



CCADET

CENTRO DE CIENCIAS APLICADAS Y DESARROLLO
TECNOLÓGICO
UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

Aplicación para la simulación de la trayectoria aparente del sol

Benjamín Valera Orozco
Gerardo Antonio Ruíz Botello

Septiembre 2014

Proyecto interno

Aplicación para la simulación de la trayectoria aparente del sol

Benjamín Valera Orozco y Gerardo Antonio Ruíz Botello

Centro de Ciencias Aplicadas y Desarrollo Tecnológico, Universidad Nacional Autónoma de México,
Circuito Exterior S/N, Ciudad Universitaria AP 70-186, C.P. 04510, México D.F. México

Resumen

El cálculo de la trayectoria aparente del sol ha sido utilizado ampliamente en las áreas de astrofotografía y energía solar. En estas áreas, un dispositivo de observación localizado en una posición fija sobre la tierra, es orientado para seguir al objeto celestial. Un ejemplo es el seguidor solar, en donde un panel fotovoltaico o lentes concentradores son dirigidos hacia el sol con el objetivo de maximizar la energía colectada. Estos dispositivos de observación se pueden orientar utilizando las ecuaciones que describen la trayectoria del sol con las cuales es posible determinar la posición del sol a partir de la posición sobre la tierra, la fecha y hora en esa posición. En el presente informe se reporta el desarrollo de una aplicación para teléfono celular del tipo "Smartphone" que simula la trayectoria solar con gráficos 3D a partir de los datos de posición y tiempo que son ingresados por el operador u obtenidos automáticamente del dispositivo. La aplicación permite mostrar al usuario, de manera amigable, la trayectoria aparente del sol. El objetivo de este trabajo es comprobar y comprender las ecuaciones solares para posteriormente desarrollar aplicaciones en el ámbito de las energías renovables, en particular, la energía solar.

Indice

Introducción	3
Definición del problema	3
Entorno actual	3
Descripción del problema a resolver y motivación	3
Relevancia y limitaciones	4
Relación con otras áreas	4
Método	4
Objetivos y resultados esperados	4
Organización del informe	5
1 Fundamentos	6
1.1 Ecuaciones de la trayectoria del sol	6
1.1.1 Registro de la fecha	6
1.1.2 Rotación de la trayectoria solar	7
1.1.3 Traslación de la trayectoria solar	8
1.1.4 Rotación de la tierra	10
1.2 Objetos geométricos	10
1.2.1 Flecha	11
1.2.2 Planet	12
2 Descripción de la aplicación	13
2.1 Procedimiento de instalación	13
2.1.1 Requisitos mínimos	13
2.1.2 Instalación de la aplicación	13
2.2 Descripción del programa en el ámbito del operador	13
2.3 Descripción del programa en el ámbito del programador	16
2.3.1 Clase Flecha	17
2.3.2 Clase GPSTracker	18
2.3.3 Clase MainActivity	19
2.3.4 Clase MyGLSurfaceView	19
2.3.5 Clase Planet	20
2.3.6 Clase PlanetRender	20
2.4 Descripción del software de apoyo	22
2.4.1 informe.pov	22
2.4.2 informe2.pov	23
3 Resultados y discusión	24
3.1 Resultados	24
3.2 Discusión	26
Conclusiones	27
Referencias	28

Introducción

El movimiento aparente del sol depende de la posición en la tierra, la hora del día y el mes del año a partir de los cuales se define un punto espacial y temporal de observación. En el movimiento diario del sol de este a oeste, podemos asumir tres eventos importantes: la hora en la que sale el sol (amanecer), la hora en la que el sol alcanza su máxima altura (mediodía) y la hora en la que el sol se oculta (atardecer). Adicionalmente, con el transcurrir de los meses, la trayectoria diaria del sol se va desplazando de norte a sur en donde se definen los siguientes tres eventos anuales: el día en el que el sol tiene su máximo desplazamiento hacia el norte (solsticio de verano), el día en el que el sol amanece exactamente por el este (equinoccio) y el día en el que el sol tiene su máximo desplazamiento hacia el sur (solsticio de invierno). Los eventos descritos se muestran en la figura 1.

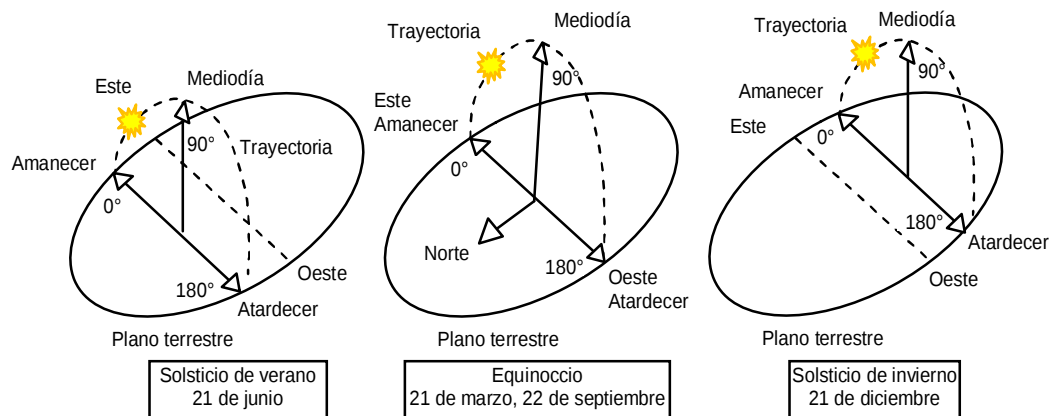


Figura 1. Movimiento aparente del sol.

Definición del problema

Los eventos mostrados en la figura 1 son importantes en el momento de calcular la trayectoria del sol y de alguna manera ayudan a comprender la mecánica solar. No obstante, con los recursos modernos de cómputo actuales es posible obtener una descripción completa y detallada del movimiento aparente del sol incluyendo la influencia de la posición del observador. De esta manera, se han desarrollado una gran variedad de simuladores del movimiento aparente del sol, que constituyen una herramienta valiosa de aprendizaje y pueden ser utilizados para evaluar dispositivos electromecánicos de seguimiento solar.

Entorno actual

En la actualidad existen diversos simuladores de la mecánica celeste como son simuladores del movimiento aparente del sol [University of Nebraska-Lincoln, 2010], calculadoras para las ecuaciones solares [Cornwall, 2014], representaciones de las estrellas, constelaciones y planetas [Vito, 2014], mapas celestes [SKY-MAP ORG, 2014], etc. Estas aplicaciones, ya sean plataformas para computadoras o dispositivos móviles, incluyen un procesamiento elevado de gráficos por computadora que facilita la comprensión del entorno celeste que nos rodea.

Un antecedente al presente trabajo es el libro [Smithwick, 2012] en donde se explican diversos conceptos de graficación 3D utilizando para ello el desarrollo de una aplicación de un sistema solar simplificado que incluye texturas y fuentes de iluminación.

Descripción del problema a resolver y motivación

En este trabajo se plantea el problema de desarrollar un simulador del movimiento aparente del sol en una aplicación para Smartphone que incluya gráficos tridimensionales y un modelo matemático que contemple la posición geográfica de observación (latitud y longitud), la fecha (mes y día) y genere la

trayectoria diaria del sol. Parte del presente trabajo es determinar los parámetros de operación (latitud, longitud, mes y día) a partir del mismo dispositivo Smartphone, ya que la tecnología actual así lo permite.

Los antecedentes descritos en éste apartado nos motivan para desarrollar una aplicación no disponible actualmente en el mercado de apps para dispositivos móviles.

Relevancia y limitaciones

La principal relevancia del presente proyecto es la comprensión de la mecánica solar para posteriormente desarrollar dispositivos electromecánicos del tipo seguidor solar e incursionar en el área de las energías renovables.

Por el contrario, las principales limitaciones están relacionadas con la naturaleza de la aplicación, ya que el proyecto descrito aquí constituye una plataforma software en donde es imposible obtener un beneficio tangible de las energías renovables.

Relación con otras áreas

Existe gran relación con las áreas de programación, graficación 3D, geometría analítica, álgebra, OpenGL y Java.

Método

El método utilizado es la programación de dispositivos móviles con sistema operativo Android utilizando OpenGL ES como interfaz gráfica 3D de programación, java como lenguaje de programación y Eclipse como entorno de desarrollo. El método empleado se muestra en la figura 2.

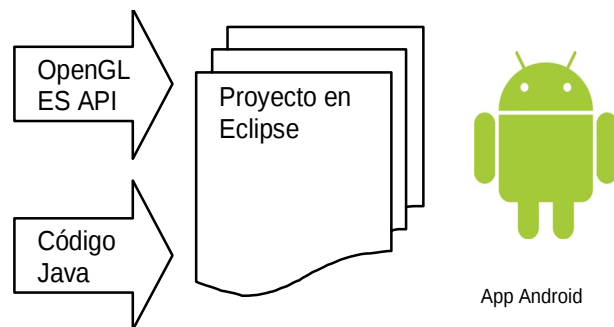


Figura 2. Metodología.

