

ESTUDIO COMPARATIVO ENTRE
LENGUAJES TEXTUALES Y
LENGUAJES VISUALES CASO:
PiCO y GraPiCO

ESTUDIO COMPARATIVO ENTRE

LENGUAJES TEXTUALES Y LENGUAJES VISUALES CASO: **PiCO y GraPiCO**

COMPILADOR

Carlos Andrés Tavera Romero

AUTORES

Luis Eduardo Espinosa Galliady
Christian Felipe Cano Castillo
Carlos Andrés Tavera Romero
Yenny Viviana Cruz Pérez
Juan David Penagos Muñoz
Paola Andrea Ramírez Arcila
Marco Antonio Triana Lozano



Estudio comparativo entre lenguajes textuales y lenguajes visuales: caso PiCO y GraPiCO / Carlos Andrés Tavera Romero [y otros]. -- Editor Edward Javier Ordoñez. -- Cali : Universidad Santiago de Cali, 2018.

149 páginas ; 24 cm.

Incluye índice de contenido.

1. Ingeniería de sistemas 2. Lenguajes de programación (Computadores) I. Tavera Romero, Carlos Andrés. II. Ordoñez, Edward Javier, editor.

005.13 cd 22 ed.

A1618660

CEP-Banco de la República-Biblioteca Luis Ángel Arango



Estudio Comparativo entre Lenguajes Textuales y Lenguajes Visuales

Caso: PiCO y GraPiCO.

© Universidad Santiago de Cali.

© **Autores:** Luis Eduardo Espinosa Galliady, Christian Felipe Cano Castillo, Carlos Andrés Tavera Romero, Yenny Viviana Cruz Pérez, Juan David Penagos Muñoz, Paola Andrea Ramírez Arcila y Marco Antonio Triana Lozano.

1a. Edición **100 ejemplares**

Cali, Colombia - 2018

ISBN: 978-958-5522-41-1

ISBN (Libro digital): 978-958-5522-42-8

Fondo Editorial

University Press Team

Carlos Andrés Pérez Galindo

Rector

Rosa del Pilar Cogua Romero

Directora General de Investigaciones

Edward Javier Ordoñez

Editor en Jefe

Comité Editorial

Editorial Board

Rosa del Pilar Cogua Romero

Monica Chávez Vivas

Edward Javier Ordoñez

Luisa María Nieto Ramírez

Sergio Molina Hincapie

Saúl Rick Fernández Hurtado

Sergio Antonio Mora Moreno

Francisco David Moya Cháves

Evaluación de contenidos/Peer-review

outcome:

Febrero (February) de 2018

Correcciones de autor/Improved version

submission:

Junio (June) de 2018

Aprobación/Acceptance:

Septiembre (September) de 2018

Diseño y diagramación

Juan Diego Tovar Cardenas

Universidad Santiago de Cali

Tel. 5183000 - Ext. 322

Cel. 301 439 7925

Impresión

SAMAVA EDICIONES

Tel: (2) 8235737

Distribución y Comercialización

Universidad Santiago de Cali

Publicaciones

Calle 5 No. 62 - 00

Tel: 518 3000, Ext. 323 - 324 - 414



La editorial de la Universidad Santiago de Cali se adhiere a la filosofía del acceso abierto y permite libremente la consulta, descarga, reproducción o enlace para uso de sus contenidos, bajo una licencia de Creative Commons Reconocimiento-NoComercial-SinObrasDerivadas 4.0 Internacional.

<http://creativecommons.org/licenses/by-nc-nd/4.0/>

CONTENIDO

AGRADECIMIENTOS 19

INTRODUCCIÓN 19

ELEMENTOS PRELIMINARES

Christian Felipe Cano Castillo

Luis Eduardo Espinosa Galliady

Carlos Andrés Tavera Romero 21

Capítulo 1

ELABORACIÓN DE HIPÓTESIS EN EXPERIMENTOS DE LENGUAJES DE PROGRAMACIÓN

Christian Felipe Cano Castillo

Luis Eduardo Espinosa Galliady

Carlos Andrés Tavera Romero 37

Capítulo 2

VARIABLES EN UN EXPERIMENTO DE LENGUAJES DE PROGRAMACIÓN

Christian Felipe Cano Castillo

Luis Eduardo Espinosa Galliady

Carlos Andrés Tavera Romero 53

Capítulo 3

UNIDADES EXPERIMENTALES UTILIZADAS EN PRUEBAS DE LENGUAJES DE PROGRAMACIÓN

Christian Felipe Cano Castillo

Luis Eduardo Espinosa Galliady

Carlos Andrés Tavera Romero 65

Capítulo 4

TRATAMIENTOS Y RÉPLICAS EN UN EXPERIMENTO DE PROGRAMACIÓN

Christian Felipe Cano Castillo

Luis Eduardo Espinosa Galliady

Carlos Andrés Tavera Romero 85

Capítulo 5

APRENDIZAJE BASADO EN PROBLEMAS (ABP) APLICADO A LOS LENGUAJES DE PROGRAMACIÓN

Christian Felipe Cano Castillo

Luis Eduardo Espinosa Gallady

Carlos Andrés Tavera Romero..... 103

Capítulo 6

LA COMUNICACIÓN EN EL ESTUDIO COMPARATIVO ENTRE LENGUAJES TEXTUALES Y LENGUAJES VISUALES

CASO: PiCO y GraPiCO

Yenny Viviana Cruz Pérez

Carlos Andrés Tavera Romero..... 121

Capítulo 7

SISTEMATIZACIÓN DE UNA EXPERIENCIA DE INVESTIGACIÓN ENTRE LA COMUNICACIÓN SOCIAL Y LA INGENIERÍA DE SOFTWARE

Paola Andrea Ramírez Arcila

Juan David Penagos Muñoz

Yenny Viviana Cruz Pérez

Carlos Andrés Tavera Romero..... 137

Capítulo 8

MODELO DE SISTEMATIZACIÓN PROPUESTO “TCACI EN DOBLE VÍA”

Paola Andrea Ramírez Arcila

Juan David Penagos Muñoz

Yenny Viviana Cruz Pérez

Carlos Andrés Tavera Romero..... 153

Capítulo 9

PASOS EN LA REALIZACIÓN DE LOS AUDIOVISUALES PEDAGÓGICOS: PiCO y GraPiCO Y EJERCICIO DE MODELACIÓN

Paola Andrea Ramírez Arcila

Juan David Penagos Muñoz

Yenny Viviana Cruz Pérez

Carlos Andrés Tavera Romero..... 169

Capítulo 10

ESTUDIO DE RESULTADOS DE PiCO y GraPiCO PARTE 1

Luis Eduardo Espinosa Gallady

Marco Antonio Triana Lozano

Carlos Andrés Tavera Romero..... 179

Capítulo 11

ESTUDIO DE RESULTADOS DE PiCO y GraPiCO

PARTE 2

Marco Antonio Triana Lozano

Carlos Andrés Tavera Romero..... 209

Capítulo 12

RECOMENDACIONES Y RESULTADOS DEL ESTUDIO COMPARATIVO

ENTRE PiCO, GraPiCO Y LOS EDITORES

Christian Felipe Cano Castillo

Luis Eduardo Espinosa Galliady

Carlos Andrés Tavera Romero..... 233

ACERCA DE LOS AUTORES 245

ANEXOS 247

ENCUESTA PiCO 249

ENCUESTA GraPiCO 257

PARES EVALUADORES 269

ÍNDICE DE FIGURAS

Figura 1. Posibles ocurrencias de errores.....	42
Figura 2. Regiones y Tipos de Prueba de Hipótesis.....	43
Figura 3. Desarrollo de la Prueba de Hipótesis.....	45
Figura 4. Elementos del estudio comparativo que intervienen para lograr el objetivo planteado.....	46
Figura 5. Aplicación E_GraPiCO.....	47
Figura 6. Editor PSPad utilizado para PiCO.....	48
Figura 7. Clasificación de las Variables.....	55
Figura 8. Aprendizaje Colaborativo.....	109
Figura 9. Modelo de comunicación de Juan C. Asinsten.....	145
Figura 10. Estaciones del modelo.....	156
Figura 11. Modelo de comunicación de Juan C. Asinsten.....	161
Figura 12. Nivel de conocimiento en lenguajes de programación de encuestados tanto para PiCO como para GraPiCO.....	184
Figura 13. Tipo de lenguaje de programación preferido de encuestados tanto para PiCO como para GraPiCO.....	184
Figura 14. Nivel de conocimiento en lenguajes de programación visual y textual diferenciado por PiCO y GraPiCO.....	185
Figura 15. Frecuencia de uso de lenguajes de programación visual y textual diferenciado por PiCO y GraPiCO.....	186
Figura 16. Interés en lenguajes de programación visual y textual diferenciado por PiCO y GraPiCO.....	186
Figura 17. Nivel de comprensión de constructor “Program” diferenciado por PiCO y GraPiCO.....	187
Figura 18. Nivel de comprensión de constructor “Context” diferenciado por PiCO y GraPiCO.....	188
Figura 19. Nivel de comprensión de constructor “Objects” diferenciado por PiCO y GraPiCO.....	189
Figura 20. Nivel de comprensión de constructor “Methods” diferenciado por PiCO y GraPiCO.....	190

Figura 21. Nivel de comprensión de constructor “Ask” diferenciado por PiCO y GraPiCO.....	191
Figura 22. Nivel de comprensión de constructor “Tell” diferenciado por PiCO y GraPiCO.....	192
Figura 23. Nivel de comprensión de constructor “MsgSend” diferenciado por PiCO y GraPiCO.	192
Figura 24. Nivel de comprensión de constructor “Value”	193
diferenciado por PiCO y GraPiCO.	193
Figura 25. Nivel de comprensión de constructor “Variable” diferenciado por PiCO y GraPiCO.	194
Figura 26. Nivel de comprensión de constructor “Argument” diferenciado por PiCO y GraPiCO.	194
Figura 27. Nivel de comprensión de constructor “Sender” diferenciado por PiCO y GraPiCO.	195
Figura 28. Nivel de comprensión de constructor “Forward” diferenciado por PiCO y GraPiCO.	196
Figura 29. Nivel de comprensión de constructor “Operators” diferenciado por PiCO y GraPiCO.	197
Figura 30. Nivel de comprensión de constructor “Relations” diferenciado por PiCO y GraPiCO.	198
Figura 31. Nivel de comprensión de constructor “Constraints” diferenciado por PiCO y GraPiCO.	198
Figura 32. Nivel de Claridad de constructores diferenciado por PiCO y GraPiCO.	199
Figura 33. Nivel de Simbología de constructores diferenciado por PiCO y GraPiCO.	200
Figura 34. Nivel de Navegación de constructores diferenciado por PiCO y GraPiCO.	201
Figura 35. Nivel de Interés de acuerdo a la estabilidad, diferenciado por PiCO y GraPiCO.	201
Figura 36. Nivel de Interés de acuerdo al diseño, diferenciado por PiCO y GraPiCO.....	202
Figura 37. Nivel de Interés de acuerdo a la modificabilidad, diferenciado por PiCO y GraPiCO.	203

Figura 38. Nivel de Interés de acuerdo a la usabilidad, diferenciado por PiCO y GraPiCO.....	203
Figura 39. Complemento de la Ingeniería en Sistemas y la Comunicación Social para explicar conceptos técnicos de los cálculos de programación GraPiCO y PiCO.....	204
Figura 40. Facilidad de entendimiento de los cálculos de programación GraPiCO y PiCO de acuerdo al material audiovisual generado.	204
Figura 41. Nivel de estética de la dinámica audiovisual expuesta.....	205
Figura 42. Facilidad de entendimiento del ejercicio de modelación en el taller de los cálculos de programación GraPiCO y PiCO.....	205
Figura 43. Nivel de conocimiento en lenguajes de programación de encuestados tanto para PiCO como para GraPiCO.....	223
Figura 44. Tipo de lenguaje de programación preferido de encuestados tanto para PiCO como para GraPiCO.....	224
Figura 45. Nivel de conocimiento en lenguajes de programación visual y textual diferenciado por genero.....	225
Figura 46. Nivel de conocimiento en lenguajes de programación de encuestados tanto para PiCO como para GraPiCO.....	226
Figura 47. Tipo de lenguaje de programación preferido de encuestados tanto para PiCO como para GraPiCO.....	227
Figura 48. Nivel de conocimiento en lenguajes de programación visual y textual diferenciado por edad.....	228

ÍNDICE DE FIGURAS

Tabla 1. Etapas del estudio comparativo	22
Tabla 2. Datos típicos para un experimento de un solo factor.....	25
Fuente: elaboración propia.....	25
Tabla 3. Datos generados en la ejecución de una réplica.....	26
Tabla 4. Forma de graficar los resultados respecto al valor experimental	27
Tabla 5. Ecuaciones para la media de acuerdo al tratamiento y bloque.....	27
Tabla 6. Forma de graficar los resultados respecto a los tratamientos y sus réplicas	28
Tabla 7. Ecuaciones para calcular la gran media o media total.....	28
Tabla 8. Etapas del estudio comparativo.....	38
Tabla 9. Tipos de Errores.....	41
Tabla 10. Etapas del Estudio Comparativo.....	54
Tabla 11. Etapas del Estudio Comparativo.....	66
Tabla 12. Tratamientos Realizados para GraPiCO y PiCO.....	76
Tabla 13. Tratamientos Realizados para GraPiCO y PiCO.....	78
Tabla 14. Otros tratamientos realizados.....	80
Tabla 15. Etapas del estudio comparativo.....	86
Tabla 16. Formas de representar los tratamientos de experimentos de un factor.....	87
Tabla 17. Formas de representar los tratamientos de experimentos de dos factores.....	88
Tabla 18. Tratamientos realizados para GraPiCO y PiCO usando el sexo y la edad.....	90
Tabla 19. Tratamientos realizados para GraPiCO y PiCO usando el semestre y la universidad.....	94

Tabla 20. Tratamientos realizados para GraPiCO y PiCO de manera comparativa	98
Tabla 21. Etapas del estudio comparativo	104
Tabla 22. Etapas del estudio comparativo	123
Tabla 23. Etapas del estudio comparativo	140
Tabla 24. Etapas del estudio comparativo	154
Tabla 25. Etapas del estudio comparativo	170
Tabla 26. Etapas del estudio comparativo	180
Tabla 27. Etapas del estudio comparativo	210
Tabla 28. Etapas del estudio comparativo	234

El estudio generador de este proyecto consiste en un análisis comparativo entre lenguajes textuales y lenguajes visuales: PiCO y GraPiCO. Sus objetivos son: establecer de manera básica qué percepción tiene un usuario frente a estos dos tipos de lenguajes y tratar de encontrar bajo qué circunstancias se considerará cuando un tipo de lenguaje se comporta mejor con respecto al otro. Para cumplir con ello, se realizará la medición de los niveles de asimilación, comprensión y aceptación de estos dos lenguajes de programación por parte de una población escogida específicamente de acuerdo a ciertos parámetros.

Agradecimientos

El presente documento es el resultado de un trabajo post doctoral financiado por la Universidad de San Buenaventura y la Universidad Autónoma de Occidente y apoyado por los grupos de investigación, LIDIS de la Universidad de San Buenavenura y COMBA I+D de la Universidad Santiago de Cali.

El proyecto es multidisciplinario, pues convergen las temáticas de Ingeniería de Sistemas, la Comunicación Social y la Estadística. Y también es interinstitucional, dado que unieron fuerzas el programa de Comunicación Social-Periodismo y el área de Estadística de la Facultad de Ciencias Básicas de la Universidad Autónoma de Occidente, así como los grupos de investigación LIDIS de la Universidad de San Buenaventura-Cali y y COMBA I+D de la Universidad Santiago de Cali.

Queremos agradecer a las instituciones, facultades, programas de estudio y grupos de investigación que aportaron para la realización del proyecto.

Gracias a la Universidad de San Buenaventura y a la Universidad Autónoma de Occidente quienes con la creación de un convenio marco posibilitaron la ejecución de la investigación.

Gracias a la Dirección de Investigaciones de la Universidad de San Buenaventura – Cali por apoyar el proyecto y por la gestión que permitió la financiación del mismo.

Gracias a la Dirección General de Investigaciones de la Universidad Santiago de Cali por su determinante apoyo y confianza en el proyecto.

Gracias a la Facultad de Ingeniería en especial al programa de Ingeniería de Sistemas de la Universidad de San Buenaventura y a la Facultad de Comunicación y Ciencias Sociales de la Universidad Autónoma de Occidente y su programa de estudios de Comunica-

ción Social y Periodismo por propender que los docentes investigadores que participaron en el proyecto contasen con los recursos necesarios.

Gracias al Grupo de Investigación LIDIS de la Universidad de San Buenaventura por poner a disposición del proyecto su experiencia y recursos, y gracias al Grupo de Investigación COMBA I+D de la Universidad Santiago de Cali por dar el apoyo y la ayuda decisivos para la culminación del proyecto.

INTRODUCCIÓN

En los años 90 surgió un movimiento en la disciplina de la computación, los lenguajes de programación visual.

La idea inicial era que por medio específicamente y únicamente de imágenes poder editar complejos programas de computadora. Es por esto que el último decenio del siglo XX fue una temporada de muchos intentos por desarrollar la próxima generación de lenguajes de programación.

Infortunadamente, los computadores en ese entonces no tenían el poder de cómputo requerido por las nuevas teorías de programación, es por esto que el avance en lenguajes de programación visual fue fundamentalmente teórico.

Pero ya en el siglo XXI las nuevas computadoras personales son incluso hasta 10 veces más potentes que las últimas del siglo pasado, permitiendo procesar complicados programas donde la parte visual tiene un rol protagónico.

Uno de estos desarrollos fue precisamente el cálculo computacional textual llamado PiCO y su posterior generación el cálculo visual GraPiCO.

Estos dos cálculos fueron propuestos por el grupo de investigación AVISPA de la Universidad del Valle y la Pontificia Universidad Javeriana de Cali, apenas ahora se pueden realizar programas netamente visuales, empleando el programa E_GraPiCO que es el editor visual de cálculo PiCO.

El editor E_GraPiCO tuvo al nacer recibimientos diametralmente opuestos, el grupo que gustan de lo textual y el grupo que apoya los lenguajes visuales.

Por esta pugna es que surge el interrogante, entre dos lenguajes semanticamente equivalentes uno textual y otro visual ¿qué lenguaje tiene mejor percepción de parte del usuario final?

Es así como surge la idea de realizar un estudio comparativo entre lenguajes visuales y lenguajes textuales por medio del cálculo textual PiCO y el cálculo visual GraPiCO.

Como público objetivo del estudio se eligió a los estudiantes de Ingeniería de Sistemas de las principales universidades del suroccidente colombiano.

Este informe científico relata paso a paso ¿cómo fue que se realizó el experimento? y ¿qué conclusiones se pudieron dilucidar?

En el presente documento se empleará la norma de la asociación IEEE para las citas y la presentación de las entradas en la sección Bibliografía, esta decisión se tomó dado que el grupo de investigación COMBA I+D de la Universidad Santiago de Cali es sede de la corporación mencionada.

ELEMENTOS PRELIMINARES

Christian Felipe Cano Castillo
Luis Eduardo Espinosa Galliad
Carlos Andrés Tavera Romero

I. Introducción

El estudio generador de este proyecto consiste en un análisis comparativo entre lenguajes textuales y lenguajes visuales: PiCO y Gra-PiCO. Sus objetivos son:

- Establecer de manera básica qué percepción tiene un usuario frente a estos dos tipos de lenguajes.
- Encontrar bajo qué circunstancias se considerará cuando un tipo de lenguaje se comporta mejor con respecto al otro.

Para cumplir con ello, se realizará la medición de los niveles de asimilación, comprensión y aceptación de estos dos lenguajes de programación por parte de una población escogida específicamente de acuerdo a ciertos parámetros.

Durante el desarrollo de este estudio se realizará una serie de pasos, que serán explicados en etapas posteriores. Ver Tabla 1.

Tabla 1. Etapas del estudio comparativo

Estudio comparativo entre lenguajes textuales y lenguajes visuales. Caso: PiCO y GraPiCO	
Etapa 1	Elaboración de hipótesis en experimentos de lenguajes de programación
Etapa 2	Variables en un experimento de lenguajes de programación
Etapa 3	Unidades experimentales utilizadas en pruebas de lenguajes de programación
Etapa 4	Tratamientos y replicas en un experimento de programación
Etapa 5	Aprendizaje Basado en Problemas (ABP) aplicado a los lenguajes de programación
Etapa 6	La comunicación en el estudio comparativo entre lenguajes textuales y lenguajes visuales: Caso PiCO y GraPiCO
Etapa 7	Sistematización de una experiencia de investigación entre la Comunicación Social y la Ingeniería de Software
Etapa 8	Modelo de sistematización propuesto “TCACI en doble vía”
Etapa 9	Pasos en la realización de los audiovisuales pedagógicos: PiCO – GraPiCO y ejercicio de modelación
Etapa 10	Estudio de resultados de PiCO y GraPiCO. Parte 1
Etapa 11	Estudio de resultados de PiCO y GraPiCO. Parte 2
Etapa 12	Recomendaciones y resultados del estudio entre PiCO, GraPiCO y editores

Fuente: elaboración propia (2018).

II. Marco teórico

Entre los aspectos incluidos en un *experimento* se encuentran: el *diseño* y el *análisis estadístico* de los datos producidos. ^{[5] [9]}¹

El *diseño del experimento* es algo muy importante a tener en cuenta durante la ejecución. Esto se debe a que esta etapa permite planificar todo lo concerniente con el estudio a realizar, obteniendo de una forma eficiente la máxima retroalimentación de los resultados generados y permitiendo que, de una forma fácil y adecuada, se pueda realizar un *análisis estadístico* enmarcado en conclusiones coherentes, justas e imparciales. ^{[5] [9]}

La importancia de esta fase radica en su utilidad a la hora de controlar los *factores perturbadores* (aspectos del *experimento* que son difíciles de manejar y pueden arriesgar su éxito), además de la presencia de cualquier *error* (el error producto de la ejecución de cualquier experimento se debe a las múltiples variaciones entre las *observaciones* o *mediciones* efectuadas) y la ejecución de diferentes técnicas y procedimientos existentes para la manipulación de las *unidades experimentales* (también conocidas como los *sujetos de estudio* o aquello que el estudio persigue analizar). ^{[5] [9] [4]}

A. Algunos diseños de experimentos

Existen diversas formas de diseñar experimentos: *Bloquización Totalmente Aleatoria*, *Cuadrados Latinos*, *Cuadrados Greco-Latinos*, *Experimento Factorial*, *Experimento Factorial 2^k*, *Experimentos Simples*, *Experimentos de Dos Factores*, *Estudios Por Observación*, *Experimento Comparativo* y *Experimento Aleatorio*, entre otras.

El *diseño* conocido como *Cuadrados Latinos* consiste en lograr que dentro de una matriz de *tratamientos* (condiciones en las que se analizarán las unidades experimentales), cada uno ocurra una vez por fila y una vez por columna. Se caracteriza por su utilidad en el control de dos fuentes de error ocurridas al mismo tiempo. ^{[5] [8] [4]}

¹ Como se comentó en la introducción, en el texto se empleará la norma IEEE para las citas bibliográficas.

El *planteamiento Cuadrados Greco-Latinos* reside en la superposición de dos *Cuadrados Latinos* entre sí. Un cuadrado se conforma por letras latinas y el otro por letras griegas, garantizando que cada letra griega sea asociada con una letra latina una sola vez. Este tipo de trasposición promueve el control de tres fuentes de error. ^{[5] [8] [4]}

Cuando en un *experimento* se considera que hay factores de interés que pueden presentar variaciones el bosquejo *Experimento Factorial* debería ser usado, debido a que en cada *réplica* (repetición del **experimento** básico) se tienen en cuenta todas las combinaciones de los niveles de los *tratamientos*. Este planteamiento surge como solución a tal interacción y significa que la diferencia existente entre sus niveles, no es igual para todos. ^{[5] [9] [4]}

En cuanto al planteamiento *Experimento factorial*, éste aparece si en un *estudio* se presentan múltiples *tratamientos* para ser estudiados. Cuenta con k factores y tiene una cantidad reducida de ejecuciones, a razón de que necesita de 2^k *observaciones* (o mediciones). ^{[5] [9] [4]}

Una forma de manipular adecuadamente las *unidades experimentales* es el diseño denominado *Bloquización*, que consiste en el agrupamiento de las unidades, mediante la asignación al azar de los diferentes *tratamientos* a los múltiples *bloques* existentes; de tal forma, que los tratamientos tengan la misma probabilidad de ocurrencia dentro de cada *bloque* y permitiendo aglutinar las *observaciones* de forma homogénea, de acuerdo a los *tratamientos* o a los *bloques*. ^{[5] [8] [4]}

Por su parte, los *estudios por observación* son aquellos para los cuales se desea hacer un *experimento* que no se puede ejecutar por cuestiones éticas. Esto último puede darse porque el investigador tiene en mente ejecutar algunas *restricciones* (tratamientos) sobre los *sujetos de estudio* hasta obtener las respuestas necesarias. Estos *tratamientos* son grupos que autoseleccionan dichas *unidades experimentales*. ^{[5] [7] [4]}

Los *experimentos aleatorios* se caracterizan porque sin importar que las condiciones de ejecución sean casi idénticas, no siempre arrojan resultados iguales, Por ello, constantemente permiten contar

con datos considerados como *muestras aleatorias*.^{[5] [6] [7] [8] [4]} Ver Tabla 2.

Tabla 2. Datos típicos para un experimento de un solo factor.[9]

Tratamiento	Observaciones			Totales	Promedios
1	y_{11}	y_{12}	y_{1n}	$y_{1.}$	$\bar{y}_1.$
2	y_{21}	y_{22}	y_{2n}	$y_{2.}$	$\bar{y}_2.$
\vdots	\vdots	\vdots	\vdots	\vdots	\vdots
a	y_{a1}	y_{a2}	y_{an}	$y_{a.}$	$\bar{y}_a.$
				$y_{..}$	$\bar{y}_{..}$

Fuente: elaboración propia (2018).

Para un *experimento* el uso de la *aleatorización* permite balancear el efecto de cualquier *variable* de ruido que pudiera influir en el resultado. En pocas palabras: facilita la disminución de diversas fuentes de error.^{[5] [6] [7]}

Si en la *experimentación* se da el caso de selección de ítems de un conjunto (ejecución de *muestreo*), se recomienda indicar la importancia que dentro del contexto del estudio tenga el orden de selección.^{[5] [6] [7]}

B. Técnica de clasificación de los experimentos, según sus niveles o factores

La importancia de esta clasificación radica en su utilidad como herramienta para reducir las diversas fuentes de error; dado que, muchas veces durante el desarrollo de un *experimento* se presentan factores que pueden afectar la respuesta generada, porque sobre ellos no hay interés alguno, aún siendo conocidos y controlables.

Esta técnica también permite controlar con precisión tanto la variación dentro de los *tratamientos* con más de dos niveles, como la diferencia entre *bloques*.^[5]

Como consecuencia, los datos generados por la ejecución de una *réplica* (repetición del *experimento* básico) utilizando la *aleatorización* para un factor con “a” niveles y “b” bloques, tienen una representación característica.^[5] Ver Tabla 3.

Tabla 3. Datos generados en la ejecución de una *réplica*.^[9]

Tratamientos	Bloques				Totales	Promedios
	1	2	...	b		
1	y_{11}	y_{12}	...	y_{1b}	$y_{1.}$	$\bar{y}_{1.}$
2	y_{21}	y_{22}	...	y_{2b}	$y_{2.}$	$\bar{y}_{2.}$
⋮	⋮	⋮		⋮	⋮	⋮
A	y_{a1}	y_{a2}	...	y_{ab}	$y_{a.}$	$\bar{y}_{a.}$
Totales	$y_{.1}$	$y_{.2}$...	$y_{.b}$	$y_{..}$	$\bar{y}_{..}$
Promedios	$\bar{y}_{.1}$	$\bar{y}_{.2}$...	$\bar{y}_{.b}$		$\bar{y}_{..}$

Fuente: elaboración propia (2018).

El *Experimento Comparativo* ocurre cuando se establece un conjunto de restricciones sobre los objetos de estudio mediante dos factores: *Tratamientos* y *Bloques*, y se desea comparar entre sí sus diferentes respuestas. Para este tipo de diseño se tiene “a” *Tratamientos*, “b” *Bloques* y el respectivo valor experimental para cada combinación entre “a” y “b”. Estos, tienen una forma particular de ser representados.^[4] Ver Tabla 4.

Tabla 4. Forma de graficar los resultados respecto al valor experimental.^[8]

Tratamientos	Bloques				
	1	2	...	B	
1	x_{11}	x_{12}	...	x_{1b}	\bar{x}_1
2	x_{21}	x_{22}	...	x_{2b}	
.	\bar{x}_2
.	
.	\bar{x}_a
A	x_{a1}	x_{a2}	...	x_{ab}	
	\bar{x}_1	\bar{x}_2	...	\bar{x}_b	

Fuente: elaboración propia (2018).

En el *tratamiento* j y en el *bloque* k se denota el valor x_{jk} . Mientras que la media de los valores en la fila j , se define por \bar{x}_j donde $j = 1 \dots a$. Por su parte, la media de los valores en la columna k se identifica con \bar{x}_k donde $k = 1 \dots b$ y, finalmente, la gran media o media total se denota por \bar{x} .^[8] Ver Tabla 5.

Tabla 5. Ecuaciones para la media de acuerdo al tratamiento y bloque.^[8]

$\bar{x}_j = \frac{1}{b} \sum_{k=1}^b x_{jk}$	$\bar{x}_k = \frac{1}{a} \sum_{j=1}^a x_{jk}$	$\bar{x} = \frac{1}{ab} \sum_{j,k} x_{jk}$
---	---	--

Fuente: elaboración propia (2018).

Si en un *experimento* se establecen restricciones sobre las *unidades experimentales* mediante un solo factor (*Tratamiento*), nace el *Experimento Simple o de un Factor*. Este diseño tiene "a" grupos de *muestras* independientes con sus correspondientes *tratamientos*

para cada grupo y “b” réplicas de cada uno de los *tratamientos*^[5]. Ver Tabla 6.

Tabla 6. Forma de graficar los resultados respecto a los *tratamientos* y sus *réplicas*.^[8]

Tratamiento 1	x_{11}	x_{12}	...	x_{1b}	\bar{x}_1
Tratamiento 2	x_{21}	x_{22}	...	x_{2b}	\bar{x}_2
⋮	⋮	⋮		⋮	
Tratamiento a	x_{a1}	x_{a2}	...	x_{ab}	\bar{x}_a

Fuente: elaboración propia (2018).

Se tiene “a” filas y “b” columnas, denotando x_{jk} la medida en la fila “j” y la columna “k,” donde $j = 1, 2, \dots, a$ y $k = 1, 2, \dots, b$. La media de los valores en la fila j se define como \bar{x}_j donde $j = 1 \dots a$. Y, por último, la gran media o media total es la media de todas las medias en todos los grupos y se nombra como \bar{x} .^[8] Ver Tabla 7.

Tabla 7. Ecuaciones para calcular la gran media o media total.^[8]

$\bar{x}_j = \frac{1}{b} \sum_{k=1}^b x_{jk}$	$\bar{x} = \frac{1}{ab} \sum_{j,k} x_{jk} = \frac{1}{ab} \sum_{j=1}^a \sum_{k=1}^b x_{jk}$
---	--

Fuente: elaboración propia (2018).

III. Historia

Para el año 2000 el grupo AVISPA de Univalle-Javeriana, presenta el cálculo textual PiCO.

“En el año 2007, producto de un trabajo doctoral, se propone el cálculo visual GraPiCO, lenguaje expresivamente equivalente a PiCO; de esta misma investigación, nace la aplicación que permite la creación de programas GraPiCO: el editor visual E_GraPiCO. Se

propone como proyecto de investigación la extensión del resultado de la tesis antes mencionada, que consiste en la elaboración de un estudio comparativo entre los lenguajes visuales y textuales (los cálculos PiCO y GraPiCO)^{[1][2]}

E_GraPiCO fue creado siguiendo cada una de las especificaciones formales generadas en la semántica de PiCO y GraPiCO, gracias a esto fue posible la demostración de su corrección y completitud. Es por lo anterior, que E_GraPiCO es una aplicación estable y funcional. Pero requiere, en las recomendaciones de usuario final, un último conjunto de modificaciones de forma, surgidas de un estudio comparativo que permitirá apreciar las ventajas de la utilización de un editor visual frente a uno textual y, de igual forma, generará recomendaciones para el mejoramiento del editor de programas gráficos ya existente.

Al no contar con dos lenguajes disponibles de equivalencia expresiva demostrada formalmente, no existían elementos para efectuar un estudio como el que aquí se propone. Después del desarrollo de PiCO y GraPiCO, esta dificultad fue superada ^[3].

De otro lado, en la actualidad se discute mucho sobre qué tipo de lenguaje es mejor, el visual o el textual, porque con la llegada de procesadores cada vez más veloces, se posibilita desplegar información con mayor grado de complejidad. Esto ha traído como consecuencia que sistemas operativos y lenguajes de programación más visuales ya estén dentro del mercado y también hayan logrado entrar a la academia. Sin embargo, estos mismos procesadores que permiten la utilización de lenguajes visuales, también se convierten en su principal limitante al exigir el código textual para su análisis y almacenamiento, mediante los mecanismos conocidos en la actualidad. Lo anterior ha producido que los lenguajes visuales sean utilizados estrictamente para la presentación de información. Grupos de investigación internacional como el IRCAM de Francia y nacionales como AVISPA, han estado creando aplicaciones y desarrollando investigaciones relacionadas con los lenguajes de programación visuales y textuales, pero hasta el momento no se han adelantado estudios comparativos desde la visión de la didáctica, que permitan obtener información de la interacción humano-lenguaje de programación.

IV. Objetivos

Objetivo general:

Efectuar un estudio comparativo entre los editores de lenguajes visuales y los textuales caso PiCO y E_GraPiCO, teniendo en cuenta las preferencias y recomendaciones de usuario final.

Los objetivos específicos son:

- Realizar un *estudio comparativo* entre la edición de programas PiCO y GraPiCO.
- Realizar un análisis del *estudio comparativo* e identificación del conjunto de características que se deben mejorar.
- Socializar las conclusiones del *estudio comparativo*.

V. Diseño

Para la realización del proyecto y obtención de la información de interés sobre el estudio comparativo entre los lenguajes visuales y los lenguajes textuales caso PiCO y GraPiCO se siguieron los siguientes pasos:

1. Realización de dos productos audiovisuales didácticos que expliquen los conceptos básicos de los constructores necesarios para programar en PiCO y E_GraPiCO.
2. Diseño y selección de un ejemplo representativo, a manera de corpus lingüístico de PiCO y GraPiCO.
3. Realización de un producto audiovisual a partir del ejemplo representativo de utilización de los lenguajes PiCO y GraPiCO.
4. Elaboración de un cuestionario donde se evalúen las nociones tratadas en los tres productos audiovisuales (con respecto a los temas explicados se evaluará: comprensión de conceptos, aplicabilidad y percepción de la experiencia).

5. Ejecución de un conjunto de *experimentos* de acuerdo a los siguientes pasos:
 - a. Exposición de los tres productos audiovisuales a un público determinado.
 - b. Sesión de discusión y clarificación de dudas con el público participante.
 - c. Diligenciamiento del cuestionario evaluativo por parte del público.
 - d. Análisis y sistematización de los resultados del cuestionario.
6. Estudio de resultados de los experimentos.
7. Consolidación de resultados y socialización de conclusiones.

VI. Metodología

En la siguiente serie de etapas se llevará un paso a paso del proceso realizado como parte de este estudio comparativo.

Durante el desarrollo de la primera etapa se expondrán múltiples aspectos que giran en torno a la *hipótesis* del estudio (primer paso a definir en el diseño de un *experimento*), la ejecución de *la prueba de hipótesis* y la verificación.

Con respecto a la segunda etapa se presentarán algunos conceptos relacionados con las *variables* del estudio (aspecto clave en la realización de cualquier investigación), como la definición de *variables* y la clasificación (herramienta útil en el diseño previo a la ejecución del proyecto).

Ya en la tercera etapa, se exhibirán conceptos claves relacionados con las *unidades experimentales*, su definición y las múltiples formas de usarlas dentro del contexto de la *experimentación*. Se podrá visualizar el manejo que se ha dado a los sujetos de estudio, haciendo referencia a los *tratamientos* efectuados.

Para abordar la cuarta etapa se mostrarán algunas teorías acerca de la definición de *tratamientos* y su implementación. También se especificará la cantidad de *observaciones* o *réplicas* realizadas al respecto.

Como quinta etapa se analizará el estudio desde el punto de vista del *Aprendizaje Basado en Problemas*, exponiendo las diferentes posturas existentes y su contribución.

Mientras que en la sexta etapa se presentarán algunos resultados de los análisis estadísticos, aplicados a la información obtenida en el proceso de la encuesta de cada lenguaje y de sus respectivos constructores.

En séptima etapa se mostrarán otros resultados de los análisis estadísticos aplicados a la información obtenida en el proceso de la encuesta de cada lenguaje y sus respectivos constructores.

Y, por último, habrá una octava etapa donde se definirán las *observaciones* y recomendaciones finales del proyecto y del editor E_GraPiCO.

VII. Resultados Esperados

En este documento se ha presentado de manera general y a grandes rasgos el estudio comparativo entre lenguajes textuales y lenguajes visuales: Caso PiCO y GraPiCO. Más adelante, se expondrá la temática correspondiente a los diferentes momentos del estudio:

Inicialmente, se tendrán en cuenta diferentes teorías, con el objetivo de aceptar o rechazar la hipótesis: en cuanto a su planteamiento y comprobación.

También se trabajará en la definición de *variables* y su clasificación (herramienta útil en el diseño previo a la ejecución de un estudio).

Se seleccionarán los *sujetos de estudio* y el tratamiento que se les dará dentro de la investigación.

De igual forma, se describirá el uso dado a algunos *tratamientos* y las *observaciones* realizadas a estos.

En cuanto a la metodología pedagógica utilizada, se explicará el *Aprendizaje Basado en Proyectos*, el cual fue empleado en la realización del taller de modelamiento de los lenguajes de programación del presente estudio.

Y, por último, se presentará el análisis de los resultados obtenidos a partir de las encuestas como medio de retroalimentación.

VIII. Bibliografía

- [1] C. Tavera y J. Díaz, Nuevo cálculo visual: GraPiCO, En II Congreso Colombiano de Computación, Universidad Javeriana. Bogotá, 2007.
- [2] C. Tavera y J. Díaz, Breve Discusión de las Ventajas de los Lenguajes Visuales frente a los Textuales: Caso de Estudio el Cálculo GraPiCO, En III Congreso Colombiano de Computación. Medellín, 2008.
- [3] C. Tavera y J. Díaz, «Alternativa de Comprobación Sintáctica de VLP: Gsig Parsing. Aspectos formales y el caso de estudio: E GraPiCO,» In XIII Congreso Argentino de Ciencias de la Computación. Argentina, 2008.
- [4] R. O. Kuehl, Diseño de experimentos. Principios estadísticos para el diseño y análisis de investigaciones., Thomson Learning, México, 2000.
- [5] D. C. Montgomery, Diseño y análisis de esperimentos. México D.F.: Limusa, 2004.
- [6] R. E. Walpole, Probabilidad y estadística para ingenieros. México D.F: Prentice Hall, 1998.
- [7] A. L. Webster, Estadística aplicada a los negocios y la economía. Bogotá: Mc Graw Hill, 2001.
- [8] M. R. Spiegel, Probabilidad y Estadística, México D.F.: McGraw-Hill, 1977.
- [9] D. C. Montgomery y G. C. Runger, Applied Statistics and Probability for Engineers. United States of America: John Wiley & Sons, Inc.

ELABORACIÓN DE HIPÓTESIS EN EXPERIMENTOS DE LENGUAJES DE PROGRAMACIÓN

*Christian Felipe Cano Castillo
Luis Eduardo Espinosa Gallady
Carlos Andrés Tavera Romero*

I. Introducción

Luego de haber determinado el tipo de investigación que se iba a desarrollar, es necesario construir las bases teóricas desde el punto de vista de la estadística, para poder contar con la validez que ofrecen los métodos científicos.

En esta etapa se presentarán algunas teorías acerca de la prueba de hipótesis que se utilizará durante el transcurso del estudio.

II. Marco Teórico

Durante el progreso de este artículo se explicarán los aspectos que giran en torno a la *hipótesis* (primer paso a definir en el diseño de un *experimento*): la definición (especificando la *hipótesis nula* y la *alternativa*) y la ejecución de la *prueba de hipótesis*. Ver Tabla 8.

Tabla 8. *Etapas del estudio comparativo.*

Estudio comparativo entre lenguajes textuales y lenguajes visuales. Caso: PiCO y GraPiCO	
Etapa 1	Elaboración de hipótesis en experimentos de lenguajes de programación.
Etapa 2	Variables en un experimento de lenguajes de programación.
Etapa 3	Unidades experimentales utilizadas en pruebas de lenguajes de programación.
Etapa 4	Tratamientos y replicas en un experimento de programación.
Etapa 5	Aprendizaje Basado en Problemas (ABP) aplicado a los lenguajes de programación.
Etapa 6	La comunicación en el estudio comparativo entre lenguajes textuales y lenguajes visuales. Caso PiCO y GraPiCO.
Etapa 7	Sistematización de una experiencia de investigación entre la Comunicación Social y la Ingeniería de Software.
Etapa 8	Modelo de sistematización propuesto “TCACI en doble vía”.
Etapa 9	Pasos en la realización de los audiovisuales pedagógicos: PiCO – GraPiCO y ejercicio de modelación.
Etapa 10	Estudio de resultados de PiCO y GraPiCO. Parte 1.
Etapa 11	Estudio de resultados de PiCO y GraPiCO. Parte 2.
Etapa 12	Recomendaciones y resultados del estudio entre PiCO, GraPiCO y editores.

Fuente: elaboración propia (2018).

La *hipótesis* es una aseveración, inferencia, supuesto, conjetura, enunciado o afirmación creada a partir de *datos experimentales*

respecto a una o más *poblaciones* con el fin de servir como apoyo en la toma de decisiones.^{[3][5][6][9]}

La *hipótesis* es una herramienta científica utilizada para reconocer las premisas que se van a evaluar utilizando el método *científico*. Éste último es un proceso de experimentación que se utiliza en la exploración de *observaciones* (hacer y responder preguntas de carácter científico).

Los pasos generales para trabajar bajo los parámetros de este método son:

- a. Definir el problema o situación a estudiar.
- b. Hacer una investigación a fondo de los antecedentes del problema.
- c. Establecer la *hipótesis*, identificando las *variables*.
- d. Establecer los procedimientos que se llevarán a cabo en el *experimento*.
- e. Probar la *hipótesis* a través del *experimento*.
- f. Analizar los resultados.
- g. Obtener conclusiones y definir si la *hipótesis* es correcta o no.

La *hipótesis* tiene mucha relevancia dentro de un experimento, ya que establece un conjunto de circunstancias y sus consecuencias. Su definición influye en la selección de los *tratamientos* y en el éxito de la investigación.^[4]

III. Metodología

Durante el proceso de la toma de decisiones se hace necesario tener información valiosa. Esa información se puede obtener mediante un instrumento analítico llamado *prueba de hipótesis* (tam-

bién conocido como *prueba de significación* o *reglas de decisión*), que permite analizar los *tratamientos* de forma objetiva, teniendo en cuenta los riesgos a los que conlleva una conclusión errónea. ^[1]
[3][5][6][9]

Dentro de la *prueba de hipótesis* solo pueden existir dos tipos de *hipótesis*: *hipótesis nula* (H_0) e *hipótesis alternativa* (H_1).

La primera tiene relación con cualquier *hipótesis* que se desea probar y es denotada con H_0 . Cuando es declarada con respecto a un parámetro poblacional siempre se establecerá de modo que especifique un valor exacto del parámetro.

La segunda hace referencia a cualquier hipótesis que difiera de una *hipótesis* dada y es denotada con H_1 . Cuando es formulada con respecto a un parámetro poblacional siempre permitirá varios valores y su aceptación es el resultado del rechazo de la *hipótesis nula*.
[1][3][5][9]

Cabe resaltar que la *prueba de hipótesis* tiene como objetivo aceptar o refutar H_0 , para ello se debe tener en cuenta el criterio de evaluación del estudio y la posibilidad de que en los resultados ocurra lo siguiente: ^{[1][6]}

- a) Resultados cuya probabilidad de aparecer es mínima, como para dudar que tiene su origen en la población estudiada en H_0 .
- b) Resultados probabilísticamente compatibles con la H_0 , pero no son suficientes para que la H_0 sea cierta.

Los resultados descritos se pueden notar gráficamente en zonas que permitirán aceptar o rechazar la H_0 :

- a) Región o Zona de Rechazo de la H_0 .
- b) Región o Zona de Aceptación de la H_0 .

Estas regiones deben tener un peso probabilístico determinado que permita delimitarlas. En el caso de la zona de rechazo el peso

probabilístico se denomina nivel de significancia α , y este valor comúnmente debe ser pequeño: 0.05, 0.01, 0.1. ^{[3][5]}

Al efectuar una *prueba de hipótesis* es muy probable incurrir en errores debido al tamaño de la muestra utilizada en el experimento. Existen dos tipos de errores: *Error Tipo I* o *Error α* y *Error Tipo II* o *Error β* . El primero sucede en el instante en que la *hipótesis nula* se rechaza cuando es verdadera, y el segundo ocurre justo en el momento en que la *hipótesis nula*, siendo falsa, no se rechaza. Ver Tabla 9. ^{[3][5][6]}

Tabla 9. Tipos de Errores.²

Decisión	Estado	
	H ₀ Cierta	H ₁ Cierta
Rechazar H ₀	Error Tipo I	Decisión Correcta
No Rechazar H ₀	Decisión Correcta	Error Tipo II

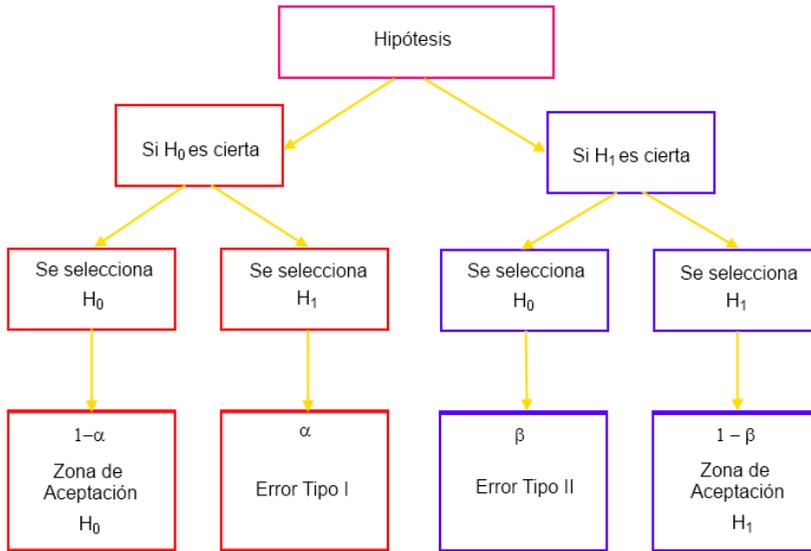
Fuente: elaboración propia (2018).

En la prueba de significación, si el diseñador del experimento selecciona una región crítica adecuada puede influir en la posible ocurrencia *del error tipo I*. Mientras que si se aumenta el tamaño de la muestra se podría reducir la probabilidad de ocurrencia de los *errores tipo II*. Ver Figura 1. ^{[3][5]}

² Tabla 2 y Figura 1, fueron realizadas basándose en la información de las paginas:

- <http://www.virtual.unal.edu.co/cursos/odontologia/2002890/lecciones/hipot2/hipo2.htm>
- http://www.virtual.unal.edu.co/cursos/sedes/manizales/4030006/lecciones/capitulotres/prueba2_1.html
- http://colposfesz.galeon.com/inferencia/teoria/conhip_archivos/image004.jpg

Figura 1. Posibles ocurrencias de errores.



Fuente: elaboración propia (2018).

Cuando se prueba la *hipótesis* establecida en el experimento se debe tener presente que cometer el *error tipo I* trae más complicaciones que cometer el *tipo II*.

La *hipótesis nula* debe formularse de forma que sea rechazada si la evidencia del *experimento* apoya esta decisión; es decir: debe proporcionarse evidencia suficiente para rechazar la *hipótesis nula* y demostrar que la *hipótesis alternativa* es la correcta. Para trabajar eficientemente, antes de seleccionar el tamaño de la muestra aleatoria, se aconseja escoger el tamaño del *error tipo I*. [3][5][6]

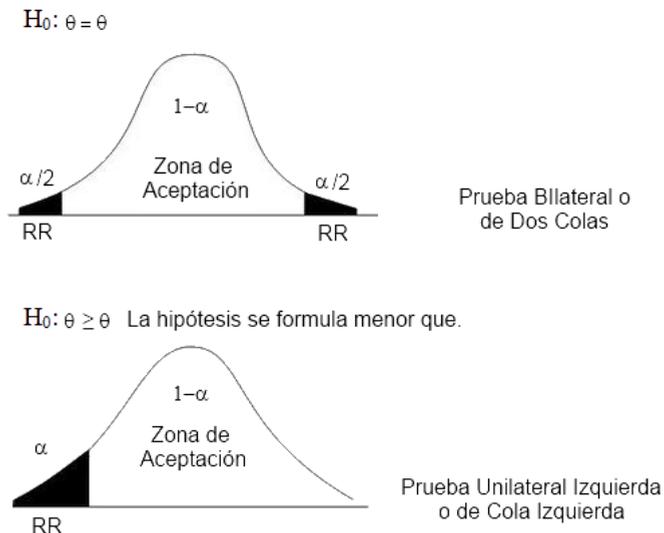
Planteándose lo anterior, puede ocurrir que la *hipótesis nula* no permite su rechazo con el valor escogido. Debe evitarse aumentar el tamaño del error con la idea de rechazar la *hipótesis nula*, porque se estaría incidiendo en una grave decisión, a causa de que no se

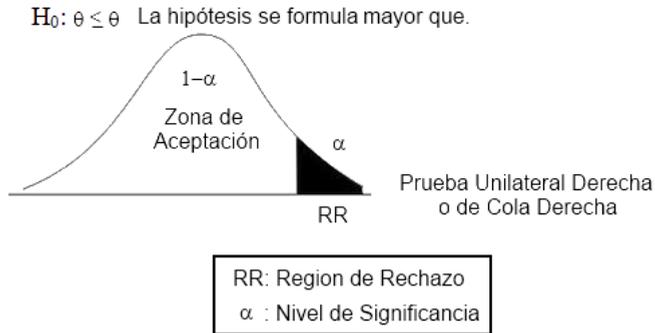
pueden manipular o modificar las condiciones en las que se ejecutó el estudio y el tamaño de la muestra.^[6] Ver Figura 2.

