

Año **2020** Número **1216**
Código de alumno

Primer examen

Araoz Andrade, Ivan Alexander

Apellidos y nombres del alumno (letra de imprenta)

Ivan Araoz

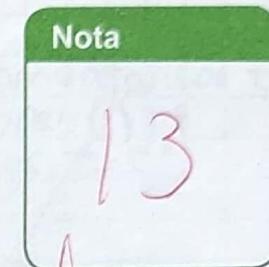
Firma del alumno

Curso: TP

Horario: 434

Fecha: 10/10/22

Nombre del profesor: E. Gómez Montoya



J.M.
Firma del profesor

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 este examen calificado, tome nota de las sugerencias que se le dan en la contracarátula del cuadernillo.

Presente aquí su trabajo

Zona exclusiva para
cálculos y desarrollos
(borrador)

PREGUNTA
1

Presentado

Muy bien

10/10

```
#include <iostream>
#include <fstream>

using namespace std;

#include "Funciones.hpp"

int main() {
    ifstream archUsuarios ("usuarios.txt", ios::in);
    if(not archUsuarios.is_open()){
        cout << "ERROR: usuarios.txt" << endl;
        exit(1);
    }

    ifstream archTiktoks ("Tiktoks.txt", ios::in);
    if(not archTiktoks.is_open()){
        cout << "ERROR: Tiktoks.txt" ;
        exit(1);
    }

    ifstream archComentarios ("comentarios.txt", ios::in);
    if(not archComentarios.is_open()){
        cout << "ERROR: comentarios.txt" ;
        exit(1);
    }

    ifstream archPaises ("paises.txt", ios::in);
    if(not archPaises.is_open()){
        cout << "ERROR: paises.txt" ;
        exit(1);
    }

    ofstream archReporte ("ReporteTiktoks.txt", ios::out);
    if(not archReporte.is_open()){
        cout << "ERROR: ReporteTiktoks";
        exit(1);
    }

    procesarReporte (archTiktoks, archUsuarios,
                     archComentarios, archPaises, archReporte);

    return 0;
}
```

Presente aquí su trabajo

#ifndef FUNCIONES_HPP
#define FUNCIONES_HPP

#include <iostream>
#include <fstream>
#include <iomanip>

using namespace std;

void procesarReporte (ifstream &, ifstream &,
 ifstream &, ifstream &, ofstream &);
void imprimirCabecera (ofstream &, int, int);
void imprimirLinea (char, int, ofstream &);
void imprimirCabeceraTikTok (ofstream &, int, int);
void buscarFecha (ofstream &, ifstream &, ifstream
&, ifstream &, int, int, int &, int &, int &);
void imprimirMejorTikTok (ofstream &, int, int, int,
 ifstream &);
void imprimirComentarios (ofstream &, ifstream &,
 ifstream &, ifstream &, int &, int &, int &, int &,
 int &, int &, bool &, int, int, int &);
void imprimirResumen (ofstream &, int, int, int, int,
 int, bool);
void imprimirUsuario (ofstream &, ifstream &, int,
 int);
void imprimirComentario (ofstream &, ifstream &);
void verificarPublico (ifstream, int, int, bool &);
void imprimirPais (ofstream &, bool, int, ifstream &);

Zona exclusiva para
cálculos y desarrollo
(borrador)

Presente aquí su trabajo

```
using namespace std;
```

```
#define TANLINEA 80
```

```
#include "Funciones.hpp"
```

void procesarReporte(ifstream &archTiktoks, ifstream &archUsuarios, ifstream &archComentarios, ifstream &archLikes, ofstream &archReporte){
 int ddI, mmI, yyyyI, ddF, mmF, yyyyF; fechaI, fechaF;
 cout << "Ingrese un rango de fechas" << endl;
 cin >> ddI >> mmI >> yyyyI >> ddF >> mmF >> yyyyF;
 fechaI = yyyyI * 10000 + mmI * 100 + ddI;
 fechaF = yyyyF * 10000 + mmF * 100 + ddF;
 • imprimirCabecera(archReporte, fechaI, fechaF);
 int dd, mm, yyyy, fecha, likes, vistas, contador=1;
 int mejorFecha = 0, mejorUsuario1, mejorUsuario2;
 char extra;
 while(true){
 archTiktoks >> dd;
 if(archTiktoks.eof()) break;
 archTiktoks >> extra >> mm >> extra >> yyyy;
 fecha = yyyy * 10000 + mm * 100 + dd;
 if(Fecha >= fechaI && Fecha <= fechaF){
 archTiktoks >> likes >> vistas;
 • imprimirCabeceraTiktok(archReporte, fecha, contador);
 • buscarFecha(archReporte, archUsuarios,
 archComentarios, archLikes, fecha, likes,
 vistas, mejorFecha, mejorUsuario1, mejorUsuario2);
 contador++;
 }
 else
 archTiktok.ignore(1000, '\n');
 }
 imprimirMejorTiktok(archReporte, mejorFecha,
 mejorUsuario1, mejorUsuario2, archUsuarios);
}

