



# Informe de pruebas

## Proyecto VIGIA

### **Autores:**

Versión: 4.1

Rafael Martínez González  
Israel Rodríguez Martín  
Jesús Manuel Rodríguez Sánchez

**Fecha: 24/01/2009**



## Tabla de contenido

---

<b>1ª Iteración</b>	<b>4</b>
Conexión Mando - PC	4
<b>2ª Iteración</b>	<b>7</b>
Aplicación que mueve la cámara	7
Aplicación de comunicación Cliente-Servidor	7
Aplicación CMC (Coordenadas Mando Consola)	8
LED's	9
Recorte de imágenes	10
<b>3ª Iteración</b>	<b>13</b>
Integración: enviar órdenes	13
<b>4ª Iteración</b>	<b>15</b>
Integración: envío de órdenes / recepción de frames	15
Integración y calibración del sistema	16
<b>Resultados de pruebas</b>	<b>18</b>
<b>Matriz de trazabilidad</b>	<b>21</b>

# 1ª Iteración

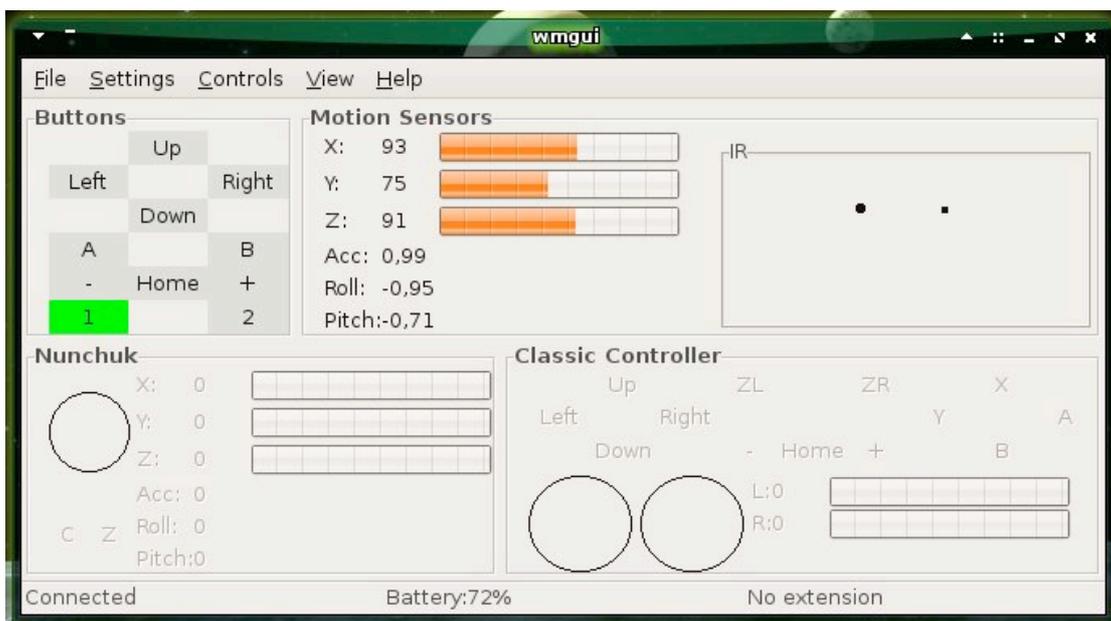
## Conexión Mando - PC

Para las pruebas de conexión entre el mando de la Wii (Wiimote) y el PC, hemos usado dos programas, uno para Linux y otro para Windows.

### Conexión Mando-PC en Linux

Hemos usado la librería Cwiid, que trae algunos programas de ejemplo para usar el Wiimote, los cuales pasamos a describir a continuación:

- **Wmgui:** es una interfaz gráfica que permite saber qué botones pulsas, cómo movemos el mando... además de otros parámetros. En concreto, como en la ventana podemos ver la detección de la luz infrarroja, hemos probado principalmente hasta qué distancia se capta la señal, que en nuestro caso ha llegado hasta unos dos metros aproximadamente. Se observa que, cuanto más nos acercamos, más grande aparece el punto que representa al infrarrojo;



por el contrario, si nos alejamos, el grosor de dicho punto disminuye, llegando a veces a parpadear.

