

**Grado en Ingeniería Informática**  
**Informática Gráfica y Visualización**  
**Curso 2024/2025**



Universidad de Jaén

**Proyecto de Prácticas**  
**SISTEMA SOLAR**

Rafael Osuna Ventura  
rov00005@red.ujaen.es

# ÍNDICE

|   |           |
|---|-----------|
| <b>1. Introducción al Proyecto.....</b>     | <b>2</b>  |
| <b>2. Manual de Usuario.....</b>            | <b>3</b>  |
| Funciones principales.....                  | 3         |
| Navegación entre vistas.....                | 3         |
| Zoom y movimiento de cámara.....            | 3         |
| Selección de planetas.....                  | 3         |
| Interacción con satélites.....              | 4         |
| Controles adicionales.....                  | 4         |
| Animación.....                              | 4         |
| <b>3. Explicación de la Estructura.....</b> | <b>5</b>  |
| 3.1. Texturas.....                          | 6         |
| 3.1.2. Fondo.....                           | 6         |
| 3.2. Planetas.....                          | 6         |
| 3.3. Iluminación.....                       | 6         |
| 3.4. Cámara.....                            | 7         |
| 3.5. Grafo de Escena.....                   | 7         |
| <b>4. Diseño de clases.....</b>             | <b>8</b>  |
| 4.1. Clase Camara.....                      | 8         |
| 4.2. Clase igvTextura.....                  | 8         |
| 4.3. Clase Planeta.....                     | 9         |
| 4.3. Clase NodoGrafoEscena.....             | 9         |
| 4.4. Clase GrafoSatelite.....               | 10        |
| 4.5. Clase Sistema Solar.....               | 10        |
| <b>5. Vídeo Resultado.....</b>              | <b>11</b> |

# 1. Introducción al Proyecto

El proyecto realizado es un Explorador del sistema solar, el cual nos permite una visualización detallada y dinámica de la configuración orbital de los nueve planetas, incluido Plutón, en relación con la Estrella principal, el Sol.

Para poder ver el sistema solar con mayor detalle hemos implementado diferentes acciones como zoom, movimiento de la cámara y navegación entre los planetas. Todo esto permite al usuario el estudio de nuestro sistema solar con “precisión” ya que la representación de él está tomada con valores reales.

## 2. Manual de Usuario

Este manual detalla cómo interactuar con la aplicación, sus funcionalidades principales, y cómo personalizar la experiencia para diferentes tipos de usuarios, ya sea educativo, lúdico o técnico.

### Funciones principales

#### Navegación entre vistas

- **Cambiar vistas:** Puedes alternar entre diferentes vistas usando la tecla **v**. Las vistas disponibles incluyen:
  - Vista Normal: Visualización predeterminada con perspectiva central.
  - Vista de Planta: Perspectiva desde arriba que permite analizar las posiciones orbitales.
  - Vista de Perfil: Perspectiva lateral ideal para ver alineaciones.
- **Vista Dividida:** Presiona **f** para alternar entre una vista única y una división en varias vistas. En esta configuración, puedes observar simultáneamente las vistas de planta, perfil y normal.

#### Zoom y movimiento de cámara

- Usa **+** para acercar la cámara (zoom in) y **-** para alejarla (zoom out). Esto es útil para estudiar detalles específicos de un planeta o su órbita.
- Puedes mover la cámara arrastrando el ratón con el botón izquierdo presionado. Este control permite una navegación intuitiva.

#### Selección de planetas

- Cambia entre planetas seleccionados usando las teclas numéricas (**1-9**). Cada número está asociado a un planeta específico en el sistema.
- Haz clic derecho para abrir un menú contextual y seleccionar planetas directamente. Este menú también incluye opciones avanzadas como ajustes de órbita.

### **Interacción con satélites**

- Los satélites incluyen paneles solares y segmentos rotativos. Usa las siguientes teclas para controlarlos:
  - **i / I**: Rotar el panel izquierdo hacia arriba o abajo.
  - **d / D**: Rotar el panel derecho hacia arriba o abajo.
  - **u / j**: Rotar segmentos del cuerpo del satélite hacia adelante o atrás.
- Presiona **r** o **R** para aumentar/disminuir la velocidad de rotación del satélite.

### **Controles adicionales**

- Presiona **c** para mostrar u ocultar los controles en pantalla. Esto incluye un resumen de todas las teclas activas.
- Usa **Esc** para salir de la aplicación en cualquier momento.

