



Universitat Oberta
de Catalunya

PRESENTACIÓN TRABAJO FIN DE GRADO

Autor: José Ignacio Bengoechea Isasa

Consultora: María Isabel March Hermo

Área: Redes de computadores

Junio 2017



Diseño de un sistema de control remoto
REMOTE ANYWHERE

<http://www.remoteanywhere.net>



Todas las imágenes de esta presentación tienen licencia Creative Commons.
Esta presentación está sujeta a una licencia de Reconocimiento no comercial sin obra derivada [3.0 España de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)



Índice

OBJETIVOS

METODOLOGÍA

PLANIFICACIÓN

DESARROLLO: FASE 1,2,3,4,5 y 6

TECNOLOGÍAS USADAS

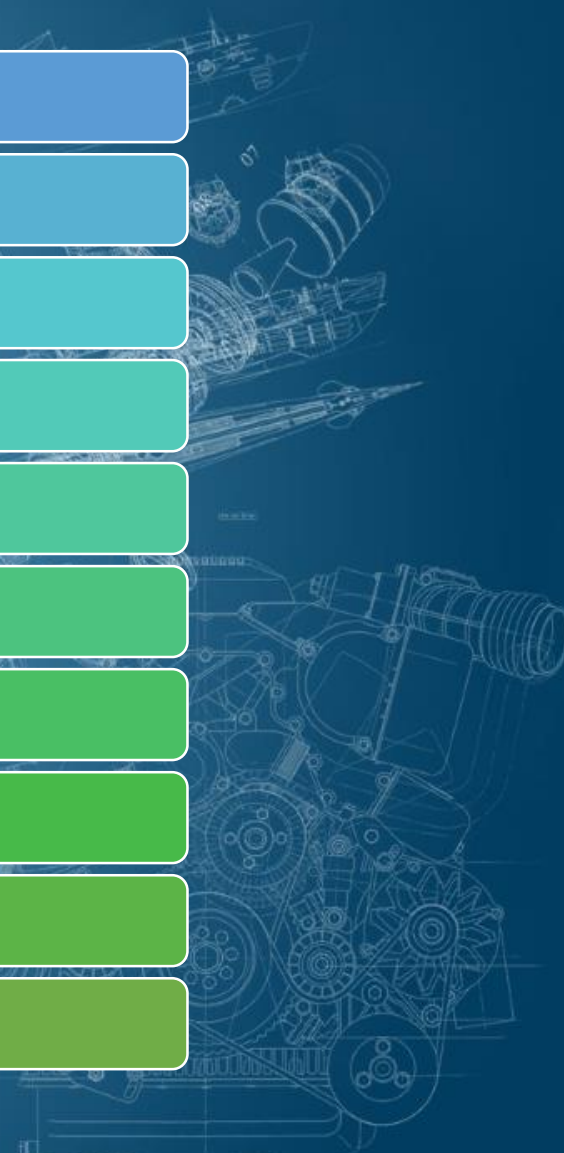
ESTRUCTURA

FUNCIONAMIENTO

INTERFACE

FUTURAS AMPLIACIONES

CONCLUSIONES



Objetivos

Objetivo

- Desarrollo de un sistema de control remoto que permita:
 - Recibir la captura de pantalla de un servidor remoto.
 - Enviar eventos relacionados del ratón.
 - Enviar eventos relacionados del teclado.

Filosofía

- Usar una tecnología que no se haya visto durante el grado en ingeniería informática.
- El desarrollo debe ser opensource.
- La arquitectura debe permitir futuras ampliaciones.
- La elección del nombre, *Remote Anywhere*, la podemos traducir como control remoto desde cualquier sitio.



