# ALGORITMOS A1, A2, A3.1, A3.2.

Estos cuatro algoritmos parten del algoritmo base realizado en el TFG. A partir de éste, se añaden nuevas restricciones y se crean los algoritmos A1, A2, A3.1, Y A3.2.

**A1:** En este algoritmo se toma que cada cirujano puede pertenecer a **más de una subunidad** quirúrgica, y por tanto tener para cada una de ellas un nivel de experiencia distinto. Además, se limita el número de equipos que operan en un quirófano por turno a **un solo equipo**.

**A2:** Se tienen en cuenta las restricciones del algoritmo A1, y además se crean distintos **tipos de consultas**. En este caso tendremos consultas Generales, Nominativas y Específicas de una subunidad. Se da como dato los candidatos a cada consulta y la unidad.

**A3:** A partir del algoritmo A1 (A3.1) o A2 (A3.2), se añaden dos condicionantes más. Por un lado, ha de tenerse en cuenta el **tipo de anestesia** que el paciente necesita en su intervención a la hora de asignarle quirófano. Los quirófanos se dividen en aquellos que permiten administración de anestesia general y aquellos que no. Por tanto, los pacientes que requieran anestesia general han de ser asignados a los quirófanos que sí tengan recursos para ello. Por otro lado, se añade un **tiempo mínimo** que ha de pasar entre la cirugía y la consulta posoperatoria.

## VERSION1. ‘EXTENSIONTFGlargo’. TIEMPO DE EJECUCIÓN DEMASIADO ALTO

El **algoritmo A1** sigue exactamente la misma estructura que el algoritmo base del TFG. Simplemente, en el código se han cambiado las condiciones para asignar un cirujano a una intervención en la función busquedaOR. También se han añadido restricciones en esta misma función para que sólo haya un equipo quirúrgico en un OR cada día.

El **algoritmo A2**, sin embargo, sí tiene una estructura ligeramente diferente al algoritmo base del TFG. Hay que asignar pacientes y cirujanos a las consultas Nominativas y de Especialidad, pero sin embargo en las Generales sólo se asigna el cirujano. En la función solución DS1, se realiza la asignación de pacientes a las consultas nominativas y de especialidad, y finalmente se asignan aquellos cirujanos que hayan quedado libres a las consultas generales. Se asignan primero aquellos que lleven menos horas de consulta acumuladas.

En la función DS2, como la asignación de cirujanos a consulta es dato de entrada, simplemente se añade una condición que verifica que la consulta sea de la unidad a la que pertenece el paciente.

Dado que hay distintos tipos de consulta, no se puede hacer un algoritmo Iterative Greedy para la asignación de cirujanos a consulta. Por ello, en vez de hacer un algoritmo IG, se realiza un Local Search, insertando todos los posibles cirujanos en las consultas de Especialidad, y el algoritmo se queda con aquella combinación que mejores resultados obtenga.

La estructura general quedaría como se muestra a continuación:

Diagrama

Descripción generada automáticamente

El pseudocódigo del Local Search del vector cirujanosconsultas es el siguiente:

Texto

Descripción generada automáticamente

El **algoritmo A3.1** parte del algoritmo A1, y el **algoritmo A3.2** parte del algoritmo A2. A estos se les añade las condiciones previamente mencionadas que requiere el algoritmo A3, pero no existe ningún cambio de estructura. La restricción del tipo de anestesia se añade en la función busquedaOR. Además, para tener en cuenta el tiempo mínimo que debe pasar entre la operación y la consulta postoperatoria, se comprueba que el paciente lleve en la lista los días necesarios antes de llamar a la función busquedaPOSTOP. Así, no se asignará a una consulta posoperatoria si no ha pasado el suficiente tiempo aún.

## VERSIÓN 2. ‘EXTENSIONTFGmedio’. TIEMPO DE EJECUCIÓN ACEPTABLE (HORAS)

Esta versión es una modificación sobre el algoritmo final, A3.2, para acortar el tiempo de ejecución. Se han eliminado aquellos procesos que más tiempo consumían. Estos son los procesos NEH y Búsqueda Local. Por tanto, en esta versión, en vez de realizar un proceso NEH, se inicializa la secuencia por EDD (Earliest Due Date). A continuación, se muestra la estructura de A3.2:

Diagrama

Descripción generada automáticamente

Siguiendo este algoritmo, el tiempo de ejecución para una instancia de 1538 pacientes es de aproximadamente 6 horas.

## VERSIÓN 3. EXTENSIONTFGcorto’. TIEMPO DE EJECUCIÓN MUY CORTO (MINUTOS)

En esta versión se consigue que el tiempo de ejecución del algoritmo A3.2 sea de segundos o minutos para una instancia grande de unos 1500 pacientes en la lista de espera. Esto se consigue siguiendo la misma estructura que la versión anterior y, además, la construcción se realiza de manera aleatoria. Es decir, se destruyen 4 pacientes de la secuencia y se insertan en 4 posiciones aleatorias de la secuencia. El proceso de destrucción+construcción se realiza de manera iterativa hasta cumplir un tiempo de parada. En este caso se ha establecido en 30 segundos.

Texto

Descripción generada automáticamente

Siguiendo esta versión, el tiempo de ejecución de este algoritmo para una instancia de 1538 pacientes es de aproximadamente 5 minutos.