Por otra parte, también hemos comprobado que, cuanto más nos acercamos al sensor LED, menos movimiento lateral podemos captar, lo cual es lógico, ya que el campo de visión es más reducido. En esta prueba también observamos que el punto que representa al infrarrojo disminuye de tamaño cuanto más se aleja lateralmente.

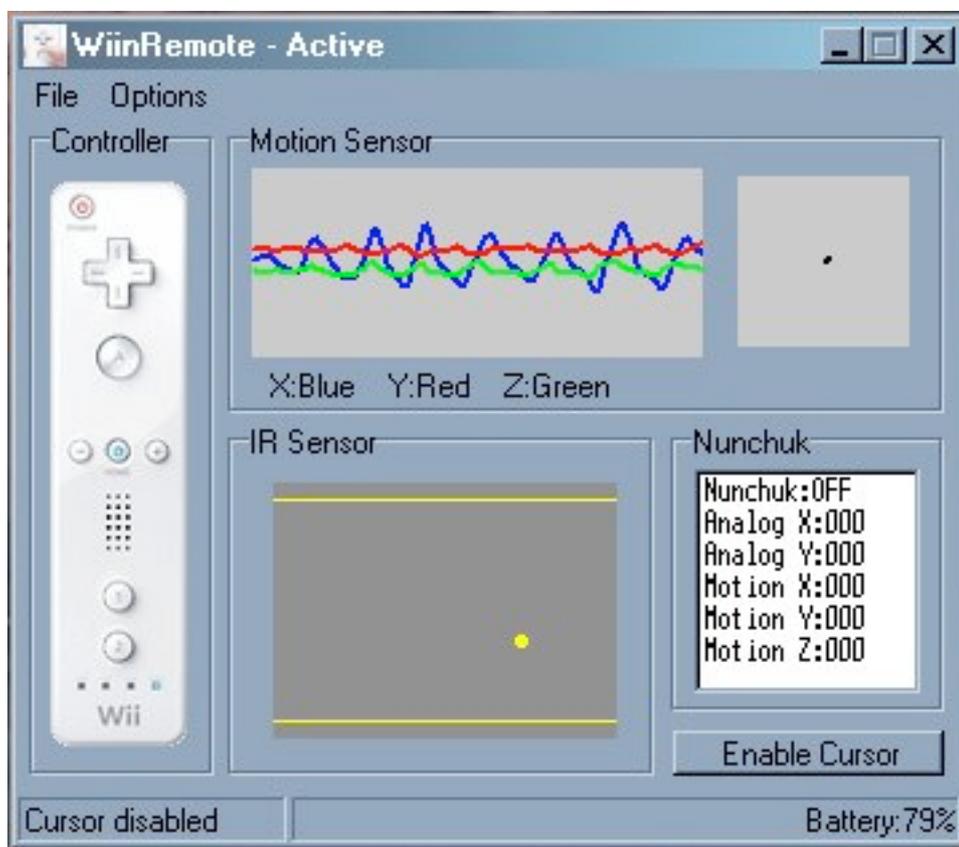
- **Wminput:** este programa nos permite usar el Wiimote como ratón. Podemos configurar el funcionamiento de cada botón del mando para que responda según prefiramos. La detección de la señal es bastante buena y muy realista, aunque, al principio, cuesta manejarse con esta funcionalidad, ya que no estamos acostumbrados a usar un ratón de esta forma.

Para probarlo, intentamos mover ventanas, maximizarlas, abrir carpetas, etc. También lo probamos con el 'Planet Penguin Racer', un juego de carreras con el simpático pingüino Tux.

- **Wmdemo:** esta herramienta permite monitorizar los sensores del Wiimote, visualizando las coordenadas del mando con cada movimiento que realizamos con él, los botones que pulsamos, los valores de aceleración, etc. Como desventaja tiene que no presenta una interfaz tan agradable al usuario como Wmgui.