En la metodología mostrada en la figura 2 se utiliza la Interfaz para la Programación de Aplicaciones (API) conocida como OpenGL ES para producir los gráficos 3D que requiere la simulación de la trayectoria solar y la representación gráfica del sol y la tierra. Con la API OpenGL ES es posible construir elementos geométricos a partir de primitivas, proporcionarles color o textura, desplegarlos, animarlos e iluminarlos en un ambiente global de coordenadas tridimensional. Por otra parte, la API de Java permite manipular controles comunes de cualquier aplicación, manejar los recursos hardware del dispositivo móvil y programar los algoritmos específicos de la aplicación. Finalmente, Eclipse es el ambiente de desarrollo en donde se puede editar, depurar, ejecutar y transferir la aplicación desarrollada.

Objetivos y resultados esperados

Desarrollar una aplicación para Smartphone con sistema operativo Android para simular la trayectoria aparente del sol al especificar una posición espacio-temporal ya sea de modo manual u obtenida directamente del dispositivo.

Organización del informe

En el capítulo uno del informe se describen los fundamentos matemáticos para el modelo de trayectoria solar y la construcción de los objetos gráficos que serán desplegados en la simulación. En el capítulo dos se describe la aplicación desarrollada en los ámbitos del programador y del operador. El capítulo tres contiene los resultados obtenidos de la aplicación. Se incluye un CD conteniendo la siguiente estructura de directorios.

Informe
CodigoFuente
Programa
Figuras

1 Fundamentos

En la presente sección se describen los fundamentos matemáticos para modelar la trayectoria del sol y la construcción de los objetos geométricos que se despliegan en la aplicación.

1.1 Ecuaciones de la trayectoria del sol

Si se observa el cielo desde la tierra, da la impresión de que el sol se mueve alrededor del plano terrestre, como se muestra en la figura 3.

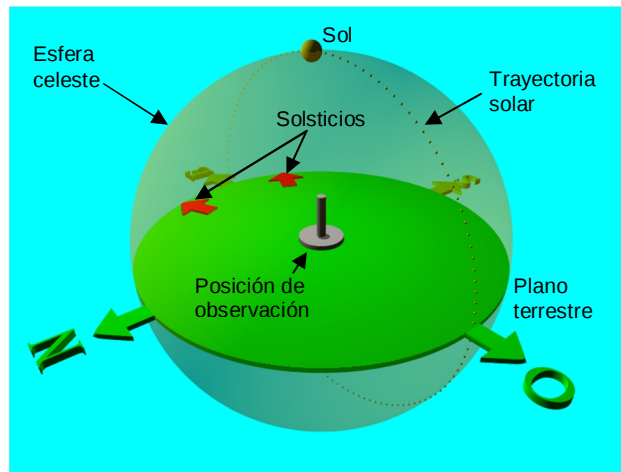


Figura 3. Modelo de la trayectoria aparente del sol.

Al modelo de la figura 3 se le conoce como “trayectoria aparente del sol” ya que da la impresión de que el observador permanece fijo y el sol se desplaza alrededor de él navegando por una esfera que circunscribe la trayectoria solar. El código para generar la figura 3 se describe en la sección 2.4. El punto de partida para el modelo se establece en la posición terrestre con latitud cero (ecuador) y la fecha el 21 de marzo o 22 de septiembre (equinoccio). Bajo estas condiciones la trayectoria solar es una circunferencia que pasa exactamente por el este y el oeste geográficos y abarca el máximo radio de la esfera celeste, como se muestra en la figura 3.

En éste modelo, la variación de los días del año o fecha se puede representar como una traslación de la trayectoria solar sobre el eje que pasa entre el norte y el sur, desplazándose entre los solsticios. Por otra parte, la posición de observación solo es afectada por la latitud y se representa en el modelo como una rotación alrededor del eje que pasa por el este y el oeste. Cabe mencionar que todas las posiciones geográficas con igual longitud observan la misma trayectoria solar. De esta manera el modelo de la trayectoria solar contempla las siguientes transformaciones:

- Registro de la fecha.
- Rotación
- Traslación

No obstante que la longitud terrestre no afecta el modelo descrito, es importante considerarlo para proporcionar una representación realista en la aplicación que se describe en el presente informe.

1.1.1 Registro de la fecha

La variación de los días del año o *Fecha* (mes y día) propicia que la trayectoria solar se mueva de norte a sur y viceversa. En el modelo, la *Fecha* primero debe ser un número *Entero* para que se puedan realizar operaciones matemáticas. Posteriormente, ese número *Entero* debe relacionarse con un *Movimiento* de traslación positivo para los días de verano y negativo para los días de invierno. Además, para simplificar

la transformación de *Fecha* a número *Entero*, se considera que febrero tiene 28 días. La relación entre número *Entero* y el movimiento entre los solsticios se normaliza en la variable *mesDia* para posteriormente proporcionarle la escala conveniente al factor de traslación geométrica. La figura 4 muestra la relación descrita que se considera más conveniente para su manipulación matemática en ambientes gráficos.

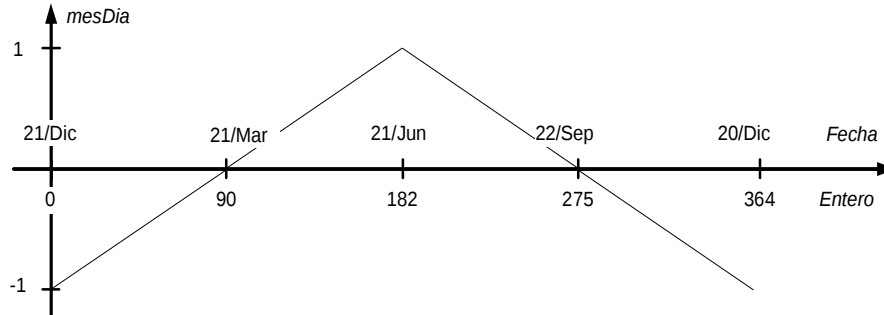


Figura 4. Relaciones Fecha-Entero-Movimiento.

Es importante mencionar que los cruces por cero de la gráfica ocurren precisamente en los solsticios en donde la trayectoria solar cambia de dirección. De la figura 4 se obtienen las siguientes relaciones:

$$mesDia = \frac{2}{182} Entero - 1 \text{ para } 0 \leq Entero \leq 182 \quad (1)$$

$$mesDia = -\frac{2}{182} (Entero - 182) + 1 \text{ para } 183 \leq Entero \leq 364 \quad (2)$$

1.1.2 Rotación de la trayectoria solar

El modelo de la figura 3 se puede representar sobre los ejes coordenados como se muestra en la figura 5.

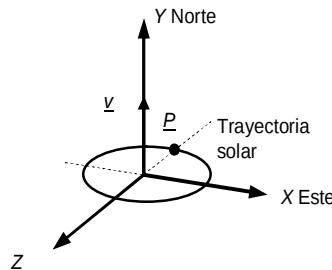


Figura 5. Trayectoria aparente del sol en los ejes coordenados.

La figura 5 muestra la trayectoria solar que se forma en la latitud cero durante el equinoccio con el planeta tierra sobre el origen. La trayectoria solar es una circunferencia en el plano X-Z que se obtiene al rotar por un ángulo α , $0 \leq \alpha < 2\pi$, el punto $\underline{P} = [0, 0, -P_z]$ alrededor del vector director $\underline{v} = [0, v_y, 0]$.

La variación de la latitud L en el punto del observador ocasiona que el vector $\underline{v}$ rote alrededor del eje X un ángulo L , como se muestra en la figura 6.

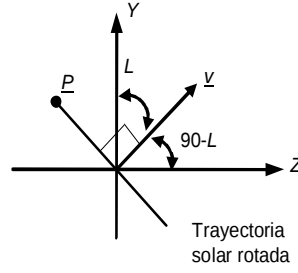


Figura 6. Variación de la latitud.

De tal manera que ahora la trayectoria solar se obtiene al rotar por un ángulo α , $0 \leq \alpha < 2\pi$, el punto $\underline{P} = [0, P_y, P_z]$ alrededor del vector director $\underline{v} = [0, v_y, v_z]$.

De la figura 6, la forma general para el vector director es

$$\underline{v} = [0, v_y, v_z] = [0, \sin(90-L), \cos(90-L)] \quad (3)$$

en donde L = Latitud.

