



Conference Paper

Computer Graphics of the Regular Polygons and their Applications

César Villacís, Walter Fuertes, Margarita Zambrano, Carlos Cabascango, Edwin Cuichán, Ramiro Benavides, Fernando Solís, and Margarita Kostikova

Departamento de Ciencias de la Computación, Universidad de las Fuerzas Armadas ESPE, Sangolquí, ESPE

Abstract

The computer graphics of regular polygons and their applications is a scarcely studied area that allows to create situations of significant learning by its mathematical and geometric content. This research presents the design and programming of regular polygons and composite sacred figures using computational analytical geometry and development tools such as C#, GDI+ graphics engine and Java with SWING graphical interface. In order to achieve this, the Agile Extreme Programming (XP) methodology has been used to translate computer graphics software applications, with the purpose of understanding how computer graphics work to generate combinations of geometric figures based on regular polygons, fully parameterizable. The proof of concept, which included the evaluation of the application performance in both the .NET framework and the NetBeans IDE were carried out with a student's group of engineering in Computer Graphics subject.

Keywords: Computer Graphics, Computational Geometric Analytic, Regular Polygons, Sacred Geometry.

Corresponding Author:

César Villacís

cjvillacis@espe.edu.ec

Received: 28 July 2017

Accepted: 5 September 2017

Published: 30 January 2018

Publishing services provided by
Knowledge E

© César Villacís et al. This article is distributed under the terms of the [Creative Commons](#)

[Attribution License](#), which permits unrestricted use and redistribution provided that the original author and source are credited.

Selection and Peer-review under the responsibility of the SIIPRIN Conference Committee.

1. Introducción

La computación gráfica es un área de la informática que posibilita generar imágenes visuales y representar en 2D y 3D objetos del mundo real en la computadora con el uso de algoritmos matemáticos y geométricos [1], [2]. Abre nuevas posibilidades de innovación e implementación de nuevos modelos que facilitan el proceso de aprendizaje de los estudiantes en la abstracción del conocimiento de la manipulación de gráficos e imágenes generados por computadora. Además, el aprendizaje de la computación gráfica enfocada a la creación de gráficos por computadora de polígonos y figuras geométricas ayuda al desarrollo matemático, lógico y espacial de los estudiantes de ingeniería para la creación de interfaces gráficas de usuario en aplicaciones desktop,

OPEN ACCESS

Web y móvil. Sin embargo, el área de la computación gráfica de figuras geométricas como polígonos regulares y figuras compuestas no se ha desarrollado suficientemente, especialmente en el ámbito educacional para estudiantes de ingeniería, ya que el costo de desarrollar estas herramientas puede ser elevado.

Ante este escenario, en el ámbito computacional, algunas investigaciones [1], [3], han demostrado que los gráficos por computadora obedecen a modelos matemáticos que pueden ser automatizados utilizando lenguajes de programación como Java, Matlab, etc. Otros trabajos han demostrado que se pueden combinar diferentes áreas de las matemáticas como geometría, trigonometría, álgebra lineal, geometría analítica, cálculo vectorial para representar funciones y figuras tanto en 2D como en 3D [4], [5], [6], [7], [8]. Sobre la base de las consideraciones anteriores se puede apreciar el interés por la comunidad científica en el desarrollo de aplicaciones de gráficos por computadora con base en modelos geométricos y matemáticos enfocados al área de la Computación Gráfica.

El presente estudio tiene como objetivo, diseñar y desarrollar modelos geométricos en 2D para representar polígonos regulares y aplicaciones relacionadas a estos mediante el uso de gráficos en la computadora, totalmente parametrizables y funcionales a base de fórmulas matemáticas. Para llevarlo a cabo se inició con la investigación descriptiva de la computación gráfica de figuras geométricas básicas como círculos, cuadrados, rectángulos, triángulos y polígonos regulares. Seguidamente, se cumplió las fases de la metodología ágil Extreme Programming (XP) y su aplicabilidad para el desarrollo de gráficos por computadora con base en modelos geométricos y matemáticos. Finalmente se implementó la interfaz en dos dimensiones, que fue construida tanto en el lenguaje C# como en Java, para analizar el rendimiento de los gráficos por computadora en ambos lenguajes. Los resultados muestran que este tipo de aplicaciones ayudan al entendimiento de los estudiantes que trabajan en la creación de gráficos por computadora dentro del área de la Computación Gráfica.

Como principal contribución se presentan las fórmulas para la obtención de los vértices de los 10 primeros polígonos regulares en función de los lados que conforman a cada polígono, permitiendo graficar un polígono de cualquier tamaño solo ingresando el valor del lado. Adicionalmente se presentan cuatro aplicaciones de figuras geométricas compuestas por polígonos regulares las cuales son: (1) flor margarita, compuesta por pentágonos; (2) baldosa con diseño, compuesta por distintas figuras como cuadrados, circunferencias, rombos, octágonos, estrellas de ocho puntas, lazos de cuatro puntas; (3) estrellas de David concéntricas; (4) gema, compuesta por varios

triángulos y decágonos. Todas estas figuras geométricas son totalmente parametrizables en lo referente a su tamaño y forma, utilizando fórmulas de geometría analítica, geometría básica, trigonometría y álgebra elemental.

