



UNIVERSIDAD
NACIONAL
DE COLOMBIA

UNIVERSIDAD NACIONAL DE COLOMBIA
FACULTAD DE CIENCIAS
DEPARTAMENTO DE ESTADÍSTICA
Programación de computadores - Grupo 19
Michel Mendivenson Barragán Zabala

Para esta práctica debe usar el cajero desarrollado en el Challenge 2 y adaptarlo a una arquitectura de programación basada en funciones de la siguiente manera:

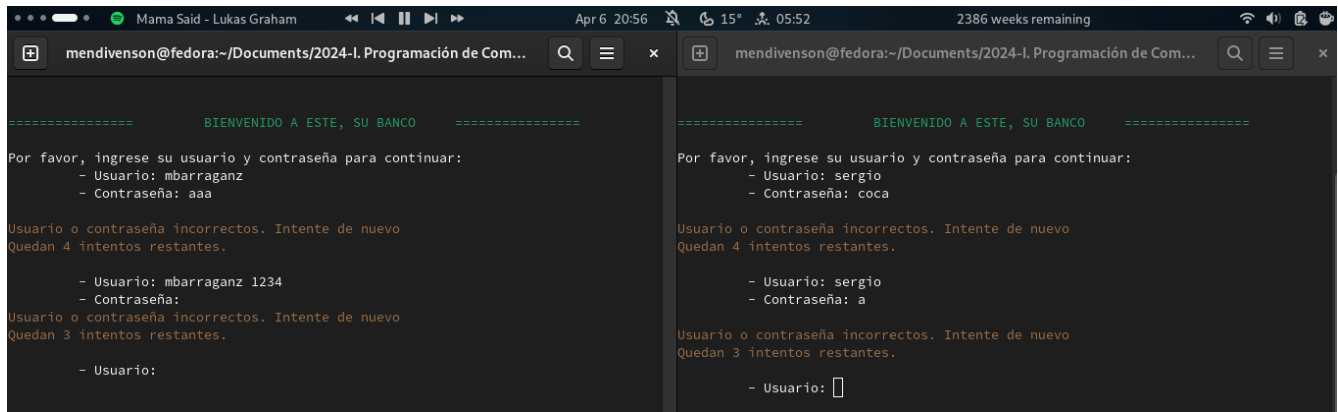
```
1 #include <iostream>
2 #include <map>
3
4 using namespace std;
5 // ===== USUARIOS Y CONTRASEÑAS =====
6 map<string, string> usuarios = {
7     {"mbarraganz", "1234contraseña"},
8     {"sergio", "cocacola"}
9 };
10
11 // ===== USUARIOS Y SALDOS CUENTAS CORRIENTE =====
12 map<string, long> saldosCorriente = {
13     {"mbarraganz", 10000000},
14     {"sergio", 10000000}
15 };
16
17 // ===== USUARIOS Y SALDOS CUENTAS AHORROS =====
18 map<string, long> saldosAhorros = {
19     {"mbarraganz", 15000000},
20     {"sergio", 15000000}
21 };
22
23 // ===== USUARIOS Y CUPOS CUENTAS DE CRÉDITO =====
24 map<string, long> cuposCredito = {
25     {"mbarraganz", 10000000}
```

- Una función para validar el usuario y la contraseña

```
1 string seguridad(){
2     string usuario; string contraseña; int intento = 1;
3     cout << "Por favor, ingrese su usuario y contraseña para continuar:" << endl;
4     while (intento < 6){
5         cout << "\t - Usuario: "; cin >> usuario;
6         cout << "\t - Contraseña: "; cin >> contraseña;
7         if (usuarios.find(usuario) != usuarios.end() && usuarios[usuario] == contraseña){
8             intento = 6;
9         } else {
10             cout << "\033[33m\nUsuario o contraseña incorrectos. Intente de nuevo" << endl;
11             cout << "Quedan " << 5 - intento << " intentos restantes.\n\033[0m" << endl;
12             ++ intento;
13             usuario = "\033[31mUsuario o contraseña incorrectos\033[0m";
14         }
15     }
```

```
16 return usuario;
17 }
```

Esta función revisa constantemente si el usuario y la contraseña coinciden con los usuarios y contraseña guardados y da hasta cinco intentos para que el usuario ingrese los correctos.



