PROBLEMA DEL MILENIO: CONJETURA DE BIRCH Y SWINNERTON-DYER

Sergio Juan Díaz Carmona

ANALISIS Y ALGORITMOS: EFRAIN ALBERTO PADILLA ZEPEDA

Conjetura de Birch y Swinnerton-Dyer:

La conjetura de Birch y Swinerton-Dyer es una conjetura matemática, enunciada en 1965 por los matemáticos ingleses Bryan Birch y Peter Swinerton-Dyer.

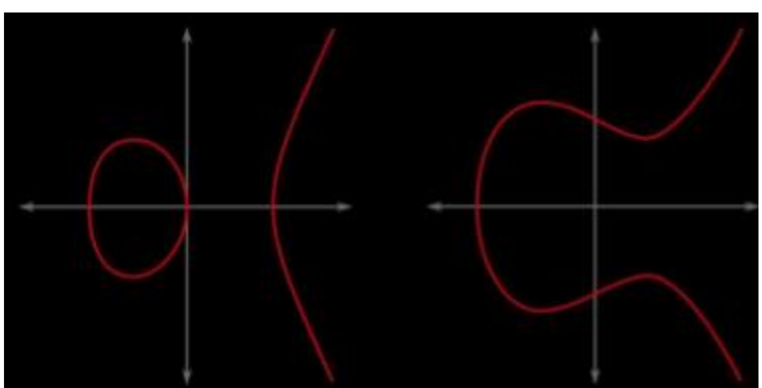
La conjetura relaciona los datos aritméticos asociados a una curva elíptica E sobre un cuerpo numérico K con el comportamiento de la Función L de Hasse-Weil L(E, s) de E en s = 1. Concretamente, se conjetura que el rango del grupo abeliano E(Q) de puntos de E es igual al orden del cero de L(E, s) en s = 1, y el primer coeficiente distinto de 0 en la expansión de Taylor de L(E, s) en s = 1 es dado por un mejor refinamiento de datos aritméticos ligados a E sobre Q. En particular, asegura que si L(E, 1) = 0, entonces el grupo E(Q) es infinito, y recíprocamente, si L(E, 1) ≠ 0, entonces E(Q) es finito.

Dicho de otra forma: La conjetura de Birch y Swinnerton-Dyer, que une geometría algebraica y teoría de números, pide estudiar las soluciones racionales a ecuaciones que definen una curva elíptica.

A principios de la década de 1960, Peter Swinnerton-Dyer usó la computadora EDSAC del laboratorio de informática de la universidad de Cambridge para calcular el número de puntos módulo p (denotado por Np) para un número largo de primos p sobre curvas elípticas cuyo rango era conocido. De esos resultados numéricos Bryan Birch y Swinnerton-Dyer conjeturaron que Np para una curva E con rango r obedecía una ley asintótica.

$$\prod_{p \leq x} \frac{N_p}{p} \approx C \log(x)^r \text{ cuando } x \to \infty$$

Esto, a su vez, llevó a hacer una conjetura general sobre el comportamiento de la función L de una curva L(E ,s) en s = 1, es decir, que podría tener un cero de orden r en ese punto. Esta conjetura se hizo con visión de futuro en ese momento, dado que la continuación analítica de L(E,s) se estableció sólo para curvas bajo multiplicación compleja, las cuales eran también la principal fuente de ejemplos numéricos. (Nótese que el recíproco de la L-función es desde algunos puntos de vista un objeto más natural de estudio; en ocasiones, esto significa que se podrían considerar los polos en lugar de los ceros)



Representación grafica de la conjetura de Birch y Swinnerton-Dyer sobre curvas elípticas.

La conjetura fue ampliada posteriormente para incluir la predicción precisa del coeficiente de Taylor principal de la función L en s = 1. Esto fue conjeturado mediante:

$$\frac{L^{(r)}(E,1)}{r!} = \frac{\#\mathrm{Sha}(E)\Omega_E R_E \prod_{p|N} c_p}{(\#E_{\mathrm{Tor}})^2}$$

donde las cantidades del miembro de la derecha son invariantes de la curva, estudiadas por Cassels, Tate, Shafarevich y otros: éste incluye el orden del grupo de torsión, el orden del grupo de Tate–Shafarevich, y las alturas canónicas de una base de puntos racionales.