void imprimirCabecera(ofstream &archReporte,
 int fechaI, int fechaF){
 archReporte << setw(35) << " " << " @tiktok_tp" <<
 endl << setw(20) << " " << " REPORTE PARA LA"
 CAMPANA : Parcial 2022-1 << endl;
 char extra = '/';
 archReporte << setw(15) << " " << " Fecha Inicial : "
 int ddI, mmI, yyyyI, ddF, mmF, yyyyF;
 ddI = fechaI % 100;
 ddF = fechaF % 100;
 fechaI = fechaI / 100;
 fechaF = fechaF / 100;
 mmI = fechaI % 100;
 mmF = fechaF % 100;
 fechaI = fechaI / 100;
 fechaF = fechaF / 100;
 yyyyI = fechaI;
 yyyyF = fechaF; // setw(2), fill('0') para esto ↓
 archReporte << ddI << extra << mmI << extra << yyyyI
 << Fecha Final : << ddF << extra << mmF << extra << yyyyF
 << endl;

2022-10-12

80 -
9
70 12
30

8'

~ 80 -
80
30

Presente aquí su trabajo

Zona exclusiva para
cálculos y desarrollos
(borrador)

```
imprimirLinea(' = ', TAMLINEA, archReporte);
```

```
void imprimirLinea( char c, int tam, ofstream &arch) {  
    for( int i = 0; i <= tam; i++ ) {  
        arch.put(c);  
    }  
    arch.put( '\n' );  
}
```

```
void imprimirCabeceraTikTok( ofstream &archReporte, int fecha,  
    int contador ) {  
    archReporte << "TikTok" << contador << " " ;  
    int dd, mm, yyyy;  
    dd = fecha % 100;  
    fecha /= 100;  
    mm = fecha % 100;  
    yyyy = fecha / 100;  
    char extra = ( '/' );  
    archReporte.fill('0');  
    archReporte << setw(2) << dd << extra << setw(2) <<  
        mm << extra << yyyy << endl << endl;  
    archReporte.fill(' ');  
    archReporte << "COMENTARIO" << setw(20) << "AUTOR"  
        PEL COMENTARIO << setw(20) << "CANTIDAD DE"  
        LIKES << setw(20) << "PAÍS DEL AUTOR" <<  
        endl;  
    imprimirLinea(' - ', TAMLINEA, archReporte);  
}
```

```
void buscarFecha( ofstream &archReporte, ifstream &archUsuarios,  
    ifstream &archComentarios,  
    ifstream &archPaises, int fecha, int likes,  
    int vistas, int &mejorF, int &mejorU1, int  
    &mejorU2 ) {  
    int dd, mm, yyyy, fechaA, nComentarios = 0, nPublicos = 0;  
    int nPrivados = 0; bool viral; int likes;  
    char extra;  
    while( true ) {  
        archComentarios >> dd;  
        if( archComentarios.eof() ) break;  
        archComentarios >> extra >> mm >> extra >>  
            yyyy;  
        fechaA = yyyy * 10000 + mm * 100 + dd;  
        if( fechaA != fecha )  
            archComentarios.ignore( 1000, '\n' );  
        else {  
            * imprimirComentarios( archReporte, archComentarios,  
                archUsuarios, archPaises, mejorF, mejorU1,  
                mejorU2, nComentarios, nPublicos, nPrivados,  
                viral, fecha, vistas, likes );  
        }  
    }  
    imprimirResumen( archReporte, likes, vistas, nComentarios,  
        nPublicos, nPrivados, viral );  
}
```