### Conexión Mando-PC en Windows

En primer lugar, instalamos el programa BlueSoleil, que permite crear una conexión Bluetooth entre el PC y el Wiimote sin que se pierda con el tiempo. Una vez que sincronizamos el mando y Bluesoleil encuentra este dispositivo, ejecutamos el WiinRemote, una aplicación que nos permite usar el mando de la Wii. Este programa tiene una interfaz parecida a la del Wmgui: en ella, podemos visualizar varios parámetros del Wiimote, como pueden ser los botones pulsados, el acelerómetro, la señal emitida, etc.



Además, también podemos usar el mando de la Wii como ratón, al igual que con Wminput. La precisión del movimiento es bastante buena, quizás un poco menos que su homólogo de Linux. Con este funcionamiento, cuando el cursor llega a alguno de los bordes de la pantalla, el Wiimote vibra indicando al usuario que no puede seguir en esa dirección.

## 2ª Iteración

### Aplicación que mueve la cámara

Para las siguientes pruebas, hemos usado una clase de test que nos pide a través de la consola cómo queremos mover la base de la cámara: de forma automática o introduciendo nosotros mismos manualmente los parámetros.

#### Movimiento automático de la base de la cámara

Al elegir la opción 'Auto' en la consola, la cámara ha respondido como estaba previsto ante las acciones que le hemos indicado en la clase de prueba. Éstas han consistido, principalmente, en desplazamientos horizontales y verticales de la base en ambos sentidos. Entre cada desplazamiento, hemos insertado un tiempo de pausa para poder percibir los movimientos por separado.

#### Movimiento de la base de la cámara ante una acción propuesta por el usuario

Tras elegir la opción 'Manual' en la consola de Java, hemos introducido los parámetros 'x', 'y' y 'z' para que la base reaccionara desplazándose según esos valores. Al igual que en el caso anterior, la base ha respondido correctamente para varias combinaciones (x, y, z).

#### Movimiento de la base de la cámara ante una serie de acciones del usuario

Una vez escogida la opción 'Manual' en la consola de Java, hemos introducido una terna (x, y, z) y la base ha reaccionado a partir de los valores que hemos escogido. A continuación, de nuevo hemos escrito una terna para que la base efectuara el movimiento correspondiente, y así sucesivamente hasta en cinco ocasiones. En todas ellas, la respuesta de la base ha sido inmediata y según lo previsto.

### Aplicación de comunicación Cliente-Servidor

Para esta prueba, ha sido necesaria la implementación de dos clases de test: una para el servidor y otra para el cliente.

#### Establecimiento de una conexión Cliente-Servidor con envío de frames

En la clase de prueba del servidor, en primer lugar, hemos creado un socket de escucha con el cliente. A continuación, se reciben los comandos que envía el cliente (log y petición de frames). El servidor lee las imágenes y le manda frames al cliente.

Por su parte, en la clase de prueba del cliente, primeramente se comunica con el servidor a partir de la dirección IP y los puertos TCP y UDP que hemos escogido. Seguidamente, una vez que ya está establecida la conexión, el cliente recibe los frames desde el servidor. Finalmente, se cierra la comunicación entre el cliente y el servidor.

Las pruebas realizadas se han desarrollado como se esperaba. Como resultado, hemos obtenido el número de frames por segundo que se ha podido transmitir entre el servidor y el cliente, y que de media nos ha salido unos 28 fps.

## Aplicación CMC (Coordenadas Mando Consola)

Para todas las pruebas, se ha usado una clase de test que se basa en mostrar por pantalla las coordenadas del mando respecto a los LED's infrarrojos. Como fuente de infrarrojos, se han usado dos mandos a distancia que han simulado los posibles movimientos del observador.

### Detección del movimiento horizontal de los LED's

Para este caso, se han realizado movimientos horizontales con un solo mando a distancia y con los dos a la vez, teniendo el Wiimote en un lugar fijo, y los resultados obtenidos han sido los esperados.

### Detección del movimiento vertical de los LED's

Mismo caso de antes pero realizando movimientos verticales en lugar de movimientos horizontales. La respuesta del sistema ha sido, igualmente, satisfactoria para todas las pruebas realizadas.

