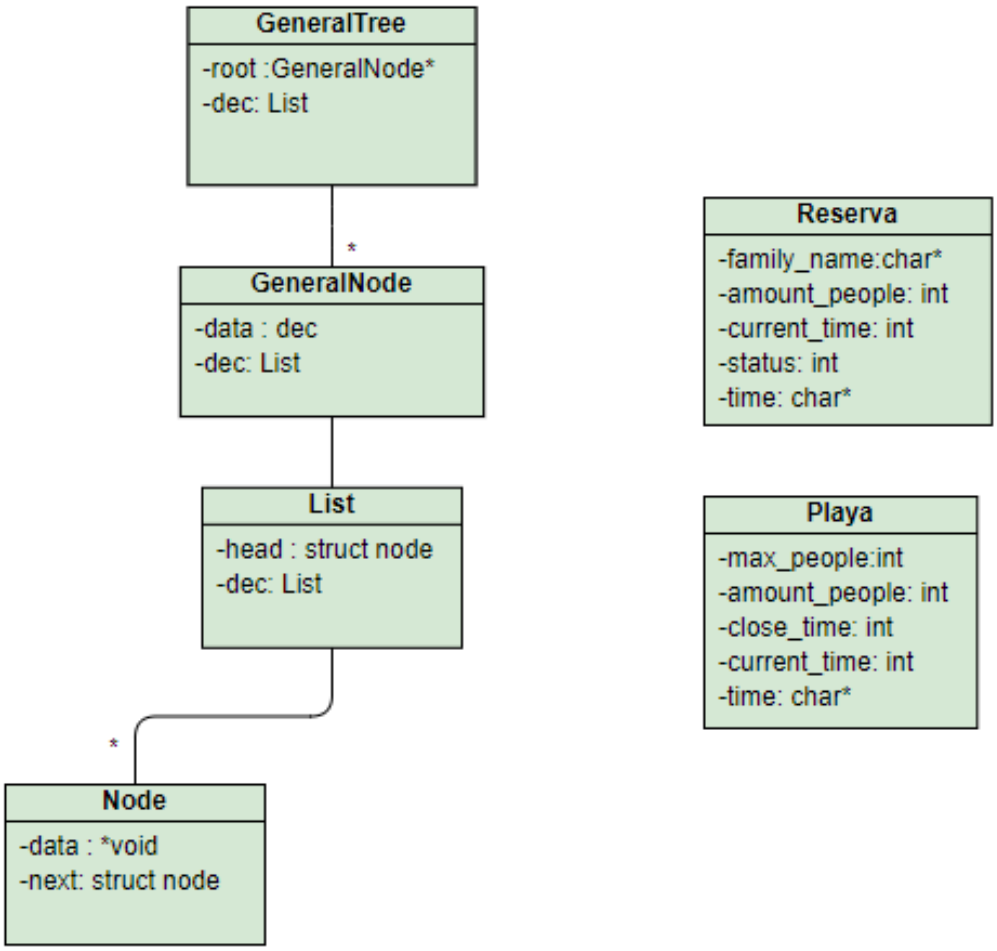
**PROYECTO - ENTREGA 1**

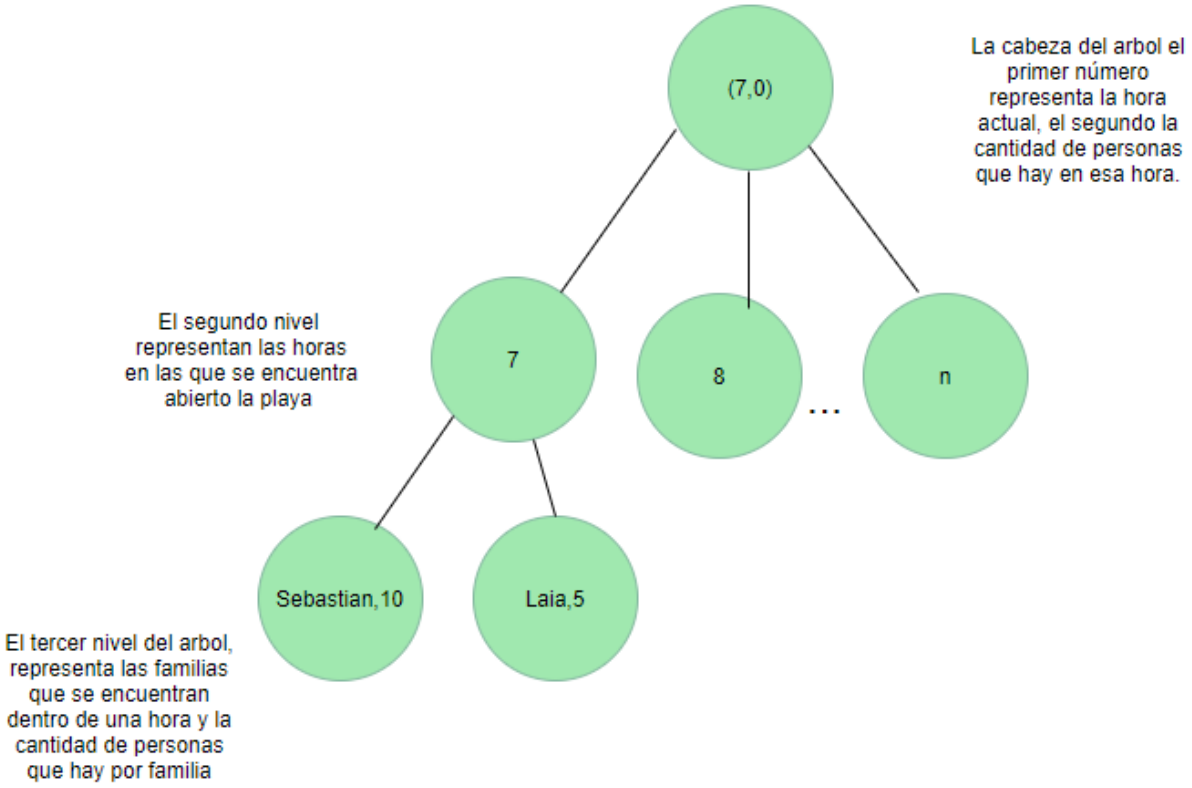
**JOSE DANIEL NIÑO MUÑOZ**
**JORGE LUIS ESPOSITO ALBORNOZ**
**JUAN SEBASTIÁN HERRERA GUAITERO**


**RICARDO HJALMAR GONZÁLES GARCÍA**
**PONTIFICIA UNIVERSIDAD JAVERIANA**
**FACULTAD DE INGENIERÍA**
**SISTEMAS OPERATIVOS**
**BOGOTA D.C**
**27 DE ABRIL DE 2021**

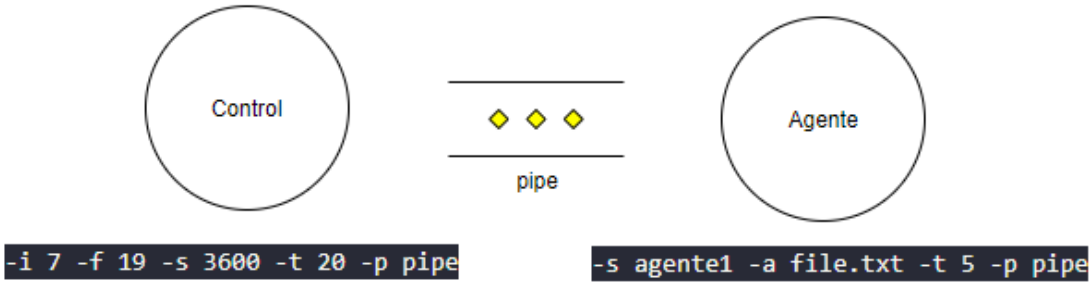
**Relación entre estructuras**



## Almacenamiento de la información



Por el momento solo se está utilizando un pipe para la comunicación entre los procesos controlador y agente.




Funciones en control.c

- **set_tree**:
  Función de tipo void, recibe como parámetro un puntero de un puntero a una variable de tipo “GeneralTree” y dos char pointers. La función se encarga de inicializar el árbol, y después dentro de un for se insertan los nodos de cada hora de un día hasta la hora de cierre.

- printTree:
  Función de tipo void, recibe un puntero a un “GeneralNode”. Esta función imprime la información que encuentra en el puntero que le es enviado de forma recursiva para ir recorriendo nodo a nodo.