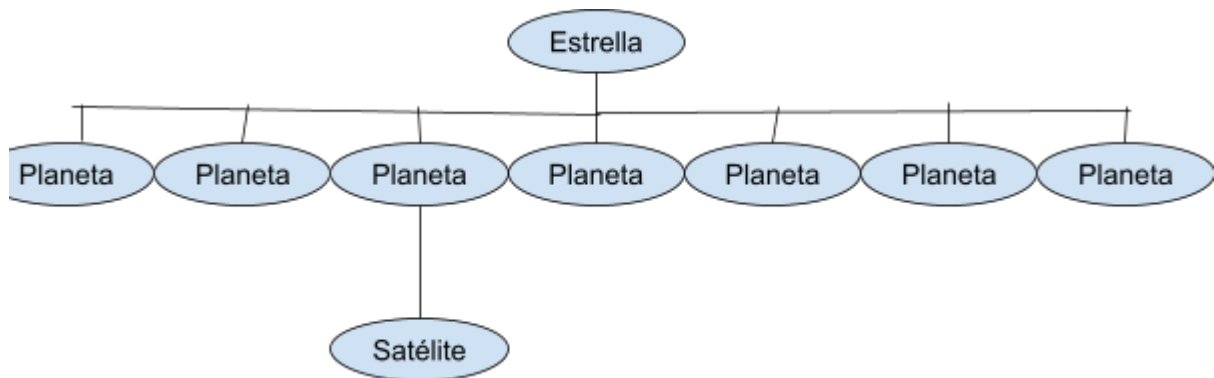
### **Animación**

- Activa o desactiva la animación del satélite con la tecla **t**. Esto controla si el satélite sigue girando de forma automática.
- Ajusta la velocidad de órbita con **a** (acelerar) y **A** (desacelerar). Esto afecta a todos los planetas simultáneamente.

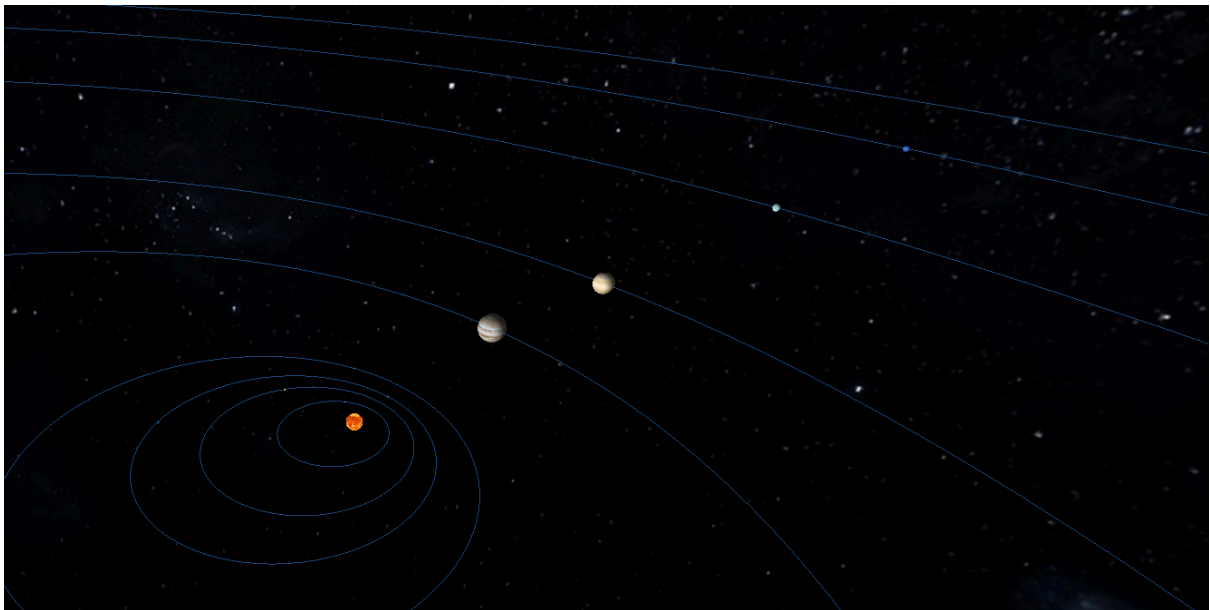
### 3. Explicación de la Estructura

Se requería una estructura jerárquica que se representará de manera similar a un árbol. En esta estructura, el nodo raíz se corresponde con la Estrella, que es el centro del sistema solar. A continuación, se encuentran los Planetas como nodos secundarios y, finalmente, las lunas de cada planeta, que son consideradas nodos hoja dentro de esta jerarquía.

