

Parcial 1 informática.

Por: Sergio Hoyos Mejía.

2023.

Universidad de Antioquía.

El propósito del presente trabajo es mostrar evidencias del proceso de elaboración del código para satisfacer las solicitudes o puntos a resolver del parcial y presentar los diferentes fallos en el proceso.

Planteamiento del problema.

Para iniciar a dar solución al problema, la primera meta fue planear cómo elaborar y diseñar el print de la matriz de horarios con el fin de crear la interfaz y comenzar a generar ideas para diseñar el apartado del funcionamiento.

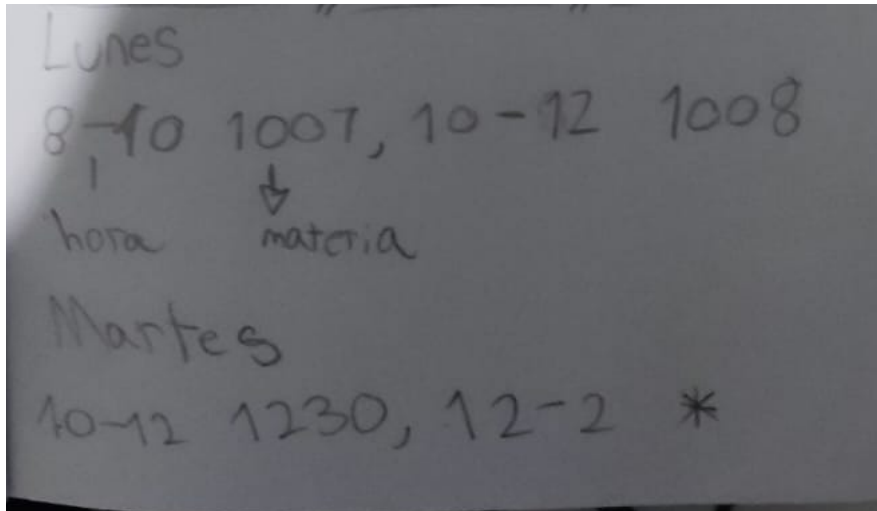
Formato  
\* = Libre - = ocupado

X	L	M	W	T	V
6-8	*	*	-	*	-
8-10	*	*	-	*	-
10-12	*	-	*	-	-

aquí sirve para mostrar si esta libre

Como se presenta en la anterior imagen, se diseñó la idea de cómo sería el print de materias, donde se trabajaría de lunes a viernes en un horario común de clases, luego en las columnas estarán los bloques de horas desde las 6 AM hasta las 6 PM y dentro de se llenaría los espacios con los caracteres '\*' para decir que el horario está libre y puede ingresar una materia el usuario, este se implementa de una manera similar a un punto de la práctica dos donde se realizaba un sistema de registro de sillas del cine, también se ideó el sistema para tomar los datos de las materias, al principio se planteó cómo crear el arreglo y qué información se le va a pedir al usuario sobre la materia, en este caso se le pediría: código de la materia, iniciales de los días que ve la materia, horas que ve la materia en semana, hora en

que empieza a ver la materia y los créditos, con eso planeaba hacer un sistema de asignación automática con la información recibida con el fin de que en el horario ya se generará las adecuaciones.



Como se muestra en la imagen, los datos a pedir y a organizar serían: Día de la clase, hora de inicio de clase y fin de clase y el código de la materia, luego se planteó el otro formato para darlo de una manera más completa y poder guardarlo correctamente en el txt.

Inicio de aplicación.

Lo primero que se pensó fue empezar a formular la estructura para asignar materias y el cómo se guardarán para luego tomar la información y poder usarla para imprimir la matriz con la información o guardarla en el archivo de texto.

```

10
11     cout<<"ingrese el numero de materias: ";
12     cin>>Num;
13
14     Materias = new char**[Num];
15
16     for(int i=0; i<Num; i++){
17         Materias[i]=new char*[6];
18
19         Materias[i][0] = new char[6];
20         cout<<"Ingrese el codigo de la materia ["<<i+1<<"] : ";
21         cin>>Materias[i][0];
22         cout<<endl;
23         Materias[i][1] = new char[25];
24         cout<<"Ingrese el nombre de la materia : ";
25         cin>>Materias[i][1];
26         cout<<endl;
27         Materias[i][2] = new char[3];
28         cout<<"Ingrese Horas de clase teorica : ";
29         cin>>Materias[i][2];
30         cout<<endl;
31         Materias[i][3] = new char[3];
32         cout<<"Ingrese Horas de clase practica: ";
33         cin>>Materias[i][3];
34         cout<<endl;
35         Materias[i][4] = new char[3];
36         cout<<"Ingrese el numero de creditos de la materia: ";
37         cin>>Materias[i][4];
38         cout<<endl;
39     }
40
41     //printer de info
42     for (int a = 0; a < Num; a++) {
43         for (int b = 0; b < 5; b++) {
44             for (int c = 0; c < 25; c++) {
45                 cout<<Materias[a][b][c];
46                 if (Materias[a][b][c]=='\0') break;