De [Rogers, 1989] la forma general para la rotación α , $0 \leq \alpha < 2\pi$, del punto $\underline{P} = [P_x, P_y, P_z]$ alrededor del vector director $\underline{v} = [0, v_y, v_z]$ es

$$\begin{aligned} P'_x &= c P_x - v_z s P_y + v_y s P_z \\ P'_y &= v_z s P_x + [v_y^2 (1-c) + c] P_y + v_y v_z (1-c) P_z \\ P'_z &= -v_y s P_x + v_y v_z (1-c) P_y + [v_z^2 (1-c) + c] P_z \end{aligned} \quad (4)$$

En donde

$$\begin{aligned} c &= \cos(\alpha) \\ s &= \sin(\alpha) \end{aligned}$$

y $\underline{P}' = [P'_x, P'_y, P'_z]$ es el nuevo punto rotado.

La variable *Latitud* entonces es usada para rotar la trayectoria solar de acuerdo a la posición especificada como variable de entrada a la aplicación.

1.1.3 Traslación de la trayectoria solar

Como ya se mencionó, la variación de la fecha introduce en el modelo una traslación de la trayectoria solar sobre el eje que une el norte con el sur. En realidad la traslación ocurre efectivamente sobre el vector director $\underline{v}$. La fórmula para la traslación del punto rotado $\underline{P}' = [P'_x, P'_y, P'_z]$ sobre el eje $\underline{v} = [0, v_y, v_z]$ es la siguiente [Rogers, 1989]

$$\begin{aligned} SunPos_x &= P'_x + (k \text{ mesDia } v_x) \\ SunPos_y &= P'_y + (k \text{ mesDia } v_y) \\ SunPos_z &= P'_z + (k \text{ mesDia } v_z) \end{aligned} \quad (5)$$

En donde $\underline{SunPos} = [SunPos_x, SunPos_y, SunPos_z]$ es el nuevo punto rotado y trasladado y k es un factor de escala para proporcionarle magnitud visible a la presentación gráfica.

La variable *mesDia* entonces es usada para trasladar la trayectoria solar de acuerdo a la fecha especificada como variable de entrada a la aplicación.

Adicionalmente ocurre que al trasladar la trayectoria solar cambia su radio r_c ya que la circunferencia de la trayectoria está circunscrita a la esfera celeste, como se aprecia en la figura 7.

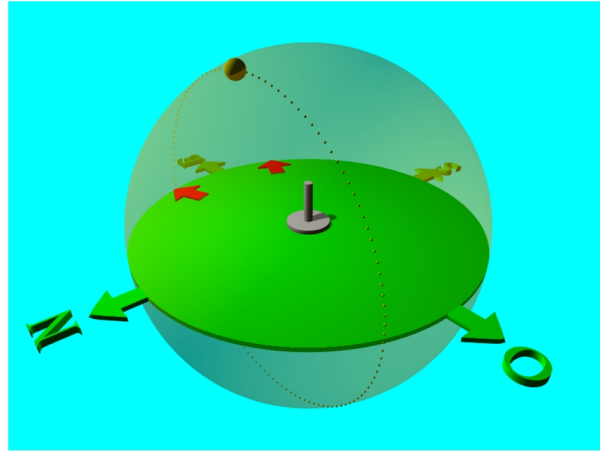


Figura 7. Traslación de la trayectoria aparente del sol.

Proyectando la trayectoria solar sobre los ejes coordenados se obtiene el arreglo que se muestra en la figura 8.

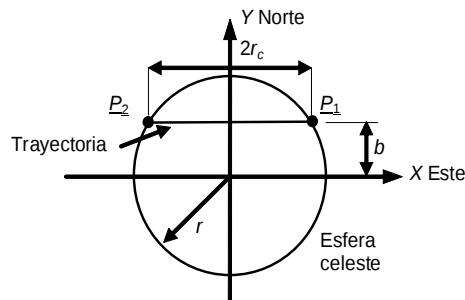


Figura 8. Radio de la trayectoria solar.

Para obtener el radio de la trayectoria solar r_c , primero se obtiene la ecuación de la circunferencia correspondiente a la esfera celeste

$$x^2 + y^2 = r^2$$

en donde r es el radio de la esfera celeste que es igual a la magnitud conveniente para la representación gráfica. Por otra parte, la ecuación de la recta que representa la trayectoria solar

$$y = b$$

en donde $b = k \text{ mesDia}$, la magnitud que se trasladó la trayectoria. Resolviendo simultáneamente las dos últimas ecuaciones para obtener los puntos de intersección $\underline{P_1}$ y $\underline{P_2}$

$$\begin{aligned} x &= \sqrt{r^2 - b^2} \\ \underline{P_1} &= [\sqrt{r^2 - b^2}, b] \\ \underline{P_2} &= [-\sqrt{r^2 - b^2}, b] \end{aligned}$$

El radio de la trayectoria r_c entonces es la mitad de la distancia entre $\underline{P_1}$ y $\underline{P_2}$

$$r_c = \frac{1}{2} \sqrt{\left(-\sqrt{r^2 - b^2} - \sqrt{r^2 - b^2}\right)^2 + (b - b)^2} \quad (6)$$

Si bien el efecto descrito aquí en donde cambia el radio de la trayectoria solar, no es muy significativo, éste se utilizó para generar la figura 7 mediante el código que se describe en la sección 2.4.

1.1.4 Rotación de la tierra

Por omisión, la textura del mapa terrestre que se utiliza en la aplicación se encuentra alineada con el punto de observación a 90° de longitud este, es decir +90° de *Longitud*. Lo que quiere decir que si se establece el punto de observación a 0° este aparecerá rotado a +90° en la aplicación, como se muestra en la figura 9.

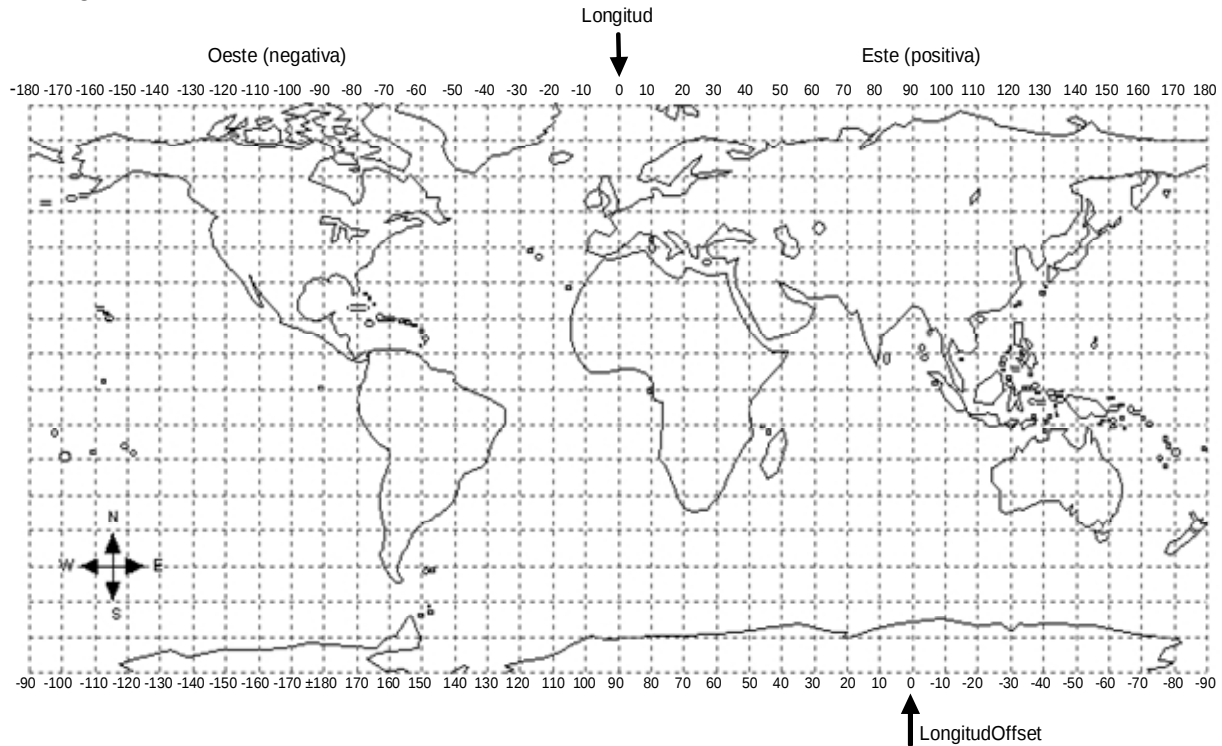


Figura 9. Longitud real del mapa terrestre.

