

**UNIVERSIDAD AUTONOMA DE MADRID**

**ESCUELA POLITECNICA SUPERIOR**



## **PROYECTO FIN DE CARRERA**

**DISEÑO, IMPLEMENTACIÓN Y VALIDACIÓN DE UN SISTEMA DE  
CONTROL PARA SMARTPHONE BASADO EN LA TECNOLOGÍA DE  
GAZE-TRACKING**

**Tamara Moreno Martín**

**ENERO 2013**



# **DISEÑO, IMPLEMENTACIÓN Y VALIDACIÓN DE UN SISTEMA DE CONTROL PARA SMARTPHONE BASADO EN LA TECNOLOGÍA DE GAZE-TRACKING**

**AUTOR: Tamara Moreno Martín**

**TUTOR: Pablo Varona Martínez**

**Grupo de Neurocomputación Biológica**

**Dpto. de Ingeniería Informática**

**Escuela Politécnica Superior**

**Universidad Autónoma de Madrid**

**Enero de 2013**



# Agradecimientos

En primer lugar, quería darle las gracias a mi tutor Pablo Varona, por darme la oportunidad de hacer este trabajo y por ayudarme a hacer esta memoria. También darle las gracias a David Rozado por haberme mostrado todos sus conocimientos acerca de la tecnología de *Eye Tracking* y *Gaze Gestures* y a Javier San Agustín por permitirme utilizar y también ayudarme con el software *ITU Gaze Tracker*. No quería olvidarme de Paco y Uwe, que me ayudaron a construir el prototipo.

Por otro lado, quiero agradecerles a mis padres, Goyo y Fuencis, el hecho de haberme dado la oportunidad de poder estudiar esta carrera y por todo su apoyo y dedicación durante estos años. También quiero agradecer a mis hermanos, Sara y Eduardo su apoyo incondicional.

A Antonio, por no haber permitido que caiga en la pereza y animarme a terminar las cosas que empiezo, así como el haberse leído esta memoria. Gracias por estar a mi lado durante este tiempo.

A mis Vikingas (y no solo Vikingas), mis amigas de siempre, de Zarzuela, sois únicas, gracias por estar siempre ahí, dispuestas a salir y por todas nuestras risas, ideas disparatadas, canciones raras, fiestas y cotilleos varios.

A todos los amigos que he conocido a lo largo de la carrera, que son muchos, gracias por tanto. Quiero agradecer especialmente a Raquel sus consejos y apoyo, a Celia ser tan divertidamente sincera, a Javi, que siempre le hacen gracia mis cosas y a Bruno, ser tan detallista (y su café siempre tan mítico). No puedo mencionar aquí a todos, pero desde aquel ya lejano Septiembre de 2006, habéis sido mis compañeros de prácticas, clase, comidas, descansos, etc. haciendo balance en todo este tiempo, solo me quedo con cosas buenas.

Por último, y no menos importante, a mis compañeros, ya también amigos, de Deimos, por tantas conversaciones y risas a las horas del café y la comida, me hacéis mucho más amena la jornada laboral.

**Tamara Moreno Martín**

**Enero 2013**



*Es más fácil obtener lo que se desea con una  
sonrisa que con la punta de la espada*

William Shakespeare





# Palabras Clave

Interacción, Teléfono, Coste, Seguimiento, Mirada, Gesto, Ojo, Prototipo, Suavizado, Interpolación, Cámara, Infrarrojo, Luz, Navegación, Fijación

## Resumen

El objetivo de este proyecto es la implementación y prueba de un sistema propuesto de bajo coste que permita a una persona interactuar con un teléfono móvil inteligente o *Smartphone* a través de los ojos. La posibilidad de ejecutar comandos con los ojos en un teléfono tipo *Smartphone*, de forma análoga a lo que se hace de forma táctil en una pantalla, es una opción muy interesante para personas con movilidad reducida o que no puedan utilizar las manos en ese momento.

Este sistema utiliza la tecnología de *Eye Tracking* (seguimiento de los ojos), para interactuar a través de gestos oculares, también llamados *Gaze Gestures*. Mediante un software de *Eye Tracking* se obtienen las posiciones a lo largo del tiempo del ojo. Una vez obtenidas dichas coordenadas, a través del algoritmo de reconocimiento de los *Gaze Gestures* se puede saber si el usuario ha realizado o no un gesto.

Lo primero que se ha llevado a cabo en este trabajo es la construcción de un soporte que sirva, además de para sujetar el teléfono, para tener acopladas las luces y la cámara IR que necesita para su funcionamiento. Dicho dispositivo construido de forma artesanal, ha tenido un coste final estimado de 40€.

Una vez implementado el prototipo, se ha pasado a validar el funcionamiento del reconocimiento de gestos oculares a través de una serie de pruebas que han mostrado cuanto de preciso es el sistema, si se dan falsos positivos en él, si es rápido o no y si influye en el mismo el aprendizaje. Existen dos tipos de realización del gesto: con y sin tiempo de fijación (*dwell time*). El sistema propuesto es algo más preciso con *dwell time*, pero del orden del doble de lento que sin *dwell time*, además de arrojar un número muy bajo de falsos positivos en ambos casos.

## Key words

Interaction, Smartphone, Cost, Tracking, Gaze, Gesture, Eye, Setup, Smoothing, Interpolation, Camera, Infrared, Light, Browsing, Dwell

## Abstract

The purpose of this project is the implementation and validation of a low cost system that allows a person to interact with a Smartphone with his/her eyes. Executing commands with eyes on a Smartphone, similar to what is done by touch, is an interesting choice for disabled people or people who cannot use their hands for some reason.

This system uses Eye Tracking technology, to interact through eye gestures, also called Gaze Gestures. The Eye Tracking software keeps track of the positions of the eye in time. Through these coordinates, the Gaze Gestures algorithm can tell if the user has made a gesture or not.

Firstly a support is built for holding the telephone and integrating the lights and IR camera that the system needs to function. This device is built by hand and has an estimated cost of 40€.

Once the prototype is implemented, the next step is validating the performance of Gaze Gesture recognition via some tests that show how accurate the system is, if some false positives occur, if the system is fast or slow and the effect of learning. There are two different forms to create gaze gestures: with or without dwell time. The systems works better with dwell time but is two times slower than the system with no dwell time, and the number of false positives is low in both cases. However, accuracy is better with dwell time.

# ÍNDICE DE CONTENIDOS

<b>1. INTRODUCCIÓN</b> .....	<b>1</b>
1.1. MOTIVACIÓN .....	1
1.2. OBJETIVOS.....	1
1.3. ORGANIZACIÓN DE LA MEMORIA .....	2
<b>2. ESTADO DEL ARTE</b> .....	<b>3</b>
2.1. INTRODUCCIÓN.....	3
2.2. INTERACCIÓN HOMBRE-MÁQUINA TRADICIONAL.....	4
2.3. INTERFACES BASADAS EN <i>EYE TRACKING</i> .....	7
2.4. <i>GAZE GESTURES</i> .....	10
2.4.1. <i>Redes de memorias temporales jerárquicas, HTM</i> .....	11
2.4.2. <i>Algoritmo de Needleman-Wunsch</i> .....	12
<b>3. DISEÑO</b> .....	<b>15</b>
3.1. CONSTRUCCIÓN DEL SOPORTE.....	15
3.1.1. <i>Luces</i> .....	15
3.1.2. <i>Cámara</i> .....	17
3.1.3. <i>Montaje y estructura</i> .....	18
3.2. COMUNICACIÓN ENTRE EL MÓVIL Y EL ORDENADOR.....	19
3.3. PROCESO DE RECONOCIMIENTO DE GESTOS .....	20
3.3.1. <i>Captura de imágenes con la cámara</i> .....	21
3.3.2. <i>Eye Tracking</i> .....	21
3.3.2.1. <i>Gaze Tracker: interfaz</i> .....	21
3.3.2.2. <i>Gaze Tracker: procesado de la imagen</i> .....	23
3.3.2.2.1. <i>Métodos de interpolación</i> .....	24
3.3.2.3. <i>Obtención del punto de mira en la pantalla</i> .....	25
3.3.2.3.1. <i>Algoritmos basados en velocidad y dispersión</i> .....	26
3.3.3. <i>Gaze Gestures</i> .....	28
3.3.3.1. <i>Algoritmo de reconocimiento de los Gaze Gestures</i> .....	28
3.3.3.2. <i>Adaptación del reconocimiento de Gaze Gestures al Smartphone</i> .....	30
<b>4. RESULTADOS</b> .....	<b>33</b>
4.1. INICIALIZACIÓN DEL SISTEMA .....	33
4.1.1. <i>Configuración del teléfono</i> .....	33
4.1.2. <i>Configuración del Eye Tracker</i> .....	33
4.1.3. <i>Configuración del algoritmo de reconocimiento de Gaze Gestures</i> .....	34
4.2. VALIDACIÓN DEL CONTROL CON <i>GAZE GESTURES</i> .....	34
4.3. DESCRIPCIÓN DE LAS PRUEBAS .....	34
4.3.1. <i>Por tipo de la realización del gesto</i> .....	35
4.3.2. <i>Por tipo de uso</i> .....	35
4.3.3. <i>Medida de la rapidez de un gesto</i> .....	36
4.3.4. <i>Pruebas de aprendizaje</i> .....	36
4.4. RESULTADOS OBTENIDOS.....	36
4.4.1. <i>Experimento de accuracy</i> .....	37
4.4.2. <i>Experimento de browsing</i> .....	38
4.4.3. <i>Tiempo de realización de un gesto</i> .....	39

