

Escenarios de aprendizaje personalizados a partir de la evaluación del pensamiento computacional para el aprendizaje de competencias de programación mediante un entorno *b-Learning* y gamificación.

Tesis doctoral

Doctorando

Arturo Rojas López

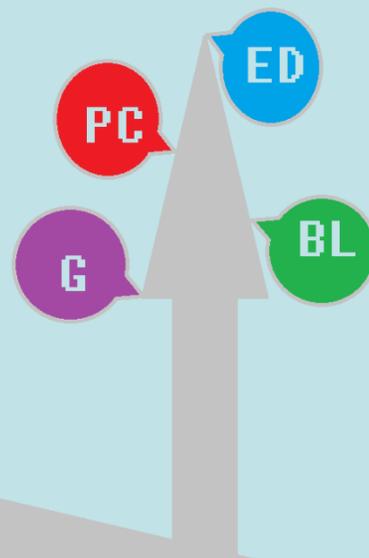
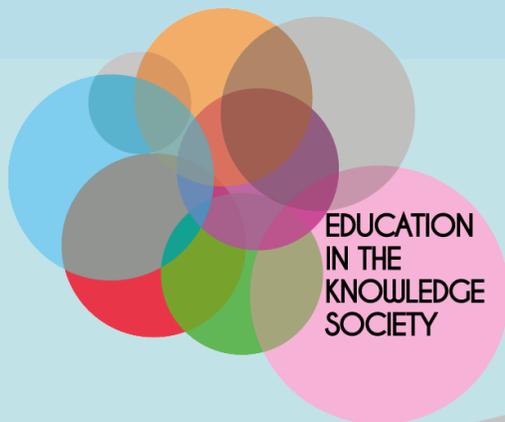
Director

Francisco José García Peñalvo



**VNiVERSiDAD
D SALAMANCA**

CAMPUS DE EXCELENCIA INTERNACIONAL



Escenarios de aprendizaje personalizados a partir de la evaluación del pensamiento computacional para el aprendizaje de competencias de programación mediante un entorno *b-Learning* y gamificación.

Tesis doctoral

Doctorando

Arturo Rojas López

Director

Francisco José García Peñalvo

Octubre 2019

Escenarios de aprendizaje personalizados a
partir de la evaluación del pensamiento
computacional para el aprendizaje de
competencias de programación mediante un
entorno b-Learning y gamificación.

Tesis doctoral

Director

Francisco José García Peñalvo

Doctorando

Arturo Rojas López

Octubre 2019

D. Francisco José García Peñalvo, Catedrático de Universidad del Departamento de Informática y Automática de la Universidad de Salamanca, en calidad de director del trabajo de tesis doctoral titulado “Escenarios de aprendizaje personalizados a partir de la evaluación del pensamiento computacional para el aprendizaje de competencias de programación mediante un entorno b-Learning y gamificación” y realizado por D. Arturo Rojas López

HACE CONSTAR

Que dicho trabajo tiene suficientes méritos teóricos contrastados adecuadamente mediante las validaciones oportunas, publicaciones relacionadas y aportaciones novedosas. Por todo ello considera que procede su defensa pública.

En Salamanca, a 22 de julio de 2019.

Dr. D. Francisco José García Peñalvo

Universidad de Salamanca

Cita recomendada:

A. Rojas-López. Escenarios de aprendizaje personalizados a partir de la evaluación del pensamiento computacional para el aprendizaje de competencias de programación mediante un entorno b-Learning y gamificación. Tesis doctoral, Programa de Doctorado en Formación en la Sociedad del Conocimiento, Universidad de Salamanca, Salamanca, España, 2019.

Resumen

El docente es una pieza clave de la funcionalidad y eficiencia de un modelo educativo universitario, pues, al determinar bajo su experiencia y creatividad un ambiente de aprendizaje, puede lograr que los estudiantes alcancen competencias profesionales específicas, sobre todo aquel docente que ha cambiado el paradigma centrado en sí mismo y se ha enfocado en el estudiante, además de hacer uso de estrategias educativas que apoyen el proceso de enseñanza - aprendizaje. La programación de computadoras es una actividad que crea tecnología de *software* y sigue siendo una profesión con alta demanda global de personal calificado, lo que ha llevado a las universidades a ofrecer diferentes alternativas de oferta educativa, es decir, el modelo de aprendizaje presencial ya no es la única forma en que se ofrece la adquisición de competencias respectivas a la creación de *software*.

En las hojas de asignatura o programas de estudio de las materias de programación generalmente está indicado lo que se va a enseñar y una orientación de la estrategia para alcanzar los objetivos planteados, es decir, está indicado el contenido del curso, los métodos y las técnicas, así como los medios y materiales didácticos, pero no hacen mención de tecnologías o de apoyarse de un medio tecnológico. Internet bien puede proporcionar información a los estudiantes acerca de los contenidos de la clase, ofrecer códigos de programas y vídeos que ilustran el desarrollo con tecnologías nuevas, pero existe el problema de la adquisición de las competencias básicas para aprender la programación de computadoras y el desarrollo de *software* en general, principalmente la capacidad cognitiva para resolver problemas a través de una metodología. La propuesta, planeación e intervención en clase de estrategias educativas, son las mejores acciones que puede hacer el docente en la búsqueda de ayudar al estudiante a superar el exceso de información que puede encontrar en Internet y así crear un plan de trabajo para innovar en la educación superior.

La presente tesis doctoral, resultado del trabajo de investigación realizado, tiene la finalidad de contribuir a la adquisición de las competencias básicas de programación de los estudiantes que participen en un ambiente *b-Learning* diseñado con base a su singularidad de aprendizaje y

personalización de contenidos a partir del nivel cognitivo de su pensamiento computacional, además de agregar gamificación como acción motivadora.

Para lograr el objetivo anterior, se proponen estrategias educativas que logren motivar a los estudiantes en la adquisición de las competencias básicas de programación: i. evaluación del pensamiento computacional, ii. educación personalizada y iii. gamificación.

La investigación tiene claramente diferenciadas dos fases, la primera donde se pudo realizar una planeación de las actividades que permitieran la evaluación del pensamiento computacional, la creación de escenarios de aprendizaje con base a los resultados obtenidos y la creación de los elementos de gamificación; la segunda etapa en donde se llevó a cabo la intervención de las estrategias en el aula y la recolección de información que midió la eficacia de las acciones propuestas para lograr el objetivo planteado. Hay que destacar que la fase de planeación fue posible replicarla en una universidad y país diferente, INACAP – Chile, exponiendo la aportación y valor agregado de la investigación debido a su adaptabilidad para un diferente contexto educativo.

La conclusión principal de la investigación es que la motivación del estudiante, a través del reconocimiento de las habilidades de pensamiento computacional que posee, la oferta de opciones de estudio considerando su individualidad en el proceso de aprendizaje de las competencias básicas de programación en un ambiente de gamificación y el uso de un entorno *b-Learning*, permite reducir el porcentaje de la deserción universitaria sin sacrificar el nivel de calidad educativo; sin embargo, la falta de autonomía de los estudiantes es un fuerte reto a vencer considerando los constantes cambios que las tendencias tecnológicas marcan en ambientes de creación de *software* y que de forma autónoma tendrán que aprender los egresados universitarios.

Palabras clave: Pensamiento computacional, educación personalizada, gamificación, *b-Learning*, educación superior, curso de programación de computadora, aprendizaje de programación, enseñanza de programación, estudiantes de nuevo ingreso.

Abstract

The teacher is a key piece of the functionality and efficiency of a university educational model, because, by determining a learning environment under his experience and creativity, he can get that students achieve specific professional competences, especially that teacher who has changed the paradigm focused on himself and has focused on the student, in addition to making use of educational strategies that support the teaching - learning process. Computer programming is an activity that creates software technology and continues to be a profession with high global demand for qualified personnel, which has led universities to offer different alternatives of educational offer, that is, the model of face-to-face learning is not the only way to offer the acquisition of respective competences to the creation of software.

In the subject sheets or study programs of the programming courses, it is usually indicated what is going to be taught and an orientation of the strategy to achieve the proposed objectives, that is, the content of the course is indicated, the methods and the techniques, as well as, the media and didactic materials, but do not mention technologies or rely on a technological medium. The Internet may well provide information to students about the contents of the class, offer program codes and videos that illustrate the development with new technologies, but there is the problem of acquiring the basic skills to learn computer programming and development of software in general, mainly the cognitive ability to solve problems through a methodology. The proposal, planning and intervention in class of educational strategies, are the best actions that the teacher can do in the search to help the student overcome the excess of information that can be found on the internet and thus create a work plan to innovate in the higher education.

The present doctoral thesis, the result of research work carried out, has the purpose of contributing to the acquisition of the basic skills of programming of students that participate in a b-Learning environment designed based on their singularity of learning and personalization of contents from the cognitive level of computational thinking, besides adding gamification as a motivating activity.

To achieve the above objective, educational strategies are proposed that motivate students to acquire the basic skills of programming: i. computational thinking evaluation, ii. personalized education and iii. gamification.

The research has clearly differentiated two phases, the first where it was possible to carry out a planning of the activities that allowed the evaluation of the computational thinking, the creation of learning scenarios based on the obtained results and the creation of the elements of gamification; the second phase where the intervention of strategies was carried out in the classroom and the collection of information that measured the effectiveness of the proposed actions to achieve the stated objective. It should be noted that the planning phase was replicable in a different university and country, INACAP - Chile, exposing the contribution and added value of the research due to its adaptability for a different educational context.

The main conclusion of the investigation is that motivation of the student, through the recognition of the computational thinking skills that he possesses, the offer of study options considering his individuality in the learning process of the basic programming competences in an environment of gamification and the use of a b-Learning environment, allows reducing the percentage of university dropout without sacrificing the level of educational quality; however, the lack of autonomy of students is a strong challenge to overcome considering the constant changes that technological trends mark in software creation environments and that university graduates will have to learn independently.

Keywords: Computational thinking, personalized education, gamification, b-Learning, higher education, computer programming course, programming learning, programming teaching, new students.

Contenido

CAPÍTULO 1. INTRODUCCIÓN.....	1
1.1 Presentación	1
1.2 Planteamiento del problema	2
1.3 Objetivos de la investigación	8
1.4 Pregunta de la investigación.....	9
1.5 Justificación de la investigación.....	10
1.6 Viabilidad del estudio.....	11
1.7 Evaluación de las deficiencias en el conocimiento del problema	12
1.8 Metodología	12
1.9 Población y muestra	17
1.10 Marco de trabajo.....	18
1.11 Estructura del documento.....	20
BLOQUE I: MARCO TEÓRICO	23
CAPÍTULO 2. CONCEPTOS TEÓRICOS	25
2.1 Educación personalizada.....	25
2.2 Pensamiento computacional.....	29
2.3 <i>b-Learning</i>	40
2.4 Gamificación	44
2.5 Conclusiones	52
CAPÍTULO 3. REVISIÓN SISTEMÁTICA DE LA LITERATURA	55
3.1 Introducción	55
3.2 Planificación del proceso de revisión.....	57
3.2.1 Identificación de la necesidad	58
3.2.2 Protocolo de revisión.....	58
3.2.2.1 Preguntas de investigación	58
3.2.2.2 Estrategia de búsqueda	59
3.2.2.3 Criterios de inclusión / exclusión	59
3.3 Ejecución de revisión	60
3.3.1 Selección de materiales	60
3.3.2 Valoración de materiales	61
3.3.3 Extracción y síntesis de datos.....	62

3.4	Informe de resultados	68
3.5	Limitaciones de la revisión	74
3.6	Conclusiones	74
BLOQUE II: MARCO EMPÍRICO		77
CAPÍTULO 4. INTERVENCIÓN CON PENSAMIENTO COMPUTACIONAL, EDUCACIÓN PERSONALIZADA Y B-LEARNING		79
4.1	Introducción	79
4.2	Metodología	80
4.2.1	Población y muestra	80
4.2.2	Fase planeación	81
4.2.2.1	Educación personalizada	81
4.2.2.2	Evaluación del pensamiento computacional	84
4.2.2.3	Escenarios de aprendizaje	92
4.2.3	Instrumentos	94
4.2.4	Fase intervención.....	96
4.2.5	Fase analítica	106
4.2.6	Fase discusión	111
4.3	Estancia en INACAP – Chile	116
4.3.1	Contextualización del trabajo.....	116
4.3.2	Fase planeación – diseño del experimento	117
4.3.2.1	Escenarios de aprendizaje	121
4.3.2.2	Instrumentos	125
4.4	Conclusiones	127
CAPÍTULO 5. INTERVENCIÓN CON GAMIFICACIÓN, EDUCACIÓN PERSONALIZADA Y B-LEARNING		131
5.1	Introducción	131
5.2	Metodología	132
5.2.1	Población y muestra	132
5.2.2	Fase planeación	133
5.2.2.1	Educación personalizada	133
5.2.2.2	Gamificación	135
5.2.3	Instrumentos	143

5.2.4	Fase intervención.....	145
5.2.5	Fase analítica	151
5.2.6	Fase discusión	156
5.3	Conclusiones	162
BLOQUE III: DISCUSIÓN Y CONCLUSIONES		166
CAPÍTULO 6. DISCUSIÓN.....		168
6.1	Introducción	168
6.2	Discusión.....	169
6.3	Conclusiones	171
CAPÍTULO 7. CONCLUSIONES.....		172
7.1	Introducción	172
7.2	Conclusiones	172
7.3	Contribuciones y futuras líneas de investigación.....	174
7.4	Publicaciones.....	177
APÉNDICES.....		180
APÉNDICE A. Indicadores por grupo del 2009 al 2016 para el curso Metodología de la programación.....		182
APÉNDICE B. Indicadores por año del 2009 al 2016 para el curso Metodología de la programación.....		188
APÉNDICE C. Indicadores por grupo del 2010 al 2017 para el curso Programación.....		190
APÉNDICE D. Indicadores por año del 2010 al 2017 para el curso Programación.		196
APÉNDICE E. Detalle de información clave por material SCOPUS tema gamificación		198
APÉNDICE F. Detalle de información clave por material WOS tema gamificación.....		204
APÉNDICE G. Detalle de información clave por material SCOPUS tema pensamiento computacional		210
APÉNDICE H. Detalle de información clave por material WOS tema pensamiento computacional		216
Referencias.....		222

Índice de figuras

Figura 1. Tendencia de indicadores acerca del estatus académico del curso MP del 2009 al 2016. Fuente: Elaboración propia	6
Figura 2. Tendencia de indicadores acerca del estatus académico del curso Programación del 2010 al 2017. Fuente: Elaboración propia	7
Figura 3. Diseño de investigación mixta tipo triangulación concurrente. Fuente: Elaboración propia	16
Figura 4. Etapas y actividades de la revisión sistemática en la tesis doctoral. Fuente: Elaboración propia.....	57
Figura 5. Diagrama de flujo de la selección de materiales en la revisión sistemática. Fuente: Basado en Ferreras-Fernández (2016)	63
Figura 6. Aspecto del curso Metodología de la programación en la plataforma Moodle. Fuente: Elaboración propia	84
Figura 7. Descripción del móvil con una expresión: $(-3 \ -1 \ 1) \ (1 \ 1) \ (2 \ 3)$. Fuente: Bebras (2015). 87	87
Figura 8. Opciones de respuestas del reactivo móvil. Fuente: Bebras (2015)	88
Figura 9. La tarea de recoger las manzanas. Fuente: Bebras (2015).....	88
Figura 10. Seis espías intercambiando información. Fuente: Bebras (2015).....	89
Figura 11. Cinco espías intercambiando información. Fuente: Bebras (2015)	90
Figura 12. Ejemplo de cinco castores al cruzar un hoyo. Fuente: Talent Search (2015)	90
Figura 13. Siete castores buscando cruzar hoyos. Fuente: Talent Search (2015)	91
Figura 14. Salto de Charcos. Fuente: Talent Search (2015).....	91
Figura 15. Porcentajes reactivo móvil iteración 1. Fuente: Elaboración propia.....	97
Figura 16. Porcentajes reactivo canguro iteración 1. Fuente: Elaboración propia	97
Figura 17. Porcentajes reactivo espías iteración 1. Fuente: Elaboración propia	98
Figura 18. Porcentajes reactivo castores en movimiento iteración 1. Fuente: Elaboración propia ...	98
Figura 19. Porcentajes reactivo salto de charcos iteración 1. Fuente: Elaboración propia	98
Figura 20. Porcentajes reactivo móvil iteración 2. Fuente: Elaboración propia.....	103
Figura 21. Porcentajes reactivo canguro iteración 2. Fuente: Elaboración propia	103
Figura 22. Porcentajes reactivo espías iteración 2. Fuente: Elaboración propia	104
Figura 23. Porcentajes reactivo castores iteración 2. Fuente: Elaboración propia	104
Figura 24. Porcentajes reactivo salto de charcos iteración 2. Fuente: Elaboración propia	104
Figura 25. Resultados de la iteración septiembre – diciembre 2016. Fuente: Elaboración propia .	109
Figura 26. Resultados por escenarios en los grupos experimentales septiembre – diciembre 2016. Fuente: Elaboración propia	110
Figura 27. Porcentajes que favorecen la iteración 1 de la intervención realizada. Fuente: Elaboración propia	113
Figura 28. Resultados grupos experimentales septiembre – diciembre 2017 iteración 2. Fuente: Elaboración propia	115
Figura 29. Insignias utilizadas durante el curso de Programación. Fuente: Elaboración propia.....	138
Figura 30. Medallero al final del cuatrimestre iteración 1. Fuente: Elaboración propia.....	145
Figura 31. Medallero final de la segunda iteración. Fuente: Elaboración propia.....	148

Figura 32. Producción científica realizada durante el periodo de la tesis doctoral. Fuente:
Elaboración propia 177

Índice de tablas

Tabla 1. Indicadores del estado académico en el curso MP del 2009 al 2016. Fuente: Elaboración propia.....	5
Tabla 2. Indicadores del estado académico en el curso Programación del 2009 al 2016. Fuente: Elaboración propia	7
Tabla 3. Población de datos históricos. Fuente: Elaboración propia.....	17
Tabla 4. Datos de grupos de control y experimentales. Fuente: Elaboración propia	18
Tabla 5. Algunos estudios sobre gamificación en la educación. Fuente: Elaboración propia con la Dr. Elvira Rincón- Flores.....	49
Tabla 6. Preguntas de investigación. Fuente: Elaboración propia.....	59
Tabla 7. Fórmulas de búsqueda. Fuente: Elaboración propia	60
Tabla 8. Cantidad de materiales obtenidos en las bases de datos. Fuente: Elaboración propia	61
Tabla 9. Cantidad de materiales incluidos en la revisión por estrategia. Fuente: Elaboración propia	62
Tabla 10. Identificación de los 5 materiales obtenidos de SCOPUS de la estrategia gamificación. Fuente: Elaboración propia	64
Tabla 11. Identificación de los 6 materiales obtenidos de WOS de la estrategia gamificación. Fuente: Elaboración propia	65
Tabla 12. Identificación de los 5 materiales obtenidos de SCOPUS de la estrategia pensamiento computacional. Fuente: Elaboración propia.....	66
Tabla 13. Identificación de los 7 materiales obtenidos de WOS de la estrategia pensamiento computacional. Fuente: Elaboración propia.....	67
Tabla 14. Resumen de materiales por país. Fuente: Elaboración propia.....	68
Tabla 15. Datos de grupos de control y experimentales. Fuente: Elaboración propia	81
Tabla 16. Conocimientos generales del curso Metodología de la programación. Fuente: Elaboración propia.....	83
Tabla 17. Indicadores de evaluación en listas de cotejo. Fuente: Elaboración propia	84
Tabla 18. Relación de habilidades del pensamiento computacional y conocimientos del curso MP. Fuente: Elaboración propia	86
Tabla 19. Instrumento encuesta para obtener información CUAN-CUAL a mitad del cuatrimestre en intervención 1. Fuente: Elaboración propia	95
Tabla 20. Instrumento encuesta para obtener información CUAN-CUAL al final del cuatrimestre en intervención 1. Fuente: Elaboración propia	96
Tabla 21. Número de estudiantes por escenario iteración 1. Fuente: Elaboración propia	99
Tabla 22. Encuesta a mitad de cuatrimestre iteración 1. Fuente: Elaboración propia.....	99
Tabla 23. Resultados al final del cuatrimestre iteración 1. Fuente: Elaboración propia	101
Tabla 24. Encuesta al final del cuatrimestre iteración 1. Fuente: Elaboración propia	102
Tabla 25. Número de estudiantes por escenario iteración 2. Fuente: Elaboración propia	105
Tabla 26. Resultados evaluativos en iteración 2 y comparación con los datos históricos. Fuente: Elaboración propia	106
Tabla 27. Encuesta al final del cuatrimestre iteración 2. Fuente: Elaboración propia	107

Tabla 28. Información comparativa de los últimos 8 años primer cuatrimestre con iteración 1. Fuente: Elaboración propia	108
Tabla 29. Planeación del curso Fundamentos de programación en INACAP. Fuente: Elaboración propia.....	118
Tabla 30. Aprendizajes esperados por unidad de aprendizaje. Fuente: Elaboración propia	118
Tabla 31. Criterios de evaluación y actividades, unidad de aprendizaje 1. Fuente: Elaboración propia.....	119
Tabla 32. Criterios de evaluación y actividades, unidad de aprendizaje 2. Fuente: Elaboración propia.....	119
Tabla 33. Criterios de evaluación y actividades, unidad de aprendizaje 3. Fuente: Elaboración propia.....	119
Tabla 34. Criterios de evaluación y actividades, unidad de aprendizaje 4. Fuente: Elaboración propia.....	120
Tabla 35. Criterios de evaluación y actividades, unidad de aprendizaje 5. Fuente: Elaboración propia.....	120
Tabla 36. Relación de habilidades del pensamiento computacional con unidades de aprendizaje. Fuente: Elaboración propia	121
Tabla 37. Determinación de los escenarios de aprendizaje en INACAP. Fuente: Elaboración propia	122
Tabla 38. Encuesta a mitad del semestre para INACAP. Fuente: Elaboración propia.....	126
Tabla 39. Encuesta al final del semestre para INACAP. Fuente: Elaboración propia	126
Tabla 40. Datos de grupos de control y experimentales intervención 2. Fuente: Elaboración propia	133
Tabla 41. Conocimientos generales del curso Programación. Fuente: Elaboración propia	134
Tabla 42. Sistema gamificado. Fuente: Elaboración propia.....	136
Tabla 43. Meta-evaluación en la dimensión formativa. Fuente: Basada en el modelo Stufflebeam (2011)	141
Tabla 44. Meta-evaluación en la dimensión proactiva. Fuente: Basada en el modelo Stufflebeam (2011)	142
Tabla 45. Organización de la actividad de gamificación en la segunda iteración. Fuente: Elaboración propia	143
Tabla 46. Instrumento encuesta para intervención 2 al final de la iteración 1. Fuente: Elaboración propia.....	144
Tabla 47. Información adicional de la encuesta para la iteración 2. Fuente: Elaboración propia ...	145
Tabla 48. Resultados de encuesta final iteración 1 intervención 2. Fuente: Elaboración propia	147
Tabla 49. Información de la encuesta al final del curso en la iteración 2. Fuente: Elaboración propia	150
Tabla 50. Información adicional de la encuesta de la iteración 2. Fuente: Elaboración propia	151
Tabla 51. Detalle de resultados en grupo de control y experimental, iteración 2.	155
Tabla 52. Concentrado de resultados en grupo de control y experimental, iteración 2.....	156
Tabla 53. Comparativo de información en los últimos 8 años y de intervención 2.	157

CAPÍTULO 1. INTRODUCCIÓN

1.1 Presentación

El nuevo modelo educativo para la educación obligatoria en México, educar para la libertad y la creatividad (Secretaría de Educación Pública, 2017) propone elementos de adaptación y flexibilidad que pueden ser considerados dentro de la educación personalizada; de los cinco ejes principales que determinan al modelo, las siguientes notas son recogidas:

- I. El planteamiento curricular. Un enfoque humanista, otorgando a las escuelas un margen de autonomía curricular, con lo cual podrán adaptar los contenidos educativos a las necesidades y contextos específicos de sus estudiantes.
- II. La escuela al centro del sistema educativo. La escuela debe enfocarse en alcanzar el máximo logro de aprendizajes de todos sus estudiantes.
- III. Formación y desarrollo profesional docente. El profesor genera ambientes de aprendizaje incluyentes, y es capaz de adaptar el currículo a su contexto específico.
- IV. Inclusión y equidad. El sistema educativo es para todos los estudiantes y debe ofrecer las bases para que cada uno cuente con oportunidades efectivas para el desarrollo de todas sus capacidades, reconociendo su contexto social y cultural.
- V. La gobernanza del sistema educativo. Se reconoce la pluralidad de actores involucrados en el proceso educativo.

Resulta clara la influencia de considerar al estudiante como un ser humano único en su proceso de aprendizaje, de apoyarlo con las herramientas que considere necesarias el docente; pues la reforma educativa actual pretende que los estudiantes egresados de los diferentes niveles educativos del país, desarrollen su potencial en beneficio personal y de la sociedad, en consecuencia, colaborar con el aprendizaje para toda la vida. Así, la enseñanza en la educación superior ha de agregar actividades de investigación que contribuyan en sus egresados en el enriquecimiento de las sociedades del

Introducción

conocimiento; favorecer el cambio de paradigma de la educación escolarizada, que en México se caracteriza por el aprendizaje de forma masiva y estandarizada (Universidad del Valle de México, 2019).

La educación personalizada (Ferraro, Álvarez, & Peñalvo, 2004; Berlanga & García, 2005; Berlanga & García, 2005; Berlanga & García, 2008; Lerís & Sein-Echaluce, 2011), el pensamiento computacional (Lye & Koh, 2014; Swaid, 2015; García-Peñalvo & Cruz-Benito, 2016; García-Peñalvo F. J., 2016; García-Peñalvo F. J., 2016; García-Peñalvo F. J., Reimann, Tuul, Rees, & Jormanainen, 2016; García-Peñalvo & Mendes, 2018), la gamificación (Caponetto, Earp, & Ott, 2014; Surendeleg, Murwa, Yun, & Kim, 2014; Villalustre & Del Moral, 2015; Carreño-León, Sandoval-Bringas, Álvarez-Rodríguez, & Camacho-González, 2018; Barna & Fodor, 2018), y el *b-Learning* (Ramírez, 2012; González, 2015; Rosser-Limiñana & Martínez, 2015), al ser utilizados como estrategias educativas en la Educación Superior en México, teniendo como caso de estudio el nivel Técnico Superior Universitario en la Universidad Tecnológica de Puebla en la División de Tecnologías de la Información y Comunicación, permitieron crear la presente tesis doctoral.

1.2 Planteamiento del problema

En similitud a otras universidades, varios son los factores que enfrentan los estudiantes para mantener una vida académica universitaria como los que comentan Echegaray et al. (2017): “desmotivación de los estudiantes, confusiones con respecto a la elección del grado, falta de información acerca de la vida universitaria o los planes y contenidos de los grados, confusión con el propio diseño de la estructura universitaria, deficiencias en la formación académica previa, sentimientos de inseguridad acerca de las propias capacidades, etc.”. Particularmente, el aprendizaje-enseñanza de la programación de computadoras no es una actividad sencilla para los estudiantes-docentes; Selby (2015) particularmente aborda el problema del aprendizaje, destacando que algunas de las razones

identificadas son: “inadecuada comprensión de cómo trabaja un modelo computacional, una incapacidad para dominar la lectura, rastreo y escritura de código, así como comprender conceptos de alto nivel tales como el diseño”. Milne y Rowe (2002) presentaron los resultados de un análisis estadístico a una encuesta realizada en línea a 66 participantes, entre estudiantes y profesores de programación alrededor del Reino Unido, indicando que la incapacidad para comprender qué le está ocurriendo al programa en la memoria de la computadora dificulta el entendimiento de conceptos avanzados tales como apuntadores o los relacionados a la gestión de la memoria, en consecuencia, sugieren que a través de la aclaración de tal proceso computacional, los estudiantes sean capaces de crear un claro modelo mental de la ejecución de instrucciones. Sakhnini y Hazzan (2008) presentaron conclusiones en beneficio del proceso enseñanza-aprendizaje de los tipos de datos abstractos. Primero, los profesores deben ser conscientes que los estudiantes tienden a tratar con problemas desconocidos, confiando en los términos e ideas que le son familiares desde su vida diaria o desde sus estudios previos, por lo tanto, las diferencias entre los nuevos problemas y los familiares deben ser explícitamente discutidas con los estudiantes, y deben ser expuestos a un amplio número de problemas de diferentes tipos y niveles con el fin de mejorar sus habilidades para tratar con problemas no familiares. Segundo, con el fin de guiar a los estudiantes en la concepción de los tipos de datos abstractos como objetos, recomiendan separar la definición y uso desde su implementación, es decir, iniciar enseñando el concepto y posteriormente su implementación. Y tercero, los estudiantes deben ser expuestos a muchos métodos, estrategias y herramientas de resolución de problemas; tales como, el método divide y vencerás o las recomendaciones de Pólya (1957) relacionadas al proceso de resolución de problemas; primero, comprender el problema; segundo, encontrar la conexión entre los datos y lo desconocido, se pueden considerar problemas auxiliares si una conexión inmediata no puede ser encontrada, se debe eventualmente obtener un plan de la solución; tercero, llevar a cabo el plan, y cuarto, examinar la solución obtenida.

Introducción

La frustración que puede causar en los estudiantes durante los primeros meses de estudio el no poder crear soluciones básicas de *software* a problemas de la industria o mercado laboral, puede ser el factor principal para que tome la decisión de abandonar los estudios. La demanda de programadores es aún requerida en el sistema económico de cualquier país y su diversidad de industrias: automotriz, dispositivos embebidos, academia, tecnologías de negocios, publicitaria y de pagos (Mazaika, 2017), particularmente la Oficina de Estadísticas Laborales de Estados Unidos (Bureau_of_Labor_Statistics, 2019) pronostica un incremento del 24% del 2016 al 2026; por lo que la pertinencia de competencias adquiridas y su funcionalidad en el ejercicio profesional refleja el éxito de un modelo educativo universitario. Además, hay que considerar el proceso de automatización de empleos que hará necesaria la actividad profesional de analistas de datos, expertos en inteligencia artificial y aprendizaje de máquinas (García-Vega, 2019). En los últimos trece años el modo habitual de impartir la materia inicial a la programación en la Universidad Tecnológica de Puebla (UTP) ha sido cien por ciento presencial, buscando unificar el aprendizaje de todos los estudiantes basándose completamente en lo indicado en la hoja de asignatura diseñada en su momento a nivel nacional por la Coordinación General de Universidades Tecnológicas y Politécnicas. Factores como la deserción, índice alto y variable de reprobación, falta de desempeño académico de los estudiantes fuera de las instalaciones hacen necesario proponer otras formas que beneficien la calidad educativa.

Los indicadores que se consideran para determinar el estado académico de la división de Tecnologías de la Información y Comunicación en la UTP son:

- Porcentaje de acreditación (PA – aprobados sobre matriculados),
- Calificación (C – promedio del grupo), y
- Porcentaje de deserción (PD – matriculados al siguiente cuatrimestre sobre matriculados al inicio).

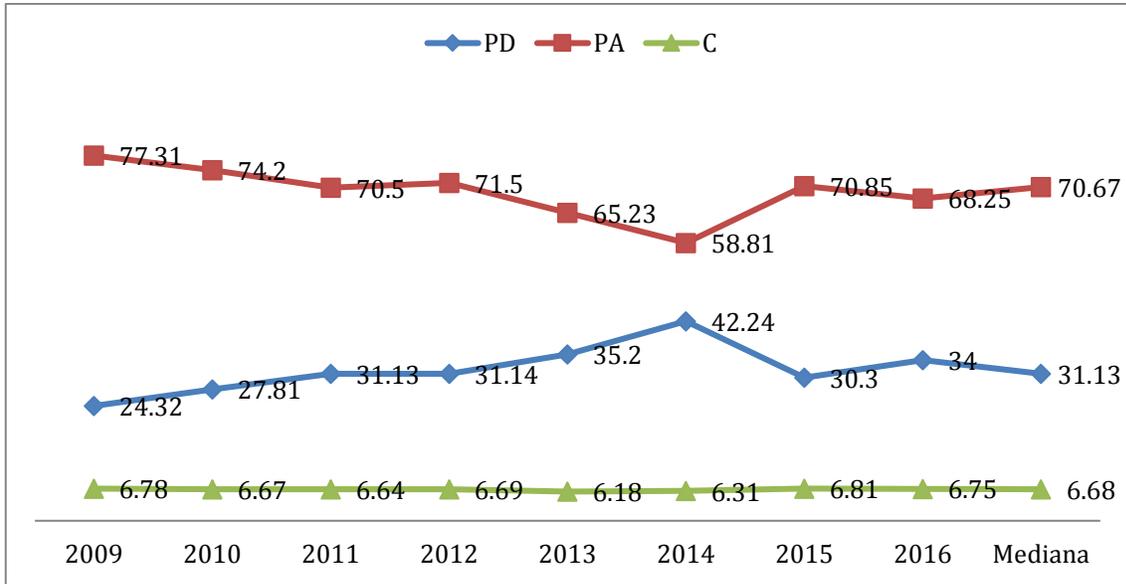
Los datos obtenidos de información interna para el curso Metodología de la programación (MP) se muestran completos en el Apéndice A; los promedios del PA, C y PD del 2009 al 2016 se muestran en el Apéndice B.

La mediana del porcentaje de deserción estudiantil (PD) al terminar el primer cuatrimestre del 2009 al 2016 fue de 31,13. La mediana del porcentaje de acreditación estudiantil (PA) del curso Metodología de la programación en el periodo mencionado anteriormente fue de 70,67. La mediana de calificación (C) en el mismo curso y periodo de los estudiantes que acreditaron fue de 6,68. La Tabla 1 contiene la información de los indicadores de cada cuatrimestre de los últimos 8 años. La Figura 1 ilustra que PD ha ido en aumento, mientras que PA ha ido reduciéndose y C se ha mantenido baja.

Tabla 1. Indicadores del estado académico en el curso MP del 2009 al 2016. Fuente: Elaboración propia

Año	Porcentajes Primer cuatrimestre		Promedio de calificación
	Deserción	Acreditación	
2009	24,32	77,31	6,78
2010	27,81	74,20	6,67
2011	31,13	70,50	6,64
2012	31,14	71,50	6,69
2013	35,20	65,23	6,18
2014	42,24	58,81	6,31
2015	30,30	70,85	6,81
2016	34,00	68,25	6,75
Mediana	31,13	70,67	6,68

Introducción



*Figura 1. Tendencia de indicadores acerca del estatus académico del curso MP del 2009 al 2016.
Fuente: Elaboración propia*

Los datos obtenidos de información interna para el curso Programación se muestran completos en el Apéndice C; los promedios del PA, C y PD del 2010 al 2017 se muestran en el Apéndice D.

La mediana del porcentaje de deserción estudiantil (PD) al concluir el segundo cuatrimestre del 2010 al 2017 fue de 30,76. La mediana del porcentaje de acreditación estudiantil (PA) del curso Programación en el periodo anterior fue de 71,89. La mediana de calificación (C) en el mismo curso y periodo de los estudiantes que acreditan fue de 7,03. La Tabla 2 contiene la información de los indicadores de cada cuatrimestre en los últimos 8 años. La Figura 2 ilustra que PD ha ido en aumento llegando casi a duplicarlo, mientras que PA ha ido reduciéndose desde el 2010, y C se ha mantenido baja.

A partir de los datos anteriores, dos problemas principales ocuparon el trabajo de tesis doctoral:

1. Determinar estrategias para reducir la deserción escolar, y
2. Proporcionar competencias básicas para la programación de computadoras, funcionales a los requerimientos de la industria del *software*.

Tabla 2. Indicadores del estado académico en el curso Programación del 2009 al 2016. Fuente: Elaboración propia

Año	Porcentajes Segundo cuatrimestre		Promedio de Calificación
	Deserción	Acreditación	
2010	16,48	85,20	7,0
2011	33,69	70,45	6,29
2012	29,99	73,07	6,92
2013	28,22	73,87	7,07
2014	31,38	71,62	7,24
2015	30,89	71,09	6,94
2016	30,63	72,17	7,05
2017	32,25	69,57	7,09
Mediana	30,76	71,89	7,03

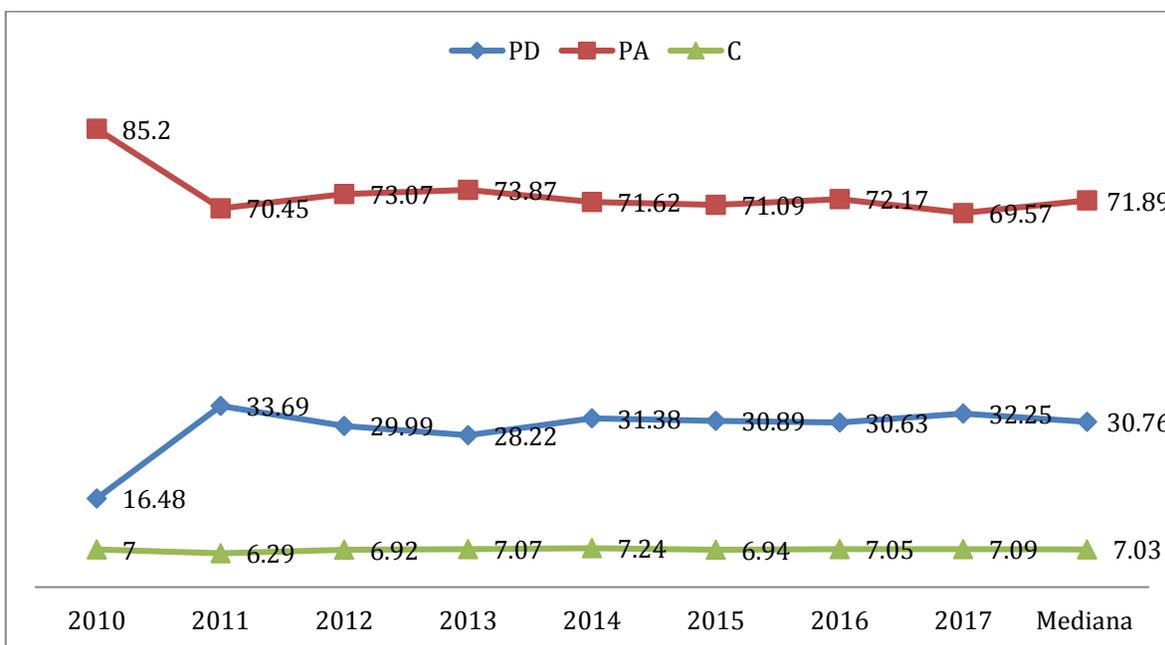


Figura 2. Tendencia de indicadores acerca del estatus académico del curso Programación del 2010 al 2017. Fuente: Elaboración propia

Hernández et al. (2006) proponen cinco elementos para plantear un problema: “los objetivos que persigue la investigación, las preguntas de investigación, la justificación de la investigación, la viabilidad del estudio y la valuación de las deficiencias en el conocimiento del problema”. Los siguientes subtemas abordan cada uno de los mencionados elementos.

1.3 Objetivos de la investigación

Una vez determinado el problema principal, el objetivo de esta investigación es demostrar que a partir de la combinación de estrategias educativas es posible reducir la deserción escolar en los cursos iniciales de programación de computadoras permitiendo al mismo tiempo la adquisición de las correspondientes competencias básicas en la creación de *software*. Existen publicaciones que han trabajado el apoyo de la enseñanza de programación en educación superior por medio de diferentes estrategias. Por ejemplo, la gamificación ha sido usada principalmente como medio de motivación y novedad para cursos de programación (Carreño-León, Sandoval-Bringas, Álvarez-Rodríguez, & Camacho-González, 2018; Barna & Fodor, 2018; Fotaris, Mastoras, Leinfellner, & Yasmine, 2015; Lopes & Mesquita, 2015; Lopes, 2014). El pensamiento computacional ha sido utilizado para mejorar las habilidades en la resolución de problemas, actividad fuertemente ligada con el diseño y creación de software (Chen, 2017; Romero, Lepage, & Lille, 2017; Ni, 2017; Ying & Pingping, 2017; Gao Q., 2014; Yinnan & Chaosheng, 2012). La educación personalizada ha sido promovida para favorecer la individualidad de aprendizaje (Lerís & Sein-Echaluce, 2011; Tekin, Braun, & Van-Der-Schaar, 2015; Laksitowening & Hasibuan, 2015; Tejeda-Lorente, Bernabé-Moreno, Porcel, Galindo-Moreno, & Herrera-Viedma, 2015). El *b-Learning* ha sido un entorno de aprendizaje que auxilia la actividad docente dentro y fuera del salón de clases (Kostolányová, 2017; Fidalgo-Blanco, Martínez-Nuñez, Borrás-Gene, & Sanchez-Medina, 2017; Olelewe & Agomuo, 2016; Shi, Zhang, & Sun, 2016). En ninguno de los trabajos publicados se ha realizado una combinación de las estrategias, lo anterior representa el valor agregado de la investigación en la tesis doctoral, además de realizar la intervención de diseños experimentales en curso de programación de computadoras en educación superior. En resumen, el objetivo general es:

- Diseñar escenarios de aprendizaje personalizados, considerando las habilidades del pensamiento computacional de los estudiantes de nuevo ingreso, que puedan ser utilizados en modalidad *b-Learning* para reducir el abandono de estudios a través de la gamificación

como estrategia de motivación y permita la adquisición de competencias básicas de Programación.

Los objetivos específicos que permiten apoyar el logro del objetivo general son:

1. Evaluar el contenido actual del sistema gestor de aprendizaje, que se utiliza en la asignatura Metodología de la programación y Programación tomando como caso de estudio la división de Tecnologías de la Información y Comunicación en la Universidad Tecnológica de Puebla.
2. Revisar el estado del arte a través de una revisión sistemática de la literatura, sobre los conceptos de educación personalizada, pensamiento computacional, gamificación y *b-Learning* en la enseñanza de programación en educación superior.
3. Determinar el impacto que tiene la intervención de la educación personalizada, pensamiento computacional, gamificación y *b-Learning* en la reducción de la baja escolar en los estudiantes de nuevo ingreso.

1.4 Pregunta de la investigación

Con la intención de establecer el sentido de la investigación, se determina la pregunta de investigación para aclarar y delimitar el trabajo a realizar. Previamente se ha comentado que estrategias educativas se han aplicado en la educación superior y específicamente en las áreas de ciencias de la computación e informática con la meta de favorecer entre otras cosas el desempeño académico de los estudiantes. El pensamiento computacional, educación personalizada, *b-Learning* y gamificación por separado han sido estrategias utilizadas en cursos de programación. En cada publicación, las mejoras en la adquisición de competencias para el desarrollo de software están sujetas al mapa o malla curricular de la institución educativa. Por lo anterior, la pregunta de investigación de la tesis doctoral es la siguiente:

¿Qué competencias básicas de Programación obtienen los estudiantes de nuevo ingreso al participar en un ambiente de aprendizaje donde se apliquen las estrategias educativas de educación

Introducción

personalizada, pensamiento computacional, gamificación y *b-Learning*, y contribuyan a la reducción del abandono escolar?

La respuesta a la pregunta planteada es relevante debido a su ausencia. La posibilidad de obtenerla es posible a través de los resultados empíricos generados de su intervención en los cursos iniciales de programación con los estudiantes de nuevo ingreso. Finalmente, además de lograr la adquisición de competencias y reducir el abandono escolar, se genera un aporte al conocimiento de la innovación educativa en educación superior.

1.5 Justificación de la investigación

La justificación de la investigación indica el porqué del estudio, exponiendo sus razones, necesidad e importancia. Particularmente para la investigación de la tesis doctoral se tienen los siguientes beneficios:

Conveniente, debido a que ayuda a resolver un problema, la baja escolar. Las instituciones de educación superior tienen entre otros indicadores un porcentaje de retención que permite conocer la cantidad de estudiantes que continúan con sus estudios, favoreciendo el prestigio de la institución misma y su valor en la generación de profesionales en la sociedad. El trabajo de investigación aporta de una forma nueva el aprendizaje enfocado en el estudiante para reducir la deserción.

Relevancia educativa, motivar a los estudiantes para conseguir competencias en el desarrollo de *Software* que impacta en egresados mejor capacitados para la industria. El aprendizaje significativo tiene un alto valor en los cursos formativos debido a que permite a los estudiantes adaptarse a las dinámicas tecnológicas de cambio en el desarrollo de aplicaciones para la industria.

Implicaciones prácticas, estrategias en el proceso enseñanza-aprendizaje para tener mejores resultados en el aula. Los resultados obtenidos en la aplicación de las estrategias de forma separada para cursos de programación en educación superior son aportaciones valiosas. La combinación de tales estrategias en intervenciones para una población específica, es decir, estudiantes de nuevo

ingreso, representa un beneficio práctico en el quehacer docente de muchas universidades a nivel global.

Valor teórico, se sugieren ideas, recomendaciones e hipótesis en la mejora del modelo educativo de las universidades tecnológicas, particularmente en lo referente al 70% práctica y 30% teórico, tales aportaciones pueden ser aplicables a otros modelos educativos similares al área de ciencias de la computación. Por otro lado, se pueden generar nuevas líneas de investigación a partir de los resultados obtenidos con la intención de poder generalizar el caso de estudio a contextos más amplios a los realizados con los grupos experimentales.

Utilidad metodológica, se hace uso de la metodología mixta, aportando un caso más de su empleabilidad. Específicamente se utilizó un diseño de triangulación concurrente que destaca por la inclusión de dos iteraciones de las fases definidas, es decir, las fases de planeación, intervención, análisis y discusión se ejecutaron en dos ocasiones, favoreciendo el resultado final del estudio donde se trianguló la información cuantitativa y cualitativa obtenida de forma simultánea.

1.6 Viabilidad del estudio

La factibilidad de la investigación está determinada por tres tipos de recursos: financieros, humanos y materiales. Respecto a los recursos financieros, no se requieren servicios de terceras personas para llevar a cabo el estudio, no hay necesidad de ejercer un gasto por el pago de algún trabajo, el investigador realizará todo el trabajo de investigación. Respecto a los recursos humanos, las intervenciones de las estrategias educativas se realizaron en dos ocasiones dentro del periodo de los estudios del programa de doctorado, es decir, en los cuatrimestres de septiembre – diciembre y enero – abril en correspondencia con los grados de primero y segundo del 2016, 2017 y 2018; durante el cuatrimestre de mayo – agosto se realizaron las actividades de planificación y creación de contenidos o ajustes para la siguiente intervención. Finalmente, existieron grupos de control y experimentales donde se aplicaron las estrategias educativas. Respecto a los materiales, no se requirió adquirir equipo

Introducción

de cómputo o dispositivos para realizar las estrategias, materiales tales como encuestas o contenidos para ambiente *b-Learning* son recursos digitales, excepto para el caso de gamificación, pero la creación del medallero no representó un problema de recurso material.

Hay que hacer notar que se tuvo acceso al lugar y contexto donde se desarrollaron las intervenciones sin ningún tipo de restricción, por lo que fue posible realizar la investigación en un tiempo determinado.

1.7 Evaluación de las deficiencias en el conocimiento del problema

Hernández et al. (2006) propone los siguientes cuestionamientos para abordar este punto del planteamiento del problema. ¿Qué más se necesita saber del problema? Debido a que se tomó como caso de estudio la problemática de la Universidad Tecnológica de Puebla, no se requirió más conocimiento de la situación respecto a los indicadores que sirvieron para justificar el desarrollo del estudio.

¿Qué no se ha considerado? Circunstancias personales de los estudiantes, es decir, la dimensión familiar, económica o laboral que tuvieron un impacto no registrado en los resultados de la investigación, así como eventualidades dentro de la universidad como suspensiones de actividades no controladas por el investigador. No se anticipa algún efecto negativo o perjudicial en las intervenciones, en los grupos experimentales y de control se verán los mismos temas y obtención de las mismas competencias.

1.8 Metodología

La metodología de la investigación ofrece un camino para llevar a cabo una investigación científica, así como las "actividades que el investigador debe realizar al desarrollar un estudio, con el propósito de producir conocimiento y teorías o resolver problemas prácticos" (Hernández, Fernández, & Baptista, 2006). La investigación científica se concibe como un conjunto de procesos sistemáticos y

empíricos que se aplican al estudio de un fenómeno; es dinámica, cambiante y evolutiva. Actualmente se utilizan tres enfoques: cuantitativo, cualitativo y mixto (Hernández, Fernández, & Baptista, 2006; Martínez, 2014).

El enfoque cuantitativo, cuyas características de utilizar estadística, medir fenómenos, emplear experimentación y el análisis causa-efecto, permite un proceso secuencial, deductivo y probatorio en la generación de resultados.

El enfoque cualitativo, que se conduce básicamente en ambientes naturales y los significados se extraen de los datos permite un proceso que da un contexto al fenómeno y profundidad de ideas.

El enfoque mixto (García-Peñalvo, Moreno-López, & Sánchez-Gómez, 2018) es un proceso que recolecta, analiza y vincula datos cuantitativos y cualitativos en una misma investigación para responder a un planteamiento del problema. Ofrece una visión mucho más completa de la realidad social, al recuperar los aspectos favorables del modelo cuantitativo y los integra con los del modelo cualitativo (Martínez, 2014); Rodríguez y Valdeoriola (2009) comentan también que una metodología mixta permite neutralizar o eliminar sesgos al usar solo el cuantitativo o cualitativo de forma aislada, permitiendo que los resultados de un método contribuyan al desarrollo del otro.

La dualidad cuantitativo – cualitativo para Sánchez-Gómez (2015) contempla a ambos enfoques como compatibles y complementarias, donde los dos pueden proporcionar información útil al integrar sus datos para producir un refuerzo de los resultados y conclusiones. Hernández et al. (2006) argumentan que el uso de un enfoque mixto o matrimonio cuantitativo-cualitativo es conveniente en un mismo estudio, solo si es el enfoque que mejor puede ayudar a responder las preguntas de investigación que se han establecido. Por lo anterior, un enfoque mixto es el que se lleva a cabo para lograr los objetivos. La investigación de la tesis doctoral generará nuevo conocimiento al ofrecer una solución integral en la adquisición de competencias básicas de programación a los estudiantes que estudian una ingeniería o licenciatura en alguna de las variaciones denominadas en relación al

Introducción

desarrollo de *software*; además de atender dos cuestiones básicas: "cómo es posible enseñar a los estudiantes a aprender con mayor eficacia y la medida en que es posible enseñar a todos los estudiantes a aprender con mayor potencia" (Ramírez, 2012).

Se han concebido diferentes diseños mixtos que combinan los enfoques cuantitativo y cualitativo. Creswell (2013), en la cuarta edición de su libro propone tres diseños de métodos mixtos: paralelos convergentes, secuencial explicativo, y secuencial exploratorio. Hernández et al. (2006) en la cuarta edición de su libro destacan cuatro: de dos etapas, de enfoque principal, en paralelo y complejos; en la quinta edición una nueva propuesta de cuatro diseños es presentada: concurrentes, secuenciales, de conversión y de integración (Hernández, Fernández, & Baptista, 2010) y se mantiene para la sexta edición (Hernández, Fernández, & Baptista, 2014). Lo anterior permite observar el desarrollo que aún tiene el enfoque mixto en comparación con el cualitativo y cuantitativo. Específicamente, para la investigación realizada, se utiliza como base un diseño de triangulación concurrente debido a las siguientes características:

- Se utiliza cuando el investigador pretende confirmar o corroborar resultados y efectuar validación cruzada entre datos cuantitativos y cualitativos, permitiendo aprovechar las ventajas de cada enfoque y minimizar sus debilidades.
- De manera concurrente (simultánea) se recolectan o analizan datos cuantitativos y cualitativos sobre el problema de investigación.
- Durante la discusión se terminan de explicar las dos clases de resultados.
- Representa un grado máximo de integración, debido a que el objetivo es el reconocimiento por parte de los dos enfoques del mismo aspecto de la realidad social para la convergencia o superposición de los resultados (García-Peñalvo, Moreno-López, & Sánchez-Gómez, 2018).

El estudio de investigación mixto basado en el diseño de triangulación concurrente, comprendió un proceso cuantitativo-CUAN y cualitativo-CUAL simultáneamente. La fase inicial de la metodología

de investigación fue la revisión sistemática de la literatura (RSL) con base a los conceptos: pensamiento computacional, educación personalizada, gamificación y *b-Learning*. Posteriormente se tiene la fase planeación donde se diseñaron los experimentos que aportarían datos para ambos enfoques considerando la problemática a resolver, en esta etapa se definieron las características de las intervenciones a realizar. La siguiente etapa correspondió a la fase intervención, es decir, la ejecución con los estudiantes de los diseños experimentales. La fase análisis destaca por la actividad de organizar los resultados generados de las intervenciones desde cada enfoque. Durante el trabajo de investigación fue posible realizar dos iteraciones de las intervenciones definidas, lo cual es muy conveniente dentro del modelo mixto, pues los resultados cuantitativos y cualitativos que se obtuvieron de forma concurrente de la primera iteración en un primer ejercicio de triangulación de la fase discusión, permitieron realizar ajustes en un segundo ciclo de la fase planeación e intervención; con la intención de mejorar los porcentajes de acreditación, calificación y baja escolar, en consecuencia, tener mejores resultados conclusivos. Después de la segunda iteración se llegó a la fase de triangulación final que aborda la discusión de resultados para comparar, confirmar, explicar y obtener el producto final del estudio. La Figura 3 ilustra el diseño metodológico descrito previamente.

En la investigación se utilizó una muestra no probabilística de estudiantes a los cuales se les aplicaron instrumentos tipo encuesta. El enfoque CUAN aportó los datos duros que midieron los resultados de evaluación del pensamiento computacional, la satisfacción de los escenarios de aprendizaje, uso del sistema gestor de aprendizaje, la adquisición de competencias y los índices de retención escolar, así como los resultados obtenidos de la experiencia de gamificación en tres dimensiones: cognitiva, emotiva y social. El enfoque CUAL se focalizó en los aspectos de motivación y satisfacción del entorno *b-learning* y gamificación, a partir de los grupos experimentales y de manera más específica para obtener los puntos de vista por parte de los estudiantes y observación del investigador. La combinación de los dos enfoques permitió complementar, explicar y reforzar la interpretación de los

Introducción

resultados con el objetivo de solucionar la problemática del estudio, establecer la información que se reportara en publicaciones y argumentar las conclusiones elaboradas para la tesis doctoral.

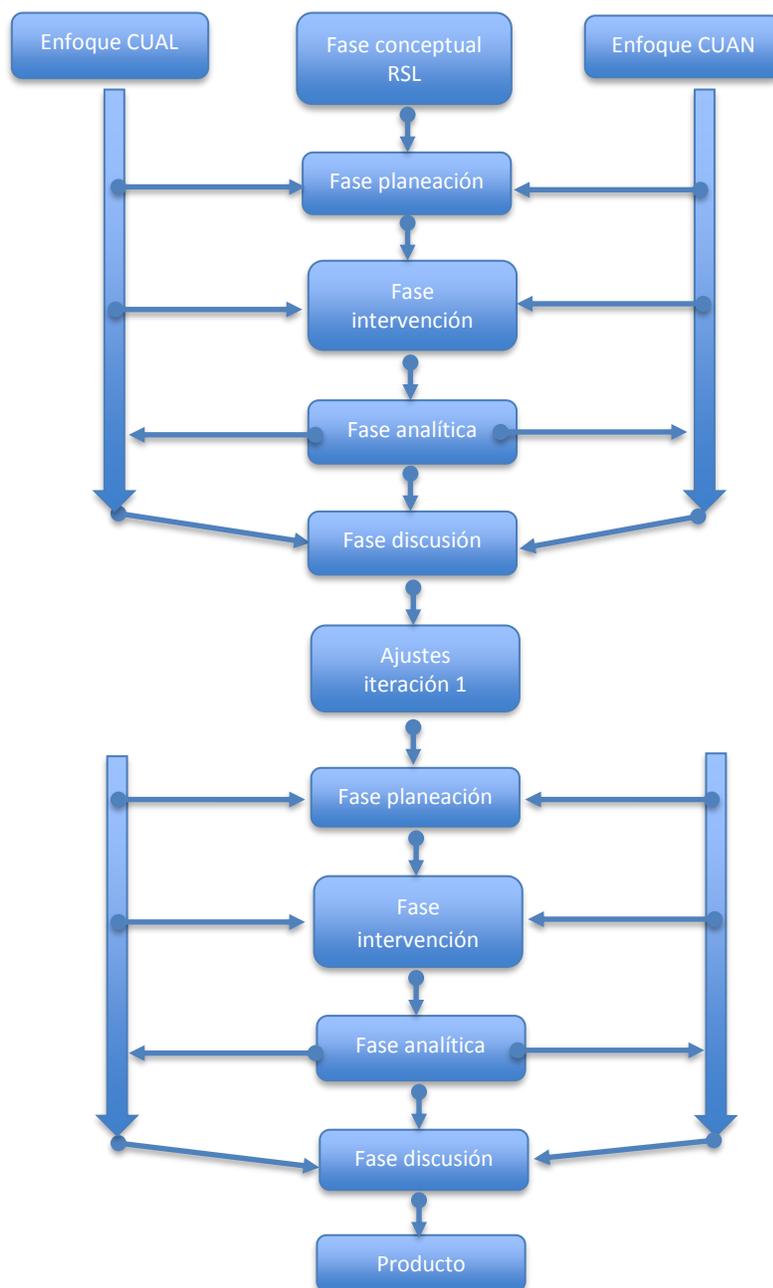


Figura 3. Diseño de investigación mixta tipo triangulación concurrente. Fuente: Elaboración propia

1.9 Población y muestra

Tomando en cuenta el caso de estudio, inicialmente se comentan los datos históricos que comprenden a la población de estudiantes que cursaron la materia de Metodología de la programación y Programación. El curso Metodología de la programación se imparte a los estudiantes de nuevo ingreso durante el primer cuatrimestre que corresponde al periodo septiembre – diciembre, hubo ocasiones en donde se ofreció también durante el periodo enero – abril; para el presente trabajo los datos de acreditación, deserción y calificación promedio del 2009 al 2016 dan un total de 3659 estudiantes. El curso de Programación, que se imparte a los estudiantes del segundo cuatrimestre y corresponde al periodo enero – abril, también hubo ocasiones que se ofreció para el periodo mayo – agosto, los datos de deserción, acreditación y calificación promedio que se obtuvieron del 2010 al 2017 dieron un total de 2633 estudiantes. La Tabla 3 contiene los valores de los datos históricos mencionados previamente, los cuales sirven para poder comparar los resultados de las intervenciones realizadas.