La *prueba de hipótesis* ocurre durante la ejecución de los siguientes pasos: ^{[1] [3] [5] [6]}

- Identificar el parámetro de la población en la que se ejecuta el *experimento*.
- Plantear las *hipótesis* (tanto *nulas* como *alternativas*).
- Establecer el Nivel de Significación (frecuentemente es denotado por α y hace referencia a la probabilidad máxima de cometer un error de *tipo I* en la *prueba de hipótesis*).
- Tomar una muestra aleatoria,
- Calcular un estadístico Z o t, y definir la región crítica (región de rechazo) del estadístico Z o t.
- Conformar la regla de decisión, interpretación de la información y formulación de conclusiones.

Figura 2. Regiones y Tipos de Prueba de Hipótesis.





Fuente: elaboración propia.

Existen dos tipos de *prueba de hipótesis*: la *prueba de hipótesis de una cola* y la *prueba de hipótesis de dos colas*.

La primera es conocida como *prueba de hipótesis unilateral* porque usa los valores extremos del estadístico Z o su correspondiente Z a un lado de la media. Su región crítica se ubica a un lado de la distribución con un área igual al nivel de significación.

La segunda es conocida también como *prueba de hipótesis bilateral*, a causa del uso de los valores extremos del estadístico Z o su correspondiente Z a ambos lados de la media. ^[5]

IV. Implementación

La verdad o falsedad de la *hipótesis* se puede conocer únicamente cuando se examine absolutamente toda la población; debido a esto, la aceptación de una *hipótesis* implica que no hay suficiente evidencia para su rechazo.

Es recomendable indicar que el rechazo de la *hipótesis nula* ha fallado, mientras que el rechazo implique su refutación por parte de la evidencia muestral existente.

La *hipótesis nula* nunca debe ser aceptada como verdadera ya que se supone verdadera hasta que haya evidencia alguna indicando lo contrario. Ver Figura 3. [9]

Figura 3. Desarrollo de la Prueba de Hipótesis.



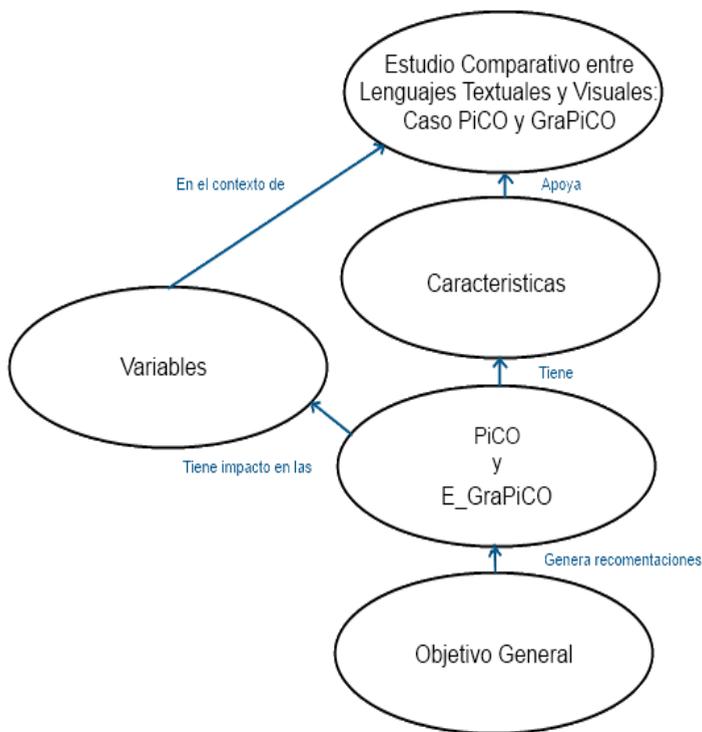
Fuente: elaboración propia (2018).

la probabilidad de rechazo de la *hipótesis nula* cuando la *hipótesis alternativa* es verdad se calcula mediante $(1 - \alpha)$. Se deben utilizar los valores α (los menores niveles de significancia para llevar la *hipótesis nula* al rechazo) para concluir la *prueba de hipótesis* de forma adecuada [5]

Teniendo en cuenta lo anterior, en la mayoría de proyectos de ingeniería se deben precisar las *hipótesis*, empleándose algunas estrategias:

- Identificar la característica que más incide directamente en la solución del problema. A través de esto se puede establecer una *hipótesis* en términos de una variable que mida el interés del objeto de estudio.
- Reconocer las variables que hacen parte del contexto del problema y crear una *hipótesis* en términos de las variables que más intervienen en el logro del objetivo del estudio. Ver Figura 4.

Figura 4. Elementos del estudio comparativo que intervienen para lograr el objetivo planteado.



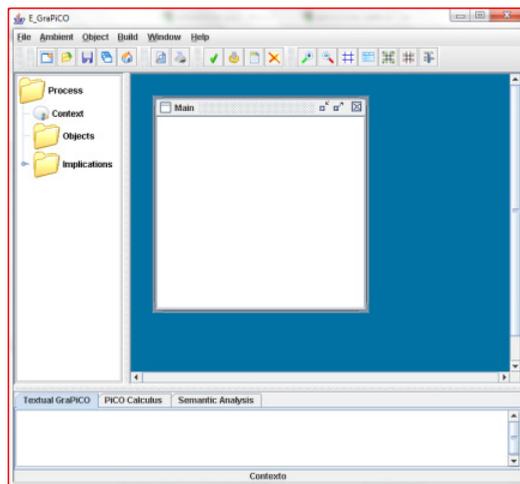
Fuente: elaboración propia (2018).

V. Planteamiento

El planteamiento del problema está enfocado en dos interrogantes, ¿En cuáles circunstancias se considera que un tipo de lenguaje se comporta mejor con respecto al otro? y ¿Qué mejoras recomienda el usuario a E_GraPiCO³ para fortalecer su utilización y su difusión?.

Se desea enfatizar en esos dos interrogantes, con el fin de soportar el conjunto de modificaciones de forma que requiere el editor E_GraPiCO (basadas en las recomendaciones del usuario final). Dichos cambios surgirán del estudio comparativo en el que se apreciarán las ventajas de la utilización de un editor visual frente a uno textual y viceversa. Además se harán recomendaciones enfocadas en el mejoramiento del editor grafico de programas existente. Ver Figura 5.

Figura 5. Aplicación E_GraPiCO

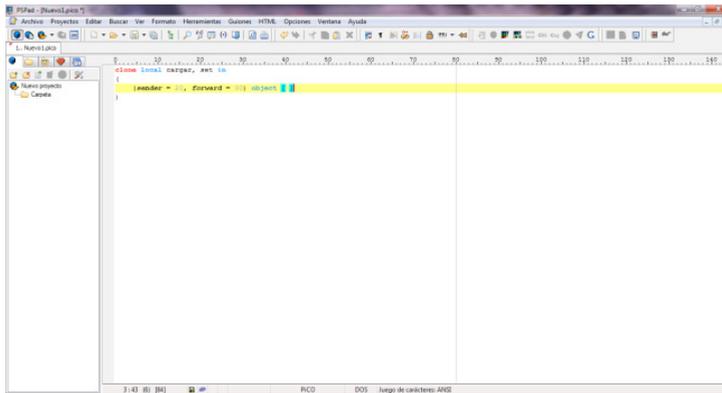


Fuente: captura de pantalla de la aplicación E_GraPiCO (2018).

³ E_GraPiCO fue creado siguiendo cada una de las especificaciones formales que surgieron de la semántica de PiCO y GraPiCO; gracias a esto fue posible la demostración de su corrección y completitud; haciendo posible que E_GraPiCO sea una aplicación estable y funcional.

Hasta el momento no existían elementos para efectuar un estudio del tipo que se propone, porque no había dos lenguajes, uno textual y otro visual, con equivalencia expresiva formalmente demostrada⁴. Después del desarrollo de PiCO y GraPiCO esta dificultad fue superada. Ver Figura 6.

Figura 6. Editor PSPad utilizado para PiCO.



Fuente: Captura de pantalla (2018).

En la actualidad se discute mucho qué tipo de lenguaje es mejor, el visual o el textual. Con la llegada de los procesadores rápidos se posibilita desplegar información muy compleja. Esto ha traído como consecuencia que sistemas operativos y lenguajes de programación visuales estén en el mercado y en la academia.

Para efectos de este *experimento* se propuso como *hipótesis nula* la siguiente *sentencia*:

Existen características en el programa aplicativo (estabilidad, diseño, modificabilidad, usabilidad) y en el lenguaje (claridad, simbología, navegación, comprensión), que influyen en los niveles de

4 La frase “equivalencia expresiva formalmente demostrada” aplicada a dos lenguajes y hace referencia a que estos generan sus múltiples *sentencias*, expresando exactamente lo mismo tanto para el uno como para el otro.

asimilación, comprensión y aceptación de los usuarios hacia los tipos de lenguaje de programación textual y visual.

Mientras que como la *hipótesis alternativa* se planteó el siguiente enunciado:

No hay razón alguna que explique la aceptación o no de los usuarios hacia los tipos de lenguaje de programación textual y visual.

Habría que tener en cuenta lo siguiente para esta investigación:

- El *error tipo I* se presentaría si se rechaza la *hipótesis nula* conociendo la existencia de cualidades o aspectos capaces de influir en los niveles de asimilación, comprensión y aceptación de los usuarios hacia los tipos de lenguaje de programación.
- El *error tipo II* se originaría justo en el momento que la *hipótesis nula* sea aceptada aunque no exista forma alguna que influya al usuario en la aceptación o rechazo de los tipos de lenguaje de programación.
- Se realizará el análisis de los datos recolectados a través de las encuestas respectivas a los dos lenguajes de programación. Requerirá del procesamiento de múltiples cruces de *variables* involucradas y el estudio posterior de las gráficas que se generen. Se manipularán algunas variables relacionadas con el nivel académico, la edad y el sexo, buscando que los resultados obtenidos se establezcan siempre en la región de aceptación y así lograr un buen término en la investigación del estudio.

VI. Conclusión

En este documento se presentó la *prueba de hipótesis* con el fin de dar soporte al *estudio comparativo* entre lenguajes textuales y lenguajes visuales: Caso PiCO y GraPiCO.

En este *estudio por observación* se planteó una *hipótesis*. Esta será desarrollada durante la investigación, con el objetivo de aceptarla o no.

La *hipótesis* tiene su importancia en todos los estudios experimentales, debido a que permite un enlace fundamental entre las teorías y las observaciones. Crea un camino en donde cualquier investigación sugiere pasos y procedimientos para llegar al conocimiento esperado. En nuestro caso el *estudio comparativo*.

Más adelante, se expondrá la temática correspondiente a los diferentes momentos del estudio:

Se trabajará en la definición de *variables* y su clasificación (herramienta útil en el diseño previo a la ejecución de un estudio).

Se seleccionarán los *sujetos de estudio* y el tratamiento que se les dará dentro de la investigación.

De igual forma, se describirá el uso dado a algunos *tratamientos* y las *observaciones* realizadas a estos.

En cuanto a la metodología pedagógica utilizada, se explicará el *Aprendizaje Basado en Proyectos*, el cual fue empleado en la realización del taller de modelamiento de los lenguajes de programación del presente estudio.

Y, por último, se presentará el análisis de los resultados obtenidos a partir de las encuestas como medio de retroalimentación.

VII. Bibliografía

- [1] A. L. Webster, Estadística aplicada a los negocios y la economía, Bogotá: Mc Graw Hill, 2001.
- [2] R. E. Walpole, Probabilidad y estadística para ingenieros, Mexico D.F.: Prentice Hall, 1998.
- [3] D. C. Montgomery, Diseño y análisis de esperimentos, Mexico D.F.: Limusa, 2004.
- [4] R. O. Kuehl, Diseño de experimentos. Principios estadísticos para el diseño y análisis de investigaciones., Thomson Learning, México, 2000.
- [5] M. R. Spiegel, Probabilidad y Estadística, Mexico D.F.: McGraw-Hill, 1977.
- [6] D. C. Montgomery y G. C. Runger, Applied Statistics and Probability for Engineers, United States of America: John Wiley & Sons, Inc., New York, 2003.
- [7] E. K. Antonsson, Development and Testing of Hypotheses in Engineering Design Research, Journal of Mechanisms, Transmissions, and Automation in Design, June, 1987, Volume 109, Issue 2, ISSN:1050-0472, eISSN: 1528-9001
- [8] C. C. George, Probabilidad y Estadística. Aplicaciones y Métodos, McGraw-Hill, México, 1999.
- [9] C. Tavera y J. Díaz, Nuevo cálculo visual: GraPiCO, En II Congreso Colombiano de Computación, Universidad Javeriana. Bogotá, 2007.
- [10] C. Tavera y J. Díaz, Breve Discusión de las Ventajas de los Lenguajes Visuales frente a los Textuales: Caso de Estudio el Cálculo GraPiCO, En III Congreso Colombiano de Computación. Medellín, 2008.

Capítulo 2

VARIABLES EN UN EXPERIMENTO DE LENGUAJES DE PROGRAMACIÓN

*Luis Eduardo Espinosa Galliady
Christian Felipe Cano Castillo
Carlos Andrés Tavera Romero*

I. Introducción

En la sección anterior se logró establecer las hipótesis que nos ayudarán a orientar hacia dónde debemos encaminar el proyecto. Falta ahora encontrar las variables del experimento, las mismas que nos clarificarán cómo hacer y cuando medir los experimentos y que datos son importantes de tener en cuenta.

II. Marco teórico

En la Etapa1 mediante la definición de la *hipótesis* se mencionó el objeto de análisis de este experimento y su proceso de verificación en futuras etapas. Esto permitirá llegar a una conclusión al finalizar el estudio.

Respecto a la presente etapa se presentarán algunos conceptos relacionados con las *variables* (la definición de las variables y su clasificación) utilizadas en esta investigación. Ver Tabla 10.

Tabla 10. Etapas del Estudio Comparativo.

Estudio Comparativo entre lenguajes Textuales y lenguajes visuales. Caso: PiCO y GraPiCO	
Etapa 1	Elaboración de hipótesis en experimentos de lenguajes de programación.
Etapa 2	Variables en un experimento de lenguajes de programación.
Etapa 3	Unidades experimentales utilizadas en pruebas de lenguajes de programación.
Etapa 4	Tratamientos y replicas en un experimento de programación.
Etapa 5	Aprendizaje Basado en Problemas (ABP) aplicado a los lenguajes de programación.
Etapa 6	La comunicación en el estudio comparativo entre lenguajes textuales y lenguajes visuales. Caso PiCO y GraPiCO.
Etapa 7	Sistematización de una experiencia de investigación entre la Comunicación Social y la Ingeniería de Software.
Etapa 8	Modelo de sistematización propuesto “TCACI en doble vía”.
Etapa 9	Pasos en la realización de los audiovisuales pedagógicos: PiCO – GraPiCO y ejercicio de modelación.
Etapa 10	Estudio de resultados de PiCO y GraPiCO. Parte 1.
Etapa 11	Estudio de resultados de PiCO y GraPiCO. Parte 2.
Etapa 12	Recomendaciones y resultados del estudio entre PiCO, GraPiCO y editores.

Fuente: elaboración propia (2018).

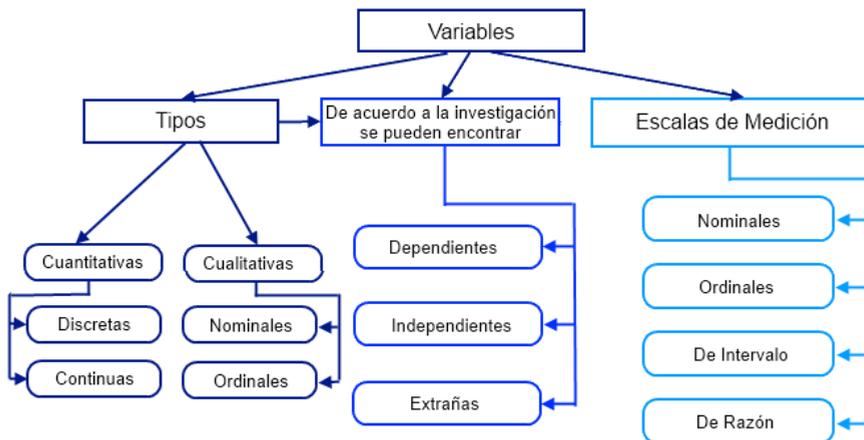
Las *variables* son características que están presentes en un objeto, individuo, fenómeno o proceso estudiado en la muestra poblacional, con la posibilidad de adoptar diferentes valores. También se

considera *variable*, si la modificación de alguna de las características anteriormente mencionadas causan el cambio en otro objeto, individuo, fenómeno o proceso.^[1]

La *variable* se mide con las *escalas de medición*. Las *escalas de medición* son un recurso así como un instrumento que establecen ciertas reglas y normas numeradas. Posibilitan la comparación del objeto que se va a medir, debido a la asignación de una identificación matemática que representa su magnitud y da un significado a esa cantidad. Ver Figura 7.

La elección de la *escala de medición* dependerá de dos cosas, el tipo de la *variable* en estudio y el manejo estadístico de la información.

Figura 7. Clasificación de las Variables.



Fuente: elaboración propia (2018).

Las *variables* se clasifican así:

a. *Variables Cualitativas*: Son atributos o categorías no numéricas, y solo pueden ser: *nominales* u *ordinales*.^[1]

- *Nominales*: Son las que se permiten clasificar pero no tienen ningún tipo de orden.

Ejemplo: sexo, nacionalidad, profesión, entre otros.

- *Ordinales*: Son las que se admiten a clasificar con un ordenamiento natural, teniendo en cuenta un orden de grado o de nivel.

Ejemplo: jerarquía del empleado, meses del año, días de la semana, entre otros.

b. Variables cuantitativas: Son los atributos o categorías que se expresan mediante cantidades numéricas. Pueden ser:^[1]

- *Discretas*: Variables a medir que sólo pueden adoptar un sólo valor numérico entero y que en muchas ocasiones presentan valores intermedios que carecen de sentido.

Ejemplo: número de exámenes, número de hijos, entre otros.
^{[1][2][3][5]}

- *Continuas*: Variables que pueden adquirir cualquier valor dentro de un intervalo de valores. En muchas ocasiones están limitadas a la precisión del dispositivo o aparato medidor.

Ejemplo: el peso, la talla, la altura, la presión arterial, entre otros.
^{[1][2][3][5]}

c. Variables independientes: son variables utilizadas por el investigador del experimento con el fin de manipular y estudiar la incidencia en las variables dependientes.^{[1][5]}

d. Variables dependientes: Son propiedades o características observadas, medidas que se tratan de cambiar, para determinar el efecto de las variables independientes en el experimento.^{[1][5]}

e. Variables extrañas: Son variables independientes no relacionadas con el objetivo del estudio o experimento, pero que puede presentar efectos sobre las variables dependientes.

Las *escalas de medición* determinan los métodos estadísticos que se usan para analizar los datos. Por ello, es importante definir las características que se van a medir. Las *escalas de medición* más comunes son:

1. *Escala nominal*: es un recurso estadístico utilizado para identificar las clases, o conjunto de elementos con características comunes que resultan de una clasificación basada en criterios como la calidad, condición, etc. No tiene propiedades cuantitativas y comúnmente los datos que se emplean son frecuencias de valores o números de incidencia en cada clase de forma tabulada. Los datos que se evalúan en este tipo de escala se llaman:
 - a. Observaciones cualitativas, debido a que describen la calidad de lo estudiado.
 - b. Observaciones categóricas, debido a que los valores se agrupan en categorías.

Este tipo de información generalmente se muestra en gráficos de barras o tablas de contingencia, porque los datos nominales o cualitativos se describen en términos de porcentajes o proporciones.

2. *Escala ordinal*: es un recurso estadístico en el cual las clases se diferencian unas de otras como en la nominal, pero además mantiene una relación entre sí, asignando un lugar específico a cada objeto de un mismo conjunto, de acuerdo con la intensidad, y la fuerza, entre otros. Al igual que en la escala nominal, se emplean porcentajes y proporciones.
3. *Escala de intervalo*: es un recurso estadístico que establece distancias equivalentes entre los objetos y la propia escala. Su utilización puede indicar con exactitud la separación entre dos puntos, garantizando que los objetos que se miden estén con la misma separación de distancia o magnitud mostrada en la escala.
4. *Escala de razón*: Es un recurso estadístico llamado también escala de proporciones. En la medición se establece la exis-

tencia de un cero natural y absoluto como origen, indicando la posibilidad de que el objeto que se analiza no tenga la propiedad que se mide. Se permite todo tipo de operaciones aritméticas y los números de la escala representan cantidades reales de lo que se esté midiendo.

III. Modelamiento

Durante la ejecución de un *experimento* se desea siempre investigar diferentes aspectos de una o más *poblaciones*.

Dichos aspectos son representados en el contexto estadístico y *matemático* mediante *variables*. Estas sirven de apoyo en la toma de decisiones de la población, motivo de análisis.

En el transcurso del *experimento*, el diseñador se puede encontrar con dos tipos de *variables*: *variable cuantitativa* y *variable cualitativa*. La primera, hace referencia a observaciones expresadas de forma numérica; y la segunda, alberga *observaciones* enunciadas de forma no numérica.

Para efectos del *estudio comparativo* entre PiCO y GraPiCO, la palabra *variable* será manejada como evento, esto dado a que las *variables* representarán los sucesos desencadenados en la interacción de los estudiantes con cada uno de estos dos lenguajes.

Aclarado lo anterior; las *variables cuantitativas* y *cualitativas* a su vez, pueden presentarse como *variables independientes* (también conocidas como explicativas o de regresión) y *dependientes*.

Durante el estudio, las *variables* que se desean explicar o predecir regularmente se presentan de forma independiente; la ocurrencia de un evento no interfiere en la probabilidad de ocurrencia de otro. Si la probabilidad de ocurrencia de un evento es cero o uno, significa que es independiente de sí mismo. Si por el contrario un evento impacta en la probabilidad de ocurrencia de otro, se dice que la *variable* o evento es *dependiente* y por lo general son las *variables* de respuesta o resultado de la ejecución del *experimento*.^{[1][2][3]}

IV. Implementación

En el estudio *comparativo* entre PiCO y GraPiCO se buscó abarcar de la mejor forma posible cada uno de los eventos generados en la experiencia del usuario con la interacción de los lenguajes de programación.

Teniendo en cuenta que las observaciones generadas serían cualitativas (ya que describen la calidad de la interacción del usuario con estos dos lenguajes), una de las mejores estrategias para mostrar y analizar su información es mediante los gráficos de barras.

Se determinó que la *escala de medición* adecuada sería la *nominal*, y se emplearon las siguientes *variables* (clasificadas como *dependientes, independientes y de bloqueo*):

Con el fin de controlar el experimento, los resultados generados y garantizar que siempre se trabaje dentro de la región de aceptación definida, se emplearon en el estudio las variables dependientes presentadas a continuación:

- Universidad (u)
- Semestre (s)
- Edad (e)
- Sexo (se)

Para tener completo control sobre la interacción del usuario con PiCO y GraPiCO, en el estudio se emplearon las siguientes variables independientes:

El nivel de conocimiento y la preferencia del usuario en los lenguajes de programación:

- Nivel de conocimiento en lenguajes de programación (nclp)
- Tipo de lenguaje de programación preferido (tlp)

En la medición del nivel de conocimiento y la frecuencia de uso del lenguaje visual GraPiCO y del lenguaje textual PiCO, se definió:

- Nivel de conocimiento en lenguajes de programación visual(nclt)
- Nivel de conocimiento en lenguajes de programación textual(nclt)
- Frecuencia de uso de aplicaciones (software) que de alguna forma empleen lenguajes de programación visual (fult)
- Frecuencia de uso de aplicaciones (software) que de alguna forma empleen lenguajes de programación textual(fult)
- Interés en lenguajes de programación visual (ilt)
- Interés en lenguajes de programación textual(ilt)

En la medición de la comprensión de los diferentes constructores de GraPiCO y PiCO, se emplearon las siguientes variables:

- Nivel de comprensión del constructor “Program”(cp).
- Nivel de comprensión del constructor “Context”(cc).
- Nivel de comprensión del constructor “Objects”(co).
- Nivel de comprensión del constructor “Methods”(cm).
- Nivel de comprensión del constructor “Ask”(ca).
- Nivel de comprensión del constructor “Tell”(ct).
- Nivel de comprensión del constructor “MsgSend”(cms).
- Nivel de comprensión del constructor “Value”(cv).
- Nivel de comprensión del constructor “Variable”(cva).
- Nivel de comprensión del constructor “Argument”(car).
- Nivel de comprensión del constructor “Sender”(cse).

- Nivel de comprensión del constructor “Forward”(cfo).
- Nivel de comprensión del constructor “Operators”(cop).
- Nivel de comprensión del constructor “Relations”(crel).
- Nivel de comprensión del constructor “Constraints”(ccon).

En las mediciones correspondientes a la retroalimentación de la experiencia del usuario con los lenguajes de programación, se estableció:

- Claridad de constructores(ccla)
- Simbología de constructores(csim)
- Navegación de constructores(cnav)

En la medición del nivel de interés del usuario respecto a los editores usados en la modelación con PiCO y GraPiCO, surgieron las siguientes variables:

- Nivel de interés de acuerdo a la estabilidad (iest)
- Nivel de interés de acuerdo al diseño (idis)
- Nivel de interés de acuerdo a la modificabilidad (imod)
- Nivel de interés de acuerdo a la usabilidad (iusa)

Para evitar posibles eventualidades que arriesgaran el proyecto se establecieron las correspondientes variables de bloqueo:

- Comentarios del público.
- Prejuicios del público.
- Correcto desarrollo de la formación académica.
- Desinterés por la temática de la presentación.
- Desinterés por la tecnología.
- Estado emocional en el momento de la presentación.
- Estado físico durante la presentación (sueño).
- Inasistencia del público por motivos de fuerza mayor, tales como, accidente o alto tráfico vehicular.

V. Conclusión

Este documento presentó las *variables* que darán el soporte al *estudio comparativo* entre lenguajes textuales y lenguajes visuales: Caso PiCO y GraPiCO.

Existen diversas formas de definir o referirse al término “*variable*”, pero todas tienen una cosa en común, utilizada y enfocada en *experimentación*.

En este *estudio por observación*, se desea estudiar o analizar los niveles de asimilación, comprensión y aceptación de los usuarios hacia los tipos de lenguaje de programación textual y visual. Dicho estudio, se llevará a cabo teniendo en cuenta las *variables* mencionadas y para su empleo se tendrán en cuenta las teorías presentadas en esta etapa.

Más adelante, se expondrá la temática correspondiente a los diferentes momentos del estudio:

Se seleccionarán los sujetos de estudio y el tratamiento que se les dará dentro de la investigación.

De igual forma, se describirá el uso dado a algunos tratamientos y las observaciones realizadas a estos.

En cuanto a la metodología pedagógica utilizada, se explicará el Aprendizaje Basado en Proyectos, el cual fue empleado en la realización del taller de modelamiento de los lenguajes de programación del presente estudio.

Y, por último, se presentará el análisis de los resultados obtenidos a partir de las encuestas como medio de retroalimentación.

VI. Bibliografía

- [1] A. L. Webster, Estadística aplicada a los negocios y la economía, Bogotá: Mc Graw Hill, 2001.
- [2] R. E. Walpole, Probabilidad y estadística para ingenieros, Mexico D.F: Prentice Hall, 1998.
- [3] C. M. Cuadras, Problemas de Probabilidades y Estadística, Barcelona: Promociones y Publicaciones Universitarias S.A., 1990.
- [4] Yañez, M; Gómez de la Vega, H; Valbuena, G., Ingeniería de Confiabilidad y Análisis Probabilístico de Riesgo. Reliability & Risk Management CA. ISBN: 980-12-12-0116-9, 2004.
- [5] D. C. Montgomery, Diseño y análisis de experimentos, Mexico D.F.: Limusa, 2004.
- [6] [C. Tavera y J. Díaz, Nuevo cálculo visual: GraPiCO, En II Congreso Colombiano de Computación, Universidad Javeriana. Bogotá, 2007.
- [7] C. Tavera y J. Díaz, Breve Discusión de las Ventajas de los Lenguajes Visuales frente a los Textuales: Caso de Estudio el Cálculo GraPiCO, En III Congreso Colombiano de Computación. Medellín, 2008.

UNIDADES EXPERIMENTALES UTILIZADAS EN PRUEBAS DE LENGUAJES DE PROGRAMACIÓN

*Luis Eduardo Espinosa Gallady
Christian Felipe Cano Castillo
Carlos Andrés Tavera Romero*

I. Introducción

Despues de definir las variables en el experimento, se requiere escoger los elementos a los que les modificarán factores (de manera controlada y planificada) para revisar su reacción y respuesta. En otras palabras necesitamos esa muestra que es necesario producir en una condición. Unidad que le aplica un único tratamiento, que puede resultar en la combinación de muchos elementos, en cada oportunidad del experimento.

Esos elementos son las unidades experimentales. En este capítulo se procederá a definir las unidades experimentales.

II. Marco Teórico

En etapas anteriores del estudio, se definió la *hipótesis*, y su forma de verificación, los tipos de *variables* usados y se explicaron las razones de su selección.

En la presente etapa, se exhibirá conceptos claves relacionados con las *unidades experimentales*, su definición y las múltiples formas de usarlas dentro del contexto de la *experimentación*.

Se podrá visualizar el funcionamiento, haciendo referencia a los tratamientos efectuados. Ver Tabla 11.

Tabla 11. Etapas del Estudio Comparativo

Estudio Comparativo entre lenguajes Textuales y lenguajes visuales. Caso: PiCO y GraPiCO	
Etapa 1	Elaboración de hipótesis en experimentos de lenguajes de programación.
Etapa 2	Variables en un experimento de lenguajes de programación.
Etapa 3	Unidades experimentales utilizadas en pruebas de lenguajes de programación.
Etapa 4	Tratamientos y replicas en un experimento de programación.
Etapa 5	Aprendizaje Basado en Problemas (ABP) aplicado a los lenguajes de programación.
Etapa 6	La comunicación en el estudio comparativo entre lenguajes textuales y lenguajes visuales. Caso PiCO y GraPiCO.
Etapa 7	Sistematización de una experiencia de investigación entre la Comunicación Social y la Ingeniería de Software.
Etapa 8	Modelo de Sistematización propuesto “TCACI en doble vía”.
Etapa 9	Pasos en la realización de los audiovisuales pedagógicos: PiCO – GraPiCO y ejercicio de modelación.
Etapa 10	Estudio de resultados de PiCO y GraPiCO. Parte 1.
Etapa 11	Estudio de resultados de PiCO y GraPiCO. Parte 2.
Etapa 12	Recomendaciones y resultados del estudio entre PiCO, GraPiCO y editores.

Fuente: elaboración propia (2018).

Con los avances tan abrumadores de la tecnología actual se observa cada vez más el empleo de ambientes visuales, tal vez por el desarrollo de chips muy avanzados tales como procesadores, memorias, entre otros.

El usuario que con mayor regularidad tiene antes la posibilidad de una elevada inmersión en ambientes multimedia intuitivos (permiten al usuario sumergirse en los diversos sistemas sin que se percate) es capaz de acrecentar capacidades como el aprendizaje, la memoria, la pedagogía y los procesos cognitivos.

En el establecimiento de las *unidades experimentales o sujetos de estudio*⁵ del proyecto, tema de esta etapa, primero se observaron las diferentes tendencias y opiniones referentes al uso de los múltiples tipos de lenguajes de programación en las personas.

Luego, se establecieron las *unidades experimentales* y los medios que se usarían en el análisis (por ejemplo, el diseño y selección de un conjunto representativo de problemas solucionables desde la ciencia de la computación y la elaboración de un cuestionario que evaluara las nociones tratadas, para obtener la mayor retroalimentación posible del usuario).

Existen diversas formas de definir y referirse al *objeto de estudio* de un *experimento*, pero todas tienen una cosa en común, dicha expresión es utilizada y enfocada hacia la *experimentación*.

Todo *experimento* en su elaboración desde el inicio debe tener claros los **objetivos de estudio** o las *unidades experimentales*. Tienen múltiples formas de ser gestionadas; se asignan los diferentes *tratamientos* de acuerdo a ciertos criterios, de tal forma que cada uno tenga la misma probabilidad de tomar o recibir algún *tratamiento*.^{[2][1]} Las *unidades* definidas serán explicadas durante el desarrollo del escrito.

Algunos de los medios utilizados para el análisis de los sujetos de estudio son la *aleatorización*, *aleatorización por bloques*, *bloques aleatorios*, *bloquización totalmente aleatoria*, *diseño totalmen-*

5 Para efectos de este escrito, tanto "Sujetos de Estudio" como "Unidades Experimentales", significan: lo que en el experimento se desea estudiar o analizar.

te aleatorizado y completa aleatoriedad. Serán brevemente explicados más adelante.

A continuación, se mostrará una inducción a conceptos claves de este contexto. ^{[2][7]}

Para comenzar, el concepto de *punto muestral* se define como cada uno de los resultados de un experimento aleatorio agrupados en un espacio muestral. ^[7]

Luego, se debe tener en cuenta que *muestra aleatoria* hace referencia a las observaciones realizadas de forma independiente y al azar, teniendo en cuenta que la probabilidad de cada observación es la misma. ^{[4][6][1]}

El *muestreo* es el proceso mediante el cual se obtienen muestras para un experimento; sin embargo en lo que concierne a este documento, el *muestreo aleatorio* se define como el procedimiento empleado para seleccionar *muestras aleatorias* en un *experimento aleatorio*.

El *muestreo aleatorio simple* es una variación del *muestreo aleatorio*, de una forma simple se escogen *observaciones* al azar para conformar una *muestra aleatoria*. ^{[6][1]}

Para comprobar la *aleatoriedad* de las *muestras* se debe utilizar una *prueba de rachas*. El término racha hace referencia a un consecutivo de símbolos asignados a cada observación del experimento.

Se define como una prueba no paramétrica de aleatoriedad a la forma de encontrar algún parámetro en las rachas del proceso de muestreo. Sí se observa una cantidad grande o pequeña de rachas en ese parámetro, se dice que está en presencia de la ausencia de aleatoriedad. ^[6]

Una *variable aleatoria* también conocida como *Variable Estocástica* o *Función Estocástica*, es denotada por una letra mayúscula y hace referencia a un número real (representado por una letra minúscula) asociado a cada punto de un *espacio muestral* de un su-

ceso aleatorio. Cada uno de los *puntos muestrales* que conforman esta función posee su propia probabilidad de ocurrencia. ^{[5][6][1]}

Variable aleatoria discreta es una *variable aleatoria* que representa datos contabilizados, debido a que se le puede contar su conjunto de resultados posibles. Este tipo de variable sólo puede tomar valores enteros, es decir, un número finito o infinito de valores “numerables o contables” con probabilidad no nula. ^{[5][6][1]}

Variable aleatoria continua es una *variable aleatoria* que representa datos medidos; puede tomar valores en una escala continua. Con frecuencia, sus posibles valores son los mismos que contienen el *espacio muestral continuo*. ^{[5][6][1]}

Las *variables aleatorias* son estadísticamente independientes, si su probabilidad conjunta es igual al producto de sus respectivas probabilidades marginales. En el caso de las *variables aleatorias discretas*, supóngase que X y Y son *variables aleatorias discretas*. Si los sucesos $X = x$, $Y = y$ son sucesos independientes para todo x , y entonces se dice que X , Y son *variables aleatorias independientes*.

Las *variables aleatorias continuas* son *variables aleatorias independientes* si los sucesos:

$$X \leq x \qquad Y \leq y$$

Son sucesos independientes para todo x , y .

Las *variables aleatorias* pueden ser normalizadas de acuerdo a la definición: sea X una *variable aleatoria* con media μ y desviación típica σ ($\sigma > 0$), entonces se puede establecer una *variable aleatoria normalizada* como:

$$X^* = \frac{X - \mu}{\sigma}$$

X^* es una cantidad adimensional o sin dimensiones que tiene media cero y varianza 1 (no tiene unidades aunque X las tenga).

Los valores de una *variable normalizada* se denominan algunas veces como referencias tipificadas (X está representada en unidades tipificadas). Son útiles para comparar diferentes distribuciones.

Los experimentos aleatorios son aquellos que tienen resultados diferentes a pesar de que las condiciones sean aproximadamente idénticas. Siempre existe la posibilidad de encontrar el error experimental a causa de la existencia de una diferencia entre las corridas u observaciones individuales (suelen existir fluctuaciones, implicando que la variable de respuesta es una variable aleatoria).

III. Teorías acerca del uso de las unidades experimentales

A continuación, se presentan algunas formas de gestionar y manipular las *unidades experimentales*. Estas teorías fueron tenidas en cuenta durante el desarrollo del estudio (la *aleatoriedad* fue muy utilizada en este estudio por las facilidades que ofrece cuando se debe controlar factores perturbadores que puedan poner en riesgo la práctica).

A. Aleatorización

La *aleatorización* es la asignación aleatoria de *tratamientos* a las *unidades experimentales*; de forma más general se define como la determinación al azar de la asignación del material experimental (orden en que se realizarán las corridas o ensayos individuales del experimento). Su importancia radica en la estimación válida de la varianza del error para los métodos de inferencia estadística, justificados en las *pruebas de hipótesis* durante el desarrollo de un *experimento*, permitiendo mitigar el riesgo de los factores perturbadores sobre el experimento.^[1]

Pasos para *aleatorizar*:

1. Primero, se debe asignar una secuencia de números a las *unidades experimentales*.^[2]

2. Luego, basta con obtener una permutación aleatoria de la secuencia de números y anotarlos en el orden de la permutación. Una permutación aleatoria se obtiene tomando una sucesión de números con dos o tres dígitos de una tabla de números aleatorios y ordenándolos de menor a mayor. Los números ordenados constituyen una permutación aleatoria. Cada número del proceso de permutación se iguala con el número asignado a la *unidad experimental* en el paso anterior. Esta permutación asegura que cada una de las asignaciones de tratamiento posible tiene la misma probabilidad de ocurrencia.^[2]
3. Por último, queda asignar una cantidad previamente determinada de números a cada *tratamiento* hasta finalizar la permutación anteriormente determinada, como por ejemplo los tres primeros números al primer tratamiento, los siguientes tres números al siguiente tratamiento y así sucesivamente.^[2]

B. **Diseño aleatorizado en bloques**

En este diseño los datos son tomados como una *muestra aleatoria* de una distribución normal. Consiste en colocar las *observaciones* en grupos homogéneos, de esta forma se clasifican por *bloques* y por *tratamientos*, teniendo siempre presente que la *aleatorización* es restringida debido a que las combinaciones de los tratamientos solo se aleatorizan dentro del *bloque*. Su principal propósito es reducir la variación dentro de los *tratamientos*.

C. **Bloques aleatorios**

Para este diseño es necesario tener un conjunto completo de *tratamientos* asignados a cada bloque. Consiste en introducir en orden aleatorio los *tratamientos* dentro de cada *bloque*. Una de sus ventajas es que permite controlar una fuente de error o variabilidad, principalmente la diferencia en bloques.^[7]

D. **“Bloquización totalmente aleatoria” con un sólo criterio de bloquización**

Teniendo en cuenta que la *bloquización* es el agrupamiento de *unidades experimentales* para controlar la variación del entorno, se establece que el diseño proporciona un medio para reducir el error experimental, ya que usa una restricción única sobre la asignación aleatoria de los tratamientos a las múltiples *unidades experimentales*, siempre teniendo en cuenta que todos los *tratamientos* deben ocurrir igual número de veces en cada bloque.^[2]

E. **Diseño totalmente aleatorizado**

En este diseño, los *tratamientos* se asignan en las diferentes *unidades experimentales* al azar, todo esto siempre teniendo en cuenta que cada unidad experimental debe tener la misma posibilidad de recibir cualquier tratamiento. Su importancia se basa en que proporciona poco control sobre la variación del entorno.^[2]

F. **Completa aleatoriedad**

Este diseño consiste en asignar cada *tratamiento* en los diferentes bloques escogidos al azar. Una de sus ventajas es que utiliza la aleatoriedad para eliminar varias fuentes de error.^[7]

IV. Implementación

Durante la ejecución del *estudio comparativo* entre lenguajes visuales y textuales se realizaron dinámicas audiovisuales para la presentación de los conceptos técnicos de cada uno de los lenguajes de programación.

Se exhibieron temas muy importantes como el *empleo de restricciones*, el uso de *constructores*, *variables*, *valores*, *métodos*, *objetos* y *procesos concurrentes*. Se realizó un taller para modelar una situación cotidiana, con el fin de ofrecer una mejor forma de presentar las principales características de estos dos lenguajes y evaluar los niveles de captación de estos lenguajes en los participantes del estudio. Ambiente con muchos aspectos por controlar como la edad, el género, el nivel de escolaridad y la disposición de

los usuarios a estudiar, dando la validez al uso de la *aleatorización* para esta investigación.

Cabe aclarar previamente que algunas variables se verán repetidas, pero esto es debido a fueron usadas en dos micro-estudios, el del lenguaje textual y el del lenguaje visual.

De forma muy general, se pretende que el usuario dé su conocimiento y preferencia por los lenguajes de programación.

Durante la ejecución del *experimento* se determinó que para el manejo de las *unidades experimentales* en este caso era mejor el uso de *tratamientos*.

Se definieron como *unidades experimentales*: Nivel de asimilación (na), Nivel de comprensión (nc) y Nivel de aceptación (nat), representadas en las siguientes *variables*:

C. **Respecto a los lenguajes de programación:**

- Nivel de conocimiento en lenguajes de programación (nclp)
- Tipo de lenguaje de programación preferido (tlp)

A continuación, se presentan las *variables* correspondientes al conocimiento, frecuencia de uso e interés del usuario respecto al lenguaje visual (GraPiCO) sujeto a estudio.

D. **Respecto a los lenguajes de programación visual:**

- Nivel de conocimiento en lenguajes de programación visual (nclt)
- Frecuencia de uso de lenguajes de programación visual (fult)
- Interés en lenguajes de programación visual (ilt)

A continuación, se pueden visualizar las *variables* correspondientes al conocimiento, frecuencia de uso e interés del usuario respecto al lenguaje textual PiCO.

E. Respecto a los lenguajes de programación textual:

- Nivel de conocimiento en lenguajes de programación textual (nclt).
- Frecuencia de uso de lenguajes de programación textual (fult).
- Interés en lenguajes de programación textual (ilt)

Las siguientes *variables* permiten realizar *observaciones* sobre el nivel de comprensión del usuario respecto a cada constructor existente en los dos lenguajes sujetos a estudio PiCO y GraPiCO:

F. Respecto a los constructores de cada uno de los dos lenguajes en cuestión:

- Comprensión de “Program” (cp)
- Comprensión de “Context” (cc)
- Comprensión de “Objects” (co)
- Comprensión de “Methods” (cm)
- Comprensión de “Ask” (ca)
- Comprensión de “Tell” (ct)
- Comprensión de “MsgSend” (cms)
- Comprensión de “Value” (cv)
- Comprensión de “Variable” (cva)
- Comprensión de “Argument” (car)
- Comprensión de “Sender” (cse)
- Comprensión de “Forward” (cfo)

- Comprensión de “Operators” (cop)
- Comprensión de “Relations” (crel)
- Comprensión de “Constraints” (ccon)

Por último, están las *variables* que permiten obtener la retroalimentación del usuario respecto a los dos editores (textual y visual):

G. Respecto a cada uno de los dos editores, textual y visual:

- Claridad (ccla)
- Simbología (csim)
- Navegación (cnav)
- Estabilidad (iest)
- Diseño (idis)
- Modificabilidad (imod)
- Usabilidad (iusa)

Se presentarán algunas tablas, con el fin de poder visualizar el manejo dado a las *unidades experimentales* (algunas asignaciones de los *tratamientos* realizados a las *variables* que representan las *unidades experimentales*).

Tabla 12. *Tratamientos Realizados para GraPiCO y PiCO.*

			Tratamientos o cruces	
			Sexo	Edad
			se	e
Unidades experimentales	Nivel de conocimiento en lenguajes de programación	nclp		
	Tipo de lenguaje de programación preferido	tlp		
	Nivel de conocimiento en lenguajes de programación (visual o textual)	nclt		
	Frecuencia de uso de lenguajes de programación (visual o textual)	fulf		
	Interés en lenguajes de programación (visual o textual)	ilt		
	Comprensión de "Program"	cp		
	Comprensión de "Context"	cc		
	Comprensión de "Objects"	co		
	Comprensión de "Methods"	cm		
	Comprensión de "Ask"	ca		
	Comprensión de "Tell"	ct		
	Comprensión de "MsgSend"	cms		
	Comprensión de "Value"	cv		
	Comprensión de "Variable"	cva		
	Comprensión de "Argument"	car		

Unidades experimentales	Comprensión de "Sender"	cse	
	Comprensión de "Forward"	cfo	
	Comprensión de "Operators"	cop	
	Comprensión de "Relations"	crel	
	Comprensión de "Constraints"	ccon	
	Claridad GraPiCo	ccla	
	Simbología GraPiCo	csim	
	Navegación GraPiCo	cnav	
	Estabilidad GraPiCo	iest	
	Diseño GraPiCo	idis	
	Modificabilidad GraPiCo	imod	
	Usabilidad GraPiCo	iusa	

Fuente: elaboración propia (2018).

Tabla 13. *Tratamientos Realizados para GraPiCO y PiCO.*

			Tratamientos o cruces	
			Semestre	Universidad
			S	U
Unidades experimentales	Nivel de conocimiento en lenguajes de programación	nclp		
	Tipo de lenguaje de programación preferido	tlp		
	Nivel de conocimiento en lenguajes de programación (visual o textual)	nclt		
	Frecuencia de uso de lenguajes de programación (visual o textual)	fult		
	Interés en lenguajes de programación (visual o textual)	ilt		
	Comprensión de "Program"	cp		
	Comprensión de "Context"	cc		
	Comprensión de "Objects"	co		
	Comprensión de "Methods"	cm		
	Comprensión de "Ask"	ca		
	Comprensión de "Tell"	ct		
	Comprensión de "Objects"	co		
	Comprensión de "Methods"	cm		
	Comprensión de "Ask"	ca		

Unidades experimentales	Comprensión de "Tell"	ct		
	Comprensión de "Objects"	co		
	Comprensión de "Methods"	cm		
	Comprensión de "Ask"	ca		
	Comprensión de "Tell"	ct		
	Comprensión de "MsgSend"	cms		
	Comprensión de "Value"	cv		
	Comprensión de "Variable"	cva		
	Comprensión de "Argument"	car		
	Comprensión de "Sender"	cse		
	Comprensión de "Forward"	cfo		
	Comprensión de "Operators"	cop		
	Comprensión de "Relations"	crel		
	Comprensión de "Constraints"	ccon		
	Claridad GraPiCo	ccla		
	Simbología GraPiCo	csim		
	Navegación GraPiCo	cnav		
Estabilidad GraPiCo	iest			
Diseño GraPiCo	idis			

Fuente: elaboración propia (2018).

Tabla 14. Otros tratamientos realizados.

		GraPiCO											
		cp	cc	co	cm	ca	ct	cms	cv	cva	car	cse	
PiCO	cp	x	x	x	x	x	x	x	x	x	x	x	x
	cc	x	x	x	x	x	x	x	x	x	x	x	x
	co	x	x	x	x	x	x	x	x	x	x	x	x
	cm	x	x	x	x	x	x	x	x	x	x	x	x
	ca	x	x	x	x	x	x	x	x	x	x	x	x
	ct	x	x	x	x	x	x	x	x	x	x	x	x
	cms	x	x	x	x	x	x	x	x	x	x	x	x
	cv	x	x	x	x	x	x	x	x	x	x	x	x
	cva	x	x	x	x	x	x	x	x	x	x	x	x
	car	x	x	x	x	x	x	x	x	x	x	x	x
	cse	x	x	x	x	x	x	x	x	x	x	x	x
	cfo	x	x	x	x	x	x	x	x	x	x	x	x
	cop	x	x	x	x	x	x	x	x	x	x	x	x
	crel	x	x	x	x	x	x	x	x	x	x	x	x
	ccon	x	x	x	x	x	x	x	x	x	x	x	x
	ccla	x	x	x	x	x	x	x	x	x	x	x	x
	csim	x	x	x	x	x	x	x	x	x	x	x	x
	cnav	x	x	x	x	x	x	x	x	x	x	x	x
	iest	x	x	x	x	x	x	x	x	x	x	x	x
	idis	x	x	x	x	x	x	x	x	x	x	x	x
imod	x	x	x	x	x	x	x	x	x	x	x	x	
iusa	x	x	x	x	x	x	x	x	x	x	x	x	
		cfo	cop	crel	ccon	ccla	csim	cnav	iest	idis	imod	iusa	
		GraPiCO											

Fuente: elaboración propia (2018).