Metodología

Metodología

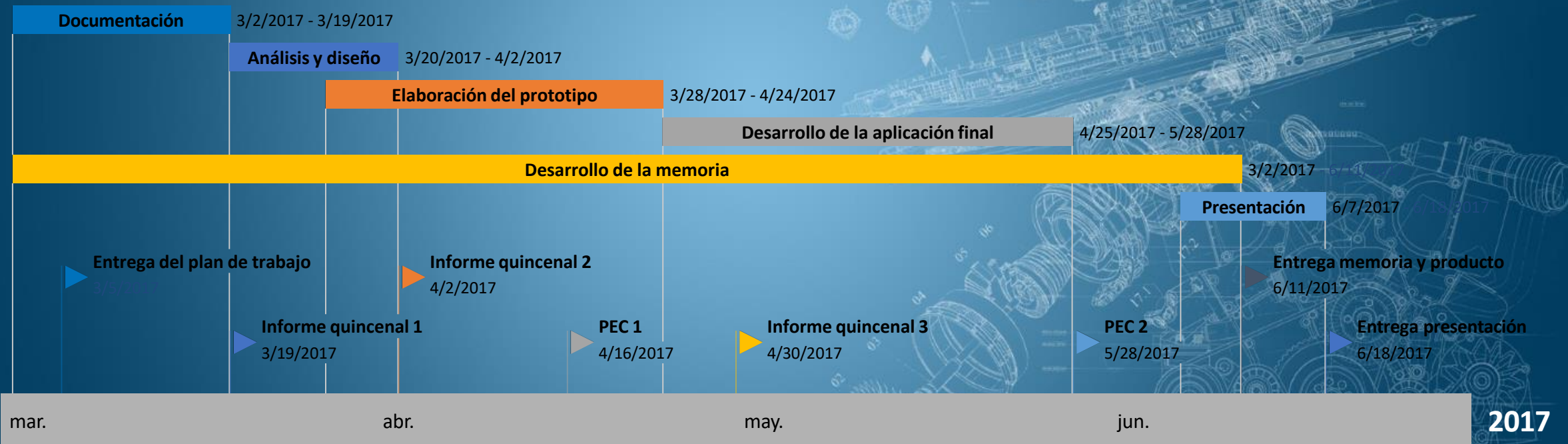
- La metodología empleada es EDT, estructura de descomposición del trabajo.
- En cada fase se genera documentación que se integra en la memoria final.



Planificación

Planificación

- La planificación se estructura a partir del calendario de entregas.
- El proceso de entregas permite revisar con la consultora, María Isabel March, la evolución del proyecto



Fase 1: Documentación y especificaciones

Recogida de documentación

- Documentación sobre desarrollo de aplicaciones cliente/servidor.
- Documentación sobre desarrollo de arquitecturas de n capas.
- Revisión de las funcionalidades de los sistemas de control remoto del mercado. Como teamviewer, vnc, remote desktop, logmein.



Especificaciones

- Tres modos de trabajo de la aplicación. Modo cliente, modo servidor, modo mixto.
- El puerto del servidor es configurable.
- El servidor tiene un listado de usuarios y contraseñas editable.
- El servidor guarda los registros de acceso.
- El cliente tiene un listado de servidores editable.



Fase 2: Análisis y diseño

Diagramas de análisis

- Diagramas de casos de uso.
- Diagramas de clases.
- Elaboración de *mockups*, o bosquejos de prototipado.

Clientes		Servidor		Ayuda	
Conectar	Configura	LOGO			
Modificar	Permisos				
Salir	Registro				
Permitir control remoto		Controlar remotamente			
Su ip interna: 192.168.1.20		CASA: 192.168.1.10:4433 OFICINA: 93.45.23.45:4543			
Su ip externa: 92.55.67.123					
Su puerto: 4444					

Arquitectura

- Se utiliza una combinación de arquitecturas.
- Arquitectura cliente/servidor para la comunicación.
- Arquitectura basada en capas para el desarrollo funcional.
 - Capa de presentación.
 - Capa de lógica de negocios.
 - Capa de acceso a datos.
 - Capa de comunicación entre cliente y servidor.

Comunicación

Presentación

Lógica

Datos

Fase 3: Desarrollo del prototipo



Elección de la tecnología

Se selecciona la tecnología Net de Microsoft. La cual no se ha visto en el grado. El IDE es Visual Studio 2017

Se define Windows como el sistema operativo de la aplicación.

