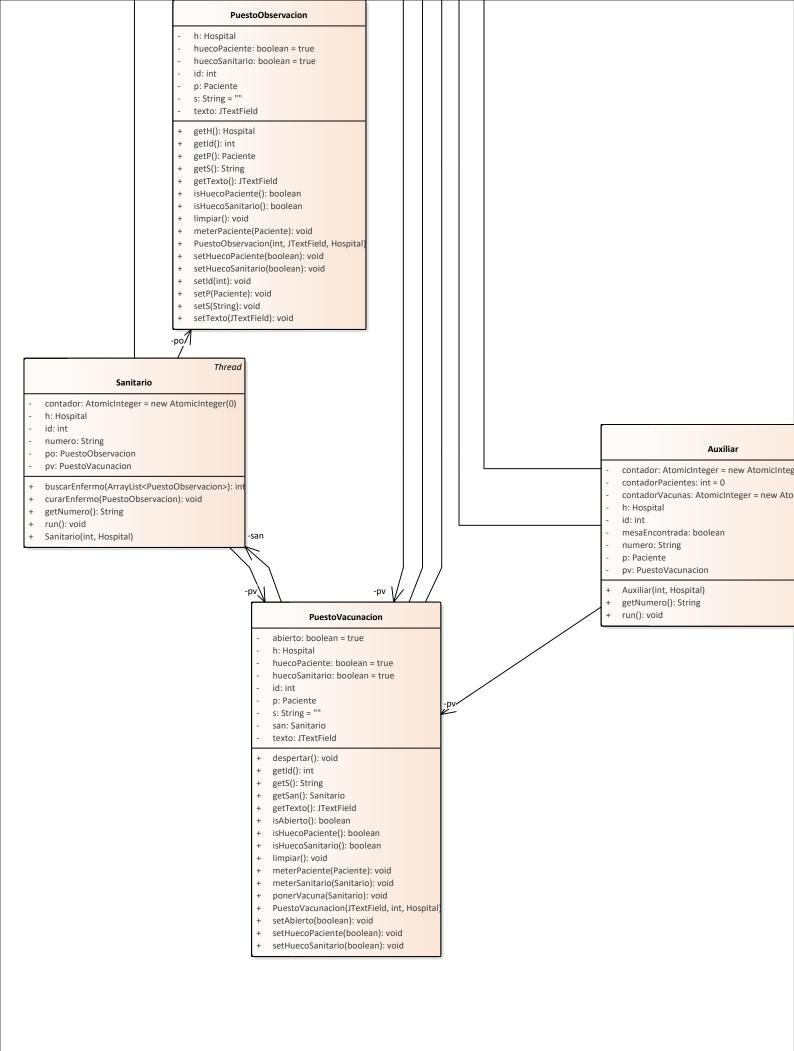
Hospital aux1: JTextField aux2: JTextField cerrojo: Lock = new ReentrantLock() colaEspera: JTextArea comprobarDatos: BlockingQueue<Paciente> = new LinkedBlock... descansoAux: ConcurrentLinkedQueue = new ConcurrentL... descansoSan: ConcurrentLinkedQueue = new ConcurrentL... mesaAsiganada: BlockingQueue<PuestoVacunacion> = new LinkedBlock... nuevaEjecucion: boolean = true pacienteAtendiendo: JTextField puestoObservacionAsignado: BlockingQueue<PuestoObservacion> = new LinkedBlock. puestosObservacion: ArrayList<PuestoObservacion> puestosObservacionLibres: BlockingQueue<PuestoObservacion> = new LinkedBlock... puestosVacunaciones: ArrayList<PuestoVacunacion> puestosVacunacionLibres: BlockingQueue<PuestoVacunacion> = new LinkedBlock... recepcion: ConcurrentLinkedQueue<Paciente> = new ConcurrentL... salaDescanso: JTextPane salaObservacionSemaforo: Semaphore = new Semaphore(20) salaVacunacionSemaforo: Semaphore = new Semaphore(10) vacunas: AtomicInteger = new AtomicInteger(0) vacunasDisp: JTextField volanteVacuna: BlockingQueue = new LinkedBlock... añadirVacunas(): AtomicInteger getAux1(): JTextField getAux2(): JTextField getColaEspera(): JTextArea getComprobarDatos(): BlockingQueue<Paciente> getDescansoAux(): ConcurrentLinkedQueue getDescansoSan(): ConcurrentLinkedQueue getMesaAsiganada(): BlockingQueue<PuestoVacunacion> getPacienteAtendiendo(): JTextField getPuestoObservacion(): PuestoObservacion getPuestoObservacionAsignado(): BlockingQueue<PuestoObservacion> getPuestosObservacion(): ArrayList<PuestoObservacion> getPuestosObservacionLibres(): BlockingQueue<PuestoObservacion> getPuestosVacunaciones(): ArrayList<PuestoVacunacion> getPuestosVacunacionLibres(): BlockingQueue<PuestoVacunacion> getRecepcion(): ConcurrentLinkedQueue<Paciente> getSalaDescanso(): JTextPane getSalaObservacionSemaforo(): Semaphore getSalaVacunacionSemaforo(): Semaphore getVacunas(): AtomicInteger getVacunasDisp(): JTextField getVolanteVacuna(): BlockingQueue Hospital(JTextArea, JTextField, JTextField libreSanitario(): PuestoVacunacion meterLog(String): void puestoConHuecoPaciente(PuestoVacunacion): void puestoObsConHuecoPaciente(PuestoObservacion): void recorrerColaDescanso(Hospital): String recorrerColaEspera(ConcurrentLinkedQueue<Paciente>): String recorrerColaObservacion(ConcurrentLinkedQueue<Paciente>): String setPuestosObservacion(ArrayList<PuestoObservacion>): void setPuestosObservacionLibres(BlockingQueue<PuestoObservacion>): void setPuestosVacunaciones(ArrayList<PuestoVacunacion>): void setPuestosVacunacionLibres(BlockingQueue<PuestoVacunacion>): void vacunar(): AtomicInteger -h/\ -h //\\ -h -h Thread **Paciente** Thread h: Hospital ServidorSockets id: int numero: String conexion: Socket ocupado: CountDownLatch = new CountDownLa conexion2: Socket po: PuestoObservacion entrada: DataInputStream pv: PuestoVacunacion entrada1: DataInputStream Reaccion: boolean = false id: int getHospital(): Hospital salida: ObjectOutputStream getNumero(): String servidor1: ServerSocket getOcupado(): CountDownLatch servidor2: ServerSocket isReaccion(): boolean Paciente(int, Hospital) run(): void run(): void ServidorSockets(Hospital, int) setReaccion(boolean): void

~po



Thread

er(0)

micInteger(0)