

# Reporte de Proyecto Individual Unidad 3

## Razones Trigonométricas en el Círculo Unitario

Osiel Alejandro Ordoñez Cruz\*

\*Ingeniería en Tecnologías de la Información

Universidad Politécnica de Victoria

**Resumen**—En el presente proyecto se desarrolló una aplicación que calcula las razones trigonométricas mediante un círculo unitario, dicho proyecto se desarrolló con Python y con componentes gráficos de la librería PyQt5. Los resultados fueron satisfactorios y cumplen con el objetivo de mostrar el proceso de obtención de las razones trigonométricas en los distintos ángulos mediante animaciones.

### I. INTRODUCCIÓN

El tema principal del presente proyecto son las razones trigonométricas en el círculo unitario [1], este círculo es de suma utilidad ya que con él es posible mostrar las razones trigonométricas [1].

El objetivo principal del proyecto, es construir dicho círculo unitario, y en base a él demostrar cómo los valores de dicho círculo arrojan las funciones trigonométricas, este proyecto está basado en una aplicación de geogebra de título, Razones Trigonométricas en el Círculo Unitario.

Para el desarrollo de este proyecto, se debe utilizar algunas herramientas, las cuales son Python y PyQt5 [2], estas herramientas serán de suma importancia para el desarrollo, ya que Python permite el cálculo de los valores relevantes, mientras que PyQt5 da la posibilidad de poder visualizar estos datos en una gráfica.

La importancia de este objetivo radica en el poder demostrar cómo se pueden obtener las razones trigonométricas fundamentales en base a dicho círculo.

### II. DESARROLLO EXPERIMENTAL

Para el dibujado de todo, se usó una escala, la cual controla el tamaño de todos los elementos gráficos, esto para que el usuario pueda acercar o alejar la vista de los elementos a su gusto. Para el funcionamiento de las funciones anteriores, se usó el `KeyPressEvent` [3], con las teclas Arriba y Abajo.

Posteriormente, se dibujó el círculo unitario, dicho círculo se encuentra en el centro de la pantalla y tiene radio proporcional a la escala, posterior a ello, se dibujaron las vertices de los círculos en base a las coordenadas del círculo y el radio.

Después, se hizo el dibujado del punto C, dicho punto tiene que recorrer la circunferencia del círculo unitario, para esto, se usó un `QTimer`, el cual llama a la función `actualizarPuntos`, en dicha función, se genera el punto C en base a los grados que tiene, esto último gracias a la función `getPoint` [4], la cual obtiene un punto en base a la posición, el tamaño del círculo y los grados del punto, adicionalmente, se configuró

la tecla Space para pausar la animación, esto se hace mediante la deshabilitación del avanzado del punto C.

Todos los puntos son almacenados en un diccionario llamado puntos, este diccionario contiene todos los puntos existentes y sus coordenadas, los cuales se encuentran en la tabla I.

En base al punto C y el punto A, se genera el punto I y K, con estos puntos y los vertices, se pueden generar las líneas h, cos, sen, A-B, A-D, A-F, A-G, i, j, todas las líneas del programa son objetos `QLineF` [5], y son guardadas en el diccionario llamado líneas, dichas líneas se pueden encontrar en la tabla II.

A partir de este punto, se generó una línea tangente llamada l, esta línea se calcula a partir de la fórmula de la pendiente [6], la fórmula de punto-pendiente [6], calcular la tangente con respecto al punto C [7] y el proceso de graficación de líneas a partir de su ecuación de punto-pendiente [8].

En base a la línea l, fue posible generar los demás puntos y líneas de la tabla I y tabla II. Para esto último se desarrollaron funciones, tales como `obtenerInterseccion` y `obtenerLineaEntrePuntos`, dichas funciones hacen uso de funciones del objeto `QLineF`, más específicamente las funciones `intersects` y las funciones de obtención de los puntos x y y del `QLineF`.

Todos los puntos y líneas generadas son dibujadas con distinto color, esto para diferenciarlos a la hora visualizar las razones trigonométricas, esto último se logra mediante el objeto `QPainter` [9]. Para dibujar los puntos, se utiliza la función `drawEllipse` del `QPainter`, mientras que para las líneas se utiliza el método `drawLine`, estos elementos son dibujados cada que el timer se actualiza, es decir, cada que el punto C cambia su posición en base a los grados.

