

Gradually evolution towards competence development in a Programming course.

Fabiola María Montoya Ramírez, Máster¹

¹Universidad Latina de Costa Rica, Costa Rica, fabiola.montoya@ulatina.cr

Abstract– The competence development in the university is so essential to face a future professional, it is necessary to make a chance inside the classroom to get it. In a programming course of Latina de Costa Rica University, is incorporated an authentic evaluation and instructional design. The above with the purpose to influence positive to the all students that enroll the course, for logic development and problems resolution.

Keywords-- Evaluation, Programming, Engineering, Education, Syllabus.

Digital Object Identifier (DOI):
<http://dx.doi.org/10.18687/LACCEI2020.1.1.190>
ISBN: 978-958-52071-4-1 ISSN: 2414-6390

Evolucionando paulatinamente hacia un desarrollo por competencias en un curso de Programación.

Gradually evolution towards competence development in a Programming course.

Fabiola María Montoya Ramírez, Máster¹

¹Universidad Latina de Costa Rica, Costa Rica, fabiola.montoya@ulatina.cr

Resumen– *El desarrollo de competencias en la universidad es indispensable para afrontar un futuro profesional, por lo que es necesario hacer cambios dentro del aula para lograrlo. En un curso de programación de la Universidad Latina de Costa Rica, se realizan cambios al incluir una evaluación más auténtica y un diseño instruccional. Lo anterior con el propósito de influenciar positivamente a los estudiantes que matriculan el curso para el desarrollo de la lógica y resolución de problemas.*

Palabras Clave– *Evaluación, Programación, Ingeniería, Educación, Syllabus.*

Abstract– *The competence development in the university is so essential to face a future professional, it is necessary to make a change inside the classroom to get it. In a programming course of Latina de Costa Rica University, is incorporated an authentic evaluation and instructional design. The above with the purpose to influence positive to the all students that enroll the course, for logic development and problems resolution.*

Keywords– *Evaluation, Programming, Engineering, Education, Syllabus.*

I. INTRODUCCIÓN

Para nadie es un secreto que la mayoría de los docentes universitarios en ingeniería y tecnologías de información no tienen una formación en docencia. Esto implica que el docente tenga un proceso de evolución para ir desarrollando aspectos relacionados con la metodología, evaluación y planificación dentro del aula.

La forma como se enseña es importante, pero sino se tiene claro la evaluación de los aprendizajes y el diseño de la clase, probablemente no se logró medir correctamente el desarrollo de competencias en los estudiantes.

Ref. [1] indica que “no existen competencias sin conocimientos; los saberes son esenciales, aquéllas no se desarrollan en el vacío. Toda competencia implica una movilización de saberes. Si alguien posee conocimientos sobre un asunto, pero es incapaz de usarlos, se dice que no es competente. En cambio, alguien sí lo es cuando “aprende a hacer” algo y, además, es consciente de “por qué y para qué” lo hace (aprende a conocer) y puede comunicarlo de modo efectivo”.

El Reglamento de Evaluación de los Aprendizajes de Ref. [2], en el artículo 3 del capítulo I, define la evaluación de los aprendizajes como un proceso continuo cuyo propósito es la recopilación de información cuantitativa y cualitativa del

estudiante, para fomentar la enseñanza y aprendizaje en el aula.

La evaluación no es un proceso que se realiza en determinado momento, es algo continuo que refleja los avances y posibles limitaciones de los estudiantes.

En Ref. [3] se menciona que la evaluación es parte de la enseñanza y por esto debe tomarse en cuenta en la planificación de un curso y así establecer el mejor momento para recoger información del estudiante y realizar retroalimentación para contribuir al logro de aprendizajes.

En la literatura se menciona que la evaluación puede ser diagnóstica, formativa y sumativa. Ref. [4] define estos 3 tipos de evaluación de la siguiente manera:

- Diagnóstica: permite medir el nivel de conocimiento de los estudiantes al inicio de un curso o un tema. Esta evaluación se puede comparar con la prueba final, para determinar el avance de un estudiante.
- Formativa: para dar seguimiento y orientación al docente del avance que tienen los estudiantes.
- Sumativa: la más conocida y consiste en otorgar una calificación final a los estudiantes, asociada con el aprendizaje.

Ref.[5] relaciona la evaluación con las competencias del perfil de salida de una carrera por medio del término evaluación auténtica. Menciona que la evaluación auténtica es un proceso formal e informal de recolección de información sobre las competencias que se desarrollan en los estudiantes a partir de instrumentos basados en situaciones reales.

Ref.[6] incluye 2 aspectos en los que se basa la evaluación autentica:

- Ambientes estimulantes para evaluar el aprendizaje de los estudiantes.
- Tener un enfoque proactivo y provechoso de la evaluación de aprendizajes.

Ref.[7] explica que la evaluación auténtica es una “reacción ante una tradición pedagógica muy extendida en la que el alumno se limita a escuchar, repetir, copiar y memorizar”. Esto hace referencia que la evaluación suele ser sumativa y se trata al estudiante como un número según resultados obtenidos.

La evaluación tradicional fomenta que el estudiante repita lo que dice y hace el profesor, obteniendo una nota como resultado de su capacidad de retención.

En cambio, la evaluación auténtica permite desarrollar habilidades en el estudiante necesarias para su futuro trabajo.

Digital Object Identifier (DOI):
<http://dx.doi.org/10.18687/LACCEI2020.1.1.190>
ISBN: 978-958-52071-4-1 ISSN: 2414-6390

Para lograr esto, se realizan actividades que incluyan: resolución de problemas, creatividad, trabajo con los demás, aprender de los errores, toma de decisiones, entre otras.

Además de las habilidades genéricas, la evaluación auténtica permite a los estudiantes desarrollar habilidades propias del área, por ejemplo: la habilidad para trabajar con la computadora, con el software Visual Studio para realizar programaciones, etc.