4.4.4.	<i>Efectos del aprendizaje</i> .....	41
4.4.4.1.	Aprendizaje en el experimento de <i>accuracy</i> .....	41
4.4.4.2.	Aprendizaje en el tiempo de realización de un gesto.....	42
4.4.5.	<i>Opinión de los usuarios</i> .....	43
<b>5.</b>	<b>CONCLUSIONES Y TRABAJO FUTURO</b> .....	<b>47</b>
5.1.	CONCLUSIONES.....	47
5.2.	TRABAJO FUTURO .....	49
	<b>REFERENCIAS</b> .....	<b>51</b>
	<b>GLOSARIO DE ACRÓNIMOS</b> .....	<b>55</b>
	<b>PUBLICACIONES</b> .....	<b>57</b>
	<b>PRESUPUESTO</b> .....	<b>I</b>
	<b>PLIEGO DE CONDICIONES</b> .....	<b>III</b>

## ÍNDICE DE FIGURAS

Figura 1.	Representación de la evolución de las interfaces .....	4
Figura 2.	Teclado clásico de móvil.....	6
Figura 3.	Ejemplo de gesto táctil en <i>Smartphone</i> .....	7
Figura 4.	Amplificación por fotomultiplicación (Smith and Warter, 1959).....	9
Figura 5.	Esquema de un sistema genérico de <i>Eye Tracking</i> (Santis and Iacoviello, 2009) .....	9
Figura 6.	Gaze Gestures de un trazo (Mollenbach, 2010).....	10
Figura 7.	HTM: Modelo generador (izda.) y red bayesiana (dcha.). (Rozado et al., 2011) .....	11
Figura 8.	Algoritmo de <i>Needleman-Wunsch</i> .....	13
Figura 9.	Esquema del circuito de luces IR.....	16
Figura 10.	Resultado final luces IR.....	17
Figura 11.	Cámara <i>Sandberg NightCam 2</i> .....	17
Figura 12.	Prototipo de <i>Gaze Tracking</i> propuesto sin teléfono (izda.) y con teléfono (dcha.)...	18
Figura 13.	Esquema que representa la interconexión entre ordenador y teléfono móvil .....	19
Figura 14.	Interfaz de <i>TeamViewer</i> .....	20
Figura 15.	Pasos del reconocimiento de gestos .....	20
Figura 16.	Interfaz del programa <i>ITU Gaze Tracker</i> .....	21
Figura 17.	Detalle de los submenús del menú de configuración del <i>ITU GazeTracker</i> .....	22
Figura 18.	Pasos que sigue el programa <i>ITU Gaze Tracking</i> .....	23
Figura 19.	Ejemplo de uso del programa <i>ITU Gaze Tracker</i> para un solo ojo .....	23
Figura 20.	Efecto de la etapa de suavizado. Adaptado de (Agustin, 2010).....	27

Figura 21. Efecto del suavizado en la coordenada y respecto al tiempo. Adaptado de (Agustin, 2010) .....	28
Figura 22. Áreas de la pantalla del móvil .....	29
Figura 23. Lista de los <i>Gaze Gestures</i> definidos en el algoritmo.....	30
Figura 24. Pantalla de calibración de <i>ITU Gaze Tracker</i> en el ordenador .....	31
Figura 25. Ejemplo de realización de un gesto en la pantalla de un móvil.....	35
Figura 26. Media y error en el porcentaje de acierto para el experimento de <i>accuracy</i> .....	38
Figura 27. Media y error para el tiempo de la realización de un gesto, sin y con <i>dwell time</i> .....	40
Figura 28. Porcentaje de acierto medio y errores para el experimento de <i>accuracy</i> con y sin <i>dwell time</i> .....	42
Figura 29. Comparativa de los efectos de aprendizaje para el tiempo por gesto con y sin <i>dwell time</i> .....	43
Figura 30. Gráfica comparativa de la opinión del 1 al 5 de los usuarios acerca del sistema con y sin <i>dwell time</i> .....	45

## ÍNDICE DE TABLAS

Tabla 1. Porcentaje de acierto para el experimento de <i>accuracy</i> , sin y con <i>dwell time</i> .....	37
Tabla 2. Número de falsos positivos en el experimento de <i>browsing</i> , sin y con <i>dwell time</i> .....	39
Tabla 3. Tiempo de la realización de un gesto, sin y con <i>dwell time</i> , en segundos .....	40
Tabla 4. Porcentaje de acierto para el aprendizaje en el experimento de <i>accuracy</i> sin <i>dwell time</i> .....	41
Tabla 5. Porcentaje de acierto para el aprendizaje en el experimento de <i>accuracy</i> con <i>dwell time</i> .....	41
Tabla 6. TPG en el aprendizaje en el experimento de <i>accuracy</i> sin <i>dwell time</i> .....	42
Tabla 7. TPG en el aprendizaje en el experimento de <i>accuracy</i> con <i>dwell time</i> .....	42
Tabla 8. Opinión del 1 al 5 de los usuarios acerca del sistema sin <i>dwell time</i> .....	44
Tabla 9. Opinión del 1 al 5 de los usuarios acerca del sistema con <i>dwell time</i> .....	44
Tabla 10. Precios de diferentes sistemas de <i>Eye Tracking</i> .....	48



# 1. Introducción

---

## 1.1. Motivación

La posibilidad de interactuar con una máquina sin necesidad de hacer uso de las interfaces típicas destinadas a tal efecto (teclado, ratón, micrófono, etc.) es una atractiva alternativa para situaciones en las cuales es imposible o incómodo utilizar las manos o la voz. Esta situación se puede presentar en el caso de personas con algún tipo de discapacidad física, tales como tetraplejas, amputaciones de extremidades, lesiones crónicas, etc. o también en personas que por necesidad o simplicidad prefieran no tener que interactuar de esa forma.

En el caso de un dispositivo móvil (un *Smartphone*, más concretamente), el problema es aún mayor, ya que debido a su reciente implantación en el mercado, no existe de momento mucha literatura que hable de este supuesto. Adicionalmente, existe el problema añadido de ser un dispositivo portátil: la luz exterior; además de problemas para calibrarlo y de tener, generalmente, una pantalla más reducida que un ordenador convencional.

En este proyecto se planteará una manera de posibilitar esa interacción: con los ojos, más concretamente a través de gestos pupilares (Hiley et al., 2006). El reconocimiento de gestos pupilares requiere el seguimiento de los ojos, también llamado *Eye Tracking*. Ese seguimiento de los ojos a lo largo del tiempo con un objetivo se conoce como seguimiento de la mirada o *Gaze Tracking*.

*Gaze Tracking* es una tecnología aún poco utilizada para el control de dispositivos portátiles (tabletas, *Smartphones*, libros electrónicos), y ordenadores en general, que puede facilitar el control intuitivo de los mismos. Existen secuencias de movimientos oculares ya predefinidas, conocidas como *Gaze Gestures*, basadas en la tecnología de *Gaze Tracking* y cuyo funcionamiento ya ha sido probado con ordenadores en la literatura (Rozado et al., 2012a). En definitiva, los *Gaze Gestures* son señales (gestos) hechas únicamente a base de movimientos oculares, cuyo uso para el control de dispositivos puede ser interesante en las situaciones anteriormente descritas.

## 1.2. Objetivos

El objetivo del proyecto es el diseño, implementación y validación de un sistema para un teléfono móvil tipo *Smartphone* que utilice los citados *Gaze Gestures*. El *Smartphone* que se va a utilizar tendrá sistema operativo *Android*, por estar basado en *Linux*, tener licencia libre y ser de código abierto.

En la fase de diseño, se ideará y construirá un prototipo para *Eye Tracking* para *Smartphone* de bajo coste, buscando los elementos más apropiados tales como la cámara, el soporte o los LEDs infrarrojos que servirán de iluminación, teniendo en cuenta una serie de limitaciones que se detallarán posteriormente.