Tabla 3. Población de datos históricos. Fuente: Elaboración propia

Datos históricos	Población	Curso
2009 – 2016	3659	Metodología de la programación
2010 – 2017	2633	Programación

Aunque generalmente se realiza la investigación en una muestra de la población por economía de tiempo y recursos, en el caso de estudio fue posible que en los grupos experimentales seleccionados, se aplicara el diseño del experimento a todos los estudiantes, en consecuencia, se tuvo una muestra no probabilística o dirigida debido a las características de la investigación, pues la elección de los estudiantes no fue dependiente de la probabilidad; y fue exploratoria en el enfoque cualitativo debido a que permitió documentar la experiencia de las estrategias educativas de los estudiantes. Durante las intervenciones, los grupos de control cuando fue posible obtenerlos, se caracterizaron por ser clases presenciales en el aula o laboratorio, no utilizaron la plataforma *Moodle* como sistema de gestión de aprendizaje, y se realizaron evaluaciones en dos momentos indicados por la respectiva academia que

Introducción

podieron ser a través de examen escrito, prácticas de laboratorio, tareas y proyecto final; por lo general, el profesor del curso optó por usar el lenguaje de programación Java o C#.

Para el cuatrimestre septiembre – diciembre de 2016, los grupos experimentales, 1° C y 1° D fueron de 32 y 33 estudiantes en cada grupo, para el cuatrimestre enero – abril de 2017 el grupo experimental fue de 14 estudiantes del grupo 2° C, los grupos experimentales del cuatrimestre septiembre – diciembre de 2017 fueron el 1° B y 1° H con 35 estudiantes en cada grupo, finalmente, para el cuatrimestre enero – abril de 2018 el grupo experimental fue de 29 estudiantes del grupo 2° I. En cada grupo experimental el autor de este trabajo fue profesor. La Tabla 4 contiene la información descrita previamente.

Tabla 4. Datos de grupos de control y experimentales. Fuente: Elaboración propia

Cuatrimestre	Año	Grupos	
		Control	Experimental
Septiembre – diciembre	2016	165	65
	2017	0	70
Enero – abril	2017	0	14
	2018	31	29

1.10 Marco de trabajo

El programa de doctorado Formación en la Sociedad del Conocimiento (García-Peñalvo F. J., 2013; García-Peñalvo F. J., 2014; García-Peñalvo F. J., 2015), establecido dentro del Instituto Universitario de Ciencias de la Educación de la Universidad de Salamanca, es donde se desarrolla la presente tesis doctoral y ejemplifica claramente su vocación de presentar los procesos enseñanza-aprendizaje como auténticos motores de las sociedades del conocimiento. El enfoque interdisciplinario del programa permite cubrir las siguientes áreas de investigación:

- Evaluación educativa y orientación.
- Interacción y e-Learning.
- Investigación-Innovación en tecnología educativa.

- Medios de comunicación y educación.
- Medicina y educación.
- Robótica educativa.
- Ingeniería y educación.
- Educación y sociedad de la información

El programa es además soportado varios grupos de investigación reconocidos de la Universidad de Salamanca tales como VISUALMED, Robótica y Sociedad y E-LECTRA, destacando por su reconocimiento de excelencia de la junta de Castilla y León el GRupo de InterAcción y eLearning (GRIAL), GITE y OCA. Particularmente, el grupo de investigadores que pertenecen a GRIAL consolidan las siguientes líneas de investigación:

- Humanidades digitales
- Metodologías e-learning
- TIC e innovación educativa.
- Ciencias de la información
- Sistemas de aprendizaje interactivo
- Tecnologías de aprendizaje
- Calidad y evaluación en educación.
- Responsabilidad social e inclusión
- Gestión estratégica del conocimiento y la tecnología.
- Ecosistemas tecnológicos
- Analítica visual
- Ingeniería web y arquitectura de *software*

La producción científica de GRIAL tiene evidencias en los proyectos de investigación y las tesis doctorales dentro del programa de doctorado Formación en la Sociedad del Conocimiento (García-

Introducción

Holgado, 2018; Rincón-Flores E. G., 2018; González-Pérez, 2019). Por lo tanto, las actividades interdisciplinarias y multiculturales permiten enmarcar el trabajo de investigación dentro de la línea Interacción y *e-Learning*.

1.11 Estructura del documento

El documento se organiza en siete capítulos y un apartado de apéndices. El primer capítulo tiene la función de introducir al lector a la tesis doctoral, mientras que los seis restantes cumplen una función específica dentro de los tres bloques determinados: 1) Marco teórico, 2) Marco empírico, y 3) Discusión y conclusiones.

En el primer bloque, se presenta toda la investigación llevada a cabo para dar un contexto teórico, y en consecuencia justificar la resolución de emplear cuatro pilares estratégicos en la tesis doctoral.

El primer pilar lo representa la educación personalizada, debido a que la adquisición de competencias para la creación de *software* no es similar en cada persona, pues es un hecho que cada programador tiene su estilo, cada estudiante es individual en la forma de adquirir y hacer suyo el conocimiento, por lo tanto, hay que ofrecer opciones para que cada quien elija los medios que mejor le permitan lograr ser competente en la programación de computadoras. La sección 2.1 contiene la definición de la educación personalizada y resultados reportados de beneficios educativos.

El segundo pilar lo constituye el pensamiento computacional, para los estudiantes de nuevo ingreso que están conociendo un entorno diferente de aprendizaje, la intención es tener un diagnóstico inicial que permita personalizar su aprendizaje. La sección 2.2 incluye la definición del pensamiento computacional y algunas de las investigaciones que refuerzan su uso en la educación superior.

El tercer pilar lo aporta el *b-Learning*, la sección 2.3 contiene la explicación del concepto y algunos de los casos donde su aplicación demuestra que es una opción de modelo educativo vigente.

El cuarto pilar lo integra la gamificación, una estrategia que permite tener estudiantes motivados, en consecuencia, con menor deserción escolar; en la sección 2.4 se incluye la definición del concepto y los casos de éxito desde lo empresarial a lo educativo.

El capítulo dos contiene cada uno de los conceptos establecidos. El capítulo tres describe la actividad que permitió enriquecer de información el primer bloque, la revisión sistemática de la literatura.

En el segundo bloque, se describen las intervenciones realizadas empleando los cuatro pilares determinados.

El capítulo cuatro contiene la primera intervención del pensamiento computacional para ofertar escenarios de aprendizaje con la intención de personalizar el aprendizaje del curso Metodología de la programación, en este panorama se hace uso del *b-Learning* para crear un ambiente de aprendizaje híbrido. El capítulo contiene también el trabajo de planeación realizado durante una estancia en INACAP – Chile para replicar el diseño experimental realizado en México, agregando un valor a la investigación al constatar que es posible trasladar a otros contextos de educación superior lo elaborado en la tesis doctoral.

El capítulo cinco desarrolla la intervención de la gamificación para crear motivación en los estudiantes y lograr reducir la deserción escolar, además de procurar la adquisición de las competencias propias del curso de Programación, nuevamente se hace uso del *b-Learning* y el enfoque de educación personalizada.

En el tercer bloque, se encuentran dos capítulos de síntesis al trabajo de investigación.

El capítulo seis contiene una discusión de los resultados obtenidos en comparación con los objetivos determinados en la tesis doctoral.

El capítulo siete incluye las conclusiones de la tesis doctoral, destacando las aportaciones y contribuciones delineadas por la respuesta a la pregunta de investigación.

BLOQUE I: MARCO TEÓRICO

La elaboración del marco teórico usualmente comprende dos etapas:

1. La revisión de la literatura correspondiente, y
2. La adopción de una teoría o desarrollo de una perspectiva teórica (Hernández, Fernández, & Baptista, 2006).

La revisión de la literatura consiste en detectar, consultar y obtener materiales útiles para los propósitos del estudio, de los cuales se extrae y recopila información relevante y necesaria para el problema de investigación. El desarrollo de una perspectiva teórica establece un conjunto de conceptos interrelacionados, definiciones y proposiciones que presentan una visión sistemática de los fenómenos al especificar las relaciones entre variables, con el propósito de explicar y predecir los fenómenos (Hernández, Fernández, & Baptista, 2010). En este sentido, el capítulo 2 de la tesis doctoral describe los conceptos teóricos que permiten fortalecer la propuesta de investigación y el capítulo 3 la revisión sistemática realizada para determinar la ubicación de la investigación en el conjunto de materiales publicados y determinar el valor agregado al final del estudio realizado.

CAPÍTULO 2. CONCEPTOS TEÓRICOS

2.1 Educación personalizada

La educación personalizada ayuda a cada estudiante reconociendo la individualidad de aprendizaje, para que el docente adapte la metodología de enseñanza, entendiendo que cada estudiante es singular como persona y aprende de modos y a ritmos distintos, ya sea presencial o en línea. Existe la confusión de identificar a la educación personalizada con la teoría de las inteligencias múltiples (Punset, 2011; Magro-Mazo, 2015), y debido a que esta considera que el estudiante no posee una inteligencia única, fomentando la construcción de aprendizaje a través de diferentes vías; ocho son las inteligencias propuestas: Inteligencia lingüística, inteligencia lógico-matemática, inteligencia espacial, inteligencia musical, inteligencia corporal y cinestésica, inteligencia intrapersonal, inteligencia interpersonal e inteligencia naturalista. Generalmente en los primeros niveles educativos se destaca ofrecer contenidos y procedimientos enfocados en enseñar y evaluar los dos primeros tipos de inteligencia: Lingüística y lógico-matemática, sin embargo, resulta insuficiente en el proyecto formativo de educar a personas en su respectivo contexto social. A partir de definir inteligencia como la capacidad de solucionar problemas o elaborar bienes valiosos, la obtención tradicional de inteligencia académica ha cambiado con la integración de las TIC en el actual proceso de aprendizaje. No cabe duda que la vida humana requiere del desarrollo de varios tipos de inteligencia, pero también atender la individualidad, en consecuencia, se justifica el uso de la educación personalizada como una estrategia educativa empleada en esta tesis doctoral. En este sentido, se inicia ofreciendo una definición del concepto y la descripción de algunos escenarios donde se ha implementado para destacar la aportación del trabajo realizado.

Definición de educación personalizada

A partir del concepto de educación personalizada, ofrecido por Bernardo et al. (2011), como una concepción educativa ajena a un modo de entender la educación enmarcado en alguna corriente

Conceptos teóricos

filosófica, psicológica o pedagógica concreta, sino que está abierta a todas las corrientes razonables de pensamiento que contribuyen a la formación de toda la persona; las intervenciones presentadas en el marco empírico de la tesis doctoral, buscan atender lo que los estudiantes tienen en común y lo que tienen de propio dentro del contexto académico de los cursos donde se aplicaron las estrategias educativas, a través de un sistema gestor de aprendizaje y la oferta de opciones se logró estimular a cada estudiante para que vaya perfeccionando libre y responsablemente la capacidad de dirigir su propio conocimiento. La educación personalizada (Ferraro, Álvarez, & Peñalvo, 2004; Berlanga & García, 2005; Berlanga & García, 2005; Berlanga & García, 2008; Lerís & Sein-Echaluce, 2011) y su relación con las TIC ofrece un enfoque que permite “ayudar a cada ser humano a establecer y mantener vínculos valiosos con la realidad” (Calderero, Aguirre, Castellanos, Peris, & Perochena, 2014) y para Gao (2014) tomará el lugar de la educación estándar. Cambiar de una educación tradicional hacia una personalizada, no es una estrategia aislada, así lo buscó también el trabajo de Hart (2016) que propuso una “adaptación de la educación considerando las características específicas de cada estudiante”. El trabajo de investigación de Sadovaya et al. (2016) comenta que “es necesario desarrollar un nuevo modelo de estrategias educativas basadas en los principios de la personificación, el diálogo, la subjetividad, el enfoque individual y la complementariedad”. Tekin et al. (2015) enmarcaron su investigación dentro de los sistemas educativos basados en la Web, destacando su uso como un complemento “y, en algunos casos, en alternativas viables a la enseñanza tradicional en el aula”; en comparación a los cursos masivos y abiertos en línea (Massive Open Online Course – MOOC) resaltaron la observación de que “continúan siendo de una sola talla para todos”, así propuso un “método sistemático para diseñar un sistema educativo personalizado basado en la Web”, el trabajo anterior destaca al comentar tres desafíos al momento de personalizar la educación, tales desafíos son: “(i) los estudiantes deben recibir enseñanza y capacitación personalizada según sus contextos (por ejemplo, clases ya tomadas, métodos de aprendizaje preferidos, etc.), (ii) para cada contexto específico, el mejor método de enseñanza y capacitación (por ejemplo, el tipo y orden de los materiales de enseñanza que se mostrarán) debe ser aprendido, (iii) la enseñanza y la capacitación

deben adaptarse en línea, en función de los puntajes / retroalimentación (por ejemplo, pruebas, cuestionarios, examen final, gustos / disgustos, etc.) de los estudiantes”. El trabajo de Laksitowening y Hasibuan (2015) aporta también otros componentes que “juegan un papel importante en la personalización: estructura de aprendizaje integrada, modelo de aprendizaje, escenario de aprendizaje personalizado, selección de contenido personalizado y evaluación basada en evidencias” que les permitió proponer una arquitectura de *e-Learning* personalizada que cumple con los estándares que hacen evidente el aprendizaje de competencias. Otro enfoque a considerar es el trabajo de Tejeda-Lorente et al. (2015) donde a partir de las metas educativas del proceso de Bolonia, se presenta un “sistema de recomendación para proporcionar actividades personalizadas a los estudiantes para reforzar su educación individualizada” y sirve como ayuda a los profesores para proporcionar a los estudiantes un seguimiento personalizado de sus estudios. En el ámbito de las tecnologías móviles personales, Kucirkova y Littleton (2017) argumentan que para integrarse en las escuelas, estas deben integrar la “noción de que el aprendizaje de los niños necesita ser adaptado a las aspiraciones de los individuos (es decir, personalizado) y participativo, implicando la consideración de múltiples perspectivas (es decir, pluralizadas)”; así, enfatizando el papel vital que desempeñan los educadores, comenta que “la personalización y pluralización deben conceptualizarse como fuerzas complementarias dentro de la reforma educativa del siglo XXI”, como lo elabora Humanante et al. (2015), que presentan la conceptualización y estructura de un entorno de aprendizaje personal móvil teniendo “al estudiante como el centro del entorno”.

La educación superior en México no está exenta de aspirar a la implementación de la educación personalizada, pero es una acción que no puede aplicarse en las clases tradicionales cara a cara donde los estudiantes no pueden ser tratados individualmente como lo expone Kostolányová (2017), sino que debe auxiliarse el profesor en tecnologías web o *e-Learning*. En la educación pública “algunos estudiantes pueden verse obstaculizados y aburridos por la enseñanza colectiva”. Kostalányová expone que “la individualización de la instrucción refleja el estilo de aprendizaje, las habilidades y el

Conceptos teóricos

conocimiento ya adquirido de cada estudiante. En el momento de la informatización de la sociedad, Internet y los instrumentos adecuados de *software* y *hardware*, la instrucción basada en computadora puede realizarse fácilmente”. Así, se tiene un caso de éxito reportado por Zhao (2016) en ingeniería química donde propone un método de tres etapas para mejorar la educación personalizada a lo largo de la carrera, no solo en una asignatura: “(1) equipamiento de sistema tutorial profesional con maestros de orientación profesional a estudiantes de primer año para orientar sus actividades de aprendizaje y proporcionar orientación profesional; (2) abrir proyecto experimental: establecer proyectos experimentales abiertos a estudiantes de segundo y tercer año para elegir libremente; (3) módulo de educación individualizado: configuración de 10 módulos de educación individualizados diferentes para estudiantes de último año a seleccionar”. Un caso similar, pues se enfoca en la enseñanza del lenguaje de programación ‘C’, es el trabajo de Chrysafiadi et al. (2015) que reportan la identificación de las necesidades individuales de cada estudiante para completar el programa de entrenamiento en su propio ritmo y habilidades de aprendizaje, a lo que denomina el modelo del estudiante; que a la par del trabajo de Chrysafiadi et al. (2015) se detalla qué modelar, cómo y por qué. Barba y Chancellor (2015), toman en cuenta a estudiantes de edad mayor que están interesados en aprender ciencias de la computación debido a la competitividad laboral, pero los métodos tradicionales no serían los adecuados para dicho aprendizaje, en consecuencia, en su propuesta toman en cuenta el estilo de vida de las personas y proponen opciones de cómo elegir la plataforma de aprendizaje, la estructura de contenidos, mantener la motivación del estudiante, y proyectos finales autoguiados. Otros trabajos de investigación que contribuyen al uso de la educación personalizada son Sun et al. (2016) y Morrowy et al. (2016), en el primero se proponen cinco actividades de aprendizaje de visualización aplicadas al aprendizaje de conceptos geométricos de matemáticas elementales con el objetivo de cultivar las habilidades del estudiante en el aprendizaje independiente; en el segundo se proponen decisiones algorítmicas que faciliten la recomendación de los horarios de cursos personalizados según los antecedentes y los intereses de un estudiante dado. Finalmente, asumiendo la importancia de la educación personalizada Villegas et al. (2017) desarrollan técnicas

de minería de datos aplicadas a los sistemas de gestión del aprendizaje, particularmente *Moodle*, para proporcionar información útil a los profesores en el objetivo de ofrecer educación adaptada a las necesidades de los estudiantes.

2.2 Pensamiento computacional

El pensamiento computacional (Wing J. M., 2006; Selby, 2015) es un proceso cognitivo que permite la generación de soluciones a problemas a través del uso de habilidades específicas, tales como abstracción, descomposición, generalización, evaluación y diseño algorítmico. Actualmente existe un gran interés por enseñar el pensamiento computacional a una edad temprana (García-Peñalvo & Mendes, 2018; Chiazzese, Fulantelli, Pipitone, & Taibi, 2018; Chiazzese, Fulantelli, Pipitone, & Taibi, 2017; García-Peñalvo F. J., Reimann, Tuul, Rees, & Jormanainen, 2016), debido a que la resolución de problemas es una competencia útil para las personas independientemente de su perfil profesional; la aplicación del pensamiento computacional en niños sirve también para detectar talento afín a las áreas de ciencia, tecnología, ingeniería y matemáticas, además, particularmente se considera un elemento clave para el éxito de estudiantes en las ciencias de la computación. Por lo anterior, es útil conocer el nivel cognitivo del estudiante en pensamiento computacional a través de una evaluación que permita obtener no solo qué habilidades posee en la resolución de problemas, sino además contribuir en la creación del ambiente de aprendizaje creado por el docente para favorecer el proceso enseñanza – aprendizaje en las asignaturas iniciales de programación.

Definición de pensamiento computacional

El término pensamiento computacional fue hecho popular por Jeannette M. Wing, en su definición “involucra la solución de problemas, diseño de sistemas, y comprensión de la conducta humana, basándose en los conceptos fundamentales de las ciencias de la computación” (Wing J. M., 2006). Wing revisó el concepto y proporcionó una nueva definición: “el pensamiento computacional es el proceso de pensamiento involucrado en la formulación de problemas y sus soluciones para que las

Conceptos teóricos

soluciones estén representadas en una forma que pueda ser efectivamente llevada a cabo por un agente de procesamiento de información” (Wing J. M., 2011). García-Peñalvo lo define como “la aplicación de un alto nivel de abstracción y un enfoque algorítmico para resolver cualquier tipo de problema” (García-Peñalvo F. J., 2016). *Google for Education* (2016) lo establece como un conjunto de habilidades y técnicas para “el proceso de solución de un problema, tales como, ordenamiento lógico, análisis de datos y creación de soluciones usando una serie de pasos ordenados que en consecuencia, es esencial para el desarrollo de aplicaciones computacionales”. En el documento elaborado por García-Peñalvo et al. (2016) se presentan definiciones que ejemplifican la falta de consenso para establecer una definición formal del término pensamiento computacional, pero a pesar de eso, destacan las siguientes dos:

“El pensamiento computacional es el proceso de reconocer aspectos de computación en el mundo que nos rodea, y aplicar herramientas y técnicas desde la ciencia de la computación para comprender y razonar acerca de sistemas y procesos artificiales y naturales” (Royal Society, 2012).

“El pensamiento computacional es un proceso de resolución de problemas que incluye (pero no está limitado a) las siguientes características: formulación de problemas de una forma que nos permita usar una computadora y otras herramientas para ayudar a resolverlos; organización lógica y análisis de datos; representación de datos a través de abstracciones tales como modelos y simulaciones; soluciones automáticas a través de pensamiento algorítmico (una serie de pasos ordenados); identificar, analizar e implementar posibles soluciones con la meta de lograr la combinación más eficiente y efectiva de pasos y recursos; generalizar y transferir este proceso de solución de problemas a una amplia variedad de problemas” (Barr & Stephenson, 2011) .

Xia (2016) comenta la definición propuesta por Zhou Yizhen del 2006: es el uso de los conceptos fundamentales de ciencias de la computación para la resolución de problemas, comprensión de la conducta humana además de actividades de pensamiento en el diseño de sistemas.

Zapata-Ros (2015) elaboró un estudio con la intención de contextualizar la definición del pensamiento computacional y la relación de habilidades asociadas, desde un análisis y de una elaboración interdisciplinar en las teorías del aprendizaje. Como resultado, establece los siguientes componentes:

- Análisis ascendente. En ocasiones, la creación de un algoritmo implica un proceso de diseño de submétodos funcionales que permiten determinar un método general de resolución.
- Análisis descendente. La resolución de un problema complejo se puede llevar a cabo a partir de resolver primero los problemas más concretos para pasar después a resolver los más abstractos.
- Heurística. Conjunto de procedimientos prácticos o informales que se obtienen de la observación, el análisis y el registro para resolver problemas.
- Pensamiento divergente. Pensamiento ajeno a un esquema rígido de pensar y de formular ideas en el aprendizaje, con el objetivo de obtener ideas creativas e innovadoras. Produce soluciones no convencionales e implica fluidez y capacidad para generar una gran cantidad de visiones e ideas sobre el problema que se trabaja, para cambiar de unas a otras, y para establecer asociaciones inusuales.
- Creatividad. Amplia variedad de comportamientos, hábitos e ideas que permiten converger el pensamiento convergente (estructurar los conocimientos de una forma lógica y para aplicar sus leyes) y divergente para producir una novedad en un campo reconocible por una sociedad.
- Resolución de problemas. El pensamiento computacional es una variante del dominio metodológico que se conoce como resolución de problemas.
- Pensamiento abstracto. Es la capacidad de operar con modelos ideales abstractos de la realidad, abstrayendo las propiedades de los objetos que son relevantes para un estudio.
- Recursividad. Metodología que permite resolver un problema con la característica de poder resolverse a partir de una remisión a otro problema de las mismas características o naturaleza, pero más pequeño que sí puede ser resuelto.

Conceptos teóricos

- Iteración. Componente importante del pensamiento computacional que permite crear procedimientos repetitivos, con una extensa proyección en otras representaciones cognitivas y en procedimientos complejos que son la base de importantes actividades y tareas en la resolución de problemas.
- Métodos por aproximaciones sucesivas. Ensayo – error. Método de resolución de problemas en donde se confrontan las ideas formadas de la realidad, a través de los sentidos, la experimentación y la representación de las ideas obtenidas de las experiencias, para aceptar o rechazar el conocimiento que la realidad ofrece e inducirlo.
- Métodos colaborativos. De una aproximación inicial de entender el trabajo colaborativo como el que se produce en una situación en la que dos o mas personas aprenden algo juntos, se nutre el concepto para llevarlo a superar el aprendizaje para trabajar juntos hacia encontrar una cultura en común, unas referencias y unas experiencias que hagan que esa forma de trabajar fluya.
- Patrones. Desde el punto de vista del análisis de la programación, los patrones tienen la facultad útil de evitar el trabajo tedioso que supone repetir procedimientos que en esencia se repiten, pero aplicados a contextos y situaciones distintas; lo que exige la capacidad de distinguir lo que tienen de común situaciones distintas.
- Sinéctica. Estudio enfocado en organizar la integración de diferentes individuos que componen un grupo para la resolución de problemas. Es un punto de confluencia de las teorías que tratan de explicar y estudian la creatividad, de las técnicas de trabajo en grupo como medio para exteriorizar flujos e impulsos que de otra forma no serían observables y por tanto analizados, mejorados y compartidos.
- Metacognición. Condición necesaria en combinar la situación y los recursos cognitivos propios de las personas para que se lleve a cabo un aprendizaje; se trata de un plan de actuación que implica habilidades y destrezas (que la persona ha de poseer previamente) y de una serie de técnicas que se aplican en función de las tareas a desarrollar para lograr el

aprendizaje, sobre las que la persona decide y sobre las que tiene una intención de utilizar de forma consciente. En relación al pensamiento computacional adquiere una importancia en la tarea de cómo afrontar un problema y cómo resolverlo.

La definición que sirve de apoyo al trabajo de investigación es la propuesta por Selby (2015) que incluye las habilidades de abstracción, descomposición, diseño algorítmico, generalización y evaluación. El estudio detona un elemento muy importante debido a que explora la relación entre el pensamiento computacional, enseñar programación y la taxonomía de Bloom. Inicialmente desarrolla una definición de las cinco habilidades.

- Generalización: la habilidad para expresar la solución de un problema en términos genéricos, la cual pueda ser aplicada a diferentes problemas que comparten algunas de las mismas características como el problema original.
- Descomposición: fraccionar a piezas más pequeñas, fáciles de resolver, partes de un problema.
- Abstracción: habilidad para decidir qué detalles de un problema son importantes y qué detalles se pueden omitir.
- Diseño Algorítmico: habilidad para crear un conjunto de instrucciones que indiquen paso a paso la solución de un problema para un dispositivo.
- Evaluación: habilidad para reconocer y determinar los alcances de realizar procesos, en términos de eficiencia y uso de recursos.

Posteriormente, mapea el dominio cognitivo de la taxonomía de Bloom para concentrarla de la siguiente forma: en el nivel de Aplicación la habilidad de Generalización, en el nivel de Análisis las habilidades de Abstracción y Descomposición, en el nivel de Síntesis la habilidad de Diseño algorítmico y en el nivel de Evaluación corresponde la habilidad del mismo nombre. Finalmente,

Conceptos teóricos

relaciona conocimiento de Programación (tipos de datos, análisis de un problema, diseño algorítmico, creación de programas, ejecución y evaluación de resultados) con el mapeo descrito anteriormente.

Debido a que el trabajo de investigación está enfocado a la educación superior, el perfil de las investigaciones publicadas acerca del pensamiento computacional puede ser dividido en dos grupos, aquellas que van dirigidos a los niveles educativos preuniversitarios y a los que corresponden al nivel educativo superior o universitario.

Preuniversitario.

El interés por enseñar el pensamiento computacional a una edad temprana es ejemplificado a través del documento *Standards in Computer Science* en Norte América (CSTA, 2011), que ha sido fuente de referencia y guía para las propuestas curriculares de los profesores. Un segundo documento que expone la importancia del tema es el titulado *Computer our Future* concentrando los esfuerzos que hace la Comunidad Europea por promover la codificación entre niños y adolescentes (European Schoolnet, 2015). TACCLE 3 Coding, tomando como referencia que los planes de estudio en Europa incluyen el pensamiento computacional en estudios preuniversitarios, ofrece recursos a los profesores para que en las aulas ejerciten la lógica, algoritmos, el control de cosas y la creación y depuración de programas en tres niveles, principiantes, medio y avanzados (TACCLE 3 Consortium, 2017; García-Peñalvo F. J., 2016; García-Peñalvo F. J., 2016; García-Peñalvo, Reimann, & Maday, 2018). En consecuencia, el impacto es importante, así los autores James y George (2009) lo consideran un elemento clave para el éxito de los estudiantes en ciencias de la computación debido a que enfatiza la abstracción y la automatización (Park, Hyun, & Heuilan, 2015).

Sáez-López et al. (2016), muestran los resultados favorables de enseñar conceptos de computación en los primeros años escolares debido a la motivación que causa el entorno *Scratch*, para muchos profesores es una herramienta nueva y representa un reto aprender a usarlo e integrarlo en el aula, pero altamente justificable el esfuerzo debido a la aceptación de los niños, considerando que para

ellos también es reto a vencer (Sentance & Csizmadia, 2017; Wong, Cheung, Ching, & Huen, 2015; Saari, Blanchfield, & Hopkins, 2016; Saari, Blanchfield, & Hopkins, 2015; Byrne, Fisher, & Tangney, 2015; Zeng, 2015; Bustillo & Garaizar, 2015; Calao, Moreno-León, Correa, & Robles, 2015; Dorling & White, 2015). Otras Instituciones y profesores han usado entornos alternativos a *Scratch* porque coinciden en los beneficios que aportan en los estudiantes (Cheng, Fu, & Chen, 2016; Worrell, Brand, & Repenning, 2015; Jenkins, 2015; Berland & Wilensky, 2015; Han, Kim, & Wohn, 2015; Sarmiento, Reis, Zaramella, Almeida, & Tacla, 2015) e inclusive han propuesto contenidos temáticos o cursos para niños de 11-12 años (Duncan & Bell, 2015; Hui-Chi, Chiu-Fan, Cheng-Chih, & Yu-Tzu, 2015; Yadav, Mayfield, Zhou, Hambruch, & Korb, 2014) y la organización de torneos abiertos para niños de hasta 14 años indica una confianza global por parte de las Instituciones educativas (UK Bebras Computational Thinking Challenge, 2015; Talent Search, 2015). El pensamiento computacional en estudios preuniversitarios es una herramienta que, en combinación con otras metodologías, los maestros tienen para crear escenarios y actividades de aprendizaje que son efectivas para ayudar a los jóvenes a resolver problemas mediante el uso de la tecnología con que viven a diario (García-Peñalvo F. J., 2018; Llorens-Largo, García-Peñalvo, Molero-Prieto, & Vendrell-Vidal, 2017). Destaca el trabajo de Repenning et al. (2015) respecto a la cantidad de estudiantes que participaron, 10 000, con el objetivo de motivarlos para engancharlos en el pensamiento computacional a través de su método de diseño de juegos y simulaciones. Zhong et al. (2016) consideran al pensamiento computacional una habilidad fundamental, por tal razón presentan una propuesta para evaluarlo en el nivel educativo básico.

Universitario.

Resulta de interés que en general el pensamiento computacional apoye a los estudiantes no importando su perfil profesional, pero para el caso de género, Hamlin et al. (2010) reportan en su estudio, acerca de resolver problemas de ingeniería usando herramientas de computación y programación a través de un ejercicio de hoja de cálculo, que las “mujeres subestimaron sus

Conceptos teóricos

habilidades mientras que los hombres sobreestimaron sus propias habilidades”, aunque en general, no hay diferencia entre hombres y mujeres para las habilidades del pensamiento computacional, la elección de la profesión es la que hace parecer que el género masculino predomina en las áreas de la computación (Espino & González, 2015).

El pensamiento computacional representa una propuesta adecuada para fomentar el aprendizaje de habilidades que beneficien a los estudiantes que ingresan en las áreas de ciencia, tecnología, ingeniería y matemáticas (Science, Technology, Engineering, and Mathematics - STEM) (Swaid, 2015; Lye & Koh, 2014; García-Peñalvo & Cruz-Benito, 2016; García-Peñalvo F. J., 2016; García-Peñalvo F. J., 2016; García-Peñalvo F. J., Reimann, Tuul, Rees, & Jormanainen, 2016). Representa el “foco de la innovación educativa por el conjunto de habilidades de solución de problemas que debe ser adquirido por las nuevas generaciones de estudiantes” y su evaluación es un tema de investigación que ha permitido la creación de pruebas en educación superior (Román, Pérez, & Jiménez, 2015), vinculación con el aprendizaje de la programación y la taxonomía de Bloom (Selby, 2015), y ha servido de forma efectiva para determinar escenarios de aprendizaje (Rojas-López & García-Peñalvo, 2016). Las habilidades pueden ser aprendidas por medio de un modelo de juego y a su vez introducir conceptos de programación que eviten la “frustración y la demanda de actividad de los estudiantes” (Kazimoglu, Kiernan, Bacon, & MacKinnon, 2012) sin olvidar que el pensamiento computacional no es sinónimo de programación. Trabajos que relacionan el pensamiento computacional con el estudio inicial de programación de computadoras, tal como Larkins y Harvey (2010) consideran no solo a estudiantes del área de ciencias de la computación sino aquellos que vienen de una variedad de ciencias e ingenierías para solucionar problemas en sus áreas, inclusive a nivel doctoral; para el caso específico de ciencias de la computación Lingling et al. (2015) comentan que se tiene que ir más allá de “entrenar a los estudiantes para solucionar problemas usando un lenguaje de programación en práctica”, se tiene que buscar su motivación pues “es un factor importante para mejorar el desempeño del estudiante” (Ramirez-Lopez & Muñoz, 2015). Walker (2015) defiende el hecho de que usar un

lenguaje de programación ofrece precisión de las propuestas de los estudiantes para mejorar sus habilidades de pensamiento computacional para analizar una correcta solución y comparar soluciones alternativas, pues existe el problema de la adquisición de las competencias básicas para aprender la programación de computadoras y el desarrollo de *software* en general; el motivo principal “desde el punto de vista del estudiante es que no se tienen las capacidades mentales adecuadas para la solución de problemas” (Olivares, Jiménez, Ortiz, & Rodríguez, 2015). Particularmente, existen dos trabajos realizados por Czerkowski et al. (2015) y Weese (2016). El primero se propone revisar el estado actual del campo en educación superior y discutir si las habilidades del pensamiento computacional son relevantes fuera de los campos de la ciencia, tecnología, ingeniería y matemáticas (STEM- Science, Technology, Engineering and Mathematics). El segundo confirma el interés en el pensamiento computacional para el desarrollo curricular universitario de programación visual. “Es imperativo desarrollar las habilidades del pensamiento computacional entre los docentes y estudiantes de las áreas STEM para sostener la revolución científica” (Swaid, 2015). Así, el trabajo de la aplicación del pensamiento computacional en la educación superior sigue siendo fuente de investigación para diferentes aristas educativas, no exclusivas de las ciencias de la computación (Shiflet & Shifleta, 2012; Chilana, y otros, 2015). Específicamente, para el caso de usar pensamiento computacional en programación de computadoras en la educación superior, existen las siguientes propuestas concretas.

Romero et al. (2017) inicialmente establecieron el impacto que tiene la creatividad para la creación de soluciones de codificación, lo que denominaron programación creativa, permitiendo enmarcar el pensamiento computacional en dicho tipo de programación. Así, determinaron una relación de componentes del pensamiento computacional con las fases de la solución de problemas colaborativos que indica el programa para la evaluación internacional del estudiante. Posteriormente, propusieron una metodología para evaluar el pensamiento computacional en las actividades de la programación creativa. Los resultados sugirieron la necesidad de una evaluación humana de la programación

Conceptos teóricos

creativa al tiempo que señalan los límites de la herramienta analítica automatizada usada, debido a que no refleja la diversidad creativa de los proyectos entregados por los estudiantes en el entorno de *Scratch* y anula la complejidad algorítmica. Yinnan y Chaosheng (2012) también comentan en su trabajo el uso del pensamiento computacional para incrementar la capacidad creativa de los estudiantes. En su propuesta organizan el contenido de enseñanza y elaboran experimentos para entrenar la capacidad de pensamiento durante un curso de programación. Zhang et al. (2011) también trabajan con la adecuación de currículo en el curso de programación *Visual Basic*. El impacto que tiene el entrenamiento del pensamiento computacional de los estudiantes recae en la intención de activar el análisis requerido en la solución de problemas que a su vez permita innovar en la programación de computadoras. Chen (2017) también determina una relación de habilidades del pensamiento computacional y el conocimiento de un lenguaje de programación, sin especificar la plataforma, para mejorar el desempeño de los estudiantes. Hace uso de la metodología de aprendizaje basada en problemas para reforzar el material teórico. Zhi-Mei y Xiang (2016) también hacen uso de la misma metodología, pero en primera estancia para mejorar el entrenamiento del pensamiento computacional a través de una secuencia de actividades, y en consecuencia, que los estudiantes mejoren su capacidad durante el proceso de resolución de problemas y finalmente impacte en el aprendizaje del lenguaje de programación C. Por otro lado, Ni (2017) enfoca su trabajo de investigación para el aprendizaje del lenguaje de programación C++. Promueve el pensamiento computacional de los estudiantes y usa la metodología de aprendizaje basada en casos reales. Xia (2016) hace uso de otras filosofías de enseñanza avanzadas, como así las denomina, para que el entrenamiento del pensamiento computacional favorezca las habilidades del estudiante y su proceso de aprendizaje de programación. Al igual que los anteriores trabajos de investigación, busca complementar la ejercitación del pensamiento computacional. La enseñanza por casos, el manejo de tareas, micro lecciones y MOOC, representan a tales filosofías.

Gao (2014) basa su investigación en dos habilidades del pensamiento computacional: Abstracción y automatización. En sus resultados, indica que a través del pensamiento computacional es posible resolver la contradicción entre enseñanza y aprendizaje en la resolución de problemas durante un curso de programación en C. Los estudiantes deben modelar adecuadamente para mejorar la efectividad del aprendizaje, analizar el modelo, verificar en un escenario práctico, y complementar con investigación, así el estudiante tiene el rol principal en el proceso de aprendizaje y el docente es un líder.

Huang et al. (2009) sin especificar que habilidad del pensamiento computacional trabajan con los estudiantes, proponen cultivar el pensamiento computacional para beneficiar la solución de problemas y el diseño algorítmico en el contexto de la programación orientada a objetos.

Ying y Pingping (2017) en su investigación comentan clases específicas de pensamiento computacional para entrenar a los estudiantes, con la intención de que descubran, analicen, resuelvan y generalicen problemas. En el proceso de enseñanza que proponen, los estudiantes deberán de producir diferentes mecanismos algorítmicos para resolver el mismo problema durante las clases presenciales. Posteriormente, se pasa a la etapa de codificación y poder obtener resultados de programación. Con la anterior estrategia, buscan resolver la problemática planteada: “altas calificaciones y bajas habilidades”.

Compañ-Rosique et al. (2015) enfocan su trabajo de investigación desde otra perspectiva. El desarrollo de habilidades de pensamiento computacional para la resolución de problemas, así como el aprendizaje de la programación de computadoras sigue siendo el objetivo, pero la estrategia que proponen para conseguirlo es diferente. Establecen tres etapas para conseguir el aprendizaje: oír, ver y hacer, pero reelaboran su significado al momento de realizar la intervención docente, es decir, apelan a escuchar, observar y practicar.

Conceptos teóricos

Michaelson (2015) elabora una pedagogía para el aprendizaje de la programación a través del uso del pensamiento computacional. Las habilidades de descomposición, abstracción, reconocimiento de patrones y diseño algorítmico, las considera como un conjunto de etapas que se empalman e interactúan para generar la resolución de problemas, calificándola como una actividad creativa. Así, argumenta que a través de la información se debe estructurar la computación, es decir, iniciar con una concreta instancia del escenario del problema para que al usar el pensamiento computacional, el estudiante elabore buenas preguntas que le permitan desenredar estructuras de información conocidas y en consecuencia guiar el diseño de la computación.

2.3 b-Learning

En los niveles educativos preuniversitarios de educación pública, el profesor con base a los planes de estudio establece y determina los objetivos, tareas y tiempo para su ejecución; en la mayoría de los casos bajo una supervisión y corrección presencial, es decir, una formación que se imparte físicamente en un espacio, el estudiante y docente interrelacionan directamente sin el uso de algún apoyo tecnológico (García-Peñalvo F. J., 2015), todo en el aula y bajo la acción del profesor; así, en consecuencia los estudiantes se limitan a ejecutar la secuencia didáctica establecida por el docente, de modo que en el nivel universitario carecen de una planificación y gestión del tiempo personal, pero no es el único problema que enfrentan los jóvenes de nuevo ingreso, además de los requerimientos administrativos de la Institución y los distractores sociales de la vida universitaria, pueden experimentar un cambio radical en lo que se refiere a los estilos de enseñanza de cada docente, haciendo uso de competencias digitales y formalizando escenarios de aprendizaje que son nuevos para los estudiantes. *Moodle* representa una herramienta con una muy buena aceptación en el ambiente académico que sirve principalmente de auxiliar en la comunicación entre el profesor y el estudiante. Por lo anterior, el presente apartado establece el concepto de *b-Learning* y algunos resultados de investigación que lo han usado en la enseñanza de programación de computadoras, contribuyendo a fortalecer el ambiente de aprendizaje en el que está basado la tesis doctoral.

Definición de b-Learning

La modalidad de aprendizaje no presencial, permitió la distribución de información e instrucción haciendo uso de una variedad de medios electrónicos tales como emisión satelital, audio-video en cintas o CD-DVD, pero principalmente Internet, lo que generó el concepto de aprendizaje electrónico *e-Learning* (*electronic learning*). Con Internet se dio un nuevo avance en el desarrollo de procesos de enseñanza - aprendizaje no presenciales con la combinación de servicios síncronos y asíncronos, no como sustituto de la formación presencial tradicional, sino más como un complemento que se ha de adaptar según las necesidades y nivel de madurez del público receptor (García-Peñalvo F. J., 2005); esta modalidad de aprendizaje destaca por su flexibilidad en el proceso de formación, evitando la necesidad física del profesor al mismo tiempo que del estudiante. Gómez y Caicedo (2015) concluyen en su investigación que el *e-Learning* tiene otros aspectos positivos: permite que los estudiantes mantengan su propio ritmo, atiende diversas demandas de capacitación, proporciona un recorrido educativo a personas cuyo lugar de residencia no acepta otra forma, enriquece la propuesta con la combinación de recursos multimedia, amplía el tiempo disponible para los estudiantes y expande los espacios educativos que no solo se reducen al aula física. Por el lado del docente, que hasta ahora pudo manejar su clase y logró cumplir con sus deberes, cumple con algunas tareas nuevas para las que no tiene capacitación, y tal vez esta falta no sea su culpa. *e-Learning* requiere muchos servicios de apoyo para la enseñanza; sin ellos, el trabajo del profesor es muy limitado y, en consecuencia, se pierden todas las posibilidades formativas (García-Peñalvo F. J., 2008). El *e-Learning* se ha ido convirtiendo en una herramienta al servicio de los procesos enseñanza-aprendizaje, de manera que se está perdiendo esa concepción binaria por la que una formación era online o no, para integrarse de una forma mucho más transparente en los procesos educativos y de autoaprendizaje en función de las necesidades de los involucrados (García-Peñalvo & Seoane-Pardo, 2015). Los ambientes híbridos de aprendizaje son aquellos que combinan la instrucción presencial con la enseñanza a través de las

Conceptos teóricos

Tecnologías de la Información y Comunicación. Sousa et al. (Sousa-Santos, Peset-González, & Muñoz-Sepúlveda, 2017) indican que para delimitar si un aprendizaje presencial se puede considerar híbrido hay que establecer un planteamiento pedagógico que combine la efectividad y socialización que proporciona la clase presencial junto a las posibilidades de mejora del aprendizaje que procura el entorno en línea. Así, se determinan los ambientes híbridos o *b-Learning* (*blended learning*-Mixto, Combinado, Mezclado) que combinan lo mejor equilibrado que sea posible las clases presenciales con las extra clases a través de una plataforma tecnológica (Ramírez, 2012). Staker y Horn (2012) proponen una definición revisada del concepto *b-Learning*: El aprendizaje mixto es un programa de educación formal en el que un estudiante aprende, al menos en parte, a través de la entrega en línea de contenido e instrucción con algún elemento de control del estudiante sobre el tiempo, el lugar, el camino y / o el ritmo, y al menos en parte, a través de un bloque supervisado. También aportan cuatro modelos del *b-Learning*: Rotación, Flexible, *Self-Blend*, y Virtual-Enriquecido. El modelo de rotación es un programa en el cual dentro de un curso o materia, los estudiantes rotan en un horario fijo o a discreción del profesor entre modalidades de aprendizaje, en donde al menos una de ellas es aprendizaje en línea. El modelo flexible es un programa en el que el contenido y la instrucción se entregan principalmente a través de Internet, los estudiantes pasan a un horario fluido, individualmente personalizado entre las modalidades de aprendizaje, y el maestro está en el sitio brindando apoyo personal en una base flexible y adaptable según sea necesario. El modelo *Self-Blend* describe un escenario en el que los estudiantes deciden tomar uno o más cursos completamente en línea para complementar sus cursos tradicionales y el maestro en línea es el mismo del curso. Finalmente, el modelo virtual-enriquecido es una experiencia completamente escolar en la que dentro de cada curso, los estudiantes dividen su tiempo entre asistir presencialmente y aprender de forma remota mediante la entrega de contenido e instrucción.

García-Cabrera et al. (2017) expresa que el *b-Learning* “es una de las maneras más eficaces para la autoformación y el aprendizaje a lo largo de la vida”, permitiendo a los egresados participar en cursos

de formación complementarios o de especialización que se ajusten a horarios disponibles por su trabajo. Su implementación requiere cambios en las estrategias y prácticas de los estudiantes, así como en el papel de los docentes como facilitadores de la interacción en línea y el acceso a diferentes fuentes de información para el estudio independiente (Rosser-Limiñana & Martínez, 2015). González-Rogado et al. (2013) han confirmado parcialmente que el nivel de aprendizaje y la satisfacción de los estudiantes será mayor después de la implementación de nuevas metodologías de enseñanza, incluido el *b-Learning*, que en contextos de enseñanza más tradicionales.

Sein-Echaluze et al. (2015) hacen uso de la plataforma *Moodle* para determinar las actividades en su propuesta de *Micro Flip Teaching* (Enseñanza Inversa), haciendo uso de los foros, tareas, cuestionarios y videos. *Moodle* representa una herramienta ampliamente aceptada en el ambiente académico, su uso en 230 países y 92, 974 sitios registrados lo confirman (*Moodle*, 2019). El desarrollo de actividades en una plataforma como *Moodle* contribuye al crecimiento académico del estudiante, entendiendo que cada uno es singular como persona y aprende de modos y ritmos distintos. Fidalgo et al. (2017; Fidalgo-Blanco, Sein-Echaluze, & García-Peñalvo, 2015) aplicaron metodología de enseñanza inversa apoyada en *b-Learning* para "llevar la lección a casa y los deberes a la clase" en la asignatura de Programación, es decir, en lugar de que el estudiante asista a clase presencial para ver la conferencia del maestro y luego ir a casa para practicar lo que aprendió, el estudiante puede ver la clase en casa y luego asistir a clase presencial para practicar lo que han aprendido (Bergmann & Sams, 2013); ejemplificando el uso de herramientas tecnológicas para evitar cursos repetitivos de experiencias pasadas y lograr una transferencia de conocimiento, "a través de evaluar la efectividad del aprendizaje y la motivación del estudiante con su participación activa de los recursos institucionales". *Moodle*, a pesar de no ser la única plataforma que se puede utilizar en las instituciones, su libre uso es un factor determinante para la mayoría de los centros educativos que la han implementado, personalizando su presentación y manteniendo en uso las últimas versiones que permiten dar valor agregado a los cursos que se ofertan a los estudiantes.

Conceptos teóricos

Olelewe y Agomuo (2016) exponen en su trabajo de investigación aplicado en Nigeria, que un modelo *b-Learning* fue más efectivo que un método presencial en la mejora de los logros académicos de Programación; con un valor agregado Shi et al. (2016) propusieron un modelo *b-Learning* basado en roles para enseñarles a los programadores principiantes aprender la programación del lenguaje C, reportando que tal combinación es efectiva para mejorar las habilidades de los estudiantes. En un estado intermedio y conciliador González (2015) expone que el modelo presencial y el modelo a distancia encuentran en el *b-Learning* una fórmula que combina lo mejor de los dos elementos y que es muy difícil establecer diferencias y desventajas entre cada modo, ya que ambos son complementarios y ambos forman parte del mismo método educativo. Particularmente para España, ha sido un trabajo arduo debido al “predominio de una gama de estilos de enseñanza tradicional sobre modelos de actividad basada en el estudiante” que usan *b-Learning* en la universidad (Tirado-Morueta, Aguaded-Gomez, & Hernando-Gómez, 2014).

2.4 Gamificación

A lo largo de la historia se ha observado una evolución en la educación, desde las teorías conductistas (Skinner B. F., 1984) hasta el constructivismo (Piaget, 1986) y el constructivismo social (Ausubel, 1981). El objetivo de estas teorías ha sido explicar el fenómeno educativo y ofrecer caminos pedagógicos al profesor, que permitan el desarrollo creativo de estrategias que promuevan el enganche-compromiso y aprendizaje significativo de sus estudiantes (Ertmer & Newby, 1993; Hernández G. , 2002).

En la didáctica moderna se pueden encontrar diversas estrategias didácticas que promueven el enganche, tales como los métodos de aprendizaje activo o aprendizaje basado en problemas, entre otros, las cuales favorecen el aprendizaje de los estudiantes (Davis & McPartland, 2012). No obstante, la apatía o falta de compromiso por parte de los jóvenes sigue siendo una preocupación hoy en día (Torres, 2006; Lee & Hammer, 2011; Marín, 2015). Hamari et al. (2016) definen el enganche como

la capacidad de inmersión o el nivel de flujo alcanzado en el desarrollo de una actividad mientras que para Gruman et al. (2010) es la cantidad de energía aplicada en acciones y logro de tareas.

El enganche es la pasión por participar y completar las actividades de aprendizaje asignadas (Skinner & Belmont, 1993). En otras palabras, el enganche es el impulso positivo para la realización de algo. Fredricks et al. (2004) proponen tres tipos de enganche, el conductual, cognitivo y emocional. El primero se refiere a la actitud positiva del estudiante para participar y poner atención en clases, el segundo a la disposición para pensar y comprender un tema o concepto, en cuyo proceso interviene la autorregulación (Deci & Ryan, 2008). El emocional se relaciona con las reacciones o sentimientos de los estudiantes dentro del salón de clase, como felicidad, apatía, ansiedad, interés o aburrimiento.

Por otro lado, Csikszentmihalyi (1990) desarrolló una teoría sobre el flujo, el cual consiste en un estado de absorción en una actividad que se disfruta de manera intrínseca, los individuos que se encuentran en este estado se muestran gozosos y exitosos (Admiraal, Huizenga, Akkerman, & Dam, 2011). Recientemente, se han desarrollado diversos estudios en los que se pretende inducir en un estado de flujo a los estudiantes o usuarios a partir de la solución de retos o desafíos (Hou & Li, 2014; Tsai, Huang, Hou, Hsu, & Chiou, 2016; Hamari & Koivisto, 2015; Chang & Wei, 2016). Por lo que un adecuado enganche puede generar un estado favorable de flujo (Hamari, y otros, 2016).

Al mismo tiempo, en la teoría de la autorregulación se muestra que la motivación intrínseca se relaciona con tres factores: la autonomía, la competencia y la socialización (Deci & Ryan, 2008; Sailer, Hense, Mayr, & Mandl, 2017) mientras que la motivación extrínseca se rige por la regulación externa, introyección, identificación e integración (Topirceanu, 2017). En este sentido, tanto el enganche como el flujo pueden impactar en la motivación tanto intrínseca como extrínseca del estudiante o usuario, ya que se provocan cambios en la conducta de la persona al aplicar diversos mecanismos o estrategias didácticas con el propósito de generar compromiso (Alsawaier, 2017).

Conceptos teóricos

Una de esas estrategias didácticas es la gamificación, diversos estudios alrededor del mundo muestran como dicha estrategia ha incrementado el nivel de enganche tanto en estudiantes como en usuarios de la gamificación (Borras-Gene, Martínez-nunez, & Fidalgo-Blanco, 2017; Freudmann & Bakamitsos, 2014; Kyewski & Krämer, 2018; Hamari & Koivisto, 2015), la cual se ha aplicado en el área empresarial, la medicina y recientemente en la educación (Magaña-Valladares, y otros, 2016; Morganti, y otros, 2017; Hamari, 2017).

Las competencias que permiten a los estudiantes crear *software* a partir de algún lenguaje de programación requieren de mucha práctica, de la recreación de circunstancias que propicien el razonamiento para codificar una solución y la integración de conocimientos a partir de la comprensión de los conceptos del paradigma de programación actual, es decir, la Programación Orientada a Objetos (POO). Si el proceso de enseñanza-aprendizaje se desarrolla desde la perspectiva del docente, ejercitando las prácticas establecidas o diseñadas sin que los estudiantes comentan errores para agilizar las clases, se está obstaculizando la experiencia del aprendizaje a partir de las equivocaciones, las cuales si reciben una guía y oportuna corrección, en lugar de generar conocimiento erróneo o frustración, permiten crear el aprendizaje significativo; no es necesario que el profesor refrende su conocimiento o habilidades de programación, son los estudiantes quienes deben hacer suya una nueva habilidad, tanto hombres como las mujeres. La gamificación como estrategia educativa permite crear escenarios de aprendizaje significativos con una motivación sostenida para la adquisición de habilidades específicas, por lo anterior, se presenta su conceptualización y el desarrollo que ha tenido en los últimos años.

Definición de gamificación

Aunque algunas personas creen que la gamificación es reciente, existe evidencia de su aplicación en las últimas décadas. Por ejemplo, en la milicia se han otorgado insignias y rangos por el buen desempeño de los soldados (Dicheva, Dichev, Agre, & Angelova, 2015). Empresas como *eBay*, *Heineken* y *Nike*, entre otras, han incursionado en la gamificación para aumentar la fidelización de

sus clientes. Las pioneras en aplicarlas en los negocios fueron las empresas *Starbucks* y *Amazon*, obteniendo resultados exitosos (Williams, 2015). *Starbucks*, con su campaña *Starbucks My Reward*, consiste en un sistema de recompensas que al acumular cierto número de estrellas los clientes reciben algún tipo de recompensa (Petersen, 2013). En el 2012 los usuarios de *My Reward* totalizaron cerca de 4,5 millones de dólares, en la actualidad solamente las tarjetas generan 3 millones de dólares al año (Chou, 2013). En el ámbito de la medicina se cuentan con diversas investigaciones (Cain, y otros, 2014; Gomez-Galvez, Suárez-Mejías, & Fernandez-Luque, 2015; Magaña-Valladares, y otros, 2016; Sardi, Idri, & Fernández-Alemán, 2017; Garrett & Young, 2018) en las que se ha motivado a los usuarios a mejorar su salud y calidad de vida a través de un sistema de recompensas que utiliza insignias y tableros de clasificación, los cuales son otorgados según los logros alcanzados, así como por compartir sus experiencias relacionadas con ciertas enfermedades crónicas o degenerativas. La gamificación ha impactado incluso en el área de las Ciencias Jurídicas (Escutia-Romero & Pamplona-Roche, 2017) no solo para ofrecer una experiencia educativa al estudiante, también una experiencia de innovación al docente. La gamificación forma parte de las nuevas estrategias que se han incorporado a las tecnologías de aprendizaje que permiten la evolución de las pedagogías (Gros & García-Peñalvo, 2016).

El término gamificación fue acuñado por Nick Pelling en el 2002 pero no fue hasta el 2010 que tomó mayor auge en el ámbito empresarial para luego incorporarse en el ámbito educativo (Rodríguez & Santiago, 2015).

En el ámbito educativo, la gamificación tiene el propósito de colocar al estudiante en escenarios o simulaciones que impliquen el logro de retos atractivos de tal forma que aumente su nivel de compromiso y competitividad (Villalustre & Del Moral, 2015). En otras palabras, la gamificación se refiere a introducir elementos y experiencias de juego en el diseño de procesos de aprendizaje de cualquier área de estudio, dirigidos no solo al aprendizaje, sino a desarrollar ciertas habilidades y actitudes transversales como la colaboración, autorregulación del aprendizaje y creatividad

Conceptos teóricos

(Caponetto, Earp, & Ott, 2014). También provee la oportunidad de explorar y de aprender de los errores gracias a la retroalimentación inmediata y al número de intentos permitidos (Koivisto, Multisilta, Niemi, Katajisto, & Eriksson, 2016).

