

Universidad Autónoma Metropolitana Unidad Azcapotzalco

División de Ciencias Básicas e Ingeniería

Licenciatura en Ingeniería en Computación

**Desarrollo de aplicaciones SIG bajo globo virtual
WorldWind en NASA Ames Research Center**

Proyecto de Integración en modalidad de Estancia Profesional

Trimestre 18-Invierno

Alumno y autor: Del Castillo Hoffman Guillermo Miguel 209330771

al209330771@alumnos.azc.uam.mx

migueldelcastillo@live.com

Asesor: Dr. Francisco Cervantes de la Torre

Profesor – investigador, Departamento de Sistemas

UAM Azcapotzalco

fcdt@correo.azc.uam.mx

Coasesor: M.Sc. Patrick Hogan

WorldWind Project Manager, National Aeronautics and Space Administration

patrick.hogan@nasa.gov

6 de abril del 2018

Yo, FRANCISCO CERVANTES DE LA TORRE, *declaro que aprobé el contenido del presente Reporte de Proyecto de Integración y doy mi autorización para su publicación en la Biblioteca Digital, así como en el Repositorio Institucional de UAM Azcapotzalco*



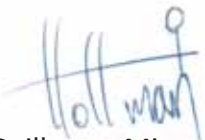
Dr. Francisco Cervantes De La Torre

Yo, PATRICK HOGAN, *declaro que aprobé el contenido del presente Reporte de Proyecto de Integración y doy mi autorización para su publicación en la Biblioteca Digital, así como en el Repositorio Institucional de UAM Azcapotzalco*



M.Sc. Patrick Hogan

Yo, GUILLERMO MIGUEL DEL CASTILLO HOFFMAN, *doy mi autorización a la Coordinación de Servicios de Información de la Universidad Autónoma Metropolitana, Unidad Azcapotzalco para publicar el presente documento en la Biblioteca Digital, así como en el repositorio Institucional de UAM Azcapotzalco.*



Guillermo Miguel Del Castillo Hoffman

Tabla de contenido

I.	Resumen	2
II.	Antecedentes	3
	<i>WorldWind</i> , el globo virtual de la NASA.....	3
	Publicaciones relacionadas	3
	Usuarios actuales y trabajos previos.....	4
III.	Justificación	5
IV.	Objetivos	5
	Objetivo general	5
	Objetivos específicos	5
V.	Marco teórico.....	6
	Elementos orbitales	6
	Datos orbitales en sistemas de cómputo	7
	Sistemas de Información Geoespacial	8
VI.	Desarrollo del proyecto.....	9
	Módulo de adquisición de datos.....	9
	Módulo de visualización orbital	9
	Módulo de interfaz de usuario.....	9
VII.	Resultados	11
VIII.	Análisis y discusión de resultados	12
	Estructura del proyecto	12
	Modificaciones realizadas a <i>Web WorldWind</i>	16
	Áreas de oportunidad para mejoras	17
	Desarrollos adicionales	18
IX.	Recursos	18
X.	Entregables	19
XI.	Agradecimientos	19
	Universidad Autónoma Metropolitana	19
	Agencia Espacial Mexicana	19
	National Aeronautics and Space Administration	20
	European Space Agency	20
XII.	Bibliografía y referencias.....	20
XIII.	Referencias de tablas	23
XIV.	Referencias de imágenes.....	23

I. Resumen

A lo largo de una estancia profesional llevada a cabo desde julio hasta septiembre del 2016 y concertada mediante un acuerdo entre la Agencia Espacial Mexicana [1], la Universidad Autónoma Metropolitana Unidad Azcapotzalco, y la *National Aeronautics and Space Administration* (NASA) [2], se diseñaron y desarrollaron en conjunto con un equipo de estudiantes internos en el Centro de Investigación Ames en Moffet Field, California [3], tres aplicaciones SIG (Sistemas de Información Geoespacial) bajo la versión web de *WorldWind*, un SDK de aplicaciones basado en un globo terráqueo virtual en 3D de código abierto desarrollado por la NASA.

El alumno colaboró con el desarrollo de los tres proyectos, dirigiendo el diseño y desarrollo de uno de ellos: Un rastreador satelital web en tiempo real. El proyecto exitosamente concluyó y se presentó durante la estancia, logrando la representación en tiempo real de alrededor de 15,000 satélites en órbita terrestre (cargas útiles, etapas de cohetes y escombros espaciales) actualmente rastreados por el Comando Espacial de Estados Unidos (*USSPACECOM*), describiendo todo el entorno orbital terrestre conocido (objetos en órbita mayores de ~5 cm de diámetro que pueden ser rastreados por radar) [4].

La aplicación desarrollada, utilizando el globo virtual de NASA *WorldWind* como base, representa fielmente en tres dimensiones y en tiempo real el entorno orbital terrestre (con datos de septiembre del 2016, fecha del término de la estancia, pero existiendo la posibilidad de operar con un conjunto de datos actualizado), en una aplicación web con la capacidad de visualizarse en cualquier dispositivo con un navegador web moderno (en computadoras de escritorio o dispositivos móviles).

El proyecto es de código abierto y su repositorio de código de la aplicación se encontrará disponible de manera indefinida en *GitHub*, estando señaladas las aportaciones de código proporcionadas por el alumno [5]:

<https://github.com/NASAWorldWindResearch/SpaceBirds>

Se puede visualizar en operación en el siguiente enlace con cualquier navegador web que soporte WebGL:

<https://worldwind.arc.nasa.gov/spacebirds/>

Se cumplieron los objetivos de la estancia, diseñando, desarrollando y entregando una aplicación de código abierto funcional, actualmente en operación y sirviendo como base para futuros desarrollos para quien desee contribuir o modificar el proyecto.

II. Antecedentes

WorldWind, el globo virtual de la NASA

WorldWind es definido por la NASA como una Interfaz de Aplicación (API) y Kit de Desarrollo de Software (SDK) libre y de código abierto (licencia NOSA 1.3) [6] para aplicaciones basadas en un globo terráqueo virtual [7]. “Permite a un desarrollador crear visualizaciones interactivas para un globo tridimensional, mapas e información geográfica.” [8].

De acuerdo con información obtenida durante la estancia, el desarrollo de *WorldWind* se inició en el 2003 como iniciativa del coasesor de la presente propuesta para su uso en computadoras personales como un proyecto del Departamento de Educación de NASA, siendo el propósito inicial del el desarrollo del globo en 3D de código abierto como herramienta educativa. La versión original estaba desarrollada en C#, dependiendo del *framework* .NET y operando exclusivamente bajo Windows. Posteriormente se reescribe bajo Java (siendo esta versión aún vigente y en uso por múltiples organizaciones), cambiando los propósitos del proyecto a un **SDK multiplataforma orientado a desarrolladores**, al contrario de una aplicación destinada al usuario final del estilo de *Google Earth*. El SDK incluye un conjunto de demostraciones básicas y ejemplos para el desarrollo de aplicaciones. En años posteriores gracias a la proliferación de plataformas móviles, se desarrollaron versiones para Android y iOS, recientemente añadiéndose una versión *web* desarrollada en JavaScript bajo la API de gráficos *WebGL*, operando en cualquier navegador moderno sin la necesidad de *plugins*. Actualmente son vigentes tres versiones independientes entre sí, aunque con características, arquitectura y capacidades similares:

Nombre	Plataforma	Lenguaje	Página
WorldWind Java	PCs de escritorio (Win, Linux, Mac)	Java	https://worldwind.arc.nasa.gov/java
Web WorldWind	Navegadores <i>web</i>	JavaScript	https://worldwind.arc.nasa.gov/web
WorldWind Android	Dispositivos móviles con <i>Android</i>	Java	https://worldwind.arc.nasa.gov/android

Tabla 1. Versiones de *WorldWind*

Publicaciones relacionadas

De acuerdo con la publicación en la IEEE, *NASA WorldWind: Opensource GIS for Mission Operations* [9], *WorldWind* combina el uso de imágenes de mapas obtenidas ya sea cartográficamente o mediante imágenes satelitales, con un modelo digital de elevación (*DEM*) basado en el radio nominal de globo y aumentado con datos de elevación de alta resolución en lugares donde éstos se encuentren disponibles. La NASA opera servidores que proven los datos tanto de imágenes como de elevación para *WorldWind*. Adicionalmente, el SDK también provee de las facilidades necesarias para consumir datos de imágenes y elevación desde fuentes externas, obedeciendo a estándares abiertos de la comunidad geoespacial.

PowerGlobe, un programa basado en la versión de escritorio de *WorldWind* que permite a los usuarios crear y visualizar presentaciones sobre el globo terráqueo, fue desarrollado como proyecto de tesis por Sofiia Sandomirskaja, estudiante de Ingeniería de *Software* de la *National Research University Higher School of Economics (HSE)* en la Federación Rusa. Provee a *WorldWind* de capacidades similares a las de *PowerPoint* en un contexto de información geoespacial: “[Las presentaciones hechas en *PowerGlobe*] pueden ser utilizadas en cualquier área donde la visualización de datos de la superficie terrestre sea requerida: pronósticos de clima, paseos virtuales, monitorización de congestión de tráfico, polución del aire, niveles de desempleo, [...] o cualquier otro indicador, dependiendo del terreno. [...] *PowerGlobe*

permite editar diapositivas, y colocar transiciones entre, así como la adición de elementos por parte del usuario, tales como texto, fotos, y capas *WMS* [10].

En la página de proyectos terminales concluidos de Ingeniería en Computación [11] no se encontraron desarrollos relacionados con visualización de datos de sistemas SIG en globos terráqueos o mapas.