En la fase de validación, se probará el funcionamiento del sistema implementado. Para poder valorar su funcionamiento, se realizarán pruebas con usuarios, las cuales pueden dividirse en tres grandes grupos: las primeras, medirán la tasa de acierto del sistema, las segundas, servirán para obtener la tasa de falsos positivos, y las terceras, irán encaminadas a dar una idea acerca de la rapidez del sistema. Además, también se medirá la influencia del aprendizaje en dicha tecnología. Con estos resultados, se puede extraer una medida de la fiabilidad y robustez del sistema implementado.

## 1.3. Organización de la memoria

La memoria consta de los siguientes capítulos:

- **Capítulo 1.** Introducción: Motivación, objetivos y organización de la memoria.
- **Capítulo 2.** Estado del arte: Repaso a las interfaces tradicionales, recorrido a la historia de las interfaces basadas en *Eye Tracking* y presentación de los *Gaze Gestures*.
- **Capítulo 3.** Diseño: Construcción del soporte y descripción del proceso de reconocimiento de gestos.
- **Capítulo 4.** Resultados: Inicialización del sistema, descripción de las pruebas y resultados obtenidos a partir de las mismas.
- **Capítulo 5.** Conclusiones y trabajo futuro: Breve repaso a las ventajas e inconvenientes del sistema propuesto y posibles mejoras del mismo de cara al futuro.



## 2. Estado del arte

---

### 2.1. Introducción

La interacción hombre-máquina (*Human Computer Interaction* , HCI) se define como la forma que tiene una persona de interactuar con un ordenador/es o con máquinas que contengan dichos ordenadores (Yang and Chen, 2009). La forma más habitual desde el origen de los ordenadores, sobre los años 50, para comunicarse con ellos, ha sido, salvo alguna excepción, manual. Esto es debido a que el ser humano, desde el inicio de los tiempos ha utilizado las manos para cualquier uso cotidiano: trabajar, escribir, comer, etc. Por este motivo, las interfaces típicas como el ratón o el teclado han sido prácticamente universales para la comunicación con cualquier tipo de máquina.

El teclado ha sido la primera interfaz asociada a un ordenador, la disposición de las teclas está basada en la de las antiguas máquinas de escribir, que surgieron a finales del siglo XIX. El ratón, en cambio, tiene sus orígenes en la década de los 60, como parte de un proyecto que buscaba la mejora de las comunicaciones hombre-máquina. A partir de este hecho surge una pregunta: una persona que no pueda (por motivos de invalidez o de ocupación en ese momento) o no quiera utilizar sus manos (simplemente por comodidad), ¿podría interactuar con una máquina?

Una opción muy utilizada es el control por voz (Gaikwad et al., 2010), que requiere condiciones de uso delimitadas. Otra opción muy interesante y complementaria sería la posibilidad de interactuar con los ojos, más conocida como *Eye Tracking*. Mediante esta tecnología se consigue que un cursor apunte al lugar del monitor del ordenador al cual se esté mirando. De esta posibilidad se hablará a continuación más detalladamente.

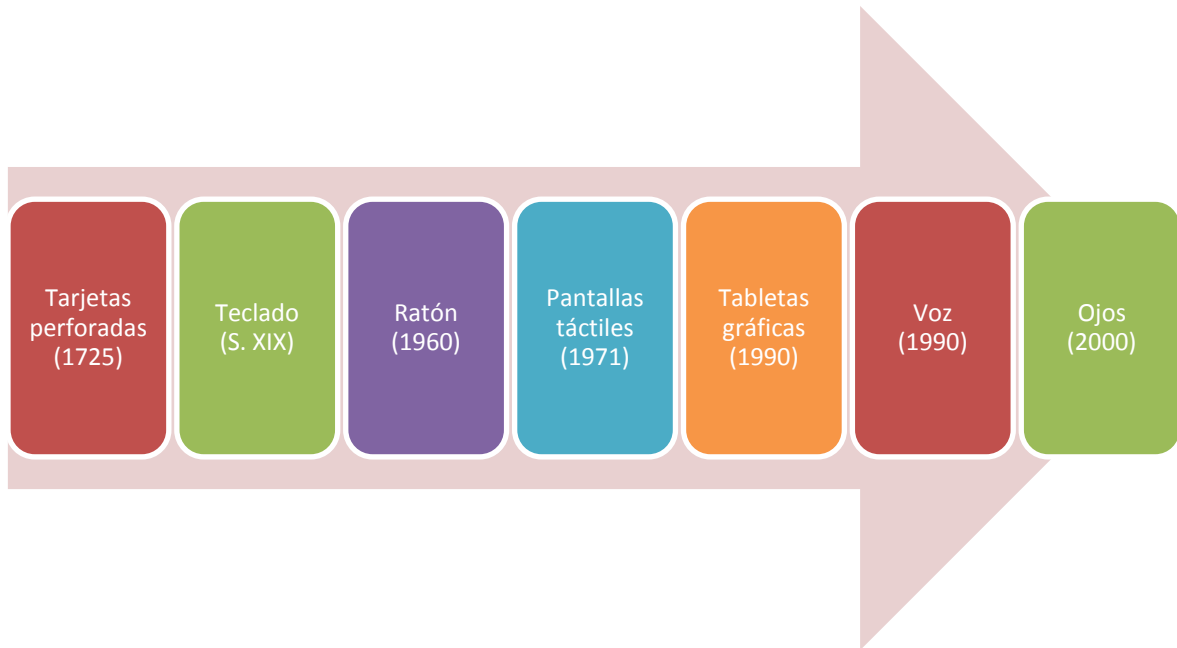
Por otra parte, la tecnología táctil ha cobrado un nuevo sentido gracias al aumento de dispositivos como *Smartphones* o tabletas, que se basan en el uso de los gestos dactilares para todo tipo de funciones, tales como acceder a aplicaciones, leer el correo, navegar por internet o bloquear y desbloquear el móvil mediante un patrón de desbloqueo, por poner solo algunos ejemplos básicos del uso.

A partir del *Eye Tracking*, existen unas secuencias de gestos oculares ya predefinidas llamadas *Gaze Gestures*, las cuales se basan en la idea de los gestos dactilares anteriormente descrita. Dichas secuencias facilitarían ese control intuitivo mediante los ojos de la máquina en cuestión, que en este caso será un dispositivo móvil.

En un dispositivo móvil tipo Smartphone, esta utilidad sería aún más intuitiva si cabe, ya que como se ha comentado antes, con dichos dispositivos se interactúa mediante gestos dactilares. La idea básica es extender el uso de gestos en un Smartphone más allá de las manos, es decir, con los ojos.

## 2.2. Interacción hombre-máquina tradicional

La comunicación entre los seres humanos está basada fundamentalmente en una mezcla del lenguaje visual y el auditivo (Jaimes and Sebe, 2007), de ahí provienen los avances en sistemas de procesamiento de imagen, audio ó video. Este lenguaje no solo está formado por las palabras (lenguaje verbal), sino también por los gestos, expresiones o comportamiento, es decir, el lenguaje no verbal. Este lenguaje no verbal tiene un significado también en cada cultura, es decir, dependiendo del interlocutor al que se refiera un mismo símbolo puede representar cosas diferentes (Rehm and Leichtenstern, 2011).



**Figura 1.** Representación de la evolución de las interfaces

En la Figura 1, pueden observarse las interfaces hombre máquina más relevantes ordenadas de forma cronológica, las cuales serán detalladas a continuación. Cabe destacar que de las interfaces expuestas algunas ya han quedado obsoletas y que no todas se utilizan en todas las máquinas.

Las tarjetas perforadas fueron usadas por primera vez sobre el año 1725 para controlar telares textiles en Francia, aunque su uso como interfaz hombre-máquina surge a partir del año 1900. Entre esa fecha y 1950 fue la forma más utilizada para introducir y almacenar información en un ordenador. Este sistema fue reemplazado primero por las cintas magnéticas, y posteriormente, los discos duros.

El teclado, como ya se comentó en el apartado anterior, tiene su origen en las antiguas máquinas de escribir mecánicas que surgieron a finales del siglo XIX. El teclado como interfaz para ordenador personal apareció en el mercado en el año 1981, pero no fue hasta finales de esa década cuando se convirtió junto con el ratón, en la interfaz más usada.

El ratón, ya mencionado anteriormente, fue diseñado en la década de 1960. En 1981 *Apple* lanzó la primera computadora con ratón incluido, pero su uso no se popularizó hasta 1984, con la llegada del sistema operativo *Macintosh*, de hecho, en la actualidad es una de las interfaces más usadas para trabajar con una máquina.

La interfaz táctil para pantallas se crea en el año 1971, siendo de uso muy común desde entonces en cajeros automáticos, PDAs o terminales de puntos de venta. Últimamente su uso se ha popularizado debido al auge de los teléfonos móviles táctiles, como el de la Figura 3 o las tabletas.