V. Conclusión

En este documento se han presentado las *unidades* o *sujetos* experimentales que se utilizarán para el estudio comparativo entre lenguajes textuales y lenguajes visuales: Caso PiCO y GraPiCO.

Bloquear y *aleatorizar* son formas eficientes para asignar los tratamientos a las *unidades experimentales*, sin tener el problema de considerar un tratamiento en especial.

Para bloquear, se recomienda siempre que sea posible dividir las *unidades experimentales* en grupos de unidades similares. Y *aleatorizar* para prevenir los sesgos en este estudio comparativo.

Más adelante, se expondrá la temática correspondiente a los diferentes momentos del estudio:

Se describirá el uso dado a algunos tratamientos y las observaciones realizadas a estos.

En cuanto a la metodología pedagógica utilizada, se explicará el *Aprendizaje Basado en Proyectos*, el cual fue empleado en la realización del taller de modelamiento de los lenguajes de programación del presente estudio.

Y, por último, se presentará el análisis de los resultados obtenidos a partir de las encuestas como medio de retroalimentación.

VI. Bibliografía

- [1] D. C. Montgomery, Diseño y análisis de experimentos, Mexico D.F.: Limusa, 2004.
- [2] R. O. Kuehl, Diseño de experimentos. Principios estadísticos para el diseño y análisis de investigaciones, Thomson Learning, México 2000.
- [3] Yañez, M; Gómez de la Vega, H; Valbuena, G., Ingeniería de Confiabilidad y Análisis Probabilístico de Riesgo. Reliability & Risk Management CA. ISBN: 980-12-12-0116-9, 2004.
- [4] R. E. Walpole, Probabilidad y estadística para ingenieros, Mexico D.F: Prentice Hall, 1998.
- [5] C. M. Cuadras, Problemas de Probabilidades y Estadística, Barcelona: Promociones y Publicaciones Universitarias S.A., 1990.
- [6] A. L. Webster, Estadística aplicada a los negocios y la economía, Bogotá: Mc Graw Hill, 2001.
- [7] M. R. Spiegel, Probabilidad y Estadística, Mexico D.F.: McGraw-Hill, 1977.
- [8] C. C. George, Probabilidad y Estadística. Aplicaciones y Métodos, McGraw-Hill, 1999.
- [9] C. Tavera y J. Díaz, Nuevo cálculo visual: GraPiCO, En II Congreso Colombiano de Computación, Universidad Javeriana. Bogotá, 2007.
- [10] C. Tavera y J. Díaz, Breve Discusión de las Ventajas de los Lenguajes Visuales frente a los Textuales: Caso de Estudio el Cálculo GraPiCO, En III Congreso Colombiano de Computación. Medellín, 2008.

Capítulo 4

TRATAMIENTOS Y RÉPLICAS EN UN EXPERIMENTO DE PROGRAMACIÓN

*Luis Eduardo Espinosa Galliady
Christian Felipe Cano Castillo
Carlos Andrés Tavera Romero*

I. Introducción

Esta sección presenta algunas teorías acerca de la gestión de tratamientos, algunas bases acerca del desarrollo de replicas y la formulación de sujetos de estudio, temas que serán usados a lo largo del experimento.

II. Marco teórico

En previas ocasiones correspondientes a las fases de este experimento se habló de la *hipótesis* usada y la forma de verificación de la misma; los tipos de *variables* usados y las razones de uso; los *sujetos de estudio* y su utilización en el marco referente a este estudio. Esta vez se presentarán algunas teorías sobre los *tratamientos* y sus implementaciones. También se especificará la cantidad de *observaciones* o *replicas* realizadas. ^{[2][3]} Ver Tabla 15.

Tabla 15. *Etapas del estudio comparativo.*

Estudio comparativo entre lenguajes textuales y Lenguajes Visuales. Caso: PiCO y GraPiCO	
Etapa 1	Elaboración de hipótesis en experimentos de lenguajes de programación.
Etapa 2	Variables en un experimento de lenguajes de programación.
Etapa 3	Unidades experimentales utilizadas en pruebas de lenguajes de programación.
Etapa 4	Tratamientos y replicas en un experimento de programación.
Etapa 5	Aprendizaje Basado en Problemas (ABP) aplicado a los lenguajes de programación.
Etapa 6	La comunicación en el estudio comparativo entre lenguajes textuales y lenguajes visuales: Caso PiCO y GraPiCO.
Etapa 7	Sistematización de una experiencia de investigación entre la Comunicación Social y la Ingeniería de Software.
Etapa 8	Modelo de sistematización propuesto “TCACI en doble vía”.
Etapa 9	Pasos en la realización de los audiovisuales pedagógicos: PiCO – GraPiCO y ejercicio de modelación.
Etapa 10	Estudio de resultados de PiCO y GraPiCO. Parte 1.
Etapa 11	Estudio de resultados de PiCO y GraPiCO. Parte 2.
Etapa 12	Recomendaciones y resultados del estudio entre PiCO, GraPiCO y editores.

Fuente: elaboración propia (2018).

Durante la ejecución de un *experimento* o *estudio*, se deben realizar múltiples observaciones de las *unidades experimentales*. Por facilidad y viabilidad del estudio dichas observaciones son clasificadas o agrupadas por medio de los tratamientos, los cuales estable-

cen las condiciones bajo las cuales serán analizadas las *unidades experimentales*.^[2]

Los *tratamientos* son creados por el diseñador del *experimento* siguiendo los lineamientos impuestos en las *hipótesis* formuladas, permitiendo asimilar de una mejor forma los diferentes procesos involucrados en el experimento.^{[2][3]}

En la práctica, los **tratamientos** se presentan como determinados grupos de medidas, agrupando sus diferentes valores dentro de múltiples niveles de factores.^[2]

Los *tratamientos* pueden ser *cualitativos* y *cuantitativos*; los primeros cuentan con niveles que corresponden a valores de una escala de medición definida y los segundos denotan tipos de magnitudes.^[3]

Dependiendo del diseño del *experimento*, los *tratamientos* se usan solos o acompañados. Si el *experimento* es un *experimento simple* (conocido también como *experimento de un factor*), para cada uno de los grupos independientes de muestras se realiza un número determinado de *observaciones* comúnmente conocidas como *tratamientos*; una forma de representar los resultados obtenidos mediante este tipo de experimento es a través de una tabla que consta de “j” filas de tratamientos, “k” columnas de medidas y una última columna que denota la media de cada tratamiento mediante la Ecuación 1, para denotar la media de todos los tratamientos, se usa la Ecuación 2.^{[2][3][4]} Ver Tabla 16, Ecuación 1 y Ecuación 2.

Tabla 16. Formas de representar los *tratamientos* de experimentos de un factor.

	$Medida_{k=1}$	$Medida_{k=2}$
$Tratamiento_{j=1}$	$Medida_{j=1,k=1}$	$Medida_{j=1,k=2}$
$Tratamiento_{j=2}$	$Medida_{j=2,k=1}$	$Medida_{j=2,k=2}$

Fuente: elaboración propia (2018).

$$\bar{x}_j = \frac{1}{b} \sum_{K=1}^b x_{jk}$$

Ecuación 1. **Media de cada tratamiento**

$$\bar{x} = \frac{1}{ab} \sum_{j,k} x_{jk} = \frac{1}{ab} \sum_{j=1}^a \sum_{K=1}^b x_{jk}$$

Ecuación 2. **Media de todos los tratamientos**

Si por el contrario el experimento es de *dos factores*, los *tratamientos* son acompañados por *bloques*. La representación de los resultados generados en esta combinación de *tratamientos* y *bloques* se realiza en una tabla la cual consta de “j” filas de tratamientos, “k” columnas de bloques y una columna que denota la media de cada tratamiento mediante la Ecuación 3. Para calcular la media de cada bloque se utiliza la Ecuación 4 y en la media total, se usa la Ecuación 5.^[3]Ver Tabla 17, Ecuación 3, Ecuación 4.

Tabla17. *Formas de representar los tratamientos de experimentos de dos factores.*

	<i>Bloque_{k=1}</i>	<i>Bloque_{k=2}</i>
<i>Tratamiento_{j=1}</i>	<i>Bloque_{j=1,k=1}</i>	<i>Bloque_{j=1,k=2}</i>
<i>Tratamiento_{j=2}</i>	<i>Bloque_{j=2,k=1}</i>	<i>Bloque_{j=2,k=2}</i>

Fuente: elaboración propia (2018).

$$\bar{x}_j = \frac{1}{b} \sum_{K=1}^b x_{jk}$$

Ecuación 3. **Media de cada tratamiento**

$$\bar{x}_k = \frac{1}{a} \sum_{j=1}^a x_{jk}$$

Ecuación 4. **Media de cada bloque**

$$\bar{x} = \frac{1}{ab} \sum_{j,k} x_{jk}$$

Ecuación 5. **Media total**

En el *experimento* la *hipótesis* que se establece define una serie de circunstancias con sus posibles consecuencias; los *tratamientos* se generan cuando se crean dichas circunstancias que se desea estudiar.

Es importante reconocer cuales son los *tratamientos* que se evaluarán en la *hipótesis*. Si desde el principio no se establece de forma precisa, detallada y clara el objetivo de estudio y la *hipótesis*, es muy factible que ni los *tratamientos*, ni el estudio sean exitosos.^[3]

III. Modelamiento

Teniendo como punto de partida que las unidades experimentales establecidas fueron: *nivel de asimilación (na)*, *nivel de comprensión (nc)* y *nivel de aceptación (nat)*. Se plantearon los siguientes cruces de variables para los tratamientos. Ver Tabla 18.

Tabla 18. *Tratamientos realizados para GraPiCO y PiCO usando el sexo y la edad.*

		Tratamientos o cruces	
		Sexo	Edad
		se	E
Unidades experimentales	Nivel de conocimiento en lenguajes de programación	nclp	
	Tipo de lenguaje de programación preferido	tlp	
	Nivel de conocimiento en lenguajes de programación (visual o textual)	nclt	
	Frecuencia de uso de lenguajes de programación (visual o textual)	fult	
	Interés en lenguajes de programación (visual o textual)	ilt	
	Comprensión de "Program"	cp	
	Comprensión de "Context"	cc	
	Comprensión de "Objects"	co	
	Comprensión de "Methods"	cm	
	Comprensión de "Ask"	ca	
	Comprensión de "Tell"	ct	

Comprensión de "MsgSend"	cms	
Comprensión de "Value"	cv	
Comprensión de "Variable"	cva	
Comprensión de "Argument"	car	
Comprensión de "Sender"	cse	
Comprensión de "Forward"	cfo	
Comprensión de "Operators"	cop	
Comprensión de "Relations"	crel	
Comprensión de "Constraints"	ccon	
Claridad GraPiCo	ccla	
Simbología GraPiCo	csim	
Navegación GraPiCo	cnav	
Estabilidad GraPiCo	iest	
Diseño GraPiCo	idis	
Modificabilidad GraPiCo	imod	
Usabilidad GraPiCo	iusa	

Fuente: elaboración propia (2018).

En la Tabla 18 y la Tabla 19, la variable *Sexo* tiene como niveles masculino y femenino. Y para la variable *Edad*, se cuenta con los siguientes niveles:

- Menos de 18 años.
- De 18 a 20 años.
- De 20 a 22 años.
- Más de 22 años.

La variable *Universidad* cuenta con los niveles:

- Universidad de San Buenaventura - Cali.
- Universidad Pontificia Javeriana.
- Universidad Autónoma de Occidente.
- Universidad ICESI.

En la variable *Semestre* los niveles usados fueron Segundo, Tercero, Cuarto y Quinto.

En **Sexo**, **Edad**, **Semestre** y **Universidad** se realizaron cruces en los dos lenguajes de programación PiCO y GraPiCO (Agrupado de acuerdo a niveles).

Los que cuentan con *Nulo*, *Principiante*, *Intermedio*, *Avanzado* y *Experto*, en su posible selección:

Nivel de conocimiento en lenguajes de programación (nclp).

Nivel de conocimiento en lenguajes de programación (visual o textual) (nclt).

La opción *Visual* y *Textual* está en:

- Tipo de lenguaje de programación preferido (tlp)

En niveles *Nada*, *Poco*, *Intermedio*, *Suficiente* y *Mucho* está:

- Frecuencia de uso de lenguajes de programación (visual o textual) (fult).

- Interés en lenguajes de programación (visual o textual) (ilt).

Respondiendo a la necesidad de captar múltiples aspectos de los dos editores se tiene:

Para GraPiCO:

- Claridad GraPiCO (ccla).
- Simbología GraPiCO (csim).
- Navegación GraPiCO (cnav).
- Estabilidad GraPiCO (iest).
- Diseño GraPiCO (idis).
- Modificabilidad GraPiCO (imod).
- Usabilidad GraPiCO (iusa).

Para PiCO:

- Claridad PiCO(ccla).
- Simbología PiCO(csim).
- Navegación PiCO(cnav).
- Estabilidad PiCO(iest).
- Diseño PiCO(idis).
- Modificabilidad PiCO(imod).
- Usabilidad PiCO(iusa).

Otrasvariables que cuentan con *Nulo, Principiante, Intermedio, Avanzado y Experto* son:

Nivel de comprensión del usuario respecto a cada uno de los constructores:

- Comprensión de “Program” (cp)
- Comprensión de “Context” (cc)
- Comprensión de “Objects” (co)

- Comprensión de “Methods” (cm)
- Comprensión de “Ask” (ca)
- Comprensión de “Tell” (ct)
- Comprensión de “MsgSend” (cms)
- Comprensión de “Value” (cv)
- Comprensión de “Variable” (cva)
- Comprensión de “Argument” (car)
- Comprensión de “Sender” (cse)
- Comprensión de “Forward” (cfo)
- Comprensión de “Operators” (cop)
- Comprensión de “Relations” (crel)
- Comprensión de “Constraints” (ccon)

Tabla 19. *Tratamientos realizados para GraPiCO y PiCO usando el semestre y la universidad*

		Tratamientos o cruces	
		Semestre	Universidad
		S	u
Nivel de conocimiento en lenguajes de programación	nclp		
Tipo de lenguaje de programación preferido	tlp		
Nivel de conocimiento en lenguajes de programación (visual o textual)	nclt		

Frecuencia de uso de lenguajes de programación (visual o textual)	fult	
Interés en lenguajes de programación (visual o textual)	ilt	
Comprensión de "Program"	cp	
Comprensión de "Context"	cc	
Comprensión de "Objects"	co	
Comprensión de "Methods"	cm	
Comprensión de "Ask"	ca	
Comprensión de "Tell"	ct	
Comprensión de "MsgSend"	cms	
Comprensión de "Value"	cv	
Comprensión de "Variable"	cva	
Comprensión de "Argument"	car	
Comprensión de "Sender"	cse	
Comprensión de "Forward"	cfo	
Comprensión de "Operators"	cop	
Comprensión de "Relations"	crel	
Comprensión de "Constraints"	ccon	

Claridad GraPiCo	ccla		
Simbología GraPiCo	csim		
Navegación GraPiCo	cnav		
Estabilidad GraPiCo	iest		
Diseño GraPiCo	idis		
Modificabilidad Gra-PiCo	imod		
Usabilidad GraPiCo	iusa		

Fuente: elaboración propia (2018).

En otros tratamientos se seleccionaron los siguientes niveles:

- i. *Hora Noche (nh)*
- ii. *Sistema operativo Windows (sow)*
- iii. *Hardware Máquina óptima (hmo)*

En este estudio comparativo se realizaron cruces (Tabla 17) entre algunas variables usadas en PiCO y GraPiCO relacionadas con los constructores, características y editores de los lenguajes:

- *Comprensión de "Program" (cp)*
- *Comprensión de "Context" (cc)*
- *Comprensión de "Objects" (co)*
- *Comprensión de "Methods" (cm)*
- *Comprensión de "Ask" (ca)*
- *Comprensión de "Tell" (ct)*
- *Comprensión de "MsgSend" (cms)*
- *Comprensión de "Value" (cv)*
- *Comprensión de "Variable" (cva)*
- *Comprensión de "Argument" (car)*
- *Comprensión de "Sender" (cse)*
- *Comprensión de "Forward" (cfo)*

- *Comprensión de “Operators” (cop)*
- *Comprensión de “Relations” (crel)*
- *Comprensión de “Constraints” (ccon)*
- *Claridad GraPiCO (ccla)*
- *Simbología GraPiCO (csim)*
- *Navegación GraPiCO (cnav)*
- *Estabilidad GraPiCO (iest)*
- *Diseño GraPiCO (idis)*
- *Modificabilidad GraPiCO (imod)*
- *Usabilidad GraPiCO (iusa)*
- *Claridad PiCO (ccla)*
- *SimbologíaPiCO (csim)*
- *Navegación PiCO (cnav)*
- *Estabilidad PiCO (iest)*
- *Diseño PiCO (idis)*
- *Modificabilidad PiCO (imod)*
- *Usabilidad PiCO (iusa)*

Tabla 20. *Tratamientos realizados para GraPiCO y PiCO de manera comparativa.*

		GraPiCO										
		cp	cc	co	cm	ca	ct	cms	cv	cva	car	cse
PiCO	cp	X	X	X	X	X	X	X	X	X	X	X
	cc	X	X	X	X	X	X	X	X	X	X	X
	co	X	X	X	X	X	X	X	X	X	X	X
	cm	X	X	X	X	X	X	X	X	X	X	X
	ca	X	X	X	X	X	X	X	X	X	X	X
	ct	X	X	X	X	X	X	X	X	X	X	X
	cms	X	X	X	X	X	X	X	X	X	X	X
	cv	X	X	X	X	X	X	X	X	X	X	X
	cva	X	X	X	X	X	X	X	X	X	X	X
	car	X	X	X	X	X	X	X	X	X	X	X
	cse	X	X	X	X	X	X	X	X	X	X	X
	cfo	X	X	X	X	X	X	X	X	X	X	X
	cop	X	X	X	X	X	X	X	X	X	X	X
	crel	X	X	X	X	X	X	X	X	X	X	X
	ccon	X	X	X	X	X	X	X	X	X	X	X
	ccla	X	X	X	X	X	X	X	X	X	X	X
	csim	X	X	X	X	X	X	X	X	X	X	X
	cnav	X	X	X	X	X	X	X	X	X	X	X
	iest	X	X	X	X	X	X	X	X	X	X	X
	idis	X	X	X	X	X	X	X	X	X	X	X
imod	X	X	X	X	X	X	X	X	X	X	X	
iusa	X	X	X	X	X	X	X	X	X	X	X	
		cfo	cop	crel	ccon	ccla	csim	cnav	iest	idis	imod	iusa
		GraPiCO										

Fuente: elaboración propia (2018).

IV. Implementación

Con el fin de obtener información significativa para la toma de decisiones durante la ejecución de un *experimento* es necesario efectuar *réplicas* del mismo. Es decir reproducir el *experimento* básico en múltiples ocasiones.

Implica el empleo de todos los *tratamientos* independiente a las diferentes *unidades experimentales*, demostrando que bajo las condiciones planteadas, los resultados pueden ser reproducidos, de tal forma que el error experimental se ve reducido considerablemente y el impacto de un factor en el experimento es conocido.^[2]

La cantidad de *réplicas* depende del nivel de significancia y está condicionada por los costos y viabilidad implicados en el experimento. Depende directamente del tamaño de la muestra a tomarse es calculada frecuentemente teniendo en cuenta experiencias anteriores.

Por lo anterior es necesario desde el inicio tener claros los objetivos de estudio durante la ejecución de la investigación y definir cuáles son asignados a los diferentes tratamientos y tienen la misma probabilidad de tomar algún tratamiento.

En la investigación se realizaron 104 observaciones o repeticiones en todos los *tratamientos*. Se efectuaron dos sesiones (PiCO y GraPiCO), 52 encuestas para cada sesión (13 por cada universidad estudiada) a estudiantes de ingeniería de sistemas que cursan entre segundo y quinto semestre académico y tienen edades entre 16 y 23 años, de las Universidades (USB-CALI, UAO, PUJ, ICESI) de la ciudad de Cali. Se buscaba conservar el experimento dentro de la región de aceptación y controlar factores que pudieran poner en riesgo el estudio, siempre teniendo en cuenta que se va analizar u observar el nivel de asimilación, nivel de comprensión y nivel de aceptación de quienes usan los lenguajes de programación PiCO y GraPiCO.^{[2][3]}

V. Conclusión

En este documento se han presentado los *tratamientos*, la formulación de *réplicas* y los *sujetos de estudio* que darán el soporte al *estudio comparativo* entre lenguajes textuales y lenguajes visuales: Caso PiCO y GraPiCO.

Existen diversas formas de definir o referirse a los términos *tratamiento*, *réplica* y *objeto de estudio*, pero todas tienen una cosa en común, dichas expresiones son utilizadas y enfocadas hacia la *experimentación*.

En este *estudio por observación* se realizarán múltiples *repeticiones* del *experimento* básico, aplicando a los diversos *tratamientos* los diferentes *sujetos de estudio* como son la asimilación y la comprensión de los usuarios de PiCO y GraPiCO. Además se realizarán múltiples observaciones de las medidas (presentación, estabilidad, desempeño, usabilidad, confiabilidad, diseño, flexibilidad, funcionalidad y satisfacción).

Más adelante, se expondrá la temática correspondiente a los últimos momentos del estudio:

Como la metodología pedagógica utilizada, se explicará el *Aprendizaje Basado en Proyectos*, el cual fue empleado en la realización del taller de modelamiento de los lenguajes de programación del presente estudio.

Y, para finalizar, se presentará el análisis de los resultados obtenidos a partir de las encuestas como medio de retroalimentación.

VI. Bibliografía

- [1] A. L. Webster, Estadística aplicada a los negocios y la economía, Bogotá: Mc Graw Hill, 2001.
- [2] D. C. Montgomery, Diseño y análisis de experimentos, Mexico D.F.: Limusa, 2004.
- [3] R. O. Kuehl, Diseño de experimentos. Principios estadísticos para el diseño y análisis de investigaciones., Thomson Learning, 2000.
- [4] M. R. Spiegel, Probabilidad y Estadística, Mexico D.F.: McGraw-Hill, 1977.
- [5] C. Tavera y J. Díaz, Nuevo cálculo visual: GraPiCO, En II Congreso Colombiano de Computación, Universidad Javeriana. Bogotá, 2007.
- [6] C. Tavera y J. Díaz, Breve Discusión de las Ventajas de los Lenguajes Visuales frente a los Textuales: Caso de Estudio el Cálculo GraPiCO, En III Congreso Colombiano de Computación. Medellín, 2008.

APRENDIZAJE BASADO EN PROBLEMAS (ABP) APLICADO A LOS LENGUAJES DE PROGRAMACIÓN

*Luis Eduardo Espinosa Galliady
Christian Felipe Cano Castillo
Carlos Andrés Tavera Romero*

I. Introducción

En esta etapa del estudio se mostrará el punto de vista aportado por el *Aprendizaje Basado en Problemas* en el estudio comparativo, presentando diversas teorías existentes usadas en el taller de modelación realizado con cada uno de estos lenguajes de programación.

II. Marco teórico

En etapas anteriores se hizo referencia a la *hipótesis* y su forma de verificación. También se explicaron las *variables* empleadas, las *unidades experimentales* establecidas y su respectivo uso. Además, se presentaron algunos *tratamientos* empleados y la cantidad de *observaciones* realizadas. Ver Tabla 21.

Tabla 21. Etapas del estudio comparativo

Estudio comparativo entre lenguajes textuales y lenguajes visuales. Caso: PiCO y GraPiCO	
Etapa 1	Elaboración de hipótesis en experimentos de lenguajes de programación.
Etapa 2	Variables en un experimento de lenguajes de programación.
Etapa 3	Unidades experimentales utilizadas en pruebas de lenguajes de programación.
Etapa 4	Tratamientos y replicas en un experimento de programación.
Etapa 5	Aprendizaje Basado en Problemas (ABP) aplicado a los lenguajes de programación.
Etapa 6	La Comunicación en el Estudio Comparativo entre lenguajes textuales y lenguajes visuales: Caso PiCO y GraPiCO.
Etapa 7	Sistematización de una experiencia de investigación entre la Comunicación Social y la Ingeniería de Software.
Etapa 8	Modelo de sistematización propuesto “TCACI en doble vía”.
Etapa 9	Pasos en la realización de los audiovisuales pedagógicos: PiCO – GraPiCO y ejercicio de modelación.
Etapa 10	Estudio de resultados de PiCO y GraPiCO. Parte 1.
Etapa 11	Estudio de resultados de PiCO y GraPiCO. Parte 2.
Etapa 12	Recomendaciones y resultados del estudio entre PiCO, GraPiCO y editores.

Fuente: elaboración propia (2018).

Esta vez, se presentará el estudio desde el punto de vista del *Aprendizaje Basado en Problemas*, exponiendo las diferentes posturas existentes. Fue usado como apoyo en la realización de un taller para modelar una situación cotidiana con el fin de ofrecer una

manera de exhibir los conceptos técnicos de estos dos lenguajes (constructores, implicaciones y restricciones).

El taller consistió en la realización de una práctica (construir un programa que mida la captación de los lenguajes de programación) a partir de un ejemplo de la vida real representado en un corto video.

III. Introducción al Aprendizaje Basado en Problemas (ABP)

El **Aprendizaje Basado en Problemas** (ABP) también conocido como *Problem Based Learning* (PBL) se define como una corriente educativa multi-metodológica y multi-didáctica. Parte de un *problema* real solucionado por un grupo de estudiantes (propiciando la composición de conocimiento).^{[6][2][1]}

IV. Historia

En la década de los sesenta, en la Facultad de Ciencias de la Salud de la Universidad de McMaster (Hamilton, Ontario, Canadá), Howard Barrows (un educador físico y médico), Tamblyn y el Comité Curricular del programa académico (conformado por un grupo de educadores) desarrollaron originalmente un método conocido como *Proceso de Razonamiento Hipotético Deductivo*, que posteriormente sería definido como el *Aprendizaje Basado en Problemas* (ABP) o *Problem Based Learning* (PBL). Consiste en el aprendizaje producto de estudiar, analizar, comprender o solucionar un problema. Fue creado con el objetivo de aportar eficazmente en la formación de profesionales en áreas de la salud, porque ayudaba a los estudiantes componer, aplicar y reutilizar la información aprendida (apoyar en la adquisición de habilidades para la solución de problemas).

Inicialmente, dicho método consistía en obtener información, generar *hipótesis* y probarlas con base en información ganada en el proceso de aprendizaje; posteriormente Imidieo Nerici resumió el

ABP bajo el nombre de *Técnica Problémica y el Método de Solución de Problemas*.^{[6][5][3][2][1]}

El primer propósito educativo del ABP, fue hacer que los estudiantes de medicina adquirieran conocimiento referente a las ciencias básicas, debido a que se facilita para ser memorizado y consultado por ellos en su contexto profesional.^{[6][5]}

V. Marco teórico

A. *Constructivismo y Aprendizaje Activo*

El ABP se apoya en Barrel, Jerónimo Bruner, Ausbel, Cousinet, Dewey y Vigotsky, padres del *constructivismo del aprendizaje*. Al igual que Ferrière, que contribuyó a la *Escuela Activa*.^{[6][5][3][2][1]}

Dewey: “la práctica educativa ofrece los datos, la materia que forma los problemas de la indagación (...) estas prácticas educativas son también la prueba final del valor de las conclusiones de todas las investigaciones”

El *Constructivismo* permite mejorar los *métodos de aprendizaje*. Es un producto de la implementación de los procesos de observación, hallazgos, experimentación, retroalimentación basada en errores, invención, construcción y solución de problemas. Utiliza la teoría, la práctica y el pensamiento activo del alumno con el profesor (toma un rol de asesor, apoyando la gestión de la información generada y recolectada por el alumno). Se enriquece el entorno de estudio con actividades propuestas por el estudiante de acuerdo con su propia iniciativa.

El *pensamiento activo* a diferencia de otras corrientes, consiste en dar al estudiante el centro de atención en su enseñanza, apoyado en la conciencia y responsabilidad que él tiene. Otorga la suficiente libertad para que por iniciativa propia se investigue y dinamice el desarrollo del programa de estudio.^{[6][5][4][3]}

Rousseau (1971): "No hay duda de que se adquieren nociones más claras y más exactas de las cosas que cada uno aprende por sí mismo, que las que reciben con las enseñanzas de otro; y además de no acostumbrar a su razón a someterse servilmente al criterio de autoridad, se ejercita el ingenio mucho más encontrando relaciones de los fenómenos, practicando la asociación de ideas, inventado los instrumentos, que cuando se acepta todo tal como se nos da y dejamos abatir nuestro espíritu en la inactividad.... Entre tantos métodos admirables para abreviar el estudio de las ciencias, necesitaríamos que alguien nos proporcionara alguno para aprenderlas con esfuerzo"

VI. Aprendizaje colaborativo

El ABP se apoya en Dewey, Kilpatrick, Piaget, Kelson y Distlerhorst, grandes contribuidores al *Aprendizaje Colaborativo*.^{[5][3][2]}

El *Aprendizaje Colaborativo* radica en el *trabajo en proyectos*, los cuales son encarados por grupos de debate y discusión, conformados con pocos estudiantes y con un trabajo equitativo.

Los proyectos surgen a raíz de la presencia de un problema actual que sea muy atractivo, que genere el entusiasmo necesario en los estudiantes, para resolver a través de las herramientas metodológicas e instrumentales existentes y disponibles.^{[6][3][2][1]}

VII. Características

Se caracteriza en la solución de un *problema* (situación seleccionada, planteada y presentada por el tutor) en uno o varios grupos de trabajo, conformados por pocos estudiantes que interactúan de forma colaborativa, principal rol en su *proceso de aprendizaje*, promoviendo el *aprendizaje significativo, colaborativo, auto-dirigido, autorregulado, activo y cooperativo*. Se fomenta la *auto-formación, autonomía cognoscitiva y la auto-evaluación*.^{[6][5][4][3][1]}

Los estudiantes haciendo, uso de su autonomía, toman el rol principal en el *proceso de aprendizaje*. Tienen la libertad necesaria para escoger el contenido temático que aprenderán y la forma

como lo harán, siendo ellos los garantes de su propio *proceso cognitivo*.^{[6][4][3][1]}

Además, de seleccionar o construir la *situación problemática*, el tutor ejerce el rol de guía durante el proceso de solución del problema, centrando el proceso de aprendizaje en el alumno y en el fácil acceso de los recursos necesarios para ello. Tiene como principal objetivo, fomentar el desarrollo de habilidades académicas, cognitivas y humanas en el estudiante (destrezas tales como creatividad, autonomía, investigación, análisis, discernimiento y solución de problemas). Ver Figura 8.^{[5][3][2]}

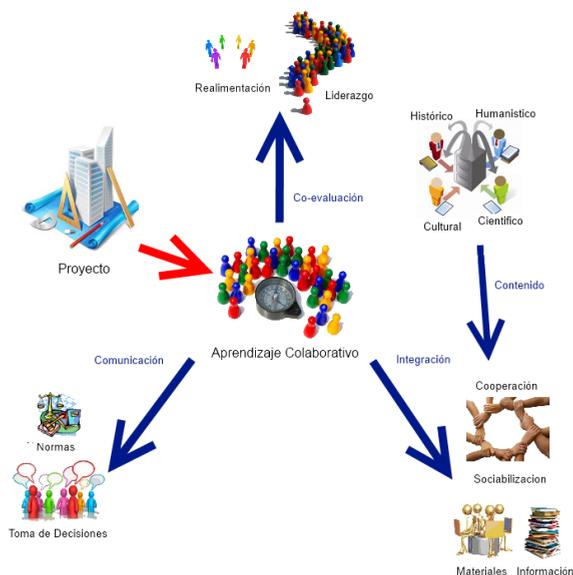
Los *problemas* deben ser situaciones lo más reales posible, ancladas con los contextos profesionales y/o sociales de los estudiantes, acordes con el contenido temático del curso y cimentadas en las diversas técnicas y estilos de enseñanza.

Los *problemas* se clasifican según su nivel de estructuración en *Problemas Brunerianos* y *Problemas No Brunerianos*. Los primeros se caracterizan por ser poco estructurados, deficientes en su redacción y definición. No poseen una única respuesta. Los segundos se caracterizan por ser muy estructurados, y mediante una serie de pasos brindan al estudiante una guía en el proceso de solución del problema.^[1]

VIII. Objetivo

Si bien el ABP se apoya en la solución de un *problema* para desarrollar el proceso de aprendizaje, cabe aclarar que la solución de este no es su principal enfoque. Su objetivo primordial es fomentar el desarrollo de destrezas y capacidades del estudiante, en búsqueda de mejorar su rendimiento en los ámbitos profesionales, habilidades meta-cognitivas, investigativas, analíticas, científicas, de solución de problemas en un trabajo en grupo.^[2]

Figura 8. Aprendizaje Colaborativo⁶



Fuente: elaboración propia (2018).

6 La Figura 8 fue inspirada en la información obtenida en la pagina: http://4.bp.blogspot.com/_3zwwR9CBpC4/S3A_NEsZwgl/AAAAAAAAAAs/7JPZm3F6VQA/s400/Imagen1.png y se utilizaron imágenes encontradas con el buscador de google, las direcciones web son :

- http://3.bp.blogspot.com/_Ji8Mmpev3xA/TIWCNWWYS2I/AAAAAAAAAEk/XgT_WdZ741A/s1600/Proyectos.jpg
- <http://sorprendemos.com/consultoresdocumentales/wp-content/uploads/2008/12/34-recursos-humanos-y-materiales.gif>
- <http://2.bp.blogspot.com/-4EWGDPY98Z8/TZUQ8cZmGkl/AAAAAAAAAEM/9Szl9dr3DAc/s1600/lideres.jpg>
- http://3.bp.blogspot.com/_BLBwanLvFgA/TBC8XO7kqwl/AAAAAAAAA-BEA/EI7yb4M7nrg/s1600/aprendizaje+apsicopce.jpg
- <http://mcimino.files.wordpress.com/2010/04/colaboracion.jpg>
- http://t2.gstatic.com/images?q=tbn:ANd9GcTWXn59YVhMPi-rjow-GaXWO8nJQx_P3cyVp3zcOzta6xWPDVw-G
- http://t2.gstatic.com/images?q=tbn:ANd9GcS79iQDWo3xRpEUsv8WM-c_e9Nww37usZ9oDleu57LGgg7dr4EH
- http://t0.gstatic.com/images?q=tbn:ANd9GcTMgr9EDLp3cXVa254ZJ9ul-WzkP_kOOj8FSgZY9f-h1E04MKwsE
- <http://t0.gstatic.com/images?q=tbn:ANd9GcQybV08f-Xul9fY85wEL-6rk->

IX. Aspectos a tener en cuenta durante la implementación del ABP

Durante la planificación de un curso ABP, se debe tener en cuenta el *diseño del problema*, la cantidad de alumnos, las características de los mismos (rutinas y técnicas de estudio, gusto por otros idiomas, situación social), la duración del curso, recursos, información necesaria y el nivel educativo. Conviene tener claros aspectos como los objetivos temáticos y humanos, fichas de apoyo, rol del profesor, herramientas de evaluación (de los estudiantes como del curso), técnicas y estilos pedagógicos (estudio independiente, grupo de discusión, experimentación, inducción-deducción, tutoría). Lo anterior define el conjunto de actividades o fases que abordarán los estudiantes en el proceso de aprendizaje.^{[6][5][4][3][2][1]}

X. Implementación

Las fases en la implementación del ABP en este estudio comparativo son:

H. *Diseño del problema*

Desde el origen del proyecto se tuvo claro que el principal objetivo era detectar de cierta forma las características que en el programa aplicativo (estabilidad, diseño, modificabilidad, usabilidad) y en el lenguaje (claridad, simbología, navegación, comprensión), influían en los niveles de asimilación, comprensión y aceptación de los usuarios hacia los tipos de lenguaje de programación textual y visual.

Teniendo en cuenta aspectos tan importantes como: la edad, el semestre de estudios universitarios y el sexo de los integrantes del público a estudiar, al igual que el tiempo para generar conocimientos previos del tema (esto debido a que los lenguajes en estudio para el proyecto son nuevos y poco conocidos), se tomó la decisión de utilizar en el taller una situación de la vida cotidiana, con el fin de aliviar la carga temática, hacer fácil la asimilación del tema y disminuir la

V1S5jbrBchr_K0wkA6O6Hq5cgrl

- <http://4.bp.blogspot.com/-0hdmqFLEx4o/TWP6NoDmW8I/AAAAAAAAA4/mhHdlZVOW30/s1600/leyes2.gif>

complejidad de la práctica.

La situación estudiada, debía ser lo más granular posible como para dividir la intensidad del ejercicio en faces idóneas; por ello, se planteó un macro problema subdividido en pequeños y sencillos casos a resolver.

El *diseño del problema*, pasó por múltiples cuestionamientos y análisis con el propósito de dar con el mejor escenario posible para que el ABP aportara sus cualidades. Todas estas ideas debían cumplir con las características de los lenguajes estudiados, tales como: el uso de *restricciones, constructores y procesos concurrentes*.

Teniendo en cuenta esto, se definió que para la sesión de ABP se haría uso de un *Problema No Bruneriano*, dividido en micro-problemas y representaría una situación cotidiana en un salón de clases.

I. **El Problema**

Se pensó en una salón de clase, lugar muy referenciado por cualquier persona en el rol de estudiante y profesor.

Para llegar al problema del salón de clase, tal y como definitivamente se resolvió, en el proyecto se debatieron múltiples alternativas, desde propuestas algo cómicas y dramáticas, hasta el típico y simple salón de clase.

Se buscaba hacer más llamativa la actividad, por ello se propiciaron factores muy importantes como el *casting* de los actores que representarían el video de representación de la situación escogida, el lugar o ambientación del salón (se buscó lo más natural posible, recreando aspectos muy importantes como útiles de clase, pupitres y tablero) y lo más necesario, el profesor y los alumnos, los cuales fueron particionados⁷ en dos conjuntos, 'egoístas' y 'generosos'. Esto se hizo con la intención de inducir al empleo de restricciones (cualidad de los lenguajes de programación sujetos de estudio).

La dinámica de la actividad consistiría en modelar "la petición del

⁷ La Figura 8 fue inspirada en la información obtenida en la pagina:
http://4.bp.blogspot.com/_3zwwR9CBpC4/S3A_NEsZwgl/AAAAAAAAAAs/7JpZm3F6VQA/s400/Imagen1.png y se utilizaron

profesor de obtener prestado un esfero de tinta roja en voz alta.”

Básicamente, basta con modelar dos alumnos (un egoísta y un generoso), un profesor y el proceso de verificación y petición de un esfero.

J. **Detalle del Problema**

Se presentan, a continuación, los diversos procesos que deben ser modelados con sus correspondientes componentes necesarios para su completo funcionamiento.

Un *proceso* correspondiente al ‘alumno egoísta’, el cual está conformado por un *objeto* prestador de esferos (sin *métodos*, con una *Restricción de Recepción de mensajes enviados* por una instancia igual al número 20 y una *Restricción de Delegación de mensajes enviados* por una instancia igual al número 30), *Mensaje* cargador de esferos (una *implicación* con sus correspondientes *Antecedente* y *Consecuente*).

El primero, cuenta con el *Envío del Mensaje* cargar y el segundo con un *Proceso Nulo* y un *Objeto Cartuchera* (una *Restricción de Recepción de mensajes enviados* por una instancia igual al número 0, no posee *Restricciones de Delegación*, un *método* cuyo nombre es “cargar”, el cual recibe los *parámetros* “Color” y “Estado”, lógica para la Creación de un *Objeto* [una *Restricción de Recepción de mensajes enviados* por una instancia igual al valor del *Parámetro* color, dos *métodos*; uno llamado “get”, con una *variable*, lógica necesaria para imponer el valor del *parámetro* “estado” a dicha *variable* y para ejecutar el *Envío del Mensaje* cargar, el cual contiene un *objeto* con los *valores* actuales de los *parámetros* color y estado; y otro llamado “set”, recibe como *parámetro* la *variable* “estado new” y tiene la lógica necesaria para hacer el *Envío del Mensaje* cargar, el cual contiene un *objeto* con los *valores* color y estado new]).

Un *proceso* correspondiente al ‘alumno generoso’, el cual se compone de un *objeto* prestador de esferos (*Restricción de Recepción de mensajes enviados* por una instancia igual al número 30 y un *método* cuyo nombre es “prestar esferos”, el cual recibe como *parámetro* la *variable* “Color” y sirve para el *Envío del Mensaje* set, el cual contiene el número 0 como *parámetro* y luego de ser ejecutado

cuenta con un *Proceso Nulo*), un *Mensaje* cargador de esferos (*Implicación*, la cual tiene sus respectivos *antecedente* y *consecuente*; el primero se utiliza para el *Envío del Mensaje* cargar y el segundo es un *Proceso Nulo*) y un *objeto* Cartuchera {*Restricción de Recepción de mensajes enviados* por una instancia igual al número 0 y el *método* cuyo nombre es “cargar”, recibe como *parámetros* las *variables* “Color” y “Estado”, se encarga de la creación de un *Objeto* [cuenta con una *Restricción de Recepción de mensajes enviados* por una instancia igual al valor del *parámetro* color y los *métodos* get (posee la *variable*, la *Imposición del Valor del Parámetro* estado en la *variablez* y el *Envío del Mensaje* cargar, el cual contiene un *objeto* con los *Valores* actuales de los *parámetros* color y estado) y set (recibe el *parámetro* estado new y el *Envío del Mensaje* cargar, el cual contiene un *objeto* con los *valores* color y estado new)].

Un *proceso* correspondiente al “Profesor”, el cual se compone de un *objeto* “Petición de Esfero” (Una *Restricción de Recepción de mensajes enviados* por una instancia igual al número 0 y un *método* cuyo nombre es “pedir esferos”, recibe como *parámetro* la *variable* “Color” y se encarga de hacer que el número 20 realice el *Envío del Mensaje* prestar esfero con el *parámetro* color y luego de ser ejecutado cuenta con un *Proceso Nulo*).

Un *proceso* correspondiente a la “Verificación y petición de Esfero”, cuya lógica radica en una *Implicación*, la que a su vez, consiste del *antecedente* (el cual cuenta con la *Petición del Mensaje* get, dicha petición es a cargo del número 0) y el *Consecuente* [*Implicación*, que tiene su respectivo *Antecedente* (el cual cuenta con la *Consulta de la Restricción* de igualdad entre el *Identificador* e y el número 1) y *Consecuente* (que posee solo la *Implicación*, que cuenta con su *Antecedente*, el cual se encarga de hacer la *petición del Mensaje* pedir Esfero, dicha petición es a cargo del número 0)].

K. **Cantidad de alumnos**

Debido a la cantidad de estudiantes asistentes al taller y teniendo en cuenta que para el ABP, es necesario manejar poca cantidad; se estableció que lo más conveniente era realizar dos sesiones contando cada una con 52 estudiantes, los cuales estarían divididos en 8 grupos de 5 integrantes y 2 grupos de 6 integrantes cada uno.

L. Características de los alumnos (rutinas y técnicas de estudio, gusto por otros idiomas, situación social)

La sesión de ABP está dirigida a estudiantes de segundo, tercero, cuarto y quinto semestre ingeniería de sistemas, quienes se encuentran actualmente cursando estudios en las Universidades de USB-CALI, PUJ, UAO, e ICESI, cuyas edades oscilen en los grupos: menos de 18 años, de 18 a 20 años, de 20 a 22 años y más de 22 años. Se hizo con el fin de controlar las *variables* durante el proyecto, propiciar un mejor ambiente a la experiencia (ya que, de esta forma, es más fácil preparar el material a usar, las fichas de apoyo, equipos y entorno de ejecución de los lenguajes de programación y audiovisuales que enriquezcan la temática tratada).

M. Duración del curso

Buscando mantener una curva estable de atención y acrecentar el nivel de éxito, se esperaría que la sesión de ABP tuviera una duración de aproximadamente 40 minutos, los cuales serían distribuidos en cuanto a la carga académica, siempre usando como principal herramienta, las intervenciones del tutor y los varios puntos de corte impuestos por los vacíos de conocimiento previo en los educandos.

N. Recursos e Información necesaria

Persiguiendo siempre el buen ambiente durante la rutina de ABP, valdría la pena tener en cuenta, recursos tales como los *estudiantes*, los *profesores*, los *computadores* (muy bien configurados para la actividad realizada por cada grupo, capaz de ofrecer al lenguaje de programación y al editor un muy buen ambiente de ejecución), dos *presentaciones audiovisuales*, una capaz de generar *conocimiento previo* en los espectadores acerca de los ítems a tratar del lenguaje de programación respectivo (el *empleo de restricciones*, el uso de *constructores*, *identificadores*, *variables* y demás). Y otra tan explícita, como para exponer de la mejor forma posible el *problema* estudiado.

O. Objetivos temáticos y humanos

La implementación del ABP en este estudio, persigue dos objetivos principales. Primero, superar algunas barreras existentes (crea-

das por factores muy arraigados en los usuarios; aspectos como prevención, creencias y expectativas previamente creadas (experiencias anteriores) frente al uso de lenguajes de programación y en la solución de problemas. Segundo, incrementar el nivel de las capacidades del estudiante en la creación de artefactos de software, como los *métodos, programas, procesos concurrentes, implicaciones, procesos nulos y objetos, manipulación de antecedentes, consecuentes y parámetros, envío de mensajes e imposición de valores a variables como de restricciones de recepción y delegación*.

Si bien es cierto que se contaba con cierta deficiencia en los estudiantes, debido a que estos no conocían por completo el tema referente a lenguajes de programación visual y textual, específicamente lenguajes PiCO y GraPiCO. Teniendo en cuenta el material utilizado y la poca complejidad del macroproblema, se esperaba que los estudiantes lograsen resolver correctamente por lo menos el 75% de los microproblemas propuestos en la actividad.

Durante la solución del macro-problema, respecto a *componentes de software*, se desea que los alumnos generen como mínimo cuatro *objetos* (dos “Prestador de Esferos”, un “Cartuchera” y un “Petición de Esfero”), siete *métodos* (un “cargar”, dos “get”, dos “set”, un “prestar esferos” y un “pedir esferos”) y diez *restricciones e implicaciones* (cinco *restricciones de recepción*, una de *delegación* y cuatro *implicaciones*).

Por el desarrollo de los puntos mencionados anteriormente, se esperaba una actividad en la cual los estudiantes se sintiesen motivados hacia la temática propuesta (ya que es enriquecida por la colaboración de la comunicación social, la cual contribuyó a la presentación de la forma más simple y atractiva de los aspectos concernientes a los lenguajes de programación y al taller de modelación).

P. *Fichas de apoyo (guía de trabajo)*

En la *solución del problema* se contó con *alumnos*, quienes previamente habrían visto una presentación introductora al problema en cuestión y los constructores del lenguaje de programación utilizados para la solución del mismo; también se tuvo en cuenta a los

profesores o tutores de la actividad, quienes servirán de apoyo a los alumnos en la solución de dudas tanto del problema como del lenguaje de programación y sus constructores (siendo estos el apoyo necesario para que en ningún momento se precipite la motivación de los alumnos y siempre haya una fuente de solución de dudas).

Teniendo en cuenta los conocimientos previamente generados en los *alumnos*, a través de la presentación del lenguaje de programación se solicitará a los mismos la *hipótesis* y el conjunto de pasos lógicos a seguir para llegar a la *solución del problema* en cuestión.

Q. **Rol de profesor**

El o los *profesores* intervendrán en la actividad de forma activa para referirse en concreto a la creación de cada uno de los subproblemas (los cuales servirán como punto de corte entre cada una de las fases de la actividad; cada fase constará de aproximadamente 10 minutos) y de forma inactiva para la solución de dudas acerca del problema que se esté actualmente resolviendo o también para la aclaración de incertidumbres acerca del lenguaje de programación. Las intervenciones inactivas se realizarán cuando el alumno lo solicite y las activas serán cuando el tiempo para cada fase se haya terminado y se avance a la siguiente.

R. **Herramientas de evaluación tanto de los estudiantes como del curso**

Para la evaluación se tendrán como herramientas, un taller, el cual servirá para evaluar el desempeño del grupo de *estudiantes* respecto a la temática propuesta, esto se hará teniendo en cuenta que el problema ha sido resuelto cuando el 75% de los *componentes de software*, relacionados con el mismo, hayan sido generados; una *encuesta* para obtener la retroalimentación de los estudiantes, lo que permitirá evaluar el proceso de aprendizaje y obviamente el diálogo constante entre las partes, permitiendo mejorar el *proceso cognitivo* de forma continua.

XI. Conclusión

La estrategia ABP, da la posibilidad de desarrollar problemáticas en forma contextualizada, pertinente y útil. Igualmente permite en grado satisfactorio, potenciar la ejemplificación, la *participación activa* y *retroalimentación* oportuna entre los participantes del *experimento*.

El uso de la estrategia ABP, en el estudio comparativo entre lenguajes textuales y lenguajes visuales: caso PiCO y GraPiCO, apoya la obtención de resultados, por la pertinencia de los temas, problemáticas, y aprendizaje en contexto, esenciales para la comprensión de teorías, fundamentación de opiniones propias, argumentación y análisis de la información.

Más adelante, se expondrá la temática correspondiente a los últimos momentos del estudio: se presentará el análisis de los resultados obtenidos, a partir de las encuestas como medio de retroalimentación.

