

Pràctica 2: El laboratori de l'IMC

Programació 2

Curs 2024-2025

Aquesta pràctica del curs consisteix en el desenvolupament d'un sistema de gestió d'analítiques. **Per resoldre aquesta pràctica pots utilitzar els elements de C++ vistos a les transparències del Tema 1, Tema 2 i Tema 3 de teoria.**

Condicions de lliurament

- La data límit de lliurament d'aquesta pràctica és el **divendres 28 de març**, fins a les **23:59**
- Has de lliurar un únic fitxer anomenat `prac2.cc` amb el codi de totes les funcions

Codi d'honor



Si es detecta còpia (total o parcial) a la teua pràctica, tindràs un **0** en el lliurament i s'informarà a la direcció de l'EPS perquè adopte mesures disciplinàries



És correcte discutir amb els teus companys possibles solucions a les pràctiques
És correcte apuntar-te a una acadèmia si serveix per obligar-te a estudiar i fer pràctiques



És incorrecte copiar codi d'altres companys o demanar a ChatGPT que et faci la pràctica
És incorrecte apuntar-te a una acadèmia perquè et facen les pràctiques



Si necessites ajuda, dirigeix-te al teu professor/a
No copies

Normes generals

- Has de lliurar la pràctica exclusivament a través del servidor de pràctiques del Departament de Llenguatges i Sistemes Informàtics (DLSI). S'hi pot accedir de dues maneres:
 - Pàgina principal del DLSI (<https://www.dlsi.ua.es>), opció "LLIURAMENT DE PRÀCTIQUES"
 - Directament a l'adreça <https://pracdlsi.dlsi.ua.es>
- Qüestions que has de tenir en compte en fer el lliurament:
 - L'usuari i la contrasenya per lliurar pràctiques són els mateixos que utilitzes a UACloud
 - Pots lliurar la pràctica diverses vegades, però només es corregirà l'últim lliurament
 - No s'acceptaran lliuraments per altres mitjans, com el correu electrònic o UACloud
 - No s'acceptaran lliuraments fora de termini
- La teua pràctica ha de poder ser compilada sense errors amb el compilador de C++ existent a la distribució de Linux dels laboratoris de pràctiques

- Si la teua pràctica no es pot compilar, la seua qualificació serà 0
- Al començament de tots els fitxers font lliurats has d'incloure un comentari amb el teu NIF (o equivalent) i el teu nom. Per exemple:

```

prac2.cc

// DNI 12345678X GARCIA GARCIA , JUAN MANUEL
...

```

- Superar **totes** les proves de l'autocorrector et dóna dret a presentar-te a l'examen de pràctiques. Si falla alguna prova no podràs presentar-te a l'examen i la teua qualificació en aquesta pràctica serà 0
- El càlcul de la nota de la pràctica i la seua rellevància en la nota final de l'assignatura es detallen a les transparències de presentació de l'assignatura (*Tema 0*)

1 Descripció de la pràctica

Es vol desenvolupar un sistema de gestió dels informes d'analítiques (en aquest cas de l'índex de massa corporal) d'un laboratori. En el sistema es donaran d'alta els pacients per una banda i els resultats de cadascuna de les anàlisis que es realitzen. El sistema permetrà fer un seguiment de les dades introduïdes i detectarà situacions de risc.

2 Detalls d'implementació

Al Moodle de l'assignatura es publicaran diversos fitxers que necessitaràs per a la correcta realització de la pràctica:

- `prac2.cc`. Aquest fitxer conté un esquelet de programa sobre el qual realitzar la teua pràctica. Descarrega'l i afegeix el teu codi en ell. Aquest fitxer conté la següent informació:
 - Els registres (`struct`) necessaris per a fer la pràctica
 - Un tipus enumerat `Error` que conté totes les possibles classes d'error que es poden donar en aquesta pràctica (per exemple, `ERR_OPTION`)
 - Una funció `error` que s'encarrega de mostrar el corresponent missatge d'error per pantalla en funció del paràmetre que se li passe. Per exemple, quan la funció rep el paràmetre `ERR_OPTION`, mostrarà per pantalla el missatge `ERROR: wrong option`
 - Una funció `showMenu` que mostra per pantalla el menú del programa
 - Una funció `main` que implementa la gestió del menú principal i crida a les funcions corresponents segons l'opció triada per l'usuari
- `autocorrector-prac2.tgz`. Conté els fitxers de l'autocorrector per avaluar la pràctica amb diferents proves d'entrada. La correcció automàtica de la pràctica, després del lliurament, es realitzarà amb aquestes mateixes proves i serà necessari superar-les totes per poder presentar-te a l'examen d'aquesta pràctica
- `prac2`. Fitxer executable de la pràctica (compilat per a màquines Linux de 64 bits) desenvolupat pel professorat de l'assignatura, perquè el pugues provar amb les entrades que vulgues i veure la sortida correcta esperada.