Las tabletas gráficas surgieron debido a la necesidad de una interacción más “humana” con el ordenador, es decir, la capacidad de procesar cualquier gráfico, véase una firma, un dibujo, etc. Son usadas en algunos comercios a la hora de firmar en el pago con tarjeta.

La voz utilizada en interfaces hombre máquina está cobrando una mayor relevancia, ya que con la llegada de los teléfonos inteligentes y tabletas, están surgiendo aplicaciones de reconocimiento de voz que sirven desde para convertir un mensaje vocal en texto hasta para dar órdenes al móvil, así como poner una alarma o encender la cámara de fotos. Este sistema se encuentra disponible en móviles como el *Samsung Galaxy S3* con *SVoice* o el *iPhone 4* con *Siri*.

Por último, existe la posibilidad de interactuar con la máquina a través de los ojos, tema que será abordado en este proyecto más adelante. Cabe precisar que estos modos de interacción con el ordenador no son excluyentes, sino que la combinación de ellos puede ser más óptima que el uso de uno solo, véase el uso conjunto de las interfaces de teclado y ratón, comúnmente usadas en ordenadores personales.

La comunicación entre una persona y una máquina está basada más bien en otro tipo de interacción, que es, en principio, manual (teclado, ratón, etc.). Pero debido sobre todo al tipo de interacción existente entre humanos, las nuevas aplicaciones se están alejando cada vez más de ese estándar, propiciado en cierta manera por la evolución de los ordenadores, que cada vez son más rápidos, tienen más memoria y también más capacidades (Jaimes and Sebe, 2007).

Las máquinas actuales están habilitadas para soportar cualquier tipo de interacción. Los ordenadores personales a diferencia de los antiguos ordenadores por lotes, permiten el control de la máquina en tiempo real por parte del usuario.

De todas formas, como ya se comentó en la introducción de este capítulo, la forma común y más utilizada que tiene el ser humano de interactuar con una máquina es a través de las manos, ya sea a través de palabras mediante el teclado o de gestos hechos con el ratón, que son los dos ejemplos más característicos.

La interacción hombre máquina idealmente debería ser similar a la existente entre humanos, salvo en que un ordenador no tiene capacidad creativa, ni de escuchar, ni atiende a explicaciones que se le pueda dar, como hace un humano. Por otro lado, una máquina ni se cansa ni se ofende como podría hacerlo un ser humano. (Moore and Churchill, 2011)

Ahora se pasará a hablar de otro tipo de máquinas más concreto, los nuevos teléfonos móviles con sistema operativo propio, también llamados *Smartphones*. Los primeros teléfonos móviles cumplían sólo la función de llamar por teléfono, para la cual se utilizaba el clásico teclado numérico. Pronto, a esa función básica inicial poco a poco se le fueron añadiendo otro tipo de funcionalidades como pudieron ser el envío de mensajes de texto o SMS, juegos, acceso a las opciones de configuración, ver la agenda, etc. cuya interacción en un principio se resolvió con dicho teclado.



**Figura 2.** Teclado clásico de móvil

En el teclado de la Figura 2 puede verse, por ejemplo, que con las teclas del 0 al 9, el asterisco y la almohadilla, podía teclearse cualquier tipo de palabra o mensaje, valiéndose de 12 teclas combinadas de distinta forma únicamente para ese fin.

La llegada al mercado de los *Smartphones*, teléfonos móviles inteligentes, y otros dispositivos portátiles táctiles como las tabletas, supuso una revolución respecto a la forma de interactuar con el teléfono móvil, se pasó de utilizar un teclado como el representado en la Figura 2, análogo al teclado de un ordenador personal, a interactuar a través de gestos táctiles, es decir, similar a la interacción que se produce en un ordenador personal a través de un ratón.

En un Smartphone, además de la posibilidad de utilizar el dedo como su fuera el ratón de un ordenador personal, también tiene la capacidad adicional de utilizar gestos dactilares, ya sea para navegar por una página web, ampliar la pantalla, o desbloquear el teléfono. En la Figura 3, por ejemplo, aparece un ejemplo del uso de un gesto en el patrón de desbloqueo de un teléfono *Android*.

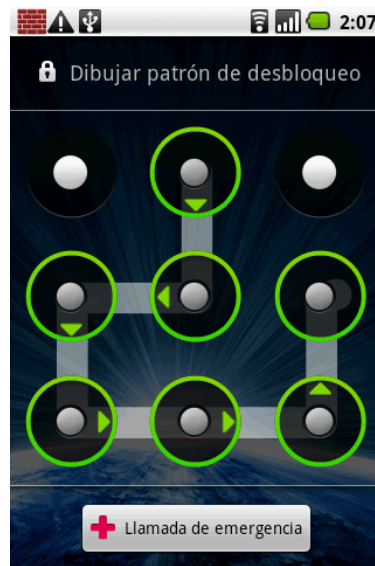


Figura 3. Ejemplo de gesto táctil en Smartphone

### 2.3. Interfaces basadas en *Eye Tracking*

Como ya se ha comentado en el capítulo anterior, hay situaciones en las que no se puede, es incómodo, o sencillamente, no se quiere hacer uso de las manos o de la voz para interactuar con el móvil o la tableta. Para estas situaciones, es interesante tener otra forma de interactuar con el dispositivo, como puede ser a través de la mirada. *Eye tracking* es una tecnología que permite realizar un seguimiento de los ojos, por ejemplo: conseguir que un cursor apunte hacia el punto al cual se está mirando en la pantalla, sin necesidad de coordinar la mirada con las manos, como si ocurre en el caso de un ratón de un ordenador (Hiley et al., 2006). Esta tecnología aún no ha sido muy explotada, debido, principalmente al alto coste de los dispositivos de *Eye Tracking* comerciales.

Al hilo de la definición de la tecnología *Eye Tracking* anterior, se puede definir *Gaze Tracking* como el uso de *Eye Tracking* para realizar seguimiento del ojo a lo largo del tiempo, con un objetivo (Zhu and Ji, 2004).

El uso de esta tecnología puede ser muy útil en diversas situaciones, como las que se detallan a continuación (Drewes, 2010):

- Uno de los entornos en los cuales se puede utilizar *Eye Tracking* es para hacer estudios de mercado y de publicidad y así analizar cuánto capta el ser humano en analizar logotipos, cuánto influye un color, etc.
- Otro entorno comercial, sería para estudios acerca de la usabilidad de páginas web, ya que dependiendo de su distribución, serán más atractivas para el ojo o no.
- Un campo muy interesante es la accesibilidad para personas con movilidad reducida, cuya única interacción solo puede ser visual.
- Para investigación de enfermedades en los ojos o problemas neuronales.

- En coches, ya que al ser difícil interactuar con elementos del coche con las manos, queda una puerta abierta al *Eye Tracking*, de hecho, ya los fabricantes de automóviles investigan en estas técnicas.

También, tiene una serie de ventajas, que están más detalladas en (Drewes, 2010):

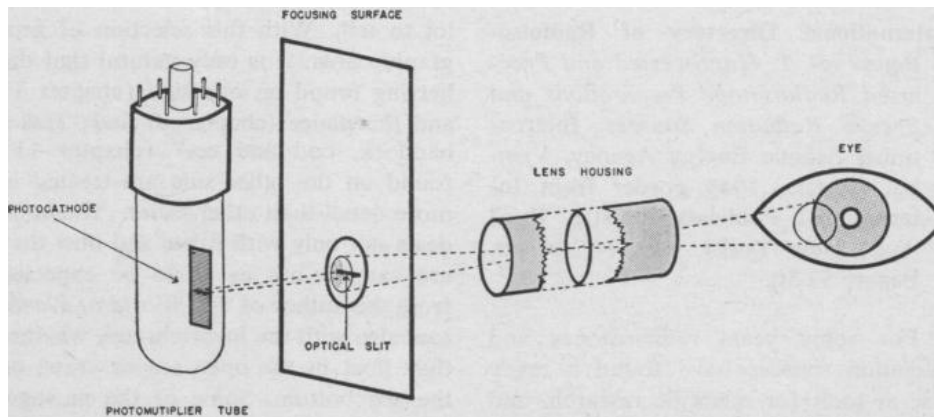
- Fácil de usar: no requiere usar las manos para interactuar.
- Interacción rápida, se interactúa a la velocidad de los ojos.
- No necesita mantenimiento de piezas mecánicas, ni limpieza del dispositivo, como ocurre con ratones y teclados.
- Se puede interactuar de forma remota, al no necesitar nada más que la mirada.
- Es higiénico, no ensucia nada al interactuar, por ser remoto.