```

C:\Qt\Tools\QtCreator\bin\qtcreator\_proces...

```

ingrese el numero de materias: 2
Ingrese el codigo de la materia [1] : 1
Ingrese el nombre de la materia : q
Ingrese Horas de clase teorica : 3
Ingrese Horas de clase practica: 0
Ingrese el numero de creditos de la materia: 2
Ingrese el codigo de la materia [2] : 123
Ingrese el nombre de la materia : w
Ingrese Horas de clase teorica : 2
Ingrese Horas de clase practica: 0
Ingrese el numero de creditos de la materia: 3

1 | q | 3 | 0 | 2 |
123 | w | 2 | 0 | 3 |
Press <RETURN> to close this window...

```

El proceso inicial no presenta problemas para guardar las materias en el arreglo dinámico, en este caso el arreglo es una matriz de tres dimensiones donde las dos primeras definen filas y columnas y la última el espacio asignado para almacenar la información que se ingrese, el único detalle fue el apartado del orden del print, que salía desordenado debido a los espacios que imprime la matriz, ya que imprime hasta encontrar el carácter de finalización(\0) y los datos de los códigos tienen distinta longitud, por lo que afecta el print.

Después de verificar el correcto funcionamiento del proceso, se decidió el implementar el paso de guardar la información en un archivo de texto, por lo que se implementaron las librerías de fstream y istream que nos dan las funciones para poder abrir archivos de texto, escribir en ellos y cerrarlos para dejarlos creados en el apartado del build del compilador.

```
1  #include <iostream>
2  #include <fstream>
3  #include <istream>
4
5  using namespace std;
6
7  int main(void)
8  {
9      //////////Declaracion e inicializacion de Variables//////////
10
11      ofstream Archivo;//Variable para escribir el txt.
12      char ***Materias=nullptr; // Arreglo dinamico con ptr
13      int Num=0;//Numero de materias;
14      char documento[11];//Numero de documento
15
```

```
char *Caja=new char [50];
int paso=-1;
//////////Pasar datos a Arreglo Lineal//////////
for (int a = 0; a < Num; a++) {
    for (int b = 0; b < 5; b++) {
        for (int c = 0; c < 25; c++) {
            if (Materias[a][b][c]=='\0') break;

            if (Materias[a][b][c]!='\0'){
                paso++;
                Caja[paso]= Materias[a][b][c];
            }
            paso++;
            Caja[paso]='|';
        }
        paso++;
        Caja[paso]='\n';
    }
}
```

Se creo una funcion que permite el almacenar en “Caja” la información de la matriz materias, de ahí pasara Caja a almacenar la información que tiene adentro a un archivo txt y nombrandolo con el documento del estudiante. El formato de recepción de información se

cambio ya que la idea era que el estudiante ingresara la información de de la materia, es decir, código, nombre, horas teóricas y de práctica(de laboratorio) y los créditos, no los días que la ve ni las horas, eso sería del siguiente apartado, el registro del horario.