La gamificación usa elementos que favorecen la motivación intrínseca y extrínseca, por ejemplo, el ofrecimiento de premios favorece a lo extrínseco mientras que lograr un reto favorece a lo intrínseco (Surendeg, Murwa, Yun, & Kim, 2014). También ofrece la oportunidad de experimentar con reglas, emociones y roles sociales (Lee & Hammer, 2011). En otras palabras, al involucrar la gamificación elementos del juego, permite que los aspectos cognitivos, emotivos y sociales converjan en el proceso de aprendizaje (Nisbet & Williams, 2009; Rincón-Flores, Ramírez-Montoya, & Mena, 2016). El aspecto cognitivo se da cuando el estudiante obtiene retroalimentación inmediata y se le ofrecen varios intentos de tal manera que se le lleva a un proceso metacognitivo o bien cuando se le enfrenta a un reto (Rincón-Flores, Ramírez-Montoya, & Mena, 2016). El aspecto emotivo se da cuando el estudiante obtiene un reconocimiento por su logro (Mekler, Brühlmann, Opwis, & Tuch, 2013) y el aspecto social sucede cuando los logros son socializados a través de un tablero de liderazgo o bien cuando los estudiantes trabajan colaborativamente para lograr un reto o misión (Domínguez, y otros, 2013; Hanus & Fox, 2015). Existen diversos modelos de gamificación (Bunchball, 2012; Kapp, 2012; Robson, Plangger, Kietzmann, McCarthy, & Pitt, 2015; Yildirim, 2017) los cuales parten de la misma idea: “tomar elementos del juego en contextos que no son de juego” (Deterding, Sicart, Nacke, O’Hara, & Dixon, 2011), la Tabla 5 muestra algunas investigaciones en el ámbito educativo. Se observa que mayormente se utilizaron tableros, retos e insignias. En la mayor parte de las investigaciones se consiguió enganchar y motivar a los participantes; incluir la gamificación para el diseño de un ambiente o actividades de aprendizaje también favorece el proceso creativo del docente (Marín, 2015). El diseño de un sistema gamificado generalmente depende de los recursos con los que se cuente y de las necesidades educativas del grupo al que va dirigido. Por lo que es importante considerar que la gamificación no solamente se da aplicando tecnología en los viejos modelos de

enganche (Burke, 2012), pues esta podría ser una gran limitación, ya que la gamificación digitalizada suele ser costosa en términos de tiempo y dinero, además, la gamificación sí que puede valerse de la tecnología, pero más bien como un medio que como un fin (Hew, Huang, Chu, & Chiu, 2016) para lograr incrementar tanto el enganche como el flujo del estudiante y colocarlo en un estado de inmersión (Keusch & Zhang, 2015; Hamari, y otros, 2016).

Tabla 5. Algunos estudios sobre gamificación en la educación. Fuente: Elaboración propia con la Dr. Elvira Rincón- Flores

Referencia	Alcance	Resultados	Dinámicas, Mecánicas y Componentes
Mekler, Brühlmann, Opwis y Tuch, 2013	Incrementar motivación y autonomía	Incremento en el enganche mejorando el desempeño, más no hubo evidencia sobre aumento de la motivación intrínseca.	D: progresión M: niveles C: puntos, tablero.
Domínguez et al., 2013	Incrementar enganche	Los estudiantes que completaron la experiencia gamificada tuvieron mejor puntaje en los exámenes prácticos pero su desempeño fue pobre en los exámenes escritos y participación en clase.	D: progresión M: niveles C: puntos, tablero.
Rincón-Flores, Ramírez-Montoya y Mena, 2016	Incrementar enganche y promover el aprendizaje a largo plazo	Incremento en el enganche y se logró un aprendizaje a largo plazo en la mayor parte del tema en el que se aplicó la gamificación.	D: progresión y emociones M: retos, competición. C: insignias, tablero
Hew, Huang, Chu y Chiu, 2016	Impacto de usar mecanismos de juego en el enganche cognitivo y conductual de estudiantes Asiáticos.	La calidad de las actividades del grupo experimental fue mejor que la del grupo de control. El grupo experimental disfrutó la experiencia gamificada.	D: progresiones M: retos C: puntos, insignias, y tablero.
Yildirim, 2017	Mejorar las actitudes y logros de los estudiantes.	Efectos positivos en logros y actitudes de los estudiantes hacia las lecciones.	D: emociones, restricciones, progresiones y narrativa. M: oportunidad, competición, cooperación, intercambio y reto C: puntos, insignias, nivel y tablero.
(Khalil, Ebner, & Admiraal, 2017)	Incrementar el enganche y motivación del estudiante.	La tasa terminal aumentó con los años, 2014 con un 17,54%, 2015 con un 19,74% y 2016 con un 26,05%	D: progresiones M: retroalimentación C: barra de progreso
Kyewski y Krämer, 2018.	Incrementar el enganche y motivación del estudiante.	Los resultados muestran que las insignias tienen menos impacto en la motivación y el rendimiento de lo que comúnmente se supone. Contrariamente a lo esperado, las insignias que solo podían ser vistas por los propios estudiantes fueron evaluadas de manera más positiva que aquellas que también podrían ser vistas por otros.	D: progresiones M: retroalimentación C: insignias y tablero.

Conceptos teóricos

La gamificación es una estrategia didáctica con diversos estudios alrededor del mundo que muestran cómo ha incrementado el nivel de enganche tanto en estudiantes como en usuarios de la gamificación (Borras-Gene, Martínez-nunez, & Fidalgo-Blanco, 2017; Freudmann & Bakamitsos, 2014; Kyewski & Krämer, 2018; Hamari & Koivisto, 2015; Morganti, y otros, 2017; Hamari, 2017). “Las metodologías docentes basadas en la clase magistral y la resolución de ejercicios ficticios se transforman en nuevas propuestas que acercan al estudiante a la realidad”.

La gamificación aporta a la educación superior compromiso, flexibilidad, competición y colaboración capaz de producir una inspiradora reflexión: “No enseñamos, compartimos; no corregimos, sugerimos; no valoramos, acompañamos” (Marqués & Aguilar-Paredes, 2017).

Específicamente, para el caso de usar gamificación en programación de computadoras en la educación superior, existen las siguientes propuestas concretas.

Carreño-León et al. (2018) usan gamificación para el curso introductorio Desarrollo de *software* en la ingeniería para hacer más atractivas las tareas existentes, que generalmente los estudiantes las consideran aburridas. Usan el elemento de recompensa por completar las mencionadas tareas, es decir, dentro del salón de clases resolver pequeños ejercicios para el diseño de algoritmos con tres niveles: básico, intermedio y avanzado; generando un ambiente de competitividad al momento de hacer visibles las recompensas entre los jugadores. Barna y Fodor (2018) además de usar también el elemento de recompensa, ofertan rutas de aprendizaje alternativo, opciones de retroalimentación y plataformas de interacción social a través de emplear la plataforma *Moodle*. Fotaris et al. (2015) para fortalecer el elemento de recompensa, pero usando un método de aprendizaje basado en problemas, definen una secuencia de puntajes de evaluación en tiempo real, agregan retroalimentación del instructor asignado al curso durante la intervención de gamificación mientras se desarrolla la actividad de codificación en el lenguaje de programación *Python*. Iosup y Epema (2014) reportan su

trabajo de implementación con mayor detalle y un elemento de novedad en la estrategia de gamificación. Describen las mecánicas y dinámicas, pero comentan las estéticas como el elemento que refuerza a las primeras dos, debido a que el arte del diseño de juego es muy importante para el estudiante-jugador. También determinan tipos de jugadores-estudiantes: exploradores, sociales, logros y ganadores para enmarcar su actividad inicial en la gamificación. Jayasinghe y Dharmaratne (2013) aportan resultados para un tema específico de la programación, los métodos de ordenamiento, y destaca su elemento de recompensa debido a que es una pulsera.

Khasianov et al. (2016) definen claramente los elementos de gamificación usados en su trabajo de investigación: metas, retos, puntos e insignias, los denominan entidades claves en la plataforma que desarrollaron, la cual está disponible como un servicio en la nube para su acceso en cualquier tipo de dispositivo móvil e interfaz web, una vez descargada puede funcionar con o sin conexión a internet. Con la herramienta de gamificación digital, es posible conocer el progreso del curso para cada uno de los estudiantes a través de las insignias que obtienen al completar tareas, y asignar nuevas metas de aprendizaje. Una diferencia con otros trabajos, es la inclusión de insignias extracurriculares, es decir, por realizar tareas culturales, deportivas o de comunidad.

Lopes y Mesquita (2015) reportan un mecanismo de recompensa como el elemento clave del método de gamificación usado durante el proceso de aprendizaje en un curso de ciencias de la computación; también elaboran una clasificación de los estudiantes, así como una constante y actualizada retroalimentación de su progreso dentro del curso. Lopes (2014) de forma similar da un peso a la recompensa, pero le llama premios para motivar y calificar a los estudiantes por medio de una plataforma en línea. Las secciones en el currículo de la asignatura se transforman en niveles, cada nivel está representado por un castillo en un mapa, el castillo es conquistado tan pronto como el estudiante resuelve los desafíos requeridos. Solo después de conquistar-terminar todos los niveles se puede terminar exitosamente la asignatura. Un elemento clave durante el proceso de aprendizaje-

Conceptos teóricos

juego, es poder ganar monedas virtuales que se pueden canjear por herramientas que ayuden a resolver tareas durante la conquista de un castillo.

2.5 Conclusiones

En este capítulo se ha hecho un resumen de 4 conceptos que representan una justificación de la propuesta de la tesis doctoral con el objetivo de determinar la aportación de la investigación. Se han establecido definiciones de cada uno de los conceptos, así como la visión de otros trabajos de investigación, pero principalmente para los conceptos pensamiento computacional y gamificación se han podido exponer resultados de trabajos vinculados con su aplicación en la enseñanza de programación de computadoras en educación superior, que representa el escenario de investigación de interés. En consecuencia, se puede determinar una combinación de los conceptos para crear una aportación novedosa, pues no existe un estudio reportado donde se mezcle su uso.

El pensamiento computacional al tener una estrecha relación con los conceptos de las ciencias de la computación y las habilidades para la resolución de problemas, representa un elemento clave en el actual proceso de enseñanza-aprendizaje de la programación de computadoras. La complejidad del proceso mencionado está asociada con la capacidad creativa del estudiante y docente. La estrategia de gamificación puede aportar motivación y enganche al establecer un espacio creativo de aprendizaje a través de mecánicas, dinámicas y componentes.

La educación personalizada en la educación superior tiene muchas oportunidades que ofrecer a los estudiantes, pero el trabajo del docente en ocasiones se ve rebasado por la cantidad de matriculados en los cursos universitarios, principalmente en los primeros ciclos, donde los grupos son amplios. Existe documentación de actividades que llevan a mejorar la experiencia educativa y destaca la capacidad de adaptarse a la individualidad de los participantes usando las Tecnologías de la Información, particularmente un entorno de aprendizaje *b-Learning*. En consecuencia, considerar la educación personalizada para la enseñanza de Programación de computadoras va más allá de usar un

lenguaje de programación, está asociado con la creación de una metodología y escenario de aprendizaje que en cada estudiante es diferente, de hecho, refuerza un estilo personal para el desarrollo de *software*. En consecuencia, la suma de los conceptos anteriores justifica su integración en la tesis doctoral. En el proceso enseñanza-aprendizaje se ha de buscar ofrecer opciones a los estudiantes, aprovechando las actividades presenciales y extendiendo el aula con actividades en línea por medio de un entorno *b-Learning*.

CAPÍTULO 3. REVISIÓN SISTEMÁTICA DE LA LITERATURA

3.1 Introducción

La actividad de investigación sobre un tema específico tiene un momento importante al permitir conocer lo que otras investigaciones han realizado, no solo para detectar áreas de oportunidad en el sentido de encontrar lo que otros no han investigado, sino también en conocer el contexto donde se han realizado las investigaciones y qué similitudes existen con el problema a resolver. La determinación del Marco Teórico se enriquece con la actividad denominada Revisión Sistemática de la Literatura. La guía de Kitchenham (2004), adaptada a las necesidades de los investigadores de ingeniería de software, está basada a su vez en tres guías para revisiones sistemáticas destinadas a ayudar a los investigadores médicos; como resultado, propone tres fases principales de la revisión: i) Planificación, ii) Conducción, y iii) Reporte. Además, para cada fase propone etapas; las etapas relacionadas con la planificación son: identificación de la necesidad de una revisión y desarrollo de un protocolo de revisión; las etapas asociadas con la conducción son: identificación de la revisión, selección de estudios primarios, estudio de evaluación de calidad, extracción de datos y monitoreo, y síntesis de datos; el reporte es una fase de solo una etapa. Una nota importante que destaca Kitchenham es la importancia de reconocer que muchas de las etapas involucran iteración, es decir, pareciera que ofrece una lista de etapas en orden secuencial, aunque algunas actividades son inicializadas durante la etapa del desarrollo del protocolo, son refinadas cuando la propia revisión se está ejecutando. A partir de la guía propuesta por Kitchenham, se pueden ajustar las etapas a las necesidades particulares de una investigación, pero sin perder la esencia misma de la revisión sistemática: evaluar e interpretar toda la relevante investigación disponible a una pregunta de investigación particular, área, o fenómeno de interés, a través de una metodología auditable, rigurosa y confiable.

Revisión sistemática de la literatura

Para Urra y Barría (2010) las revisiones sistemáticas poseen relevancia por su credibilidad en la búsqueda, recolección, ordenamiento y análisis de las investigaciones que se han realizado en un periodo de tiempo; las etapas que la componen son: definir su propósito, formular la pregunta, buscar literatura, evaluar los datos, analizar los datos y presentar los resultados.

Para Manterola et al. (2011) una revisión sistemática es un artículo de síntesis de la evidencia disponible, en el que se realiza una revisión de aspectos cuantitativos y cualitativos de estudios primarios, con el objetivo de resumir la información existente respecto de un tema en particular; las etapas que la componen son: formulación del problema, localización y selección de estudios primarios, evaluación de la calidad metodológica de los estudios, extracción de datos, análisis y presentación de resultados.

En el contexto de la medicina, Linares-Espinós et al. (2018), establecen la importancia de realizar una revisión sistemática, que trasladada a la presente investigación, permite obtener evidencia apoyada en la mejor información científica disponible para aplicarla en la determinación del marco teórico y usarla en el marco empírico; en consecuencia el proceso de revisión debe estar bien desarrollado y planificado de antemano para reducir sesgos y eliminar estudios irrelevantes o de mala calidad, los pasos que proponen incluyen: i) formular correctamente la pregunta a responder, ii) desarrollo de un protocolo (criterios de inclusión y exclusión), iii) realizar una búsqueda detallada y amplia, iv) cribar los resúmenes de los trabajos identificados en la búsqueda y posteriormente de los textos completos seleccionados.

En la presente tesis doctoral se realizó la revisión sistemática en tres etapas como lo propone Kitchenham (2004) y lo ejecuta también Briz-Ponce (Briz-Ponce, 2016) y Hernández et al. (2017), las secciones siguientes contienen el desarrollo de las etapas y actividades indicadas a detalle en la Figura 4.

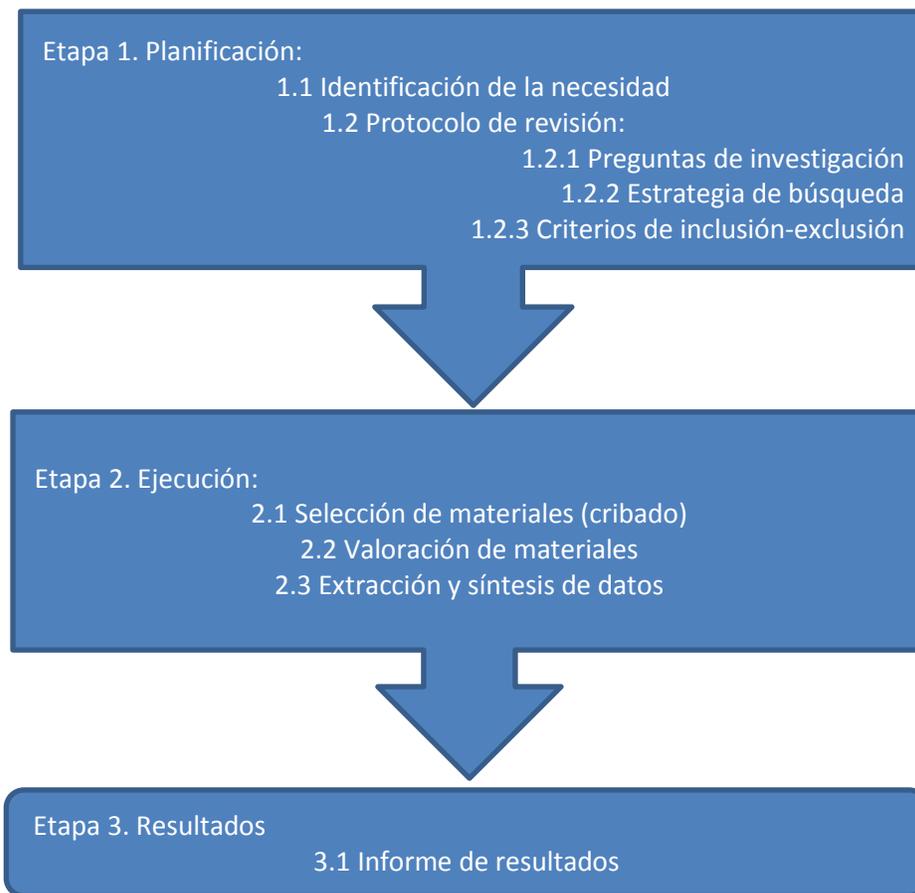


Figura 4. Etapas y actividades de la revisión sistemática en la tesis doctoral. Fuente: Elaboración propia

3.2 Planificación del proceso de revisión

Dos son las actividades realizadas en la etapa de planificación, la identificación de la necesidad y el protocolo de revisión. Kitchenham (2004) comenta que “la necesidad de una revisión sistemática surge desde el requerimiento de los investigadores por resumir toda la información existente acerca de un fenómeno en una manera completa e imparcial. Los investigadores deben asegurar que una revisión sistemática es necesaria.” Una vez determinada la necesidad, se procede a determinar el protocolo de revisión que declara los métodos que serán usados para emprender una revisión sistemática específica.

3.2.1 Identificación de la necesidad

En el caso de esta tesis doctoral el proceso de revisión sistemática es necesario para ubicar la problemática definida y argumentar el tema de estudio. Kitchenham (2004) sugiere que “los investigadores deben identificar y revisar algunas revisiones sistemáticas existentes del fenómeno de interés usando un apropiado criterio de evaluación” para contribuir con la identificación de la necesidad misma; para su evaluación se pueden usar las preguntas: ¿Cuáles fueron los objetivos de la revisión? ¿Qué fuentes fueron buscadas para identificar estudios primarios? ¿Hubo algunas restricciones? ¿Cuáles fueron los criterios de inclusión/exclusión y cómo fueron aplicados? ¿Qué criterios fueron usados para evaluar la calidad de los estudios primarios y cómo fueron aplicados? ¿Cómo fueron extraídos los datos desde los estudios primarios? ¿Cómo fueron sintetizados los datos? ¿Cómo fueron las diferencias entre los estudios investigados? ¿Cómo fueron combinados los datos? ¿Fue razonable combinar los estudios? ¿Las conclusiones fluyen desde la evidencia?

3.2.2 Protocolo de revisión

El protocolo de revisión realizado en la tesis doctoral está definido por tres actividades: i) preguntas de investigación, ii) estrategia de búsqueda (bases de datos y fórmula de búsqueda), iii) criterios de inclusión y exclusión. A continuación, se documenta el procedimiento realizado en cada una de las actividades.

3.2.2.1 Preguntas de investigación

El objetivo de esta actividad en el protocolo de revisión es tener preguntas que permitan acotar adecuadamente el problema en cuestión, es decir, determinar estrategias para reducir la deserción escolar, y proporcionar competencias básicas para la programación de computadoras, funcionales a los requerimientos de la industria del *software*, así, se determinan las preguntas indicadas en la Tabla 6 tomando como marco las estrategias de educación personalizada, pensamiento computacional, *b-Learning* y gamificación.

Tabla 6. Preguntas de investigación. Fuente: Elaboración propia

Identificación (ID) de la Pregunta de Investigación (PI)	Pregunta de Investigación
PI.1	¿Cuál es la definición usada del concepto Pensamiento Computacional/Educación Personalizada/ <i>b-learning</i> /gamificación?
PI.2	¿Cuál ha sido la actividad de las investigaciones de Pensamiento Computacional/Educación Personalizada/ <i>b-learning</i> /gamificación en Educación Superior?
PI.3	¿Cómo se ha usado el Pensamiento Computacional/Educación Personalizada/ <i>b-learning</i> /gamificación en la enseñanza de la Programación de computadoras en la Educación Superior?

3.2.2.2 Estrategia de búsqueda

Las bases de datos seleccionadas para la revisión sistemática en la tesis doctoral son *Web of Science* (WOS) y SCOPUS, debido a que representan las dos principales bases de datos mundiales de referencias bibliográficas y citas de publicaciones periódicas, además de tener acceso como estudiante de la Universidad de Salamanca a través de las licencias gestionadas por la Fundación Española para la Ciencia y Tecnología.

Las fórmulas de búsqueda, realizadas en inglés, ajustadas a cada una de las bases de datos y estrategias están indicadas en la Tabla 7.

La búsqueda se solicitó en el Título del material entre los años 2000 – 2018, sin restricción del tipo del mismo. Lo anterior ayudó a conocer el contexto donde consideraron los autores enmarcar su trabajo de investigación.

3.2.2.3 Criterios de inclusión / exclusión

Inclusión:

Cualquier tipo de material.

Idioma: inglés.

Materiales entre los años 2000 y 2018.

Revisión sistemática de la literatura

Exclusión:

Idioma diferente al inglés o español.

Materiales antes del 2000.

Tabla 7. Fórmulas de búsqueda. Fuente: Elaboración propia

ID Formula	Formula
WOS	
FWPC	TÍTULO: (computational thinking) AND TÍTULO: (teaching programming) OR TÍTULO: (computational thinking) AND TÍTULO: (higher education)
FWEP	TÍTULO: (personalized education) AND TÍTULO: (teaching programming) OR TÍTULO: (personalized education) AND TÍTULO: (higher education)
FWBL	TÍTULO: (b-learning) AND TÍTULO: (teaching programming) OR TÍTULO: (b-learning) AND TÍTULO: (higher education)
FWG	TÍTULO: (gamification) AND TÍTULO: (teaching programming) OR TÍTULO: (gamification) AND TÍTULO: (higher education)
SCOPUS	
FSPC	((computational AND thinking AND higher AND education) OR (computational AND thinking AND teaching AND programming)) 17 resultados
FSEP	((personalized AND education AND higher AND education) OR (personalized AND teaching AND programming)) 13 resultados
FSBL	((b-learning AND higher AND education) OR (b-learning AND teaching AND programming)) 7 resultados
FSG	((gamification AND higher AND education) OR (gamification AND teaching AND programming)) 39 resultados

3.3 Ejecución de revisión

3.3.1 Selección de materiales

En esta sección se describen los materiales obtenidos a partir de ejecutar las fórmulas de búsqueda en las bases de datos WOS y SCOPUS el 15 y 16 de noviembre de 2018 y en una segunda ocasión el 13 y 18 de febrero de 2019. La Tabla 8 contiene los resultados en WOS y SCOPUS de la primera y segunda ejecución de la revisión.

Tabla 8. Cantidad de materiales obtenidos en las bases de datos. Fuente: Elaboración propia

Resultado	Búsqueda / Observaciones
Primera: 22 Segunda: 23	Buscó: TÍTULO: (computational thinking) AND TÍTULO: (teaching programming) OR TÍTULO: (computational thinking) AND TÍTULO: (higher education) Período de tiempo: 2000-2018. Bases de datos: WOS, CCC, DIIDW, KJD, MEDLINE, RSCI, SCIELO. Idioma de búsqueda=English. Solo un material nuevo en la búsqueda del 13 de febrero.
17	Buscó: TÍTULO: (personalized education) AND TÍTULO: (teaching programming) OR TÍTULO: (personalized education) AND TÍTULO: (higher education) Período de tiempo: 2000-2018. Bases de datos: WOS, CCC, DIIDW, KJD, MEDLINE, RSCI, SCIELO. Idioma de búsqueda=English. Sin cambios en la búsqueda del 13 de febrero.
Primera: 8 Segunda: 9	Buscó: TÍTULO: (b-learning) AND TÍTULO: (teaching programming) OR TÍTULO: (b-learning) AND TÍTULO: (higher education) Período de tiempo: 2000-2018. Bases de datos: WOS, CCC, DIIDW, KJD, MEDLINE, RSCI, SCIELO. Idioma de búsqueda=English. Solo un material nuevo en la búsqueda del 13 de febrero.
Primera: 38 Segunda: 47	Buscó: TÍTULO: (gamification) AND TÍTULO: (teaching programming) OR TÍTULO: (gamification) AND TÍTULO: (higher education) Período de tiempo: 2000-2018. Bases de datos: WOS, CCC, DIIDW, KJD, MEDLINE, RSCI, SCIELO. Idioma de búsqueda=English 9 materiales nuevos en la búsqueda del 13 de febrero.
Total WOS: Primera: 85 Segunda: 96	
Primera: 17 Segunda: 20	TITLE ((computational AND thinking AND teaching AND programming) OR (computational AND thinking AND higher AND education)) AND PUBYEAR > 1999. 3 materiales nuevos en la búsqueda del 13 de febrero.
Primera: 12 Segunda: 13	TITLE ((personalized AND education AND higher AND education) OR (personalized AND education AND teaching AND programming)) AND PUBYEAR > 1999. Solo un material nuevo en la búsqueda del 13 de febrero.
7	TITLE ((b-learning AND higher AND education) OR (b-learning AND teaching AND programming)) AND PUBYEAR > 1999. Sin cambios en la búsqueda del 13 de febrero.
Primera: 39 Segunda: 44	TITLE ((gamification AND higher AND education) OR (gamification AND teaching AND programming)) AND PUBYEAR > 1999. 5 materiales nuevos en la búsqueda del 13 de febrero.
Total SCOPUS: Primera: 75 Segunda: 84	
TOTAL Primera:160 Segunda: 180	

3.3.2 Valoración de materiales

Para el proceso de selección de los resultados una vez realizadas las búsquedas, se determinaron los siguientes criterios:

Inicialmente se eliminaron los materiales duplicados (17 repetidos), obteniendo un total de 143 en la ejecución primera de la revisión mientras que para la segunda se obtendrían 163. Posteriormente se respondió a la siguiente pregunta:

Revisión sistemática de la literatura

¿El título o resumen del material hace referencia al pensamiento computacional / educación personalizada / *b-learning* / gamificación en la enseñanza de programación en la educación superior?

En caso afirmativo, la lectura del resumen determinó la búsqueda del material completo para su consideración en la creación del Marco Conceptual, se consideraron 24 materiales en la primera ejecución, pero 2 no fue posible obtener el texto completo, en consecuencia, se incluyeron 22 materiales en la revisión. En la segunda ejecución se agregaron a consideración 2 nuevos materiales, pero uno de ellos no fue posible obtener el texto completo, en consecuencia, se incluyeron 23 materiales en la revisión. La Figura 5 ilustra el proceso de selección orientado por el trabajo realizado por Ferreras-Fernández (2016). El detalle de la valoración por cada una de las estrategias se encuentra en la Tabla 9 para cada una de las bases de datos consultadas en ambas ejecuciones. Se observa que no se obtuvieron materiales para el caso de educación personalizada y *b-Learning* en ambas bases de datos usando las consultas específicas.

Tabla 9. Cantidad de materiales incluidos en la revisión por estrategia. Fuente: Elaboración propia

WOS – Primera: 85 Segunda: 96				SCOPUS – Primera: 75 Segunda: 84				Total
Pensamiento Computacional	Educación personalizada	B- Learning	Gamificación	Pensamiento Computacional	Educación personalizada	B- Learning	Gamificación	
Obtenidos								
22 / 23	17	8 / 9	38 / 47	17	12	7	39	160 / 180
Incluidos								
7	0	0	5 / 6	5	0	0	5	22 / 23

3.3.3 Extracción y síntesis de datos

La siguiente etapa del proceso consistió en la revisión de los 22 materiales obtenidos de la valoración realizada en la primera ejecución. La Tabla 10 contiene la identificación de los 5 materiales para la estrategia de gamificación en la base de datos SCOPUS. La Tabla 11 contiene la identificación de materiales para la misma estrategia en la base de datos WOS, durante la segunda ejecución se obtuvo un nuevo material. Con el objetivo de obtener respuestas a las preguntas de investigación, se realizó una síntesis de los puntos clave, es decir, ¿Cuál es la definición de gamificación?, ¿Cuál es la actividad

de investigación? y ¿Cuál es el uso de gamificación en programación en educación superior? La información encontrada está contenida en el Apéndice E y Apéndice F por cada material revisado.

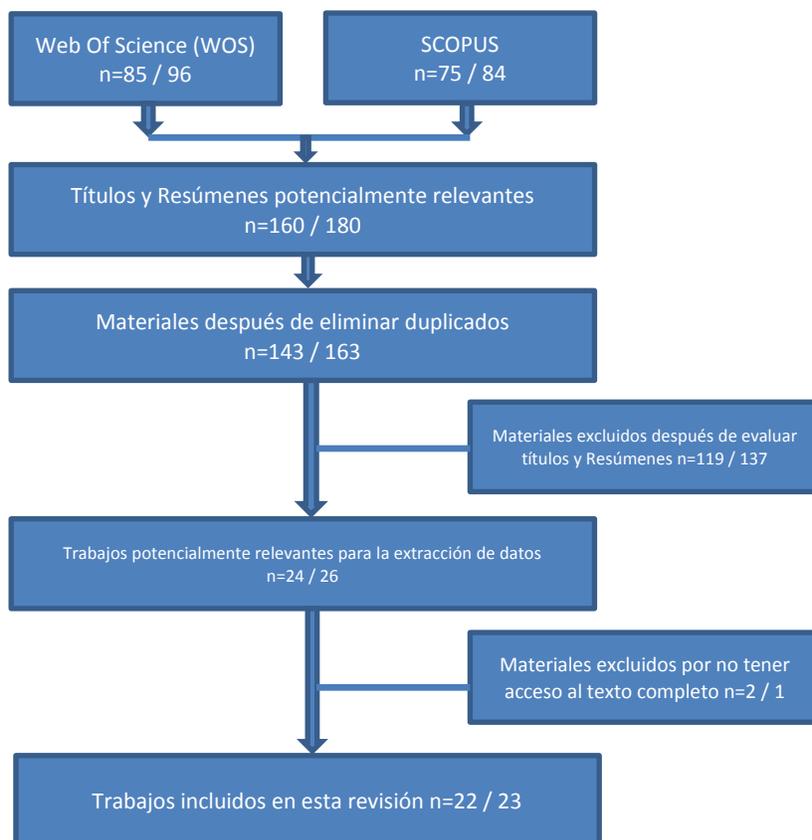


Figura 5. Diagrama de flujo de la selección de materiales en la revisión sistemática. Fuente: Basado en Ferreras-Fernández (2016)

La Tabla 12 contiene la identificación de los 5 materiales para la estrategia de pensamiento computacional en la base de datos SCOPUS. La Tabla 13 contiene la identificación de materiales para la misma estrategia en la base de datos WOS. Nuevamente, con el objetivo de obtener respuestas a las preguntas de investigación, se realizó una síntesis de los puntos clave, es decir, ¿Cuál es la definición de pensamiento computacional?, ¿Cuál es la actividad de investigación? y ¿Cuál es el uso del pensamiento computacional en programación en educación superior? La información encontrada se encuentra en el Apéndice G y Apéndice H por cada material revisado.

Tabla 10. Identificación de los 5 materiales obtenidos de SCOPUS de la estrategia gamificación.
Fuente: Elaboración propia

ID – Citas	SCOPUS_G_01 – 0
Título	Gamification technique for teaching programming
Autores	Mónica Carreño-León, Andrés Sandoval-Bringas, Francisco Álvarez-Rodríguez, Yolanda Camacho-González.
País	México
Publicación / Año	IEEE Global Engineering Education Conference (EDUCON) / 2018
DOI	10.1109/EDUCON.2018.8363482
ID – Citas	SCOPUS_G_02 – 4
Título	An empirical study on the use of gamification on IT courses at higher education
Autores	Balázs Barna, Szabina Fodor
País	Hungría
Publicación / Año	Advances in Intelligent Systems and Computing / 2018
DOI	10.1007/978-3-319-73210-7_80
ID – Citas	SCOPUS_G_03 – 2
Título	From hiscore to high marks: Empirical study of teaching programming through gamification
Autores	P.Fotaris, T. Mastoras, R. Leinfellner, R. Yasmine
País	Reino Unido - Grecia
Publicación / Año	Proceedings of the European Conference on Games-based Learning / 2015
DOI	No proporcionado por SCOPUS
ID – Citas	SCOPUS_G_04 – 74
Título	An Experience Report on Using Gamification in Technical Higher Education
Autores	Alexandru Iosup, Dick Epema
País	Países Bajos
Publicación / Año	Proceedings of the 45th ACM Technical Symposium on Computer Science Education / 2015
DOI	10.1145/2538862.2538899
ID – Citas	SCOPUS_G_05 – 10
Título	Game base learning vs gamification from the higher education students' perspective
Autores	Udeni Jayasinghe, Anuja Dharmaratne
País	Sri Lanka
Publicación / Año	Proceedings of 2013 IEEE International Conference on Teaching, Assessment and Learning for Engineering / 2013
DOI	10.1109/TALE.2013.6654524

Tabla 11. Identificación de los 6 materiales obtenidos de WOS de la estrategia gamificación.

Fuente: Elaboración propia

ID – Citas	WOS_G_01 – 0
Título	La gamificación en la educación superior: una revisión sistemática
Autores	Carolina Lozada Ávila, Simón Betancur Gómez (Lozada & Betancur, 2017)
País	Colombia
Publicación / Año	Revista Ingenierías Universidad de Medellín, vol. 16, No. 31 / 2017
DOI	10.22395/rium.v16n31a5
ID – Citas	WOS_G_02 – 0
Título	GAMIFICATION IN HIGHER EDUCATION: KAZAN FEDERAL UNIVERSITY PRIMER
Autores	Airat KHASIANOV, Irina SHAKHOVA, Bulat GANIEV
País	Rusia
Publicación / Año	Elearning Vision 2020!, vol. I Colección: eLearning and Software for Education / 2016
DOI	10.12753/2066-026X-16-075
ID – Citas	WOS_G_03 – 0
Título	Who wants to be a pythonista? Using gamification to teach computer programming.
Autores	Panagiotis Fotaris, Theodoros Mastoras, Richard Leinfellner, Yasmine Rounally
País	Reino Unido – Grecia
Publicación / Año	7 th International Conference on Education and New learning Technologies / 2015
DOI	No proporcionada por WOS
ID – Citas	WOS_G_04 – 0
Título	Evaluation of a gamification methodology in higher education
Autores	Rui Pedro Lopes, Cristina Mesquita
País	Portugal
Publicación / Año	7 th International Conference on Education and New learning Technologies / 2015
DOI	No proporcionada por WOS
ID – Citas	WOS_G_05 – 0
Título	An Award system for gamification in higher education
Autores	Rui Pedro Lopes
País	Portugal
Publicación / Año	7 th International Conference of Education, Research and Innovation / 2014
DOI	No proporcionada por WOS
ID - Citas	WOS_G_01_2 – 0
Título	Gamification as a didactic strategy for teaching / learning programming: a systematic mapping of the literature
Autores	Oscar Revelo Sánchez, Cesar Alberto Collazos Ordoñez, Javier Alejandro Jiménez Toledo (Revelo, Collazos, & Jiménez, 2018)
País	Colombia
Publicación / Año	Revista digital Lampsakos / 2018
DOI	No proporcionada por WOS

Tabla 12. Identificación de los 5 materiales obtenidos de SCOPUS de la estrategia pensamiento computacional. Fuente: Elaboración propia

ID – Citas	SCOPUS_PC_01 – 0
Título	Computational thinking development through creative programming in higher education
Autores	Margarida Romero, Alexandre Lepage y Benjamin Lille.
País	Francia – Canadá
Publicación / Año	International Journal of Educational Technology in Higher Education / 2017
DOI	10.1186/s41239-017-0080-z
ID – Citas	SCOPUS_PC_02 – 0
Título	The computational thinking- oriented inquiry teaching mode for advanced programming language course
Autores	Qian Gao
País	China
Publicación / Año	BioTechnology: An Indian Journal / 2014
DOI	No disponible por SCOPUS
ID – Citas	SCOPUS_PC_03 – 1
Título	Training for Computational Thinking Capability on Programming Language Teaching
Autores	ZHANG Yinnan, LUO Chaosheng
País	China
Publicación / Año	Proceedings of 2012 7th International Conference on Computer Science and Education / 2012
DOI	10.1109/ICCSE.2012.6295420
ID – Citas	SCOPUS_PC_04 – 0
Título	Research of VB programming teaching mode based on the core of computational thinking ability training
Autores	Chunying Zhang, Xiao Chen, JianJing Li
País	China
Publicación / Año	6th International Conference on Computer Science and Education, Final Program and Proceedings / 2011
DOI	10.1109/ICCSE.2011.6028861
ID – Citas	SCOPUS_PC_05 – 2
Título	Programming courses teaching method for ability enhancement of computational thinking
Autores	HUANG Wenming, DENG Zhenrong, Dong Rongsheng
País	China
Publicación / Año	International Association of Computer Science and Information Technology - Spring Conference / 2009
DOI	10.1109/IACSIT-SC.2009.52

Tomando en consideración el país indicado por los autores en cada uno de los materiales, la Tabla 14 muestra la cantidad de cada uno distinguiendo la base de datos y estrategia por medio del ID, destacando por mucho China con 9 publicaciones para la estrategia de pensamiento computacional. Posteriormente, Portugal, Colombia y la dupla Reino Unido-Grecia con dos materiales para la estrategia de gamificación, y con solo un material el resto de los países obtenidos.

Tabla 13. Identificación de los 7 materiales obtenidos de WOS de la estrategia pensamiento computacional. Fuente: Elaboración propia

ID – Citas	WOS_PC_01 – 0
Título	Programming Language Teaching Model Based on Computational Thinking and Problem-based Learning
Autores	Guang-ming Chen
País	China
Publicación / Año	Proceedings of the 2nd International Seminar on Education Innovation and Economic Management / 2017 DOI – No proporcionado por WOS
ID – Citas	WOS_PC_02 – 0
Título	Discussion on case teaching method based on computational thinking in programming teaching
Autores	Zhao Ni
País	China
Publicación / Año	Proceedings of the International Conference on Social Science, Education and Humanities Research / 2017 DOI – No proporcionado por WOS
ID – Citas	WOS_PC_03 – 0
Título	Research on the teaching of programming language based on computational thinking
Autores	Lu Ying, Liu Pingping
País	China
Publicación / Año	Proceedings of the International Conference on Social Science, Education and Humanities Research / 2017 DOI – No proporcionado por WOS
ID – Citas	WOS_PC_04 – 0
Título	The PBL Teaching Method Research Based on Computational Thinking in C Programming
Autores	Zhi-Mei Cheng, Xiang Li
País	China
Publicación / Año	2 nd International Conference on Modern Education and Social Science / 2016
DOI	No proporcionado por WOS
ID – Citas	WOS_PC_05 – 0
Título	On the basis of the program design teaching and research of cultivation of computational thinking ability
Autores	Xia Aiyue
País	China
Publicación / Año	Proceedings of the 2 nd International Conference on Social Science and Higher Education / 2016 DOI – No proporcionado por WOS
ID – Citas	WOS_PC_06 – 0
Título	Enseñando a programar: un camino directo para desarrollar el Pensamiento Computacional
Autores	Patricia Compañ-Rosique, Rosana Satorre-Cuerda, Faraón Llorens-Largo, Rafael Molina-Carmona
País	España
Publicación / Año	Revista de Educación a Distancia / 2015
DOI	10.6018/red/46/11
ID – Citas	WOS_PC_07 – 1
Título	Teaching programming with computational and informational thinking
Autores	Greg Michaelson
País	Reino Unido
Publicación / Año	Journal of Pedagogic Development / 2015 DOI – No proporcionado por WOS

Revisión sistemática de la literatura

La revisión de los materiales respecto al número de citas, destaca SCOPUS_G_04 con 74 menciones, SCOPUS_G_05 con 10 y SCOPUS_G_02 con 4, posteriormente, SCOPUS_G_03 y SCOPUS_PC_05 con 2 citas, finalmente, SCOPUS_PC_03 y WOS_PC_07 con una cita. El resto de los materiales revisados aún no ha sido citado.

Para terminar con la extracción de datos, se comenta la información de los materiales encontrados respecto al año de publicación. Seis materiales corresponden al 2015, seguido de cinco materiales el 2017, tres del 2016 y 2018, y dos materiales del 2014. Finalmente, solo un material para los años 2013, 2012, 2011 y 2009.

Tabla 14. Resumen de materiales por país. Fuente: Elaboración propia

ID	País	Cantidad
SCOPUS_PC_02, SCOPUS_PC_03, SCOPUS_PC_04, SCOPUS_PC_05, WOS_PC_01, WOS_PC_02, WOS_PC_03, WOS_PC_04, WOS_PC_05	China	9
SCOPUS_G_03, WOS_G_03	Reino Unido – Grecia	2
WOS_G_04, WOS_G_05	Portugal	2
WOS_G_01, WOS_G_01_2	Colombia	2
SCOPUS_G_01	México	1
SCOPUS_G_02	Hungría	1
SCOPUS_G_04	Países bajos	1
SCOPUS_G_05	Sri Lanka	1
WOS_G_02	Rusia	1
SCOPUS_PC_01	Francia – Canadá	1
WOS_PC_06	España	1
WOS_PC_07	Reino Unido	1

3.4 Informe de resultados

Las respuestas a las preguntas de investigación, con base a la información encontrada en los materiales obtenidos de las consultas realizadas por cada una de las estrategias son las siguientes.

A la PI.1 ¿Cuál es la definición usada del concepto pensamiento computacional? De los 13 materiales, todos los autores indican su uso para la resolución de problemas, algunos retoman el planteado inicialmente en 2006 por la autora Jeannette M. Wing (SCOPUS_PC_02, SCOPUS_PC_04, WOS_PC_02, WOS_PC_07) o hicieron alusión a una variación de la definición inicial, incluso Jeannette en el 2011 presentó una nueva definición. En el material WOS_PC_05 el autor atribuye el concepto a una persona diferente a Wing en el mismo año. El concepto de pensamiento computacional está fuertemente relacionado con las ciencias de la computación, así se comenta en 6 materiales (SCOPUS_PC_01, SCOPUS_PC_02, SCOPUS_PC_03, WOS_PC_03, WOS_PC_04, WOS_PC_05). Algunos autores incluyen habilidades específicas en la definición (SCOPUS_PC_05, WOS_PC_01, WOS_PC_07). En consecuencia, se tiene un marco de referencia del concepto pensamiento computacional:

El pensamiento computacional es un tipo de pensamiento analítico, un conjunto de estrategias cognitivas y metacognitivas emparejadas con procesos, habilidades y métodos de ciencias de la computación (análisis, abstracción, descomposición, razonamiento heurístico, planeación, programación, modelo, reconocimiento de patrones, algoritmo); su esencia es pensar acerca de datos e ideas, y usar y combinar estos recursos para resolver problemas, diseñar sistemas y comprender la conducta humana, de tal forma que una computadora – humana o máquina – pueda efectivamente llevar a cabo la solución.

A la PI. 2 ¿Cuál ha sido la actividad de las investigaciones de pensamiento computacional en educación superior?

Los materiales revisados, indican el uso del pensamiento computacional con la intención de promover, entrenar y cultivar habilidades en la resolución de problemas (SCOPUS_PC_03, SCOPUS_PC_04, SCOPUS_PC_05, WOS_PC_01, WOS_PC_02, WOS_PC_03) para mejorar la efectividad de aprendizaje en cursos de programación, a través del uso de lenguajes de programación específicos, tales como C, C++, visual basic o el entorno Scratch (SCOPUS_PC_02,

Revisión sistemática de la literatura

SCOPUS_PC_04, SCOPUS_PC_01), en consecuencia, los resultados de investigación indican que el pensamiento computacional es un apoyo en educación superior para entrenar la capacidad de pensamiento (SCOPUS_PC_01, WOS_PC_04, WOS_PC_05), y en consecuencia los estudiantes tengan mejores resultados académicos usando metodologías educativas (WOS_PC_06, WOS_PC_07). Destaca la actividad reportada en WOS_PC_03 que busca obtener una solución al problema de altas calificaciones y bajas habilidades de los estudiantes.

A la PI. 3 ¿Cómo se ha usado el pensamiento computacional en la enseñanza de la Programación de computadoras en la educación superior?

SCOPUS_PC_01 destaca por relacionar 6 componentes del pensamiento computacional con las 4 fases de la solución de problemas colaborativos de PISA 2015, para que la evaluación del pensamiento computacional impacte en las actividades de la programación de computadoras, es decir, organiza el contexto y contenido de la enseñanza como lo propone también SCOPUS_PC_03 y SCOPUS_PC_05. SCOPUS_PC_02 de igual forma determina dos habilidades del pensamiento computacional que sirven de impulso para mejorar la efectividad del aprendizaje de los estudiantes, pero auxiliándose en el método de enseñanza basado en la investigación o en casos reales como lo hacen en WOS_PC_02 y WOS_PC_05 o basado en problemas que se emplea en WOS_PC_04. En este sentido, la combinación de otras estrategias ofrece mejores resultados, SCOPUS_PC_03 lo refuerza también con los experimentos realizados o WOS_PC_05 que usa MOOC y micro lecciones. SCOPUS_PC_04 emplea el pensamiento computacional con la intención de favorecer el pensamiento de programación buscando tener impacto fuera del ambiente universitario, es decir, trabajo y vida del estudiante, o específicamente para mejorar el desempeño en cualquier plataforma de programación como da uso en WOS_PC_01. WOS_PC_06 destaca por establecer tres etapas para conseguir el aprendizaje: oír, ver, y hacer; que al aplicarlas para el desarrollo de habilidades del pensamiento computacional, se beneficia la resolución de problemas en el curso de programación propuesto. Finalmente, WOS_PC_07 usa el pensamiento computacional para que los estudiantes puedan hacer

buenas preguntas que permitan desenredar las estructuras de información conocidas y guiar el diseño de soluciones a problemas nuevos.

A la PI.1 ¿Cuál es la definición usada del concepto gamificación?

La mayoría de los materiales parten del concepto que indica el uso de elementos de juego en un contexto de no juego, lo que implica que no es exclusivo usarlo en el área de la educación; a excepción de SCOPUS_G_04, WOS_G02, WOS_G_03 y WOS_G_05 que no incluyen una definición. En SCOPUS_G_04 se expone la relación de gamificación con el uso de elementos de juego pero desde un punto de vista del efecto social que causa, en WOS_G_02 y WOS_G03 incluyen la motivación de los estudiantes como el objetivo de la gamificación, indicando que son actividades lúdicas del aprendizaje. Finalmente, WOS_G_05 comenta los beneficios de la gamificación con base a los trabajos revisados en las referencias. Destaca el material WOS_G_04 con una redacción formal del concepto: “El neologismo gamificación es aplicado para usar pensamiento de juego o mecanismo de juego para resolver problemas o enganchar a las audiencias”.

En consecuencia, se tiene un marco de referencia del concepto gamificación:

La gamificación es conocida como aquella que usa o integra elementos de juego y sus técnicas, reglas o mecánicas de diseño, en un contexto o escenario que no son de juego para ofrecer recompensa o motivación a los estudiantes en el diseño de cursos y su intervención.

A la PI. 2 ¿Cuál ha sido la actividad de las investigaciones de gamificación en educación superior?

Es precisamente la medición del impacto en el uso de la estrategia de gamificación en la enseñanza de programación la principal actividad de investigación en los materiales (SCOPUS_G_01, SCOPUS_G_03, SCOPUS_G_05, WOS_G_01_2, WOS_G_04), aunque no necesariamente en áreas

Revisión sistemática de la literatura

afines a las ciencias de la computación, tal es el caso de SCOPUS_G_02 que aplica gamificación en estudiantes de ciencias económicas, también no exclusivamente en educación superior, como en SCOPUS_G_05 que investiga la gamificación para mejorar el proceso de aprendizaje de estudiantes en varios niveles de edad. De forma muy clara SCOPUS_G_04 establece la pregunta de investigación: “¿Puede la gamificación ser efectiva para la enseñanza en cursos de educación superior?”, en consecuencia, se reportan resultados de mejora en los porcentajes de acreditación y participación, así como el aumento de satisfacción de los estudiantes. Para llevar a cabo la investigación, en WOS_G_02 se desarrolló una plataforma especial que incluye servicios accesibles en la nube, similarmente, en WOS_G_05 se diseñó un sistema de premios para ofertar una evaluación autorregulada, y así mantener informado al estudiante de lo que ha aprendido y lo que aún necesita aprender. En WOS_G_01_2 se indica el uso de gamificación como una estrategia en solitario o en combinación con otras estrategias para tener un impacto positivo en el aprendizaje de programación de computadoras. WOS_G_03 extiende la evaluación de la gamificación como estrategia para obtener motivación de los estudiantes e investiga la afectación que tiene en el estudiante para recordar, y en consecuencia también afecte el desempeño en un curso de programación de computadoras.

A la PI. 3 ¿Cómo se ha usado la gamificación en la enseñanza de la Programación de computadoras en la educación superior?

Las diferencias entre la forma de usar gamificación representan el valor del trabajo de investigación de cada uno de los materiales. SCOPUS_G_01 propone recompensas por las tareas completadas y en consecuencia un ambiente de competitividad al hacer visibles las obtenidas por los estudiantes. SCOPUS_G_02 describe que además de usar recompensas en la propuesta de gamificación, se auxilia de la plataforma *e-Learning Moodle* para proponer rutas de aprendizaje alternativo, así como varias opciones de retroalimentación; WOS_G_01_2 reafirma el uso de gamificación en combinación con otras estrategias. SCOPUS_G_03 trabaja con puntajes como elemento de gamificación para la etapa

de evaluación en el curso de programación *Python*. SCOPUS_G_04 detalla la propuesta de gamificación indicando las mecánicas, dinámicas y estéticas, este último elemento es novedoso en el sentido de cuidar el diseño de arte del juego. Un material que aplica gamificación en un curso diferente a programación pero si del área es SCOPUS_G_05, la intervención se lleva a cabo en el curso de estructura de datos y algoritmos, y destaca por incluir como recompensa una pulsera, es decir, un elemento físico que va más allá de una recompensa visual pero digital, tales como las insignias en WOS_G_02, que junto a las metas, retos y puntos forman su propuesta de gamificación para extender su uso fuera del salón de clases (actividades culturales, deportivas o de comunidad). De forma similar WOS_G_03, ofrece a los estudiantes una recompensa física, en este caso una playera al equipo sobresaliente, pero además un reconocimiento basado en un nombramiento simbólico, ser el pitonisa del año por responder correctamente preguntas de opción múltiple relacionadas con el lenguaje de programación *Python*. El uso que tiene la gamificación en WOS_G_04 está en sintonía con los materiales anteriores, pero además incluye su aplicación para proporcionar una clasificación automática de los estudiantes, como lo hace también SCOPUS_G_04 al determinar clases de jugadores: explorador, logros, social y ganador; en consecuencia, el profesor puede ofrecer una constante retroalimentación del progreso del estudiante durante el curso. Finalmente, WOS_G_05 utiliza varios mecanismos de diseño de juegos, como desafíos adaptativos, recompensas, curiosidad y la oportunidad de aumentar el tiempo que los estudiantes pasan trabajando, experimentando y aprendiendo. Las secciones en el currículo de la asignatura se transforman en niveles, cada uno representado por un castillo en un mapa, otorgando estrellas por logros cada vez más complejos. El material comenta el uso del concepto moneda, denominado *BitPoints*, que se puede utilizar para principalmente reducir la carga de trabajo, es decir, el dinero virtual se puede gastar en una tienda virtual, donde el estudiante puede invertir en herramientas que pueden ayudar a resolver algunas tareas: bloques de código, instrucciones y otras técnicas para la solución de los retos, así como un factor de oportunidad, dando mayor motivación a los estudiantes.

Revisión sistemática de la literatura

Para las estrategias de educación personalizada y *b-Learning* ningún material pudo aportar respuestas a las preguntas de investigación.

3.5 Limitaciones de la revisión

Aunque la búsqueda de material a través de la base de datos SCOPUS y WOS fue garantía de información con un alto nivel de calidad, solo se revisaron para las estrategias de pensamiento computacional y gamificación, por lo que será necesario consultar otras fuentes que aporten resultados de investigación en los temas de educación personalizada y *b-Learning*. La valoración de los materiales obtenidos solo fue realizada por el investigador, debido al interés por conocer algunos de los resultados obtenidos de otros investigadores y determinar el alcance del estudio al momento de establecer el plan de investigación, en consecuencia, se tiene un análisis subjetivo de los resultados y una interpretación del análisis desde un punto de vista personal. A pesar de lo anterior, se ha incluido en la tesis doctoral una rica referencia de investigaciones en especial para el tema de pensamiento computacional y gamificación. La educación personalizada está comenzando su periodo de publicación de resultados y será necesario revisar a futuro la variedad y el contraste con lo reportado hasta este momento, pero lo incluido en la tesis doctoral tiene un alto valor por la carencia de estudios en el contexto de la enseñanza de programación de computadoras.

3.6 Conclusiones

La revisión sistemática de literatura que se llevó a cabo permitió responder las preguntas de investigación de dos temas de interés en los últimos 10 años. El resultado obtenido se logró a partir de la ejecución de la metodología propuesta al inicio del capítulo. Dos fueron las bases de datos consultadas que arrojó 23 materiales para su estudio, es decir, SCOPUS y WOS para las estrategias de pensamiento computacional y gamificación. Debido al seguimiento del protocolo y la estrategia

de revisión no se reportaron materiales para las estrategias de educación personalizada y *b-Learning*. La segunda ejecución de la revisión establecida, con una diferencia de tiempo de tres meses, permite validar el protocolo y los resultados obtenidos inicialmente, todos los potenciales materiales encontrados de la primera ejecución fueron encontrados en la segunda, así se cumple un requisito de la revisión sistemática indicado al inicio del presente capítulo: una metodología auditable, rigurosa y confiable.

Los materiales revisados muestran el gran interés que existe por incluir el pensamiento computacional en el currículo de los estudiantes desde edad temprana, la promoción de su aplicación en la solución de problemas ha permitido crear competencias muy bien estructuradas e inclusive olimpiadas que han servido para la motivación de los participantes y guías académicas para los docentes de los niveles básicos de educación. A pesar de que no existe una definición definitiva del concepto, algunos autores coinciden en indicar las habilidades y su descripción para comprender qué es el pensamiento computacional. Respecto a la educación superior y su uso en el aprendizaje de programación, los materiales revisados reportan que durante los primeros cursos de los estudiantes se ha implementado el pensamiento computacional para fortalecer habilidades matemáticas y lógicas, en casos específicos para la solución de problemas sin importar el perfil profesional, es decir, una carrera del perfil diferente a las ciencias de la computación.

La gamificación, es generalmente utilizada para motivar a los estudiantes en un ambiente de aprendizaje que resulta complejo y en ocasiones aburrido, por tal motivo los materiales revisados reportan el uso de los elementos de juego como una combinación de actividades presenciales para mejorar los resultados académicos y enganchar a los estudiantes con el contenido de los cursos. La mayoría de los autores están de acuerdo en definir la gamificación como la estrategia que usa elementos de juego en contexto que no lo son. Sobresale el uso de recompensas físicas o virtuales que ganan los estudiantes a partir de la obtención de puntos, con base al cumplimiento de tareas o resolver retos en el ambiente determinado por el docente.

Revisión sistemática de la literatura

La revisión sistemática realizada, como se mencionó anteriormente, careció de materiales para las estrategias de educación personalizada y *b-Learning* en educación superior en cursos de programación. Respecto a la educación personalizada, ésta es un reto por la meta de ofrecer aprendizaje individualizado, particularmente en el ambiente de la educación superior pública, la tarea del docente es abrumadora en todos los niveles educativos; así, el uso de las tecnologías de la información como herramienta puede permitir auxiliar su implementación, desde oferta curricular adaptable, elección del orden de los contenidos, hasta flexibilidad de evidencias evaluativas, en consecuencia, la inclusión inevitable del *b-Learning* como entorno de aprendizaje.

Una consecuencia inmediata de la revisión sistemática es la justificación del escenario de investigación de la tesis doctoral. Una combinación de los temas de interés, es decir, usar la evaluación del pensamiento computacional para determinar una guía académica al momento de proponer educación personalizada, que aporte resultados en beneficio de la enseñanza de la programación de computadoras en la educación superior con indicadores específicos, tales como el porcentaje de acreditación, deserción y calificación. El complemento a la combinación comentada se encuentra en primer lugar en el uso de la gamificación como estrategia de motivación y en segundo lugar del *b-Learning* como escenario de aprendizaje.

BLOQUE II: MARCO EMPÍRICO

El bloque contiene propiamente la implementación empírica del trabajo de investigación. Las intervenciones descritas en los dos capítulos que le integran permiten visualizar el proceso de la metodología mixta realizado una vez concluida la revisión sistemática de la literatura. Inicialmente se detalla el diseño experimental, es decir, la fase de planeación donde se determinaron usar en la intervención uno la combinación de las estrategias educativas: pensamiento computacional, educación personalizada y *b-Learning*. En la segunda intervención se determinó usar las estrategias de gamificación, educación personalizada y *b-Learning*. La justificación de tal selección está asociada con la necesidad de establecer el punto de partida de cada estudiante de nuevo ingreso, la evaluación del pensamiento computacional representó el medio para realizarlo. En consecuencia, para la segunda intervención, al no existir la necesidad primaria de conocer el estatus del estudiante, se determinó trabajar con la motivación. Por otro lado, cada intervención tuvo dos iteraciones, para cada iteración se indica qué enfoque va aportando en el experimento (CUAN, CUAL), así como desde que enfoque se fue obteniendo información. Lo anterior proporciona una aportación a la investigación realizada pues se describen resultados a partir de dos ejercicios empíricos, validando el trabajo realizado en la fase de planeación, enriqueciendo la fase de intervención y generando datos en la fase de análisis. Después de los resultados obtenidos de la primera intervención, se tuvo la oportunidad de realizar la fase de planeación para otra institución educativa, no solo diferente al modelo educativo de la UTP, sino también de país diferente a México. La última sección del capítulo 4, contiene el trabajo de la estancia realizada en INACAP Chile. Es de vital importancia el trabajo llevado a cabo debido a que permite exponer la adaptabilidad, flexibilidad y articulación de parte del trabajo de investigación realizado en un contexto diferente al caso de estudio para el cual fue diseñado.

