Recibido: 01-08-2021 / Revisado: 23-10-2021
Aceptado: 23-10-2021 / Publicado: 31-01-2022

CodES: herramienta de visualización para desarrollo de pensamiento algorítmico

CodES: visualization tool for developing algorithmic thinking

Javier A. Jiménez Toledo[1], Cesar Collazos[2], Manuel Ortega Cantero[3]

[1] Universidad CESMAG, Colombia
[2] Universidad del Cauca, Colombia
[3] Universidad de Castilla-La Mancha, España

jajimenez@unicesmag.edu.co , ccollazo@unicauca.edu.co , manuel.ortega@uclm.es

RESUMEN. CodES (CODificación con Entradas y Salidas) es una herramienta de visualización que basa su accionar en el artefacto más simple de análisis computacional que es el diagrama de entrada/salida, con el propósito de generar procesos de abstracción para el diseño y escritura de algoritmos, permitiendo que el estudiante centre su atención en comprender el problema a solucionar mediante sus elementos esenciales y a la vez intuir desde un inicio la interfaz computacional a construir con sus diagramas de diseño y codificación. CodES fue validado utilizando un enfoque cuantitativo, con investigación de tipo descriptiva y mediante diseño experimental con grupo de control y pos prueba, además, se utilizó una técnica paramétrica para comprobar la diferencia estadística existente entre datos obtenidos en el proceso investigativo, que junto con un test de usabilidad y una técnica de seguimiento ocular permitieron sugerir el uso de CodES en un primer curso de programación de computadores.

ABSTRACT. CodES (CODification with Inputs and Outputs) is a visualization tool that bases its action on the simplest artifact of computational analysis, which is the input and output diagram, for the purpose of generating an abstraction process for algorithm design and writing, for the student to focus on first understanding the problem to be solved through its essential elements while intuiting the computational interface to be built and their respective design and coding diagrams from the outset. CodES was validated using a quantitative approach with descriptive type research and by experimental design with control group and post-test. In addition, a parametric technique was used to check the statistical difference between data obtained in the research process which together with a Usability test and an Eye-tracking technique, suggested the use of CodES in a first computer programming course.

PALABRAS CLAVE: CodES, Programación, Algoritmo, Pensamiento, Visualización.

KEYWORDS: CodES, Programming, Algorithm, Thinking, Visualization.

Jiménez Toledo, J. A.; Collazos, C.; Ortega Cantero, M. (2022). CodES: herramienta de visualización para desarrollo de pensamiento algorítmico. *Campus Virtuales*, 11(1), 21-33. https://doi.org/10.54988/cv.2022.1.809

1. Introducción

El diseño de algoritmos en un primer curso de programación de computadores (CS1) es una tarea compleja para el estudiante novato debido a que se asume que éste posee diversas habilidades, entre ellas, para resolver problemas y aplicar modelos mentales o matemáticos (Jiménez-Toledo et al., 2019). Además, en un CS1, las metodologías de enseñanza inapropiadas, la falta de visualización de resultados inmediatos al proceso, la carencia de experiencia previa (Jiménez et al., 2015), la falta de interés en la programación de computadores (Malliarakis et al., 2014), la complejidad cognitiva requerida (Insuasti, 2016), los débiles niveles de desarrollo de pensamiento matemático (Silva et al., 2016), la gran cantidad de conceptos nuevos (Ortega et al., 2017), los problemas de aprendizaje en el estudiante (Dann et al., 2006), entre otros, hacen que este tenga mucha importancia puesto que el éxito de un futuro programador de computadores depende en gran medida de este primer curso.

Además, existen varias herramientas de software para afrontar un CS1, Jiménez et al. (2019) las clasifican en cinco categorías: "Visualización o simuladores de algoritmos, herramientas de evaluación automática, juegos educativos centrados en la enseñanza de una unidad específica de aprendizaje, Juegos educativos centrados en la enseñanza de unidades múltiples de aprendizaje y Ambientes colaborativos" (p.29).

A su vez, las herramientas de visualización o también llamadas simuladores de algoritmos permiten seguir de manera continua las instrucciones codificadas a través de representaciones gráficas a sus elementos o mediante valores. El propósito de estas herramientas es orientar al usuario en la representación y evaluación de conceptos abstractos mediante procesos simbólicos en entornos computacionales (Costelloe, 2004).

En este artículo presentamos CodES, una herramienta de visualización especialmente diseñada para generar procesos de abstracción en estudiantes novatos que le permitan el desarrollo de pensamiento algorítmico a través de una interfaz sencilla y partiendo únicamente del diagrama de entrada/salida mediante el cual es posible construir variados ejemplos computacionales.

CodES utiliza tips o claves de identificación que le permitirán de una forma sencilla al estudiante el manejo de entradas y salidas e identificar cuando debe incorporar estructuras condicionales y repetitivas en sus algoritmos. Es así como CodES se construyó pensando en los procesos de abstracción que debe adquirir un estudiante para desarrollar su pensamiento algorítmico, necesarios para afrontar un CS1.

