

1 Ejercicio 1

En este ejercicio se implemento el uso del timestamp en segundos y microsegundos del lado del servidor.

```

    }
    Respuesta resp(7200);
    cout << "Servidor iniciado...\n";
    while (true) {
        struct mensaje *msj = resp.getRequest();
        if(msj != NULL) {
            char msjRecib[1024];
            memcpy(msjRecib,&msj->arguments,34);

            timeval actual;
            actual.tv_sec = 0;
            actual.tv_usec = 0;

            gettimeofday(&actual,NULL);
            cout <<"Server.cpp:"<< actual.tv_sec <<" : "<< actual.tv_usec<< endl;
            string regs;
            regs += to_string(actual.tv_sec)+"@"+ to_string(actual.tv_usec);
            write(destino,msjRecib, 34);
            write(destino,regs.c_str(), regs.length());
            write(destino,"\n", 1);

            resp.sendReply((char *)&actual);
        }else{//si es duplicado
            timeval actual;
            actual.tv_sec = 0;
            actual.tv_usec = 0;
            cout << "Telefono ya registrado" << endl;
            resp.sendReply((char *)&actual);
        }
    }
    close(destino);
}

```

Figure 1: Modificacion del servidor

```

using namespace std;
int randNum();
int main(int argc, char *argv[]){
    //El cliente debe recibir el parametro n en la línea de comandos...
    if(argc != 4){
        cout<<"Forma de usa: nombre_programa ip nombre_archivo num_registros"<<endl;
        exit(0);
    }
    ifstream archivo(argv[2], ios::binary);
    char cadena[256];
    string s;
    int regactual=0;
    int res = 0;
    //leerá n registros de un archivo...
    int n = atoi(argv[3]);
    while (regactual < n){
        //irá leyendo cada registro del archivo...
        getline(archivo,s);
        memcpy(cadena,s.c_str(),s.size());
        //cout<<cadena<<endl;//comprobación

        // y los manda en mensajes UDP(un registro en cada mensaje UDP) hacia el servidor.;
        Solicitud s;
        timeval res;
        memcpy(&res, s.doOperation(argv[1], 7200, 1, (char *)&cadena), sizeof(timeval));

        cout <<"Client.cpp:"<<res.tv_sec<<" : "<< res.tv_usec << endl;
        regactual++;
    }
    return 0;
}

```

Figure 2: Modificacion del cliente

1.1 Preguntas Ejercicio 1

¿Sería aceptable en esta aplicación utilizar un timestamp que solo incluya la variable de segundos `time_t`? ¿por qué?

R: NO, porque dependiendo el numero de votos esta variable se va a repetir a mas de un votante por lo que es necesario mostrar ambas variables, segundos y microsegundos.

¿Existe la posibilidad de que el servidor devuelva un timestamp repetido a dos votantes distintos?

R: En cuanto a la variable segundos SI, pero con respecto a la variable microsegundos NO.

¿Por qué sería necesario en esta aplicación distribuida el uso de la función `fsync()` después de ejecutar la función `write()` en el lado del servidor?

R: Para que el programa descargue sus buffers a los medios físicos terminando la verificación.

2 Ejercicio 2

Se debe de realizar una verificación del registro del celulares, para realizar las búsquedas utilice la clase vector de STL y el método `binary_search`.

Se creo una archivo de cabecera que contiene una estructura con los datos del celular, curp, partido, una cadena construida de estos datos y un operador booleano.

```
1 #include <string>
2 #include <iostream>
3 using namespace std;
4 struct celulares{
5     char celular[11];
6     char CURP[19];
7     char partido[4];
8     string toString()
9     {
10         return string(celular)+"^@"+string(CURP)+"^@"+string(partido);
11     }
12     bool operator<(const celulares & other) const
13     {
14         return atoi(celular) < atoi(other.celular);
15     } //
16 };
```

Se modifica el servidor de nuevo agregando el vector, y el metodo de busqueda `binary_search`.

```

Respuesta resp(7200);
vector<int> celulares;
cout << "Servidor iniciado....\n";
while (true) {
    struct mensaje *msj = resp.getRequest();
    if(msj != NULL) {
        char msjRecib[1024];
        struct celulares re;
        memcpy(&re,&msj->arguments,sizeof(struct celulares));
        timeval actual;
        actual.tv_sec = 0;
        actual.tv_usec = 0;

        //Ejercicio 2
        if(!binary_search(celulares.begin(), celulares.end(),atoi(re.celular))){
            celulares.push_back(atoi(re.celular));
            sort(celulares.begin(), celulares.end());
            gettimeofday(&actual,NULL); //Ejercicio1
            //cout <<"Server.cpp:"<< actual.tv_sec <<" : "<< actual.tv_usec<< endl;
            string regs = re.toString();
            regs += "^@"+to_string(actual.tv_sec)+"^@"+ to_string(actual.tv_usec);
            //write(destino,msjRecib, 34);
            write(destino,regs.c_str(), regs.length());
            write(destino,"\n", 1);

            resp.sendReply((char *)&actual);
        }else{//si es duplicado
            timeval actual;
            actual.tv_sec = 0;
            actual.tv_usec = 0;
            cout << "Telefono ya registrado" << endl;
            resp.sendReply((char *)&actual);
        }
    }
}

```

Figure 3: binary_search

3 Prueba con 50

Aunque en esta practica no fueron requeridas, se muestra una prueba con 50 registros simplemente para detallar el funcionamiento de los programas, se eliminaron las impresiones en pantalla para realizar unas pruebas de 7 mil y de 70 mil registros que estaran disponibles en el zip con los archivos enviados.