A partir de todos estos elementos generados, se obtiene la longitud de las líneas llamadas sen, cos, tan, cot, sec, csc, estas longitudes corresponden a las razones trigonométricas, y son mostradas mediante el método del `QPainter`, `drawText`.

Finalmente, el ángulo A-C-B es dibujado, esto mediante el ángulo de la línea h, y el método de `QPainter` llamado `drawArc`, que brinda la posibilidad de dibujar arcos. El valor del ángulo también es mostrado con el método `drawText`.

### III. RESULTADOS

Los resultados obtenidos fueron satisfactorios, esto debido a que, efectivamente se obtuvieron las razones trigonométricas en base al círculo unitario.

Tabla I: Puntos utilizados en la aplicación

Punto	Descripción
A	Origen del círculo unitario
B	Vertice derecho del círculo unitario
G	Vertice superior del círculo unitario
D	Vertice izquierdo del círculo unitario
F	Vertice inferior del círculo unitario
P	0.043 unidades a la izquierda del punto A
Q	0.044 unidades abajo del punto A
C	Punto que rodea la circunferencia
I	Punto con valor x de C y valor y de A
K	Punto con valor x de A y valor y de C
L	Intersección entre línea X y l
M	Intersección entre línea Y y l
R	Intersección entre línea q y r
O	Intersección entre línea m y n
S	Intersección entre línea fl y h
T	Intersección entre línea gl y h

Tabla II: Líneas utilizadas en la aplicación

Línea	Descripción
X	Eje X
Y	Eje Y
h	Línea entre puntos A y C
cos	Línea entre puntos A e I
sen	Línea entre puntos A y K
A-B	Línea entre puntos A y B
A-D	Línea entre puntos A y D
A-F	Línea entre puntos A y F
A-G	Línea entre puntos A y G
i	Línea entre puntos I y C
j	Línea entre puntos K y C
l	Recta tangente al círculo con punto de tangencia C
hl	Línea entre puntos L y C
il	Línea entre puntos M y C
r	Línea que recorre todo en X pasando por y del punto M
fl	Línea que recorre todo en Y pasando por x del punto B
q	Línea que recorre todo en Y pasando por x del punto Q
n	Línea que recorre todo en Y pasando por x del punto L
m	Línea que recorre todo en X pasando por y del punto P
csc	Línea entre puntos R y Q
sec	Línea entre puntos P y O
tan	Línea entre puntos S y B
gl	Línea que recorre todo en X pasando por y del punto G
cot	Línea entre puntos T y G
kl	Línea entre puntos S y C
jl	Línea entre puntos C y T

Adicionalmente, es posible pausar la animación y tanto aumentar como disminuir la vista para poder apreciar de mejor manera cómo funciona el círculo.

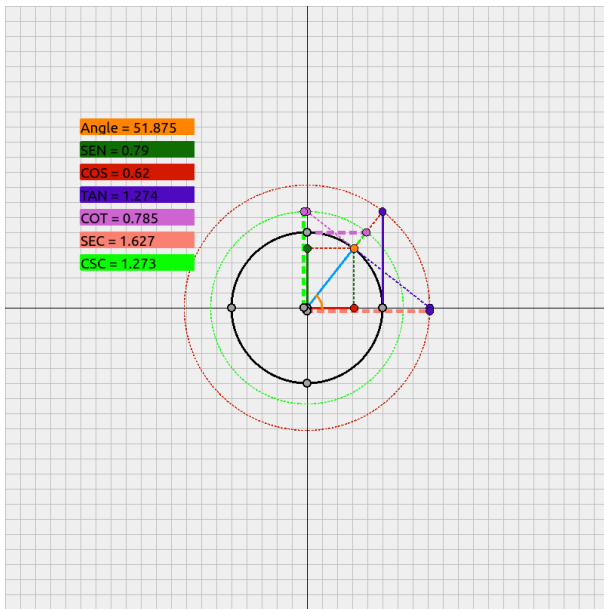
En la figura 1 se muestra la aplicación en el valor inicial de 52 grados. En la figura 1(a) Cuenta con el acercamiento por defecto. En la figura 1(b) se muestra un acercamiento mayor. En la figura 1(c) se muestra un alejamiento de la vista.

En la figura 2 se muestra la aplicación con distintos grados, para apreciar la animación de la misma.