2. Revisión de la literatura

Se propuso como objetivo de la revisión de literatura el caracterizar las herramientas de visualización bajo los principios de un mapeo sistemático, cuya principal pregunta de investigación fue ¿Cuáles son las herramientas de visualización utilizadas en un CS1 y que son reportadas en la literatura científica?, además, en la exploración de los estudios primarios se utilizó una cadena de búsqueda compuesta por un término clave ("Visualizador de algoritmos") seguido de sinónimos ("paso a paso", "herramienta de seguimiento", "visualización", etc.) para garantizar mayor cantidad de resultados. Dicha cadena de búsqueda se empleó en las siguientes bases de datos electrónicas: SCOPUS, ScienceDirect, IEEE Xplore y ACM Digital Library. Además, se establecieron criterios de inclusión como: estudios con validación, artículos completos, publicados en cualquier año, etc., y criterios de exclusión como: artículos no relacionados con educación, documentos como presentaciones o resúmenes, etc. Finalmente se realizó la extracción de datos y se presenta un análisis de las características de las herramientas de visualización encontradas. Entre los visualizadores reportados para abordar un CS1 se encuentran:

Collece 2.0. Utiliza un entorno colaborativo y síncrono con un sistema de control de versiones donde los usuarios pueden trabajar proyectos completos con credenciales de usuario para solucionar problemas de programación mediante la edición, compilación y ejecución de programas en Java y en C de forma concurrentemente (Lacave et al., 2019).

FLINT. Es el acrónimo de Flowchart Interpreter y a través de una interfaz simple permite construir

algoritmos utilizando diagramas de flujo, los cuales pueden seguirse combinando herramientas gráficas y valores para variables (Silva et al., 2016).

FreeDfd. Antes llamado Smart DFD y es un editor, depurador e intérprete de diagramas de flujo (Cárdenas et al., 1998), cuenta con opciones de seguimiento paso a paso y evaluación de variables en tiempo de ejecución (Del Prado & Lamas, 2014).

Jeliot 3. Visualiza programas codificados en Java, permite rastrear métodos, variables y operaciones mediante animación (UEF, 2020), posibilitando visualización total o semiautomática de los flujos de datos y control.(Sánchez et al., 2018).

PSeInt. Es el acrónimo de PSeudo Intérprete y permite construir algoritmos simples utilizando Pseudo-lenguaje con el propósito de centrar la atención del estudiante en los conceptos fundamentales de la algoritmia computacional (Novara, 2020). Cuenta con opciones de seguimiento paso a paso, variables y generación de diagrama de flujo.

Raptor. Desarrollado por el Departamento de Ciencias de la Computación de la Academia de la Fuerza Aérea de los Estados Unidos de América y es un entorno de programación basado en diagramas de flujo (Wilson et al., 2020) que permite visualizar el funcionamiento de un algoritmo y cuenta con herramientas de seguimiento y ejecución (Carlisle et al., 2005).

La tabla 1 presenta las características de Graficación, codificación y compilación que incorporan las herramientas descritas.

Software	Graficación			Codificación		Compilación		
	Flujograma	Diagrama E/S	Visualizador Interfaz de usuario	Pseudo código	Exportar código	Verificador estático	Impresora estética	Online
CodES	X	X	X	X	X	X	X	X
Collece 2.0					X		X	X
FLINT	X				X			
FreeDfd	X							
Jeliot 3					X			
PSeInt	X			X	X			
Raptor	X				X			

Tabla 1. Características software de visualización. Fuente: Elaboración propia.

3. Metodología

Este artículo describe a CodES y presenta los resultados del proceso investigativo mediante el cual se hace evaluaciones con estudiantes, expertos en usabilidad y experiencia de usuario (UX).

En lo relacionado con la metodología de investigación para evaluar CodES con estudiantes, se realizó bajo el paradigma positivista por lo que se fundamentó en el conocimiento científico, con enfoque cuantitativo que permitió examinar datos de manera numérica, utilizando el método empírico analítico porque los datos fueron tratados con técnicas estadísticas y bajo un tipo de investigación descriptivo que permitió medir el grado de relación que tuvieron las variables en estudio, además, se aplicó un diseño experimental con grupos de control y experimental utilizando post prueba con 147 estudiantes distribuidos en seis grupos diferentes pertenecientes a dos Universidades de la ciudad de Pasto (Col) en un CS1. Los datos obtenidos en esta etapa se sometieron a un análisis estadístico con la distribución de probabilidad T de Student, con el propósito de establecer la diferencia de los resultados de los grupos de control y experimentales.

Finalmente, CodES fue analizado mediante una prueba de Eye Tracking y evaluado por un profesional

experto mediante el Expert Review Checkpoint de Travis con el propósito de determinar su nivel de usabilidad.

4. Resultados

4.1. CodES

CodES basa su funcionamiento en el diagrama de entrada salida y propone una estrategia didáctica de enseñanza basada en tips o claves de identificación para que el estudiante reconozca si el enunciado del problema a solucionar es de transformación de salidas o tiene estructuras condicionales o cíclicas.

El entorno de trabajo de CodES consta principalmente de los siguientes componentes (Figura 1):

Toolbar. Provisto de cinco elementos básicos: entrada, proceso, salida, rótulo y continuidad; mediante los cuales se realiza todo el proceso de desarrollo inicial de pensamiento algorítmico.
Export. Permite exportar los algoritmos construidos en CodES a formato nativo java, C y Python.
BlackBox. Es el principal componente de CodES y es donde el usuario construye el diagrama de entrada salida con los 5 elementos básicos del Toolbar
Rich Graphical Interface. Permite visualizar cómo será la interfaz inicial computacional de los elementos ubicados en el BlackBox. Además, esta vista le presenta al estudiante cómo distribuir básicamente dichos elementos en una interfaz de usuario y los corresponde con el BlackBox mediante un código de colores.
Flow Chart. Visualiza los resultados del BlackBox mediante un diagrama de flujo, cuyos componentes se relacionan mediante un código de color con los elementos del Rich Graphical Interface y BlackBox.
Pseudocode. Presenta la codificación del algoritmo generado en el Flow Chart con sus correspondientes indicadores de colores.

