

```
1  /*
2  * Archivo:    main.cpp
3  * Autor: Oscar Dueñas Damian - Oscar DD.
4  * Código PUCP: 20180146
5  * Created on 20 de septiembre de 2020, 08:08 PM
6  */
7
8  #include <iostream>
9  #include <iomanip>
10 #include <fstream>
11 #include "funciones_aux.h"
12 using namespace std;
13
14 int main(int argc, char** argv) {
15     crearArchivosBin();
16     ifstream enero = abrirArchL("Enero.bin", 'B');
17     ifstream febrero = abrirArchL("Febrero.bin", 'B');
18     ifstream marzo = abrirArchL("Marzo.bin", 'B');
19     ifstream registro = abrirArchL("MovEnCuentas.txt", 'T');
20     ofstream reporte = abrirArchE("Reporte.txt", 'T');
21     generarReporte(reporte, registro, enero, febrero, marzo);
22     return (EXIT_SUCCESS);
23 }
24
```

```
1  /*
2  * Archivo:   funciones_aux.h
3  * Author:   Oscar Dueñas Damian Oscar DD.
4  *Codigo PUCP: 20180146
5  * Created on 20 de septiembre de 2020, 09:48 PM
6  */
7
8  #ifndef FUNCIONES_AUX_H
9  #define FUNCIONES_AUX_H
10 #include <fstream>
11 using namespace std;
12
13 void imprimeLinea (char , int , ofstream &);
14 ifstream abrirArchL (const char *, char );
15 ofstream abrirArchE (const char *, char );
16 void llenarArchivoBin (ofstream &, ifstream &);
17 void crearArchivosBin ();
18 void generarReporte (ofstream &, ifstream &, ifstream &,ifstream &, ifstream &);
19 void imprimeCabecera (int , ofstream &);
20 void procesarDatosCliente (char *, ifstream &, ofstream &, char &, double &);
21 void imprimirMonedaCuenta (char , ofstream &);
22 void procesarTransaccionesCliente (char , ifstream &, ofstream &, ifstream &,
23     ifstream &, ifstream &, double);
24 void procesarTransaccionesFecha (char , int , ifstream &,
25     ofstream &, ifstream &, ifstream &, ifstream &, double &, double &,
26     int &, int &);
27 double calcularFator (char , char , int , ifstream &, ifstream &,
28     ifstream &);
29 double buscarCambio (char , ifstream &);
30 void imprimirResumen (int , int , double , double , double , char , ofstream &);
31
32 #endif /* FUNCIONES_AUX_H */
33
```

```

1  /*
2  * Archivo:   funciones_aux.cpp
3  * Autor: Oscar Dueñas Damian - Oscar DD.
4  *Codigo PUCP: 20180146
5  * Created on 20 de septiembre de 2020, 08:17 PM
6  */
7
8  #include <iostream>
9  #include <iomanip>
10 #include <fstream>
11 #include <cstring>
12 #define MAX_CAR 110
13 using namespace std;
14
15 void imprimeLinea (char car, int num, ofstream &arch) {
16     for (int i = 0; i < num; i++) arch.put(car);
17     arch << endl;
18 }
19
20 ifstream abrirArchL (const char *nombre, char modo) {
21     ifstream arch;
22     if (modo == 'T') arch.open(nombre, ios::in);
23     else arch.open(nombre, ios::in | ios::binary);
24     if (!arch) {
25         cout << "ERROR: no se pudo abrir el archivo " << nombre << endl;
26         exit(1);
27     }
28     return arch;
29 }
30
31 ofstream abrirArchE (const char *nombre, char modo) {
32     ofstream arch;
33     if (modo == 'T') arch.open(nombre, ios::out);
34     else arch.open(nombre, ios::out | ios::binary);
35     if (!arch) {
36         cout << "ERROR: no se pudo abrir el archivo " << nombre << endl;
37         exit(1);
38     }
39     return arch;
40 }
41
42 void llenarArchivoBin (ofstream &archB, ifstream &arch) {
43     char car, moneda[15];
44     double cambio;
45     while (1) {
46         arch >> car;
47         if (arch.eof()) break;
48         arch >> cambio >> moneda;
49         archB.write(reinterpret_cast<const char*>(&car), sizeof(char));
50         archB.write(reinterpret_cast<const char*>(&cambio), sizeof(double));
51         archB.write(reinterpret_cast<const char*>(&moneda), sizeof(char)*15);
52     }
53 }
54
55 void crearArchivosBin () {
56     ofstream eneroBin = abrirArchE("Enero.bin", 'B');
57     ofstream febreroBin = abrirArchE("Febrero.bin", 'B');
58     ofstream marzoBin = abrirArchE("Marzo.bin", 'B');