Presente aquí su trabajo

Zona exclusiva para
cálculos y desarrollos
(borrador)

```
void imprimirMejorTikTok( ostream &archReporte,
    int fecha, int codP, int identificador, ifstream &archUsuarios,
    archReporte << endl << "RESUMEN MEJOR TIKTOK" <<
    endl << "FECHA : " << setw(5) << " ";
    int dd, mm, yyyy;
    dd = fecha % 100;
    fecha /= 100;
    mm = fecha % 100;
    yyyy = fecha / 100;
    char extra = '\n';
    archReporte.fill('0');
    archReporte << setw(2) << dd << extra << setw(2) <<
    mm << extra << yyyy << endl;
    archReporte.fill(' ');
    archReporte << "Seguidor que obtuvo mayor numero
    de likes." << setw(5) << " ";
    imprimirUsuario(archReporte, archUsuarios, codP,
        identificador);
    archReporte.put('\n');
    imprimirLinea(' - ', +AMLINEA, archReporte);
}
```

```
void imprimirComentarios( ostream &archReporte, ifstream &archComentarios,
    ifstream &archUsuarios, ifstream &archPublicos, int &mejorF, int &mejorU1, int
    &mejorU2, int &nComentarios, int &nPublicos, int &
    int &nPrivados, bool &viral, int fecha, int vistos,
    int codP, id, contador=0, nLikes, mayorLike=0,
    dobleRatioV; int likesT=0,
    char extra; bool publico;
    while (true) {
        extra = archComentarios.get();
        if (extra == '\n') break;
        archComentarios >> codP >> extra >> id;
        imprimirComentario(archReporte, archComentarios);
        nComentarios++;
        *imprimirUsuario(archReporte, archUsuarios, codP,
            id); verificarPublico(archUsuarios, codP, id, publico);
        archComentarios >> nLikes; likesT += nLikes;
        if (nLikes > mayorLike) {
            if (publico)
                mejorF = fecha;
            mejorU1 = codP;
            mejorU2 = id;
            nPublicos++;
        } else
            nPrivados++;
    }
    *imprimirPublico(archReporte, publico, codP, archPublicos);
    archReporte.put('\n');
    imprimirLinea(' - ', +AMLINEA, archReporte);
    ratioV = vistos / likesT * 100;
    if (nComentarios > ratioV)
        viral = true;
    likes = likesT;
}
```

Presente aquí su trabajo

Zona exclusiva para
cálculos y desarrollos
(borrador)

```
void imprimirResumen(ofstream & archReporte, int likes, int vistas, int nComentarios, int nPublicos, int nPrivados, bool viral) {
    archReporte << "Resumen TIKTOK" << setw(15) << "REPRODUCCIONES:" << endl;
    archReporte << "LIKES:" << likes << setw(15) << endl;
    archReporte << "VISTAS:" << vistas << endl;
    archReporte << "Comentarios:" << nComentarios << endl;
    archReporte << "Publicos:" << nPublicos << endl;
    archReporte << "Privados:" << nPrivados << endl;
    archReporte << "Seguidores NO Publicos:" << setw(20) << endl;
    if (viral)
        archReporte << "TIKTOK VIRAL" << endl;
    else
        archReporte << "TIKTOK NO VIRAL" << endl;
    imprimirLinea(' ', TAMLINEA, archReporte);
}
```

```
void imprimirUsuario(ofstream & archReporte, ifstream & archUsuarios, int codPB, int idB, int codP, int contador=0) {
    char extra;
    archUsuarios. clean();
    archUsuarios. seekg(0, ios::beg);
    while (true) {
        archUsuarios >> codP;
        if (archUsuarios.eof()) break;
        archUsuarios >> extra >> id;
        if (codP == codPB and id == idB) {
            archUsuarios >> ws;
            while (true) {
                extra = archUsuarios.get();
                if (extra == '\n') break;
                archReporte.put(extra);
                contador++;
            }
            break;
        }
    }
    archUsuarios.ignore(4000, '\n');
    for (int i=0; i < 30-contador; i++) {
        archReporte.put(' ');
    }
}
```

```
void imprimirComentarios(ofstream & archReporte,
    ifstream & archComentarios) {
    archComentarios >> ws;
    int contador = 0;
    char c;
    while (true) {
        c = archComentarios.get();
        if (c == '\n') break;
        archReporte.put(c);
        contador++;
    }
}
```

Presente aquí su trabajo

```
for(int i = 0; i < 30 - contador; i++) {
    archReporte.put(' ');
}

void verificarPublico(ifstream &archUsuarios, int codPB,
                     int idB, bool &publico) {
    int codP, id, seguidores;
    char extraP publicoC;
    archUsuarios.clear();
    archUsuarios.seekg(0, ios::beg);
    while(true) {
        archUsuarios.read(codP, 1);
        if(archUsuarios.eof()) break;
        archUsuarios.read(id, 1);
        if(codP == codPB and id == idB) {
            archUsuarios.read(seguidores, 1);
            archUsuarios.read(publicoC, 1);
            if(publicoC == 'V') {
                publico = true;
            } else {
                publico = false;
            }
            break;
        }
    }
    archUsuarios.ignore(1000, '\n');
}

void imprimirPaís(ofstream &archReporte, bool publico,
                   int codPB, ifstream &archPaíses) {
    int codP, char c;
    archPaíses.clear();
    archPaíses.seekg(0, ios::beg);
    while(true) {
        archPaíses.read(codP, 1);
        if(archPaíses.eof()) break;
        if(publico) {
            if(codP == codPB) { archPaíses.read(ws, 1);
                while(true) {
                    c = archPaíses.get();
                    if(c == ' ') break;
                    archReporte.put(c);
                }
            }
            break;
        } else {
            archPaíses.ignore(1000, '\n');
        }
    }
    archReporte >> "Información NO Pública";
}
```