Figura 1. Entorno CodES. Fuente: Elaboración propia.

Debido a que CodES está diseñado pensando en el desarrollo de habilidades algorítmicas para estudiantes de un CS1, consta de labels que le permiten enriquecer la interfaz de usuario con el propósito de obtener entornos de interacción apropiados desde los primeros programas computacionales, asegurando de esta manera involucrar elementos básicos pero importantes a la hora de tener un producto software que incluyen un título del aplicativo hasta rótulos que guíen tanto la captura como la impresión de datos, que generalmente se descuida en los primeros ejercicios de programación.

CodES no contempla la operación de asignación por lo que es necesario realizar las operaciones aritméticas en las salidas del BlackBox, esto le permite al estudiante asociar cada proceso con su correspondiente elemento y así fundamentar el desarrollo inicial del pensamiento algorítmico.

Jiménez Toledo, J. A.; Collazos, C.; Ortega Cantero, M. (2022). CodES: herramienta de visualización para desarrollo de pensamiento algorítmico. *Campus Virtuales,* 11(1), 21-33. https://doi.org/10.54988/cv.2022.1.809

Asimismo, CodES plantea un tip o clave de identificación para enseñar el concepto de condicional, el cual se basa en que al tener una salida con dos alternativas de solución se debe pensar en la utilización de un condicional compuesto, como lo presenta la Figura 2.

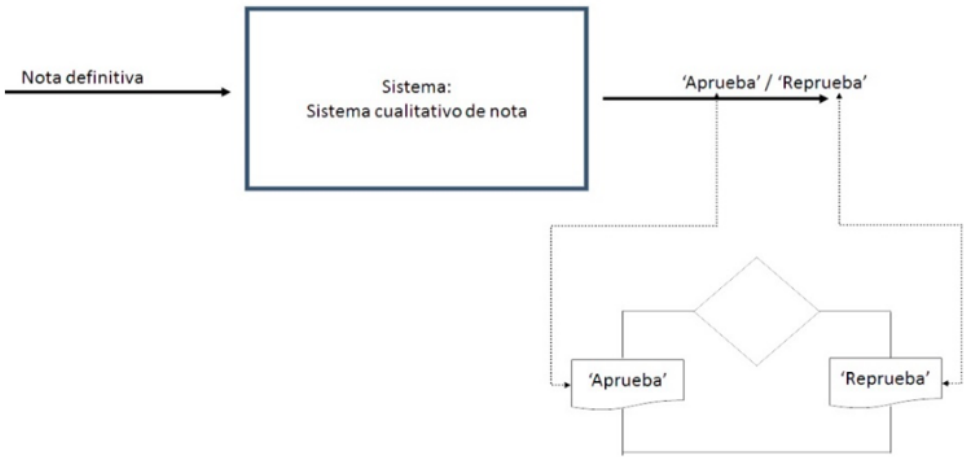

Figura 2. Condicional compuesto. Fuente: Elaboración propia.

Además, si una salida tiene más de dos alternativas de solución (n), se está ante un condicional compuesto donde n-1 será la cantidad de condiciones requeridas (Figura 3), claro está sin la utilización de operadores lógicos como el And o el Or.

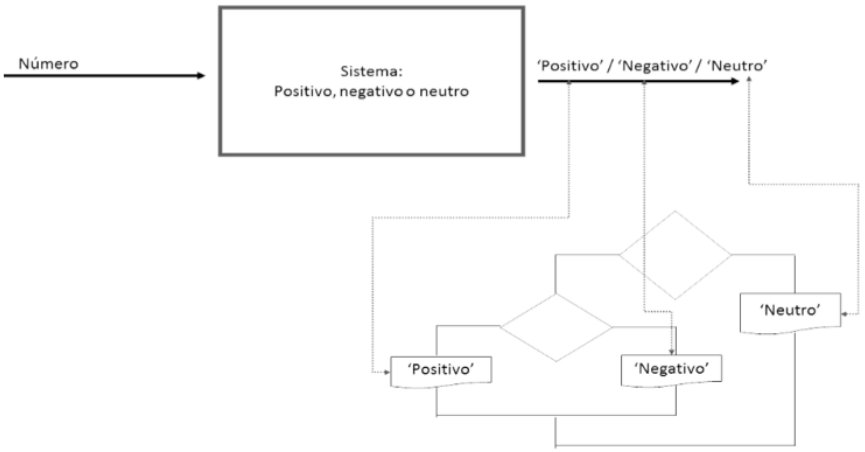

Figura 3. Condicional anidado. Fuente: Elaboración propia.

La Figura 4 presenta la implementación con CodES del ejemplo contenido en la Figura 2, teniendo en cuenta que el estudiante solo construye el BlackBox correspondiente y por cada instrucción se realiza la implementación respectiva en Rich Graphical Interface, Flow Chart y Pseudocode.