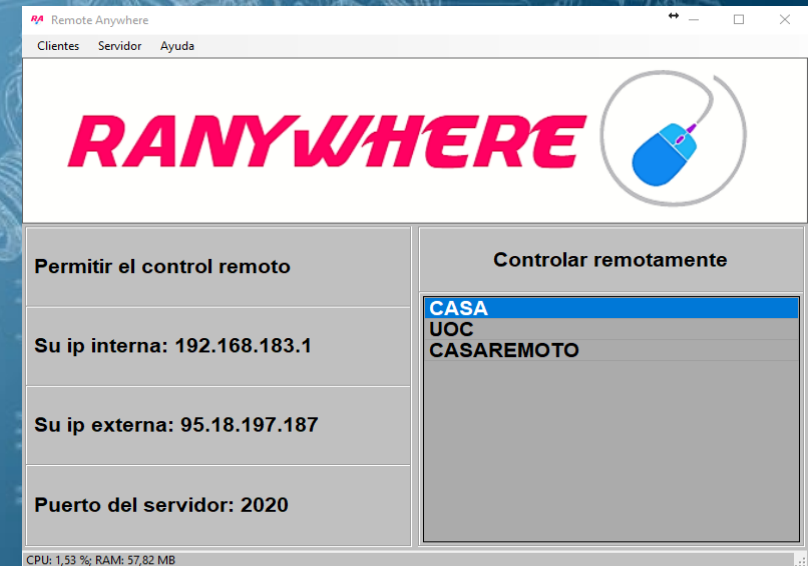
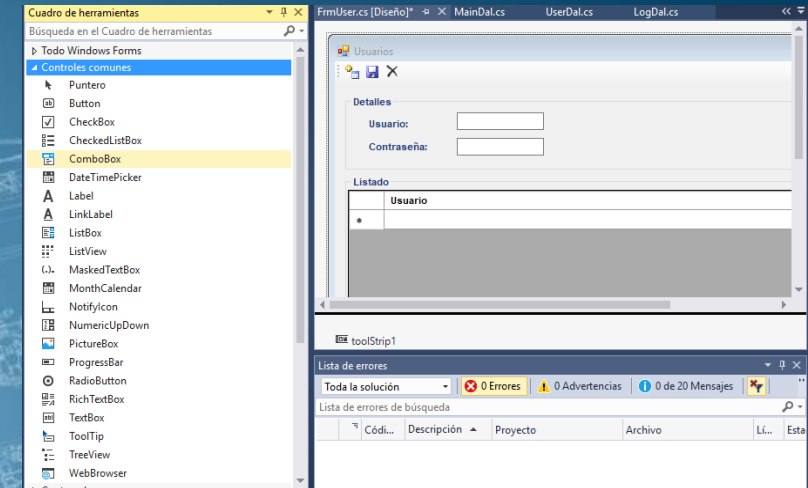
Para la capa de comunicación se selecciona WCF, que nos permite comunicar con objetos de procesos remotos.

Establecimiento del prototipo

El entorno visual del prototipo usa Windows Forms.

La división de las funcionalidades en capas permite dividir el trabajo

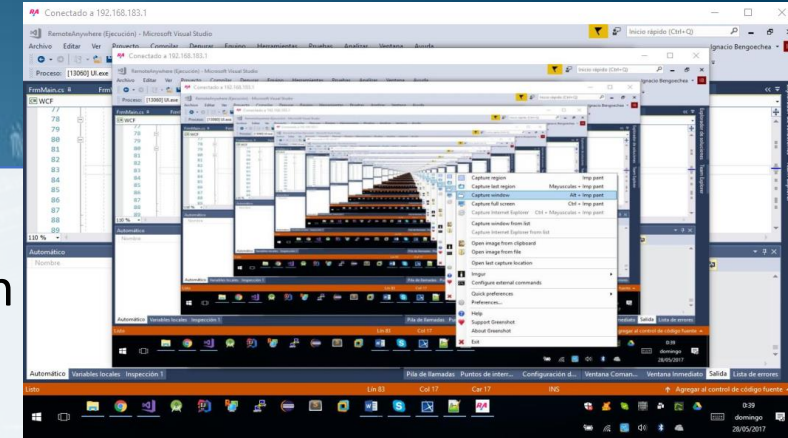
En esta fase el prototipo no nos permite la conexión, pero si la creación de usuarios y de servidores.