### Detección del movimiento en profundidad de los LED's

El caso de la detección de la profundidad ha sido un poco más complejo, ya que se requiere que los dos mandos estuvieran alineados entre sí y, a su vez, con el Wiimote, de manera que el Wiimote fuera capaz de calcular la distancia entre él y los LED's. Siguiendo los pasos descritos, se ha logrado comprobar que realmente el valor de profundidad mostrado por la clase de test iba cambiando según nos acercábamos o alejábamos del Wiimote.

### Detección de la posición de varios LED's simultáneamente

Esta prueba en realidad ya se ha probado indirectamente con las anteriores, sin embargo se ha vuelto a comprobar que la aparición y desaparición de LED's en el campo visual del Wiimote eran detectados correctamente (se han realizados diferentes variantes de las anteriores).

## Detección de la posición de los LED's en un entorno con otras fuentes de luz

Esta prueba se ha llevado a cabo usando uno de los mandos a distancia como emisor de infrarrojos y una linterna como fuente de luz externa al sistema. Sólo se han apreciado pequeñas interferencias cuando la linterna se ponía junto a la fuente de infrarrojos y ambas fuentes se encontraban muy cerca del wiimote. Es por ello que la respuesta obtenida por el sistema parece ser bastante buena, sobre todo si tenemos en cuenta que los emisores de LED's infrarrojos no son mandos a distancia y no los LED's que se usarán en el sistema final.

## LED's

### Detección de máxima distancia de los LED's

A partir de un metro y medio, el sensor sólo captura un punto. A dos metros y medio sigue captando bien los leds y tiene cierto rango de desplazamiento.

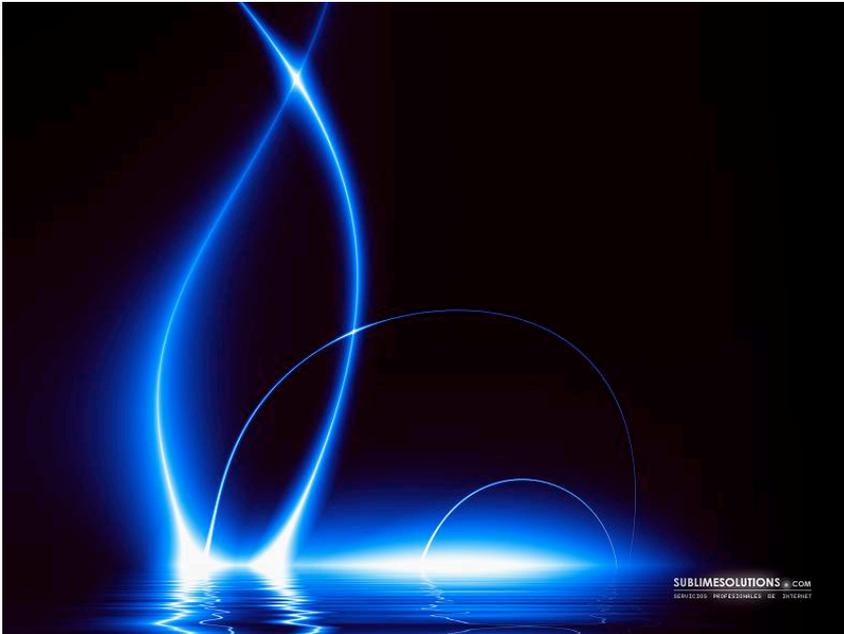
Si trazáramos un triángulo, tendría una distancia de dos metros o dos metros y medio y un desplazamiento horizontal de medio metro. A esta distancia, el desplazamiento vertical es de un metro.

### Detección del máximo ángulo de incidencia

A unos pocos centímetros de distancia, ya que en este caso sólo interesa el ángulo. El led azul es captado a con un ángulo máximo de cuarenta y cinco grados. El led blanco llega casi a un ángulo recto.