La figura 9 muestra que existe un offset de longitud el cual hay que corregir para que la aplicación muestre la longitud real del mapa alineada con el punto de observación. La corrección se logra al rotar la tierra alrededor del eje Y con una magnitud *LongitudOffset* que se calcula a partir de la figura 9 de la siguiente manera

$$\begin{aligned} \text{LongitudOffset} &= -\text{Longitud} + 90 \quad \text{para } -90 \leq \text{Longitud} \leq 180 \\ \text{LongitudOffset} &= -\text{Longitud} - 270 \quad \text{para } -180 \leq \text{Longitud} \leq -90 \end{aligned} \quad (7)$$

1.2 Objetos geométricos

La aplicación desarrollada contempla la creación de dos objetos geométricos a partir de los cuales se representa un sistema solar simple en donde el sol es rotado para generar la trayectoria diaria. Los objetos geométricos se construyen a partir de primitivas triangulares y son los siguientes:

- Flecha, utilizada para indicar el norte geográfico del planeta tierra.
- Planet, utilizado para representar la tierra y el sol.

1.2.1 Flecha

El objeto flecha está compuesto por cinco primitivas que forman las cinco caras del prisma triangular de la figura 10.

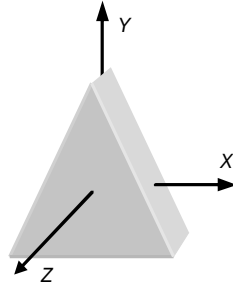


Figura 10. Flecha.

Cada primitiva está compuesta por sus vértices, vectores normales de cada vértice y triángulos descritos por sus vértices como se muestra en la tabla 1.

Primitiva	Figura	Vértices	Normales	Triángulos
tfan1		0 [0.00, 0.25,0.05] 1 [0.25,-0.25,0.05] 2 [-0.25,-0.25,0.05]	[0.00, 0.00, 1.00] [0.00, 0.00, 1.00] [0.00, 0.00, 1.00]	0, 2, 1
tfan2		3 [0.00, 0.25, 0.05] 4 [0.25, -0.25, 0.05] 5 [0.25, -0.25, -0.05] 6 [0.00, 0.25, -0.05]	[0.707, 0.707, 0.000] [0.707, 0.707, 0.000] [0.707, 0.707, 0.000] [0.707, 0.707, 0.000]	3, 4, 5 3, 5, 6
tfan3		7 [0.00, 0.25,-0.05] 8 [0.25,-0.25,-0.05] 9 [0.25,-0.25,-0.05]	[0.000, 0.000,-1.000] [0.000, 0.000,-1.000] [0.000, 0.000,-1.000]	9, 7, 8

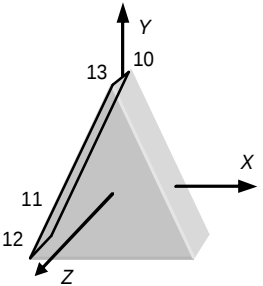
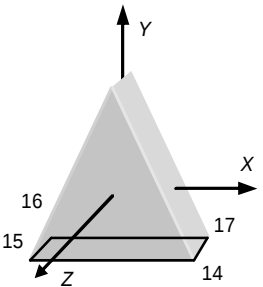
tfan4		10 [0.00, 0.25,-0.05] 11 [-0.25,-0.25,-0.05] 12 [-0.25,-0.25, 0.05] 13 [0.00, 0.25, 0.05]	[-0.707, 0.707, 0.000] [-0.707, 0.707, 0.000] [-0.707, 0.707, 0.000] [-0.707, 0.707, 0.000]	11,13,10 11,12,13
tfan5		14 [0.25,-0.25, 0.05] 15 [-0.25,-0.25, 0.05] 16 [-0.25,-0.25,-0.05] 17 [0.25,-0.25,-0.05]	[0.000,-1.000, 0.000] [0.000,-1.000, 0.000] [0.000,-1.000, 0.000] [0.000,-1.000, 0.000]	16, 17, 14 16, 14, 15

Tabla 1. Objeto Flecha.

1.2.2 Planet

El Objeto Planet se forma por triángulos segmentados a partir de la generación de pilas (stacks) y rebanadas (slices), ambos elementos circulares conformando una esfera como se muestra en la figura 11.

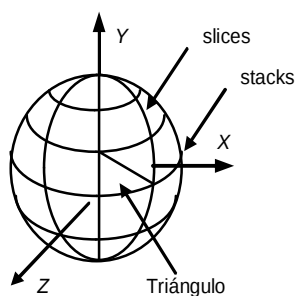


Figura 11. Planet.

El algoritmo utilizado se describe ampliamente en [Smithwick, 2012]. El objeto Planet permite variar el número de pilas, rebanadas, radio y la textura asociada a él, con lo cual se obtiene una representación muy aproximada a la de cualquier planeta del sistema solar.

2 Descripción de la aplicación

En la presente sección se describe el desarrollo de la aplicación en los ámbitos del operador y del programador así como el software de apoyo que se utilizó para ilustrar algunos conceptos.

2.1 Procedimiento de instalación

Si bien la aplicación descrita en el presente informe puede ser distribuida en el mercado de aplicaciones de Google (PlayStore) e instalada siguiendo el método típico, se recomienda seguir el siguiente proceso de instalación a partir del CD anexo.

2.1.1 Requisitos mínimos

Dispositivo móvil o Smartphone con las siguientes características:

- Sistema operativo Android 2.3 (Gingerbread) o superior.
- Sensor de posicionamiento global o GPS.
- Resolución 240X320 o superior.
- Conexión de datos USB a una computadora anfitriona.

Computadora anfitriona con las siguientes características:

- Procesador Intel Core i7 3.4GHz.
- Memoria RAM 32GB.
- Sistema operativo Windows 8.
- Eclipse ADT Bundle (Android Developer Tools) con Android SDK (Software Development Kit) [].
- 500GB de espacio libre en disco duro.
- Puerto USB.

2.1.2 Instalación de la aplicación

- Conectar el dispositivo móvil al puerto USB de la computadora anfitriona donde reside el CD anexo.
- En algunos dispositivos móviles, habilitar la transferencia de datos USB.
- Copiar el archivo del CD anexo "Programa/SeguidorSolar.apk" al dispositivo móvil (memoria de teléfono o memoria de almacenamiento interno).
- En la configuración del Smartphone, permitir instalar aplicaciones que no estén en el Android Market.
- Ejecutar el instalador de paquetes para el archivo almacenado en el Smartphone.
- Permitir a la aplicación acceder a la ubicación.

2.2 Descripción del programa en el ámbito del operador

Una vez terminado el proceso de instalación de forma exitosa, la pantalla que se despliega al iniciar la aplicación (SeguidorSolar) es la mostrada en la figura 12.



Figura 12. Pantalla de inicio de la aplicación SeguidorSolar.

La aplicación tiene dos formas de ingresar las variables de entrada al simulador:

- Editar datos manualmente: Se ingresa el mes del año, el día del mes, la longitud y la latitud de la posición terrestre de observación. Un ejemplo se muestra en la figura 13 en donde se desea simular la trayectoria del sol para el 30 de marzo en la Ciudad de México.

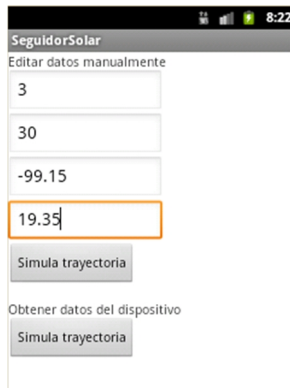


Figura 13. Trayectoria solar para la Ciudad de México el 30 de marzo.

- Obtener datos del dispositivo: Se obtienen los datos de fecha y posición geográfica directamente del dispositivo móvil (calendario y sensor GPS).

Una vez ingresados los datos iniciales, la aplicación genera la trayectoria solar como se muestra en la figura 14.

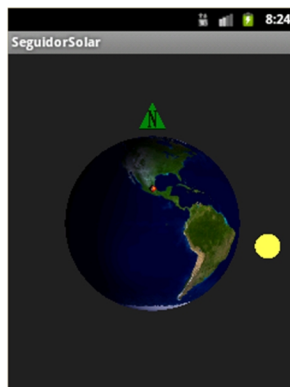


Figura 14. Simulador de trayectoria.

La pantalla despliega al planeta tierra y el sol girando a su alrededor. Es importante mencionar que la luz esta animada en conjunto con la posición del sol, es decir, cuando el sol pase por el frente de la tierra ésta se iluminará. Otros aspectos a destacar son la flecha que indica la posición geográfica del norte y el punto rojo indicando la latitud y longitud de la posición de observación actual.

Sobre la pantalla se puede rotar al sistema solar global en los ejes X y Y al desplazar con el dedo, como se muestra en la figura 15.

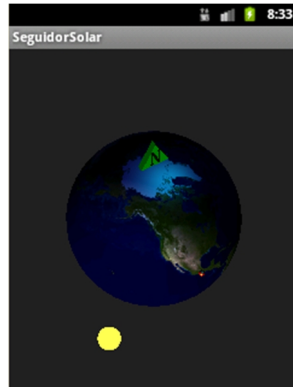


Figura 15. Rotación $X - Y$.

También se cuenta con la operación de acercamiento y alejamiento al pellizcar la pantalla con dos dedos, como se muestra en las figuras 16 y 17.

