

## Proyecto II - Servidor

Generated by Doxygen 1.8.14



# Contents

|          |  |          |
|----------|--|----------|
| <b>1</b> | <b>File Index</b>                        | <b>1</b> |
| 1.1      | File List . . . . .                      | 1        |
| <b>2</b> | <b>File Documentation</b>                | <b>3</b> |
| 2.1      | clnt_hand.c File Reference . . . . .     | 3        |
| 2.1.1    | Detailed Description . . . . .           | 3        |
| 2.1.2    | Function Documentation . . . . .         | 3        |
| 2.1.2.1  | err_send() . . . . .                     | 3        |
| 2.1.2.2  | handle_client() . . . . .                | 4        |
| 2.2      | servidor.c File Reference . . . . .      | 4        |
| 2.2.1    | Detailed Description . . . . .           | 5        |
| 2.2.2    | Function Documentation . . . . .         | 5        |
| 2.2.2.1  | error() . . . . .                        | 5        |
| 2.2.2.2  | main() . . . . .                         | 5        |
| 2.3      | servidor.h File Reference . . . . .      | 6        |
| 2.3.1    | Detailed Description . . . . .           | 7        |
| 2.3.2    | Macro Definition Documentation . . . . . | 7        |
| 2.3.2.1  | ASK_POK . . . . .                        | 7        |
| 2.3.2.2  | BAIT . . . . .                           | 7        |
| 2.3.2.3  | BALL . . . . .                           | 7        |
| 2.3.2.4  | BUFF_SIZE . . . . .                      | 7        |
| 2.3.2.5  | ENCOUNTER . . . . .                      | 7        |
| 2.3.2.6  | END_SESSION . . . . .                    | 8        |
| 2.3.2.7  | ERROR_CODE . . . . .                     | 8        |

|          |                        |    |
|----------|------------------------|----|
| 2.3.2.8  | ESCAPE                 | 8  |
| 2.3.2.9  | MAX_POK                | 8  |
| 2.3.2.10 | NO                     | 8  |
| 2.3.2.11 | NO_TRIES               | 8  |
| 2.3.2.12 | NO_USER                | 8  |
| 2.3.2.13 | OK                     | 8  |
| 2.3.2.14 | POKEDEX                | 9  |
| 2.3.2.15 | ROCK                   | 9  |
| 2.3.2.16 | RUN                    | 9  |
| 2.3.2.17 | SERV_PORT              | 9  |
| 2.3.2.18 | SUCCESS                | 9  |
| 2.3.2.19 | TIMEOUT                | 9  |
| 2.3.2.20 | TRY_USER               | 9  |
| 2.3.2.21 | USER                   | 9  |
| 2.3.3    | Function Documentation | 9  |
| 2.3.3.1  | err_send()             | 9  |
| 2.3.3.2  | error()                | 10 |
| 2.3.3.3  | handle_client()        | 10 |

# Chapter 1

## File Index

### 1.1 File List

Here is a list of all documented files with brief descriptions:

|                             |   |   |
|-----------------------------|---|---|
| <a href="#">clnt_hand.c</a> | Definiciones/Mensajes del servidor y el cliente . . . . . | 3 |
| <a href="#">servidor.c</a>  | Programa del Servidor . . . . .                           | 4 |
| <a href="#">servidor.h</a>  | Definiciones/Mensajes del servidor y el cliente . . . . . | 6 |



## Chapter 2

# File Documentation

### 2.1 clnt\_hand.c File Reference

Definiciones/Mensajes del servidor y el cliente.

```
#include "servidor.h"
```

#### Functions

- void \* [err\\_send](#) (int clntfd, char code)  
*Envia un codigo de error al cliente.*
- void \* [handle\\_client](#) (void \*vargp)  
*Funcion que atiende al cliente.*

#### 2.1.1 Detailed Description

Definiciones/Mensajes del servidor y el cliente.

##### Date

11 Diciembre 2018 Aqui se atienden las peticiones del cliente.

#### 2.1.2 Function Documentation

##### 2.1.2.1 err\_send()

```
void* err_send (
    int clntfd,
    char code )
```

Envia un codigo de error al cliente.

**Parameters**

|               |                                   |
|---------------|-----------------------------------|
| <i>clntfd</i> | Descriptor de archivo del cliente |
| <i>code</i>   | El codigo de error                |

**Returns**

NULL?

**2.1.2.2 handle\_client()**

```
void* handle_client (
    void * vargp )
```

Funcion que atiende al cliente.