XII. Bibliografía

- [1] B. R. Gómez, «Aprendizaje Basado en Problemas (ABP): Una Innovación Didáctica para la Enseñanza Universitaria,» vol. 8, 2005.
- [2] P. M. Bueno y V. L. Fitzgerald, «Aprendizaje Basado En Problemas Problem – Based Learning vol. 13.
- [3] S. F. Campos, El Aprendizaje Basado en Problemas como Propuesta Educativa para las Disciplinas Económicas y Sociales Apoyadas en El B-Learning, vol. 2, nº 40, 2006.
- [4] V. H. Dueñas, El aprendizaje basado en problemas como enfoque pedagógico en la educación en salud, vol. 32, nº 004, 2001.
- [5] R. Delisle, How use Problem-Based Learning In The Classroom, Alexandria, Virginia USA: Association for Supervision and Curriculum Development, 1997.
- [6] A. Escribano y Á. d. V. del Valle, El aprendizaje Basado en Problemas (ABP) Una propuesta Metodológica en Educación Superior, Madrid: Narcea S.A. de ediciones, 2008.
- [7] C. Tavera y J. Díaz, Nuevo cálculo visual: GraPiCO, En II Congreso Colombiano de Computación, Universidad Javeriana. Bogotá, 2007.
- [8] C. Tavera y J. Díaz, Breve Discusión de las Ventajas de los Lenguajes Visuales frente a los Textuales: Caso de Estudio el Cálculo GraPiCO, En III Congreso Colombiano de Computación. Medellín.

LA COMUNICACIÓN EN EL ESTUDIO COMPARATIVO ENTRE LENGUAJES TEXTUALES Y LENGUAJES VISUALES. CASO: PiCO y GraPiCO

*Yenny Viviana Cruz Pérez
Carlos Andrés Tavera Romero*

I. Resumen

Este es un proyecto interdisciplinario entre la Ingeniería de Software y la Comunicación Social. Es un trabajo interinstitucional entre la Universidad de San Buenaventura Cali –USBC– y la Universidad Autónoma de Occidente– UAO–, que se gestó para dar apoyo al Laboratorio de Investigación para el Desarrollo de la Ingeniería de Software (LIDIS), de la USBC, en el trabajo posdoctoral de su entonces director, el ingeniero de sistemas Carlos Andrés Tavera Romero, consistente en el estudio comparativo entre el Cálculo textual: PiCO y el Cálculo visual: GraPiCO^{[1][2]}, los cuales hacen parte del programa E_GraPiCO, para conocer bajo qué condiciones es más adecuado un lenguaje que otro y qué mejoras requieren.

El objetivo desde la comunicación fue socializar los conceptos técnicos básicos propios de estos dos lenguajes de programación por medio de material comunicativo audiovisual que, a su vez, sirviera como una herramienta didáctica mediadora entre los exponentes del proyecto y el auditorio.

II. Introducción

El Estudio Comparativo entre Lenguajes Textuales y Lenguajes Visuales. Caso: PiCO y GraPiCO, surgió en el año 2010 como parte de un trabajo posdoctoral desarrollado en la Universidad de San Buenaventura-Cali, más exactamente en el Laboratorio de la Investigación para el Desarrollo de la Ingeniería de Software LIDIS, liderado por su director, Ingeniero de Sistemas Carlos Andrés Tavera Romero PhD.

Los propósitos principales de esta investigación fueron: saber qué percepción tienen los usuarios sobre estos dos tipos de lenguaje y cuáles son las ventajas al utilizar un cálculo frente el otro. Como valor agregado, también se buscó conocer cuáles son las posibles mejoras de forma para el editor de programas gráficos E_GraPiCO, el cual es una aplicación que comprende el lenguaje de programación: visual GraPiCO.

En la consecución de estos objetivos, además de la Ingeniería de Software, se vincularon componentes de comunicación y estadística, gracias a la participación de la Facultad de Comunicación Social y la Facultad de Ciencias Básicas de la Universidad Autónoma de Occidente de la ciudad de Cali.

Durante el proceso de esta investigación se trabajó una serie de pasos que articulan las diferentes etapas del proyecto. A continuación se especifica la etapa a desarrollar en el presente capítulo. Ver Tabla 22.

Tabla 22. Etapas del estudio comparativo

Estudio comparativo entre lenguajes textuales y lenguajes visuales. Caso: PiCO y GraPiCO	
Etapa 1	Elaboración de hipótesis en experimentos de lenguajes de programación.
Etapa 2	Variables en un experimento de lenguajes de programación.
Etapa 3	Unidades Experimentales utilizadas en pruebas de lenguajes de programación.
Etapa 4	Tratamientos y replicas en un experimento de programación.
Etapa 5	Aprendizaje Basado en Problemas (ABP) aplicado a los lenguajes de programación.
Etapa 6	La comunicación en el estudio comparativo entre lenguajes textuales y lenguajes visuales: Caso PiCO y GraPiCO.
Etapa 7	Sistematización de una experiencia de investigación entre la Comunicación Social y la Ingeniería de Software.
Etapa 8	Modelo de sistematización propuesto “TCACI en doble vía.”
Etapa 9	Pasos en la realización de los audiovisuales pedagógicos: PiCO – GraPiCO y ejercicio de modelación.
Etapa 10	Estudio de resultados de PiCO y GraPiCO. Parte 1.
Etapa 11	Estudio de resultados de PiCO y GraPiCO. Parte 2.
Etapa 12	Recomendaciones y resultados del estudio entre PiCO, GraPiCO y editores.

Fuente: elaboración propia (2018).

III. Cómo llegan a vincularse otros saberes dentro del Estudio Comparativo entre Lenguajes Textuales y Lenguajes Visuales. Caso: PiCO y GraPiCO

En la búsqueda de los objetivos propuestos se midieron los niveles de asimilación, aprehensión y aceptación de estos dos lenguajes de programación por parte de los usuarios finales, en términos estadísticos: la población, que fue escogida de acuerdo a ciertos parámetros específicos que serán explicados más adelante. Para estos fines, se vio la necesidad de incorporar tanto un componente comunicativo como un componente estadístico, que permitieran un acercamiento con los usuarios, a fin de conocer su percepción frente a los dos cálculos objeto de estudio, y adicionalmente, saber cuáles podrían ser las mejoras requeridas para la aplicación E_GraPiCO.

A partir de la idea inicial de realizar este estudio comparativo apareció entonces la inquietud de parte de los investigadores en Ingeniería de Software de la Universidad de San Buenaventura-Cali, de vincular otros saberes que pudieran aportar desde su campo para la consecución de los objetivos propuestos.

De esta forma, y pensando en la importancia de diseñar e implementar una estrategia comunicativa que pudiera facilitar el acercamiento a los posibles usuarios finales, se dio un acercamiento con la Facultad de Comunicación Social de la Universidad Autónoma de Occidente (Cali), la cual ofreció tanto el talento humano como toda la infraestructura necesaria para hacer un trabajo conjunto.

Adicionalmente, también se hizo necesaria la vinculación de una disciplina que tuviera la experticia en la realización de estudios comparativos, y fue así como se incluyó el componente estadístico, con la participación de un investigador de la Facultad de Ciencias Básicas de la misma Universidad Autónoma de Occidente.

IV. La Comunicación en el Estudio Comparativo entre Lenguajes Textuales y Lenguajes Visuales. Caso: PiCO y GraPiCO

No fue fácil pensar un proyecto en el que *tuvieran* un papel preponderante en determinadas etapas del proceso, una disciplina de ingeniería como la Ingeniería de Software^[3] y la Comunicación, que carga el estigma de no tener una definición clara de lo que es, pues los teóricos más vanguardistas ya la asumen como una disciplina^[4], y los más tradicionalistas siguen defendiendo que es un campo del saber en construcción^[5].

Es importante dejar claro que la comunicación hace parte del conjunto de las ciencias sociales, que –como paréntesis–, según Pierre Bourdieu, han presentado un desarrollo tardío propiciado gracias a una batalla interna por la autoridad científica a la hora de “producir”^[6], es decir: hay una lucha de poderes y protagonismo que indiscutiblemente también salpica al campo en cuestión y que, además, por obvias razones también está presente en cualquier tipo de investigación donde converjan ciencias tan disímiles.

Siguiendo con la propuesta, fue necesario asumir una postura para definir lo que iba a ser la Comunicación dentro de esta investigación, y con este propósito, entraron los planteamientos de Miguel de Moragas, quien con argumentos muy sólidos, elimina cualquier discusión sobre si es una disciplina o un campo de estudio: “Podemos entender pues a la comunicación como una disciplina y también como un campo de estudio: como disciplina porque investiga los paradigmas de la comunicación en sus múltiples facetas y modelos; interpreta las relaciones, las medicaciones o las interacciones que posibilita la comunicación; busca los aspectos comunicativos comunes entre los fenómenos humanos y sociales, y de estos con los fenómenos naturales y tecnológicos. Pero la comunicación es también un ‘campo de estudio’ que se beneficia de las aportaciones a la comprensión de la comunicación desde otras disciplinas, como la antropología, la semiología, la sociología, la economía o la historia.”^[4]. Por tanto, dentro de esta investigación no existe la dua-

lidad de si es un campo de estudio o una disciplina, y como consecuencia, se concibe de igual a igual con la Ingeniería de Software.

Desde esta perspectiva, uno de los ejercicios iniciales y más complejos fue el de generar conciencia entre los ingenieros de la importancia de la comunicación dentro de su proceso investigativo, ya que desde el mismo campo hay enormes vacíos que aún en la academia están sin resolver y que es fundamental tener presentes a la hora de justificar por qué la comunicación está inmersa en todo tipo de interacción humana, incluyendo los trabajos relacionados con diferentes disciplinas del saber, aunque no hagan parte de las ciencias sociales o humanas.

En pleno siglo XXI la investigación en comunicación sigue presentando deficiencias propias de épocas anteriores, que aún hoy, los intelectuales y estudiosos del campo no han podido superar. Es así, como la investigación en comunicación se encuentra en una permanente crisis y los mismos investigadores no saben –no sabemos– qué rumbo tomar porque el hecho de ser un área tan amplia no facilita la conceptualización ni la teorización de la misma y, hasta el momento, no hay una noción “ideal” que concentre los fenómenos sociales en toda su magnitud y complejidad.

Así, que fue indispensable para los comunicadores, empezar por comprender ellos mismos, que el hecho de pertenecer a un campo en construcción, no le quitaba trascendencia ni rigor a los procesos investigativos en los cuales pudieran participar y que si bien se hace parte de las ciencias sociales, también se está haciendo ciencia, sólo que desde una mirada diferente y se estaba en las mismas condiciones competitivas que los ingenieros de software para participar en el estudio comparativo, aunque que con un enfoque comunicativo y no tan tecnológico.

Para argumentar por qué la comunicación es tan necesaria dentro de cualquier proceso de interacción humana, se pueden recordar los postulados de Gastón Bachelard, quien planteó que hay una marcada diferencia con la concepción de la comunicación como parte instintiva –animal– del ser humano^[7]; pues, por el contrario, aquí el hombre produce estructura y transforma el lenguaje, lo que da pie a la aparición del idioma; pero no de forma connatural sino en

calidad de una condición que le posibilita desarrollar un sistema de comunicación complejo donde se intercambia información.

Complementando lo anterior, Ferdinand de Saussure manifestaba que la comunicación es mucho más que sólo el lenguaje oral, pues el proceso en sí se vale del organismo casi en su totalidad, y también es mucho más que la mera transmisión de información, porque implica planear, ordenar, tener un principio de interés, de disposición, de intercambio y de compartir dicha información, y fue esto último lo que precisamente evidenció la necesidad de vincular al estudio comparativo investigadores con una formación en el campo específico de la comunicación.

Esta sinergia permitió que los ingenieros tuvieran una interacción más efectiva con el público participante en los ejercicios experimentales, y con esa finalidad fue que se planteó y diseñó de forma conjunta, entre comunicadores e ingenieros, una estrategia comunicativa que fuera más allá de la transmisión de mensajes y sus alcances llegaran hasta la mediación entre los investigadores y los usuarios finales.

V. El concepto organizacional “Estrategia de Comunicación”, aplicado al estudio comparativo entre lenguajes textuales y lenguajes visuales. Caso: PiCO y GraPiCO

De acuerdo con las teorías de la comunicación la noción “estrategia de comunicación” es un término aplicado a las organizaciones, grupos sociales (comunidades) o instituciones. Para el presente proyecto se concibió a los participantes del estudio comparativo como una comunidad, siguiendo la definición expuesta por Amalio Blanco: “una pluralidad de individuos que se congregan en torno a una estructura normativa, valorativa e incluso comparten algunos modelos de conducta formando una serie de grupos más o menos extensos, caracterizados por un estilo y una frecuencia de relación de la que se carece en otras formas de asociación a la que nor-

malmente subyace algún tipo de proximidad o de semejanza por razones biológicas o sociales”^[8].

Desde la perspectiva de la estadística, la población seleccionada aleatoriamente para el Estudio Comparativo entre Lenguajes visuales y Lenguajes textuales fueron: *104 estudiantes de 2º, 3º, 4º y 5º semestre de Ingeniería de Sistemas de la Universidad de San Buenaventura, Pontificia Universidad Javeriana, Universidad Autónoma de Occidente e ICESI.*

Bajo esta misma línea conceptual y en el marco de esta investigación, hablar del diseño y aplicación de una estrategia comunicativa es pertinente, ya que ésta última se puede entender como “una serie de acciones, programadas y planificadas, que se implementan a partir de ciertos intereses y necesidades, en un espacio de interacción humana, en una gran variedad de tiempos”^[9].

Asimismo, toda estrategia comunicativa comprende dos funciones, una referida a informar y la otra a comunicar, pero donde prevalece “la intención de compartir o poner en común una misma visión”, que finalmente es el objetivo primordial de la comunicación.

El problema de los investigadores en Comunicación Social se dirigió a identificar cuál sería la estrategia comunicativa más efectiva para la socialización de conceptos técnicos propios de la Ingeniería de Software, más específicamente de los lenguajes de programación textual PiCO y visual GraPiCO, tales como constructores, implicaciones y restricciones, entre otros; además de presentar de forma dinámica y atractiva, un problema hipotético que pudiera ser modelado en un taller práctico donde se aplicaran los conceptos aprehendidos, siendo éstas las principales necesidades comunicativas que se presentaron por parte de los investigadores en Ingeniería de Software.

Para ello, se tuvo en cuenta el tipo de comunicación que sería más efectiva de acuerdo al sistema de comunicación a aplicar; es decir: la elaboración de la información (producción), la manera de ponerla a circular (distribución) y la accesibilidad para quienes la deseen o requieran (consumo). En consecuencia, y siguiendo lo que al respecto expone Martín Serrano, se analizaron las cuatro

diferentes modalidades de comunicación: por asamblea, por emisarios, red de distribución y la tecnológica^[10]; considerando la última como aquella que podría ajustarse más a lo requerido, por trabajar con una serie de mediaciones cognitivas y estructurales, y estar dirigida a públicos amplios. Las otras tres no cumplían con las exigencias del proyecto.

Fue aquí cuando se vio la necesidad de que el equipo de investigadores, tanto de ingeniería como de comunicación, trabajaran conjuntamente en la primera etapa del sistema comunicativo: la de producción, es decir, que todos juntos construyeran la información que sería socializada. Se dedicaron dos meses a la selección de los conceptos técnicos y a la elaboración de definiciones claras, concretas y sencillas, que facilitarían su comprensión por parte de la población participante en el estudio. Todo ello, pensando en que el discurso a distribuir sería la materia prima de la investigación y que los mensajes tenían que ser pertinentes y apropiados para ser asimilados, aprehendidos y aceptados^[9].

Como parte de la estrategia también se contempló la planeación de los tiempos y las etapas de circulación de la información. Así, se determinó que el medio de comunicación más adecuado sería el audiovisual y que se elaborarían tres productos de este tipo para apoyar el estudio comparativo. Dado que no se encontraron antecedentes exactos que dieran una idea de cómo debía procederse en este caso, se trabajó con el método de aprendizaje de ensayo y error en el cual se aprende sobre la base de tanteos y equívocos hasta establecer una conexión adecuada entre la situación y la respuesta considerada como correcta.

En consecuencia, hubo un gran número de transformaciones en la estructura de los contenidos, pensados más desde su forma que desde su fondo, y propiciados por la necesidad de adaptar los mensajes a la narrativa audiovisual requerida por cada una de las posibilidades que se aplicaron antes de llegar a los que serían los productos definitivos.

Para hacer más viable que los sujetos participantes del estudio estuvieran atentos, se tuvieron presentes las leyes de la atención propuestas desde la Psicología de la Educación:

1. La intensidad es la capacidad de concentrarse en un objeto y se determina por la fuerza del estímulo a atender y por el atractivo de lo que se presenta, además de los motivos externos y ajenos a la voluntad del sujeto.
2. La duración es su grado de sostenibilidad, la atención frente a un objeto o una idea, en condiciones normales es fluctuante y breve, ya que el resto del tiempo lo que el individuo realmente hace es fijarse en otros aspectos relacionados con su eje de concentración.
3. Y el alcance es el número de objetos o ideas que pueden lograr la fijación mental, cuando algo es muy intenso se constituye como unidad porque opaca lo demás. Aún no se ha podido precisar si una persona puede fijar su atención en varios elementos al tiempo^[11].

En concordancia con lo anterior, el tiempo de distribución de la información a través de los videos se pensó con una extensión de máximo cinco minutos, para hacer tener mayores opciones de lograr la atención de la población y, por ende, motivar su participación en todo el proceso.

De igual manera, inicialmente se pensó en trabajar los audiovisuales con “voz en off”, es decir, que sólo se escuchara narración, pero no se viera la imagen de quien hablaba. Sin embargo, con el fin de llamar la atención de la audiencia, al final se decidió optar por una presentadora. Para ello se hizo un casting y se escogió a Laura Pérez Beracasa, una estudiante de Comunicación Social de la Universidad Autónoma de Occidente, quien trabaja como conductora del programa juvenil *Login* del canal regional Telepacífico. La idea era llegarle al público con una imagen conocida y fresca, que generara impacto y credibilidad. Esto último fue garantizado después de hacer un experimento piloto con un grupo de estudiantes que evaluaron muy bien la participación de la presentadora seleccionada. Su experiencia le permitió manejar la información con la seguridad y el convencimiento que podría transmitir cualquier ingeniero de software conocedor del tema.

En este punto, los comunicadores e ingenieros trabajaron más unidos que siempre, pues en la elaboración del guión técnico y el libreto era indispensable la participación de ambos equipos de investigadores para que los términos que se emplearan en los audiovisuales fueran descritos correctamente y no generaran confusión entre los usuarios finales. El propósito también era que se manejara un lenguaje, que si bien fuera técnico, pudiese ser comprendido por estudiantes de ingeniería de sistemas de segundo a quinto semestre, o sea, por el público seleccionado para participar en el ejercicio experimental.

Después de la etapa de reproducción de los primeros dos audiovisuales que serían presentados al mismo público, pero dividido en dos grupos: los que verían exclusivamente el video de PiCO y los que verían el de GraPiCO, se realizó la producción: la grabación y montaje de los productos visuales. Posterior a esto, se pasó a la etapa de postproducción, donde se editó el material y se hicieron las correcciones sugeridas por los ingenieros de sistemas y por el experto en pedagogía. Después, se elaboró el tercer audiovisual, que fue el ejemplo hipotético, el cual sería presentado a los dos grupos de estudiantes por igual, para que modelaran el ejercicio a partir de los conceptos explicados en ambos videos.

Al ser éste un trabajo basado en el método de ensayo y error, hubo que hacer varias correcciones que surgieron de los comentarios hechos por los estudiantes participantes en la prueba piloto, y por el especialista en pedagogía.

La estrategia comunicativa contó también con una etapa de evaluación, basada en una corta serie de preguntas, inmersas en la encuesta general que fue aplicada al terminar el taller de modelación.

Se resalta que el formato general de dicha encuesta fue elaborado conjuntamente entre los ingenieros, los comunicadores y el experto en estadística, a fin de ser consecuentes con el lenguaje manejado en los audiovisuales, pero sin perder la esencia del estudio comparativo. También se brindó a los investigadores en ingeniería una orientación dirigida hacia el fortalecimiento de la expresión corporal, la expresión oral y el manejo de público, pensada en las

jornadas de aplicación de los ejercicios experimentales entre los estudiantes seleccionados como muestra.

Finalmente, la estrategia comprendió la socialización de la investigación por medio del presente informe escrito que, a manera de libro, describe todo el proceso ejecutado. En esta etapa de producción intelectual, los comunicadores participaron redactando sus respectivos capítulos y colaborando con la edición de estilo de todo el texto.

VI. Cómo se concibe el enfoque didáctico en el Estudio Comparativo entre Lenguajes Textuales y Lenguajes Visuales. Caso: PiCO y GraPiCO

Es necesario enfatizar que esta investigación tuvo un marcado enfoque didáctico, por tratar un problema básicamente de carácter académico y cuyo público fue un grupo de estudiantes universitarios.

Con esta base, se consultaron diferentes expertos en pedagogía y en comunicación, quienes después de revisar los objetivos del proyecto y el tipo de información a comunicar (los conceptos técnicos y el problema hipotético), reafirmaron que la estrategia debía estar mediada por la tecnología; en otras palabras, que se hicieran uno o varios productos comunicativos para presentar los contenidos de la manera más clara, precisa, dinámica y atractiva que fuera posible, buscando garantizar el interés y de los estudiantes de ingeniería de sistemas seleccionados dentro de la muestra.

Aprender algo requiere la máxima atención para asimilar todos los procesos que implica procesar la nueva información y sólo podremos recordar aquello a lo que hayamos prestado suficiente atención.^[12] Por tanto, los tres productos audiovisuales y el taller mismo, tendrían que ser pensados de manera estratégica para que los mensajes logran ser percibidos por los sujetos; es decir, que pudieran interpretarlos y darles significado, todo esto enmarcado en el aprendizaje.

En la implementación de la estrategia comunicativa, y con el fin de socializar los mensajes, se adoptó el *aprendizaje cognoscitivo*, pues era necesario que la audiencia de los productos audiovisuales (quienes fueron los mismos usuarios finales) involucrara “la memoria, el pensamiento, la resolución de problemas y el lenguaje”^[13] dentro de todo el ejercicio experimental, elementos básicos de cualquier proceso mental que vaya más allá del condicionamiento. Estos últimos ayudarían a complementar el *aprendizaje por observación*, que sería el implementado por los investigadores en Ingeniería de Software, a la hora de hacer el taller práctico de modelación y después de presentar los videos de los conceptos técnicos de los lenguajes de programación y del problema hipotético. Entendiendo, entonces, el modelamiento como “cualquier proceso en que la información se suministra por el ejemplo, antes de permitir la práctica directa”^[14].

Por último, dentro del enfoque didáctico se trabajó con el *Aprendizaje Basado en Problemas* (ABP), en inglés: *Problem Based Learning* (PBL) el cual fue aplicado por los ingenieros durante el taller teórico práctico. Esta corriente iniciada en Canadá, es conocida por sus características de multimetodología y multididáctica y parte de un problema real para ser estudiado, analizado, comprendido o solucionado por un grupo pequeños de estudiantes, con el fin de propiciar entre ellos la composición de conocimiento. Y, con esta perspectiva fue que se diseñó el audiovisual último, a ser presentado tanto a la audiencia de PiCO como a la de GraPiCO, pues se pensó en una situación real y cotidiana, pero que pudiese ser fácilmente asimilada para ser modelada en el contexto profesional de los estudiantes de Ingeniería de Software.

VII. Conclusiones

Es importante tener presente que la comunicación está inmersa en todos los procesos humanos, por tanto, cualquier proyecto de la disciplina, ciencia o campo del saber que sea, debe contemplar un componente comunicativo dentro de la investigación, porque así sea al finalizar el proyecto, es decir, al momento de socializar la información, siempre será necesario entablar un proceso de comunicación.

En los procesos de comunicación es indispensable contar con el apoyo de expertos en el tema, pues si bien la capacidad de comunicarse es inherente a la naturaleza humana, hay estrategias que manejan técnicas y metodologías especializadas para tener mayores probabilidades de éxito en el envío del mensaje.

Las estrategias de comunicación en el campo organizacional pueden adaptarse a las investigaciones con enfoque didáctico, pues el público objetivo es una comunidad u organización como cualquier otra, por tanto, las metodologías aplicadas en las organizaciones no son muy lejanas de este tipo de procesos que también tienen un componente comunicativo.

Para trabajar en un proyecto con enfoque didáctico es indispensable pensar en las características del público objetivo y cuál sería la estrategia de comunicación más adecuada, dependiendo del tipo de aprendizaje que se adapta más a la investigación en desarrollo. En este punto es fundamental trabajar de la mano con profesionales de otras áreas que cuenten con la experticia necesaria para dar el enfoque requerido por el proyecto.

VIII. Bibliografía

- [1] J. F. Díaz y C. A. Tavera. Nuevo cálculo visual: GraPiCO. En II Congreso Colombiano de Computación, Universidad Javeriana. Bogotá, 2007.
- [2] J. F. Díaz y C. A. Tavera. Breve Discusión de las Ventajas de los Lenguajes Visuales frente a los Textuales: Caso de Estudio el Cálculo GraPiCO. En III Congreso Colombiano de Computación. Medellín, 2008.
- [3] I. Sommerville. Ingeniería del Software. Séptima Edición. Pearson Educación S.S. Madrid, 2005.
- [4] M. De Moragas. Citado por: J. L. Piñuel Raigada. En: La docencia universitaria de las teorías de la comunicación. Universidad Complutense Madrid. Revista Diálogos de la Comunicación. No. 78. Enero-julio de 2009. p. 166.
- [5] J. M. Pereira. La comunicación, un campo de conocimiento en construcción. Reflexiones sobre la Comunicación Social en Colombia. Bogotá: Universidad Javeriana. Revista investigación y desarrollo vol. 13, n° 2, 2005.
- [6] P. Bourdieu. Le champ scientifique. En Actes de la Recherche en Sciences Sociales, 2-3, 1976.
- [7] G. Bachelard. Conocimiento común y conocimiento científico. En: El racionalismo aplicado. Traducción de Irene Ramos. Buenos Aires: Paidós, 1978. 10 p.A. L. Webster, Estadística aplicada a los negocios y la economía. Bogotá: Mc Graw Hill, 2001.
- [8] A. Blanco. La psicología comunitaria ¿una nueva utopía del siglo XX? En: Psicología Comunitaria. Madrid, Textos Visor, 1993.
- [9] E. C. Arellano. La Estrategia de Comunicación como un principio de integración/interacción dentro de las organizaciones. En Revista Razón y Palabra. Suplemento Especial, Año 3. Enero-Marzo, 1998.
- [10] M. Martín Serrano. La Producción Social de Comunicación. Editorial Alianza. México, 1994.

- [11] W. A. Kelly. *La Psicología de la Educación*. 7ª Edición. Morata, S. A. Madrid, 1982.
- [12] D. Goleman. *El punto ciego*. Plaza & Janés Editores. España, 1997.
- [13] D. Coon. *Psicología*. 10ª Edición. Cengage Learning Editores. México, 2005.
- [14] T. L. Rosenthal y T. A. Steffek. *Modeling methods*. En F.H. Kanfer y A.P. Goldstein. 4ª Edición. *Helping people change. A textbook of methods*, 1991.

SISTEMATIZACIÓN DE UNA EXPERIENCIA DE INVESTIGACIÓN ENTRE LA COMUNICACIÓN SOCIAL Y LA INGENIERÍA DE SOFTWARE

*Paola Andrea Ramírez Arcila
Juan David Penagos Muñoz
Yenny Viviana Cruz Pérez
Carlos Andrés Tavera Romero*

I. Introducción

En el año 2010 la Universidad de San Buenaventura Cali, en el marco del Grupo de Investigación para el Desarrollo de la Ingeniería de Software (LIDIS) emprendió el proyecto posdoctoral de su entonces director, el ingeniero de sistemas Carlos Andrés Tavera Romero, buscando realizar un estudio comparativo entre el cálculo textual PiCO y el cálculo visual GraPiCO^{[1][2]}, que hacen parte del programa E_GraPiCO^[3], esto con el fin de conocer bajo qué condiciones era más adecuado un lenguaje que otro y hacer mejoras a dicho software.

En la consecución de estos objetivos, además de la Ingeniería de Software, se vincularon componentes de comunicación y estadística, gracias a la participación de la Facultad de Comunicación Social y la Facultad de Ciencias Básicas de la Universidad Autónoma de Occidente de la ciudad de Cali.

En este trabajo los ingenieros de sistemas de la USBC necesitaron del apoyo de un grupo de comunicadores sociales de la UAO que se encargaran de la parte comunicacional del proyecto: la realización de dos audiovisuales para explicar conceptos básicos de Ingeniería de Software^[4] a un público objetivo (estudiantes de Ingeniería de Sistemas, seleccionados aleatoriamente), además de otro video que mostrara un caso específico para ser modelado bajo el editor E_GraPiCO.

Desde el equipo de comunicadores también se dio apoyo en aspectos como redacción de textos, instrucción en expresión corporal, oratoria, manejo de público y otros elementos que permitieron a los ingenieros adquirir las herramientas necesarias para exponer su proyecto ante el público.

A su vez, los comunicadores se apropiaron de este como su proyecto de grado, y decidieron por objetivo principal sistematizar la experiencia vivida dentro de esta investigación posdoctoral interdisciplinaria. Ya en los objetivos específicos optaron por diseñar y desarrollar un modelo de sistematización que permitiera mostrar cada una de las etapas experimentadas durante el proceso y, finalmente, a partir de los resultados obtenidos, hacer una reflexión con respecto a los aportes que la Comunicación Social hizo a la Ingeniería de Sistemas.

Fue así, como adoptaron los conceptos de Juan Carlos Asisten, un especialista argentino en diseño y comunicación, y se tomó su modelo como ejemplo para realizar el proceso de producción de los audiovisuales.

De la misma manera, se tuvieron en cuenta autores que han estudiado el tema de la sistematización: Arizaldo Carvajal Burbano, sociólogo y especialista en Teoría y Métodos de Investigación en Sociología de la Universidad del Valle, y Óscar Jara, director general del Centro de Estudios y Publicaciones de Alforja y Coordinador del Programa Latinoamericano de Apoyo a la Sistematización de

Experiencias de CEAAL, fueron los escogidos como punto de partida, gracias a las posturas teóricas que usaron en sus respectivas investigaciones.

En esta investigación se empleó la metodología aplicada y, en correspondencia, se utilizaron conocimientos prácticos para desarrollarlos con un grupo específico de personas, en este caso: los estudiantes de Ingeniería de Sistemas seleccionados como público objetivo. El enfoque investigativo utilizado fue del tipo hermenéutico, por emplear la interpretación en momentos como la definición de los conceptos y su funcionalidad.

Como parte del proceso desarrollado: aciertos, falencias y demás, se conformó un documento titulado Una propuesta pedagógica de sistematización interdisciplinar entre la Ingeniería de Sistemas y la Comunicación Social, trabajo de grado resultante de la investigación y que se enmarcó específicamente dentro de la perspectiva de los comunicadores.

Durante el proceso de esta investigación se trabajó una serie de pasos que articulan las diferentes etapas del proyecto. A continuación se especifica la etapa a desarrollar en el presente capítulo. Ver Tabla 23.

Tabla 23. Etapas del estudio comparativo

Estudio comparativo entre lenguajes textuales y lenguajes visuales. Caso: PiCO y GraPiCO	
Etapa 1	Elaboración de hipótesis en experimentos de lenguajes de programación.
Etapa 2	VARIABLES en un experimento de lenguajes de programación.
Etapa 3	Unidades experimentales utilizadas en pruebas de lenguajes de programación.
Etapa 4	Tratamientos y replicas en un experimento de programación.
Etapa 5	Aprendizaje Basado en Problemas (ABP) aplicado a los lenguajes de programación.
Etapa 6	La comunicación en el estudio comparativo entre lenguajes textuales y lenguajes visuales: Caso PiCO y GraPiCO.
Etapa 7	Sistematización de una experiencia de investigación entre la Comunicación Social y la Ingeniería de Software.
Etapa 8	Modelo de sistematización propuesto "TCACI en doble vía."
Etapa 9	Pasos en la realización de los audiovisuales pedagógicos: PiCO – GraPiCO y ejercicio de modelación.
Etapa 10	Estudio de resultados de PiCO y GraPiCO. Parte 1.
Etapa 11	Estudio de resultados de PiCO y GraPiCO. Parte 2.
Etapa 12	Recomendaciones y resultados del estudio entre PiCO, GraPiCO y editores.

Fuente: elaboración propia (2018).

II. ¿Qué es sistematizar una experiencia?

Basándose en que la práctica experimentada fue de gran importancia para la producción de conocimiento y cooperación con la Ingeniería de Sistemas, el grupo de Comunicación decidió sistematizar esta experiencia tomando como base el proceso descrito en la monografía de Arizaldo Carvajal: Teoría y práctica de la sistematización de experiencias, cuyo principal planteamiento expone que: “la sistematización es un proceso metodológico cuyo objeto es que el educador o promotor de un proyecto recupere su relación con la acción, organizando lo que sabe de su práctica para darla a conocer a otros”^[5].

Cabe anotar que, según este sociólogo: “No sistematizamos para informarnos de lo que sucede y seguir haciendo lo mismo, sino para mejorar, enriquecer, transformar nuestras prácticas.” Por lo que este proyecto se convierte en un aporte significativo para el campo de la Comunicación al ponerla al servicio de una disciplina diferente y, con ello, ampliar el campo de acción y las prácticas mismas.

Siguiendo en sintonía con la sistematización, diversas apreciaciones se encuentran cuando se indagan conceptos y teóricos que definen la idea. Del trabajo Marco Conceptual de la Sistematización de Experiencias de la Dirección General de Políticas de Desarrollo Social Dirección de investigación y Desarrollo Social Lima, 2010. El grupo de comunicadores recopiló varias de estas posturas que ayudan a reconocer lo que puede hacerse en este tipo de procesos:

“Es la reflexión acerca de la experiencia propia del proyecto a fin de identificar y explicitar las lecciones aprendidas en el proceso (tanto positivas como negativas)”^[6]

“Un proceso de reflexión que pretende ordenar u organizar lo que ha sido la marcha, los procesos, los resultados de un proyecto, buscando en tal dinámica las dimensiones que pueden explicar el curso que asumió el trabajo realizado”^[7]

“Es el momento final de la secuencia y está dirigido a organizar los resultados del aprendizaje realizado a través de los momentos anteriores. Este proceso implica el desarrollo, entre otras destrezas,

de la habilidad para establecer y manejar criterios de selección y ordenamiento; de la habilidad para discernir entre lo importante y trascendente y lo que es superfluo o circunstancial y de la capacidad de establecer categorías y jerarquías de valoración.”^[8]

Para cerrar este apartado, se trae de nuevo a colación al sociólogo Carvajal Burbano, quien a través de otro de sus planteamientos ayuda a justificar una razón más por la cual el grupo de comunicación tomó la determinación de sistematizar esta experiencia investigativa: “la difusión de la sistematización, permite que otros profesionales, inmersos en realidades parecidas y enfrentando problemas similares, puedan aprender de nuestra experiencia, permitiéndoles no partir siempre de cero y no repetir los errores que podamos haber cometido”^[5].

III. Relación entre la Comunicación y la Pedagogía

Con el propósito de apoyar esta investigación posdoctoral se hizo necesario cruzar la Ingeniería de Software y la Comunicación Social mediante el uso de material pedagógico audiovisual para la realización del estudio comparativo de los dos lenguajes de programación, y es en este punto específico donde entra a mediar la comunicación.

Y, precisamente, por ser explicativos y enseñar los términos utilizados por los ingenieros, además de mostrar la usabilidad de los dos lenguajes de programación, es que los productos audiovisuales diseñados dentro de este proyecto contemplan el componente pedagógico que lleva una alta carga comunicativa para cumplir a cabalidad su función educativa.

El investigador uruguayo Mario Kaplún, plantea que: “..los múltiples mensajes que a diario son emitidos en el escenario social: un periódico sindical..., un video para la prevención del SIDA..., un folleto sobre cooperativismo..., una campaña ecológica en pro de la preservación del medio ambiente..., incluso la prédica de un sacerdote bien mirados, todos ellos son mensajes educativos...”^[9]

Entonces, el audiovisual en este proyecto se hace indispensable en la medida en que captura dos de los cinco sentidos por medio de los cuales aprehendemos y aprendemos el conocimiento; y, en este caso, incluso, posibilita la utilización de herramientas para la socialización de los conceptos técnicos básicos, a través del uso de un lenguaje menos complejo que el empleado desde lo técnico.

También es importante tener presente que: “los medios audiovisuales presentan la ventaja de mantener el interés del alumno, y utilizados adecuadamente facilitan la presentación y comprensión de los contenidos”^[10], lo que ayuda a la comunicación para enviar el mensaje con menos posibilidades de distorsión y distracción.

IV. La Comunicación vista desde la interdisciplinariedad

Diferentes autores mencionan que: “la comunicación nunca ha tenido un campo disciplinar propio”^[11], sino más bien un dominio de estudio interdisciplinar en las ciencias sociales.

Profundizando más en el asunto, la interdisciplinariedad hace referencia a la ayuda o trabajo entre una ciencia y otra. Pero, aunque su idea central es superar la fragmentación del conocimiento que reflejan las disciplinas particulares, no se debe considerar como una simple suma de saberes de diferentes especialidades o como la sumatoria de puntos de vista distintos y limitados, si no mejor como una contribución a un todo para la consecución de algo nuevo e integrado.

En consecuencia, va más allá de lo previsto en ella, porque los participantes de las investigaciones *transcinden* las propias disciplinas y logran crear nuevos mapas comunes sobre los problemas en cuestión; entonces, la interdisciplinariedad se aparta de la norma de la eficacia propiciada por los procesos de las investigaciones monodisciplinarias. La “inter”, pues, actúa en un proceso investigativo como el equilibrio y la cooperación entre la parte interior de una ciencia o campo –para este proyecto, la Ingeniería de Software– y la exterior de otra –la Comunicación Social–.

V. Teorías de la comunicación utilizadas en el proyecto

A través de la historia el estudio de la comunicación se ha centrado en diferentes objetos de estudio, desde diferentes teorías y autores, y también detractores.

En un principio, la comunicación tomada desde el enfoque positivista con teóricos como Shannon y Weaver, planteó un modelo de máquina transmisora que buscaba *“sólo la verdad al transferir un mensaje con exactitud”*. Esto da cuenta de una comunicación lineal en la que aparece de primera una fuente de comunicación o emisor, sigue el mensaje con signos o señales, luego un codificador o emisor que lo transforma a través de un canal utilizado para transportar los signos, y de ahí se encuentra un descodificador o receptor que reconstruye el mensaje a partir de los signos. Esta explicación se fundamenta en el comunicado mostrado en ^[12]

Esta postura también sostiene que el único impedimento para no verificar la comunicación es por desconocimiento del código, lo que la reduce a estancias meramente técnicas.

Ya en el modelo físico-matemático desarrollado por Harold Laswell se asume la postura desde diferentes preguntas: *¿Quién dice? ¿Qué dice? ¿Qué canal? ¿A quién lo dice? ¿Con qué efectos?* ^[13]. Este arquetipo, surgido a partir de las telecomunicaciones, sostiene que hay una transmisión de información entre un origen y un destino.

En consonancia cronológica se conoce el modelo funcionalista, que centra su investigación en los medios masivos de comunicación, diferentes autores (como: Talcott Parsons, Robert Nerton, George Mead, Ivan Pavlov, Frederick Skinner y Harold Laswell, entre otros) indagaron sobre la eficiencia e influencia de los medios en la sociedad y sus consecuencias, con el fin de ser efectivos en el mensaje enviado; su postura hacía referencia a la estandarización del mensaje.

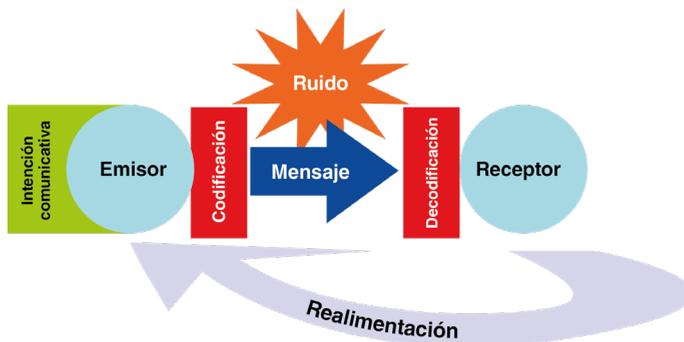
Desde la Teoría Crítica de la Escuela de Frankfurt se estudió el impacto de los medios de comunicación en la sociedad, sus pro-

cesos de dominación y poder. Los teóricos (como: Theodor Adorno, Max Horkheimer, Herbert Marcuse, Walter Benjamin, Michael Foucault, Antonio Gramsci y Jürgen Habermas) sostienen que los medios de comunicación tienden a estandarizar al individuo bajo estereotipos. “Los medios de comunicación están en manos de los mismo dueños de los modos de producción”^[14].

A finales de los sesenta, aparece una nueva forma de concebir la comunicación, desde la hermenéutica, es decir, desde la interpretación. Para estos teóricos el mensaje no llega estandarizado, cada uno elabora un proceso de interpretación. “La comunicación se convierte en la acción dadora de sentido, en producción simbólica. Intercambio simbólico, desciframientos y ocultamientos. Ya no sólo el trabajo sino también el amor, el deseo, la risa, el comer y hasta enterrar a los muertos. Toda actividad humana está instalada en un trasfondo de orden comunicativo. El hombre es el ser simbólico”^[12].

Siguiendo con teorías de la comunicación y apelando a la que le compete a los comunicadores sociales participantes en este proyecto, desde la interpretación de autores más actuales hicieron parte del proceso: el diseñador y comunicador argentino Juan Carlos Asisten, propone un modelo más complejo que fue adoptado por el grupo comunicadores en la búsqueda de esas bases teóricas que les permitieran una práctica más efectiva^[15]. Cabe acordar, que aquí no se maneja un sentido meramente lingüístico, si no que se incluye lo semántico.

Figura 9. Modelo de comunicación de Juan C. Asisten



Fuente: Modelo de Juan C. Asisten en el libro [15] (2018).

En este proyecto lo audiovisual tuvo un papel importante para el desarrollo de la explicación. Éste fue usado como una herramienta de trabajo necesaria para propagar el mensaje, y es por esta razón que se tuvo muy presente la etapa de planificación para la elaboración y selección de las imágenes, como lo predica José Miguel Pereira “...requiere, como cualquier otro proceso de comunicación intencional, la debida planificación. No debemos dejarnos subyugar por ‘imágenes bonitas’. Orientémonos a crear o seleccionar aquellas adecuadas a nuestra intención comunicativa”^[11]

El proceso de selección de planos, imágenes, sonidos, diseños interactivos que hacen parte del audiovisual, fueron escogidos estrictamente para mantener la intención comunicativa que necesitaban en ese caso específico los ingenieros de sistemas.

Según, Mayra Teresa Millán, en su investigación sobre Modelos y Teorías de Comunicación, un signo puede ser definido como toda cosa que sustituye a otro de modo que desencadene en relación a un tercero, un complejo análogo de reacciones^[16]. Desde Charles Peirce se han tratado de establecer los principios de la semiótica, llegándose a clasificar en: ícono y símbolo, y sus niveles del signo en: sintáctico, semántico y pragmático. En este proyecto esto se vio reflejado en el uso de signos, imágenes, sonidos y composiciones visuales dentro de los audiovisuales que, en un sentido explicativo, buscaban enviar una idea específica al receptor.

VI. Objetivos del proyecto desde la comunicación

El objetivo general de este proyecto fue sistematizar la experiencia del proceso de articulación entre la Comunicación Social y la Ingeniería de Software, además de contribuir desde la comunicación misma con una investigación posdoctoral interdisciplinaria, algo quizás un poco revolucionario frente a lo que tradicionalmente se trabaja.

Entre las muchas preguntas que surgieron, el grupo de comuni-

cadores sociales logró resolver las siguientes:

¿De qué manera contribuir con otros procesos investigativos contando la experiencia vivida y las fases desarrolladas?

Este documento hace parte de una serie de informes compilados. De la misma manera, estos escritos se publicarán a manera de artículos en diferentes revistas interesadas en temas como la sistematización de experiencias, la investigación interdisciplinaria y el diseño y uso de audiovisuales pedagógicos; y, en lo posible, se espera seguir socializando los resultados en congresos, seminarios y otros eventos similares.

¿Cómo diseñar y articular dos videos pedagógicos que permitieran presentar los conceptos básicos de los lenguajes de programación textual –PiCO– y visual –GraPiCO– de una forma clara, precisa, equilibrada y dinámica?

Se realizaron dos videos expositivos y pedagógicos que permitieron ilustrar los conceptos necesarios para el grupo de ingenieros de sistemas y que, además, les permitieron desarrollar su proyecto de investigación; la comunicación sirvió entonces como una herramienta.

También se realizó un tercer audiovisual, a manera de ejercicio de modelación, para que los estudiantes asistentes al auditorio pusieran en práctica lo aprehendido de los otros dos videos.

¿De qué manera demostrar la utilidad de los audiovisuales por medio del análisis de las dos encuestas realizadas?

Se ejecutó una serie de encuestas al público objetivo de la investigación (estudiantes de Ingeniería de Sistemas seleccionados aleatoriamente), quienes serían los observadores de los tres videos. Los resultados arrojaron respuestas satisfactorias que evidenciaron cómo los dos primeros audiovisuales permitieron entender los conceptos básicos de los lenguajes de programación en estudio, y fueron claves para modelar el ejercicio final presentado en el tercer video.

VII. Metodología empleada en el proyecto

El enfoque investigativo en este proyecto fue Hermenéutico dado que, a través de la comunicación, se interpretó la definición de los conceptos del Cálculo textual: PiCO y el Cálculo visual: GraPiCO, y se describió la funcionalidad de ambos lenguajes de programación, haciendo uso de herramientas audiovisuales para transmitir el mensaje de una manera sencilla y precisa, al público objetivo.

Según Arizaldo Carvajal, “la sistematización hermenéutica es el proceso que permite dar cuenta de la racionalidad interna de las experiencias estudiadas y el sentido que tienen para sus actores. Es un proceso de comprensión de sentidos en contextos específicos, –en este caso contexto de comunicadores e ingenieros– en los cuales diversas interpretaciones buscan su legitimación”^[5].

Para llevar a cabo este proyecto se necesitó consultar fuentes primarias que facilitaron al grupo de Comunicadores Sociales apropiarse de las definiciones y explicaciones sobre cada uno de los conceptos del trabajo de investigación. Así, los investigadores de la USBC hicieron numerosas charlas explicativas sobre el uso y las definiciones de cada lenguaje de programación, y se entrevistó varias veces a Carlos Andrés Tavera, Director del grupo LIDIS, para profundizar en los temas y aclarar las dudas finales.

Por último, entre las fuentes secundarias y o documentales se tomaron diferentes teorías de comunicación, que aportaron desde el campo de conocimiento propio, el uso de un lenguaje apropiado para que la información fuera totalmente clara. Además, se consultaron monografías y documentos sobre la sistematización de experiencias, que permitieron elaborar un proceso en la recolección de la información y una organización a la hora de elaborar los productos audiovisuales.

VIII. Conclusiones

- El trabajo interdisciplinar da cuenta de un mundo en el que hay que aunar esfuerzos para lograr un determinado objetivo, ciencia; las disciplinas se unen para resolver un problema que solo pueden solucionar de manera conjunta.

- Este proyecto dio cuenta de que la unión estratégica entre la Comunicación Social y la Ingeniería de Sistemas se complementó de manera satisfactoria, de manera tal, que los productos escogidos, en este caso audiovisuales, tuvieron aceptación por parte de los estudiantes de Ingeniería, quienes manifestaron mejor entendimiento ante los conceptos de cálculos de lenguajes de programación; gracias a los audiovisuales presentados, los estudiantes de primeros semestres de Ingeniería lograron comprender e interiorizar de manera eficaz dichos conceptos.
- Así como fueron de gran importancia los audiovisuales para el entendimiento de los conceptos informáticos, la parte estética de los videos también jugó un papel de suma importancia, pues sin la sonorización, las imágenes, animaciones y demás efectos adecuados, el objetivo no se habría logrado. Estos fueron pieza clave para su interiorización, porque en definitiva, con una mala utilización de los elementos mencionados el mensaje podría distorsionarse.
- Se demostró que la comunicación fue vital en el proyecto, su aporte fue de suma importancia y permitió a los ingenieros seguir con su investigación y terminarla de manera satisfactoria.
- Se corroboró que la interdisciplinariedad es una apuesta de la academia para dar respuestas al mundo actual y hacer énfasis en dar soluciones a un fin desde diferentes ópticas, posturas, campos o disciplinas.
- Finalmente y sin duda alguna, sistematizar experiencias es un proceso de aprendizaje incalculable que ayuda a plasmar y analizar el paso a paso de un proyecto para no caer en errores cometidos, a dar cuenta de cada fase y a evaluar el desarrollo del mismo, y lo más importante, a servir de antecedentes, guías o ejemplos a otros investigadores.

IX. Bibliografía

- [1] J. F. Díaz y C. A. Tavera. Nuevo cálculo visual: GraPiCO. En II Congreso Colombiano de Computación, Universidad Javeriana. Bogotá, 2007.
- [2] J. F. Díaz y C. A. Tavera. Breve Discusión de las Ventajas de los Lenguajes Visuales frente a los Textuales: Caso de Estudio el Cálculo GraPiCO. En III Congreso Colombiano de Computación. Medellín, 2008.
- [3] Tavera Romero, C. A., Días Frias, J. F., Soto Vargas, A., Gallego Varona, J. J., & Jojoa, A. (2007). Alternativa de Comprobación sintáctica de VLP: Gsig Parsing. Aspectos formales y el caso de estudio: E_GraPiCO. En R. d. Informática, G. R. Simari, & A. E. De Giusti (Edits.), Anales del XIII Congreso Argentino de Ciencias de la Computación (págs. 1632 - 1644). Argentina: RedUNCI.
- [4] I. Sommerville. Ingeniería del Software. Séptima Edición. Pearson Educación S.S. Madrid, 2005.
- [5] A. Carvajal Burabano. Teoría y Práctica de la Sistematización de Experiencia. 3ra Edición - Santiago de Cali, Agosto 2007. Escuela de Trabajo social y Desarrollo Humano - Universidad del Valle. Educación incluyente / Educación para adultos / Pagina web <http://www.colombiaaprende.edu.co/html/home/1592/article-228165.html> - En linea
- [6] Villa, V., & Ocampos, A. (2015). Citada en Guía para la sistematización de experiencias de intervenciones públicas. En M. d. Perú. Lima, Perú.
- [7] Berdegué Sacristan, J. A. (2015). Citado en Guía para la sistematización de experiencias de intervenciones públicas. En República del Perú, Ministerio de Salud, Programa de Apoyo a la Reforma del Sector Salud PARSALUD II (1ª ed., pág. 13). Lima, Perú.
- [8] Tréllez Solís, E. (2015). Citada en Guía para la sistematización de experiencias de intervenciones públicas. En República del Perú, Ministerio de Salud, Programa de Apoyo a la Reforma del Sector Salud PARSALUD II (1ª ed., pág. 13). Lima, Perú.