- Una función para consultar el saldo

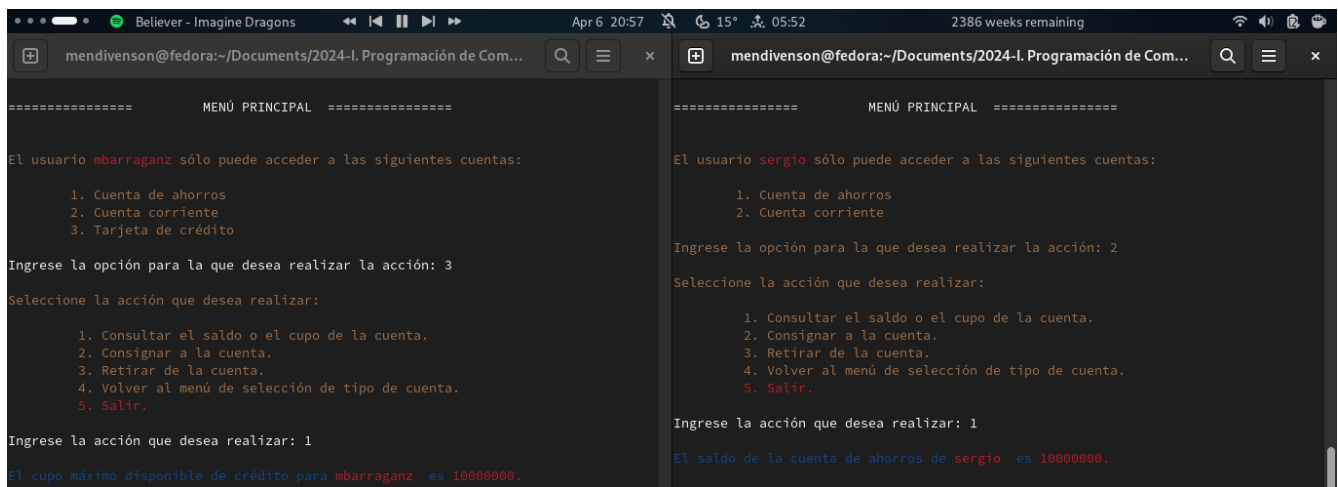
```
1 int TipoCuenta;
2 cout << "\n\033[33mEl usuario \033[31m" << usuario << " \033[33msólo puede acceder a las siguientes cuentas:\n"
<< endl;
3 if (saldosAhorros.find(usuario) != saldosAhorros.end()){
4     cout << "\t1. Cuenta de ahorros" << endl;
5 }
6
7 if (saldosCorriente.find(usuario) != saldosCorriente.end()){
8     cout << "\t2. Cuenta corriente" << endl;
9 }
10
11 if (cuposCredito.find(usuario) != cuposCredito.end()){
12     cout << "\t3. Tarjeta de crédito \033[0m" << endl;
13 }
14
15 cout << "\nIngrese la opción de la que desea conocer información: ";
16 cin >> TipoCuenta;
17 return TipoCuenta;
18 }
19
20 long consultar(int cuenta, string usuario){
21     switch(cuenta){
22     case 1:
23         cout << " \033[34m\nEl saldo de la cuenta de ahorros de \033[31m" << usuario << " \033[34m es \033[31m" <<
saldosAhorros[usuario] << " \033[31m.\033[0m" << endl;
24         return saldosAhorros[usuario];
25         break;
26     case 2:
27         cout << " \033[34m\nEl saldo de la cuenta de ahorros de \033[31m" << usuario << " \033[34m es \033[31m" <<
saldosCorriente[usuario] << " \033[31m.\033[0m" << endl;
28         saldosCorriente[usuario];
```

```

29 break;
30 case 3:
31 cout << "\033[34m\nEl cupo máximo disponible de crédito para \033[31m" << usuario << " \033[34m es \033[31m" << cuposCredito[usuario] << "\033[31m.\033[0m" << endl;
32 return cuposCredito[usuario];
33 break;
34 default: {
35 cout << "\033[31m ha seleccionado una opción incorrecta.\033[0m" << endl;
36 return 0;
37 }
38 }
39 };