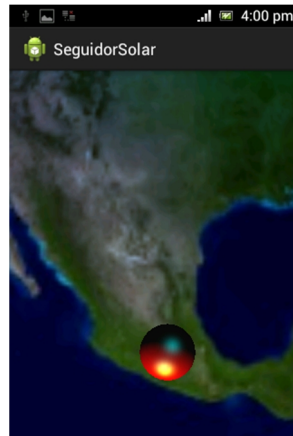


Figura 16. Acercamiento.



Figura 17. Alejamiento.

2.3 Descripción del programa en el ámbito del programador

La aplicación es un conjunto de archivos organizados en un proyecto Eclipse [Montero, 2012] haciendo uso de los siguientes recursos propios de un Smartphone:

- Gráficos 3D.
- Sensor de posición global (GPS).
- Calendario (Fecha)
- Pantalla táctil (desplazamiento y pellizco).

Estos recursos en conjunto con los elementos de control comunes a las aplicaciones (cajas de texto, botones, etc.) posibilitan la creación de programas altamente intuitivos en su operación pero con una gran complejidad en su código fuente.

La figura 18 muestra la organización del proyecto y a continuación una breve descripción del contenido útil de las carpetas mostradas.

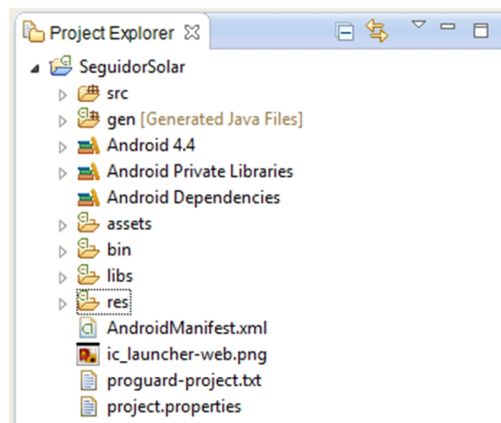


Figura 18. Proyecto SeguidorSolar.

- Carpeta src. Contiene el código fuente escrito en Java.
- Carpeta gen. Contiene el código fuente construido por omisión por el ambiente de desarrollo para las clases R y BuildConfig.
- Carpeta Android 4.4. Contiene las librerías insertadas por omisión por el ambiente de desarrollo.

- Carpeta Android Private Libraries. Contiene las librerías insertadas por omisión por el ambiente de desarrollo.
- Carpeta Android Dependencies. No contiene código alguno.
- Carpeta assets. Contiene el archivo earth.png utilizado como mapa de textura para representar el planeta tierra.
- Carpeta bin. Contiene los archivos binarios en código de máquina y recursos gráficos.
- Carpeta libs. Contiene las librerías insertadas por omisión por el ambiente de desarrollo.
- Carpeta res. Contiene los recursos utilizados por la aplicación como son los mapas de textura earth.png y norte.png, los componentes de la pantalla o layout activity_main.xml y algunos valores de cadena utilizados en la aplicación.
- Archivo AndroidManifest.xml. Presenta información esencial de la aplicación al sistema operativo Android.

La parte clave de la aplicación está contenida en el código fuente de la carpeta src, mostrada en la figura 19, la cual se describe a continuación. Es importante mencionar que las clases de la figura 19 son las que se desarrollaron de manera específica para la presente aplicación y en donde residen las operaciones descritas en el capítulo 1 de éste informe. El proyecto completo se puede consultar en la carpeta “CodigoFuente” del CD anexo.

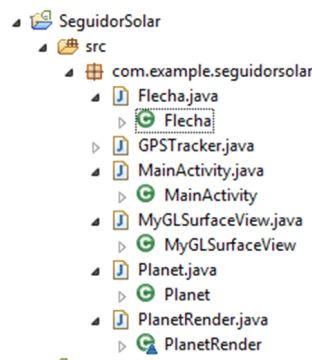


Figura 19. Clases principales de la aplicación.

2.3.1 Clase Flecha

La clase Flecha se utiliza para crear el objeto “Flecha” que se despliega en la aplicación para indicar el norte geográfico. Los métodos de esta clase son los que se muestran en la figura 20 y su descripción a continuación.

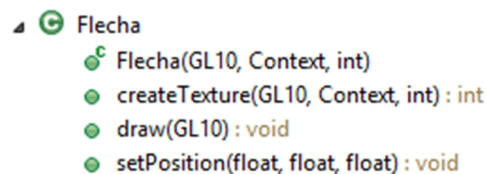


Figura 20. Clase Flecha.

`public Flecha(GL10 gl, Context context, int resourceId)`. Es el constructor de la clase. Se asigna espacio en memoria para los arreglos descritos en la tabla 1 y crea la textura con el método `createTexture` a partir del recurso recibido en la variable `resourceId`.

`public void draw(GL10 gl)`. Utiliza la primitiva OpenGL llamada `GL_TRIANGLE_FAN` para dibujar los parches triangulares que conforman el objeto geométrico “Flecha”.

`public int createTexture(GL10 gl, Context contextRegf, int resource).` Crea la textura a partir de un recurso del tipo bitmap indicado por la variable resource.

`public void setPosition(float x, float y, float z).` Establece la posición en el sistema global de coordenadas del objeto geométrico "Flecha".

2.3.2 Clase GPSTracker

La clase GPSTracker se utiliza para obtener la latitud y longitud a partir del sensor GPS integrado al dispositivo móvil [Tamada, 2012]. La figura 21 muestra los métodos de la clase y su descripción a continuación.

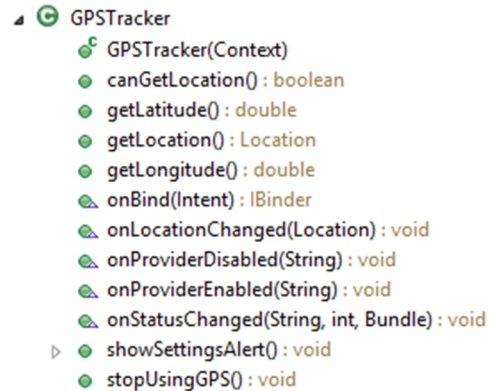


Figura 21. Clase GPSTracker.

`public GPSTracker(Context context).` Es el constructor de la clase. Ejecuta el método getLocation.

`public boolean canGetLocation().` Se utiliza para detectar si la función GPS está habilitada en la configuración del dispositivo móvil.

`public double getLatitude().` Obtiene la latitud a partir de la variable location.

`public Location getLocation().` Obtiene la localidad a partir del sensor GPS o del proveedor del servicio de datos y lo almacena en la variable location.

`public double getLongitude().` Obtiene la longitud a partir de la variable location.

`public IBinder onBind(Intent arg0).` Función sobrecargada de su clase pariente. No contiene código de implementación.

`public void onLocationChanged(Location location).` Función sobrecargada de su clase pariente. No contiene código de implementación.

`public void onProviderDisabled(String provider).` Función sobrecargada de su clase pariente. No contiene código de implementación.

`public void onProviderEnabled(String provider).` Función sobrecargada de su clase pariente. No contiene código de implementación.

`public void onStatusChanged(String provider, int status, Bundle extras).` Función sobrecargada de su clase pariente. No contiene código de implementación.

`public void showSettingsAlert()`. Función que despliega la pantalla de diálogo en donde se configura el uso del sensor de posición GPS, la red de datos o el uso de la red Wi-Fi.

`public void stopUsingGPS()`. Detiene el uso del GPS.

2.3.3 Clase MainActivity

La clase MainActivity se utiliza para registrar los datos de entrada a la aplicación y enviarlos a la clase MyGLSurfaceView para generar la simulación de la trayectoria. La figura 22 muestra los métodos de la clase y a continuación son descritos.

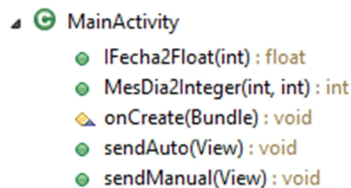


Figura 22. Clase MainActivity.

`public float IFecha2Float(int i)`. Realiza la conversión del *Entero* que representa la fecha al flotante que representa el desplazamiento *mesDia*, según las ecuaciones (1) y (2).

`public int MesDia2Integer(int mes, int dia)`. Realiza la conversión de la *Fecha* (mes y día) al número *Entero* que representa los días del año, según la gráfica de la figura 4.

`protected void onCreate(Bundle savedInstanceState)`. Crea la aplicación y despliega el layout de entrada a la aplicación, `activity_main.xml`.

`public void sendAuto(View view)`. Obtiene los datos mes y día del calendario del dispositivo y los datos de latitud y longitud de la clase GPSTracker para enviar las variables *mesDia*, *latitud* y *longitud* a la clase MyGLSurfaceView para generar la trayectoria solar.