El resto del artículo ha sido organizado como sigue: La sección 2 describe el fundamento teórico que sustenta esta investigación. En la sección 3 se explica el proceso de desarrollo con base en Extreme Programming (XP), la geometría analítica de los polígonos regulares, las figuras compuestas y la implementación de las aplicaciones en C# y Java. En la sección 4 se ilustran los resultados obtenidos. En la sección 5 se analizan los trabajos relacionados. Finalmente en la sección 6 se exponen las conclusiones y el trabajo futuro.

2. Marco teórico referencial

2.1. Polígonos regulares en la Geometría Sagrada, La estrella de David

La estrella de David consta de dos triángulos equiláteros sobrepuestos. El triángulo con la punta hacia arriba representa lo femenino en los tres niveles: cuerpo, mente y Alma, y el triángulo con la punta hacia abajo representa lo masculino, también en los tres niveles: cuerpo, mente y Alma [9], [10], [11]. El símbolo quiere decirnos que nuestra Alma nunca se divide en dos almas, femenina y masculina, nunca se divide en Adán y Eva, sino que permanece indivisa, unitaria. Dentro de cada hombre vive una mujer, y dentro de cada mujer vive un hombre. ¿Es nuestra Alma Gemela! La estrella de David afirma que lo femenino y lo masculino son inseparables

2.2. Polígonos regulares en la Geometría Sagrada, El sol de ocho puntas preincaico

El sol de ocho puntas de la cultura Pasto está constituido por dos equinoccios, dispuestos horizontalmente, y dos solsticios, dispuestos verticalmente. Los equinoccios representan la unidad de los pensamientos verdaderos con los pensamientos falsos y la unidad de los sentimientos positivos con los sentimientos negativos. En cambio, los solsticios representan la unidad del cuerpo con la naturaleza y de lo masculino con lo femenino [9], [10], [11].

2.3. Metodología Extreme Programming (XP)

De acuerdo con [12], [13], esta es una metodología ágil de desarrollo de software ideal para pequeños o medianos grupos de trabajo en el desarrollo de software cuyos requerimientos pueden ser ambiguos o cambiantes y consideran a las personas como un factor decisivo para lograr el éxito de un proyecto. Su ciclo de vida cumple con las fases de: exploración, planificación, iteraciones para liberar el producto ionizante, y el mantenimiento. Durante este estudio se realizaron doce prácticas que sirven como punto de partida de un equipo XP conformado, fueron enfocadas en el desarrollo de las aplicaciones de software de computación gráfica.

3. Diseño de la Investigación

3.1. Proceso de Desarrollo

El proceso de desarrollo de gráficas por computadora fue basado en el ciclo de vida de XP que permite ejecutar tareas de manera iterativa e incrementable. El equipo de trabajo ha llevado a cabo una entrega incremental del producto considerando tres iteraciones: (1) Diseño de los modelos matemáticos de los polígonos regulares y sus aplicaciones; (2) Diseño y desarrollo de la interfaz gráfica de usuario de las gráficas por computadora en C# y en Java; (3) Diseño y desarrollo de las librerías de gráficos por computadora utilizando los motores de gráficos GDI+ para C# y la interfaz gráfica SWING para Java. En estas iteraciones se han liberado partes del producto que fueron completadas y evaluadas para incrementar la funcionalidad y mejorar la calidad del producto comparado con versiones previas de la aplicación de software que fueron desarrolladas e implementadas. La Fig. 1 ilustra las iteraciones en XP basadas en el modelo modificado de juegos propuesto por Villacís, C., et al, en [14].

3.2. Geometría Analítica de los Polígonos Regulares

Dado el lado de un hexágono, se obtienen los valores de los vértices encontrando los valores de los segmentos '**b**' y '**Ap**', donde se consideran los valores de los siguientes ángulos: $m\hat{1} = 60^\circ$ (ángulo externo), $m\hat{2} = 120^\circ$ (ángulo interno), $m\hat{3} = 30^\circ$, como se puede ver en la Fig. 4: Dado el lado de un heptágono, se obtienen los valores de los vértices encontrando los valores de los segmentos '**a**', '**b**', '**c**', '**d**', '**e**', '**f**', '**g**', '**h**' y de la apotema '**Ap**', donde se consideran los valores de los siguientes ángulos: $m\hat{1} = 51,42^\circ$