```

Esta función funciona junto con otra función que imprime y guarda el tipo de cuenta (Solamente aquellas a las que el usuario está inscrito) para un usuario en específico.



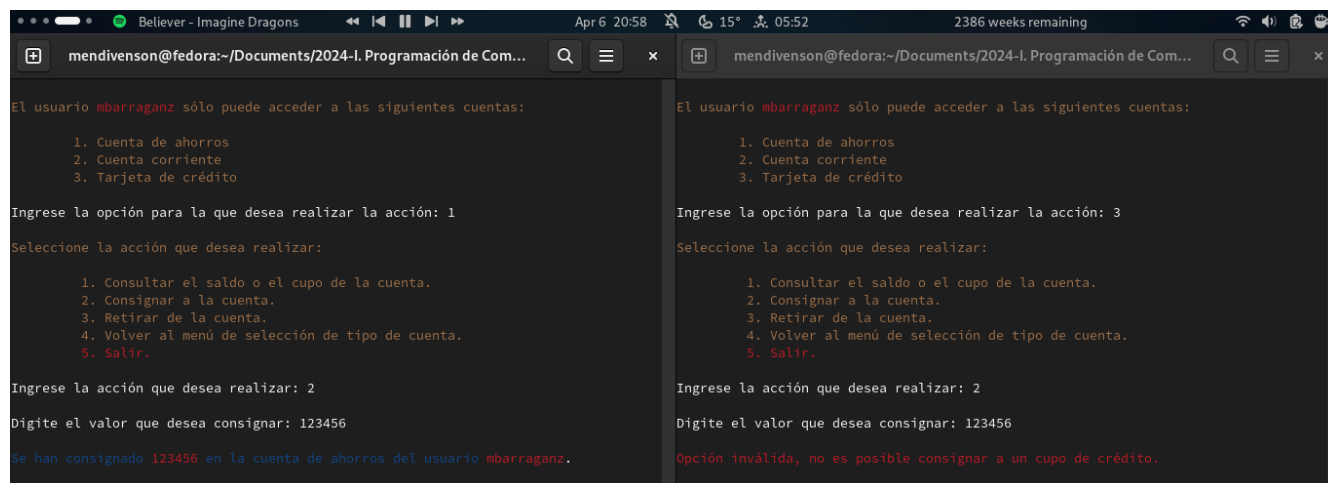
- Una función para consignar

```

1 void consignar(int cuenta, string usuario, long valor){
2 switch (cuenta){
3 case 1:
4 saldosAhorros[usuario] += valor;
5 cout << "\033[34m\nSe han consignado \033[31m" << valor << "\033[34m en la cuenta de ahorros del usuario \033[31m" << usuario << "\033[0m." << endl;
6 break;
7 case 2:
8 saldosCorriente[usuario] += valor;
9 cout << "\033[34m\nSe han consignado \033[31m" << valor << "\033[34m en la cuenta corriente del usuario \033[31m" << usuario << "\033[0m." << endl;
10 break;
11 case 3:
12 cout << "\033[31m\nOpción inválida, no es posible consignar a un cupo de crédito.\033[0m" << endl;
13 break;
14 default:
15 cout << "\033[31m\nOpción inválida.\033[0m" << endl;
16 break;
17 }
18 }