Funciones en data.h

- write_pipe:
  Función de tipo void, recibe como parámetro un int que será el modo en el que se va a usar el file descriptor, void* buf, parâmetro genérico que contiene la información que será enviada por el pipe, size_t representa el tamaño en bytes de buf, char* pipe, el nombre de la tubería por donde se van a comunicar y flag representa la forma en la que se abre el pipe.La función envía la información a través de un pipe.

- read_pipe:
  Función de tipo void, recibe como parámetro un int que será el modo en el que se va a usar el file descriptor, void* buf, parâmetro genérico que contiene la información que será enviada por el pipe, size_t representa el tamaño en bytes de buf, char* pipe, el nombre de la tubería por donde se van a comunicar los procesos y flag representa la

forma en la que se abre el pipe.La función lee  la información enviada a través de un pipe.

- drop_space:
  Función de tipo puntero a char, recibe como parámetro char* que será  el parámetro al que se le eliminará todos los espacios entre las comas para así retornar un nuevo char* con la línea sin espacios.

- int_to_char:
  Función de tipo puntero a char, recibe como parámetro dos int. El primer int es el valor que se quiere transformar a char y el segundo se encarga de definir si se le agrega una coma al inicio.Se retorna el puntero dirigido al char.

- getCurrentTime:
  Función de tipo int, recibe como parámetro un puntero a un “GeneralNode” en este caso específicamente a la raíz del árbol. La función obtiene el valor numérico que representa el tiempo actual dentro de la cabeza del árbol.

- setHeadTime:
  Función de tipo void, que recibe un doble puntero a la cabeza, un tiempo y la cantidad de gente para actualizar así la información dentro de la cabeza del árbol

- update_tree:
  Función de tipo void,actualiza la información del árbol haciendo uso de setHeadTime.

- getAmountPeopleByHour:
  Función de tipo int, recibe como parámetro un puntero a un puntero a un “GeneralTree” y un int. La función busca un nodo que tenga la hora correspondiente al int de parámetro, si lo encuentra copia la información del nodo en un auxiliar, y revisa en su totalidad, transformando la cantidad de personas de cada familia en int y haciendo una sumatoria para regresar ese número, si no encuentra un nodo o la lista está vacía, devuelve un 0.

- sufficent_space:
  Función de tipo int, recibe como parámetro un puntero a un “GeneralTree”, una variable reserva y una variable beach. La función busca en las horas que se encuentra la playa abierta un espacio para re ubicar la reserva que se le manda como parámetro, si la encuentra devuelve la nueva hora de reserva, de lo contrario envía un -1.

- answer_request:
  Función de tipo int, recibe como parámetro un puntero a un puntero a un “GeneralTree”, otro puntero a una variable reserva y otro puntero a una variable beach. La función se

encarga de ver si la reserva se puede agendar, si se puede agendar la inserta en el nodo, de lo contrario se busca una nueva hora para la reserva, esto puede acabar de dos formas. Si se encuentra una nueva hora de reserva, se reserva en el nuevo horario. Para los dos casos anteriores se retorna un 1. De lo contrario, si no se logra encontrar un espacio para la reserva, se retorna un -1.

- sortbubble:
  Función de tipo puntero a un int, recibe como parámetro un puntero a un “GeneralTree” y otro a un int. La función organiza las horas por la cantidad de personas que estaban en cada hora. Y regresa un arreglo ordenado de mayor a menor de dicha lista.

- horaPico:
  Función de tipo puntero a char, recibe como parámetro un puntero a un “GeneralTree”. Busca la hora con mayor cantidad de personas y en caso de existir 2 horas o mas con la misma cantidad de personas, y todas tienen la mayor cantidad de personas en comparación al resto de horas, se agrega a la lista de horas y se regresan las horas con mayor cantidad de personas.

- menorinfluencia:
  Función de tipo puntero a char, recibe como parámetro un puntero a un “GeneralTree”. Busca la hora con menor cantidad de personas y en caso de existir 2 horas o mas con la misma cantidad de personas, y todas tienen la menor cantidad de personas en comparación al resto de horas, se agrega a la lista de horas y se regresan las horas con menor cantidad de personas.
- report:
  Función de tipo void, recibe como parámetro un puntero a un “GeneralTree”. La función se encarga de imprimir toda la información de lo ocurrido en la simulación, imprime las horas de mayor y menor flujo de personas y la cantidad de solicitudes aceptadas, negadas y reprogramadas.