Al incluir actividades auténticas, se logra una evaluación que propicie el desarrollo de habilidades y competencias.

Ref. [8] presenta una lista de métodos de enseñanza que propician el desarrollo de competencias. Algunos de estos métodos son: exposición, diálogo y argumentación, discusión, juego de roles, trabajo en equipo, estudio de casos, método de proyectos y aprendizaje basado en problemas.

Las instituciones educativas necesitan adecuar la educación y darle más importancia a las competencias para que los perfiles de salida de las carreras satisfagan las necesidades del entorno, al contar con profesionales que puedan adaptarse a este. Ref. [9].

Para lograrlo se necesita un buen diseño del curso, tal como lo menciona Ref. [10] el syllabus es una herramienta de planificación.

Ref. [10] indica que un enfoque de competencias requiere una alineación entre el perfil de salida, el plan de estudios, las acciones para lograrlo y su evaluación. También menciona una definición de Ref. [11] con respecto a lo que es el syllabus: “herramienta...-orientada al desarrollo de proyectos formativos, a organizar nuestra actuación no como un conjunto de acciones imprevisibles y desconectadas entre sí, sino como la puesta en práctica de un plan bien pensado y articulado”.

Es común que se confunda el plan de un curso y el syllabus, pero ambos son diferentes. El plan de curso es el documento que se entrega al docente y tiene una visión general del curso, en cambio el syllabus puede ser desarrollado por el docente/equipo de trabajo por lo que es más específico, Ref. [12] indica que este formato “pretende motivar y guiar al estudiante indicando con claridad aspectos e hitos básicos del curso, tales como evaluaciones, actividades, contenidos, lecturas y reglas del juego”.

Ref. [12] agrega que un syllabus debe ser: “coherente, apuntar al logro del aprendizaje activo, monitorear y evidenciar los aprendizajes”.

Existen muchos modelos para el diseño instruccional de un curso. Para obtener resultados óptimos se siguen fases como análisis, diseño, desarrollo, implementación y evaluación. Ref. [13]

Son varios los aspectos que se deben contemplar para que un curso desarrolle competencias, pero indudablemente la relación entre la evaluación y el diseño del curso es esencial para la educación universitaria de hoy en día.

A. Objetivo

Proponer un cambio en un curso de Programación de la Facultad de Ingeniería y TI, sede Heredia, por medio de la

evaluación auténtica y diseño instruccional, que le permita al estudiante desarrollar las competencias necesarias para su perfil de salida.

B. Metodología

Hi: El rendimiento académico y satisfacción de los estudiantes de programación, se ve afectado positivamente por la incorporación de una evaluación auténtica y syllabus del curso.

En este trabajo se utiliza un diseño con un enfoque mixto CUALITATIVA+CUANTITATIVA (simultánea), por lo que el diseño es de triangulación concurrente. Esto basado en la metodología de Ref. [14] que consiste en 8 pasos:

- Idea
- Planteamiento del problema
- Revisión de la literatura
- Hipótesis cuantitativa
- Diseño de investigación
- Recolección de datos cuantitativa y cualitativa
- Análisis de datos cuantitativa y cualitativa
- Interpretación

Se recopilan los resultados de los periodos 2019-01 y 2019-02 de un curso de programación en dos grupos: G1 que es el grupo control (no tiene cambios) y el G3 tiene cambios en la evaluación e incorporación del syllabus.

Importante indicar que el grupo G3 ha tenido cambios metodológicos con la incorporación de metodologías activas en los periodos 2018-02 y 2018-03. Además, se ha profundizado aún más en la metodología de aprendizaje activo (en gran medida el aprendizaje basado en problemas) y clase invertida, aumentando la cantidad de actividades en el 2019 en comparación al 2018.

Con respecto a los datos cuantitativos se incluyen: los resultados de la evaluación docente, notas del curso y encuesta de satisfacción.

Para los datos cualitativos se utiliza la observación, cuestionario previo y encuesta de satisfacción en los grupos ya mencionados.

La población son los estudiantes de la Universidad Latina de Costa Rica matriculados en el curso de programación 1, en la sede Heredia para las carreras de Ingeniería Industrial, Ingeniería Civil e Ingeniería Electrónica.

La muestra es no probabilística de tipo por conveniencia, ya que se incluyen los estudiantes matriculados en los grupos. Ambos grupos son impartidos por el mismo profesor lo que disminuye la variabilidad en los resultados. Estos grupos presentan 94 estudiantes:

TABLA 1.
DISTRIBUCIÓN DE LOS ESTUDIANTES DE PROGRAMACIÓN

2019-01		2019-02	
Grupo1	Grupo3	Grupo1	Grupo3
25	23	26	20

El cuestionario se aplica el primer día de clases, con el propósito de conocer más del estudiante y adaptar la clase según necesidades. Al ser el curso de programación compartido por varias carreras, el docente desconoce la distribución de este.

El cuestionario está distribuido en 2 secciones, la primera solicita datos principales del estudiante como:

- Nombre completo del estudiante (pregunta abierta)
- Correo electrónico (pregunta abierta)
- Carrera (selección entre industrial, civil y electrónica)

La segunda parte se enfoca en preguntas relacionadas con el conocimiento y otras sobre el aprendizaje:

- ¿Ha programado en algún lenguaje de programación? (selección entre sí o no)
- Si en la pregunta anterior, su respuesta fue sí, indicar los lenguajes que ha programado (pregunta abierta)
- ¿Cómo le gusta aprender? (pregunta abierta)
- ¿Qué lo motiva a estudiar? (pregunta abierta)

Para medir la satisfacción del estudiante, se utiliza la encuesta de satisfacción propuesta por Ref.[15] para ser aplicada al final del cuatrimestre en ambos grupos. Esta encuesta contiene 8 preguntas Likert de 10 y 3 preguntas abiertas:

1. En general, está satisfecho con el curso.
2. La metodología de trabajo aplicada ha sido adecuada.
3. Las actividades aplicadas en el curso han sido adecuadas.
4. La documentación y materiales han sido adecuados para el desarrollo del curso.
5. Los medios y recursos didácticos utilizados han sido los adecuados.
6. Las instalaciones físicas han facilitado el desarrollo de las actividades.
7. El curso ha contribuido al desarrollo de la lógica para resolver problemas.
8. Tomando en cuenta la metodología aplicada, recomendaría este curso a otro compañero.
9. Mencione un aspecto que le gustó de este curso
10. Mencione un aspecto que le podría incluir a este curso
11. Comentarios

Para validar el cuestionario sobre satisfacción de los estudiantes, se usa el coeficiente Alfa de Cronbach.

