**INFORME**

La estructura seleccionada para este problema fue un montículo binario (POR QUE). El orden de complejidad O en:

* *Inserciones:* dado que la función Insertar agrega el elemento al final de la lista y que luego es desplazado hacia arriba con la función InfiltrarArriba, el orden de complejidad es , que representa la altura en el montículo (siendo n la cantidad de elementos).
* *Eliminaciones:* realizando el mismo análisis que con las inserciones, la función eliminarMin también tiene una complejidad . Depende de la cantidad de elementos y la cantidad de comparaciones que realiza entre esos elementos la función InfiltrarAbajo, la cual es utilizada por eliminarMin.

Para resolver este problema, tomamos en cuenta que la consigna solicita aplicar una estructura de datos adecuada para almacenar los pacientes conforme ingresan al centro de salud, utilizando dos criterios, el primero y más importante radica en un proceso que permite una gestión de riesgo clínico para manejar adecuadamente los flujos de pacientes, categorizándolos en tres niveles de riesgo, el segundo criterio que implementamos fue el del orden de llegada, asignando con la función:

self.\_\_llegada = Paciente.contador\_llegada

esta información se almacena como parte de los datos que hay de cada paciente.

En la Sala de Emergencias, se determina que la variable cola\_de\_espera es la cola de prioridad, definida en un módulo de la carpeta modules. También se denomina una variable “paciente” y se inserta a la cola de prioridad a ser acomodada entre los otros datos, tomando en cuenta su nivel de riesgo y si coincide con algún otro dato, se acomoda en base al orden de llegada