

Diseño colaborativo basado en ThinkLets como apoyo a la enseñanza de la Programación

Collaborative design based on ThinkLets to support the teaching of Programming

Oscar Revelo-Sánchez¹ , Cesar Alberto Collazos² , Andrés F. Solano³ , Habib Fardoun⁴ 

¹Universidad de Nariño, San Juan de Pasto, Colombia

²Universidad del Cauca, Popayán, Colombia

³Universidad Autónoma de Occidente, Cali, Colombia

⁴King Abdulaziz University, Arabia Saudi

orevelo@udenar.edu.co, ccollazo@unicauca.edu.co, afsolano@uao.edu.co, hfordoun@kau.edu.sa

(Recibido: 2 Mayo 2020; aceptado: 22 Septiembre 2020; Publicado en Internet: 1 Diciembre 2020)

Resumen. La incorporación del trabajo colaborativo en el ámbito educativo crece día a día, al igual que el número de grupos de investigación y proyectos asociados a este tema, dado que las actividades de aprendizaje especificadas de forma colaborativa promueven la comunicación, la coordinación y la negociación al interior de los grupos. Si bien se ha demostrado la utilidad práctica de esta estrategia en diferentes niveles de formación, no se ha hecho mucho énfasis en una parte fundamental: el diseño de los procesos colaborativos inherentes. La Ingeniería de la Colaboración destaca para ello el uso de patrones de colaboración y de *ThinkLets* como unidades primarias de diseño, lo cual tiene como objetivo primordial la reutilización. Este artículo presenta una propuesta de actividad colaborativa a implementarse como apoyo en la enseñanza de cursos iniciales de Programación en el ámbito universitario, la cual basa su diseño colaborativo en la utilización de patrones y *ThinkLets*.

Palabras clave: Actividad Colaborativa, Diseño Colaborativo, Patrones de Colaboración, *ThinkLets*, Programación de Computadores.

Abstract. The incorporation of collaborative work in the educational field grows day by day, as do the number of research groups and projects associated with this topic; since the learning activities specified collaboratively promote communication, coordination, and negotiation within the groups. While this strategy's practical usefulness has been demonstrated at different training levels, not much emphasis has been placed on a fundamental part: the design of inherent collaborative processes. Collaboration Engineering highlights collaboration patterns and ThinkLets as primary design units, which has its primary objective reuse. This article presents a proposal for a collaborative activity to be implemented to support the teaching of initial programming courses at the university level, which bases its collaborative design on patterns and ThinkLets.

Keywords: Collaborative activity, Collaborative design, Collaborative patterns, ThinkLets, Computer programming.

Tipo de artículo: Artículo de investigación.

1 Introducción

La incorporación del trabajo colaborativo en cursos de Programación ha sido identificada como una estrategia potencial que podría maximizar la participación de los estudiantes y tener un impacto positivo en el aprendizaje (Revelo-Sánchez et al., 2018). Debido a la habitual complejidad en la enseñanza/aprendizaje de la Programación, se han planteado diferentes enfoques didácticos que incorporan elementos de colaboración, y en ocasiones en combinación con otros enfoques, buscando consolidar estrategias que aporten posibles soluciones al problema (Revelo-Sánchez et al., 2018).

Un requisito indispensable para llevar a la práctica lo antes descrito, es que las actividades de aprendizaje que vayan a realizarse como parte de la estrategia, sean especificadas de forma colaborativa, promoviendo la comunicación, la coordinación y la negociación al interior de los grupos. A pesar de que se ha demostrado en diferentes trabajos de investigación y para diferentes niveles de formación, la utilidad práctica que una estrategia colaborativa de aprendizaje tiene en la enseñanza de la Programación (Revelo-Sánchez et al.,

2018), no se ha hecho énfasis en una parte fundamental, como es el diseño de los procesos colaborativos inherentes.

La Ingeniería de la Colaboración, como disciplina emergente, destaca para ello el uso de patrones de colaboración y de *ThinkLets* como unidades primarias de diseño, lo cual tiene como objetivo primordial la reutilización (Kolfshoten & de Vreede, 2007). En este artículo se presenta una propuesta de actividad colaborativa a implementarse como apoyo en la enseñanza de cursos iniciales de Programación en el ámbito universitario, la cual basa su diseño colaborativo en la utilización de patrones y *ThinkLets*.

La estructura del artículo es la siguiente: inicialmente de forma breve se describen los referentes teóricos relacionados con la evaluación por pares como actividad colaborativa fundante, y de los patrones de colaboración y los *ThinkLets* como unidades primarias de diseño; posteriormente la Sección 3 describe en detalle la actividad colaborativa propuesta incluyendo la identificación de roles, el plan de implementación y el diseño colaborativo específico. Finalmente, la Sección 4 presenta un conjunto conclusiones y trabajo futuro.

2 Referentes teóricos

2.1 Evaluación por pares

Topping et al., (2000) definen la evaluación por pares como “*un acuerdo para que los compañeros consideren el nivel, el valor, la calidad o el éxito de los productos o los resultados del aprendizaje de otros con un estatus similar*” (Topping et al., 2000). La evaluación por pares es una técnica que generalmente se considera efectiva para promover las habilidades cognitivas superiores de los estudiantes, ya que estos usan sus conocimientos y habilidades para interpretar, analizar y evaluar el trabajo de los demás a fin de aclararlo y corregirlo (Sitthiworachart & Joy, 2004). Esta técnica ha sido satisfactoriamente empleada en variedad de disciplinas académicas.