Se incluye el análisis de resultados de la encuesta de evaluación del desempeño docente de la universidad. Las preguntas son de tipo Likert de 10 y contiene una pregunta abierta:

1. El docente lleva preparada la clase
2. El docente me brinda un trato respetuoso
3. El docente me evalúa de acuerdo con la materia que he visto en clase
4. El docente me explica el programa de curso detalladamente en la primera clase

5. El docente muestra conocimiento y dominio de la materia del curso
6. El docente relaciona la materia del curso con la profesión de mi carrera
7. El docente usa el aula virtual
8. El docente usa herramientas tecnológicas adecuadas para el curso
9. El docente utiliza una metodología de enseñanza que me ayuda para comprender la materia
10. El docente valora mis conocimientos adquiridos en cursos anteriores
11. Comentarios

Para el diseño del syllabus se utiliza la propuesta presentada por Ref.[10]. En la Fig.1 se muestran los componentes para dicho diseño:

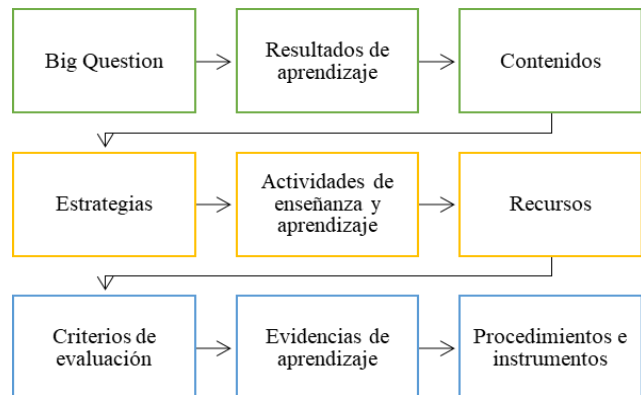


Fig. 1 Componentes para un diseño de syllabus.

Antes de desarrollar los componentes anteriores, hay que tener claro cuáles son las competencias (o parte de ellas) que deben desarrollarse en el curso de programación. Como en el programa de curso no viene definido dichas competencias, se extraen del perfil de egreso de la carrera:

- Desarrollo y análisis de sistemas de información gerencial.
- Desarrollar programas informáticos que ayuden al proceso de automatización y simulación industrial.

Es necesario mencionar que no es común el uso del syllabus en la universidad donde se realiza este estudio. Para responder a cada apartado se utiliza el programa del curso actual de programación:

TABLA 2.
DISEÑO DEL SYLLABUS PARA PROGRAMACION.

	Situación del programa	Situación Mejorada
Big Question	¿De qué manera es posible resolver problemas estructurados de ingeniería por medio de lenguajes de programación? <i>*Se especula dentro del programa, pero no está claramente establecido. *</i>	¿De qué manera es posible generar programas informáticos que ayuden al proceso de automatización y simulación industrial, utilizando la programación estructurada?
Resultados	1. Diagramas de flujo	1. <i>Construye</i> diagramas de

de aprendizaje	2. Interfaz de usuario 3. Aplicaciones informáticas que utilicen estructuras de programación 4. Bases de datos *Se cuentan con contenidos y objetivos, pero no con resultados de aprendizaje claramente establecidos. Es un enfoque tradicional (basado en la enseñanza) *	flujo, a partir de la descripción de una situación, para resolver un problema. 2. Elige controles de interfaz de usuario, según las entradas y salidas de un diagrama de flujo, para montar adecuadamente la funcionalidad de una aplicación informática. 3. Diseña aplicaciones informáticas que utilicen las estructuras de programación, a partir de los diagramas de flujo, según los requerimientos de un problema a sistematizar. 4. Utiliza códigos para la conexión con una base de datos sencilla y de pequeño tamaño, con el objetivo de manipular datos dentro de una aplicación informática.
Criterios de evaluación	Se realiza una evaluación sumativa, según lo obtenido en tareas, exámenes y proyecto.	Reconoce la simbología que se utiliza en la construcción de diagramas de flujo. Interpreta la descripción de un problema para ser traducido en un diagrama de flujo. Relaciona los diagramas de flujo con su respectiva programación. Reconoce las estructuras de programación. Soluciona problemas por medio de la programación, ya sean estructuras secuenciales, condicionales, cíclicas o una mezcla de todas. Soluciona problemas que requiera el uso de una base de datos sencilla y de pequeño tamaño.
Evidencias de aprendizaje	No se aplica	Diagramas de flujo realizados. Resultados de Actividades en línea. Programa creado en Visual Studio.
Actividades de aprendizaje	Ejercicios	Descubrir el algoritmo. Repaso con mentimeter. Comprobación de lectura con kahoot. Actividades en línea con educaplay. Ejercicios en clase individuales, pero con colaboración entre todos. Ejercicios en grupos llamados dinámicas (incluye un torneo de programación)

Los resultados de aprendizaje se analizaron utilizando la fórmula **VERBO + OBJETO + CONDICIÓN + FINALIDAD** para transformar un programa basado en un enfoque tradicional en uno orientado a competencias. Con respecto a los verbos se utiliza la taxonomía Bloom-Anderson para indicar por cada objeto el nivel que se espera que el estudiante logre al final del curso de programación. Los objetos se centran en el nivel 3 de aplicación (interfaz de

usuario y bases de datos) y el 6 de crear (diagramas de flujo, aplicaciones informáticas).