## Recorte de imágenes

Para la realización de los siguientes casos de prueba, se ha utilizado la siguiente imagen:



### Recorte de una imagen desde la esquina superior izquierda

El resultado obtenido tras aplicar un recorte desde la esquina superior izquierda ha sido el esperado, como podemos apreciar en la siguiente imagen:



### Recorte de una imagen desde la esquina superior derecha

El resultado obtenido tras aplicar un recorte desde la esquina superior derecha ha sido el esperado, como podemos ver en la imagen siguiente:



### Recorte de una imagen desde la esquina inferior izquierda

El resultado que hemos obtenido tras aplicar un recorte desde la esquina inferior izquierda coincide con el esperado, como podemos ver en la imagen siguiente:



### Recorte de una imagen desde la esquina inferior derecha

La imagen obtenida tras aplicar un recorte desde la esquina inferior derecha coincide con el resultado esperado, como podemos apreciar en la siguiente imagen:



### Recorte de una imagen en una zona central

La imagen que hemos obtenido tras aplicar un recorte en la zona central coincide con el resultado esperado, como podemos ver en la siguiente imagen:



**Nota: todas las imágenes que aparecen han sido redimensionadas para una mejor visualización y para no aumentar excesivamente el tamaño.**

## 3ª Iteración

### Integración: enviar órdenes

#### Generación de órdenes a partir de los movimientos del usuario

Las pruebas se han hecho con las gafas prototipo. En este caso, los LED's están situados más cerca de los extremos de las patillas para así tener el máximo campo posible. Precisamente, algunos de los valores obtenidos no concuerdan con lo que se esperaba: el parámetro 'pan', correspondiente al movimiento horizontal, tiene como valor mínimo -17 y como máximo 88, pero debería de estar centrado; de hecho, si nos situamos justo enfrente del Wiimote, la coordenada que devuelve es 20. Por lo tanto, tenemos el centro en 20, el límite izquierdo en -20 y el derecho en 90, si redondeamos los valores. Este problema puede deberse a algunas de las siguientes razones:

- Mala calibración por parte del Wiimote, de modo que el centro no está en la misma posición que el mando, sino que éste está desplazado hacia un lado.
- Cálculo erróneo a la hora de realizar la traducción de las coordenadas.

Por su parte, el parámetro 'tilt', correspondiente al movimiento vertical, sí toma valores que se corresponden con lo esperado, ya que cuando los LED's están por debajo del Wiimote obtenemos valores positivos, y, cuando están por encima, valores negativos.

#### Generación de órdenes a partir de movimientos simulados

En esta prueba, no hace falta usar ni el mando de la Wii (Wiimote) ni ningún LED, ya que las posiciones del usuario las generemos automáticamente con la ayuda de un bucle. En él, modificamos las supuestas coordenadas 'x' e 'y' según el índice 'i' que controla al bucle. Al contrario que en la prueba anterior, aquí sí conseguimos traducir correctamente a los parámetros 'pan' y 'tilt' tal y como hemos explicado anteriormente: valores negativos cuando el usuario se encuentra a la derecha y/o por encima del Wiimote, y valores positivos si está a la izquierda o por debajo del mando.

#### Envío de órdenes al servidor

En esta prueba, lo que se trata es que, además de generar las órdenes a partir de la posición del usuario, éstas se envíen al servidor, que se encargará de mover la cámara remota según la orden recibida. Tras conectarnos al servidor detallando qué dirección IP y qué puertos vamos a usar, hemos realizado varios movimientos tanto en horizontal, como en vertical y en profundidad. Hemos vuelto a utilizar las gafas prototipo. De nuevo, el parámetro 'pan' es el que da problemas,

ya que el centro de la coordenada 'x' no la calcula bien. Esto se podría solucionar con un 'offset' a la hora de traducir dicha coordenada a su correspondiente valor 'pan'. El parámetro 'tilt' sigue tomando valores adecuados según la coordenada 'y' en la que se encuentra el usuario. Por otra parte, las órdenes generadas se envían y llegan correctamente al servidor.

## 4ª Iteración

### Integración: envío de órdenes / recepción de frames

#### Captura de imágenes (frames) de forma local

En esta prueba, lo que se ha comprobado ha sido únicamente la posibilidad de capturar imágenes a través de la cámara web que se va a usar en el sistema final (Logitech Quickcam Pro 9000). Ha consistido en mostrar, en todo momento, la última imagen capturada por la cámara web en una ventana de prueba; a esta ventana también se le ha añadido la temporización entre imágenes de modo que se pudiera observar el tiempo transcurrido entre actualización y actualización.