Al igual que ventajas, tiene también inconvenientes, también detallados y ampliados en (Drewes, 2010):

- Requiere desarrollar una habilidad en el control de los ojos, que consiste en reducir al mínimo los movimientos que se realizan inconscientemente.
- Problema del rey Midas: al mirar a un punto, el sistema puede considerar que se desea interactuar con dicho punto y quizás solo se esté observando en ese momento, es difícil saber en qué caso se están utilizando los ojos para interactuar o para únicamente para mirar.
- Puede causar cansancio ocular debido al esfuerzo extra que ha de realizarse con la vista en este tipo de tecnología.

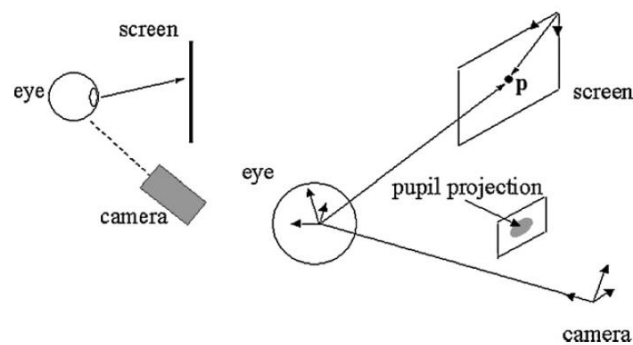
Existen dos tipos de movimientos oculares: sacádicos y suaves. Los movimientos oculares sacádicos se definen como aquellos en los cuales se cambia la fovea, que es el área de la retina en el cual se enfocan los rayos luminosos, rápidamente a un punto periférico. Esto causa a veces ausencia de visión fovea, que pasa, por ejemplo, cuando se está leyendo un texto. Dicho movimiento se caracteriza por su rapidez, y que no se puede controlar dicha velocidad conscientemente (Stark et al., 1962). Los movimientos oculares suaves, en cambio, consisten en mantener una imagen cerca de la fovea (Grossberg et al., 2012).

Las primeras técnicas para medir los movimientos oculares se remontan aproximadamente a finales de los años 50 y principios de los 60 como en (Smith and Warter, 1959) y en (Smith and Warter, 1960), en el cual se aborda dicha medida a través de técnicas fotoeléctricas. En (Smith and Warter, 1959), por ejemplo, se aborda mediante la amplificación por fotomultiplicación de la luz que pasa por una hendidura en la que se forma un campo de luz-oscuridad, como se muestra en la Figura 4.



**Figura 4.** Amplificación por fotomultiplicación (Smith and Warter, 1959)

En la actualidad, una interfaz hombre máquina genérica de *Eye Tracking* es un sistema formado por dispositivos físicos tales como cámaras, sensores, lentes, luces, etc. y algoritmos de análisis de señal, con el objetivo de determinar con suficiente precisión el punto de mira del sujeto (Santis and Iacoviello, 2009).



**Figura 5.** Esquema de un sistema genérico de *Eye Tracking* (Santis and Iacoviello, 2009)

Como puede observarse en la Figura 5, un sistema de *Eye Tracking* genérico consta de una cámara que apunta al ojo mientras este se encuentra mirando a la pantalla en cuestión. El sistema determina gracias a la proyección de la pupila en la cámara a qué punto de la pantalla se está mirando.

Hay dos tipos de sistemas dependiendo de la colocación de la cámara: remotos o tipo *head mounted*. En los remotos, tanto la cámara como las luces están situadas a distancia del usuario, normalmente cerca de la pantalla, mientras que en los de tipo *head mounted*, tanto la cámara como las luces están encima de la cabeza del usuario (Agustin, 2010), como puede ser en una visera.

Por último, sería conveniente destacar que ya existen *Smartphones* que vienen con un sistema para *Eye Tracking*, como puede ser el *Samsung Galaxy S3*, con su sistema *SmartStay*<sup>1</sup>, que evita el bloqueo del teléfono si detecta que el usuario está mirando al mismo en ese instante.

<sup>1</sup> <http://www.samsung.com/es/galaxy3/#smartstay>

## 2.4. *Gaze Gestures*

Un *Gaze Gesture* o gesto con la mirada, se puede definir como una secuencia de trazos, también llamados movimientos sacádicos, entre dos puntos. Diferentes patrones de estos trazos definen diferentes gestos (Rozado et al., 2012a).

Los gestos son una forma de comunicación muy extendida entre humanos, como se ha comentado anteriormente en la introducción, pero también entre animales, como pasa en el caso de los perros, que poseen habilidades para responder mediante gestos a los humanos (Téglás et al., 2012).

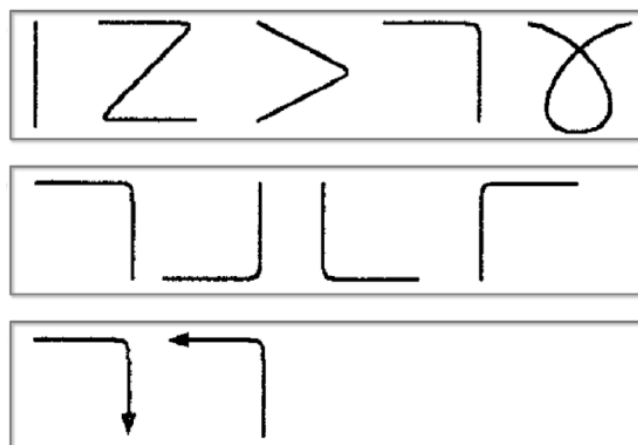
Los usos de los *Gaze Gestures* para interactuar con una máquina se pueden englobar en tres grupos genéricos:

- Los que los utilizan para elaborar un diálogo con una máquina.
- Los que los utilizan gestos de un solo trazo, ya que son más sencillos de aprender.
- Los que emplean *Gaze Gestures* como caracteres de entrada.

En el primer caso, cabe destacar el experimento *RealTourist* (Qvarfordt and Zhai, 2005), el cual se utiliza en un sistema experimental para gestión turística, llamado *iTurist*. Para interactuar con dicho sistema se usan *Gaze Gestures*.

En (Mollenbach, 2010), se estudian los resultados con los gestos de un solo trazo, como los que se observan en la Figura 6. Estos trazos pueden ser figuras distintas, el mismo trazo en distinta posición o también en distinto sentido.

También existen usos de los *Gaze Gestures* como caracteres de entrada, como se puede ver en (Wobbrock et al., 2007): en este caso, los gestos oculares son las propias letras del alfabeto, es decir, el ojo actuaría de lápiz a la hora de escribir.



**Figura 6.** Gaze Gestures de un trazo (Mollenbach, 2010)



Las acciones de reconocimiento que los humanos y algunos mamíferos somos capaces de realizar, por ejemplo, al reconocer un objeto en otra localización, con una iluminación diferente o una deformación no es tarea sencilla para una máquina (Dileep and Bobby, 2007).

El reconocimiento *Gaze Gestures* es un problema de reconocimiento de una estructura espacio temporal. Existen varios algoritmos de reconocimiento de secuencias destinados a este fin, aunque en este apartado no se abordarán todos.

En los siguientes subapartados se comentarán dos técnicas para el reconocimiento de secuencias, dependiendo del tipo de realización del gesto: si el gesto es de *gliding* o deslizamiento de la mirada a lo largo de un camino en la pantalla; o si se trata de secuencias predefinidas de movimientos sacádicos entre puntos. Para el reconocimiento de *Gaze Gestures* mediante técnicas de *gliding*, se suelen emplear redes de memorias temporales jerárquicas, también llamadas HTM, mientras para el caso de gestos con movimientos sacádicos, se utiliza el algoritmo de *Needleman-Wunsch*.

### 2.4.1. Redes de memorias temporales jerárquicas, HTM

El HTM (*Hierarchical Temporal Memory*) es un algoritmo bioinspirado en la neocorteza de los mamíferos capaz de llevar a cabo el reconocimiento de patrones, predicción de series o tareas de control, utilizando su estructura de nodos que realizan un agrupamiento espacio temporal y se organizan de forma jerárquica (Hawkins and Blakeslee, 2004).

Los HTM son conceptualmente una estructura de nodos tipo árbol (Rozado et al., 2011), con un modelo generador y una red bayesiana. Una red bayesiana es un modelo probabilístico que representa un conjunto de variables aleatorias y su dependencia condicional a través de un gráfico acíclico dirigido.