3 Funcionament del programa

El programa que desenvoluparàs consisteix en un sistema d'emmagatzematge i gestió de les analítiques que es realitzen en un laboratori.

4 Components

El programa a desenvolupar ha de poder gestionar tres elements fonamentals: els pacients, les diferents analítiques que es realitzen a cadascun dels pacients i la base de dades, que és on s'emmagatzema tota la informació dels pacients i de les analítiques. Els següents apartats descriuen l'estructura de cadascun d'aquests components amb més detall.

4.1 Pacient

Les estructures de tipus `Patient` emmagatzemen la informació relativa a un pacient del laboratori. Cada pacient estarà identificat pel seu NIF (`nif`), el seu nom (`name`) i el seu telèfon (`telephone`):

```
struct Patient{
    string nif;
    string name;
    string telephone;
};
```

També utilitzarem una estructura específica de pacient per emmagatzemar els pacients en el fitxer binari (`PatientBin`):

```
struct PatientBin{
    char nif[KMAXNIF];
    char name[KMAXNAME];
    char telephone[KMAXTELEPHONE];
};
```

Les constants `KMAXNIF`, `KMAXNAME` i `KMAXTELEPHONE` tenen el format que s'indica a continuació:

```
const int KMAXNIF=10;
const int KMAXNAME=50;
const int KMAXTELEPHONE=14;
```

4.2 Analítica

Les estructures de tipus `Analysis` emmagatzemen informació referent a les dades d'una analítica realitzada a un pacient. Cada analítica estarà identificada per un número i contindrà un identificador únic (`id`), el NIF del pacient (`nif`), la data de l'analítica (`dateAnalysis`), el pes (`weight`) i l'alçada (`height`):

```
struct Analysis{
    unsigned int id;
    char nif[KMAXNIF];
    Date dateAnalysis;
    float weight;
    float height;
};
```

El registre `Date` es defineix de la manera següent:

```
struct Date{
    unsigned int day;
    unsigned int month;
    unsigned int year;
};
```

4.3 Base de dades

L'estructura Database emmagatzema tota la col·lecció de pacients (patients) i analítiques realitzades (analysis). Conté, a més, un camp nextId que emmagatzemarà l'identificador que s'haurà d'assignar al camp id de la següent analítica que es cree. Aquest camp tindrà el valor 1 a l'inici del programa i s'incrementarà d'un en un per a cada nova analítica. És a dir, la primera analítica que es cree tindrà com a identificador 1, la segona tindrà el 2 i així successivament.

```
struct Database{
    unsigned int nextId;
    vector <Patient> patients;
    vector <Analysis> analysis;
};
```

5 Inici del programa

El main de la teua pràctica contindrà una variable de tipus Database que emmagatzemarà en memòria tota la informació dels pacients i analítiques durant l'execució del programa. Quan s'iniciï el programa, el camp nextId d'aquesta variable s'inicialitzarà a 1.

El primer que hauràs de fer és carregar en memòria les dades dels pacients emmagatzemades en el fitxer binari patients.bin. Aquest fitxer conté estructures de tipus PatientBin, una darrere l'altra, de manera que cadascuna emmagatzema la informació d'un pacient. Hauràs de crear una funció loadPatients que s'encarregue de llegir tota la informació dels pacients i carregar-la en el vector patients de la variable de tipus Database esmentada anteriorment.



