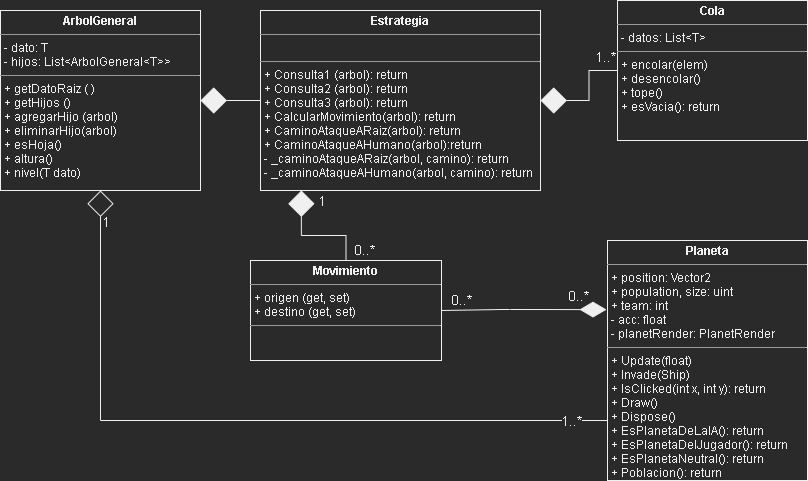
Trabajo Práctico Final   
Complejidad Temporal, Estructuras de Datos y Algoritmos.

* Docente: Leonardo Amet.
* Carrera: Ingeniería Informática.
* Tema: Arboles Generales.
* Año: 2020.
* Comisión: 03.
* Alumno: Ricardo Aguirre.
* Legajo: 48914.



**Diagrama UML:**



**Detalles de la implementación:**

Para la resolución de dicho trabajo practico que se basó en la simulación de un juego que se desarrolla como una conquista planetaria donde el enfrentamiento se da entre dos jugadores, un usuario y una entidad de software (Bot). El problema fue visto como como árbol general en donde cada nodo era un planeta que estaba interconectado a través de aristas. Para resolver la **Consulta 1** se utilizó una adaptación del recorrido con separación de niveles en donde el algoritmo recorría cada nivel en busca del planeta enviado por parámetro y en donde una variable utilizada como contador aumentaba a medida que se iba pasando de nivel en nivel hasta encontrar dicho planeta.

Para la resolución de la **Consulta 2** también se utilizó una adaptación del recorrido por niveles en la cual gracias a una variable utilizada como contador aumentaba a medida que iba encontrando planetas cuya población sea únicamente mayor a 10.

Con respecto a la resolución de la **Consulta 3**  se volvió a utilizar una adaptación del recorrido de niveles con separadores pero esta un complejo debido a lo que pedía este método. Este método pedía calcular y retornar el promedio poblacional en cada nivel del árbol pasado como parámetro. Para esto se fue calculando la cantidad de población total que se dividió con la cantidad de planetas del mismo nivel dando así el promedio, luego ese dato era guardado en una variable. Al momento de guardar ese dato y cambiar de nivel las variables que se utilizaron para llevar a cabo el conteo y suma de los datos se reiniciaban para así poder seguir calculando el promedio de población de los distintos niveles. Luego de esto variable que contiene los promedios de cada nivel era retornada e impresa por pantalla.

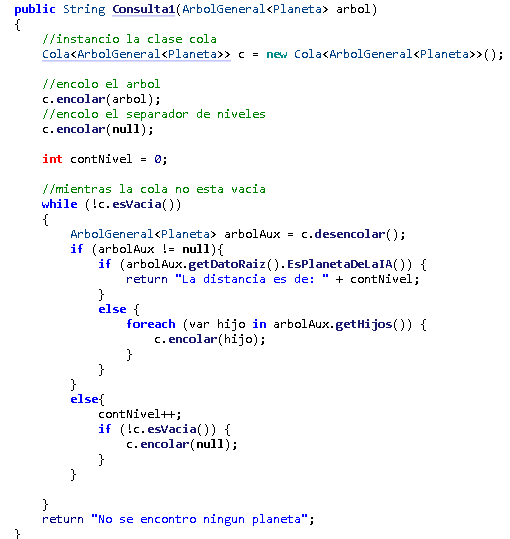
**CalcularMovimiento**  este método se basó en los movimientos apropiados para ir capturando planetas y así intentar que la entidad de software (Bot) logre ganar. Para lograr esto se pensó la estrategia de ataque de dos maneras, la primera se basa en la buscar el camino que lleve a el bot a conquistar la raíz de los planetas y para alcanzar ese objetivo se utilizó una adaptación del recorrido Pre-Orden en el cual se arma un camino desde la raíz hasta el planeta que es controlado por la el bot. Una vez capturada la raíz el próximo paso a seguir es la captura e intento de derrotar el usuario, para esto otra vez se utiliza una adaptación del recorrido Pre-Orden en el cual desde la raíz se recorre planeta por planeta en busca de o los planetas que son dominados por el usuario para así atacarlo y ganar el juego.