Planteamiento de solución:

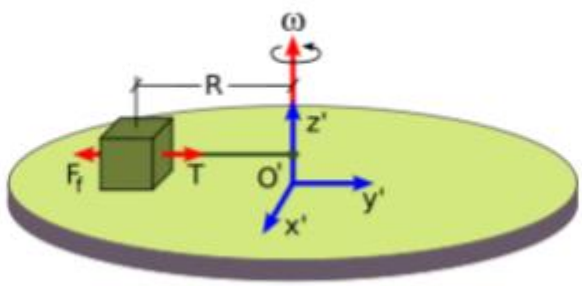
Tras el análisis del problema se encontraron semejanzas con algunas leyes que tiene formulas que nos pueden ayudar a simular este caso de elipses, utilizando de ejemplo las orbitas de los planetas del sistema solar.

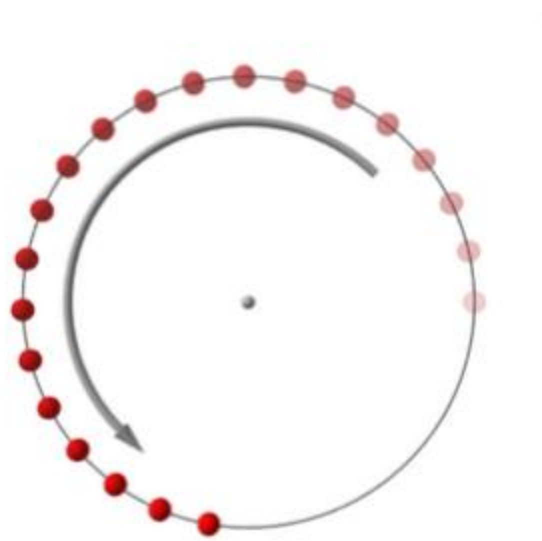
Los temas que tomamos para estos problemas son:

movimiento circular:

Se define movimiento circular como aquél cuya trayectoria es una circunferencia. Una vez situado el origen O de ángulos describimos el movimiento circular mediante las siguientes magnitudes. En cinemática, el movimiento circular es el que se basa en un eje de giro y radio constante, por lo cual la trayectoria es una circunferencia. Si, además, la velocidad de giro

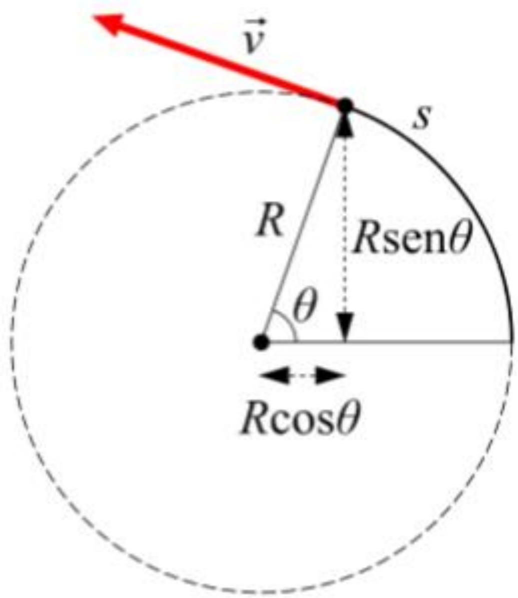
es constante, se produce el movimiento circular uniforme, que es un caso particular de movimiento circular, con radio fijo y velocidad angular constante.



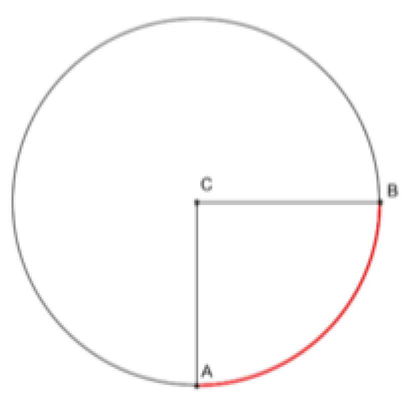


En los movimientos circulares hay que tener en cuenta algunos conceptos específicos para este tipo de movimiento:

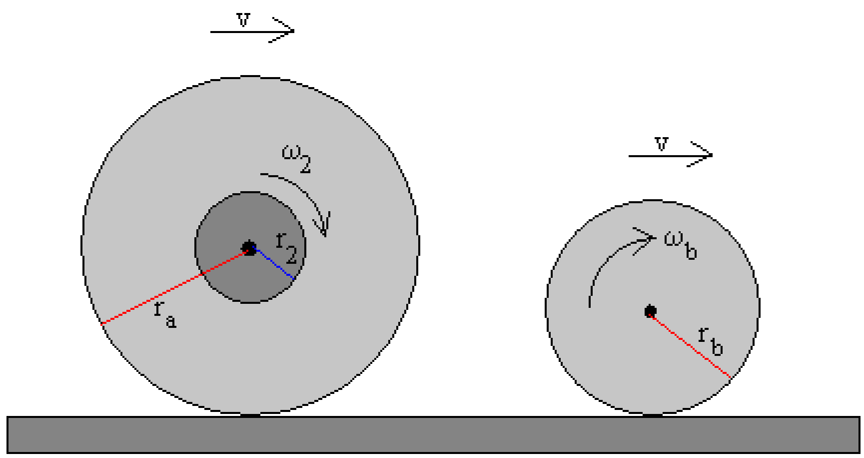
Eje de giro: es la línea alrededor de la cual se realiza la rotación, este eje puede permanecer fijo o variar con el tiempo, pero para cada instante de tiempo, es el eje de la rotación.



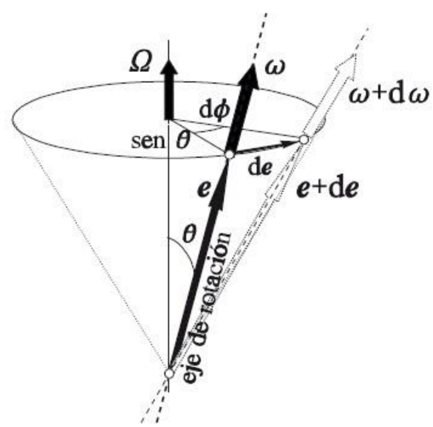
Arco: partiendo de un eje de giro, es el ángulo o arco de radio unitario con el que se mide el desplazamiento angular. Su unidad es el radián.



Velocidad angular: es la variación de desplazamiento angular por unidad de tiempo.



Aceleración angular: es la variación de la velocidad angular por unidad de tiempo.

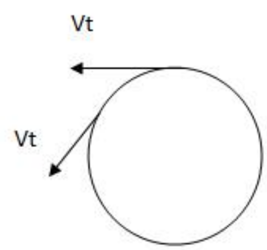


Velocidad tangencia:

La velocidad tangencial es la velocidad del móvil (distancia que recorre en el tiempo).

Por lo tanto para distintos radios y a la misma velocidad angular, el móvil se desplaza a distintas velocidades tangenciales. A mayor radio y a la misma cantidad de vueltas por segundo, el móvil recorre una trayectoria mayor, porque el perímetro de esa circunferencia es mayor y por lo tanto la velocidad tangencial también es mayor.

La velocidad tangencial se mide en unidades de espacio sobre unidades de tiempo, por ejemplo [m/s], [km/h], etc. Se calcula como la distancia recorrida en un período de tiempo.



Ecuación de la velocidad tangencial

Para calcular la velocidad tangencial se multiplica la velocidad angular por el radio (llama tangencial porque es tangente a la trayectoria).

$$V = \omega \cdot r$$

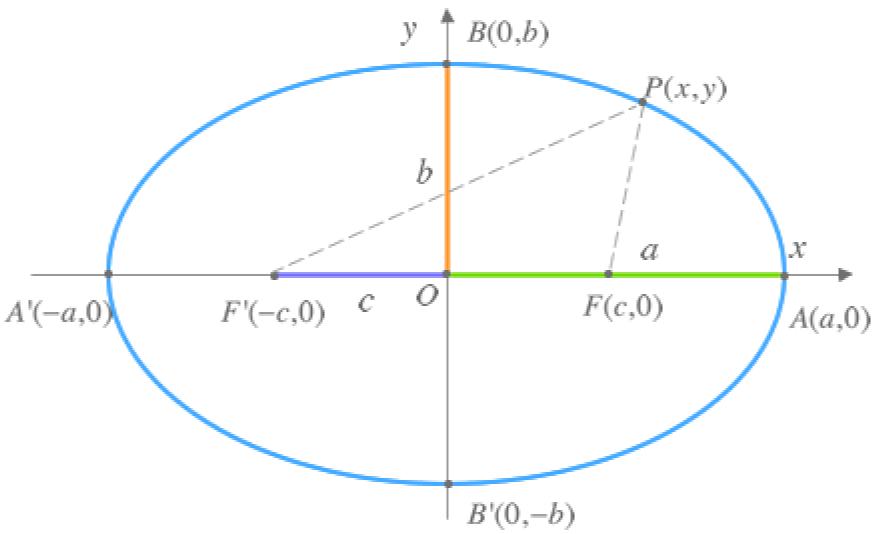
V = Velocidad tangencial [m/s]
ω = Velocidad angular = 2 π f [rad/s]
r = Radio de giro [m]
Para el ejemplo anterior la calculamos como:

$$V = \omega \cdot r = \frac{2\pi \frac{rad}{s}}{1\ s} \cdot 5\ m = 31{,}4\ \frac{m}{s}$$

En MCU la velocidad tangencial es constante (en módulo) para un mismo punto. A mayor distancia del eje, la velocidad tangencial aumenta. Su dirección varía continuamente, teniendo siempre la misma dirección que la recta tangente al punto en donde se encuentre el móvil. La velocidad tangencial es un vector, que resulta del producto vectorial del vector velocidad angular (ω) por el vector posición (*r*) referido al punto *P*.