Para el sistema del horario se implementó el código de las materias pero adaptándolo al proceso que se iba a pedir, código de materia, horas y días para poder agendar el apartado de las horas teóricas y de laboratorio si se requería.

```
#include <iostream>

using namespace std;

bool comparar(char*** id, char cad[], int c);
void imprimir(int*** list);
void llenar(int*** horario);

int main()
{

    int n=2, dia=0, hora=0;
    int eleccion=0;
    int*** horario = new int**[n];

    for (int i=0; i<n+1 ; i++){
        horario[i] = new int*[6]; //memoria para columnas
        for (int j = 0; j < 6; j++) {
            horario[i][j] = new int[7]; // memoria para dato
        }
    }

    llenar(horario);
    while(eleccion!=9){

        cout<<"-----CRONOGRAMA-----"<<endl;
```

```
while(eleccion!=0){

    cout<<"-----CRONOGRAMA-----"<<endl;

    cout<<"Ingrese que día desea registrar: "<<endl;
    |      cout<<"1-Lunes"<<endl;
    cout<<"2-Martes"<<endl;
    cout<<"3-Miercoles"<<endl;
    cout<<"4-Jueves"<<endl;
    cin>>dia;

    cout<<"Ingrese el horario donde registrara la materia: "<<endl;
    cout<<"1-> 6-8"<<endl;
    cout<<"2-> 8-10"<<endl;
    cout<<"3-> 10-12"<<endl;
    cin>>hora;

    imprimir(horario);

    cout<<"\n El usuario ha sido agregado. Presiona cualquier tecla para agregar otro usuario\n o 'x' para terminar: ";
    cin>>eleccion;
    if (eleccion=='*'){
    |      //break;
    }
    else if(eleccion=='x'){
    |      //continue;
    }
```

```

    delete [] horario;
}

void llenar(int*** horario){
    for(int i=0; i<=6; i++){
        for(int j=0; j<=7; j++){
            horario[i][j]=0;
            cout<<"entro en"<<horario[i][j]<<" ";
        }
        cout<<endl;
    }
}

void imprimir(int*** horario){
    for(int i=0; i<=6; i++){
        for(int j=0; j<=7; j++){
            horario[i][j]=0;
            cout<<horario[i][j]<<" ";
        }
        cout<<endl;
    }
}

```

En la ultima imagen se presentan las funciones, donde la importante para mantener ordenado el formato de impresion ordenado es la de llenar, donde agregar el caracter espacio para ordenar las posiciones del arreglo y al imprimir quede simetrico.

Unión de codigos y proceso final.

Para empezar a dar forma y final al parcial se unieron los codigos para completar el proceso de registro y a su vez iniciar el punto final, agendar las horas de repaso en el horario a generar, donde se presentarian las opciones de agendar las horas de repaso manual o automatico con la implementacion de las librerias time.h y stdlib.h para poder hacer el proceso de generar en automatico las horas de repaso por materia.

```

#include <iostream>
#include <fstream>
#include <istream>
#include <time.h>
#include <stdlib.h>

void llenar(char*** horario);
void imprimir(char ***Matriz, int tam2, int tam, int slots);
void formatotxt(char ***Materias, int Num, int paso, char *Caja);
void liberador(char ***Materias, char ***horario, char *Caja, char *Cajahorario, int Num);
void random(char ***horario, char ***Materias, int Num);
void rellenarArreglo(char *arreglo, int longitud, char relleno);
void formathorariotxt(char ***horario, int paso, char *Cajahorario);
int conversor(char Dato[30]);

```

```

void formathorariotxt(char ***horario, int paso, char *Cajahorario){
    for (int i = 0; i < 14; i++) {
        for (int b = 0; b < 6; b++) {
            for (int c = 0; c < 10; c++) {
                if (horario[i][b][c]=='\0') break;

                if (horario[i][b][c]!='\0'){
                    paso++;
                    Cajahorario[paso]= horario[i][b][c];
                }
            }
            paso++;
            Cajahorario[paso]='|';
        }
        paso++;
        Cajahorario[paso]='\n';
    }
}

void rellenarArreglo(char *arreglo, int longitud, char relleno) {
    int tamanoActual = 0;
    while (arreglo[tamanoActual] != '\0') {
        tamanoActual++;
    }
    int espaciosSobrantes = longitud - tamanoActual;
    if (espaciosSobrantes > 0) {
        for (int i = tamanoActual; i < longitud; i++) {
            arreglo[i] = relleno;
        }
        arreglo[longitud] = '\0';
    }
}

```

Se crearon diferentes funciones que ayudan con la elaboracion del codigo, el caso de RellenarArreglo nos permitia llenar los espacios de una ubicacion definida de la matriz de



horario para mantenerlo ordenado y guardarlo sin moverse los caracteres “|” que separan la información.

```
int conversor(char Dato[30]){  
  
    int n=0;  
    for(int i=0; Dato[i]!='\0'; i++){  
        n=n*10+(Dato[i]-'0');  
    }  
    return n;  
}
```

