7 Documentación del proyecto.

Para recoger toda la documentación relativa a la modificación del código fuente, se ha realizado un procesamiento de la información mediante la ayuda que proporciona el programa *Doxygen*, dicho programa es capaz de examinar los ficheros fuentes y generar la documentación procedente de la implementación que se ha desarrollada, y que será entregada junto a esta memoria en un CD adjunto.

Esta documentación será presentada en formato HTML, por lo que es posible leerla desde cualquier navegador web, en cualquiera de los sistemas operativos. En el interior del fichero html se encontrará un fichero con el nombre "*index.html*", este fichero es la página principal de la documentación.

A continuación se va a presentar la estructura indice que será encontrada en la documentación adjunta.

Ilustración 22: Indice estructural de la documentación del código fuente del cliente twinkle.

En la documentación se recoge todas la estructura del programa, las funciones que han sido implementadas,así como, toda la relación de variables, funciones y ficheros relativos a dicho proyecto.

Desde la página principal de la documentación se puede acceder a cualquiera de las funciones y variables implementadas en el cliente mediante cualquiera de las opciones de las que se disponen.

Se puede listar todos los directorios que forman el programa, y que se muestra en la siguiente ilustración.

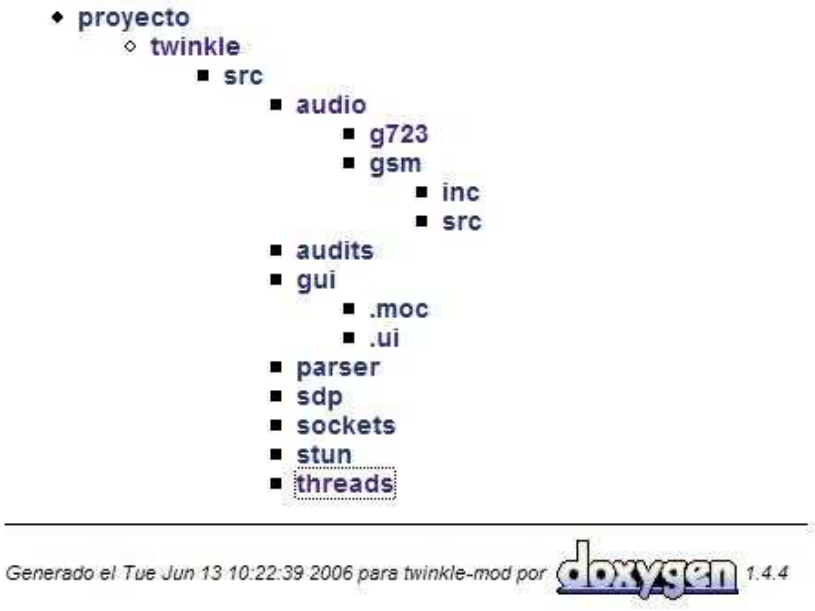

Ilustración 23: Listado de directorios que forman el cliente twinkle.

En la jerarquía de clases se muestra un listado con todas las clases usadas para el desarrollo del cliente, además en este listado también se muestra la herencia de las clases. Observando el listado se puede ver que este cliente es sencillo y hace un uso prudente de la herencia, pues prácticamente la totalidad de las clases no son heredadas.

En la siguiente ilustración se muestran algunas de las clases más importantes y determinantes en el desarrollo del cliente de VoIP.

Ilustración 24: Listado de las principales clases del cliente twinkle

Si se desea acceder al código fuente o se desea obtener mayor información de la implementación de una clase, se tiene que hacer click en el enlace de la clase en cuestión y se mostrará toda la información relativa a dicha clase.

Por ejemplo, se va a mostrar a continuación la información de las clases implementadas para

este proyecto: ***t_audio_net_tx*** y ***t_audio_play_tx***.

Referencia de la Clase t_audio_net_tx

`#include <audio_net_tx.h>`

Lista de todos los miembros.