```

Esta función verifica con la información de en qué cuenta se desea consignar si se desea consignar en una cuenta de cupo de crédito para no permitir la consignación.



```
El usuario mbarraganz sólo puede acceder a las siguientes cuentas:
1. Cuenta de ahorros
2. Cuenta corriente
3. Tarjeta de crédito

Ingrese la opción para la que desea realizar la acción: 1
Seleccione la acción que desea realizar:
1. Consultar el saldo o el cupo de la cuenta.
2. Consignar a la cuenta.
3. Retirar de la cuenta.
4. Volver al menú de selección de tipo de cuenta.
5. Salir.

Ingrese la acción que desea realizar: 2
Digite el valor que desea consignar: 123456
Se han consignado 123456 en la cuenta de ahorros del usuario mbarraganz.

El usuario mbarraganz sólo puede acceder a las siguientes cuentas:
1. Cuenta de ahorros
2. Cuenta corriente
3. Tarjeta de crédito

Ingrese la opción para la que desea realizar la acción: 3
Seleccione la acción que desea realizar:
1. Consultar el saldo o el cupo de la cuenta.
2. Consignar a la cuenta.
3. Retirar de la cuenta.
4. Volver al menú de selección de tipo de cuenta.
5. Salir.

Ingrese la acción que desea realizar: 2
Digite el valor que desea consignar: 123456
Opción inválida, no es posible consignar a un cupo de crédito.
```

- Una función para contar los billetes

```
1 void cuantos(long valor){
2     int billetes;
3     cout << "\033[90mEl dinero le será entregado en: " << endl;
4     if ((valor / 100000) > 1){
5         billetes = floor(valor/100000);
6         valor = valor - billetes * 100000;
7         cout << "\t - " << billetes << " billetes de cien mil pesos." << endl;
8     }
9     if ((valor / 50000) > 1){
10        billetes = floor(valor/50000);
11        valor = valor - billetes * 50000;
12        cout << "\t - " << billetes << " billetes de cincuenta mil pesos." << endl;
13    }
14    if ((valor / 20000) > 1){
15        billetes = floor(valor/20000);
16        valor = valor - billetes * 20000;
17        cout << "\t - " << billetes << " billetes de veinte mil pesos." << endl;
18    }
19    if ((valor / 10000) > 1){
20        billetes = floor(valor/10000);
21        valor = valor - billetes * 10000;
22        cout << "\t - " << billetes << " billetes de diez mil pesos." << endl;
23    }
24    if ((valor / 5000) > 1){
25        billetes = floor(valor/5000);
26        valor = valor - billetes * 5000;
27        cout << "\t - " << billetes << " billetes de cinco mil pesos." << endl;
28    }
29    if ((valor / 2000) > 1){
30        billetes = floor(valor/2000);
31        valor = valor - billetes * 2000;
32        cout << "\t - " << billetes << " billetes de dos mil pesos." << endl;
33    }
```

```

34 if ((valor / 1000) > 1){
35     billetes = floor(valor/1000);
36     valor = valor - billetes * 1000;
37     cout << "\t - " << billetes << " billetes de mil pesos." << endl;
38 }
39 cout << "El restante " << valor << " se le entregará en monedas.\033[0m" << endl;
40 }

```

Esta función va restando los valores de los billetes que sean posibles hasta llegar a los billetes de mil pesos dando un listado final de cuántos billetes de cada denominación hacen falta para entregar el dinero completo al usuario.

```

La transacción ha sido cancelada
El dinero le será entregado en:
    - 1 billetes de cien mil pesos.
    - 1 billetes de veinte mil pesos.
    - 1 billetes de dos mil pesos.
    - 1 billetes de mil pesos.
El restante 0 se le entregará en monedas.