Usuarios actuales y trabajos previos

Si bien *WorldWind* no es muy conocido en el ámbito académico o comercial de SIG (donde los productos de Google, ESRI y Microsoft son más conocidos, con *Google Earth*, *ArcGIS* y *Virtual Earth* respectivamente). De acuerdo con información obtenida durante la estancia, *WorldWind* es ampliamente utilizado en el ámbito gubernamental, aeroespacial y de defensa, contando entre sus usuarios a la Administración Federal de Aviación (FAA), el Departamento de Defensa de EUA (DoD), la Agencia Espacial Europea (ESA) [12], el consorcio de defensa de origen francés *Thales Group*, la Secretaría de Marina – Armada de México, Petróleos Mexicanos [13], además de la misma NASA.

Varios SIG se han desarrollado utilizando *WorldWind* como componente de mapas, sirviendo como ejemplo **NESAT3D**, sistema de administración de espacio aéreo desarrollado por la Administración Federal de Aviación de Estados Unidos (FAA, *Federal Aviation Administration*). NESAT3D permite la visualización del tráfico aéreo en tiempo real, obteniendo la información de la posición de las naves a partir de sus transpondedores con una cobertura nacional. La visualización de corredores aéreos en 3D, las cartas de navegación y los mapas topográficos de altitud se encuentran embebidos en la aplicación. Provee de herramientas para realizar mediciones de distancias y posee una variedad de mapas de cobertura de nubes y radares [14].

Shawn Gano [15], ingeniero en computación de NASA Johnson Space Center, realizó de manera aparte a su labor en la NASA, una aplicación de rastreo satelital para equipos de escritorio con capacidades avanzadas de predicción de posiciones en tiempo real, el pasado, o el futuro, utilizando *WorldWind Java* como base para una aplicación de PCs de escritorio llamada **JSatTrak**. Éstas son algunas de sus características de acuerdo con la página del proyecto [16]:

- Elementos orbitales de más de 3,000 satélites fácilmente actualizables.
- Herramienta de rastreo – incluye ángulos de rastreo, trazado polar, y predicciones de paso de satélites (en ángulo visible y por radar) considerando limitaciones de acuerdo a la elevación del terreno.
- Satélites personalizados con una interfaz de diseño de misiones para el modelado de maniobras.
- Consola de comandos.
- Análisis de cobertura terrestre.
- Múltiples vistas simultáneas en 2D, 3D y de visualización de datos.

Si bien no se utilizó ni el diseño ni el código de *JSatTrak* para el proyecto relativo al presente reporte, fue su principal inspiración.

III. Justificación

La Agencia Espacial Mexicana ha realizado una ardua labor para propiciar la formación de profesionistas mexicanos del campo espacial [17] llegando a convenios con instituciones internacionales relacionadas, siendo el principal exponente la NASA. La difusión lograda para la Universidad Autónoma Metropolitana al quedar seleccionado un alumno de la misma para una estancia en la NASA, forma parte de la justificación para dicha estancia.

Los proyectos a realizarse funcionan como demostraciones técnicas de la incipiente versión *Web* de *WorldWind* ante posibles patrocinadores, beneficiando la continuidad del desarrollo de *WorldWind* como proyecto de la NASA. Al estar basados en una plataforma de código abierto con una licencia *Open Source* de la NASA (licencia NOSA 1.3) [6], *WorldWind* y los proyectos basados en el mismo, no tienen restricciones de exportación por parte de la agencia federal espacial de Estados Unidos. Pueden distribuirse, utilizarse y modificarse en México (y en otros países) incentivando el desarrollo de plataformas de información geoespaciales para propósitos educativos, comerciales o científicos.

Los proyectos sirven como herramientas de visualización y análisis geoespacial en 3D con la conveniencia de estar disponibles para cualquier dispositivo con un navegador, sin necesidad de descargar ninguna aplicación o *plugin* adicional.

IV. Objetivos

Objetivo general

Diseñar y desarrollar una aplicación SIG de rastreo satelital, *SpaceBirds*, utilizando como base la versión web del globo 3D de la NASA, *WorldWind*.

Objetivos específicos

- Diseñar y desarrollar un módulo para la **adquisición de los datos orbitales** de todos los satélites artificiales en órbita terrestre para la aplicación *SpaceBirds*.
- Diseñar y desarrollar un módulo para la **visualización del entorno orbital global** (todos los satélites artificiales rastreados en órbita terrestre, incluyendo escombros orbitales) para la aplicación *SpaceBirds*.
- Diseñar y desarrollar un **módulo de interfaz de usuario** que permita interactuar al mismo con conjuntos de satélites y navegar el entorno orbital a través del tiempo (hacia el pasado y futuro), además de desplegar la información asociada a un satélite seleccionado.

V. Marco teórico

Elementos orbitales

Para la identificación única de una órbita específica se hacen uso parámetros orbitales también conocido como **elementos orbitales**. En mecánica celeste, estos elementos son considerados generalmente como sistemas clásicos de dos cuerpos, donde una órbita de Kepler es utilizada. Los elementos orbitales tradicionales son los seis **elementos Keplerianos**, llamados así por Johannes Kepler y su trabajo acerca de las leyes del movimiento planetario [18].

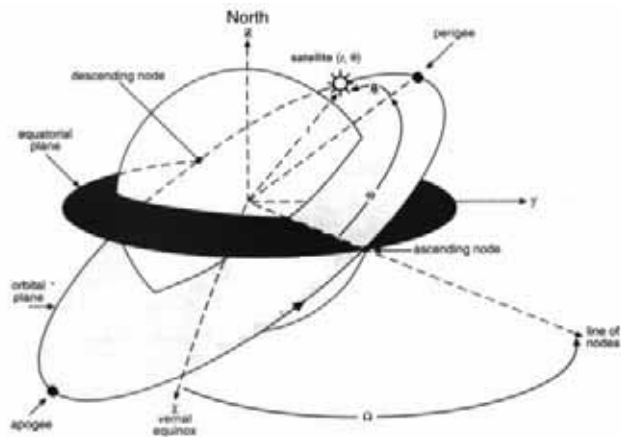


Fig. 1. Parámetros orbitales. Imagen: NASA.

Cuando son observados desde un marco de referencia inercial, dos objetos orbitales recorren trayectorias elípticas distintas. Cada una de esas trayectorias tiene su foco en un centro de masa común. Cuando se observa el sistema desde un marco de referencia no inercial centrado en uno de los cuerpos, únicamente la trayectoria del cuerpo opuesto es aparente; Los elementos Keplerianos describen estas trayectorias no inerciales. Una órbita contiene dos conjuntos de elementos Keplerianos dependiendo qué cuerpo se utiliza como marco de referencia. El cuerpo de referencia es llamado *primario*, mientras que el otro cuerpo es llamado *secundario*. El primario no necesariamente posee mayor masa con respecto al secundario, e incluso cuando ambos cuerpos son de igual masa, los elementos orbitales dependen en la elección del primario.

Los dos principales elementos que definen la forma y tamaño de la elipse son los siguientes.

- **Excentricidad** – describe qué tan elongada se encuentra la elipse con respecto un círculo.
- **Eje semimayor** – la suma de las distancias de periapsis y apoapsis (los puntos más cercanos y lejanos al primario en la trayectoria orbital del secundario, respectivamente) dividida entre dos. Para órbitas circulares, el eje semimayor es la distancia entre los centros de los cuerpos.

Dos elementos definen la orientación del plano orbital en el que se encuentra la elipse:

- **Inclinación** – la inclinación vertical de la elipse con respecto al plano de referencia, medida desde el nodo de ascenso (donde la órbita cruza ascendentemente a través del plano de referencia). El ángulo de inclinación se mide perpendicularmente a la línea de intersección entre el plano orbital y el plano de referencia. Tres puntos cualesquier en la elipse definen el plano orbital de la misma. El plano y la elipse son, ambos, objetos bidimensionales definidos en un espacio tridimensional.
- **Longitud del nodo de ascenso** – es el que orienta horizontalmente el nodo de ascenso de la elipse, (donde la órbita cruza ascendentemente a través del plano de referencia) con respecto al punto vernal del marco de referencia.

Finalmente:

- **Argumento de la periapsis** – define la orientación de la elipse en el plano orbital, siendo un ángulo medido desde el nodo ascendente a la periapsis (el punto más cercano al objeto primario en la órbita del objeto satélite).
- **Anomalía verdadera** en una **época** determinada – define la posición del objeto orbital a través de la elipse en un tiempo específico (la “época”).

Dado un marco inercial de referencia y una época arbitraria (un punto específico en el tiempo) exactamente seis parámetros son requeridos para definir sin ambigüedad una órbita arbitraria sin perturbaciones. Esto es dado que el problema contiene **seis grados de libertad**. Éstos corresponden a las tres dimensiones espacial que definen la posición (x , y , z en un sistema coordenado Cartesiano), además de la velocidad en cada una de esas dimensiones. Éstos pueden ser descritos como vectores de estado orbital, pero a menudo se utilizan los elementos Keplerianos por encontrárseles más convenientes.

Datos orbitales en sistemas de cómputo

El estándar actual de codificación de datos orbitales tiene antecedentes en el trabajo seminal de Max Lane a inicios de los 60, tratando del desarrollo de modelos matemáticos que permitieran pronosticar las posiciones de satélites basándose en un conjunto mínimo de datos. Su primera publicación en el tema, datando de 1965, introdujo la Teoría de Arrastre Analítica, que trataba principalmente sobre los efectos del arrastre causado por una atmósfera simétrica (esférica) y no rotatoria. Posteriormente K. Cranford se unió al trabajo de Lane, ambos publicando un modelo mejorado en 1969 que añadía varios efectos armónicos debidos a las interacciones entre el Sol, la Tierra y la Luna, además de varias otros factores [19].