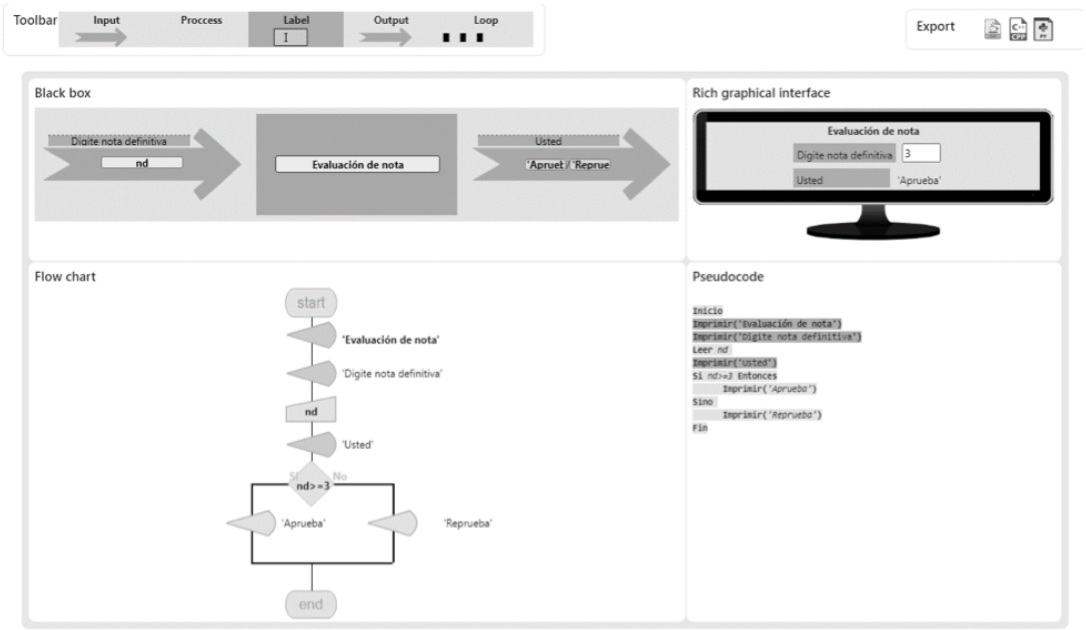

Figura 4. Condicional en CodES. Fuente: Elaboración propia.

Para la enseñanza de estructuras iniciales repetitivas conocidas como ciclos, CodES plantea la utilización del elemento continuidad representado con los tres puntos suspensivos, a través del cual en el BlackBox se plantea un diseño con al menos 4 elementos: una primera salida que indica el inicio, una segunda salida con la cual se calcula los pasos, el elemento de continuidad y finalmente una salida que determina el final, con estos elementos es posible identificar la estructura iterativa la cual puede ser visualizada en CodES mediante Ciclo Para, Mientras o Hacer mientras.

Además, se plantea una nueva extensión del concepto de ciclo de acuerdo a su estructura, para ello se propone la utilización del ciclo ascendente y descendente. El ciclo ascendente se caracteriza por a) el valor de inicio es mayor que el valor de fin, b) la condición o fin únicamente permite la utilización de los operadores relacionales menor o menor igual, c) los pasos son positivos; mientras que en el ciclo descendente a) el valor de inicio es menor que el valor de fin, b) la condición o fin únicamente permite la utilización de los operadores relacionales mayor o mayor igual y c) los pasos son negativos.

Una vez construido el BlackBox, CodES le permite al usuario evaluar dichas construcciones a través del Rich Graphical Interface, el cual permite capturar valores en las entradas, procesarlas y generar resultados. Asimismo, CodES se ejecuta Online e incorpora un verificador estático que le permite al estudiante corregir posibles errores al validar la escritura de variables y operaciones automáticamente y a la vez posee una impresora estética que utiliza código de colores y distribución de código adecuada tanto en BlackBox, Rich Graphical Interfaz, Flow chart y Pseudocódigo.

4.2. Evaluación de CodES

CodES fue evaluado bajo el paradigma positivista, con enfoque cuantitativo, utilizando el método empírico analítico, con un tipo de investigación descriptivo y mediante un diseño experimental con grupos de control y experimentales con post prueba. En el proceso investigativo con CodES participaron 147 estudiantes de dos universidades de la ciudad de San Juan de Pasto (Col) distribuidos en 6 grupos. En la tabla 2 se muestra el diseño experimental aplicado en cada universidad:

Universidad	Diseño Experimental
Universidad CESMAG - UniCesmag (Privada)	G_1 X O_1 G_2 - O_2
	G_3 X O_3 G_4 - O_4
Universidad de Nariño - UDENAR (Estatal)	G_5 X O_5 G_6 - O_6

Tabla 2. Diseño experimental por universidad. Fuente: Elaboración propia.

Los grupos G_1, G_3 y G_5 corresponden a los grupos experimentales de cada institución y G_2, G_4 y G_6 fueron sus grupos de control respectivamente, además X fue el tratamiento experimental que consistió en la estrategia didáctica de enseñanza con CodES. A su vez, O_1, O_2, O_3, O_4, O_5 y O_6 fueron las post pruebas realizadas al final del tratamiento experimental tanto para los grupos experimentales como los de control. En la tabla 3 se encuentra en detalle la información de los grupos participantes.

Universidad	Facultad	Programa	Curso	Semestre	Jornada
UniCesmag	Ingeniería	Ingeniería de Sistemas	Introducción a la programación	1	Diurna
	Ingeniería	Ingeniería de Sistemas	Introducción a la programación	1	Nocturna
UDENAR	Ingeniería	Ingeniería de Sistemas	Fundamentos de programación	1	Diurna

Tabla 3. Caracterización de los grupos participantes. Fuente: Elaboración propia.