```

- Una función para retirar que considere si se hace en la cuenta de ahorros, en la cuenta corriente o en la cuenta de crédito

```

1 void retirar(int cuenta, string usuario, long valor){
2     long ValorActual;
3     switch(cuenta){
4     case 1:
5         ValorActual = saldosAhorros[usuario];
6         if (valor <= ValorActual && valor > 0){
7             saldosAhorros[usuario] = saldosAhorros[usuario] - valor;
8             cout << "\033[34m\nLa transacción se ha realizado\033[0m" << endl;
9             cuantos(valor);
10        }
11    else {
12        cout << "\033[31m\nOpción inválida. No hay esa cantidad de dinero en la cuenta\033[0m" << endl;
13    }
14    break;
15    case 2:
16        ValorActual = saldosCorriente[usuario];
17        if (valor <= ValorActual && valor > 0){
18            cout << "\033[34m\nLa transacción se ha realizado\033[0m" << endl;
19            cuantos(valor);
20            saldosCorriente[usuario] = saldosCorriente[usuario] - valor;
21        } else { cout << "\033[31m\nOpción inválida. No hay esa cantidad de dinero en la cuenta\033[0m" << endl;}
22        break;
23    case 3:
24        ValorActual = cuposCredito[usuario];
25        if (valor <= ValorActual && valor > 0){
26            cout << "\n\033[31m\n Cuidado. Está a punto de realizar un préstamo\033[0m\n" << endl;
27            string word;

```

```

28     cout << "Si está seguro que desea continuar con la transacción escriba Y, de lo contrario se
cancelará la transacción. ";
29     cin >> word;
30     if (word == "Y" || word == "YES" || word == "yes" || word == "Yes"){
31         cout << "\033[34m\nEl crédito se ha realizado\033[0m" << endl;
32         cuposCredito[usuario] = cuposCredito[usuario] - valor;
33         cuantos(valor);
34     } else {
35         cout << "\033[34m\nLa transacción ha sido cancelada\033[0m" << endl;
36     }
37 } else { cout << "\033[31m\nOpción inválida. El usuario no cuenta con el suficiente cupo de
crédito\033[0m" << endl;}
38 break;
39 }
40 }

```

Esta función revisa los montos que se desean retirar y de qué cuenta se desean retirar. De forma que si el monto que se desea retirar supera el dinero disponible no se permita la transacción y usa la función cuantos definida anteriormente para saber qué billetes de qué denominación se entregarán.

```

Legend - The Score Apr 6 21:15 15° 05:52 2386 weeks remaining
mendivenson@fedora:~/Documents/2024-I. Programación de Com...
El usuario mbarraganz sólo puede acceder a las siguientes cuentas:
1. Cuenta de ahorros
2. Cuenta corriente
3. Tarjeta de crédito
Ingrese la opción para la que desea realizar la acción: 1
Seleccione la acción que desea realizar:
1. Consultar el saldo o el cupo de la cuenta.
2. Consignar a la cuenta.
3. Retirar de la cuenta.
4. Volver al menú de selección de tipo de cuenta.
5. Salir.
Ingrese la acción que desea realizar: 3
Digite el valor que desea retirar: 12520000
La transacción se ha realizado
El dinero le será entregado en:
- 125 billetes de cien mil pesos.
- 1 billetes de veinte mil pesos.
El restante 0 se le entregará en monedas.
Seleccione la acción que desea realizar:
1. Consultar el saldo o el cupo de la cuenta.
2. Consignar a la cuenta.
3. Retirar de la cuenta.
4. Volver al menú de selección de tipo de cuenta.