La evaluación por pares es una extensión natural del cambio de un modo de educación centrado en el profesor a uno centrado en el estudiante, que enfatiza la responsabilidad y la participación activa de los estudiantes en su proceso de aprendizaje, así como las habilidades metacognitivas y un modelo dialógico y colaborativo de enseñanza/aprendizaje. Debe diseñarse para mejorar el aprendizaje profundo mediante rondas de revisión y comentarios (Spiller, 2012) (ver [Figura 1](#)).



Figura 1. La evaluación por pares. Rondas de revisión y comentarios de los estudiantes.
Fuente: Basado en (Spiller, 2012).

La evaluación por pares, propuesta por (Topping et al., 2000), puede verse como una estrategia de enseñanza que brinda a los estudiantes la posibilidad de evaluar las fortalezas y debilidades de los trabajos de sus pares, permitiéndoles realizar modificaciones a objetivos no alcanzados, para desarrollar sus capacidades metacognitivas y de pensamiento crítico, su rendimiento en pruebas y sus habilidades profesionales. La evaluación por pares se puede aplicar como una evaluación formativa o sumativa, al proporcionar a los estudiantes las rúbricas de los profesores para evaluar el trabajo de los compañeros y dar sugerencias. Durante el proceso de evaluación entre pares, los estudiantes deben ver los trabajos desde el punto de vista de los profesores y pensar si el rendimiento de los compañeros cumple con los requisitos establecidos. Por otra parte, al evaluar el trabajo de los compañeros, los estudiantes también tienen la oportunidad de reflexionar y pensar cómo mejorar su propio trabajo; dicho proceso puede beneficiar el rendimiento académico de los “revisores” y de los “revisados”, y promover la creatividad de los estudiantes (Hwang et al., 2016).

La evaluación por pares es fundamentalmente una actividad de colaboración que se produce entre al menos dos pares. Al reconocer esto, un tema crucial es el grado de interactividad que se permite durante el proceso de evaluación. En una versión más interactiva, la evaluación por pares puede permitir o incluso requerir que los compañeros negocien sobre cómo abordar una tarea determinada, cómo dar su opinión y cómo usar los comentarios durante la revisión (van Gennip et al., 2010). Tal intercambio interactivo puede ser beneficioso porque puede evocar procesos cognitivos y discursivos que desencadenan una elaboración más profunda del material y, por lo tanto, conducen a un mejor aprendizaje (King, 2007).

2.2 Patrones de colaboración y *ThinkLets*

La Ingeniería de la Colaboración ha identificado una serie de patrones a partir de los cuales un grupo trabaja colaborativamente buscando alcanzar sus metas (Coto et al., 2016). Los patrones de colaboración definen la manera cómo los participantes de una actividad grupal van de un estado inicial a un estado final (de Vreede & Briggs, 2005). Cada patrón tiene sub-patrones que pueden relacionarse con actividades, en la descripción del proceso genérico. Los patrones de colaboración comúnmente identificados son (Briggs et al., 2005, 2006):

- **Generar:** pasar de tener menos a tener más conceptos en el conjunto de conceptos compartidos por el grupo.
 - *Reunir:* recopilar y compartir conceptos conocidos por miembros del grupo.
 - *Crear:* producir y compartir ideas nuevas que antes no conocían los miembros del grupo.
 - *Elaborar:* agregar detalles a los conceptos que ya están compartidos por el grupo.
 - *Descomponer:* caracterizar un concepto en términos de sus componentes y subcomponentes.
 - *Expandir:* agregar detalles para explicar o describir completamente un concepto.
- **Reducir:** pasar de tener muchos conceptos a centrarse en menos conceptos que el grupo considere que merecen mayor atención.
 - *Seleccionar:* elegir un subconjunto de conceptos existentes.
 - *Abstraer:* derivar conceptos más generales de instancias específicas en el conjunto existente.
 - *Resumir:* capturar la esencia de los conceptos sin eliminar los conceptos únicos.
- **Aclarar:** pasar de tener menos a tener una comprensión más compartida de los conceptos y de las palabras y frases utilizadas para expresarlos.
 - *Describir:* proponer explicaciones y formulaciones alternativas de un concepto.
- **Organizar:** pasar de menos a más comprensión de las relaciones entre los conceptos que el grupo está considerando.
 - *Clasificar:* organizar los conceptos en clústeres etiquetados.
 - *Estructurar:* crear arreglos espaciales entre conceptos para representar sus relaciones conceptuales.
- **Evaluar:** pasar de menos a más comprensión del valor relativo de los conceptos considerados.
 - *Encuestar:* evaluar la opinión del grupo con respecto a los conceptos.
 - *Clasificar:* identificar un orden de preferencia entre los conceptos.
 - *Valorar:* especificar y determinar el valor de los conceptos.
- **Construir consenso:** pasar de tener menos a tener más miembros del grupo que estén dispuestos a comprometerse con una propuesta.

- **Medir:** valorar el grado en que los interesados están dispuestos a comprometerse con una propuesta.
- **Diagnosticar:** buscar entendimiento de las causas subyacentes del disentimiento.
- **Defender:** persuadir a otros para que adopten y acepten una posición.
- **Resolver:** buscar formas de superar las causas subyacentes del disentimiento.