- [9]Kaplún, M. (2002). Una pedagogía de la comunicación. En O. M. Pérez (Ed.). Habana, Cuba: Caminos.
- [10] Pérez Romero, J. T. (2003). Cuerpo de maestros: Temario común. CEP Editorial.
- [11] J. M. Pereira, La comunicación: un campo de conocimiento en construcción, reflexiones sobre la comunicación social en Colombia, (2005) http://ciruelo.uninorte.edu.co/pdf/invest_desarrollo/13-2/9_La%20comunicacion.pdf PIAGET jean – [En línea] [consultado 24 de Octubre de 2012] – Disponible en: http://webdelprofesor.ula.ve/ciencias/ricardo/PDF/La_interdisciplinaridad.pdf
- [12] Asociación Colombiana de Facultades y Programas Universitarios de Comunicación-AFACOM. (2004). Exámenes de calidad de la educación superior - ECAES en comunicación e información: Marco de fundamentación conceptual y especificaciones de la prueba ECAES en Comunicación e Información. Bogotá, Colombia.
- [13] Lasswell, H. D. (1948). The Structure and Function of Communication in Society. Harper & Bros .
- [14] Osorio, S. N. (2007). La Teoría Crítica de la Sociedad de la Escuela de Frankfurt: algunos presupuestos teórico-críticos. Educación y Desarrollo Social , 1 (1), 104-119.
- [15] Asinsten, J. C. (2017). Comunicación visual y tecnología de gráficos en computadora. educ.ar y Ministerio de Educación Ciencia y Tecnología.
- [16] Millán Tapia, M. T. (2012). *Modelos y teorías de la Comunicación*. Londres, Inglaterra: Universidad de Londres

MODELO DE SISTEMATIZACIÓN PROPUESTO “TCACI EN DOBLE VÍA”

*Paola Andrea Ramírez Arcila
Juan David Penagos Muñoz
Yenny Viviana Cruz Pérez
Carlos Andrés Tavera Romero*

I. Introducción

Este es un proyecto interdisciplinario e interinstitucional entre la Ingeniería de Sistemas y la Comunicación Social, donde participan la Universidad de San Buenaventura Cali –USBC– (por parte de los ingenieros) y la Universidad Autónoma de Occidente UAO (por parte de los comunicadores), el proyecto en cuestión se gestó con el fin de apoyar al Laboratorio de Investigación para el Desarrollo de la Ingeniería de Software (LIDIS) de la USBC, en un trabajo posdoctoral^[1] consistente en el estudio comparativo entre el cálculo textual: PiCO^[2] y el cálculo visual: GraPiCO^{[3][4]}, que hacen parte del editor de programas visuales de E_GraPiCO^[5]. Tal apoyo fue necesario para conocer bajo qué condiciones es más adecuado un lenguaje frente al otro y qué mejoras requieren.

Si bien fue importante el apoyo que la Comunicación le dio a la Ingeniería, igual fue la labor de los comunicadores al proponer un modelo de sistematización desde la interdisciplinariedad, lo que se convirtió en un gran reto, no solo por adentrarse y apropiarse de conceptos de otra disciplina, sino por atreverse a contar la experiencia bajo un modelo propio.

Basados en diferentes apreciaciones de autores expertos en la sistematización como Arizaldo Carvajal y Óscar Jara, los comunicadores de la Universidad Autónoma de Occidente trabajaron esta metodología para contar aciertos y desaciertos del proceso adelantado con los ingenieros de sistemas de la Universidad de San Buenaventura Cali, y del proyecto posdoctoral sobre el estudio comparativo entre dos cálculos computacionales.

El equipo de estudiantes comunicadores decidió enfocar desde esta experiencia el trabajo final de su carrera, dando como resultado el modelo de sistematización TCACI en doble vía, que permitió mostrar las etapas del proceso y que es uno de los objetivos específicos de su proyecto de pregrado. Al final, se hace una reflexión frente a los aportes que la Comunicación Social hizo a la Ingeniería de Sistemas, a partir de los resultados obtenidos en la última fase.

Durante esta investigación se trabajó una serie de pasos que articulan las etapas del proyecto vividas por todo el equipo de investigadores: ingenieros de sistemas y comunicadores sociales. A continuación, se especifica la etapa a desarrollar en el presente informe en el marco del macroproyecto. Ver la Etapa 8 en la Tabla 24.

Tabla 24. *Etapas del estudio comparativo*

Estudio comparativo entre lenguajes textuales y lenguajes visuales. Caso: PiCO y GraPiCO	
Etapa 1	Elaboración de hipótesis en experimentos de lenguajes de programación.
Etapa 2	Variables en un experimento de lenguajes de programación.
Etapa 3	Unidades experimentales utilizadas en pruebas de lenguajes de programación.
Etapa 4	Tratamientos y replicas en un experimento de programación.
Etapa 5	Aprendizaje Basado en Problemas (ABP) aplicado a los lenguajes de programación.

Etapa 6	La comunicación en el estudio comparativo entre lenguajes textuales y lenguajes visuales: Caso PiCO y GraPiCO.
Etapa 7	Sistematización de una experiencia de investigación entre la Comunicación Social y la Ingeniería de Software.
Etapa 8	Modelo de sistematización propuesto “TCACI en doble vía.”
Etapa 9	Pasos en la realización de los audiovisuales pedagógicos: PiCO – GraPiCO y ejercicio de modelación.
Etapa 10	Estudio de resultados de PiCO y GraPiCO. Parte 1.
Etapa 11	Estudio de resultados de PiCO y GraPiCO. Parte 2.
Etapa 12	Recomendaciones y resultados del estudio entre PiCO, GraPiCO y editores.

Fuente: elaboración propia (2018).

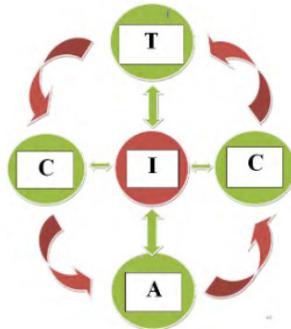
El modelo consta de cinco estaciones. Son cuatro las encargadas de guiar el trabajo del investigador respecto al tema específico que está abordando: *Tematiza, Conoce, Aplica y Concluye*. Seguido a ello, cabe una quinta estación denominada *Inquieta*, basada en realizar preguntas que conduzcan al mejoramiento de las anteriores estaciones y los elementos que las componen. Es decir, es la fase evaluativa de cada punto y la que permite saber sí, en realidad, el modelo está funcionando o se debe corregir. Además, el modelo propuesto contiene las etapas desarrolladas durante el proyecto, el trabajo realizado con los ingenieros y la experiencia como tal. Estas etapas se titularon: Ingeniería comunicacional y Comunicando sapiencias.

En resumen, este artículo expondrá el trabajo logrado entre ingenieros y comunicadores, el paso a paso recorrido, y los aciertos y desaciertos del proceso.

II. Estaciones del Modelo

El modelo consta de las siguientes partes ver la Figura 10.

Figura 10. Estaciones del modelo.



Fuente: elaboración propia (2018).

A. Tematiza

En esta primera estación se reflexiona respecto al tema que se trabajó, por ejemplo: ¿Qué se pretendió con ese tema?, ¿Por qué ese tema?, ¿Qué se planteó con ese tema?

B. Conoce

Esta etapa se basa en las investigaciones realizadas frente al tópico trabajado, entrevistas, autores, artículos y demás elementos que sirvieron como fuente y estado del arte para el desarrollo del trabajo.

C. Aplica

Da cuenta de lo que se hizo durante la investigación. Productos realizados, su explicación y cómo se realizaron. Caben aquí entrevistas, documentales, crónicas, modelos, etc.

D. Concluye

En este punto se presentan los aportes del proyecto, ganancias (intelectuales), aportes y reflexiones que permitan demostrar la importancia de la investigación.

E. Inquieta

Este último punto es la evaluación del modelo. Desde aquí, se plantean preguntas pertinentes para el mejoramiento o afianzamiento del trabajo realizado y de cada una de las estaciones por las que pasó el proceso para la sistematización. Se realizan preguntas tales como; ¿Para qué se hizo?, ¿por qué se hizo?, ¿qué faltó?, ¿cómo se hizo?, ¿qué se hizo?, ¿fue suficiente?. En síntesis, cualquier pregunta que permita evaluar lo que se hizo y su real funcionalidad. A su vez, *Inquieta*, es la estación que permite apreciar la sistematización fase por fase, y muestra la efectividad del modelo.

III. ¿Cómo debe usarse el modelo?

Comienza con la estación T y gira en sentido contrario a las manecillas del reloj, siguiendo en el orden de C, A, C, pero con la salvedad del punto central denominado *Inquieta*, ya que este punto hace énfasis en cada una de las estaciones que componen el modelo.

Llegando al punto *Conoce*, se realiza una comunicación hacia *Inquieta*. Aquí se hacen las preguntas pertinentes para evaluar los elementos del punto anterior. Cuando las preguntas se realizan y son respondidas de manera positiva, se regresa a la estación *Conoce* y, posteriormente, se pasa a *Aplica* y *Concluye*. Por esta razón, se dice que hay una “doble vía” en la comunicación entre todas las estaciones. En el caso de que las respuestas en *Inquieta* sean negativas o poco favorecedoras para el trabajo, se regresa a la estación anterior. Pero buscando solucionar o mejorar los elementos que no tuvieron la respuesta esperada.

IV. Aplicación del Modelo de Sistematización

El modelo se aplicó en el proyecto llamado: Propuesta pedagógica de sistematización interdisciplinaria entre la Ingeniería de Sistemas y la Comunicación Social. La experiencia por estaciones fue la siguiente.

A. Tematiza

Sistematización de una experiencia interdisciplinar entre la Ingeniería de Sistemas y la Comunicación Social.

1) ¿Para qué se hizo?

Para proponer un modelo de sistematización de experiencias en un proyecto de comunicación interdisciplinar que, además, fuera aplicado y diera cuenta del proceso y la unión entre ingenieros de sistemas y comunicadores sociales.

Para reflexionar sobre los aportes de la comunicación social en ese proceso interdisciplinario, a partir de los resultados arrojados en el proyecto.

2) ¿Por qué se hizo?

Porque el grupo de comunicadores sociales quiso contar la experiencia de este proyecto interdisciplinar a partir de su aporte a los ingenieros, y porque no se conocía otro modelo para sistematizar experiencias entre ingeniería y comunicación.

3) ¿Fue suficiente?

Sí fue suficiente. Pues por medio del proyecto se aportaron argumentos a la tesis que defiende la interdisciplinariedad y que la define como necesaria para resolver problemas del mundo actual, donde se plantea la necesidad de unir sapiencias de dos o más ciencias o campos en pro de un mismo objetivo.

B. Conoce

Se realizaron investigaciones y apropiaciones de temas que serían necesarios para todo el proceso. Por ejemplo, los comunicadores se apropiaron de conceptos informáticos manejados por los ingenieros y viceversa. Así, se realizó una entrevista con el director de programa de Maestría en Ingeniería de Software en la Universidad de San Buenaventura-Cali. Con él se compartieron varios de los conceptos apropiados por el grupo, significado del proyecto y consistencia del proceso.

Con el fin de documentarse para la realización de la presentación audiovisual, se realizaron investigaciones previas de proyectos in-

terdisciplinarios y metodologías empleadas en modalidades didácticas y pedagógicas de las dos disciplinas ya mencionadas.

Durante el proceso de investigación, también se buscó la manera de trasladar la información más densa de los conceptos programáticos a un lenguaje menos técnico y de mayor accesibilidad para el público objetivo.

1) *¿Para qué se hizo?*

Se hizo con el fin de compenetrarse con el proyecto, conocer antecedentes y estar preparados para la etapa de producción del proceso.

2) *¿Por qué se hizo?*

Porque en todo proyecto de investigación debe hacerse una documentación previa y un estado del arte que presente los antecedentes del trabajo.

3) *¿Qué faltó?*

Faltó indagar en las posibilidades de proyectos similares, de manera que se disminuyeran los obstáculos a partir de las experiencias previas. Sin embargo, esto no se cumplió porque no se encontraron proyectos de sistematización con condiciones de interdisciplinariedad entre dos disciplinas tan distantes como la comunicación social y la ingeniería de sistemas.

4) *¿Cómo se hizo?*

Todo el material obtenido en la investigación se obtuvo en jornadas de estudio y búsqueda bibliográfica y ejercicios de indagación con fuentes de información.

C. Aplica

Aquí se realizan los tres audiovisuales didácticos y pedagógicos planteados en el proyecto de investigación.

D. Audiovisual PiCO

Este video explica los conceptos manejados en el cálculo de lenguaje de programación textual. Es decir, para qué sirve, qué lo compone y cómo funciona.

E. Audiovisual GraPiCO

El video del cálculo computacional visual GraPiCO fue equivalente al del cálculo computacional textual PiCO, los dos videos contaron con la misma narrativa visual para garantizar el equilibrio en el estudio comparativo que debían realizar los ingenieros entre cálculos computacionales textuales y visuales. Con esto se evitó arriesgar la objetividad de los resultados por afectaciones de carácter estético.

1) ¿Fue suficiente?

Sí, fue suficiente. Los audiovisuales PiCO y GraPiCO tuvieron aceptación entre el público objetivo y cumplieron el propósito pedagógico para el cual fueron realizados.

Este proyecto sirvió para mostrar un modelo de sistematización de experiencias, práctico, paso a paso, completo y, sobre todo, que reconoce la importancia de la retroalimentación, pues se desarrolló en doble vía, y esto permitió evaluar cada estación en su momento, sin necesidad de llegar a la última para retornar al principio.

2) ¿Qué faltó?

En este punto falta reconocimiento del modelo de sistematización de experiencias dentro de la sociedad académica, de manera que sea aplicado en otros proyectos.

V. La Metodología utilizada en el Proyecto

Dado que, a través de la comunicación se interpretó la definición de los conceptos y la funcionalidad en los dos cálculos computacionales para transmitirlos a los estudiantes de Ingeniería de Sistemas de una manera clara y sencilla, se trabajó bajo el enfoque hermenéutico.

Como se puede encontrar en el libro presentado en ^[6], la sistematización hermenéutica es el proceso que permite dar cuenta de la racionalidad interna de las experiencias estudiadas y el sentido que tienen para sus actores. Es un proceso de comprensión de sentidos en contextos específicos, —en este caso contexto de comunicadores e ingenieros— en los cuales diversas interpretaciones buscan su legitimación.

Esta investigación se enmarcó en la sistematización de la experiencia a partir de la contribución de dos productos audiovisuales desde la comunicación para facilitar a los ingenieros de sistemas del grupo LIDIS, un mejoramiento del editor de programas visuales E_GraPiCO.

Según Oscar Jara en el informe presentado en [7]: “para la sistematización es muy importante el ordenamiento y la clasificación de la información, estructurando de manera precisa categorías, relaciones, posibilitando la constitución de base de datos organizados”.

Teniendo en cuenta lo anterior, para llevar a cabo este proyecto fueron necesarias la comprensión y definición de conceptos de los dos calculos computacionales con sus diferencias en la funcionalidad (PiCO-textual, GraPiCO-visual). Posteriormente, cuando ya se había hecho la apropiación de los conceptos mencionados, se realizó un esquema para la producción de los audiovisuales.

Juan Carlos Asinsten, especialista argentino en diseño y comunicación, afirma en el documento presentado en [8] que es mínima la comunicación basada exclusivamente en imágenes. “En la mayor parte de la comunicación visual tienen preponderancia las imágenes, pero se complementan con textos, sonidos, locuciones, que acotan y precisan su sentido, su significación” ver la Figura 11. De la misma forma, se pretendió estructurar los dos productos audiovisuales. El fondo de la presentación es el mismo editor E_GraPiCO, la presentadora, se encuentra justificada hacia a la derecha y en la parte izquierda, se muestran las funcionalidades de los conceptos, que se exponen con sonidos y nombres.

Figura 11. Modelo de comunicación de Juan C. Asinsten



Fuente: Modelo de Juan C. Asinsten en el libro [15] (2018).

VI. Etapas en el proceso

A continuación se discuten las dos etapas en las que el proceso puede ser dividido desde la Comunicación Social:

A. Etapa 1: Ingeniería comunicacional

En esta primera etapa, los comunicadores se apropiaron de conceptos informáticos que los ingenieros de sistemas manejaron o utilizaron en su proyecto con los cálculos computacionales GraPiCO y PiCO y que necesitaban trasladar a un lenguaje comprensible para dar a entender su trabajo. La interpretación e interiorización de dichos conceptos se realizó para que los comunicadores lograran presentar la información de forma menos técnica para el mayor entendimiento del auditorio.

Durante el proceso de aprendizaje se realizaron los guiones de videos y se puso en marcha la producción de las piezas audiovisuales, con el fin de lograr los objetivos de comunicadores e ingenieros.

Para realizar los videos visual (GraPiCO) y textual (PiCO), los comunicadores tuvieron en cuenta la importancia de cada concepto aprendido dentro de su lenguaje de programación para establecer el orden de aparición en el producto final. Sin embargo, contaron con el aporte de los ingenieros, quienes ratificaron la estructura establecida. Constantes reuniones de ambos grupos para concertar una estructura fueron fundamentales en la organización de la información suministrada en la etapa de investigación.

Acto seguido, se realizó la grabación del video prototipo y se dio inicio a la producción audiovisual. La primera prueba se ejecutó en el estudio de televisión de la Universidad San Buenaventura Cali, donde las personas elegidas como presentadoras de los audiovisuales fueron las dos mujeres del grupo de comunicación que contaban con la ventaja de conocer el proyecto y el funcionamiento de los lenguajes de programación. Además, sirvió como piloto para definir si alguna de las dos podría ser la conductora de los audiovisuales.

Por otro lado, se realizó la composición visual basada en el video clip estadounidense de Anne Leonard "La historia de las cosas," un

modelo interactivo y dinámico para explicar un tema tan extenso como este.

Realizado el prototipo, se llevó a cabo la retroalimentación con los ingenieros que hicieron los respectivos aportes para la producción de los audiovisuales. La decisión final fue buscar una presentadora externa y la escogida fue Laura Pérez Beracasa. Comunicadora Social de la Universidad Autónoma de Occidente, quien tiene reconocimiento como conductora del canal regional Telepacífico.

Después, el trabajo radicó en apropiarse de los conceptos utilizados en GraPiCO y PiCO. Se estableció la diferencia entre ambos, además, del proceso explicativo, en el cual no se cuidaron los detalles sobre la usabilidad de ambos lenguajes de programación. Con este paso se finalizó la investigación y la explicación sobre el funcionamiento de ambos lenguajes de programación, donde los comunicadores quedaron con total claridad y prestos a la preproducción de los piezas audiovisuales.

En la producción de los videos fue necesario contar con un estudio de grabación, cámaras, camarógrafo, luces, luminotécnico, set de grabación, personajes extras, presentadora, libretos y guiones.

La postproducción se hizo en dos pequeñas fases: la preedición de los videos obtenidos, en la que se seleccionaron las imágenes que quedarían en los audiovisuales para ser entregados a los ingenieros, y la edición del material escogido. Se realizó el montaje de dicho material pertinente para la entrega: sonido, musicalización, imagen, logos, créditos, y se procedió a la entrega del material a los ingenieros, quienes dieron sus opiniones para hacer los cambios sugeridos. En esta etapa también se realizó la corrección del material audiovisual a partir de las observaciones, para nuevamente entregar el material debidamente terminado.

Haciendo uso de la compenetración con la ingeniería, los comunicadores decidieron llevar a cabo un proceso de *ingeniería comunicacional*, la cual se distinguió por ser elaborada con precisión con los detalles que los ingenieros quisieron cuidar y los comunicadores, supieron realizar.

B. Etapa 2: Comunicando sapiencias

La nueva etapa fue la realización de los talleres de comunicación para los ingenieros, quienes tomaron en cuenta diferentes aspectos referentes a la expresión corporal, dicción y manejo del público, como herramientas que les permitieran comunicar su mensaje con mayor efectividad.

Los ingenieros se basaron en un guion para su presentación en público, fue la recomendación porque en caso de olvido no se perdería el hilo conductor y tendrían pautas de la continuación de los tópicos a tratar durante la presentación. Por esa razón, se decidió no optar por el uso de un libreto.

Se corrigieron problemas en dicción y articulación, se dieron técnicas para el control de nervios, la postura corporal, gesticulación y demás. Después de estas sugerencias, los ingenieros se prepararon para la demostración y, posteriormente, la crítica constructiva para mejorar. Por último y después de un ensayo exhaustivo y riguroso, los comunicadores dieron el visto bueno a los ingenieros para realizar su intervención en la presentación.

Con el material audiovisual ya terminado y los talleres realizados para la exposición de los ingenieros, el grupo de comunicadores se dio a la tarea de realizar la recopilación de la información, de manera tal que pudiera sistematizarse incluyendo errores y aciertos; además de los resultados y aporte de la comunicación en el proyecto posdoctoral que dieron cuenta del proceso.

VII. Conclusiones

Sistematizar una experiencia es un proceso de aprendizaje complejo, pero que permite un crecimiento profesional porque ayuda a plasmar y analizar el paso a paso de un proyecto para no caer en errores cometidos, a dar cuenta de cada fase y a evaluar el desarrollo de ese estudio.

Esta sistematización permitió al grupo de comunicación plasmar el paso a paso del proyecto con errores, falencias y aciertos, de mane-

ra tal que aportó una experiencia de la cual puede alimentarse otro trabajo interdisciplinar.

El modelo TCACI sirvió para evidenciar el proceso realizado en el trabajo interdisciplinar y quedó como un aporte de la Comunicación, que permite sistematizar experiencias diferentes a la propuesta, lo que lo convierte en un modelo estándar.

VIII. Bibliografía

- [1] Carlos Andrés Tavera Romero, Christian Felipe Cano Castillo, and Luis Eduardo Espinosa Gallady. (2012) Biblioteca Digital de la Universidad de San Buenaventura Cali. [Online]. <http://biblioteca-digital.usb.edu.co/handle/10819/1338>
- [2] Camilo Rueda et al., Integrating Constraints and Concurrent Objects in Musical Applications: A Calculus and its Visual Language, Constraints, vol. 6, no. 1, pp. 21-52, january 2001.
- [3] Carlos Andrés Tavera Romero and Juan Francisco Díaz Frías, Nuevo cálculo visual GraPiCO: Presentación de sus características fundamentales, en Memorias del 2do Congreso Colombiano de Computación, Bogota, 2007.
- [4] Carlos Andrés Tavera Romero and Juan Francisco Díaz Frías, Breve Discusión de las Ventajas de los Lenguajes Visuales frente a los Textuales: Caso de Estudio el Cálculo GraPiCO, in III Congreso Colombiano de Computación, Medellín, 2008.
- [5] Carlos Andrés Tavera Romero, Juan Francisco Díaz Frías, Aybert Soto Vargas, Juan Javier Gallego Varona, and Anderson Jojoa, Alternativa de comprobación sintáctica de VLP: Gsig Parsing. Aspectos formales y el caso de estudio: E_GraPiCO, en Anales del XIII Congreso Argentino de Ciencias de la Computación, Guillermo Ricardo Simari and Armando Eduardo De Giusti, Eds. Argentina: RedUNCI, 2007, pp. 1632 - 1644.
- [6] Arizando Carvajal Burbano, Teoría y Práctica de la Sistematización de Experiencias, 4th ed. Cali, Valle, Colombia: Universidad del Valle, Escuela de Trabajo Social y Desarrollo Humano, 2010.
- [7] Oscar Jara Holliday, Orientaciones Teórico-Prácticas para la Sistematización de Experiencias. Costa Rica: Centro de Estudios y Publicaciones Alforja.
- [8] Juan Carlos Asinsten, Comunicación visual y tecnología de gráficos en computadora: educ.ar y Ministerio de Educación Ciencia y Tecnología, 2017.

- [9] José Miguel Pereira González, La comunicación, un campo de conocimiento en construcción. Reflexiones sobre la Comunicación Social en Colombia, Investigación & Desarrollo, vol. 13, no. 2, pp. 412-441, 2005.

PASOS EN LA REALIZACIÓN DE LOS AUDIOVISUALES PEDAGÓGICOS: PiCO – GraPiCO Y EJERCICIO DE MODELACIÓN

*Paola Andrea Ramírez Arcila
Juan David Penagos Muñoz
Yenny Viviana Cruz Pérez
Carlos Andrés Tavera Romero*

I. Introducción

Este es un proyecto interdisciplinario e interinstitucional entre la Ingeniería de Sistemas y la Comunicación Social, la Universidad de San Buenaventura de Cali USBC (por parte de los Ingenieros) y la Universidad Autónoma de Occidente UAO (por parte de los Comunicadores); el proyecto en cuestión se gestó con el fin de dar apoyo al Laboratorio de Investigación para el Desarrollo de la Ingeniería de Software (LIDIS), de la USBC, en un trabajo posdoctoral consistente en el estudio comparativo entre el Cálculo textual: PiCO y el Cálculo visual: GraPiCO, los cuales hacen parte del programa E_GraPiCO. Este apoyo fue necesario para conocer bajo qué condiciones es más adecuado un lenguaje que otro y qué mejoras requieren. El objetivo desde la comunicación durante el proceso fue socializar los conceptos técnicos básicos propios de estos dos lenguajes de programación (PiCO Y GraPiCO) por medio de material comunicativo audiovisual que, a su vez, serviría como una herramienta didáctica mediadora entre los exponentes del proyecto y el auditorio.

Realizar productos audiovisuales tiene un alto grado de complejidad, y si a esto se le añade que hace parte de un proyecto post doctoral, su elaboración se dificulta aún más, sin embargo, el grupo de comunicadores decidió apoyar este proyecto y se embarcó en la realización de la pieza fundamental que hizo parte del proceso de investigación de los ingenieros.

En este proyecto se trabajaron diferentes pasos que articularon las diferentes etapas experimentadas por todo el equipo de investigadores. A continuación se especifica la etapa a desarrollar en el presente capítulo. Ver Tabla 25.

Tabla 25. Etapas del estudio comparativo

Estudio comparativo entre lenguajes textuales y lenguajes visuales. Caso: PiCO y GraPiCO	
Etapa 1	Elaboración de hipótesis en experimentos de lenguajes de programación.
Etapa 2	Variables en un experimento de lenguajes de programación.
Etapa 3	Unidades experimentales utilizadas en pruebas de lenguajes de programación.
Etapa 4	Tratamientos y replicas en un experimento de programación.
Etapa 5	Aprendizaje Basado en Problemas (ABP) aplicado a los lenguajes de programación.
Etapa 6	La comunicación en el estudio comparativo entre lenguajes textuales y lenguajes visuales: Caso PiCO y GraPiCO.
Etapa 7	Sistematización de una experiencia de investigación entre la Comunicación Social y la Ingeniería de Software.
Etapa 8	Modelo de sistematización propuesto “TCACI en doble vía”.
Etapa 9	Pasos en la realización de los audiovisuales pedagógicos: PiCO – GraPiCO y ejercicio de modelación.

Etapa 10	Estudio de resultados de PiCO y GraPiCO. Parte 1.
Etapa 11	Estudio de resultados de PiCO y GraPiCO. Parte 2.
Etapa 12	Recomendaciones y resultados del estudio entre PiCO, GraPiCO y editores.

Fuente: elaboración propia (2018).

Por otra parte, el grupo de comunicadores inmerso en este proyecto decidió trabajar desde la comunicación con un modelo de Juan Carlos Asinsten, especialista argentino en diseño y comunicación. Se tomó esta decisión para realizar el proceso de producción de los audiovisuales, ya que el modelo de comunicación del especialista antes mencionado sirvió de apoyo y de orientación en gran manera a la hora de realizar los productos comunicativos.

Durante la realización de los productos audiovisuales se llevaron a cabo varios pasos:

1. Investigación metodológica.
2. Etapa de guionización de la información analizada.
3. Realización de Prototipo Audiovisual. Preproducción, producción, posproducción
4. Presentación del video-prototipo a ingenieros.
5. Realización de los videos calculo textual y video calculo gráfico.
6. Retroalimentación por parte de los ingenieros al grupo de comunicadores para las correcciones de los audiovisuales.
7. Corrección de audiovisuales: preproducción, producción, posproducción.
8. Realización de un taller práctico para ayudar a los ingenieros en su presentación ante público.
9. Jornada de presentación de los videos y recolección de datos.
10. Etapa de análisis de los datos recolectados de las encuestas, relacionados con el componente de comunicación.
11. Socialización de resultados.

II. Investigación metodológica

En esta parte del proyecto los comunicadores recibieron la explicación de los conceptos de los cálculos de lenguajes de programación GraPiCO y PiCO por parte del grupo de investigación LIDIS, llevando así un proceso de adhesión al funcionamiento de cada uno de los conceptos y estableciendo la diferencia entre ambos cálculos. Con la explicación y apropiación de los conceptos, esta etapa se finalizó contando con los conocimientos necesarios para la elaboración de los audiovisuales.

III. Guionización

Para realizar estos guiones; uno visual (GraPiCO) y otro textual (PiCO) los comunicadores tuvieron en cuenta la importancia de cada concepto dentro de su lenguaje de programación para establecer el orden de aparición y la continuidad; también contaron con el aporte de los ingenieros, quienes ratificaron la estructura establecida; además, constantes reuniones de ambos grupos sirvieron para concretar una estructura clave en la organización de la información suministrada en la etapa de investigación.

Trasladar la información investigada a un guion, fue un factor fundamental en esta parte del proceso donde los ingenieros fueron participes claves en la realización de los guiones que sirvieron para llevar a cabo los audiovisuales.

IV. Realización del Prototipo Audiovisual

En esta etapa se realizaron pruebas para definir la puesta en escena y la composición visual de los videos, donde la realización del guion de manera previa fue fundamental para poder dar inicio a la producción de los audiovisuales.

La primera prueba se realizó en el estudio de televisión de la Universidad San Buenaventura-Cali, allí los presentadores fueron las dos mujeres del grupo de comunicación que contaban con la ven-

taja de conocer el proyecto y el funcionamiento de los lenguajes de programación que se iban a exponer; este video piloto sirvió como para definir si eran las comunicadoras sociales las personas indicadas para conducir los audiovisuales.

Durante el proceso de la etapa de los audiovisuales se vivieron tres fases, las cuales fueron; Pre–pro–posproducción; estas se explican a continuación:

S. **Pre-Producción**

Se apropiaron los conceptos técnicos de los lenguajes de programación y posteriormente se realizó el guión.

T. **Producción**

En primera instancia se utilizó el estudio de la Universidad San Buenaventura Cali y sus equipos, además de contar con el guion que se realizó de manera previa.

Las presentadoras del prototipo fueron integrantes del grupo de comunicación debido a su cercanía con el medio y el conocimiento sobre el proyecto que a su vez, daría mayor claridad y seguridad de la información que se transmitiría al auditorio. Por otro lado, se realizó la composición visual basada en el video clip La historia de las cosas, un modelo interactivo y dinámico para explicar un tema tan extenso como éste; sin embargo, resultó muy complejo para las presentadoras ya que no dominaban a plenitud la técnica de presentación.

U. **Post-Producción**

Luego de la grabación de este audiovisual se decidió de manera unánime entre los ingenieros y los comunicadores, repetir la grabación, ya que las presentadoras que sirvieron para hacer el piloto no dominaban a plenitud la práctica de la presentación; además, con el prototipo realizado se conocieron las falencias que tenía el desarrollo de los productos audiovisuales, elementos que se debían corregir.

V. Presentación del video prototipo a ingenieros

Debido a las equivocaciones durante los días de grabación por parte de las presentadoras no se pudo realizar la exposición de los audiovisuales y ante esto, ambas partes llegaron a la conclusión de contratar una presentadora con experiencia. También se decidió utilizar el estudio de televisión de la Universidad Autónoma de Occidente y sus equipos, mejorando la producción de los audiovisuales, pues las instalaciones con las que se contaba en ese momento, no eran las adecuadas.

VI. Realización de los videos Cálculo textual y Cálculo visual

Luego de las correcciones realizadas entre los comunicadores e ingenieros sobre el prototipo audiovisual, se decidió contratar a la presentadora Laura Pérez, Comunicadora Social de la Universidad Autónoma y presentadora oficial de un programa juvenil en el canal regional Telepacífico. Debido a que se necesitaban equipos lo suficientemente profesionales para la realización de los videos, se decidió trabajar con los equipos de la Universidad Autónoma, institución líder en la región por su alto desempeño en tecnología audiovisual y comunicación; además, se utilizaron los estudios de edición de la UAO para garantizar la calidad en estos productos audiovisuales que servirían más adelante como herramientas claves en un proyecto post-doctoral.

Fueron dos jornadas de grabación que contaron con la presencia del Director del grupo LIDIS, Carlos Andrés Tavera; él fue el veedor de la Universidad San Buenaventura e hizo presencia para que se cumpliera a cabalidad todo lo propuesto en un principio.

Durante la mañana del primer día de grabación se explicó el proyecto a la presentadora y la funcionalidad de cada concepto, luego, en la tarde, se inició la grabación del Cálculo visual GraPiCO y al día siguiente el Cálculo textual PiCO.

Cuando los audiovisuales se terminaron, se realizó la edición de los dos productos basados en un guión de montaje elaborado de manera previa por el grupo de comunicadores, guión que estableció el orden de aparición de cada concepto en cada audiovisual con sus respectivos sonidos, imágenes y animaciones; animaciones que fueron suministradas por los ingenieros para culminación de ambos productos.

VII. Retroalimentación

Después de haber realizado los audiovisuales requeridos por el grupo de ingenieros, y de haber editado el material, se concertó una reunión con estos, para conocer cuáles eran los comentarios, apreciaciones, aportes o críticas que tenían de los productos, si servían, si no servían, qué cambios deseaban hacer y demás.

Dicha reunión arrojó como resultado diferentes consideraciones. Los comentarios en general fueron animadores para los comunicadores, pero se debían corregir puntos específicos para el producto final, como mejorar la sonorización de algunos conceptos que se estaban dando a conocer por medio de los videos, por ejemplo Clone. Se consideró que era importante insertarle una sonorización mucho más llamativa, pero a la vez agradable y coherente con lo que el video trataba de exponer, ya que Clone significaba una funcionalidad importante en el Cálculo PiCO (textual).

Como segunda apreciación por parte de los ingenieros, se solicitó que en la explicación de cada concepto apareciera la presentadora escogida, Laura Pérez, ya que ella aportaba el dinamismo que se buscaba para los videos, teniendo en cuenta que se estaban manejando conceptos informáticos sumamente pesados en cuanto a interiorización intelectual se refiere; además, ella lograba captar de sobremanera la atención de los receptores de los audiovisuales o productos elaborados.

Luego de debatir respecto a lo que querían los ingenieros, el trabajo se trasladó a una de las salas de edición audiovisual de la Universidad Autónoma de Occidente, UAO, donde se corrigieron los puntos sugeridos. Cuando todo estuvo listo y después de una ardua

y nueva revisión de los videos, el material fue entregado para una nueva observación de los ingenieros, en la cual, los comunicadores recibieron el aval de su trabajo.

Por último, los miembros del grupo LIDIS manifestaron que los comunicadores lograron cumplir con las expectativas y retos que se les habían trazado. Retos y expectativas que consiguieron su fin principal, aportar al estudio comparativo de los cálculos de lenguajes de Programación PiCO y GraPiCO.

VIII. Conclusiones

La comunicación se articuló con la ingeniería en sistemas y apoyó su proceso de investigación, aportando no solamente como herramienta meramente técnica sino además en la construcción del discurso, es decir en su fondo y forma. Este componente comunicacional jugó un papel indispensable en la medida en que sirvió de puente entre unos exponentes y el auditorio.

La realización de audiovisuales fue una gran experiencia para los comunicadores ya que les permitió elaborar un discurso visual a un público totalmente ajeno a su campo de estudio, siendo ésta una ardua tarea en estos dos productos el entendimiento de cada uno de los conceptos para posteriormente poder plasmar ese conocimiento en un resultado agradable, dinámico y sobre todo de enseñanza.

El acompañamiento por parte de los ingenieros en sistemas fue un apoyo fundamental en la apropiación del manejo y definición de cada uno de los conceptos ya que permitieron elaborar un contenido sencillo y contundente, en esta tarea para que los estudiantes de ingeniería entendieran la funcionalidad de los dos lenguajes de programación.

IX. Bibliografía

- [1] Carlos Andrés Tavera Romero, Christian Felipe Cano Castillo, y Luis Eduardo Espinosa Galliady. (2012) Biblioteca Digital de la Universidad de San Buenaventura Cali. [Online]. <http://bibliotecadigital.usb.edu.co/handle/10819/1338>
- [2] Camilo Rueda et al., Integrating Constraints and Concurrent Objects in Musical Applications: A Calculus and its Visual Language, Constraints, vol. 6, no. 1, pp. 21-52, january 2001.
- [3] Carlos Andrés Tavera Romero y Juan Francisco Díaz Frías, Nuevo cálculo visual GraPiCO: Presentación de sus características fundamentales, en Memorias del 2do Congreso Colombiano de Computación, Bogota, 2007.
- [4] Carlos Andrés Tavera Romero y Juan Francisco Díaz Frías, Breve Discusión de las Ventajas de los Lenguajes Visuales frente a los Textuales: Caso de Estudio el Cálculo GraPiCO, en III Congreso Colombiano de Computación, Medellín, 2008.
- [5] Carlos Andrés Tavera Romero, Juan Francisco Díaz Frías, Aybert Soto Vargas, Juan Javier Gallego Varona, y Anderson Jojoa, Alternativa de comprobación sintáctica de VLP: Gsig Parsing. Aspectos formales y el caso de estudio: E_GraPiCO, en Anales del XIII Congreso Argentino de Ciencias de la Computación, Guillermo Ricardo Simari y Armando Eduardo De Giusti, Eds. Argentina: RedUNCI, 2007, pp. 1632 - 1644.
- [6] Juan Carlos Asinsten, Comunicación visual y tecnología de gráficos en computadora: educ.ar y Ministerio de Educación Ciencia y Tecnología, 2017.

ESTUDIO DE RESULTADOS DE PiCO y GraPiCO

PARTE 1

*Luis Eduardo Espinosa Galliady
Marco Antonio Triana Lozano
Carlos Andrés Tavera Romero*

I. Introducción

Luego de realizar los experimentos y las encuestas, se requiere analizar los datos desde el punto de vista estadístico para poder sacar las conclusiones.

Para cumplir con ello, se realizará la medición de los niveles de asimilación, comprensión y aceptación de estos dos lenguajes de programación por parte de una población escogida específicamente de acuerdo a ciertos parámetros.

Este documento presenta parte del análisis de los resultados estadísticos obtenidos en el proyecto.

II. Marco Teórico

En etapas previas se explicó la *hipótesis* y su verificación, las *variables* empleadas, los *sujetos de estudio* (y el manejo dado a las

mismas), las diferentes asignaciones hechas a las *unidades experimentales* mediante el uso de *tratamientos* y las *observaciones* efectuadas sobre los mismos; también se presentó la contribución del *Aprendizaje Basado en Proyectos*, el diseño y elaboración de las encuestas empleadas para la captura de datos a analizar y el apoyo ofrecido por la *Comunicación Social*, para la elaboración de todo el material concerniente a la interacción de la población escogida con el experimento en cuestión; todo esto ha presentado la evolución del proceso de ejecución de este estudio; sin embargo, aún queda faltando presentar el análisis correspondiente a los resultados obtenidos mediante la presentación de cada uno de estos dos lenguajes de programación.

En la presentación de los lenguajes de programación, se generó una sesión de discusión y aclaración de dudas acerca de los mismos; posteriormente, se realizó un taller para modelar una situación cotidiana; proceso que arrojó información muy importante para el estudio a través de encuestas; siendo todo esto, el tema a tratar en este documento. ^[5] Ver Tabla 26.

Tabla 26. *Etapas del estudio comparativo.*

Estudio comparativo entre lenguajes textuales y lenguajes visuales. Caso: PiCO y GraPiCO	
Etapa 1	Elaboración de hipótesis en experimentos de lenguajes de programación.
Etapa 2	Variables en un experimento de lenguajes de programación.
Etapa 3	Unidades experimentales utilizadas en pruebas de lenguajes de programación.
Etapa 4	Tratamientos y replicas en un experimento de programación.
Etapa 5	Aprendizaje Basado en Problemas (ABP) aplicado a los lenguajes de programación.
Etapa 6	La comunicación en el estudio comparativo entre lenguajes textuales y lenguajes visuales: Caso PiCO y GraPiCO.

Etapa 7	Sistematización de una experiencia de investigación entre la Comunicación Social y la Ingeniería de Software.
Etapa 8	Modelo de sistematización propuesto “TCACI en doble vía”.
Etapa 9	Pasos en la realización de los audiovisuales pedagógicos: PiCO – GraPiCO y ejercicio de modelación.
Etapa 10	Estudio de resultados de PiCO y GraPiCO. Parte 1.
Etapa 11	Estudio de resultados de PiCO y GraPiCO. Parte 2.
Etapa 12	Recomendaciones y resultados del estudio entre PiCO, GraPiCO y editores.

Fuente: elaboración propia (2018).

Las encuestas o cuestionarios utilizados para la recolección de datos en este experimento, fueron diseñados según los lineamientos que serán presentados a continuación.^[5]

Los cuestionarios son muy utilizados en el ámbito de la investigación, puesto que generan escalas e índices para facilitar la medición (ya que estandariza la información recogida por el cuestionario) a costos relativamente bajos llegando a captar información de una mayor cantidad de participantes y permitiendo el análisis de absolutamente toda la información, de una forma muy práctica y fácil.^[5]

En el momento de elaborar un cuestionario, se debe tener absolutamente claro todo el conocimiento acerca de lo que se va a medir, al igual que ciertos conocimientos estadísticos muy útiles.^[5]

Un buen cuestionario, se caracteriza por:

1. Ser adecuado para el problema a medir y tener un contenido válido (lo cual implica que permita medir lo que se pretende).^[5]
2. Ser fiable, es decir tener la mínima cantidad posible de errores en la medida a realizar.^[5]
3. Ser flexible a los cambios de los participantes encuestados.^[5]
4. Tener sus componentes lo más delimitados posible.^[5]

Para obtener un cuestionario adecuado, fiable, y flexible, se deben seguir los siguientes pasos:

1. Definir aquello que se desea medir.^[5]
2. Establecer el contenido del cuestionario, teniendo en cuenta aspectos tales como la población a la que va dirigido, la forma de distribución y el formato del mismo.^[5]
3. La correcta composición de los ítems (unidad básica del mismo) del cuestionario.^[5]
4. Cantidad de ítems.^[5]
5. Definir el contenido.^[5]
6. Orden adecuado de los ítems.^[5]
7. Contar con preguntas muy bien definidas y estructuradas, lo cual implica tener en cuenta el lenguaje y la cultura del encuestado; por lo tanto, se recomienda que éstas sean cortas y fáciles de entender.^[5]

III. Modelamiento

Las encuestas a los estudiantes se hicieron, utilizando la herramienta *LimeSurvey*. *LimeSurvey* es una aplicación *opensource* para la realización de encuestas en línea, está desarrollado en PHP y brinda la posibilidad a los usuarios aunque no tengan conocimientos de programación, de facilitar el desarrollo, la publicación y recolección de respuestas en las encuestas; incluye plantillas web y utilidades básicas para analizar estadísticamente los resultados obtenidos. La aplicación permite exportar los datos en formato de Excel.

Para analizar, gestionar y procesar los datos recopilados de las encuestas, se utilizó la herramienta *Statgraphics Centurion* para Windows; permite la representación gráfica de todo tipo de estadísticas y el desarrollo de *experimentos* y simulaciones en función del comportamiento de los valores, de manera sencilla.

Stagraphics cuenta con cuatro módulos que lo caracterizan:

- StatReport: Editor estadístico que permite generar informes con datos variables.
- StatWizard: Asistente estadístico que nos permite escoger los métodos más convenientes para analizar los datos.
- StatFolio: Libro de análisis, con características similares a Excel (columnas, filas y hojas).
- StatLink: Permite enlazar estadísticamente los libros de análisis con los datos fuentes y un enlace estadístico.

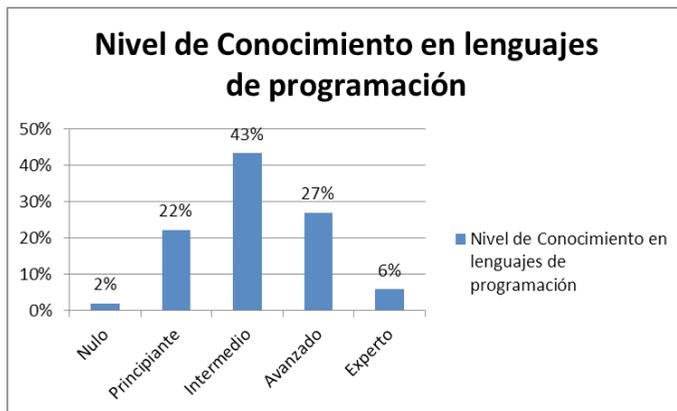
Con estas dos herramientas se procesaron los datos obtenidos de las encuestas y se realizó el estudio pertinente de acuerdo al caso PiCO y GraPiCO.

IV. Resultados

Con el fin siempre de conservar el experimento dentro de la región de aceptación y poder controlar factores que pudieran poner en riesgo el estudio, las encuestas realizadas tanto para PiCO como para GraPiCO, a estudiantes de ingeniería en sistemas que cursan entre segundo y quinto semestre académico, con edades entre 16 y 23 años, de las universidades de San Buenaventura, Autónoma de Occidente, Javeriana y el ICESI de la ciudad de Cali; arrojaron los siguientes resultados presentados y explicados en el transcurso de este escrito.

De la información obtenida, se observó que un mayor porcentaje (70%) de estudiantes encuestados tienen buen conocimiento en lenguajes de programación, mientras que el 6% de estos, opinan que tienen mucho conocimiento de los lenguajes de programación. Además, tan sólo el 2% de estos estudiantes considera no tener conocimiento alguno de los lenguajes de programación; La Figura 12 muestra un gráfico de barras ejemplificando esta situación.

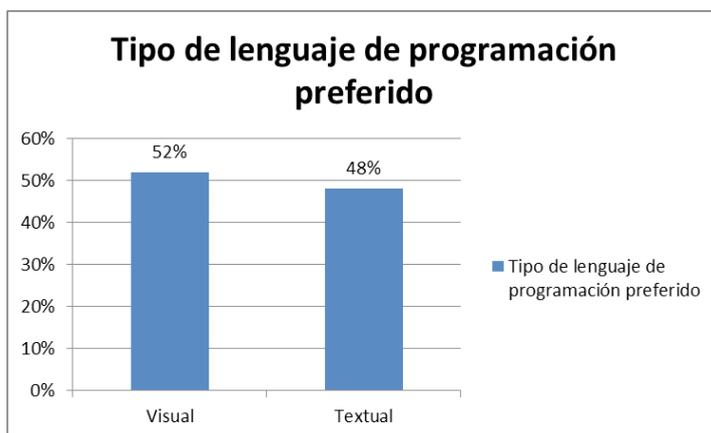
Figura 12. Nivel de conocimiento en lenguajes de programación de encuestados tanto para PiCO como para GraPiCO.



Fuente: elaboración propia (2018).

En la Figura 13 se muestra un gráfico de barras, en el cual se observa que un mayor porcentaje (52%) de estudiantes encuestados prefiere lenguajes de programación visuales con respecto a los textuales.

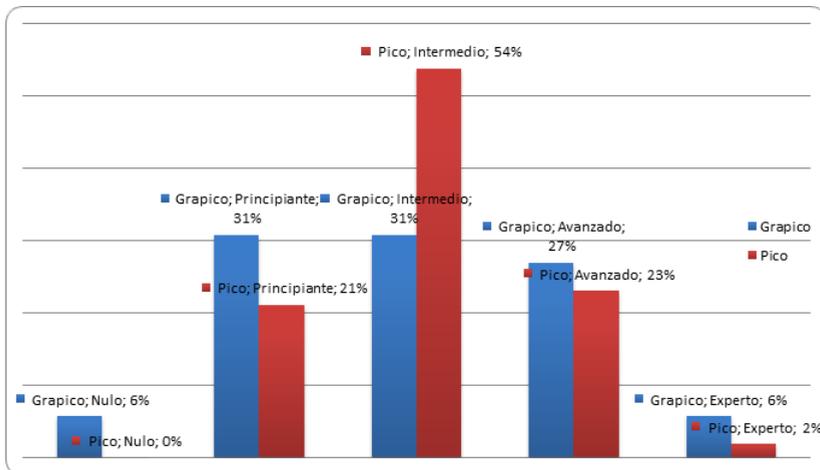
Figura 13. Tipo de lenguaje de programación preferido de encuestados tanto para PiCO como para GraPiCO.



Fuente: elaboración propia (2018).