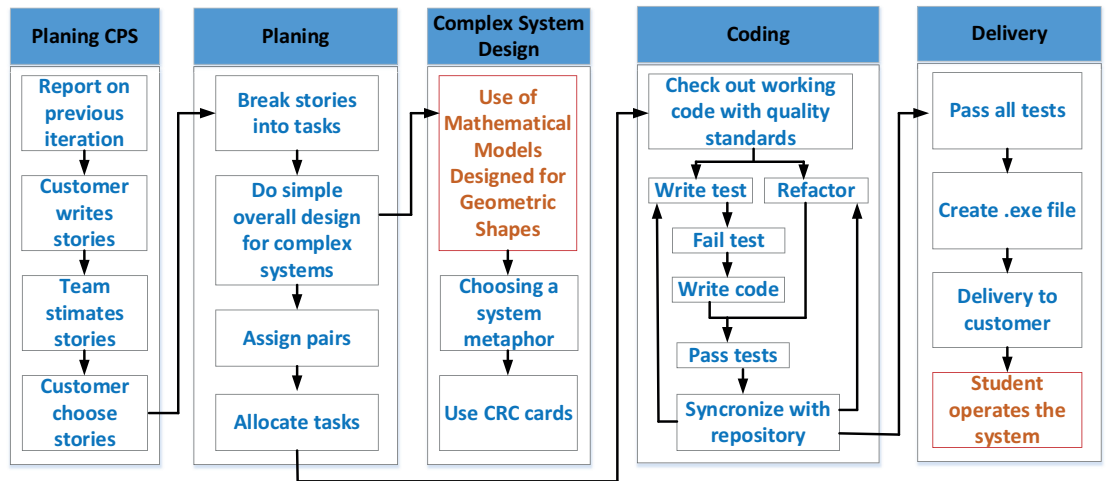


Figura 1: Modelo modificado de juegos propuesto por Villacís, C., et al. Fuente: Adaptación de [14].

(ángulo externo), $m\hat{2} = 38,54^\circ$, $m\hat{3} = 64,28^\circ$, $m\hat{4} = 128,57^\circ$ (ángulo interno). Dado el lado de un octágono, se obtienen los valores de los vértices encontrando los valores de los segmentos 'b' y de la apotema 'Ap', donde se consideran los valores de los siguientes ángulos: $m\hat{1} = 45^\circ$ (ángulo externo), $m\hat{2} = 67,5^\circ$, $m\hat{3} = 22,5^\circ$, $m\hat{4} = 135^\circ$ (ángulo interno), como se puede ver en la Fig. 2.

Dado el lado de un nonágono, se obtienen los valores de los vértices encontrando los valores de los segmentos 'a', 'b', 'c', 'd', 'e', 'f', 'g' y de la apotema 'Ap', donde se consideran los valores de los siguientes ángulos: $m\hat{1} = 40^\circ$ (ángulo externo), $m\hat{2} = 20^\circ$, $m\hat{3} = 70^\circ$, $m\hat{4} = 10^\circ$, $m\hat{5} = 30^\circ$, $m\hat{6} = 140^\circ$ (ángulo interno), como se puede ver en la Fig. 5. De forma análoga se obtendrán los valores de los vértices del decágono y undecágono. (Véase Fig. 3).

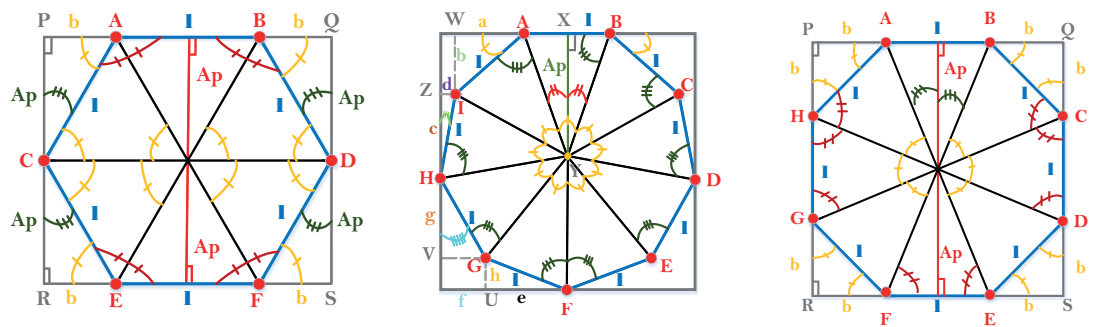


Figura 2: Geometría Analítica del hexágono, heptágono y octógono.