- El teu programa només gestionarà una única variable de tipus Database que contindrà totes les dades de pacients i anàlisis.
- Si en iniciar el programa el fitxer patients.bin no existeix, no has de mostrar cap tipus d'error. El programa s'iniciarà amb normalitat deixant el vector patients de Database buit.
- Assumirem que el contingut del fitxer patients.bin és correcte. No cal fer cap tipus de comprovació sobre el contingut durant la càrrega.

6 Menú

En executar la pràctica, i després de carregar la informació dels pacients (si n'hi ha), es mostrarà per pantalla el menú principal del programa, quedant a l'espera que l'usuari trie una opció:

Terminal

```
1- Add patient
2- View patient
3- Delete patient
4- Save patients
5- Add analysis
6- Export analysis
7- Import analysis
8- Statistics
q- Quit
Option:
```

Les opcions vàlides que pot introduir l'usuari són els números del 1 al 8 i la lletra q. Si l'opció triada no és cap d'aquestes (per exemple un 9, una x o un ;), s'emetrà l'error ERR_OPTION cridant la funció error amb aquest paràmetre. Quan l'usuari trie una opció correcta, s'ha d'executar el codi associat a aquesta opció. En finalitzar-la, es tornarà a mostrar el menú principal i es demanarà una altra opció, fins que l'usuari decidisca eixir del programa utilitzant l'opció q.

7 Opcions

En els següents apartats es descriu el funcionament que haurà de tenir cadascuna de les opcions que es mostren en el menú principal del programa.

7.1 Add patient

Aquesta opció permet donar d'alta els pacients dels quals podrem introduir anàlisis a l'aplicació. El seu codi s'ha d'incloure en una funció anomenada `addPatient`. La informació que es sol·licitarà en primer lloc és el NIF del pacient a donar d'alta:

```
Terminal
Enter NIF:
```

Si s'introdueix un NIF en blanc, s'haurà de tornar al menú principal. S'ha de garantir que es introdueixen els nou caràcters del NIF, compost per vuit números i una lletra. En cas contrari, es mostrarà l'error `ERR_WRONG_NIF` i es tornarà a sol·licitar.

Es comprovarà amb la funció `searchPatient` si aquest existeix. Aquesta funció retornarà `-1` si el pacient no existeix. En cas contrari, retornarà la posició del pacient dins del vector `patients`. Si ja existís un pacient amb aquest NIF, es mostrarà el missatge d'error `ERR_PATIENT_EXISTS` i el programa tornarà a sol·licitar el NIF. Si no existeix, demanarà la resta d'informació i afegirà un nou registre de pacient. En primer lloc, demanarà el nom:

```
Terminal
Enter name:
```

S'ha de comprovar que en el camp nom s'introdueixen almenys tres caràcters. En cas contrari, es mostrarà el missatge `ERR_WRONG_NAME` i es tornarà a sol·licitar. A continuació, es demanarà el telèfon:

```
Terminal
Enter telephone:
```

S'ha de comprovar que el número de telèfon comença pel símbol `+` (per al prefix telefònic) i va seguit per entre 10 i 12 números. Si no fora així, es mostrarà el missatge d'error `ERR_WRONG_TELEPHONE` i es tornarà a demanar el número de telèfon.



- Per saber si un caràcter és un número, pots utilitzar la funció `isdigit()` de la llibreria `cctype`.
- Per saber si un caràcter és una lletra, pots utilitzar la funció `isalpha()` de la llibreria `cctype`.

7.2 View patient

Aquesta opció permet visualitzar la informació d'un pacient. S'ha d'implementar una funció anomenada `viewPatient`. En primer lloc, es sol·licitarà el NIF amb el missatge:

```
Terminal
Enter NIF:
```

En cas que no hi haja cap pacient amb aquest NIF (s'ha d'utilitzar la funció `searchPatient` per localitzar-lo), es mostrarà el missatge d'error `ERR_PATIENT_NOT_EXISTS` i el programa tornarà a preguntar el NIF. Si s'introdueix la cadena buida, es tornarà al menú principal sense fer res.

