

# TÉCNICAS DE PROGRAMACIÓN

## SEGUNDA PRÁCTICA CALIFICADA

### SEMESTRE ACADÉMICO 2022-2

Horario: Todos

Duración: 110 minutos  
Elaborado por los profesores del curso.

#### ADVERTENCIAS:

- Todo dispositivo electrónico (teléfono, tableta, computadora u otro) deberá permanecer apagado durante la evaluación en su mochila.
- Coloque todo aquello que no sean útiles de uso autorizado durante la evaluación en la parte delantera del aula, por ejemplo, mochila, maletín, cartera o similar, y procure que contenga todas sus propiedades. La apropiada identificación de las pertenencias es su responsabilidad.
- Si se detecta omisión a los dos puntos anteriores, la evaluación será considerada nula y podrá conllevar el inicio de un procedimiento disciplinario en determinados casos.
- Es su responsabilidad tomar las precauciones necesarias para no requerir la utilización de servicios higiénicos: durante la evaluación, no podrá acceder a ellos, de tener alguna emergencia comunicárselo a su jefe de práctica.
- En caso de que el tipo de evaluación permita el uso de calculadoras, estas no podrán ser programables.
- Quienes deseen retirarse del aula y dar por concluida su evaluación no lo podrán hacer dentro de la primera mitad del tiempo de duración destinado a ella.

#### INDICACIONES:

- No se pueden usar apuntes de clase ni calculadoras.
- LAS SOLUCIONES DEBERÁN DESARROLLARSE BAJO UN ESTRICTO DISEÑO DESCENDENTE, por lo que NO SE CALIFICARÁN aquellos módulos que son llamados por otros que estén incompletos. Cada módulo no debe sobrepasar las 30 líneas de código aproximadamente.
- NO SE PUEDEN EMPLEAR ARCHIVOS DE DATOS AUXILIARES NI VARIABLES GLOBALES. NO podrá implementar funciones en el archivo main.cpp, las funciones se deberán implementar en archivos independientes (.h/.hpp y .cpp).
- No se calificará el código puesto como comentario.
- En la calificación se tomará en cuenta el buen uso de los nombres de los identificadores, y el eficaz uso de comentarios.

La finalidad de esta práctica es la de reforzar los conceptos del capítulo 6: “Aplicaciones con arreglos y punteros”.

#### DEBE LEER TODA LA PRUEBA ANTES DE EMPEZAR A DESARROLLAR EL PROGRAMA

(20 puntos) Una clínica especializada en enfermedades psicológicas requiere contar con un reporte para controlar las citas registradas de sus pacientes. Para realizar esta labor cuenta con tres archivos de textos como los que se muestran a continuación:

*medicos.txt*

552775	A	374.50
675432	B	563.33
...	...	...

En cada línea se muestran los datos de cada médico de la empresa. Primero aparece el código del médico, después aparece la categoría de descuento que aplica dicho médico a aquellos pacientes que poseen un seguro de la institución (A el 10%, B el 5%, C el 2%, otro carácter no aplica descuento), finalmente viene la tarifa que cobra el médico por hora de consulta. Se sabe que se tiene un máximo de 50 médicos registrados.

*pacientes.txt*

56401189	772346621	V
64572344	809500124	F
...	...	...

En cada línea se muestran los datos de los pacientes registrados en la empresa. Primero aparece el DNI del paciente seguido de su teléfono, al final un caracter que indica si el paciente está asegurado o no. Aproximadamente la clínica atiende a 200 pacientes.

*citas.txt*

64572344	675432	14/9/2022	11:34:10	13:22:35
11223344	235468	12/8/2022	9:00:25	9:45:00
...	...	...	...	...

En cada línea se muestran los datos de las citas que han tenido los diferentes pacientes. Primero aparece el DNI del paciente seguido del código del médico que lo atendió, la fecha de la cita y la hora en la que entró y



se retiró el paciente de la consulta. Un paciente puede haber tenido varias citas con el mismo o diferentes médicos. El descuento del médico solo se aplica a aquellos pacientes asegurados. En esta base de datos se cuenta con un máximo de 500 citas registradas.