Los patrones de colaboración son una guía general de cómo se ejecutará un proceso. Sin embargo, en sí mismos no presentan mayor detalle metodológico. Para ello se crearon los *ThinkLets*, que son técnicas de facilitación repetibles, transferibles y predecibles para llevar a un grupo a alcanzar el objetivo acordado (Briggs et al., 2003). Un *ThinkLet* se refiere a la unidad más pequeña de conocimiento requerido para ser capaces de reproducir la colaboración entre personas que trabajan hacia un objetivo común (Santanen & de Vreede, 2004). Los *ThinkLets* pueden ser usados como unidades conceptuales de construcción en el diseño de procesos que involucran la colaboración como elemento primordial, ya que cada *ThinkLet* proporciona el conocimiento necesario sobre los pasos o instrucciones a seguir para implementarlos como parte del proceso grupal (Briggs et al., 2006).

Antes de que se llamara “*ThinkLet*”, este concepto tenía otras denominaciones menos adecuadas: receta, técnica, módulo de razonamiento. El término “*ThinkLet*” fue acuñado por David Tobey en marzo de 2000 mientras bregaba por una denominación más expresiva. “Es como un applet...”, dijo, “... pero es un *ThinkLet*” (Briggs et al., 2001).

Los *ThinkLets* se detallan completamente y son modificables. Pueden usarse para construir nuevos grupos de procesos. Son recetas que pueden ser utilizadas por usuarios sin mucha experiencia, dado que pueden ser fácilmente aprendidos, recordados y se pueden adaptar fácilmente a un diseño de proceso (Briggs et al., 2003).

La documentación estándar de un *ThinkLet* presenta la siguiente información (Briggs et al., 2001): un nombre metafórico o representativo relacionado con el patrón o los patrones de colaboración que soporta, los criterios para decidir cuándo escogerlo o no, una información general (como entradas y salidas), cómo usarlo (pasos que lo conforman), una historia exitosa que ayuda a clarificar las circunstancias bajo las cuales es útil, y una explicación del nombre, la cual hace más fácil recordarlo. En la [Tabla 1](#) se esquematiza la estructura original que tienen los diferentes *ThinkLets* (Briggs et al., 2001).

Tabla 1. Estructura de un *ThinkLet* según la documentación original.

Nombre del <i>ThinkLet</i> : Patrón de Colaboración Asociado
Seleccione este <i>ThinkLet</i> si... ... No seleccione este <i>ThinkLet</i> si... ... Información general ... Entradas: ... Salidas: ... Cómo usar <Nombre del <i>ThinkLet</i> > Herramientas ... Configuración 1. ... 2. Pasos 1. ... 2. Ideas sobre <Nombre del <i>ThinkLet</i> > ... Historia exitosa ... ¿Por qué el nombre? ...

Fuente: (Briggs et al., 2001)

3 La actividad colaborativa propuesta

Se propone una actividad colaborativa, la cual se ha denominado “Evaluación de Código por Pares”, que permita a los estudiantes de un curso inicial de Programación, seguir trabajando en grupo fuera del aula.

3.1 Identificación de roles

Para la realización de la actividad propuesta, se identifican los roles que se muestran en la Figura 2, y que se describen a continuación:

- Profesor: quien es el encargado de motivar y coordinar la actividad.
- Estudiantes evaluadores: son los grupos de estudiantes quienes en la etapa de proceso toman el rol de evaluadores del código de sus compañeros autores.
- Estudiantes autores: son los grupos de estudiantes quienes en la etapa de proceso toman el rol de autores del código y son evaluados por sus compañeros evaluadores.

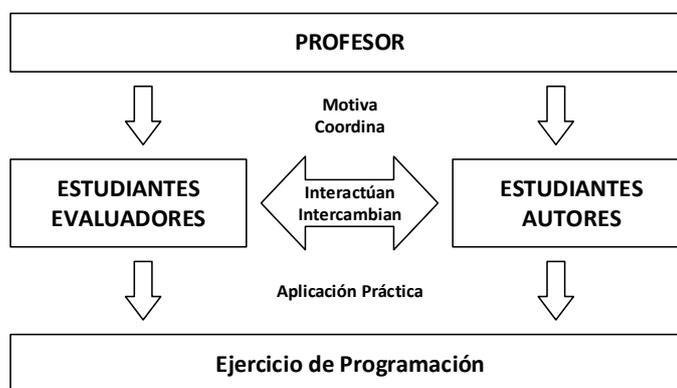


Figura 2. Identificación de roles en la evaluación de código por pares.

El rol de evaluador y de autor se desempeña en paralelo, dado que mientras el estudiante se encuentra evaluando el código de sus compañeros, al mismo tiempo, su código está siendo también evaluado por sus compañeros.