CAPÍTULO 4. INTERVENCIÓN CON PENSAMIENTO COMPUTACIONAL, EDUCACIÓN PERSONALIZADA Y B-LEARNING

4.1 Introducción

La intervención que se describe en el presente capítulo se fundamenta en la evaluación del pensamiento computacional a los estudiantes de nuevo ingreso, particularmente las habilidades de generalización, descomposición, abstracción, diseño algorítmico y evaluación; con base al resultado de la evaluación del pensamiento computacional se ofertaron escenarios de aprendizaje, es decir, educación personalizada, así como el uso de la plataforma *Moodle* como herramienta para un entorno *b-Learning*.

En la primera intervención se propuso rediseñar la práctica docente para impactar en el aprendizaje de la programación, específicamente la asignatura Metodología de la programación (MP), con el objetivo de mejorar los indicadores de estudiantes acreditados, la calificación promedio del grupo y reducir el porcentaje de deserción.

La estrategia de educación personalizada estuvo fundamentada por 4 elementos:

- i. Contenido.
- ii. Formas de trabajo
- iii. Ritmos y tiempo
- iv. Opciones de evaluación.

El trabajo realizado tuvo la novedad de evitar un contenido único para todos los estudiantes de la asignatura MP que está incluida en el primer cuatrimestre de la carrera Técnico Superior Universitario de la división de Tecnologías de la Información y Comunicación (TIC) en la Universidad Tecnológica de Puebla (UTP); cada septiembre inicia una nueva generación y desde el departamento de servicios escolares se van formando los grupos para el turno matutino y vespertino que en promedio suman

Intervención con pensamiento computacional, educación personalizada y *B-Learning*

doce de aproximadamente treinta estudiantes cada uno. La dirección de la división de TIC asigna a los profesores que imparten la materia con base a la disponibilidad reportada por los profesores de asignatura o tiempo completo.

Es importante mencionar que la intervención se realizó en dos ocasiones, es decir, tuvo dos iteraciones. La primera en el cuatrimestre septiembre – diciembre de 2016, donde siguiendo la metodología de investigación se elaboró la fase planeación determinando la forma de implementar las estrategias educativas de pensamiento computacional, educación personalizada y *b-Learning*; posteriormente se ejecutó la fase intervención y en consecuencia se tuvieron datos para la fase analítica. En la fase discusión de la primera iteración se detectaron acciones de mejora para la segunda iteración en el cuatrimestre septiembre – diciembre de 2017 y en fase final de esta última iteración tener resultados conclusivos desde ambos enfoques de la metodología mixta de triangulación concurrente.

4.2 Metodología

4.2.1 Población y muestra

Acerca de los grupos de control, para la intervención en el cuatrimestre septiembre – diciembre de 2016, los grupos 1°A, 1°B, 1°E, 1°F y 1°G sirvieron para comparar resultados de los grupos experimentales, debido a que los profesores respectivos compartieron las calificaciones de los estudiantes. Para la intervención en el cuatrimestre septiembre – diciembre de 2017 no compartieron la calificación ninguno de los grupos ajenos a los grupos experimentales.

Los grupos experimentales de la primera iteración para el curso MP en el cuatrimestre septiembre – diciembre 2016 fueron los grupos 1°C y 1°D con 32 y 33 estudiantes en cada grupo; para la segunda iteración en el cuatrimestre septiembre – diciembre 2017 fueron los grupos 1°B y 1°H con 35 estudiantes en cada grupo. La Tabla 15 concentra los datos mencionados.

Tabla 15. Datos de grupos de control y experimentales. Fuente: Elaboración propia

Cuatrimestre	Año	Grupos	
		Control	Experimental
Septiembre - Diciembre	2016	165 1°A, 1°B, 1°E, 1°F y 1°G	65 1°C y 1°D
	2017	0	70 1°B y 1°H

4.2.2 Fase planeación

La fase planeación durante la metodología mixta desarrollada está formada por tres acciones principales. Primero, el trabajo realizado para ofertar educación personalizada a través de cuatro elementos que la gestionan. Segundo, la creación de la evaluación del pensamiento computacional. Y tercero, el resultado de la aplicación de los dos primeros por medio del establecimiento de los escenarios de aprendizaje que han de fomentar un entorno *b-Learning*. El trabajo realizado de esta forma, tuvo la intención de crear el contexto adecuado para obtener información desde el enfoque CUAN y CUAL para la fase analítica.

4.2.2.1 Educación personalizada

Los cuatro elementos de la estrategia de educación personalizada son:

i Contenido

Los conocimientos generales del curso MP están indicados en la Tabla 16. La estrategia de educación personalizada fue apoyada por el uso de la plataforma institucional *Moodle* que permitió organizar los contenidos de tal forma que los estudiantes pudieron elegir sus herramientas de aprendizaje. Se crearon tres unidades, pero etiquetadas como etapas, con el fin de indicar un aprendizaje progresivo; cada etapa corresponde a las unidades temáticas del curso y contiene el objetivo para que el estudiante conozca cuando ha superado la respectiva etapa. En el encabezado del curso se agregaron dos archivos, el primero contiene el encuadre de la materia y el segundo una definición del concepto tecnologías de la información, además del objetivo de la asignatura y la encuesta final, que fue

Intervención con pensamiento computacional, educación personalizada y *B-Learning*

habilitada en su respectivo momento al igual que la encuesta a mitad del cuatrimestre, ambas encuestas midieron principalmente la aceptación de la combinación de estrategias. También en cada etapa hay carpetas tituladas como Lecturas, Audios y Evaluaciones donde se encuentran los materiales de estudio en su respectivo formato y archivos de texto con los ejercicios de evaluación; se agregaron actividades tipo Lección para alojar videos y actividades en línea (crucigramas y sopa de letras) de los temas según la unidad temática; finalmente, se encuentran foros donde los estudiantes responden a preguntas de los temas, suben las evidencias de evaluación y comentan las aportaciones de los compañeros. La Figura 4.1 muestra el aspecto del curso Metodología de la programación.

ii Formas de trabajo

Las actividades que se propusieron para ofrecer al estudiante un entorno inicial de aprendizaje con base a los reactivos contestados correctamente de la evaluación del pensamiento computacional, están determinadas en 10 escenarios de aprendizaje usando tres espacios de trabajo: presencial, semi-presencial y en línea. En el espacio presencial las clases se llevaron a cabo de forma tradicional, es decir, actividades prácticas y teóricas como lo indica el currículo de la asignatura en el horario asignado por la dirección de la carrera. En el espacio semi-presencial se aprovechó el uso de la plataforma *Moodle* para adaptar los conocimientos que debía aprender el estudiante según los reactivos correctos de la evaluación del pensamiento computacional, es en esta modalidad donde se dieron mayores opciones a los estudiantes y se organizaron días de asesorías. En el espacio en línea el estudiante podía solicitar asesorías para aclarar dudas o entregar las evidencias de evaluación.

iii Ritmos y tiempos

El objetivo del curso y de cada unidad temática fue el mismo para todos los estudiantes, pero cada estudiante determinó su ritmo de aprendizaje, fue una oportunidad para que el estudiante eligiera qué y a que ritmo aprender.

iv Opciones de evaluación

En la primera iteración septiembre – diciembre de 2016, los ejercicios y actividades evaluativas fueron ofrecidas para que el estudiante las eligiera, tales como codificar programas, diseñar diagramas de flujo o pseudocódigo y de acuerdo a la forma de trabajo la revisión de las evidencias fue presencial o en línea; había una fecha límite de entrega, pero los estudiantes acordaron con el docente el día para su revisión. En la segunda iteración, realizada en septiembre – diciembre de 2017, se incluyeron además listas de cotejo para los productos de evaluación determinados por la academia de profesores, así los estudiantes conocieron los puntos clave a considerar en la evaluación, con el objetivo de tener una mejor comunicación durante el cuatrimestre y no solo durante la revisión de los trabajos, productos o evidencias de evaluación. Cinco fueron las listas de cotejo integradas en la segunda iteración de la intervención, la Tabla 17 describe los elementos principales de evaluación de cada una de las listas.

Tabla 16. Conocimientos generales del curso Metodología de la programación. Fuente: Elaboración propia

Unidad Temática	Conocimientos
1	Tipos de datos Identificadores de variables
2	Operadores aritméticos Operadores lógicos Operadores relacionales Jerarquía de operadores Resolver expresiones
3	Uso de variable contador y acumulador Estructura de selección (condicional) Estructura de repetición (ciclo) Diagrama de flujo Diseño de algoritmos



Figura 6. Aspecto del curso Metodología de la programación en la plataforma Moodle. Fuente: Elaboración propia

Tabla 17. Indicadores de evaluación en listas de cotejo. Fuente: Elaboración propia

Lista de cotejo Producto	Habilidad o conocimiento a evaluar	Correspondencia con unidad temática del curso
1	Tipos de datos, uso correcto de variables, constantes, operaciones aritméticas básicas e identificación de valores de entrada y salida.	1. Conceptos básicos
2	Creación de expresiones algorítmicas, jerarquía de operadores aritméticos, lógicos y algebraicos.	2. Expresiones
3	Estructuras de control secuenciales y condicionales en algoritmos	3. Algoritmos y diagramas de flujo
4	Estructura de control repetición en algoritmos	3. Algoritmos y diagramas de flujo
5	Creación de métodos o funciones con parámetros de entrada y/o salida	3. Algoritmos y diagramas de flujo

4.2.2.2 Evaluación del pensamiento computacional

El objetivo de la asignatura, de cada unidad temática, así como la descripción del Saber y Saber Hacer de cada tema está redactado en la hoja de asignatura con base a una categoría cognitiva de la

taxonomía de Bloom. Para determinar los escenarios de aprendizaje, se definió una relación entre los conocimientos de la asignatura MP y el pensamiento computacional como se describe a continuación:

- La unidad temática 1 titulada “*Conceptos básicos*”, tiene seis temas y en la redacción del pilar *Saber* están los verbos Identificar, Reconocer y Describir que corresponden al nivel dos de la taxonomía de Bloom, en el pilar *Saber Hacer* el verbo Determinar, también del nivel dos de Bloom y otro del cuarto, Esquematizar, por lo que se encuentran en el nivel de Comprender y Analizar, en consecuencia, se eligió que correspondan a las habilidades de Abstracción y Descomposición.
- La unidad temática 2 nombrada “*Expresiones*”, usa en los tres temas el verbo Identificar para el pilar *Saber*, que corresponde al nivel dos de Bloom, mientras que en el pilar *Saber Hacer* se usa en tres ocasiones verbos del nivel tres Aplicar por lo que se eligió corresponda a la habilidad de Generalización.
- La unidad temática 3: “*Algoritmos y diagramas de flujo*” tiene cuatro temas, en el pilar *Saber* aparecen verbos del nivel cognitivo dos Comprender y en el pilar *Saber Hacer* un verbo del nivel cinco Evaluar, además de indicar la programación de ejercicios usando el paradigma de programación estructurada por lo que se eligió que correspondan a las habilidades de Evaluación y Diseño algorítmico.

La Tabla 18 contiene la relación establecida de las unidades temáticas del curso Metodología de la programación y las habilidades del pensamiento computacional: Descomposición, generalización, abstracción, diseño algorítmico y evaluación, así como el nombre del reactivo que la evalúa.

Por lo anterior, aprovechando la relación mencionada y la forma en que están determinados los contenidos de la materia se eligieron los reactivos para evaluar el pensamiento computacional de los estudiantes de nuevo ingreso y ofrecer un ambiente inicial de aprendizaje donde se motive a cursar la materia con base a las habilidades detectadas.

Tabla 18. Relación de habilidades del pensamiento computacional y conocimientos del curso MP.

Fuente: Elaboración propia

Unidad Temática	Verbos Usados	Nivel taxonomía Bloom	Habilidad PC – ejercicio
1 Conceptos básicos, 6 temas	Saber: Identificar, Reconocer y Describir	2 Comprender	Abstracción Canguro
	Saber Hacer: Esquematizar y Determinar	4 Analizar	Descomposición Móviles
2 Expresiones, 3 temas	Saber: Identificar	2 Comprender	Generalización Espías
	Saber Hacer: Localizar, Resolver y Convertir	3 Aplicar	
3 Algoritmos y diagramas de flujo, 4 temas	Saber: Reconocer, Identificar y Describir	2 Comprender	Evaluación Salto de charcos
	Saber Hacer: Comparar y Resolver	5 Evaluar	Diseño algorítmico Castores en movimiento

Los reactivos se eligieron de dos fuentes reconocidas a nivel Internacional: el concurso sobre fluidez informática y computacional en edades escolares *UK Bebras Computational Thinking Challenge* (2015), y la Olimpiada de computación de búsqueda de talento (Talent Search, 2015) que pretende orientar a los estudiantes sudafricanos con más aptitudes en pensamiento computacional hacia carreras técnicas; debido a que su redacción original es en inglés, se realizó una traducción al español. El motivo por el cual solo se seleccionó un ejercicio para cada una de las habilidades del pensamiento computacional, es el tiempo utilizado para responder la evaluación, un promedio de 40 minutos, no es intención que el estudiante invierta mucho tiempo en responder ejercicios; la validación de cada pregunta está garantizada al ser instrumentos de fuentes internacionales usadas en eventos de competencia, aunque cada ejercicio seleccionado no se limita a evaluar exclusivamente la habilidad para la cual fue seleccionada. Para la evaluación del pensamiento computacional a través de los reactivos seleccionados, se creó un formulario en las aplicaciones de *Google* desde la cuenta

institucional de la UTP, los estudiantes la contestaron dentro de la universidad para asegurar que se realizara.

El ejercicio de Bebras denominado Móviles evalúa en gran medida la habilidad de Descomposición.

La redacción del ejercicio es la siguiente:

Un móvil es una pieza de arte que cuelga del techo, generalmente en los dormitorios. Un móvil consiste de palos y figuras. Cada palo tiene unos cuantos puntos donde figuras u otros palos pueden ser atados. Además, cada palo tiene un punto para colgar, donde se cuelga a un palo hacia abajo (o hacia el techo). La Figura 7 es un ejemplo de móvil que puede ser descrito usando números y paréntesis, Pregunta: ¿Cuál de los siguientes móviles (Figura 8) puede ser construido usando las siguientes instrucciones? $(-3 (-1 4) (2 (-1 1) (1 1))) (2 (-1 6) (2 3))$.

El estudiante debe seleccionar la imagen correcta a partir de establecer que la relación de la imagen con la expresión de los números y paréntesis conduce a determinar que el móvil es una composición de pequeños móviles. La respuesta correcta es la opción A.



Figura 7. Descripción del móvil con una expresión: $(-3 (-1 1) (1 1)) (2 3)$. Fuente: Bebras (2015)

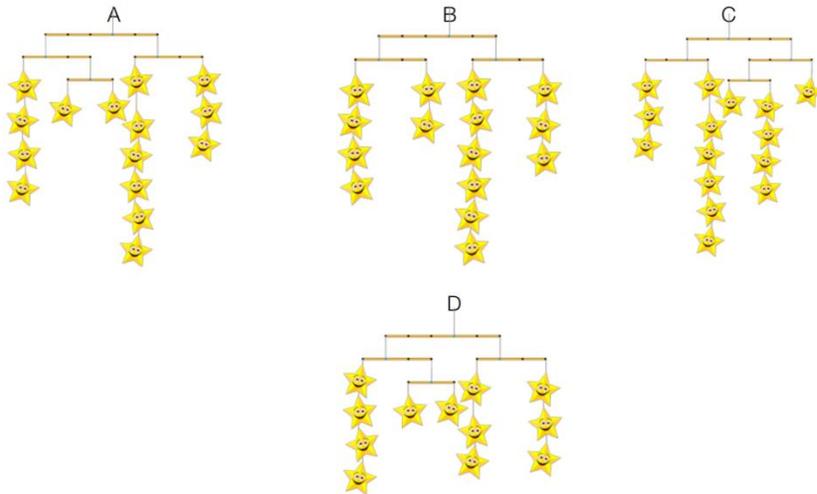


Figura 8. Opciones de respuestas del reactivo móvil. Fuente: Bebras (2015)

El ejercicio nombrado Canguro, también de Bebras, destaca por su evaluación de la Abstracción pues el estudiante decide, analiza y determina qué información le va asistiendo para obtener la respuesta correcta del problema planteado, ver Figura 9, la respuesta es Plato I.

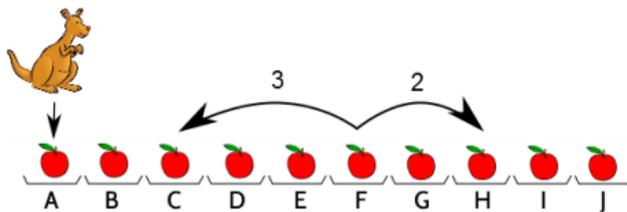


Figura 9. La tarea de recoger las manzanas. Fuente: Bebras (2015)

La redacción del ejercicio anterior en la evaluación es la siguiente:

Hay 10 platos en una fila. Hay una manzana en cada plato. Al canguro Tomás le encanta saltar. Primero, el salta desde el plato más a la izquierda con la letra A. En cada salto después del inicial, salta dos platos hacia adelante, o tres platos hacia atrás. (Un ejemplo de los dos posibles saltos desde un plato es mostrado con flechas en la imagen.)

Tomás sólo salta hacia platos con una manzana. Si el salta hacia un plato, recoge la manzana. Pregunta: Si Tomás recoge todas las 10 manzanas, ¿cuál manzana recoge al final? A, B, C, D, E, F, G, H, I, o J.

El tercer ejercicio tomado de Bebras con nombre Espías, se enfoca en evaluar la habilidad de Generalización, pues una vez que verifica la solución indicada para el problema inicial, aplica a una situación similar la forma de resolver el problema que contiene características parecidas. La respuesta correcta es 4. La redacción del ejercicio es la siguiente:

Cada viernes, seis espías intercambian toda la información que han reunido durante la semana. Un espía nunca puede ser visto con más de otro espía al mismo tiempo. Así, tienen que tener varios encuentros de reunión en pares y compartir la información que poseen. El grupo de 6 espías sólo necesitan tres encuentros para distribuir todos sus secretos. Antes del encuentro cada espía mantiene una sola pieza de información (espía 1 conoce 'a', espía 2 conoce 'b', etc.). En el primer encuentro espía 1 y 2 se encuentran y comparten información entonces ahora ambos conocen 'ab'. La Figura 10 muestra cuales espías se encuentran en cada reunión a través de una línea. También muestra cuáles piezas de información tienen todos. Después de tres encuentros toda la información ha sido distribuida. Pregunta: Después de un incidente internacional un espía ha dejado de atender los encuentros. ¿Cuál es el número mínimo de reuniones necesitadas por los cinco espías restantes para intercambiar toda la información? (ver Figura 11).

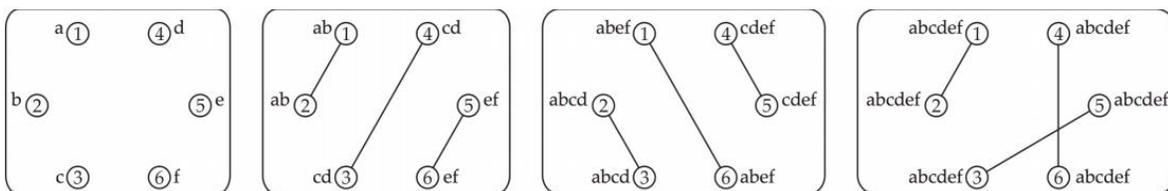


Figura 10. Seis espías intercambiando información. Fuente: Bebras (2015)

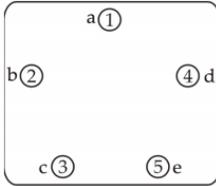


Figura 11. Cinco espías intercambiando información. Fuente: Bebras (2015)

Dos reactivos se tomaron de la Olimpiada de Computación. El primero nombrado Castores en movimiento se eligió para evaluar la habilidad del Diseño algorítmico debido a que el estudiante que lo resuelva muestra su capacidad de entender la organización de instrucciones para resolver un problema, en este caso las indicaciones para que los castores crucen los hoyos. La respuesta es 2 1 6 5 3 4 7. La redacción del ejercicio en la evaluación es la siguiente:

Una colonia de castores está viajando a través de un bosque oscuro. El camino es estrecho, así que tienen que viajar en una fila sin pasar uno del otro. Algunas veces hay un hoyo en el camino. Un hoyo es cruzado de la siguiente manera:

- Primero saltan tantos castores sean necesarios para llenar el hoyo.
- La colonia entera pasará entonces a través del hoyo.
- Los castores que saltaron treparán para salir del hoyo, y unirse al final de la línea.

La Figura 12 muestra cómo cinco castores pasan un pequeño hoyo que se llena con tres castores.

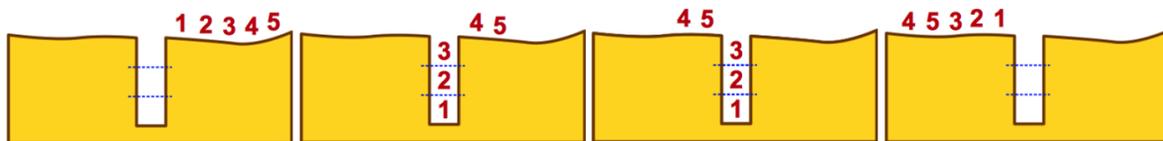


Figura 12. Ejemplo de cinco castores al cruzar un hoyo. Fuente: Talent Search (2015)

Una colonia de 7 castores pasa a través del bosque. Cruzan 3 hoyos. El primer hoyo se ajusta a 4 castores, el segundo se ajusta a 2, y el último hoyo se ajusta a 3 castores, ver

Figura 13, ¿En qué orden se encontrarán los castores después de que hayan pasado el tercer hoyo?

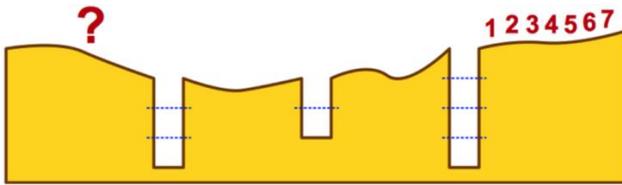


Figura 13. Siete castores buscando cruzar hoyos. Fuente: Talent Search (2015)

El segundo reactivo, de la Olimpiada de Computación, con nombre Salto de charcos, se escogió para determinar la habilidad de Evaluación, es decir, realizar paso a paso las instrucciones indicadas para completar el ejercicio y obtener la respuesta solicitada. La solución es Berta, Dora, Carlos, Ana y Luisa. La redacción del reactivo es la siguiente:

Ana (edad 7), Berta (edad 8), Carlos (edad 9), Dora (edad 10) y Luisa (edad 11) están jugando un juego donde saltan de un charco a otro. Ellos han ubicado flechas entre los charcos, y todos inician del lado izquierdo como se indica en la Figura 14. Cuando un niño salta dentro de un charco él o ella espera la llegada de un segundo niño. El niño mayor en el charco entonces saltará de acuerdo a la flecha gruesa, el más joven sigue la flecha delgada. ¿Cuál es el orden (de arriba hacia abajo) en el cual los niños terminarán a la derecha?

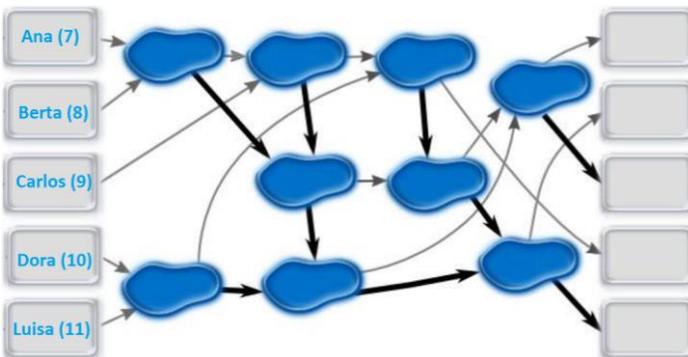


Figura 14. Salto de Charcos. Fuente: Talent Search (2015)

Intervención con pensamiento computacional, educación personalizada y *B-Learning*

Los reactivos para la iteración 2 fueron los mismos usados en la iteración 1, no hubo necesidad de modificarlos o elegir otros debido a que ya no están disponibles al público y no existió comentario alguno que indicara que no se comprendían en las encuestas realizadas en la iteración 1. Respecto al contenido de la plataforma en *Moodle*, no se alteró su organización. El ajuste realizado en la segunda iteración en la fase planeación que se llevó a cabo con la intención de tener un seguimiento puntual de los estudiantes fue la integración de listas de cotejo.

4.2.2.3 Escenarios de aprendizaje

La descripción de cada escenario de aprendizaje para impactar en el logro de los objetivos del curso MP de forma personalizada es la siguiente:

Escenario 1. 5 respuestas correctas

El estudiante tiene derecho al material del curso en línea y puede realizar quince días después de iniciado el cuatrimestre la evaluación de las unidades temáticas a través de los exámenes elaborados usando la plataforma *Moodle* y entregando los ejercicios prácticos determinados por la academia, de ser aprobatorios los resultados, el estudiante acredita el curso, en caso contrario se determina una fecha para la revisión presencial docente-estudiante de los ejercicios, se aclaran dudas y si es decisión del estudiante se determinan nuevos ejercicios prácticos a evaluar o tomar el curso en línea.

Escenario 2. Incorrecto el ejercicio de Castores en movimiento

Reforzar el diseño algorítmico con ejercicios a través de la plataforma *Moodle*, realizar un mes después de iniciado el cuatrimestre las evaluaciones del curso completo y determinar si acredita la materia o elige tomarlo en línea.

Escenario 3. Incorrecto el ejercicio de Salto de charcos

Reforzar las estructuras secuenciales, decisión y repetición codificando ejercicios específicos que resuelven un problema usando las estructuras de control. Calendarizar asesorías presenciales para esos temas y prácticas de laboratorio, realizar un mes después de iniciado el cuatrimestre las evaluaciones del curso completo y determinar si aprueba la materia o elige tomarla en línea.

Escenario 4. Incorrectos los ejercicios de Castores en movimiento y Salto de charcos

El estudiante tiene derecho al material del curso en línea y realizar actividades de práctica para programar los ejercicios de programación estructurada propuestos por la academia. Puede calendarizar tres sesiones de asesoría presencial con el profesor y solicitar al mes de iniciado el cuatrimestre la evaluación de los exámenes en línea y los ejercicios prácticos. Si acredita sus evaluaciones, también el cuatrimestre, en caso contrario puede elegir por tomar el curso en línea o semi-presencial.

Escenario 5. Incorrecto el ejercicio de Espías

Reforzar la evaluación de expresiones aritméticas, lógicas y relacionales considerando la jerarquía de operadores por medio de codificar ejercicios que usen cálculo de operaciones y condicionales; realizar un mes después de iniciado el cuatrimestre las evaluaciones del curso completo y determinar si acredita la materia o elige tomarlo en línea.

Escenario 6. Incorrecto los ejercicios de Castores en movimiento, Salto de charcos y Espías

El estudiante puede trabajar un modelo semi-presencial, donde el material del curso y las actividades deberán ser liberados con base al tiempo determinado en la hoja de asignatura; el estudiante puede solicitar sesiones de asesoría con el profesor del curso, realizar las prácticas y los exámenes en los momentos marcados por la academia.

Intervención con pensamiento computacional, educación personalizada y *B-Learning*

Escenario 7. Correcto el ejercicio de Canguro o Móvil

El estudiante tiene habilidades básicas, pero necesarias para el aprendizaje particularmente de programación, por lo que se puede optar por un entorno semi-presencial, pero con una atención de asesoría presencial regular en las prácticas de laboratorio. Realizar las evaluaciones departamentales indicadas por la academia y ejercicios prácticos evaluativos para desarrollar las habilidades que requieren refuerzo, es decir, Generalización y Diseño algorítmico.

Escenario 8. Incorrecto en todos los reactivos

Existe la posibilidad de que el estudiante no tenga habilidades para el estudio de la carrera y en la unidad temática tres sea difícil la comprensión de las estructuras de control y diseño algorítmico, por lo que se recomienda en reunión presencial docente-directivo-estudiante valorar el perfil vocacional de este último. Es muy seguro que el curso presencial sea la mejor opción para el estudiante.

Escenario 9. Incorrectos los ejercicios de Móviles y Canguro

El estudiante trabajará con un modelo semi-presencial, deberá elegir un día a la semana para práctica de laboratorio.

Escenario 10. Correctos los ejercicios de Castores en movimiento y Salto de charcos

El estudiante puede estudiar en línea las primeras dos unidades temáticas, después de cinco semanas trabajar con un modelo semi-presencial.

4.2.3 Instrumentos

Finalmente, dentro de esta fase, se establecieron los instrumentos que permitieron obtener información para cada una de los enfoques. Para el enfoque CUAN, además de obtener los resultados de la evaluación del pensamiento computacional, fue importante obtener la satisfacción de los

escenarios de aprendizaje (educación personalizada), en combinación con el uso del sistema gestor de aprendizaje (*b-Learning*) y la adquisición de competencias propias del curso de Metodología de la programación. Por ello se elaboraron dos encuestas para su aplicación a mitad (Tabla 19) y final del cuatrimestre (Tabla 20). La primera parte del instrumento al final del curso preguntó por los conocimientos adquiridos, posteriormente se cuestionó la aprobación de la modalidad de aprendizaje, así como la evaluación inicial del pensamiento computacional, a través de las habilidades de descomposición, generalización, abstracción, diseño algorítmico y evaluación; finalmente se pidieron recomendaciones para estudiantes de futuras generaciones. Para el enfoque CUAL, se agregaron en las encuestas reactivos abiertos, además de considerar la observación que realizara el docente del grupo durante la intervención en sus dos iteraciones.

Tabla 19. Instrumento encuesta para obtener información CUAN-CUAL a mitad del cuatrimestre en intervención 1. Fuente: Elaboración propia

Pregunta	Opción de respuesta
La modalidad de trabajo te parece adecuada con tu expectativa de aprendizaje	Si No
¿Conoces los objetivos a los que tienes que llegar o tienes claro el conocimiento y lo que debes saber hacer al final del curso?	Si No
¿Te sientes perdido usando la plataforma, no sabes qué hacer y para qué?	Si No
De los recursos contenidos en la plataforma ¿Cuál usaste?	Audio Video Lectura Actividades
¿Qué acción sugieres para mejorar el aprendizaje o estás de acuerdo con tu entorno de aprendizaje?	Abierta

Tabla 20. Instrumento encuesta para obtener información CUAN-CUAL al final del cuatrimestre en intervención 1. Fuente: Elaboración propia

Pregunta	Opción de respuesta
Selecciona los conceptos con los que estás familiarizado	Opción Múltiple
Tipos de datos	
Operadores aritméticos	
Creación de identificadores para variables	
Operadores lógicos	
Operadores relacionales	
Jerarquía de operadores	
Resolver expresiones aritméticas, lógicas, y relacionales	
Uso de una variable contador y acumulador	
Estructura de selección	
Estructura de repetición	
Definición y creación de un algoritmo	
¿La modalidad de aprendizaje fue adecuada para adquirir las competencias del curso?	Si No
¿La evaluación de tus habilidades al inicio del cuatrimestre fue una actividad acertada para determinar el mejor entorno de aprendizaje?	Si No
¿Qué recomendación tienes para las futuras generaciones acerca de la forma de aprender el contenido del curso?	Abierta

4.2.4 Fase intervención

Iteración septiembre – diciembre de 2016

En los primeros días de septiembre de 2016, todos los estudiantes de ambos grupos experimentales, un total de 65 estudiantes de nuevo ingreso, hicieron la evaluación del pensamiento computacional en el laboratorio asignado para los grupos 1°C y 1°D. La Figura 15 muestra los porcentajes según la respuesta elegida por los estudiantes al ejercicio de Móvil, solo un 33,8% contestó correctamente. Para el reactivo Canguro un 52,3% contestó correctamente como lo muestra la Figura 16. Un 36,9% de los estudiantes contestó correctamente el ejercicio relacionado con los Espías, el resto de los porcentajes está indicado en la Figura 17. El cuarto reactivo Castores en movimiento obtuvo el mayor porcentaje de respuesta correcta con un valor de 73,8%, la Figura 18 contiene también los porcentajes de las otras opciones elegidas pero que son incorrectas. Finalmente, el último ejercicio Salto de charcos tuvo un porcentaje del 67,7% de respuestas correctas, la Figura 19 muestra la distribución de los porcentajes para el último reactivo. Hay que observar que en todos los reactivos los estudiantes eligieron todas las opciones, no existe un contraste entre respuesta correcta y una incorrecta, lo que

generó que se aplicaran más de tres de los escenarios propuestos de aprendizaje. La cantidad de estudiantes que se tuvo para cada escenario de los 2 grupos experimentales, después de la evaluación del pensamiento computacional, está indicada en la Tabla 21, así como el acuerdo grupal más destacable según el escenario de aprendizaje.

Ejercicio 1

¿Cuál de los anteriores móviles puede ser construido usando éstas instrucciones? (-3 (-1 4) (2 (-1 1) (1 1))) (2 (-1 6) (2 3))

65 respuestas

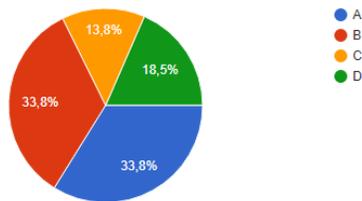


Figura 15. Porcentajes reactivo móvil iteración 1. Fuente: Elaboración propia

Ejercicio 2

Si Tomás recoge todas las 10 manzanas, ¿Cuál manzana recoge al final? A, B, C, D, E, F, G, H, I, o J.

65 respuestas

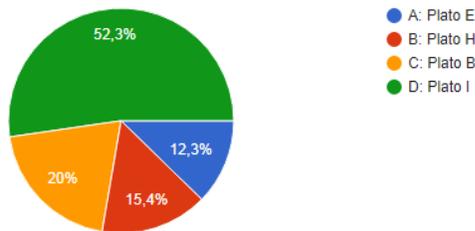


Figura 16. Porcentajes reactivo canguro iteración 1. Fuente: Elaboración propia

Ejercicio 3

¿Cuál es el número mínimo de reuniones necesitadas por los cinco espías restantes para intercambiar toda la información?

65 respuestas

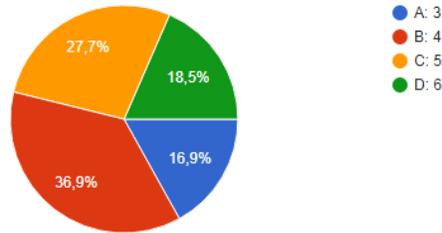


Figura 17. Porcentajes reactivo espías iteración 1. Fuente: Elaboración propia

Ejercicio 4

¿En qué orden se encontrarán los castores después de que hayan pasado el tercer hoyo?

65 respuestas

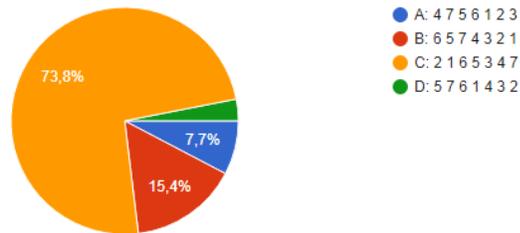


Figura 18. Porcentajes reactivo castores en movimiento iteración 1. Fuente: Elaboración propia

Ejercicio 5

¿Cuál es el orden (de arriba hacia abajo) en el cual los niños terminarán a la derecha?

65 respuestas

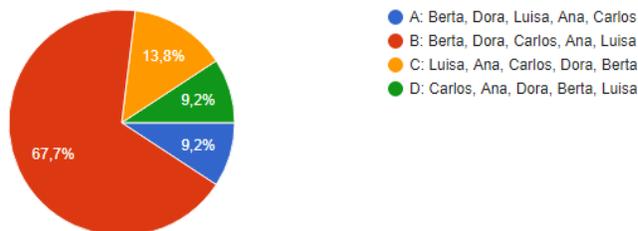


Figura 19. Porcentajes reactivo salto de charcos iteración 1. Fuente: Elaboración propia

Tabla 21. Número de estudiantes por escenario iteración 1. Fuente: Elaboración propia

Grupo	Escenario	Estudiantes	Observaciones
1°C	5	6	11 de octubre fecha para revisión de productos
	7	5	Viernes para práctica en laboratorio
	9	8	Jueves para práctica en laboratorio
	10	13	18 de octubre inicio de etapa 3
1°D	1	3	30 de septiembre revisión de evidencias de todo el curso
	5	2	7 de octubre revisión de evidencias de todo el curso
	7	10	Lunes para práctica en laboratorio
	9	14	Los estudiantes indicaron que día querían asesoría
	10	4	17 de octubre inicio de etapa 3

A mitad del cuatrimestre se solicitó a los estudiantes contestar voluntariamente la encuesta diseñada, los valores porcentuales de las respuestas están indicados en la Tabla 22, desafortunadamente solo participaron 24 de los 65 estudiantes.

Tabla 22. Encuesta a mitad de cuatrimestre iteración 1. Fuente: Elaboración propia

Pregunta	Opción de respuesta
La modalidad de trabajo te parece adecuada con tu expectativa de aprendizaje	Si – 75% No – 25%
¿Conoces los objetivos a los que tienes que llegar o tienes claro el conocimiento y lo que debes saber hacer al final del curso?	Si – 66,7% No – 33,3%
¿Te sientes perdido usando la plataforma, no sabes qué hacer y para qué?	Si – 16,7% No – 83,3%
De los recursos contenidos en la plataforma ¿Cuál usaste?	Audio – 45,8% Video – 54,2% Lectura – 70,8% Actividades – 83,3%
¿Qué acción sugieres para mejorar el aprendizaje o estás de acuerdo con tu entorno de aprendizaje?	Respecto a la modalidad de aprendizaje: Estar de acuerdo – 54,2% No está de acuerdo – 16,6% Sin comentario – 29,2% Respecto a sugerencias: Modalidad presencial – 37,5% Más ejercicios – 12,5% Contenido correcto – 29,2% Sin comentario – 20,8%

Intervención con pensamiento computacional, educación personalizada y *B-Learning*

Los resultados por escenario de aprendizaje fueron los siguientes:

Escenario 1. Del grupo 1^oD los dos estudiantes con calificación Autónomo fueron coherentes al usar un entorno de aprendizaje en línea, recurrieron a las asesorías y presentaron las evidencias de evaluación cuando lo determinaron, obteniendo una calificación aprobatoria. El tercer estudiante obtuvo Destacado debido a que la entrega de evidencias fue del 90% de lo acordado, pero suficiente para acreditar el curso.

Escenario 5. Los 8 estudiantes de ambos grupos optaron por el modelo semi-presencial al mes de haber iniciado el cuatrimestre. Se determinó un día a la semana para práctica en laboratorio, pero a pesar de estar el material de estudio en línea no lo usaban, pues no se observó avance en laboratorio, se tenía que iniciar con explicación teórica y entonces ejercitar la programación. Tres estudiantes obtuvieron Destacado, dos Satisfactorio, y tres acreditaron en periodo extraordinario.

Escenario 7. Los 15 estudiantes de los dos grupos eligieron el modelo semi-presencial, aunque se determinó un día de clase teórica presencial y otra de práctica. La evaluación fue con base al tiempo indicado en la hoja de asignatura y los ejercicios determinados por la academia. Un estudiante obtuvo Satisfactorio, 2 Destacado, 5 Autónomo y 7 acreditaron en extraordinario.

Escenario 9. Los estudiantes que estudiaban el material en línea fueron los que mejor utilizaron el tiempo de laboratorio, 13 acreditaron el curso en periodo extraordinario y 9 fueron baja.

Escenario 10. Sin problema trabajaron las dos primeras unidades temáticas en línea, para la última unidad, a pesar de pasar a un modelo semi-presencial no tuvieron el mismo desempeño de autonomía. El diseño de algoritmos y la codificación fue la actividad principal cuando solicitaron horas de laboratorio que no estaban consideradas en el escenario. De los 17 estudiantes, 14 acreditaron en extraordinario y 3 desertaron.

Al final del cuatrimestre el resultado de acreditados en los periodos ordinario y extraordinario, así como el número de estudiantes que no acreditaron o se dieron de baja están concentrados en la Tabla 23. La calificación aprobatoria corresponde a los valores de Satisfactorio-8, Destacado-9, y Autónomo-10.

Tabla 23. Resultados al final del cuatrimestre iteración 1. Fuente: Elaboración propia

Grupo	Ordinario			Extraordinario	Aprobados	Baja
	SA	DE	AU			
A	10	5	6	5	26	8
B	8	4	5	8	25	8
C	2	3	5	18	28	4
D	1	3	2	19	25	8
E	0	1	2	9	12	21
F	3	1	1	8	13	20
G	0	1	6	19	26	6

Al final del cuatrimestre la respectiva encuesta voluntaria fue realizada y nuevamente la participación fue baja, solo 15 estudiantes respondieron; los valores a las preguntas están en la Tabla 24.

Iteración septiembre – diciembre de 2017

En esta ocasión se extendió la invitación de realizar la evaluación del pensamiento computacional a los estudiantes de nuevo ingreso tan pronto se conocieron los resultados de admisión a la división de TIC, a través del correo electrónico de los jóvenes aceptados, se les hizo llegar una carta de bienvenida y la dirección electrónica de la evaluación, los estudiantes del primer proceso de admisión fueron los que más participaron. Un total de 110 estudiantes de nuevo ingreso realizaron la evaluación del pensamiento computacional en los meses de julio y agosto de 2017, además de los que fueron asignados al 1°B y 1°H que no lo habían realizado anteriormente, debido a que fueron los grupos donde el autor de este trabajo fue profesor. El grupo 1°B corresponde al área de sistemas informáticos mientras que el 1°H al área de redes y telecomunicaciones, lo anterior representa una diferencia en comparación a la primera iteración, aunque el curso fue el mismo para ambas especialidades los estudiantes tienen la idea de recibir mejor contenido si pertenecen al área de sistemas informáticos,

Intervención con pensamiento computacional, educación personalizada y *B-Learning*

por el otro lado, los estudiantes de redes y telecomunicaciones tienen la idea de que no deben recibir un aprendizaje profundo de programación porque no lo usarán en su ejercicio profesional; para ambos casos son erróneas las ideas. Cada uno de los grupos tuvo a 35 estudiantes matriculados.

Tabla 24. Encuesta al final del cuatrimestre iteración 1. Fuente: Elaboración propia

Pregunta	Opción de respuesta
Selecciona los conceptos con los que estás familiarizado	
Tipos de datos	73,3%
Operadores aritméticos	60,0%
Creación de identificadores para variables	66,7%
Operadores lógicos	73,3%
Operadores relacionales	66,7%
Jerarquía de operadores	73,3%
Resolver expresiones aritméticas, lógicas, y relacionales	66,7%
Uso de una variable contador y acumulador	73,3%
Estructura de selección	26,7%
Estructura de repetición	73,3%
Definición y creación de un algoritmo	66,7%
¿La modalidad de aprendizaje fue adecuada para adquirir las competencias del curso?	Si – 86,7% No – 13,3%
¿La evaluación de tus habilidades al inicio del cuatrimestre fue una actividad acertada para determinar el mejor entorno de aprendizaje?	Si – 73,3% No – 26,7%
¿Qué recomendación tienes para las futuras generaciones acerca de la forma de aprender el contenido del curso?	Estudiar – 46,6% Presencial – 20% Ninguna – 20% Lo mismo para todos – 6,7% Me gustó y motivó a investigar y aprender más por mi cuenta – 6,7%

De los 110 estudiantes que contestaron la evaluación, se presentan los siguientes resultados. Un 38,2% obtuvo correcto el reactivo de evaluación de la habilidad Descomposición. El resto de los porcentajes está indicado en la Figura 20. Se obtuvo un porcentaje mayor al de la primera iteración, pero siguió la diversidad de respuestas por parte de los estudiantes al igual que en el resto de los reactivos. Para la evaluación de la habilidad de Abstracción, solo un 45,5% contestó correctamente, un porcentaje menor al de la iteración 1 (Figura 21).

Ejercicio 1

¿Cuál de los anteriores móviles puede ser construido usando éstas instrucciones? $(-3 \ (-1 \ 4) \ (2 \ (-1 \ 1) \ (1 \ 1))) \ (2 \ (-1 \ 6) \ (2 \ 3))$

110 respuestas

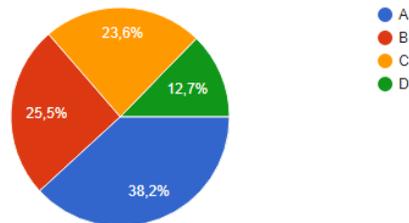


Figura 20. Porcentajes reactivo móvil iteración 2. Fuente: Elaboración propia

Ejercicio 2

Si Tomás recoge todas las 10 manzanas, ¿Cuál manzana recoge al final? A, B, C, D, E, F, G, H, I, o J.

110 respuestas

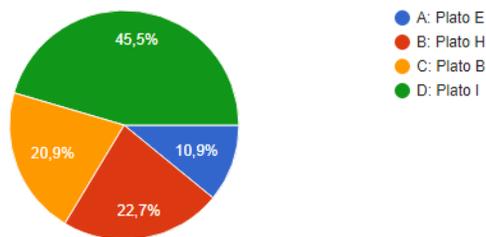


Figura 21. Porcentajes reactivo canguro iteración 2. Fuente: Elaboración propia

El ejercicio de espías que evaluó la habilidad de Generalización tuvo un 29,1% de acierto, la Figura 22 muestra los porcentajes de las otras opciones que eligieron los jóvenes; para el ejercicio que evaluó el Diseño algorítmico un 68,2% contestó correctamente, más de la mitad como lo muestra la Figura 23. Finalmente, para el reactivo de nombre Salto de charcos que evaluó la habilidad de Evaluación, un 58,2% respondió adecuadamente como se ilustra en la Figura 24.

Ejercicio 3

¿Cuál es el número mínimo de reuniones necesitadas por los cinco espías restantes para intercambiar toda la información?

110 respuestas

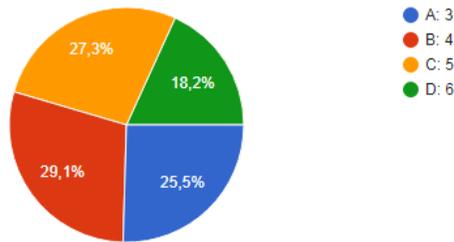


Figura 22. Porcentajes reactivo espías iteración 2. Fuente: Elaboración propia

Ejercicio 4

¿En qué orden se encontrarán los castores después de que hayan pasado el tercer hoyo?

110 respuestas

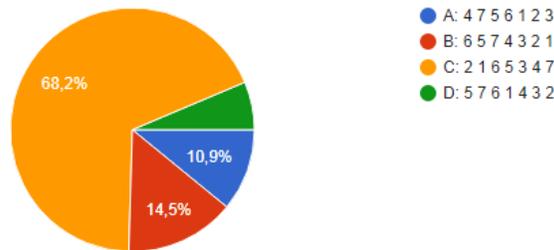


Figura 23. Porcentajes reactivo castores iteración 2. Fuente: Elaboración propia

Ejercicio 5

¿Cuál es el orden (de arriba hacia abajo) en el cual los niños terminarán a la derecha?

110 respuestas

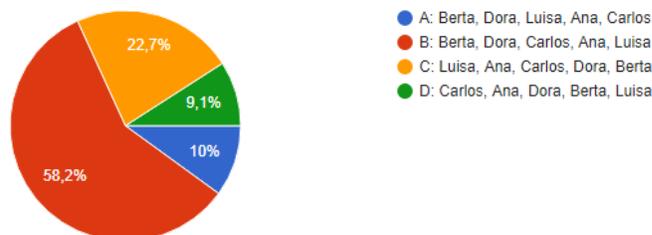


Figura 24. Porcentajes reactivo salto de charcos iteración 2. Fuente: Elaboración propia

La Tabla 25 muestra la cantidad de estudiantes por escenario en cada grupo experimental, además de observaciones iniciales para la iteración 2, hay que indicar que algunos estudiantes no se presentaron desde el inicio del curso, motivo por el cual de los 35 matriculados por grupo, 5 del 1°B y 2 del 1°H no están incluidos.

Tabla 25. Número de estudiantes por escenario iteración 2. Fuente: Elaboración propia

Grupo	Escenario	Estudiantes	Observaciones
1°B 30	1	4	Más estudiantes con 5 reactivos correctos. Viernes 13 de octubre revisión de productos de evaluación.
	3	1	El estudiante elige el escenario 10 debido a la cantidad de compañeros que forman parte del mismo.
	5	2	2 de octubre se revisan productos de evaluación
	7	6	Se usan dos horas a la semana de clase para práctica en laboratorio
	8	2	Los estudiantes comentan haber elegido correctamente la carrera y se integran al escenario 10
	9	3	Se usan dos horas a la semana para clase práctica en laboratorio.
	10	12	La mayor cantidad de estudiantes, desde el 9 de octubre comienza el modelo semi-presencial.
1°H 33	5	1	Se integra al grupo del escenario 7
	6	1	Se integra al grupo del escenario 7
	7	14	Eligen un modelo semi-presencial pero dan más peso a lo presencial solicitando 4 horas clase frente a grupo
	9	12	Trabajan el modelo semi-presencial.
	10	2	9 de octubre para trabajo en modelo semi-presencial
	8	3	Los estudiantes comentan haber elegido correctamente la carrera y se integran al escenario 7

La encuesta a mitad del cuatrimestre no se pudo realizar debido a la contingencia causada por un sismo de 7.1 registrado el 19 de septiembre, a 12 días de haber iniciado el curso. La oferta de escenarios fue afortunadamente llevada a cabo, las revisiones de productos planeados también, pero se priorizaron las actividades presenciales una vez reanudadas las clases 20 días después del sismo. La contingencia causó que el servidor de la plataforma *Moodle* se apagara y no fue reanudado el servicio hasta que personal de protección civil permitió el acceso del administrador. La mayoría de los estudiantes pudo bajar contenido del curso y las listas de cotejo de los primeros tres productos, lo

Intervención con pensamiento computacional, educación personalizada y *B-Learning*

que afectó el aprendizaje solo de los estudiantes en la modalidad presencial, aunque se mantuvo comunicación a través de la red social *Facebook*.

Los resultados evaluativos al finalizar el cuatrimestre están indicados en la Tabla 26 por grupo, así como los datos históricos.

Tabla 26. Resultados evaluativos en iteración 2 y comparación con los datos históricos. Fuente: Elaboración propia

	1° B	1° H	2009-2016 Mediana
Calificación	Cantidad		
Autónomo	8	5	
Destacado	6	2	
Satisfactorio	5	10	
Extraordinario	4	8	
No presente	12	10	
Porcentaje de acreditación	65,71%	71,43%	70,67%
Calificación promedio	5.54	4.94	6,68
Porcentaje de deserción	34,29%	28,57%	31,13%

La evaluación final solo fue contestada por 10 estudiantes del 1°B, la información se encuentra concentrada en la Tabla 27, hay que destacar que entre los encuestados 1 obtuvo los 5 reactivos correctos, su autonomía fue mayor, y el resto siguió el modelo semi-presencial.

4.2.5 Fase analítica

Acerca de ofrecer educación personalizada a través de escenarios de aprendizaje en un entorno *b-Learning* determinados por la evaluación del pensamiento computacional:

Para la iteración 1, a pesar de que existieron mismos escenarios en los dos grupos (excepto el escenario 1 para el 1°D) la variación en las cantidades mostró la individualidad de los estudiantes y la justificación de personalizar su entorno de aprendizaje, es decir, reflejó la diversidad de conocimiento y habilidades con el que ingresaron a la universidad. Para la iteración 2, nuevamente existió una diversidad de escenarios que se ofrecieron a los estudiantes después de la evaluación del

pensamiento computacional, lo que confirma por segunda vez la individualidad e importancia de personalizar su entorno de aprendizaje.

Tabla 27. Encuesta al final del cuatrimestre iteración 2. Fuente: Elaboración propia

Pregunta	Opción de respuesta
Selecciona los conceptos con los que estás familiarizado	
Tipos de datos	90,0%
Operadores aritméticos	90,0%
Creación de identificadores para variables	100%
Operadores lógicos	90,0%
Operadores relacionales	80,0%
Jerarquía de operadores	90,0%
Resolver expresiones aritméticas, lógicas, y relacionales	80,0%
Uso de una variable contador y acumulador	90,0%
Estructura de selección	60,0%
Estructura de repetición	100%
Definición y creación de un algoritmo	80,0%
¿La modalidad de aprendizaje fue adecuada para adquirir las competencias del curso?	Si – 100% No – 0%
¿La evaluación de tus habilidades al inicio del cuatrimestre fue una actividad acertada para determinar el mejor entorno de aprendizaje?	Si – 100% No – 0%
¿Qué recomendación tienes para las futuras generaciones acerca de la forma de aprender el contenido del curso?	Aceptación total, motivó a investigar y aprender más de forma autónoma.

Acerca de mejorar los indicadores de acreditación, deserción escolar y calificación del curso Metodología de la programación:

Para el cuatrimestre septiembre – diciembre de 2016 iteración 1, los valores obtenidos de los porcentajes de deserción, acreditación y calificación promedio, representan una mejora en comparación con los datos históricos. El porcentaje de deserción fue del 24,24 para el 1°C y del 27,27 para el 1°D, que es menor en 6.89 y 3.86 por ciento respecto a la mediana de los últimos 8 años (31,13); el porcentaje de acreditación del curso Metodología de la programación fue del 81.82 para el 1°C y del 72.73 para el 1°D que es mayor en 11.14 y 2.05 por ciento respecto a la mediana de los últimos 8 años (70,67); el promedio de calificación del mismo curso en los estudiantes del 1°C fue

Intervención con pensamiento computacional, educación personalizada y *B-Learning*

del 7,12 y del 1°D fue del 6,81, mayor en 0,43 y 0,13 respecto a la mediana de los últimos 8 años (6,68). La Tabla 28 concentra la información comparativa comentada anteriormente.

Tabla 28. Información comparativa de los últimos 8 años primer cuatrimestre con iteración 1.

Fuente: Elaboración propia

Año	Porcentajes Primer cuatrimestre		Promedio de calificación
	Deserción	Acreditación	
2009	24,32	77,31	6,78
2010	27,81	74,20	6,67
2011	31,13	70,50	6,64
2012	31,14	71,50	6,69
2013	35,20	65,23	6,18
2014	42,24	58,81	6,31
2015	30,30	70,85	6,81
2016	34,00	68,25	6,75
Mediana	31,13	70,67	6,68
1°C 2016	24,24	81,82	7,12
1°D 2016	27,27	72,73	6,81

Por otro lado, el número de estudiantes aprobados en los grupos experimentales es similar a los grupos de control, así como el número de estudiantes que no acreditaron o se dieron de baja, excepto de forma muy drástica con el 1° E y 1° F. El dato de observación está en el número de estudiantes que acreditaron en el periodo extraordinario con el profesor del curso que es mayor en los grupos experimentales, para el caso del 1° G los aprobados en extraordinario fueron promovido en un periodo posterior al cuatrimestre por la directora de la carrera. La acción anterior no aporta una comparación correcta con la investigación entre ese grupo de control y los experimentales.

Específicamente para el cuatrimestre de la iteración 1, el número de estudiantes acreditados fue poco más del 80% (53 de 65 estudiantes), un 18,5% de deserción (12 estudiantes) y un promedio del 65,45% de nivel académico de egreso, porcentajes similares a grupos que no trabajaron la propuesta en el mismo periodo y mucho mejor a dos grupos en particular. La Figura 25 ilustra el comparativo de lo mencionado.

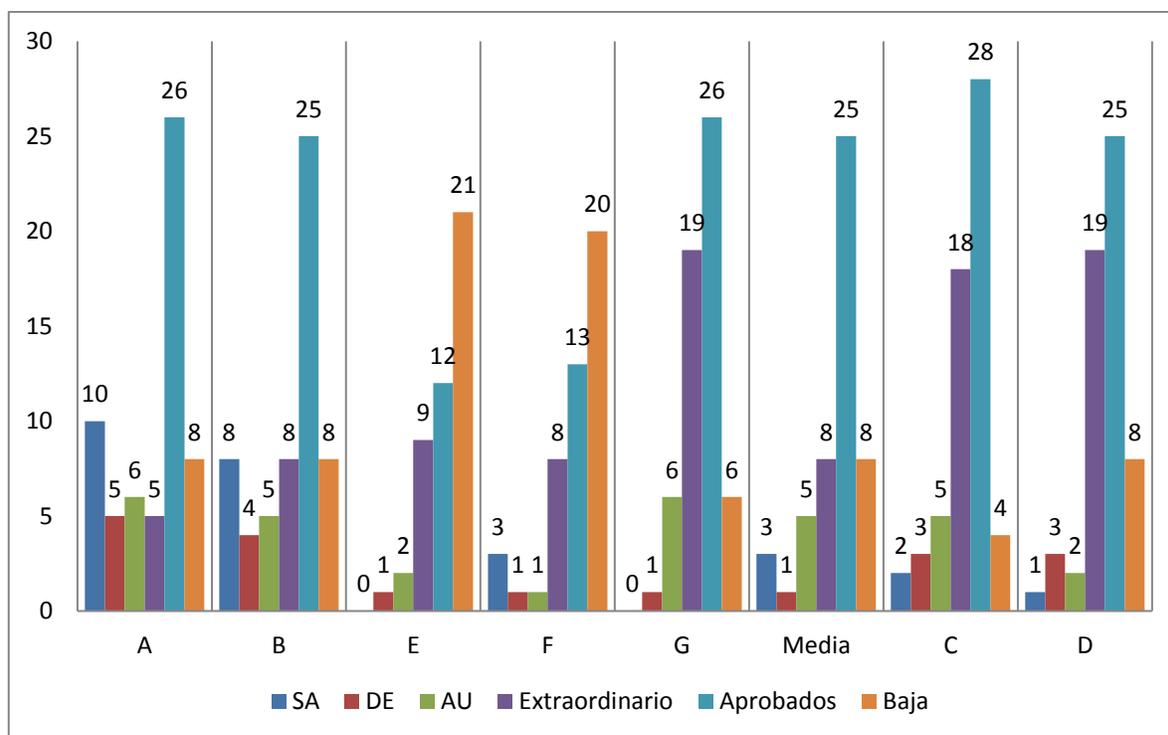


Figura 25. Resultados de la iteración septiembre – diciembre 2016. Fuente: Elaboración propia

La Figura 26 muestra los resultados de aprobación y baja escolar por escenario. Destaca la cantidad de estudiantes que aprueban en extraordinario principalmente en los escenarios 7, 9 y 10. En el escenario 7 se indicó que los estudiantes poseían las habilidades básicas y requerían de asesoría presencial regular, pero la mayoría de los estudiantes no asistió a las horas determinadas por el docente. Los escenarios 9 y 10 tienen una alta promoción de la autonomía del estudiante, lo que refuerza la falta de organización del tiempo de estudio y compromiso con su proceso educativo, solo al final del curso los estudiantes tuvieron la presión de entregar los productos de evaluación. Las bajas escolares se obtuvieron en los escenarios 9 y 10.