Descripción detallada

Definición en la línea **21** del archivo **audio_net_tx.h**.

Métodos públicos

	t_audio_net_tx (**t_audio_session** *_audio_session, **t_twinkle_rtp_session** *_rtp_session, **t_audio_codec_g723** _codecg723, unsigned int _pba)
	~t_audio_net_tx ()
void	**set_running** (bool running)
void	**run** (void)
t_line *	**get_line** (void) const
void	**lock** (void)
void	**unlock** (void)

Atributos privados

t_audio_session *	**audio_session**
t_twinkle_rtp_session *	**rtp_session**
t_audio_codec_g723	**codecg723**
unsigned char *	**sample_buf**
unsigned int	**pbaBuffer**
bool	**is_running**
bool	**stop_running**

Ilustración 25: Información extendida de la clase t_audio_net_tx creada para este proyecto.

Documentación del constructor y destructor

```
t_audio_net_tx::t_audio_net_tx ( t_audio_session *       _audio_session,
                                 t_twinkle_rtp_session * _rtp_session,
                                 t_audio_codec_g723      _codecg723,
                                 unsigned int            _pba
                               )
```

t_audio_net

Parámetros:

_audio_session	referencia al objeto padre que realiza la creaciÃ³n de hilo audio_net
_rtp_session	referencia a la session creada para utilizar las librerias de ccRTP
_codecg723	means setting to codec g723.1
_pba	means algorithm use to buffer playout

Definición en la línea **55** del archivo **audio_net_tx.cpp**.

Hace referencia a **audio_session**, **codecg723**, **is_running**, **MEMMAN_NEW_ARRAY**, **pbaBuffer**, **rtp_session**, **sample_buf**, **SAMPLE_BUF_SIZE**, y **stop_running**.

Ilustración 26: Documentación generada para el constructor de la clase t_audio_net_tx.

En la ilustración 26 se puede ver detalladamente la información extendida que se va a encontrar en la documentación adjunta. Para cada clase tenemos la información completa, esto

incluye métodos y variables. Además se indica el número de linea del fichero que contiene la definición, y también incluye información acerca de los métodos que han sido referenciados o utilizados en la función examinada.

Descripción detallada

Definición en la línea **27** del archivo **audio_play_tx.h**.

Métodos públicos

	t_audio_play_tx (**t_audio_io** *_playback_device, **t_audio_codec** _codec, **t_audio_codec_g723** _codecg723, unsigned short _ptime=0)
	~t_audio_play_tx ()
void	**set_running** (bool running)
void	**run** (void)
void	**lock** (void)
void	**unlock** (void)
void	**retain_for_concealment** (unsigned char *buf, unsigned short **len**)
void	**conceal** (short num)
void	**clear_conceal_buf** (void)
void	**play_pcm** (unsigned char *buf, unsigned short **len**, bool only_3rd_party=false)

Atributos privados

t_audio_io *	**playback_device**
t_audio_codec	**codec**
unsigned short	**ptime**
t_audio_codec_g723	**codecg723**
gsm	**gsm_decoder**
unsigned char *	**sample_buf**
unsigned char *	**conceal_buf** [MAX_CONCEALMENT]
unsigned short	**conceal_buflen** [MAX_CONCEALMENT]
short	**conceal_pos**
short	**conceal_num**
unsigned short	**soundcard_buf_size**
unsigned char *	**jitter_buf**
unsigned short	**jitter_buf_len**
bool	**load_jitter_buf**
bool	**is_running**
bool	**stop_running**

Ilustración 27: Información extendida de la clase t_audio_play_tx creada para este proyecto.

Al igual que con la documentación que se ha mostrado en las ilustraciones 25 y 26, las ilustraciones 27 y 28 son generadas con ayuda de la herramienta doxygen. Esta documentación recoge toda la información de la clase y la ordena para que pueda ser accedida de manera cómoda y sencilla.