A continuación se adjuntan imágenes y pruebas de la correcta funcionalidad del programa.

Consulta 1:

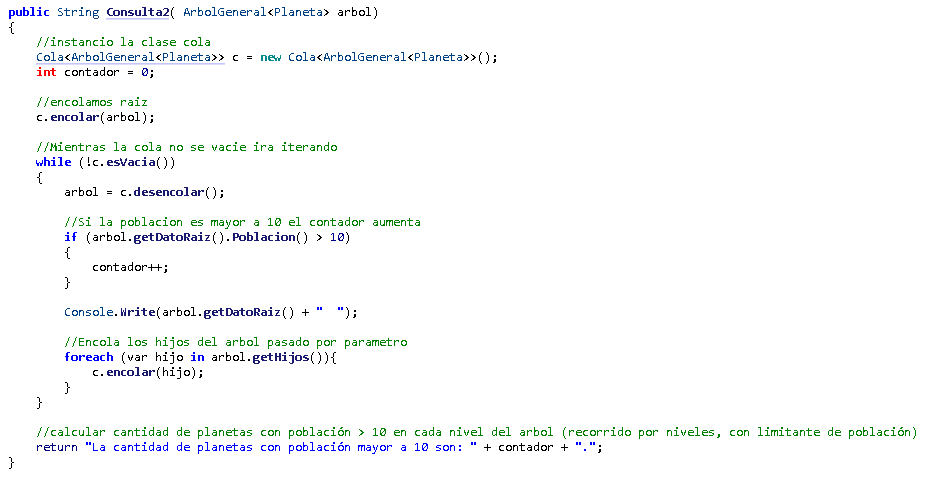
En este método se crea una clase cola que en primer lugar almacena el árbol pasado como parámetro y a continuación un null el cual servirá como separador de niveles, también se hace uso de un contNivel que sirve para contar los niveles. Luego pasa a entrar a un bucle que iterara mientras la cola creada anteriormente no se vacié. Dentro del bucle While se crea un árbol auxiliar que sirve para ir desencolando los valores almacenados en la primer cola para así poder en primer lugar comparar para ver si ese árbol es el que se buscó y paso por parámetro, en caso contrario a ese árbol se le pedirá sus árboles hijos que serán almacenados en la cola para luego ser procesados. Una vez pedido el contador de niveles se incrementara para dar a conocer que ya el algoritmo se encuentra en un nivel más del que comenzó agregando a continuación otro null si es que la cola no se vacío para no caer en un bucle infinito que sirve para llevar un mejor control de los niveles.

Al encontrar el dato el método retornara con un mensaje en pantalla (visto al final) dando a conocer el nivel del planeta que contiene a el bot.



Consulta 2:

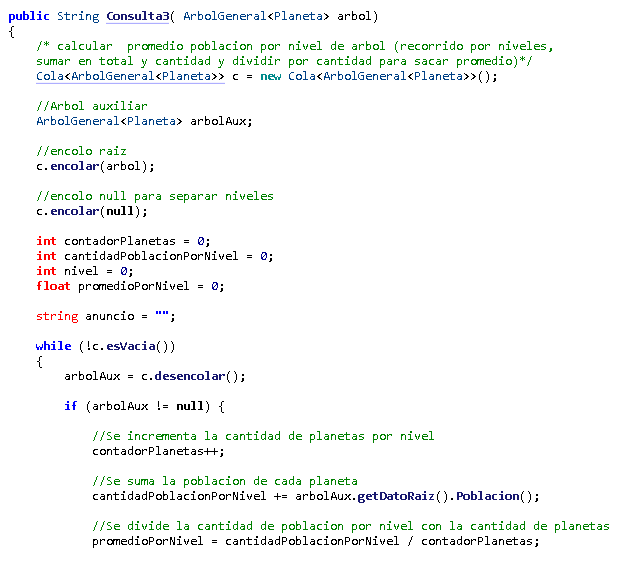
Este método es muy similar al explicado con anterioridad, la diferencia es que no hace falta usar un separador de niveles ya que lo que busca conocer únicamente son a aquellos planetas que poseen una población mayor a 10, para esto el algoritmo recorre y compara la población de cada planeta, al encontrar a alguno que supere los 10 un contador se incrementara. Al llegar al último planeta o hoja se retornara un mensaje en donde se detallara la cantidad total de planetas que cumplen con la condición dada.