La siguiente organización jerárquica empleada nos permite visualizar la relación y la estructura de manera lógica y ordenada entre los elementos del sistema solar:



El resultado sería el siguiente:



## 3.1.Texturas

Para poder visualizar las texturas se ha empleado la biblioteca “lodepng”. Esta nos permite la carga de imágenes desde archivos en formatos PNG y se utiliza comúnmente en aplicaciones de gráficos, incluidos proyectos que involucran OpenGL. Todas las imágenes usadas como textura son ofrecidas por la página de la NASA.

### 3.1.2. Fondo

El mapa que se ha diseñado se trata de un fondo de estrellas “esférico”. Digo qué es un fondo esférico ya que se quería conseguir que a la hora de mover la cámara siempre hubiera el fondo de estrellas, para ello el fondo es una escena esférica completa que alberga todo el conjunto de planetas, de esta forma aunque la cámara se mueva en todos los ejes el fondo nunca desaparece.

## 3.2. Planetas

Este diseño encapsula tanto los datos orbitales como los aspectos gráficos de los planetas. Desde la representación de la órbita y la posición hasta la texturización y visualización, esta clase permite la manipulación y renderizado de planetas, permitiendo así una integración en un entorno tridimensional.

Los atributos tenidos en cuenta para la representación de un planeta han sido obtenidos de la [siguiente tabla](#), la cual nos la proporciona una página de la NASA.

## 3.3. Iluminación

A la hora de implementar la iluminación se ha realizado lo siguiente, la iluminación proviene desde la estrella del sistema solar, como sería lo normal en el espacio.

Al simular la luz solar, se puede ver como afecta la luz tanto a la distancia de los planetas, siendo los más lejanos que le llegue menos luz, teniendo en cuenta el ciclo de día y noche a la hora de la rotación de los planetas. Por lo tanto se ha tenido en cuenta, que a la hora de renderizar los shaders en las texturas de los planetas se iluminara la cara que está mirando al Sol, por lo tanto como es de esperar la parte que no le da la luz del sol se mantendrá sin ninguna iluminación simulando el ciclo de día y noche natural de cada planeta.

### 3.4. Cámara

La clase Camara está diseñada para proporcionar una experiencia inmersiva al explorar el sistema solar, permitiendo observar el planeta seleccionado desde cualquier ángulo mediante rotaciones libres con el ratón. Además, ofrece la capacidad de cambiar entre diferentes perspectivas para analizar los planetas y sus órbitas desde ángulos predefinidos, o incluso visualizar varias perspectivas simultáneamente dividiendo la ventana en múltiples vistas.

### 3.5. Grafo de Escena

En el proyecto se utilizan dos grafos de escena principales: uno para representar la jerarquía del sistema solar y otro para modelar el satélite. Ambos grafos están diseñados para estructurar las relaciones jerárquicas entre los objetos y permitir transformaciones independientes y acumulativas, asegurando una representación coherente y realista en la simulación. Las animaciones de los grafos son implementadas con la función `glutIdleFunc()`.

- **Grafo del sistema solar:** representa la relación jerárquica entre el Sol, los planetas y sus satélites. El Sol es el nodo raíz del grafo, y cada planeta es un nodo hijo del Sol. Si un planeta tiene satélites, estos se representan como nodos hijos del planeta correspondiente. Por ejemplo, la Luna es un nodo hijo de la Tierra.

Este diseño jerárquico permite que las transformaciones, como las rotaciones y traslaciones, se acumulen a lo largo del grafo. Por ejemplo, cuando la Tierra orbita alrededor del Sol, la Luna también lo hace de manera coherente, ya que su posición depende de la de la Tierra. Cada nodo en el grafo puede tener transformaciones locales (rotación propia, órbita, inclinación) que se combinan con las transformaciones globales de sus nodos padres.

- **Grafo Satélite :** El satélite se modela como un grafo de escena independiente, pero utiliza el mismo enfoque jerárquico. Está compuesto por varios nodos que representan sus partes móviles, como los paneles solares y el cuerpo principal. Cada parte es un nodo en el grafo, con transformaciones específicas que permiten movimientos independientes, como la rotación de los paneles o la inclinación del cuerpo.

El uso de este grafo jerárquico permite una interacción realista con el satélite. Los usuarios pueden controlar la rotación de cada componente de manera individual, respetando las restricciones físicas y manteniendo la cohesión del modelo general. Por ejemplo, al rotar un panel solar, la posición de este se ajusta correctamente en relación con el cuerpo principal del satélite.

## 4. Diseño de clases

El proyecto está compuesto por varias clases que modelan los elementos principales del sistema solar, como planetas, texturas, luces, cámaras y un grafo de escena para manejar la jerarquía de objetos. Cada clase está diseñada para ser modular y escalable, permitiendo futuras expansiones como la inclusión de nuevos planetas o mejoras gráficas.