`public void sendManual(View view)`. Obtiene los datos mes, día, latitud y longitud del layout `activity_main.xml` para enviar las variables *mesDia*, *latitud* y *longitud* a la clase MyGLSurfaceView para generar la trayectoria solar.

2.3.4 Clase MyGLSurfaceView

La clase MyGLSurfaceView se utiliza para administrar los eventos que ocurren en la vista gráfica de la aplicación. La figura 23 muestra los métodos de la clase que se explican a continuación.

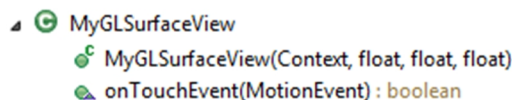


Figura 23. Clase MyGLSurfaceView.

`public MyGLSurfaceView(Context context, float longitud, float latitud, float mes)`. Es el constructor de la clase. Establece a la clase PlanetRender como render de la aplicación y envía los parámetros para generar la simulación de la trayectoria solar.

`public boolean onTouchEvent(MotionEvent ev)`. Administra los eventos de la pantalla táctil para enviar los parámetros de rotación y escala a la clase PlanetRender y modificar la apariencia de la simulación. El desplazamiento en X sobre la pantalla táctil se interpreta como una rotación en X en el sistema global de

coordenadas al modificar la variable PlanetRender.angleX. El desplazamiento en Y sobre la pantalla táctil se interpreta como una rotación en Y en el sistema global de coordenadas al modificar la variable PlanetRender.angleY. El pellizco sobre la pantalla táctil se interpreta como un desplazamiento de la posición de observación en el sistema global de coordenadas al modificar la variable PlanetRender.distancia.

2.3.5 Clase Planet

La clase Planet se utiliza para crear el objeto geométrico “Planet” según la descripción en 1.2.2 y la referencia [Smithwick, 2012]. La figura 24 muestra los métodos de esta clase que se describen a continuación.

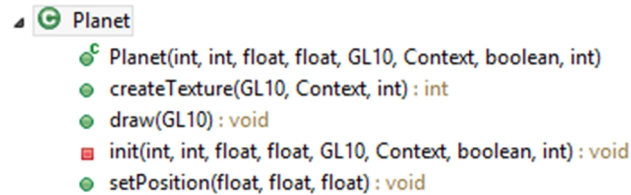


Figura 24. Clase Planet.

public Planet(int stacks, int slices, float radius, float squash, GL10 gl, Context context, boolean imageld, int resourceId). Es el constructor de la clase. Inicia las variables de acuerdo a los parámetros que recibe y llama al método init.

public int createTexture(GL10 gl, Context contextRegf, int resource). Crea la textura a partir de un recurso del tipo bitmap indicado por la variable resource.

public void draw(GL10 gl). Utiliza la primitiva OpenGL llamada GL_TRIANGLE_FAN para dibujar los parches triangulares que conforman el objeto geométrico “Planet”

private void init(int stacks,int slices, float radius, float squash, GL10 gl, Context context, boolean imageld, int resourceId). Crea los vectores para almacenar los vértices, vectores normales y colores a partir de la generación de las pilas y rebanadas descritas en la sección 1.2.2.

public void setPosition(float x, float y, float z). Establece la posición en el sistema global de coordenadas del objeto geométrico “Planet”.

2.3.6 Clase PlanetRender

La clase PlanetRender se utiliza para realizar la animación fotorrealista (render) de las geometrías involucradas en la aplicación. Además, se realizan las transformaciones de coordenadas apropiadas para animar la aplicación y generar la simulación de la trayectoria solar diaria. La figura 25 muestra los métodos de la clase que se explican a continuación.

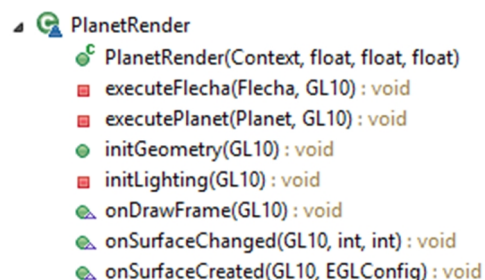


Figura 25. Clase PlanetRender.

public PlanetRender(Context context, float longitud, float latitud, float mes). Es el constructor de la clase. Inicia las variables longitud, latitud y mes a partir de la aplicación que crea éste objeto.

private void executeFlecha(Flecha m_Flecha, GL10 gl). Crea el objeto “Flecha” y lo traslada a la posición definida en las variables Flecha.m_Pos[i], i = 0, 1, 2 y lo dibuja en el sistema de coordenadas OpenGL.

private void executePlanet(Planet m_Planet, GL10 gl). Crea el objeto “Planet” y lo traslada a la posición definida en las variables Planet.m_Pos[i], i = 0, 1, 2 y lo dibuja en el sistema de coordenadas OpenGL.

public void initGeometry(GL10 gl). Establece la posición inicial de los objetos geométricos en el sistema de coordenadas de OpenGL como se muestra en la figura 26. El código para generar la figura 26 se muestra en la sección 2.4. Carga los recursos correspondientes a las texturas del planeta y de la flecha. Calcula la longitud offset de acuerdo a la ecuación (7) y calcula el vector director $\underline{v}$ mediante la ecuación (3).

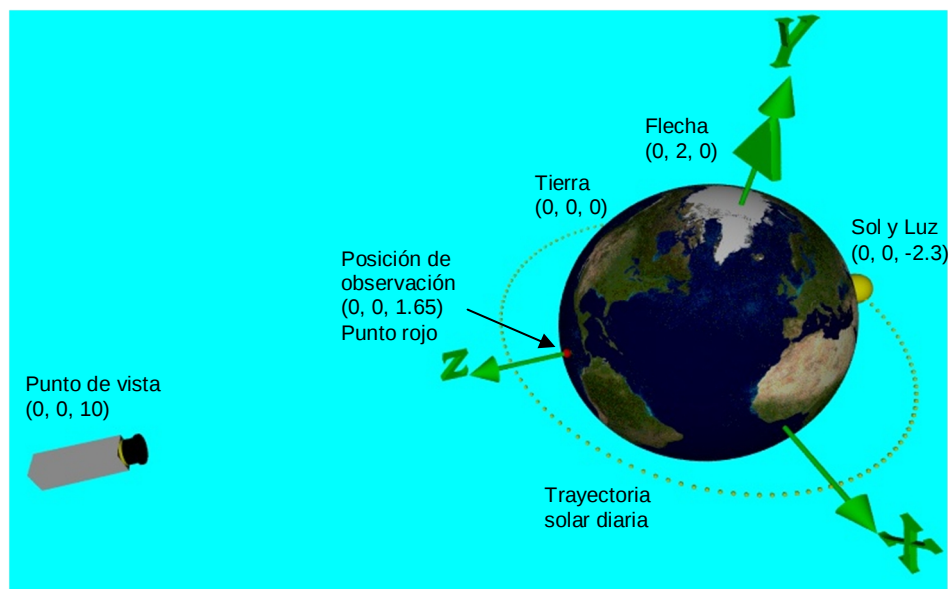


Figura 26. Posición de los objetos en el sistema de coordenadas.

private void initLighting(GL10 gl). Inicia las propiedades de tres fuentes de luz: dos fijas en las posiciones posFill1=[20.0, 50.0, 50.0] y posFill2=[-20.0, 50.0, 50.0] y una móvil siguiendo la trayectoria del sol.

public void onDrawFrame(GL10 gl). Este método se ejecuta periódicamente para generar la trayectoria diaria del sol. En cada iteración se incrementa la variable $\alpha = \alpha + 1.2$ (°). Se calcula la trayectoria solar mediante las ecuaciones (4) y (5). Se rota el ambiente global usando las variables angleX y angleY generadas por el desplazamiento táctil del usuario en la clase MyGLSurfaceView. Se traslada el punto de observación mediante la variable distancia generada por el pellizco táctil del usuario en la clase MyGLSurfaceView. Compensa el offset en la longitud al rotar la tierra para alinear la textura con la longitud real ingresada por el usuario. Rota la posición de observación denotada por el punto rojo de la figura 26 para hacerla coincidir con la latitud ingresada. Posiciona la flecha que indica el norte geográfico y dibuja todos los elementos.

public void onSurfaceChanged(GL10 gl, int width, int height). Establece las propiedades del área de proyección del ambiente global de coordenadas sobre la pantalla.

public void onSurfaceCreated(GL10 gl, EGLConfig config). Inicia las condiciones geométricas y de iluminación.

2.4 Descripción del software de apoyo

En la edición del presente informe se utilizó un software de traza de rayos conocido como PovRay [POV-Ray, 2013] para generar las figuras fotorrealistas que apoyan las explicaciones. PovRay es un ambiente de programación que genera ambientes fotorrealistas a partir de descripciones geométricas de objetos, fuentes de iluminación y texturas ingresadas en modo texto. De esta manera es posible describir un objeto a partir de sus coordenadas globales en modo texto y obtener la figura en la pantalla de la computadora.

