

# 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*

<b>Estudio comparativo entre lenguajes textuales y lenguajes visuales. Caso: PiCO y GraPiCO</b>	
<b>Etapa 1</b>	Elaboración de hipótesis en experimentos de lenguajes de programación.
<b>Etapa 2</b>	Variables en un experimento de lenguajes de programación.
<b>Etapa 3</b>	Unidades experimentales utilizadas en pruebas de lenguajes de programación.
<b>Etapa 4</b>	Tratamientos y replicas en un experimento de programación.
<b>Etapa 5</b>	Aprendizaje Basado en Problemas (ABP) aplicado a los lenguajes de programación.
<b>Etapa 6</b>	La comunicación en el estudio comparativo entre lenguajes textuales y lenguajes visuales: Caso PiCO y GraPiCO.
<b>Etapa 7</b>	Sistematización de una experiencia de investigación entre la Comunicación Social y la Ingeniería de Software.
<b>Etapa 8</b>	Modelo de sistematización propuesto "TCACI en doble vía".
<b>Etapa 9</b>	Pasos en la realización de los audiovisuales pedagógicos: PiCO – GraPiCO y ejercicio de modelación.
<b>Etapa 10</b>	Estudio de resultados de PiCO y GraPiCO. Parte 1.
<b>Etapa 11</b>	Estudio de resultados de PiCO y GraPiCO. Parte 2.
<b>Etapa 12</b>	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<sup>[4]</sup> es un mensaje que tiene una utilidad específica, la cual mediante un proceso de razonamiento y análisis, se convierte en conocimiento.

El software<sup>[4]</sup> 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<sup>[1]</sup>. 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*,<sup>[1][5][6]</sup> 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*<sup>[1]</sup>, 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 <sup>[1][7]</sup> 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* <sup>[3]</sup> 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* <sup>[2]</sup> 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* <sup>[2][8]</sup> 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 gráficas, 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.