59     ifstream enero = abrirArchL("Enero.txt", 'T');
60     ifstream febrero = abrirArchL("Febrero.txt", 'T');
61     ifstream marzo = abrirArchL("Marzo.txt", 'T');
62     llenarArchivoBin(eneroBin, enero);
63     llenarArchivoBin(febreroBin, febrero);
64     llenarArchivoBin(marzoBin, marzo);
65     eneroBin.close(); enero.close();
66     febreroBin.close(); febrero.close();
67     marzoBin.close(); marzo.close();
68 }
69
70 void imprimeCabecera (int opc, ofstream &reporte) {
71     imprimeLinea('-', MAX_CAR, reporte);
72     if (opc == 1) {
73         reporte << "CLIENTE"

```

```

74         << setw(50) << "CODIGO DE CUENTA"
75         << setw(15) << "MONEDA"
76         << setw(35) << "SALDO INICIAL" << endl;
77     } else {
78         reporte << "FECHA"
79         << setw(30) << "RETIROS"
80         << setw(35) << "DEPOSITOS"
81         << setw(30) << "SALDO" << endl;
82         imprimeLinea('-',MAX_CAR,reporte);
83     }
84 }
85
86 void imprimirMonedaCuenta (char M, ofstream &reporte) {
87     if (M == 'S') reporte << setw(20) << "SOLES";
88     else if (M == '$') reporte << setw(20) << "DOLARES";
89     else if (M == '&') reporte << setw(20) << "EUROS";
90     else if (M == 'Y') reporte << setw(20) << "YEN";
91     else if (M == 'U') reporte << setw(20) << "RUBLO";
92     else if (M == 'P') reporte << setw(20) << "PESO";
93     else if (M == 'B') reporte << setw(20) << "BOLIVIANO";
94     else if (M == 'C') reporte << setw(20) << "COLON";
95     else if (M == 'L') reporte << setw(20) << "LIBRA";
96     else if (M == 'A') reporte << setw(20) << "AFGANI";
97     else reporte << setw(20) << "FRANCO";
98 }
99
100 void procesarDatosCliente (char *cadena, ifstream &registro, ofstream &reporte,
101     char &M, double &saldo) {
102     char nombre[15];
103     int codigo, lon = strlen(cadena)+1;
104     imprimeCabecera(1,reporte);
105     reporte << cadena << " ";
106     while (1) {
107         registro >> codigo;
108         if (registro.fail()) {
109             registro.clear();
110             registro >> nombre;
111             reporte << nombre << " ";
112             lon += strlen(nombre) + 1;
113         } else break;
114     }
115     reporte << setw(45-lon) << " " << setw(5) << codigo;
116     registro >> M >> saldo;
117     imprimirMonedaCuenta(M,reporte);
118     reporte << setw(32) << saldo << endl;
119     imprimeCabecera(2,reporte);
120 }
121
122 double buscarCambio (char moneda, ifstream &arch) {
123     char mon;
124     double cambio;
125     int tamDelRegistro = sizeof(char)*16 + sizeof(double);
126     arch.seekg(0, ios::end);
127     int tamDelArch = arch.tellg();
128     arch.seekg(0, ios::beg);
129     int numRegistros = tamDelArch/tamDelRegistro;
130     for (int i = 0; i < numRegistros; i++) {
131         arch.seekg(i*tamDelRegistro, ios::beg);
132         arch.read(reinterpret_cast<char*>(&mon), sizeof(char));
133         arch.read(reinterpret_cast<char*>(&cambio), sizeof(double));
134         if (mon == moneda) return cambio;
135     }
136 }
137
138 double calcularFator(char M, char mon, int mes, ifstream &enero, ifstream &febrero,
139     ifstream &marzo) {
140     double cambio, cuenta;
141     if (mes == 1) {
142         cuenta = buscarCambio(M,enero);
143         cambio = buscarCambio(mon,enero);
144     } else if (mes == 2) {
145         cuenta = buscarCambio(M,febrero);
146         cambio = buscarCambio(mon,febrero);

```

```

147     } else {
148         cuenta = buscarCambio(M,marzo);
149         cambio = buscarCambio(mon,marzo);
150     }
151     return cambio/cuenta;
152 }
153
154 void procesarTransaccionesFecha (char M, int mes, ifstream &registro,
155     ofstream &reporte, ifstream &enero, ifstream &febrero, ifstream &marzo,
156     double &subtotal_depositos, double &subtotal_retiros, int &depositos,
157     int &retiros) {
158     char tipo, mon, tipoAnt;