#### Captura / recepción de imágenes (frames) de forma remota

Esta prueba es un paso más sobre la anterior, y en ella se ha hecho uso de la arquitectura cliente-servidor desarrollada para el sistema. La parte del servidor es la encargada de capturar las imágenes a través de la cámara web, y la parte del cliente es la encargada de recibirlas y mostrarlas. Con esta prueba, lo que se ha comprobado es que toda la parte relacionada con el envío/recepción de imágenes, dentro de la comunicación entre cliente y servidor, es correcta.

#### Envío de órdenes simuladas / captura de imágenes de forma remota

Ésta es la primera prueba en la que se ha probado la comunicación cliente-servidor en ambos sentidos simultáneamente, ya que, anteriormente, sólo se habían enviado órdenes o sólo se habían enviado imágenes, con lo que la comunicación era unidireccional. Además de comprobar que la comunicación bidireccional se lleva a cabo de forma correcta, también se ha comprobado que las imágenes enviadas por el servidor responden a movimientos simulados a través del envío de órdenes ficticias.

#### Envío de órdenes reales / captura de imágenes de forma remota

La prueba inicial del sistema al completo ha sido ésta, aunque en ella todavía no se ha comprobado la parte de información de errores ni el tratamiento de las imágenes obtenidas por la cámara web. Con esta prueba, lo que sí se ha comprobado ha sido el funcionamiento básico de la "ventana virtual", es decir, se ha comprobado que las imágenes que se envían desde el servidor y se muestran en la ventana del usuario son las correspondientes a su posición actual frente al monitor y que éstas cambian a la par que sus movimientos en todo momento.

## Integración y calibración del sistema

### Detección cámara-ordenador

Esta prueba ha sido uno de los puntos conflictivos en el desarrollo del sistema, debido a que las pruebas iniciales con la cámara web Logitech QuickCam Pro 9000 no fueron satisfactorias por problemas de detección/configuración de la misma. Por este motivo, las pruebas se han centrado en la detección de la cámara en el ordenador que actuará como servidor en el sistema final.

El problema se debía a una mala configuración de la cámara en el sistema que actuaba de servidor, lo que provocaba la aparición de un "pantallazo azul de la muerte"; sin embargo, tras realizar el proceso de instalación del servidor desde el principio (incluyendo el software de TrackerPod y el software de Logitech QuickCam Pro 9000), y fijándose con detenimiento en la configuración de la cámara, se ha conseguido solucionar el fallo, por lo que ya podemos hacer uso de ésta cámara sin ningún problema.

### Integración de órdenes reales / captura de imágenes / informe de log

Las pruebas sobre el informe de errores en el sistema no han sido satisfactorias en un primer momento, como consecuencia de un mal manejo en algunos de los mensajes de información devueltos por el servidor web integrado de la base móvil. Tras corregir los errores en el manejo de la información, se probó de nuevo el sistema y las pruebas fueron satisfactorias, por lo que ya es posible informar al usuario de los fallos que se produzcan en el sistema.

En cuanto a la integración, ésta no ha provocado la aparición de ningún fallo en la funcionalidad que ya ofrecía en sistema.

### Integración de algoritmos de recorte y zoom

La prueba de integración de los algoritmos de recorte y zoom se ha llevado a cabo de forma correcta y sin la aparición de errores en el funcionamiento del sistema. Esta prueba ha sido bastante simple, ya que, anteriormente, se habían probado por separado ambos algoritmos; lo único nuevo aquí ha sido comprobar que, realmente, las imágenes capturadas por la cámara web eran procesadas por estos algoritmos.

### Funciones avanzadas de la ventana de usuario

Esta prueba se ha centrado en probar las opciones avanzadas disponibles en la aplicación del usuario. Para ello, se han realizado pruebas en las que la dirección IP y puertos del servidor no eran los de por defecto y se han capturado imágenes de la ventana virtual para guardar en archivos.