El programa deberá elaborar un reporte en el que aparezca cada paciente con las respectivas citas que ha tenido. El reporte será similar al siguiente:

#### Reporte.txt

CLINICA PSICOLOGICA TP.  
RELACION DE CITAS POR PACIENTE

DNI: 56401189  
Telefono: 772346621

Seguro: EL PACIENTE ESTA ASEGURADO

CITAS REALIZADAS:

FECHA	ID MEDICO	DESCUENTO	TARIFA	LLEGADA	SALIDA	TIEMPO (hrs.)	PAGO	DESCUENTO
02/03/2022	552775	10%	374.50	09:14:33	11:05:23	1.8472	622.60	69.17
15/05/2022	345215	0%	256.17	14:33:15	15:14:06	0.6808	174.40	0.00
...	...	...	...	...	...	...	...	...
Total de gastos por citas:		6785.56						
Total de descuentos por citas:		752.34						

DNI: 64572344  
Telefono: 809500124

Seguro: EL PACIENTE NO ESTA ASEGURADO

CITAS REALIZADAS:

FECHA	ID MEDICO	DESCUENTO	TARIFA	LLEGADA	SALIDA	TIEMPO (hrs.)	PAGO	DESCUENTO
02/03/2022	675432	0%	563.33	10:00:00	11:00:00	1.0000	363.33	0.00
...	...	...	...	...	...	...	7563.33	...
Total de gastos por citas:		...						
Total de descuentos por citas:		...						

TOTAL RECAUDADO POR LA CLINICA: 967546.89

Contidad de Pacientes que recibieron descuentos de tipo A: 175

Contidad de Pacientes que recibieron descuentos de tipo B: 55

Contidad de Pacientes que recibieron descuentos de tipo C: 20

#### CONSIDERACIONES IMPORTANTES:

- El pago por cita debe considerar las horas y fracción.
- En el reporte, al pago del cliente se le ha realizado el descuento.
- En el reporte, el total de gastos por citas es la suma de los pagos de las consultas del cliente.
- En el reporte, el total recaudado por la clínica no incluye los descuentos.

**PARA DESARROLLAR EL REPORTE, LOS ARCHIVOS MEDICOS.TXT, PACIENTES.TXT Y CITAS.TXT SOLO SE PUEDEN LEER UNA SOLA VEZ.**

Coloque en arreglos los datos de los archivos utilizados que crea conveniente para mostrar en el reporte.

San Miguel, 26 de septiembre del 2022



Año				Número			
2	0	2	0	1	2	1	6

Código de alumno

## Práctica

Araoz Andrade, Ivan Alexander  
Apellidos y nombres del alumno (letra de imprenta)

Ivan Araz  
Firma del alumno

Curso: TP

Práctica N°: 2

Horario de práctica: P-434

Fecha: 26/09/22

Nombre del profesor: E. Gómez

Nota
16

C.H.  
Firma del jefe de práctica  
Nombre y apellido: C.H.  
(iniciales)

## INDICACIONES

1. Llene todos los datos que se solicitan en la carátula, tanto los personales como los del curso.
2. Utilice las zonas señaladas del cuadernillo para presentar su trabajo en limpio. Queda terminantemente prohibido el uso de hojas sueltas.
3. Presente su trabajo final con la mayor claridad posible. No desglose ninguna hoja de este cuadernillo. Indique de una manera adecuada si desea que no se tome en cuenta alguna parte de su desarrollo.
4. Presente su trabajo final con la mayor pulcritud posible. Esto incluye lo siguiente:
  - cuidar el orden, la redacción, la claridad de expresión, la corrección gramatical, la ortografía y la puntuación en su desarrollo;
  - escribir con letra legible, dejando márgenes y espacios que permitan una lectura fácil;
  - evitar borrones, manchas o roturas;
  - no usar corrector líquido;
  - realizar los dibujos, gráficos o cuadros requeridos con la mayor exactitud y definición posibles.
5. No seguir estas indicaciones influirá negativamente en su calificación.
6. Al recibir esta práctica calificada, tome nota de las sugerencias que se le dan en la contracarátula del cuadernillo.



Zona exclusiva para  
cálculos y desarrollos  
(borrador)