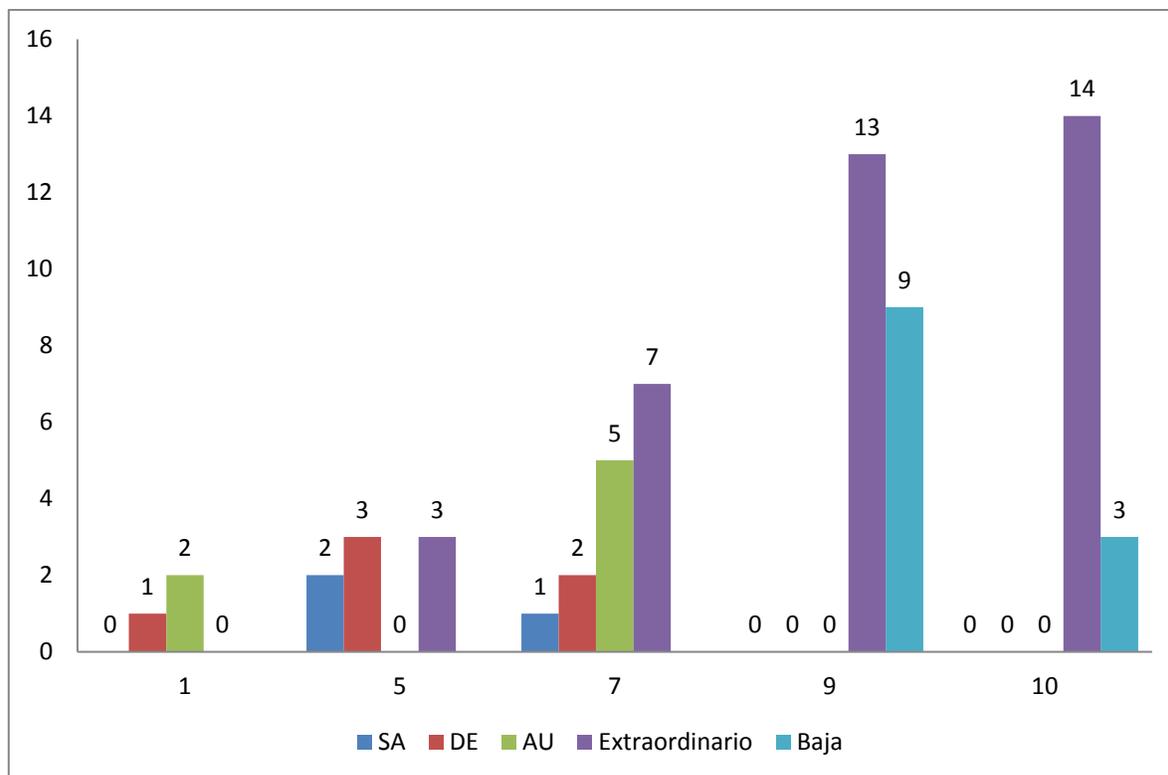


Figura 26. Resultados por escenarios en los grupos experimentales septiembre – diciembre 2016. Fuente: Elaboración propia

Para el cuatrimestre septiembre – diciembre de 2017 iteración 2, existió un número elevado de estudiantes que ya no se presentaron después del regreso de actividades tras el sismo magnitud 7.1 que padeció la comunidad académica de la Universidad Tecnológica de Puebla, así como la población completa de la ciudad, la intervención en su iteración 2 se adaptó a las circunstancias a pesar de no haber sido alterada en su contenido, lo que resultó en un ejemplo más de la adaptabilidad que deben tener las estrategias educativas. Debido a la contingencia, no fue posible tener una mejora respecto al porcentaje de deserción y acreditación del 2009 al 20016, en consecuencia, el promedio de la calificación también es menor a la mediana de los últimos 8 años. Al no existir grupos de control, se carece de un análisis comparativo, pero a pesar de esto se considera la experiencia.

Acerca de las competencias básicas de Programación que adquirieron los estudiantes:

De la iteración 1 los estudiantes desconocen estar familiarizados al menos con un concepto, lo que coincide con el promedio de calificación obtenido. El 73,3% de los estudiantes reconoce estar familiarizado con los conceptos de tipos de datos, operadores lógicos, jerarquía de operadores, uso de variable contador y acumulador, y estructura de repetición; el 66,7% con los conceptos de creación de identificadores, operadores relacionales, resolver expresiones aritméticas, lógicas y relacionales, y la definición y creación de algoritmos; el 60% con el concepto de operadores aritméticos; solo un 26,7% con el concepto de estructura de selección. Los anteriores porcentajes coinciden con la recomendación de los estudiantes para futuras intervenciones donde aconsejan estudiar más, un 46,6% lo indican, la modalidad de aprendizaje no es un problema que ellos perciban, un 86,7% la considera adecuada y solo un 6,7% recomienda que sea la misma forma de aprendizaje para todos. De la segunda iteración los resultados son mejores, dos conceptos son 100% familiarizados por los estudiantes, creación de identificadores y estructura de repetición; cinco conceptos tienen el 90% de familiaridad, tipos de datos, operadores aritméticos, operadores lógicos, jerarquía de operadores y uso de variable contador y acumulador; tres conceptos tienen el 80%, operadores relacionales, resolver expresiones aritméticas, lógicas y relacionales y definición y creación de un algoritmo; nuevamente el concepto estructura de selección tiene el porcentaje más bajo, 60% lo que indica un foco de atención para modificar el material en la plataforma y reforzar su práctica en las sesiones presenciales. La mejoría en los resultados coincide con el hecho de que el 100% de los encuestados considera adecuada la modalidad de aprendizaje para la adquisición de las competencias y no indican una recomendación negativa para las siguientes intervenciones, sus comentarios son de aceptación y motivación por investigar y aprender más de forma autónoma.

4.2.6 Fase discusión

Desde la perspectiva cuantitativa, los indicadores académicos obtenidos son mejores y la intervención que combinó el pensamiento computacional, educación personalizada y *b-Learning* representó una novedad favorable para el caso de estudio y puede servir para otros contextos similares a nivel global.

Intervención con pensamiento computacional, educación personalizada y *B-Learning*

Desde la perspectiva cualitativa, a partir de las encuestas realizadas a mitad y final de cuatrimestre en la iteración 1 se tienen las siguientes observaciones con la intención de corroborar los resultados cuantitativos. A mitad del cuatrimestre, un 75% de los encuestados estuvo de acuerdo con su modalidad de aprendizaje, un 66,7% tenía claro los objetivos de aprendizaje del curso, un 83,3% utilizó sin dificultad la plataforma *Moodle* (destaca el uso del contenido de tipo actividades, lectura, y visual) y los recursos contenidos, destacando el uso de las actividades con un 83,3%, finalmente, de las acciones sugeridas por los estudiantes, indicaron que la modalidad presencial era la mejor opción de los encuestados. Al final del cuatrimestre, la evaluación de las habilidades del pensamiento computacional al inicio del curso tuvo una aprobación final del 73,3%, lo que produjo una aceptación del entorno de aprendizaje del 86,7%, un resultado muy relevante que justifica la propuesta y su intervención en el aula; aunque la mayoría de los estudiantes acreditaron en el periodo extraordinario, los escenarios de aprendizaje tuvieron aceptación y tuvieron el impacto favorable en los indicadores académicos.

Se puede asegurar, sobre todo de la primera iteración que se obtuvieron los siguientes resultados favorables:

- Pertinente relación de las unidades temáticas del curso y las habilidades del pensamiento computacional, como también lo propuso Romero et al. (2017).
- Motivación de los estudiantes al reconocer sus habilidades del pensamiento computacional, beneficiando su confianza en la capacidad de resolución de problemas, que como en los trabajos de Yinnan y Chaosheng (2012), y Zhang et al. (2011) impacta en el aprendizaje de la programación de computadoras.
- Aceptación de elegir un entorno de aprendizaje inicial para el curso Metodología de la programación, que permitió a varios estudiantes acreditar el curso y tomar el control de su proceso de aprendizaje, en semejanza con lo reportado por Gao (2014), Chen (2017) y Ni (2017) que parten de la cultivación del pensamiento computacional. La Figura 27 muestra

que el éxito de la propuesta radica en la aceptación de la estrategia, los estudiantes tuvieron claro el objetivo del curso y el uso de la plataforma.

- Un porcentaje de deserción menor al registrado en los últimos 8 años.

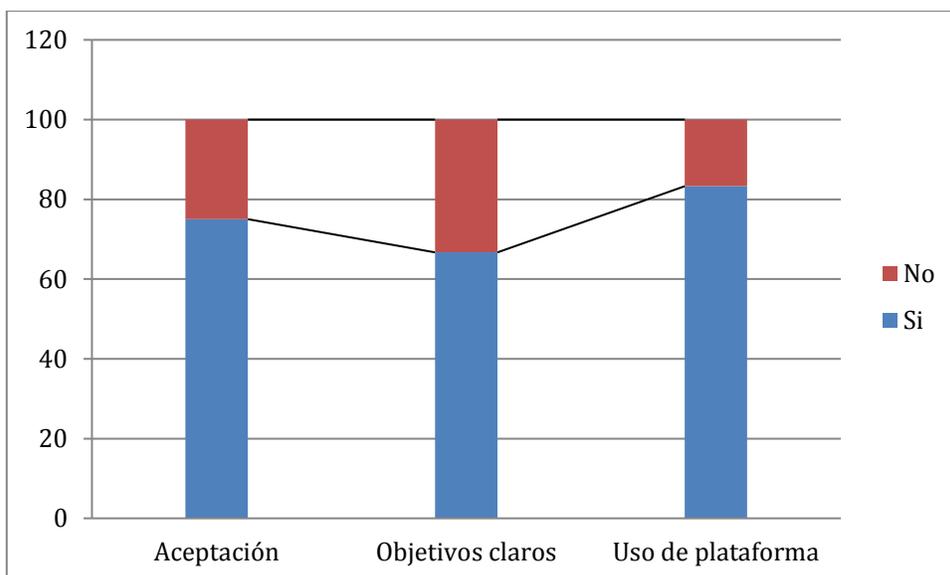


Figura 27. Porcentajes que favorecen la iteración 1 de la intervención realizada. Fuente: Elaboración propia

Los estudiantes no están preparados para un aprendizaje autónomo, pues se observaron tres conductas comunes:

- Requieren de recordarles las actividades de estudio,
- Fecha de entrega de trabajos generalmente determinadas por el profesor y
- Limitar su fuente de conocimiento a lo proporcionado por la universidad.

La falta de una disciplina de estudio hizo difícil que los estudiantes tomaran el control de su proceso educativo, así lo indica el número alto de acreditados durante el periodo extraordinario de los escenarios 9 y 10 en comparación con el número de aprobados en extraordinario en los grupos de control que tenían clase presencial; la mayoría de los estudiantes indicó estar cómodo en un ambiente

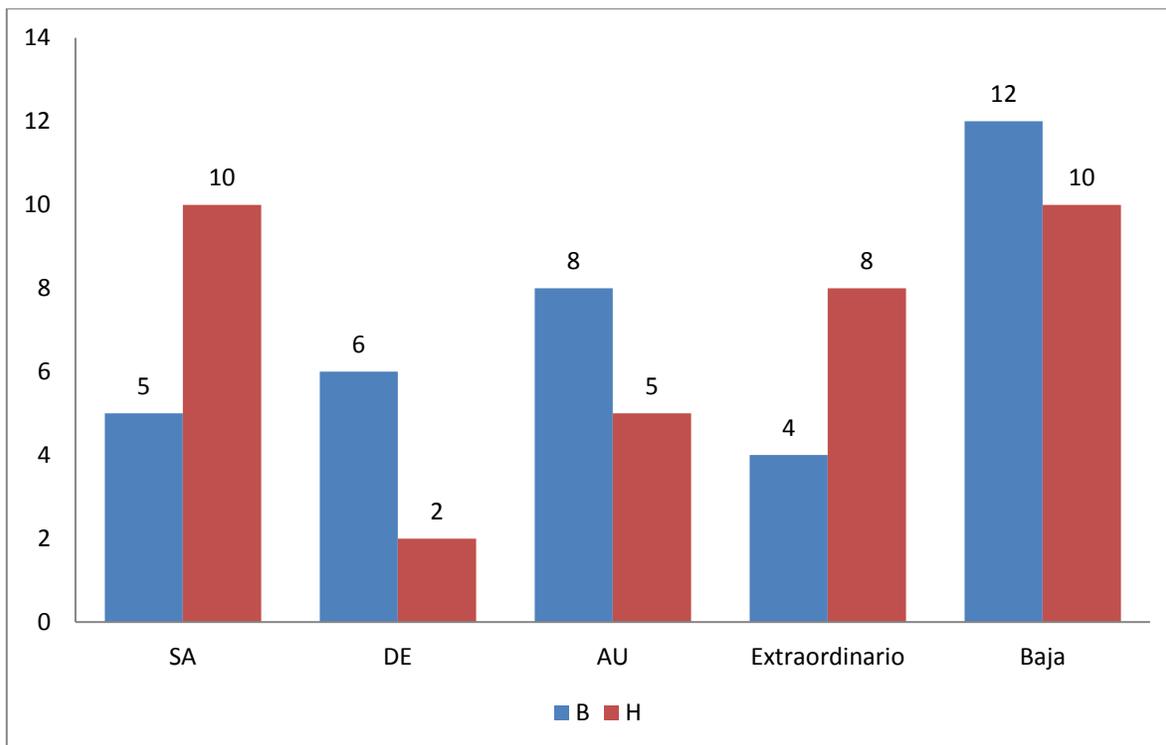
Intervención con pensamiento computacional, educación personalizada y *B-Learning*

presencial donde la responsabilidad solo sea del docente y las estrategias que realice frente a grupo. Esta actitud coincide con lo reportado por Ying et al. (2017). Existe un entusiasmo por el trabajo semi-presencial y la forma de presentar proyectos chicos de programación, en semejanza a lo propuesto por Cheng et al. (2016), pero deberá trabajarse una planeación, creación y seguimiento de actividades más puntual que permita al estudiante sentir el control externo de su proceso educativo, como lo reporta Xia (2016) o Michaelson (2015), así como agregar las recomendaciones de Compañeros et al. (2015) para el entorno presencial.

De la poca información cualitativa de la segunda iteración, se tiene que la mayoría reportó una familiaridad alta con los conceptos aprendidos en el curso, excepto con un 60,0% para el tema de estructura de selección, es decir, uso de condicionales. En las demás preguntas las respuestas son muy favorables y satisfactorias.

Al observar los resultados de los estudiantes que aprobaron el curso para los dos grupos experimentales en la Figura 28 se tienen 13 con calificación Autónomo, más que en la primera iteración, lo cual queda muy justificado al reconocer su capacidad de aprendizaje auto regulado considerando la contingencia del sismo; 15 estudiantes obtuvieron el mínimo satisfactorio y 8 Destacado, es decir, 36 estudiantes de 48 a través del escenario de aprendizaje acreditaron en periodo ordinario, y un total de 12 también, pero en periodo extraordinario. El anterior resultado es un reflejo de la importancia de buscar el balance entre la modalidad presencial y *b-Learning*, pues de no haber sido por la falla del servidor posiblemente se hubieran tenido mejores resultados en las evaluaciones finales. Los estudiantes ante la falta de clases no contaron con el entorno virtual y los beneficios que ofrece la plataforma *Moodle*. El trabajo de evaluación incrementó por el uso de las listas de cotejo, se tuvieron que realizar 240 revisiones de los productos de evaluación y promediarlas para generar la calificación final, pero bien aportó una puntualización de los elementos de acreditación, lo cual era una mejora a resolver respecto a la iteración 1. El trabajo a futuro consiste ahora en diseñar posibles acciones de automatización de la evaluación para evitar el exceso de trabajo manual.

Figura 28. Resultados grupos experimentales septiembre – diciembre 2017 iteración 2. Fuente: Elaboración propia



Por otro lado, a pesar de que dos generaciones de estudiantes han realizado los reactivos seleccionados en la intervención, su vigencia de uso se reafirma porque representan una complejidad adecuada al nivel de egreso del nivel pre universitario. Los pocos estudiantes que contestaron la encuesta final en la segunda iteración, exhiben la importancia de reconocer las habilidades del pensamiento computacional, lo cual representa novedoso en comparación con el contexto pre universitario, la otra novedad es la oferta de escenarios de aprendizaje donde se valora su capacidad de autonomía, sin demeritar la riqueza que puede ofrecer la clase presencial. La contingencia causada por el sismo confirma la acción de crear un adecuado modelo educativo híbrido para el sistema de Universidades Tecnológicas, donde el aprendizaje a distancia pueda apoyar cuando el aprendizaje presencial no sea posible y en sentido contrario, poder aprender durante un periodo a distancia y aprovechar el momento presencial cuando ya sea posible.

4.3 Estancia en INACAP – Chile

4.3.1 Contextualización del trabajo

El modelo educativo del INACAP (2015) se integra por cinco tendencias de cambio en las funciones y estructuras de la educación superior: Sociedad del conocimiento, formación de competencias, flexibilidad y articulación, cobertura y gestión de la calidad. En la Sociedad del conocimiento, el foco de la educación está puesto en la persona, reconociendo la heterogeneidad de su perfil. En la formación de competencias, las instancias de demostración de desempeños han de permitir al docente retroalimentar oportunamente a los estudiantes con el objetivo de asegurar su capacidad de actuar profesionalmente una vez egresado; desde el punto de vista de la formación, implica contar con arquitecturas progresivas de aprendizaje, diversificación de metodologías pedagógicas y la asignación protagónica al estudiante. En la flexibilidad y articulación, se han de reconocer aprendizajes previos obtenidos no solamente en procesos educativos formales, para articular aprendizaje de acuerdo con las necesidades e intereses propios del estudiante. El trabajo de investigación realizado en la estancia del 1 de marzo al 30 de junio de 2019, tuvo el objetivo de realizar el diseño y la planeación de escenarios iniciales de aprendizaje para la asignatura Fundamentos de programación, a partir de la evaluación del pensamiento computacional a los estudiantes de nuevo ingreso de las carreras Ingeniería en informática y Analista programador de la Universidad Tecnológica de Chile y del Centro de Formación Técnica respectivamente, para favorecer la motivación y autonomía de estudio a través del reconocimiento de habilidades y el uso del diseño instruccional del curso presencial. La propuesta de planeación realizada estuvo fundamentada en corresponder con tres de las tendencias de cambio comentadas previamente. Respecto a la sociedad del conocimiento, propiciar el reconocimiento de la individualidad del estudiante como persona que realizará estudios universitarios, es decir, los escenarios responden al hecho de que cada persona aprende de forma diferente. En la formación de competencias, contribuir con acciones preventivas que comunique el docente cuando se detecta carencia de habilidades específicas. Finalmente, en la flexibilidad y articulación, aportar una herramienta diagnóstica que favorezca el reconocimiento de competencias

previas para tener un articulado inicio de estudios a partir de las necesidades del estudiante. En consecuencia, contribuir con el modelo educativo del INACAP a través de replicar la intervención realizada en México, es decir, la combinación del pensamiento computacional, el fomento de la educación personalizada y el entorno de aprendizaje *b-Learning*. INACAP tiene porcentaje de retención de primer año (2016-2017) para el Centro de Formación Técnica del 70,2%, y para la Universidad Tecnológica del 67,0% (SIES, 2019).

Tomando como referencia la asignatura presencial Fundamentos de programación, primer curso en ambas mallas curriculares para el desarrollo de software (Ingeniería en informática / Analista programador), donde en la descripción se indica una “asignatura práctica orientada a desarrollar el pensamiento lógico del estudiante mediante el análisis y resolución de problemas”, se puede considerar una evaluación del pensamiento computacional para determinar las habilidades que posee un estudiante en la resolución de problemas, y obtener escenarios iniciales de aprendizaje que representen una primer guía académica para el docente ante la dificultad que puede tener el estudiante cuando se desarrolle el contenido del curso; con el objetivo de favorecer la motivación y autonomía de estudio, además de reducir la deserción escolar a través de acciones preventivas en lugar de correctivas durante el desarrollo de la asignatura.

En trabajo colaborativo con las direcciones de Investigación y Desarrollo (I+D), INACAP online y del Área de Informática y Telecomunicaciones, se logró una propuesta de investigación robusta y factible en su ejecución. Lo anterior debido a la fuerte vinculación de la intervención realizada en México con el modelo educativo del INACAP y el diseño instruccional de la asignatura.

4.3.2 Fase planeación – diseño del experimento

El curso presencial Fundamentos de Programación tiene cinco unidades de aprendizaje y 90 horas asignadas para el semestre de 18 semanas. Debido al modelo educativo de INACAP el diseño instruccional integra horas de trabajo en línea (72 presenciales y 18 en línea para el curso), el detalle

Intervención con pensamiento computacional, educación personalizada y *B-Learning*

de la distribución de horas, semanas, ponderación y número de criterios de evaluación está indicado en la Tabla 29. En cada unidad se describe el aprendizaje esperado (ver Tabla 30) y una lista de los criterios de evaluación (ver Tabla 31 a Tabla 35), para cada asignatura se realizan tres acciones evaluativas: diagnóstica, formativa y sumativa; las dos primeras corresponden a actividades sugeridas que contribuirán a mejorar los resultados de aprendizaje y permitirá conocer el avance del estudiante en distintos momentos de la asignatura, la evaluación sumativa es de carácter obligatorio y se realiza según lo estipule el curso. Cada unidad de aprendizaje aporta un porcentaje de la evaluación sumativa, la suma de las cinco en el curso genera un 80%, el 20% faltante lo especifica el docente a su criterio por medio de evaluar presentaciones, proyectos o trabajos extras.

Tabla 29. Planeación del curso Fundamentos de programación en INACAP. Fuente: Elaboración propia

Unidad de aprendizaje	Horas			Semanas	Criterios de Evaluación/ Ponderación
	Presenciales	En línea	Totales		
1. Fundamentos de procesamiento de datos	12	3	15	3	5 / 10%
2. Estructuras de control en DFD	16	4	20	4	6 / 20%
3. Estructuras de control en pseudocódigo	20	5	25	5	6 / 20%
4. Estructura de arreglo	12	3	15	3	5 / 15%
5. Subrutinas	12	3	15	3	5 / 15%

Tabla 30. Aprendizajes esperados por unidad de aprendizaje. Fuente: Elaboración propia

Unidad de aprendizaje	Aprendizaje esperado
1	Resuelve problemas de procesamiento de datos, aplicando principios de almacenamiento y tablas de verdad.
2	Representa gráficamente la solución de un problema mediante diagramas de flujo de datos aplicando estructuras de control.
3	Desarrolla Algoritmos en Pseudocódigo, aplicando estructuras de control en la solución de un problema planteado.
4	Desarrolla Algoritmos en Pseudocódigo, utilizando Arreglos unidimensionales y bidimensionales en la solución de un problema planteado.
5	Desarrolla algoritmos básicos en Pseudocódigo, mediante subrutinas en la solución de un problema planteado.

Tabla 31. Criterios de evaluación y actividades, unidad de aprendizaje 1. Fuente: Elaboración propia

Criterio de evaluación	Actividad – modalidad
1.1.1. Identificando datos de entrada, proceso y salida a partir de un problema de procesamiento de datos. 1.1.2. Aplicando las etapas de la metodología de Polya en el análisis de la solución del problema planteado. 1.1.3. Considerando operaciones de entrada, procesos y salida de datos en memoria. 1.1.4. Aplicando operadores lógicos en la solución de problemas de procesamiento de datos.	- Evaluación diagnóstica – presencial - Actividad formativa 1 – en línea - Guía de ejercicios 1 / evaluación formativa – presencial - Actividad formativa 2 – en línea - Guía de ejercicios 2 / evaluación formativa – presencial - Evaluación sumativa – presencial

Tabla 32. Criterios de evaluación y actividades, unidad de aprendizaje 2. Fuente: Elaboración propia

Criterio de evaluación	Actividad – modalidad
2.1.1. Aplicando estructuras de decisión en la solución del problema. 2.1.2. Incorporando operadores lógicos en la solución del problema. 2.1.3. Utilizando estructuras de repetición en la solución del problema. 2.1.4. Considerando la validación de datos en la solución del problema. 2.1.5. Realizando la traza de la solución propuesta.	- Evaluación diagnóstica – presencial - Guía de ejercicios 3 / evaluación formativa – presencial - Foro – en línea - Guía de ejercicios 4 / evaluación formativa – presencial - Evaluación sumativa – presencial

Tabla 33. Criterios de evaluación y actividades, unidad de aprendizaje 3. Fuente: Elaboración propia

Criterio de evaluación	Actividad – modalidad
3.1.1. Aplicando estructuras de decisión en la solución del problema. 3.1.2. Utilizando operadores lógicos en la construcción de algoritmos. 3.1.3. Incorporando estructuras de repetición en algoritmos en Pseudocódigo. 3.1.4. Considerando la validación de datos en la solución del problema. 3.1.5. Realizando la traza de la solución propuesta.	- Evaluación diagnóstica – presencial - Actividad formativa – en línea - Guía de ejercicios 5 / evaluación formativa – presencial - Foro – en línea - Evaluación sumativa – presencial

Tabla 34. Criterios de evaluación y actividades, unidad de aprendizaje 4. Fuente: Elaboración propia

Criterio de evaluación	Actividad – modalidad
4.1.1. Ingresando datos en arreglos unidimensionales y bidimensionales. 4.1.2. Realizando un recorrido sobre arreglos unidimensionales o bidimensionales. 4.1.3. Realizando búsquedas en arreglos unidimensionales o bidimensionales. 4.1.4. Realizando la traza de la solución propuesta.	- Evaluación diagnóstica – presencial - Actividad formativa – en línea - Guía de ejercicios 6 / evaluación formativa – presencial - Foro – en línea - Evaluación sumativa – presencial

Tabla 35. Criterios de evaluación y actividades, unidad de aprendizaje 5. Fuente: Elaboración propia

Criterio de evaluación	Actividad – modalidad
5.1.1. Incorporando paso de parámetros a la subrutina. 5.1.2. Incorporando el retorno de datos al programa principal. 5.1.3. Realizando llamadas a las subrutinas creadas. 5.1.4. Realizando la traza de la solución propuesta.	- Evaluación diagnóstica – presencial - Guía de ejercicios 7 / evaluación formativa – presencial - Foro – en línea - Evaluación sumativa – presencial

Inicialmente, en similitud a la intervención realizada en México, se establece una relación de conocimientos de la asignatura Fundamentos de programación con las habilidades del pensamiento computacional como se indica en la Tabla 36.

En cada relación se indica la justificación y los criterios de evaluación con los que se determina un impacto, además del nombre del reactivo seleccionado, los cuales también corresponden a los usados previamente (Rojas-López & García-Peñalvo, 2018).

Tabla 36. Relación de habilidades del pensamiento computacional con unidades de aprendizaje.

Fuente: Elaboración propia

Unidad de aprendizaje	Habilidad Pensamiento Computacional / reactivo	Justificación
1. Fundamentos de procesamiento de datos	Abstracción / Canguro	La abstracción ayuda a determinar los datos que contribuyen a desarrollar la resolución de alguna problemática como lo indica el aprendizaje esperado de la unidad. Impactará en los criterios de evaluación: 1.1.1, 1.1.3, 1.1.4
2. Estructuras de control en DFD	Evaluación / Salto de Charcos	La evaluación permite reconocer y determinar el alcance de ejecución de los diagramas de flujo de datos en la resolución de un problema que es parte del aprendizaje esperado de la unidad. Impactará en el criterio de evaluación: 2.1.5
3. Estructuras de control en pseudocódigo	Diseño algorítmico / Castores	El diseño algorítmico permite el desarrollo de algoritmos en pseudocódigo (o lenguaje de programación) que es objetivo del aprendizaje esperado de la unidad. Impactará en los criterios de evaluación: 3.1.1, 3.1.2, 3.1.3, 3.1.4
4. Estructura de arreglos	Generalización / Espías	El manejo de arreglos representa el primer incremento en la complejidad del diseño algorítmico, debido a que el estudiante ha visualizado la usabilidad de la estructura para resolver problemas donde previamente había creado una solución, pero ahora ha incorporado la comprensión de características similares, así puede generalizar una solución de forma óptima. Impactará en los criterios de evaluación: 4.1.2, 4.1.3
5. Subrutinas	Descomposición / Móviles	La creación de subrutinas o funciones representa la habilidad para fragmentar (descomponer) un problema en bloques funcionales y de menor tamaño. Impactará en los criterios de evaluación: 5.1.1, 5.1.2, 5.1.3

4.3.2.1 Escenarios de aprendizaje

La interpretación de los resultados que se obtienen con la evaluación del pensamiento computacional, está determinada por los siguientes 8 escenarios que el estudiante puede enfrentar en la modalidad presencial del curso. El número de escenarios corresponde a los casos que representan desde la detección de posibles talentos (cinco reactivos correctos) hasta carencia de habilidades para la resolución de problemas (cinco reactivos incorrectos). Se consideran escenarios donde se establecen

Intervención con pensamiento computacional, educación personalizada y *B-Learning*

posibles carencias desde el inicio, a mitad o final del curso respecto a las unidades de aprendizaje con los reactivos correspondientes. Por lo anterior, los escenarios propuestos no son todas las combinaciones de respuestas correctas de los cinco reactivos, solo de aquellos según la relación con la unidad de aprendizaje. Las unidades 1 y 2 tienen un nivel básico y de gran importancia para el curso por el contenido formativo y de conceptos. La unidad 3 representa un nivel medio de aprendizaje al repasar el concepto de estructuras de control en pseudocódigo. Finalmente, las unidades 3 y 4 se asocian con un nivel alto de aprendizaje dentro del curso. La Tabla 37 muestra la determinación de los escenarios a partir de lo comentado previamente.

El primer escenario corresponde a los estudiantes que exhiben habilidades en la resolución de problemas, por lo que se considera que no tendrán problemáticas con los contenidos de aprendizaje del curso y pueden tener un alto grado de autonomía. El segundo escenario determina el caso contrario del primero. A partir del tercero hasta el octavo se toman en cuenta los reactivos correctos según la unidad de aprendizaje con la que fue relacionada.

Tabla 37. Determinación de los escenarios de aprendizaje en INACAP. Fuente: Elaboración propia

Nivel	Habilidad	Unidad	Reactivo 1 – correcto, 0 – incorrecto	Escenarios							
				1	2	3	4	5	6	7	8
Básico	Abstracción	1	Canguro	1	0	0	1	0	1	0	1
	Evaluación	2	Salto de charcos	1	0	0	1	0	1	0	1
Medio	Diseño algorítmico	3	Castores	1	0	1	0	1	0	0	1
Alto	Generalización	4	Espías	1	0	1	0	0	1	1	0
	Descomposición	5	Móviles	1	0	1	0	0	1	1	0

Escenario 1. 5 respuestas correctas

El estudiante no tendrá ninguna problemática para acreditar siguiendo el diseño instruccional, realizando las actividades presenciales y las establecidas para la asignatura en línea a través de la plataforma, es decir, uso de las actividades formativas, guía de aprendizaje y ejercicios, así como las evaluaciones formativas y sumativas de cada unidad de aprendizaje. Un escenario alternativo para el

estudiante, si le fuera posible, podría ser avanzar el desarrollo de la asignatura a su propio ritmo y solo recibir la guía del docente cuando la solicite. Finalmente, también puede ser un buen candidato para tomar el curso en línea.

Escenario 2. Incorrecto en todos los reactivos

Existe la posibilidad de que el estudiante no tenga habilidades para el estudio de la asignatura y en la unidad de aprendizaje tres sea difícil la comprensión de las estructuras de control y diseño algorítmico al momento de crear algoritmos en pseudocódigo, por lo que se recomienda en reunión presencial docente-directivo-estudiante valorar el perfil vocacional de este último. Es muy seguro que un curso presencial sea la mejor opción para el estudiante.

Escenario 3. Incorrectos los ejercicios de Canguro y Salto de charcos.

El estudiante requiere observación y trabajo presencial del docente para que desde las unidades iniciales adquiera aprendizaje significativo. El estudiante podría acreditar el curso presencial sin inconveniente teniendo en cuenta reforzar en su momento (en la semana 4 del curso), de forma presencial o a través de la plataforma la segunda unidad de aprendizaje, Estructuras de control en DFD, debido a que sería su primer contacto con las estructuras de control (secuenciales, decisión y repetición) para la definición de algoritmos, codificando ejercicios extras a la guía correspondiente que resuelven un problema usando tales estructuras. En caso contrario, posiblemente tenga dificultad desde la unidad de aprendizaje 3 y se dificulte su acreditación.

Escenario 4. Correctos los ejercicios de Canguro y Salto de charcos

El estudiante exhibe habilidades básicas para la actividad creativa del diseño de algoritmos. Posiblemente no tendrá ninguna problemática para acreditar los criterios de evaluación en las primeras tres semanas siguiendo el diseño instruccional, es decir, realizando las actividades presenciales y en línea establecidas para la asignatura en la plataforma. A partir de la cuarta semana,

Intervención con pensamiento computacional, educación personalizada y *B-Learning*

trabajar una atención especial con una práctica y retroalimentación de ejercicios extras en las unidades de aprendizaje dos y tres de forma presencial, así garantizar también la prevención de alguna problemática en las últimas dos unidades de aprendizaje.

Escenario 5. Correcto el ejercicio de Castores en movimiento

El estudiante requiere observación y trabajo presencial del docente para que desde las unidades iniciales adquiera aprendizaje significativo. Habrá que reforzar la evaluación de expresiones aritméticas, lógicas y relacionales considerando la jerarquía de operadores por medio de codificar ejercicios que usen cálculo de operaciones y condicionales en la unidad de aprendizaje dos (semana 4 del curso). Lo anterior para prevenir problemáticas al momento de abordar la unidad de aprendizaje 4 y enfrente sin conflicto el primer incremento de complejidad en el curso.

Escenario 6. Incorrecto el ejercicio de Castores en movimiento

Si le fuera posible al estudiante, podría avanzar el desarrollo de la asignatura a su propio ritmo y solo recibir la guía del docente cuando la solicite a través de la plataforma misma en las primeras dos unidades de aprendizaje, en caso contrario, seguir la planeación indicada en el curso presencial. El estudiante posiblemente tenga problema con la unidad de aprendizaje 3, Estructuras de control con pseudocódigo, por ello sería conveniente reforzar el diseño algorítmico con ejercicios extras a la guía correspondiente a través de la plataforma o de forma presencial cuando se llegue a esta unidad de aprendizaje (en la semana 8 del curso), así garantizar también la prevención de alguna problemática en las últimas dos unidades de aprendizaje.

Escenario 7. Correctos los ejercicios de Espías y Móviles

El estudiante requiere observación y trabajo presencial del docente para que desde las unidades iniciales adquiera aprendizaje significativo, además de atención para que el material del curso y las actividades se realicen puntualmente con base al tiempo determinado en el diseño instruccional,

también puede requerir sesiones de asesoría con el profesor, así que deberán ser alentadas para que la confianza del estudiante se incremente respecto a la capacidad de crear programas, y planeadas para que no representen un trabajo extra del docente fuera del horario presencial. En caso contrario, posiblemente tenga dificultad desde la unidad de aprendizaje 3 y se dificulte su acreditación.

Escenario 8. Incorrectos los ejercicios de Espías y Móviles

El estudiante tiene habilidades básicas, pero necesarias para el aprendizaje particularmente de programación de computadoras, por lo que el diseño instruccional creado es favorable, pero con una atención de asesoría presencial regular en las prácticas de laboratorio que diseñe el docente.

Las circunstancias ideales para la intervención del experimento, es aquel donde todos los estudiantes que inician el curso realicen la evaluación del pensamiento computacional en línea, como una actividad durante la primera semana. Posteriormente, determinar los escenarios iniciales que posiblemente enfrentarán los docentes-estudiantes durante el curso a partir de los reactivos correctos que obtuvo cada uno de los jóvenes. Establecer los grupos de control y experimental. El o los grupos experimentales han de realizar las acciones o condiciones preventivas que consideran las recomendaciones de cada escenario para su sección respectiva del profesor. Durante el curso registrar y medir la correspondencia del escenario con los resultados de los criterios de evaluación de los estudiantes en su respectivo momento, es decir, verificar que en los grupos de control se ha de observar un porcentaje de deserción y acreditación en correspondencia con los datos históricos del área Informática y Telecomunicaciones de los últimos 3 a 5 años; para los grupos experimentales, se espera favorecer la motivación y autonomía de estudio para acreditar el curso y en consecuencia aumentar el porcentaje de retención escolar.

4.3.2.2 Instrumentos

Es recomendable realizar dos encuestas para la obtención de datos de la misma forma en que se realizó en México, las preguntas modificadas se indican en la Tabla 38 y Tabla 39, la primera para ser

Intervención con pensamiento computacional, educación personalizada y *B-Learning*

contestada en la semana 9 del curso y la segunda al final del semestre por los estudiantes del o los grupos experimentales.

Tabla 38. Encuesta a mitad del semestre para INACAP. Fuente: Elaboración propia

Pregunta	Opción de respuesta
La modalidad de trabajo te parece adecuada con tu expectativa de aprendizaje	Si No
¿Conoces los objetivos a los que tienes que llegar o tienes claro el conocimiento y lo que debes saber hacer al final del curso?	Si No
¿Te sientes perdido usando la plataforma, no sabes qué hacer y para qué?	Si No
De las actividades contenidas en la plataforma ¿Cuál te aportó más?	Selección múltiple - Evaluación diagnóstica - Actividad formativa - Guía de ejercicios / evaluación formativa - Foro - Evaluación sumativa
¿Qué acción sugieres para mejorar el aprendizaje o estás de acuerdo con tu entorno de aprendizaje?	Abierta

Tabla 39. Encuesta al final del semestre para INACAP. Fuente: Elaboración propia

Pregunta	Opción de respuesta
Selecciona los criterios de aprendizaje con los que estás familiarizado - Selección múltiple Identificando datos de entrada, proceso y salida a partir de un problema de procesamiento de datos. Aplicando operadores lógicos en la solución de problemas de procesamiento de datos. Aplicando estructuras de decisión en la solución del problema. Utilizando estructuras de repetición en la solución del problema. Considerando la validación de datos en la solución del problema. Realizando la traza de la solución propuesta. Utilizando operadores lógicos en la construcción de algoritmos. Incorporando estructuras de repetición en algoritmos en Pseudocódigo. Ingresando datos en arreglos unidimensionales y bidimensionales. Realizando un recorrido sobre arreglos unidimensionales o bidimensionales. Realizando búsquedas en arreglos unidimensionales o bidimensionales. Incorporando paso de parámetros a la subrutina. Incorporando el retorno de datos al programa principal. Realizando llamadas a las subrutinas creadas.	
¿La modalidad de aprendizaje fue adecuada para adquirir las competencias del curso?	Si No
¿La evaluación de tus habilidades al inicio del cuatrimestre fue una actividad acertada para determinar el mejor entorno inicial de aprendizaje?	Si No
¿Qué recomendación tienes para las futuras generaciones acerca de la forma de aprender el contenido del curso?	Abierta

4.4 Conclusiones

La intervención presentada en este capítulo, con sus dos iteraciones y diseño experimental para INACAP – Chile, fundamentada en la evaluación del pensamiento computacional a los estudiantes de nuevo ingreso, particularmente las habilidades de generalización, descomposición, abstracción, diseño algorítmico y evaluación, y con base al resultado de la evaluación del pensamiento computacional ofertar escenarios de aprendizaje personalizados para el curso Metodología de la programación en México y Fundamentos de programación en Chile, así como el uso de la plataforma *Moodle* como herramienta para un entorno *b-Learning*, genera los siguientes resultados concretos a la investigación de la tesis doctoral.

1. Impacta directamente en el objetivo general. Se diseñaron escenarios de aprendizaje personalizados para la asignatura Metodología de la programación / Fundamentos de programación dirigido a los estudiantes de nuevo ingreso, estableciendo una relación entre las habilidades evaluadas del pensamiento computacional y los temas del curso, usando un entorno *b-Learning*. Los escenarios propuestos representaron la acción de trabajo con el estudiante ante la respuesta incorrecta de una habilidad específica, por ejemplo, si contestó erróneamente el reactivo que evalúa la habilidad de Descomposición, que corresponde en parte a la unidad temática 1 Conceptos y al escenario 9, el entorno propone al estudiante trabajar en los temas respectivos y favorecer el aprendizaje autónomo. La iteración 1 de la intervención permitió obtener un porcentaje de deserción menor a los registrados en los últimos 8 años al finalizar el primer cuatrimestre, así como un porcentaje de acreditación mayor en el mismo periodo. En consecuencia, se redujo el abandono de estudios a partir de incrementar el porcentaje de acreditación en el curso. Respecto al promedio de la calificación, no se registró un incremento significativo, pero es una evidencia de que se ha mantenido la calidad evaluativa.
2. Respecto a las competencias básicas de Programación que adquirieron los estudiantes, de la iteración 1 el 73,3% de los estudiantes reconoce estar familiarizado con los conceptos de

tipos de datos, operadores lógicos, jerarquía de operadores, uso de variable contador y acumulador, y estructura de repetición; el 66,7% con los conceptos de creación de identificadores, operadores relacionales, resolver expresiones aritméticas, lógicas y relacionales, y la definición y creación de algoritmos; el 60% con el concepto de operadores aritméticos; solo un 26,7% con el concepto de estructura de selección. De la segunda iteración los resultados son mejores, dos conceptos son 100% familiarizados por los estudiantes, creación de identificadores y estructura de repetición; cinco conceptos tienen el 90% de familiaridad, tipos de datos, operadores aritméticos, operadores lógicos, jerarquía de operadores y uso de variable contador y acumulador; tres conceptos tienen el 80%, operadores relacionales, resolver expresiones aritméticas, lógicas y relacionales y definición y creación de un algoritmo; el concepto estructura de selección tiene el porcentaje más bajo, 60%.

3. Desde la primera iteración se observaron dos resultados positivos. En primer lugar, existen estudiantes que sienten motivación por que sean reconocidas sus habilidades y en segundo lugar, la evaluación permite determinar un entorno de aprendizaje inicial. El trabajo a futuro incluye mejorar la empatía por la propuesta de trabajo para aumentar la participación en las encuestas a mitad y final del cuatrimestre y así tener mejores niveles de evaluación cualitativa del trabajo educativo propuesto.
4. De la estancia en INACAP – Chile. Los ocho escenarios de aprendizaje para los estudiantes de nuevo ingreso del curso Fundamentos de programación están propuestos para beneficiar el porcentaje de retención del INACAP. Su diseño está basado a partir del estudio de los contenidos de las unidades de aprendizaje y el material disponible en la plataforma, es decir, considerando los criterios de evaluación y el diseño instruccional del curso.
5. En INACAP – Chile se realizó una planificación pertinente y con un diseño experimental robusto para su posible ejecución si los directivos lo consideran para el próximo semestre de nuevo ingreso. Por lo anterior, se tiene un resultado muy favorable para la investigación hasta

ahora realizada en México, es decir, es posible adaptar y ajustar el uso de estrategias educativas que mejoren los indicadores de retención de Chile con los resultados y diseño experimental de investigación previamente ejecutado en México.

Para las futuras intervenciones, se proponen con certeza las siguientes acciones:

- i. Realizar la evaluación del pensamiento computacional, usando los reactivos previamente elegidos, de preferencia durante los tres días iniciales del cuatrimestre a los grupos donde se aplicará la estrategia educativa.
- ii. Ofrecer una inducción en el uso de la plataforma *Moodle*, así como garantizar su disponibilidad.
- iii. Exponer los escenarios de aprendizaje, con detalle de las actividades y recursos con los que cuenta el estudiante, a todo el grupo experimental, destacar la importancia de las listas de cotejo en beneficio de los productos evaluativos que puede entregar.
- iv. De forma individual indicar el escenario de aprendizaje que corresponde al estudiante después de haber realizado la evaluación del pensamiento computacional, lo anterior para evitar influencia grupal en la decisión personal, pero sin dejar de ser flexible, la elección final la tiene el estudiante.
- v. Es muy recomendable realizar la encuesta a mitad del cuatrimestre para conocer el resultado de la experiencia que está teniendo el estudiante con su entorno de aprendizaje, y realizar los ajustes que sean necesarios en beneficio de los resultados académicos, emocionales y sociales.
- vi. Crear nuevos contenidos y ejercicios que involucren diferentes niveles de complejidad y puedan ser ofrecidos a los estudiantes con libre elección de realizarlos con base a su desempeño en la plataforma *Moodle*.

Intervención con pensamiento computacional, educación personalizada y *B-Learning*

- vii. Realizar la encuesta final para detectar oportunidades de mejora en los contenidos de la plataforma y preferentemente reafirmar la aceptación de la estrategia educativa.

CAPÍTULO 5. INTERVENCIÓN CON GAMIFICACIÓN, EDUCACIÓN PERSONALIZADA Y B-LEARNING

5.1 Introducción

La intervención que se describe en el presente capítulo, nuevamente con dos iteraciones, enero – abril de 2017 y enero – abril de 2018, se fundamenta en la gamificación como elemento de motivación en la educación personalizada, así como el uso la plataforma *Moodle* como herramienta para un entorno *b-Learning* en el curso de Programación, con el objetivo de mejorar los indicadores de estudiantes acreditados, la calificación promedio del grupo y reducir el porcentaje de deserción.

El modelo de gamificación de Werbach y Hunter está estructurado por tres elementos: dinámicas, mecánicas y componentes, los cuales pueden combinarse según los objetivos a lograr en una acción educativa. Las dinámicas son los contextos en los cuales se desarrolla la gamificación (por ejemplo, emotivas, narrativas, sociales, progresiones), las mecánicas son las actividades dentro de las dinámicas (por ejemplo, retos, oportunidades, recompensas) y los componentes (por ejemplo, insignias, puntos, tableros de clasificación) son los objetos, digitales o no, utilizados dentro de las mecánicas (Werbach & Hunter, 2015); así, el modelo de gamificación está en sintonía con el incremento de “la concentración, el esfuerzo y la motivación fundamentada en el reconocimiento, el logro, la competencia, la colaboración, la autoexpresión y todas las potencialidades educativas compartidas por las actividades lúdicas” (Sánchez, 2015).

Respecto a la educación personalizada en esta intervención, el estudiante tuvo la oportunidad de elegir inicialmente la modalidad de aprendizaje, es decir, presencial, semi-presencial o en línea; tuvo la oportunidad de elegir, los ritmos y tiempos de aprendizaje y evaluación, lo que significó que el estudiante se comprometió y determinó los momentos en que entregó los productos de evaluación y el tiempo que dedicó al estudio de los contenidos; finalmente tuvo la oportunidad de elegir las evidencias de evaluación.

Intervención con gamificación, educación personalizada y *B-Learning*

Los estudiantes que eligieron la modalidad de aprendizaje semi-presencial participaron en una experiencia de gamificación no digitalizada basada en dinámicas de progresión, emotivas y sociales, mecánicas de retos y recompensas usando insignias y un medallero como componentes.

En el desarrollo del capítulo se describe el trabajo realizado siguiendo la metodología de investigación mixta determinada. Se inicia con la fase planeación para establecer el uso de las estrategias educativas de gamificación y educación personalizada en un entorno *b-Learning*. Posteriormente se ejecutó la fase intervención que permitió tener los resultados obtenidos durante dos iteraciones para la fase analítica. De la primera iteración se detectaron acciones de mejora para su aplicación en la segunda. Así, al llegar a la fase discusión de la segunda iteración realizar la triangulación concurrente que permitió tener las conclusiones.

5.2 Metodología

5.2.1 Población y muestra

Tomando en consideración la opción de elección que tienen los estudiantes en el marco de la educación personalizada, el grupo experimental de la primera iteración para el curso Programación en el cuatrimestre enero – abril 2017 fue de 14 estudiantes del grupo 2°C, de un total de 25 debido a que 11 estudiantes decidieron no participar en la intervención; para la segunda iteración en el cuatrimestre enero – abril 2018, se determinó aplicar la gamificación al grupo 2°I con 29 estudiantes, que tuvo la característica de ser de la especialidad de redes y telecomunicaciones, pues tuvo mejor disposición en implementar la estrategia educativa a diferencia del grupo 2°B con 31 estudiantes de la especialidad de sistemas informáticos que eligió solo programar el producto del proyecto final, en consecuencia, se pudo tener un grupo control y experimental. La Tabla 40 concentra los datos mencionados. El grupo experimental de la segunda iteración estuvo formado por 14 mujeres y 15 hombres, sus edades oscilan entre los 18 y 20 años; había tenido en su curso de Metodología de la programación la experiencia de usar el entorno *PSeInt* para aprender a crear programas basados en la programación estructurada. En los primeros días de clases, no se aplicó la evaluación del pensamiento

computacional pues se consideró importante comenzar con los temas de la programación orientada a objetos y que conocieran el entorno de programación de *Visual Studio C#*, que fue el determinado en reunión de academia.

El grupo de control estuvo formado 7 mujeres y 24 hombres, sus edades oscilan entre los 18 y 21 años; la formación previa en el desarrollo de software fue mejor en comparación al grupo experimental, pero corresponde al diferente lenguaje de programación que usaron.

Tabla 40. Datos de grupos de control y experimentales intervención 2. Fuente: Elaboración propia

Cuatrimestre	Año	Grupos	
		Control	Experimental
Enero - Abril	2017	0	14 - 2°C
	2018	31 - 2°B	29 - 2°I

5.2.2 Fase planeación

La fase planeación durante la metodología mixta desarrollada para la segunda intervención, está formada por dos acciones principales. Primero, el trabajo realizado para ofertar educación personalizada a través de los cuatro elementos que la gestionan, pero adaptados para trabajar con la estrategia de gamificación. Segundo, la creación de la actividad de gamificación. De forma implícita se incluye el uso del entorno *b-Learning* dentro del diseño de la intervención. El trabajo realizado de esta forma, nuevamente tuvo la intención de crear el contexto adecuado para obtener información desde el enfoque CUAL y CUAL.

5.2.2.1 Educación personalizada

Con la intención de continuar reconociendo la individualidad de cada estudiante, identificada y reconocida en la intervención 1 a partir de la evaluación del pensamiento computacional, la estrategia de educación personalizada estuvo nuevamente fundamentada por 4 elementos:

i. Contenido

Los conocimientos generales del curso están indicados en la Tabla 41 y fueron los mismos para las dos iteraciones de la intervención.

Tabla 41. Conocimientos generales del curso Programación. Fuente: Elaboración propia

Unidad Temática	Conocimientos
1	Fundamentos y características de la Programación Orientada a Objetos
2	Generalidades y características del ambiente de desarrollo
3	Clases y Objetos Estructuras de control Encapsulamiento Herencia Polimorfismo
4	Arreglos
5	Manejo de excepciones

ii. Formas de trabajo

En la modalidad presencial las clases se llevaron a cabo de forma tradicional, actividades prácticas y teóricas como lo indica el currículo de la asignatura en el horario asignado. En la modalidad semi-presencial se aprovechó una vez más el uso de la plataforma *Moodle* para trabajar en un entorno *b-Learning*, es en esta modalidad donde se dieron mayores opciones, tales como, los ritmos y tiempos de aprendizaje y evaluación; se ofreció una diversidad de materiales de lectura en inglés o español, autonomía en el tiempo que dedicara el estudiante al estudio de los contenidos y se organizaron días de asesorías. Las horas de práctica en laboratorio sirvieron para aplicar la propuesta de gamificación. Finalmente, en la modalidad en línea el estudiante podía solicitar asesorías para aclarar dudas o entregar las evidencias de evaluación, ningún estudiante optó por esta opción. Otro elemento de elección fue el lenguaje de programación que pudo elegir el estudiante para entregar las evidencias de software sin importar la modalidad de aprendizaje.

iii. Ritmos y tiempo

El objetivo del curso y de cada unidad temática fue el mismo para todos los estudiantes, pero cada uno determinó su ritmo de aprendizaje, fue una oportunidad para que el estudiante eligiera qué y a qué ritmo aprender.

iv. Opciones de evaluación

En similitud a la primera intervención, los ejercicios y actividades evaluativas fueron ofrecidas para que el estudiante las eligiera, tales como codificar programas en el lenguaje de programación que eligiera el estudiante y de acuerdo a la forma de trabajo la revisión de las evidencias fue presencial o en línea; había una fecha límite de entrega, pero los estudiantes acordaron con el docente el día para su revisión.

En la segunda iteración, los grupos de control y experimental fueron evaluados con listas de cotejo que fueron desarrolladas por un grupo de profesores que impartieron el curso, los cuales se reunieron en academia y determinaron 9 listas de cotejo que evaluarían los productos académicos, lo que permitió tener un mismo parámetro de evaluación de competencias de los estudiantes, además sirvió como un elemento de medición de la eficiencia de usar la estrategia de gamificación en la adquisición de habilidades de programación, es decir la dimensión cognitiva.

5.2.2.2 Gamificación

La experiencia de gamificación en la iteración 1, consiste en la aplicación de cinco retos colaborativos a lo largo del curso cuyo objetivo fue crear un *software* con alguna funcionalidad específica, el logro de cada reto se recompensó con una serie de insignias, las cuales tenían un valor determinado de acuerdo al reto logrado. En la segunda iteración, la gamificación se usó al final del cuatrimestre y no a lo largo del periodo enero - abril, debido a que se observó un amplio espacio de tiempo entre los desafíos propuestos, ahora la propuesta consistió en utilizar las últimas 5 sesiones de trabajo de dos horas que se tuvieron frente a grupo para que fueran resueltos los retos y se generó el producto de

Intervención con gamificación, educación personalizada y *B-Learning*

proyecto final indicado por la academia de la asignatura de Programación. En la Tabla 42 se resume el sistema gamificado que se utilizó en la intervención.

Tabla 42. Sistema gamificado. Fuente: Elaboración propia

Dinámicas	Mecánicas	Componentes
Emocional: Retroalimentación inmediata, publicación de resultados.	Retos: Desafíos desarrollados de manera individual y colaborativa.	Puntos: Se otorgaron por el logro de los retos.
Narrativa: Historia contextual del reto.	Oportunidades: Ofrecer varios intentos para llegar a la respuesta correcta.	Insignias: Mostraban el avance en el medallero.
Progresión: Visualización de avances.		Avatares: El equipo diseñó su propio avatar que lo identificó.
Social: Trabajo colaborativo		Medallero: Fue el objeto en el que se mostraron públicamente los resultados.

Las dinámicas sobre las que se desarrolló la experiencia gamificada fueron: la emocional, presente en el sentido de que los estudiantes recibieron retroalimentación inmediata por parte del profesor, cuyo papel se limitó a ser guía y a alentar a los discentes, ya sea reconociendo su trabajo o motivándolos a lograr el reto; la de progresión, debido a que los participantes podrían ver sus avances; y la social, los estudiantes pudieron trabajar colaborativamente en la solución de alguno de los retos, permitiendo que entre ellos conocieran sus debilidades y fortalezas para promover el estímulo entre compañeros.

Cada reto se enmarcó dentro de una historia (dinámica narrativa) escrita que serviría de enganche para el desarrollo del programa de computadora, con el fin de crear emoción, curiosidad e interés. Durante el curso, los participantes podían visualizar sus logros por medio de un medallero en el cual se encontraban los nombres de los equipos identificados con su respectivo avatar, así como los nombres de los estudiantes, los retos superados, y las insignias ganadas por equipo. En cada uno de los retos, se permitió el trabajo en equipo para motivar la finalización de las actividades solicitadas.

La retroalimentación inmediata juega un papel importante y el ofrecer varias oportunidades para lograr el reto motiva el enganche para el logro de las actividades, pues “fallar es una fase del aprendizaje”. La finalización del reto hizo acreedor al equipo de 10 puntos. La funcionalidad completa del programa indica que el reto fue completado, si no fue programada alguna funcionalidad el equipo gana 7 puntos, la sola participación del equipo sin entregar algún programa en el reto otorga 3 puntos. También se ofrecen puntos extras que pueden ser asignados a los estudiantes por evidenciar una aportación extra en la codificación, participación o trabajo colaborativo. En cuanto a las insignias, estas fueron diferentes y de acuerdo al tipo de reto logrado. Las insignias se describen a continuación y se pueden ver en la Figura 29.

Insignia POO – cuando el equipo explica y ejemplifica los fundamentos de la programación orientada a objetos, es decir, los conceptos de abstracción, herencia, encapsulamiento y polimorfismo, así como clase y objeto.

Insignia IDE – cuando el equipo demuestra el uso correcto del entorno de desarrollo para poder crear, abrir, guardar, compilar, ejecutar y depurar un proyecto de programación en el lenguaje de programación elegido.

Insignia CLASE – cuando el equipo define una clase de forma correcta considerando la sintaxis del lenguaje de programación determinado para definir los atributos, propiedades y métodos de la clase, así como el uso del método constructor cuando sea adecuado definirlo.