Presente aquí su trabajo

```
#include <iostream>
#include <fstream>
#define MAXTIKTOKS 1000
using namespace std;

#include "Funciones.hpp"

int main() {
    int ddI, mmI, yyyyI, ddF, mmF, yyyyF, fechaI,
        fechaF; char extra;
    cout << "Ingrese el rango: " << endl;
    cin >> ddI >> extra >> mmI >> extra >> yyyyI >> ddF >>
        extra >> mmF >> extra >> yyyyF;
    fechaI = yyyyI * 10000 + mmI * 100 + ddI;
    fechaF = yyyyF * 10000 + mmF * 100 + ddF;
```

```
int arrFecha[MAXTIKTOKS], arrNComentarios [
    MAXTIKTOKS] = {3}, arrLikes[MAXTIKTOKS] = {3},
    arrNReproducciones[MAXTIKTOKS], arrSegPublicos
    [MAXTIKTOKS] = {3}, nTiktoks = 0;
double arrRatio[MAXTIKTOKS];
bool arrViral[MAXTIKTOKS];

int arrCodP[100], arrId[100], nUsuarios = 0;
bool publico[100];
```

```
leer Usuarios(arrCodP, arrId, arrPublico, nUsuarios);
leer Tiktoks(arrFecha, arrLikes, arrNReproducciones,
    nTiktoks, fechaI, fechaF);
```

procesar Comentarios (

Incompleto

return 0;

PREGUNTA
2

Zona exclusiva para
cálculos y desarrollos
(borrador)

Presente aquí su trabajo

```
#ifndef FUNCIONES_HPP
#define FUNCIONES_HPP

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

void leerUsuarios( int * , int * , bool * , int & );
void leerTiktoks( int * , int * , int * , int & , int ,
int );

#endif
```

Presente aquí su trabajo

Zona exclusiva para
cálculos y desarrollos
(borrador)

```
using namespace std;
```

```
#include "Funciones.hpp"
```

```
void leerUsuarios ( int *arrCodP, int *arrId, bool  
*arrPublico, int & nUsuarios ) {
```

```
    int seguidores;
```

```
    char extra; publico;
```

```
    ifstream archUsuarios ("usuarios.txt" ior :: in);
```

```
    if (not archUsuarios. is - open ()) {
```

```
        cout << "ERROR : usuarios.txt" << endl;
```

```
        exit (1);
```

```
}
```

```
while (true) {
```

```
    archUsuarios >> arrCodP [nUsuarios];
```

```
    if (archUsuarios .eof ()) break;
```

```
    archUsuarios >> extra >> arrId [nUsuarios];
```

```
    archUsuarios >> ws;
```

```
    while (true) {
```

```
        extra = archUsuarios .get ();
```

```
        if (extra == ' ') break;
```

```
}
```

```
    archUsuarios >> seguidores >> publico;
```

```
    if (publico == 'V')
```

```
        arrPublico [nUsuarios] = true;
```

```
    else
```

```
        arrPublico [nUsuarios] = false;
```

```
    nUsuarios ++;
```

```
}
```

3.0

falta

```
void leertiktoks ( int * arrFecha, int * arrNLikes,  
int * arrNReproducciones, int & ntiktoks, int fechaI,  
int dd, mm, yyyy, fechaI, fechaF );
```

```
char extra;
```

```
ifstream archTiktoks ("tiktoks.txt", ior :: in);
```

```
if (not archTiktoks. is - open ()) {
```

```
    cout << "ERROR : tiktoks.txt" << endl;
```

```
    exit (1);
```

```
while (true) {
```

```
    archTiktoks >> dd;
```

```
    if (archTiktoks .eof ()) break;
```

```
    archTiktoks >> extra >> mm >> extra >> yyyy;
```

```
    fecha = yyyy * 10000 + mm * 100 + dd;
```

```
    if (fecha >= fechaI and fecha <= fechaF) {
```

```
        arrFecha [ntiktoks] = fecha;
```

```
        archTiktoks >> arrNLikes [ntiktoks] >> arrNReproducciones
```

```
        [ntiktoks];
```

```
        ntktoks ++;
```

```
    else
```

```
        ignore (1000, '\n');
```

```
3
```