El primer grupo experimental G_1 fue conformado por 33 estudiantes del curso Introducción a la programación de primer semestre de Ingeniería de Sistemas del segundo periodo académico del 2019 a los cuales se les suministró el tratamiento experimental X y finalmente se le aplicó una prueba posterior O_1. El grupo de control G_2 estuvo conformado por 27 estudiantes del periodo académico I-2019 del mismo semestre y curso del grupo experimental a quienes no se les aplicó tratamiento experimental y las notas obtenidas fueron consideradas como O_2.

El segundo grupo experimental G_3 estuvo constituido por 11 estudiantes de la jornada diurna de la misma universidad, curso y semestre que G_1, correspondientes al periodo académico I-2020 a quienes también se les aplicó el mismo tratamiento X con su pos evaluación O_3 donde su grupo de control G_4 fueron los 14 estudiantes de la jornada nocturna del mismo periodo académico, universidad, semestre y curso del grupo experimental a quienes tampoco se les aplicó CodES, considerando también las notas obtenidas como O_4.

El tercer grupo experimental G_5 estuvo constituido por 29 estudiantes del curso de Fundamentos de programación de primer semestre de Ingeniería de Sistemas correspondientes al periodo académico II-2019 a quienes también se les aplicó el mismo tratamiento X con su correspondiente evaluación O_5 y su grupo de control G_6 fueron los 33 estudiantes del periodo académico I-2018 del mismo semestre y curso del grupo experimental a quienes tampoco se les aplicó tratamiento experimental, considerando también las notas obtenidas como O_6. En la tabla 4 se aprecia la caracterización de los estudiantes participantes.

Grupo	Tipo	Estudiantes	Hombres	Mujeres	Edad (años)	Repiten asignatura
G_1	Experimental	33	27	6	18 a 21	0
G_2	Control	27	23	4	16 a 22	2
G_3	Experimental	11	11	0	18 a 35	0
G_4	Control	14	14	0	22 a 40	0
G_5	Experimental	29	24	5	18 a 20	0
G_6	Control	33	30	3	18 a 19	0

Tabla 4. Caracterización para estudiantes de Introducción a la Programación. Fuente: Elaboración propia.

Los grupos de control G_2, G_4 y G_6 fueron orientados por el mismo profesor utilizando la metodología de enseñanza tradicional donde se expone la temática de estudio, se llevan a cabo una serie de ejemplos y ejercicios para finalmente realizar evaluaciones.

El proceso investigativo para los tres grupos de control consistió en la realización de cuatro actividades evaluativas por cada unidad de competencia por cada semestre, que consistieron en la aplicación de dos talleres grupales con la participación de dos estudiantes (40%) y dos seguimientos individuales (60%), con el propósito de establecer cuantitativamente el proceso de apropiación de cada una de las unidades mencionadas con la metodología tradicional de clase. Los resultados de postprueba aplicados para los cursos de los grupos de control se muestran en la Tabla 5.

Grupo	Unidad de competencia	Taller 1	Taller 2	Total Talleres	Examen 1	Examen 2	Total Examen	Definitiva
G_2	Entrada/salida	4,30	4,00	4,15	3,80	4,00	3,90	4,00
	Condicionales	3,30	3,20	3,25	3,10	3,00	3,05	3,13
	Ciclos	4,00	3,80	3,90	3,70	3,60	3,65	3,75
G_4	Entrada/Salida	4,10	4,00	4,05	3,50	3,50	3,50	3,72
	Condicionales	3,90	3,80	3,85	3,70	3,20	3,45	3,61
	Ciclos	4,00	3,50	3,75	3,60	3,30	3,45	3,57
G_6	Entrada/salida	4,70	4,40	4,55	4,30	4,20	4,25	4,37
	Condicionales	4,30	4,00	4,15	4,00	4,00	4,00	4,04
	Ciclos	4,20	4,30	4,30	4,10	4,20	4,15	4,21

Tabla 5. Promedio notas grupos control. Fuente: Elaboración propia.

Los resultados de las notas mostradas en la Tabla 5 y que fueron obtenidos en los tres cursos para grupos de control, evidencian dificultad en el estudio de condicionales, un mejor resultado en el manejo de ciclos y una buena apropiación en el manejo de captura y salida de datos.

Los resultados definitivos evidencian de igual manera la influencia que tiene el proceso de selección de los estudiantes en la universidad pública, frente a los de las dos universidades privadas cuya política de ingreso es abierta.

Una vez aplicado el tratamiento investigativo en cada grupo experimental G_1, G_3 y G_5 utilizando CodES, se procedió a la realización de las mismas actividades evaluativas realizadas en los grupos de control. Los resultados de postprueba aplicados para los tres cursos muestran en la Tabla 6.

Grupo	Unidad de competencia	Taller 1	Taller 2	Total Talleres	Examen 1	Examen 2	Total Examen	Definitiva
G1	Entrada/salida	4,50	4,70	4,60	4,60	4,40	4,50	4,54
	Condicionales	4,10	4,00	4,05	3,90	3,80	3,85	3,93
	Ciclos	4,40	4,30	4,35	4,00	4,00	4,00	4,14
G3	Entrada/salida	4,40	4,80	4,60	4,20	4,40	4,30	4,42
	Condicionales	4,60	4,60	4,60	4,20	4,40	4,30	4,42
	Ciclos	4,50	4,40	4,45	4,30	4,10	4,20	4,30
G5	Entrada/salida	5,00	4,80	4,80	4,80	4,88	5,00	4,80
	Condicionales	4,90	4,30	4,60	4,45	4,63	4,90	4,30
	Ciclos	4,65	4,60	4,80	4,70	4,68	4,65	4,60

Tabla 6. Promedio notas grupos experimentales. Fuente: Elaboración propia.