Comparando el nivel de conocimiento entre lenguajes de programación visuales y textuales, se encontró que un mayor porcentaje (75%) de estudiantes encuestados tiene algo de conocimiento en lenguaje textual con respecto al lenguaje visual (62%), mientras que, es mayor el porcentaje (33%) de estudiantes encuestados que tienen mucho conocimiento en lenguaje visual con respecto al lenguaje textual (25%), fenómeno que se puede evidenciar en la Figura 14.

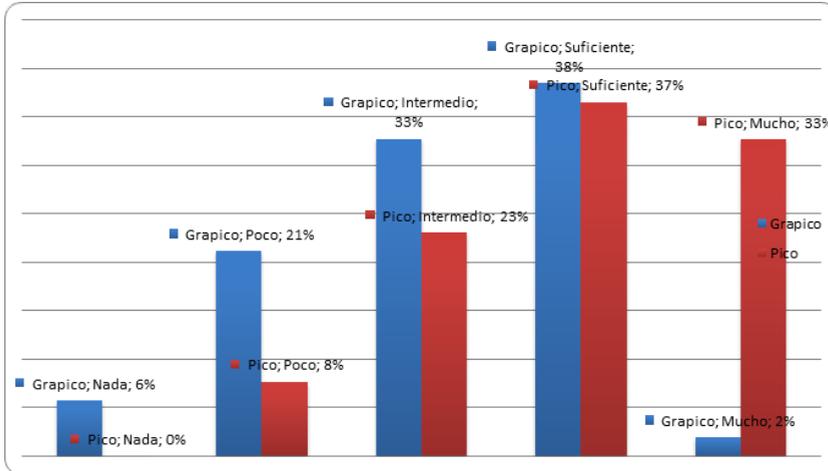
Figura 14. Nivel de conocimiento en lenguajes de programación visual y textual diferenciado por PiCO y GraPiCO.



Fuente: elaboración propia (2018).

Según la Figura 15, comparando la frecuencia de uso de aplicaciones que de alguna manera utilizan lenguajes de programación visuales y textuales, se concluyó que un mayor porcentaje (54%) de estudiantes encuestados utiliza con alguna frecuencia el lenguaje visual con respecto al lenguaje textual (31%), mientras que, es mayor el porcentaje (70%) de estudiantes encuestados que utiliza con más frecuencia el lenguaje textual con respecto al lenguaje visual (40%).

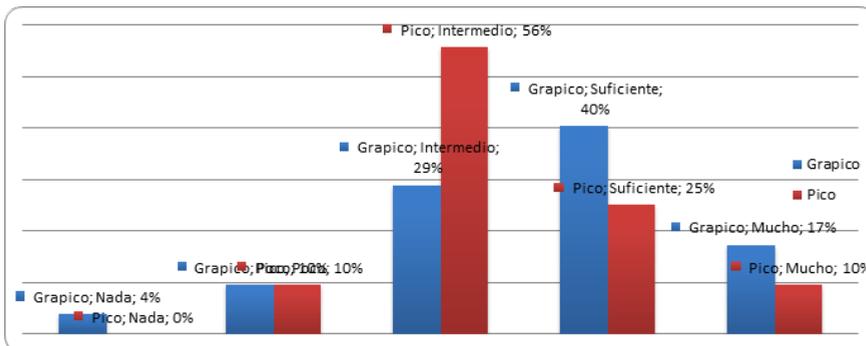
Figura 15. Frecuencia de uso de lenguajes de programación visual y textual diferenciado por PiCO y GraPiCO.



Fuente: elaboración propia (2018).

Visualizando la Figura 16, se puede hallar que un mayor porcentaje (66%) de estudiantes encuestados, tiene algo de interés en lenguajes textuales con respecto a los lenguajes visuales (39%), mientras que, se observa que un mayor porcentaje (57%) de los estudiantes encuestados está muy interesado en lenguajes visuales con respecto a los lenguajes textuales (35%).

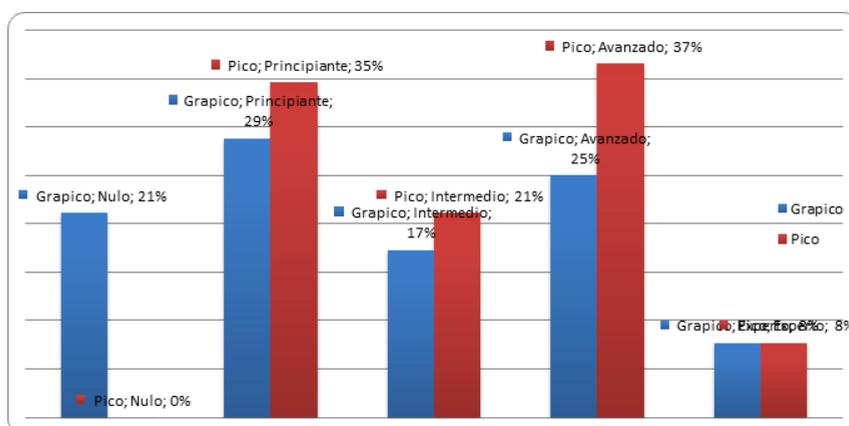
Figura 16. Interés en lenguajes de programación visual y textual diferenciado por PiCO y GraPiCO.



Fuente: elaboración propia (2018).

De acuerdo a la Figura 17, se puede visualizar claramente que un mayor porcentaje (56%) de los estudiantes encuestados conocen y comprenden algo acerca del manejo del *constructor* “program” en lenguaje textual PiCO con respecto al lenguaje visual GraPiCO (46%); también; es mayor el porcentaje (45%) de estudiantes que conoce, comprende y utiliza muy bien el *constructor* “program” en lenguaje textual PiCO con respecto al lenguaje visual GraPiCO (33%).

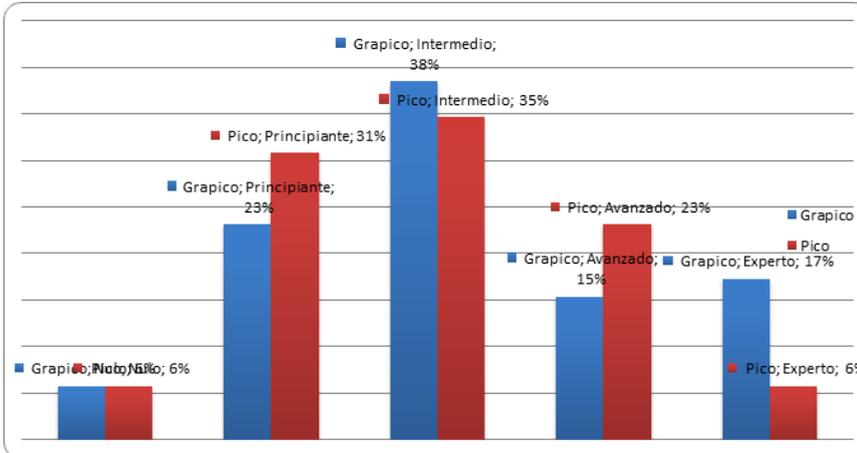
Figura 17. Nivel de comprensión de *constructor* “Program” diferenciado por PiCO y GraPiCO.



Fuente: elaboración propia 2018.

Con la Figura 18, comparando el nivel de comprensión del *constructor* “context” entre lenguajes de programación visuales y textuales, se puede analizar fácilmente que un mayor porcentaje (66%) de estudiantes encuestados conoce y comprende algo acerca del manejo del *constructor* “context” en lenguaje textual PiCO con respecto al lenguaje visual GraPiCO (61%), mientras que, es mayor el porcentaje (32%) de estudiantes encuestados que conoce, comprende y utiliza muy bien el *constructor* “context” en lenguaje visual GraPiCO con respecto al lenguaje textual PiCO (29%).

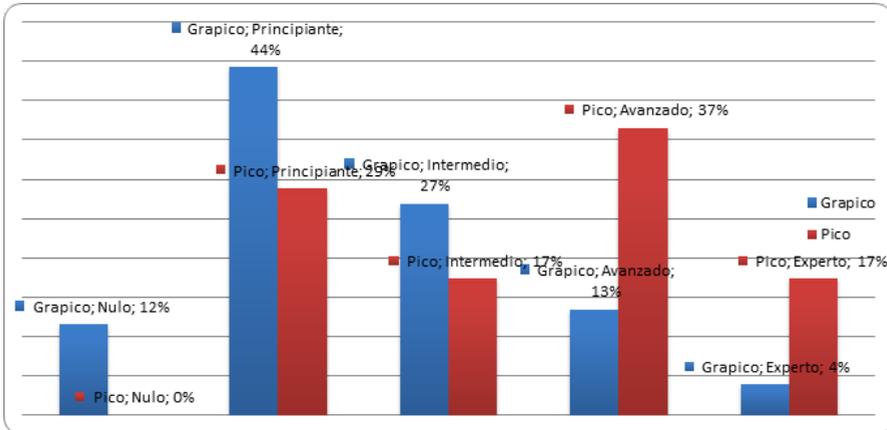
Figura 18. Nivel de comprensión de constructor “Context” diferenciado por PiCO y GraPiCO.



Fuente: elaboración propia (2018).

Comparando el nivel de comprensión del *constructor* “Objects” entre lenguajes visuales y textuales de programación, se observa en la Figura 19, que un mayor porcentaje (71%) de estudiantes encuestados conoce y comprende algo acerca del manejo del *constructor* “Objects” en lenguaje visual GraPiCO, con respecto al lenguaje textual PiCO (46%), mientras que se observa que un mayor porcentaje (54%) de estudiantes encuestados conoce, comprende y utiliza muy bien el constructor “Objects” en lenguaje **textual PiCO con respecto al lenguaje visual GraPiCO (17%)**.

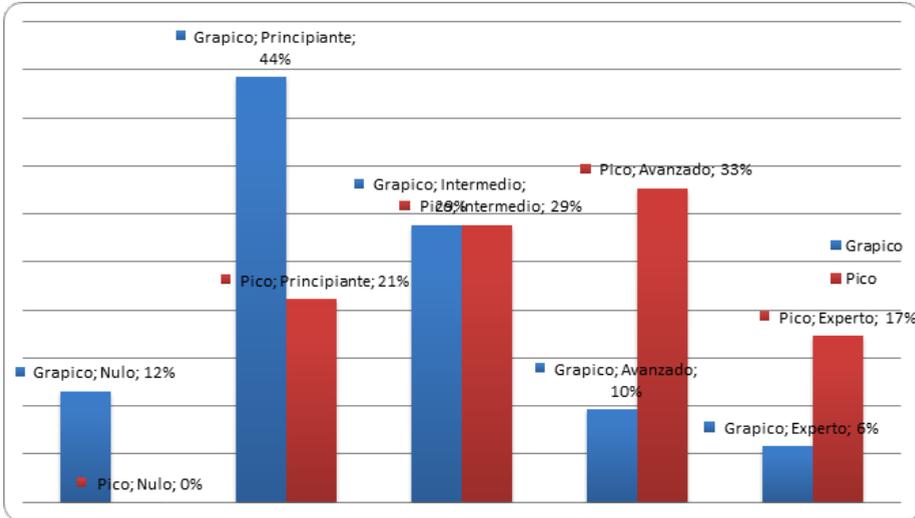
Figura 19. Nivel de comprensión de constructor “Objects” diferenciado por PiCO y GraPiCO.



Fuente: elaboración propia (2018).

Analizando el nivel de comprensión del *constructor* “Methods” entre lenguajes visuales y textuales de programación y usando la Figura 20, se puede concluir que un mayor porcentaje (73%) de estudiantes encuestados conoce y comprende algo acerca del manejo del *constructor* “Methods” en lenguaje visual GraPiCO con respecto al lenguaje textual PiCO (50%), mientras que se observa que un mayor porcentaje (50%) de los estudiantes encuestados conoce, comprende y utiliza muy bien el *constructor* “Methods” en lenguaje textual PiCO con respecto al lenguaje visual GraPiCO (16%).

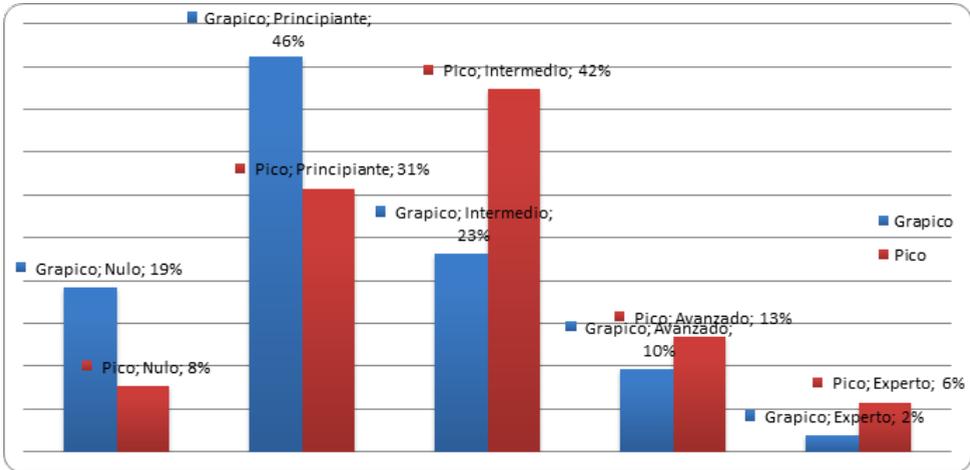
Figura 20. Nivel de comprensión de constructor “Methods” diferenciado por PiCO y GraPiCO.



Fuente: elaboración propia (2018).

Cotejando el nivel de comprensión del *constructor* “Ask” entre lenguajes visuales y textuales de programación, mediante la Figura 21, se observó que un mayor porcentaje (73%) de estudiantes encuestados conoce y comprende algo acerca del manejo del constructor “Ask” en lenguaje textual PiCO con respecto al lenguaje visual GraPiCO (69%), también; es mayor el porcentaje (19%) de estudiantes encuestados que conoce, comprende y utiliza muy bien el *constructor* “Ask” en lenguaje textual PiCO con respecto al lenguaje visual GraPiCO (12%).

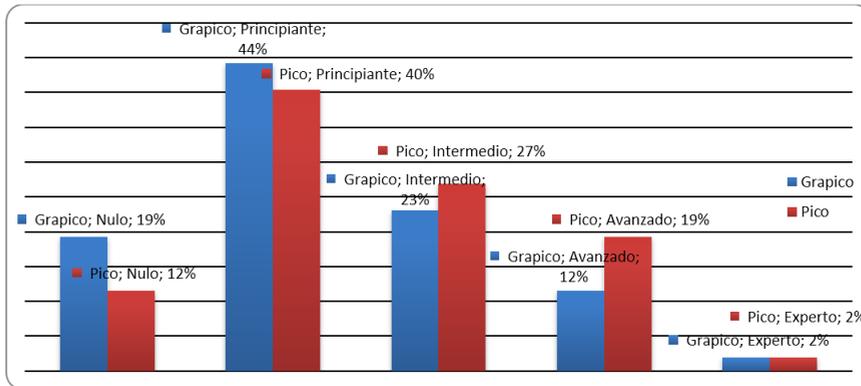
Figura 21. Nivel de comprensión de constructor “Ask” diferenciado por PiCO y GraPiCO.



Fuente: elaboración propia (2018).

Confrontando el nivel de comprensión del *constructor* “Tell” entre lenguajes visuales y textuales de programación, se puede mediante la Figura 22 observar que igual proporción (67%) de estudiantes encuestados conoce y comprende algo acerca del manejo del *constructor* “Tell” en lenguaje textual PiCO y lenguaje visual GraPiCO, mientras que, se observa un mayor porcentaje (21%) de estudiantes encuestados que conoce, comprende y utiliza muy bien el *constructor* “Tell” en lenguaje textual PiCO con respecto al lenguaje visual GraPiCO (14%).

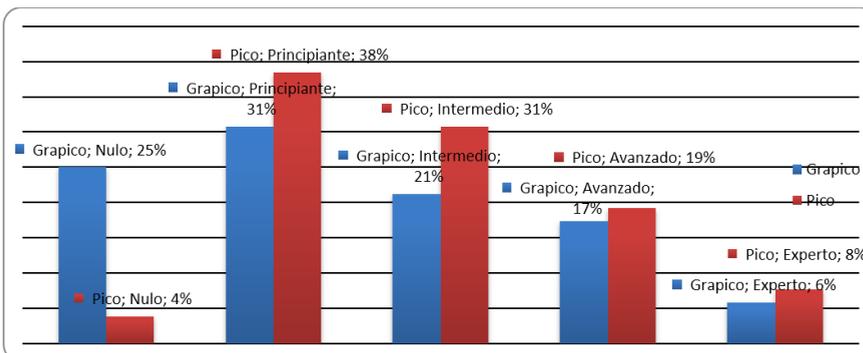
Figura 22. Nivel de comprensión de constructor “Tell” diferenciado por PiCO y GraPiCO.



Fuente: elaboración propia (2018).

Comparando el nivel de comprensión del *constructor* “MsgSend” entre lenguajes visuales y textuales de programación, se observó con la ayuda de la Figura 23, que un mayor porcentaje (69%) de estudiantes encuestados conoce y comprende algo acerca del manejo del *constructor* “MsgSend” en lenguaje textual PiCO con respecto al lenguaje visual GraPiCO (52%), también; es mayor el porcentaje (27%) de estudiantes encuestados que conoce, comprende y utiliza muy bien el *constructor* “MsgSend” en lenguaje textual PiCO con respecto al lenguaje visual GraPiCO (23%).

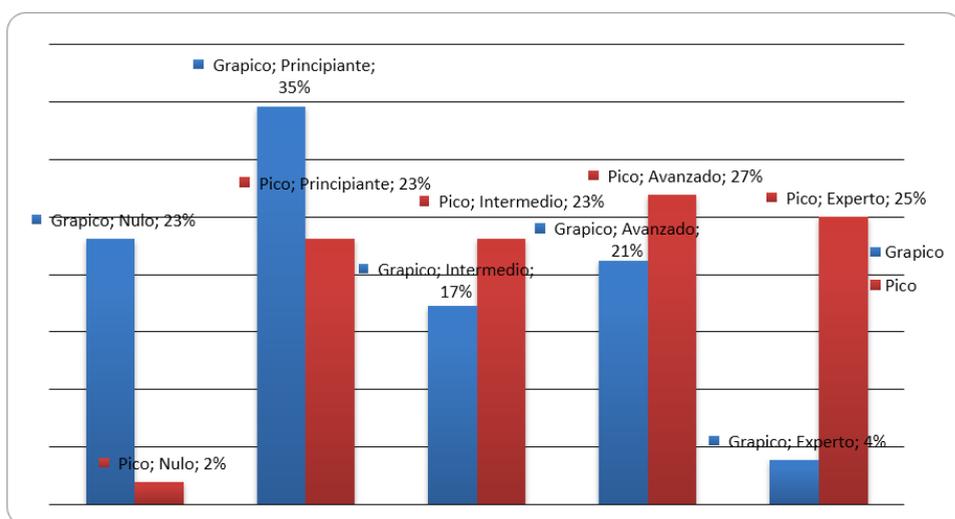
Figura 23. Nivel de comprensión de constructor “MsgSend” diferenciado por PiCO y GraPiCO.



Fuente: elaboración propia (2018).

Llevando a cabo una comparación del nivel de comprensión del *constructor* “Value,” entre lenguajes de programación visuales y textuales, se observó que un mayor porcentaje (52%) de estudiantes encuestados conoce y comprende algo acerca del manejo del *constructor* “Value” en lenguaje visual GraPiCO con respecto al lenguaje textual PiCO (46%), mientras que se observa un mayor porcentaje (52%) de estudiantes encuestados que conoce, comprende y utiliza muy bien el *constructor* “Value” en lenguaje textual PiCO con respecto al lenguaje visual GraPiCO (25%), siendo esto algo muy diciente de la Figura 24.

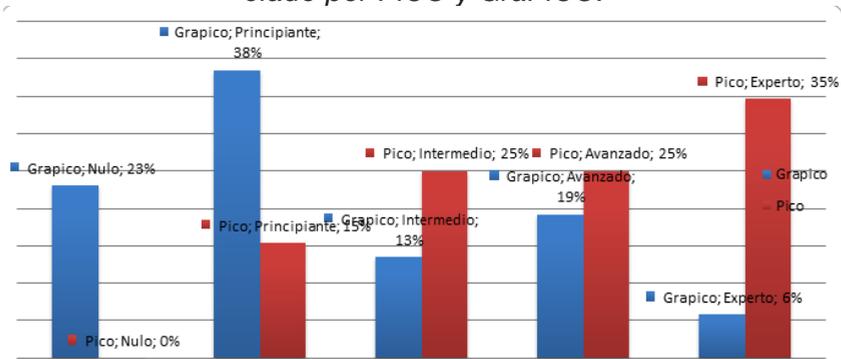
Figura 24. Nivel de comprensión de constructor “Value” diferenciado por PiCO y GraPiCO.



Fuente: elaboración propia (2018) .

Según la Figura 25, comparando el nivel de comprensión del *constructor* “Variable” entre lenguajes de programación visuales y textuales, observamos que un mayor porcentaje (51%) de estudiantes encuestados conoce y comprende algo acerca del manejo del *constructor* “Variable” en lenguaje visual GraPiCO con respecto al lenguaje textual PiCO (40%), mientras que se observa un mayor porcentaje (60%) de estudiantes encuestados que conoce, comprende y utiliza muy bien el *constructor* “Variable” en lenguaje textual PiCO con respecto al lenguaje visual GraPiCO (25%).

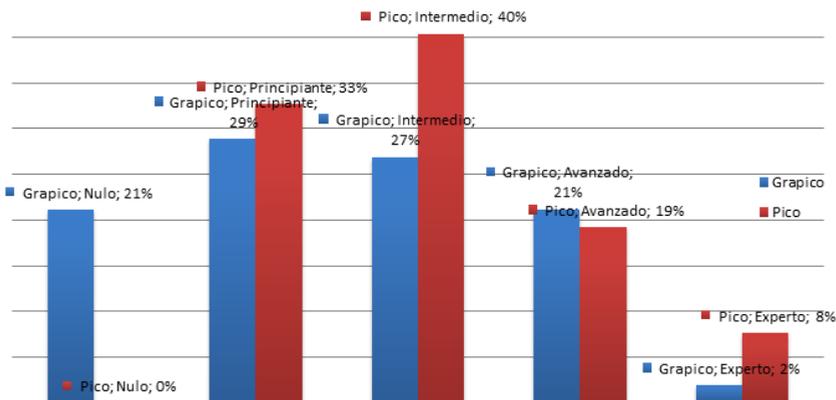
Figura 25. Nivel de comprensión de constructor “Variable” diferenciado por PiCO y GraPiCO.



Fuente: elaboración propia (2018).

Realizando un análisis de la Figura 26, que muestra la comparación del nivel de comprensión del constructor “Argument” entre lenguajes de programación visuales y textuales, se observó que un mayor porcentaje (73%) de estudiantes encuestados conoce y comprende algo acerca del manejo del constructor “Argument” en lenguaje textual PiCO con respecto al lenguaje visual GraPiCO (56%), también; es mayor el porcentaje (27%) de estudiantes encuestados que conoce, comprende y utiliza muy bien el constructor “Argument” en lenguaje textual PiCO con respecto al lenguaje visual GraPiCO (23%).

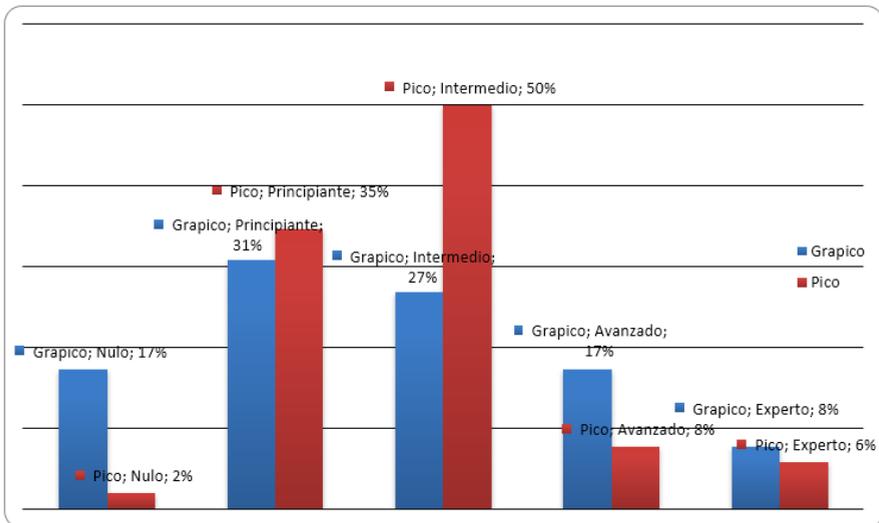
Figura 26. Nivel de comprensión de constructor “Argument” diferenciado por PiCO y GraPiCO.



Fuente: elaboración propia (2018)

Usando la Figura 27, para visualizar el nivel de comprensión del *constructor* “Sender” entre lenguajes de programación visuales y textuales, se observó que un mayor porcentaje (85%) de estudiantes encuestados conoce y comprende algo acerca del manejo del *constructor* “Sender” en lenguaje textual PiCO con respecto al lenguaje visual GraPiCO (58%), mientras que, se observa un mayor porcentaje (25%) de estudiantes encuestados que conoce, comprende y utiliza muy bien el *constructor* “Sender” en lenguaje visual GraPiCO con respecto al lenguaje textual PiCO (14%).

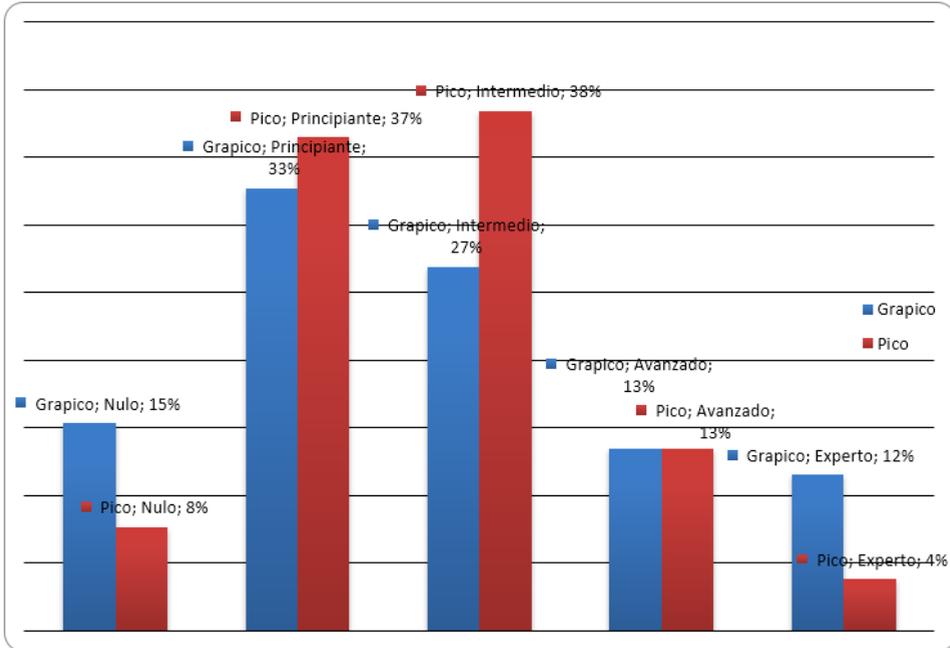
Figura 27. Nivel de comprensión de *constructor* “Sender” diferenciado por PiCO y GraPiCO.



Fuente: elaboración propia (2018).

Visualizando la Figura 28, se expone el nivel de comprensión del *constructor* “Forward” entre lenguajes de programación visuales y textuales, observando que un mayor porcentaje (75%) de estudiantes encuestados conoce y comprende algo acerca del manejo del *constructor* “Forward” en lenguaje textual PiCO con respecto al lenguaje visual GraPiCO (58%), mientras que, se observa un mayor porcentaje (25%) de estudiantes encuestados que conoce, comprende y utiliza muy bien el *constructor* “Forward” en lenguaje visual GraPiCO con respecto al lenguaje textual PiCO (17%).

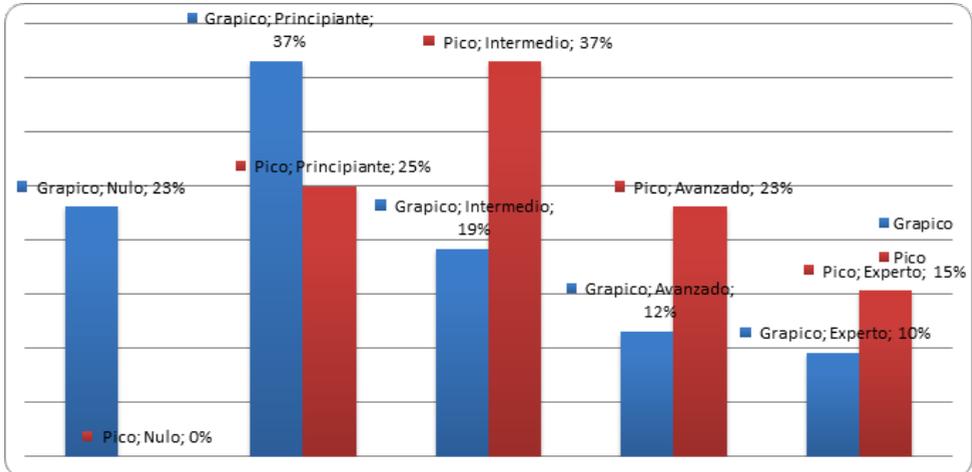
Figura 28. Nivel de comprensión de constructor “Forward” diferenciado por PiCO y GraPiCO.



Fuente: elaboración propia (2018).

Apoyándose en la Figura 29, Comparando el nivel de comprensión del *constructor* “Operators” entre lenguajes de programación visuales y textuales, se observa que un mayor porcentaje (62%) de estudiantes encuestados conoce y comprende algo acerca del manejo del *constructor* “Operators” en lenguaje textual PiCO con respecto al lenguaje visual GraPiCO (56%), también; es mayor el porcentaje (38%) de estudiantes encuestados que conoce, comprende y utiliza muy bien el *constructor* “Operators” en lenguaje textual PiCO con respecto al lenguaje visual GraPiCO (22%). Además, se observa que un 23% de estudiantes no tiene ningún conocimiento y tampoco comprende el manejo del *constructor* “Operators” en lenguaje visual GraPiCO.

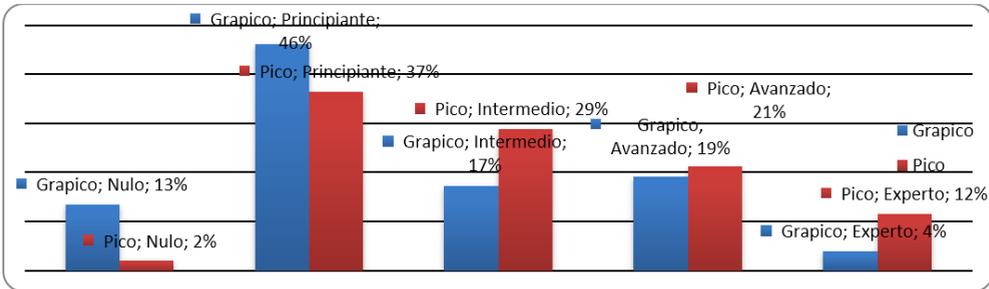
Figura 29. Nivel de comprensión de constructor “Operators” diferenciado por PiCO y GraPiCO.



Fuente: elaboración propia (2018).

Comparando el nivel de comprensión del *constructor* “Relations” entre lenguajes de programación visuales y textuales y soportando el análisis en la Figura 30, se observa que un mayor porcentaje (66%) de estudiantes encuestados conoce y comprende algo acerca del manejo del constructor “Relations” en lenguaje textual PiCO con respecto al lenguaje visual GraPiCO (63%), también; es mayor el porcentaje (33%) de estudiantes encuestados que conoce, comprende y utiliza muy bien el *constructor* “Relations” en lenguaje textual PiCO con respecto al lenguaje visual GraPiCO (23%). Además, se observa que es mayor el porcentaje de estudiantes que no tiene ningún conocimiento y tampoco comprende el manejo del constructor “Relations” en lenguaje visual GraPiCO con respecto al lenguaje textual PiCO.

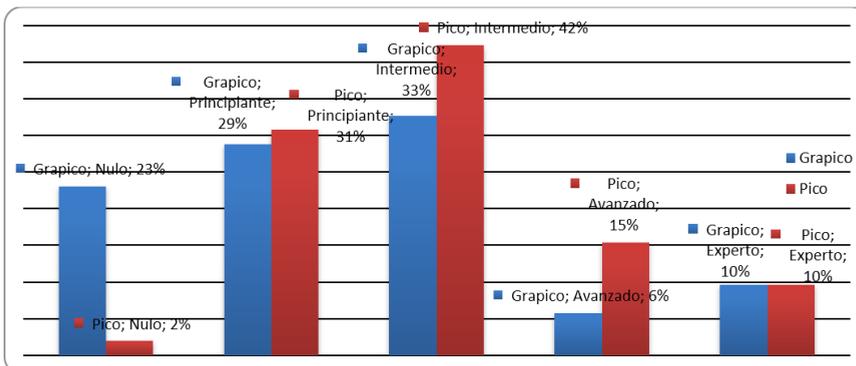
Figura 30. Nivel de comprensión de constructor “Relations” diferenciado por PiCO y GraPiCO.



Fuente: elaboración propia (2018).

Ajustándose a la Figura 31 y comparando el nivel de comprensión del constructor “Constraints” entre lenguajes de programación visuales y textuales, se concluye que un mayor porcentaje (73%) de estudiantes encuestados conoce y comprende algo acerca del manejo del constructor “Constraints” en lenguaje textual PiCO con respecto al lenguaje visual GraPiCO (62%), también; es mayor el porcentaje (25%) de estudiantes encuestados que conoce, comprende y utiliza muy bien el *constructor* “Constraints” en lenguaje textual PiCO con respecto al lenguaje visual GraPiCO (16%). Además, se observa que es mayor el porcentaje de estudiantes que no tiene ningún conocimiento y tampoco comprende el manejo del *constructor* “Constraints” en lenguaje visual GraPiCO con respecto al lenguaje textual PiCO.

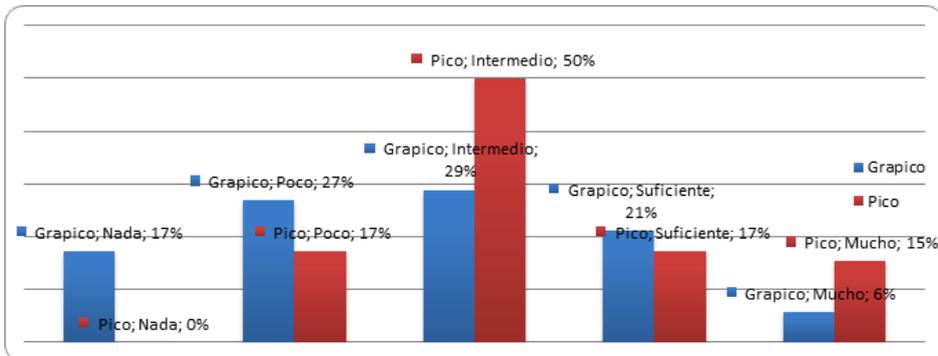
Figura 31. Nivel de comprensión de constructor “Constraints” diferenciado por PiCO y GraPiCO.



Fuente: elaboración propia (2018).

Según la Figura 32 y realizando una comparación del grado de claridad de los *constructores* entre lenguajes de programación visuales y textuales, se encontró que un mayor porcentaje (67%) de estudiantes encuestados tiene algo de claridad con respecto a la interpretación de las características y propiedades de los *constructores* en lenguaje textual PiCO con respecto al lenguaje visual GraPiCO (56%), también; es mayor el porcentaje (32%) de estudiantes encuestados que tiene una adecuada interpretación de las características y propiedades de los *constructores* en lenguaje textual PiCO con respecto al lenguaje visual GraPiCO (27%). Además, se observa que un 17% de estudiantes considera que no tiene claridad con respecto al manejo de los *constructores* en lenguaje visual GraPiCO.

Figura 32. Nivel de Claridad de constructores diferenciado por PiCO y GraPiCO.

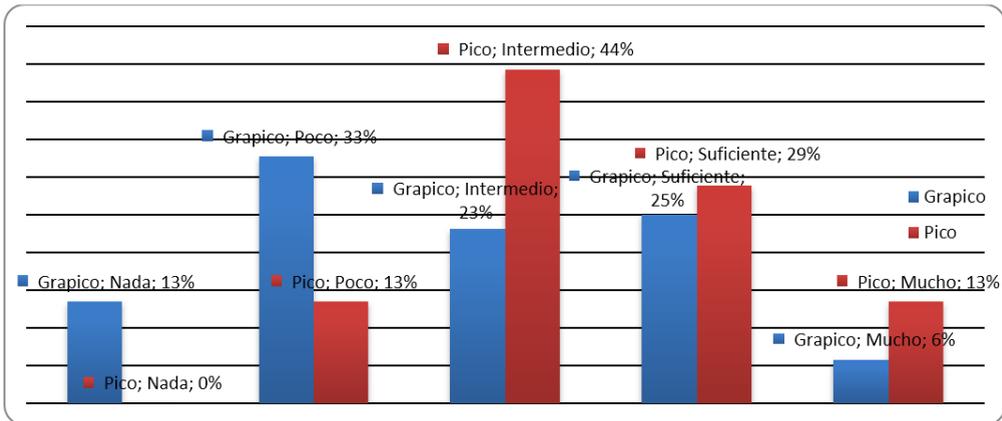


Fuente: elaboración propia (2018).

Comparando el nivel de comprensión de la simbología de los *constructores*, entre lenguajes de programación visuales y textuales, se encontró que un mayor porcentaje (57%) de estudiantes encuestados comprende algo acerca de la simbología utilizada para los constructores en lenguaje textual PiCO con respecto al lenguaje visual GraPiCO (56%), según Figura 33, también; es mayor el porcentaje (42%) de estudiantes encuestados que comprende claramente la simbología utilizada para los constructores en lenguaje textual PiCO con respecto al lenguaje visual GraPiCO (31%). Además, se observa que un 13% de estudiantes no comprende en

absoluto la simbología utilizada para los constructores en lenguaje visual GraPiCO.

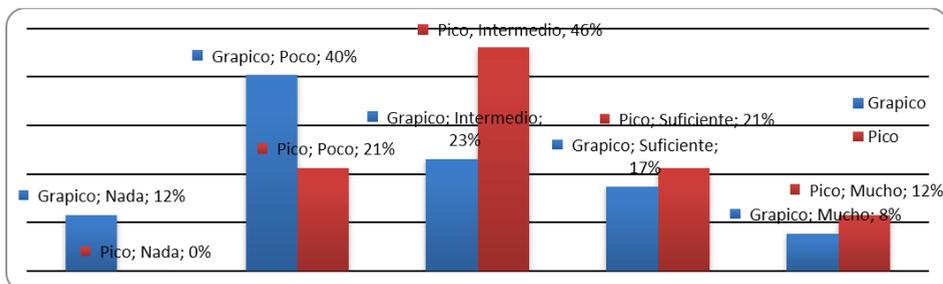
Figura 33. Nivel de Simbología de constructores diferenciado por PiCO y GraPiCO.



Fuente: elaboración propia (2018).

Comparando la navegación de los *constructores* entre lenguajes de programación visuales y textuales, se encontró que un mayor porcentaje (67%) de estudiantes encuestados considera que las ayudas para declarar los *constructores* no es tan buena en lenguaje textual PiCO con respecto al lenguaje visual GraPiCO (63%); también, es mayor el porcentaje (33%) de estudiantes encuestados que considera que las ayudas para declarar los *constructores* es muy buena en lenguaje textual PiCO con respecto al lenguaje visual GraPiCO (25%). Además, apoyándose en la Figura 34, se observa que un 12% de estudiantes considera que las ayudas para declarar los *constructores* son muy malas en lenguaje visual GraPiCO.

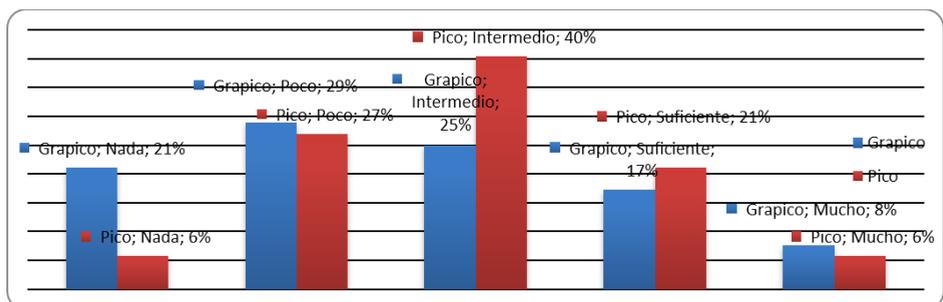
Figura 34. Nivel de Navegación de constructores diferenciado por PiCO y GraPiCO.



Fuente: elaboración propia (2018).

Comparando el nivel de interés entre el editor textual (aplicación) para PiCO y la aplicación E_GraPiCO de acuerdo a su estabilidad, se observa mediante la Figura 35, que un mayor porcentaje (67%) de estudiantes encuestados tiene algo de interés en el editor textual con respecto a la aplicación E_GraPiCO de acuerdo a su estabilidad (54%), también; es mayor el porcentaje (27%) de estudiantes encuestados que tiene mucho interés en el editor textual con respecto a la aplicación E_GraPiCO (25%). Además, se observa que es mayor el porcentaje de estudiantes que no tiene ningún interés en la aplicación E_GraPiCO con respecto al editor textual (aplicación) para PiCO.

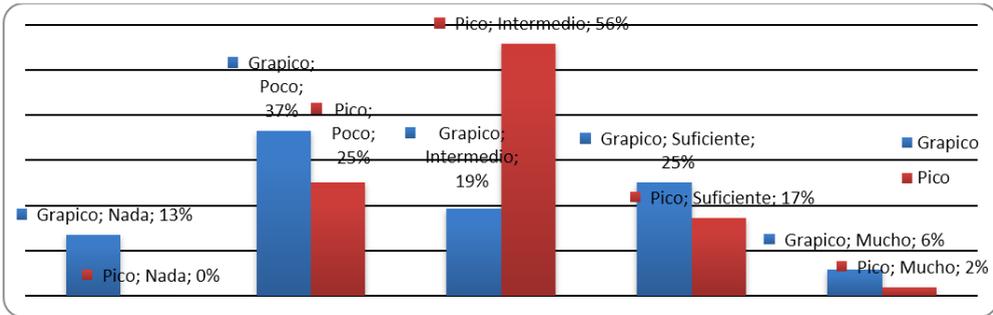
Figura 35. Nivel de Interés de acuerdo a la estabilidad, diferenciado por PiCO y GraPiCO.



Fuente: elaboración propia (2018).

Comparando el nivel de interés entre el editor textual (aplicación) para PiCO y la aplicación E_GraPiCO de acuerdo a su diseño, apoyándose en la Figura 36, se concluye que un mayor porcentaje (81%) de estudiantes encuestados tiene algo de interés en el editor textual con respecto a la aplicación E_GraPiCO de acuerdo a su diseño (56%), mientras que, se observa que un mayor porcentaje (31%) de estudiantes encuestados tiene mucho interés en la aplicación E_GraPiCO con respecto al editor textual para PiCO (19%). Además, se observa que el 13% de estudiantes no tiene ningún interés en la aplicación E_GraPiCO de acuerdo a su diseño.

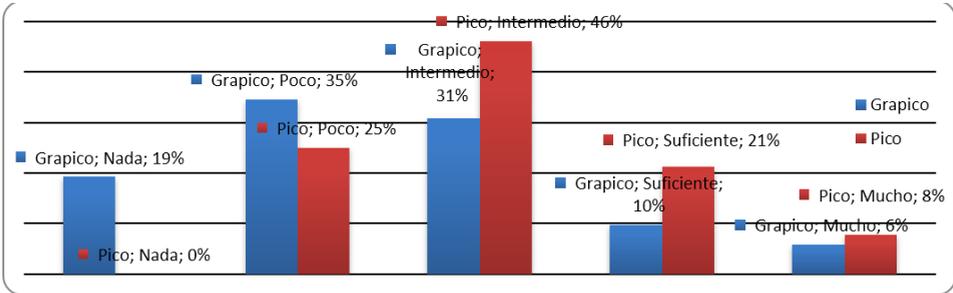
Figura 36. Nivel de Interés de acuerdo al diseño, diferenciado por PiCO y GraPiCO.



Fuente: elaboración propia (2018).

Soportando la Figura 37, se puede comparar el nivel de interés entre el editor textual (aplicación) para PiCO y la aplicación E_GraPiCO de acuerdo a su modificabilidad, observando que un mayor porcentaje (71%) de estudiantes encuestados tiene algo de interés en el editor textual con respecto a la aplicación E_GraPiCO de acuerdo a su modificabilidad (61%); también, es mayor el porcentaje (29%) de estudiantes encuestados que tiene mucho interés en el editor textual con respecto a la aplicación E_GraPiCO (16%). Además, se observa que el 19% de estudiantes no tiene ningún interés en la aplicación E_GraPiCO de acuerdo a su modificabilidad.

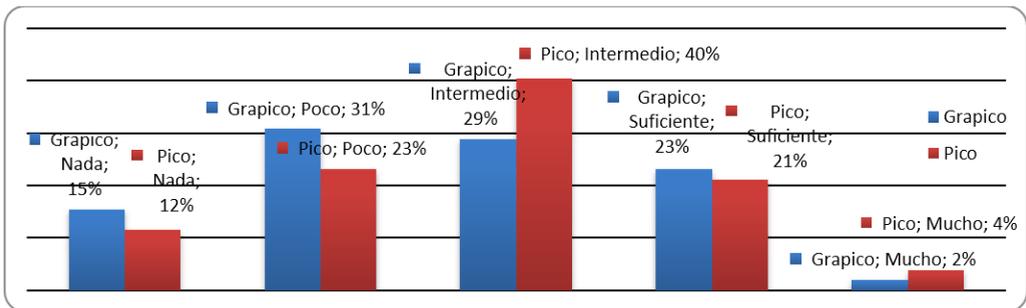
Figura 37. Nivel de Interés de acuerdo a la modificabilidad, diferenciado por PiCO y GraPiCO.



Fuente: elaboración propia (2018).

Comparando el nivel de interés entre el editor textual (aplicación) para PiCO y la aplicación E_GraPiCO de acuerdo a su usabilidad, se observa que un mayor porcentaje (63%) de estudiantes encuestados tienen algo de interés en el editor textual con respecto a la aplicación E_GraPiCO de acuerdo a su usabilidad (60%), también, mediante la ayuda brindada por la Figura 38; se observa que es igual el porcentaje (25%) de estudiantes encuestados que tiene mucho interés en el editor textual y la aplicación E_GraPiCO. Además, vemos que es mayor el porcentaje de estudiantes que no tiene ningún interés en la aplicación E_GraPiCO con respecto al editor textual (aplicación) para PiCO.

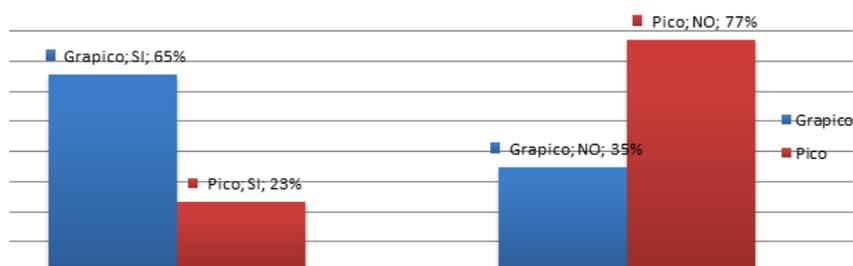
Figura 38. Nivel de Interés de acuerdo a la usabilidad, diferenciado por PiCO y GraPiCO.



Fuente: elaboración propia (2018).

Observando los resultados encontrados, se puede concluir mediante la visualización de la Figura 41, que un mayor porcentaje de estudiantes encuestados (65%) opina que la dinámica audiovisual expuesta, imágenes, efectos, sonorización y la presentación es estética para el cálculo de programación visual con respecto al cálculo de programación textual (23%).

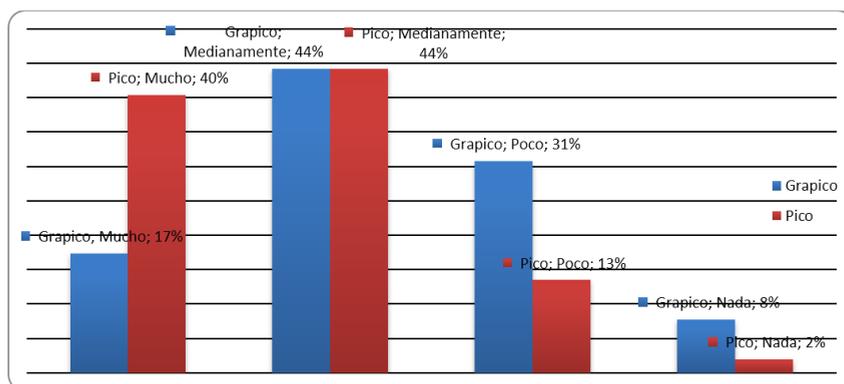
Figura 41. Nivel de estética de la dinámica audiovisual expuesta.



Fuente: elaboración propia (2018).

Por último, teniendo en cuenta los resultados generados y la Figura 42, se puede concluir que un mayor porcentaje de estudiantes encuestados, considera que fue muy importante utilizar un material audiovisual (video dramatizado) porque facilitó el entendimiento del ejercicio de modelación en el taller para el cálculo de programación textual (84%) y el cálculo de programación visual (61%).

Figura 42. Facilidad de entendimiento del ejercicio de modelación en el taller de los cálculos de programación GraPiCO y PiCO.



Fuente: elaboración propia (2018).