## Presente aquí su trabajo

4  
18.  
13/9/5

```
using namespace std;
#include "funciones.hpp"
#define MAXMEDICOS 50
#define MAXPACIENTES 200
#define MAXCITAS 500
#define TAMLINEA 135
int main() {

    int arrCodMedico[MAXMEDICOS], arrDniPaciente,
    arrTelefPaciente[MAXPACIENTES], arrDniCitas[MAXPACIENTES],
    arrCodMedCitas[MAXCITAS], arrDdCitas[MAXCITAS],
    arrMmCitas[MAXCITAS], arrYyyCitas[MAXCITAS],
    arrHhIcitas[MAXCITAS], arrHhFcitas[MAXCITAS],
    arrSsIcitas[MAXCITAS], arrHhFcitas[MAXCITAS],
    arrMinFcitas[MAXCITAS], arrSsFcitas[MAXCITAS];
    int nMedicos = nCitas = nPacientes = 0;
    double arrTarifaMedico[MAXMEDICOS];
    char arrDescMedico[MAXMEDICOS];
    bool arrSeguro[MAXPACIENTES];

    ofstream archReporte("Reporte.txt", ios::out);
    if (not archReporte.is_open()) {
        cout << "Error: El archivo Reporte.txt no pudo abrirse";
        exit(1);
    }
    // tambien puede ir en imprimirReporte

    // Lectura de archivos
    leerMedicos(arrCodMedico, arrTarifaMedico, arrDescMedico);
    leerPacientes(arrDniPaciente, arrTelefPaciente, arrSeguro);
    leerCitas(arrDniCitas, arrCodMedCitas, arrDdCitas,
    arrMmCitas, arrYyyCitas, arrHhIcitas, arrMinIcitas,
    arrSsIcitas, arrHhFcitas, arrMinFcitas, arrSsFcitas,
    nCitas);

    // Imprimir Reporte
    imprimirReporte(// todos los arr mencionados, nMedicos,
    nCitas, nPacientes, archReporte);
    return 0;
}
```



# Presente aquí su trabajo

ACU

```
#ifndef FUNCIONES.HPP
#define FUNCIONES.HPP
```

```
void leerMedicos(int *, double *, char *, int &);
void leerPacientes(int *, int *, bool *, int &);
void leerCitas(//11 int *, int &);
void imprimirReporte(//14 int *, int, int, int, double *,
    bool *, char *, ofstream &);
void imprimirCaberera(int *, int *, bool *, int,
    ofstream &);
void imprimirLinea(char, int, ofstream &);
void imprimirCitas(//14 todos los arreglos x contadores,
    double &, ofstream &);
void imprimirResumen(ofstream &, double, char *);
void buscarMedico(int, int *, double *, char *,
    int, double &, double &);
void imprimirResumen2(ofstream &, double, double);
```

Zona exclusiva para  
cálculos y desarrollo  
(borrador)

```
using namespace std;
#include <iomanip>
#include <iostream>
#include <fstream>
#endif
```

ESTO VA  
ARRIBA



Zona exclusiva para  
cálculos y desarrollos  
(borrador)

## Presente aquí su trabajo

```
#include "Funciones.hpp"
// como es en papel el diseño descendente no es perfecto
void leerMedicos(int *arrCodMedico, double
*arrTarifaMedico, char *arrDescMedico, int
&nMedicos){
    ifstream archMedicos("medicos.txt", ios::in);
    if (not archMedicos.is_open()) {
        cout << "ERROR: archMedicos" << endl;
        exit(1);
    }

    while (true) {
        archMedicos >> arrCodMedico[nMedicos];
        if (archMedicos.eof()) break;
        archMedicos >> arrDescMedico[nMedicos] >>
        arrTarifaMedico[nMedicos];
        nMedicos++;
    }
}

void leerPacientes(int *arrDniPaciente, int *arrTelefPaciente,
bool *arrSeguro, int &nPacientes) {
    ifstream archPacientes("pacientes.txt", ios::in);
    if (not archPacientes.is_open()) {
        cout << "ERROR: pacientes.txt" << endl;
        exit(1);
    }

    char car;
    while (true) {
        archPacientes >> arrDniPaciente[nPacientes];
        if (archPacientes.eof()) break;
        archPacientes >> arrTelefPaciente[nPacientes] >> car;

        if (car == 'V') {
            arrSeguro[nPacientes] = true;
        }
        else {
            arrSeguro[nPacientes] = false;
        }
        nPacientes++;
    }
}
```



# Presente aquí su trabajo

void leerCitas (// todos los arr definidos en main con \*) { su tipo de dato

ifstream archCitas ("citas.txt", ios::in);

if (not archCitas.is\_open()) {  
cout << "ERROR: citas.txt" << endl;  
exit(1);  
}

char slash, extra;

while (true) {

archCitas >> arrDniCitas[nCitas];

if (archCitas.eof()) break;

archCitas >> arrCodMedCitas[nCitas] >> arrDdCitas[nCitas]

>> extra >> arrMmCitas[nCitas] >> slash >> arrYmCitas[nCitas]

>> arrHhI[nCitas] >> extra >> arrMinI[nCitas]

>> extra >> arrSsICitas[nCitas] >> arrHhFCitas[nCitas]