Ecuación de la elipse

La elipse se define como una línea curva cerrada tal que la suma de las distancias a dos puntos fijos, *F* y *F'* , llamados focos, es constante.



Elipse
Se trata de una circunferencia achatada que se caracteriza porque la suma de las distancias desde cualquiera de sus puntos P hasta otros dos puntos denominados focos (F y F') es siempre la misma.

Ten en cuenta que para cualquier punto de la elipse siempre se cumple que:

$$d(P,F)+d(P,F')=2\cdot a$$

Donde *d(P,F)* y *d(P,F')* es la distancia de un punto genérico *P* al foco *F* y al foco *F'* respectivamente.

Los siguientes elementos se encuentran en cada elipse:

1. Centro: Es el punto de intersección de los ejes. Es, además, centro de simetría.
2. Eje principal o focal: Es el eje en el que se encuentran los focos. Es un eje de simetría.
3. Eje secundario: Es el eje perpendicular al eje principal, mediatriz del segmento que une los focos.
4. Vértices: Puntos de intersección de la elipse con los ejes.

5. Distancia focal: Distancia entre los focos. Su longitud es 2·c.
6. Semidistancia focal: Distancia entre el centro y cada foco. Su longitud es c.
7. Semieje mayor o principal: Segmento entre el centro y los vértices del eje principal. Su longitud es a.
8. Semieje menor o secundario: Segmento entre el centro y los vértices del eje secundario.
9. Radio vectores: Cada punto de la elipse cuenta con dos radio vectores que son los segmentos que unen dicho punto a cada uno de los focos.

Leyes de kepler

Al empezar con las implementaciones nos dimos cuenta que obtener la posición ‘X’ y ‘Y’ necesitaba mas que los temas ya explicados, así que al buscar encontramos las formulas de la velocidad areolar, pero para poder entender esta necesitamos explicar las leyes de Kepler.

El boom que representa el descubrimiento de planetas fuera del sistema solar ha reactivado el interés en el movimiento planetario. Nada hace creer que estos exo-planetas sigan leyes distintas a las ya conocidas y válidas del sistema solar: las leyes de Kepler.

Johannes Kepler fue el astrónomo que sin la ayuda del telescopio y utilizando las observaciones de su mentor Tycho Brahe, creó un modelo matemático que describe el movimiento de los planetas alrededor del Sol.

Dejó plasmado este modelo en las tres leyes que llevan su nombre y que siguen siendo tan vigentes el día de hoy como en 1609, cuando estableció las dos primeras y en 1618, fecha en que enunció la tercera.

En lenguaje actual, las tres leyes de Kepler dicen así:

1. Las órbitas de todos los planetas son elípticas y el Sol está en un foco.

2. El vector de posición que va desde el Sol a un planeta barre áreas iguales en tiempos iguales.

3. El cuadrado del período orbital de un planeta es proporcional al cubo del semieje mayor de la elipse descrita.

Un planeta tendrá una velocidad lineal, al igual que cualquier objeto conocido que se mueva. Y aún hay más: al escribir la segunda ley de Kepler en forma matemática, surge un nuevo concepto llamado velocidad areolar, propia de cada planeta.

¿Por qué los planetas se mueven elípticamente alrededor del Sol?

La Tierra y los demás planetas se mueven alrededor del Sol gracias a que este ejerce una fuerza sobre ellos: la atracción gravitatoria. Lo mismo sucede con cualquier otra estrella y los planetas que conformen su sistema, si los tiene.

Esta es una fuerza del tipo conocido como fuerza central. El peso es una fuerza central con la que todos están familiarizados. El objeto que ejerce la fuerza central, sea el Sol o una estrella lejana, atrae a los planetas hacia su centro y estos se mueven describiendo una curva cerrada.

En principio esta curva se puede aproximar como una circunferencia, tal como lo hizo Nicolás Copérnico, astrónomo polaco creador de la teoría heliocéntrica.