### 4.1. Clase Camara

La clase `Camara` tiene como propósito gestionar cómo los usuarios pueden observar el sistema solar desde diferentes ángulos y con distintos niveles de detalle. No se trata de una cámara completamente libre, sino que siempre está anclada al planeta seleccionado, permitiendo rotar alrededor de este con el ratón para explorar sus características desde cualquier ángulo. Además, la cámara incluye funcionalidades que permiten cambiar entre vistas predefinidas, como vistas superiores o laterales, e incluso dividir la ventana para mostrar varias perspectivas al mismo tiempo.

La clase también maneja el nivel de zoom, permitiendo al usuario acercarse para observar detalles del planeta o alejarse para obtener una vista más general. Esto se combina con la capacidad de ajustar automáticamente el campo de visión en función del tamaño de la ventana, asegurando que el sistema solar siempre se visualice correctamente.

**Métodos principales:**

- `cambiarVista()`: Cambia entre modos de vista predefinidos (como planta o perfil) para ofrecer diferentes perspectivas del sistema solar.
- `moverRaton(int x, int y)`: Usa las coordenadas del ratón para rotar la cámara alrededor del planeta seleccionado.
- `setZoom(double nuevoZoom)`: Ajusta la distancia entre la cámara y el planeta, modificando el nivel de zoom.
- `ajustarVista(int ancho, int alto)`: Adapta el campo de visión según las dimensiones actuales de la ventana de visualización.

### 4.2. Clase igvTextura

La clase `igvTextura` es responsable de la gestión de texturas en la simulación. Una textura es una imagen que se aplica a un objeto 3D para darle más realismo. En este proyecto, las texturas se utilizan principalmente para los planetas y el fondo de la escena. La clase permite cargar texturas desde archivos PNG usando la biblioteca `lodepng`, asegurando la compatibilidad con estándares comunes. Estas texturas se aplican a objetos mediante las funciones de OpenGL, como `glTexImage2D`, lo que facilita la integración con la escena 3D.

Adicionalmente, esta clase incluye métodos para configurar un fondo esférico texturizado, que actúa como un entorno visual envolvente, aumentando la inmersión del usuario en la simulación.



#### Métodos principales:

- **asociarTextura()**: Asocia la textura cargada a un objeto de la escena, como un planeta.
- **fondo()**: Crea y renderiza un fondo texturizado en forma de esfera para el sistema solar.

### 4.3. Clase Planeta

La clase **Planeta** modela los planetas del sistema solar, incluyendo sus propiedades físicas, orbitales y visuales. Cada planeta tiene atributos como su nombre, tamaño, período orbital, inclinación y texturas que lo identifican. También se incluyen datos como el aphelion y perihelion (distancias máximas y mínimas al Sol), los cuales son escalados para adaptarse al contexto de la simulación.

Además, los planetas tienen una estructura jerárquica: un planeta puede tener un padre (por ejemplo, la Luna tiene como padre a la Tierra) y puede actuar como padre de otros objetos. Esto permite que sus movimientos orbitales y rotacionales sean realistas y respeten su dependencia jerárquica.

#### Métodos principales:

- **Orbita(double grado, double escalaRotacionEje)**: Calcula la posición del planeta en su órbita, aplicando las transformaciones correspondientes.
- **dibujarPlaneta()**: Renderiza el planeta, aplicando texturas, iluminación y las transformaciones acumulativas necesarias.
- **set\_padre(Planeta\* nuevoPadre)**: Establece relaciones jerárquicas entre planetas o satélites.

### 4.3. Clase NodoGrafoEscena

La clase **NodoGrafoEscena** define una estructura genérica para manejar relaciones jerárquicas entre objetos en una escena 3D. Es utilizada tanto para representar el sistema solar como para modelar componentes específicos, como el satélite. Cada nodo del grafo puede realizar transformaciones independientes (traslación, rotación, escalado), pero estas transformaciones se acumulan de manera jerárquica. Por ejemplo, si un nodo padre rota, sus nodos hijos también serán afectados.

Esto resulta especialmente útil para representar relaciones espaciales complejas, como los movimientos de los planetas alrededor del Sol o la rotación de un panel solar respecto al cuerpo principal de un satélite.

#### Métodos principales:

- **agregarHijo(NodoGrafoEscena\* nodo)**: Agrega un nodo hijo a la estructura jerárquica actual.
- **dibujar()**: Aplica las transformaciones acumulativas y renderiza tanto el nodo actual como sus hijos.