Los modelos de Lane fueron utilizados ampliamente por la milicia y NASA a finales de los 60. La versión mejorada se convirtió en el primer modelo estándar para NORAD a inicios de los 70, el cual desencadenó en el desarrollo del estándar de codificación **Two Line Element set (TLE)** definido conjuntamente entre NORAD y NASA [20].

El formato TLE, codifica una lista de parámetros orbitales, incluidos los 6 elementos Keplerianos que describen la órbita de cualquier objeto en órbita terrestre en un momento dado en el tiempo (*epoch*, conocido en español como “época de las coordenadas”). Mediante el uso de una fórmula adecuada de predicción, el estado del objeto (su posición y velocidad) puede ser estimado con cierta precisión. *WorldWind* no posee funciones de extrapolación para mecánica orbital y esto debe ser integrado a la aplicación.

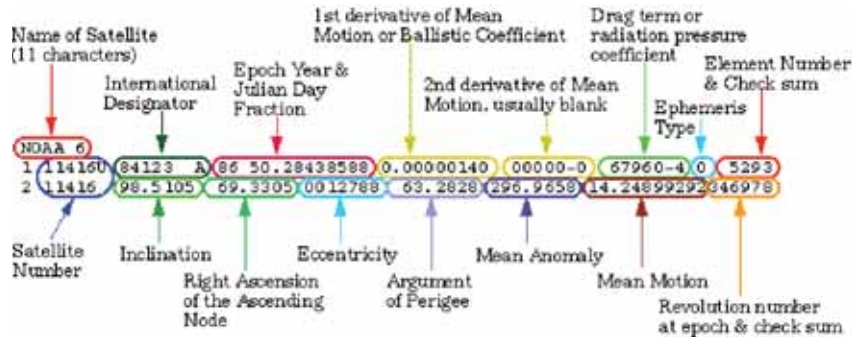


Fig. 2. Formato de datos TLE. Imagen: NASA.

Los datos TLE se pueden utilizar como fuente de datos para modelos de cómputo, conocidos como modelos simplificados de perturbaciones, utilizados para calcular los vectores de estado orbital de satélites relativos a un sistema de coordenadas inercial centrado en la Tierra.

Este conjunto de modelos usualmente es referido como SGP4. Este modelo relativamente ligero en cuanto a requerimientos computacionales se utiliza a menudo para procesar los datos TLE de NORAD y NASA.

Estos modelos predicen el efecto de las perturbaciones ocasionadas por la forma de la Tierra, el arrastre atmosférico, la radiación y efectos gravitacionales de otros cuerpos como el Sol y la Luna. Los modelos de Perturbaciones Generales Simplificadas (SGP por sus siglas en inglés) se aplican a objetos cercanos a la Tierra con periodos orbitales menores a 225 minutos. Las Perturbaciones Simplificadas de Espacio Profundo (SDP por sus siglas en inglés) se aplican a objetos con un periodo orbital mayor a 255 minutos, lo cual corresponde a una altitud de 5,877.5 km, asumiendo una órbita circular.

Los modelos de espacio profundo SDP4 y SDP8 únicamente utilizan modelos de arrastre simplificados. La precisión no es la prioridad en estos casos dado que los casos con alto arrastre no se mantienen en espacio profundo por mucho tiempo, convirtiéndose la órbita en casi circular rápidamente. SDP4 también añade perturbaciones gravitacionales lunares y solares a todas las órbitas, y términos en las ecuaciones de resonancia Terrestre específicamente para órbitas geostacionarias de 24 horas y *Molniya* de 12 horas [21].

Existen muchas bibliotecas de software que implementan los modelos SGP y SDP en diferentes lenguajes de programación. Se puede utilizar la biblioteca *satellite.js* [22], escrita en JavaScript y adaptada a partir de una implementación en Python de los modelos SGP4/SDP4 antes mencionados.

Sistemas de Información Geoespacial

Los **Sistemas de Información Geoespacial (SIG)**, son sistemas diseñados para capturar, almacenar, manipular, analizar, administrar y presentar información geográfica o espacial (algunas veces las siglas se identifican como “Sistemas de Información Geográfica”). En general, el término describe cualquier sistema de información que integra, almacena, edita, analiza, comparte y visualiza información geoespacial. Las aplicaciones SIG son herramientas que permiten a los usuarios realizar búsquedas interactivas, analizar información geoespacial, editar datos en mapas, y presentar los resultados de todas estas operaciones [23]. *WorldWind* puede utilizarse como el componente de mapas dentro de un SIG.

VI. Desarrollo del proyecto

Se propone una aplicación que comprende una parte de *frontend* (la aplicación que se ejecuta en cualquier navegador web y con la que interactúa el usuario) y el *backend* (*scripts* que se ejecutan del lado del servidor).

Módulo de adquisición de datos

Componente del *backend*. El módulo de adquisición de datos opera en el servidor, combinando la información de dos fuentes de datos distintas (y en formatos distintos) para la obtención de los datos ya filtrados y procesados que comprenden la información de todos los objetos en órbita terrestre rastreados actualmente. El archivo resultante es el que alimenta la aplicación web para el despliegue de todos los satélites y su información asociada.

Módulo de visualización orbital

Componente del *frontend*. Este módulo basa su funcionalidad en la versión web de *WorldWind*, componente de mapas diseñado como *SDK* para aplicaciones SIG. El globo 3D es aprovechado para desarrollar la parte principal de la aplicación web que otorga la vista al usuario, con la idea de mostrar el entorno orbital global con todos los satélites en órbita terrestre rastreados por el Comando Espacial de los Estados Unidos (US SPACECOM) [24]. *WorldWind*, como *SDK*, provee de herramientas de visualización de datos para colocar marcas de posición que representan los satélites en su posición geográficamente correcta sobre el globo, actualizándola en tiempo real. Sin embargo, *WorldWind* no posee capacidades para consumir parámetros orbitales (es decir, datos en términos de elementos Keplerianos, ver Fig. 1) por lo que este módulo requiere del procesamiento de parámetros orbitales para representarlos en posiciones geográficas en términos de latitud, longitud y altitud. También se aprovecharán las capacidades tridimensionales de *WorldWind* para representar los trayectos orbitales y los modelos de los satélites en 3D, además del horizonte terrestre del satélite – esto es, la zona aproximada de la Tierra desde donde es posible ver el satélite en el cielo.

Módulo de interfaz de usuario

Componente adicional del *frontend*, pero separado de *WorldWind*, este módulo provee al usuario la posibilidad de filtrar la visualización a satélites de su interés de acuerdo a diversos criterios (*e.g.* para únicamente visualizar satélites de cierto tipo, con ciertas características orbitales, o de cierto país/operador). Del mismo modo, el usuario puede visualizar el trayecto orbital pasado y futuro de un satélite seleccionado, o bien, colocar las posiciones de todos los objetos en órbita en algún momento arbitrario del pasado o futuro.

Se proponen los siguientes casos de uso, tanto del servidor (Fig. 2) como del cliente (Fig. 3).

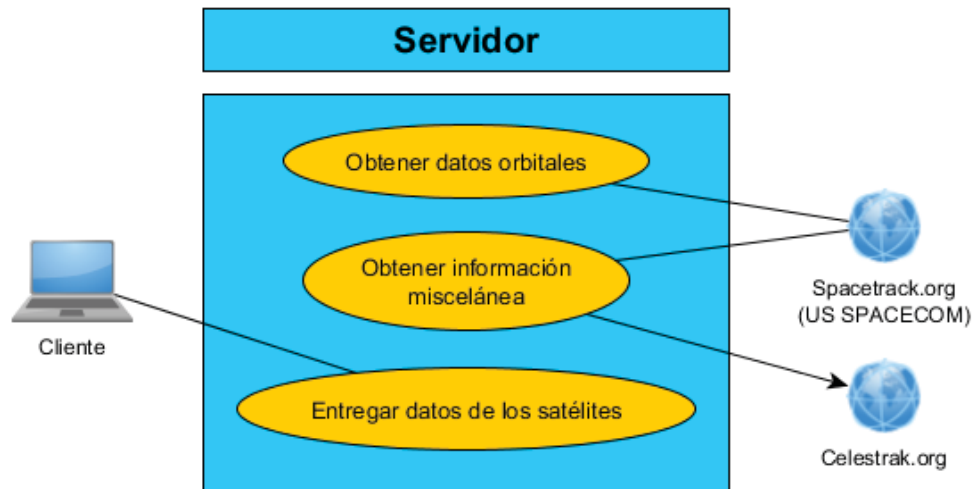


Fig. 3. Diagrama de casos de uso de backend. Imagen: Autoría propia.

En breve, el *backend*, comprendido por el módulo de adquisición de datos, se encarga de obtener la información de los satélites: Tanto los parámetros que permiten calcular su órbita como su información miscelánea asociada (e.g. Identificador del satélite, fecha y lugar de lanzamiento, país u operador, tipo de objeto, etc.) a partir de dos fuentes distintas que se complementan. Ya procesada y filtrada la

información, se entrega al cliente para su despliegue.