Es con ayuda de la documentación generada cuando cualquier otro desarrollador de software puede continuar con el trabajo realizado y analizar todo el código fuente de una manera fácil y sencilla.

La documentación forma parte de la implementación y desarrollo de cualquier software libre, cuanto mejor sea la calidad de la documentación generada más fácil será su comprensión por un tercero y más aceptación tendrá en la comunidad.

Documentación del constructor y destructor

```
t_audio_play_tx::t_audio_play_tx ( t_audio_io *          _playback_device,
                                   t_audio_codec         _codec,
                                   t_audio_codec_g723    _codecg723,
                                   unsigned short        _ptime = 0
                                 )
```

Constructor del objeto que va a manejar la reproducciÃ³n de las tramas de audio que van llegando a nuestro equipo.

Parámetros:

_playback_device	nos indica cual va a ser la referencia para poder enviar los datos PCM al sistema de audio
_codec	nos indica el codec que va a ser utilizado en la session RTP
_codecg723	means setting to codec g723.1
_ptime	es el periodo de tiempo de la trama que contiene el paquete RTP (playout time)

Definición en la línea **54** del archivo **audio_play_tx.cpp**.

Hace referencia a **codec**, **CODEC_G711_ALAW**, **CODEC_G711_ULAW**, **CODEC_G723**, **CODEC_G723AR53**, **CODEC_G723R53**, **CODEC_G723R63**, **CODEC_GSM**, **codecg723**, **conceal_buf**, **conceal_buflen**, **conceal_num**, **conceal_pos**, **t_audio_io::get_buffer_size()**, **gsm_decoder**, **INFO**, **initDecoderG723()**, **initVadG723()**, **is_running**, **jitter_buf**, **jitter_buf_len**, **JITTER_BUF_SIZE**, **load_jitter_buf**, **MAX_CONCEALMENT**, **MEMMAN_NEW_ARRAY**, **playback_device**, **ptime**, **PTIME_G711_ALAW**, **PTIME_G711_ULAW**, **PTIME_G723**, **PTIME_GSM**, **sample_buf**, **SAMPLE_BUF_SIZE**, **soundcard_buf_size**, y **stop_running**.

```
t_audio_play_tx::~t_audio_play_tx ( )
```

Definición en la línea **135** del archivo **audio_play_tx.cpp**.

Hace referencia a **conceal_buf**, **gsm_decoder**, **is_running**, **jitter_buf**, **MAX_CONCEALMENT**, **MEMMAN_DELETE_ARRAY**, **sample_buf**, y **stop_running**.

Ilustración 28: Documentación generada para el constructor de la clase t_audio_play_tx.

La documentación del código fuente permitirá que cualquier programador interesado en el proyecto pueda continuar con el trabajo, justo en el estado en que se ha quedado. Esta documentación facilita la compresión y el análisis del programa permitiendo al programador bucear por las clases y funciones de manera sencilla y ordenada.

Antes de finalizar este apartado mencionar que la documentación relativa a las librerías ccRTP utilizadas por el cliente twinkle no están incluidas en el CD adjunto, pero se pueden consultar en la siguiente dirección:

- http://data.gnutelephony.org/ccrtp/refman/html/index.html

En la ilustración 29 se muestra el diagrama de bloques de la función principal del cliente twinkle, si se observa detenidamente no se va a encontrar ninguna referencia a las clases t_audio_net_tx y t_audio_play_tx, estas clases son creadas como consecuencia del inicio de una conversación de voz entre dos extremos. Sin embargo, en la ilustración se puede observar como twinkle ha sido diseñado desde el principio como una aplicación multihilo, ejecutando las tareas en procesos independientes unos de los otros y favoreciendo la creación de nuevos hilos de ejecución.