## 4.4. Clase GrafoSatelite

El `GrafoSatelite` es una especialización de `NodoGrafoEscena` diseñada específicamente para modelar un satélite. Este grafo incluye nodos que representan las diferentes partes del satélite, como el cuerpo principal, los paneles solares y los brazos mecánicos. Cada componente puede moverse de manera independiente, respetando restricciones físicas y manteniendo la cohesión general del modelo.

Por ejemplo, los paneles solares pueden rotar alrededor de su eje para simular el seguimiento del Sol, mientras que el cuerpo principal puede inclinarse o moverse sin afectar las relaciones entre los nodos.

### Métodos principales:

- `inicializar()`: Configura las transformaciones iniciales de los componentes del satélite, como colores y posiciones.
- `dibujar()`: Renderiza el satélite completo, aplicando las transformaciones de cada nodo según su jerarquía.

## 4.5. Clase Sistema Solar

La clase `SistemaSolar` es el núcleo principal del proyecto, encargada de coordinar y gestionar todos los elementos que componen el sistema solar, incluyendo los planetas, sus órbitas, y el satélite. Actúa como controlador central, inicializando y organizando las diferentes entidades y configurando su interacción dentro de la escena.

Esta clase maneja la creación de los planetas y sus relaciones jerárquicas (como la Tierra y el satélite), además de configurar parámetros globales como la velocidad de las órbitas y las propiedades visuales. También gestiona eventos de usuario, como la selección de planetas, el cambio de vistas y las transformaciones interactivas de los objetos.

En términos de diseño, `SistemaSolar` se apoya en otras clases, como `Camara`, `Planeta`, y `NodoGrafoEscena`, para delegar tareas específicas, asegurando que la simulación sea modular y escalable.

### Métodos principales

- `crearPlanetas()`: Crea e inicializa todos los planetas del sistema solar, estableciendo sus propiedades (nombre, tamaño, órbita, textura) y jerarquías. Los datos están escalados por un factor de 20 y se obtienen de la [página oficial de la NASA](#).
- `dibujarMovimiento()`: Renderiza toda la escena del sistema solar. Aplica las transformaciones a los planetas y el satélite en función de sus posiciones actuales, además de manejar las vistas y modos interactivos como la vista dividida.
- `orbita()`: Calcula las posiciones orbitales de todos los planetas y satélites según el tiempo transcurrido. Actualiza las rotaciones y traslaciones necesarias para que las órbitas sean precisas y dinámicas.
- `representarSistemaSolar()`: Es el punto de entrada principal para inicializar el sistema solar. Configura la cámara, los planetas, el satélite, las texturas del fondo y

los eventos interactivos (teclado, ratón). También inicializa la lógica de animaciones y órbitas.

- **mostrarControles()**: Renderizar en pantalla un cuadro con información sobre las teclas y controles disponibles. Esto permite a los usuarios consultar cómo interactuar con la simulación sin abandonar la aplicación.
- **mostrarInfoPlaneta()**: Imprime en la consola información detallada del planeta seleccionado, incluyendo nombre, aphelion, perihelion y período orbital. Es útil para comprender las propiedades específicas de cada planeta.
- **dibujarSatelite()**: Renderiza el satélite alrededor de un planeta, aplicando las transformaciones acumuladas para que se mueva de forma realista según la rotación y posición orbital del planeta.
- **actualizarAnimacion()**: Controla las animaciones del satélite. Incrementa los ángulos de rotación en función del tiempo y reinicia las posiciones si las animaciones completan un ciclo.
- **movermeEntrePlanetas(int indice)**: Cambia el planeta seleccionado en función del índice proporcionado. Ajusta automáticamente el zoom de la cámara para ofrecer una vista óptima del nuevo planeta.
- **menuSeleccion(int opcion)**: Implementa la lógica para un menú contextual de selección de planetas. Este menú aparece al hacer clic derecho y permite seleccionar planetas rápidamente.
- **crearMenuPlanetas()**: Construye un menú dinámico con todos los planetas del sistema solar. Se asocia al botón derecho del ratón, permitiendo acceder a una lista interactiva de planetas.
- **reescalarVentana(int w, int h)**: Ajusta el tamaño de la ventana y recalcula las proporciones de la cámara para asegurar que la escena se vea correctamente tras un redimensionado.
- **renderizarEscena()**: Es el método que aplica las transformaciones generales de la cámara y renderiza cada elemento de la escena, como planetas, satélites y el fondo.
- **mostrarInfoPlanetaEnVentana()**: Renderiza en pantalla información detallada del planeta seleccionado, mostrando atributos como su nombre y distancias orbitales de manera visual.

## 5. Vídeo Resultado

[Enlace video](#)