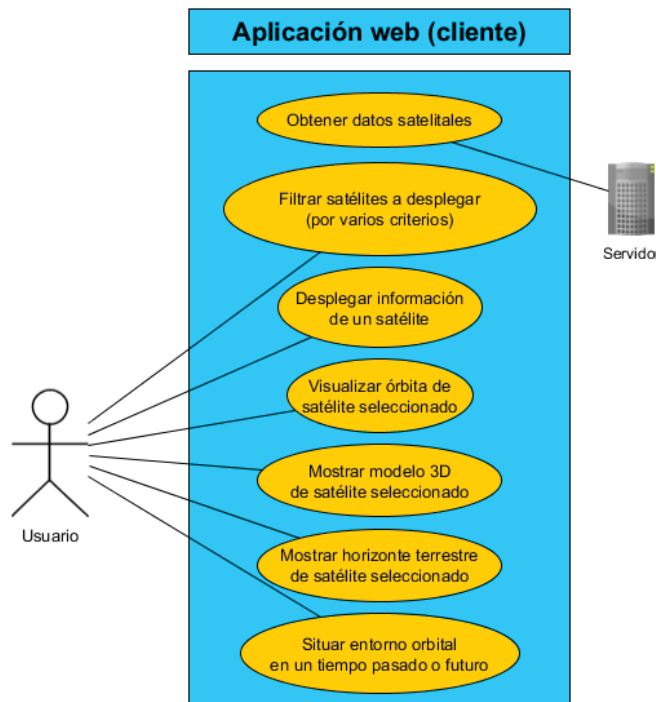


Fig. 4. Diagrama de casos de uso del frontend. Imagen: Autoría propia.

El *frontend*, comprendido por el módulo de visualización del entorno orbital global y de interfaz de usuario, después de obtener los datos satelitales desde el *backend*, desplegará todos los satélites a la vez, actualizando su posición en tiempo real, y permitiendo al usuario filtrar el despliegue de los mismos por diversos criterios (e.g. ‘mostrar todas las etapas de cohetes y cargas útiles’, ‘mostrar únicamente satélites geoestacionarios’, etc.), además de mostrar información asociada a un satélite cuando éste es seleccionado por el usuario, mostrando también su órbita, un modelo 3D del satélite, y el horizonte terrestre del satélite explicado en el módulo de visualización orbital. Adicionalmente, la

interfaz permitirá situar la visualización en un tiempo dado por el usuario (fecha y hora pasada, presente o futura), situando todos los objetos del entorno orbital en el momento en el tiempo deseado.

VII. Resultados

El *frontend*, siendo una aplicación web, se programó en *JavaScript*, lenguaje en el que también se encuentra desarrollado *Web WorldWind*. **La programación de gráficos de bajo nivel de *Web WorldWind* está descrita en *shaders de WebGL***, variante de la API de gráficos *OpenGL* orientada a navegadores web [25]. Para los elementos 2D usuales de una aplicación web (formas, tablas, cajas de texto, etc.) se hace uso de HTML. Se utiliza CSS para estilización y formato visual de estos elementos. Se utilizan bibliotecas UI como *Bootstrap* [26], *Bulma* [27] y *perfect-scrollbar* [28]. También se hace uso de la biblioteca *jQuery* [29] que facilita el manejo tanto de eventos de en el navegador – esto es, facilita las interacciones entre el código en *JavaScript* y la API del navegador para interactuar con él, llamada *Document Object Model* [30]).

En cuanto al *backend*, se eligió *Node* [31], la implementación de *JavaScript* desarrollada inicialmente como lenguaje del lado de servidor (en línea de comandos), manteniendo homogénea la elección de lenguaje tanto en el *backend* como el *frontend* de toda la solución, si bien hay diferencias importantes entre el *runtime* de *Node* y la manera en la que los navegadores web manejan *JavaScript*.

Como entorno de desarrollo, el equipo de la NASA recomendó usar el IDE *WebStorm* [32]. También se hicieron uso de diversas herramientas de desarrollo de *software* libre escritas en *Node*, tales como el administrador de tareas *Grunt* [33] y el servidor mínimo de pruebas *node-http-server* [34].

Como componente de mapas, por supuesto, fue utilizado *Web WorldWind*, encapsulando su funcionalidad mediante un *canvas* de HTML. *WorldWind* provee de su propio conjunto de eventos para selección (realizar *click* sobre objetos de la escena) y control de la escena 3D misma (rotar la orientación de la cámara o el globo mismo), además de contener facilidades para la obtención de datos geoespaciales (típicamente, imágenes satelitales sobre el globo y un modelo digital de elevación).

Para la obtención de datos, se accedieron a los parámetros orbitales de todos los objetos rastreados por el Comando Espacial de los Estados Unidos mediante un servicio web *REST* [35] público (requiriendo éste una cuenta de acceso gratuita). En cualquier momento dado, *USSPACECOM* rastrea alrededor de 15,000 objetos con diámetro mayor a 5 cm en órbita terrestre, entre satélites (en operación y abandonados), etapas de cohetes aún en órbita y otros escombros espaciales. El proveedor del servicio *REST*, *space-track.org* [36], entrega los datos en diversos formatos según se le solicite. Por consistencia con el resto del proyecto, se elige que los datos se provean en formato *JavaScript Object Notation (JSON)* [37]. Es importante señalar que *space-track.org* no permite el consumo de sus datos desde una aplicación web por poseer restricciones habilitadas de *Cross Origin Resource Sharing (CORS)* [38].

Para completar los datos informativos de los satélites, se pueden utilizar los datos de la página *celestrak.com* [39] que están contenidos en archivos de texto plano. Se puede realizar una referencia cruzada entre los dos conjuntos de datos para obtener registros completos. Por la restricción *CORS* del primer proveedor y dado el volumen significativo de datos, esta tarea no se realiza en el lado del cliente,

desarrollando en su lugar un script de *Node* en la línea de comandos con la intención de ejecutarse en el servidor. Se puede automatizar la ejecución de dicho script mediante el automatizador de tareas *CRONjobs* [40] para actualizar el conjunto de datos con la frecuencia que se desee.

Los parámetros orbitales únicamente entregan la velocidad y dirección de los satélites cuando se rastrearon físicamente mediante radar. Para conocer la posición del satélite en cualquier momento dado, y considerando la relativa estabilidad del movimiento de un objeto en órbita, es posible extrapolar los datos para obtener la posición del objeto a lo largo de toda su órbita en cualquier momento dado. Esto permite rastrear al satélite en tiempo real.

Los parámetros orbitales de todos los objetos se entregan en JSON en un formato de datos definido por la NASA y NORAD denominado “conjunto de elemento de dos líneas” o “*TLE*” (Fig. 2).

VIII. Análisis y discusión de resultados

La estancia comprendió el periodo desde el 6 de junio hasta el 13 de septiembre del 2016 en el centro de investigación de la NASA antes mencionado, implicando 40 horas de trabajo por semana durante 14 semanas. El tiempo de trabajo total resulta en 560 horas.

Se realizó la obtención del conjunto de datos de los ~15,000 objetos en órbita terrestre de los dos proveedores externos (*Space-track* y *Celestrak*), procesándolos mediante el script de *Node* que forma parte del *backend* para obtener sus datos asociados (nombre, tipo de objeto, número de identificación, fecha y lugar de lanzamiento, tipo de órbita, operador, etc.), además de eliminar algunos campos no necesarios que incrementan el tamaño de los archivos. Con los datos procesados, se alimentan éstos a la aplicación, aprovechando las facilidades que otorga *WorldWind* para la visualización de todos los objetos simultáneamente, calculando mediante la biblioteca de propagadores SGP4/SDP4 su posición en tiempo real o en el tiempo seleccionado por el usuario. Al seleccionar un satélite con el puntero, se muestra el área sobre la Tierra desde donde se pueden observar, además de su trayecto orbital pasado y futuro.

Estructura del proyecto

La aplicación realizada cumple con los tres objetivos específicos mencionados en el presente reporte. Se hace referencia a cada módulo en los archivos y carpetas especificados en el repositorio de código:

El módulo de adquisición de datos comprende dos elementos, presentes en el *backend*: Un comando de consola que obtiene el archivo JSON del proveedor REST de datos satelitales (localizado en ‘Backend_nodejs_stuff/curl_script/curl_spacetrack_script.txt’ en el repositorio) [41], y un programa en NodeJS que filtra el archivo obtenido para obtener únicamente los satélites actualmente rastreados (archivo en ‘Backend_nodejs_stuff/orbitalObjectsFilter.js’) [42]. La ejecución de ambos se puede automatizar fácilmente mediante un script de *Bash* en un servidor con los privilegios suficientes para hacerlo. En el servidor de NASA en el que *SpaceBirds* se encuentra actualmente en operación [43], no se le proporcionan estos privilegios a los internos por razones de seguridad.

El **módulo de visualización de datos** comprende a la parte en JavaScript y HTML de la aplicación visualizable en un navegador web. Dichos archivos se localizan en la carpeta ‘/standalone/SatTracker2.js’ [44] y ‘/standalone/index.html’ [45]. La biblioteca de *WorldWind* se encuentra alojado en ‘/standalone/util/worldwindlib.js’, ya “minificada” (con todo el *whitespace* eliminado) para ocupar menos memoria en el navegador y requerir de menor tiempo de transferencia en red. El módulo asimila todos los eventos de las interacciones del usuario, además de desplegar visualmente las posiciones de todos los satélites (en tiempo real o en un tiempo elegido por el usuario) como marcas de posición sobre el globo de *WorldWind*. Así mismo, el cálculo y dibujado de las órbitas es realizado por este modulo.