Etapas. Teniendo en cuenta que un proceso de aprendizaje colaborativo está formado por varias tareas o actividades que deben ser desarrolladas tanto por el mediador cognitivo o facilitador como por los grupos de aprendices, como lo proponen (Guerrero et al., 1999), éstas se organizan en tres etapas:

Primera etapa (pre-proceso). Se desarrolla una actividad colaborativa en línea donde los estudiantes deben resolver un problema planteado por el docente realizando el análisis, el diseño y la implementación en el lenguaje de programación seleccionado para el curso. Para la parte del análisis y el diseño en pseudocódigo de la solución se utilizan documentos compartidos en línea a través de Google Docs. Cada estudiante desde su computador accede a Google Docs y al documento compartido. La participación de cada estudiante se incorpora al mismo documento y todos tendrán la misma versión para trabajar. Esta participación puede ser realizada en forma asincrónica, y también se puede hacer en forma sincrónica donde todos los estudiantes que comparten el documento pueden ver que están haciendo sus compañeros. Una vez que el equipo acuerda una solución en Google Docs, deben convertir el pseudocódigo al lenguaje de programación. Esta actividad dura 7 días.

Segunda etapa (proceso). Cada grupo entrega el código fuente y el código ejecutable al docente del curso. Esto se realiza a través de un sistema de gestión de aprendizaje (*Learning Management System – LMS*). El docente realiza un cruce de soluciones en forma anónima entre los diferentes grupos, de manera que cada equipo pueda corregir y verificar la solución que le fue asignada. Se le envía a cada grupo el código escrito en el lenguaje seleccionado, el programa ejecutable, las pautas para realizar la actividad y las pautas de corrección. Para llevar a cabo esta tarea y como parte del trabajo del equipo se deberá establecer quien cumple el rol de “autor” de la solución y quien cumple el rol “evaluador” de la solución. Cada equipo presenta un informe en línea con las correcciones y comentarios efectuados al código. Estas

correcciones son visualizadas por el grupo evaluador y el docente a través del LMS. Esta actividad dura 4 días.

Tercera etapa (post-proceso). El docente recibe las correcciones de cada grupo evaluador y las redistribuye a cada grupo autor. El docente revisa las correcciones y selecciona una o varias soluciones que mejor representen la solución al problema y que cumplan con las pautas de calidad de programación establecidas. Con las soluciones seleccionadas, se trabaja en el aula presencial y se ponen en discusión los códigos presentados y se construye, junto al docente, una solución acordada entre todos los participantes. Por último, se publica esa solución en el LMS. En este nuevo espacio compartido cada grupo compartirá su solución corregida. Las actividades del tercer momento duran 4 días.

3.2 Plan de implementación

A continuación, en la [Tabla 2](#) se describe, como lo proponen (Collazos & Mendoza, 2006), un plan de implementación, en el que se incluyen las actividades en la etapa de inicio (configuración inicial) y aquellas que se requieren para garantizar los principios básicos de una actividad colaborativa.

Tabla 2. Estructura de un *ThinkLet* según la documentación original.

Configuración inicial				
Objetivos	Duración	Recursos/Materiales	Conformación de grupos	Distribución de estudiantes
Resolver un problema de programación realizando el análisis, el diseño y la implementación en lenguaje. Enviar a cada grupo el código, las pautas para realizar la actividad y las pautas de corrección. Presentar un informe en línea con las correcciones y comentarios efectuados al código. Construir conjuntamente la solución al problema.	15 días	Documentación de análisis, diseño y pseudocódigo compartida en GD. Código fuente y ejecutable implementado en lenguaje de programación y publicado en LMS. Solución final publicada en LMS.	Homogéneos con respecto a los rasgos de personalidad de los estudiantes 3 o 4 estudiantes	Para la primera y segunda etapas, trabajo virtual sincrónico y asincrónico en GD y LMS. Para el tercer momento, trabajo en aula física de clase.
Principios básicos				
Principios		Actividades		
1. Interdependencia positiva		<ul style="list-style-type: none"> • Productos solicitados: documentación de análisis, diseño y pseudocódigo, y, código fuente y ejecutable. • Roles secundarios asignados: para los autores, analista, diseñador e implementador; para los evaluadores, relator y revisores. Por cada ronda se intercambian estos roles. • Evaluación grupal y bonificaciones por promedio grupal que exceda un mínimo establecido por el profesor. 		
2. Responsabilidad individual		<ul style="list-style-type: none"> • Conformación de grupos de máximo 4 integrantes. • Evaluación individual y bonificaciones por promedio grupal que exceda un mínimo establecido por el profesor. 		
3. Habilidades cognitivas e interpersonales		<ul style="list-style-type: none"> • Implementación adecuada de la solución. • Comunicación. • Colaboración. • Solución de conflictos. 		
4. Interacción simultánea		<ul style="list-style-type: none"> • Interacción sincrónica y asincrónica en GD. 		

	<ul style="list-style-type: none"> • Interacción sincrónica y asincrónica en LMS a través de chat y foro de discusión.
5. Evaluación y reflexión	<ul style="list-style-type: none"> • Las entregas en las fechas acordadas. • Envío o no de la solución ejecutable. • La cantidad de accesos y la cantidad y calidad de líneas de código en el documento compartido en GD. • La entrega de la corrección solicitada. • La corrección realizada de acuerdo a las pautas fijadas. • Los tipos de correcciones realizadas. • Nivel de participación en clase. • La actividad finalmente conforma una nota que será promediada junto a otras evaluaciones que se realizan en el curso para conformar la nota final de cada estudiante.
6. Actividades de extensión	N/A (Dado el diseño de la actividad)

Fuente: (Collazos & Mendoza, 2006)

3.3 Diseño colaborativo

Para el diseño colaborativo de la actividad propuesta, se emplea la metodología para el desarrollo de procesos colaborativos propuesta por (Kolfshoten & de Vreede, 2007), la cual se especifica a continuación en cada una de sus fases.