```

- [Extrachallenge] Sin usar la estructura “While” ni “for” ni “If else”, es decir solo usando funciones hacer que el programa pueda volver al menú principal siempre que finalice cualquier transacción y funcione correctamente.

```

1 void menu(string usuario, int Cuenta){
2
3 int Accion; long valor;
4 if (Cuenta == 0){

```

```

5  Cuenta = tipoCuenta(usuario);
6  }
7
8  // Elección de la acción a realizar:
9  cout << "\n\033[33mSeleccione la acción que desea realizar: " << endl;
10 cout << "\n\t 1. Consultar el saldo o el cupo de la cuenta." << endl;
11 cout << "\t 2. Consignar a la cuenta." << endl;
12 cout << "\t 3. Retirar de la cuenta." << endl;
13 cout << "\t 4. Volver al menú de selección de tipo de cuenta." << endl;
14 cout << "\t \033[31m5. Salir. \033[0m\n" << endl;
15 cout << "Ingrese la acción que desea realizar: ";
16 cin >> Accion;
17
18
19 switch(Accion){
20 case 1:
21     consultar(Cuenta, usuario);
22     break;
23 case 2:
24     cout << "\nDigite el valor que desea consignar: ";
25     cin >> valor;
26     consignar(Cuenta, usuario, valor);
27     break;
28 case 3:
29     cout << "Digite el valor que desea retirar: ";
30     cin >> valor;
31     retirar(Cuenta, usuario, valor);
32     break;
33 case 4:
34     Cuenta = 0;
35     std::cout << "\033[2]\033[1;1H";
36     break;
37 case 5:
38     cout << "\033[32m\n ===== Gracias por preferirnos, qué tenga buen día ===== \
\033[0m\n" << endl;
39     return;
40     break;
41 default:
42     cout << "\033[31m\n Opción inválida.\033[0m" << endl;
43     break;
44 }
45 menu(usuario, Cuenta);
46 }
47
48
49 int main() {

```

```

50
51 std::cout << "\033[2]\033[1;1H";
52 cout << "\033[32m\n\n===== \t BIENVENIDO A ESTE, SU BANCO\t
===== \033[0m\n" << endl;
53 string usuario = seguridad(); // Usuario y contraseña
54 std::cout << "\033[2]\033[1;1H";
55
56 if (usuario == "\033[31mUsuario o contraseña incorrectos\033[0m"){
57     cout << usuario << endl;
58     return 0;
59 } else {
60     cout << "\n===== \t MENÚ PRINCIPAL\t =====\n" << endl;
61     menu(usuario, 0);
62 }
63 return 0;
64 }

```

La función responsable del bucle es la función llamada menú. A grandes rasgos, imprime el menú de selección de tipo de cuenta y a continuación da 5 opciones para cada una de las cuentas. El bucle se generará debido a que la última línea de la función es un llamado de nuevo a la función menú y además para la opción 4 de escoger otro tipo de cuenta con el mismo usuario, se limpia la consola con la línea `std::cout << "\033[2]\033[1;1H"`.

```

El usuario mbarraganz sólo puede acceder a las siguientes cuentas:
1. Cuenta de ahorros
2. Cuenta corriente
3. Tarjeta de crédito

Ingrese la opción para la que desea realizar la acción: 3
Seleccione la acción que desea realizar:
1. Consultar el saldo o el cupo de la cuenta.
2. Consignar a la cuenta.
3. Retirar de la cuenta.
4. Volver al menú de selección de tipo de cuenta.
5. Salir.

Ingrese la acción que desea realizar: 1
El cupo máximo disponible de crédito para mbarraganz es 10000000.
Seleccione la acción que desea realizar:
1. Consultar el saldo o el cupo de la cuenta.
2. Consignar a la cuenta.
3. Retirar de la cuenta.
4. Volver al menú de selección de tipo de cuenta.
5. Salir.

Ingrese la acción que desea realizar: 2
Digite el valor que desea consignar: 123
Opción inválida, no es posible consignar a un cupo de crédito.
Seleccione la acción que desea realizar:
1. Consultar el saldo o el cupo de la cuenta.
2. Consignar a la cuenta.
3. Retirar de la cuenta.
4. Volver al menú de selección de tipo de cuenta.
5. Salir.

Ingrese la acción que desea realizar:

```