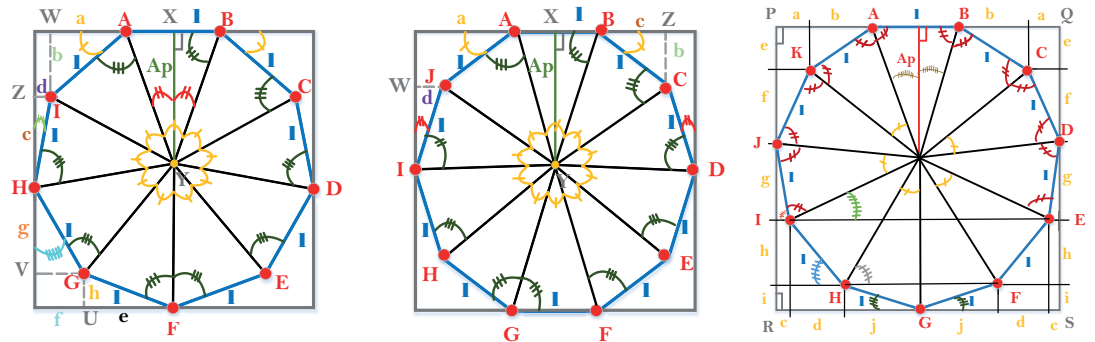


Figura 3: Geometría Analítica del nonágono, decágono y undecágono.

3.3. Figura Geométrica Sagrada de un Mosaico Pre-incaico

Dado el lado de un octágono regular, dibujar una figura geométrica sagrada que corresponde a un mosaico pre-incaico que es la combinación de distintas figuras como: dos cuadrados, dos circunferencias, un rombo, un octágono, una estrella de ocho puntas, dos lazos de cuatro puntas. Se obtienen los valores de los vértices encontrando los valores de los segmentos 'a', 'b', 'd', 'e', 'h', de la apotema 'Ap', y del radio 'r' donde se consideran los valores de los siguientes ángulos: $m\hat{1} = 45^\circ$, $m\hat{2} = 22,5^\circ$, $m\hat{3} = 26,2^\circ$, $m\hat{4} = 18,8^\circ$, como se puede ver en la Fig. 4.

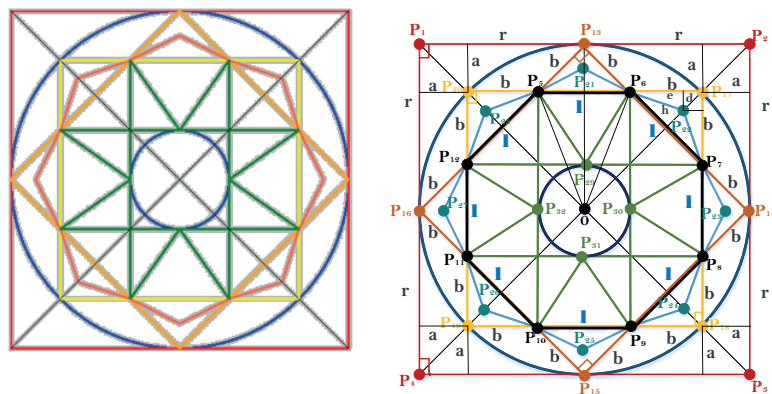


Figura 4: Figura Geométrica Sagrada de un Mosaico Pre-incaico.

3.4. Figura Geométrica Sagrada de la Estrella de David Judía

Dado el lado de un hexágono regular, dibujar una figura geométrica sagrada que tiene tres estrellas de David concéntricas. Se obtienen los valores de los vértices encontrando los valores de los segmentos 'b' y de la apotema 'Ap', donde se consideran los valores de los siguientes ángulos: $m\hat{1} = 60^\circ$, $m\hat{2} = 120^\circ$, y $m\hat{3} = 30^\circ$. (Véase Fig. 5).

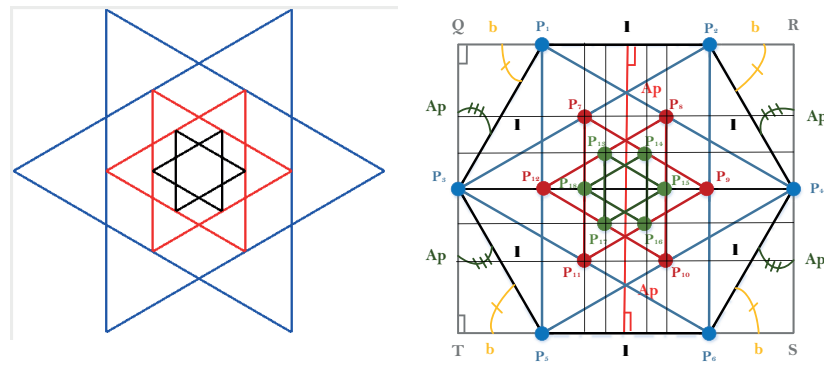


Figura 5: Figura Geométrica Sagrada de la Estrella de David Judía.

3.5. Figura Geométrica Sagrada del Cubo de Metatrón

Dado el lado de un hexágono regular, dibujar una figura geométrica sagrada que corresponde al cubo de Metatrón que es la combinación de varias estrellas hexagonales. Se obtienen los valores de los vértices encontrando los valores de los segmentos 'a', 'b', 'c', 'd', 'e', 'g', 'h', 'j', 'm', de la apotema 'Ap', y del radio 'r' donde se consideran los valores de los siguientes ángulos: $m\hat{1} = 60^\circ$, $m\hat{2} = 120^\circ$, y $m\hat{3} = 30^\circ$, como se puede ver en la Fig. 6.