Diagnóstico de la actividad.

Meta. Realizar evaluación de código por pares como actividad colaborativa, que permita a los estudiantes de un curso inicial de Programación, continuar trabajando fuera del aula de clase.

Entregables. Documentos de análisis, diseño y pseudocódigo, y, código fuente y ejecutable, tanto grupales como unificados.

Objetivo. Colaborar para obtener una perspectiva más amplia de la temática en desarrollo.

Grupo. Estudiantes de cursos iniciales de programación del ámbito universitario, organizados homogéneamente por rasgos de personalidad en grupos de 3 o 4 estudiantes.

Roles. Profesor, quien es el encargado de motivar y coordinar la actividad; estudiantes evaluadores, son los grupos de estudiantes quienes en la etapa de proceso toman el rol de evaluadores del código de sus compañeros autores; y, estudiantes autores, son los grupos de estudiantes quienes en la etapa de proceso toman el rol de autores del código y son evaluados por sus compañeros evaluadores.

Requerimientos. Documentos compartidos en línea (Google Docs), soporte de LMS disponible, aula física de clase, tres momentos de interacción con una duración aproximada de 15 días entre los estudiantes de un curso inicial de programación.

Descomposición de la actividad. Utilizando en el enfoque de descomposición del proceso (Kolfshoten & de Vreede, 2007), la actividad se podría descomponer bajo la siguiente estructura (ver [Tabla 3](#)):

Tabla 3. Descomposición de la actividad.

Nombre de la actividad: Evaluación de código por pares		
No.	Sub-actividades	Colaborativa
1	Análisis, diseño e implementación inicial de la solución.	Si
2	Corrección y verificación de soluciones.	Si
3	Construcción de una solución acordada.	Si

Participantes: Profesor, estudiantes.

Tiempo de duración estimado: 15 días

Sub-Actividad 1: Análisis, diseño e implementación inicial de la solución

Actividades relacionadas: Utilización de documentos compartidos en línea y de LMS.

Descripción: Se desarrolla una actividad colaborativa en línea donde los estudiantes deben resolver un problema planteado por el profesor realizando el análisis, el diseño y la implementación en el lenguaje de programación seleccionado para el curso.

Entradas: Problema planteado por el profesor; documento base de análisis, diseño y pseudocódigo.

Resultados esperados: Documentación de análisis, diseño y pseudocódigo compartida en GD; código fuente y ejecutable implementado en lenguaje de programación y publicado en LMS.

Participantes: Estudiantes.

Patrón	Justificación
Generación	Este patrón permite a los estudiantes al interior de cada grupo, realizar aportes para la construcción colectiva de una solución al problema planteado por el profesor. El objetivo de este patrón es pasar de tener pocos a tener muchos conceptos e ideas compartidas por el grupo..

Sub-Actividad 2: Corrección y verificación de soluciones

Actividades relacionadas: Utilización de documentos compartidos en línea y de LMS.

Descripción: Se desarrolla una actividad colaborativa en línea en la que el profesor realiza un cruce de soluciones en forma anónima entre los diferentes grupos, de manera que cada equipo pueda corregir y verificar la solución que le fue asignada.

Entradas: Código fuente y ejecutable de las soluciones de los diferentes grupos; pautas para realizar la actividad, pautas de corrección.

Resultados esperados: Informe en línea con las correcciones y comentarios efectuados al código.

Participantes: Profesor, estudiantes.

Patrón	Justificación
Generación	Este patrón permite a los estudiantes al interior de cada grupo, realizar aportes para la corrección y verificación colectiva de las diferentes soluciones al problema planteado por el profesor. El objetivo de este patrón es pasar de tener pocos a tener muchos conceptos e ideas compartidas por el grupo.

Sub-Actividad 3: Construcción de una solución acordada

Actividades relacionadas: Utilización de documentos compartidos en línea y de LMS.

Descripción: Se desarrolla una actividad colaborativa en aula presencial, en la que el profesor revisa las correcciones y selecciona una o varias soluciones que mejor representen la solución al problema y que cumplan con las pautas de calidad de programación establecidas; esas soluciones se ponen en discusión y se construye, junto al profesor, una solución acordada entre todos los participantes de la actividad.

Entradas: Soluciones corregidas y comentadas.

Resultados esperados: Código fuente y ejecutable de la solución acordada.

Participantes: Profesor, estudiantes.

Patrón	Justificación
Construcción de consenso	Este patrón permite a los estudiantes al interior de cada grupo, realizar aportes para la construcción consensuada de una solución al problema planteado por el profesor. El objetivo de este patrón es pasar de tener pocos a tener muchos miembros del grupo que estén dispuestos a comprometerse con una propuesta

Fuente: (Kolfschoten & de Vreede, 2007)

Selección de ThinkLets. A continuación, se presenta la descripción de los *ThinkLets* a emplearse en las sub-actividades definidas como colaborativas (ver [Tabla 4](#) y [Tabla 5](#)). Algunos aspectos de la estructura de los *ThinkLets* han sido modificados levemente, pero manteniendo la estructura básica y objetivo de cada uno de ellos.