Los resultados de todas las pruebas realizadas han sido satisfactorios, y, por tanto, actualmente disponemos de un sistema versátil que no depende de una configuración fija por defecto y que permite hacer capturas de lo que se ve a través de la ventana virtual, entre otras cosas.

### Calibración del sistema completo

Ésta última prueba ha sido la más compleja de todas. Ha consistido en probar el funcionamiento al completo del sistema y calibrar este funcionamiento, cambiando los parámetros de las fórmulas implementadas para procesar el movimiento del usuario y las imágenes. Para conseguir este objetivo, el modo de trabajo ha sido mediante prueba y error, y contando con la opinión de todos los integrantes del grupo de forma que la calibración fuese lo más objetiva posible.

Por otra parte, se ha solucionado el problema relacionado con el parámetro 'pan', ya que el parámetro de lectura correspondiente (la coordenada 'x') se recibía desplazada con respecto al centro.

También se han observado ciertos movimientos bruscos de la cámara cuando el usuario se salía del ángulo de recepción del mando Wiimote, hecho que también ha sido solucionado por los implementadores en el transcurso de la prueba.

Finalmente, y tras la aprobación general del grupo, la prueba ha sido considerada como satisfactoria y, por tanto, el sistema ya debería estar listo para su despliegue en el entorno real del usuario.

# Resultados de pruebas

## Listado de requisitos

Identificador	Requisito
R1	Detectar la posición del usuario
R2	Mostrar las imágenes remotas en pantalla
R3	Mover la cámara a partir de los movimientos del usuario
R4	Responder a los movimientos horizontales, verticales y en profundidad
R5	Capturar por pantalla la ventana virtual

## Listado de casos de prueba

Identificador	Nombre
CP1	Conexión Mando-PC en Linux
CP2	Conexión Mando-PC en Windows
CP3	Movimiento automático de la base de la cámara
CP4	Movimiento de la base de la cámara ante una acción propuesta por el usuario
CP5	Movimiento de la base de la cámara ante una serie de acciones del usuario
CP6	Establecimiento de una conexión Cliente-Servidor con envío de frames
CP7	Detección del movimiento horizontal de los LED's
CP8	Detección del movimiento vertical de los LED's
CP9	Detección del movimiento en profundidad de los LED's
CP10	Detección de la posición de varios LED's simultáneamente
CP11	Detección de la posición de los LED's en un entorno con otras fuentes de luz
CP12	Detección de máxima distancia de los LED's
CP13	Detección del máximo ángulo de incidencia
CP14	Recorte de una imagen desde la esquina superior izquierda
CP15	Recorte de una imagen desde la esquina superior derecha
CP16	Recorte de una imagen desde la esquina inferior izquierda
CP17	Recorte de una imagen desde la esquina inferior derecha
CP18	Recorte de una imagen en una zona central
CP19	Generación de órdenes a partir de los movimientos del usuario
CP20	Generación de órdenes a partir de movimientos simulados
CP21	Envío de órdenes al servidor
CP22	Captura de imágenes (frames) de forma local
CP23	Captura / recepción de imágenes (frames) de forma remota
CP24	Envío de órdenes simuladas / captura de imágenes de forma remota
CP25	Envío de órdenes reales / captura de imágenes de forma remota
CP26	Detección cámara-ordenador
CP27	Integración de órdenes reales / captura de imágenes / informe de log
CP28	Integración de algoritmos de recorte y zoom
CP29	Funciones avanzadas de la ventana de usuario
CP30	Calibración del sistema completo