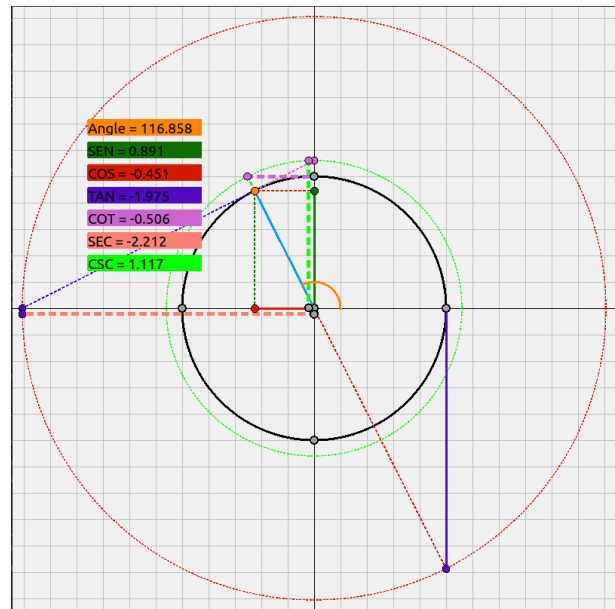
#### IV. CONCLUSIÓN

En el presente trabajo se desarrolló una solución eficiente para el objetivo planteado, puesto que se llegaron a los objetivos principales, los cuales son obtener las razones trigonométricas en base al círculo unitario utilizando Python y PyQt5.

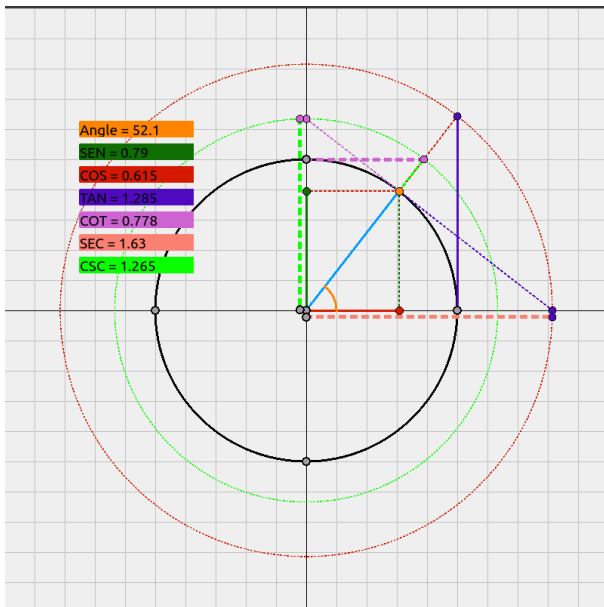
Las razones trigonométricas se obtuvieron mediante la animación de distintos puntos, líneas, ángulos y coordenadas, y la animación de estos se logró mediante un QTimer [10] y el objeto Painter con sus respectivas funciones.