Con respecto a la evaluación el programa del curso presenta los siguientes rubros y su respectiva distribución:

- Tareas: 15%
- Prueba I: 20%
- Prueba II: 20%
- Prueba final: 25%
- Proyectos: 20%

Efectivamente la evaluación es 100% sumativa, con un 65% de la nota enfocada en exámenes. No se hace evaluación del trabajo en clase.

Con objetivo de ir hacia una evaluación auténtica, se considera la evaluación de la siguiente manera:

Sumativa:

- Tareas programadas: 10%
- Actividades en línea: 10%
- Dinámicas en clase: 10%
- Comprobación pre-clase: 10%
- Prueba I: 20%
- Prueba II: 20%
- Proyecto: 20%

Formativa:

- Los estudiantes que obtengan menos del 55% de preguntas correctas en la comprobación de lectura con kahoot, podrán repetir la comprobación en Moodle.
- Las actividades en línea pueden realizarse las veces que sean necesarias.
- Se da retroalimentación en las dinámicas realizadas en clase.

A pesar de que el objetivo de las actividades en línea, comprobaciones y dinámicas es de un enfoque más formativo, si tienen un porcentaje asignado.

En el syllabus se incorporan rúbricas de evaluación para cada actividad con el propósito de fomentar la evaluación y desglosar aspectos como: trabajo en equipo, entregables, resolución de problemas, entre otros.

Para el montaje del syllabus se utiliza el formato propuesto por Ref. [10] con variaciones para adaptarlo a la situación del curso:

I. IDENTIFICACIÓN DEL CURSO

- Código y nombre del curso
- Créditos
- Número de sesiones por semana
- Horas teóricas/prácticas
- Horas individuales
- Horas totales semanales
- Carreras
- Tipo de curso
- Docente
- Año

- Ciclo lectivo
- Modalidad
- Requisitos
- Lugar

II. DESCRIPCIÓN DEL CURSO

- Descripción e intención formativa del curso
- Competencias del perfil de egreso que se favorecen en su desarrollo
- Gran pregunta (BQ) que se abordará y que articulará el curso con el resto de la formación
- Metodología
- Evaluación

III. RESULTADOS DE APRENDIZAJE

- Resultado de aprendizaje
- Criterios de evaluación
- Evidencias de Aprendizaje
- Contenidos de aprendizaje
- Semana
- Actividad(es) de aprendizaje
- Recursos
- Modalidad

IV. CRONOGRAMA DE ACTIVIDADES DETALLADO

V. RÚBRICAS

TABLA 3.
EXTRACTO DE PROPUESTA GENERADA PARA EL
RESULTADO DE APRENDIZAJE 1.

Resultado de aprendizaje 1	Construye diagramas de flujo, a partir de la descripción de una situación, para resolver un problema.		
Criterios de evaluación	Reconoce la simbología que se utiliza en la construcción de diagramas de flujo. Interpreta la descripción de un problema para ser traducido en un diagrama de flujo.		
Evidencias de Aprendizaje	Diagramas de flujo realizados Resultados de Actividades en línea		
Contenidos de aprendizaje	<ul style="list-style-type: none"> • Conceptos: algoritmo, diagrama de flujo • Componentes de un diagrama de flujo de datos: Inicio/Fin, Proceso, Flujo, Variable, Operadores, Entrada/Salida, Decisión, Ciclo. • Diagramación: Representar la resolución de problemas básicos utilizando diagramas de flujo. 		
Semana	Actividad(es) de aprendizaje	Recursos	Modalidad
1	Descubrir el algoritmo secuencial	Computadora, internet, presentación, tijeras, goma, papel, lapiceros, instrucciones.	Trabajo en grupo
2	Repaso con mentimeter Ejercicios sobre diagramas de flujo, Blockly games	Computadora, internet, papel, lapiceros, presentación	Individual

Para el grupo control (G1) se utiliza el programa oficial del curso de programación y para el grupo modificado (G3) el syllabus propuesto y evaluación auténtica.

Se incluye como parte final a este estudio la observación para incluir aspectos más cualitativos, ya que los elementos anteriores se enfocan en gran parte en un enfoque cuantitativo.

II. RESULTADOS OBTENIDOS

Los resultados al incorporar la evaluación auténtica y diseño instruccional en un curso de programación se dividen

en 4 aspectos: cuestionario previo, calificaciones, encuesta de satisfacción y evaluación docente.

A. Cuestionario previo

En el cuestionario se logra observar que la mayoría de los estudiantes no tiene experiencia previa con algún lenguaje de programación. En la Fig.2 se muestra el detalle:

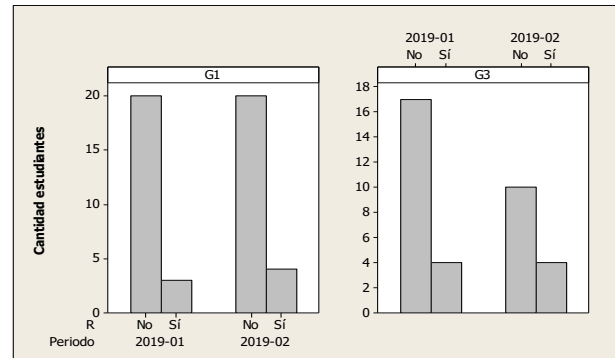


Fig. 2 Cantidad de estudiantes según si han o no programado, por periodo en el curso de programación 1, grupo 1 y 3.

Los que indican que han utilizado algún lenguaje de programación, mencionan en su mayoría Visual Basic.Net, lenguaje oficial del curso. Muchos de estos son estudiantes repitentes. Otros pocos mencionan C++, Python, Rockwell y Zelio.

La gran mayoría les gusta aprender practicando. Otros en menor medida indican que con actividades interactivas y visualmente.

