



PROGRAMACIÓ 1

CURS: 2022/2023-Q2

PRÀCTICA

1 Sistema de gestió de pacients

El servei de salut pública ens ha demanat ajuda pel desenvolupament d'una prova de concepte (PoC) per millorar la gestió de pacients. Vol que implementem la simulació d'un nou sistema de gestió dels pacients d'un servei d'urgències per decidir si l'aplicarà en hospitals de diverses localitats.

El sistema a dissenyar ha de permetre donar d'alta i de baixa pacients al servei d'urgències, modificar l'estat d'un pacient, fer cerques de pacients de forma eficient i gestionar l'atenció mèdica tenint en compte la gravetat de l'estat de salut del pacient. A més, el sistema ha de permetre programar visites dels pacients a metges perquè puguin ser atesos més endavant.

En el registre d'un/a pacient hi constaran el seu nom complet, edat, motiu pel qual ingressa a urgències i la gravetat (baixa, mitjana o alta) del seu estat de salut. El sistema ha de mantenir els registres de tots els pacients que ingressen al servei d'urgències així com una llista pels pacients que estan esperant a ser atesos.

Quan un pacient arriba al servei d'urgències, se'l dona d'alta al sistema fent servir el seu nom complet com a identificador. En funció de la gravetat del seu problema de salut, se situa en un lloc o altre de la llista dels pacients que esperen a ser atesos, de manera que els casos més greus siguin atesos primer. Un cop atès, s'elimina d'aquesta llista. Si el cas ho requereix, per tal de fer un seguiment de l'evolució del pacient, es pot programar una visita del pacient a un/a doctor/a per una data concreta. Les visites no s'atenen al servei d'urgències sinó a les consultes de cada doctor/a i, per aquest motiu, les visites s'associen als doctors.

Atès que es tracta d'una prova de concepte, ara per ara, no es dissenyaran totes les funcionalitats d'un sistema real.

2 Dades d'entrada i resultats

Per veure els detalls concrets del format de les dades d'entrada, convé consultar el joc de proves públic.

Els valors per indicar la gravetat d'un problema de salut seran: 1 per gravetat alta, 2 per mitjana i 3 per baixa. Les dates per programar visites seran sempre correctes, és a dir, no caldrà comprovar la seva validesa.



Els fitxers d'entrada contindran un conjunt d'operacions que sempre finalitzaran amb l'operació `fi`.

Per facilitar la llegibilitat, totes les operacions s'escriuran a partir de la primera columna juntament amb tots els seus paràmetres separats per un espai. Els resultats s'escriuran a partir de la columna 3 i, en el cas que una operació requereixi escriure més d'un element en la mateixa línia, aquests estaran separats per un espai. Per a més detalls, consulteu el joc de proves públic.

3 Detall de les operacions

En aquest apartat es descriuen les funcionalitats previstes pel nostre sistema. El repertori d'operacions és el següent:

- **alta_pacient**: S'introdueixen les dades d'un pacient (nom complet, edat, motiu de l'ingrés i gravetat). Si el pacient ja existeix en el sistema, es produeix un error; sinó, el pacient és donat d'alta en el sistema i és afegit a la llista d'espera en funció de la gravetat del seu estat de salut.
- **baixa_pacient**: S'introdueix el nom d'un pacient. Si el pacient no existeix, es produeix un error. En cas contrari, el pacient es dona de baixa del sistema completament, per tant, també cal donar-lo de baixa de la llista d'espera i cancel·lar totes les visites que pogués tenir programades (si és el cas).
- **alta_doctor**¹: S'introdueix el nom del doctor/a, que farà d'identificador. Si ja existeix en el sistema, es produeix un error. En cas contrari, el doctor/a és donat d'alta en el sistema.
- **llista_espera**: No té paràmetres. S'escriuen les dades de tots els pacients que constin a la llista d'espera.
- **tractar_seguent_pacient**: No té paràmetres. Si no hi ha cap pacient a la llista d'espera, es produeix un error; sinó, s'elimina el primer pacient de la llista d'espera, però no es dona de baixa del sistema.
- **modificar_estat_pacient**: S'introdueix el nom d'un pacient i el nou nivell de gravetat. Si el pacient no existeix en el sistema o el nivell de gravetat no és vàlid, es produeix un error; sinó, s'actualitza el valor de gravetat del pacient en el sistema i es reassigna el pacient en la llista d'espera en funció del nou nivell de gravetat.
- **programar_visita**: S'introdueixen el nom d'un pacient, el nom d'un doctor i una data. Si el pacient o el doctor no existeixen en el sistema, es produeix un error; sinó, s'afegeix una nova visita al doctor amb el pacient i data indicats. Sempre serà possible

¹No es preveuen, per ara, baixes de doctors.



afegir una nova visita a qualsevol doctor del sistema, encara que ja tingui altres visites programades per la mateixa data.

- **cancellar_visita:** S'introdueixen el nom d'un pacient, el nom d'un doctor i una data. Si el pacient o el doctor no existeixen en el sistema, o bé la visita a cancel·lar no havia estat programada, es produeix un error; sinó, s'elimina la visita al doctor amb el pacient i data indicats.
- **mostrar_programacio_visites:** No té paràmetres. Per cada doctor que hi hagi en el sistema, s'escriu el seu nom i la llista de visites (data i nom pacient) que té programades **ordenada** per data. En el cas que hagi dues dates coincidents, s'escriuen segons l'ordre en què hagin estat introduïdes al sistema.
- **fi:** No té paràmetres. S'acaba l'execució de la simulació.