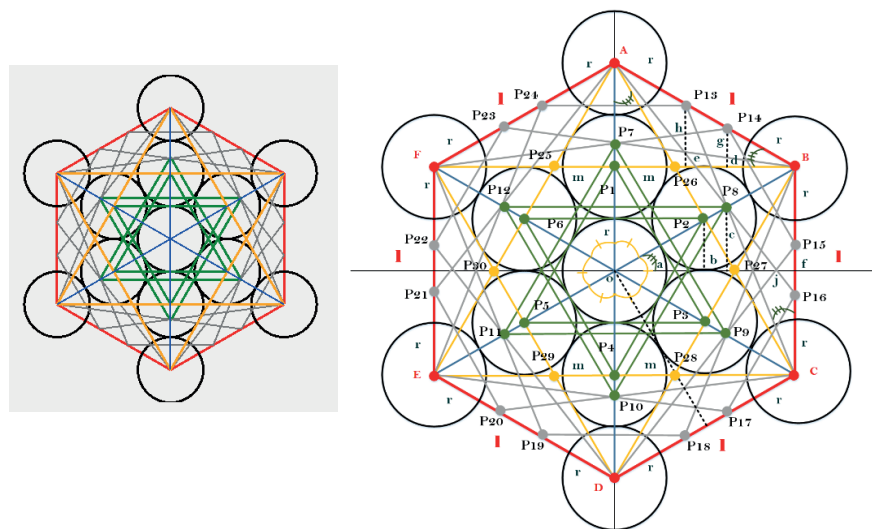


Figura 6: Figura Geométrica Sagrada del Cubo de Metatrón.

Se utilizan las fórmulas de traslación de ejes: $X_p = (x_{(P-1)}) * SF + 300$; $Y_p = (-1) * (y_{(P-1)}) * SF + 200$, para poder graficar la figura desde el centro de coordenadas cartesianas. Uniendo ciertos puntos del cubo de Metatrón se obtienen los cinco sólidos Platónicos y la figura sagrada Cristiana de la Trinidad, como se puede ver en la Fig. 7.

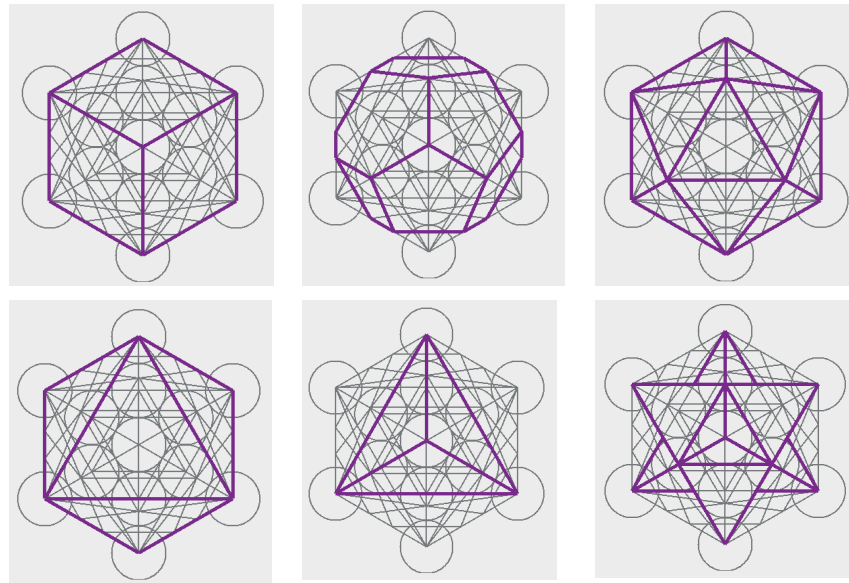


Figura 7: Sólidos Platónicos y la figura sagrada Cristiana de la Trinidad.

3.6. Figura Geométrica Sagrada del Hombre Pitagórico

Dado el lado de un hexágono regular, dibujar una figura geométrica sagrada que corresponde al cubo de Metatrón que es la combinación de varias estrellas hexagonales. Se obtienen los valores de los vértices encontrando los valores de los segmentos 'a', 'b', 'c', 'd', 'e', 'g', 'h', 'j', 'm', de la apotema 'Ap', y del radio 'r' donde se consideran los valores de los siguientes ángulos: $m\hat{1} = 60^\circ$, $m\hat{2} = 120^\circ$, y $m\hat{3} = 30^\circ$, como se puede ver en la Fig. 8.