## Resultados de las pruebas

Caso de prueba	Resultado	Causa/s en caso de fallo	Observaciones	Ingeniero de pruebas
CP1	Correcto	-	-	Rafael Martínez González
CP2	Correcto	-	-	Rafael Martínez González
CP3	Correcto	-	-	Rafael Martínez González
CP4	Correcto	-	-	Rafael Martínez González
CP5	Correcto	-	-	Rafael Martínez González
CP6	Correcto	-	-	Rafael Martínez González
CP7	Correcto	-	-	Jesús Manuel Rodríguez Sánchez
CP8	Correcto	-	-	Jesús Manuel Rodríguez Sánchez
CP9	Correcto	-	La profundidad de los LED's se mide de forma imprecisa a veces, debido a que sólo se detecta uno de ellos	Jesús Manuel Rodríguez Sánchez
CP10	Correcto	-	-	Jesús Manuel Rodríguez Sánchez
CP11	Correcto	-	Existen pequeñas interferencias en caso de que la fuente de luz sea bastante intensa y se encuentre muy cerca de la fuente infrarroja y del Wiimote	Jesús Manuel Rodríguez Sánchez
CP12	Correcto	-	-	Israel Rodríguez Martín
CP13	Correcto	-	-	Israel Rodríguez Martín
CP14	Correcto	-	-	Israel Rodríguez Martín
CP15	Correcto	-	-	Israel Rodríguez Martín
CP16	Correcto	-	-	Israel Rodríguez Martín
CP17	Correcto	-	-	Israel Rodríguez Martín
CP18	Correcto	-	-	Israel Rodríguez Martín
CP19	Correcto	Se han usado dos mandos de televisión, ya que los LED's no estaban disponibles; también es posible que se hayan traducido mal las coordenadas (corregido)	El parámetro 'pan' no toma valores negativos cuando el usuario se encuentra a la derecha del Wiimote	Rafael Martínez González e Israel Rodríguez Martín
CP20	Correcto	Es posible que se hayan traducido mal las coordenadas o que el mando esté mal calibrado	El parámetro 'pan' no toma valores negativos siempre que el usuario se encuentra a la derecha del Wiimote	Rafael Martínez González e Israel Rodríguez Martín
CP21	Correcto	Es posible que se hayan traducido mal las coordenadas o que el mando esté mal calibrado (corregido)	El parámetro 'pan' no toma valores negativos siempre que el usuario se encuentra a la derecha del Wiimote	Rafael Martínez González e Israel Rodríguez Martín
CP22	Correcto	-	Han aparecido problemas con JMF y la cámara Logitech Quickcam Pro 9000 en algunos equipos	Jesús Manuel Rodríguez Sánchez
CP23	Correcto	-	-	Jesús Manuel Rodríguez Sánchez
CP24	Correcto	El cliente sólo puede conectar la primera vez con el servidor (corregido)	Error en sucesivas conexiones con el servidor, corregidas posteriormente	Jesús Manuel Rodríguez Sánchez
CP25	Correcto	-	-	Rafael Martínez González, Israel Rodríguez Martín y Jesús Manuel Rodríguez Sánchez

Caso de prueba	Resultado	Causa/s en caso de fallo	Observaciones	Ingeniero de pruebas
CP26	Correcto	Configuración errónea de la cámara (corregido)	Los problemas aparecidos con la cámara Logitech Quickcam Pro 9000 han sido solucionados	Jesús Manuel Rodríguez Sánchez
CP27	Correcto	Tratamiento incorrecto de los mensajes de información de la base móvil (corregido)	-	Jesús Manuel Rodríguez Sánchez
CP28	Correcto	-	-	Rafael Martínez González, Israel Rodríguez Martín y Jesús Manuel Rodríguez Sánchez
CP29	Correcto	-	-	Jesús Manuel Rodríguez Sánchez
CP30	Correcto	-	Cuando el usuario se sale del ángulo de recepción del mando y vuelve a él, la cámara realiza movimientos bruscos (se ha corregido posteriormente). Además, se ha corregido el problema del parámetro 'pan' de pruebas anteriores	Rafael Martínez González, Israel Rodríguez Martín y Jesús Manuel Rodríguez Sánchez

# Matriz de trazabilidad

Matriz de trazabilidad

		Requisitos				
		R1	R2	R3	R4	R5
Casos de prueba	CP1					
	CP2					
	CP3					
	CP4					
	CP5					
	CP6					
	CP7					
	CP8					
	CP9					
	CP10					
	CP11					
	CP12					
	CP13					
	CP14					
	CP15					
	CP16					
	CP17					
	CP18					
	CP19					
	CP20					
	CP21					
	CP22					
	CP23					
	CP24					
	CP25					
	CP26					
	CP27					
	CP28					
	CP29					
	CP30					