Como se puede apreciar en la Tabla 6, los resultados obtenidos en los talleres (que son grupales y estuvieron conformados por 2 estudiantes) son superiores a los exámenes individuales, lo cual demuestra que el estudiante adquiere mejores resultados trabajando en grupo que haciéndolo individual, lo cual concluye que el proceso colaborativo llevado con el conjunto de estudiantes incide de manera positiva en sus resultados grupales.

De igual manera, se puede observar que la unidad de competencia relacionada con "Condicionales" es la que menor valor cuantitativo tiene frente a las otras dos unidades de competencia en los tres grupos experimentales, lo indica que los procesos lógicos que requieren evaluación de proposiciones tienen un cierto grado de dificultad para los estudiantes.

Por otro lado, se puede apreciar en los tres grupos que en la unidad de competencia relacionada con ciclos el estudiante se siente más a gusto ya que sus notas mejoran con relación a la unidad de condicionales.

Finalmente se puede concluir que en la unidad de competencia "Captura y salida de datos" hay una buena apropiación por parte del estudiante ya que es la que más valor cuantitativo tiene en cada grupo con relación a las otras unidades de competencia. Además, Los resultados de las notas están por encima de la escala cuantitativa de 4.0, lo cual demuestra que hay una apropiación buena por parte de los estudiantes en forma general a finalizar dichos cursos.

4.3. Discusión

Una vez terminado el proceso investigativo, se hace el paralelo entre el grupo de control y el experimental para cada uno de los cursos de este estudio. En la Tabla 7 se exhibe los resultados finales obtenidos por el grupo de control y experimental.

Grupo	Unidad de competencia	Grupo de control			Grupo experimental		
		Total Talleres	Total Examen	Definitiva	Total Talleres	Total Examen	Definitiva
G1	Entrada/Salida	4,15	3,90	4,00	4,60	4,50	4,54
G2	Condicionales	3,25	3,05	3,13	4,05	3,85	3,93
	Ciclos	3,90	3,65	3,75	4,35	4,00	4,14
G3	Entrada/Salida	4,05	3,50	3,72	4,60	4,30	4,42
G4	Condicionales	3,85	3,45	3,61	4,60	4,30	4,42
	Ciclos	3,75	3,45	3,57	4,45	4,20	4,30
G5	Entrada/Salida	4,55	4,25	4,37	4,80	5,00	4,80
G6	Condicionales	4,15	4,00	4,04	4,60	4,90	4,30
	Ciclos	4,30	4,15	4,21	4,80	4,65	4,60

Tabla 7. Grupo de control vs grupo experimental de Fundamentos de Programación. Fuente: Elaboración propia.

La Tabla 7 muestra la incidencia que tuvo el tratamiento experimental propuesto en esta investigación para el curso de Introducción a la Programación del Programa de Ingeniería de Sistemas de la Institución Universidad Cesmag. Los datos presentados evidencian que las notas obtenidas por los grupos experimentales G_1 y G_2, son mayores a las registradas por el grupo de control tanto en las actividades grupales (Talleres) como en las individuales (Exámenes).

Así mismo, muestra la incidencia que tuvo también el tratamiento experimental para el curso de Fundamentos de Programación del programa de Ingeniería de Sistemas de la Universidad de Nariño. Al igual que en la UniCesmag, los datos obtenidos evidencian que los resultados del grupo experimental son mejores a los del grupo de control.

Finalmente, se realiza un análisis estadístico para determinar mediante la distribución de probabilidad T de Student, la cual se utiliza para examinar la diferencia entre dos muestras independientes y pequeñas (Sanchez Turcios, 2015), la diferencia que existe entre las notas obtenidas por el grupo experimental y las del grupo de control.

En la Tabla 8 se muestra el análisis comparativo entre el grupo experimental G_1 y el de control G_2 respecto a las tres unidades de competencia planteadas en la investigación para el CS1 de Introducción a la programación.

Grupo	Concepto	Grupo experimental	Grupo de control
Entrada/Salida	Media	4,49575758	4,02962963
	Varianza	0,73820227	0,67062678
	Observaciones	33	27
	Varianza agrupada	0,70790981	
	Diferencia hipotética de las medias	0	
	Grados de libertad	58	
	Estadístico t	2,13490917	
	P(T<=t) una cola	0,01850238	
	Valor crítico de t (una cola)	1,67155276	
	P(T<=t) dos colas	0,03700477	
	Valor crítico de t (dos colas)	2,00171748	
Condicionales	Media	3,87545455	3,1437037
	Varianza	0,57988182	1,09901652
	Observaciones	33	27
	Varianza agrupada	0,81259738	
	Diferencia hipotética de las medias	0	
	Grados de libertad	58	
	Estadístico t	3,1281585	
	P(T<=t) una cola	0,0013754	
	Valor crítico de t (una cola)	1,67155276	
	P(T<=t) dos colas	0,0027508	
	Valor crítico de t (dos colas)	2,00171748	
Ciclos	Media	4,12484848	3,66148148
	Varianza	0,55394451	1,0152208
	Observaciones	33	27
	Varianza agrupada	0,76072353	
	Diferencia hipotética de las medias	0	
	Grados de libertad	58	
	Estadístico t	2,04726897	

Tabla 8. T de Student para G_1 y G_2 curso Introducción a la Programación. Fuente: Elaboración propia.