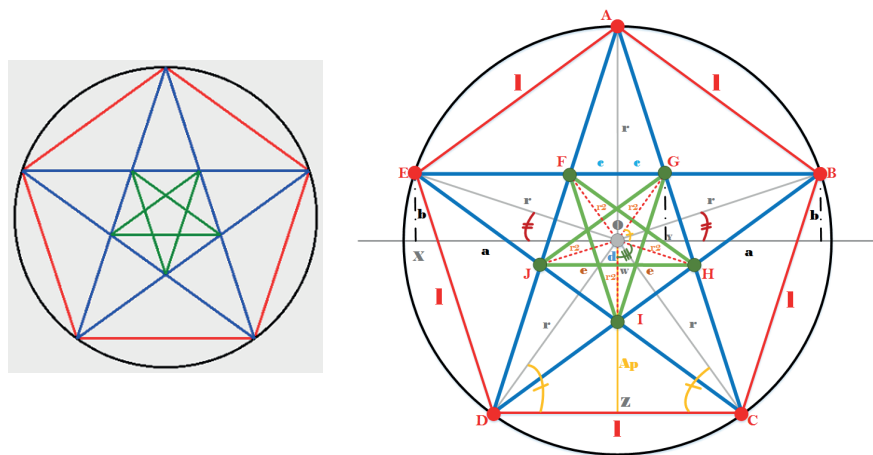


Figura 8: Figura Geométrica Sagrada del Hombre Pitagórico.

Las coordenadas de los puntos de la figura son:

$$A(0, r); B(a, b); C\left(\frac{1}{2}, -Ap\right); D\left(-\frac{1}{2}, -Ap\right); E(-a, b); G(c, b); H(e, -d); I(0, -r2); J(-e, -d).$$

También se utilizan las fórmulas de traslación de ejes para poder graficar la figura con respecto al centro del plano cartesiano.

3.7. Figura Geométrica Sagrada de la Trinidad en relación a su origen Judío-Cristiano

Dado el lado de un hexágono regular, dibujar una figura geométrica sagrada que corresponde a la Trinidad en relación a su origen Judío-Cristiano que es la combinación de un hexágono con una estrella hexagonal dentro de un triángulo equilátero. Se obtienen los valores de los vértices encontrando el valor del segmento 'a' y de la apotema 'Ap', donde se considera el valor del siguiente ángulo: $m\hat{1} = 60^\circ$, como se puede ver en la Fig. 9.

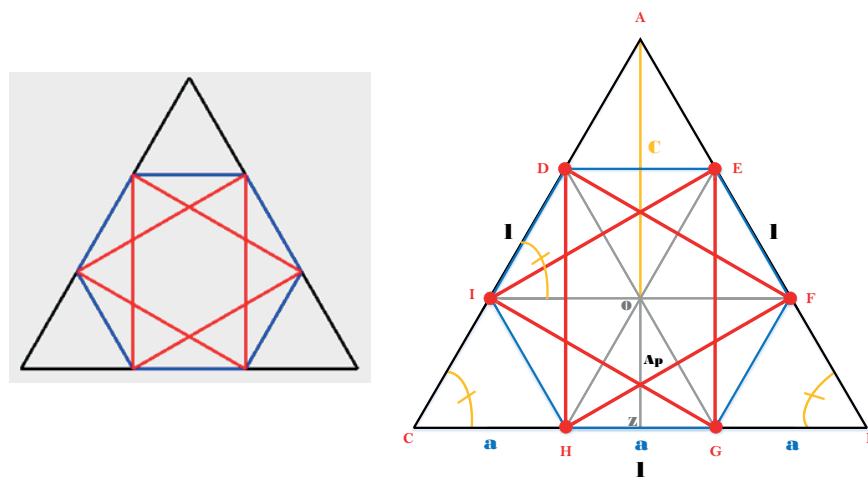


Figura 9: Figura Geométrica Sagrada de la Trinidad en relación a su origen Judío-Cristiano.

Las coordenadas de los puntos de la figura son:

$$A(0, c); B\left(\frac{3}{2}, -Ap\right); C\left(\frac{-31}{2}, -Ap\right); D\left(-\frac{a}{2}, Ap\right); E\left(\frac{a}{2}, Ap\right); F(a, 0); G\left(\frac{a}{2}, -Ap\right); H\left(\frac{-a}{2}, -Ap\right); I(-a, 0)$$

3.8. Implementación y Pruebas de la Aplicación

La Fig. 10 muestra la interfaz gráfica de usuario de la aplicación de gráficas por computadora desarrollado en C# con el motor de gráficos GDI+, donde es posible observar la imagen sagrada del Cubo de Metatrón.

La Fig. 11 muestra la interfaz gráfica de usuario de la aplicación de gráficas por computadora desarrollado en Java con la interfaz gráfica SWING, donde es posible observar la imagen sagrada del Cubo de Metatrón.