4 Es demana

Es demana que apliqueu el *disseny modular* per especificar i implementar un conjunt de classes i un programa principal que donin resposta a les operacions detallades en l'apartat anterior.

4.1 Classes

Per resoldre aquesta pràctica ja se us dona l'especificació (incompleta) d'algunes de les classes. Caldrà que completeu les seves especificacions afegint els mètodes que us calguin en funció de la implementació triada pel tipus. Després, implementeu els mètodes en els fitxers `.cpp`. D'altres classes, les haureu d'especificar i implementar completament.

El conjunt de classes és el següent:

- **Data:** Representa una data amb format dia, mes i any (ja donada).
- **Visita:** Representa una visita, que consta d'una data i un pacient (ja donada).
- **Pacient.hpp:** Representa el registre d'un pacient al servei d'urgències (especificació donada, caldrà implementar-la).
- **Doctor:** Ha de representar un doctor amb el seu conjunt de visites programades i oferir els mètodes necessaris per poder-hi operar (caldrà especificar-la i implementar-la).
- **Hospital:** Ha de representar tota la informació necessària per gestionar el servei d'urgències format pel conjunt de pacients, el conjunt de doctors i la llista d'espera, i oferir els mètodes necessaris per poder-hi operar (caldrà especificar-la i implementar-la).



També se us dona l'especificació i la implementació de dues classes que haureu de **completar**. Busqueu el text `PROGRAM THIS METHOD` en els fitxers `.cpp` corresponents.

- **PriorityQueue**: Especificació (`PriorityQueue.hpp`) i implementació (`PriorityQueue.cpp`) d'una cua amb prioritat. Partint de la classe `Queue` amb què heu treballat, s'ha modificat el mètode `push` per tal que es comporti com una cua amb prioritat de manera que els elements que s'afegeixin a la cua ho facin per ordre de “menor” a “major” del seu valor, en lloc de per ordre d'arribada. Heu d'implementar, en el fitxer corresponent, el mètode `remove(T value)` per tal que elimini de la cua l'element `value` donat.
- **BST**: Especificació (`BST.hpp`) i implementació (`BST.cpp`) d'un arbre binari de cerca (*Binary Search Tree*). **Heu de fer servir** aquesta classe per emmagatzemar els pacients del servei d'urgències, per tal que les cerques per pacient (nom) siguin eficients. Dins d'aquesta classe, heu d'afegir un mètode **recursiu** que us permeti fer aquesta cerca.

5 Normativa, avaluació i lliuraments

5.1 Normativa

1. La pràctica és **obligatòria**. La no presentació de la pràctica resultarà en un **NO PRESENTAT** com a nota final de l'assignatura.
2. S'ha de realitzar en grups de **dues persones**. Serà motiu de no acceptació del treball l'haver estat realitzat per més de dues persones. La formació de grups de pràctiques queda al vostre criteri.
3. La qualificació final de la pràctica té un pes del 25% sobre la nota total de l'assignatura. La nota de la pràctica està formada per dues parts: programació i prova de validació **individual** (PVI). La nota de la PVI serà un valor real entre 0 i 1. La fórmula per calcular la nota final de la pràctica és la següent:

$$\text{nota pràctica} = \text{nota programació} * \text{nota PVI}$$

La no compareixença d'un/a estudiant a la PVI farà que la seva nota de la pràctica sigui **NO PRESENTAT**, encara que hagi fet el lliurament del codi.

4. Si dos o més treballs presentats són considerats **iguals**, no serà acceptat **cap** dels treballs. En tractar-se d'una còpia, suposarà un 0 del treball i de l'assignatura.



5.2 Avaluació

Les pràctiques acceptades s'avaluaran mitjançant:

- La correctesa de l'especificació i/o comentaris del codi. Atenció especial amb:
 - les Pre/Post de les operacions més importants
 - les capçaleres dels mètodes pel que fa al bon ús de **const** (tant sobre mètodes com sobre paràmetres)
 - evitar l'ús de tipus “pesants” en el pas de paràmetres per valor o de retorn de funcions
- Els mètodes pertanyen a les classes adequades. Es fa una bona distinció entre mètodes privats i públics.
- L'execució del programa.

Una pràctica que no funcioni pot ser igualment acceptada i avaluada tot i que, si no passa cap joc de prova, la nota de la pràctica serà un 0.

5.3 Lliurament i PVI

1. El lliurament s'ha de fer a través d'Atenea.
2. El lliurament consistirà en un únic fitxer comprimit, amb nom **Pr-dni1-dni2.zip** format pels DNI dels membres del grup, ordenats de menor a major. Aquest fitxer contindrà **únicament** els fitxers de codi necessaris per compilar i executar la pràctica.
3. Se us dona un fitxer **Makefile** per facilitar la generació de l'executable així com per passar els jocs de proves. Aquest **Makefile** es farà servir per la correcció de la pràctica.
4. Se us facilitaran jocs de prova públics.

5.3.1 Dates

Data límit lliurament: diumenge 4 de juny de 2023 a les 23:59
--

Especificació i implementació de les classes, amb un programa principal que respongui a totes les funcionalitats demanades a l'enunciat. Se us donaran instruccions sobre com fer el lliurament així com jocs de prova públics que us serviran d'exemple.

Data PVI: dimarts 23 de maig de 2023 a les sessions de laboratori de cada grup

Preguntes individuals a cada membre del grup de pràctiques sobre el disseny i la implementació de la pràctica.