En la Tabla 8 se aprecia el resultado de la aplicación del tratamiento experimental en cada una de las unidades de competencia establecidas, de tal forma que G_1 posee un valor estadístico t (2,13490917, 3,1281585 y 2,04726897) mayor tanto al valor crítico de t de una cola (1,67155276 en cada uno) como al

valor crítico para dos colas (2,00171748 en cada uno), además, el registro para una y dos colas de P fue inferior al 5% en los tres casos, lo cual concluye que las notas obtenidas por G_1 frente a las del G_2 en cada temática son diferentes estadísticamente.

De igual manera, se aplicó T de Student para los grupos G_3 y G_4 también del curso de Introducción a la programación, donde G_3 obtuvo un valor estadístico t (2,31176389, 2,16226104 y 2,40061536) mayor tanto al valor crítico de t de una cola (1,71387153 en cada uno) como al valor crítico para dos colas (2,06865761 en cada uno), así también, el registro para una y dos colas de P fue inferior al 5% en los tres casos, concluyendo que las notas conseguidas por G_3 con relación a las del G_4 son diferentes estadísticamente en cada temática.

Por último, al aplicar T de Student en cada una de las unidades de competencia en G_5 y G_6 para el curso de Fundamentos de programación, el grupo G_5 obtuvo un valor estadístico t (2,10911757, 2,26839432 y 2,39107334) mayor tanto al valor crítico de t de una cola (1,67064886 en cada uno) como al valor crítico para dos colas (2,00029782 en cada uno), de igual forma, el registro para una y dos colas de P fue inferior al 5% también en los tres casos, así mismo, se concluye que las notas alcanzadas por el grupo G_5 son diferentes estadísticamente a las obtenidas por G_6 en cada temática.

Por lo tanto, el análisis estadístico anterior demuestra la incidencia que tiene CodES en esta investigación en los grupos experimentales frente a los grupos de control, estableciendo que el desarrollo de pensamiento algorítmico en un CS1 se puede obtener al incorporar CodES como una estrategia didáctica para disminuir la complejidad de las temáticas y así obtener resultados académicos que benefician de una manera directa a los estudiantes.

Además, para el primer ejemplo de estudio de entradas y salidas con CodES, correspondiente al cálculo del cuadrado de un número, se realizó una prueba de Eye Tracking con los 11 estudiantes del grupo experimental G_3 en la que puede observarse que el mapa de calor promedio del grupo indica su interés en primer lugar en la zona de trabajo de BlackBox, seguido del Rich Graphical Interface, luego del Flow Chart y finalmente del Pseudocódigo, como lo muestra la Figura 5.

Figura 5. Mapa de calor CodES. Fuente: Elaboración propia.

Asimismo, debido a que CodES funciona Online, se implementó el Expert Review Checkpoint de Travis (Travis, 2016) para usabilidad en entornos web, considerando 247 preguntas en 9 categorías, el cual fue evaluado por un experto obteniendo en promedio un 80,2% que lo categoriza como un buen resultado y con elementos por mejorar como su sistema de búsquedas y el diseño gráfico de su Frontend. Los resultados del test se presentan en la Tabla 9.

	Puntos	No. Preguntas	No. Respuestas	Puntaje
Página principal	17	20	20	85,0%
Orientación a tareas y funcionalidades del sitio	36	44	44	81,8%
Navegación	27	29	29	93,1%
Formularios e ingresos de datos	20	23	23	87,0%
Confianza y credibilidad	12	13	13	92,3%
Escritura y calidad de contenido	19	23	23	82,6%
Diseño páginas	29	38	38	76,3%
Búsquedas	8	20	20	40,0%
Ayuda, retroalimentación y tolerancia a fallos	31	37	37	83,8%
Total				80,2%

Tabla 9. Resultados de Expert Review Checkpoint para CodES. Fuente: Elaboración propia.

5. Conclusiones

CodES es una herramienta de visualización que permite desarrollar pensamiento algorítmico en un CS1 mediante el artefacto más simple de análisis computacional que es el diagrama de entrada y salida.

Además, CodES se puede utilizar en un CS1 con el propósito de generar proceso de abstracción en el estudiante novato tanto para el diseño de algoritmos como para su escritura.

Asimismo, este enfoque permite que el estudiante centre su atención en comprender en primer lugar el problema a solucionar a través de sus elementos esenciales y a la vez intuir desde un inicio la interfaz computacional a construir y sus respectivos diagramas de diseño y codificación

CodES utiliza tips o claves de identificación que le permitirán de una forma sencilla al estudiante el manejo de entradas y salidas e identificar cuando debe incorporar estructuras condicionales y repetitivas en sus algoritmos.

A su vez, los procesos de implementación de CodES en los 3 grupos experimentales permitieron evidenciar que estadísticamente los resultados son significativamente diferentes frente a los grupos de control, lo que permite sugerir el uso de CodES en un CS1 como una herramienta adicional para que el estudiante inicie el desarrollo de su pensamiento algorítmico.