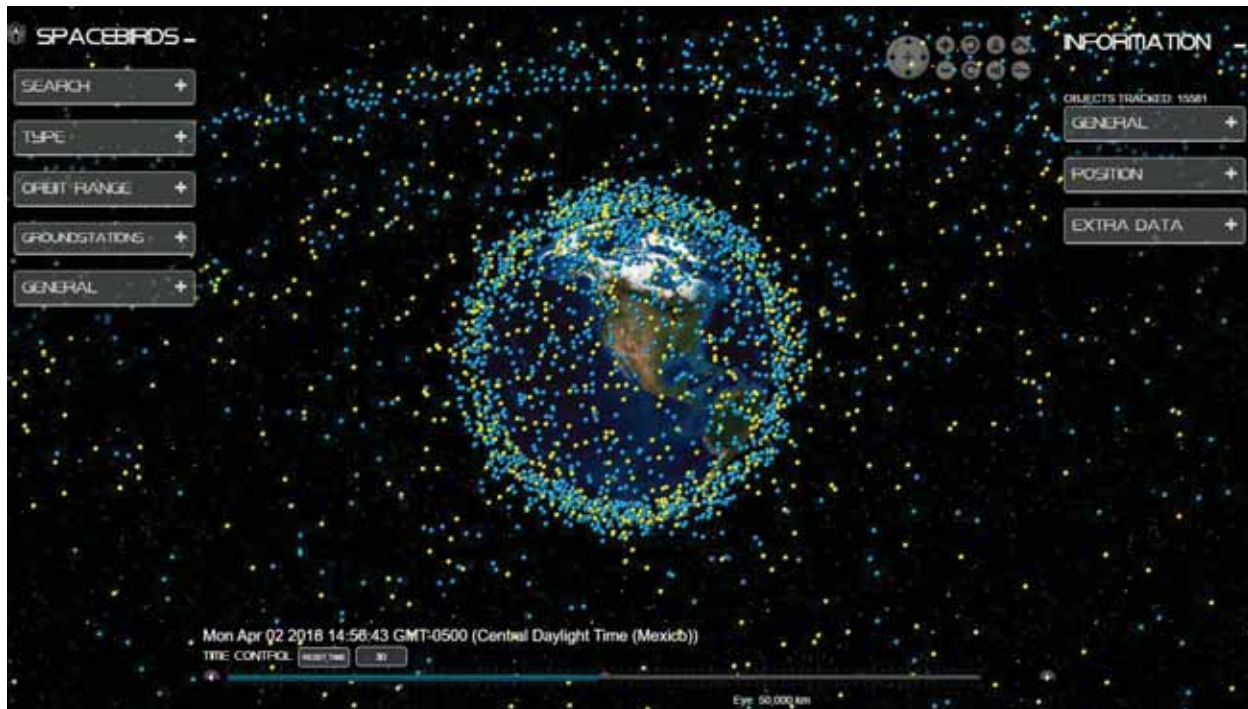


Fig. 5. Módulo de visualización de datos. Imagen: Autoría propia.

Exitosamente se presentan los ~15,000 satélites orbitando la Tierra en 3D, con sus posiciones actualizándose en una tasa calculada de acuerdo con la capacidad de actualización del dispositivo donde se visualiza la aplicación mediante una prueba inicial del tiempo de cálculo de las posiciones de todo el conjunto de datos. Al posar el cursor sobre cada objeto, se muestra la órbita del mismo en un periodo seleccionado.

Al seleccionar un satélite, es posible realizar *zoom* al objeto para visualizar un modelo 3D. En este caso, el único modelo 3D cargado es de la Estación Espacial Internacional (y el modelo fue asignado a todos los objetos), pero es posible asociar otros modelos 3D a los objetos que se desee. La selección del satélite también realiza el dibujo de su “horizonte”: El área sobre la Tierra desde donde es posible verlo en el cielo.

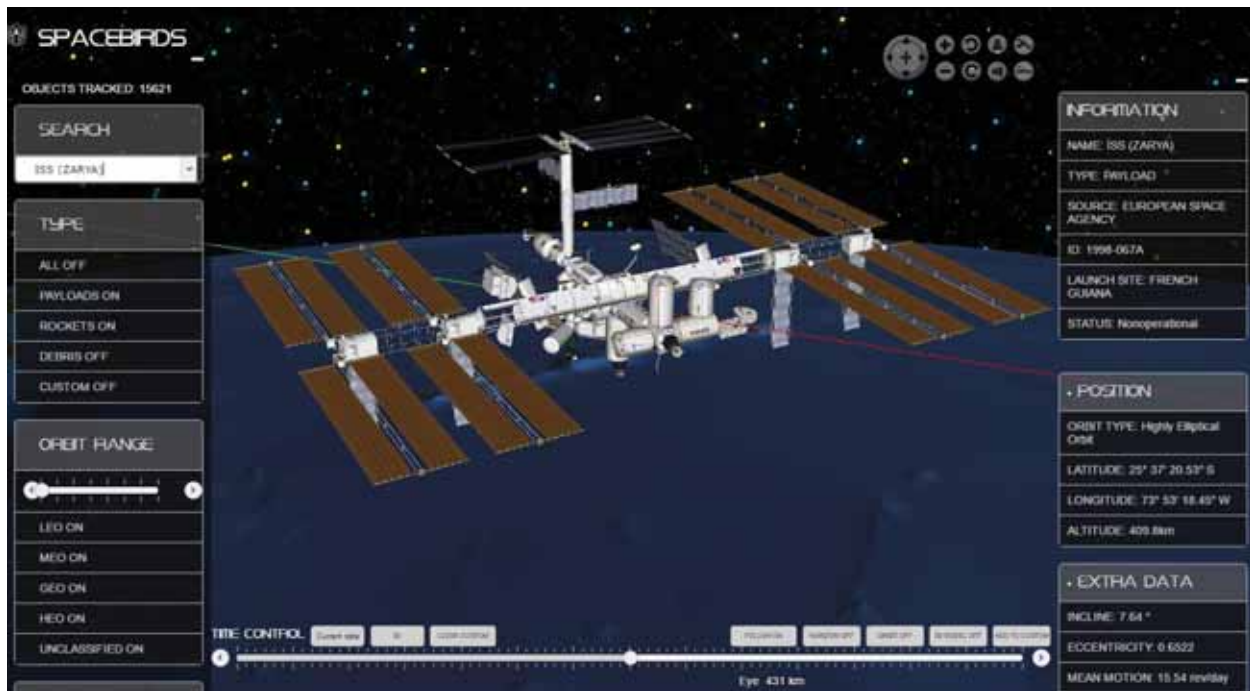


Fig. 6. Modelo 3D de la EEI en SpaceBirds. Imagen: Autoría propia.

WorldWind tiene integradas vistas basadas en diferentes proyecciones geográficas. Es posible cambiar la vista a una proyección equirectangular en 2D que permite ver el trayecto de la órbita de un satélite seleccionado como un trazado terrestre.



Fig. 7. Trazado terrestre y área de visión del Telescopio Espacial Hubble. Imagen: Autoría propia.

El módulo también provee al usuario de herramientas de búsqueda y filtrado del conjunto de satélites por tipo de objeto, tipo de órbita, y búsqueda directa por nombre de objeto. Como característica

adicional, se utilizó el conjunto de estaciones terrenas de *JSatTrak*, para opcionalmente mostrarlas como marcas de posición en el globo.

El vínculo entre el módulo de adquisición y de visualización de datos lo realizan *web workers* [46]: Programas de *JavaScript* que el navegador ejecuta de manera aislada del *DOM* [47] y asíncronamente, comunicándose con el código que los invoca mediante mensajes. Los *web workers* tienen varios casos de uso, y para propósitos de *SpaceBirds*, se utilizaron para realizar las peticiones *AJAX* [48] a los archivos *JSON* en disco, siendo éstos resultado de los datos obtenidos y procesados por el módulo de adquisición. Los *web workers* se encuentran en la carpeta *'/standalone/Workers'* [49]. Uno consume el archivo de datos satelitales, mientras que el segundo consume los datos de las estaciones terrenas.

El módulo de interfaz de usuario combina la lógica proporcionada por el módulo de visualización de datos, *Web WorldWind*, y la estilización gráfica determinada por el CSS y las bibliotecas UI mencionadas en la sección VII. En la carpeta *'/standalone/styles'*, se encuentran los archivos CSS que definen la tipografía, mientras que el archivo HTML de la aplicación [45] determina la distribución de elementos en el documento web.

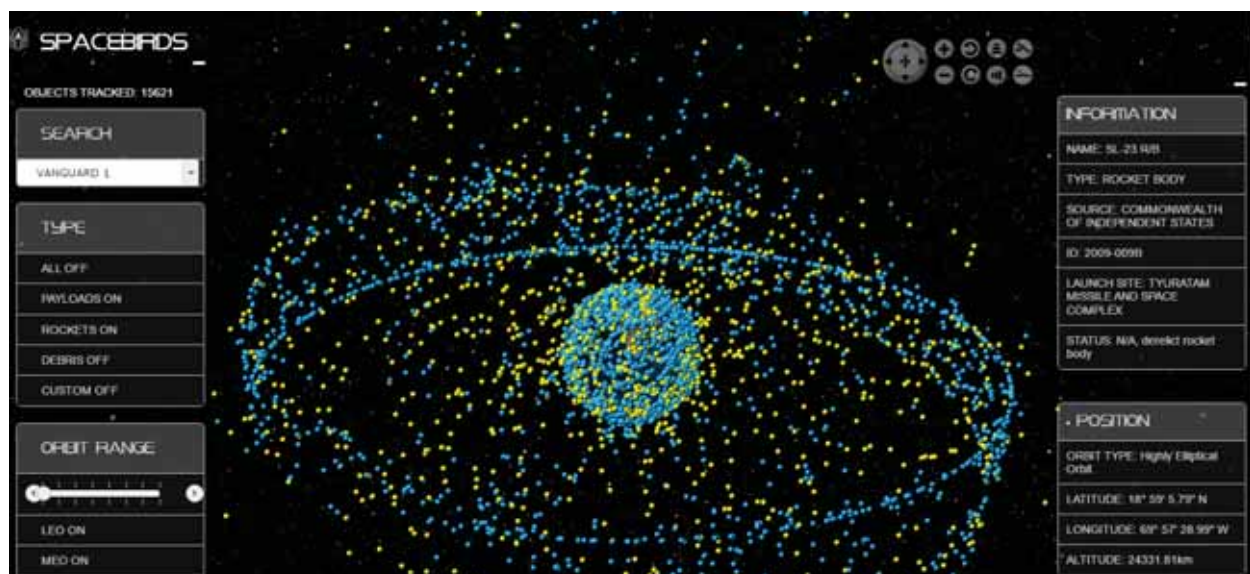


Fig. 8. Anillo de satélites geostacionarios y geosíncronos. Imagen: Autoría propia.