Fase 4: Desarrollo de la aplicación final

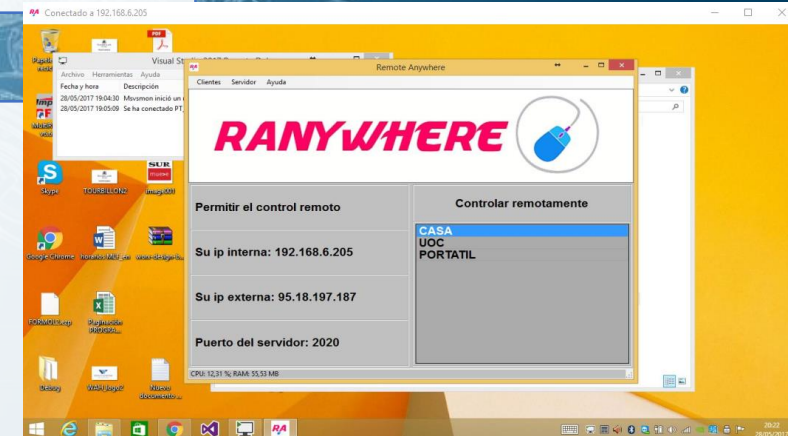
Aplicación funcional

- En esta fase se desarrolla la funcionalidad de comunicación mediante WCF.
- Nos permite conectarnos remotamente al servidor y solicitarle imágenes de su consola, o enviarle eventos del ratón y el teclado.



Establecimiento de pruebas

- Se realizan pruebas de funcionalidad de la aplicación, que nos permiten establecer si se cumplen las especificaciones de la misma.
- También se establecen pruebas de rendimiento para establecer la optimización de la aplicación.



Fase 5 y 6 : Desarrollo de la memoria y de la presentación

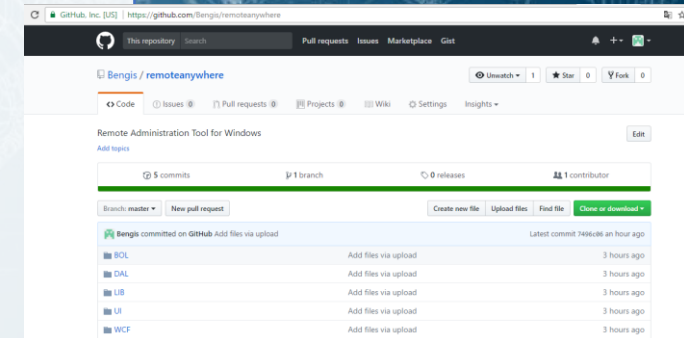
Desarrollo de la memoria

- Se integra la documentación que se ha elaborado en las fases previas.
- La memoria debe seguir un formato establecido por la UOC para la unificación de los trabajos fin de grado.



Desarrollo de la presentación

- Se elabora este documento para explicar el proceso relacionado con la gestión de este proyecto.
- Se sube el código a un repositorio en GitHub disponible en la dirección <http://www.remoteanywhere.net>, desde el que se puede descargar el código fuente y compilarlo.



Tecnologías usadas

Software de desarrollo

- El IDE es Visual Studio 2017 Community Edition.
- El entorno de programación es Net Framework.
- El lenguaje de programación es C#.
- La base de datos es Sql Compact