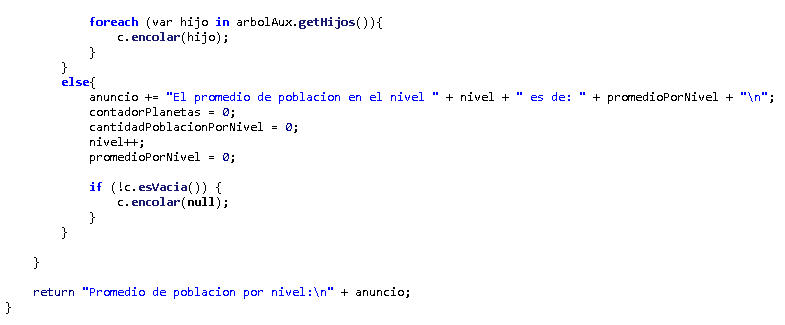


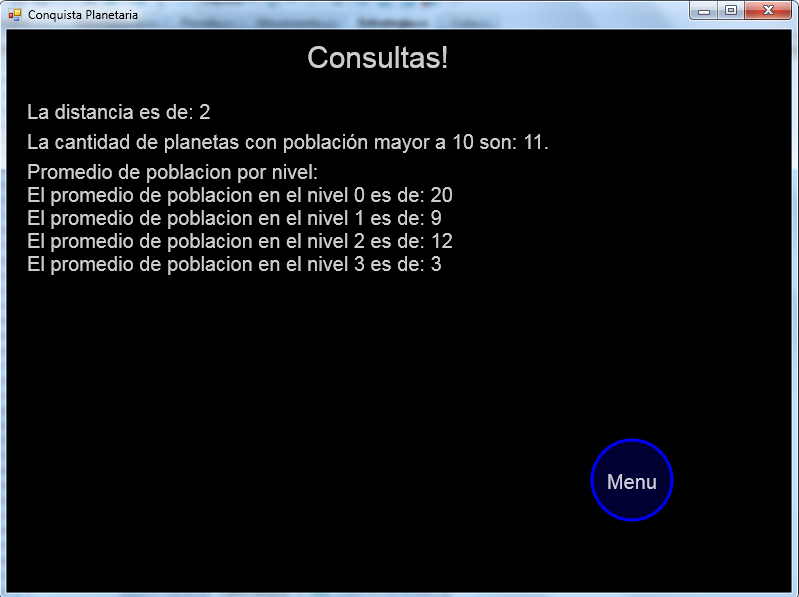
Consulta 3:

El comienzo de este método es similar a los explicados anteriormente, se crea una cola en la cual se agrega el árbol pasado como parámetro, luego se encola un null que sirve como separador de niveles y a continuación se crea un árbol auxiliar para ir controlando los datos. A su vez se instancian una variables instanciadas en 0 o vacias que servirán de ayuda para realizar las operaciones necesarias. Luego de esto se prosigue con la implementación de un bucle que ira iterando hasta que la cola creada no se vacié. El dato guardado en la cola se usa para instanciar el árbol auxiliar, a continuación se hace una comparación para saber si ese dato es un null o no, en caso de no ser un null se incrementaría el contador de planetas y se sumaría la población que posee el mismo, por consiguiente se haría la operación que calcule el promedio que será guardado en otra variable. Luego de esto se pide al árbol los hijos que tuviera para ser encolados y luego procesados.

Al cambiar de nivel las variables que se utilizaron para realizar las operaciones matemáticas se reinician y se incrementa el contador de nivel y se guarda en otra variable el anuncio que será retornado al final del programa con el promedio según el nivel.