Como se ha mencionado anteriormente, *SpaceBirds* fue desarrollado por un equipo de internos seleccionados por NASA para el proyecto de *WorldWind*. (Tabla 2). Se desglosan las aportaciones de cada integrante del equipo como sigue, con una estimación del volumen de código aportada:

Interno	Aportaciones
Bert Stewart	Desarrollador principal junto con el alumno. Diseñador de primera maqueta de UI, implementador de funcionalidades adicionales al despliegue de los datos en la aplicación, como las herramientas de búsqueda y filtrado de satélites, la navegación en el tiempo, y el despliegue de toda la información asociada a cada satélite. <ul style="list-style-type: none"> • Visualización orbital: 60% • Interfaz de usuario: 25%
Miguel Del Castillo	Diseñador general de la aplicación, desarrollador del <i>backend</i> , y junto con Bert Stewart, desarrollador principal de la aplicación. Encargado del consumo y correcto despliegue de los datos de los satélites, su posición, órbita, actualización en tiempo real y su campo de visión de acuerdo con su altura. Programador de los <i>web workers</i> que consumen los archivos de datos satelitales para entregárselos al módulo de visualización. <ul style="list-style-type: none"> • Visualización orbital: 40% • Adquisición de datos: 100%
Farah Salah	Diseñadora de la interfaz de usuario final tanto en lo estético como en la disposición de los elementos HTML, siendo la única especialista en CSS. <ul style="list-style-type: none"> • Interfaz de usuario: 75%
Julia Semenenko	Desarrolladora que por cuestiones de visado llegó posteriormente al equipo, realizando valiosas aportaciones en <i>debugging</i> y optimización adaptativa del cálculo de órbitas y posiciones en el módulo de visualización orbital.
Jenipher Gonzalez	Diseñadora del logotipo de la aplicación, así como de la página de <i>about</i> . También diseñó la primera propuesta de UI que no fue utilizada.

Tabla 2. Desarrolladores de SpaceBirds y sus aportaciones.

Modificaciones realizadas a Web WorldWind

Se requirió modificar la base de código de la biblioteca misma de *Web WorldWind* para visualizar el volumen tridimensional que comprende al entorno orbital terrestre en la aplicación. Considerando los casos de uso comunes de *WorldWind*, se encuentra optimizado para mostrar objetos e información cerca de la superficie terrestre, por lo que normalmente el volumen de renderización está truncado de tal modo que únicamente se visualiza el hemisferio terrestre con cara hacia la cámara virtual, ocultando (realizando *culling*) todo lo que se encuentre más allá del plano que corta al globo en dicho hemisferio (Fig. 9).

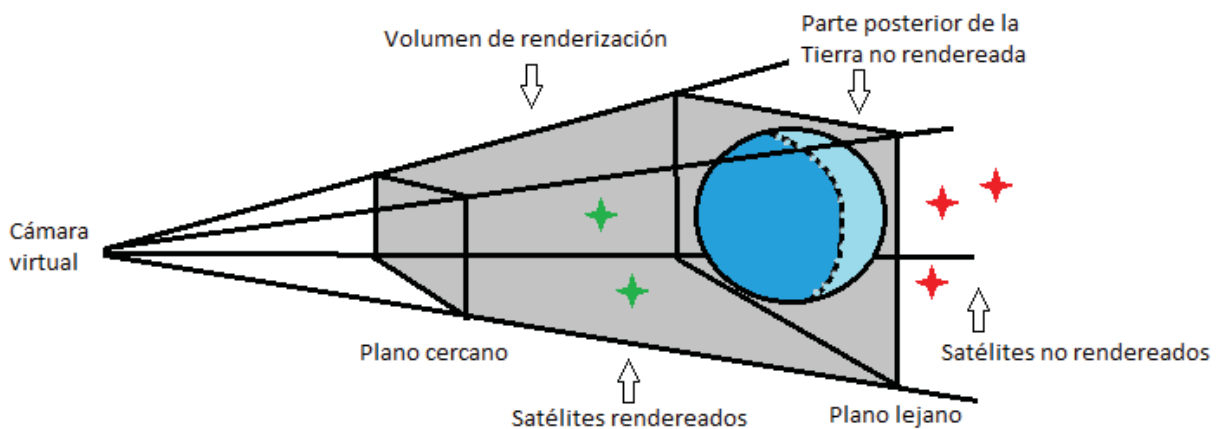


Fig. 9. Configuración estándar del volumen de renderización de Web WorldWind. Imagen: Autoría propia.

Dicho de otro modo, el plano lejano del volumen de renderización corta a la mitad al globo, y siendo que el tamaño del volumen no es una característica configurable por medio de la *API* proveída por *WorldWind*, fue necesario modificar los cálculos de distancia del plano lejano en la biblioteca para alejarlo a mucho más allá del globo (alrededor de unos 3 diámetros terrestres) para que se visualizaran los satélites que se encuentran atrás de la Tierra con respecto a la cámara virtual (Fig. 5). Por tanto, la versión de *Web WorldWind* que utiliza *SpaceBirds* está personalizada.

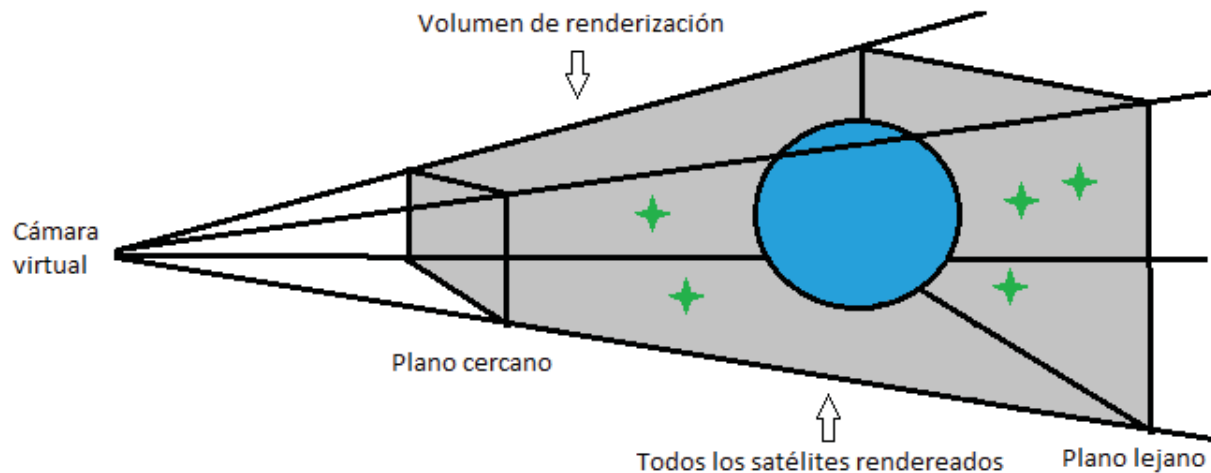


Fig. 10. Ampliación al volumen de renderización de *Web WorldWind*. Imagen: Autoría propia.

En el repositorio de código se encuentran mucho más carpetas y archivos de los mencionados anteriormente. Éstos comprenden todos los módulos de *Web WorldWind* necesarios para construir el archivo de la biblioteca que *SpaceBirds* consume para mostrar el globo, incluida la modificación pertinente al plano lejano en el archivo `'/src/navigate/Navigator.js'` [50].

Áreas de oportunidad para mejoras

Web WorldWind no está optimizado para presentar 15,000 marcas de posición (los elementos utilizados para representar a los satélites) en movimiento y de manera concurrente. En este sentido, las versiones de Java y Android son más capaces, parcialmente por las ventajas inherentes del entorno en donde se ejecutan. ***SpaceBirds* consume muchos recursos de cómputo tanto del GPU como del CPU**, a menos que se reduzca la cantidad de satélites desplegados mediante los filtros proporcionados. La biblioteca realiza un *draw call* [51] al hardware de video por cada marca de posición (y otros objetos, como las trayectorias orbitales), lo cual ralentiza la renderización de manera significativa. Si bien en teoría *SpaceBirds* puede ejecutarse en el navegador de un dispositivo móvil, éste requiere ser de prestaciones de hardware considerables. La tasa de cuadros por segundo y la velocidad de respuesta de la aplicación es muy baja si el equipo en el que se ejecuta no es potente. Esto a pesar de la optimización realizada del cálculo de la posición de todos los objetos de acuerdo con la velocidad del equipo que ejecuta la aplicación. Es posible, sin embargo, realizar más modificaciones a *WorldWind* para dibujar todos los objetos sobre el globo en una cantidad menor de *draw calls* al hardware de video.

La interfaz gráfica, si bien utilizable y funcional, podría ser más amigable. Para colocar el entorno orbital en una fecha dada, se utilizó un control deslizante. Hubiera sido más adecuado un *widget* que permitiera introducir fecha y hora.

La posibilidad de **actualizar el conjunto de datos desde la aplicación web sería una adición importante**. Sin embargo los archivos involucrados son demasiado grandes como para actualizarse directamente desde el *frontend* sin tener acceso de escritura a disco en el servidor.

