

# 1 Anexos

## 1.1. Código

### 1.1.1. Script

```
#!/bin/sh

case "$1" in
start)
echo "Iniciando servicio... "
# Aquí comando a ejecutar para arrancar el servicio
    cold /etc/cold.properties
;;
stop)
# Aquí comando a ejecutar para detener el servicio
    pid='pidof cold'
    if [ -z $pid ]; then
        echo "No se puede detener el servicio, el
            demonio no esta corriendo"
    else
        echo "Deteniendo servicio..."
        kill -TERM $pid
    fi
#TERM (15) senhal de detenerse
;;
restart)
# Aquí comando a ejecutar para reiniciar el servicio
    pid='pidof cold'
    if [ -z $pid ]; then
        echo "No se puede reiniciar el servicio, el
            demonio no esta corriendo"
    else
        echo "Reiniciando servicio..."
        kill -TERM $pid
        cold /etc/cold.properties
    fi
#TERM (15)
```

```
;;
reload)
# Aquí comando a ejecutar para recargar el servicio
pid='pidof cold'
if [ -z $pid ]; then
    echo "No se puede recargar.. el demonio no
        esta corriendo.."
else
    echo "Recargando archivo de configuracion del
        servicio..."
    kill -HUP $pid
fi
#HUP (1)

;;
status)
pid='pidof cold'
if [ -z $pid ]; then
    echo "El demonio no esta activo.."
else
    echo "El demonio esta activo PID= $pid"

fi

;;
*)
echo "Modo de empleo: sudo service coldaemon {start|stop|
    restart|reload|status}"
exit 1
;;
esac
exit 0
```

### 1.1.2. coldaemon.h

```
#include <sys/types.h>
#include <sys/stat.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <fcntl.h>
#include <errno.h>
#include <unistd.h>
#include <time.h>
#include <syslog.h>
#include <signal.h>
```

```
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <string.h>
#include <postgresql/libpq-fe.h>
// #include <netbd.h>

/*

    Todas las variables en minúsculas, se usa guion bajo si
    es necesario
    ej; var, var_muy_larga
    Todos los define en mayúsculas
    ej: #define MACRO valor_macro
    todos los nombres de función en minúsculas

*/

// Definiciones
#define STR_LEN 512
#define PKG_LEN 16384

// Estructuras de datos
struct thread_list
{
    pthread_t hilo;
    int thread_index;
    struct thread_list * siguiente;
};

typedef struct
{
    int threads; // cantidad de hilos (thread manager)
    int puerto; // puerto de escucha
    char acl[STR_LEN]; // lista de control de acceso
    char log[STR_LEN]; // bitácora del demonio
    int timeout; // tiempo límite de espera
    struct thread_list * lista_hilo; // hilos en ejecución
    int thread_index; // identificador de hilo (thread
        worker)
    int socket_descriptor; // descriptor de socket (thread
        worker)
    struct sockaddr_in socket; // estructura socket (thread
        worker)
} thread_arg;
```

```

struct parameters
{
    thread_arg arg;
    struct parameters * siguiente;
};

typedef struct
{
    char *codser; //3 DIGITOS codigo de servicio
    int numtran; //6 DIGITOS numero de transaccion
    char *fechahora; //14 DIGITOS fecha y hora de
        transaccion
    char *tipofact; // 3 DIGITOS tipo de factura
    char *comprobante; //11 DIGITOS numero de comprobante
    long int monto; //12 DIGITOS monto de la factura
    char *vencimiento; //8 DIGITOS fecha de vencimiento
    int verificador; //1 DIGITOS digito verificador
    char *prefijo; //4 DIGITOS prefijo
    char *numero; //7 DIGITOS numero telefonico
    char *nummed; //15 DIGITOS numero de medidor
    char *abonado; //9 DIGITOS numero de abonado
    char *mensaje; //20 CARACTERES DE MENSAJE
}SERVICIO;

int ready;
pthread_mutex_t lock;

// Biblioteca de Funciones de:
// Hash, postgres, semaforos

#define OK 0
#define ARGUMENTOS_INVALIDOS 1
#define NO_CONFIG_FILE 2
#define INVALID_CONFIG_FILE 3
#define CANT_OPEN_ACL 13
#define LOG_ERROR 11
#define SOCK_DESCRIPTOR_ERROR 4
#define BINDING_ERROR 5
#define LISTENNING_ERROR 6
#define ACCEPT_CONNECTION_ERROR 7
#define CANT_FORK 8
#define SESSION_ERROR 9
#define CHDIR_ERROR 10
#define CONFIG_ERROR 12
#define NULL_THREAD 14
#define CANT_CLOSE_SOCKET 15

```

```
#define DB_EXIT_NICELY 20

#define INVALID_COD_SERV 30
#define INVALID_YEAR 31
#define INVALID_DAY 32
#define INVALID_MONTH 33
#define INVALID_HOUR 34
#define INVALID_MIN 35
#define INVALID_SEC 36

/*
    acl_file es un puntero al nombre del fichero que tiene
    los datos de autenticación
    usuario es un puntero al nombre usuario
    clave es un puntero a la clave
    la función retorna 0 si la autenticación es exitosa y se
    debe definir códigos
    de error para cada caso de error
*/
//char authentication(char * acl_file, char * usuario, char *
    clave);

/*
    config_parser lee un archivo de configuración y
    establece los parámetros del demonio a partir del
    mismo
    retorna 0 si no hubo errores, establecer sus códigos de
    error en otro caso
    config_file, puntero al nombre del archivo de
    configuración
    puerto, threads, timeout, logpath, logfile, parámetros
    del demonio.
*/
int config_parser (char * config_file, int * puerto, int *
    threads, int * timeout, char ** logpath, char ** logfile,
    char ** aclpath, char ** aclfile);

/*
    EL CORE DAEMON
*/
void * coredaemon(void * argumento);
// Helper para limpiar el #015 que telnet envía como Retorno de
    Línea
void limpiar_telnet(char * cadena);
```

```

/*
    Parser de patrones de entrada
*/
char col_parser (SERVICIO *servicio , char * patron , int log_fd);
char rev_parser (SERVICIO *reversa , char * patron);

/*
    Módulo de Base de Datos
*/
int db_module(char * operacion , SERVICIO serv , char * usuario ,
    int log_fd , char * resp);

/*
    UTILERIA GENERAL
*/
void writelog(int log_fd , const char * mensaje);
void thread_add(struct thread_list **lista , int index);
pthread_t * thread_get(struct thread_list *lista , int index);
void thread_del(struct thread_list **lista , int index);
unsigned long hash( char * str);
char authentication (char * acl_file , char * user , unsigned long
    pass_buscado);

/*
    Administrador de Hilos
*/
void * thread_manager(void * argumento);

/*
    Módulo de Configuración
*/
char config_module(char * config_file , thread_arg * argumento);
void dbg_print_thread_arg(thread_arg * argumento);

```

### 1.1.3. coldaemon.c