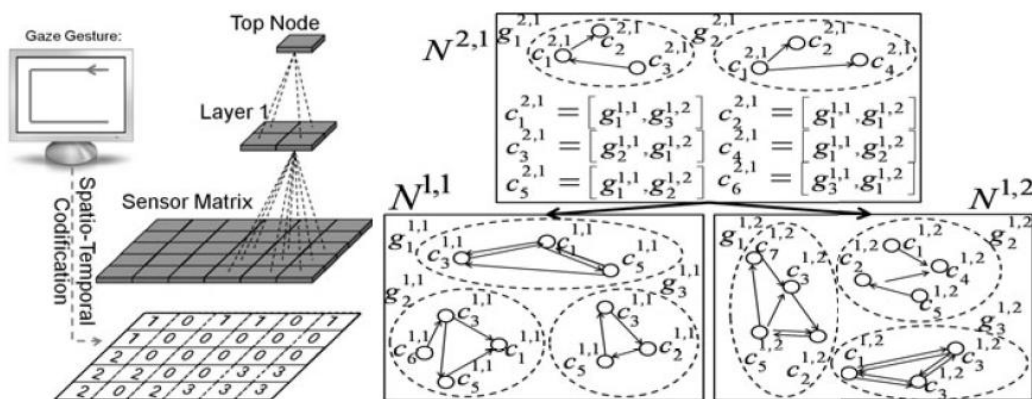


Figura 7. HTM: Modelo generador (izda.) y red bayesiana (dcha.). (Rozado et al., 2011)

En la Figura 7 puede verse como el modelo generador se basa fundamentalmente en ir encapsulando la información del gesto desde las capas más bajas del modelo hasta el nivel superior. Cada nodo de cada nivel tiene asociados unos patrones de coincidencia ( $c_1, c_2, \dots, c_n$ ) y unas cadenas de Markov ( $g_1, g_2, \dots, g_n$ ). Una cadena de Markov es un proceso estocástico discreto en el que la probabilidad de que ocurra un evento depende del evento inmediatamente anterior. Esto último se encuentra representado en la red bayesiana de la Figura 7.

Las cadenas de Markov representan las ocurrencias de las secuencias de los hijos del nodo en cuestión. Cada cadena de Markov es definida como un subconjunto del conjunto de patrones de coincidencia del nodo. Los patrones de coincidencia capturan la estructura de los nodos que están debajo en la jerarquía, que representa vectorialmente la co-activación de las cadenas de Markov en los nodos hijo (Rozado et al., 2011).

El principal problema de los HTM es la necesidad del uso de una fase de entrenamiento previo para poder construir el modelo (Rozado, 2011), debido a que hay que construir el modelo bayesiano de la estructura.

#### 2.4.2. Algoritmo de *Needleman-Wunsch*

El algoritmo de *Needleman-Wunsch* sirve para realizar alineamientos globales entre dos secuencias, es decir, medir el parecido entre la secuencia de puntos de los gestos oculares sacádicos realizados en la pantalla respecto a la secuencia de gestos a reconocer, a partir de la programación dinámica (Needleman and Wunsch, 1970). Este algoritmo, se ideó en un principio para ver las similitudes entre dos secuencias de aminoácidos, pero es extensible a la hora de comparar dos secuencias cualesquiera.

El funcionamiento de este algoritmo se representa en la Figura 8, donde la matriz representada a la izquierda de la figura, llamada a partir de ahora matriz D, representa el alineamiento entre las secuencias A y B, siendo A la secuencia del gesto realizada y B la secuencia del gesto que se ha de reconocer.

Cada punto  $D(i, j)$  de la matriz anterior se define mediante la Ecuación 1:

$$D(i, j) = \max \begin{cases} D(i-1, j-1) + s(x_i, y_i) \\ D(i-1, j) + g \\ D(i, j-1) + g \end{cases} \quad \text{Ecuación 1}$$

Dónde  $s(x, y)$ , es una constante que tiene valor distinto en caso de coincidencia o fallo entre las dos secuencias en ese punto y  $g$  es un valor de salto (*gap*), ambos valores son asignados por el programador (Rozado et al., 2012b).

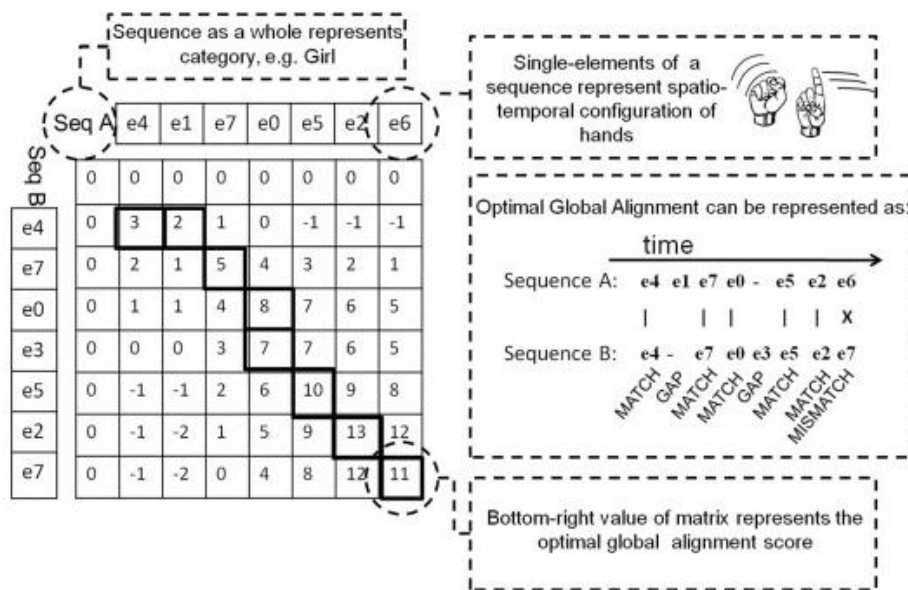


Figura 8. Algoritmo de Needleman-Wunsch

En el ejemplo de la Figura 8,  $s(x,y)$  vale en caso de coincidencia, 3 y en caso de no coincidencia, -2. El valor de  $g$  es en este caso es -1. Por ejemplo, en el cálculo de  $D(5,5)$ , al no haber coincidencia entre las secuencias ( $e_0 \neq e_3$ ),  $s(5,5) = -2$  y  $g = -1$ .  $D(4,4) = 4$ ,  $D(4,5) = 8$  y  $D(5,4) = 3$ , por lo que, según la Ecuación 1, su valor sería el máximo de la secuencia  $\{2,7,2\}$ , o sea, valdría 7.

Una vez se ha rellenado la matriz, comenzando en el punto  $D(1,1)$  hasta el punto  $D(n,n)$  siendo  $n$  la longitud de ambas secuencias, que ha de ser la misma, se parte de la esquina inferior derecha, es decir, el último valor calculado, y se buscan los máximos haciendo el camino inverso. El camino obtenido es la secuencia de máximo alineamiento. El valor de ese punto, da una puntuación a ese parecido entre secuencias, siendo mayor cuanto más similares son.

Este algoritmo ha sido elegido para ser implementado en el reconocimiento de los gestos oculares en este trabajo, por dos motivos: porque los gestos que se utilizarán para probar el prototipo son sacádicos y por la simplicidad y rapidez del algoritmo.



# 3. Diseño

---

## 3.1. Construcción del soporte

Para poder llevar a cabo el experimento, lo primero que hay que hacer es construir un dispositivo de *Eye Tracking* para un *Smartphone*, el cual tenga una característica fundamental: que sea de bajo coste, ya que como se ha comentado anteriormente, el coste de esta tecnología ha sido siempre la principal barrera para su desarrollo.

Para poder realizar *Eye Tracking* con el móvil, se necesitarán fundamentalmente, tanto fuentes de luz como una cámara, que han de estar adaptadas para ser usadas en dicho *Smartphone*, además de un soporte que sirva tanto para acoplar los elementos como para colocar el teléfono.

### 3.1.1. Luces

La mayoría de los sistemas de *Eye Tracking* usan iluminación de tipo infrarrojo, a partir de ahora, IR, aunque algunos sistemas pasivos utilizan la luz natural. La ventaja que tiene la luz IR es que proporciona a la escena una luz más estable. También influye el hecho de ser una luz invisible para el ojo humano, ya que este es sensible a longitudes de onda entre 400 nm y 700 nm y el espectro infrarrojo va de 700 nm a 100  $\mu$ m, lo cual hace también que no cause distracción. Además, ayuda a eliminar los reflejos producidos por otras fuentes de luz externas (Agustin, 2010).

Para nuestro dispositivo de bajo coste, sería necesario un foco pequeño, fácil de colocar y barato de luz infrarroja. Como no se ha encontrado nada que reúna estas características, se ha optado por fabricar uno a partir de un componente de luz infrarroja ya existente.