SpaceBirds podría ser extendido para **incluir características avanzadas de predicción de avistamientos satelitales** en un punto dado sobre la Tierra, aprovechando el modelo digital de elevación de *WorldWind* para incluso tomar en cuenta el relieve terrestre y los ángulos óptimos de visión para la instalación de estaciones terrenas, como lo hace *JSatTrak*. La Agencia Espacial Europea (ESA) desarrolló una aplicación bajo *Web WorldWind* que muestra no solo la órbita y posición de los satélites, sino sus planes de adquisición y sus tiempos de conexión y descarga de datos a estaciones terrenas [52].

Desarrollos adicionales

Además de *SpaceBirds*, el alumno contribuyó con aportaciones menores a otros dos proyectos realizados en la estancia por el equipo de internos de *WorldWind*:

- **QuakeHunter**, una herramienta web de visualización del historial de sismos en 3D (con el epicentro de los sismos dibujado bajo la superficie terrestre en su profundidad correcta) [53].
- **WorldWeather**, una herramienta de visualización de imágenes satelitales (principalmente climatológicas) de diversos servicios web publicados por diversas organizaciones como NASA, ESA, US NAVY, DLR, entre otras [54].

IX. Recursos

Los recursos económicos necesarios de alojamiento, viáticos, alimentos, etc. para la estancia fueron estipulados en la convocatoria de la AEM y aprobados y proporcionados por la UAM.

El único equipo necesario comprende una laptop, proporcionada por el alumno. La conexión a Internet es proveída por la infraestructura en el centro de la NASA.

El servidor donde actualmente se puede acceder a *SpaceBirds* es propiedad de NASA. Como con cualquier aplicación web, es posible copiar el repositorio a un servidor web para publicar en Internet la aplicación de manera independiente.

X. Entregables

- 1) La aplicación en su totalidad fue diseñada por el alumno y fue programada junto con varios internos. El proyecto queda disponible como software de código abierto en un repositorio de código en línea (*GitHub*) para su difusión y modificación, incluyendo el histórico completo del desarrollo mediante Git. En el repositorio de *GitHub* se pueden visualizar los *commits* realizados por el alumno [5].
- 2) La documentación del proyecto se integra al código mismo mediante comentarios, utilizando la herramienta *JSDocs* [55] de generación automática de documentación, imitando el método de documentación de *Web WorldWind* mismo.
- 3) Se realizó un video que describe la funcionalidad y operación de *SpaceBirds*. Dicho video fue hecho por el alumno junto con Bert Stewart, quien provee la voz [56].
- 4) El Departamento de Educación del Centro de Investigación Ames hace una entrega de un diploma al término exitoso de la estancia.
- 5) El archivo digital de un afiche de presentación de los proyectos ante el personal del Centro de Investigación Ames.

El presente reporte se entrega en un disco compacto al Coordinador de Estudios de Ingeniería en Computación, junto con el código fuente de la aplicación en un archivo comprimido, manteniéndose también disponible en línea en su repositorio de *GitHub* por tiempo indefinido [57]. Como entregables adicionales, en el mismo disco compacto se encuentra el scan del diploma mencionado en el cuarto inciso, además del afiche referido en el inciso quinto. Del mismo modo se adjunta el video de presentación del proyecto que también se encuentra disponible en *YouTube* [56]. La versión del video presente en el CD se encuentra subtitulada en español.

XI. Agradecimientos

El alumno extiende sus agradecimientos a todas las personas que hicieron posible la estancia, sin ningún orden en particular:

Universidad Autónoma Metropolitana

Julieta Roldán, Christopher Bravo, Cristina Larios Malo, Romualdo López Zárate, Miriam Aguilar Sánchez, Jorge Gabriel Ortiz Leroux, Dulce María Castro Val, Armando Alonso Navarrete, Ernesto Noriega Estrada, Luis Enrique Noreña Franco, Silvia González Brambila, Rafael Pérez Flores, Georgii Khachaturov, Gabriela del Valle, Margarita González Brambila, Roberto Alfonso Alcántara Ramírez, Jesús Isidro González Trejo, Salvador Vega y León, sin los cuales no hubiese sido posible obtener el apoyo para realizar la estancia, y por supuesto, a mi asesor Francisco Cervantes de la Torre.

Agencia Espacial Mexicana

Blanca Rebollar, Isaí Fajardo, Octavio Arellano, César Enoc Brindis y Carlos Duarte que hicieron posible el convenio de estancias entre NASA y AEM, además de haber tenido el placer de trabajar con ellos durante mi servicio social en la AEM.

National Aeronautics and Space Administration

Mi coasesor y Project manager Patrick Hogan, y el equipo de *WorldWind* durante la estancia: David Collins, Zach Glueckert, Bruce Schubert y Randolph Kim. Personal adicional de NASA ARC: Ali Guarneros, Friedemann Freund, Ken Galal, George Sarver, Roger Hunter, Andrés Martínez, Benjamin Klamm y Sasha Victoria, así como a Shawn Gano de NASA JSC por su rastreador de satélites JSatTrak que sirvió como inspiración inicial del proyecto. También a mis compañeros de estancia: Farah Salah, Bert Stewart, Julia Semenenko, Jenipher Gonzalez, Gabriel Militão, Benjamin Chang, Khaled Sharif, Nidhi Jain, Erika Biswas, Nancy Hernandez y Édgar González Zaleta, que hicieron de la estancia una experiencia inolvidable.

European Space Agency

Yann Voumard, Project lead del equipo de desarrollo de *Web WorldWind* por parte de ESA, y quien en la práctica también fue coasesor al proporcionarnos un ejemplo de visualización de un satélite y su órbita en *Web WorldWind*.

A James Yoder, por su fabulosa visualización de <http://stuffin.space/>, que sirvió como modelo a seguir para *SpaceBirds* como aplicación web.

A quienes fueron el principal apoyo familiar en la aventura de llegar a la NASA: Mi tío Antonio Del Castillo y a mi hermana Katrina Dobrolinsky, a quien conocí en persona por vez primera en NASA ARC.

Y a Ender también, porque obviamente lo más lógico es dedicarle tu titulación a Ender. ~(=^_^)

XII. Bibliografía y referencias

- [1] *Sitio de la Agencia espacial Mexicana*. Revisado el 3 de marzo del 2017. <http://www.aem.gob.mx/>
- [2] *Oportunidad de Estancias en NASA-AMES (The National Aeronautics and Space Administration) Anexo II - guía de proyectos*, Agencia Espacial Mexicana. Revisado el 3 de marzo del 2017. http://www.educacionespacial.aem.gob.mx/images/estancias/20151124%20ANEXO%20II%20CONVOCATORIA%20ESTANCIAS%20NASA_dise.pdf
- [3] *Sitio del Centro de Investigación Ames de la NASA*. Revisado el 3 de marzo del 2017. <https://www.nasa.gov/centers/ames/home/>
- [4] *AU-18 Space Handbook, Aerospace Control, space surveillance*. US Air Force. Revisado el 2 de julio del 2017. <http://www.au.af.mil/au/awc/awcgate/au-18/au18004c.htm>
- [5] *Repositorio de código de SpaceBirds, Commits del alumno*. Revisado el 2 de julio del 2017. <https://github.com/NASAWorldWindResearch/SpaceBirds/commits?author=Beak-man>
- [6] *NASA Open Source Agreement v1.3*. Sitio de Open Source Initiative. Revisado el 28 de Agosto del 2017. <https://opensource.org/licenses/NASA-1.3>
- [7] *Web WorldWind Virtual Globe code repository*. GitHub / NASA. Revisado el 3 de marzo del 2017. <https://github.com/NASAWorldWind/WebWorldWind>
- [8] *Sitio de WorldWind*. NASA. Revisado el 3 de marzo del 2017. <https://worldwind.arc.nasa.gov/>
- [9] D. G. Bell et al., "NASA WorldWind: Opensource GIS for Mission Operations," 2007 IEEE Aerospace Conference, Big Sky, MT, 2007, pp. 1-9. doi: 10.1109/AERO.2007.352954. Revisado el 12 de julio del 2017. <http://ieeexplore.ieee.org/abstract/document/4161692/?part=1>