Corroborar que el usuario tenga su archivo correspondiente, obtiene los datos y comienza el ciclo de captura. Se validan las acciones del usuario, así como si se capturan o no los pokemon, también se calcula si los pokemon escapan. (el conteo de las pokebolas igual)

Al final se pregunta si se quiere ver el pokedex del usuario en consola y se termina el thread.

**Parameters**

|              |  |
|--------------|--|
| <i>vargp</i> | Descriptor de archivo (desde la creación del thread) |
|--------------|--|

**Returns**

NULL?

## 2.2 servidor.c File Reference

Programa del Servidor.

```
#include "servidor.h"
```

**Functions**

- void [error](#) (char \*errorMessage)  
*Función para el manejo de errores.*
- int [main](#) (int argc, char \*\*argv)  
*Función para establecer conexión con los clientes.*



## 2.2.1 Detailed Description

Programa del Servidor.

### Date

11 Diciembre 2018 Se inicializa el socket, se define la dirección del puerto, y se crean los threads correspondientes a los clientes que piden conexión.

## 2.2.2 Function Documentation

### 2.2.2.1 error()

```
void error (
    char * errorMessage )
```

Función para el manejo de errores.

Muestra un error del lado del servidor.

#### Parameters

|                     |                     |
|---------------------|---------------------|
| <i>errorMessage</i> | El mensaje de error |
|---------------------|---------------------|

#### Returns

Void

### 2.2.2.2 main()

```
int main (
    int argc,
    char ** argv )
```

Función para establecer conexión con los clientes.

Aquí se crea/inicia el socket para escuchar peticiones (ciclo infinito), cuando se acepta un cliente, se crea un thread para poder tener varias conexiones a la vez y se empieza a atender al cliente.

#### Parameters

|             |                       |
|-------------|-----------------------|
| <i>argc</i> | Numero de argumentos  |
| <i>argv</i> | Argumentos en arreglo |

## Returns

...

## 2.3 servidor.h File Reference

Definiciones/Mensajes del servidor y el cliente.

```
#include <stdio.h>
#include <sys/socket.h>
#include <sys/time.h>
#include <string.h>
#include <unistd.h>
#include <errno.h>
#include <stdlib.h>
#include <arpa/inet.h>
#include <pthread.h>
#include <time.h>
```

## Macros

- `#define MAX_POK` 151
- `#define SERV_PORT` 9999
- `#define BUFF_SIZE` 2049
- `#define ASK_POK` 10
- `#define BALL` 11
- `#define ROCK` 12
- `#define BAIT` 13
- `#define RUN` 14
- `#define TRY_USER` 15
- `#define USER` 20
- `#define ENCOUNTER` 21
- `#define ESCAPE` 22
- `#define SUCCESS` 23
- `#define NO_TRIES` 24
- `#define POKEDEX` 25
- `#define OK` 30
- `#define NO` 31
- `#define END_SESSION` 32
- `#define TIMEOUT` 40
- `#define NO_USER` 41
- `#define ERROR_CODE` 42

## Functions

- void `error` (char \*errorMessage)  
*Función para el manejo de errores.*
- void \* `handle_client` (void \*vargp)  
*Funcion que atiende al cliente.*
- void \* `err_send` (int clntfd, char code)  
*Envia un codigo de error al cliente.*

### 2.3.1 Detailed Description

Definiciones/Mensajes del servidor y el cliente.

#### Date

11 Diciembre 2018 Se define la tabla de códigos del cliente y servidor, así como de uso general y de error. Además se definen funciones adicionales para el servidor.