La fuerza responsable es la atracción gravitatoria. Esta fuerza depende directamente de las masas de la estrella y el planeta en cuestión y es inversamente proporcional al cuadrado de la distancia que los separa.

El problema no es tan fácil, pues en un sistema solar, todos los elementos interactúan de esta manera, añadiendo complejidad al asunto. Además, no son partículas, pues estrellas y planetas tienen tamaño mensurable.

Por esta razón, el punto central de la órbita o circuito recorrido por los planetas no está exactamente centrado en la estrella, sino en un punto conocido como el centro de gravedad del sistema sol-planeta.

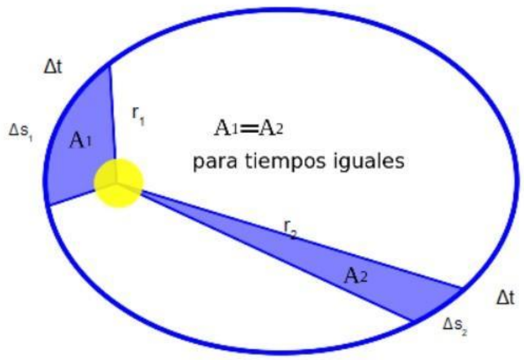
La órbita resultante es elíptica. La siguiente imagen lo muestra, tomando como ejemplo la Tierra y el Sol:

La órbita de la Tierra es elíptica, con el Sol ubicado en uno de los focos. Cuando la Tierra y el Sol están a su máxima distancia se dice que la Tierra está en afelio. Y si la distancia es mínima entonces hablamos de perihelio. El afelio es la posición más lejana de la Tierra al Sol, mientras que el perihelio es el punto más cercano. La elipse puede ser más o menos achatada, según las características de sistema estrella – planeta.

Los valores de afelio y perihelio varían anualmente, ya que los demás planetas causan perturbaciones. Para otros planetas, estas posiciones se denominan apoastro y periastro respectivamente.

La magnitud de la velocidad lineal de un planeta no es constante

Kepler descubrió que cuando un planeta órbita alrededor del Sol, durante su movimiento barre áreas iguales en tiempos iguales. La figura 2 muestra gráficamente el significado de esto:



Sol es r. Cuando el planeta describe su órbita recorre un arco de elipse Δs en un tiempo Δt.

Matemáticamente, el hecho de que A1 sea igual a A2 se expresa así:

$$A_1 = A_2$$

Los arcos recorridos Δs son pequeños, de manera que cada área puede aproximarse a la de un triángulo:

$$\frac{1}{2}\Delta s_1 r_1 = \frac{1}{2}\Delta s_2 r_2$$

Como Δs=vΔt, donde v es la velocidad lineal del planeta en un punto dado, al sustituir tenemos:

$$\frac{1}{2}v_1 \Delta t r_1 = \frac{1}{2}v_2 \Delta t r_2$$

Y puesto que el intervalo de tiempo Δt es el mismo, se obtiene:

$$v_1 r_1 = v_2 r_2$$

$$v_1 = v_2 \frac{r_2}{r_1}$$

Como r2 > r1, entonces v1 > v2, en otras palabras, la velocidad lineal de un planeta no es constante. De hecho, la Tierra va más deprisa cuando está en el perihelio que cuando está en el afelio.

Por lo tanto, la velocidad lineal de la Tierra o de cualquier planeta alrededor del Sol no es una magnitud que sirva para caracterizar el movimiento de dicho planeta.

La velocidad areolar

La segunda ley de Kepler sugiere una nueva magnitud llamada velocidad areolar. Se define como el área barrida por unidad de tiempo y es constante. Para calcularla se utiliza la siguiente figura:

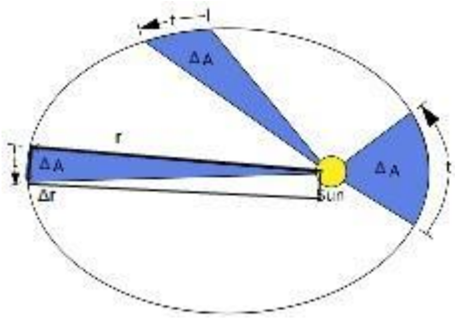


Figura 3. El vector de posición de la Tierra (o planeta) respecto al Sol es r, y al moverse, la Tierra experimenta un desplazamiento, también vectorial Δr.

Se escoge una pequeña área barrida por la Tierra mientras realiza su circuito elíptico, la cual denotaremos como ΔA. El tiempo necesario para ello es Δt.