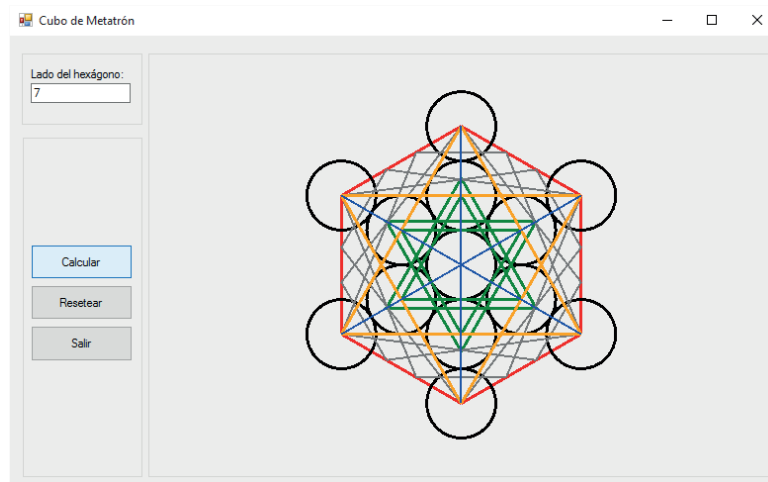


Figura 10: Interfaz gráfica de usuario de la aplicación desarrollado en C# con GDI+.

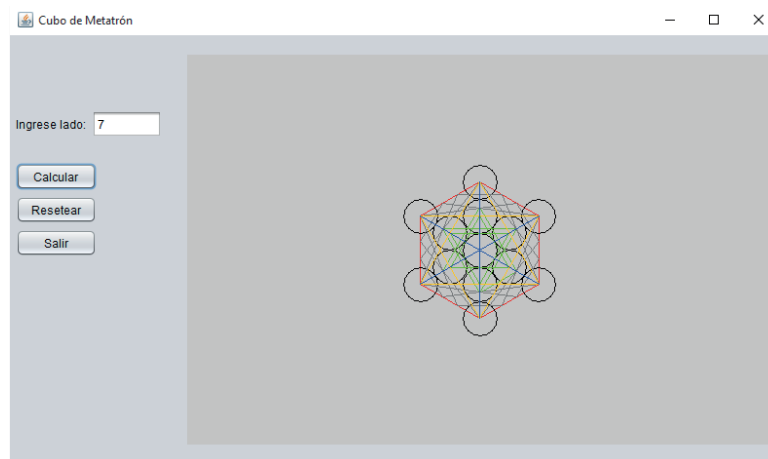


Figura 11: Interfaz gráfica de usuario de la aplicación desarrollado en Java con SWING.

Las pruebas unitarias ejecutadas en toda la aplicación de gráficas por computadora proporcionan una operación adecuada sobre todos los módulos de las figuras geométricas y en el código desarrollado. Las pruebas de aceptación han sido llevadas a cabo al final de cada iteración para evaluar los requerimientos de funcionalidad diseñados para el cliente final, quien en este caso son los estudiantes de ingeniería de sistemas e informática.

4. Evaluación de Resultados

Para probar las aplicaciones se seleccionó como proyecto piloto una universidad pública del cantón Rumiñahui, para lo cual se tomaron cuatro muestras representativas de cuatro cursos durante cuatro semestres en la asignatura de Computación Gráfica, dictado por el mismo profesor, como se indica a continuación: i) Semestre Marzo

2015-Agosto 2015: En este semestre de un total de 23 estudiantes, lograron aprobar 12 equivalente al 52,17%, con un promedio de 14,13 sobre 20; ii) Semestre Octubre 2015-Febrero 2016: En este semestre de un total de 26 estudiantes, lograron aprobar 19 equivalente al 73,08%, con un promedio de 14,82 sobre 20; iii) Semestre Marzo 2016-Agosto 2016: En este semestre de un total de 28 estudiantes, lograron aprobar 22 equivalente al 78,57%, con un promedio de 15,48 sobre 20; iv) Semestre Octubre 2016-Febrero 2017: En este semestre de un total de 20 estudiantes, lograron aprobar 18 la asignatura correspondiente al 90,00%, con un promedio de 16,67 sobre 20.

En los dos primeros semestres i) y ii) se trabajó la geometría analítica de los polígonos regulares y la graficación de los mismos. Mientras que en los semestres iii) y iv) se añadió el contexto de lo sagrado a la geometría analítica computacional haciendo que el aprendizaje se vuelva significativo.

Por otro lado se realizó un análisis de rendimiento de la CPU con respecto a las aplicaciones de gráficas por computadora realizadas en dos lenguajes de programación, obteniéndose los siguientes resultados considerando una computadora con un procesador i5: a) En el lenguaje C# con el motor de gráficos de GDI+, el consumo de la CPU fue del 3%, a velocidades que fluctúan entre 0,77 a 0,81 GHz; b) En el lenguaje Java con la interfaz gráfica SWING, el consumo de la CPU fue del 5%, a velocidades que fluctúan entre 0,77 a 0,84 GHz. Estos resultados muestran que es mejor trabajar con el lenguaje C# para desarrollar aplicaciones de gráficas por computadora, mientras que en Java hay un mayor consumo de recursos.