- [10] *PowerGlobe: Software for Creating Presentations Based on NASA WorldWind*. Soffia Sandomirskaja. Revisado el 12 de julio del 2017. <https://www.hse.ru/en/edu/vkr/182648157>
- [11] *Proyectos Terminales concluidos de Ingeniería en Computación*, CBI UAM Azcapotzalco. Revisado el 23 de abril del 2017. <http://kali.azc.uam.mx/buscadorcbi/>
- [12] *NASA WorldWind projects presentation*. Portal de *Scientific Exploitation of Operational Missions*. Agencia Espacial Europea. Revisado el 28 de agosto del 2017. http://seom.esa.int/openscience15/docs/default-source/15c12_presentations/D3_C1_04_1000_Hogan.pdf
- [13] *Commits de World Wind Java relacionados con Petróleos Mexicanos*. GitHub. Revisado el 28 de agosto del 2017. <https://github.com/nasa/World-Wind-Java/blob/master/WorldWind/src/gov/nasa/worldwindx/applications/worldwindow/features/Navigation.java#L22>
- [14] *NASA Delivers New Air Traffic Spacing Tool to FAA*. NASA. Revisado el 6 de abril del 2018. <https://www.nasa.gov/aero/nasa-delivers-traffic-spacing-tools-tss>
- [15] *Perfil de LinkedIn de Shawn Gano, autor de JSatTrak*. LinkedIn. Revisado el 6 de abril del 2018. <https://www.linkedin.com/in/shawngano/>
- [16] *JSatTrak, página del Proyecto*. Portal personal de Shawn Gano. Revisado el 6 de abril del 2018. <http://www.gano.name/shawn/JSatTrak/>
- [17] *Portal de Educación Espacial*. Agencia Espacial Mexicana. Revisado el 6 de abril del 2018. <http://www.educacionespacial.aem.gob.mx>
- [18] *Keplerian Elements*, AMSAT. Revisado el 6 de abril del 2018. <http://www.amsat.org/amsat-new/tools/keps.php>
- [19] Lane, Max H, and K H. Cranford. *An Improved Analytical Drag Theory for the Artificial Satellite Problem*. New York: American Institute of Aeronautics and Astronautics, 1969. Revisado el 6 de abril del 2018. <http://www.worldcat.org/title/improved-analytical-drag-theory-for-the-artificial-satellite-problem/oclc/122930989>
- [20] *NORAD Two-Line Element Set Format*, Celestrak.com. Revisado el 3 de marzo del 2017. <https://www.celestrak.com/NORAD/documentation/tle-fmt.asp>
- [21] *Models for Propagation of NORAD Element Sets*, Celestrak.com. Revisado el 3 de marzo del 2017. <https://celestrak.com/NORAD/documentation/spacetrk.pdf>
- [22] *Satellite-js, Modular set of functions for SGP4 and SDP4 propagation of TLEs*, Shashwatak/GitHub. Revisado el 3 de marzo del 2017. <https://github.com/shashwatak/satellite-js/>
- [23] *Geographic information system: Old principles with new capabilities*, Vida Maliene, Vytautas Grigonis. Springer, 2011. Revisado el 3 de marzo del 2017. <https://link.springer.com/article/10.1057%2Fudi.2010.25>
- [24] *US Air Force Space Command*. Fuerza Aérea de los Estados Unidos. Revisado el 3 de marzo del 2017. <http://www.afspc.af.mil/>
- [25] *WebGL Overview*. Khronos Group. Revisado el 3 de marzo del 2017. <https://www.khronos.org/webgl/>
- [26] *Bootstrap website*, Twitter. Revisado el 3 de marzo del 2017. <http://getbootstrap.com/>
- [27] *Bulma – A modern CSS framework*. Revisado el 13 de julio del 2017. <http://bulma.io/>
- [28] *Repositorio de código de perfect-scrollbar*. GitHub. Revisado el 13 de julio del 2017. <https://github.com/noraesae/perfect-scrollbar>
- [29] *jQuery*. Revisado el 13 de julio del 2017. <https://jquery.com/>

- [30] *Document Object Model (DOM)*. Mozilla Developer Network. Revisado el 13 de julio del 2017.
https://developer.mozilla.org/en-US/docs/Web/API/Document_Object_Model
- [31] *Node.js website*, Node Foundation. Revisado el 3 de marzo del 2017.
<https://nodejs.org/en/>
- [32] *WebStorm IDE*. JetBrains. Revisado el 13 de julio del 2017.
<https://www.jetbrains.com/webstorm/>
- [33] *Grunt: The JavaScript task runner*. Revisado el 13 de julio del 2017.
<https://gruntjs.com/>
- [34] *Node http server*. Repositorio de paquetes npm. Revisado el 13 de julio del 2017.
<https://www.npmjs.com/package/node-http-server>
- [35] *What Are RESTful Servicios web?* Oracle Docs. Revisado el 3 de marzo del 2017.
<http://docs.oracle.com/javaee/6/tutorial/doc/gijqy.html>
- [36] *Space-track.org*, SAIC. Revisado el 3 de marzo del 2017.
<https://www.space-track.org>
- [37] *JSON*. Mozilla Developer Network. Revisado el 13 de julio del 2017.
https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/JSON
- [38] *HTTP Access Control (CORS)*. Mozilla Developer Network. Revisado el 13 de julio del 2017.
https://developer.mozilla.org/en-US/docs/Web/HTTP/Access_control_CORS
- [39] *Portal de Celestrak*, Celestrak.com. Revisado el 6 de abril del 2018.
<http://celestrak.com/>
- [40] *CRON job website*, B1G Software. Revisado el 28 de agosto del 2017.
<https://cron-job.org/en/>
- [41] *Comandos de obtención de datos satelitales brutos*. Repositorio de SpaceBirds, GitHub. Revisado el 6 de abril del 2018.
https://github.com/NASAWorldWindResearch/SpaceBirds/tree/master/Backend_nodejs_stuff/curl_script
- [42] *Filtro de datos satelitales*. Repositorio de SpaceBirds, GitHub. Revisado el 6 de abril del 2018.
https://github.com/NASAWorldWindResearch/SpaceBirds/blob/master/Backend_nodejs_stuff/orbitalObjectsFilter.js
- [43] *SpaceBirds*. NASA WorldWind Lab, NASA. Revisado el 6 de abril del 2018.
<https://worldwind.arc.nasa.gov/spacebirds/>
- [44] *Controlador del módulo de visualización de datos*. Repositorio de SpaceBirds, GitHub. Revisado el 6 de abril del 2018.
<https://github.com/NASAWorldWindResearch/SpaceBirds/blob/master/standalone/SatTracker2.js>
- [45] *Vista del módulo de visualización de datos*. Repositorio de SpaceBirds, GitHub. Revisado el 6 de abril del 2018.
<https://github.com/NASAWorldWindResearch/SpaceBirds/blob/master/standalone/index.html>
- [46] *Using Web Workers*. Mozilla Developer Network. Mozilla. Revisado el 6 de abril del 2018.
https://developer.mozilla.org/en-US/docs/Web/API/Web_Workers_API/Using_web_workers
- [47] *Introduction to the DOM*. Mozilla Developer Network. Mozilla. Revisado el 6 de abril del 2018.
https://developer.mozilla.org/en-US/docs/Web/API/Document_Object_Model/Introduction
- [48] *AJAX web docs*. Mozilla Developer Network. Mozilla. Revisado el 6 de abril del 2018.
<https://developer.mozilla.org/en-US/docs/Web/Guide/AJAX>
- [49] *Web workers para obtención de datos satelitales y de estaciones terrenas*. Repositorio de SpaceBirds, GitHub. Revisado el 6 de abril del 2018.
<https://github.com/NASAWorldWindResearch/SpaceBirds/tree/master/standalone/Workers>

- [50] *Módulo de Navegador de Web WorldWind, modificado para SpaceBirds*. Repositorio de SpaceBirds, GitHub. Revisado el 6 de abril del 2018.
<https://github.com/NASAWorldWindResearch/SpaceBirds/blob/master/src/navigate/Navigator.js>
- [51] *The WebGL API: 2D and 3D graphics for the web*. Mozilla Developer Network web docs. Mozilla. Revisado el 6 de abril del 2018. https://developer.mozilla.org/en-US/docs/Web/API/WebGL_API
- [52] *Copernicus Sentinel app (ESA)*. Página de descarga de Google Play. Google. Revisado el 6 de abril del 2018. <https://play.google.com/store/apps/details?id=esa.sentinel&hl=en>
- [53] *QuakeHunter*. NASA WorldWind Lab, NASA. Revisado el 6 de abril del 2018.
<https://worldwind.arc.nasa.gov/quakehunter/>
- [54] *WorldWeather*. NASA WorldWind Lab, NASA. Revisado el 6 de abril del 2018.
<https://worldwind.arc.nasa.gov/worldweather/>
- [55] *JSDoc, automated documentation generation for JavaScript*, JSDoc 3 Documentation project. Revisado el 28 de agosto del 2017. <http://usejsdoc.org/>
- [56] *NASA SpaceBirds: Satellite Tracker*. Video realizado por Bert Stewart y Miguel Del Castillo. YouTube. Revisado el 6 de abril del 2018. <https://youtu.be/ojp8Tqf2j0k>
- [57] *Repositorio de SpaceBirds*, GitHub. Revisado el 6 de abril del 2018.
<https://github.com/NASAWorldWindResearch/SpaceBirds>

XIII. Referencias de tablas

- [1] *Versiones de WorldWind*. Autoría original.
- [2] *Desarrolladores de SpaceBirds y sus aportaciones*.

XIV. Referencias de imágenes

- [1] *Parámetros orbitales*. NASA Glossary of Terms, Keplerian elements. NASA. Revisado el 13 de julio del 2017. <https://earth-www.larc.nasa.gov/ceresweb/IWG/glossary/k.html>
- [2] *Two Line Element Set*. NASA Spaceflight. NASA. Revisado el 6 de abril del 2018.
https://spaceflight.nasa.gov/realdata/sightings/SSapplications/Post/JavaSSOP/SSOP_Help/tle_def.html
- [3] *Diagrama de casos de uso de backend*. Imagen de autoría original.
- [4] *Diagrama de casos de uso de frontend*. Imagen de autoría original.
- [5] *Módulo de visualización de datos*. Imagen de autoría original.
- [6] *Modelo 3D de la EEI en Spacebirds*. Imagen de autoría original.
- [7] *Trazado terrestre y área de visión del Telescopio Espacial Hubble*. Imagen de autoría original.
- [8] *Anillo de satélites geoestacionarios y geosíncronos*. Imagen de autoría original.
- [9] *Configuración estándar del volumen de renderización de Web WorldWind*. Imagen de autoría original.
- [10] *Ampliación al volumen de renderización de Web WorldWind*. Imagen de autoría original.

Todas las imágenes publicadas por la NASA (exceptuando logotipos y sellos oficiales de la agencia) pueden utilizarse para propósitos educacionales, de acuerdo con lo que dice en *Media Usage Guidelines* en el sitio oficial de NASA: <http://www.nasa.gov/multimedia/guidelines/index.html> (revisado el 3 de marzo del 2017).