B. Calificaciones de los estudiantes

Con respecto a los resultados de las calificaciones, se muestra el histórico desde el 2016-1 hasta 2019-02 para ambos grupos:

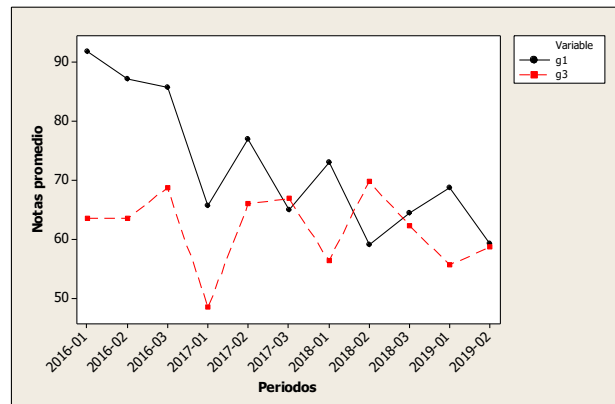


Fig. 3 Histórico de las calificaciones de los estudiantes por periodo en el curso de programación 1, grupo 1 y 3.

En la Fig.4 se muestra el histórico con relación a la cantidad de reprobados por cada grupo desde el 2016-01 hasta el 2019-02:

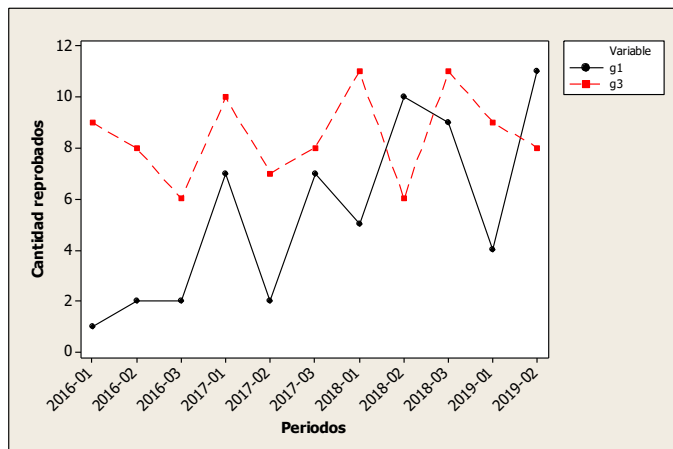


Fig. 4 Histórico de estudiantes reprobados por periodo en el curso de programación 1, grupo 1 y 3.

Propiamente en el 2019-01, los resultados se desglosan a continuación:

- El 51% de las tareas no fueron entregadas por los estudiantes.
- El 37% de las actividades en línea no fueron realizadas por los estudiantes.
- El 20% de las comprensiones de lectura no fueron realizadas por los estudiantes.
- El 17% de las dinámicas no fueron realizadas por los estudiantes.
- El promedio de las actividades en línea fue de 53. Del total, 12 estudiantes tuvieron una nota promedio inferior a 70 (52%).
- El promedio de las comprensiones fue de 48. Del total, 18 estudiantes tuvieron una nota promedio inferior a 70 (78%)
- Para la comprobación de lectura 1, 16 estudiantes no realizaron la segunda oportunidad (70%). El 75% de quienes lo hacen, mejoran la nota.
- Para la comprobación de lectura 2, 16 estudiantes no realizaron la segunda oportunidad (70%). El 43% de quienes lo hacen, mejoran la nota.
- Para la comprobación de lectura 3, 14 estudiantes no realizaron la segunda oportunidad (61%). El 67% de quienes lo hacen, mejoran la nota.
- Para la comprobación de lectura 4, 20 estudiantes no realizaron la segunda oportunidad (87%), El 67% de quienes lo hacen, mejoran la nota.

Propiamente en el 2019-02, los resultados se desglosan a continuación:

- El 45% de las tareas no fueron entregadas por los estudiantes.

- El 28% de las actividades en línea no fueron realizadas por los estudiantes.
- El 26% de las comprensiones de lectura no fueron realizadas por los estudiantes.
- El 15% de las dinámicas no fueron realizadas por los estudiantes.
- El promedio de las actividades en línea fue de 56. Del total, 10 estudiantes tuvieron una nota promedio inferior a 70 (50%).
- El promedio de las comprensiones fue de 48. Del total, 11 estudiantes tuvieron una nota inferior a 70 (50%)
- Para la comprobación de lectura 1, 18 estudiantes no realizaron la segunda oportunidad (90%).
- Para la comprobación de lectura 2, 19 estudiantes no realizaron la segunda oportunidad (95%).
- Para la comprobación de lectura 3, 15 estudiantes no realizaron la segunda oportunidad (75%).
- Para la comprobación de lectura 4, 20 estudiantes no realizaron la segunda oportunidad (100%).
- De los que realizaron las comprensiones por segunda ocasión, mejoraron.

D. Encuesta de satisfacción

Desde el cuatrimestre 2018-2 se ha aplicado una encuesta de satisfacción a los estudiantes del curso de programación:

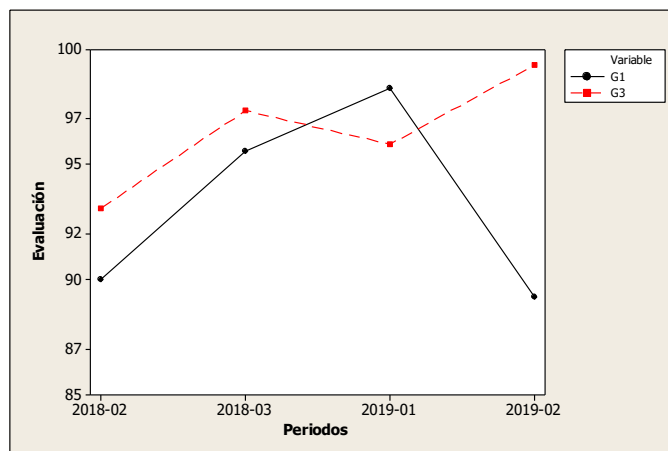


Fig. 5 Resultados encuesta de satisfacción de programación 1, grupo 1 y 3

Con respecto al periodo 2019-1, tenemos lo siguiente:

- Desviación estándar es más alta en el grupo 3.
- En el grupo 3 la pregunta mejor valorada es la correspondiente a la metodología. La más baja la relacionada con el desarrollo de la lógica. Sin embargo, la diferencia es de 0.86.
- En el grupo control (g1), la pregunta mejor valorada es si recomendaría este curso considerando la metodología del curso. La más baja es con respecto a si las instalaciones facilitan las actividades del curso.