Un aspecto importante a considerar en el desarrollo del presente proyecto son los sistemas de coordenadas en OpenGL y PovRay ya que son distintos, como se muestra en la figura 27. Debido a que la simulación de la trayectoria solar está sobre la base del sistema de coordenadas de OpenGL, es necesario considerar el cambio del eje X por el eje Z cuando se ejemplifica en PovRay.

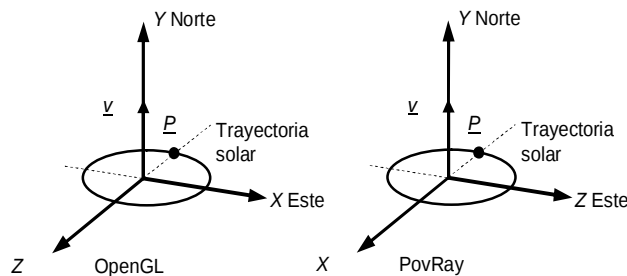


Figura 27. Sistemas de coordenadas.

Se desarrollaron dos programas que principalmente generan ambientes globales de coordenadas para apoyar lo descrito en el presente informe.

2.4.1 informe.pov

El primer programa “informe.pov”, que se encuentra en el directorio “Figuras” del CD anexo, se utiliza para describir la trayectoria solar a partir de los siguientes datos de entrada:

- Posicion. La posición angular del sol sobre la trayectoria solar en donde $0 \leq \text{Posición} \leq 2\pi$.
- Latitud. Corresponde a la variable Latitud descrita en la sección 1.1.2.
- Mes. Corresponde a la variable mesDia descrita en la sección 1.1.1, pero multiplicada por un factor de escala 15, apropiado para el sistema de coordenadas definido en PovRay.

Y de los siguientes elementos geométricos que se dibujan en el ambiente global:

- Flecha. Para representar los solsticios y las coordenadas geográficas. El objeto resulta de la unión CSG de dos intersecciones entre cilindros (cylinder) y cajas (box).
- Plano. Para representar el plano de tierra. El objeto resulta de la intersección CSG entre una esfera (sphere) y una caja.
- Poste. Para representar la posición de observación. El objeto resulta de la unión entre la intersección de una esfera y una caja con un cilindro.
- Boveda solar. Para representar la esfera celeste que circunscribe la trayectoria circular del sol. El objeto resulta de una esfera semi-transparente.
- Sol. El sol resulta de una fuente de luz combinada con esfera.
- Ruta del sol. La trayectoria diaria del sol y su posición se generan a partir de los siguientes elementos: los datos de entrada, la translación indicada por la variable Mes, la rotación indicada por la variable Latitud, el radio indicado por la variable Mes de la ecuación (6) y la posición en la trayectoria solar indicada por la variable posición.

Con el primer programa se generaron las figuras 3 y 7 del presente informe.

2.4.2 informe2.pov

El segundo programa “informe2.pov”, que se encuentra en el directorio “Figuras” del CD anexo, se utiliza para definir las condiciones iniciales del ambiente global de la aplicación.

Los siguientes elementos geométricos se dibujan en el ambiente global:

- Cámara. Para representar el punto de observación que se establece en el simulador. El objeto se compone de una caja y tres cilindros.
- Flecha. Para representar la flecha que indica el norte geográfico. El objeto se compone de la intersección de un cono y una caja.
- Ejes coordenados. Para representar los ejes coordenados X Y y Z . Los objetos se componen de un cono, un cilindro y el texto que indica el eje coordenado cada uno.
- Tierra. Para representar el planeta tierra. El objeto se compone de una esfera cuya textura es el mapa bidimensional del planeta.
- Sol. Para representar el sol. El objeto se compone de una esfera amarilla.
- Posición. Para representar la posición terrestre de observación. El objeto se compone de una esfera roja.
- Ruta del sol. La trayectoria diaria del sol se genera con una serie de esferas amarillas dibujadas en el plano X - Z .

Con el segundo programa se generó la figura 26 del presente informe.

3 Resultados y discusión

En la presente sección se describen los resultados obtenidos de la aplicación para la simulación de la trayectoria solar diaria.

3.1 Resultados

Se desarrolló una aplicación para Smartphone que simula la trayectoria solar diaria en un ambiente gráfico 3D. La simulación se ejecuta bajo el esquema de gráficos con animación en tiempo de ejecución del dispositivo móvil. Lo anterior significa que la trayectoria diaria se completa en unos cuantos segundos, de manera que el usuario puede observar el movimiento del sol a una velocidad superior de lo que ocurre en la realidad (aproximadamente 3 s por trayectoria diaria). La serie de pantallas capturadas que se muestran en la figura 28 muestran el progreso de una simulación en donde se especifica la fecha 30 de marzo en la Ciudad de México (aproximadamente durante el equinoccio). La figura 29 muestra la simulación de la trayectoria para la misma localidad el 21 de diciembre (solsticio de invierno). La figura 30 muestra la simulación de la trayectoria para la misma localidad el 21 de junio (solsticio de verano).



Figura 28. Trayectoria solar para el 30 de marzo en la Ciudad de México.

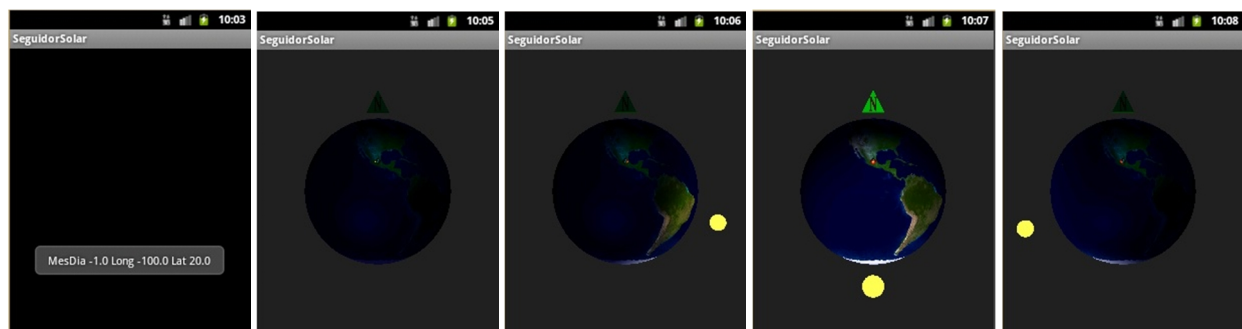


Figura 29. Trayectoria solar para el 21 de diciembre en la Ciudad de México.

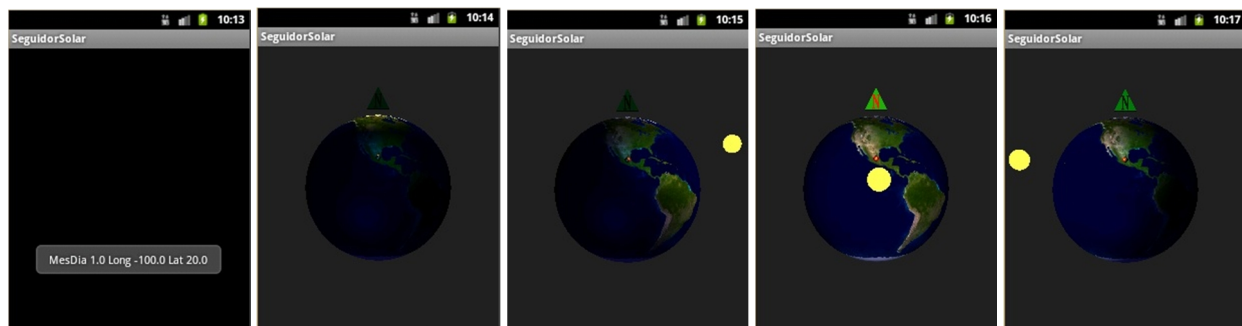


Figura 30. Trayectoria solar para el 21 de junio en la Ciudad de México.

En las figuras 28 a 30 se aprecia el efecto de desplazamiento que sufre la trayectoria solar durante el año: Para el equinoccio el sol amanece exactamente por el este y se oculta por el oeste. Para el solsticio de invierno el sol alcanza su máximo desplazamiento hacia el sur y para el solsticio de verano el sol alcanza su máximo desplazamiento hacia el norte.

La variación en latitud se muestra en las series de figuras 31 a 33 donde se especifica la localidad en la Ciudad de Sídney Australia. La figura 31 muestra la simulación de la trayectoria para el 21 de marzo (equinoccio). La figura 32 muestra la simulación de la trayectoria en la misma localidad el 21 de diciembre (solsticio de invierno). La figura 30 muestra la simulación de la trayectoria para la misma localidad el 21 de junio (solsticio de verano).