La función conversor nos facilitaba la posibilidad de convertir datos char a int, esta se utilizaba principalmente para poder hacer el calculo de las horas de repaso para cada materia, convirtiendo los datos de los credits, horas teoricas y practicas de char como se ingresa al int para hacer la operación.

```
void random(char ***horario, char ***Materias, int Num){  
  
    for(int i = 0; i<Num; i++){  
  
        int aux = (conversor(Materias[i][4])*3)-(conversor(Materias[i][2])+conversor(Materias[i][3]));  
        cout<<"la operacion da: "<<aux<<endl;  
  
        int Q = 0;  
        while(Q<aux){  
            int dia = 1+rand()%(6-1);  
  
            int hora = 1+rand()%(14-1);  
  
            cout<<dia<<" "<<hora<<endl;  
  
            if(horario[hora][dia][0] == ' '){  
  
                cout<<"Funciona :D"<<endl;  
                Q++;  
                horario[hora][dia] = Materias[i][0];  
                rellenarArreglo(horario[hora][dia], 10, ' ');  
            }  
        }  
    }  
  
    imprimir(horario, 14, 6, 10);  
}
```

La función random permite la posibilidad de asignar las horas de repaso de manera automatica si el usuario lo desea así, aquí se implementan la función de convertir de char a int y rand para aleatorizar los datos de día y hora para poder asignar poco a poco las horas de repaso de cada materia, esta función tiene un while que funcionara hasta que Q sea igual a aux, aux es la variable que almacenara la cantidad de horas de repaso de cada materia, por lo que cada que entre al condicional donde verifica que la posición que se asigna al tomar el día

y la hora, verifica si esa casilla es un espacio para poder aumentar Q y anexar el código de la materia, de lo contrario repetirá el proceso hasta encontrar un espacio vacío.

Errores que se presentaron frecuentemente durante la solución.

a continuación se presentarán los errores más frecuentes durante la elaboración de la entrega final y su respectiva solución:

Error comparador de códigos de materia: se resolvió realizando otro método de verificación donde se implementa el conversor de materias y un bool que será true o false dependiendo si se cumple que son iguales o no (Solución en la segunda imagen).

```
Archivo.open(documento);
Archivo << Caja;
Archivo.close();

liberador(Materias, Num);
}
|
/*bool Comparar(char** Materias, char cad[], int c){
    for(int i = 1; i < c-1; i++){
        if(Materias[i][0]==Materias[i+1][0] || Materias[i][0]==Materias[i-1][0]){
            return true;
        }
    }
    return false;
}*/

void imprimirmate(char ***Materias, int Num){
    for (int a = 0; a < Num; a++) {
        for (int b = 0; b < 5; b++) {
            for (int c = 0; c < 25; c++) {
                cout<<Materias[a][b][c];
                if (Materias[a][b][c]=='\0') break;
            }
            cout<<" | ";
        }
        cout<<endl;
    }
}

void liberador(char ***Materias, int Num){

    for (int a = 0; a < Num; a++) {
        for (int b = 0; b < 5; b++) {
            delete[] Materias[a];
        }
    }
    delete[] Materias;
    Materias=nullptr;
}
```

```

        if(i>0){
            bool H = true;

            while(H){
                cout << "Ingrese el codigo de la materia ["<<i+1<<"] : ";
                cin >> Materias[i][0];

                int a = conversor(Materias[i][0]);

                H = false;
                for(int k = 0; k < i; k++){
                    int b = conversor(Materias[k][0]);
                    if(a == b){
                        cout << "La materia ya fue ingresada, intente otra." << endl;
                        H = true;
                        break; // Salir del ciclo for
                    }
                }
            }
        }
        else{
            cout<<"Ingrese el codigo de la materia ["<<i+1<<"] : ";
            cin>>Materias[i][0];
        }
    }
}

```

Error verificador de materias: La solución fue implementar un break justo donde el condicional verifica que si las materias son iguales o no y asignado al bool H como true para que siga iterando

```

cout<<"ingrese el numero de materias: ";
cin>>Num;
////////// Matriz dinamica de Datos//////////

Materias = new char**[Num];

for(int i=0; i<Num; i++){
    Materias[i]=new char*[0];

    Materias[i][0] = new char[0];

    if(i>0){
        while(H!=false){

            cout<<"Ingrese el codigo de la materia ["<<i+1<<"] : ";
            cin>>Materias[i][0];

            int a = conversor(Materias[i][0]);

            for(int k = 0; k<i; k++){
                int b = conversor(Materias[k][0]);
                if(a==b){
                    cout<<"La materia ya fue ingresada, intente otra "<<endl;
                }
                else{
                    H=false;
                }
            }
        }
    }
    else{
        cout<<"Ingrese el codigo de la materia ["<<i+1<<"] : ";
        cin>>Materias[i][0];
    }

    cout<<endl;
    Materias[i][1] = new char[25];
    cout<<"Ingrese el nombre de la materia : ";
    cin>>Materias[i][1];
    cout<<endl;
    Materias[i][2] = new char[0];
}