Tabla 4. *ThinkLet* Comparative Brainstorm

<i>ThinkLet</i> Comparative Brainstorm	
Escoger este ThinkLet...	<ul style="list-style-type: none"> • Para generar una lluvia de ideas sobre soluciones de problemas. • Cuando sepa de antemano cuál será el criterio para juzgar si una solución es buena o no. • Cuando el tiempo es esencial.
No escoger este ThinkLet...	Cuando desea dar libertad a las personas en su búsqueda de soluciones. Use <i>FreeBrainstorm</i> en su lugar.
Información general	<p>El equipo genera una lluvia de ideas de soluciones para un problema en respuesta a una secuencia de pautas comparativas ofrecidas por un moderador.</p> <p>Entradas: Un conjunto de criterios para decidir qué soluciones son buenas y cuáles no lo son.</p> <p>Salidas: Un conjunto de soluciones potenciales.</p>
Configuración	<ol style="list-style-type: none"> 1. Desarrolle un conjunto de pautas comparativas basado en los criterios de calidad requeridos para la solución. 2. Publique un enunciado como pregunta para la lluvia de ideas.

	3. Cree una página de lluvia de ideas para cada participante, más una adicional, más otra adicional por cada 10 personas en el grupo (si es el caso).
Pasos	<ol style="list-style-type: none"> 1. Asegúrese de que los participantes entiendan el problema. 2. Diga esto: <ol style="list-style-type: none"> a. “Cada uno tome su página de lluvia de ideas”. b. “Cada uno de ustedes está comenzando una página diferente”. c. “Escriba la mejor solución que se le ocurra para el problema”. d. “Cuando termine de escribir su solución, no la diga. Solo cruce sus brazos para saber que ya terminó”. 3. Espere hasta que todos hayan terminado de escribir su primera idea, luego diga esto: <ol style="list-style-type: none"> a. “Ahora intercambien páginas”. b. Dé una de las indicaciones comparativas. 4. Repita el patrón de intercambio de páginas y responda a las indicaciones comparativas hasta que el grupo se quede sin tiempo o ideas.
¿Por qué el nombre?	Este <i>ThinkLet</i> lleva su nombre porque se le pide a los participantes que comparen sus nuevas ideas con las ideas que ya existen, y que rechacen cualquier contribución.

Tabla 5. *ThinkLet* Moodring

<i>ThinkLet</i> MoodRing	
Escoger este <i>ThinkLet</i>...	<ul style="list-style-type: none"> • Para detectar patrones de consenso sobre un solo tema en tiempo real. • Para saber cuándo es el momento de dejar de hablar y tomar una decisión.
No escoger este <i>ThinkLet</i>...	Si necesita abordar varios problemas al mismo tiempo. Use <i>StrawPoll</i> en su lugar.
Información general	<p>Los participantes registran su opinión sobre un solo tema, luego comienzan una discusión oral. Mientras hablan, si escuchan algo que cambia su opinión en cualquier dirección, cambian su voto.</p> <p>Entradas: Una sola opinión de un problema.</p> <p>Salidas: Un patrón siempre cambiante de consenso grupal.</p>
Configuración	<ol style="list-style-type: none"> 1. Publique una opinión sobre un problema en un medidor de opinión. 2. Dé a conocer el medidor de opinión a los participantes.
Pasos	<ol style="list-style-type: none"> 1. Asegúrese de que el grupo entienda el problema. Diga esto: “Si tiene preguntas aclaratorias sobre el tema en cuestión, por favor levantar la mano”. 2. Si alguien levanta la mano, facilite una discusión verbal para abordar cualquier dificultad de comprensión. Si es necesario, vuelva a formular el problema. 3. Diga esto: <ol style="list-style-type: none"> a. “Por favor registre su opinión en el medidor de opinión”. b. “Ahora hablemos del problema. Si escucha algo que le haga cambiar de parecer en cualquier dirección, cambie su voto. Seguiremos hablando hasta que hayamos llegado a algún tipo de consenso sobre este tema”.
¿Por qué el nombre?	Este <i>ThinkLet</i> lleva su nombre dada la novedad del “anillo anímico” que apareció en el mercado en la década de los 60’s. Se suponía que el anillo debía revelar el estado de ánimo de la persona que lo portaba. Si la persona se sentía deprimida, se decía que el anillo se volvía negro. Si la persona se sentía feliz, el anillo se ponía azul. Si la persona se sentía enojada, se suponía que el anillo se ponía rojo. De hecho, los colores eran sensibles a la temperatura y no al estado de ánimo. No obstante, al igual que el legendario anillo anímico, este <i>ThinkLet</i> revela la actitud de los miembros del grupo en tiempo real. Proporciona una idea del estado de ánimo del equipo.

Documentación del diseño. A partir de la información obtenida en las etapas anteriores, a continuación, se documenta el diseño colaborativo de la actividad propuesta.

Construcción de agenda. La agenda es el documento que presenta de manera clara y detallada la información de las actividades que forman parte del proceso diseñado. En la [Tabla 6](#) se muestra en detalle la construcción de este documento.

Modelo de facilitación del proceso (FPM). Es un diagrama utilizado para mostrar el flujo del proceso y los elementos del mismo. En la [Figura 3](#) se muestra la especificación de este diagrama para la actividad bajo diseño.

