativa del dictamen del profesor, pues desde el punto de vista competitivo también genera ventaja en caso de que resulte un empate en cuanto a puntos.

La distribución de estudiantes por equipos y grupos responde a los valores representados en la Tabla 1.



Tabla 1. Cantidad de integrantes por equipo en cada grupo en que se realiza la clase.

	<b>Grupo 1</b>	<b>Grupo 2</b>	<b>Grupo 3</b>
<b>Equipo A</b>	11	11	15
<b>Equipo B</b>	11	11	15
<b>Total de estudiantes</b>	22	22	30

Al terminar la competición, el profesor realiza una sumatoria de todos los puntos acumulados por los integrantes de cada equipo y determina qué porcentaje representa con respecto al total de puntos posibles. Obteniendo así el puntaje por equipos para dictaminar la sentencia, conforme a la motivación de los estudiantes. Sin embargo, para el profesor se obtienen como positivo la reafirmación de conocimiento, la participación activa de los estudiantes, el trabajo en equipo y la evaluación de los estudiantes. El puntaje por equipos se representa en la Figura 3, donde se refieren los equipos, en correspondencia con la Tabla 1, de la siguiente forma: Eq. A-G1, para el caso del equipo A del grupo 1 y así sucesivamente hasta Eq. B-G3 que representa al equipo B del grupo 3.

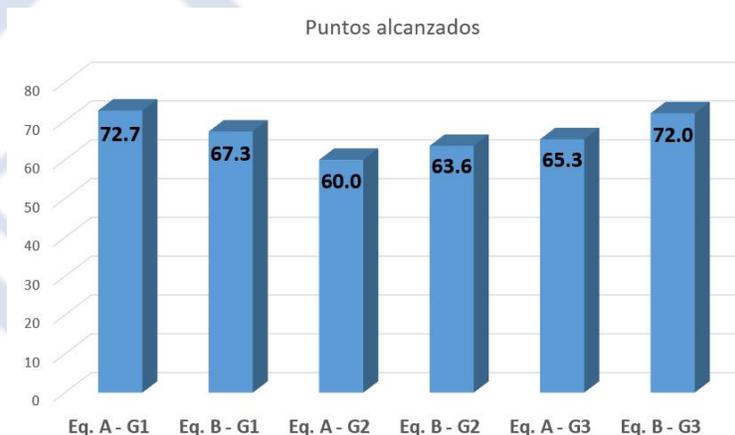


Figura 3. Puntaje obtenido por los equipos al terminar la clase.

Un análisis importante que salta a la vista es el comportamiento de los estudiantes ante la posibilidad de consumo de los comodines. Es evidente que ningún estudiante usaría un comodín si considera que conoce la respuesta correcta. Los equipos que consumen menor número de comodines, están mostrando mayor seguridad al responder, aunque no siempre coincide estar seguro de una respuesta y que sea correcta, esto ocurre en la minoría de los participantes, ver Figura 4.



Los comodines, por su parte, son catalogados por los estudiantes conforme al nivel de ayuda que le proporcionen, así que se escuchan comparativas en el ardor de la competición con afirmaciones como: “Cambio de pregunta es mejor que Menos 50%”, “deja el mejor para el final” o “no lo gastes ahora, arriégate”. Lo cierto es que en cada equipo fueron usados como un recurso para acercarse a la respuesta correcta, minimizando así, la tendencia de muchos estudiantes de abandonar antes de intentar obtener respuestas a través del razonamiento. El comodín más usado fue Consultar equipo, y el menos usado fue Nuevo intento.

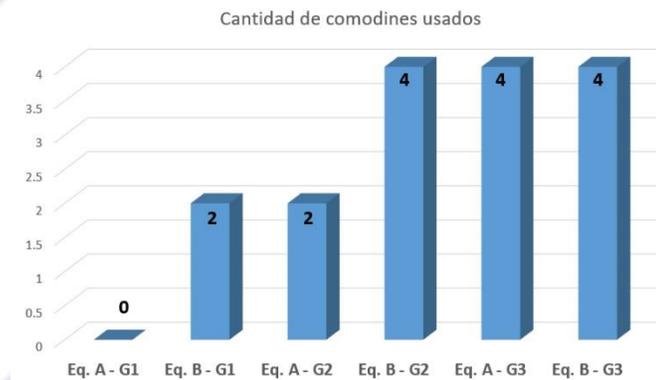


Figura 4. Comportamiento del uso de los comodines por equipo.

Como parte del propósito de la clase, cada estudiante recibe su propia evaluación, la cual se obtiene a partir del desempeño realizado con la aplicación de software y reforzada con el criterio del profesor sobre la calidad de las respuestas en cada caso. De esta forma objetiva y divertida, resultan evaluados todos los estudiantes presentados a la clase, en tiempo real y con conformidad por parte de los evaluados, pues su desempeño ha sido público e imparcialmente valorado. Los resultados generales de la evaluación de esta actividad docente se evidencian en la Figura 5.