```

```

C:\Qt\Tools\QtCreator\bin\qtcreator_process_stu...
ingrese su numero de documento: 123
documento dos H123
ingrese el numero de materias: 4
Ingrese el codigo de la materia [1] : 1

Ingrese el nombre de la materia : 1

Ingrese Horas de clase teorica : 1

Ingrese Horas de clase practica: 1

Ingrese el numero de creditos de la materia: 1

Ingrese el codigo de la materia [2] : 1
La materia ya fue ingresada, intente otra
Ingrese el codigo de la materia [2] : 2

Ingrese el nombre de la materia : 2

Ingrese Horas de clase teorica : 2

Ingrese Horas de clase practica: 2

Ingrese el numero de creditos de la materia: 2

Ingrese el nombre de la materia : _

```

Error desorden de impresión de materias y guardado en su Txt: La solución era implementando la función de rellenar arreglo, para anexarle el caracter espacio y con ello poder organizar el print que quede simetrico(No se implemento).

```

90      cout<<endl;
91      Materias[i][0] = new char[3];
92      cout<<"Ingrese Horas de clase practica: ";
93      cin>>Materias[i][0];
94      cout<<endl;
95      Materias[i][1] = new char[3];
96      cout<<"Ingrese el numero de creditos de la materia: ";
97      cin>>Materias[i][1];
98      cout<<endl;
99  }
100
101  ////////////////////////////////////Imprimir Datos////////////////////////////////////
102
103  imprimir(Materias, Num, 5, 10);
104  cout<<endl;
105  char *almacen=new char [50];
106  int paso=1;
107  ////////////////////////////////////Pasos datos a Arreglo lineal////////////////////////////////////
108
109  formatotxt(Materias, Num, paso, almacen);
110  ////////////////////////////////////Escribiendo el Archivo////////////////////////////////////
111
112  Archivo.open(documento);
113  Archivo <<almacen ;
114  Archivo.close();
115
116
117
118
119
120
121
122  ////////////////////////////////////crear matriz para horarios////////////////////////////////////
123  horario = new char**[4];
124  for (int i = 0; i < 4; i++) {
125      horario[i] = new char*[10];
126      for (int j = 0; j < 6; j++) {
127          horario[i][j] = new char[10];
128      }
129  }
130
131  llenar(horario);
132
133  imprimir(horario, 14, 8, 10);

```

```

Ingrese el numero de creditos de la materia: 3
Ingrese el codigo de la materia [3]: 234432
Ingrese el nombre de la materia : algebra
Ingrese Horas de clase teorica : 4
Ingrese Horas de clase practica: 0
Ingrese el numero de creditos de la materia: 3
Ingrese el codigo de la materia [4]: 345543
Ingrese el nombre de la materia : Informatica
Ingrese Horas de clase teorica : 4
Ingrese Horas de clase practica: 3
Ingrese el numero de creditos de la materia: 4

| 1 | | 1 | | 1 | | 1 | | 1 | | 0 | | 3 | |
| 1 2 3 2 1 | | calculo | | 4 | | 0 | | 3 | |
| 2 3 4 4 3 2 | | algebra | | 4 | | 0 | | 3 | |
| 3 4 5 5 4 3 | | informatica | | 4 | | 3 | | 4 | |
Press <RETURN> to close this window...