Si el pacient existeix a la base de dades, es visualitzaran les seues dades i els resultats de totes les analítiques que se li hagen realitzat, una en cada línia, seguint aquest format:

Terminal

```
NIF: 11111111D
Name: Pedro Pi
Telephone: +34612345689
Id   Date       Height Weight
23   11/11/2027  188    95
45   17/02/2028  188    93
```

Les columnes `Id`, `Date`, `Height` i `Weight` estan separades per tabuladors (caràcter `'\t'`). Si no hi ha analítiques per a un pacient, es mostrarà només la seua informació, sense posar `Id`, `Date`, etc. Si el número que representa el mes o l'any és inferior a 10, s'haurà d'incloure un 0 davant perquè el format complisca amb dos dígitos per al dia i dos per al mes. Per exemple: 05/07/2026.

7.3 Delete patient

Aquesta opció permet esborrar la informació d'un pacient. S'ha d'implementar una funció anomenada `deletePatient`. En primer lloc, es sol·licitarà el NIF amb el següent missatge:

Terminal

```
Enter NIF:
```

Si s'introdueix un NIF en blanc, s'haurà de tornar al menú principal. En el cas que no es tinga informació d'un pacient amb aquest NIF, es mostrarà l'error `ERR_PATIENT_NOT_EXISTS` i es tornarà a sol·licitar el NIF.

Si el pacient existeix, s'eliminarà del vector `patients` de la base de dades i s'eliminaran totes les analítiques d'aquest pacient emmagatzemades en el vector `analysis`.

7.4 Save patients

Aquesta opció guarda tots els pacients de la base de dades en el fitxer `patients.bin`. Si ja existeix un fitxer amb aquest nom, el sobreescrirà amb la informació actual de la base de dades. S'ha d'implementar una funció `savePatients` per fer-ho.

Per guardar la informació d'un pacient (`Patient`) en el fitxer binari, primer s'ha de transformar en una estructura de tipus (`PatientBin`). Això implica que potser hauràs de retallar el nom del pacient perquè no supere la mida màxima `KMAXNAME`.

7.5 Add analysis

Aquesta opció permet afegir els resultats d'una analítica per a un pacient. S'implementarà en una funció `addAnalysis`. En primer lloc, es sol·licitarà el NIF del pacient:

Terminal

```
Enter NIF:
```

Si s'introdueix un NIF en blanc, s'haurà de tornar al menú principal. Si no existís un pacient amb aquest NIF, es mostrarà l'error `ERR_PATIENT_NOT_EXISTS` i es tornarà a sol·licitar. Si existeix, es demanarà la data de l'analítica amb el següent missatge:

Terminal

```
Enter date (day/month/year):
```

L'usuari haurà d'introduir la data amb el format indicat en el missatge. Per exemple: 12/5/2027. Assumim que l'usuari sempre introdueix el format correcte. No fa falta comprovar-ho.

El que sí que s'ha de comprovar és que la data és vàlida. Per a això, el dia haurà d'estar comprès entre els valors 1 i 31, el mes entre 1 i 12, i l'any entre 2025 i 2050, inclosos. No és necessari comprovar que el número de dies és correcte per a un mes concret. Assumim que tots els mesos poden tenir 31 dies (per exemple, el 30 de febrer seria correcte). Aquesta informació s'haurà d'emmagatzemar als camps `day`, `month` i `year` de `dateAnalysis`. En cas d'error, es mostrarà el missatge `ERR_WRONG_DATE` i es tornarà a demanar que s'introdueixi la data. A continuació, es demanarà el pes (en quilograms) amb el següent missatge:

Terminal

Enter weight:

S'haurà de verificar que el camp pes siga positiu. En cas contrari, es llançarà l'error `ERR_WRONG_NUMBER` i es tornarà a sol·licitar. Finalment, es demanarà l'alçada (en centímetres):

Terminal

Enter height:

Igualment, s'haurà de verificar que els camp alçada siga positiu. En cas contrari, es llançarà l'error `ERR_WRONG_NUMBER` i es tornarà a sol·licitar.

Finalment, s'assignarà al camp `id` el valor de `nextId` de la base de dades i s'incrementarà aquest valor per a la següent analítica.

7.6 Export analysis

Emmagatzema en un fitxer anomenat `analysis.bin` tots els registres d'analítiques emmagatzemats a la base de dades. S'ha d'implementar una funció `exportAnalysis` per a això. En cas que el fitxer ja existís, s'haurà de sobreesciure.