Insignia EC – cuando el equipo utiliza las estructuras de control para resolver el problema determinado en el reto 3 o siguiente del curso. Al menos se debe usar una estructura de selección y repetición.

Insignia ENCAPSULAMIENTO (EN) – cuando el equipo haga uso adecuado de los modificadores de acceso en la definición de las clases que correspondió a las necesidades de funcionalidad del programa desarrollado a partir del reto 3.

Intervención con gamificación, educación personalizada y *B-Learning*

Insignia HERENCIA (HE) – cuando el equipo realice la aplicación del concepto de herencia a partir del reto 4, generando una clase base y al menos dos clases derivadas según el problema a resolver.

Insignia POLIFORMISMO (PO) – cuando el equipo, a partir del reto 4, aplique en la definición de los métodos involucrados en las clases el concepto de polimorfismo para resolver el problema indicado.

Insignia ARREGLO (AR) – cuando el equipo, a partir del reto 3, utilice la estructura de datos arreglo para el manejo de información del mismo tipo.

Insignia ME – última insignia que el equipo puede ganar en el curso a partir del reto 3, cuando agregue a su codificación el manejo de excepciones para evitar problemas de ejecución del programa.

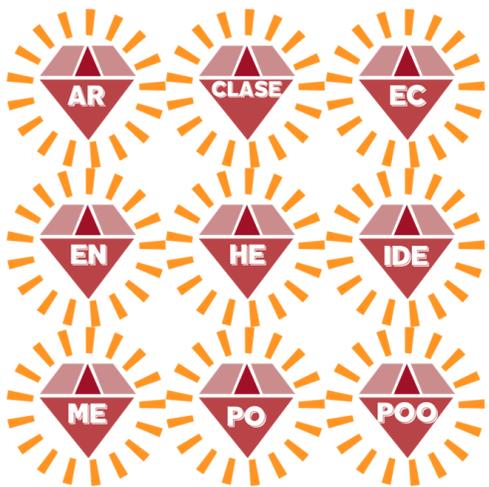


Figura 29. Insignias utilizadas durante el curso de Programación. Fuente: Elaboración propia

Los cinco retos creados iniciaban con la siguiente narrativa:

El año, 2017, las condiciones muy diferentes a la forma de desarrollar software en comparación al inicio del siglo y nada que decir de la extensión de dispositivos a los que hay que crear programas, ya no solo a las computadoras ahora también a la amplia cantidad de teléfonos móviles y aparatos electrónicos que interactúan entre sí o están conectados a la nube; el desafío parece mayor y unos cuantos los que capitalizan a escala mundial. ¿Cuál es el proceso para desarrollar software? ¿Qué reglas hay que seguir? ¿Cuándo saber si

alguien está listo como programador? ¿Qué plataforma de desarrollo aprender? Las preguntas en los estudiantes son muchas y la inseguridad por saber si el tiempo se aprovecha crece cuando se observa todo lo que se puede crear en videojuegos, app, realidad virtual, internet de las cosas y robótica. Hay que iniciar con el primer paso, tener una metodología para crear programas y una vez superada esa prueba crecer como programador para vencer los desafíos que se tienen para ser autónomo en el aprendizaje, activo en el trabajo en equipo y creativo en las soluciones de software; ser parte de una élite de programación. ¿Crees poder pertenecer a esta élite? Te invito a demostrarlo jugando Élite de programación.

Posteriormente se describía el reto a resolver, el primero se muestra a continuación:

El primer reto del juego consiste en demostrar tus habilidades para modelar diagramas de clases que permiten ofrecer una solución de software, al mismo tiempo que muestras el uso de los conceptos de la programación orientada a objetos, tales como abstracción, encapsulamiento, herencia, y polimorfismo. Lee con atención la problemática siguiente y resuelve las actividades indicadas, así podrás obtener la insignia POO y avanzar con el siguiente desafío.

Como se comentó anteriormente, el contenido del curso fue el mismo en ambos cuatrimestres, pero en la segunda iteración hubo más práctica de ejercicios de programación en laboratorio por cada uno de los temas, es decir, práctica del entorno de programación Visual Studio C# para crear aplicaciones del tipo *Windows Forms*, inicialmente se usaron los controles para lectura de datos (cajas de texto), escritura (etiquetas) y botones para hacer cálculos sencillos incluyendo la lógica de la solución en el evento clic del botón; integración de ejercicios en programas más complejos, distribución del material de estudio teórico de los conceptos de clases y objetos en la red social *Facebook* del grupo, así como los conceptos de abstracción, encapsulamiento, herencia y polimorfismo, y no se solicitó un estudio previo a la clase presencial de teoría. En consecuencia, los

Intervención con gamificación, educación personalizada y *B-Learning*

estudiantes del grupo experimental estuvieron en mejores condiciones para resolver todos los retos cubriendo todos los temas de las unidades temáticas del curso de Programación.

De la primera iteración, se obtuvieron aspectos de mejora y fortalecimiento de la experiencia de gamificación a partir del análisis de los resultados obtenidos, principalmente de los cualitativos, es decir, las opiniones de los estudiantes y de la observación reflexiva del investigador que realizó la intervención. Además, se utilizó una rúbrica de meta-evaluación basada en el modelo de Stufflebeam (2011), el cual coadyuva a delinear, obtener y aplicar información descriptiva y juiciosa sobre la utilidad, la factibilidad y la precisión de una evaluación y su naturaleza sistemática. La meta-evaluación permite desarrollar una reflexión sistematizada cuyo propósito es determinar las fortalezas y áreas de oportunidad de una estrategia didáctica, en este caso de la gamificación.

La Tabla 43 contiene la rúbrica respondida para la dimensión formativa de la meta-evaluación, mientras la Tabla 44 contiene la rúbrica contestada para la dimensión proactiva. Con base en el análisis meta-evaluativo, se obtuvieron los siguientes aspectos de mejora y que se aplicaron en la segunda iteración de la experiencia gamificada:

- Se observó que el tiempo entre las clases donde se realizaban las sesiones de gamificación fue amplio en días, lo que posiblemente produjo la falta de atención y compromiso para realizar las actividades previas que permitieran preparar a los estudiantes.
- Se observó falta de emoción por hacer un reto después del otro a pesar de informar días antes que se realizaría una sesión de gamificación.

Por lo anterior, como se comentó previamente, para la segunda iteración de la estrategia, enero – abril de 2018, se propuso reducir el tiempo entre las sesiones de gamificación, en lugar de realizarse a lo largo del cuatrimestre se llevaría a cabo en las cinco últimas semanas del curso. Se diseñó un proyecto final de programación que requería un tiempo promedio de 10 horas para su desarrollo, así se determinaron 5 sesiones de gamificación de dos horas de duración distribuyendo el desarrollo del

software por etapas. Un elemento nuevo que se integró fue considerar que si los equipos no terminaran la funcionalidad completa del programa en las dos horas de la sesión, obtendrían 7 puntos y podrían entregar posteriormente el programa y entonces poder obtener las insignias respectivas. Los demás elementos de gamificación se conservaron sin ningún cambio.

Tabla 43. Meta-evaluación en la dimensión formativa. Fuente: Basada en el modelo Stufflebeam (2011)

Dimensión	Pregunta	Respuesta	Mejoras
Formativa	¿Qué usos fueron dados al resultado de evaluación?	Determinar la calificación del curso de Programación al promediar los productos generados en cada sesión de gamificación.	Determinar listas de cotejo para cada producto de evaluación que sean comunes entre un grupo experimental y uno de control, incluir una sección de observaciones donde se puedan anotar las acciones a mejorar en caso de no cumplir completamente con el producto de software además de los comentarios del trabajo en equipo o individual si se considera adecuado.
	¿Cuántos reportes fueron dados a los estudiantes? ¿Fueron suficientes para fines de evaluación y aprendizaje?	Ninguno, la retroalimentación se daba previo al fin de la sesión de gamificación, posteriormente no se entregaba por escrito, solo el resultado que se mostraba en el medallero servía de evidencia y la calificación del curso fue el promedio de las sesiones.	Continuar con la retroalimentación previa al final de la sesión, pero entregar posteriormente un reporte de los resultados observados según el contenido de la sesión, es decir, la lista de cotejo con sus respectivas observaciones.
	¿Fueron suficientemente claros los resultados para los estudiantes? ¿Consiguieron comprender los datos y/o retroalimentación relacionada a su proceso de aprendizaje?	Si, durante la retroalimentación se explicaba el resultado, además de responder a preguntas específicas de los estudiantes, no solo hubo aceptación de los resultados, además conocieron las mejoras que podían hacer.	Generar una evidencia de los resultados obtenidos por los participantes que pudieran usar para su consulta después de cada sesión de gamificación.
	¿Qué decisiones académicas han sido realizadas a partir de los resultados?	Debido a que hubo tiempo entre las sesiones de gamificación, se fue modificando el alcance de los productos de evaluación para ajustarse al avance que se observó, el cual desafortunadamente se tuvo que reducir.	Hacer de forma continua las sesiones de gamificación para evitar que se pierda un ritmo de trabajo. Trabajar previamente ejercicios similares en el alcance de los productos de evaluación.

Intervención con gamificación, educación personalizada y *B-Learning*

Tabla 44. Meta-evaluación en la dimensión proactiva. Fuente: Basada en el modelo *Stufflebeam* (2011)

Dimensión	Preguntas	Respuestas	Mejoras
Proactiva	¿Qué resultados han sido alcanzados?	Trabajo en equipo, programación orientada a objetos, diseño y codificación de clases y objetos, abstracción y encapsulación.	Trabajar ejercicios previos de herencia, polimorfismo, manejo de excepciones y arreglos básicos y de objetos para poder cumplir con los productos de evaluación de las dos últimas sesiones.
	¿Son congruentes con los objetivos?	Solo en las primeras dos sesiones, para las tres sesiones restantes se modificaron restando el alcance de evaluación, incluso se llegó a repetir un producto.	Se propone que con la ejercitación previa de ejercicios correspondientes a las tres últimas sesiones sea completa la congruencia de los resultados.
	¿Existen algunos efectos negativos?	Algunos estudiantes sintieron frustración por el trabajo en equipo, pero más bien correspondió a un exceso de confianza y no haberse preparado para las sesiones con estudio previo.	Si la actividad de gamificación se deja al final del cuatrimestre, se pueden ejercitar las habilidades previas de programación.
	¿Existen algunos efectos positivos?	Sí, motivación por la forma de exponer la situación del problema a resolver.	Reforzar este hecho cada vez que se aplique la actividad.
	¿Los resultados sugieren que las metas, diseños y procesos deban ser modificados?	Sí, Incrementar el hecho de que los estudiantes puedan realizar el producto de software con una ejercitación previa usando una problemática similar.	Cubrir el contenido del curso antes de realizar las actividades de gamificación. Que la comprensión y codificación de los temas de herencia, polimorfismo, manejo de excepciones y arreglos, previamente practicados de mejores resultados a los estudiantes.
	Costo: De acuerdo a las horas del proceso de gamificación demandadas al profesor, ¿Existe un resultado positivo? De acuerdo a los materiales, licencias o cualquier otro gasto para las actividades de aprendizaje, ¿Existe un resultado positivo?	Si, el contexto de resolver un reto bajo condiciones de tiempo y características de programación permite que los estudiantes organicen mejor sus propuestas de diseño y codificación de forma individual o en el trabajo de equipo. El costo de los elementos visuales de gamificación no es alto y genera buenos resultados de motivación en los estudiantes.	Trabajar la propuesta de una forma continua o con menos días entre las sesiones.
	Impacto: ¿Qué impacto fue realizado con la audiencia objetivo? ¿Los resultados sugieren que el proyecto es exitoso? Es posible mostrar o inferior que los resultados tienen alguna conexión entre el diseño y la implementación para alcanzar las metas de aprendizaje.	El mayor impacto fue que los estudiantes enfrentaron el trabajo en equipo en un contexto donde se tenía que cumplir con requisitos de tiempo y de desarrollo de software, si no tenían su trabajo previo de estudio no aportaron mucho al trabajo colaborativo e incluso no fue posible que terminaran los productos de evaluación, lo que llevó a realizar ajustes de contenidos, la motivación para realizar los desafíos fue clara pero su exceso de confianza en considerar un simple juego la actividad de gamificación sorprendió a los estudiantes, aun así la propuesta fue bien aceptada y reflejó resultados favorables tanto de evaluación de las metas del curso como el trabajo grupal.	Cada grupo es diferente y tendrá que verificarse si es buen candidato para hacer la actividad de gamificación, un factor a considerar es la capacidad de autonomía de los estudiantes y el trabajo en equipo, así como haber cubierto los temas del curso previamente para después hacer los desafíos.
	En general, ¿Qué valiosos fueron los resultados e impactos del esfuerzo?	Todos los estudiantes que participaron aprobaron el curso de programación y reportaron motivación en hacer las actividades de gamificación, solo en las dos primeras sesiones se puede considerar que hubo éxito completo y fue desafortunado haber tenido que modificar el alcance de las tres sesiones finales debido a la falta de estudio previo de los estudiantes.	Si es posible aplicar en una segunda ocasión la propuesta entonces que los estudiantes tengan mayor práctica de programación respecto a los alcances de desarrollo de software de las últimas tres sesiones de gamificación, así como reducir el tiempo entre cada desafío.

La Tabla 45 contiene la información de las fechas en que se llevaron a cabo las sesiones de gamificación de la iteración 2, el alcance de temas, habilidades y conocimientos que en cada sesión

se evaluó en el desarrollo de los respectivos desafíos, así como el número-nombre de la lista de cotejo usada para su evaluación con el número de indicadores que evaluó.

Tabla 45. Organización de la actividad de gamificación en la segunda iteración. Fuente: Elaboración propia

Sesión	Fecha – 2018	Conocimientos competencias	Lista de cotejo	Duración horas
1	22 de Marzo	Conceptos de Programación Orientada a Objetos Diseño de diagramas de clase	1-Programación Fundamentos (10-Indicadores)	2
2	10 de Abril	Uso del entorno de desarrollo visual C#	2-Ambiente Desarrollo (10-Indicadores)	2
3	12 de Abril	Diseño y codificación de una clase. Uso correcto de tipos de datos. Encapsulación de datos. Uso de estructuras de control	3-ProgramaciónPOO (12-Indicadores) 4-PrograEstructurasControl (9-Indicadores)	2
4	17 de Abril	Diseño y codificación de herencia y polimorfismo Definición de una clase base y clases derivadas, sobre escritura de un método común en las clases derivadas.	5-Herencia, (10-Indicadores) 6-Polimorfismo, (10-Indicadores) 7-Arreglos (9-Indicadores)	2
5	19 de Abril	Uso del manejo de excepciones	8-ManejoExcepciones (4-Indicadores) 9-Integrador (9-Indicadores)	2

5.2.3 Instrumentos

El instrumento que se diseñó para evaluar la percepción del estudiante sobre la gamificación del curso está enfocado en tres dimensiones: cognitiva, emotiva y social. Las preguntas se encuentran en la Tabla 46.

Tabla 46. Instrumento encuesta para intervención 2 al final de la iteración 1. Fuente: Elaboración propia

Dimensión	Pregunta	Opciones
Cognitiva	La retroalimentación inmediata por parte del profesor en la actividad gamificada me dio oportunidad de analizar mejor mi respuesta.	Totalmente de acuerdo
	Escuchar las opiniones de los compañeros de equipo me permitió comprender mejor el reto gamificado.	De acuerdo
	El intercambio de opiniones con mis compañeros de equipo de las actividades gamificadas se dio de manera respetuosa (por ejemplo, yo escuchaba con atención la opinión de mis compañeros y viceversa).	En desacuerdo
	Resolver los retos gamificados me ayudó a comprender mejor los temas correspondientes.	Totalmente en desacuerdo
	El nivel de dificultad de los retos me pareció adecuado a mis conocimientos previos sobre el tema.	
Emotiva	El ambiente de competencia que se vivió en la actividad gamificada me motivó a resolver los retos.	
	Cada vez que mi equipo lograba resolver un reto me sentía contento y motivado.	
	Ver los avances de los otros equipos en el tablero me motivó a concentrarme más en la actividad gamificada.	
	Ver el avance de mi equipo en el tablero nos producía una emoción positiva.	
Social	Me agradaba que en cada parcial hubiera una actividad gamificada.	
	Mi equipo se motivaba a avanzar más rápido en cada nivel cuando veía el tablero de avance en las actividades gamificadas la mayoría de las veces.	
	Las actividades gamificadas basadas en retos son mejores cuando se desarrollan en equipo que individualmente.	
	Mi participación hubiera sido mejor en la actividad gamificada si me hubieran tocado otros compañeros en el equipo.	

Finalizada la primera iteración, se determinó también realizar ajustes en el instrumento para mejorar la información que se obtuvo sobre todo en el enfoque CUAL. La Tabla 47 contiene los elementos

agregados al instrumento encuesta usado al finalizar la segunda iteración, destaca que las respuestas sean abiertas.

Tabla 47. Información adicional de la encuesta para la iteración 2. Fuente: Elaboración propia

Dimensión	Pregunta
Abierta Cognitiva	– Haber desarrollado el proyecto final a través de los 5 desafíos, ¿Qué te permitió aprender de nuevo? ¿Haber desarrollado los desafíos como proyecto final, te permitieron aplicar los temas del curso? ¿Cuáles temas aplicaste?
Abierta Emocional	– ¿Te motivaron los desafíos? Describe por qué. Explica por favor, ¿Qué tipo de sentimientos o emociones te causaron ver tus avances en el medallero?
Abierta Social	– ¿De qué manera trabajar con otros compañeros, favoreció o no, la resolución de los desafíos? ¿Qué te pareció compartir los resultados de tu proyecto con los otros equipos?
Abierta Mejorar	– ¿Qué se puede mejorar de la actividad para las siguientes generaciones?

5.2.4 Fase intervención

Iteración enero – abril de 2017

La Figura 30 muestra el medallero al final de cuatrimestre, la lectura de los resultados en laboratorio de cada sesión fue la siguiente.



Figura 30. Medallero al final del cuatrimestre iteración 1. Fuente: Elaboración propia

Intervención con gamificación, educación personalizada y *B-Learning*

Primer reto.

Los cuatro equipos obtuvieron 7 puntos cada uno debido a que ninguno pudo entregar la funcionalidad completa del producto solicitado.

Segundo reto.

Aunque durante la actividad la mayoría de los estudiantes podía ejecutar la funcionalidad solicitada, ningún equipo termino el programa. Los cuatro equipos obtuvieron 3 puntos cada uno, con la opción de poder reportar el producto posteriormente. Solo un equipo entregó la evidencia solicitada por lo que se le otorgó la insignia de IDE.

Tercer reto.

Tres equipos logran terminar el producto solicitado, pero solo uno con la funcionalidad completa (10 puntos) y ganó la insignia EN, un equipo ganó tres insignias (IDE, CLASE y EN) a pesar de tener limitada la funcionalidad (7 puntos).

Cuarto y Quinto reto.

Desafortunadamente los resultados son bajos al no tener productos terminados, prácticamente asisten a la actividad sin estudio previo tratando de resolver la problemática hasta donde el tiempo les permita.

Al finalizar el cuatrimestre los 13 estudiantes que terminaron (1 se dio de baja) y aprobaron el curso realizaron la encuesta acerca de su percepción sobre la gamificación del curso. Los porcentajes a las preguntas se encuentran en la Tabla 48.

Tabla 48. Resultados de encuesta final iteración 1 intervención 2. Fuente: Elaboración propia

Dimensión	Pregunta	Porcentajes
Cognitiva	La retroalimentación inmediata por parte del profesor en la actividad gamificada me dio oportunidad de analizar mejor mi respuesta.	61,5% Totalmente de acuerdo 30,8% De acuerdo 7,7% En desacuerdo
	Escuchar las opiniones de los compañeros de equipo me permitió comprender mejor el reto gamificado.	Para ambas preguntas 53,8% Totalmente de acuerdo 38,5% De acuerdo 7,7% En desacuerdo
	El intercambio de opiniones con mis compañeros de equipo de las actividades gamificadas se dio de manera respetuosa (por ejemplo, yo escuchaba con atención la opinión de mis compañeros y viceversa).	53,8% Totalmente de acuerdo 38,5% De acuerdo 7,7% En desacuerdo
	Resolver los retos gamificados me ayudó a comprender mejor los temas correspondientes.	61,5% De acuerdo 30,8% Totalmente de acuerdo 7,7% En desacuerdo
	El nivel de dificultad de los retos me pareció adecuado a mis conocimientos previos sobre el tema.	53,8% De acuerdo 38,5% Totalmente de acuerdo 7,7% En desacuerdo
	Emotiva	El ambiente de competencia que se vivió en la actividad gamificada me motivó a resolver los retos.
Cada vez que mi equipo lograba resolver un reto me sentía contento y motivado.		46,2% De acuerdo 30,8% Totalmente de acuerdo 23,1% En desacuerdo
Ver los avances de los otros equipos en el tablero me motivó a concentrarme más en la actividad gamificada.		46,2% Totalmente de acuerdo 38,5% De acuerdo 15,4% En desacuerdo
Ver el avance de mi equipo en el tablero nos producía una emoción positiva.		46,2% Totalmente de acuerdo 46,2% De acuerdo 7,7% En desacuerdo
Me agradaba que en cada parcial hubiera una actividad gamificada.		46,2% Totalmente de acuerdo 46,2% De acuerdo 7,7% En desacuerdo
Social		Mi equipo se motivaba a avanzar más rápido en cada nivel cuando veía el tablero de avance en las actividades gamificadas la mayoría de las veces.
	Las actividades gamificadas basadas en retos son mejores cuando se desarrollan en equipo que individualmente.	53,8% Totalmente de acuerdo 30,8% De acuerdo 15,4% En desacuerdo
	Mi participación hubiera sido mejor en la actividad gamificada si me hubieran tocado otros compañeros en el equipo.	61,5% En desacuerdo 15,4% Totalmente de acuerdo 15,4% De acuerdo 7,7% Totalmente en desacuerdo

Intervención con gamificación, educación personalizada y *B-Learning*

Iteración enero – abril de 2018

En la Figura 31 se muestra el medallero al final del cuatrimestre del grupo experimental. Como se puede observar, un par de equipos, particularmente los equipos “Brazos cruzados” y “AJME” no lograron ganar suficientes insignias, lo que indica que sus productos de software no tuvieron los elementos requeridos del tema a evaluar.



Figura 31. Medallero final de la segunda iteración. Fuente: Elaboración propia

A continuación, se explica brevemente como se desarrolló cada reto en general por el grupo experimental.

Primer reto.

Todos los equipos indicaron que habían terminado a tiempo el desafío, por lo que se dio retroalimentación 20 minutos antes de terminar la clase, se detectaron algunas carencias comunes en el producto entregado en cada uno de los equipos por lo que obtuvieron 7 puntos de 10, pero quienes describieron correctamente los conceptos de Programación Orientada a Objetos ganaron la insignia POO, cuatro de los seis equipos lo lograron, se consideró buena participación de todos los estudiantes.

Segundo reto.

Solo un equipo no pudo terminar el reto (Brazos cruzados), un segundo equipo tuvo varias deficiencias en el código del programa solicitado, tres equipos entregaron el trabajo correctamente, pero no terminaron la descripción de la funcionalidad del entorno de desarrollo, no obstante, posteriormente ganaron la Insignia IDE cuando enviaron el trabajo completo. Un equipo cumplió satisfactoriamente con toda la actividad (Grupo Delta) aunque no se observó trabajo colaborativo, ganaron los 10 puntos y representó un reto a vencer para algunos de los otros equipos.

Tercer reto.

Todos los equipos nuevamente se esforzaron por entregar el producto completo, pero se encontraron problemas al momento de hacer la retroalimentación, motivo por el cual se condicionaron las insignias del desafío que solo ganaron posteriormente tres equipos, el resto se conformó con haber logrado los 7 puntos. Hay que hacer notar que el equipo NASUL'S se propuso alcanzar los 10 puntos para igualar al equipo Delta.

Cuarto reto.

El contenido del reto fue el más complicado de todas las sesiones, solo un equipo logró casi terminar el trabajo por lo que se condicionaron las insignias que ganaron posteriormente, cinco equipos lograron el mínimo de tres puntos y no se preocuparon por ganar insignias.

Quinto reto.

El gran triunfo del equipo integrado solo por mujeres (NASUL's) obtuvieron los 10 puntos y ganaron la insignia ME, se observó claramente su esfuerzo y confianza del resto de los equipos que a pesar de tratar de terminar no cumplían con mucha de la funcionalidad del programa y al ser la última actividad no se preocuparon por obtener la insignia, se concentraron en entregar el programa completo.

Intervención con gamificación, educación personalizada y *B-Learning*

Nuevamente al terminar el curso, la mayoría de los estudiantes respondieron a la encuesta final, 23 de 29 estudiantes. La Tabla 49 y Tabla 50 contienen la información de la encuesta de la iteración 2, la primera semejante a la usada en la iteración 1 y en la segunda las preguntas abiertas que fueron agregadas.

Tabla 49. Información de la encuesta al final del curso en la iteración 2. Fuente: Elaboración propia

Dimensión	Pregunta	Porcentajes
Cognitiva	La retroalimentación inmediata por parte del profesor en la actividad gamificada me dio oportunidad de analizar mejor mi respuesta.	43,5% Totalmente de acuerdo 56,5 De acuerdo
	Escuchar las opiniones de los compañeros de equipo me permitió comprender mejor el desafío gamificado.	34,8% Totalmente de acuerdo 60,9% De acuerdo 4,3% En desacuerdo
	Resolver los desafíos gamificados me ayudó a comprender mejor los temas correspondientes.	26,1% Totalmente de acuerdo 73,9% De acuerdo
	El nivel de dificultad de los desafíos me pareció adecuado a mis conocimientos previos sobre el tema.	21,7% Totalmente de acuerdo 78,3 De acuerdo
Emotiva	El ambiente de competencia que se vivió en la actividad gamificada me motivó a resolver los desafíos.	26,1% Totalmente de acuerdo 73,9% De acuerdo
	Ver los avances de los otros equipos en el medallero me motivó a concentrarme más en la actividad gamificada.	52,2% Totalmente de acuerdo 47,8% De acuerdo
	Cada vez que mi equipo lograba resolver un desafío me sentía contento y motivado.	73,9% Totalmente de acuerdo 26,1% De acuerdo
	Ver el avance de mi equipo en el medallero nos producía una emoción positiva.	52,2% Totalmente de acuerdo 47,8% De acuerdo
	Me agradó que el proyecto final fuera una actividad gamificada.	47,8% Totalmente de acuerdo 52,2% De acuerdo
Social	El intercambio de opiniones con mis compañeros de equipo de las actividades gamificadas se dio de manera respetuosa (ej. yo escuchaba con atención la opinión de mis compañeros y viceversa)	39,1% Totalmente de acuerdo 47,8% De acuerdo 8,7% En desacuerdo 4,3% Totalmente en desacuerdo
	Mi equipo se motivaba a avanzar más rápido en cada nivel cuando veía el tablero de avance en las actividades gamificadas la mayoría de las veces.	56,5% Totalmente de acuerdo 39,1% De acuerdo 4,3% En desacuerdo
	Las actividades gamificadas basadas en desafíos son mejores cuando se desarrollan en equipo que individualmente.	43,5% Totalmente de acuerdo 52,2 De acuerdo 4,3% En desacuerdo
	Mi participación hubiera sido mejor en la actividad gamificada si me hubieran tocado otros compañeros en el equipo.	17,4% Totalmente de acuerdo 26,1% De acuerdo 39,1% En desacuerdo 17,4% Totalmente en desacuerdo

Tabla 50. Información adicional de la encuesta de la iteración 2. Fuente: Elaboración propia

Dimensión	Pregunta	Resultado
Abierta – Cognitiva	Haber desarrollado el proyecto final a través de los 5 desafíos, ¿Qué te permitió aprender de nuevo?	7 trabajo en equipo 10 repasar 6 un tema específico
	¿Haber desarrollado los desafíos como proyecto final, te permitieron aplicar los temas del curso? ¿Cuáles temas aplicaste?	21 Herencia 19 Polimorfismo 8 POO 5 Arreglos 5 Manejo de errores 3 Encapsulamiento 2 Métodos 2 Abstracción
Abierta – Emocional	¿Te motivaron los desafíos? Describe por qué.	11 Aplicación de conocimientos 8 Competir entre los equipos 4 Trabajo en equipo
	Explica por favor, ¿Qué tipo de sentimientos o emociones te causaron ver tus avances en el medallero?	17 Alegría, Felicidad, Satisfacción 7 Preocupación, Nerviosismo, Estrés 2 Orgullo 2 Motivación, Superación 1 Impotencia 1 Desconfianza 1 Frustración
Abierta – Social	¿De qué manera trabajar con otros compañeros, favoreció o no, la resolución de los desafíos?	18 Favoreció diversidad de ideas, aportación, complementar conocimiento 5 No favoreció, dificultad de comunicación.
	¿Qué te pareció compartir los resultados de tu proyecto con los otros equipos?	21 Muy bien 2 Incomodidad
Abierta – Mejorar	¿Qué se puede mejorar de la actividad para las siguientes generaciones?	14 Ningún cambio 3 Ofrecer pistas 3 Organizar equipos de otra forma 2 Más tiempo 1 Mostrar medallero al final

5.2.5 Fase analítica

De la primera iteración, como observador de la intervención se presentan algunos comentarios que se tuvieron por cada una de las sesiones en la actividad de gamificación, es decir, por cada uno de los retos presentados a los estudiantes.

Intervención con gamificación, educación personalizada y *B-Learning*

Primer reto.

Los estudiantes no completaron el ejercicio, indicaron que por falta de tiempo, pero se debió a que realizaron trabajo de investigación durante la clase en lugar de haberlo hecho previamente. Se observa trabajo colaborativo en la organización de las actividades para completar el reto.

Segundo reto.

A pesar de haber recibido la recomendación de estudiar previamente el material teórico; gastaron nuevamente tiempo en el laboratorio.

Tercer reto.

Se observó conformismo en el equipo que solo obtuvo 3 puntos. No hubo trabajo colaborativo durante el desarrollo del reto, de forma aislada trabajaron los estudiantes para programar, al finalizar el tiempo entre ellos eligieron que programa presentar para evaluación de cada uno de los equipos.

Cuarto y Quinto reto.

Durante el cuatrimestre se realizaron ajustes de los últimos retos con base a la evidencia de aprendizaje de los estudiantes.

A diferencia de la primera iteración, a continuación, se presentan algunos comentarios que como observador de la intervención en su segunda iteración aportan información acerca de la forma en que trabajó cada uno de los equipos en el grupo experimental y que coincide también con los resultados en el medallero.

Equipo NASULS

Desde el desafío 1, mostró un trabajo en equipo organizado, incluso observaban el trabajo de quien tomaba el control y hacían pertinentes observaciones para ir haciendo la actividad paso a paso, fue un equipo integrado únicamente por mujeres, los comentarios entre ellas permitieron aprender cómo

crear el código de los programas solicitados, después de las sesiones se observó que seguían trabajando en equipo para resolver completamente los desafíos que no pudieron terminar, motivo por el cual lograron obtener todas las insignias.

Equipo BRAZOS CRUZADOS

Tres fueron los integrantes del equipo y desafortunadamente nunca hubo compromiso con las sesiones de gamificación, a veces faltaba un estudiante y en un desafío faltaron dos estudiantes, no se observó trabajo en equipo, quien tomaba el control del trabajo no consultaba a su compañero y viceversa. Los resultados fueron muy por debajo de lo esperado.

Equipo LOS SUPER CONOCIDOS

Tomaron con mucha seriedad cada desafío y desde la sesión 1 dos estudiantes tomaron el control del equipo, los demás estudiantes aportaron con base a su experiencia de programación, observaban y daban su opinión. Fue el equipo que consciente del tiempo que tenía por desafío trataban de terminar las actividades solicitadas, pero siempre les faltó tiempo y entregaban posteriormente los productos solicitados con la funcionalidad básica, pero sin cubrir los requerimientos adecuados para obtener las insignias finales.

Equipo AJME

En todos los desafíos tuvieron problemas de liderazgo, se les hizo la observación desde la sesión 1 pero no lograron organizarse para que el desarrollo de las actividades fuera exitoso, un estudiante iniciaba el control del equipo y después de un tiempo otro estudiante asumía el control, internamente se hacían grupos y solo solapaban las acciones de quienes en su momento hacían las actividades del desafío. En las últimas dos sesiones de trabajo una estudiante mejor hacía trabajo de las otras materias, el trabajo en equipo no fue bueno.

Intervención con gamificación, educación personalizada y *B-Learning*

Equipo JADEBY

A pesar de haber sido un equipo formado por una variedad de estudiantes respecto a su nivel de programación, esta característica llevó a los integrantes a sesiones donde había mucho diálogo para determinar la forma de realizar las tareas de cada desafío, fueron más lentos en su organización, pero su trabajo en equipo en cada sesión reportó funcionalidades básicas de los programas solicitados. Es el equipo que mejor trabajo colaborativo logro.

Equipo DELTA

Los integrantes trabajaron de forma individual, eligieron de entre ellos alguien que realizara las actividades del desafío mientras los demás hacían otras cosas de carácter personal, incluso vieron videos musicales. Se observó mucho juego entre ellos sin tener aportaciones efectivas al equipo, la responsabilidad de resolver el desafío fue casi siempre del mismo estudiante en las primeras sesiones, pero para las últimas las actividades rebasaban su destreza para resolver la problemática por sí solo.

En general fue hasta el desafío tres donde todos los equipos lograron una mejor organización, concentrados en la actividad donde un líder tomaba el control del equipo y los demás estudiantes aportaban, pero los productos de software no fueron correctos. Se enfrentaron a una forma muy próxima del trabajo en equipo donde se tiene que hacer un software, pero no es fácil entenderse o darse a entender con los compañeros de trabajo, se vio en ocasiones indiferencia, falta de compromiso y sobre todo dejar la responsabilidad a otro. El contenido de la retroalimentación fue también sobre tales observaciones del trabajo en equipo.

Acerca de la motivación, se observó que tuvieron entusiasmo en hacer la evaluación del proyecto final a través de la actividad de gamificación, pero con el paso de las sesiones algunos fueron perdiendo el interés debido a los resultados que lograban en el equipo, al no lograr definir un rol de trabajo o tomar definitivamente a juego el trabajo.

Cada grupo es diferente y a pesar de no tener mejores resultados en la dimensión cognitiva, la emotiva y social aportó mucho a los estudiantes, permitiendo hacer una mejor lectura de los resultados obtenidos en la segunda iteración. Aunque la mayoría de los estudiantes aprobó el proyecto final, hubo algunos que aprobaron el curso hasta el periodo extraordinario debido a que no habían acreditado la primera evaluación del curso, no existió desaliento para tales estudiantes que trabajaron bien en equipo ante tal situación, haber hecho la actividad fue motivación para que en periodo extraordinario acreditaran el curso.

La Tabla 51 contiene un comparativo de los resultados que se obtuvieron de cada una de las listas de cotejo en ambos grupos control y experimental de la segunda iteración. La Tabla 52 resume usando solo los totales a partir de la evaluación con las listas de cotejo. Los resultados promedio en todas las listas de cotejo son mayores en el grupo de control que en el grupo experimental.

Tabla 51. Detalle de resultados en grupo de control y experimental, iteración 2.

Equipo / Lista de cotejo	1	2	3	4	5	6	7	8	9	Total /
	10	10	12	9	10	10	9	4	9	83
1	7	8	9	9	10	10	8	4	9	74
2	9	8	10	9	10	10	8	4	9	77
3	8	8	10	9	6	9	7	4	8	69
4	5	8	10	7	7	10	8	4	9	68
5	8	6	8	9	7	9	6	4	9	66
6	5	4	9	7	6	6	6	4	7	54
7	8	8	7	7	8	9	5	4	8	64
8	7	8	9	7	7	10	8	4	9	69
Promedio	7.13	7.2	9.0	8.0	7.6	9.1	7.0	4.0	8.5	68
NASULS	6	6	9	8	10	10	8	4	9	70
BRAZOS CRUZADOS	5	3	3	0	2	1	0	0	2	16
LOS SUPER CONOCIDOS	6	7	8	6	3	2	4	1	4	41
AJME	3	6	4	1	3	2	2	1	2	24
JADEBY	7	6	8	7	6	6	5	2	8	55
DELTA	7	4	8	5	6	8	6	1	7	52
Promedio	5.6	5.3	6.6	4.5	5.0	4.8	4.1	1.5	5.3	43

Tabla 52. Concentrado de resultados en grupo de control y experimental, iteración 2.

<i>Equipo control</i>	<i>Promedio</i>	<i>Equipo experimental</i>	<i>Promedio</i>
1	89	NASULS	84
2	93	BRAZOS CRUZADOS	19
3	83	LOS SUPER CONOCIDOS	49
4	82	AJME	29
5	80	JADEBY	66
6	65	DELTA	63
7	77		
8	83		

5.2.6 Fase discusión

Tomando en consideración la metodología de investigación mixta basada en el diseño de triangulación concurrente utilizada en la tesis doctoral, se presenta la discusión de la intervención que combina gamificación, educación personalizada y *b-Learning*, a partir de la recolección de resultados de forma concurrente que tuvo dos iteraciones.

Los resultados del 2°C en el cuatrimestre enero – abril de 2017 son los siguientes: el porcentaje de deserción fue del 8,0 que es menor en 22,76 por ciento respecto a la mediana de los últimos 8 años (30,76); el porcentaje de acreditación del curso Programación fue del 92,0 que es mayor en 20,11 respecto a la mediana de los últimos 8 años (71,89); el promedio de calificación del mismo curso en los estudiantes del 2°C fue de 7,09 mayor en 0,06 por ciento respecto a la mediana de los últimos 8 años (7,03).

Los resultados del 2°I en el cuatrimestre enero – abril de 2018 son los siguientes: el porcentaje de deserción fue del 9,68 que es menor en 21,08 por ciento respecto a la mediana de los últimos 8 años; el porcentaje de acreditación del curso Programación fue del 90,32 que es mayor en 18.43 respecto a la mediana de los últimos 8 años; el promedio de calificación del mismo curso en los estudiantes del

2° I fue de 7,51 mayor en 0,48 por ciento respecto a la mediana de los últimos 8 años. La Tabla 53 reúne la información de las dos iteraciones.

Tabla 53. Comparativo de información en los últimos 8 años y de intervención 2.

Año	Porcentajes Segundo cuatrimestre		Promedio de Calificación
	Deserción	Acreditación	
2010	16,48	85,20	7,0
2011	33,69	70,45	6,29
2012	29,99	73,07	6,92
2013	28,22	73,87	7,07
2014	31,38	71,62	7,24
2015	30,89	71,09	6,94
2016	30,63	72,17	7,05
2017	32,25	69,57	7,09
Mediana	30,76	71,89	7,03
2°C 2017	8,0	92,0	7,09
2°I 2018	9,68	90,32	7,51

Por lo anterior, desde la perspectiva cuantitativa, los indicadores académicos obtenidos son mejores y la intervención que combinó gamificación, educación personalizada y *b-Learning* representó una novedad favorable para el caso de estudio y puede servir para otros contextos similares a nivel global.

Desde la perspectiva cualitativa, a partir de la encuesta realizada a final de cuatrimestre en ambas iteraciones, se tienen las siguientes observaciones con la intención de corroborar los resultados cuantitativos.

En cuanto a la dimensión cognitiva de la primera iteración, se observa que más del 90% de los estudiantes que aprobaron el curso consideró que el nivel de dificultad de los retos fue acorde a los conocimientos adquiridos, que el intercambio de opiniones con los compañeros de equipo facilitó la comprensión y que el reto en sí mismo contribuyó a la comprensión del tema que implicaba, lo que corresponde con el porcentaje de acreditación obtenido. En cuanto a la dimensión emotiva, la mayor parte de los estudiantes se sintieron motivados por el ambiente de competencia y por observar los

Intervención con gamificación, educación personalizada y *B-Learning*

logros en el medallero. Con respecto a la dimensión social, se observa que más del 70% de los estudiantes están de acuerdo en que el trabajo colaborativo favoreció el logro de los retos.

Los 11 estudiantes que optaron por la clase presencial en la iteración 1, no utilizaron ningún elemento de las opciones ofrecidas que permitieron una flexibilidad en el curso; el hecho de haber tenido a esos once estudiantes en clases tradicionales permitió personalizar ejercicios, trabajos y tareas, no en el contexto de lo ofrecido usando la plataforma *Moodle* ni participando en la gamificación, pero sí considerando en primer lugar que ellos eligieron aprender de esa forma y en consecuencia ajustase a sus necesidades para la adquisición de las competencias respectivas. Los 14 estudiantes del modelo semi-presencial fueron los que indicaron querer administrar su tiempo de estudio con los contenidos que hubo en la plataforma *Moodle* y entregar evidencias extras de evaluación para acreditar el cuatrimestre considerando los resultados desfavorables en las últimas dos sesiones de gamificación. El estudiante fue el centro de atención independientemente de su modalidad de aprendizaje en el curso de Programación, todos acreditaron, aunque hubo dos bajas académicas del grupo por otras razones.

Uno de los ajustes que se aplicó en la segunda iteración fue el tiempo y momento de la aplicación de la experiencia gamificada, en la primera se aplicó durante todo el curso y en la segunda se aplicó al final. Esto permitió que la atención estuviera focalizada en un lapso de tiempo más corto y como consecuencia los estudiantes estuvieron más motivados en resolver por lo menos los tres primeros retos con respecto a la primera iteración. Esto coincide con Hou y Li (2014), Hamari y Koivisto (2015) y Hamari et al. (2016) quienes encontraron que los retos favorecen a la motivación intrínseca y el flujo. Así que los retos pueden ser una estrategia didáctica para este tipo de cursos, no obstante, debe cuidarse la planeación y diseño de los mismos.

Aunque en la segunda iteración, se observó que dos equipos de seis lograron terminar los últimos dos desafíos (en la primera iteración ninguno lo logró), persistió el decaimiento en el logro de estos retos por lo que vale la pena reflexionar sobre su diseño, quizás disminuir la dificultad o quizás la cantidad de retos debería reducirse. Shuptrine (2013) observó algo similar, el flujo y engancho disminuyó en

los estudiantes que no contaban con las habilidades necesarias para resolver el reto asignado, por ello tanto el diseño como la cantidad de retos en un periodo de tiempo es algo que debe considerarse seriamente para que la estrategia sea una experiencia positiva para los estudiantes.

Con respecto a la dimensión cognitiva, los resultados de la segunda iteración fueron mayores que en la primera. Además, los estudiantes comentaron que aplicaron sus conocimientos, que aprendieron sobre un tema en específico y que repasaron los temas. Esto coincide con lo encontrado por Hamari et al. (2016) y Rincón-Flores et al. (2016), por lo que esta dimensión debe considerarse en el diseño de este tipo de actividades; más del 96% de los estudiantes estuvo de acuerdo con que el reto gamificado favoreció a su aprendizaje, ya que la retroalimentación inmediata les permitió mejorar el análisis de su respuesta y comprender de mejor manera los temas. La mejora de la intervención en su segunda iteración concuerda con lo aprendido por Llorens-Largo et al. (2016) en su propuesta, los estudiantes aplicaron de forma autónoma las habilidades que les permiten adquirir conocimiento.

En cuanto a la dimensión emotiva, en la segunda iteración, el 100% de los participantes que respondieron el cuestionario están de acuerdo en que la estrategia favorece esta dimensión, pues les agradó el ambiente de competencia y tuvieron emociones de alegría y satisfacción al observar sus logros en el tablero, esto es similar a lo encontrado por Hamari (2017) y Mekler et al. (2013). La motivación fue un elemento clave en el diseño de la intervención con el objetivo de obtener estudiantes satisfechos, receptivos en el proceso de enseñanza – aprendizaje y, por lo tanto, con menos probabilidad de que abandone los estudios universitarios en coincidencia con González-Rogado et al. (2014). No obstante, algunos sintieron preocupación y nerviosismo, tal y como sucedió en el estudio de Hanus y Fox (2015), por lo que es importante el uso de avatares para conservar el anonimato de los participantes.

Con respecto a la dimensión social, se observa que más del 95% de los estudiantes están de acuerdo en que el trabajo colaborativo favoreció el logro de los retos, ya que el intercambio de opiniones con los compañeros de equipo facilitó la comprensión del reto, favoreció el intercambio de ideas y la

Intervención con gamificación, educación personalizada y *B-Learning*

completitud de conocimiento, este hallazgo coincide con lo encontrado en el estudio de Nisbet y Williams (2009) así como en el de Domínguez et al. (2013). Sin duda, la dimensión social favorece el desarrollo de actitudes y habilidades que forman parte de la educación integral del estudiante.

Finalmente, para la iteración 2, los resultados de la lista de cotejo revelaron que los estudiantes del grupo control tuvieron un mejor desempeño en el desarrollo de los productos de software que los estudiantes del grupo experimental. Esto pudo deberse a diversas causas, por ejemplo, los estudiantes del grupo control pertenecen a una especialidad relacionada directamente con la programación, por lo que esto pudo condicionar de manera positiva su desempeño. También puede ser que las actividades del grupo control estuvieron más perfiladas a la evaluación de la lista de cotejo. En general, se observó entusiasmo en hacer la evaluación del proyecto final a través de la actividad de gamificación, particularmente en los primeros tres desafíos, no obstante, con el paso de las sesiones algunos fueron perdiendo el interés debido a los resultados que se lograban como equipo, en este sentido vale la pena reflexionar en el diseño de las actividades y su dosificación.

Respecto a las preguntas abiertas, se tiene el siguiente análisis para cada una de las dimensiones.

Preguntas abiertas-dimensión cognitiva.

Haber desarrollado el proyecto final a través de los 5 desafíos, ¿Qué te permitió aprender de nuevo?

Siete estudiantes respondieron aprendizaje nuevo en el trabajo de equipo, cuatro en un tema específico (2-Herencia, 1- Polimorfismo y 1-Manejo de excepciones), destacan 10 que indicaron un repaso de los temas del curso y uno en general de Programación; llama la atención el estudiante que respondió Paciencia como el aprendizaje nuevo que tuvo.

¿Haber desarrollado los desafíos como proyecto final, te permitieron aplicar los temas del curso? ¿Cuáles temas aplicaste?

Los estudiantes respondieron con más de un tema, solo se comenta la ocurrencia decreciente del primero mencionado. 12-Herencia, 6-Polimorfismo, 2-Abstracción-Clases, 1-Encapsulamiento.

Preguntas abiertas-dimensión emotiva.

¿Te motivaron los desafíos? Describe por qué.

Todos respondieron que sí les motivaron los desafíos con las siguientes descripciones: trece porque fue posible resolverlos, cuatro porque sintieron una competencia sana y auto superación, tres porque obtuvieron buenos resultados en equipo y tres porque pudieron obtener las insignias.

Explica por favor, ¿Qué tipo de sentimientos o emociones te causaron ver tus avances en el medallero?

Algunos estudiantes respondieron con más de 2 emociones, se presentan las respuestas de las primeras dos mencionadas en el formato número de estudiantes-emoción. Primera emoción positiva: 12-Alegría, 2-sentirse bien, 2-satisfacción, 1-motivación; Primera emoción negativa: 2-estrés, 1-impotencia-presión-preocupación-frustración. Segunda emoción positiva: 3-entusiasmo, 2-satisfacción, 2-superación, 2-orgullo, 1-confianza-tranquilidad-competencia; Segunda emoción negativa: 3-preocupación, 1-desconfianza, 1-estrés, 6-no indicaron otra emoción.

Preguntas abiertas-dimensión social.

¿De qué manera trabajar con otros compañeros, favoreció o no, la resolución de los desafíos?

Cuatro estudiantes indicaron que no favoreció (3-poca ayuda de sus compañeros, 1-desorganización), mientras 19 respondieron que sí favoreció de las siguientes maneras: 14-cooperación, 4-conocer mejor a los compañeros, 1-no describe.

¿Qué te pareció compartir los resultados de tu proyecto con los otros equipos?

Intervención con gamificación, educación personalizada y *B-Learning*

Dos estudiantes indicaron que no les gusto debido a que sintieron incomodidad, mientras que a 21 les pareció agradable que otros vieran los resultados.

Pregunta abierta para mejorar estrategia.

¿Qué se puede mejorar de la actividad para las siguientes generaciones?

Trece estudiantes respondieron que no deben hacerse cambios a la actividad de gamificación, 3 pidieron agregar pistas durante los desafíos, 2 proponen más tiempo entre desafíos y 1-que la actividad fuera grupal-mostrar el medallero al final-la información de la actividad fuera desde el inicio-diferente asignación de equipos-mayor compromiso de los estudiantes.

5.3 Conclusiones

La intervención presentada en este capítulo, con sus dos iteraciones, fundamentada en la combinación de las estrategias de gamificación, educación personalizada y *b-Learning*, genera los siguientes resultados concretos a la investigación de la tesis doctoral.

1. Impacta directamente en el objetivo general. Se diseñaron escenarios de aprendizaje personalizados para la asignatura Programación, en ellos se ofreció opciones de evidencias evaluativas, ritmo de aprendizaje y tiempo de entrega de productos determinados por los estudiantes, opciones en los formatos de los contenidos y diferentes modalidades de aprendizaje (presencial, semi-presencial y en línea) para ofrecer una educación personalizada, haciendo uso de un entorno *b-Learning* y gamificación para los estudiantes que trabajaron en la modalidad semi-presencial; en consecuencia, permitieron obtener un porcentaje de deserción menor a los registrados en los últimos 8 años, así como un porcentaje de acreditación mayor en el curso de Programación en el mismo periodo, respecto al promedio de la calificación nuevamente no se registró un incremento significativo. Hay que destacar el caso de los estudiantes que decidieron no usar alguna de las opciones ofrecidas, indicando que, a pesar de tal decisión, haber respetado su elección es un indicador favorable

como docente, se fue consciente de que no todos los estudiantes aprenden de la misma forma y haber escuchado cuál era su entorno de aprendizaje fue la mejor decisión para concluir que todo el grupo recibió educación personalizada. La gamificación es una estrategia didáctica que favorece la motivación de los estudiantes, sin embargo, es muy importante tomar cuidado en el diseño de los retos, que sean apropiados y de acuerdo con la habilidad de los estudiantes, de lo contrario, los resultados pueden ser opuestos a lo planeado.

2. Respecto a las competencias básicas de Programación que adquirieron los estudiantes, cada uno de los desafíos fue diseñado para un concepto en particular, pero desde el punto de vista del estudiante a partir de la encuesta final, destaca el concepto y práctica de herencia, posteriormente el de polimorfismo, con menor familiaridad el de POO, arreglos y manejo de excepciones, finalmente el de encapsulamiento, métodos y abstracción. También indican los estudiantes del grupo experimental la habilidad del trabajo en equipo. Una posible razón de que los resultados de evaluación fueran mejor en el grupo de control en la iteración 2, puede ser a que el grupo pertenece a la especialidad de sistemas informáticos y en el cuatrimestre anterior habían tenido experiencia con el entorno de programación, el grupo experimental no tenía esa experiencia, pero a pesar de ello, la mayoría de los estudiantes lograron obtener buenos resultados.
3. Uno de los inconvenientes de la realización en la iteración 1 fue que no hubo una revisión de la lectura asignada en el modelo de aprendizaje semi-presencial, así varios de los estudiantes asistieron a los retos sin haber leído, lo cual representó un obstáculo para resolverlos. Por lo tanto, aunque esta situación está más relacionada con el diseño del curso que con la gamificación en la intervención, es un aspecto que debe ser considerado. Si una estrategia didáctica no genera los resultados deseados la primera vez que es aplicada, no debe ser una razón para desalentarse, en su lugar se tiene que tomar la oportunidad de poder reflexionar en el diseño de las actividades bajo una perspectiva de mejora continua. Lo anterior benefició la segunda iteración. En consecuencia, en la encuesta final se confirma la gamificación como

Intervención con gamificación, educación personalizada y *B-Learning*

estrategia de motivación debido a que los estudiantes reportan emociones de alegría y orgullo con el uso del medallero, aunque, por otro lado, algunos estudiantes pueden sentir incomodidad que su ejecución sea expuesta en el medallero y puede afectar su motivación, por lo tanto, el uso de avatares es recomendado para proteger el anonimato.

Para las futuras intervenciones, se proponen con certeza las siguientes acciones:

- i. El diseño de la estrategia de gamificación ofreció la motivación necesaria a los estudiantes, cada grupo es diferente y debe determinarse la aplicación de la intervención al final o a lo largo del cuatrimestre a consulta del grupo.
- ii. Para asegurar la ejecución completa de los retos, debe previamente reforzarse cada sesión con ejercicios de programación con el propósito de asegurar que el estudiante está preparado para terminarlos, para ello se propone elaborar foros en la plataforma *Moodle* que permitan un trabajo colaborativo además de estudio teórico previo, también se pueden diseñar retos similares de tal manera que los estudiantes puedan tener varios intentos y que con ello puedan aprender de sus errores.
- iii. Las insignias pueden diversificarse dentro del mismo curso hacia otras habilidades cognitivas, actitudinales y de valores. También vale la pena que, para evaluar la dimensión cognitiva, se incluyan instrumentos que evalúen competencias como la de resolución de problemas, de tal manera que la evaluación cognitiva no quede solamente en manos de un instrumento tradicional.
- iv. La creatividad del profesor es muy importante para ser capaz de crear retos que sean de interés al estudiante, es decir, debe comprender cuáles son las dinámicas que sus estudiantes juegan.

BLOQUE III: DISCUSIÓN Y CONCLUSIONES

El tercer bloque de la tesis doctoral en sus dos capítulos que le integran, expone la actividad de discusión a partir de verificar el cumplimiento de los objetivos planteados, además de responder las preguntas del trabajo de investigación. En consecuencia, se presenta finalmente el producto conclusivo con base en toda la información y evidencia recabada por el seguimiento del modelo mixto de triangulación concurrente. Lo anterior permite establecer las aportaciones a la Sociedad del Conocimiento y particularmente a los investigadores afines con la búsqueda de estrategias educativas en la educación superior del área de ciencias de la computación en todas sus variaciones. Se presenta el trabajo a futuro de la tesis doctoral y los productos científicos publicados que permiten validar el trabajo realizado.

CAPÍTULO 6. DISCUSIÓN

6.1 Introducción

Dos principales problemas ocuparon el trabajo de tesis doctoral: determinar estrategias para reducir la deserción escolar y proporcionar competencias básicas para la programación de computadoras.

Ante tales problemas se propusieron dos acciones para solucionarlos:

- Crear intervenciones donde sea posible ofrecer aprendizaje personalizado usando una modalidad *b-Learning*, para los estudiantes de nuevo ingreso del curso Metodología de la programación, usar la evaluación del pensamiento computacional para determinar escenarios iniciales, para los estudiantes del curso de Programación, usar la gamificación como estrategia de motivación. Así para ambas intervenciones, permitir la adquisición de competencias de programación y reducir el abandono de estudios.
- Evaluar el impacto que tuvieron las intervenciones realizadas en el proceso enseñanza – aprendizaje de los cursos Metodología de la programación y Programación.

Otras acciones realizadas que contribuyeron al cumplimiento de los objetivos fueron:

- Modificar el contenido y la organización del actual sistema gestor de aprendizaje usado para las intervenciones.
- Establecer el estado del arte de los conceptos teóricos en el contexto de la educación superior y la enseñanza de programación.
- Verificar el correcto diseño de la intervención con educación personalizada, pensamiento computacional y *b-Learning* en un contexto educativo diferente al caso de estudio, es decir, la elaboración de un diseño experimental en INACAP - Chile.

El presente capítulo, discute el alcance logrado de cada uno de los objetivos enmarcados en la definición del problema.

6.2 Discusión

Al menos en tres ocasiones, en la división de tecnologías de la información y comunicación se ha instalado un servidor *Moodle* para tener una herramienta educativa, a partir del trabajo de investigación propuesto, la plataforma ha tenido un uso continuo por parte de los estudiantes que han participado en las intervenciones. El contenido actual para los cursos de Metodología de la programación y Programación ofrece variedad de contenidos y espacios para evidenciar la adquisición de competencias de los participantes. En comparación a los contenidos anteriores y la frecuencia de uso, como se comentó en el capítulo cuatro, la importancia de la plataforma *Moodle* quedó expuesta, el modelo *b-Learning* creado permitió la continuación del aprendizaje a pesar de no haber actividad presencial, aquellos que descargaron el material y lo difundieron con sus compañeros lo favorecieron. En una discusión más detallada, la eficiencia individual estuvo determinada por la oferta de los escenarios de aprendizaje en cada estudiante, es decir, para la mayoría de los jóvenes fue la primera vez que usaron una plataforma de ese tipo, lo que implicó agregar una curva de aprendizaje de *Moodle*, pero lo que fue más importante, balancear lo proporcionado por la plataforma y las actividades presenciales. La combinación del ambiente híbrido con la educación personalizada resulto un reto difícil de asumir para muchos; tomar conciencia que el proceso de aprendizaje debía ser controlado por el mismo estudiante. Más de uno llegó a exclamar en las encuestas o de forma presencial que *quería su clase tradicional*. Las estrategias no dispararon la deserción por falta de adaptación del estudiante, pero si comenzó a crear en algunos jóvenes su capacidad de autonomía. ¿Es eficiente el ambiente de aprendizaje *b-learning*? Sí. En un modelo educativo donde 70% de la actividad debe ser práctica y 30% teórica, el *b-learning* ofrece a los estudiantes que la definición, la identificación y la aplicación de conceptos teóricos se lleve a cabo a su propio ritmo de aprendizaje a través de una variedad de contenidos (lecturas, audio, videos) seleccionados y organizados a partir de la guía del docente; flexibilidad en las evidencias evaluativas y reconocimiento de habilidades adquiridas, en este sentido la propuesta del ambiente híbrido se enriqueció con la evaluación del pensamiento computacional.