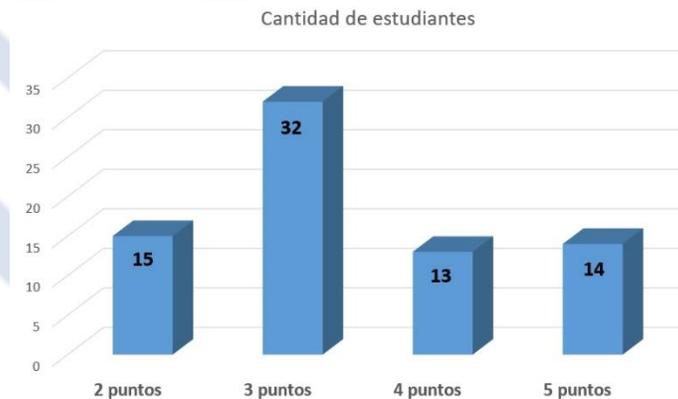


Figura 5. Estadística general de evaluación obtenida por los estudiantes.



## CONCLUSIONES

Luego del desarrollo de esta experiencia resaltan aspectos que propician satisfacción tanto a estudiantes como al profesor. La búsqueda insaciable de lo novedoso y la calidad en la enseñanza de la educación superior dándose de la mano. A continuación, algunas de las experiencias y evaluaciones del desarrollo de la estrategia.

Tras una breve introducción a la aplicación de software y explicar lo que acontecería durante la clase, los estudiantes comprendieron el funcionamiento de la aplicación en muy poco tiempo de iniciada la actividad. Comenzaron a relacionarse con la aplicación con familiaridad, y se explicaban la dinámica entre ellos a los que tardaron un poco más en aprender a usarla.

Mientras más avanzado el desarrollo de la actividad se interesaban más por responder las preguntas, pues el concepto de evaluación cambió en sus mentes por el concepto de nivel; sin embargo, continuaban respetando los requerimientos de una evaluación, pues lo asumían como si formara parte de las reglas de un juego.

Cada estudiante se enfrentó a más de una pregunta durante la actividad, no todos acertaron en cada respuesta, pero no se observa rasgo de frustración en ellos, y sí el deseo de que les correspondiera el turno nuevamente.

Aunque todo resultó en el cumplimiento del objetivo y bien recibido por los estudiantes, la aplicación no tiene todas las condiciones para aplicarla con equipos, fue implementada para el desarrollo individual. Sin embargo, con un mínimo de ajuste llevado a cabo manualmente por el profesor se desarrolló exitosamente la actividad evaluativa. Lo cual implica que es posible incorporar esa modalidad de equipos a la aplicación de software.

Esta dinámica ejercida entre los estudiantes y guiada por el profesor provoca el enfoque continuo de cada participante sobre la actividad. Respondiendo preguntas y generando una energía agradable y motivadora sobre el contenido. Pues mientras los estudiantes consideran que están jugando, la realidad es que están siendo evaluados y participando del PEA.

## REFERENCIAS

- [1] I. Fernández, "Las TICS en el ámbito educativo," Educación 9, 2010.
- [2] A. Torres, L. M. Rodríguez, A. Rodríguez, "Ludificación y sus posibilidades en el entorno de blended learning: revisión documental," Revista Iberoamericana de Educación a Distancia, vol. 21, no. 1, pp. 95-111, 2018.
- [3] M. Sailer, L. Homner, "The gamification of learning: A meta-analysis". 2020.



- [4] R. Landers, "Developing a theory of gamified learning: Linking serious games and gamification of learning," *Simulation & Gaming*, vol. 45, no. 6, pp. 752-768, 2014.
- [5] O. García, N. Acosta, "Estrategia de integración de las tecnologías informáticas al proceso de formación del profesional de la educación," *Pedagogía Universitaria*, vol. 18, no. 1, pp. 78-86, 2013.
- [6] D. López, A. Velamazán, and F. Muniesa, "Aprendizaje adaptativo en Moodle: tres casos prácticos". No. ART-2015-103697. 2015.
- [7] J. Colao, "With 24 Million Students, Codecademy Is Bigger Than You Thought." Online: <http://www.forbes.com/sites/jjcolao/2014/04/23/with-24-million-students-codecademy-is-bigger-than--you-thought>. 2014.
- [8] R. Aguilar, A. Rosete, "Los cursos masivos en línea en Coursera y su empleo potencial en los programas de ingeniería en América Latina," *Lámpsakos*, no. 14, pp. 61-70, 2015.
- [9] S. Gavarri, "El aprendizaje de lenguas extranjeras mediado por las TIC: aprender Inglés con Duolingo". *El toledo de Astier*, vol. 7, no. 12, pp. 56-65, 2016.
- [10] O. Rodríguez, "La motivación, motor del aprendizaje," *Revista Ciencias de la Salud*, vol. 4, 2006.
- [11] J. Flores, "Using gamification to enhance second language learning," *Digital Education Review*, no. 27, pp. 32-54, 2015.
- [12] J. Mullins, R. Sabherwal, "Gamification: A cognitive-emotional view," *Journal of Business Research*, vol. 106, pp. 304-314, 2020.
- [13] S. Deterding, D. Dixon, R. Khaled, and L. Nacke, "From game design elements to gamefulness: defining gamification". In *Proceedings of the 15th international academic MindTrek conference: Envisioning future media environments*, pp. 9-15, 2011.
- [14] Y. Pascuas, E. Vargas, J. Muñoz, "Experiencias motivacionales gamificadas: una revisión sistemática de literatura," *Innovación educativa*, vol. 17, no. 75, pp. 63-80, 2017.
- [15] R. César, "Aplicación de la metodología de la gamificación a través de las TIC en 3º de ESO". Tesis de Maestría. 2016.



LaSalle  
Universidad