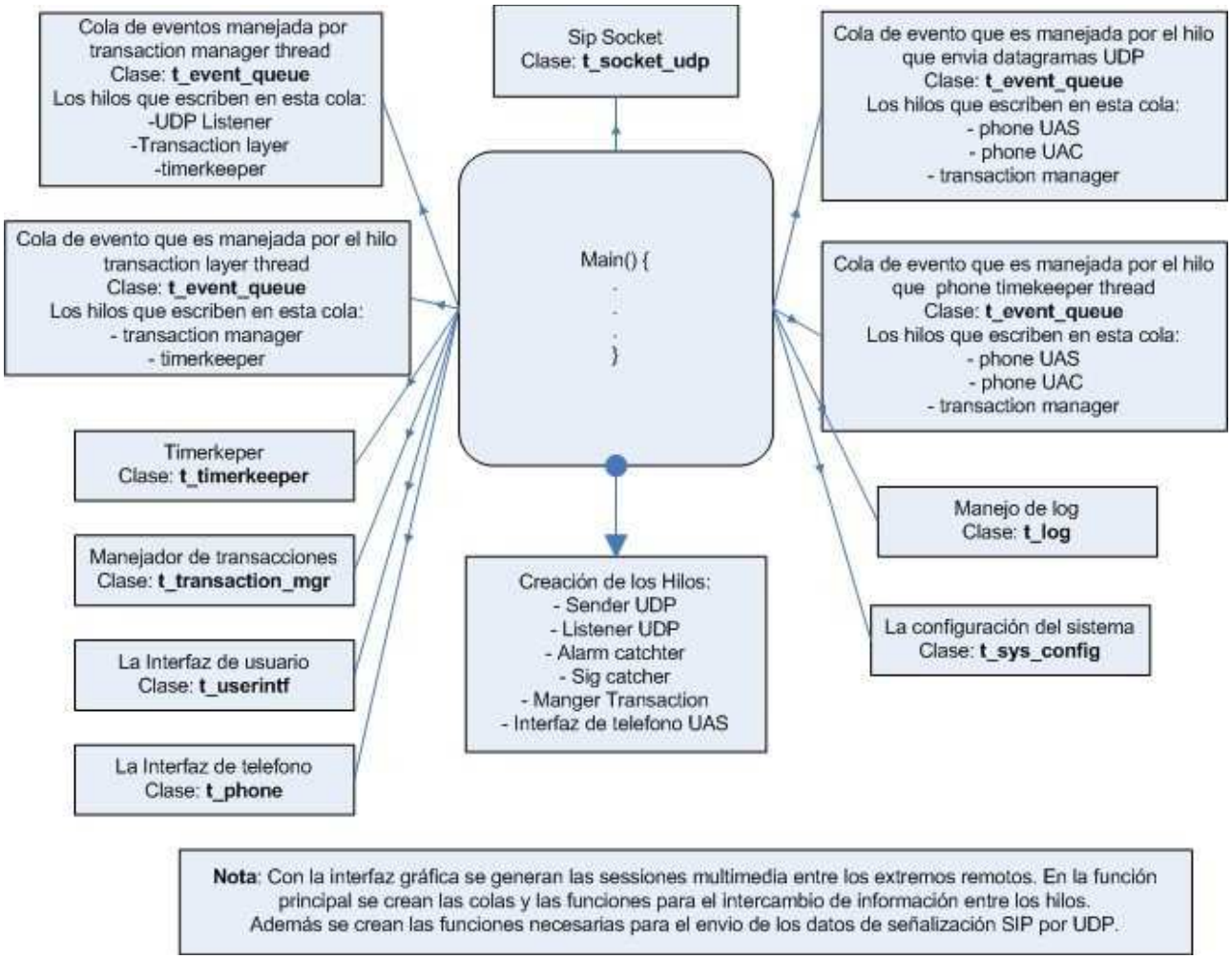

Ilustración 29: Diagrama de bloques de la función Main() del cliente twinkle.

Además de la documentación del código fuente en el CD adjunto también se encuentra el código fuente generado y utilizado para el desarrollo de este proyecto.