

1. Servidor

| Servicios | | |
|---|--------------|----------------|
| Cliente | Servidor | Notificaciones |
| "0/" Desconexión | | |
| "1/nombre" Log in, Conectarse | | |
| "2/" Consultar Saldo | "1/Cantidad" | |
| "3/Ingreso" Ingresar Dinero | | |
| "4/Nombre/Cantidad" "Admin" hace cargo a cliente | | |
| "5/Cantidad" "Admin" carga la cantidad a pagar para todos los clientes conectados | | "2/" |

2. pthread_mutex_t mutex = PTHREAD_MUTEX_INITIALIZER;
ListaClientes milistaclientes;

```
3. void *AtenderCliente(void *Socket){
    char peticion[200];
    char notificacion[200];
    char respuesta[200];
    int ret;

    char nom[20];
    int sock_conn = *(int *socket)
    int terminar=0;
    while(!terminar) {
        ret = read(sock_conn,peticion,sizeof(peticion));
        peticion[ret]="0\0";

        char *p = strtok(peticion, "/");
        int codigo = atoi(p);
        if(codigo==0){
            //Desconexió

            pthread_mutex_lock(&mutex);
            TomarNotaClienteDesconectado(&milistaclientes,sock_conn);
            pthread_mutex_unlock(&mutex);
            terminar = 1;
        }
        else if(codigo == 1){
            p=strtok(NULL,"/");
            strcpy(nom,p);
            pthread_mutex_lock(&milistaclientes,nom,sock_conn);
```

```

        pthread_mutex_unlock(&mutex);
    }
    else if(codigo ==2){
        int saldo = DameSaldoCliente(&milistaclientes,nom);
        sprintf(respuesta,"1/%d",saldo);
        write(sock_conn,respuesta);
    }
    else if(codigo==3){
        p=strtok(NULL,"/");
        int cantidad = atoi(p);
        pthread_mutex_lock(&mutex);
        HacerIngreso(&milistaclientes,nom);
        pthread_mutex_unlock(&mutex);
    }
    else if(codigo==4){
        p=strtok(NULL,"/");
        char cliente[20];
        strcpy(cliente,p);
        p=strtok(NULL,"/");
        int cargo = atoi(p);
        pthread_mutex_lock(&mutex);
        HacerCargoCliente(&milistaclientes,cargo,cliente);
        pthread_mutex_unlock(&mutex);
    }
    else if(codigo==5){
        p=strtok(NULL,"/");
        int comision = atoi(p);
        pthread_mutex_lock(&mutex);
        HacerComision(&milistaclientes,comision);
        pthread_mutex_unlock(&mutex);

        int conectados = DameSocketsConectados(&milistaclientes,socket);
        strcpy(notificacion,"2/");
        for(int i=0;i<num;i++)
            write(sockets[i].notificacion,strlen(notificacion));
    }

}

Close(sock_conn);
} //AtenderCliente

```

```

public partial class Form1:Form
{
    Socket server;
    Thread atender;

    public Form1()
    {
        InitializeComponent();
    }
    private void Conectarbtn_Click(object sender, EventArgs e)
    {
        string mensaje = "1/"+nombreBox.text;
        byte[] msg = System.Text.Encoding.ASCII.GetBytes(mensaje);
        server.Send(msg);
    }
    private void SaldoBtn_Click(object sender, EventArgs e)
    {
        string mensaje = "2/";
        byte[] msg = System.Text.Encoding.ASCII.GetBytes(mensaje);
        server.Send(msg);
    }
    private void IngresoBtn_Click(object sender, EventArgs e)
    {
        string mensaje = "3/"+cantidad_i_Box.Text;
        byte[] msg = System.Text.Encoding.ASCII.GetBytes(mensaje);
        server.Send(msg);
    }
    private void CargoBtn_Click(object sender, EventArgs e)
    {
        string mensaje = "4/"+clienteBox.Text+"/"+cantidad_C_Box.Text;
        byte[] msg = System.Text.Encoding.ASCII.GetBytes(mensaje);
        server.Send(msg);
    }
    private void ComisionBtn_Click(object sender, EventArgs e)
    {
        string mensaje = "5/"+comisionBox.Text;
        byte[] msg = System.Text.Encoding.ASCII.GetBytes(mensaje);
        server.Send(msg);
    }
    private void EntrarBancoBtn_Click()
    {
        //al que deseamos conectarnos
        IPAddress direc = IPAddress.Parse("XXX");
        IPEndPoint ipep = new IPEndPoint(direc, 8007);

        //Creamos el socket
        server = new Socket(AddressFamily.InterNetwork, SocketType.Stream, ProtocolType.Tcp);
        try
        {
            server.Connect(ipep); //Intentamos conectar el socket
            ThreadStart ts = delegate { AtenderServidor(); };
            atender = new Thread(ts);
            atender.start();
        }
    }
}

```

```

        MessageBox.Show("Conectado");

    }
    catch (SocketException ex)
    {
        //Si hay excepcion imprimimos error y salimos del programa con return
        MessageBox.Show("No he podido conectar con el servidor");
        return;
    }
    }
    private void AtenderServidor()
    {
        while(true)
        {
            int codigo
            byte msg = newbyte[80];
            server.Recive(msg);
            string mensaje =
system.text.Encoding.ASCII.GetString(msg).Split('\0')[0];
            string trozos = mensaje.split('/');
            codigo = Convert.ToInt32(trozos[0]);
            switch(codigo){
                case 1:
                    this.Invoke(new Action(()=>
                    {
                        respuestaLbl.text = trozos[1];
                    }));
                    break;
                case 2:
                    this.Invoke(new Action(()=>
                    {
                        notificacionLbl.text = "Se ha efectuado un
carga";
                    }));
                    break;
            }
        }
    }
}

```