7.7 Import analysis

Llegirà la informació del fitxer `analysis.bin`. S'ha d'implementar en una funció `importAnalysis`. Si el fitxer no existís, s'haurà de mostrar el missatge d'error `ERR_FILE_NOT_EXISTS` i es tornarà a la funció principal sense fer res més.

Per a cadascuna de les analítiques emmagatzemades en el fitxer, s'haurà de comprovar prèviament que existeix un pacient amb aquest NIF en el vector `patients` de la base de dades. En cas contrari, s'escriurà el NIF d'aquest pacient en el fitxer de text `wrong_patients.txt` (un per línia) i no es carregarà aquesta analítica en el sistema. Aquest fitxer no es sobreesciurà en cas que ja existís, sinó que s'afegirà la informació a continuació. Si el fitxer `wrong_patients.txt` no es pot obrir, es mostrarà també l'error `ERR_FILE_NOT_EXISTS` i es tornarà al menú principal sense fer res més.

Per a cada analítica que es carregue en el sistema, se li haurà d'assignar el `id` que li corresponga com si s'estigués inserint des de l'opció `Add analysis`, segons el valor emmagatzemat a la variable `Database`. És a dir, s'ignora el `id` que tenia l'analítica quan es va emmagatzemar en el fitxer binari. Les analítiques que puguen existir abans de carregar el fitxer no s'esborraran.

7.8 Statistics

Aquesta opció recorrerà el fitxer d'analítiques i mostrarà el grau d'obesitat per a cadascuna de les analítiques dels pacients. Un pacient pot tenir múltiples analítiques, així que es mostraran totes elles. S'implementarà en una funció anomenada `statistics`. Una mesura de l'obesitat es determina mitjançant l'índex de massa corporal (IMC), que es calcula amb la següent fórmula:

$$IMC = \frac{\text{pes (kg)}}{[\text{alçada (m)}]^2}$$

Per exemple, una persona d'1,85 metres d'alçada i 94 quilograms de pes tindria un IMC de $94/1,85^2 = 27,5$. Segons l'Institut Nacional del Cor, els Pulmons i la Sang dels Estats Units (NHLBI), el sobrepès es defineix com un IMC de 25 o més. Es considera que una persona és obesa si el seu IMC és superior o igual a 30. La següent taula mostra els diferents tipus de composició corporal i el seu corresponent IMC:

Composició corporal	Índex de massa corporal (IMC)
Underweight	Menys de 18.5
Healthy	18.5 – 24.9
Overweight	25.0 – 29.9
Obesity	Igual o major de 30.0

De cada analítica es mostrarà el NIF del pacient, la data de l'analítica (en format dd/mm/yyyy), el pes, l'alçada i la composició corporal (Underweight, Healthy, Overweight o Obesity). Tots aquests camps es mostraran separats per un ' ; ':

```
Terminal
11111111D;02/05/2025;110;180;Obesity
11111111D;04/07/2025;115;180;Obesity
22222222G;13/11/2025;75;186;Healthy
```

Només es mostrarà la informació dels pacients que tinguen almenys una analítica. La mateixa informació que es mostra en pantalla i amb aquest mateix format (separant cada camp amb ' ; '), es guardarà en un fitxer de text anomenat `statistics.txt`. Si el fitxer ja existís, es sobreescriurà.



- En mostrar la informació per pantalla i desar-la en un fitxer, si el número que representa el mes o l'any és inferior a 10, s'haurà d'incloure un 0 davant perquè el format complisca amb dos dígits per al dia i dos per al mes

8 Arguments del programa

En aquesta pràctica, el programa ha de permetre a l'usuari indicar algunes accions a realitzar mitjançant el pas d'arguments per línia de comandament. Hauràs d'utilitzar els paràmetres `int argc` i `char *argv[]` de la funció `main` per poder gestionar-los. El programa admetrà els següents arguments per línia de comandament:

- `-f <fitxer de text>`. Permet importar analítiques d'un fitxer de text, el nom del qual s'haurà d'indicar després de l'argument `-f`, iniciant el programa a continuació. El format del fitxer de text haurà de ser el mateix que el que es genera amb l'opció `Statistics` (no cal comprovar-ho). S'haurà d'ignorar l'IMC de l'analítica llegida del fitxer i guardar tota la resta d'informació en registres de tipus `Analysis` en el vector `analysis` de la base de dades. Per a cada analítica que es carregue en el sistema, se li haurà d'assignar el id que li corresponga com si s'estigués inserint des de l'opció `Add analysis`. Igual que succeïa en l'opció `Import analysis` del menú, si un NIF no existeix a la base de dades, s'escriurà el NIF d'aquest pacient en el fitxer `wrong_patients.txt`.