Como trabajo futuro se pretende explorar el concepto de CodES en estudiantes de educación primaria y bachillerato dada la facilidad de aprendizaje propuesta en esta herramienta, además se pretende combinar CodES con estrategias de aprendizaje colaborativas y evaluar sus resultados. También, se proyecta adicionar nuevas funcionalidades para el manejo modular y cobertura de ejemplos de mayor complejidad computacional.

Agradecimientos

Especial agradecimiento a los integrantes de los grupos de investigación IDIS de Universidad del Cauca, CHICO de la Universidad Castilla - La Mancha y Tecnofilia de la Universidad CESMAG, los cuales permitieron la realización de esta investigación.

Cómo citar este artículo / How to cite this paper

Jiménez Toledo, J. A.; Collazos, C.; Ortega Cantero, M. (2022). CodES: herramienta de visualización para desarrollo de pensamiento algorítmico. *Campus Virtuales,* 11(1), 21-33. https://doi.org/10.54988/cv.2022.1.809

Referencias

Cárdenas, F.; Castillo, N.; Daza, E. (1998). Editor e intérprete de algorítmos representados en diagramas de flujo. Informática Educativa, 11(1), 101-106.

Carlisle, M.; Wilson, T.; Humphries, J.; Hadfield, S. (2005). Raptor: A visual programming environment for teaching algorithmic problem solving. SIGCSE Bull, 37(1), 176-180.

Costelloe, E. (2004). Teaching Programming. The State of the Art. In CRITE Technical Report (Institute). (https://www.scss.tcd.ie/disciplines/information_systems/crite/crite_web/publications/sources/programmingv1.pdf).

Dann, W.; Copper, S.; Pausch, R. (2006). Learning to program with Alice. Upper Saddle River, NJ.: P. Hall (ed.).

Del Prado, A.; Lamas, N. (2014). Alternativas para la enseñanza de pseudocódigo y diagrama de flujo. Riect, 5(3), 102-113.

Insuasti, J. (2016). Problemas de enseñanza y aprendizaje de los fundamentos de programación * Problems of teaching and learning the basics of programming Problemas de ensino e aprendizagem dos fundamentos de programação. Revista Educación y Desarrollo Social, 10(2), 12011-15318. doi:10.18359/reds.1701.

Jiménez-Toledo, J. A.; Collazos, C.; Revelo-Sánchez, O. (2019). Considerations in the teaching-learning processes for a first course in computer programming: a systematic review of the literature. TecnoLógicas, 22, 83-117. doi:10.22430/22565337.1520.

Jiménez, J.; Collazos, C.; Hurtado, J.; Pantoja, W. (2015). Collaborative strategy in three-dimensional environments as a didactic strategy for learning iterative structures in computational programming. Investigium IRE Ciencias Sociales y Humanas, 6(2), 80-92. doi:10.1007/s13398-014-0173-7.2.

Lacave, C.; Garcia, M. A.; Molina, A. I.; Sanchez, S.; Redondo, M. A.; Ortega, M. (2019). COLLECE-2.0: A real-time collaborative programming system on Eclipse. 2019 International In Symposium on Computers in Education (SIIE) (pp. 1-6). doi:10.1109/SIIE48397.2019.8970132.

Malliarakis, C.; Satratzemi, M.; Xinogalos, S. (2014). Educational games for teaching computer programming. Research on E-Learning and ICT in Education: Technological, Pedagog- Ical and Instructional Perspectives. Springer, 1(June), 99-118. doi:10.1007/978-1-4614-6501-0.

Novara, P. (2020). PSeInt. PSeInt. (http://pseint.sourceforge.net/).

Ortega, M.; Redondo, M. A.; Molina, A. I.; Bravo, C.; Lacave, C.; Arroyo, Y.; Sánchez, S.; García, M. A.; Collazos, C. A.; Jiménez, J. J.; Luna-García, H.; Velázquez-Iturbide, J. A.; Gómez-Pastrana, R. A. (2017). IProg: Development of immersive systems for the learning of programming. In ACM International Conference Proceeding Series, Part F1311, 6. doi:10.1145/3123818.3123874.

Sánchez, S.; García, M. Á.; Lacave, C.; Molina, A. I.; González, C.; Vallejo, D.; Redondo, M. Á.; Sanchez, E. S.; Gmarin, M.; Lacave, C.; Molina, A.; Gonzalez, C.; Vallejo, D.; Redondo, M. (2018). Applying Mixed Reality Techniques for the Visualization of Programs and Algorithms in a Programming Learning Environment. In ELmL 2018 : The Tenth International Conference on Mobile, Hybrid, and On-Line Learning Applying (pp. 84-89).

Sanchez Turcios, R. A. (2015). T-Student, usos y abusos. Revista Mexicana de Cardiologia, 26(1), 59-61.

Silva, G.; Arjona, P.; Castillo, F. (2016). More Time or Better Tools? A Large-Scale Retrospective Comparison of Pedagogical Approaches to Teach Programming. IEEE Transactions on Education, 59(4), 274-281. doi:10.1109/TE.2016.2535207.

Travis, D. (2016). 247 web usability guidelines. Userfocus. (https://www.userfocus.co.uk/resources/guidelines.html).

UEF (2020). Jeliot 3. (https://cs.joensuu.fi/jeliot/description.php).

Wilson, T.; Carlisle, M.; Humphries, J.; Moore, J. (2020). RAPTOR home page. Raptor. (https://raptor.martincarlisle.com/).