159     double saldo, factor;
160     subtotal_depositos = 0; subtotal_retiros = 0; depositos = 0; retiros = 0;
161     while (registro.get() != '\n') {
162         registro >> tipo;
163         if (tipo == 'D' || tipo == 'R') {
164             registro >> mon >> saldo;
165             factor = calcularFator(M,mon,mes,enero,febrero,marzo);
166             if (tipo == 'D') {
167                 subtotal_depositos += saldo*factor;
168                 depositos++;
169             } else {
170                 subtotal_retiros += saldo*factor;
171                 retiros++;
172             }
173             tipoAnt = tipo;
174         } else {
175             registro >> saldo;
176             factor = calcularFator(M,tipo,mes,enero,febrero,marzo);
177             if (tipoAnt == 'D') {
178                 subtotal_depositos += saldo*factor;
179                 depositos++;
180             } else {
181                 subtotal_retiros += saldo*factor;
182                 retiros++;
183             }
184         }
185     }
186 }
187
188 void imprimirResumen (int num_depositos, int num_retiros, double total_depositos,
189     double total_retiros, double saldo, char M, ofstream &reporte) {
190     imprimeLinea('=',MAX_CAR,reporte);
191     reporte << "RESUMEN" << endl;
192     reporte << "CANTIDAD TOTAL DE RETIROS: "
193         << setw(10) << num_retiros
194         << setw(30) << "TOTAL DE RETIROS: "
195         << setw(5) << M
196         << setw(12) << total_retiros << endl
197         << "CANTIDAD TOTAL DE DEPOSITOS: "
198         << setw(8) << num_depositos
199         << setw(30) << "TOTAL DE DEPOSITOS: "
200         << setw(5) << M
201         << setw(12) << total_depositos << endl
202         << "SALDO FINAL: "
203         << setw(5) << M
204         << setw(12) << saldo << endl;
205 }
206
207 void procesarTransaccionesCliente (char M, ifstream &registro, ofstream &reporte,
208     ifstream &enero, ifstream &febrero, ifstream &marzo, double saldo_ini) {
209     int dd, mm, aa, depositos, retiros, num_depositos = 0, num_retiros = 0;
210     double subtotal_depositos, subtotal_retiros, total_depositos = 0,
211         total_retiros = 0, saldo = saldo_ini;
212     char car;
213     while (1) {
214         registro >> dd;
215         if (registro.fail()) {
216             registro.clear();
217             break;
218         }
219         registro >> car >> mm >> car >> aa;

```

```

220     reporte.fill('0');
221     reporte << setw(2) << dd << '/' << setw(2) << mm << '/' << setw(4) << aa;
222     reporte.fill(' ');
223     procesarTransaccionesFecha(M,mm,registro,reporte,enero,febrero,marzo,
224         subtotal_depositos,subtotal_retiros,depositos,retiros);
225     saldo += (subtotal_depositos - subtotal_retiros);
226     num_depositos += depositos; num_retiros += retiros;
227     total_retiros += subtotal_retiros; total_depositos += subtotal_depositos;
228     reporte << setw(15) << M
229         << setw(12) << subtotal_retiros
230         << setw(20) << M
231         << setw(15) << subtotal_depositos
232         << setw(20) << M
233         << setw(12) << saldo << endl;
234 }
235 imprimirResumen(num_depositos,num_retiros,total_depositos,total_retiros,
236     saldo, M, reporte);
237 }
238
239 void generarReporte (ofstream &reporte, ifstream &registro, ifstream &enero,
240     ifstream &febrero, ifstream &marzo) {
241     char cadena[15], M;
242     double saldo_ini;
243     reporte.precision(2); reporte << fixed;
244     reporte << setw(60) << "ESTADO DE CUENTAS DEL BANCO ABCD" << endl;
245     while (1) {
246         registro >> cadena;
247         if (registro.eof()) break;
248         procesarDatosCliente(cadena,registro,reporte,M,saldo_ini);
249         procesarTransaccionesCliente(M,registro,reporte,enero,febrero,marzo,
250             saldo_ini);
251     }
252 }

```