```

error print desordenado de horario: La solución fue rellenar arreglo con ayuda de la función rellenar arreglo con el caracter espacio.

```

cout << "Ingrese el segundo día que desea registrar: " << endl;
cout << "1-Lunes" << endl;
cout << "2-Martes" << endl;
cout << "3-Miercole" << endl;
cout << "4-Jueves" << endl;
cout << "5-Viernes" << endl;
cin >> dia2;

cout << "Ingrese el horario donde registrara la materia: " << endl;
cout << "1-> 6" << endl;
cout << "2-> 7" << endl;
cout << "3-> 8" << endl;
cout << "4-> 9" << endl;
cout << "5-> 10" << endl;
cout << "6-> 11" << endl;
cout << "7-> 12" << endl;
cout << "8-> 13" << endl;
cout << "9-> 14" << endl;
cout << "10-> 15" << endl;
cout << "11-> 16" << endl;
cin >> hora;

if (horario[hora][dia][0] == ' ' && horario[hora][dia2][0] == ' ')
{
    cout<<endl;
    imprimir(Materias, Num, 5, 25);
    cout<<endl;
    cout << "Ingresa el codigo de la materia: ";
    cin >> horario[hora][dia];

    horario[hora+1][dia]=horario[hora][dia];

    cout << endl;
    imprimir(horario, 14, 6, 10);
}
else {
    cout<< "Ya esta ocupado el horario, intenta con otro. "<<endl;
}

```

```

7-> 12
8-> 13
9-> 14
10-> 15
11-> 16
1

| 1 | | 1 | | 1 | | 1 | | 1 | |
| 2 | | 2 | | 2 | | 2 | | 2 | |
| 3 | | 3 | | 3 | | 3 | | 3 | |

Ingresa el codigo de la materia: 1

| H o r a | | 1 . L u n e s | | 2 . M a r t e s | | 3 . M i e r c | | | | |
| 6 | | 1 | | | | | | | | |
| 7 | | 1 | | | | | | | | |
| 8 | | | | | | | | | | |
| 9 | | | | | | | | | | |
| 10 | | | | | | | | | | |
| 11 | | | | | | | | | | |
| 12 | | | | | | | | | | |
| 13 | | | | | | | | | | |
| 14 | | | | | | | | | | |
| 15 | | | | | | | | | | |
| 16 | | | | | | | | | | |
| 17 | | | | | | | | | | |
| 18 | | | | | | | | | | |

La materia ha sido agendada. Presiona cualquier tecla para agendar otra materia
o '*' para terminar:

```

```

while (arreglo[tamanoActual] != '\0') {
    tamanoActual++;
}

int espaciosSobrantes = longitud - tamanoActual;
if (espaciosSobrantes > 0) {
    for (int i = tamanoActual; i < longitud; i++) {
        arreglo[i] = ' ';
    }
    arreglo[longitud] = '\0';
}

int conversor(char Dato[80]){
    int n=0;
    for(int i=0; Dato[i]!='\0'; i++){
        n=n*10+(Dato[i]-'0');
    }
    return n;
}

void random(char ***horario, char ***Materias, int Num){

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

        int aux = (conversor(Materias[i][0])*3)-(conversor(Materias[i][1])*4)+5;
        cout<<"la operacion da: "<<aux<<endl;

        int Q = 0;
        while(Q<aux){
            int dia = 1+rand()%(6-1);

            int hora = 1+rand()%(4-1);

            cout<<dia<<" " <<hora<<endl;

            if(horario[hora][dia][0] == ' '){

                cout<<"Funciona :D"<<endl;
                Q++;
            }
        }
    }
}

```

```

188     horario[hora+1][dia]-horario[hora][dia];
189     rellenarArreglo(horario[hora][dia], 10, ' ');
190
191     horario[hora][dia2] = horario[hora][dia];
192     horario[hora+1][dia2] = horario[hora][dia2];
193
194
195
196     cout << endl;
197     imprimir(horario, 10, 6, 10);
198
199     else {
200         cout<< "Ya esta ocupado el horario, intenta con otro."<<endl;
201     }
202
203     //break;
204
205
206     /*case 2:
207         ...*/
208
209
210     cout << "\nla materia ha sido agendada. Presiona cualquier tecla para agendar
211     cin >> eleccion;
212     cout << endl;
213
214 } while (eleccion != '*');
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

Situaciones que se impidieron agregar verificadores u otras funciones.

Como en toda situación, el tiempo es un factor clave y determinante para resolver las diferentes dificultades o problemas a resolver, en este caso para implementar mas verificadores de datos se requería mas tiempo, el programa funcionaria sin problemas para el usuario perfecto que no cometería errores, para un usuario común o no perfecto faltarian filtros para la información, de resto todas las soluciones se implementaron de la forma mas eficiente posible desde el conocimiento establecido hasta el momento.