Discusión

La coherencia entre las habilidades evaluadas y los temas del curso Metodología de la programación orientan al estudiante cómo iniciar su proceso de adquisición de competencias relativas a la creación de *software*. Se reconoce una habilidad, se reafirma su comprensión y aplicación y se acredita la competencia, o se detecta una carencia, se ofertan escenarios de aprendizaje y se flexibiliza la evaluación para evidenciar la acreditación de la competencia; en ambos casos, se propone un modelo *b-Learning* y personalización dando paso al aprendizaje centrado en el estudiante.

No se obtuvieron resultados casuales o espontáneos, son reflejo de aportaciones encontradas en la revisión de investigaciones realizadas previamente por una variedad de investigadores, pero que permitieron crear una nueva propuesta para el contexto de universidades tecnológicas en México, particularmente en la ubicada en la ciudad de Puebla. En este punto se destaca la importancia de la tesis doctoral por su usabilidad en otro contexto de educación superior: INACAP – Chile. En trabajo colaborativo con el área de informática y telecomunicaciones, INACAP online y la dirección de I+D fue posible hacer un diseño experimental ajustado al curso Fundamentos de programación con el objetivo de aumentar el porcentaje de retención y proporcionar las competencias respectivas de la asignatura, es decir, metas similares de la tesis doctoral.

La gamificación fue un elemento importante para la intervención dos y que enriqueció la tesis doctoral. La gamificación fue una estrategia didáctica que favoreció la motivación de los estudiantes, no se obtuvieron los resultados deseados en su primera iteración, pero fue una razón para reflexionar el diseño de la estrategia y proponer una variación, así de la segunda ocasión de usarla en una intervención, se confirman las observaciones del trabajo en equipo y ofreció nuevamente resultados favorables, principalmente en la dimensión emotiva y social.

En resumen, la experiencia didáctica de gamificación fue gratificante debido a que permitió observar áreas de oportunidad tanto en el diseño mismo de la gamificación y en actividades del curso, las cuales conducen a un proceso de mejora continua que al mismo tiempo permite validar si la estrategia didáctica supera lo que ya fue hecho, es decir, si realmente es una innovación.

6.3 Conclusiones

La innovación educativa tiene beneficios para el estudiante y para el docente, pero también genera resistencia al cambio; el estudiante puede elegir, pero debe ser más autónomo, el docente ofrece opciones de contenido y aprendizaje, pero tiene que elaborarlos, el estudiante quiere su clase tradicional y el docente debe escucharlo. Escuchar al estudiante es la clave inicial para ofrecer educación personalizada, una modalidad *b-Learning* y gamificación. Realizar la evaluación del pensamiento computacional sin realizar un diálogo a partir de los resultados obtenidos para ofrecer los escenarios de aprendizaje no solo sería una falsa innovación, sería nuevamente centrar el aprendizaje en el docente que no busca que los estudiantes aprendan algo nuevo a través de la apropiación del conocimiento, sino solo una réplica del conocimiento del docente, con sus virtudes, pero desafortunadamente con sus defectos, “sigue este camino y aprenderás como yo”. Trabajar con una modalidad híbrida sin abonar aquellas competencias digitales para que el estudiante aproveche el espacio virtual de aprendizaje o no reforzar la ejercitación en línea de los conceptos teóricos, no influirá positivamente cuando las actividades presenciales se lleven a cabo. Continuar con métodos de evidencia evaluativa impresa no solo afecta la coherencia de perfil profesional, desestima los esfuerzos por centralizar los productos académicos en la plataforma. Finalmente, el uso de gamificación permite expresar la creatividad docente, su uso o no aplicación en un curso está condicionada a las características de cada grupo, nuevamente a la atenta escucha de la aceptación a las estrategias, pero lo que es inevitable obtener, es la motivación por recrear un escenario de competitividad que ofrece satisfacción por superar un reto dentro de un trabajo colaborativo.

Con las estrategias aplicadas en la tesis doctoral, lo que se propone son elementos que permitan una mejora continua, una innovación sustentable que se pueda realizar y mantener usando el tiempo de docencia determinado por el centro de trabajo, evitar llevar trabajo a casa y en consecuencia desgastar la vocación docente.

CAPÍTULO 7. CONCLUSIONES

7.1 Introducción

Este capítulo tiene el objetivo de presentar las conclusiones de la respuesta a la pregunta de investigación, además de exponer las contribuciones obtenidas principalmente en el bloque empírico de la tesis doctoral.

¿Qué competencias básicas de Programación obtienen los estudiantes de nuevo ingreso al participar en un ambiente de aprendizaje donde se apliquen las estrategias educativas de educación personalizada, pensamiento computacional, gamificación y *b-Learning*, y contribuyan a la reducción del abandono escolar?

Siete son las conclusiones que se listan a continuación, además de responder a la pregunta, fortalecen la hipótesis.

7.2 Conclusiones

Conclusión 1

Las competencias básicas que obtienen los estudiantes de nuevo ingreso en un modelo *b-Learning*, integrando escenarios de aprendizaje con una orientación de educación personalizada a través de la evaluación del pensamiento computacional son:

Para el curso Metodología de la programación:

Diferenciación de tipos de datos, identificación y uso de operadores aritméticos, lógicos y relacionales en expresiones, jerarquía de operadores, creación de identificadores válidos, uso de variables contador y acumulador, estructuras de repetición y definición y creación de algoritmos.

Para el curso Programación:

Polimorfismo, programación orientada a objetos, arreglos, manejo de excepciones, encapsulamiento, métodos, abstracción y trabajo en equipo.

Conclusión 2

El modelo *b-Learning* es pertinente para el sistema de universidades tecnológicas, 70% práctico – 30% teórico, el uso de la plataforma *Moodle* debe ser balanceado con las expectativas del trabajo presencial, rico en contenidos y actividades de retroalimentación, flexible en las evidencias de evaluación, moldeable a las habilidades de cada estudiante y adaptativo a su crecimiento cognitivo.

Conclusión 3

El pensamiento computacional a través de las habilidades de abstracción, generalización, diseño algorítmico, descomposición y evaluación, tienen una relación estrecha con los contenidos temáticos del curso Metodología de la programación, por lo cual su evaluación en los estudiantes de nuevo ingreso permite reconocer habilidades y detectar carencias remediabiles para lograr la adquisición de competencias básicas en el desarrollo de *software*. La relación comentada fue verificada también en INACAP – Chile para los contenidos temáticos de la asignatura Fundamentos de programación.

Conclusión 4

La educación personalizada ofrece el control al estudiante, haciéndolo gestor y partícipe activo de su propio aprendizaje, el cambio de paradigma centrado en el docente hacia el estudiante conlleva la oferta de los mejores métodos y recursos académicos para todos los estudiantes, sobre todo para aquellos que sean menos autónomos.

Conclusión 5

La gamificación es una estrategia motivadora para el estudiante y el docente, si es diseñada apropiadamente para el grupo experimental, considerando el cumplimiento cognitivo de los participantes para realizar los retos y determinando los componentes adecuados para el

Conclusiones

reconocimiento de los logros, el trabajo colaborativo generará resultados emocionales y sociales muy altos en los estudiantes.

Conclusión 6

La evaluación del pensamiento computacional a los estudiantes de nuevo ingreso, particularmente las habilidades de generalización, descomposición, abstracción, diseño algorítmico y evaluación, y con base al resultado de la evaluación del pensamiento computacional ofertar escenarios de aprendizaje personalizados, así como el uso de la plataforma *Moodle* como herramienta para un entorno *b-Learning* para el curso Metodología de la programación; la combinación de las estrategias de gamificación, educación personalizada y *b-Learning* para el curso de Programación generó en la Universidad Tecnológica de Puebla en la división de tecnologías de la información y comunicación, un porcentaje de deserción menor a los registrados en los últimos 8 años. En consecuencia, se redujo el abandono de estudios a partir de incrementar el porcentaje de acreditación en el curso. Respecto al promedio de la calificación, no se registró un incremento significativo, pero es una evidencia de que se ha mantenido la calidad evaluativa.

Conclusión 7

El diseño experimental de evaluar el pensamiento computacional a los estudiantes de nuevo ingreso, particularmente las habilidades de generalización, descomposición, abstracción, diseño algorítmico y evaluación, y con base al resultado de la evaluación del pensamiento computacional ofertar escenarios de aprendizaje personalizados, también es posible en INACAP- Chile, considerando además que usan la plataforma *Moodle* como herramienta para un entorno *b-Learning* para el curso Fundamentos de programación.

7.3 Contribuciones y futuras líneas de investigación

Se contribuyó a la reducción de la deserción escolar proporcionando las competencias de programación de computadoras sin reducir la calidad educativa. Se contribuyó estableciendo una

relación de las habilidades del pensamiento computacional con temas específicos del curso Metodología de la programación para el caso de estudio en México, así como para la asignatura Fundamentos de programación en INACAP – Chile, lo que confirma la similitud de contenidos que se pueden ofrecer a nivel global, en consecuencia, se pueden usar los resultados comentados para su adecuación según lo considere algún docente interesado en usar el trabajo empírico. Se contribuyó con un trabajo de investigación relacionado con la evaluación del pensamiento computacional para ofrecer escenarios de aprendizaje, educación personalizada que fue novedosa para los estudiantes que experimentan compensación y reconocimiento, así como la guía para comenzar la adquisición de competencias pertinentes al desarrollo de *software*. Se contribuyó con una investigación promotora de la autonomía del estudiante en su proceso educativo, dando el control a través de la oferta de diferentes recursos, flexibilidad de evaluación y valoración de productos académicos organizados en tiempo y ritmo por los propios estudiantes. Finalmente, se contribuyó con una estrategia didáctica para la motivación del estudiante por medio de la gamificación, también para la adquisición de competencias relacionadas con la Programación.

Si bien la pregunta de investigación ha sido resuelta confirmando la hipótesis planteada, a pesar de tener un trabajo de investigación finalizado dentro del alcance establecido en la tesis doctoral, existen elementos que deben considerarse y enfocarse en el trabajo a futuro.

Los actores principales del proceso enseñanza-aprendizaje son el estudiante y el docente, así se llevaron a cabo las intervenciones, pero también interviene el área directiva de la universidad que no solo permitió la ejecución de los experimentos educativos, sino que puede evaluar y valorar los resultados obtenidos, principalmente por los estudiantes, y ofrecer acciones de proyección educativa para ajustar el plan académico de la división de TIC, con el objetivo de obtener resultados similares de motivación, bajo porcentaje de deserción y adquisición de competencias en otros cursos, al menos usando la educación personalizada, el modelo *b-Learning* y la gamificación, en consecuencia, a futuro se debe realizar trabajo de divulgación y discusión con compañeros docentes y directivos.

Conclusiones

El tercer cuatrimestre de estudios tiene una acción integradora de los cursos pasados mientras se adquieren nuevas competencias, la complejidad del proceso educativo aumenta y es una oportunidad para trabajar con las estrategias de la tesis doctoral o bien modificar, ajustar o hallar nuevas estrategias que sean oportunas al contexto integrador, en cualquier caso, su documentación y evaluación también es un trabajo a futuro valioso por el impacto directo en la experiencia del docente y su mejora continua.

Desafortunadamente, la experiencia de los participantes en los grupos experimentales queda limitada, truncada y frenada en los cursos posteriores, la capacidad y nivel de conciencia logrado de autonomía se ve eclipsado por el retorno a las acciones tradicionales de los compañeros docentes, lo que se ha observado hasta el momento, es la actitud crítica de los estudiantes frente a los escenarios propuestos, por lo cual se deberá investigar a futuro el porcentaje de estudios terminales de los participantes de las intervenciones realizadas, que permita tener un métrica del impacto logrado en la conclusión de la carrera.

En comparación con las investigaciones encontradas y revisadas durante el ejercicio del marco teórico, es importante destacar que se debe actualizar e incrementar las acciones realizadas para poder obtener información no detectada debido a la inexperiencia del investigador.

La ejecución del diseño experimental creado para INACAP – Chile puede ser realizado para el semestre inicial de 2020, pero requiere de un seguimiento del área de informática y telecomunicaciones y de INACAP online. El interés existe y la disposición del investigador también para obtener datos de la intervención, en consecuencia, tener continuidad de las fases en la metodología de investigación mixto y llegar a nuevos productos de la tesis doctoral.

Finalmente, se debe mejorar la manipulación de la información a través del estudio o trabajo colaborativo con un especialista en técnicas estadísticas para la presentación y operación de datos cuantitativos que impacte en la difusión futura de resultados de investigación.

7.4 Publicaciones

En el último apartado del capítulo, se indican por tipo las 10 publicaciones relacionadas con la investigación desarrollada en la tesis doctoral. En resumen, se han publicado 4 artículos en revistas indexadas, 5 trabajos en actas de conferencias internacionales y 1 capítulo de libro; cada uno de ellos fortaleció y validó el contenido de la tesis doctoral a partir de la retroalimentación proporcionada por los respectivos revisores durante el proceso de publicación. La Figura 32 ilustra la producción realizada por año y tipo.

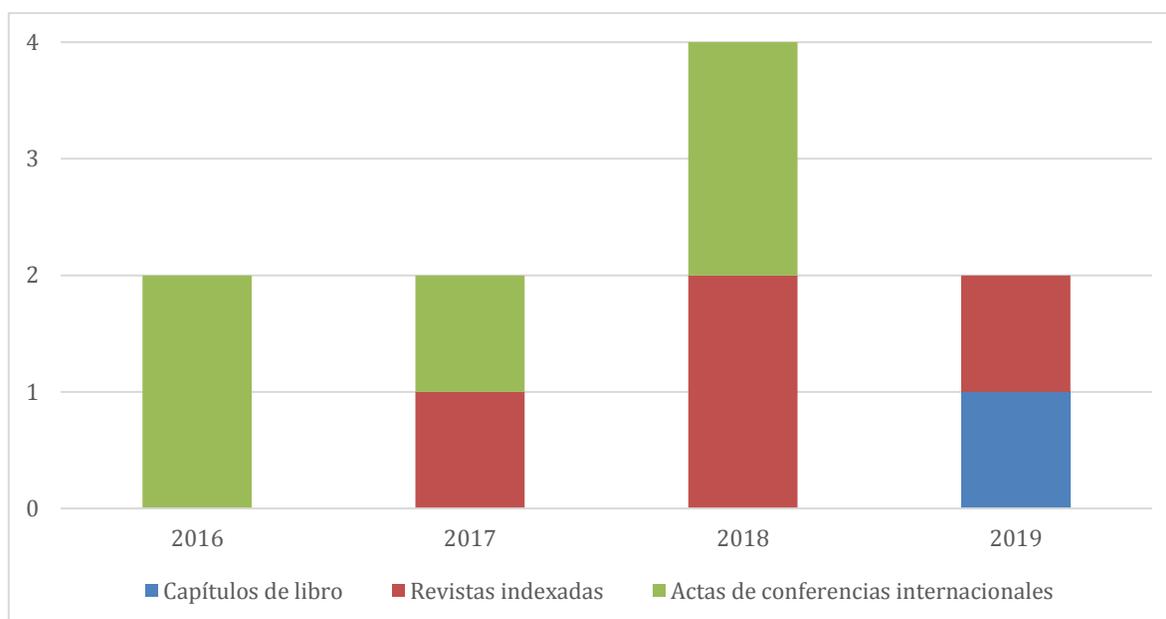


Figura 32. Producción científica realizada durante el periodo de la tesis doctoral. Fuente: Elaboración propia

- Actas de conferencias internacionales
1. Rojas-López A. and García-Peñalvo F.J. (2016) Relationship of knowledge to learn in programming methodology and evaluation of computational thinking, in Proceedings of the Fourth International Conference on Technological Ecosystems for Enhancing Multiculturality (TEEM'16) (Salamanca, Spain, November 2-4, 2016), F. J. García-Peñalvo, Ed. ICPS: ACM International Conference Proceeding Series, New York, NY, USA: ACM, pp. 73-77. doi:10.1145/3012430.3012499

Conclusiones

2. Rojas-López A. and García-Peñalvo F.J. (2016) Personalized contents based on cognitive level of student's computational thinking for learning basic competencies of programming using an environment b-learning, in Proceedings of the Fourth International Conference on Technological Ecosystems for Enhancing Multiculturality (TEEM'16) (Salamanca, Spain, November 2-4, 2016), F. J. García-Peñalvo, Ed. ICPS: ACM International Conference Proceeding Series, New York, NY, USA: ACM, 2016, pp. 1139-1145. doi: 10.1145/3012430.3012660
 3. Rojas-López A. and García-Peñalvo F.J. (2017) Personalized education using computational thinking and b-learning environment: classroom intervention, in Proceedings of the Fifth International Conference on Technological Ecosystems for Enhancing Multiculturality (TEEM'17) (Cádiz, Spain, October 18-20, 2017), F. J. García-Peñalvo, Ed. ICPS: ACM International Conference Proceeding Series, New York, NY, USA: ACM, 2017, article 9. doi:10.1145/3144826.3145357
 4. Rojas-López A. and García-Peñalvo F.J. (2018) in Proceedings of the Sixth International Conference on Technological Ecosystems for Enhancing Multiculturality (TEEM'18) (Salamanca, Spain, October 24 - 26, 2018), F. J. García-Peñalvo, Ed. ICPS: ACM International Conference Proceeding Series, New York, NY, USA: ACM, 2018, pp. 31-35. doi: 10.1145/3284179.3284187
 5. Rojas-López A. and Rincón-Flores E.G. (2018) Gamification as Learning Scenario in Programming Course of Higher Education. In: Zaphiris P., Ioannou A. (eds) Learning and Collaboration Technologies. Learning and Teaching. LCT 2018. Lecture Notes in Computer Science, vol 10925. Springer, Cham. doi:10.1007/978-3-319-91152-6_16
- Revistas indexadas
1. Rojas-López, A. (2017) Intervención de tres estrategias educativas para cursos de programación en educación superior. Education In The Knowledge Society (EKS), 18(4), 21-34. doi:10.14201/eks20171842134

2. Rojas-López A. and García-Peñalvo F.J. (2018) Learning Scenarios for the Subject Methodology of Programming from Evaluating the Computational Thinking of New Students, in *IEEE Revista Iberoamericana de Tecnologías del Aprendizaje* (Volume: 13, Issue: 1, Feb. 2018), pp. 30 – 36. Date of Publication: 01 March 2018. Electronic ISSN: 1932-8540. doi: 10.1109/RITA.2018.2809941. Publisher: IEEE
 3. Rojas-López, A. and García-Peñalvo, F. J. (2018) 4 elements proposed to offer personalized education in the university superior technical level. *Advanced Science Letters*. 24(11), 7980-7982. doi: 10.1166/asl.2018.12471.
 4. Rojas-López, A. Rincón-Flores, E.G., Mena, J., García-Peñalvo, F.J. and Ramírez-Montoya, M.S. (2019) Engagement in the course of Programming in Higher Education through the use of Gamification. *Universal Access in the Information Society (UAIS)*. Volume: 18, Issue: 3, pp. 583-597. doi: 10.1007/s10209-019-00680-z.
- Capítulo de libro
1. Rojas-López A. and García-Peñalvo F.J. (2019) Personalized Education for a Programming Course in Higher Education, en *Innovative Trends in Flipped Teaching and Adaptive Learning*, M. L. Sein-Echaluce, Á. Fidalgo-Blanco y F. J. García-Peñalvo, Eds. *Advances in Educational Technologies and Instructional Design (AETID)*, pp. 203-227, Hershey, PA, USA: IGI Global, 2019. doi: 10.4018/978-1-5225-8142-0.ch0010.

Conclusiones

APÉNDICES

APÉNDICE A. Indicadores por grupo del 2009 al 2016 para el curso Metodología de la programación.

Grupo	Cuatrimestre septiembre – diciembre de 2009		
	Indicador		
	PA	C	PD
A	76,92	6,66	23,08
B	86,21	7,58	17,24
C	89,29	7,45	21,43
D	72,00	6,75	28,00
E	60,71	5,74	39,29
F	75,86	5,89	24,14
G	96,30	8,39	11,11
H	75,00	6,35	25,00
I	72,73	6,64	27,27
J	95,65	7,77	4,35
K	92,00	7,40	8,00
L	85,19	7,51	14,81
M	63,64	6,59	36,36
N	80,65	7,20	22,58
O	92,59	7,89	11,11
P	55,56	5,64	44,44
Q	85,19	7,02	14,81
R	62,50	6,05	41,67
S	66,67	5,76	33,33
T	61,54	5,47	38,46

Grupo	Cuatrimestre enero – abril de 2010		
	Indicador		
	PA	C	PD
A	72,41	5,95	27,59

	Cuatrimestre septiembre – diciembre de 2010		
	Indicador		
Grupo	PA	C	PD
A	81,25	7,78	18,75
B	80,65	6,75	51,61
C	75,68	7,43	24,32
D	71,43	7,30	28,57
E	46,15	5,00	53,85
F	61,76	6,19	38,24
G	69,23	5,97	30,77
H	76,67	7,04	23,33
I	90,63	6,98	9,38
J	93,33	8,05	6,67
K	66,67	6,02	33,33
L	76,47	6,18	23,53
M	66,67	6,38	33,33
N	70,00	6,89	30,00
O	88,24	6,89	11,76

	Cuatrimestre enero – abril de 2011		
	Indicador		
Grupo	PA	C	PD
A	58,33	5,75	41,67
B	50,00	4,62	50,00

	Cuatrimestre septiembre – diciembre de 2011		
	Indicador		
Grupo	PA	C	PD
A	89,66	8,21	13,79
B	79,41	7,16	20,59
C	55,88	6,65	44,12
D	72,22	6,86	38,89
E	58,62	6,28	44,83
F	70,97	6,45	29,03
G	74,36	6,65	25,64
H	58,06	5,00	48,39
I	82,35	8,53	17,65
J	77,14	6,69	22,86
K	83,87	7,85	16,13
L	64,29	6,07	35,71
M	82,35	6,88	17,65

Indicadores por grupo del 2009 al 2016 para el curso Metodología de la programación

Grupo	Cuatrimestre enero – abril de 2012		
	Indicador		
	PA	C	PD
A	59,38	5,13	68,75
B	63,89	7,03	36,11

Grupo	Cuatrimestre septiembre – diciembre de 2012		
	Indicador		
	PA	C	PD
A	74,29	6,60	25,71
B	76,32	6,79	23,68
C	66,67	6,21	33,33
D	72,97	7,38	27,03
E	71,88	7,28	34,38
F	65,63	6,91	34,38
G	90,00	7,57	10,00
H	66,67	6,10	33,33
I	71,88	6,00	28,13
J	79,17	7,67	20,83
K	70,83	6,33	29,17

Grupo	Cuatrimestre enero – abril de 2013		
	Indicador		
	PA	C	PD
A	48,94	5,98	51,06

	Cuatrimestre septiembre – diciembre de 2013		
	Indicador		
Grupo	PA	C	PD
A	70,27	6,60	29,73
B	82,35	7,51	20,59
C	64,71	6,59	35,29
D	59,38	5,72	43,75
E	63,64	5,89	36,36
F	69,44	6,69	30,56
G	50,00	4,80	50,00
H	48,48	5,05	51,52
I	87,10	7,43	12,90
J	71,43	6,29	28,57
K	69,70	6,47	30,30
L	50,00	4,82	50,00
M	77,78	6,79	22,22

	Cuatrimestre enero – abril de 2014		
	Indicador		
Grupo	PA	C	PD
A	27,78	5,18	75,00
B	61,90	7,51	38,10

	Cuatrimestre septiembre – diciembre de 2014		
	Indicador		
Grupo	PA	C	PD
A	56,25	6,14	43,75
B	65,63	6,70	34,38
C	44,83	5,72	55,17
D	69,70	7,00	39,39
E	72,73	7,12	27,27
F	48,48	5,76	51,52
G	57,58	6,47	42,42
H	58,82	6,12	41,18
I	60,00	6,20	42,86
J	61,76	5,84	38,24
K	50,00	5,19	50,00
L	87,88	7,47	12,12

Indicadores por grupo del 2009 al 2016 para el curso Metodología de la programación

	Cuatrimestre enero – abril de 2015		
	Indicador		
Grupo	PA	C	PD
A	89,74	7,95	15,38

	Cuatrimestre septiembre – diciembre de 2015		
	Indicador		
Grupo	PA	C	PD
A	70,00	7,08	30,00
B	70,97	6,76	32,26
C	83,33	7,60	16,67
D	46,67	5,77	53,33
E	70,00	7,12	30,00
F	34,48	4,33	65,52
G	46,67	5,45	53,33
H	66,67	6,33	33,33
I	91,18	8,28	8,82
J	89,29	7,86	14,29
K	84,38	7,66	18,75
L	66,67	6,05	33,33
M	56,25	5,58	43,75
N	90,63	7,78	9,38
O	76,67	7,50	26,67

	Cuatrimestre septiembre – diciembre de 2016		
	Indicador		
Grupo	PA	C	PD
A	76,47	6,78	26,47
B	75,76	7,73	27,27
C	81,82	7,12	24,24
D	72,73	6,82	27,27
E	36,36	4,52	63,64
F	39,39	4,88	60,61
G	78,13	7,39	25,00
H	81,25	7,42	21,88
I	65,71	7,04	34,29
J	88,57	7,43	11,43
K	51,61	5,44	51,61
L	89,29	8,91	17,86

APÉNDICE B. Indicadores por año del 2009 al 2016 para el curso Metodología de la programación.

Año	Metodología de la Programación		
	Indicador		
	PA	C	PD
2009	77,31	6,79	24,32
2010	74,20	6,68	27,81
2011	70,50	6,64	31,13
2012	71,50	6,69	31,14
2013	65,23	6,19	35,20
2014	58,81	6,32	42,24
2015	70,85	6,82	30,30
2016	68,25	6,75	34,00
Mediana	70,67	6,68	31,13

APÉNDICE C. Indicadores por grupo del 2010 al 2017 para el curso Programación.

Grupo	Cuatrimestre enero – abril de 2010		
	Indicador		
	PA	C	PD
A	76,19	6,33	23,81
B	91,67	7,54	8,33
C	95,00	7,43	5,00
D	94,12	7,66	5,88
E	83,33	6,73	16,67
F	86,36	7,05	13,64
G	92,00	7,44	8,00
H	69,57	5,75	39,13
I	84,21	6,63	15,79
J	75,00	6,81	25,00
K	90,48	7,11	14,29
L	91,30	7,53	13,04
M	90,91	7,86	18,18
N	80,00	6,82	20,00
O	44,44	4,64	55,56
P	81,82	7,09	18,18
Q	93,33	7,41	13,33
R	100,00	7,97	0,00
S	93,33	7,28	6,67
T	90,91	6,98	9,09

Grupo	Cuatrimestre mayo – agosto de 2010		
	Indicador		
	PA	C	PD
A	51,61	4,60	54,84

Grupo	Cuatrimestre enero – abril de 2011		
	Indicador		
	PA	C	PD
A	66,67	6,93	22,22
B	82,14	7,21	17,86
C	88,89	7,35	11,11
D	67,74	6,71	32,26
E	75,00	6,47	25,00
F	95,65	7,26	4,35
G	82,61	6,72	17,39
H	75,00	6,88	25,00
I	80,00	6,70	28,00
J	52,17	5,06	47,83
K	100,00	7,76	0,00
L	95,83	7,28	45,83
M	87,50	7,38	25,00
N	17,65	3,93	82,35
O	79,17	6,54	20,83

Grupo	Cuatrimestre mayo – agosto de 2011		
	Indicador		
	PA	C	PD
A	58,82	6,17	47,06
B	32,00	3,47	72,00
C	50,00	5,19	61,11

Grupo	Cuatrimestre enero – abril de 2012		
	Indicador		
	PA	C	PD
A	68,18	7,36	36,36
B	55,56	7,22	44,44
C	91,30	8,04	8,70
D	91,67	7,75	12,50
E	85,71	7,76	14,29
F	41,67	6,50	66,67

Indicadores por grupo del 2010 al 2017 para el curso Programación

G	63,64	5,14	36,36
H	80,00	6,60	40,00
I	72,73	6,18	27,27
J	88,46	8,00	11,54
K	64,71	7,29	35,29
L	100,00	8,00	0,00
M	70,00	7,40	30,00

Cuatrimestre mayo – agosto de 2012			
Indicador			
Grupo	PA	C	PD
A	52,63	4,42	52,63
B	73,33	6,73	33,33
C	69,57	6,43	30,43

Cuatrimestre enero – abril de 2013			
Indicador			
Grupo	PA	C	PD
A	86,67	7,63	23,33
B	76,67	7,13	23,33
C	78,26	6,96	21,74
D	96,00	8,43	16,00
E	59,38	5,97	43,75
F	63,64	7,05	36,36
G	52,38	6,19	47,62
H	93,33	8,00	6,67
I	52,63	6,26	47,37
J	88,00	8,00	12,00

Cuatrimestre mayo – agosto de 2013			
Indicador			
Grupo	PA	C	PD
A	75,86	7,30	24,14
B	63,64	5,99	36,36

	Cuatrimestre enero – abril de 2014		
	Indicador		
Grupo	PA	C	PD
A	60,00	7,52	43,33
B	96,77	8,24	3,23
C	75,00	7,52	25,00
D	56,25	5,50	43,75
E	92,00	8,16	12,00
F	59,09	7,43	40,91
G	56,52	7,30	43,48
H	94,44	8,14	5,56
I	76,67	7,35	23,33
J	63,16	6,76	36,84

	Cuatrimestre mayo – agosto de 2014		
	Indicador		
Grupo	PA	C	PD
A	83,33	7,94	41,67
B	75,86	6,81	27,59
C	41,94	5,52	61,29

	Cuatrimestre enero – abril de 2015		
	Indicador		
Grupo	PA	C	PD
A	77,27	7,45	22,73
B	53,85	6,62	46,15
C	79,17	7,58	20,83
D	82,35	8,32	17,65
E	78,95	6,76	21,05
F	55,00	4,75	45,00
G	50,00	6,07	50,00
H	78,95	7,00	26,32
I	73,08	6,48	26,92
J	59,09	6,23	40,91
	Cuatrimestre mayo – agosto de 2015		
	Indicador		
Grupo	PA	C	PD
A	87,50	8,17	29,17
B	91,30	8,22	8,70
C	57,69	6,65	46,15

Indicadores por grupo del 2010 al 2017 para el curso Programación

Grupo	Cuatrimestre enero – abril de 2016		
	Indicador		
	PA	C	PD
A	56,00	6,62	44,00
B	73,91	6,96	26,09
C	70,37	7,52	29,63
D	61,11	6,11	38,89
E	80,95	7,71	19,05
F	66,67	6,83	40,00
G	28,57	4,79	71,43
H	77,78	6,47	22,22
I	57,14	6,68	42,86
J	96,00	7,94	8,00
K	76,19	6,95	23,81
L	87,50	8,03	43,75
M	95,00	9,05	5,00
N	74,07	6,91	25,93
O	81,25	7,31	18,75

Grupo	Cuatrimestre enero – abril de 2017		
	Indicador		
	PA	C	PD
A	62,50	7,29	37,50
B	70,83	7,79	33,33
C	92,00	8,06	8,00
D	42,86	5,05	61,90
E	82,35	7,91	17,65
F	77,78	7,69	22,22
G	66,67	6,37	44,44
H	83,33	7,33	16,67
I	56,67	7,52	43,33
J	67,74	5,87	32,26
K	100,00	8,27	0,00
L	54,55	7,00	45,45

APÉNDICE D. Indicadores por año del 2010 al 2017 para el curso Programación.

	Programación		
	Indicador		
Año	PD	PA	C
2010	16,48	85,20	7,0
2011	33,69	70,45	6,29
2012	29,99	73,07	6,92
2013	28,22	73,87	7,07
2014	31,38	71,62	7,24
2015	30,89	71,09	6,94
2016	30,63	72,17	7,05
2017	32,25	69,57	7,09
Mediana	30,76	71,89	7,03

APÉNDICE E. Detalle de información clave por material SCOPUS tema gamificación
SCOPUS_G_01

Definición:

El uso de elementos de juego y sus técnicas de diseño, en un contexto de no juego, es conocido como gamificación.

Actividad de investigación:

Los investigadores presentan la aplicación de una técnica de gamificación usada en la enseñanza de programación para estudiantes de cursos introductorios en la Ingeniería en Desarrollo de Software en la Universidad Autónoma de Baja California Sur (UABCS). Se presentan referencias que apoyan la propuesta y una descripción del contexto de la aplicación fue realizado. Finalmente, los resultados del uso y las conclusiones obtenidas son presentados.

Uso de gamificación en programación en educación superior:

1. Ofrecer recompensas por completar tareas
2. Tomar ventaja de la competitividad, haciendo visibles las recompensas entre los jugadores
3. Hacer más atractivas las tareas existentes que pueden ser generalmente aburridas.

Se proponen realizar tareas dentro del salón de clases con el fin de resolver pequeños ejercicios para el diseño de algoritmos.

- Definir el objetivo de la resolución del algoritmo.
- Usar tarjetas con instrucciones en pseudo código
- Resolver un algoritmo al ordenar las tarjetas con instrucciones en pseudo código.

- Trabajar en equipos y en competencia con los otros equipos, a través del uso de banderas.
- Establecer tres niveles: básico, intermedio y avanzado.

SCOPUS_G_02

Definición:

Gamificación es el uso de elementos de juego y técnicas de diseño de juego en entornos de no juego.

Actividad de investigación:

Evaluar la efectividad de una plataforma de gamificación durante un curso de Tecnologías de la Información en la Universidad Corvinus de Busapest, dirigido a estudiantes de ciencias económicas.

Uso de gamificación en programación en educación superior:

Se usa el ambiente de gamificación dentro de la plataforma *e-learning Moodle* durante el curso, los pasos de gamificación incluyen un sistema de recompensa, rutas de aprendizaje alternativo, varias opciones de retroalimentación y plataformas de interacción social. Se interesan más por la calidad del curso, para evaluar la propuesta, realizan encuestas voluntarias acerca de la satisfacción de los estudiantes que terminan el curso y los resultados del examen final.

SCOPUS_G_03

Definición:

La integración de elementos de juego, mecánicas y entornos dentro de situaciones o escenarios que no son de juego.

Detalle de información clave por material SCOPUS tema gamificación

Actividad de investigación:

El trabajo sugiere que un método de aprendizaje basado en problemas, multidimensional, gamificado puede mejorar el rendimiento, aun cuando es aplicado a una tarea compleja y tradicionalmente aburrida como la enseñanza de programación de computadoras.

Uso de gamificación en programación en educación superior:

Se emplea una secuencia en tiempo real de puntajes de evaluación, retroalimentación del instructor, y codificación en vivo para entregar una experiencia de aprendizaje completamente interactiva. Nuevamente usan una versión del juego en clase ¿Quién quiere ser millonario? Denominado Kahoot! para el curso de programación *Python*.

SCOPUS_G_04

Definición:

No indican una definición, pero comentan que existen varias. Los autores indican que su trabajo está relacionado con el uso de elementos de juego sociales en el diseño de cursos para los estudiantes.

Actividad de investigación:

Establecen claramente su pregunta de investigación ¿Puede la gamificación ser efectiva para la enseñanza en cursos de educación superior?

Uso de gamificación en programación en educación superior:

Determina las clases de jugadores: explotadores, logros, sociales y ganadores. Utilizan un entorno con mecánicas, dinámicas y estéticas; ésta última se refiere a la forma en que las dinámicas y mecánicas del juego interactúan con el arte del diseño del juego, para producir alcances culturales y emocionales. Así, identifican siete herramientas principales de gamificación (3 para mecánicas y 4 para dinámicas):

1. Sistemas de puntos.
2. Niveles.
3. Medallero.
4. Insignias.
5. Embarcar (al jugador / lo novedoso).
6. Ciclos de enganche social.
7. Contenido por descubrir.

Las intervenciones realizadas con las 7 herramientas ha permitido mejorar los porcentajes de acreditación y participación, y aumentar la satisfacción de los estudiantes.

SCOPUS_G_05

Definición:

El uso de reglas de juego a entornos que no son juego, ofrece algún tipo de recompensas a los estudiantes para motivarlos.

Área de investigación:

Aplicar los conceptos de aprendizaje basado en juegos y gamificación para mejorar el proceso de aprendizaje de estudiantes en varios niveles de edad. Ofrecer un análisis de aprendizaje basado en juegos contra gamificación en educación superior desde la perspectiva de estudiantes de educación superior (estructura de datos y algoritmos).

Uso de gamificación en programación en educación superior:

Los autores han comparado un juego que se ha implementado para dar la idea de los algoritmos de ordenación con un material gamificado, junto con una animación de acuerdo con el pseudo código, posteriormente, un cuestionario después de cada algoritmo de ordenamiento.

La intención del juego es ayudar a los estudiantes a comprender los principios fundamentales de los algoritmos de ordenación de burbuja y cubos; donde el estudiante tiene que recoger cajas y ordenarlas según sus valores mediante el uso de reglas de algoritmos de ordenación.

El estudiante puede mirar el pseudo código para un algoritmo de ordenación particular. En consecuencia, se está ordenando un conjunto de datos dado y la parte del pseudo código se está resaltando de acuerdo con la ejecución del algoritmo para que el estudiante pueda capturar la idea fundamental de un algoritmo de ordenación particular. Después de las demostraciones de algoritmos de ordenación, los estudiantes deben seguir las pruebas y podrán conocer sus estándares. Sobre la base de las calificaciones que el estudiante haya obtenido, obtendrá una pulsera para que pueda estar satisfecho con su conocimiento.

APÉNDICE F. Detalle de información clave por material WOS tema gamificación

WOS_G_01

Considerando que el material es una revisión sistemática de la gamificación, se procedió a evaluar el producto a partir de las preguntas indicadas en la sección 3.2.1.

¿Cuáles fueron los objetivos de la revisión?

El objetivo de la revisión es identificar, mediante casos aplicados en las diferentes áreas de conocimiento, cómo la gamificación ha sido empleada en la educación superior, así como también conocer mediante cadenas de búsqueda, su participación en la literatura.

¿Qué fuentes fueron buscadas para identificar estudios primarios?

Nueve bases de datos académicas: Ebsco, ACM, IEEE, springer link, Web of Science, Emerald, Science Direct, Scopus, Google Scholar.

¿Hubo algunas restricciones?

Solo se realiza la revisión para seis áreas de conocimiento definidas por el Ministerio de Educación Nacional en Colombia.

¿Cuáles fueron los criterios de inclusión/exclusión y cómo fueron aplicados?

Inclusión: Trabajos publicados del año 2000 en adelante. Todos los contextos geográficos. Publicaciones como artículos de revistas, libros, capítulos de libros, presentaciones en eventos académicos que trabajen la gamificación en Educación Superior. Trabajos que hayan implementado la gamificación en las áreas de conocimiento definidas en la revisión.

Exclusión: Trabajos que no proceden de fuentes académicas confiables (no especifica). Publicaciones que trabajen la gamificación en Educación Básica y Media u otro tipo de ambientes académicos

distintos a la Educación Superior. Referencias anteriores al 2000. Artículos que estudien la gamificación en organizaciones, manejo de clientes, capacitaciones, etc.

¿Qué criterios fueron usados para evaluar la calidad de los estudios primarios y cómo fueron aplicados?

Establece y describe cuatro criterios de evaluación con la escala de valoración Alto (5), Medio (3) y Bajo (1). Aplicación de la gamificación, Coherencia metodológica, Claridad de los argumentos, y Aporte al área de conocimiento.

¿Cómo fueron extraídos los datos desde los estudios primarios?

Elaboraron la síntesis de los artículos seleccionados

¿Cómo fueron sintetizados los datos?

Para el análisis segmentado por área de conocimiento, determinaron una variable llamada Score.

¿Cómo fueron las diferencias entre los estudios investigados?

A partir del cálculo de una variable denominada Score, determinaron porcentajes de participación en las diferentes bases de datos por área de conocimiento.

¿Cómo fueron combinados los datos?

Respecto a las áreas de conocimiento, logra determinar en cuál área existe mayor publicación del tema gamificación.

¿Fue razonable combinar los estudios?

Particularmente para el caso de estudio, Si.

¿Las conclusiones fluyen desde la evidencia?

Presenta comentarios a partir de la revisión de un artículo, no indica el porqué de la selección.

WOS_G_02

Definición:

No incluye, se comentan el beneficio de la gamificación en la motivación de los estudiantes.

Actividad de investigación:

Fue desarrollada una plataforma especial de gamificación que incluye servicios accesibles en la nube para cualquier dispositivo móvil e interfaz web. No solo se hace el actual progreso del curso fácilmente accesible para cada uno de los estudiantes, también se introducen insignias que los estudiantes pueden ganar al completar tareas complejas; así como nuevas metas de aprendizaje, una interfaz de fácil uso para el docente sin agregar sobrecarga de trabajo con tales metas, finalmente la opción de trabajar sin o con conexión a internet.

Uso de gamificación en programación en educación superior:

Trabajan con lo que denominan las cuatro entidades clave en la plataforma que desarrollaron: metas, retos, puntos e insignias. Académicas o extracurriculares, estas últimas incluyen actividades culturales, deportivas o de comunidad.

WOS_G_03

Definición:

No establece una definición o el uso de alguna de otro autor, expone el uso de actividades lúdicas en el aprendizaje.

Actividad de investigación:

Evaluar cómo la aplicación de gamificación para la evaluación puede afectar la experiencia de aprendizaje además de la motivación, la capacidad de recordar y desempeño de los estudiantes en un curso de programación de computadoras.

Uso de gamificación en programación en educación superior:

El grupo experimental revisó los mismos conceptos teóricos que el grupo de control, pero jugó una implementación del juego popular ¿Quién quiere ser millonario? que fue modificado, actualizado con preguntas de opción múltiple relacionadas con el lenguaje de programación *Python*, renombrado ¿Quién quiere ser pitonisa?, así durante las sesiones con los equipos de trabajo formados se obtenía un puntaje y el equipo ganador recibió el título de pitonisa del año, además de una playera.

WOS_G_04*Definición:*

El neologismo gamificación es aplicado para usar pensamiento de juego o mecanismo de juego para resolver problemas o enganchar a las audiencias.

Actividad de investigación:

Usar varios mecanismos de diseño de juegos en varios aspectos del proceso de aprendizaje en un curso de ciencias de la computación para evaluar el impacto de la propuesta de gamificación.

Uso de gamificación en programación en educación superior:

Un mecanismo de recompensa fue central para el método de gamificación, proporcionar clasificación automática de los estudiantes, así como constante y actualizada retroalimentación de su progreso dentro del curso.

WOS_G_05

Definición:

No determina una definición de gamificación, se comentan los beneficios con base a los trabajos revisados en sus referencias bibliográficas.

Actividad de investigación:

Este documento describe la adaptación y el diseño de un sistema de premios para motivar y calificar a los estudiantes. El enfoque proporciona un sistema de evaluación autorregulado, que brinda al estudiante acceso inmediato y constante al proceso de aprendizaje, manteniendo, en todo momento, conciencia sobre lo que ha aprendido y lo que aún necesita aprender. Además, también tiene una noción cuantitativa y cualitativa de la calificación final, lo que le permite hacer opciones con respecto a la calificación final.

Uso de gamificación en programación en educación superior:

Utiliza varios mecanismos de diseño de juegos, como desafíos adaptativos, recompensas, curiosidad y la oportunidad de aumentar el tiempo que los estudiantes pasan trabajando, experimentando y aprendiendo en una Institución de Educación Superior, particularmente en el curso de Ciencias de la computación. Las secciones en el currículo de la asignatura se transforman en niveles, otorgando estrellas por logros cada vez más complejos. Las calificaciones dependen de las experiencias de aprendizaje que los estudiantes elijan, así como del tiempo dedicado a resolver los desafíos que se les presentan. También existe el concepto de moneda, denominado *BitPoints*, que se puede utilizar para aumentar la autonomía de los estudiantes y reducir la carga de trabajo.

Una plataforma en línea está constantemente disponible para informar al estudiante sobre su progreso dentro de la materia y el grado en el momento del acceso. Cada nivel está representado por un castillo en un mapa, que se considera conquistado tan pronto como el estudiante resuelve los desafíos

requeridos. Dependiendo de la dificultad de los desafíos enfrentados, el estudiante puede enfrentar un grado inferior o superior. Solo después de terminar todos los niveles, el estudiante puede terminar exitosamente la asignatura.

El dinero virtual se puede gastar en una tienda virtual, donde el estudiante puede invertir en herramientas que pueden ayudar a resolver algunas tareas. La tienda virtual proporciona bloques de código, instrucciones y otras técnicas que pueden ser valiosas para los estudiantes. Un factor de oportunidad también se considera en la tienda, dando mayor motivación a los estudiantes.

WOS_G_01_2

Definición:

Incorporación de elementos de juego en diferentes entornos incluyendo los de aprendizaje.

Actividad de investigación:

Recopilar y analizar resultados de investigación acerca de la incorporación de la gamificación en cursos de programación.

Uso de gamificación en programación en educación superior:

Se indica que se utilizan elementos de gamificación en solitario tales como juegos serios o entornos específicos gamificados como: *Kodesh* y *Wolves and Sheep*, posteriormente se indica que se emplean elementos de gamificación combinados con otras estrategias tales como: *flipped learning*, *lighweight teams*, interacción *peer to peer*, redes sociales, MOOC, *blended learning*, *Moodle* y ambientes virtuales 3D.

APÉNDICE G. Detalle de información clave por material SCOPUS tema pensamiento computacional

SCOPUS_PC_01

Definición:

El Pensamiento Computacional es un conjunto de estrategias cognitivas y metacognitivas emparejadas con procesos y métodos de ciencias de la computación (análisis, abstracción, modelo). Puede ser relacionado a la ciencia de la computación de la misma forma como el pensamiento algorítmico es relacionado a las matemáticas.

Actividad de investigación:

El artículo introduce el Pensamiento Computacional en el contexto de las actividades de programación creativa de la Educación Superior. En este estudio, se involucra a estudiantes de pregrado en una actividad de programación creativa utilizando Scratch. Luego, se analizan las puntuaciones de Pensamiento Computacional de una herramienta de análisis automático y la evaluación humana de los proyectos de programación creativa. Los resultados sugirieron la necesidad de una evaluación humana de la programación creativa al tiempo que señalan los límites de una herramienta analítica automatizada, que no refleja la diversidad creativa de los proyectos de Scratch y anula la complejidad algorítmica.

Uso de Pensamiento Computacional en programación en educación superior:

Identifica seis componentes de la competencia de PC, dos relacionadas a la codificación y al conocimiento de tecnologías y cuatro relacionados a las cuatro fases de la solución de problemas colaborativos de PISA 2015 (Programa para la Evaluación Internacional del Estudiante – Programme for International Student Assessment – PISA). La relación es la siguiente:

Componente PC	Fase solución de problemas colaborativo Collaborative Problem Solving-CPS
COMP1: habilidad para identificar los componentes de una situación y su estructura (análisis/representación-abstracción).	CPS-A: fase de exploración y comprensión
COMP2: habilidad para organizar y modelar la situación eficientemente (organizar/modelar).	CPS-B: fase de representación y formulación.
COMP3: conocimiento de codificación.	
COMP4: conocimiento de sistemas tecnológicos (software / hardware).	
COMP5: se enfoca en la capacidad para crear un programa de computadora (programación).	CPS-C: fase de planeación y ejecución.
COMP6: habilidad para comprometerse en la evaluación y proceso iterativo de mejorar un programa de computadora.	CPS-D: fase de monitoreo y reflexión.

En consecuencia, proponen una evaluación del Pensamiento Computacional dentro del contexto de actividades de la programación creativa. El modelo #5c21 corresponde a las 5 competencias clave para la educación del siglo 21: Pensamiento Computacional, creatividad, colaboración, solución de problemas, y pensamiento crítico. Aplicado a 120 estudiantes de maestría de educación escolar primaria de la Universidad de Laval en Canadá. Estudiantes sin experiencia previa de programación ni egreso para tal perfil. Aporta discusión sobre la oportunidad del aprendizaje del paradigma de programación orientado a objetos (POO) desde los primeros pasos de las actividades de aprendizaje del Pensamiento Computacional. Examinar la oportunidad de desarrollar Pensamiento Computacional de una forma interdisciplinaria sin crear un nuevo currículo de Pensamiento Computacional en una disciplina específica. La oportunidad de desarrollar Pensamiento Computacional en diferentes niveles de educación desde educación primaria a actividades de aprendizaje a lo largo de la vida.

SCOPUS_PC_02

Definición:

Detalle de información clave por material SCOPUS tema pensamiento computacional

Comenta la definición de Wing: el Pensamiento Computacional es para resolver problemas, diseñar sistemas y comprender la conducta humana con las ciencias de la computación. Incluye una serie de actividades de pensamiento que cubren el campo de las ciencias de la computación.

Actividad de investigación:

Proponer un modo de enseñanza innovador a través de considerar comprensiblemente los conceptos de Pensamiento Computacional y la enseñanza basada en la investigación.

Uso de Pensamiento Computacional en programación en educación superior:

El desarrollo del Pensamiento Computacional tiene una gran influencia en la metodología propuesta. Se usa como caso de enseñanza el diseño del lenguaje de programación en C para mejorar la resolución de problemas de estudiantes que no son del área profesional de ciencias de la computación. Dos habilidades del Pensamiento Computacional destacan en el estudio: abstracción y automatización. El trabajo de investigación muestra que el proceso completo de enseñanza a través de un método de modelado, formalmente muestra la mejora de la efectividad del aprendizaje y la habilidad del Pensamiento Computacional. A través del análisis del modelo y la verificación por medio del caso práctico, se puede observar que el Pensamiento Computacional basado en el método de enseñanza fundamentado en la investigación puede bien resolver la contradicción entre enseñanza y aprendizaje, es un modo educativo de estudiantes como el cuerpo principal, el profesor como el líder.

SCOPUS_PC_03

Definición:

Pensamiento Computacional es utilizar los conceptos básicos de ciencias de la computación para resolver problemas, diseñar sistemas y comprender la conducta humana.

Actividad de investigación:

En el artículo se discute cómo entrenar la capacidad de pensamiento de estudiantes durante la enseñanza del lenguaje de programación.

Uso de Pensamiento Computacional en programación en educación superior:

Se proporcionan algunos consejos acerca de cómo organizar el contenido de enseñanza y son propuestos experimentos para mejorar los resultados de la enseñanza, y los caracteres del pensamiento humano son explotados para incrementar la capacidad creativa.

SCOPUS_PC_04

Definición:

Usa el concepto de Wing.

Actividad de investigación:

Como entrenar el Pensamiento Computacional de los estudiantes en el currículo del programa Visual Basic-VB con la intención de activar a los estudiantes en el análisis y solución de problemas e innovar con las computadoras.

Uso de Pensamiento Computacional en programación en educación superior:

Con base en el modo de enseñanza de VB y el Pensamiento Computacional, el estudiante puede cultivar su pensamiento de programación y habilidades en la solución de problemas, observar y comprender en su estudio, trabajo y vida los cambios en el mundo objetivo desde muchas perspectivas.

SCOPUS_PC_05

Definición:

Detalle de información clave por material SCOPUS tema pensamiento computacional

El Pensamiento Computacional es un tipo de pensamiento analítico. La esencia del Pensamiento Computacional es pensar acerca de datos e ideas, y usar y combinar estos recursos para resolver problemas. Estos conceptos fundamentales, tales como abstracción, descomposición, razonamiento heurístico, planeación, programación, entre otras, pueden ser aplicados a través de un amplio rango de disciplinas en una forma que permita nuevos avances en muchos campos.

Actividad de investigación:

Proponen un sistema de aprendizaje de un curso construido en el centro de la cultivación de la habilidad del Pensamiento Computacional y relacionado con un método reformado, el cual enfatiza la solución de problemas y el diseño algorítmico y determina la relación entre el método orientado a objetos y el método orientado a procedimientos en la enseñanza teórica, y el diseño de experimentos multi-capas y organización de un experimento basado en proyecto.

Uso de Pensamiento Computacional en programación en educación superior:

Proponen un curso mejorado en el método y estrategia de enseñanza basada en los principios de solución de problemas y el centro de enseñanza en mejorar la habilidad del Pensamiento Computacional.

APÉNDICE H. Detalle de información clave por material WOS tema pensamiento computacional

WOS_PC_01

Definición:

Es el proceso de pensamiento involucrado en la formulación de un problema y expresar su solución(es) de tal forma que una computadora – humana o máquina – pueda efectivamente llevarla a cabo. Indica cuatro técnicas que son requeridas: Descomposición, Reconocimiento de patrones, Abstracción y Algoritmo.

Actividad de investigación:

Cultivar el Pensamiento Computacional para mejorar el aprendizaje del lenguaje de programación.

Uso de Pensamiento Computacional en programación en educación superior:

Los autores proponen que a través del uso de las habilidades mencionadas en la definición de Pensamiento Computacional es posible que los estudiantes mejoren su desempeño en cualquier plataforma de programación, es decir, en el uso de cualquier lenguaje de programación. Así, establecen una relación entre las habilidades del Pensamiento Computacional y el conocimiento de un lenguaje de programación. Hacen uso de material teórico pero con un fuerte uso del aprendizaje basado en problemas.

WOS_PC_02

Definición:

Comenta la propuesta por Wing.

Actividad de investigación:

A través de promover el Pensamiento Computacional de los estudiantes, la habilidad práctica, consciente y comprensiva de los estudiantes también se promueve.

Uso de Pensamiento Computacional en programación en educación superior:

Durante el curso de C++, se combina el Pensamiento Computacional con el método de programación a través de casos reales.

WOS_PC_03*Definición:*

Es una actividad de pensamiento que cubre una amplitud de las ciencias de la computación, incluyendo el uso de conceptos básicos de la ciencia de la computación para resolver problemas, diseño de sistemas y la comprensión de la conducta humana.

Actividad de investigación:

Para resolver la problemática de “altas calificaciones y bajas habilidades” de los estudiantes, los autores proponen promover la habilidad del Pensamiento Computacional de los estudiantes en los cursos de diseño de programas.

Uso de Pensamiento Computacional en programación en educación superior:

El proceso de enseñanza se enfoca en la guía de “pensar” y “métodos” introduciendo diferentes mecanismos algorítmicos para resolver el mismo problema en las clases presenciales. Usan el Pensamiento Computacional para entrenar a los estudiantes y entonces pensaron como escribir código y obtener resultados. Las clases de Pensamiento Computacional son para entrenar a los estudiantes, usar métodos de Pensamiento Computacional para descubrir, analizar, resolver y resumir problemas.

WOS_PC_04

Definición:

Uso del concepto base de ciencias de la computación para resolver problemas, diseñar y comprender las acciones humanas.

Actividad de investigación:

Que los estudiantes mejoren su conciencia de Pensamiento Computacional, método y habilidad durante el proceso de resolución de problemas.

Uso de Pensamiento Computacional en programación en educación superior:

Debido al modo de enseñanza del lenguaje de programación C, los autores proponen el uso del método Aprendizaje Basado en Problemas para mejorar el entrenamiento del Pensamiento Computacional a través de una secuencia de actividades.

WOS_PC_05

Definición:

La definición aun no es completa y clara, indica el autor; comenta la definición propuesta por Zhou Yizhen del 2006: es el uso de los conceptos fundamentales de ciencias de la computación para la resolución de problemas, comprensión de la conducta humana además de actividades de pensamiento en el diseño de sistemas.

Actividad de investigación:

Establecer actividades para el curso de programación que fomenten habilidades del Pensamiento Computacional.

Uso de Pensamiento Computacional en programación en educación superior:

El Pensamiento Computacional, comenta el autor, debe ser entrenado en la enseñanza de programación. Al agregar filosofías de enseñanza avanzadas, tales como la enseñanza por casos, el manejo de tareas, micro lecciones o MOOCs se pueden mejorar las habilidades del estudiante.

WOS_PC_06*Definición:*

Es una habilidad fundamental para todo el mundo, no solo para los ingenieros en informática, que se basa en resolver problemas haciendo uso de conceptos fundamentales de la informática.

Actividad de investigación:

El trabajo se centra en los aspectos metodológicos acerca de cómo iniciar a un estudiante en el campo de la programación de computadoras tomando como base que no hay nada mejor para desarrollar la habilidad de resolver problemas que usar los conceptos informáticos de una asignatura de introducción a la programación.

Uso de Pensamiento Computacional en programación en educación superior:

Establecen tres etapas para conseguir el aprendizaje: oír, ver y hacer. A través del desarrollo de tales metodologías enfocadas a trabajar las etapas el curso de programación es el vehículo perfecto para desarrollar habilidades de Pensamiento Computacional ya que implica la resolución de problemas haciendo uso de conceptos informáticos.

WOS_PC_07*Definición:*

Detalle de información clave por material WOS tema pensamiento computacional

Basado en la definición inicial de Wing propone trabajar con la formulación de Kao et al. basada en cuatro etapas: descomposición, reconocimiento de patrones, generalización de patrones y diseño algorítmico. Ofrece una explicación personal de cada una de las etapas.

Área de investigación:

Formular un método a través de la información para la resolución de problemas por medio del Pensamiento Computacional.

Uso de Pensamiento Computacional en programación en educación superior:

Elaborar una pedagogía de programación. Considerando que el Pensamiento Computacional es la suma de las etapas (descomposición, abstracción, patrones y algoritmos) que se empalman e interactúan, es decir, no se pueden separar cada una de las etapas, para generar una actividad creativa (resolución de problemas), el autor argumenta que se debe permitir que la información estructure a la computación, iniciando con una concreta instancia del escenario del problema. Usar el Pensamiento Computacional para hacer buenas preguntas, para desenredar estructuras de información bien conocidas y guiar el diseño de la computación.