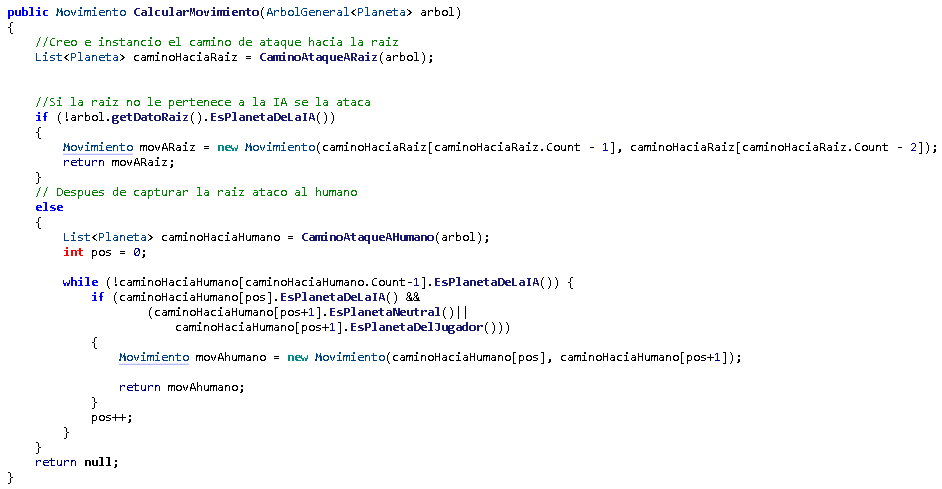


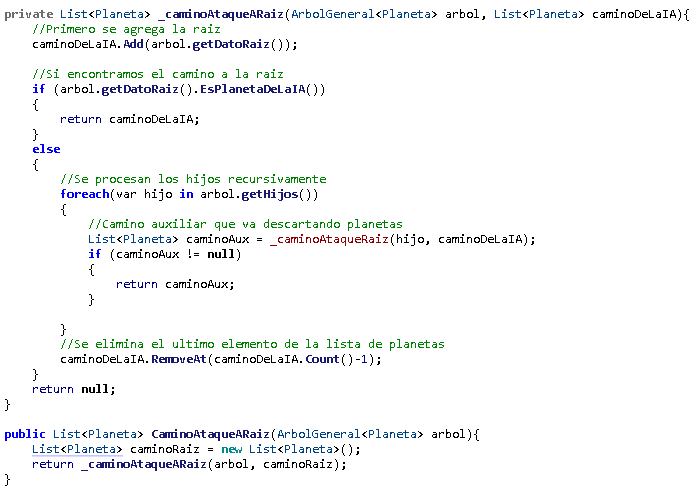


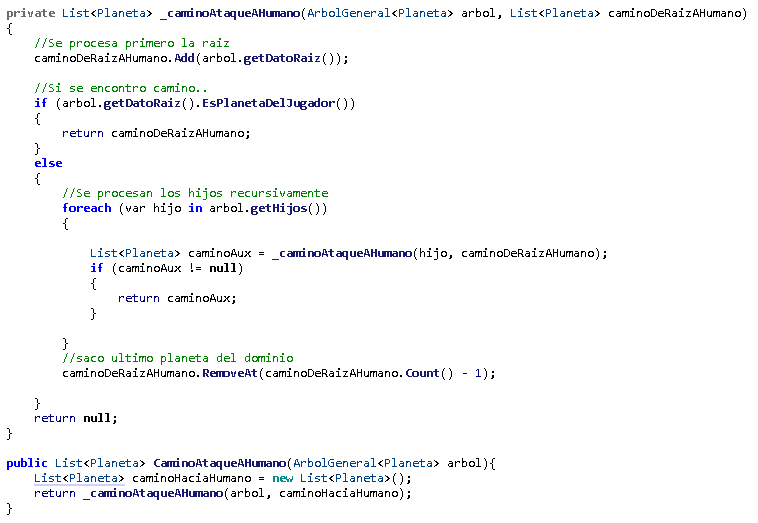
Calcular Movimiento:

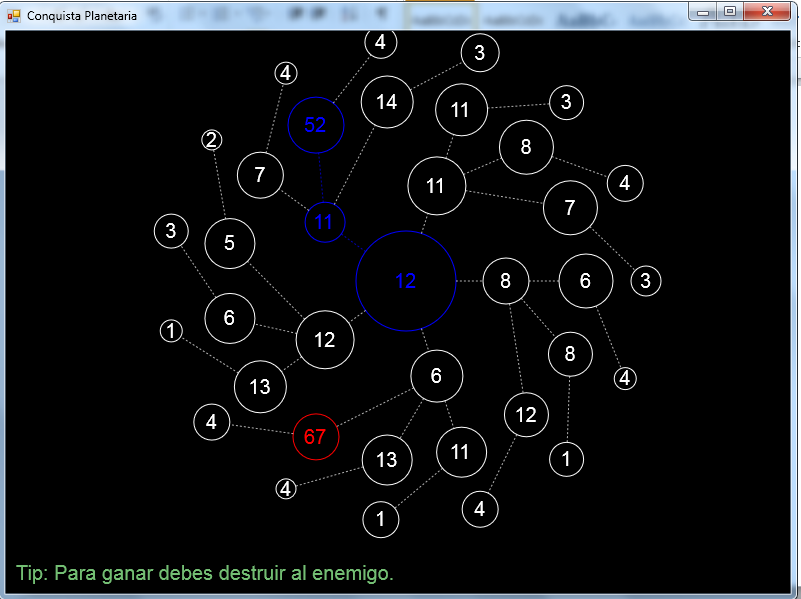
Este método toma se construye a través de otros dos métodos los cuales son CaminoAtaqueARaiz y CaminoAtaqueAJugador los cuales utilizan una adaptación el algoritmo de Pre-Orden en el cual su objetivo es la búsqueda desde el nodo pasado por parámetro a la raíz del árbol. Para esto el algoritmo procesa primeramente la raíz y luego los hijos recursivamente en busca de aquel planeta que sea igual al buscado. En caso de hallarlo el método retornara el camino.

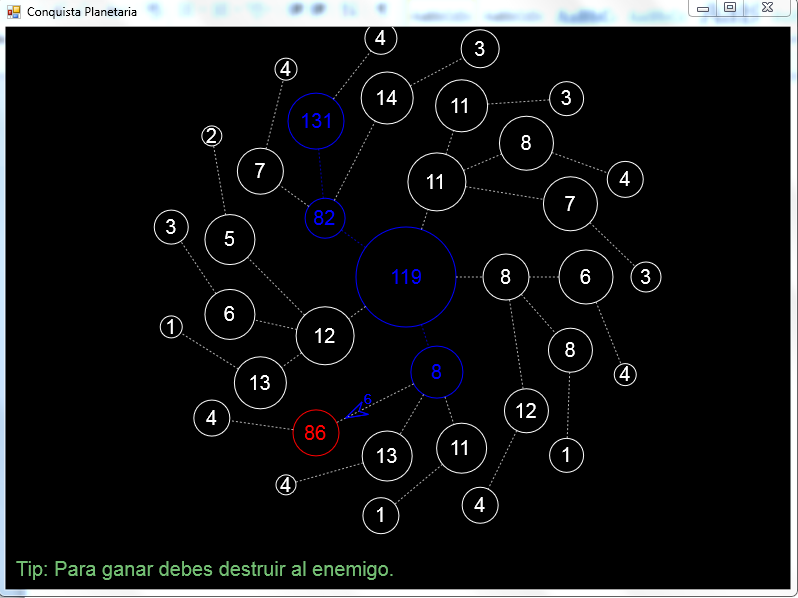
Luego de esto el método calcular movimiento se encarga de retornar aquellos movimientos de origen a destino. Para lograr esto el algoritmo ira sacando aquellos planetas dentro de la lista que recibió y los utilizara como origen en primer lugar a donde se encuentre parada la IA y como destino al planeta más cercano que lo lleve a la raíz. Debido a que los datos fueron almacenados en primer lugar la raíz y por último el planeta de la IA los Planetas guardados en la lista son des apilados, comenzando desde el ultimo que haya sido guardado.











Alguna de las ideas que se podrían llevar a cabo para mejorar el programa sería la de mejorar la IA agregándole algunas funcionalidades de movimientos más rápidos ya que el usuario tiene la posibilidad de moverse sin límite de tiempo, en cambio el bot cuenta con un movimiento cada ciertos segundos.

Otra idea sería la de modificar el movimiento a la raíz, ya que muchas veces se acumula la población en un planeta y esas tropas no son enviadas al planeta siguiente para ayudar a la captura del usuario, se debería buscar la manera de que a pesar de que la IA halla capturado la raíz esta siga enviando su población para ayudar a aquellos planetas que menos tienen.

**Reflexión:**

Con respecto al planteamiento de dicho trabajo final estoy conforme con lo pedido ya que estos son los que ayudan a practicar más y generar así los conocimientos necesarios para afrontar otro tipo de retos, debido a la versatilidad y maneras de poder resolverlo. Si bien en un comienzo se tornó difícil con el tiempo y gracias a la práctica uno puede ir pensando las cosas de otra manera lo cual va facilitando el aprendizaje. La experiencia adquirida gracias a este tipo de trabajos es mucha lo cual suma y aporta para seguir mejorando y generando esas ganas de aprender y conocer más a fondo los temas tratados.