- Con respecto a comentarios, únicamente 2 fueron negativos. Estos fueron en el grupo control.
- En el grupo 3 los aspectos que mencionan los estudiantes que les gustó del curso están enfocados en la metodología, haciendo énfasis en las dinámicas/actividades de la clase. Estos comentarios fueron 8 de 14, para un 57% de la clase.
- En el grupo control los comentarios fueron más diversos, pero se observa un enfoque hacia la aplicación del curso.

Con respecto al periodo 2019-2, tenemos lo siguiente:

- Desviación estándar es más alta en el grupo 3.
- En el grupo 3, hubo 3 preguntas con la misma valoración, la cuál fue la más alta posible. Estas fueron las preguntas sobre metodología, satisfacción del curso y si recomendaría el curso. La más baja fue si las actividades habían sido adecuadas. Sin embargo, esta diferencia fue muy baja, de sólo 0.18.
- En el grupo control, la pregunta mejor valorada fue sobre si la documentación y materiales del curso fueron adecuados. La más baja es con respecto a si las instalaciones físicas facilitan las actividades del curso.
- Únicamente 2 comentarios fueron negativos. Estos en el grupo control.
- En el grupo 3, con respecto a los aspectos que a los estudiantes les gustó del curso, 6 de 11 indican que la metodología, lo cual representa un 54.54%.
- En cambio, en el grupo control los aspectos fueron muy diversos.

E. Evaluación del Desempeño Docente

Desde el 2018-02 se han realizado cambios en el grupo 3 con respecto al grupo 1 (grupo control). En la Fig.6 y Fig.7 se muestra el histórico de evaluación del curso desde el 2016-02:

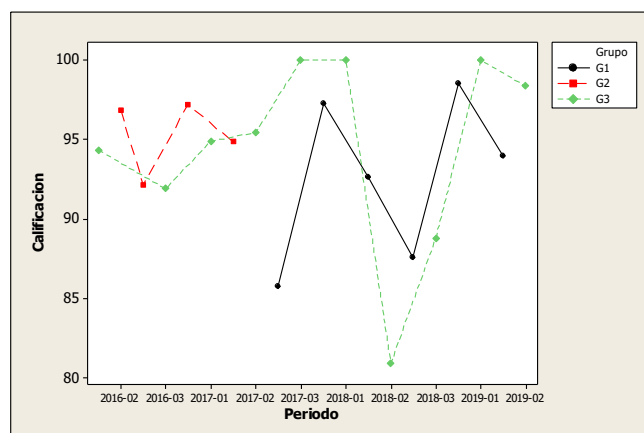


Fig. 6 Histórico de la evaluación docente por periodo en el curso de programación 1, grupo 1, 2 y 3.

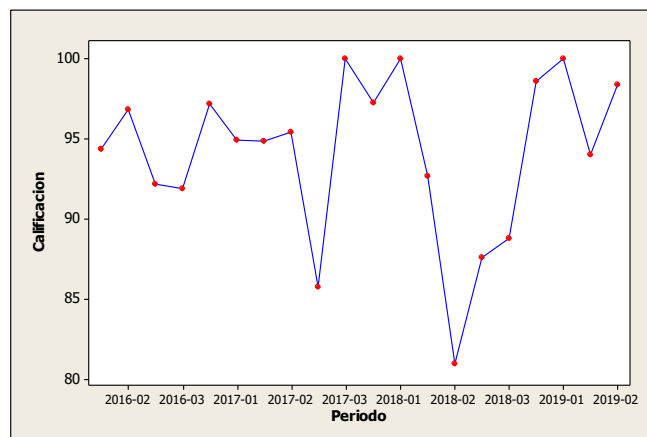


Fig. 7 Histórico de la evaluación docente por periodo en el curso de programación 1, en todos los grupos.

En los siguientes gráficos se destacan 3 preguntas del instrumento de evaluación que se utiliza actualmente en la universidad:

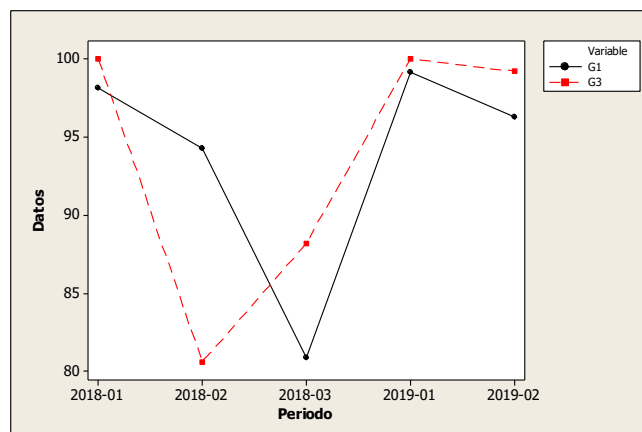


Fig. 8 Histórico de la evaluación docente por periodo en el curso de programación 1 del grupo 1 y grupo 3, con respecto a la pregunta: El docente me evalúa de acuerdo con la materia que he visto en clase.