Tabla 6. Agenda detallada

Actividad	Descripción	Pregunta/Asignación	Entregable	ThinkLet & Patrón	Tiempo (días)
1	Análisis, diseño e implementación inicial de la solución.	Utilizar documentos compartidos en línea a través de GD. Convertir el pseudocódigo a lenguaje de programación.	Documentación de análisis, diseño y pseudocódigo compartida en GD; código fuente y ejecutable implementado en lenguaje de programación y publicado en LMS.	Comparative Brainstorm (Generación)	7
2	Corrección y verificación de soluciones.	Enviar el código, las pautas para realizar la actividad y las pautas de corrección.	Informe en línea con las correcciones y comentarios efectuados al código.	Comparative Brainstorm (Generación)	4
3	Construcción de una solución acordada.	Publicar la solución acordada.	Código fuente y ejecutable de la solución acordada.	MoodRing (Construcción de consenso)	4

Validación del diseño. Se valida la especificación del proceso colaborativo a través de una prueba piloto con estudiantes de cursos iniciales de programación del ámbito universitario, con el fin de verificar si la ejecución del diseño se puede llevar a cabo en el tiempo estimado y con los recursos definidos.

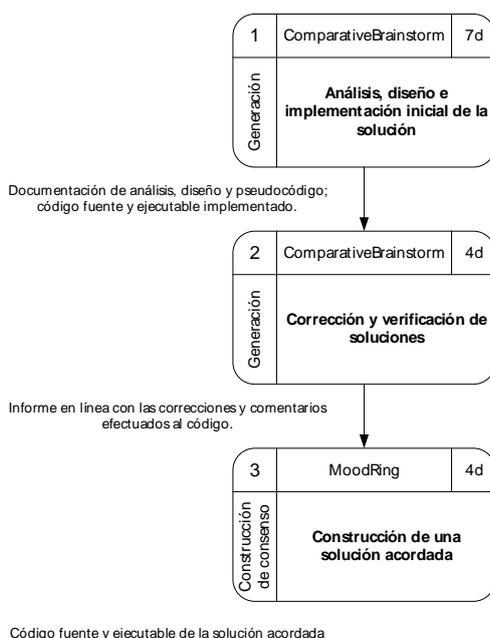


Figura 3. Modelo de facilitación del proceso.

4 Conclusiones y trabajo futuro

En el campo de la enseñanza de la Programación se hace necesario involucrar aspectos didácticos que coloquen al estudiante como elemento central y activo del proceso de aprendizaje, inculcándole la necesidad del autoaprendizaje en un ambiente colaborativo que mejore las actitudes y aptitudes de estudio y de trabajo en equipo.

En tal sentido, el desarrollo de actividades colaborativas específicas, como la descrita en este artículo: la “evaluación de código por pares”, intenta fortalecer la colaboración entre los miembros del grupo, es decir, se intenta promover la comunicación, la coordinación y la negociación, con el fin de motivar el aprendizaje al interior del grupo. La colaboración permite a los miembros del grupo unir esfuerzos intelectuales en búsqueda de un objetivo común que, en este caso particular, es lograr apropiarse de una forma alternativa y didáctica de conceptos fundamentales de la Programación.

El diseño de actividades colaborativas, como se describe y emplea en el presente artículo, ofrece un soporte útil y efectivo, especialmente para diseñadores de procesos novatos. Sin embargo, toma un tiempo considerable asimilar y usar adecuadamente la documentación y los materiales de apoyo. Esta metodología de diseño está basada en los denominados *ThinkLets*, considerados como bloques de construcción que se pueden unir para especificar cómo un determinado patrón de colaboración debe implementarse cuando el proceso está en ejecución.

Los *ThinkLets* que se describen en este documento son una pequeña muestra de los muchos que aún se están documentando, publicando y probando. Dado que los *ThinkLets* crean de forma demostrable patrones de colaboración repetibles, pueden proporcionar una ventana a los mecanismos cognitivos, sociales y de otros tipos que entran en juego cuando las personas trabajan conjuntamente hacia sus metas.

Como trabajo futuro inmediato se tiene la validación integral del diseño colaborativo propuesto. Para esto se espera contar con al menos dos grupos de experimentación, cursos iniciales de Programación del ámbito universitario, a uno de los cuales se le aplicaría un tratamiento experimental consistente en desarrollar la actividad colaborativa propuesta siguiendo el diseño y la metodología descrita en el presente artículo, y al otro se lo consideraría como grupo de control, desarrollando con él una actividad que el profesor diseñe y considere como colaborativa. En ambos casos se trabajaría sobre las mismas temáticas y se mediría los rendimientos académicos cuantificados específicamente con los resultados en las actividades propuestas. Además, se realizarían las pruebas estadísticas correspondientes para comprobar la prueba de hipótesis de la efectividad del diseño propuesto: un diseño colaborativo basado en *ThinkLets* permite obtener mejores resultados en el desarrollo de actividades colaborativas como apoyo a la enseñanza de la Programación.

Otro aspecto a tener en cuenta como trabajo futuro, pero de un alcance mayor, es una posible adecuación, generalización y estandarización de la metodología para el desarrollo de procesos colaborativos propuesta por (Kolschoten & de Vreede, 2007), orientándola específicamente hacia el diseño de actividades colaborativas de aprendizaje, incorporándole algunos elementos pedagógicos y didácticos, dado que originalmente ésta no tiene orientación hacia un escenario de trabajo específico.