Referencias

- Admiraal, W., Huizenga, J., Akkerman, S., & Dam, G.-t. (2011). The concept of flow in collaborative game-based learning. *Computers in Human Behavior*, 27(3), 1185-1194. doi:10.1016/j.chb.2010.12.013
- Alsawaier, R. S. (2017). The effect of gamification on motivation and engagement. *International Journal of Information and Learning Technology*, 35(1), 56-79. doi:10.1108/IJILT-02-2017-0009
- Ausubel, P. D. (1981). *Psicología Educativa*. Distrito Federal, México: Trillas.
- Barba, E., & Chancellor, S. (2015). Tangible Media Approaches to Introductory Computer Science. *ITiCSE '15 Proceedings of the 2015 ACM Conference on Innovation and Technology in Computer Science Education* (págs. 207-212). Vilnius, Lithuania — July 04 - 08: IEEE. doi:10.1145/2729094.2742612
- Barna, B., & Fodor, S. (2018). An Empirical Study on the Use of Gamification on IT Courses at Higher Education. *Teaching and Learning in a Digital World. ICL 2017. Advances in Intelligent Systems and Computing*. 715, págs. 684-692. Cham: Springer. doi:10.1007/978-3-319-73210-7_80
- Barr, V., & Stephenson, C. (2011). Bringing computational thinking to K-12: what is Involved and what is the role of the computer science education community? *ACM Inroads*, 2(1), 48-54. doi:10.1145/1929887.1929905
- Bergmann, J., & Sams, A. (2013). Flip Your Students' Learning. *Educational Leadership*, 70(6), 16-20. Obtenido de <http://www.ascd.org/publications/educational-leadership/mar13/vol70/num06/Flip-Your-Students%27-Learning.aspx>
- Berlaga, A. J., & García, F. J. (2008). Learning Design in Adaptive Educational Hypermedia Systems. *Journal of Universal Computer Science*, 14(22), 3627-3647. doi:10.3217/jucs-014-22-3627
- Berland, M., & Wilensky, U. (2015). Comparing Virtual and Physical Robotics Environments for Supporting Complex Systems and Computational Thinking. *Journal of Science Education and Technology*, 24(5), 628-647. doi:10.1007/s10956-015-9552-x
- Berlanga, A. J., & García, F. J. (2005). IMS LD reusable elements for adaptive learning designs. *Journal of Interactive Media in Education*, 1, 12. doi:10.5334/2005-11
- Berlanga, A. J., & García, F. J. (2005). Learning technology specifications: semantic objects for adaptive learning environments. *International Journal of Learning Technology*, 1(4), 458-472. doi:10.1504/IJLT.2005.007155

- Bernardo, J., Javaloyes, J. J., & Calderero, J. F. (2011). *Educación personalizada: principios, técnicas y recursos*. Madrid: Síntesis.
- Borras-Gene, O., Martínez-nunez, M., & Fidalgo-Blanco, A. (2017). New Challenges for the motivation and learning in engineering education using gamification in MOOC. *International Journal of Engineering Education*, 32(1), 501-512. Obtenido de <http://www.scopus.com/inward/record.url?eid=2-s2.0-84959358012&partnerID=tZOtx3y1>
- Briz-Ponce, L. (2016). *Análisis de la efectividad en las Aplicaciones m-health en dispositivos móviles dentro del ámbito de la formación médica (Tesis Doctoral)*. Salamanca, España: Universidad de Salamanca.
- Bunchball. (2012). *Gamificaiton 101: An Introduction to Game Dynamics*. Obtenido de <https://www.bunchball.com/>
- Bureau_of_Labor_Statistics. (2019). *U.S. Department of Labor*. Obtenido de Occupational Outlook Handbook, Software Developers: <https://www.bls.gov/ooh/computer-and-information-technology/software-developers.htm#tab-6>
- Burke, B. (2012). *Gamification 2020: What is the future of gamification?* Obtenido de Gartner Research: <https://www.gartner.com/en/documents/2226015>
- Bustillo, J., & Garaizar, P. (2015). Scratching the surface of digital literacy... but we need to go deeper. *2014 IEEE Frontiers in Education Conference (FIE) Proceedings* (pág. art. no. 7044224). Madrid, Spain: IEEE. doi:10.1109/FIE.2014.7044224
- Byrne, J. R., Fisher, L., & Tangney, B. (2015). Empowering teachers to teach CS — Exploring a social constructivist approach for CS CPD, using the Bridge21 model. *2015 IEEE Frontiers in Education Conference (FIE)* (pág. art. no. 7344030). El Paso, TX, USA: IEEE. doi:10.1109/FIE.2015.7344030
- Cain, J., Conway, J. M., DiVall, M. V., Erstad, B. L., Lockman, P. R., Ressler, J. C., & Nemire, R. E. (2014). Report of the 2013-2014 Academic Affairs Committee. *American Journal of Pharmaceutical Education*, 78(10). doi:10.5688/ajpe7810S23
- Calao, L. A., Moreno-León, J., Correa, H. E., & Robles, G. (2015). Developing Mathematical Thinking with Scratch - An Experiment with 6th Grade Students. *Design for Teaching and Learning in a Networked World*. 9307, págs. 17-27. Cham: Springer. doi:10.1007/978-3-319-24258-3_2
- Calderero, J. F., Aguirre, A. M., Castellanos, A., Peris, R. M., & Perochena, P. (2014). Una nueva aproximación al concepto de educación personalizada y su relación con las TIC. *Education in the Knowledge Society*, 15(2), 131-151.

Referencias

- Caponetto, I., Earp, J., & Ott, M. (2014). Gamification and education: A literature review. *Proceedings of the European Conference on Games Based Learning*, (págs. 50-57).
Obtenido de <http://search.ebscohost.com/login.aspx?direct=true&db=eue&AN=99224935&site=ehost-live>
- Carreño-León, M., Sandoval-Bringas, A., Álvarez-Rodríguez, F., & Camacho-González, Y. (2018). Gamification technique for teaching programming. *2018 IEEE Global Engineering Education Conference (EDUCON)* (págs. 2009-2014). Tenerife, Spain: IEEE.
doi:10.1109/EDUCON.2018.8363482
- Chang, J., & Wei, H. (2016). Exploring Engaging Gamification Mechanics in Massive Online Open Courses. *Educational Technology & Society*, 19, 177–203.
- Chen, G.-m. (2017). Programming Language Teaching Model Based on Computational Thinking and Problem-based Learning. *Proceedings of the 2017 2nd International Seminar on Education Innovation and Economic Management (SEIEM 2017)*. Atlantis Press. doi:10.2991/seiem-17.2018.31
- Cheng, P. H., Fu, J. M., & Chen, L. W. (2016). Knowledge transfer of software tool development for functional requirements analysis. *Computer Applications in Engineering Education*, 24(1), 131-143. doi:10.1002/cae.21679
- Chiazzese, G., Fulantelli, G., Pipitone, V., & Taibi, D. (2017). Promoting computational thinking and creativeness in primary school children. *TEEM 2017 Proceedings of the 5th International Conference on Technological Ecosystems for Enhancing Multiculturality* (pág. Article No. 6). Cádiz, Spain — October 18 - 20: ACM New York, NY, USA. doi:10.1145/3144826.3145354
- Chiazzese, G., Fulantelli, G., Pipitone, V., & Taibi, D. (2018). Involucrando a los niños de educación primaria en el Pensamiento Computacional: diseñando y desarrollando videojuegos. *Education in the Knowledge Society*, 19(2), 63-81. doi:10.14201/eks20181926381
- Chilana, P. K., Alcock, C., Dembla, S., Ho, A., Hurst, A., Armstrong, B., & Guo, P. J. (2015). Perceptions of non-CS majors in intro programming: The rise of the conversational programmer. *2015 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)* (págs. 251-259). Atlanta, GA, USA: IEEE. doi:10.1109/VLHCC.2015.7357224
- Chou, Y.-K. (2013). *Top 10 Marketing Gamification Cases You Won't Forget*. Obtenido de <http://sumo.ly/dhll>
- Chrysafiadi, K., & Virvou, M. (2015). A Novel Hybrid Student Model for Personalized Education. En *Advances in Personalized Web-Based Education, Intelligent Systems Reference Library* (Vol. 78, págs. 61-90). Springer. doi:10.1007/978-3-319-12895-5_3

- Chrysafiadi, K., & Virvou, M. (2015). Student Modeling for Personalized Education: A Review of the Literature. *Advances in Personalized Web-Based Education. Intelligent Systems Reference Library*, 78, 1-24. doi:10.1007/978-3-319-12895-5_1
- Compañ-Rosique, P., Satorre-Cuerda, R., Llorens-Largo, F., & Molina-Carmona, R. (2015). Enseñando a programar: un camino directo para desarrollar el pensamiento computacional. *Revista de Educación a Distancia*, 46(11). doi:10.6018/red/46/11
- Creswell, J. W. (2013). *Research design: Qualitative, quantitative, and mixed methods approaches* (Cuarta ed.). Thousand Oaks, CA, USA: SAGE.
- Csikszentmihalyi, M. (1990). *Flow: The Psychology of Optimal Experience, 1st Harper Perennial Modern Classics*. New York: Harper Perennial.
- CSTA. (2011). *K-12 Computer Science Standards*. Obtenido de http://csta.acm.org/Curriculum/sub/CurrFiles/CSTA_K-12_CSS.pdf
- Czerkawski, B. C., & Lyman, E. W. (2015). Exploring Issues About Computational Thinking in Higher Education. *TechTrends*, 59(2), 57–65. doi:10.1007/s11528-015-0840-3
- Davis, M. H., & McPartland, J. M. (2012). High school reform and student engagement. (S. L. Christenson, A. L. Rashly, & C. Wylie, Edits.) *Handbook of Research on Student Engagement*, 515-539.
- Deci, E. L., & Ryan, R. M. (2008). Facilitating optimal motivation and psychological well-being across life's domains. *Canadian Psychology/Psychologie Canadiennem*, 49(1), 14-23. doi:10.1037/0708-5591.49.1.14
- Deterding, S., Sicart, M., Nacke, L., O'Hara, K., & Dixon, D. (2011). Gamification. using game-design elements in non-gaming contexts. *CHI '11 Extended Abstracts on Human Factors in Computing Systems* (págs. 2425-2428). Vancouver, BC, Canada — May 07 - 12: ACM New York, NY, USA. doi:10.1145/1979742.1979575
- Dicheva, D., Dichev, C., Agre, G., & Angelova, G. (2015). Gamification in education: A systematic mapping study. *Educational Technology and Society*, 18(3), 75-88. Obtenido de <http://www.scopus.com/inward/record.url?eid=2-s2.0-84938082996&partnerID=tZOtx3y1>
- Domínguez, A., Saenz-De-Navarrete, J., De-Marcos, L., Fernández-Sanz, L., Pagés, C., & Martínez-Herráiz, J. J. (2013). Gamifying learning experiences: Practical implications and outcomes. *Computers & Education*, 63, 380-392. doi:10.1016/j.compedu.2012.12.020
- Dorling, M., & White, D. (2015). Scratch: A Way to Logo and Python. *SIGCSE '15 Proceedings of the 46th ACM Technical Symposium on Computer Science Education* (págs. 191-196). Kansas City, Missouri, USA — March 04 - 07: ACM New York, NY, USA. doi:10.1145/2676723.2677256

Referencias

- Duncan, C., & Bell, T. (2015). A Pilot Computer Science and Programming Course for Primary School Students. *WIPSCCE '15 Proceedings of the Workshop in Primary and Secondary Computing Education* (págs. 39-48). London, United Kingdom — November 09 - 11: ACM New York, NY, USA. doi:10.1145/2818314.2818328
- Echegaray, G., Barroso, N., Laskurain, I., Zuza, K., & Barragués, J. I. (2017). Investigación cualitativa para la mejora de los resultados académicos en primer curso en los grados de Ingeniería de la Escuela de Ingeniería de Gipuzkoa. *IV Congreso Internacional sobre Aprendizaje, Innovación y Competitividad-CINAIC (Zaragoza 4-6 Octubre 2017)*. doi:10.26754/CINAIC.2017.000001_093
- Ertmer, P., & Newby, T. (1993). Behaviorism, cognitivism, constructivism: comparing critical features from an instructional design perspective. *Performance Improvement Quarterly*, 50-72.
- Escutia-Romero, R., & Pamplona-Roche, S. (2017). Gamificación en la asignatura Derecho Romano: un estudio de caso. *IV Congreso Internacional sobre Aprendizaje, Innovación y Competitividad-CINAIC. Zaragoza 4-6 Octubre*. doi:10.26754/CINAIC.2017.000001_091
- Espino, E. E., & González, C. S. (2015). Influence of Gender on Computational Thinking. *Interacción '15 Proceedings of the XVI International Conference on Human Computer Interaction* (pág. art. no. 36). Vilanova i la Geltru, Spain — September 07 - 09: ACM New York, NY, USA. doi:10.1145/2829875.2829904
- European Schoolnet. (2015). *Computing our future: Computer programming and coding - Priorities, school curricula and initiatives across Europe*. Belgium. Obtenido de http://www.eun.org/documents/411753/817341/Computing+our+future_final_2015.pdf/d3780a64-1081-4488-8549-6033200e3c03
- Ferraro, M. P., Álvarez, H. L., & Peñalvo, F. G. (2004). Adaptive Educational Hypermedia Proposal Based on Learning Styles and Quality Evaluation. In P. De Bra & W. Nejdl (Eds.), *Adaptive Hypermedia and Adaptive Web-Based Systems. 3rd International Conference* (págs. 316-319). Berlin: Springer Verlag. doi:10.1007/978-3-540-27780-4_41
- Ferreras-Fernández, T. (2016). *Visibilidad e impacto de la literatura gris científica en repositorios institucionales de acceso abierto. Estudio de caso bibliométrico del repositorio Gredos de la Universidad de Salamanca. (PhD)*. Salamanca, España: Universidad de Salamanca.
- Fidalgo-Blanco, Á., Martínez-Nuñez, M., Borrás-Gene, O., & Sanchez-Medina, J. J. (2017). Micro flip teaching – An innovative model to promote the active involvement of students. *Computers in Human Behavior*, 72, 713-723. doi:10.1016/j.chb.2016.07.060
- Fidalgo-Blanco, Á., Sein-Echaluce, M. L., & García-Peñalvo, F. J. (2015). Identifying Educational Innovation Characteristics. *3rd International Conference in Technological Ecosystems for Enhancing Multiculturality (TEEM 2015)*. Porto, Portugal.

- Fotaris, P., Mastoras, T., Leinfellner, R., & Rosunally, Y. (2015). Who Wants To Be A Pythonista? Using Gamification To Teach Computer Programming. *7th annual International Conference on Education and New Learning Technologies (EDULEARN15)*.
- Fotaris, P., Mastoras, T., Leinfellner, R., & Yasmine, R. (2015). From hiscore to high marks: Empirical study of teaching programming through gamification. *Proceedings of the 9th European Conference on Games Based Learning ECGBL 2015*. Steinkjer, Norway, 8-9 October 2015. Obtenido de <http://eprints.uwe.ac.uk/28599/>
- Fredricks, J. A., Blumenfeld, P. C., & Paris, A. H. (2004). School Engagement: Potential of the Concept, State of the Evidence. *Review of Educational Research, 74*, 59–109.
- Freudmann, E. A., & Bakamitsos, Y. (2014). The Role of Gamification in Non-profit Marketing: An Information Processing Account. *Procedia - Social and Behavioral Sciences, 148*, 567-572. doi:10.1016/j.sbspro.2014.07.081
- Gao, P. (2014). Using Personalized Education to take the place of Standardized Education. *Journal of Education and Training Studies, 2*(2), 44-47.
- Gao, Q. (2014). The computational thinking- oriented inquiry teaching mode for advanced programming language course. *Bio Technology: A Indian Journal, 10*(12), 6588-6595. Obtenido de <https://www.tsijournals.com/articles/the-computational-thinking-oriented-inquiry-teaching-mode-for-advanced-programming-language-course.pdf>
- García-Cabrera, L., Ruano-Ruano, I., Balsas-Almagro, J. R., & Fuentes-Martínez, R. (2017). Metodología para realizar mini-vídeos accesibles y multilingües para formación universitaria. *IV Congreso Internacional sobre Aprendizaje, Innovación y Competitividad-CINAIC*. Zaragoza 4-6 Octubre. doi:10.26754/CINAIC.2017.000001_028
- García-Holgado, A. (2018). *Análisis de integración de soluciones basadas en software como servicio para la implantación de Ecosistemas Tecnológicos Educativos*. Salamanca, España: Tesis doctoral, Programa de Doctorado en Formación del Conocimiento, Universidad de Salamanca.
- García-Peñalvo, F. J. (2005). Estado actual de los sistemas e-learning. *Teoría de la Educación: Educación y Cultura en la Sociedad de la Información, 6*(2).
- García-Peñalvo, F. J. (2008). *Advances in E-Learning: Experiences and Methodologies*. Hershey, PA: IGI Global. doi:10.4018/978-1-59904-756-0
- García-Peñalvo, F. J. (2013). Education in knowledge society: a new PhD programme approach. *TEEM '13 Proceedings of the First International Conference on Technological Ecosystem for Enhancing Multiculturality* (págs. 575-577). Salamanca, Spain — November 14 - 15, 2013: ACM New York, NY, USA. doi:10.1145/2536536.2536624

Referencias

- García-Peñalvo, F. J. (2014). Formación en la sociedad del conocimiento, un programa de doctorado con una perspectiva interdisciplinar. *Education in the Knowledge Society*, 15(1), 4-9.
- García-Peñalvo, F. J. (2015). Cómo entender el concepto de presencialidad en los procesos educativos en el siglo XXI. *Education in the Knowledge Society*, 16(2), 6-12. doi:10.14201/eks2015162612
- García-Peñalvo, F. J. (2015). Engineering Contributions to a Multicultural Perspective of the Knowledge Society. *IEEE Revista Iberoamericana de Tecnologías del Aprendizaje*, 10(1), 17-18. doi:10.1109/RITA.2015.2391371
- García-Peñalvo, F. J. (2016). A brief introduction to TACCLE 3 — coding European project. 2016 *International Symposium on Computers in Education (SIIE)*. Salamanca, Spain: IEEE. doi:10.1109/SIIE.2016.7751876
- García-Peñalvo, F. J. (2016). Proyecto TACCLE3 - Coding. En F. G.-P. Mendes (Ed.), *XVIII Simposio Internacional de Informática Educativa, SIIE* (págs. 187-189). Salamanca, España: Ediciones Universidad de Salamanca.
- García-Peñalvo, F. J. (2016). What Computational Thinking Is. *Journal of Information Technology Research*, 9(3), v-viii.
- García-Peñalvo, F. J. (2018). Computational thinking. *IEEE Revista Iberoamericana de Tecnologías del Aprendizaje (IEEE RITA)*, 13(1), 17-19. doi:10.1109/RITA.2018.280993
- García-Peñalvo, F. J., & Cruz-Benito, J. (2016). Computational thinking in pre-university education. En F. J. García-Peñalvo (Ed.), *TEEM '16 Proceedings of the Fourth International Conference on Technological Ecosystems for Enhancing Multiculturality* (págs. 13-17). Salamanca, Spain: ACM, New York, NY, USA. doi:10.1145/3012430.3012490
- García-Peñalvo, F. J., & Mendes, A. J. (2018). Exploring the computational thinking effects in pre-university education. *Computers in Human Behavior*, 80, 407-411. doi:10.1016/j.chb.2017.12.005
- García-Peñalvo, F. J., & Seoane-Pardo, A. (2015). Una revisión actualizada del concepto de eLearning. *Décimo Aniversario. Education In The Knowledge Society*, 16(1), 119-144. doi:10.14201/eks2015161119144
- García-Peñalvo, F. J., Moreno-López, L., & Sánchez-Gómez, M. C. (2018). Empirical evaluation of educational interactive systems. *Quality & Quantity*, 52(6), 2427-2434. doi:10.1007/s11135-018-0808-4
- García-Peñalvo, F. J., Reimann, D., & Maday, C. (2018). Introducing Coding and Computational Thinking in the Schools: The TACCLE 3 – Coding Project Experience. En M. Khine (Ed.).

- Computational Thinking in the STEM Disciplines, Springer, Cham. doi:10.1007/978-3-319-93566-9_11
- García-Peñalvo, F. J., Reimann, D., Tuul, M., Rees, A., & Jormanainen, I. (2016). *An overview of the most relevant literature on coding and computational thinking with emphasis on the relevant issues for teachers*. Belgium: TACCLE3 Consortium. doi:10.5281/zenodo.16512
- García-Vega, M. Á. (2019). Automatización: así es la batalla entre trabajo y tecnología. *Revista Retina*. Obtenido de https://retina.elpais.com/retina/2019/05/24/tendencias/1558680372_855666.html?id_externo_rsoc=TW_CM
- Garett, R., & Young, S. D. (2018). Health Care Gamification: A Study of Game Mechanics and Elements. *Technology, Knowledge and Learning*, 1-13. doi:10.1007/s10758-018-9353-4
- Gómez, S. M., & Caicedo, L. M. (2015). Studying university programs under e-learning and b-learning modalities. *Revista Lasallista de Investigación*, 12(2), 94-104.
- Gomez-Galvez, P., Suárez-Mejías, C., & Fernandez-Luque, L. (2015). Social media for empowering people with diabetes: Current status and future trends. *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)* (págs. 2135–2138). Milan, Italy : IEEE. doi:10.1109/EMBC.2015.7318811
- González, M. E. (2015). El b-learning como modalidad educativa para construir conocimiento. *Opción: Revista de Ciencias Humanas y Sociales*, 31(2), 501-531.
- González-Pérez, L. I. (2019). *Protocolo de evaluación de la aceptación de los repositorios institucionales por parte de los usuarios: en el marco de una colección de recursos sobre sustentabilidad energética*. Salamanca, España: Tesis doctoral, Programa de Doctorado en Formación de la Sociedad del Conocimiento, Universidad de Salamanca.
- González-Rogado, A. B., Rodríguez-Conde, M. J., Olmos-Migueláñez, S., Borham, M., & García-Peñalvo, F. J. (2013). Experimental evaluation of the impact of b-learning methodologies on engineering students in Spain. *Computers in Human Behavior*, 29(2), 370-377. doi:10.1016/j.chb.2012.02.003
- González-Rogado, A. B., Rodríguez-Conde, M. J., Olmos-Migueláñez, S., Borham, M., & García-Peñalvo, F. J. (2014). Key Factors for Determining Student Satisfaction in Engineering: A Regression Study. *International Journal of Engineering Education (IJEE)*, 30(3), 576-584.
- Google for Education. (2016). *Exploring Computational Thinking*. Obtenido de <https://www.google.com/edu/resources/programs/exploring-computational-thinking/>
- Gros, B., & García-Peñalvo, F. J. (2016). Future Trends in the Design Strategies and Technological Affordances of E-Learning. (M. Spector, B. Lockee, & M. Childress, Edits.) *Learning, Design, and Technology*, 1-23. doi:10.1007/978-3-319-17727-4_67-1

Referencias

- Gruman, J., Holmes-Rovner, M., French, M. E., Jeffres, D., Sofaer, S., Shaller, D., & Prager, D. J. (2010). From patient education to patient engagement: implications for the field of patient education. *Patient Education and Counseling, 78*, 350-356.
- Hamari, J. (2017). Do badges increase user activity? A field experiment on the effects of gamification. *Computers in Human Behavior, 71*, 469-478. doi:10.1016/j.chb.2015.03.036
- Hamari, J., & Koivisto, J. (2015). "Working out for likes": An empirical study on social influence in exercise gamification. *Computers in Human Behavior, 50*, 333-347. doi:10.1016/j.chb.2015.04.018
- Hamari, J., Shernoff, D. J., Rowe, E., Coller, B., Asbell-Clarke, J., & Edwards, T. (2016). Challenging games help students learn: An empirical study on engagement, flow and immersion in game-based learning. *Computers in Human Behavior, 54*, 170-179. doi:10.1016/j.chb.2015.07.045
- Hamlin, B., Riehl, J., Hamlin, A. J., & Monte, A. (2010). Work in progress — What are you thinking? Over confidence in first year students. *2010 IEEE Frontiers in Education Conference (FIE)* (págs. F2H1-F2H2). Washington, DC, USA: IEEE. doi:10.1109/FIE.2010.5673354
- Han, A., Kim, J., & Wohn, K. (2015). Entry: visual programming to enhance children's computational thinking. *UbiComp/ISWC'15 Adjunct Adjunct Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2015 ACM International Symposium on Wearable Computers* (págs. 73-76). Osaka, Japan - September 07 - 11: ACM New York, NY, USA. doi:10.1145/2800835.2800871
- Hanus, M. D., & Fox, J. (2015). Assessing the effects of gamification in the classroom: A longitudinal study on intrinsic motivation, social comparison, satisfaction, effort, and academic performance. *Computers & Education, 80*, 152-161. doi:10.1016/j.compedu.2014.08.019
- Hart, S. A. (2016). Precision Education Initiative: Moving Toward Personalized Education. *Mind, Brain, and Education, 10*(4), 209-211. doi:10.1111/mbe.12109
- Hernández, G. (2002). *Paradigmas en psicología de la educación*. Distrito Federal, México: Editorial Paidós.
- Hernández, L., Muñoz, M., Mejía, J., Peña, A., Rangel, N., & Torres, C. (2017). Una Revisión Sistemática de la Literatura Enfocada en el uso de Gamificación en Equipos de Trabajo en la Ingeniería de Software. *RISTI - Revista Ibérica de Sistemas y Tecnologías de Información, 21*, 33-50. doi:10.17013/risti.21.33-50
- Hernández, R., Fernández, C., & Baptista, P. (2006). *Metodología de la investigación* (Cuarta ed.). México: Mc. Graw Hill.

- Hernández, R., Fernández, C., & Baptista, P. (2010). *Metodología de la investigación* (Quinta ed.). México: Mc. Graw Hill.
- Hernández, R., Fernández, C., & Baptista, P. (2014). *Metodología de la investigación* (Sexta ed.). México: Mc. Graw Hill.
- Hew, K. F., Huang, B., Chu, K. W., & Chiu, D. K. (2016). Engaging Asian students through game mechanics: Findings from two experiment studies. *Computers & Education*, 92-93, 221-236. doi:10.1016/j.compedu.2015.10.010
- Hou, H.-T., & Li, M.-C. (2014). Evaluating multiple aspects of a digital educational problem-solving-based adventure game. *Computers in Human Behavior*, 30, 29-38. doi:10.1016/j.chb.2013.07.052
- Huang, W., Deng, Z., & Rongsheng, D. (2009). Programming Courses Teaching Method for Ability Enhancement of Computational Thinking. *2009 International Association of Computer Science and Information Technology - Spring Conference* (págs. 182-185). Singapore, Singapore: IEEE. doi:10.1109/IACSIT-SC.2009.52
- Hui-Chi, C., Chiu-Fan, H., Cheng-Chih, W., & Yu-Tzu, L. (2015). Computational Thinking Curriculum for K-12 Education -- A Delphi Survey. *2015 International Conference on Learning and Teaching in Computing and Engineering* (págs. 213-214). Taipei, Taiwan: IEEE. doi:10.1109/LaTiCE.2015.44
- Humanante, P. R., García-Peñalvo, F. J., & Conde, M. A. (2015). Mobile personal learning environments: conceptualization and structure. *3rd International Conference in Technological Ecosystems for Enhancing Multiculturality (TEEM 2015)*. Octubre, Porto, Portugal.
- INACAP. (2015). *Modelo Educativo Institucional 2015*. Obtenido de <http://www.inacap.cl/web/acerca-de/Modelo-Educativo-2015.pdf>
- Iosup, A., & Epema, D. (2014). An experience report on using gamification in technical higher education. *SIGCSE '14 Proceedings of the 45th ACM technical symposium on Computer science education* (págs. 27-32). Atlanta, Georgia, USA — March 05 - 08: ACM New York, NY, USA. doi:10.1145/2538862.2538899
- James, J. L., & George, H. L. (2009). Thinking about computational thinking. *SIGCSE '09 Proceedings of the 40th ACM technical symposium on Computer science education*. 41, págs. 260-264. Chattanooga, TN, USA — March 04 - 07, 2009: ACM New York, NY, USA. doi:10.1145/1508865.1508959
- Jayasinghe, U., & Dharmaratne, A. (2013). Game based learning vs. gamification from the higher education students' perspective. *Proceedings of 2013 IEEE International Conference on*

Referencias

- Teaching, Assessment and Learning for Engineering (TALE)* (págs. 683-688). Bali, Indonesia : IEEE. doi:10.1109/TALE.2013.6654524
- Jenkins, C. (2015). A Work in Progress Paper: Evaluating a Microworlds-based Learning Approach for Developing Literacy and Computational Thinking in Cross-curricular Contexts. *WiPSCE '15 Proceedings of the Workshop in Primary and Secondary Computing Education* (págs. 61-64). London, United Kingdom — November 09 - 11: ACM New York, NY, USA. doi:10.1145/2818314.2818316
- Kapp, K. M. (2012). *The gamification of learning and instruction: game-based methods and strategies for training and education*. San Francisco, USA: Wiley.
- Kazimoglu, C., Kiernan, M., Bacon, L., & MacKinnon, L. (2012). Learning Programming at the Computational Thinking Level via Digital Game-Play. *Procedia Computer Science*, 9, 522-531. doi:10.1016/j.procs.2012.04.056
- Keusch, F., & Zhang, C. (2015). A Review of Issues in Gamified Surveys. *Social Science Computer Review*, 35(2), 147-166. doi:10.1177/0894439315608451
- Khalil, M., Ebner, M., & Admiraal, W. (2017). How can Gamification Improve MOOC Student Engagement? *The 11th European Conference on Game-Based Learning ECGBL 2017*, (págs. 819–828).
- Khasianov, A., Ganiev, B., & Shakhova, I. (2016). Gamification in higher education: Kazan Federal University Primer. *International Scientific Conference – eLearning and Software for Education*, 1, págs. 519-522. doi:10.12753/2066-026X-18-075
- Kitchenham, B. (2004). *Procedures for Performing Systematic Reviews*. Keele, UK: Joint Technical Report.
- Koivisto, J. M., Multisilta, J., Niemi, H., Katajisto, J., & Eriksson, E. (2016). Learning by playing: A cross-sectional descriptive study of nursing students' experiences of learning clinical reasoning. *Nurse Education Today*, 45, 22-28. doi:10.1016/j.nedt.2016.06.009
- Kostolányová, K. (2017). Adaptation of Personalized Education in E-learning Environment. En T. T. Wu, R. Gennari, Y. M. Huang, & Y. Cao (Ed.), *Emerging Technologies for Education. SETE 2016. Lecture Notes in Computer Science. 10108*, págs. 433-442. Cham: Springer. doi:10.1007/978-3-319-52836-6_46
- Kucirkova, N., & Littleton, K. (2017). Developing personalised education for personal mobile technologies with the pluralisation agenda. *Oxford Review of Education*, 43(3), 276-288. doi:10.1080/03054985.2017.1305046
- Kyewski, E., & Krämer, N. C. (2018). To gamify or not to gamify? An experimental field study of the influence of badges on motivation, activity, and performance in an online learning course. *Computers & Education*, 118, 25-37. doi:10.1016/j.compedu.2017.11.006

- Laksitowening, K. A., & Hasibuan, Z. A. (2015). Personalized e-learning architecture in standard-based education. *2015 International Conference on Science in Information Technology (ICSITech)* (págs. 110-114). Yogyakarta, Indonesia: IEEE.
doi:10.1109/ICSITech.2015.7407787
- Larkins, D. B., & Harvey, W. (2010). Introductory computational science using MATLAB and image processing. *Procedia Computer Science*, *1*(1), 913-919. doi:10.1016/j.procs.2010.04.100
- Lee, J., & Hammer, J. (2011). Gamification in Education : What , How , Why Bother ? *Academic Exchange Quarterly*, *15*(2).
- Lerís, D., & Sein-Echaluce, M. L. (2011). La personalización del aprendizaje: Un objetivo del paradigma educativo centrado en el aprendizaje. *Arbor Ciencia, Pensamiento y Cultura*, *187*(3), 123-134. doi:10.3989/arbor.2011.Extra-3n3135
- Linares-Espinós, E., Hernández, V., Domínguez-Escrig, J. L., Fernández-Pello, S., Hevia, V., Mayor, J., . . . Ribal, M. J. (2018). Methodology of a systematic review. *Actas Urológicas Españolas (English Edition)*, *42*(8), 499-506. doi:10.1016/j.acuro.2018.01.010
- Lingling, Z., Xiaohong, S., & Tiantian, W. (2015). Bring CS2013 Recommendations into c Programming Course. *Procedia - Social and Behavioral Sciences*, *176*, 194-199. doi:10.1016/j.sbspro.2015.01.461
- Llorens-Largo, F., Gallego-Durán, F. J., Villagrà-Arnedo, C. J., Compañ-Rosique, P., Satorre-Cuerda, R., & Molina-Carmona, R. (2016). Gamificación del Proceso de Aprendizaje: Lecciones Aprendidas. *VAEP-RITA*, *4*(1), 25-32.
- Llorens-Largo, F., García-Peñalvo, F. J., Molero-Prieto, X., & Vendrell-Vidal, E. (2017). La enseñanza de la informática, la programación y el pensamiento computacional en los estudios preuniversitarios. *Education in the Knowledge Society*, *18*(2), 7-17. doi:10.14201/eks2017182717
- Lopes, R. (2014). An award system for gamification in higher education. *7th International Conference of Education, Research and Innovation*, (págs. 5563-5573). Seville, Spain. Obtenido de https://bibliotecadigital.ipb.pt/bitstream/10198/12907/1/2014_iceri_award_system.pdf
- Lopes, R., & Mesquita, C. (2015). Evaluation of a gamification methodology in higher education. *7th International Conference on Education and New Learning Technologies, EDULEARN15 Proceedings, Barcelona, Spain*, (págs. 6996-7005).
- Lozada, C., & Betancur, S. (2017). La gamificación en la educación superior: una revisión sistemática. *Revista Ingenierías Universidad de Medellín*, *16*(31), 97-124. doi:10.22395/rium.v16n31a5

Referencias

- Lye, S. Y., & Koh, J. H. (2014). Review on teaching and learning of computational thinking through programming: What is next for K-12? *Computers in Human Behavior, 41*, 51-61. doi:10.1016/j.chb.2014.09.012
- Magaña-Valladares, L., González-Robledo, M. C., Rosas-Magallanes, C., Mejía-Arias, M. Á., Arreola-Ornelas, H., & Knaul, F. M. (2016). Training Primary Health Professionals in Breast Cancer Prevention: Evidence and Experience from Mexico. *Journal of Cancer Education, 33*(1), 160-166. doi:10.1007/s13187-016-1065-7
- Magro-Mazo, C. (2015). *Personalización y educación ¿Cuál era la pregunta?* Obtenido de Boletín scopeo 105: <http://scopeo.usal.es/boletin-scopeo-no-105/>
- Manterola, C., Astudillo, P., Arias, E., & Claros, N. (2011). Revisiones sistemáticas de la literatura: Qué se debe hacer acerca de ellas. *Cirugía Española, 91*(3), 149-155.
- Marín, V. (2015). La gamificación educativa. Una alternativa para la enseñanza creativa. *Digital Education Review, 27*, 5-8. Obtenido de http://revistes.ub.edu/index.php/der/article/view/12486/pdf_1
- Marqués, A., & Aguilar-Paredes, C. (2017). "S1PE": Gamificación para conocer el sector audiovisual. Una propuesta de innovación docente en el grado de Comunicación Audiovisual. *IV Congreso Internacional sobre Aprendizaje, Innovación y Competitividad-CINAIC*. Zaragoza 4-6 Octubre. doi:10.26754/CINAIC.2017.000001_036
- Martínez, H. (2014). *Metodología de la Investigación con enfoque por competencias*. México: Cengage Learning Editores.
- Mazaika, K. (2017). Will The Demand For Developers Continue To Increase? *Forbes*. Obtenido de <https://www.forbes.com/sites/quora/2017/01/20/will-the-demand-for-developers-continue-to-increase/#1272429e33ee>
- Mekler, E. D., Brühlmann, F., Opwis, K., & Tuch, A. T. (2013). Do points, levels and leaderboards harm intrinsic motivation?: an empirical analysis of common gamification elements. *Gamification '13 Proceedings of the First International Conference on Gameful Design, Research, and Applications* (págs. 66-73). Toronto, Ontario, Canada — October 02 - 04: ACM New York, NY, USA. doi:10.1145/2583008.2583017
- Michaelson, G. (2015). Teaching Programming with Computational and Informational Thinking. *Journal of Pedagogic Development, 5*(1). Obtenido de <https://www.beds.ac.uk/jpd/volume-5-issue-1-march-2015/teaching-programming-with-computational-and-informational-thinking>
- Milne, I., & Rowe, G. (2002). Difficulties in Learning and Teaching Programming—Views of Students and Tutors. *Education and Information Technologies, 7*(1), 55-66. doi:10.1023/A:1015362608943

- Moodle. (2019). *Moodle Statistics*. Obtenido de <https://moodle.net/stats/>
- Morganti, L., Pallavicini, F., Cadel, E., Candelieri, A., Archetti, F., & Mantovani, F. (2017). Gaming for Earth: Serious games and gamification to engage consumers in pro-environmental behaviours for energy efficiency. *Energy Research & Social Science*, 29, 95-102. doi:10.1016/j.erss.2017.05.001
- Morrowy, T., Sarvestani, S. S., & Hurson, A. R. (2016). Algorithmic Decision Support for Personalized Education. *2016 IEEE 17th International Conference on Information Reuse and Integration (IRI)* (págs. 188-197). Pittsburgh, PA, USA, 28-30 July 2016: IEEE. doi:10.1109/IRI.2016.32
- Ni, Z. (2017). Discussion on Case Teaching Method Based on Computational Thinking in Programming Teaching. *2017 International Conference on Social science, Education and Humanities Research (ICSEHR 2017)*. Atlantis Press. doi:10.2991/icsehr-17.2017.9
- Nisbet, S., & Williams, A. (2009). Improving students' attitudes to chance with games and activities. *Australian Mathematics Teacher*, 65(3), 25-37.
- Olelewe, C. J., & Agomuo, E. E. (2016). Effects of B-learning and F2F learning environments on students' achievement in QBASIC programming. *Computers & Education*, 103, 76-86. doi:10.1016/j.compedu.2016.09.012
- Olivares, J. C., Jiménez, J. A., Ortiz, O., & Rodríguez, N. (2015). Desarrollo de una aplicación para fortalecer el aprendizaje de los fundamentos de programación. *Revista de ciencia e ingeniería del instituto tecnológico superior de Coatzacoalcos*, 2(2).
- Park, C. J., Hyun, J. S., & Heuilan, J. (2015). Effects of gender and abstract thinking factors on adolescents' computer program learning. *2015 IEEE Frontiers in Education Conference (FIE)* (pág. art. no. 7344115). El Paso, TX, USA: IEEE. doi:10.1109/FIE.2015.7344115
- Petersen, R. (2013). *21 Companies Using Gamification to Get Better Business Results*. Obtenido de Heyo blog: <http://blog.heyo.com/21-companies-using-gamification-to-get-better-business-results/?lang=es> via @heyo
- Piaget, J. (1986). *Jean Piaget Psicología y Pedagogía*. México: Ariel.
- Pólya, G. (1957). *How to solve it*. United States of America: Anchor books edition.
- Punset, E. (2011). *De las inteligencias múltiples a la educación personalizada*. Obtenido de <http://www.rtve.es/television/20111209/inteligencias-multiples-educacion-personalizada/480968.shtml>
- Ramírez, M. S. (2012). *Modelos y estrategias de enseñanza para ambientes innovadores*. México: Editorial digital Tecnológico de Monterrey.

Referencias

- Ramirez-Lopez, A., & Muñoz, D. F. (2015). Increasing Practical Lessons and Inclusion of Applied Examples to Motivate University Students during Programming Courses. *Procedia - Social and Behavioral Sciences*, 176, 552-564. doi:10.1016/j.sbspro.2015.01.510
- Repenning, A., Webb, D. C., Koh, K. H., Nickerson, H., Miller, S. B., Brand, C., . . . Repenning, N. (2015). Scalable Game Design: A Strategy to Bring Systemic Computer Science Education to Schools through Game Design and Simulation Creation. *ACM Transactions on Computing Education (TOCE) - Special Issue II on Computer Science Education in K-12 Schools*, 15(2), Article No. 11. doi:10.1145/2700517
- Revelo, O., Collazos, C. A., & Jiménez, J. A. (2018). La gamificación como estrategia didáctica para la enseñanza/aprendizaje de la programación: un mapeo sistemático de la literatura. *Lámpsakos*, 19, 31-46. doi:10.21501/21454086.2347
- Rincón-Flores, E. G. (2018). *Gamificación en ambientes masivos de innovación abierta en el área de sustentabilidad energética*. Salamanca, España: Tesis doctoral, Programa de doctorado en Formación en la Sociedad del Conocimiento, Universidad de Salamanca.
- Rincón-Flores, E., Ramírez-Montoya, M. S., & Mena, J. J. (2016). Challenge-based gamification and its impact in teaching mathematical modeling. *Proceedings of the fourth International Conference on Technological Ecosystems for Enhancing Multiculturality, TEEM 2016*. Salamanca, Spain.
- Rincón-Flores, E., Ramírez-Montoya, M. S., & Mena, J. J. (2016). Challenge-based gamification as a teaching' Open Educational Innovation strategy in the energy sustainability area. *Proceedings of the fourth International Conference on Technological Ecosystems for Enhancing Multiculturality, TEEM 2016*. Salamanca, Spain.
- Robson, K., Plangger, K., Kietzmann, J. H., McCarthy, I., & Pitt, L. (2015). Is it all a game? Understanding the principles of gamification. *Business Horizons*, 58(4), 411-420. doi:10.1016/j.bushor.2015.03.006
- Rodríguez, F., & Santiago, R. (2015). *Gamificación: cómo motivar a tu alumnado y mejorar el clima en el aula*. Barcelona, España: Grupo Oceano.
- Rodríguez-Gómez, D., & Valldeoriola-Roquet, J. (2009). *Metodología de la investigación*. Barcelona, España: Universitat Oberta de Catalunya.
- Rojas-López, A., & García-Peñalvo, F. J. (2016). Relationship of knowledge to learn in programming methodology and evaluation of computational thinking. *Proceedings of the Fourth International Conference on Technological Ecosystems for Enhancing Multiculturality (TEEM'16)* (págs. 73-77). Salamanca, Spain, November 2-4: F. J. García-Peñalvo, Ed. ICPS: ACM International Conference Proceeding Series, New York, NY, USA: ACM. doi:10.1145/3012430.3012499

- Rojas-López, A., & García-Peñalvo, F. J. (2018). Learning Scenarios for the Subject Methodology of Programming from Evaluating the Computational Thinking of New Students. *IEEE Revista Iberoamericana de Tecnologías del Aprendizaje*, 13(1), 30-36.
doi:10.1109/RITA.2018.2809941
- Román, M., Pérez, J. C., & Jiménez, C. (2015). Test de Pensamiento Computacional: diseño y psicometría general Computational Thinking Test: design & general psychometry. *III Congreso Internacional sobre Aprendizaje, Innovación y Competitividad (CINAIC 2015)*, Octubre 14-16. Madrid, ESPAÑA.
- Romero, M., Lepage, A., & Lille, B. (2017). Computational thinking development through creative programming in higher education. *International Journal of Educational Technology in Higher Education*, 14-42. doi:10.1186/s41239-017-0080-z
- Rosser-Limiñana, A., & Martínez, R. S. (2015). Actitud de los estudiantes ante la implantación del B-Learning en la docencia universitaria. *Opción*, 31(4), 814-825.
- Royal Society. (2012). *Shut down or restart: The way forward for computing in UK schools*.
Obtenido de <https://royalsociety.org/~media/education/computing-in-schools/2012-01-12-computing-in-schools.pdf>
- Saari, E. M., Blanchfield, P., & Hopkins, G. (2015). Learning Computational Thinking Through the Use of Flash Action Scripts - Preparing Trainee Elementary School Teachers for Teaching Computer Programming. *7th International Conference on Computer Supported Education*, (págs. 75-84). Portugal. doi:10.5220/0005442600750084
- Saari, E. M., Blanchfield, P., & Hopkins, G. (2016). Computational Thinking: A Tool to Motivate Understanding in Elementary School Teachers. *Computer Supported Education. CSEDU 2015. Communications in Computer and Information Science. 583*. Cham: Springer.
doi:10.1007/978-3-319-29585-5_20
- Sadovaya, V. V., Korshunova, O. V., & Nauruzbay, Z. Z. (2016). Personalized education strategies. *Mathematics Education*, 11(1), 199-209. doi:10.12973/iser.2016.21019a
- Sáez-López, J. M., Román-González, M., & Vázquez-Cano, E. (2016). Visual programming languages integrated across the curriculum in elementary school: A two year case study using "Scratch" in five schools. *Computers & Education*, 97, 129-141.
doi:10.1016/j.compedu.2016.03.003
- Sailer, M., Hense, J. U., Mayr, S. K., & Mandl, H. (2017). How gamification motivates: An experimental study of the effects of specific game design elements on psychological need satisfaction. *Computers in Human Behavior*, 69, 371-380. doi:10.1016/j.chb.2016.12.033
- Sakhnini, V., & Hazzan, O. (2008). Reducing Abstraction in High School Computer Science Education: The Case of Definition, Implementation, and Use of Abstract Data Types.

Referencias

- Journal on Educational Resources in Computing (JERIC)*, 8(2), 1-13.
doi:10.1145/1362787.1362789
- Sánchez, F. (2015). Gamificación. *Education In The Knowledge Society*, 16(2), 13-15.
doi:10.14201/eks20151621315
- Sánchez-Gómez, M. C. (2015). Metodología de investigación en pedagogía social, avance cualitativo y modelos mixtos. *Pedagogía Social. Revista Interuniversitaria*, 26, 21-34.
doi:10.7179/PSRI_2015.26.01
- Sardi, L., Idri, A., & Fernández-Alemán, J. L. (2017). A systematic review of gamification in e-Health. *Journal of Biomedical Informatics*, 71, 31-48. doi:10.1016/j.jbi.2017.05.011
- Sarmiento, H. R., Reis, C. A., Zaramella, V., Almeida, L. D., & Tacla, C. A. (2015). Supporting the Development of Computational Thinking: A Robotic Platform Controlled by Smartphone. *International Conference on Learning and Collaboration Technologies, LCT 2015: Learning and Collaboration Technologies*. 9192, págs. 124-135. Cham: Springer. doi:10.1007/978-3-319-20609-7_13
- Secretaría de Educación Pública. (2017). *Ruta para la implementación del modelo educativo* (Segunda ed.). Ciudad de México, México: SEP. Recuperado el 4 de Julio de 2019, de https://www.gob.mx/cms/uploads/attachment/file/207248/10_Ruta_de_implementacion_del_modelo_educativo_DIGITAL_re_FINAL_2017.pdf
- Sein-Echaluce, M. L., Fidalgo-Blanco, Á., & García-Peñalvo, F. J. (2015). Metodología de enseñanza inversa apoyada en b-learning y gestión del conocimiento. *III Congreso Internacional sobre Aprendizaje, Innovación y Competitividad (CINAIC 2015)*. Madrid, España. Obtenido de <https://repositorio.grial.eu/bitstream/grial/480/1/FlipTeaching.pdf>
- Selby, C. C. (2015). Relationships: computational thinking, pedagogy of programming, and Bloom's Taxonomy. *WiPSCe '15 Proceedings of the Workshop in Primary and Secondary Computing Education* (págs. 80-87). London, United Kingdom: ACM New York, NY, USA.
doi:10.1145/2818314.2818315
- Sentance, S., & Csizmadia, A. (2017). Computing in the curriculum: Challenges and strategies from a teacher's perspective. *Education and Information Technologies*, 22(2), 469-495.
doi:10.1007/s10639-016-9482-0
- Shi, N., Zhang, P., & Sun, X. (2016). B-Learning on Novice Programmer Learning with Roles of Variables. *International Journal of Simulation Systems, Science & Technology*, 17(32), 38.1-38.7. Obtenido de <http://ijssst.info/Vol-17/No-32/paper38.pdf>
- Shiflet, G. W., & Shifleta, A. B. (2012). Introducing Life Science Doctoral Students in Oz to the Wizardry of Computational Modeling: Introducing Computational Thinking with CellDesigner™. *Procedia Computer Science*, 9, 1753-1762. doi:10.1016/j.procs.2012.04.193

- Shuptrine, C. (2013). Improving college and career readiness through Challenge-Based Learning. *Contemporary Issues in Education Research*, 6, 181–189.
- SIES. (2019). *Servicio de Información de la Educación Superior*. Obtenido de Metodología datos publicados en buscador de instituciones, www.mifuturo.cl 2018-2019: <http://portales.inacap.cl/transparencia/>
- Skinner, B. F. (1984). The shame of American education. *American Psychologist*, 39(9), 947-954.
- Skinner, E. A., & Belmont, M. J. (1993). Motivation in the classroom: reciprocal effects of teacher behavior and student engagement across the school year. *Journal of Educational Psychology*, 85, 571-581.
- Sousa-Santos, S., Peset-González, M. J., & Muñoz-Sepúlveda, J. (2017). La metodología Flipped Classroom en la enseñanza híbrida universitaria: la satisfacción de los estudiantes. *IV Congreso Internacional sobre Aprendizaje, Innovación y Competitividad-CINAIC (Zaragoza 4-6 Octubre 2017)*. doi:10.26754/CINAIC.2017.000001_063
- Staker, H., & Horn, M. B. (2012). *Classifying K–12 blended learning*. Obtenido de Innosight Institute: <https://www.christenseninstitute.org/wp-content/uploads/2013/04/Classifying-K-12-blended-learning.pdf>
- Stufflebeam, D. (2011). Meta-evaluation. *Journal of Multidisciplinary Evaluation*, 7(15), 99-158.
- Sun, N., Li, K., & Zhu, X. (2016). Action Research on Visualization Learning of Mathematical Concepts Under Personalized Education Idea: Take Learning of Geometrical Concepts of Elementary Math for Example. *Blended Learning: Aligning Theory with Practices. ICBL 2016. Lecture Notes in Computer Science*. 78, págs. 348-359. Cham: Springer. doi:10.1007/978-3-319-41165-1_31
- Surendeleg, G., Murwa, V., Yun, H.-K., & Kim, Y. S. (2014). The role of gamification in education—a literature review. *Contemporary Engineering Sciences*, 7(29), 1609-1616. doi:10.12988/ces.2014.411217
- Swaid, S. I. (2015). Bringing Computational Thinking to STEM Education. *Procedia Manufacturing*, 3, 3657-3662. doi:10.1016/j.promfg.2015.07.761
- TACCLE 3 Consortium. (2017). *TACCLE 3: Coding Erasmus + Project website*. Obtenido de <http://www.taccle3.eu/>
- Talent Search. (2015). *Elite: Grade 12+, Institute of IT Professionals South Africa*. Obtenido de <http://www.olympiad.org.za>
- Tejeda-Lorente, Á., Bernabé-Moreno, J., Porcel, C., Galindo-Moreno, P., & Herrera-Viedma, E. (2015). A Dynamic Recommender System as Reinforcement for Personalized Education by

Referencias

- a Fuzzly Linguistic Web System. *Procedia Computer Science*, 55, 1143-1150.
doi:10.1016/j.procs.2015.07.084
- Tekin, C., Braun, J., & Van-Der-Schaar, M. (2015). eTutor: Online learning for personalized education. *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (págs. 5545-5549). Brisbane, QLD, Australia: IEEE.
doi:10.1109/ICASSP.2015.7179032
- Tirado-Morueta, R., Aguaded-Gomez, I., & Hernando-Gómez, Á. (2014). Adoption of B-learning at universities in Spain: The influence of environment and personal factors. *Educational Technology Use and Design for Improved Learning Opportunities*, 286-310.
doi:10.4018/978-1-4666-6102-8.ch016
- Topirceanu, A. (2017). Gamified learning: A role-playing approach to increase student in-class motivation. *Procedia Computer Science*, 112, 41-50. doi:10.1016/j.procs.2017.08.017
- Torres, J. (2006). *La desmotivación del profesorado*. Madrid: Morata.
- Tsai, M.-J., Huang, L.-J., Hou, H.-T., Hsu, C.-Y., & Chiou, G.-L. (2016). Visual behavior, flow and achievement in game-based learning. *Computers & Education*, 98, 115-129.
doi:10.1016/j.compedu.2016.03.011
- UK Bebras Computational Thinking Challenge. (2015). *answers (2015)*, University of Oxford.
Obtenido de <http://www.bebas.org>
- Universidad del Valle de México. (2019). *Aprendizaje para toda la vida*. Ciudad de México: Centro de opinión pública. Recuperado el 4 de Julio de 2019, de <https://opinionpublicauvm.mx/sites/default/files/presentaciones/Aprendizaje%20para%20toda%20la%20vida%20vf.pdf>
- Urra, E., & Barría, R. M. (2010). La revisión sistemática y su relación con la práctica basada en la evidencia en salud. *Revista Latino-Americana de Enfermagem*, 18(4), 824-831.
doi:10.1590/S0104-11692010000400023
- Villalustre, L., & Del Moral, M. (2015). Gamification: Strategies to optimize learning process and the acquisition of skills in university contexts. *Digital Education Review*, 27, 13-31.
Obtenido de http://revistes.ub.edu/index.php/der/article/view/12486/pdf_1
- Villegas-Ch, W., & Luján-Mora, S. (2017). Analysis of data mining techniques applied to LMS for personalized education. *2017 IEEE World Engineering Education Conference (EDUNINE)* (págs. 85-89). Santos, Brazil, 19-22 March 2017: IEEE. doi:10.1109/EDUNINE.2017.7918188
- Walker, H. M. (2015). Computational thinking in a non-majors CS course requires a programming component. *ACM Inroads*, 6(1), 58-61. doi:10.1145/2727126

- Weese, J. L. (2016). Mixed Methods for the Assessment and Incorporation of Computational Thinking in K-12 and Higher Education. *ICER '16 Proceedings of the 2016 ACM Conference on International Computing Education Research* (págs. 279-280). Melbourne, VIC, Australia — September 08 - 12: ACM New York, NY, USA. doi:10.1145/2960310.2960347
- Werbach, K., & Hunter, D. (2015). *The gamification toolkit*. Philadelphia, USA: Wharton Digital Press.
- Williams, I. (2015). *Gamification as a content marketing tactic*. Obtenido de Smart Insights: <https://www.smartinsights.com/content-management/content-marketing-creative-and-formats/gamification-as-a-content-marketing-tactic/>
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33-35. doi:10.1145/1118178.1118215
- Wing, J. M. (2011). *Research Notebook: Computational Thinking--What and Why?* Obtenido de The Link, The magazine of the Carnegie Mellon University School of Computer Science: <https://www.cs.cmu.edu/link/research-notebook-computational-thinking-what-and-why>
- Wong, G. K., Cheung, H. Y., Ching, E. C., & Huen, J. M. (2015). School perceptions of coding education in K-12: A large scale quantitative study to inform innovative practices. *2015 IEEE International Conference on Teaching, Assessment, and Learning for Engineering (TALE)* (pág. art. no. 7386007). Zhuhai, China: IEEE. doi:10.1109/TALE.2015.7386007
- Worrell, B., Brand, C., & Repenning, A. (2015). Collaboration and Computational Thinking: A classroom structure. *2015 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)* (págs. 183-187). Atlanta, GA, USA: IEEE. doi:10.1109/VLHCC.2015.7357215
- Xia, A. (2016). On the Basis of the Program Design Teaching and Research of Cultivation of Computational Thinking Ability. *Proceedings of the 2016 2nd International Conference on Social Science and Higher Education*. Atlantis Press. doi:10.2991/icshe-16.2016.83
- Yadav, A., Mayfield, C., Zhou, N., Hambrusch, S., & Korb, J. T. (2014). Computational Thinking in Elementary and Secondary Teacher Education. *ACM Transactions on Computing Education (TOCE)*, 14(1), Article No. 5. doi:10.1145/2576872
- Yildirim, I. (2017). Students' perceptions about gamification of education: A Q-method analysis. *Egitim ve Bilim*, 42(191), 235-246. doi:10.15390/EB.2017.6970
- Ying, L., & Pingping, L. (2017). Research on the teaching of programming language based on Computational Thinking. *Proceedings of the 2017 International Conference on Social science, Education and Humanities Research (ICSEHR 2017)*. Atlantis Press. doi:10.2991/icsehr-17.2017.17

Referencias

- Yinnan, Z., & Chaosheng, L. (2012). Training for computational thinking capability on programming language teaching. *2012 7th International Conference on Computer Science & Education (ICCSE)* (págs. 1804-1809). Melbourne, VIC, Australia : IEEE.
doi:10.1109/ICCSE.2012.6295420
- Zapata-Ros, M. (2015). Pensamiento computacional: Una nueva alfabetización digital. *Revista De Educación a Distancia*, 46. Obtenido de <https://revistas.um.es/red/article/view/240321>
- Zeng, X. L. (2015). Educational exploration based on computational thinking capacity cultivating. *Computing, Control, Information and Education Engineering: Proceedings of the 2015 Second International Conference on Computer, Intelligent and Education Technology (CICET 2015)*, (págs. 895-898). Guilin, P.R. China.
- Zhang, C., Chen, X., & Li, J. (2011). Research of VB programming teaching mode based on the core of computational thinking ability training. *2011 6th International Conference on Computer Science & Education (ICCSE)* (págs. 1260-1263). Singapore, Singapore: IEEE.
doi:10.1109/ICCSE.2011.6028861
- Zhao, F.-q. (2016). Personalized Education Approaches for Chemical Engineering and Relevant Majors. *2016 The 3rd International Conference on Industrial Engineering and Applications (ICIEA 2016) MATEC Web of Conferences*, 68, pág. art. no. 20003.
doi:10.1051/mateconf/20166820003
- Zhi-Mei, C., & Xiang, L. (2016). The PBL teaching method research based on computational thinking in C programming. *2nd International Conference on Modern Education and Social Science*, (págs. 405-409). doi:10.12783/dtssehs/mess2016/9624
- Zhong, B., Wang, Q., Chen, J., & Li, Y. (2016). An exploration of three-dimensional integrated assessment for computational thinking. *Journal of Educational Computing Research*, 53(4), 562-590. doi:10.1177/0735633115608444