En la figura 3 se muestra el vector de posición de la Tierra respecto al Sol, denotado por r. Cuando la Tierra se mueve, experimenta un desplazamiento Δr.

Esta área corresponde a la mitad del área del rectángulo mostrado en la figura 3:

$$v_A = \frac{\Delta A}{\Delta t} = \frac{\frac{1}{2}\left|\vec{r} x \Delta\vec{r}\right|}{\Delta t} = \frac{1}{2} r \frac{\Delta r}{\Delta t}$$

El cociente Δr/Δt es precisamente la velocidad lineal de la Tierra, entonces la velocidad areolar queda como:

$$v_A = \frac{1}{2} r v$$

Las unidades de vA en el Sistema Internacional son:

$$\frac{m^2}{s}.$$

Nótese que, si bien tanto r como v varían, el producto permanece constante. Ello convierte a la velocidad areolar en una magnitud muy adecuada para caracterizar el movimiento de un planeta alrededor de su estrella.

El producto de r y v es la magnitud del momento angular L, de manera que la velocidad areolar puede expresarse como:

$$v_A = \frac{1}{2} r v$$

Implementación:

El programa del proyecto está formado basado en la jerarquía del universo.

1. Galaxias.
2. Sistemas.
3. Planetas/Focos.
4. Satelite.

A continuación, mostrare cada una de las clases:

Galaxia:

```
class GALAXIA:public CNODO
{
public:
	GALAXIA();
	~GALAXIA();
	void addSistem(SISTEMASOLAR*);

	void setName(std::string name) { nombre = name; };
	std::string getName() { return nombre; };

	void setRatio(float rat) { ratio = rat; };
	float getRatio() { return ratio; };

	void setPosition(sf::Vector2f pos) { position = pos; };
	sf::Vector2f getPosition(sf::Vector2f pos) { return position; };

	int getNumSistems() { return sistemas.size(); };

	void onInit();
	void onUpdate();
	void onDelete();
	void drawGalaxi(sf::RenderWindow &wnd, const float &spped);
	void drawSistem(sf::RenderWindow);
	void drawPlanet(sf::RenderWindow);

private:
	std::vector<SISTEMASOLAR*> sistemas;
	sf::Vector2f centroDeMasa;
	sf::CircleShape galaxi;
	float ratio;
};
```

Sistema

```
class SISTEMASOLAR :public CNODO
{
public:
	SISTEMASOLAR();
	~SISTEMASOLAR();
	void addPlanet(PLANETA*);
	void addFoco(FOCO*);

	void setName(std::string name) { nombre = name; };
```

```cpp
	std::string getName() { return nombre; };

	void setRatio(float rat) { ratio = rat; };
	float getRatio() { return ratio; };

	void setPosition(sf::Vector2f pos) { position = pos; };
	sf::Vector2f getPosition(sf::Vector2f pos) { return position; };

	int getNumFocos() { return focos.size(); };
	int getNumPlanets() { return planetas.size(); };

	void onInit();
	void onUpdate();
	void onDelete();


	void calcularCentroMasa();
	float calcularMasaTotal();
	void drawSistem(sf::RenderWindow &wnd, const float &spped);


private:

	std::vector<FOCO*> focos;
	std::vector<PLANETA*> planetas;
	sf::Vector2f centroDeMasa;
	float masaTotal;
	//sf::Vector2f position;
	//std::string nombre;
	float ratio;
	sf::CircleShape sistem;
};
```

Foco

```cpp
class FOCO :public CNODO
{
public:
	FOCO();
	~FOCO();
	void setName(std::string name) { nombre = name; };
	std::string getName() { return nombre; };

	void setRatio(float rat) { ratio = rat; };
	float getRatio() { return ratio; };

	void setPosition(sf::Vector2f pos) { position = pos; };
	sf::Vector2f getPosition(sf::Vector2f pos) { return position; };

	void setMasa(float ms) { masa = ms; };
	float getMasa() { return masa; };

	void setShape();
	void setShapeColor(sf::Color color);

	void onInit() {};
	void onUpdate();
	void onDelete() {};
```

```
	void draw(sf::RenderWindow& wnd, const float &spped);

private:
	float ratio;
	float masa;
	sf::Vector2f centroDeMasa;
	sf::Color focoColor;
	sf::CircleShape foco;
};
```