Si el fitxer passat com a argument no existeix o el fitxer `wrong_patients.txt` no es pot obrir, s'emetrà l'error `ERR_FILE_NOT_EXISTS` i s'iniciarà el programa amb normalitat sense carregar cap analítica. Tingues en compte que abans de gestionar el fitxer d'analítiques, el programa carregarà en memòria la informació de `patients.bin` perquè existeixen pacients als quals associar les analítiques.

El següent exemple importarà les dades emmagatzemades en el fitxer de text `blood_tests.txt`:

```
Terminal
$ prac2 -f blood_tests.txt
```

- `-s`. Mostra les estadístiques, igual que si l'usuari seleccionés l'opció `Statistics` del menú, eixint del programa a continuació sense arribar a mostrar el menú principal. Aquesta opció només té sentit si l'usuari ha introduït també l'argument `-f`, ja que si no, no existiria cap informació per mostrar. Per aquesta raó, la següent crida seria incorrecta:

```
Terminal
$ prac2 -s
```

Els dos arguments poden aparèixer al mateix temps en una crida al programa i també en qualsevol ordre. Independentment de l'ordre en què apareguen a la línia de comandament, l'ordre en què es processaran els arguments serà el següent:

1. En primer lloc, es processarà l'opció `-f` per carregar dades del fitxer
2. En segon lloc, s'executarà l'opció `-s` per mostrar les estadístiques

En el següent exemple, primer es carregaran les dades del programa emmagatzemades en el fitxer `blood_tests.txt` i, a continuació, es mostraran les estadístiques, eixint després del programa:

```
Terminal
$ prac2 -s -f blood_tests.txt
```

Un altre exemple, els paràmetres del qual són vàlids, seria el següent:

```
Terminal
$ prac2 -f -f -s
```

Encara que a primera vista els paràmetres puguin semblar incorrectes, en realitat no ho són. El programa rep un primer paràmetre `-f` que va seguit d'un fitxer que es diu `-f`. A continuació, rep el paràmetre `-s` per mostrar les estadístiques. Aquesta seqüència de paràmetres és, per tant, correcta.

En cas que es produísca un error en els arguments d'entrada (per exemple, que s'introduísca una opció que no existeix o es repeteix una opció), s'haurà d'emetre l'error `ERR_ARGS` i es finalitzarà el programa sense arribar a mostrar el menú principal. Si tots els arguments són correctes, es processaran les opcions introduïdes per l'usuari i després es mostrarà el menú principal del programa de la forma habitual (sempre que no s'haja passat l'opció `-s`, que mostra la informació i finalitza el programa).



- Recorda, independentment de si es passen arguments al programa o no, sempre s'haurà de carregar primer el fitxer `patients.bin` a l'inici del programa
- Els dos arguments són opcionals i poden no aparèixer en la crida al programa
- Recorda, si està present el paràmetre `-s`, obligatòriament ha d'estar `-f`. Si aparegués sol, s'haurà d'emetre l'error `ERR_ARGS`
- Un determinat argument només pot aparèixer una vegada en la crida. En cas contrari es considerarà que els arguments són incorrectes
- Si apareix algun argument diferent dels esmentats (per exemple, `-n`), es considerarà que els arguments són incorrectes
- Si darrere de `-f` no apareix el nom del fitxer, els arguments seran incorrectes també
- Els fitxers de text no han de tenir necessàriament l'extensió `.txt`, igual que els fitxers binaris no han de tenir necessàriament l'extensió `.bin`. Utilitzem aquestes extensions en els exemples per tal que quede més clar que estem treballant amb fitxers de text i binaris