Se ha comprado para realizar las luces una lámpara infrarroja<sup>2</sup> que cuesta 11€ aproximadamente. Como esta lámpara consta de 48 LEDs, no se puede utilizar directamente, ya que con menos luz bastaría y la lámpara anterior es muy grande para un prototipo para *Smartphone*, ya que la radiación IR es calor que va directamente al ojo. Por esta razón, aprovechando tanto los LEDs como el cable de alimentación de 12 V con el que viene la lámpara original, se construirá una lámpara nueva infrarroja con solo 9 LEDs, es decir, la quinta parte aproximadamente de luz de la lámpara original.

El circuito se quiere construir en forma de matriz 3x3, ya que así la luz queda más centrada a un punto, como se aprecia en la Figura 9. Es necesario añadir resistencias para limitar la corriente en los LEDs. Suponiendo iguales los tres diodos, se asume que caerá la misma tensión en los mismos. Teniendo en cuenta la simetría que presenta el circuito y que la fuente de alimentación da 12V, las tres resistencias necesarias serán iguales.

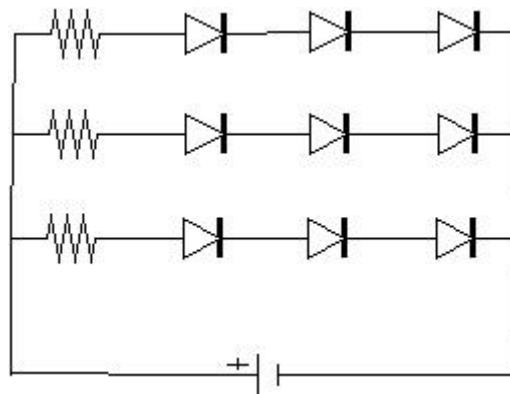
---

<sup>2</sup> <http://www.dealextreme.com/p/ir-48-led-infrared-illumination-light-for-night-vision-dc-12v-500ma-15235>

Utilizando la ley de Ohm ( $V = I \cdot R$ ), y sabiendo por las leyes de Kirchoff la corriente que pasa por cada resistencia (asumiendo simetría en el circuito de la Figura 9) es un tercio de la que sale de la fuente de alimentación, obtenemos el valor de cada una de las resistencias a partir de la Ecuación 2.

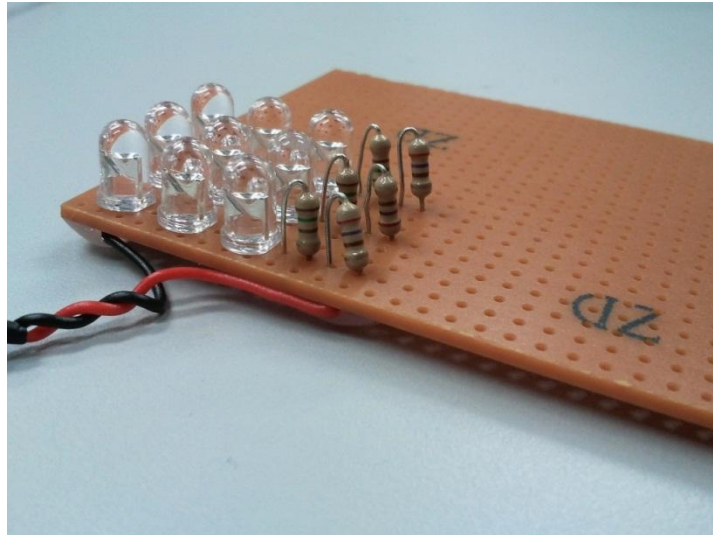
$$R(\Omega) = \frac{12 - 3V_D}{I/3} \quad \text{Ecuación 2}$$

En la Ecuación 2,  $I$  representa la corriente que sale de la fuente y  $V_D$  es la tensión que cae en cada uno de los diodos del circuito, que al ser iguales, ha de ser la misma. En la Figura 9 se muestra un esquema del circuito que se quiere construir.



**Figura 9.** Esquema del circuito de luces IR

Asumiendo que la tensión que soporta un diodo,  $V_D$  es 1.5 V y que la corriente que da la fuente son 55.5 mA, resolviendo la Ecuación 2 se obtiene un resultado para la resistencia de  $R(\Omega) = 405.41 \Omega \approx 405 \Omega$ . Al no tener resistencias de ese valor, se ha optado por la asociación en serie (suma de resistencias) de dos resistencias cuyo valor total se aproxime lo máximo posible al valor teórico. Entre las resistencias de las cuales se dispone, se eligen una de 150  $\Omega$  y otra de 270  $\Omega$ , lo que da un total de 420  $\Omega$ . O sea, nuestro circuito final tendrá un total de 6 resistencias físicas. El circuito final, ya implementado y con sus componentes soldados queda como se puede observar en la Figura 10.



**Figura 10.** Resultado final luces IR

### 3.1.2. Cámara

Al igual que la luz utilizada, la cámara será también sensible en el infrarrojo, en este caso se ha utilizado concretamente la *Sandberg NightCam 2* (ver Figura 11). Las especificaciones se pueden ver más detalladamente en la web del fabricante<sup>3</sup>. Entre sus características, cabe destacar la comunicación con el PC a través del puerto USB o que su espectro de captura también incluye el infrarrojo. A todo esto, hay que sumar también el hecho de su bajo coste, en torno a \$20 (aproximadamente 16€). Esta cámara viene con un juego de lentes.



**Figura 11.** Cámara *Sandberg NightCam 2*

---

<sup>3</sup> <http://www.sandberg.it/es-es/product/NightCam-2>

### 3.1.3. Montaje y estructura

Con todos los elementos básicos del soporte ya descritos, pasamos a construir el prototipo de *Eye Tracking* de bajo coste. Para ello, además de luces y cámara, se necesita un soporte para poder colocar el Smartphone. Dicho soporte será un soporte universal para el coche de Smartphone y GPS, que cuesta alrededor de 10€<sup>4</sup>.

En dicho soporte, se colocarán tanto la cámara como las luces, construyendo así aproximadamente, por unos 40€ un dispositivo de *Eye Tracking*. Hay que tener en cuenta a la hora de colocar los componentes en el soporte, que tanto las luces como la cámara han de ir colocados lo más cerca posible de la pantalla del teléfono para mejorar las prestaciones, pero en la zona inferior, ya que en la zona superior de la pantalla puede causar efecto de sombra en el ojo con el párpado superior al incidir la luz desde arriba. También hay que tener en cuenta que tanto la luz como la cámara han de incidir de forma paralela a la normal del plano en el que se encuentra en este caso el móvil y que entre ambas ha de haber entre 1.5 y 2 cm de separación.

En la Figura 12, se observa la implementación final del dispositivo de *Eye Tracking* de bajo coste que ha sido fabricado. Se observa que tanto las luces como la cámara se encuentran dispuestas como se ha comentado anteriormente. También se han tapado las luces infrarrojas de la cámara, que son menos potentes y pueden dar lugar a confusión en el software. Además, se ha añadido la tabla de gestos a realizar por los usuarios para que tengan a mano los ejercicios que han de hacer en las pruebas del prototipo.



Figura 12. Prototipo de *Gaze Tracking* propuesto sin teléfono (izda.) y con teléfono (dcha.)

<sup>4</sup> <http://www.amazon.co.uk/OXO-GPS-Flexible-Universal-Holder/dp/B0030T110M>



El teléfono móvil utilizado en las pruebas es el *Samsung Galaxy Mini* (Figura 12). Este dispositivo *Smartphone* tiene instalado de serie la versión *Android 2.2* y una resolución de pantalla de 290x240 píxeles.

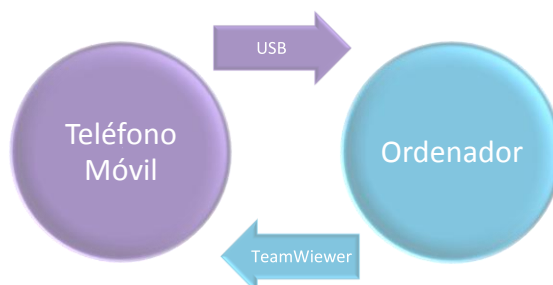
Cabe también enfatizar en que el equipo propuesto es opcional, se puede hacer uno de similares características con otras luces, otra cámara u otro soporte, siempre que cumplan los requisitos anteriormente citados. También cabe resaltar que la solución propuesta es de bajo coste: se asume en un principio que probablemente no es el mejor dispositivo para *Eye Tracking*, pero sí es una solución al alcance de cualquier bolsillo, proponiendo un prototipo de un coste estimado de 40€ frente a los miles de euros de los aparatos que se comercializan actualmente.