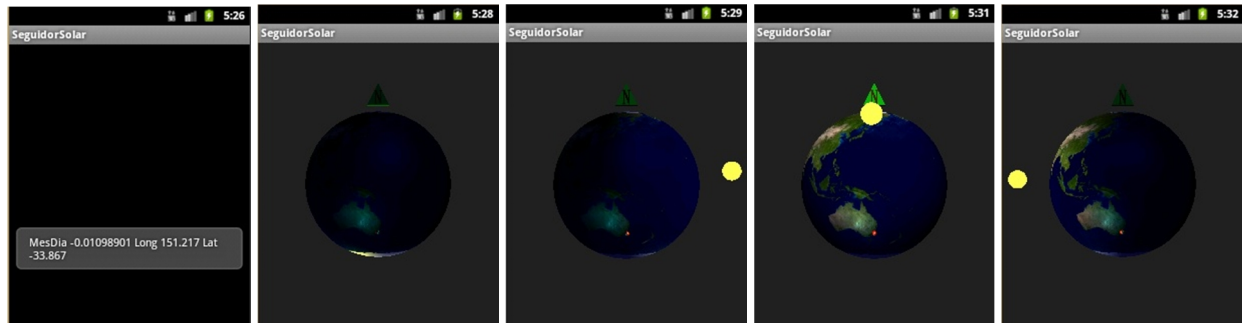


Figura 31. Trayectoria solar para el 21 de marzo en la Sídney Australia.

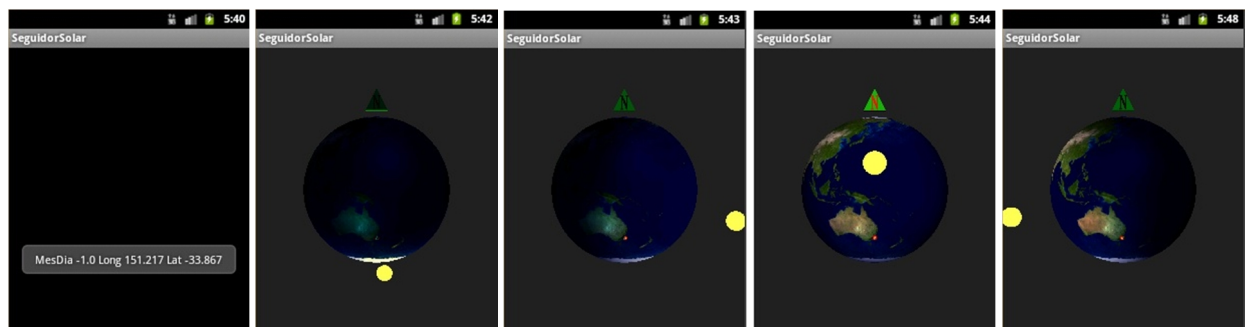


Figura 32. Trayectoria solar para el 21 de diciembre en la Sídney Australia.

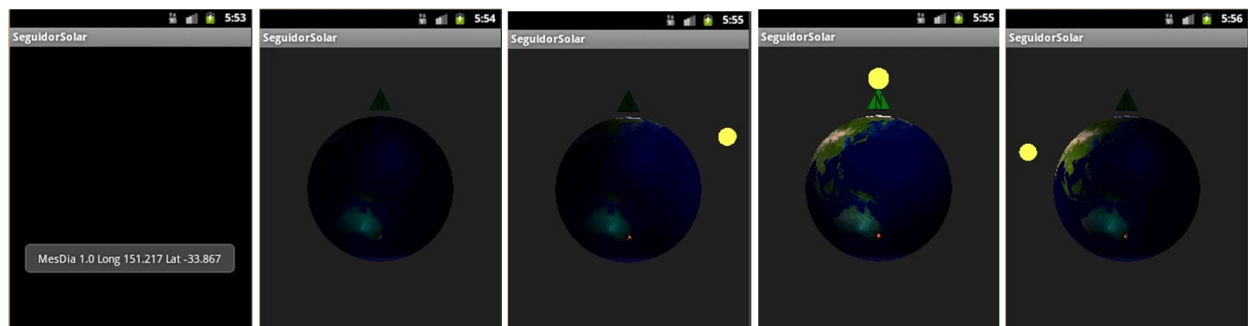


Figura 33. Trayectoria solar para el 21 de junio en la Sídney Australia.

Otro parámetro a evaluar es la obtención automática de los datos de posición mediante el sensor GPS del dispositivo móvil. La Tabla 2 muestra la comparación entre los datos de posición que se obtienen del dispositivo móvil contra aquellos obtenidos mediante Google Maps.

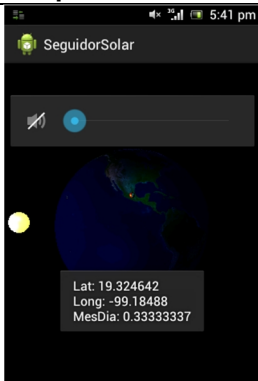
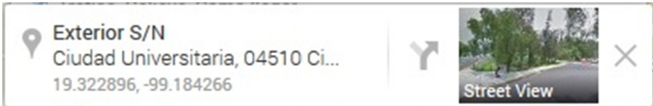
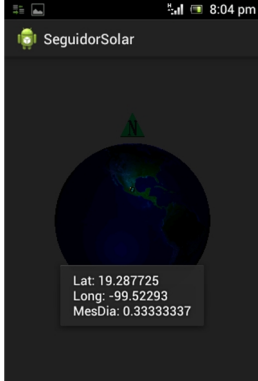
Localidad	Posición del dispositivo móvil	Posición Google Maps
México D.F.		
Toluca		

Tabla 2. Datos de posición geográfica (latitud, longitud).

3.2 Discusión

Los resultados presentados en esta sección presentan buena consistencia. En primer lugar, el aspecto general de la aplicación es bastante realista y su interface con el usuario sigue la filosofía de las aplicaciones para entorno Android en donde se busca crear interfaces intuitivas en un diseño simple.

Con respecto al modelo de la trayectoria solar, la aplicación desarrollada muestra que el efecto de la variación de la latitud del punto de observación efectivamente se manifiesta como una rotación alrededor del eje que pasa por el oeste y este. Por otra parte, la variación de la fecha efectivamente se representa en el simulador como una traslación de norte a sur.

Finalmente, la aplicación proporciona flexibilidad al poder simular diferentes trayectorias para diversas fechas del año y posiciones geográficas.

Conclusiones

Se desarrolló una aplicación para Smartphone que simula la trayectoria aparente diaria del sol en un ambiente gráfico 3D fotorrealista. La aplicación es una materialización de las ecuaciones solares que describen matemáticamente el comportamiento del sol a nuestro alrededor. La teoría descrita aquí nos permite comprender el comportamiento mencionado y a partir de esta asimilación de conocimiento, se pueden desarrollar dispositivos electromecánicos que aprovechen la energía del sol. Un ejemplo es la energía solar en donde se utilizan dispositivos conocidos como “seguidores solares” que permiten concentrar la energía del sol con fines de generación de electricidad o calor.

La aplicación no proporciona ningún beneficio tangible para el usuario. Sin embargo, la importancia de desarrollar la aplicación radica en la experiencia adquirida por los participantes ya que esto nos permitirá abordar proyectos en el área de energías renovables.

Referencias

Cornwall C, Horiuchi A y Lehman C. (2014) "Solar Position Calculator" en National Oceanic and Atmospheric Administration. [On line].
Available at: <http://www.esrl.noaa.gov/gmd/grad/solcalc/azel.html>. [Accessed on Agosto 22, 2014].

Montero MR. (2012) *Desarrollo de aplicaciones para Android*, Editorial Ra-Ma, España, pp. 350.

POV-Ray. (2013) "POV-Ray2 en Persistence of Vision. [On line]
Available at: <http://www.povray.org/>. [Accessed on Agosto 22, 2014].

Rogers DF y Adams JA. (1989) *Mathematical Elements for Computer Graphics*, Mc Graw-Hill, USA, 512 pp.

SKY-MAP.ORG. (2014) "Sky Map" en Sky-Map.org. [On line].
Available at: <http://www.sky-map.org/>. [Accessed on Agosto 22, 2014].

Smithwick M y Verma M. (2012) *Pro OpenGL ES for Android*. Apress, USA, 308 pp.

Tamada R. (2012) "Android GPS, Location Manager Tutorial" en Android Hive. [On line].
Available at: <http://www.androidhive.info/2012/07/android-gps-location-manager-tutorial/>. [Accessed on Agosto 22].

University of Nebraska-Lincoln. (2010) "Motions of the sun simulator" en Nebraska Astronomy applet project. [On line].
Available at: <http://astro.unl.edu/naap/motion3/animations/sunmotions.swf>. [Accessed on Agosto 22, 2014].

Vito Technology. (2014) "Star Walk Stargazing guide" en Vito Technology. [On line].
Available at: <http://vitotechnology.com/star-walk.html>. [Accessed on Agosto 22, 2014].