1. Análisis de alto nivel

El programa comienza su ejecución en la clase *InterfazHospital,* donde empezará a ejecutarse la primera parte del enunciado. En primer lugar, los pacientes entrarán a una cola de espera en la *Recepción* del hospital. Aquí, el *Auxiliar1* se encargará de registrar a los pacientes, teniendo en cuenta si estos tienen cita o no. Si no hay ningún problema y los pacientes están citados, procederán a la sala de vacunación. Después de registrar a 10 pacientes, este *Auxiliar* se tomará un descanso.

En la *SalaVacunacion* los pacientes accederán a un puesto aleatorio, del 1 al 10, y bien esperarán a un sanitario, o bien si ya hay un sanitario dentro, procederán a vacunarse en el tiempo aleatorio establecido. Una vez finalizada la vacunación, tendrán que acceder a una *SalaObsrvacion* para reposar 10 segundos antes de marchar. Pasado ese tiempo, es posible que al *Paciente* le de una reacción a la vacuna, por lo que tendrá que llegar un *Sanitario* para atender esa reacción. Después de tratarlo, el *Paciente* podrá marchar y el *Sanitario* volver a su trabajo.

Por otra parte, los *Sanitarios* irán derechos a la *SalaDescanso* para cambiarse al llegar al hospital. Una vez pasados por el *vestidor*, accederán a un puesto aleatorio en la *SalaVacunacion*, en la cual comenzarán a vacunar a los pacientes cuando ocupen ambos el mismo puesto. Cada vez que un *Sanitario* vacune a 15 *Pacientes*, se tomará un descanso en la *SalaDescanso*. En esta *SalaVacunacion* también encontraremos al *Auxiliar2*, que estará preparando vacunas constantemente, y accederá a la *SalaDescanso* cada vez que prepare 20 vacunas.

En el siguiente paso entraremos en *ServidorHospital*, el cual nos permite controlar de forma remota el funcionamiento del hospital, copiando los datos de la primera interfaz con un segundo de diferencia. Gracias a esta interfaz implementada con sockets, podremos visualizar el recorrido de los hilos por el hospital y tendremos la capacidad de cerrar puestos de vacunación haciendo click en el botón del puesto que queramos cerrar. Esto ocasionará que, si está lleno, los sanitarios terminen de vacunar a los pacientes y accedan a la sala de descanso, y así proceder a la limpieza del puesto cerrado.

2) – 3). Diseño general del sistema y de las herramientas de sincronización utilizados; Clases principales que intervienen con su descripción

* **Clase Recepción**: la clase Recepción será la encargada de recoger a los pacientes que llegan, meterlos en una cola, y hacer que el auxiliar los registre para poder entrar a vacunarse. A esta clase se le han pasado los atributos de JTextField necesarios para que la interfaz funcione, así como los hilos de Paciente y Auxiliar, un arrayList para guardar en cola a los pacientes, dos semáforos para control, y un lock con su respectivo condition. También se le ha pasado la clase de *EscrituraTexto*, la cual nos permitirá crear el log en un .txt

Al constructor de esta clase se le pasarán los JTextField necesarios, así como la variable creada de escritura de texto.

* + El primer método que encontramos es **auxiliarRegistra().** Encontramos un for de 10 que controlará cuándo el auxiliar ha registrado a 10 pacientes para poder tomarse un descanso. Después, cerramos el semáforo de *auxiliarRegistra*, que se liberará en el siguiente método, entrelazándolos. Escribimos en la interfaz gracias a *auxiliarTxt* así como en el log y por pantalla con las siguientes dos líneas. Después, el auxiliar se toma un tiempo aleatorio de entre 0,5 y 1 segundo para registrar al paciente, y finalmente libera el semáforo *pacienteEspera*
  + **pacienteEspera()** liberará el semáforo de *auxiliarRegistra* para que le método anterior pueda funcionar correctamente, y luego cerrará su propio semáforo con el mismo nombre para terminar de entrelazar los dos métodos.
  + **pacienteEntra()** hará que cada paciente se añada a *colaRecepcion*, la cual irá almacenando en dicho *ArrayList*. Después, imprimirá en la interfaz la cola que haya.
  + **Salir()** se encargará de sacar al paciente de la recepción, quitando a dicho paciente de la *colaRecepcion* y actualizando el método imprimir en la interfaz. Finalmente pasaremos el identificador del paciente por el recuadro de la interfaz que indicará que es el siguiente para acceder al puesto de vacunación
  + **imprimirColaEspera()** se encargará de recorrer la cola de recepción para imprimirla por la interfaz del hospital, si es que esta cola no se encuentra vacía.
* **Clase** **SalaDescanso**: esta clase manejará el descanso de los sanitarios y de los auxiliares, así como el vestidor al que los sanitarios tienen que acceder antes de entrar a trabajar. Como atributos, se le pasarán los JTextFields necesarios para imprimir en la interfaz, así como la clase EscrituraTexto y los ArrayList de los descansos de sanitario y auxliar.
  + **vestidorSanitario()** hará que el sanitario entre a cambiarse antes de vacunar. Se añadirá el sanitario a la cola de descansoS y se imprimirá por la interfaz. Después, tardará entre 1 y 3 segundos para cambiarse, se le sacará de la cola descansoS y volverá a actualizar la impresión por interfaz para removerlo.
  + **descansoSanitario()** tendrá un procedimiento bastante parecido al anterior método, solo que esta vez imprimiremos en el log cuando comienza y cuando finaliza este descanso, y cambiaremos los tiempos de descanso que tiene (siendo estos de 5 a 8 segundos). Lo protegemos en todo momento con un lock.
  + **descansoAuxiliar1()** se encargará de hacer que el Auxiliar en *la recepción* se tome un descanso cada vez que registre a 10 pacientes. Protegido con un lock, introducirá al auxiliar en su respectiva cola de descanso *(descansoA),* imprimirá esta cola por la interfaz y procederá a descansar entre 3 y 5 segundos. Después, se le eliminará de la cola y se actualizará la impresión de la interfaz. También escribiremos en el log cuándo comienza y cuándo acaba su descanso, con *et.inicioDescansoA(a*) y *et.finDescansoA(a)* respectivamente.
  + **descansoAuxiliar2()** cumple exactamente la misma función que el método anterior, solo que ajustándose a los tiempos de descanso de el auxiliar 2 después de preparar 20 vacunas, siendo este descanso de entre 1 y 4 segundos
  + **imprimirColaDescansoAuxiliar()** e **imprimirColaDescansoSanitario()** serán los métodos encargados de recorrer los ArrayList de los descansos de auxliar y sanitario para imprimirlos por la interfaz, siempre que ninunguno de estos esté vacío.