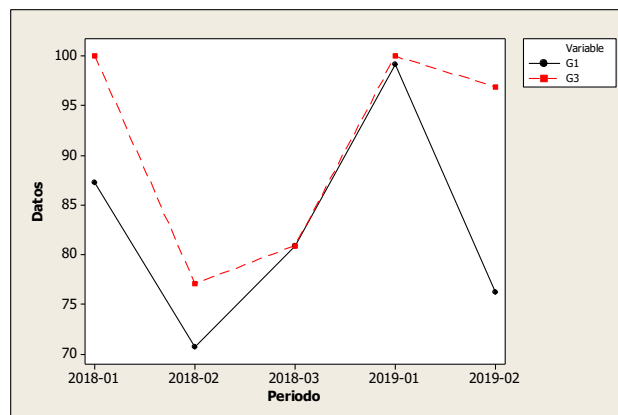


Fig. 9 Histórico de la evaluación docente por periodo en el curso de programación 1 del grupo 1 y grupo 3, con respecto a la pregunta: El docente relaciona la materia del curso con la profesión de mi carrera.

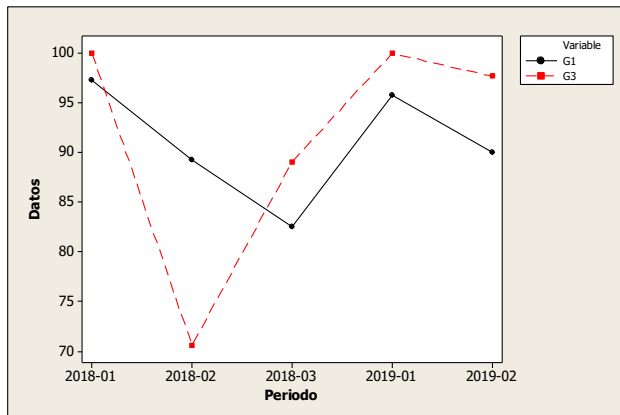


Fig. 10 Histórico de la evaluación docente por periodo en el curso de programación 1 del grupo 1 y grupo 3, con respecto a la pregunta: El docente utiliza una metodología de enseñanza que me ayuda a comprender la materia.

III. ANÁLISIS

El curso de programación tiene como competencia principal el análisis y resolución de problemas, sin embargo, actualmente no existe una definición de competencias en el plan de curso. A pesar de esto y a los requerimientos de los empresarios, es necesario evolucionar poco a poco en el desarrollo de competencias por medio de la inclusión de metodologías.

Tener un syllabus que esté asociado al plan de curso permite detallar aspectos en metodología, actividades y rúbricas de evaluación. Estos aspectos son importantes para el desarrollo de competencias porque permite empoderar al estudiante dentro del aula.

Al incluir otro tipo de acciones fuera del aula como las actividades en línea y las comprobaciones de lectura, le damos oportunidad al estudiante de repasar la materia en aquellos aspectos que son esenciales para el curso, como por ejemplo la estructura de una condición, funciones matemáticas, entre otros.

A pesar de que las actividades en línea se pueden hacer varias veces, son pocos los estudiantes que lo aprovechan. Lo mismo pasa con las comprobaciones de lectura, pocos aprovechan la oportunidad y quienes lo hacen tienden a mejorar su calificación en el rubro. Las preguntas no son siempre las mismas.

Se evidencia que las actividades individuales tienen poca participación: actividades en línea, tareas y comprobaciones.

Lo contrario sucede con las dinámicas, ya que son actividades dentro de la clase y se trabaja en grupos. La principal característica de este rubro es la retroalimentación por parte del docente, incluso se da la oportunidad al estudiante de aprender de sus errores y de sus compañeros. Las dinámicas son el rubro con los mejores resultados.

Con la implementación realizada en los periodos de análisis, no se nota una diferencia significativa en los

resultados finales de los cursos. Sin embargo, se observa una mejora en el manejo de los códigos e involucramiento en la clase.

Se logra que los estudiantes sin previo conocimiento de programación logren el desarrollo de lógica y resolución de problemas. Ellos mismos han mencionado en varias ocasiones su avance, ya que al inicio del curso no creían tener las capacidades de programar.

La mayoría de los estudiantes que llega al final del curso tiene amplias posibilidades de pasarlo. El grupo 3 tiende a tener más reprobados, muchos de estos dejan de asistir en las últimas semanas, por lo que mínimo pierden 40% de la nota.

Otro aspecto es que los estudiantes que han tenido alguna experiencia previa programando, logran en su mayoría terminar el curso. Sin embargo, las notas no suelen ser superiores a 80.

Aunque el cuestionario no es una prueba diagnóstica, le brinda al profesor una idea de las características del estudiante y actuar según sea necesario. Por ejemplo, conocer cuántos estudiantes hay matriculados por carrera permite ajustar los ejercicios con temas apropiados.

Tampoco se visualiza un efecto en la evaluación docente que hace el estudiante. A pesar de que no hay preguntas enfocadas en el syllabus o evaluación auténtica, hay 3 que tienen relación directa con los cambios realizados. Sin embargo, no hay una diferencia significativa en los resultados de estas preguntas. El grupo 3 tiende a ser más alto en los periodos del 2019.

Se observa que los resultados de satisfacción no cambian considerablemente de un grupo a otro, excepto en el último periodo. Los estudiantes mencionan que los cambios a nivel metodológico son positivos.

IV. CONCLUSIONES

Con los cambios incorporados en el grupo 3 de programación, se concluye lo siguiente:

- Una variable importante que influye en los resultados del curso es la diferencia en el horario de los grupos, debido al tipo de estudiante que matricula. El grupo 3 tiene un horario de mañana y el grupo 1 de noche. Normalmente en la noche matriculan los estudiantes que trabajan.
- No hay una diferencia significativa en los resultados cuantitativos, pero si cambios cualitativos como el comportamiento de los estudiantes dentro de la clase y manejo de la sintaxis del código.
- Los estudiantes que no tienen experiencia previa programando, han logrado hacerlo y desarrollado en cierto nivel el analizar un problema y darle una solución. Aunque a este momento no se logre medir la competencia como tal, si se evidencia en la clase.
- Contar con un syllabus con todos los elementos ya mencionados, le permite al docente y estudiante tener claro lo que se pretende con el curso y cómo lograrlo.