### 2.3.2 Macro Definition Documentation

#### 2.3.2.1 ASK\_POK

```
#define ASK_POK 10
```

(1B op) Pide por un Pokemon

#### 2.3.2.2 BAIT

```
#define BAIT 13
```

(1B op) Tira comida al Pokemon

#### 2.3.2.3 BALL

```
#define BALL 11
```

(1B op) Tira pokebola al Pokemon

#### 2.3.2.4 BUFF\_SIZE

```
#define BUFF_SIZE 2049
```

Tamaño de Buffer

#### 2.3.2.5 ENCOUNTER

```
#define ENCOUNTER 21
```

(1B op, 1B id, 1B pb) Pokemon id con pb

#### 2.3.2.6 END\_SESSION

```
#define END_SESSION 32
```

(1B op) Termino de Sesion

#### 2.3.2.7 ERROR\_CODE

```
#define ERROR_CODE 42
```

(1B op) El codigo no se esperaba

#### 2.3.2.8 ESCAPE

```
#define ESCAPE 22
```

(1B op) El Pokemon se escapo

#### 2.3.2.9 MAX\_POK

```
#define MAX_POK 151
```

Pokemon máximos

#### 2.3.2.10 NO

```
#define NO 31
```

(1B op) Rechazar

#### 2.3.2.11 NO\_TRIES

```
#define NO_TRIES 24
```

(1B op) Ya no hay mas pokebolas

#### 2.3.2.12 NO\_USER

```
#define NO_USER 41
```

(1B op) No existe el usuario

#### 2.3.2.13 OK

```
#define OK 30
```

(1B op) Aceptar

#### 2.3.2.14 POKEDEX

```
#define POKEDEX 25
```

(1B op, 151B pk) Se envia a pokedex

#### 2.3.2.15 ROCK

```
#define ROCK 12
```

(1B op) Tira piedra al Pokemon

#### 2.3.2.16 RUN

```
#define RUN 14
```

(1B op) Escapa del encuentro

#### 2.3.2.17 SERV\_PORT

```
#define SERV_PORT 9999
```

Puerto para conexion

#### 2.3.2.18 SUCCESS

```
#define SUCCESS 23
```

(1B op) El Pokemon se atrapo

#### 2.3.2.19 TIMEOUT

```
#define TIMEOUT 40
```

(1B op) Hubo un timeout

#### 2.3.2.20 TRY\_USER

```
#define TRY_USER 15
```

(1B op,nB us) Conectar con el usuario us

#### 2.3.2.21 USER

```
#define USER 20
```

(1B op) Si existe el usuario

### 2.3.3 Function Documentation

#### 2.3.3.1 err\_send()

```
void* err_send (
    int cIntfd,
    char code )
```

Envia un codigo de error al cliente.

**Parameters**

|               |                                   |
|---------------|-----------------------------------|
| <i>clntfd</i> | Descriptor de archivo del cliente |
| <i>code</i>   | El codigo de error                |

**Returns**

NULL?

**2.3.3.2 error()**

```
void error (
    char * errorMessage )
```

Función para el manejo de errores.

Muestra un error del lado del servidor.

**Parameters**

|                     |                     |
|---------------------|---------------------|
| <i>errorMessage</i> | El mensaje de error |
|---------------------|---------------------|

**Returns**

Void

**2.3.3.3 handle\_client()**

```
void* handle_client (
    void * vargp )
```

Funcion que atiende al cliente.

Corroborar que el usuario tenga su archivo correspondiente, obtiene los datos y comienza el ciclo de captura. Se validan las acciones del usuario, así como si se capturan o no los pokemon, también se calcula si los pokemon escapan. (el conteo de las pokebolas igual)

Al final se pregunta si se quiere ver el pokedex del usuario en consola y se termina el thread.

**Parameters**

|              |  |
|--------------|--|
| <i>vargp</i> | Descriptor de archivo (desde la creación del thread) |
|--------------|--|

Returns

NULL?





# Index

ASK\_POK  
servidor.h, 7

BAIT  
servidor.h, 7

BALL  
servidor.h, 7

BUFF\_SIZE  
servidor.h, 7

clnt\_hand.c, 3  
err\_send, 3  
handle\_client, 4

ENCOUNTER  
servidor.h, 7

END\_SESSION  
servidor.h, 7

ERROR\_CODE  
servidor.h, 8

ESCAPE  
servidor.h, 8

err\_send  
clnt\_hand.c, 3  
servidor.h, 9

error  
servidor.c, 5  
servidor.h, 10

handle\_client  
clnt\_hand.c, 4  
servidor.h, 10

MAX\_POK  
servidor.h, 8

main  
servidor.c, 5

NO\_TRIES  
servidor.h, 8

NO\_USER  
servidor.h, 8

NO  
servidor.h, 8

OK  
servidor.h, 8

POKEDEX  
servidor.h, 8

ROCK

servidor.h, 9  
RUN  
servidor.h, 9

SERV\_PORT  
servidor.h, 9

SUCCESS  
servidor.h, 9

servidor.c, 4  
error, 5  
main, 5  
servidor.h, 6  
ASK\_POK, 7  
BAIT, 7  
BALL, 7  
BUFF\_SIZE, 7  
ENCOUNTER, 7  
END\_SESSION, 7  
ERROR\_CODE, 8  
ESCAPE, 8  
err\_send, 9  
error, 10  
handle\_client, 10  
MAX\_POK, 8  
NO\_TRIES, 8  
NO\_USER, 8  
NO, 8  
OK, 8  
POKEDEX, 8  
ROCK, 9  
RUN, 9  
SERV\_PORT, 9  
SUCCESS, 9  
TIMEOUT, 9  
TRY\_USER, 9  
USER, 9

TIMEOUT  
servidor.h, 9

TRY\_USER  
servidor.h, 9

USER  
servidor.h, 9