Declaración de conflicto de intereses

Los autores declaran no tener conflicto de intereses con respecto a la investigación, autoría o publicación de este artículo.

Referencias

- Briggs, R. O., de Vreede, G.-J., & Nunamaker Jr., J. F. (2003). Collaboration Engineering with ThinkLets to Pursue Sustained Success with Group Support Systems. *Journal of Management Information Systems*, 19(4), 31–64. <https://doi.org/10.1080/07421222.2003.11045743>
- Briggs, R. O., de Vreede, G.-J., Nunamaker Jr., J. F., & Tobey, D. (2001). ThinkLets: Achieving predictable, repeatable patterns of group interaction with Group Support Systems (GSS). *Proceedings of the Hawaii International Conference on System Sciences*, 32. <https://doi.org/10.1109/HICSS.2001.926238>
- Briggs, R. O., Kolschoten, G. L., & de Vreede, G.-J. (2005). Toward a Theoretical Model of Consensus Building. *Proceedings of the Eleventh Americas Conference on Information Systems*, 101–110.

- <https://aisel.aisnet.org/amcis2005/12>
- Briggs, R. O., Kolfshoten, G. L., de Vreede, G.-J., & Dean, D. L. (2006). Defining Key Concepts for Collaboration Engineering. *Proceedings of the Twelfth Americas Conference on Information Systems*, 121–128. <https://aisel.aisnet.org/amcis2006/17>
- Collazos, C., & Mendoza, J. (2006). How to take advantage of “cooperative learning” in the classroom. *Educación y Educadores, Volumen 9*(4128), 61–76.
- Coto, M., Collazos, C. A., & Mora-Rivera, S. (2016). Modelo Colaborativo y Ubicuo para apoyar los procesos de enseñanza-aprendizaje a nivel Iberoamericano. *Revista de Educación a Distancia (RED)*, 48, 1–30. <https://doi.org/10.6018/red/48/10>
- de Vreede, G.-J., & Briggs, R. O. (2005). Collaboration Engineering: Designing Repeatable Processes for High-Value Collaborative Tasks. *Proceedings of the 38th Annual Hawaii International Conference on System Sciences*, 1–10. <https://doi.org/10.1109/HICSS.2005.144>
- Guerrero, L. A., Alarcon, R., Franco, F., Hiberico, V., & Collazos, C. (1999). Una Propuesta para la Evaluación de Procesos de Colaboración en Ambientes de Aprendizaje Colaborativo. *Proceedings of the International Workshop of Educative Software, TISE'99*, 1–10.
- Hwang, G.-J., Liang, Z.-Y., & Wang, H.-Y. (2016). An Online Peer Assessment-Based Programming Approach to Improving Students' Programming Knowledge and Skills. *2016 International Conference on Educational Innovation through Technology (EITT)*, 81–85. <https://doi.org/10.1109/EITT.2016.23>
- King, A. (2007). Scripting Collaborative Learning Processes: A Cognitive Perspective. En F. Fischer, I. Kollar, H. Mandl, & J. M. Haake (Eds.), *Scripting Computer-Supported Collaborative Learning* (pp. 13–37). Springer. https://doi.org/10.1007/978-0-387-36949-5_2
- Kolfshoten, G. L., & de Vreede, G.-J. (2007). The Collaboration Engineering Approach for Designing Collaboration Processes. En J. M. Haake, S. F. Ochoa, & A. Cechich (Eds.), *Groupware: Design, Implementation, and Use* (Vol. 4, pp. 95–110). Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-74812-0_8
- Revelo-Sánchez, O., Collazos-Ordóñez, C. A., & Jiménez-Toledo, J. A. (2018). El trabajo colaborativo como estrategia didáctica para la enseñanza/aprendizaje de la programación: una revisión sistemática de literatura. *TecnoLógicas*, 21(41), 115–134. <https://doi.org/10.22430/22565337.731>
- Santanen, E. L., & de Vreede, G.-J. (2004). Creative approaches to measuring creativity: comparing the effectiveness of four divergence thinkLets. *Proceedings of the 37th Annual Hawaii International Conference on System Sciences*, 1–10. <https://doi.org/10.1109/HICSS.2004.1265093>
- Sitthiworachart, J., & Joy, M. (2004). Effective peer assessment for learning computer programming. *ACM SIGCSE Bulletin*, 36(3), 122–126. <https://doi.org/10.1145/1026487.1008030>
- Spiller, D. (2012). Assessment Matters: Self-assessment and Peer Assessment. En *Teaching Development Unit*. Hamilton: University of Waikato.
- Topping, K. J., Smith, E. F., Swanson, I., & Elliot, A. (2000). Formative Peer Assessment of Academic Writing Between Postgraduate Students. *Assessment & Evaluation in Higher Education*, 25(2), 149–169. <https://doi.org/10.1080/713611428>
- van Gennip, N. A. E., Segers, M. S. R., & Tillema, H. H. (2010). Peer assessment as a collaborative learning activity: The role of interpersonal variables and conceptions. *Learning and Instruction*, 20(4), 280–290. <https://doi.org/10.1016/j.learninstruc.2009.08.010>