The image shows two terminal windows. The top window is titled 'pach@pach-Inspiron-3558: ~/SistemasDistribuidos-ESCOM/SDEP_Seguridad/Servidor'. It shows the command './server op' being executed, resulting in the output 'Servidor iniciado....'. The bottom window is titled 'pach@pach-Inspiron-3558: ~/SistemasDistribuidos-ESCOM/SDEP_Seguridad/Cliente'. It shows the command './client 127.0.0.1 70mil 50' being executed, with the output 'lient 127.0.0.1 70mil 50' visible.

```

pach@pach-Inspiron-3558: ~/SistemasDistribuidos-ESCOM/SDEP_Seguridad/Servidor
Archivo Editar Ver Buscar Terminal Ayuda
pach@pach-Inspiron-3558:~/SistemasDistribuidos-ESCOM/SDEP_Seguridad/Servidor$ ./server op
Servidor iniciado....

pach@pach-Inspiron-3558: ~/SistemasDistribuidos-ESCOM/SDEP_Seguridad/Cliente
Archivo Editar Ver Buscar Terminal Ayuda
pach@pach-Inspiron-3558:~/SistemasDistribuidos-ESCOM/SDEP_Seguridad/Cliente$ ./client 127.0.0.1 70mil 50
lient 127.0.0.1 70mil 50
pach@pach-Inspiron-3558:~/SistemasDistribuidos-ESCOM/SDEP_Seguridad/Cliente$

```

Figure 4: Correr los programas

```

Mensaje enviado
Server.cpp:1588883282 : 338664
Mensaje enviado
Server.cpp:1588883282 : 338802
Mensaje enviado
Server.cpp:1588883282 : 338981
Mensaje enviado
Server.cpp:1588883282 : 339159
Mensaje enviado
Server.cpp:1588883282 : 339304
Mensaje enviado
Server.cpp:1588883282 : 339441
Mensaje enviado
Server.cpp:1588883282 : 339577
Mensaje enviado
Server.cpp:1588883282 : 339711
Mensaje enviado
Server.cpp:1588883282 : 339876
Mensaje enviado
Server.cpp:1588883282 : 339876

Direccion: 127.0.0.1
Puerto: 7200
Client.cpp:1588883282 : 339304
ContRequest: 46
Mensaje recibido
Direccion: 127.0.0.1
Puerto: 7200
Client.cpp:1588883282 : 339441
ContRequest: 47
Mensaje recibido
Direccion: 127.0.0.1
Puerto: 7200
Client.cpp:1588883282 : 339577
ContRequest: 48
Mensaje recibido
Direccion: 127.0.0.1
Puerto: 7200
Client.cpp:1588883282 : 339711
ContRequest: 49
Mensaje recibido
Direccion: 127.0.0.1
Puerto: 7200
Client.cpp:1588883282 : 339876

```

Figure 5: prueba de 50

```

GNU nano 2.9.3 op
5504296321^@HJVG462869MCSEDU48^@PRI^@1588883282@336438
5504291606^@SCEP662113MVZQDR78^@P_T^@1588883282@336662
5504295517^@RDAC422406HDGVSP58^@MVC^@1588883282@336979
5504289482^@VFPL542983HBCVNB32^@MVC^@1588883282@337291
5504291162^@CYLR456570MBSVUT20^@MOR^@1588883282@337491
5504292790^@YNCL421454MCLJBG32^@PRD^@1588883282@337712
5504293588^@VIRA550084HSLCRX65^@PRI^@1588883282@337916
5504293806^@BENE718625HZSGGA58^@MVC^@1588883282@338070
5504292783^@HTVH259858HMCHYX41^@PNL^@1588883282@338238
5504290138^@XVDD903301MMNOFW81^@MOR^@1588883282@338387
5504293874^@IFGE079924HCCUBK12^@PAN^@1588883282@338526
5504293855^@HVGW751972HJCUKP74^@MVC^@1588883282@338664
5504292895^@FJAR570943HOCHVG07^@PRI^@1588883282@338802
5504292374^@QWLP790092HBCNBI58^@PRD^@1588883282@338981
5504289736^@EWMQ905811HGTLSI61^@P_T^@1588883282@339159
5504290660^@IFFN963585MJCHES95^@PES^@1588883282@339304
5504294175^@CFJN837931MSPQWK96^@MVC^@1588883282@339441
5504291232^@NGGC646962HDFBUN13^@PES^@1588883282@339577
5504292943^@DLFW078833HCHECE13^@VDE^@1588883282@339711
5504296134^@NRNR665666MBCKMV52^@PAN^@1588883282@339876

```

Figure 6: archivo op