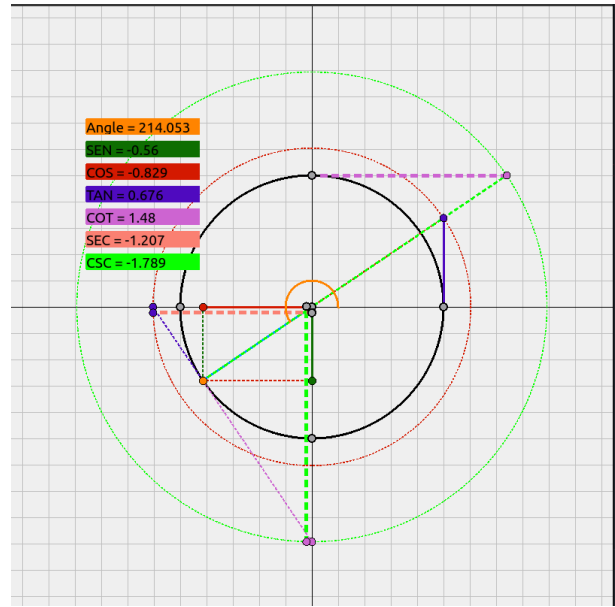
(a) 52 Grados Acercamiento por defecto



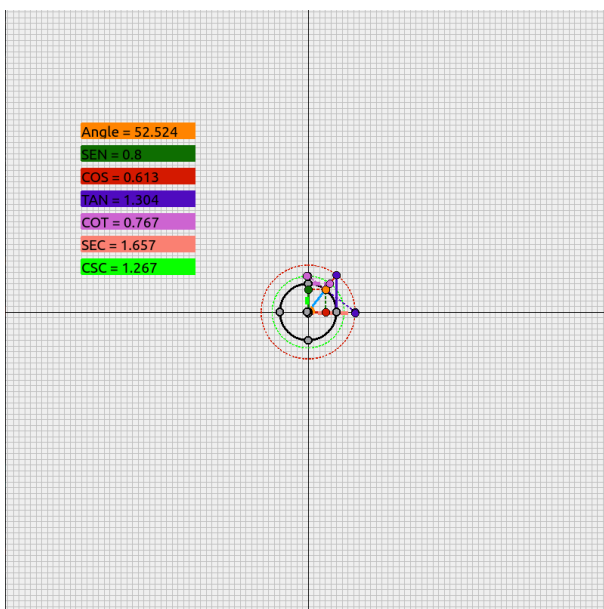
(a) 116 Grados



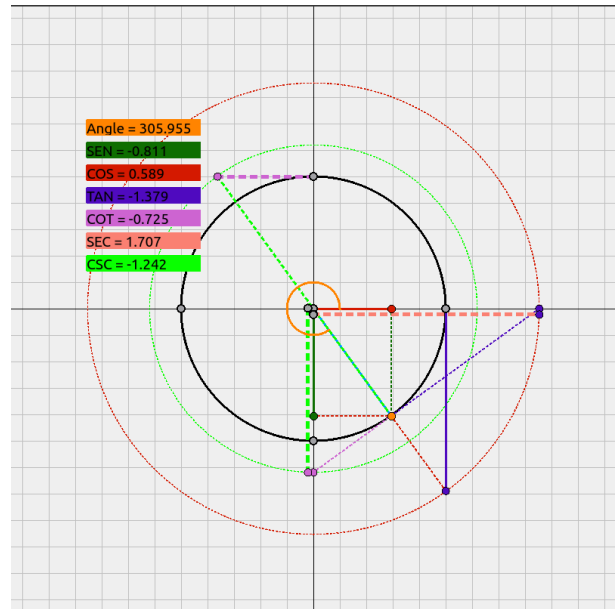
(b) 52 Grados Acercamiento mayor



(b) 214 Grados



(c) 52 Grados Alejamiento



(c) 305 Grados

## REFERENCIAS

- [1] Pedro Vicente Esteban Duarte. *Elementos generales de trigonometría*. Universidad EAFIT, pág. 20.
- [2] *The complete PyQt5 tutorial — Create GUI applications with Python*. <https://www.pythonguis.com/pyqt5-tutorial/>. Consultado el 28-11-2022.
- [3] *QKeyEvent Class*. <https://doc.qt.io/qt-6/qkeyevent.html>. Consultado el 28-11-2022.
- [4] *Calcular la posición de cualquier punto en una circunferencia*. <https://foro.rinconmatematico.com/index.php?topic=16490.0>. Consultado el 28-11-2022.
- [5] *QLineF Class*. <https://doc.qt.io/qt-6/qlinef.html>. Consultado el 28-11-2022.
- [6] *La ecuación punto-pendiente de la recta*. <https://www.superprof.es/apuntes/escolar/matematicas/analitica/recta/ecuacion-punto-pendiente-de-la-recta.html>. Consultado el 28-11-2022.
- [7] *Ecuación de una recta tangente a una circunferencia dado el centro y el punto de tangencia*. [https://www.youtube.com/watch?v=KHNU9ZN\\_iI8&t=679s&ab\\_channel=EduMate](https://www.youtube.com/watch?v=KHNU9ZN_iI8&t=679s&ab_channel=EduMate). Consultado el 28-11-2022.
- [8] *Repaso de las gráficas de rectas en la forma pendiente-ordenada al origen*. <https://es.khanacademy.org/math/algebra/x2f8bb11595b61c86:forms-of-linear-equations/x2f8bb11595b61c86:graphing-slope-intercept-equation/s/a/graphing-lines-review>. Consultado el 28-11-2022.
- [9] *QPainter Class*. <https://doc.qt.io/qt-6/qpainter.html>. Consultado el 28-11-2022.
- [10] *QTimer Class*. <https://doc.qt.io/qt-6/qtimer.html>. Consultado el 28-11-2022.