Planeta

```
class PLANETA :public CNODO
{
public:
	PLANETA();
	~PLANETA();
	void addSatelite(SATELITE*);

	void setName(std::string name) { nombre = name; };
	std::string getName() { return nombre; };

	void setRatio(float rat) { ratio = rat; };
	float getRatio() { return ratio; };

	void setPosition(sf::Vector2f pos) { position = pos; };
	sf::Vector2f getPosition(sf::Vector2f pos) { return position; };

	int getNumLunas() { return lunas.size(); };

	void setMasa(float ms) { masa = ms; };
	float getMasa() { return masa; };

	void setSpeeds(float max, float min)
	{
		speedAfelio = max; speedgHepielio = min;
		difVel = max - min;
	};

	void setDistToFoco(float max, float min)
	{
		Afelio = max; perihelio = min;
		ratioTotal = min + max;
	};


	void onInit();
	void onUpdate();
	void onDelete();
	void setShape();
	void setShapeColor(sf::Color color);
	void draw(sf::RenderWindow & wnd, const float &spped);

	void calcularCentroMasa();
	float calcularMasaTotal();
	void setDaysToComplete(float days) { dias = days; };

private:
	float time;
```

```
	float ratio;
	float Afelio;
	float perihelio;
	float speedAfelio;
	float speedgHepielio;
	float masa;
	float m_direction_X = -1;
	sf::Vector2f centroDeMasa;
	sf::Color planetColor;
	sf::CircleShape planeta;
	std::vector<SATELITE*> lunas;
	float ratioTotal = 0;
	float difVel=0;
	float speed = 0;
	float dias = 1000;
};
```

Satelite

```
class SATELITE :public CNODO
{
public:
	SATELITE();
	~SATELITE();
	void setName(std::string name) { nombre = name; };
	std::string getName() { return nombre; };

	void setRatio(float rat) { ratio = rat; };
	float getRatio() { return ratio; };

	void setPosition(sf::Vector2f pos) { relativePosition = pos; };
	sf::Vector2f getPosition(sf::Vector2f pos) { return position; };

	void setMasa(float ms) { masa = ms; };
	float getMasa() { return masa; };

	void setSpeeds(float max, float min)
	{ speedAfelio = max; speedgHepielio = min;
	difVel = max - min;
	};

	void distToPlanet(float max, float min)
	{ Afelio = max; perihelio = min;
	ratioTotal = min + max;
	};

	void onInit() {};
	void onUpdate();
	void onDelete() {};
	void draw(sf::RenderWindow & wnd, const float &spped);
	void setShape();
	void setShapeColor(sf::Color color);
	void setDaysToComplete(float days) { dias = days; };
private:
	float ratio;
	float Afelio;
	float perihelio;
	float speedAfelio;
```

```
	float speedgHepielio;
	float masa;
	sf::Vector2f centroDeMasa;
	sf::CircleShape luna;
	sf::Color lunaColor;
	float ratioTotal = 0;
	float difVel = 0;
	float speed = 0;
	float time;
	float dias=1;
};
```

Cada uno tiene estas características:

1. Posición.
2. Posición relativa.
3. Padre.
4. Un vector de Hijos.
5. Nombre

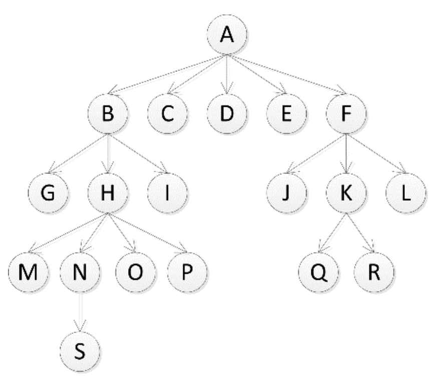
Estas características se les son heredadas de una clase **NODO**

```
class CNODO
{
public:
	CNODO();
	~CNODO();
	CNODO* parent=nullptr;
	std::vector<CNODO*>childrens;
	sf::Vector2f position;
	sf::Vector2f relativePosition;
	std::string nombre="ñ";

private:

};
```

Este nodo esta hecho para formar un árbol (Cuadro descriptivo que representa de forma gráfica las relaciones que tienen los elementos de un conjunto o las variaciones de un fenómeno), específicamente un árbol n-ario.



Echo de esta forma pensado en que una galaxia puede llegar a tener mas de un sistema, un sistema puede tener más de un foco o planetas y un planeta puede tener más de una luna.

Algoritmo de búsqueda

El algoritmo de búsqueda que desarrollamos se basa en la búsqueda del nodo deseado por su nombre, este revisa del primer al ultimo nodo, siendo el mejor caso ω(1) y el peor O(n), con una complejidad de T(n) = (n·log2(n)).

En caso de que se encuentre se regresa el nodo, de lo contrario se regresa un nodo nulo.