V. Conclusión.

Este estudio dejó muchas cosas interesantes; la preferencia actual por los lenguajes de programación Visual, aunque eso no implique que se facilite el aprendizaje de ciertos constructores del lenguaje GraPiCO, equivalentes en PiCO.

Muchos estudiantes encuestados tienen un mayor nivel de conocimiento en lenguajes de programación textual, que en lenguajes de programación visual. Se mostró el interés de muchos estudiantes por aprender más acerca de lenguajes de programación visual. Tendencia que vemos en la actualidad, debido al despliegue tecnológico de herramientas portátiles y móviles de muchos fabricantes, que implican el uso de interfaces y aplicaciones gráficas, especialmente basado en iconos, despertando el uso y desarrollo de la Programación Visual en muchos programadores. Con respecto a lo textual, es algo que no va a desaparecer, porque hasta el momento sigue siendo muy utilizado y da el soporte como infraestructura para que existan los lenguajes visuales, además de tener muchos adeptos en desarrollo de aplicaciones.

Se notó que muchas personas encuestadas usaban de manera errónea el término de *Programación Visual*, pero con este estudio y experimento, se pudo aclarar que es un tipo de programación en el cual, por medio de íconos, se pueden crear programas y que no implican el uso de lenguajes textuales; y que los lenguajes de Programación Textual, que tienen alguna interfaz gráfica para visualizar lo que se desarrolla, no es Programación Visual.

VI. Bibliografía

- [1] A. L. Webster, Estadística aplicada a los negocios y la economía, Bogotá: Mc Graw Hill, 2001.
- [2] D. C. Montgomery y G. C. Runger, Applied Statistics and Probability for Engineers, United States of America: John Wiley & Sons, Inc., 2003.
- [3] C. M. Cuadras, Problemas de Probabilidades y Estadística, Barcelona: Promociones y Publicaciones Universitarias S.A., 1990.
- [4] M. Yañez, Ingeniería de Confiabilidad y Análisis Probabilístico de Riesgo, 2003.
- [5] Matronas Profesión, vol. 5, nº 17, 2004.

ESTUDIO DE RESULTADOS DE PiCO y GraPiCO

PARTE 2

*Marco Antonio Triana Lozano
Carlos Andrés Tavera Romero*

I. Introducción

Mediante el presente documento, se logró realizar un estudio comparativo entre lenguajes textuales y lenguajes visuales, específicamente entre lenguajes de programación textual PiCO y visual GraPiCO. Por medio de un análisis estadístico, se obtuvieron algunos resultados para establecer el nivel de aceptación de estos dos tipos de lenguajes y conocer bajo qué condiciones es más adecuado un lenguaje de programación que otro; con el fin de hacer mejoras a dicho software. Inicialmente, se escogió una población (objetiva) específica definida con anterioridad por el grupo de investigadores participantes en este proyecto institucional para realizar el estudio; en el cual se realizaron mediciones acerca del nivel de conocimiento, asimilación, comprensión y aceptación de estos dos lenguajes de programación.

II. Marco Teórico

Este proyecto interinstitucional, busca dar apoyo al Laboratorio de Investigación para el Desarrollo de la Ingeniería de Software (LI-

DIS), de la Universidad de San Buenaventura Cali, en un trabajo posdoctoral consistente en el estudio entre el Cálculo textual: PiCO y el Cálculo visual: GraPiCO, los cuales hacen parte del programa E_GraPiCO.

Por esta razón, se propone un proyecto enfocado hacia el mejoramiento de E_GraPiCO, para lo cual es necesario hacer un estudio comparativo de los cálculos de programación PiCO y GraPiCO.

En primera instancia, se construyó un material audiovisual didáctico con el apoyo del grupo de investigadores de Comunicación Social, por medio del cual se socializaron los conceptos técnicos básicos a través de un lenguaje entendible para estudiantes con conocimientos básicos en Ingeniería de sistemas.

Este producto, sirvió como herramienta didáctica mediadora entre el grupo de investigadores y el público objetivo (personas a quienes se les aplicó la encuesta). Desde esta perspectiva, se hizo el diseño, desarrollo e implementación de dos videos expositivos que permitieron ilustrar de manera gráfica y sonora los componentes de manejo requeridos en la investigación. Uno de los productos audiovisuales corresponde al Cálculo textual (PiCO) y el otro al visual (GraPiCO). Ver Tabla 27.

Tabla 27. Etapas del estudio comparativo

Estudio comparativo entre lenguajes textuales y lenguajes visuales. Caso: PiCO y GraPiCO	
Etapa 1	Elaboración de hipótesis en experimentos de lenguajes de programación.
Etapa 2	Variabes en un experimento de lenguajes de programación.
Etapa 3	Unidades experimentales utilizadas en pruebas de lenguajes de programación.
Etapa 4	Tratamientos y replicas en un experimento de programación.
Etapa 5	Aprendizaje Basado en Problemas (ABP) aplicado a los lenguajes de programación.

Etapa 6	La comunicación en el estudio comparativo entre lenguajes textuales y lenguajes visuales: Caso PiCO y GraPiCO.
Etapa 7	Sistematización de una experiencia de investigación entre la Comunicación Social y la Ingeniería de Software.
Etapa 8	Modelo de sistematización propuesto “TCACI en doble vía”.
Etapa 9	Pasos en la realización de los audiovisuales pedagógicos: PiCO – GraPiCO y ejercicio de modelación.
Etapa 10	Estudio de resultados de PiCO y GraPiCO. Parte 1.
Etapa 11	Estudio de resultados de PiCO y GraPiCO. Parte 2.
Etapa 12	Recomendaciones y resultados del estudio entre PiCO, GraPiCO y editores.

Fuente: elaboración propia (2018).

En etapas previas se explicó la *hipótesis* aplicada y su verificación, las *variables* empleadas, los *sujetos de estudio*, las diferentes asignaciones hechas a las *unidades experimentales* mediante el uso de *tratamientos* y las *observaciones* efectuadas. También, se presentó la contribución del *Aprendizaje Basado en Proyectos*.

Todo lo anterior ha presentado la evolución del proceso de ejecución de este estudio; sin embargo, aún queda faltando presentar parte del análisis correspondiente a los resultados obtenidos, mediante la presentación de cada uno de estos dos lenguajes de programación, proceso que arrojó información muy importante para el estudio a través de encuestas, tema a tratar en este documento.^[7]

Las encuestas o cuestionarios utilizados para la recolección de datos en este experimento, fueron diseñados según algunos lineamientos, los cuales serán presentados a continuación.^[7]

Los cuestionarios son muy utilizados en el ámbito de la investigación, puesto que generan escalas e índices para facilitar la medición (ya que estandariza la información recogida por el cuestionario) a costos relativamente bajos llegando a captar información

de una mayor cantidad de participantes, permitiendo el análisis de toda la información de una forma muy práctica y fácil.^[7]

En el momento de elaborar un cuestionario, se debe tener claro todo el conocimiento acerca de lo que se va a medir, al igual que ciertos conocimientos estadísticos muy útiles.^[7]

Un buen cuestionario, se caracteriza por:

1. Ser adecuado para el problema a medir y tener un contenido valido (lo cual implica que permita medir lo que se pretende).^[7]
2. Ser fiable, es decir tener la cantidad mínima posible de error en la medida a realizar.^[7]
3. Ser flexible a los cambios de los participantes encuestados.^[7]
4. Tener sus componentes, lo más delimitados posible.^[7]

Para obtener un cuestionario adecuado, fiable, y flexible, se debe seguir los siguientes pasos:

1. Definir aquello que se desea medir.^[7]
2. Establecer el contenido del cuestionario, teniendo en cuenta aspectos tales como la población a la que va dirigido, la forma de distribución y el formato del mismo.^[7]
3. La correcta composición de los ítems (unidad básica del mismo) del cuestionario.^[7]
4. Cantidad de ítems.^[7]
5. Definir el contenido.^[7]
6. Orden adecuado de los ítems.^[7]
7. Contar con preguntas muy bien definidas y estructuradas, lo cual implica tener en cuenta el lenguaje y la cultura del encuestado; por lo tanto se recomienda que éstas sean cortas y fácil de entender.^[7]

III. Modelamiento

A través de este proyecto se busca conocer cual cálculo de programación (PiCO ó GraPiCO) prefieren más los usuarios, los cuales fueron caracterizados en la población objetivo.

Para realizar el estudio comparativo entre los lenguajes de programación PiCO y GraPiCO se siguieron los siguientes pasos:

1. Definición de la *Población objetivo*: Todos los estudiantes de ingenierías en sistemas de segundo, tercero, cuarto y quinto semestre académico, de la Universidad de San Buenaventura, Universidad Autónoma de Occidente, Universidad Javeriana y la Universidad Icesi de la ciudad de Cali, con edades entre 16 y 23 años de cualquier sexo.
2. Definición de las *Variables de interés*: Para determinar cuáles características se debían tener en cuenta en el estudio, se reunió el grupo de participantes en el proyecto para explicar y socializar los diferentes conceptos básicos de los cálculos de programación PiCO y GraPiCO. Las características más importantes fueron: nivel de conocimiento, preferencias, tipo de interés, nivel de asimilación, nivel de comprensión y aceptación de estos dos lenguajes de programación.
3. Después de identificar estas características se *diseñó el cuestionario* acorde a los objetivos del proyecto.
4. Con los productos audiovisuales que se desarrollaron correspondientes al Cálculo textual (PiCO) y al visual (GraPiCO), se realizaron dos exposiciones lo suficientemente ilustrativas y dinámicas, cada una en su respectivo lenguaje de programación. En cada universidad, los usuarios a encuestar (población objetivo) se dividieron en dos grupos. Posteriormente, a un grupo se le presentaron los conceptos básicos de los lenguajes de programación textual y al otro grupo los conceptos básicos de los lenguajes de programación visual.
5. Al final de cada exposición se discutieron y aclararon algunas dudas que se presentaron a las personas que vieron la presentación.

6. Con el propósito de hacer el estudio comparativo lo más homogéneo posible, es decir, condiciones similares en los dos grupos de estudio, aleatoriamente se escogieron los estudiantes de cada universidad que participaron en la investigación. Luego, estos usuarios se dividieron en dos grupos, los integrantes de cada grupo fueron escogidos al azar. También, ambos productos audiovisuales deben manejar una narrativa audiovisual igual para evitar errores sistemáticos generados por factores no controlables por el investigador y que pueden influir en los resultados introduciendo errores sistemáticos. Lo anterior significa influenciar lo menos posible, la opinión de los usuarios que participaron en el estudio acerca de estos dos lenguajes de programación.
7. Se hizo una prueba piloto para observar posibles errores en el diseño preliminar de la encuesta o cuestionario, inclusión de otras variables o eliminación de algunas de ellas.
8. Se realizó el *diligenciamiento del cuestionario* definitivo por parte de los estudiantes escogidos en la población objetivo.
9. Calificación y sistematización de los resultados de las encuestas.
10. Estudio de los resultados de la encuesta, los cuales permiten hacer una mejora en los lenguajes de programación, gracias a los resultados que nos arroje la encuesta en el estudio comparativo.
11. Análisis de la información obtenida.
12. Consolidación de resultados y socialización de conclusiones.

IV. Programa E_Grapico

E_GraPiCO fue creado siguiendo cada una de las especificaciones formales; gracias a esto fue posible la demostración de su corrección y completitud. Es por lo anterior que E_GraPiCO es una aplicación estable y funcional, pero requiere un último conjunto de

modificaciones de forma, que consisten en las recomendaciones de usuario final, que surgirán de un estudio comparativo donde se aprecien las ventajas de la utilización de un editor visual ante uno textual, y de igual forma donde se harán recomendaciones de mejoramiento del editor de programas gráficos existente. Hasta el momento, no existían elementos para efectuar un estudio del tipo que se propone, porque no había disponibles dos lenguajes, uno textual y otro visual, con equivalencia expresiva formalmente demostrada. Después del desarrollo de PiCO y GraPiCO esta dificultad fue superada.

De otro lado, la discusión sobre cuál es mejor entre lo visual y lo textual es un asunto de actualidad, porque con la llegada de los procesadores cada vez más veloces, se posibilita desplegar información cada vez más compleja. Esto, tiene como consecuencia que ya estén en el mercado y la academia, sistemas operativos y lenguajes de programación cada vez más visuales; pero los mismos procesadores que permiten la utilización de lenguajes visuales, también se convierten en su principal limitante, al exigir que el código que ellos deben procesar tiene que ser textual para su análisis y almacenamiento, mediante los mecanismos conocidos actualmente; produciendo esta situación que se utilicen los lenguajes totalmente visuales estrictamente para presentación de la información.

Grupos de investigación internacional como el IRKAM de Francia y nacionales como AVISPA, han estado creando aplicaciones y desarrollando investigaciones relacionadas con los lenguajes de programación visuales y textuales, pero hasta el momento no se han adelantado estudios comparativos desde la visión de la didáctica, que den información sobre la interacción humano-lenguaje de programación ya sea visual o textual.

V. RESULTADOS

De acuerdo al estudio comparativo entre lenguajes de programación visuales y textuales, caso PiCO y GraPiCO, que se realizó con estudiantes de Ingeniería en Sistemas que cursan entre segundo y quinto semestre académico, con edades entre 16 y 23 años, de las universidades de San Buenaventura, Autónoma de Occidente, Ja-

veriana y el ICESI de la ciudad de Cali; se encontraron los siguientes resultados: (Ver las cifras estadísticas y gráficos correspondientes que se muestran en el etapa anterior Estudio de Resultados de PiCO y GraPiCO. Parte 1):

- La mayoría de estos estudiantes, tiene un nivel intermedio en lenguajes de programación. (Ver Figura 12, de la etapa anterior. Nivel de conocimiento en lenguajes de programación).
- Un mayor porcentaje de estos estudiantes, prefiere de alguna manera el lenguaje de programación visual con respecto al lenguaje de programación textual. (Ver Figura 13, de la etapa anterior. Tipo de lenguaje de programación preferido).
- Comparando el nivel de conocimiento entre lenguajes de programación visual y textual, se encontró que es mayor el porcentaje de estudiantes que tienen algo de conocimiento en lenguaje textual PiCO con respecto al lenguaje visual GraPiCO. Caso contrario en los estudiantes que tienen mucho conocimiento en lenguajes de programación, es mayor el porcentaje para GraPiCO con respecto a PiCO. (Ver Figura 14, de la etapa anterior. Nivel de conocimiento en lenguajes de programación visual y textual).
- Comparando la frecuencia de uso de aplicaciones, que de alguna manera utilizan lenguajes de programación visuales y textuales, se encontró que es mayor el porcentaje de estudiantes que utiliza con alguna frecuencia el lenguaje visual GraPiCO con respecto al lenguaje textual PiCO. Caso contrario en los estudiantes que utilizan con más frecuencia los lenguajes de programación, es mayor el porcentaje para PiCO con respecto a GraPiCO. (Ver Figura 15, de la etapa anterior. Frecuencia de uso de lenguajes de programación visual y textual).
- Comparando el nivel de interés entre lenguajes de programación visual y textual, se encontró que es mayor el porcentaje de estudiantes que tienen algo de interés en lenguaje textual PiCO, con respecto al lenguaje visual GraPiCO. Caso contrario en los estudiantes que están muy interesados en los lenguajes de programación, es mayor el porcentaje para GraPi-

CO con respecto a PiCO. (Ver Figura 16, de la etapa anterior. Interés en lenguajes de programación visual y textual)

- Comparando el nivel de comprensión del constructor “program”, entre lenguajes de programación visuales y textuales, se encontró que es mayor el porcentaje de estudiantes que conoce, comprende y utiliza muy bien el constructor “program” en lenguaje textual PiCO con respecto al lenguaje visual GraPiCO. (Ver Figura 17, de la etapa anterior. Comprensión de constructor “Program”).
- Comparando el nivel de comprensión del constructor “context”, entre lenguajes de programación visual y textual, se encontró que es mayor el porcentaje de estudiantes que conoce y comprende algo acerca del manejo del constructor “context” en lenguaje textual PiCO con respecto al lenguaje visual GraPiCO. Caso contrario en los estudiantes que conocen, comprenden y utilizan muy bien el constructor “context”, es mayor el porcentaje para GraPiCO con respecto a PiCO. (Ver Figura 18, de la etapa anterior. **Comprensión de constructor “Context”**).
- Comparando el nivel de comprensión del constructor “Objects”, entre lenguajes de programación visuales y textuales, se encontró que es mayor el porcentaje de estudiantes que conocen y comprenden algo acerca del manejo del constructor “Objects” en lenguaje visual GraPiCO con respecto al lenguaje textual PiCO. Caso contrario en los estudiantes que conocen, comprenden y utilizan muy bien el constructor “Objects”, es mayor el porcentaje para PiCO con respecto a GraPiCO. (Ver Figura 19, de la etapa anterior. Comprensión de constructor “Objects”).
- Comparando el nivel de comprensión del constructor “Methods”, entre lenguajes de programación visuales y textuales, se encontró que es mayor el porcentaje de estudiantes que conoce y comprende algo acerca del manejo del constructor “Methods” en lenguaje visual GraPiCO con respecto al lenguaje textual PiCO. Caso contrario en los estudiantes que conocen, comprenden y utilizan muy bien el constructor “Methods”, es mayor el porcentaje para PiCO con respecto a

GraPiCO. (Ver Figura 20, de la etapa anterior. Comprensión de constructor “Methods”).

- Comparando el nivel de comprensión del constructor “Ask”, entre lenguajes de programación visuales y textuales, se encontró que es mayor el porcentaje de estudiantes que conoce, comprende y utiliza muy bien el constructor “Ask” en lenguaje textual PiCO con respecto al lenguaje visual GraPiCO. (Ver Figura 21, de la etapa anterior. Comprensión de constructor “Ask”).
- Comparando el nivel de comprensión del constructor “Tell”, entre lenguajes de programación visuales y textuales, se encontró que en igual proporción los estudiantes conocen y comprenden algo acerca del manejo del constructor “Tell” en lenguaje textual PiCO y en lenguaje visual GraPiCO. Caso diferente en los estudiantes que conocen, comprenden y utilizan muy bien el constructor “Tell”, es mayor el porcentaje para PiCO con respecto a GraPiCO. (Ver Figura 22, de la etapa anterior. Comprensión de constructor “Tell”).
- Comparando el nivel de comprensión del constructor “MsgSend”, entre lenguajes de programación visuales y textuales, se encontró que es mayor el porcentaje de estudiantes que conoce, comprende y utiliza muy bien el constructor “MsgSend” en lenguaje textual PiCO con respecto al lenguaje visual GraPiCO. (Ver Figura 23, de la etapa anterior. Comprensión de constructor “MsgSend”).
- Comparando el nivel de comprensión del constructor “Value”, entre lenguajes de programación visual y textual, se encontró que es mayor el porcentaje de estudiantes que conoce y comprende algo acerca del manejo del constructor “Value” en lenguaje visual GraPiCO con respecto al lenguaje textual PiCO. Caso contrario en los estudiantes que conocen, comprenden y utilizan muy bien el constructor “Value”, es mayor el porcentaje para PiCO con respecto a GraPiCO. (Ver Figura 24, de la etapa anterior. Comprensión de constructor “Value”).
- Comparando el nivel de comprensión del constructor “Variable”, entre lenguajes de programación visuales y textuales, se encontró que es mayor el porcentaje de estudiantes que conoce y comprende algo acerca del manejo del constructor

“Variable” en lenguaje visual GraPiCO con respecto al lenguaje textual PiCO. Caso contrario en los estudiantes que conocen, comprenden y utilizan muy bien el constructor “Variable”; es mayor el porcentaje para PiCO con respecto a GraPiCO. (Ver Figura 25, de la etapa anterior. Comprensión de constructor “Variable”).

- Comparando el nivel de comprensión del constructor “Argument”, entre lenguajes de programación visuales y textuales, se encontró que es mayor el porcentaje de estudiantes que conoce, comprende y utiliza muy bien el constructor “Argument” en lenguaje textual PiCO con respecto al lenguaje visual GraPiCO. (Ver Figura 26, de la etapa anterior. Comprensión de constructor “Argument”).
- Comparando el nivel de comprensión del constructor “Sender”, entre lenguajes de programación visual y textual, se encontró que es mayor el porcentaje de estudiantes que conoce y comprende algo acerca del manejo del constructor “Sender” en lenguaje textual PiCO con respecto al lenguaje visual GraPiCO. Caso contrario en los estudiantes que conocen, comprenden y utilizan muy bien el constructor “Sender”, es mayor el porcentaje para GraPiCO con respecto a PiCO. (Ver Figura 27, de la etapa anterior. Comprensión de constructor “Sender”).
- Comparando el nivel de comprensión del constructor “Forward”, entre lenguajes de programación visual y textual, se encontró que es mayor el porcentaje de estudiantes que conoce y comprende algo acerca del manejo del constructor “Forward” en lenguaje textual PiCO con respecto al lenguaje visual GraPiCO. Caso contrario en los estudiantes que conocen, comprenden y utilizan muy bien el constructor “Forward”, es mayor el porcentaje para GraPiCO con respecto a PiCO. (Ver Figura 28, de la etapa anterior. Comprensión de constructor “Forward”).
- Comparando el nivel de comprensión del constructor “Operators”, entre lenguajes de programación visuales y textuales, se encontró que es mayor el porcentaje de estudiantes que conoce, comprende y utiliza muy bien el constructor “Operators” en lenguaje textual PiCO con respecto al lenguaje visual

GraPiCO. (Ver Figura 29, de la etapa anterior. Comprensión de constructor “Operators”).

- Comparando el nivel de comprensión del constructor “Relations”; entre lenguajes de programación visuales y textuales, se encontró que es mayor el porcentaje de estudiantes que conoce, comprende y utiliza muy bien el constructor “Relations” en lenguaje textual PiCO con respecto al lenguaje visual GraPiCO. (Ver Figura 30, de la etapa anterior. Comprensión de constructor “Relations”).
- Comparando el nivel de comprensión del constructor “Constraints”; entre lenguajes de programación visuales y textuales, se encontró que es mayor el porcentaje de estudiantes que conoce, comprende y utiliza muy bien el constructor “Constraints” en lenguaje textual PiCO con respecto al lenguaje visual GraPiCO. (Ver Figura 31, de la etapa anterior. Comprensión de constructor “Constraints”).
- Comparando el nivel de comprensión de los constructores entre lenguajes de programación visuales y textuales, se encontró que es mayor el porcentaje de estudiantes que no tiene ningún conocimiento y tampoco comprende el manejo de estos constructores para el lenguaje visual GraPiCO con respecto al lenguaje textual PiCO. (Ver Figuras 17 - 31, de la etapa anterior).
- Comparando el grado de claridad de los constructores entre lenguajes de programación visuales y textuales, se encontró que es mayor el porcentaje de estudiantes que tiene una adecuada interpretación de las características y propiedades de los constructores en lenguaje textual PiCO con respecto al lenguaje visual GraPiCO. (Ver Figura 32, de la etapa anterior. Claridad de constructores).
- Comparando el nivel de comprensión de la simbología de los constructores entre lenguajes de programación visuales y textuales, se encontró que es mayor el porcentaje de estudiantes que comprende claramente la simbología utilizada para los constructores en lenguaje textual PiCO con respecto al lenguaje visual GraPiCO. (Ver Figura 33, de la etapa anterior. Simbología de constructores).

- Comparando la navegación de los constructores entre lenguajes de programación visual y textual, se encontró que es mayor el porcentaje de estudiantes que considera que las ayudas para declarar los constructores es muy buena en lenguaje textual PiCO con respecto al lenguaje visual GraPiCO. (Ver Figura 34, de la etapa anterior. Navegación de constructores).
- Al comparar los constructores entre lenguajes de programación visuales y textuales de acuerdo a los siguientes parámetros: claridad, simbología y navegación, se encontró una proporción considerable de estudiantes que considera que no tienen claridad con respecto al manejo de los constructores, no comprenden en absoluto la simbología utilizada para los constructores y además, consideran que las ayudas para declarar los constructores es muy mala en lenguaje visual GraPiCO. (Ver Figuras 32 - 34, de la etapa anterior).
- Comparando el nivel de interés entre el editor textual (aplicación) para PiCO y la aplicación E_GraPiCO, se encontró que es mayor el porcentaje de estudiantes que tiene mucho interés en el editor textual con respecto a la aplicación E_GraPiCO de acuerdo a su estabilidad. (Ver Figura 35, de la etapa anterior. Interés de acuerdo a la estabilidad).
- Comparando el nivel de interés entre el editor textual (aplicación) para PiCO y la aplicación E_GraPiCO, se encontró que es mayor el porcentaje de estudiantes que tiene algo de interés en el editor textual con respecto a la aplicación E_GraPiCO de acuerdo a su diseño. Caso contrario en los estudiantes que tienen mucho interés en su diseño, es mayor el porcentaje para la aplicación E_GraPiCO con respecto al editor textual para PiCO. (Ver Figura 36, de la etapa anterior. Interés de acuerdo al diseño).
- Comparando el nivel de interés entre el editor textual (aplicación) para PiCO y la aplicación E_GraPiCO, se encontró que es mayor el porcentaje de estudiantes que tiene mucho interés en el editor textual con respecto a la aplicación E_GraPiCO de acuerdo a su Modificabilidad. (Ver Figura 37, de la etapa anterior. Interés de acuerdo a la Modificabilidad).
- Comparando el nivel de interés entre el editor textual (aplicación) para PiCO y la aplicación E_GraPiCO, se encontró que

en igual proporción los estudiantes tienen mucho interés en el editor textual y la aplicación E_GraPiCO de acuerdo a su Usabilidad. (Ver Figura 38, de la etapa anterior. Interés de acuerdo a la usabilidad).

- Al comparar los constructores entre lenguajes de programación visual y textual de acuerdo a sus características, se encontró un mayor porcentaje de estudiantes que no tiene ningún interés en la aplicación E_GraPiCO con respecto al editor textual (aplicación) para PiCO de acuerdo a su estabilidad, diseño, modificabilidad y usabilidad. (Ver Figuras 35 - 38, de la etapa anterior).

De acuerdo a otros resultados encontrados en la encuesta, podemos mencionar algunas opiniones de interés:

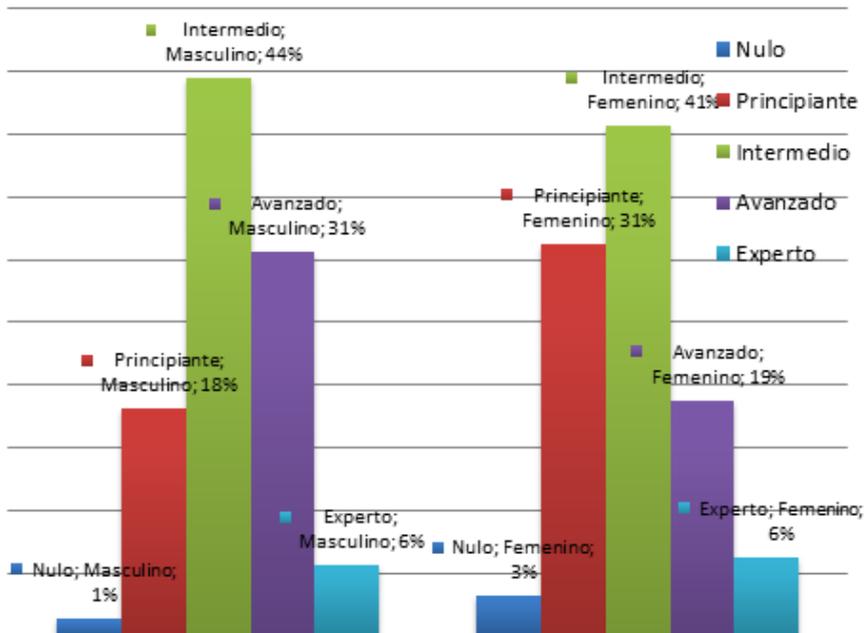
- Se encontró que es mayor el porcentaje de estudiantes que considera que la Ingeniería en Sistemas y la Comunicación Social sí se complementan en el caso específico del material audiovisual expuesto para explicar los conceptos técnicos del cálculo de programación visual con respecto al cálculo de programación textual. (Ver Figura 39, de la etapa anterior. Complemento de la Ingeniería en Sistemas y la Comunicación Social para explicar conceptos técnicos de los cálculos de programación GraPiCO y PiCO).
- Se encontró, que es mayor el porcentaje de estudiantes que considera muy importante utilizar un material audiovisual para facilitar el entendimiento del cálculo de programación, en el caso del editor textual con respecto al cálculo de programación visual. (Ver figura 40, de la etapa anterior. Facilidad de entendimiento de los cálculos de programación GraPiCO y PiCO de acuerdo al material audiovisual generado).
- Se encontró, que es mayor el porcentaje de estudiantes que opina que la dinámica audiovisual expuesta, imágenes, efectos, sonorización y la presentación es estética para el cálculo de programación visual con respecto al cálculo de programación textual. (Ver figura 41, de la etapa anterior. Nivel de estética de la dinámica audiovisual expuesta).
- Observamos que un mayor porcentaje de estudiantes considera que fue muy importante utilizar un material audiovisual (vi-

deo dramatizado) porque facilitó el entendimiento del ejercicio de modelación en el taller para el cálculo de programación textual con respecto al cálculo de programación visual. (Ver Figura 42, de la etapa anterior. Facilidad de entendimiento del ejercicio de modelación en el taller de los cálculos de programación GraPiCO y PiCO).

De acuerdo al estudio comparativo entre lenguajes de programación visuales y textuales: caso PiCO y GraPiCO; se encontraron algunos resultados teniendo en cuenta el género del estudiante:

- De acuerdo a la Figura 43 que muestra un gráfico de barras comparativo entre el nivel de conocimiento en lenguajes de programación de acuerdo al género, se encontró que es mayor el porcentaje de estudiantes de sexo masculino que tiene (mucho) mayor conocimiento en lenguajes de programación, con respecto a los estudiantes de sexo femenino.

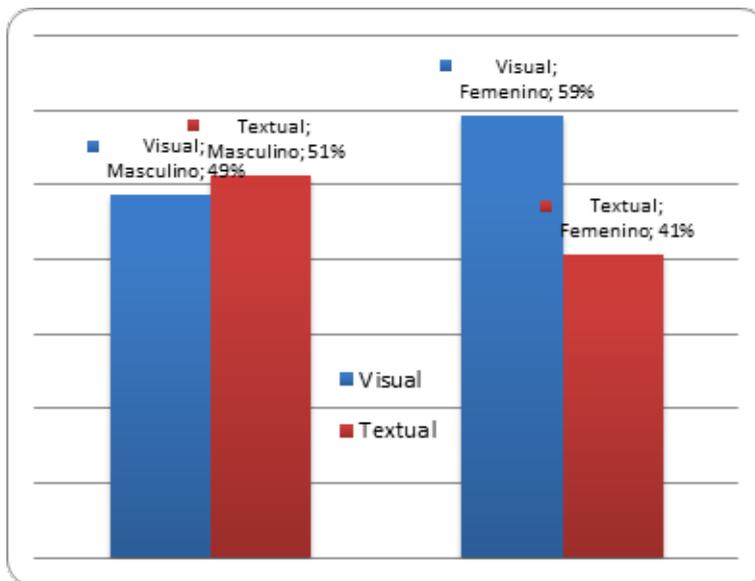
Figura 43. Nivel de conocimiento en lenguajes de programación de encuestados tanto para PiCO como para GraPiCO.



Fuente: elaboración propia (2018).

- La Figura 44 muestra un grafico de barras comparativo entre el tipo de lenguaje de programación preferido de acuerdo al genero; se encontró que es mayor el porcentaje de estudiantes de sexo femenino que prefiere de alguna manera el lenguaje de programación visual con respecto al lenguaje de programación textual comparado con los estudiantes de sexo masculino.

Figura 44. Tipo de lenguaje de programación preferido de encuestados tanto para PiCO como para GraPiCO.

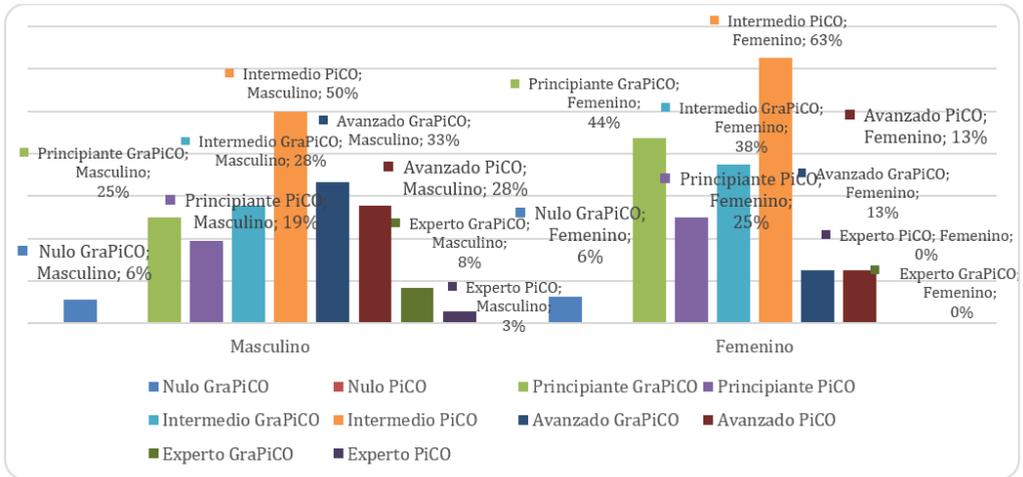


Fuente: elaboración propia (2018).

- Comparando el nivel de conocimiento entre lenguajes de programación visuales y textuales de acuerdo al genero, se encontró que es mayor el porcentaje de estudiantes de sexo femenino que tiene algo de conocimiento en lenguaje textual PiCO con respecto al lenguaje visual GraPiCO comparado con los estudiantes de sexo masculino. Caso contrario en los estudiantes que tienen mucho conocimiento en lenguajes de programación, es mayor el porcentaje para GraPiCO en los hombres con res-

pecto a PiCO comparado con los estudiantes de sexo femenino. Ver Figura 45.

Figura 45. Nivel de conocimiento en lenguajes de programación visual y textual diferenciado por genero.

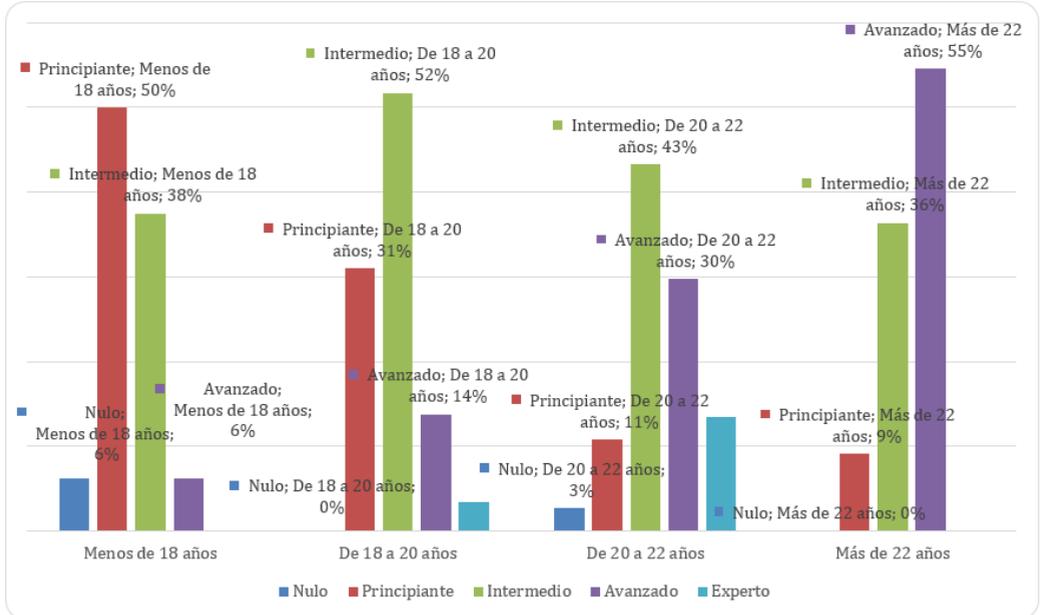


Fuente: elaboración propia (2018).

De acuerdo al estudio comparativo entre lenguajes de programación visuales y textuales: caso PiCO y GraPiCO; se encontraron otros resultados teniendo en cuenta la edad del estudiante:

- De acuerdo a la Figura 46 que muestra un gráfico de barras comparativo entre el nivel de conocimiento en lenguajes de programación de acuerdo a la edad, se encontró que es mayor el porcentaje para estudiantes mayores de 22 años que tiene mucho conocimiento en lenguajes de programación con respecto a los estudiantes de menor edad.

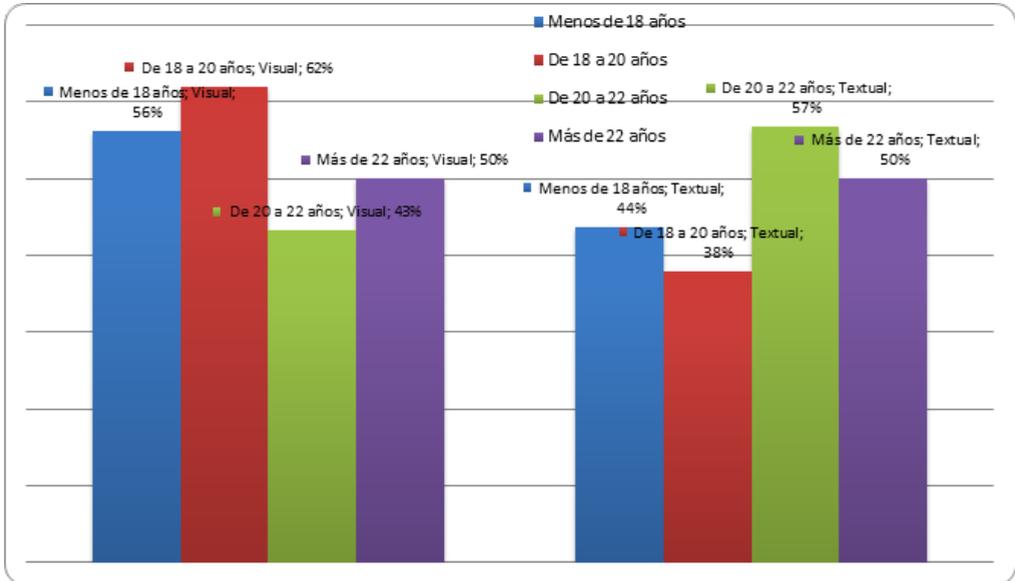
Figura 46. Nivel de conocimiento en lenguajes de programación de encuestados tanto para PiCO como para GraPiCO.



Fuente: elaboración propia (2018).

- La Figura 47 muestra un gráfico de barras comparativo entre el tipo de lenguaje de programación preferido de acuerdo a la edad; se encontró que es mayor el porcentaje para estudiantes menores de 20 años que prefiere el lenguaje de programación visual con respecto al lenguaje de programación textual comparado con los estudiantes mayores de 20 años. Caso contrario con los que prefieren el lenguaje de programación textual con respecto al visual, es mayor el porcentaje para estudiantes mayores de 20 años.

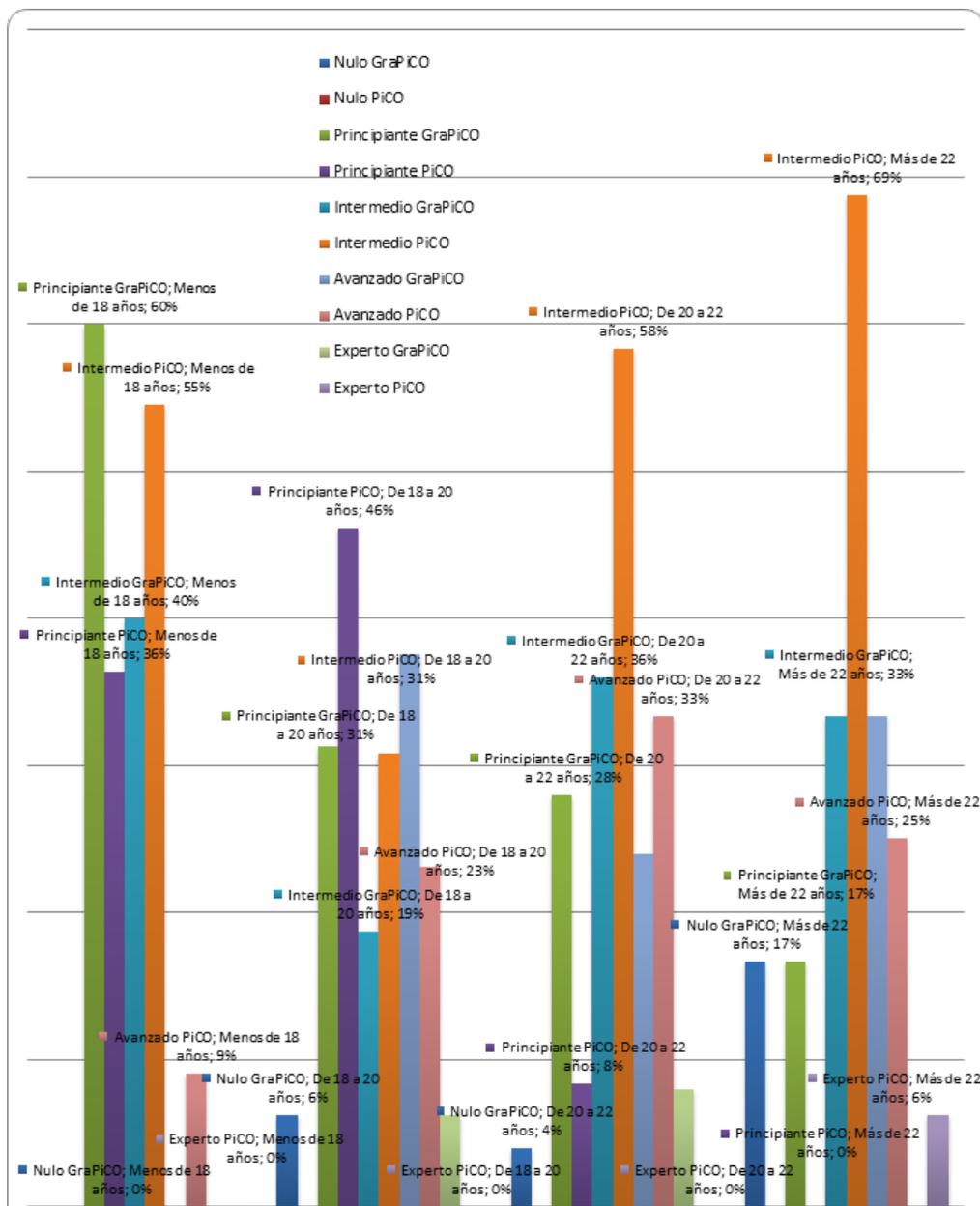
Figura 47. Tipo de lenguaje de programación preferido de encuestados tanto para PiCO como para GraPiCO.



Fuente: elaboración propia (2018).

- Comparando el nivel de conocimiento entre lenguajes de programación visuales y textuales de acuerdo a la edad, se encontró que es mayor el porcentaje para estudiantes menores de 18 años que tiene algo de conocimiento en lenguaje visual GraPiCO con respecto al lenguaje textual PiCO comparado con los estudiantes mayores de 18 años. Para los estudiantes que tienen mucho conocimiento en lenguajes de programación, es mayor el porcentaje para GraPiCO con respecto a PiCO en estudiantes con edades entre 18 y 22 años, comparado con los estudiantes menores de 18 o mayores de 22 años. Ver Figura 37.

Figura 48. Nivel de conocimiento en lenguajes de programación visual y textual diferenciado por edad.



Fuente: elaboración propia (2018).

VI. Conclusión

En este documento se ha presentado un análisis de los resultados obtenidos en el estudio comparativo de lenguajes visuales y textuales: Caso PiCO y GraPiCO. Durante los siguientes pasos a ejecutar para la culminación del experimento, se expondrán las recomendaciones, observaciones y sugerencias obtenidas a partir de la generación de encuestas pensadas como medio de retroalimentación de la experiencia.

Bibliografía

- [1] R. W. Mendenhall, R. J. Beaver, B. M. Beaver, Introducción a la Probabilidad y Estadística. Thomson.
- [2] R. Behar, M. Yepes, Estadística un enfoque descriptivo. Cali: Universidad del Valle.
- [3] R. E. Walpole, R.H. Meyer, Probabilidad y Estadística, Mexico D. F.:Mc Graw Hill, 6a. edición.
- [4] B. Grima, Estadística Aplicada. Cali: Universidad del Valle, Universidad Politécnica de Cataluña.
- [5] V J, M. Hernando, Diseño de encuestas de opinión. Ra-Ma, 2003.
- [6] M Á Cea D´ancona, Métodos de encuesta, teoría y práctica, errores y mejora. Madrid: Editorial Síntesis, 2004
- [7] «Matronas Profesión,» vol. 5, nº 17, 2004.

RECOMENDACIONES Y RESULTADOS DEL ESTUDIO COMPARATIVO ENTRE PiCO, GraPiCO Y LOS EDITORES

*Christian Felipe Cano Castillo
Luis Eduardo Espinosa Galliady
Carlos Andrés Tavera Romero*

I. Introducción

Posterior al análisis de los datos y de su presentación visual por medio de gráficas, se requiere hacer una etapa donde se haga la propuesta de las recomendaciones y el trabajo futuro. En la presente sección se mostrarán las observaciones que se consideraron pertinentes.

II. Marco Teórico

En etapas previas, explicó a la **Hipótesis** y su verificación, las **variables** empleadas, los **sujetos de estudio**, las diferentes asignaciones hechas a las **unidades experimentales** mediante el uso de **tratamientos** y las **observaciones**. También, se presentó la contribución del **Aprendizaje Basado en Problemas** en la investigación y los análisis estadísticos de la información.

En esta etapa, se culminará el proceso de análisis de los resultados y se harán las recomendaciones de PiCO, GraPiCO y sus respectivos editores. Ver Tabla 28.

Tabla 28. *Etapas del estudio comparativo*

Estudio comparativo entre lenguajes textuales y lenguajes visuales. Caso: PiCO y GraPiCO	
Etapa 1	Elaboración de hipótesis en experimentos de lenguajes de programación.
Etapa 2	Variables en un experimento de lenguajes de programación.
Etapa 3	Unidades experimentales utilizadas en pruebas de lenguajes de programación.
Etapa 4	Tratamientos y replicas en un experimento de programación.
Etapa 5	Aprendizaje Basado en Problemas (ABP) aplicado a los lenguajes de programación.
Etapa 6	La comunicación en el estudio comparativo entre lenguajes textuales y lenguajes visuales: Caso PiCO y GraPiCO.
Etapa 7	Sistematización de una experiencia de investigación entre la Comunicación Social y la Ingeniería de Software.
Etapa 8	Modelo de sistematización propuesto "TCACI en doble vía".
Etapa 9	Pasos en la realización de los audiovisuales pedagógicos: PiCO – GraPiCO y ejercicio de modelación.
Etapa 10	Estudio de resultados de PiCO y GraPiCO. Parte 1.
Etapa 11	Estudio de resultados de PiCO y GraPiCO. Parte 2.
Etapa 12	Recomendaciones y resultados del estudio entre PiCO, GraPiCO y editores.

Fuente: elaboración propia (2018).

III. Pautas a tener en cuenta en estudios de lenguajes de programación y editores

La información^[4] es un mensaje que tiene una utilidad específica, la cual mediante un proceso de razonamiento y análisis, se convierte en conocimiento.

El software^[4] es un conjunto de instrucciones normalizadas que convierten los datos (input) en información o acciones (output). Existen dos grandes familias de software:

- Las aplicaciones
- Los sistemas o equipos.

Las aplicaciones, son las que permiten hacer cosas y los sistemas, son los que permiten controlar y ejecutar equipos de cómputo.

En la ingeniería del software, es indispensable validar las diferentes propuestas (métodos, lenguajes, entre otros) para lograr una mayor madurez del software^[1]. Para ello existen tres tipos de estudios empíricos:

- Experimentos controlados
- Casos de estudio
- Encuestas.

Aunque se seleccione alguno de los tres, no se restringe el uso de otro al mismo tiempo.

Ciertas características intrínsecas del estudio a realizar, permitirán evaluar cuales serán los más o menos apropiados de acuerdo a lo que se desea probar, validar y el contexto a trabajar. Lo que se obtenga, permitirá tomar las mejores posturas, recomendaciones y observaciones a la hora de decidir la adopción de nuevas tecnologías.

Cuando se hace un estudio comparativo entre dos lenguajes de programación visual y textual, expresivamente equivalente, hay que tener en cuenta el instrumento de medición escogido para ello, en

este caso fue la encuesta, aunque esta se basaba de la percepción de los asistentes en un experimento controlado (presentación de videos con la descripción del lenguaje y sus respectivos constructores e instrucciones) y un caso de estudio (Taller con un caso de la vida real para modelar en los respectivos lenguajes PiCO y GraPiCO).

La encuesta, es un tipo de estudio empírico que permite capturar información relevante tras haber utilizado una técnica, herramienta o tecnología y poder evaluar su impacto. La encuesta, tiene como principal objetivo el comparar los conocimientos, actitudes y comportamientos principales de las personas que participan, permitiendo obtener información con la intención de describir el efecto de algo: una técnica, un proceso, una herramienta, una tecnología, entre otros. Conviene fijar claramente los resultados que se espera de lo que se desea medir y después de realizarse, analizar los resultados significativos obtenidos.

Lo importante, es no olvidar las etapas anteriores, para que sirva a otros investigadores y puedan replicar o basarse en el conocimiento adquirido durante la realización del experimento y estudio. Por ello, es fundamental comunicar en un congreso, revista, material de laboratorio o como material educativo los hallazgos encontrados.