5. Conclusiones

En esta investigación se diseñó e implementó un grupo de aplicaciones de gráficas por computadora para representar polígonos regulares y geometría sagrada utilizando técnicas de programación orientadas a objetos. Además se crearon algoritmos de computación gráfica basados en geometría analítica y geometría plana tanto en 2D como en 3D para graficar sólidos, enfocado al aprendizaje de Computación Gráfica para estudiantes de la carrera de Informática y Sistemas. Metodológicamente se inició con una investigación descriptiva de la computación gráfica de los polígonos regulares y geometría sagrada, los motores gráficos nativos de los lenguajes C# y Java, las fases de la Metodología XP y su aplicabilidad para el desarrollo de aplicaciones de gráficas por computadora. Finalmente se implementó la interfaz gráfica en 2D y 3D de las aplicaciones. Los resultados muestran que este tipo de aplicaciones ayudan a mejorar el razonamiento abstracto de los estudiantes de materias complejas como

la Computación Gráfica, mediante la solución de problemas aplicados al mundo real con narrativa estructurada, enseñando el significado de la ciencia y haciendo que el aprendizaje se vuelva significativo. Además de crear un contexto de aprendizaje dentro del cual los elementos cognitivos se integran como un todo con este entorno que rescata los principios y valores culturales de los pueblos ancestrales. Una vez integrado todos estos elementos, el proceso de enseñanza-aprendizaje de los estudiantes se hace más sencillo debido a la motivación y a la analogía que hace el aprendiz al interrelacionar sus elementos cognitivos aprendidos con los elementos cognitivos por aprender. Para validarlo se planteó a los dos últimos grupos de c) y d) de la sección 4, obteniéndose una mayor comprensión de la materia y logrando que menos estudiantes reprueben la asignatura. Desde el punto de vista de la Computación Gráfica se desarrollaron los modelos geométricos y analíticos computacionales de los polígonos regulares que no están definidos en ningún trabajo relacionado ya que solo se enfocan a graficar este tipo de figuras geométricas mediante regla y compás y no mediante modelos matemáticos y computacionales. Como trabajo futuro se plantean aplicaciones de gráficas por computadora utilizando matemática fractal para realizar procesos recursivos de construcción de figuras concéntricas y planos de mosaicos.

Referencias

- [1] Ammeraal, L.: Computer Graphics for Java Programmers. 3rd Edition. Wiley, England (2001) ISBN: 0471981427.
- [2] Mortenson, M.E.: Mathematics for Computer Graphics Applications. Second Edition, Industrial Press Inc., New York (1999). ISBN: 0-8311-3111-X.
- [3] Marschner, S., Shirley, P.: Fundamentals of Computer Graphics, Fourth Edition, AK Peters/CRC Press, USA, (2015). ISBN: 1482229394.
- [4] Solomon, J.: Numerical Algorithms: Methods for Computer Vision, Machine Learning, and Graphics, First Edition, AK Peters/CRC Press, USA, (2015). ISBN-13: 978-1482251883.
- [5] Vince, J.: Rotation Transform for Computer Graphics, Springer, 1st Edition, USA, (2011). ISBN: 0857291530.
- [6] Kindle, J.: Geometría Analítica Plana y del Espacio. 1ra edición. McGraw-Hill, México (1970). ISBN: 9684229488.
- [7] Lehmann, C.: Geometría Analítica. 1ra edición. UTEHA, México (1959).
- [8] Kletenik, D.: Problemas de Geometría Analítica. 5ta edición. Editorial MIR, Moscú (1981).

- [9] Lawlor, R., *Mitos: Dioses, Misterios. Geometría Sagrada*. Debate Decisiones, Madrid, 1993. ISBN: 84-7444-748-8.
- [10] Ghyka, M.: *Estética de las Proporciones en la Naturaleza y en las Artes*. Editorial Poseidón, Barcelona, España, 1953. ISBN: 84-85083-06-7.
- [11] Ghyka, M.: *El Número de Oro. Ritos y Ritmos Pitagóricos en el Desarrollo de la Civilización Occidental*. Editorial Poseidón, Barcelona, España, 1978.
- [12] Beck, K., 2000. *Extreme programming explained: embrace change*. Addison-Wesley Professional.
- [13] Schneider, J. G., & Johnston, L., 2003. *Extreme Programming at universities: an educational perspective*. 25th International conference on software engineering (pp. 594-599). IEEE Computer Society.
- [14] Villacís, C., Fuertes, W., Santillán, M., Aules, H., Tacuri, A., Zambrano, M. and Salguero, E. *On the Development of Strategic Games based on a Semiotic Analysis: A Case Study of an Optimized Tic-Tac-Toe*. In *Proceedings of the 18th International Conference on Enterprise Information Systems (ICEIS 2016) - Volume 1*, pages 425-432.