>> extra >> arrMinFCitas[nCitas] >> extra >> arrSsFCitas[nCitas];

nCitas++;

}

void imprimirReporte (// todos los arreglos y contadores, ofstream &archReporte) {

archReporte << setw(56) << " " << "CLINICA PSICOLOGICA TP." <<

endl << setw(52) << " " << "RELACION DE CHAS

POR PACIENTE" << endl;

for (int i = 0; i <= nPacientes; i++) {

imprimirCabecera (arrDniPaciente, arrTelefonoPaciente,  
arrSeguro, i, archReporte);

imprimirCitas (arrDniPaciente[i], arrDdCitas, ... // el  
resto de arreglos y contadores, totalRecaudado,  
archReporte);

double totalRecaudado = 0.0; // debería ir arriba

imprimirResumen (archReporte, totalRecaudado,  
arrDescMedico);

→ no implementado

void imprimirLinea (char c, int tam, ofstream &archReporte) {

for (int i = 0; i < tam; i++) {  
archReporte.put(c);

archReporte.put('\n');

Zona exclusiva para  
cálculos y desarrollos  
(borrador)

130 + AM

130-	130-
23	30
112	105
12	52
12	56



# Presente aquí su trabajo

Zona exclusiva para  
cálculos y desarrollos  
(borrador)

X → 100

✓ → desc

desc = Y. 100

✓ = Tarifa desc Real  
100

```

void imprimirCabecera( int *arrDniPaciente, int *arrTelefono,
    bool *arr Seguro, int posicion, ofstream &archReporte ) {
    imprimirLinea ('=' + AMLINEA, archReporte);
    archReporte << "DNI" << setw(8) << " " << arrDniPaciente[posicion]
        << setw(10) << " " << "Seguro";
    if (arrSeguro[posicion]) {
        archReporte << "EL PACIENTE ESTA ASEGURADO" << endl;
    }
    else {
        archReporte << "EL PACIENTE NO ESTA ASEGURADO" << endl;
        archReporte << "telefono:" << setw(3) << " " << arrTelefono
            << endl;
    }
    archReporte << "CITAS REALIZADAS:" << endl;
    imprimirLinea ('-', TAMLINEA, archReporte);
    archReporte << setw(15) << "FECHA" << setw(15) << "ID MEDICO"
        << setw(15) << "DESCUENTO" << setw(15) << "TARIFA" <<
        setw(15) << "LLEGADA" << setw(15) << "SALIDA" <<
        setw(15) << "TIEMPO (hrs.)" << setw(15) << "PAGO" <<
        setw(15) << "DESCUENTO" << endl;
    imprimirLinea ('-', TAMLINEA, archReporte);
}

int dniBuscar,
void imprimirCitas( // todos los arreglos y contadores, double
    &totalRecaudado, ofstream &archReporte ) { double tiempo;
    double totalCitas = 0.0; double Descuentos = 0.0; tarifa, desc, descReal;
    for (int i = 0; i < nCitas; i++) {
        if (arrDniCitas[i] == dniBuscar) {
            archReporte << H1('0') << setw(2) << arrDniCitas[i] <<
                "/" << setw(2) << arrMesCitas[i] << "/" << arrYrCitas[i] <<
                " " << arrCodMedCitas[i];
            BuscarMedico (arrCodMedCitas[i], arrCodMedico, arrTarifa,
                arrDescMedico, nMedicos, tarifa, desc);
            if (arrSeguro[i]) {
                descReal = desc;
            }
            archReporte << setw(15) << descReal << setw(15)
                << tarifa << fill('0') << setw(2) << arrHh[i] <<
                ":" << setw(2) << arrMin[i] << ":" <<
                setw(2) << arrSs[i] << " " <<
                // mismo formato para hora Salida
                tiempo = (Hh[i] * 3600 + Min[i] * 60 + Ss[i]) - (Hh[i] * 3600
                    + Min[i] * 60 + Ss[i]) / 3600.0;
            archReporte << " " << tiempo;
            descuento = tarifa * descReal / 100;
            pago = tarifa * tiempo - descuento;
            archReporte << setw(5) << " " << fixed << setprecision(2)
                << setw(10) << pago << setw(10) << descuento;
            totalCitas += pago;
            totalDescuentos += descuento;
        }
    }
    imprimirResumen2( archReporte, totalCitas, totalDescuentos,
        totalRecaudado += totalCitas;
}
    
```

no implementado

arr Citas →

no implementado

- reporte incompleto por falta de implementación de métodos
- no usa contadores para el reporte total de la clínica
- si usa acumuladores