El código implementado es el siguiente:

```
CNODO * MANAGER::search(std::string name, CNODO * node)
{
	if (node->nombre != name)
	{
		for (int i = 0; i < node->childrens.size(); i++)
		{
			NODE = search(name, node->childrens[i]);
			if (NODE !=nullptr)
			{
				return NODE;
			}
		}
	}
	else
	{
		return node;
	}
	return nullptr;
}
```

Sistema de movimiento:

El sistema de movimiento de cada planeta se basa en sus distancias con el foco, sus velocidades y el tiempo en días terrestres que se tarda en dar una vuelta al sol.

Primero se calcula la distancia máxima que pose, ósea la suma del perihelio y el afelio, a la vez que se calcula la diferencia de velocidad que tiene, la velocidad máxima que alcanza menos la mínima.

Se calcula el porcentaje de su posición en x respecto a su distancia total (posiciónX/distanciaTotal), luego con este porcentaje se multiplica la diferencia de velocidades y el resultado se le suma a la velocidad mínima para saber la velocidad en ese punto. Con esta velocidad se utiliza una fórmula de velocidad areolar para calcular el radio que debería tener a esa velocidad, y con eso se calcula la posición en Y de astro.

```
void PLANETA::onUpdate()
{
	time +=(1/dias)*speed;
	if (position.x < 0)
```

```
	{
		position.x = perihelio * cos(time);
		if (position.x < -perihelio)
		{
			position.x = -perihelio;
		}
	}
	else
	{
		position.x = Afelio * cos(time);
		if (position.x > Afelio)
		{
			position.x = Afelio;
		}
	}
	float por = perihelio + position.x;
	por = por / ratioTotal;
	speed = speedgHepielio+(difVel*por);
	float ratioTemp;

	ratioTemp = Afelio * (speed / speedAfelio);
	position.y = ratioTemp*sin(time);
  planeta.setPosition(position.x, position.y  );
  anteposition = position;
  for (int i = 0; i < lunas.size(); i++)
  {
	  lunas[i]->onUpdate();
  }
}
```

En el caso del satélite la formula es muy similar, solo que al principio se calcula una posición relativa al planeta para después sumarle la posición real del planeta para darle su lugar en el espacio al satélite:

```
void SATELITE::onUpdate()
{
	time += (1/dias)*speed;
	if (relativePosition.x < 0)
	{
		relativePosition.x = perihelio * cos(time);
		if (relativePosition.x < -perihelio)
		{
			relativePosition.x = -perihelio;
		}
	}
	else
	{
		relativePosition.x = Afelio * cos(time);
		if (relativePosition.x > Afelio)
		{
			relativePosition.x = Afelio;
		}
	}
	float por = perihelio + relativePosition.x;
	por = por / ratioTotal;
	speed = speedgHepielio + (difVel*por);
	float ratioTemp;
```

```
        ratioTemp = Afelio * (speed / speedAfelio);
        relativePosition.y = ratioTemp * sin(time);
        position = relativePosition + parent->position;
        luna.setPosition(position.x, position.y);
        anteposition = position;
}
```

Conclusión:

Las leyes de Kepler fueron de gran ayuda para poder entender mejor el movimiento elíptico de los planetas, y utilizando más fórmulas para poder calcular la posición de un planeta respecto aun tiempo, velocidad y distancia, es posible convertir una orbita circular a una mas elíptica, muy parecida a la que en realidad efectúan los planetas.

El sistema de búsqueda funciona muy bien con todo lo que es nuestro sistema solar, no se ha probado con mas que eso, pero en verdad el peor caso es que no se encuentre el planeta y se haya desperdiciado todo ese tiempo en buscar algo que no existe.

Referencias:

- https://www.bbc.com/mundo/noticias-45706619
- Wiles, Andrew (2006), «The Birch and Swinnerton-Dyer conjecture», en Carlson, James; Jaffe, Arthur; Wiles, Andrew, The Millennium prize problems (en inglés), American Mathematical Society, pp. 31-44, ISBN 978-0-821-83679-8, archivado desde el original el 29 de diciembre de 2009
- https://www.universoformulas.com/fisica/cinematica/movimiento-circular/
- https://sites.google.com/site/fisicacbtis162/in-the-news/movimiento-circular
- http://www.sc.ehu.es/sbweb/fisica/cinematica/circular/circular.htm
- https://www.fisicapractica.com/velocidad-tangencial-mcu.php
- https://www.universoformulas.com/fisica/cinematica/velocidad-tangencial/
- https://www.lifeder.com/velocidad-areolar/
- https://www.youtube.com/watch?v=llRL5BYco_c&feature=youtu.be