Librerías usadas

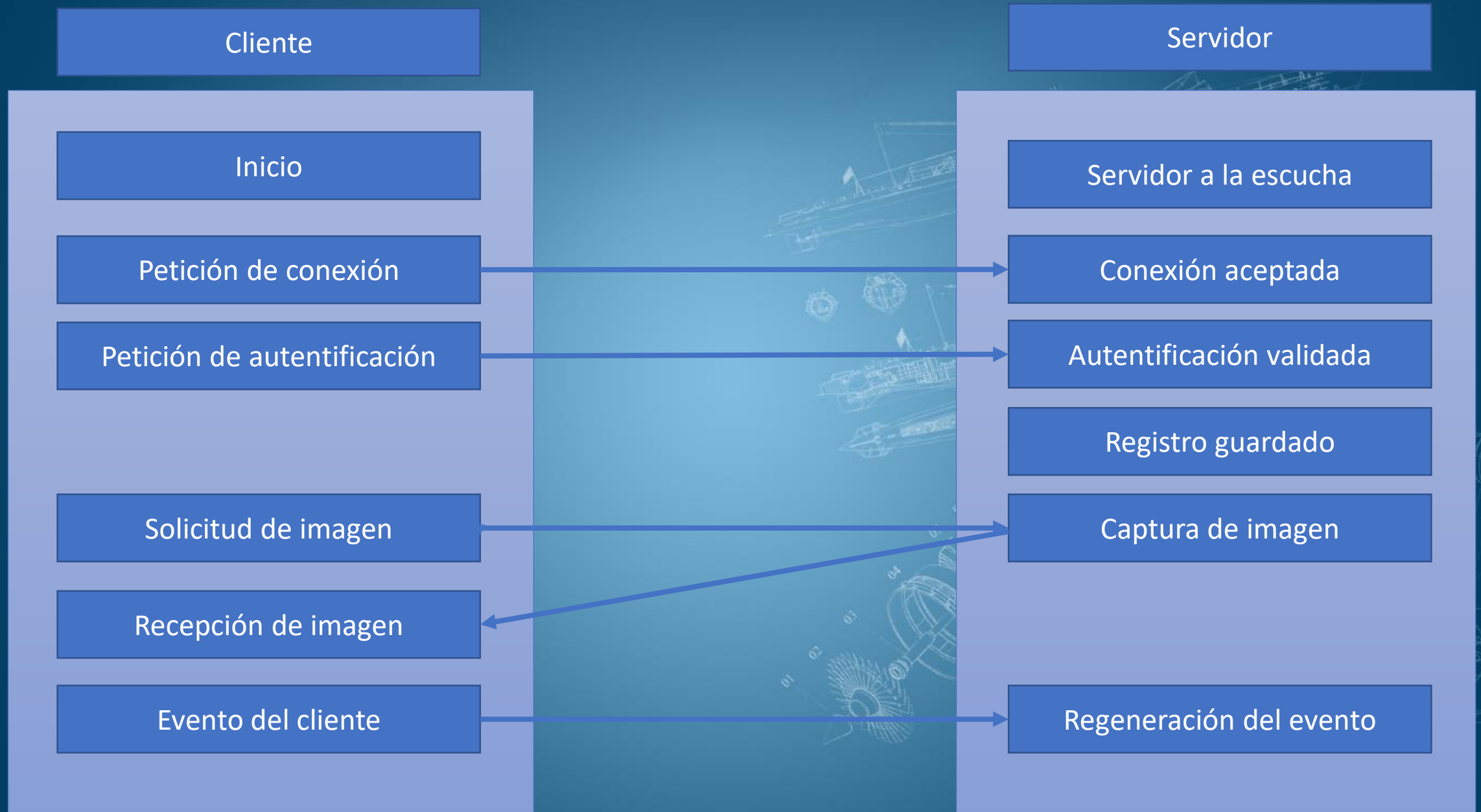
- Para la comunicación se ha usado Windows Communication Foundation.
- Para la simulación de eventos de teclado se ha usado la librería open source *InputSimulator*.
- Para la simulación de eventos de ratón se usa la API de Windows.



Estructura



Funcionamiento



Interface

The screenshot shows the 'Remote Anywhere' application window. The title bar includes the 'RANYWHERE' logo and window controls. The main menu has 'Clientes', 'Servidor', and 'Ayuda' options. The main content area is titled 'Alta de servidores y de usuarios y contraseñas' and features the 'RANYWHERE' logo and a mouse icon. The interface is divided into two main sections: 'Permitir el control remoto' and 'Controlar remotamente'. The 'Permitir el control remoto' section contains three input fields: 'Ip de conexión del servidor' (with value 192.168.183.1), 'Su ip interna: 192.168.183.1', and 'Puerto del servidor' (with value 2020). The 'Controlar remotamente' section contains a list of servers with 'CASA' selected. A red arrow points to the 'CASA' entry with the text 'Doble clic para conectar a servidores configurados previamente'. The system tray at the bottom shows 'CPU: 4,55 %; RAM: 59,28 MB'.

Remote Anywhere

Cientes Servidor Ayuda

Alta de servidores y de usuarios y contraseñas

RANYWHERE

Permitir el control remoto

Ip de conexión del servidor

Su ip interna: 192.168.183.1

Su ip externa: 95.18.205.91

Puerto del servidor

Puerto del servidor: 2020

Controlar remotamente

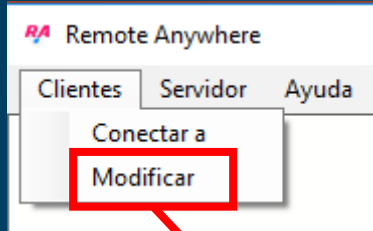
CASA

UOC

Doble clic para conectar a servidores configurados previamente

CPU: 4,55 %; RAM: 59,28 MB

Interface: Alta de servidores



Servidores

Nuevo servidor, guardar servidor, eliminar servidor

Detalles

Nombre:

Ip:

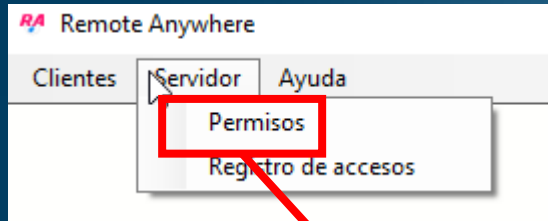
Puerto:

Listado

Nombre	IP	Puerto
CASA	192.168.0.1	4444
UOC	192.168.6.150	4444

Doble clic para conectar a servidores configurados

Interface: Alta de usuarios



Usuarios

Nuevo usuario, guardar usuario, eliminar usuario

Detalles

Usuario:

Contraseña:

Listado

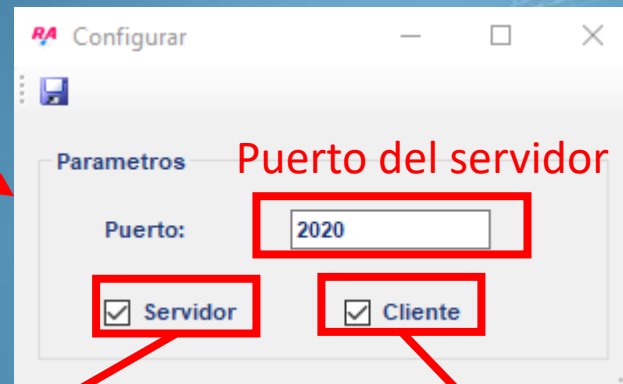
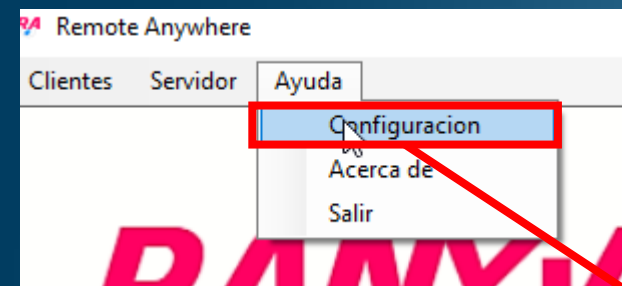
Usuario	Contraseña
PACO	1235
LUIS	1234
MARIA	1234

The main application window, titled 'Usuarios', features a toolbar with three icons: a plus sign (add), a floppy disk (save), and an 'X' (delete). A red box highlights these icons, with a red text label 'Nuevo usuario, guardar usuario, eliminar usuario' pointing to them. Below the toolbar is a 'Detalles' section with two input fields for 'Usuario' and 'Contraseña'. The 'Listado' section contains a table with three rows of user data. A red arrow from the 'Permisos' menu in the inset points to the main application window.

Interface: Configuración

Modos de la aplicación

- Modo servidor. Solo servidor
- Modo cliente. Solo cliente
- Modo mixto. Ambos modos activos.
- Los cambios requieren un reinicio



Opción de modo servidor



Opción de modo cliente

Futuras ampliaciones

Nuevas plataformas

- El desarrollo por capas nos permite incluir nuevos tipos de clientes.
- En el caso de Android y de iOS es necesario modificar la capa de comunicación y de presentación.
- Xamarin es una alternativa en Android y en iOS, pero requiere que terminen de portar el código de la librería de WCF.

Nuevas funcionalidades

- Es posible incluir un cliente en web que nos permita hacer teleasistencia a un cliente sin necesidad de que abra puertos. Requiere que se modifique la capa de comunicación para que funcione en ambos sentidos como cliente y servidor.
- Es posible ampliarlo usando nuestra propio interface de comunicación basada en UDP perforado para evitar el uso de puertos.

Conclusiones

Objetivo

- Se han cumplido los objetivos del proyecto. Esta solución permite conectarse remotamente a un servidor y controlarlo
- Se han dejado futuras ampliaciones que acercan la solución a una solución comercial. El proyecto se subirá a GitHub para compartirlo con la comunidad.

Metas personales

- Estoy satisfecho con los objetivos cumplidos. Para mi el principal objetivo era crear un proyecto de programación que me permitiera aprender sobre procesos de comunicación.
- La carga de trabajo ha sido posiblemente la mayor de todo el grado, pero el desarrollo de las entregas separadas en las PEC ha permitido la división del trabajo a realizar.

Gracias por ver la presentación. Espero que les haya resultado interesante