- Además de la evidente planificación que conlleva tener un syllabus, genera beneficios al estudiante porque tiene claridad de las actividades que se realicen dentro y fuera de la clase.
- A pesar de que no se aplica una evaluación 100% formativa, se logra aprovechar recursos dentro de la clase para fomentar la participación de los estudiantes y la realización de ejercicios que en su mayoría incorporan una metodología de aprendizaje basado en problemas. Esto fomenta la evaluación auténtica.
- A pesar de que el grupo 1 tiene una metodología no tan estructurada y es más tradicional, los estudiantes están de acuerdo con esto. Normalmente los cursos que llevan los estudiantes tienen esta dirección.
- Los estudiantes no saben lo que es una metodología.
- El torneo de programación incorporado en el año 2019 ha tenido buena recepción por parte de los estudiantes. Además, permite trabajar en aspectos de liderazgo y trabajo en equipo.
- Los cambios implementados han sido positivos, pero no se ven reflejados en los resultados cuantitativos.
- Los resultados pueden mejorar si realmente los estudiantes hicieran las actividades fuera de clase. Por algún motivo que no se ha investigado, es común que falten con los trabajos cuando son extraclase.

El desarrollo de competencias definitivamente es un reto para los docentes y estudiantes, ya que el cambio de paradigma es grande.

Para que el desarrollo de competencias pueda darse, es necesario motivar al estudiante para que se involucre en la clase y perciba la materia como un valor agregado. Esto es justamente lo complicado, buscar la manera para que los estudiantes acepten un curso llamado complementario y no del núcleo de las carreras.

Efectivamente tener un syllabus con el detalle del curso permite al estudiante una mejor organización del tiempo y claridad en los objetivos a cumplir.

Al incluir una evaluación formativa, los estudiantes son más conscientes de su aprendizaje y crean programas porque se logra desarrollar algunas competencias como lógica y resolución de problemas dentro del curso.

Se detecta que a los estudiantes no les cuesta aprender el código, incluso son capaces de entender en gran medida un programa terminado, pero les cuesta hacer el análisis desde cero.

Incluso muchos estudiantes tienen deficiencias en la matemática, lo cual es una dificultad para la programación. Esta deficiencia y mejora en esta competencia se logra al involucrar un poco más al estudiante y que este realice las actividades por su cuenta y no esperar a que los compañeros y profesor tengan la solución para copiarlo.

El estudiante al hacer ejercicios y practicar con la ayuda de herramientas tecnológicas, logra mayor comprensión de la lógica a seguir.

Definitivamente el diseño de un syllabus y la evaluación auténtica ha permitido que el nivel del curso crezca, sin embargo, es necesario seguir innovando en la clase para lograr mejores resultados en los estudiantes.

REFERENCIAS

- [1] Moreno, T. (2012). La evaluación de competencias en educación. ISSN 2007-7033
- [2] Ministerio de Educación Pública de Costa Rica, MEP. <https://www.mep.go.cr/sites/default/files/reglamento-evaluacion-aprendizajes.pdf>
- [3] Anijovich, R, Cappelletti, G. (2017). La evaluación como oportunidad. Obtenido de http://pdfhumanidades.com/sites/default/files/apuntes/ANIJOVICH%20La%20evaluacion%20como%20oportunidad_0.pdf
- [4] Orozco-Jutorán, M. (2006) "La evaluación diagnóstica, formativa y sumativa en la enseñanza de la traducción", en: Varela, M.J. (ed.) La evaluación en los estudios de traducción e interpretación. Sevilla: Bienza. p. 47-68 ISBN: 978-84-933962-8-2.
- [5] Prigent, R., Bernard, H. et Kozanitis, A. (2009). Enseigner à l'université dans une approche-programme. Presses internationales Polytechnique.
- [6] Wiggins, G. (1990). The Case for Authentic Assessment. ERIC Digest. Obtenido de <http://www.ericdigests.org/pre-9218/case.htm>
- [7] Revista iberoamericana de educación. N.º 64 (2014), pp. 11-25 (ISSN: 1022-6508) - oei/caeu1111monográfico / monográfica evaluación auténtica de los procesos educativos.
- [8] Crispín, M, Fernández, T, Ramírez, J, Ulloa, J. (2012). Guía del docente para el desarrollo de competencias. Obtenido de https://ibero.mx/formaciondeprofesores/Apoyos%20generales/Guia_doc_ente_desarrollo_competencias.pdf
- [9] Guijosa, C. (2018). El objetivo de la educación basada en competencias. Obtenido de <https://observatorio.tec.mx/edu-news/el-objetivo-de-la-educacion-basada-en-competencias>
- [10] Jerez, O. (2015). El diseño de Syllabus En La Educación Superior: Una Propuesta Metodológica.
- [11] Zabalza, M. (2004). Guía para la planificación didáctica de la docencia Universitaria en el Marco del EEES. Obtenido de https://www.udc.es/grupos/apumefyr/docs_significativos/guiadeguias.pdf
- [12] Pontificia Universidad Católica de Valparaíso (2017). Que implica el diseño de un curso en formato syllabus. Obtenido de: <http://vra.ucv.cl/ddcyf/wp-content/uploads/2017/08/que-implica-el-diseno-C3%B1o-de-un-curso-en-formato-syllabuscontinual.pdf>
- [13] Colorado, B. (2018). Diseño instruccional ¿Y eso qué es? Obtenido de <https://elearningactual.com/que-es-el-diseno-instruccional-significado/>
- [14] Hernández, R. (2010). Metodología de la investigación.
- [15] Montoya, F. (2019). Implementación de metodologías innovadoras en el curso ISC-110 Programación 1 en un 26.7% de las clases. 10.18687/LACCEI2019.1.1.258