IV. Visualización de un editor gráfico de programación

La *visualización*,^{[1][5][6]} es un medio que facilita el análisis de datos complejos a través de la exploración visual. Es muy utilizada de manera científica cuando los datos tienen una representación intrínseca; como por ejemplo, la representación visual médica y la simulación mecánica, los cuales tienen objetos reales y concretos para su representación en tecnologías 3D.

En el caso de lenguajes de programación, se habla de *visualización de información*, la cual se caracteriza por no tener representaciones intrínsecas, a causa de la intangibilidad del código o instrucciones de programación, como se muestra en la conferencia

expuesta en [11]. Cuando se utiliza el código o las instrucciones de programación, se encuentran concebidas para ser entendidas por parte de los humanos y los ordenadores, pero no presentan una realidad concreta fuera de estos. La información en este tipo de casos se caracteriza por los atributos gráficos presentes como la forma, color, textura, ubicación, orientación, etc. En el caso de los programas se deben asociar las entidades directas (paquetes, clases, métodos, líneas de código, relaciones, entre otras) y las entidades derivadas (medidas calculadas) con los atributos gráficos para visualizar sus relaciones de manera simultánea.

Uno de los principales limitantes de la *visualización* de un entorno gráfico son las interferencias, las cuales obstruyen la interpretación visual, sesgando el cálculo de los datos reales asociados con el atributo gráfico. Ejemplos de interferencias tenemos: el color frente a la textura, la distancia frente al tamaño, la forma frente a la oclusión, etc.

Otro de los limitantes es la cantidad de entidades que se muestran. Cuando en un entorno gráfico hay demasiados datos y objetos presentes, se puede sobrecargar el sistema visual y se fracasaría en el objetivo. Dado el número de clases en programas grandes, con miles de propiedades, este aspecto es fundamental en la visualización de programas. Para trabajar este aspecto se debe permitir en los editores o aplicaciones de programación visuales una navegación intuitiva, efectiva y en tiempo real, utilizando capas y organizaciones jerárquicas para los datos, y opciones que permitan acercar y alejar la información mostrada en el artículo sobre lenguajes visuales publicada en [12].

V. Marcos de visualización de software

En los editores de programación visual encontramos los *marcos de visualización*^[1], los cuales son contenedores o ventanas que agrupan elementos específicos del lenguaje de programación.

En un *marco de visualización* hay que tener en cuenta cuatro aspectos:

- La representación de entidades.
- La representación del programa.
- La navegación.
- La representación de relaciones.

Teniendo en cuenta que los programas son orientados a objetos, cuando se habla de entidades se entenderán como clases.

Los marcos son fundamentales en la visualización de un software, debido a que nos permiten comprender las propiedades locales y globales del mismo. Cuanto más agradable, eficiente y flexible sea un marco, se podrá analizar, entender y explicar las propiedades de la aplicación.

VI. Utilización de representaciones visuales un entorno gráfico

Cuando se diseña ^{[1][7]} un entorno de visualización en una aplicación, lo principal es seleccionar los *objetos gráficos* que se utilizarán para representar los elementos dentro de un programa en determinado lenguaje. Se aconseja que sea lo más simple posible. La simplicidad es fundamental para la percepción humana. El cerebro analiza una escena, principalmente, a través de la correspondencia rápida de patrones, permitiéndonos reconocer mejor formas comunes (Palmer, 1999). Por ello, las formas básicas como por ejemplo, las líneas rectas, rectángulos y cubos son procesadas muy rápidamente.

La *interfaz de usuario* ^[3] es todo el sistema, es la parte que el usuario ve, trabaja y comunica. El usuario interactúa con la aplicación, para poder realizar una tarea y obtener unos resultados. Así una interfaz pobre origina todo tipo de problemas de productividad, errores y mayor tiempo de aprendizaje. De ahí que la realimentación del usuario no sólo debe darse en los aspectos del diseño de la interfaz, sino también en toda la información ofrecida por el sistema.

VII. Calidad de un producto percibida por un usuario

La *calidad* ^[2] fue definida por Montgomery en su libro [13] como: “Calidad es el grado en que los productos cumplen con las exigencias de la gente que los utiliza,” además él establece dos tipos de calidad:

- La calidad del diseño.
- La calidad de conformidad.

El primer tipo de calidad, es el grado en que un producto posee aquellas características que se tuvieron en cuenta al crearse. Cumple a satisfacción con el objetivo por el que fue diseñado. El otro tipo, la calidad de conformidad, es el grado en que un producto está de acuerdo con la intención del diseño, mas no cumple con todos los objetivos planteados.

VIII. Percepciones y actitudes de los asistentes en una encuesta

La encuesta es la mejor forma de comprender el mundo y de hacer predicciones sobre el mismo. Pero para ello, es fundamental que los datos que se obtengan reflejen una información valida y fiable de lo que se estudia.

Cuando se diseñan los cuestionarios se debe hacer de manera cuidadosa con el fin de detallar las *percepciones* ^{[2][8]} y actitudes de los asistentes al estudio. Los participantes poseen ciertos indicadores visibles, que reflejan su satisfacción en general. Los asistentes pueden sonreír cuando se habla del producto a estudiar o pueden decir cosas buenas. Ambas acciones, son manifestaciones o indicadores de un factor que se puede denominar satisfacción de usuario.

Para la *percepción y actitud del usuario* o asistente se debe tener en cuenta lo siguiente:

- Nunca se sabe el verdadero nivel de satisfacción de nuestros clientes. Se desarrollan deducciones a partir medidas establecidas en los cuestionarios con respecto al factor de la satisfacción.
- Se busca comprender como los factores de la satisfacción del usuario y la sensibilidad percibida se relacionan entre sí.

IX. Recomendaciones obtenidas

E_GraPiCO es una aplicación estable y funcional, que requiere un último conjunto de modificaciones de forma, pero estas recomendaciones de usuario final, están dadas por el estudio comparativo realizado, en el cual se aprecian las ventajas de la utilización de un editor visual ante uno textual.

Muchos estudiantes encuestados tienen un mayor nivel de conocimiento en lenguajes de programación textual, que en lenguajes de programación visual.

Se mostró el interés de muchos estudiantes por aprender más acerca de lenguajes de programación visual. Tendencia que vemos en la actualidad, debido al despliegue tecnológico de herramientas portátiles y móviles de muchos fabricantes, que implican el uso de interfaces y aplicaciones graficas, especialmente basado en íconos, despertando el uso y desarrollo de la programación visual en muchos programadores. Con respecto a lo textual, es algo que no va a desaparecer, porque hasta el momento sigue siendo muy utilizado y da el soporte como infraestructura para que existan los lenguajes visuales, además de tener muchos adeptos en desarrollo de aplicaciones.

La percepción de los asistentes a la encuesta realizada en el caso de PiCO y GraPiCO es:

- Según lo analizado en la información recolectada es muy positiva en cuanto la asimilación de los constructores de lenguajes PiCO y GraPiCO.

- Se manifiesta una mayor comprensión en PiCO que en GraPiCO, y esto se debe en su mayoría a que hay un mayor nivel de intuitividad de PiCO que en GraPiCO.
- En cuanto al diseño de E_GraPiCO, la visualización gustó mucho, pero se manifestó inconformidad con la navegación de la aplicación.
- La interfaz de E_GraPiCO cumple con su objetivo de comunicar e interactuar con los elementos que hacen parte del marco de visualización y las jerarquías establecidas. No se establecieron inconformidades por los íconos utilizados en cada constructor de GraPiCO (representación de entidades, representación de programa), ni tampoco con sus relaciones.
- El lenguaje GraPiCO gusto mucho, aunque las personas encuestadas aceptan, que hacen falta varios controles y mejoras gráficas, pero que por sus antecedentes en conocimientos en otros lenguajes de programación textual, se les facilitaba PiCO.
- Se manifiesta que hay determinados constructores que de acuerdo al manejo visual (ícono), se entendían mejor en GraPiCO, como los constructores Context, Ask, Tell.
- Los constructores que más se dificultaron en cuanto a asimilación y comprensión en ambos lenguajes fueron Ask, Tell, Sender, Forward, MsgSend.
- Según lo analizado en la información obtenida de las encuestas, se tiene un buen interés en PiCO por su estabilidad y modificabilidad, pero se prefiere en diseño y usabilidad de GraPiCO.
- Se tiene mucho interés por el lenguaje visual GraPiCO y su editor, se espera una futura versión con los ajustes anteriormente descritos.
- En cuanto al material audiovisual, los estudiantes manifestaron la importancia de utilizar este tipo de materiales que incluían videos muy estéticos, completos, y didácticos. Por ello, creen que la Comunicación Social va de la mano con

la Ingeniería de Sistemas, especialmente en este tipo de trabajos.

- En cuanto al proyecto y la ejecución del mismo, los estudiantes participantes del mismo, mostraron satisfacción por las competencias y temas desarrollados, aunque se manifestó inconformidad por la duración del mismo.

Analizando las sugerencias y percepciones de los usuarios de la aplicación E_GraPiCO, este producto de software se puede clasificar en el tipo de Calidad de Conformidad, porque aunque gusta, no cumple con todos los requerimientos y objetivos que se han planteado desde su creación.

X. Conclusión.

En este documento se han presentado algunas teorías de cómo analizar u obtener información relevante de estudiar lenguajes de programación y sus respectivos editores, además se plasmaron las recomendaciones, sugerencias y percepciones que se obtuvieron del estudio comparativo entre lenguajes textuales y lenguajes visuales: Caso PiCO y GraPiCO.

Qué un lenguaje programación sea mejor que el otro, es complicado de establecer; además, existen muchos paradigmas que evitan la prevalencia del uno en el otro. Cada uno tiene sus ventajas y desventajas. Lo mejor es buscar y seleccionar el tipo de lenguaje que mejor aplique al caso de estudio que se desee evaluar e implementar. En cuanto a los lenguajes de programación visual, todavía dependen de los lenguajes textuales como plataforma de desarrollo y sostenibilidad. Pero a la velocidad en que los desarrollos de procesadores y de dispositivos tecnológicos avanzan se espera que el auge, desempeño y uso de los lenguajes de programación visual sean mayores y más eficiente.

En un futuro, se espera continuar con el estudio de PiCO y GraPiCO, especialmente en la usabilidad.

XI. Bibliografía

- [1] G. P. Coral Calero, *Calidad del producto y proceso software*, Madrid, España: Ra-Ma, 2010, pp. 117-132, 134-142, 145-153.
- [2] B. E. Hayes, *Cómo medir la satisfacción del cliente*, Barcelona, España: Gestión 2000, 2002, pp. 15-16, 22-24, 64-97.
- [3] C. D. Toni Granollers i Saltiveri, *Diseño de sistemas interactivos centrados en el usuario*, Barcelona, España: Editorial UOC, 2005, pp. 19-32, 51-61, 93-123.
- [4] F. G. d. I. Fuente, *Los sistemas de información en la sociedad del conocimiento*, Pozuelo de Alarcón, Madrid: ESIC Editorial, 2004, pp. 51-54, 60-64.
- [5] S. Diehl, *Software Visualization: Visualizing the Structure, Behaviour, and Evolution of Software*, Germany: Springer, 2007.
- [6] C. Ware, *Information Visualization perception for design*, San Francisco: Morgan Kaufmann, 2004.
- [7] D. Benyon, *Designing Interactive Systems: A Comprehensive Guide to Hci and Interaction Design*, Addison Wesley, 2010.
- [8] G. R. R. A. N. Hill, *Customer Satisfaction: The customer experience through the customer's eyes*, London: Cogent, 2007.
- [9] C. Tavera y J. Díaz, *Nuevo cálculo visual: GraPiCO*, In II Congreso Colombiano de Computación, Universidad Javeriana. Bogotá, 2007.
- [10] C. Tavera y J. Díaz, *Breve Discusión de las Ventajas de los Lenguajes Visuales frente a los Textuales: Caso de Estudio el Cálculo GraPiCO*, En III Congreso Colombiano de Computación. Medellín, 2008.
- [11] C. Knight, M. Munro, *Vistual but Visible Software*, The Proceedings of the IEEE International Conference on Information Visualization, 2000

- [12] O. Köth, M. Minas, Abstraction in Graph-Transformation Based Diagram Editors, *Electronic Notes in Theoretical Computer Science*, Volume 50, Issue 3, August 2001, pages 295-301.
- [13] D.C. Montgomery, Introduction to statistical quiality control, *Quality and Reliability Engineering International*, 1985, volume 2, issue 1.

ACERCA DE LOS AUTORES

Carlos Andrés Tavera Romero

Ingeniero de sistemas de la Universidad del Valle y Doctor en Ingeniería de la Universidad del Valle, profesor de la Universidad Santiago de Cali, líder de la línea de investigación de Desarrollo de Sistemas de Información del Grupo de Investigación COMBA I+D de la Universidad Santiago de Cali.

Correo: *carlos.tavera00@usc.edu.co*

Luis Eduardo Espinosa Galliady

recibió el título de Ingeniero de Sistemas de la Universidad de San Buenaventura Cali, en Cali Colombia en el año 2012.

Correo: *luis467493@gmail.com*

Christian Felipe Cano Castillo

recibió el título de Ingeniero de Sistemas de la Universidad de San Buenaventura Cali, en Cali Colombia en el año 2012.

Yenny Viviana Cruz Pérez

Comunicadora Social-Periodista de la Universidad Autónoma de Occidente, Cali Colombia. Especialista en Desarrollo Comunitario de la Universidad del Valle, Magister en Comunicación de la Universidad Autónoma de Occidente. Directora del Programa de Comunicación Social de la Fundación Universitaria Católica Lumen Gentium.

Correo: *directorcsp@unicatolica.edu.co*

Juan David Penagos Muñoz

Comunicador Social-Periodista de la Universidad Autónoma de Occidente, Cali, Colombia.

Correo: *chojuan9101@hotmail.com*

Paola Andrea Ramírez Arcila

Comunicadora Social-Periodista de la Universidad Autónoma de Occidente, Candidata a Magister en Comunicación de la Universidad Nacional Abierta y a Distancia, Cali, Colombia.

Correo: *paola_and.ramirez@uao.edu.co*

Marco Antonio Triana Lozano

Estadístico de la Universidad del Valle, Magister en Ciencias de la Computación del Instituto Tecnológico de Monterrey México, Profesor de Tiempo Completo de la Universidad Autónoma de Occidente.

Correo: *matriana@uao.edu.co, matriana@gmail.com*

ANEXOS



ENCUESTA PICO

Agradecemos su valiosa atención y colaboración a nuestro proyecto, el cual busca establecer el nivel de aprehensión de los conceptos del Lenguaje Textual PICO.

DATOS PERSONALES

A continuación se realizan preguntas personales y académicas.

* u: Universidad:

Por favor elija una SOLO una de las siguientes

Universidad en la que estudia

- Universidad de San Buenaventura - Cali.
- Universidad Pontificia Javeriana.
- Universidad Autónoma de Occidente.
- Universidad ICESI.

* s: Semestre:

Por favor elija una, SÓLO una de las siguientes

Semestre en el cual se encuentra actualmente estudiando.

- Segundo.
- Tercero
- Cuarto.
- Quinto.

* e: Edad:

Por favor elija una, SÓLO una de las siguientes

Rango de edad en el cual se encuentra usted.

- Menos de 18 años.
- De 18 a 20 años.
- De 20 a 22 años.
- Más de 22 años.

* se: Sexo:

Por favor elija una, SÓLO una de las siguientes

- Femenino.
- Masculino.

CONOCIMIENTO GENERAL

Esta sección es acerca del conocimiento y preferencia en lenguajes de programación.

* nclp: ¿Cuál considera que es su nivel de conocimiento en lenguajes de programación?

Por favor elija una, SÓLO una de las siguientes

- 1-Nulo.
- 2-Principiante.
- 3-Intermedio.
- 4-Avanzado.
- 5-Experto.

* tlp: ¿Qué tipo de lenguaje de programación prefiere?

Por favor elija una, SÓLO una de las siguientes

- | | |
|--------------------------|----------|
| <input type="checkbox"/> | Visual. |
| <input type="checkbox"/> | Textual. |

CONOCIMIENTO EN LENGUAJES VISUALES

Esta sección es acerca del conocimiento, uso e interés en lenguajes de programación visuales.

* nclt: ¿Cuál considera que es su nivel de conocimiento en lenguajes de programación textual?

Por favor elija una, SÓLO una de las siguientes

- | | |
|--------------------------|-----------------|
| <input type="checkbox"/> | 1-Nulo. |
| <input type="checkbox"/> | 2-Principiante. |
| <input type="checkbox"/> | 3-Intermedio. |
| <input type="checkbox"/> | 4-Avanzado. |
| <input type="checkbox"/> | 5-Experto. |

* fult: ¿Con qué frecuencia ha utilizado aplicaciones (Software) que de alguna forma usen lenguajes de programación textuales?

Por favor elija una, SÓLO una de las siguientes

- | | |
|--------------------------|---------------|
| <input type="checkbox"/> | 1-Nada. |
| <input type="checkbox"/> | 2-Poco. |
| <input type="checkbox"/> | 3-Intermedio. |
| <input type="checkbox"/> | 4-Suficiente. |
| <input type="checkbox"/> | 5-Mucho. |

* ilt: Califique su interés en lenguajes de programación textuales.

Por favor elija una, SÓLO una de las siguientes

- | | |
|--------------------------|---------------|
| <input type="checkbox"/> | 1-Nada. |
| <input type="checkbox"/> | 2-Poco. |
| <input type="checkbox"/> | 3-Intermedio. |
| <input type="checkbox"/> | 4-Suficiente. |
| <input type="checkbox"/> | 5-Mucho. |

PiCO

Esta sección es acerca de PiCO

* cp: Califique su nivel de comprensión del constructor "Program" del lenguaje de programación textual PiCO.

Por favor elija una, SÓLO una de las siguientes	
clone local nombre in (...)	<input type="checkbox"/> 1-Nulo.
	<input type="checkbox"/> 2-Principiante.
	<input type="checkbox"/> 3-Intermedio.
	<input type="checkbox"/> 4-Avanzado.
	<input type="checkbox"/> 5-Experto.
* cc: Califique su nivel de comprensión del constructor “Contex” del lenguaje de programación textual PiCO.	
Por favor elija una, SOLO una de las siguientes	
local	<input type="checkbox"/> 1-Nulo.
	<input type="checkbox"/> 2-Principiante.
	<input type="checkbox"/> 3-Intermedio.
	<input type="checkbox"/> 4-Avanzado.
	<input type="checkbox"/> 5-Experto.

* co: Califique su nivel de comprensión del constructor “Objects” del lenguaje de programación textual PiCO.	
Por favor elija una, SÓLO una de las siguientes	
Object[...]	<input type="checkbox"/> 1-Nulo.
	<input type="checkbox"/> 2-Principiante.
	<input type="checkbox"/> 3-Intermedio.
	<input type="checkbox"/> 4-Avanzado.
	<input type="checkbox"/> 5-Experto.
* cm: Califique su nivel de comprensión del constructor “Methods” del lenguaje de programación textual PiCO.	
Por favor elija una, SÓLO una de las siguientes	
method	<input type="checkbox"/> 1-Nulo.
	<input type="checkbox"/> 2-Principiante.
	<input type="checkbox"/> 3-Intermedio.
	<input type="checkbox"/> 4-Avanzado.
	<input type="checkbox"/> 5-Experto.
* ca: Califique su nivel de comprensión del constructor “Ask” del lenguaje de programación textual PiCO.	

		Por favor elija una, SÓLO una de las siguientes
ask	<input type="checkbox"/>	1-Nulo.
	<input type="checkbox"/>	2-Principiante.
	<input type="checkbox"/>	3-Intermedio.
	<input type="checkbox"/>	4-Avanzado.
	<input type="checkbox"/>	5-Experto.
<p>* ct: Califique su nivel de comprensión del constructor "Tell" del lenguaje de programación textual PiCO.</p> <p>Por favor elija una, SÓLO una de las siguientes</p>		
tell	<input type="checkbox"/>	1-Nulo.
	<input type="checkbox"/>	2-Principiante.
	<input type="checkbox"/>	3-Intermedio.
	<input type="checkbox"/>	4-Avanzado.
	<input type="checkbox"/>	5-Experto.
<p>* cms: Califique su nivel de comprensión del constructor "MsgSend" del lenguaje de programación textual PiCO.</p> <p>Por favor elija una, SÓLO una de las siguientes</p>		
msgsend	<input type="checkbox"/>	1-Nulo.
	<input type="checkbox"/>	2-Principiante.
	<input type="checkbox"/>	3-Intermedio.
	<input type="checkbox"/>	4-Avanzado.
	<input type="checkbox"/>	5-Experto.

		* cv: Califique su nivel de comprensión del constructor "Value" del lenguaje de programación textual PiCO.
		Por favor elija una, SÓLO una de las siguientes
Z=10	<input type="checkbox"/>	1-Nulo.
	<input type="checkbox"/>	2-Principiante.
	<input type="checkbox"/>	3-Intermedio.
	<input type="checkbox"/>	4-Avanzado.
	<input type="checkbox"/>	5-Experto.
		* cva: Califique su nivel de comprensión del constructor "Variable" del lenguaje de programación textual PiCO.
		Por favor elija una, SÓLO una de las siguientes

variable	<input type="checkbox"/>	1-Nulo.
	<input type="checkbox"/>	2-Principiante.
	<input type="checkbox"/>	3-Intermedio.
	<input type="checkbox"/>	4-Avanzado.
	<input type="checkbox"/>	5-Experto.
<p>* car: Califique su nivel de comprensión del constructor "Argument" del lenguaje de programación textual PiCO.</p> <p>Por favor elija una, SÓLO una de las siguientes</p>		
[...]	<input type="checkbox"/>	1-Nulo.
	<input type="checkbox"/>	2-Principiante.
	<input type="checkbox"/>	3-Intermedio.
	<input type="checkbox"/>	4-Avanzado.
	<input type="checkbox"/>	5-Experto.
<p>* cse: Califique su nivel de comprensión del constructor "Sender" del lenguaje de programación textual PiCO.</p> <p>Por favor elija una, SÓLO una de las siguientes</p>		
sender	<input type="checkbox"/>	1-Nulo.
	<input type="checkbox"/>	2-Principiante.
	<input type="checkbox"/>	3-Intermedio.
	<input type="checkbox"/>	4-Avanzado.
	<input type="checkbox"/>	5-Experto.
<p>* cfo: Califique su nivel de comprensión del constructor "Forward" del lenguaje de programación textual PiCO.</p> <p>Por favor elija una, SÓLO una de las siguientes</p>		
forward	<input type="checkbox"/>	1-Nulo.
	<input type="checkbox"/>	2-Principiante.
	<input type="checkbox"/>	3-Intermedio.
	<input type="checkbox"/>	4-Avanzado.
	<input type="checkbox"/>	5-Experto.

<p>* cop: Califique su nivel de comprensión del constructor "Operators" del lenguaje de programación textual PiCO.</p>	
<p>Por favor elija una, SÓLO una de las siguientes</p>	
<p>+,-,*,/</p>	<input type="checkbox"/> 1-Nulo.
	<input type="checkbox"/> 2-Principiante.
	<input type="checkbox"/> 3-Intermedio.
	<input type="checkbox"/> 4-Avanzado.
	<input type="checkbox"/> 5-Experto.
<p>* crel: Califique su nivel de comprensión del constructor "Relations" del lenguaje de programación textual PiCO.</p>	
<p>Por favor elija una, SÓLO una de las siguientes</p>	
<p>>,>=,<,<=,=,<></p>	<input type="checkbox"/> 1-Nulo.
	<input type="checkbox"/> 2-Principiante.
	<input type="checkbox"/> 3-Intermedio.
	<input type="checkbox"/> 4-Avanzado.
	<input type="checkbox"/> 5-Experto.
<p>* ccon: Califique su nivel de comprensión del constructor "Constraint" del lenguaje de programación textual PiCO.</p>	
<p>Por favor elija una, SÓLO una de las siguientes</p>	
<p>(sender=20, forward=30)</p>	<input type="checkbox"/> 1-Nulo.
	<input type="checkbox"/> 2-Principiante.
	<input type="checkbox"/> 3-Intermedio.
	<input type="checkbox"/> 4-Avanzado.
	<input type="checkbox"/> 5-Experto.
<p>* ccla: Califique la claridad de los constructores del lenguaje textual PiCO anteriormente especificados.</p>	
<p>Por favor elija una, SÓLO una de las siguientes</p>	
<p>Claridad: Fácil interpretación de las características y propiedades de los constructores.</p>	<input type="checkbox"/> 1-Nada.
	<input type="checkbox"/> 2-Poco.
	<input type="checkbox"/> 3-Intermedio.
	<input type="checkbox"/> 4-Suficiente.
	<input type="checkbox"/> 5-Mucho.
<p>* csim: Califique la simbología de los constructores del lenguaje textual PiCO anteriormente especificados.</p>	
<p>Por favor elija una, SÓLO una de las siguientes</p>	

Simbología: Modelación de los parámetros mediante iconos cercanos al mundo real.	<input type="checkbox"/>	1-Nada.
	<input type="checkbox"/>	2-Poco.
	<input type="checkbox"/>	3-Intermedio.
	<input type="checkbox"/>	4-Suficiente.
	<input type="checkbox"/>	5-Mucho.

* cnav: Califique la navegación de los constructores del lenguaje textual PICO anteriormente especificados.

Por favor elija una, SÓLO una de las siguientes

Navegación: La navegación se refiere a cómo se busca la información en las pantallas que se ven; el número de clicks, las palabras clave, el color de los vínculos, los iconos, la velocidad de despliegue para llegar a otros contextos.	<input type="checkbox"/>	1-Nada.
	<input type="checkbox"/>	2-Poco.
	<input type="checkbox"/>	3-Intermedio.
	<input type="checkbox"/>	4-Suficiente.
	<input type="checkbox"/>	5-Mucho.

* iest: Califique el nivel de interés de la aplicación E_GraPiCO de acuerdo a su estabilidad.

Por favor elija una, SÓLO una de las siguientes

Estabilidad: Un sistema es estable cuando su nivel de inducción a fallos es muy pequeño o no existe, debido al alto manejo de controles de la aplicación.	<input type="checkbox"/>	1-Nada.
	<input type="checkbox"/>	2-Poco.
	<input type="checkbox"/>	3-Intermedio.
	<input type="checkbox"/>	4-Suficiente.
	<input type="checkbox"/>	5-Mucho.

* idis: Califique el nivel de interés de la aplicación E_GraPiCO de acuerdo a su diseño.

Por favor elija una, SÓLO una de las siguientes

Diseño: Componentes, módulos, interfaces, procedimientos de prueba y datos de un sistema, que se crean para satisfacer unos requisitos específicos y que ayuda al usuario final a comprender los contenidos informativos con una presentación cómoda, eficaz y atractiva. Constructores, Ventanas, Menús, Símbolos, Signos y Colores de la Aplicación.	<input type="checkbox"/>	1-Nada.
	<input type="checkbox"/>	2-Poco.
	<input type="checkbox"/>	3-Intermedio.
	<input type="checkbox"/>	4-Suficiente.
	<input type="checkbox"/>	5-Mucho.

ENCUESTA GraPiCO

* imod: Califique el nivel de interés de la aplicación E_GraPiCO, de acuerdo a su modificabilidad.

Por favor elija una, SÓLO una de las siguientes

Modificabilidad: Un elemento de programación puede ser un elemento modificable, cuando su valor se puede modificar, o un elemento no modificable, cuando su valor se mantiene fijo una vez que se ha creado.

- 1-Nada.
- 2-Poco.
- 3-Intermedio.
- 4-Suficiente.
- 5-Mucho.

* iusa: Califique el nivel de interés de la aplicación E_GraPiCO de acuerdo a su modificabilidad.

Por favor elija una, SÓLO una de las siguientes

Usabilidad: La usabilidad es la facilidad con que las personas pueden utilizar una herramienta particular con el fin de alcanzar un objetivo concreto.

- 1-Nada.
- 2-Poco.
- 3-Intermedio.
- 4-Suficiente.
- 5-Mucho.

COMUNICACIÓN

Esta parte permite a los comunicadores evaluar la experiencia.

* com1: ¿Considera usted que la Ingeniería de Sistemas y la Comunicación Social se complementan, en el caso específico del material audiovisual expuesto para explicar conceptos técnicos del cálculo de programación GraPiCO?

Por favor elija una, SÓLO una de las siguientes

- Si.
- No.

* com2: Semestre: ¿Utilizar un material audiovisual facilitó el entendimiento del cálculo de programación GraPiCO?

Por favor elija una, SÓLO una de las siguientes

- Mucho.
- Medianamente.
- Poco.
- Nada.

* com3: La dinámica audiovisual expuesta: imágenes, efectos, sonorización, presentadora, ¿es estética?

Por favor elija una, SÓLO una de las siguientes

- Si.
- No.

Agradecemos su valiosa atención y colaboración a nuestro proyecto, el cual busca establecer el nivel de aprehensión de los conceptos del Lenguaje Visual GraPiCO.

Datos Personales

A continuación se realizan preguntas personales y académicas.

* u: Universidad:

Por favor elija una SOLO una de las siguientes

Universidad en la que estudia

Universidad de San Buenaventura - Cali.

Universidad Pontificia Javeriana.

Universidad Autónoma de Occidente.

Universidad ICESI.

* s: Semestre:

Por favor elija una, SÓLO una de las siguientes

Semestre en el cual se encuentra actualmente estudiando.

Segundo.

Tercero

Cuarto.

Quinto.

* e: Edad:

Por favor elija una, SÓLO una de las siguientes

Rango de edad en el cual se encuentra usted.

Menos de 18 años.

De 18 a 20 años.

De 20 a 22 años.

Más de 22 años.

* se: Sexo:

Por favor elija una, SÓLO una de las siguientes

Femenino.

Masculino.

Conocimiento General

Esta sección es acerca del conocimiento y preferencia en lenguajes de programación.

* nclp: ¿Cuál considera que es su nivel de conocimiento en lenguajes de programación?

Por favor elija una, SÓLO una de las siguientes

1-Nulo.

2-Principiante.

3-Intermedio.

4-Avanzado.

5-Experto.

* tlp: ¿Qué tipo de lenguaje de programación prefiere?

Por favor elija una, SÓLO una de las siguientes

- | | |
|--------------------------|----------|
| <input type="checkbox"/> | Visual. |
| <input type="checkbox"/> | Textual. |

Conocimiento en Lenguajes Visuales.

Esta sección es acerca del conocimiento, uso e interés en lenguajes de programación visuales.

* nclt: ¿Cuál considera que es su nivel de conocimiento en lenguajes de programación visual?

Por favor elija una, SÓLO una de las siguientes

- | | |
|--------------------------|-----------------|
| <input type="checkbox"/> | 1-Nulo. |
| <input type="checkbox"/> | 2-Principiante. |
| <input type="checkbox"/> | 3-Intermedio. |
| <input type="checkbox"/> | 4-Avanzado. |
| <input type="checkbox"/> | 5-Experto. |

* fult: ¿Con que frecuencia ha utilizado aplicaciones (Software) que de alguna forma usen lenguajes de programación visuales?

Por favor elija una, SÓLO una de las siguientes

- | | |
|--------------------------|---------------|
| <input type="checkbox"/> | 1-Nada. |
| <input type="checkbox"/> | 2-Poco. |
| <input type="checkbox"/> | 3-Intermedio. |
| <input type="checkbox"/> | 4-Suficiente. |
| <input type="checkbox"/> | 5-Mucho. |

* ilt: Califique su interés en lenguajes de programación visuales.

Por favor elija una, SÓLO una de las siguientes

- | | |
|--------------------------|---------------|
| <input type="checkbox"/> | 1-Nada. |
| <input type="checkbox"/> | 2-Poco. |
| <input type="checkbox"/> | 3-Intermedio. |
| <input type="checkbox"/> | 4-Suficiente. |
| <input type="checkbox"/> | 5-Mucho. |

GraPiCO	
Esta sección es acerca de GraPiCO	
<p>* cp: Califique su nivel de comprensión del constructor "Program" del lenguaje de programación visual GraPiCO.</p>	
Por favor elija una, SÓLO una de las siguientes	
<input type="checkbox"/>	1-Nulo.
<input type="checkbox"/>	2-Principiante.
<input type="checkbox"/>	3-Intermedio.
<input type="checkbox"/>	4-Avanzado.
<input type="checkbox"/>	5-Experto.
<p>* cc: Califique su nivel de comprensión del constructor "Contex" del lenguaje de programación visual GraPiCO.</p>	
Por favor elija una, SÓLO una de las siguientes	
<input type="checkbox"/>	1-Nulo.
<input type="checkbox"/>	2-Principiante.
<input type="checkbox"/>	3-Intermedio.
<input type="checkbox"/>	4-Avanzado.
<input type="checkbox"/>	5-Experto.

<p>* co: Califique su nivel de comprensión del constructor "Objects" del lenguaje de programación visual GraPiCO.</p>	
Por favor elija una, SÓLO una de las siguientes	
<input type="checkbox"/>	1-Nulo.
<input type="checkbox"/>	2-Principiante.
<input type="checkbox"/>	3-Intermedio.
<input type="checkbox"/>	4-Avanzado.
<input type="checkbox"/>	5-Experto.
<p>* cm: Califique su nivel de comprensión del constructor "Methods" del lenguaje de programación visual GraPiCO.</p>	
Por favor elija una, SÓLO una de las siguientes	
<input type="checkbox"/>	1-Nulo.
<input type="checkbox"/>	2-Principiante.
<input type="checkbox"/>	3-Intermedio.
<input type="checkbox"/>	4-Avanzado.
<input type="checkbox"/>	5-Experto.

* ca: Califique su nivel de comprensión del constructor "Ask" del lenguaje de programación visual GraPiCO.

Por favor elija una, SÓLO una de las siguientes

- | | |
|--------------------------|-----------------|
| <input type="checkbox"/> | 1-Nulo. |
| <input type="checkbox"/> | 2-Principiante. |
| <input type="checkbox"/> | 3-Intermedio. |
| <input type="checkbox"/> | 4-Avanzado. |
| <input type="checkbox"/> | 5-Experto. |

* ct: Califique su nivel de comprensión del constructor "Tell" del lenguaje de programación visual GraPiCO.

Por favor elija una, SÓLO una de las siguientes

- | | |
|--------------------------|-----------------|
| <input type="checkbox"/> | 1-Nulo. |
| <input type="checkbox"/> | 2-Principiante. |
| <input type="checkbox"/> | 3-Intermedio. |
| <input type="checkbox"/> | 4-Avanzado. |
| <input type="checkbox"/> | 5-Experto. |

* cms: Califique su nivel de comprensión del constructor "MsgSend" del lenguaje de programación visual GraPiCO.

Por favor elija una, SÓLO una de las siguientes

- | | |
|--------------------------|-----------------|
| <input type="checkbox"/> | 1-Nulo. |
| <input type="checkbox"/> | 2-Principiante. |
| <input type="checkbox"/> | 3-Intermedio. |
| <input type="checkbox"/> | 4-Avanzado. |
| <input type="checkbox"/> | 5-Experto. |

* cv: Califique su nivel de comprensión del constructor "Value" del lenguaje de programación visual GraPiCO.

Por favor elija una, SÓLO una de las siguientes

- | | |
|--------------------------|-----------------|
| <input type="checkbox"/> | 1-Nulo. |
| <input type="checkbox"/> | 2-Principiante. |
| <input type="checkbox"/> | 3-Intermedio. |
| <input type="checkbox"/> | 4-Avanzado. |
| <input type="checkbox"/> | 5-Experto. |

* cva: Califique su nivel de comprensión del constructor "Variable" del lenguaje de programación visual GraPiCO.

Por favor elija una, SÓLO una de las siguientes

	<table border="1"> <tr><td><input type="checkbox"/></td><td>1-Nulo.</td></tr> <tr><td><input type="checkbox"/></td><td>2-Principiante.</td></tr> <tr><td><input type="checkbox"/></td><td>3-Intermedio.</td></tr> <tr><td><input type="checkbox"/></td><td>4-Avanzado.</td></tr> <tr><td><input type="checkbox"/></td><td>5-Experto.</td></tr> </table>	<input type="checkbox"/>	1-Nulo.	<input type="checkbox"/>	2-Principiante.	<input type="checkbox"/>	3-Intermedio.	<input type="checkbox"/>	4-Avanzado.	<input type="checkbox"/>	5-Experto.
<input type="checkbox"/>	1-Nulo.										
<input type="checkbox"/>	2-Principiante.										
<input type="checkbox"/>	3-Intermedio.										
<input type="checkbox"/>	4-Avanzado.										
<input type="checkbox"/>	5-Experto.										
<p>* car: Califique su nivel de comprensión del constructor "Argument" del lenguaje de programación visual GraPiCO.</p> <p>Por favor elija una, SÓLO una de las siguientes</p> <table border="1"> <tr><td><input type="checkbox"/></td><td>1-Nulo.</td></tr> <tr><td><input type="checkbox"/></td><td>2-Principiante.</td></tr> <tr><td><input type="checkbox"/></td><td>3-Intermedio.</td></tr> <tr><td><input type="checkbox"/></td><td>4-Avanzado.</td></tr> <tr><td><input type="checkbox"/></td><td>5-Experto.</td></tr> </table>		<input type="checkbox"/>	1-Nulo.	<input type="checkbox"/>	2-Principiante.	<input type="checkbox"/>	3-Intermedio.	<input type="checkbox"/>	4-Avanzado.	<input type="checkbox"/>	5-Experto.
<input type="checkbox"/>	1-Nulo.										
<input type="checkbox"/>	2-Principiante.										
<input type="checkbox"/>	3-Intermedio.										
<input type="checkbox"/>	4-Avanzado.										
<input type="checkbox"/>	5-Experto.										
<p>* cse: Califique su nivel de comprensión del constructor "Sender" del lenguaje de programación visual GraPiCO.</p> <p>Por favor elija una, SÓLO una de las siguientes</p> <table border="1"> <tr><td><input type="checkbox"/></td><td>1-Nulo.</td></tr> <tr><td><input type="checkbox"/></td><td>2-Principiante.</td></tr> <tr><td><input type="checkbox"/></td><td>3-Intermedio.</td></tr> <tr><td><input type="checkbox"/></td><td>4-Avanzado.</td></tr> <tr><td><input type="checkbox"/></td><td>5-Experto.</td></tr> </table>		<input type="checkbox"/>	1-Nulo.	<input type="checkbox"/>	2-Principiante.	<input type="checkbox"/>	3-Intermedio.	<input type="checkbox"/>	4-Avanzado.	<input type="checkbox"/>	5-Experto.
<input type="checkbox"/>	1-Nulo.										
<input type="checkbox"/>	2-Principiante.										
<input type="checkbox"/>	3-Intermedio.										
<input type="checkbox"/>	4-Avanzado.										
<input type="checkbox"/>	5-Experto.										
<p>* cfo: Califique su nivel de comprensión del constructor "Forward" del lenguaje de programación visual GraPiCO.</p> <p>Por favor elija una, SÓLO una de las siguientes</p> <table border="1"> <tr><td><input type="checkbox"/></td><td>1-Nulo.</td></tr> <tr><td><input type="checkbox"/></td><td>2-Principiante.</td></tr> <tr><td><input type="checkbox"/></td><td>3-Intermedio.</td></tr> <tr><td><input type="checkbox"/></td><td>4-Avanzado.</td></tr> <tr><td><input type="checkbox"/></td><td>5-Experto.</td></tr> </table>		<input type="checkbox"/>	1-Nulo.	<input type="checkbox"/>	2-Principiante.	<input type="checkbox"/>	3-Intermedio.	<input type="checkbox"/>	4-Avanzado.	<input type="checkbox"/>	5-Experto.
<input type="checkbox"/>	1-Nulo.										
<input type="checkbox"/>	2-Principiante.										
<input type="checkbox"/>	3-Intermedio.										
<input type="checkbox"/>	4-Avanzado.										
<input type="checkbox"/>	5-Experto.										

<p>* cop: Califique su nivel de comprensión del constructor "Operators" del lenguaje de programación visual GraPiCO.</p>		
<p>Por favor elija una, SÓLO una de las siguientes</p>		
<input type="checkbox"/>	1-Nulo.	
<input type="checkbox"/>	2-Principiante.	
<input type="checkbox"/>	3-Intermedio.	
<input type="checkbox"/>	4-Avanzado.	
<input type="checkbox"/>	5-Experto.	
<p>* crel: Califique su nivel de comprensión del constructor "Relations" del lenguaje de programación visual GraPiCO.</p>		
<p>Por favor elija una, SÓLO una de las siguientes</p>		
<input type="checkbox"/>	1-Nulo.	
<input type="checkbox"/>	2-Principiante.	
<input type="checkbox"/>	3-Intermedio.	
<input type="checkbox"/>	4-Avanzado.	
<input type="checkbox"/>	5-Experto.	
<p>* ccon: Califique su nivel de comprensión del constructor "Constraint" del lenguaje de programación visual GraPiCO.</p>		
<p>Por favor elija una, SÓLO una de las siguientes</p>		
<input type="checkbox"/>	1-Nulo.	
<input type="checkbox"/>	2-Principiante.	
<input type="checkbox"/>	3-Intermedio.	
<input type="checkbox"/>	4-Avanzado.	
<input type="checkbox"/>	5-Experto.	
<p>* ccla: Califique la claridad de los constructores del lenguaje visual GraPiCO anteriormente especificados.</p>		
<p>Por favor elija una, SÓLO una de las siguientes</p>		
<p>Claridad: Fácil interpretación de las características y propiedades de los constructores.</p>	<input type="checkbox"/>	1-Nada.
	<input type="checkbox"/>	2-Poco.
	<input type="checkbox"/>	3-Intermedio.
	<input type="checkbox"/>	4-Suficiente.
	<input type="checkbox"/>	5-Mucho.
<p>* csim: Califique la simbología de los constructores del lenguaje visual GraPiCO anteriormente especificados.</p>		
<p>Por favor elija una, SÓLO una de las siguientes</p>		

Simbología: Modelación de los parámetros mediante íconos cercanos al mundo real.	<input type="checkbox"/>	1-Nada.
	<input type="checkbox"/>	2-Poco.
	<input type="checkbox"/>	3-Intermedio.
	<input type="checkbox"/>	4-Suficiente.
	<input type="checkbox"/>	5-Mucho.

* cnav: Califique la navegación de los constructores del lenguaje visual GraPiCO anteriormente especificados.

Por favor elija una, SÓLO una de las siguientes

Navegación: La navegación se refiere a cómo se busca la información en las pantallas que se ven; el número de clicks, las palabras clave, el color de los vínculos, los íconos, la velocidad de despliegue para llegar a otros contextos.	<input type="checkbox"/>	1-Nada.
	<input type="checkbox"/>	2-Poco.
	<input type="checkbox"/>	3-Intermedio.
	<input type="checkbox"/>	4-Suficiente.
	<input type="checkbox"/>	5-Mucho.

* iest: Califique el nivel de interés de la aplicación E_GraPiCO de acuerdo a su estabilidad.

Por favor elija una, SÓLO una de las siguientes

Estabilidad: Un sistema es estable cuando su nivel de inducción a fallos es muy pequeño o no existe, debido al alto manejo de controles de la aplicación.	<input type="checkbox"/>	1-Nada.
	<input type="checkbox"/>	2-Poco.
	<input type="checkbox"/>	3-Intermedio.
	<input type="checkbox"/>	4-Suficiente.
	<input type="checkbox"/>	5-Mucho.

* idis: Califique el nivel de interés de la aplicación E_GraPiCO de acuerdo a su diseño.

Por favor elija una, SÓLO una de las siguientes

Diseño: Componentes, módulos, interfaces, procedimientos de prueba y datos de un sistema, que se crean para satisfacer unos requisitos específicos y que ayuda al usuario final a comprender los contenidos informativos con una presentación cómoda, eficaz y atractiva. Constructores, Ventanas, Menús, Símbolos, Signos y Colores de la Aplicación.	<input type="checkbox"/>	1-Nada.
	<input type="checkbox"/>	2-Poco.
	<input type="checkbox"/>	3-Intermedio.
	<input type="checkbox"/>	4-Suficiente.
	<input type="checkbox"/>	5-Mucho.

<p>* imod: Califique el nivel de interés de la aplicación E_GraPiCO de acuerdo a su modificabilidad.</p>	
<p>Por favor elija una, SÓLO una de las siguientes</p>	
<p>Modificabilidad: Un elemento de programación puede ser un elemento modificable, cuando su valor se puede modificar, o un elemento no modificable, cuando su valor se mantiene fijo una vez que se ha creado.</p>	<input type="checkbox"/> 1-Nada.
	<input type="checkbox"/> 2-Poco.
	<input type="checkbox"/> 3-Intermedio.
	<input type="checkbox"/> 4-Suficiente.
	<input type="checkbox"/> 5-Mucho.
<p>* iusa: Califique el nivel de interés de la aplicación E_GraPiCO de acuerdo a su modificabilidad.</p>	
<p>Por favor elija una, SÓLO una de las siguientes</p>	
<p>Usabilidad: La usabilidad es la facilidad con que las personas pueden utilizar una herramienta particular con el fin de alcanzar un objetivo concreto.</p>	<input type="checkbox"/> 1-Nada.
	<input type="checkbox"/> 2-Poco.
	<input type="checkbox"/> 3-Intermedio.
	<input type="checkbox"/> 4-Suficiente.
	<input type="checkbox"/> 5-Mucho.
<p>Comunicación</p>	
<p>Esta parte permite a los c comunicadores evaluar la experiencia.</p>	
<p>* com1: ¿Considera usted que la Ingeniería de Sistemas y la Comunicación Social se complementan, en el caso específico del material audiovisual expuesto para explicar conceptos técnicos del cálculo de programación GraPiCO?</p>	
<p>Por favor elija una, SÓLO una de las siguientes</p>	
<input type="checkbox"/> Si.	
<input type="checkbox"/> No.	
<p>* com2: Semestre: ¿Utilizar un material audiovisual facilitó el entendimiento del cálculo de programación GraPiCO?</p>	
<p>Por favor elija una, SÓLO una de las siguientes</p>	
<input type="checkbox"/> Mucho.	
<input type="checkbox"/> Medianamente.	
<input type="checkbox"/> Poco.	
<input type="checkbox"/> Nada.	

* com3: La dinámica audiovisual expuesta: imágenes, efectos, sonorización, presentadora, ¿es estética?

Por favor elija una, SÓLO una de las siguientes

- Si.
 No.

* com5: ¿Cree usted que el video dramatizado facilitó el entendimiento del ejercicio de modelación en el taller?

Por favor elija una, SÓLO una de las siguientes

- Mucho.
 Medianamente.
 Poco.
 Nada.

RETROALIMENTACIÓN

Esta sección es acerca de la retroalimentación de la experiencia.

* obs: Observaciones

Por favor, escriba su respuesta aquí:

Enviar su cuestionario

Gracias por completar este cuestionario

Nota: Este formato de la encuesta de GraPiCO, es el obtenido de exportar las preguntas desde la aplicación LimeySurvey. Dicha aplicación, permitió implementar la encuesta en un servidor web y utilizarla vía correo electrónico con los participantes del experimento.

PARES EVALUADORES

Enrique Pardo Pérez

Universidad de Córdoba

Orcid: <https://orcid.org/0000-0002-6467-5790>

Edisson Duarte Restrepo

Universidad de Cartagena

Adriana Correa Bermúdez

Corporación Centro Internacional de Entrenamiento e
Investigaciones médicas CIDEIM

Alexander Luna Nieto

Fundación Universitaria de Popayán

Orcid: <https://orcid.org/0000-0002-9297-8043>

Alexander López Orozco

Universidad de San Buenaventura

Orcid: <https://orcid.org/0000-0003-0068-6252>

Carlos Andrés Rodríguez Torijano

Universidad de los Andes

Orcid: <https://orcid.org/0000-0002-0401-9783>

Carlos David Grande Tovar

Universidad del Atlántico

Orcid: <https://orcid.org/0000-0002-6243-4571>

Ingrid Paola Cortes Pardo

Pontificia Universidad Javeriana

Orcid: <https://orcid.org/0000-0003-0282-0259>

Jean Jader Orejarena Torres

Universidad Autónoma de Occidente

Orcid: <https://orcid.org/0000-0003-0401-3143>

John James Gómez Gallego

Universidad Católica de Pereira

Orcid: <https://orcid.org/0000-0001-6685-7099>

Juan Manuel Rubio Vera

Servicio Nacional de Aprendizaje Sena

Orcid: <https://orcid.org/0000-0003-1281-8750>

Margaret Mejía Genez

Universidad de Guanajuato

Orcid: <https://orcid.org/0000-0002-5142-5813>

María Alexandra Rendón Uribe

Universidad de Antioquia

Orcid: <https://orcid.org/0000-0002-1062-6125>

Willian Fredy Palta Velasco

Universidad de San Buenaventura

Orcid: <https://orcid.org/0000-0003-1888-0416>

Yenny Patricia Ávila Torres

Universidad Tecnológica de Pereira

Orcid: <https://orcid.org/0000-0003-1399-7922>

Diana Milena Díaz Vidal

Universidad de San Buenaventura

Orcid: <https://orcid.org/0000-0002-6428-8272>

Marco Antonio Chaves García

Fundación Universitaria María Cano

Orcid: <https://orcid.org/0000-0001-7226-4767>

Nelson Jair Cuchumbé Holguín

Universidad del Valle

Orcid: <https://orcid.org/0000-0002-9435-9289>

Ángela María Salazar Maya

Universidad de Antioquia

Orcid: <https://orcid.org/0000-0001-7599-1193>

Este libro fue diagramado utilizando fuentes tipográficas Helvetica LT Std en sus respectivas variaciones a 11 y 10 puntos, y para los títulos a 20 puntos. Se Terminó de imprimir en noviembre en los talleres de SAMAVA EDICIONES E.U.
POPAYÁN - COLOMBIA 2018

Fue publicado por la Facultad de Ingenierías de la la Universidad Santiago de Cali.