## 3.2. Comunicación entre el móvil y el ordenador

Al tratarse de un prototipo de prueba de funcionamiento de la tecnología de *Eye Tracking* aplicada a los *Gaze Gestures*, se ha optado por que toda la ejecución del proceso se lleve a cabo en el ordenador en lugar del móvil. El gesto se realizará en el móvil, pero las imágenes del ojo mirando al teléfono, que captura la cámara, serán enviadas al programa *GazeTracker* desde el puerto USB de la cámara hacia el ordenador, ya que la cámara presenta este tipo de conexión. Ya en el ordenador se realiza todo el procesado final.

La conexión del teléfono al ordenador se realiza mediante el software *TeamViewer*, de descarga gratuita en *Google Play* (antes *Android Market*), la tienda virtual de aplicaciones de *Android*, ya que el Smartphone que se utilizará será uno con dicho sistema operativo.

La aplicación *TeamViewer*<sup>5</sup> permite el acceso a un escritorio remoto si se posee un acceso a internet, así se tendrá desde el móvil con el que se realizarán las pruebas, acceso al escritorio de la máquina dónde se está ejecutando el software. El teléfono, de esta forma, tiene acceso al programa que se ejecuta en el PC para entre otras cosas, realizar la calibración del dispositivo. De esta forma se demuestra que es posible el control del teléfono, aunque el procesado se lleve a cabo en un ordenador. En la Figura 13 se muestra un sencillo esquema de la conexión existente entre el teléfono y la máquina.



**Figura 13.** Esquema que representa la interconexión entre ordenador y teléfono móvil

<sup>5</sup> <http://www.teamviewer.com>

En la Figura 14 puede verse la interfaz del programa *Teamviewer*. La imagen izquierda se corresponde con la interfaz de la aplicación para teléfonos *Android*, y la de la derecha, se corresponde con la interfaz de acceso al programa desde un ordenador.

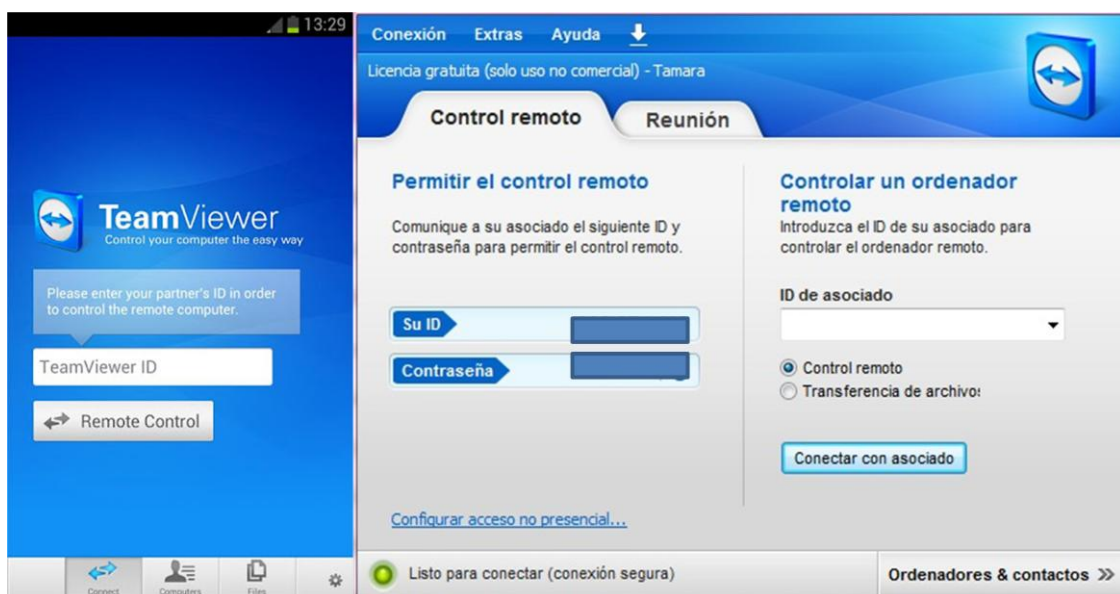


Figura 14. Interfaz de *TeamViewer*

### 3.3. Proceso de reconocimiento de gestos

El proceso de reconocimiento de gestos sigue el esquema que se puede ver en la Figura 15, y que será detallado paso a paso en este apartado.

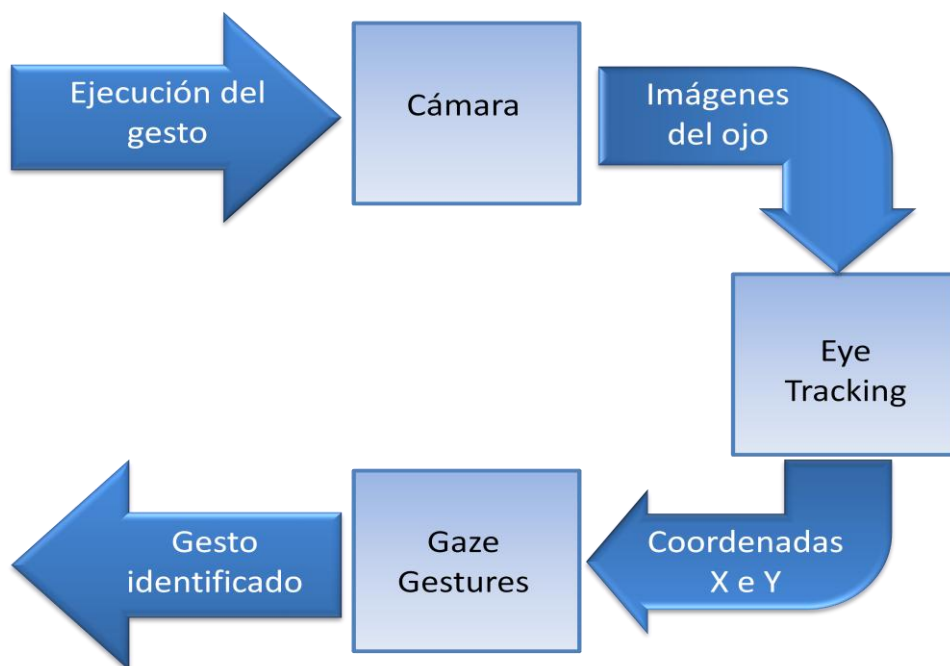


Figura 15. Pasos del reconocimiento de gestos

### 3.3.1. Captura de imágenes con la cámara

La cámara que se ha acoplado al dispositivo de *Eye Tracking* de bajo coste es la *Sandberg NightCam2*, que puede observarse en la Figura 11. Esta cámara será configurada desde el software *ITU Gaze Tracker*, del cual se hablará en el apartado siguiente, para que muestree a 30 imágenes por segundo con una resolución de la cámara de 320x240 píxeles. Se le ha colocado una lente de 16mm, que al ser la mayor disponible, al tener que realizar el experimento a una distancia aproximada de 30 cm, es la más adecuada para este fin.

### 3.3.2. *Eye Tracking*

En este apartado, se describirá el proceso de *Eye Tracking* partiendo de la descripción de la configuración del software que se utilizará, el *ITU Gaze Tracker*, así como de la forma en la que procesa la imagen para convertirla en coordenadas X e Y de la pantalla del Smartphone.

#### 3.3.2.1. *Gaze Tracker*: interfaz

El software *ITU Gaze Tracker* ha sido desarrollado por el *ITU Gaze Group*, de la universidad de Copenhague, y se puede obtener mediante descarga de la web del mismo<sup>6</sup>. Este programa se ejecuta en un sistema operativo Windows®, de hecho, ha sido desarrollado en C# y la librería se compila con el compilador Visual Studio (Agustin, 2010), de Microsoft®.



Figura 16. Interfaz del programa *ITU Gaze Tracker*

La interfaz de este programa (Figura 16), consta de una pantalla en la cual muestra lo que está viendo la cámara en este momento y tres botones: *setup*, *calibrate* y *start*. El botón de *setup* accede al menú de configuración del programa (véase más detalladamente en la Figura 17), dónde se observan las siguientes opciones:

<sup>6</sup> <http://www.gazegroup.org>

- Pestaña *Tracking*. Tipo de montaje: puede ser *head mounted*, remoto monocular (solo se utiliza un ojo) y remoto binocular (se utilizan los dos ojos). También se puede poner ajuste de los parámetros del programa en automático o ajustarlos a mano, además de poder decirle al programa dónde están localizadas las fuentes de luz IR.
- Pestaña *Calibration*. Se puede ajustar el número de puntos de la calibración (9 en nuestro caso), pero pueden ser 12, ó 16, así como el área de calibración y su color.
- Pestaña *Camera*. Aquí además de poder ajustar la resolución de la cámara o de invertir la imagen, se pueden ajustar los parámetros tales como el brillo, el contraste, etc.
- Tamaño de la pantalla de calibración: por defecto, se calibra ajustado al tamaño del monitor que se esté utilizando, pero se puede utilizar un área de calibración menor que dicho monitor.

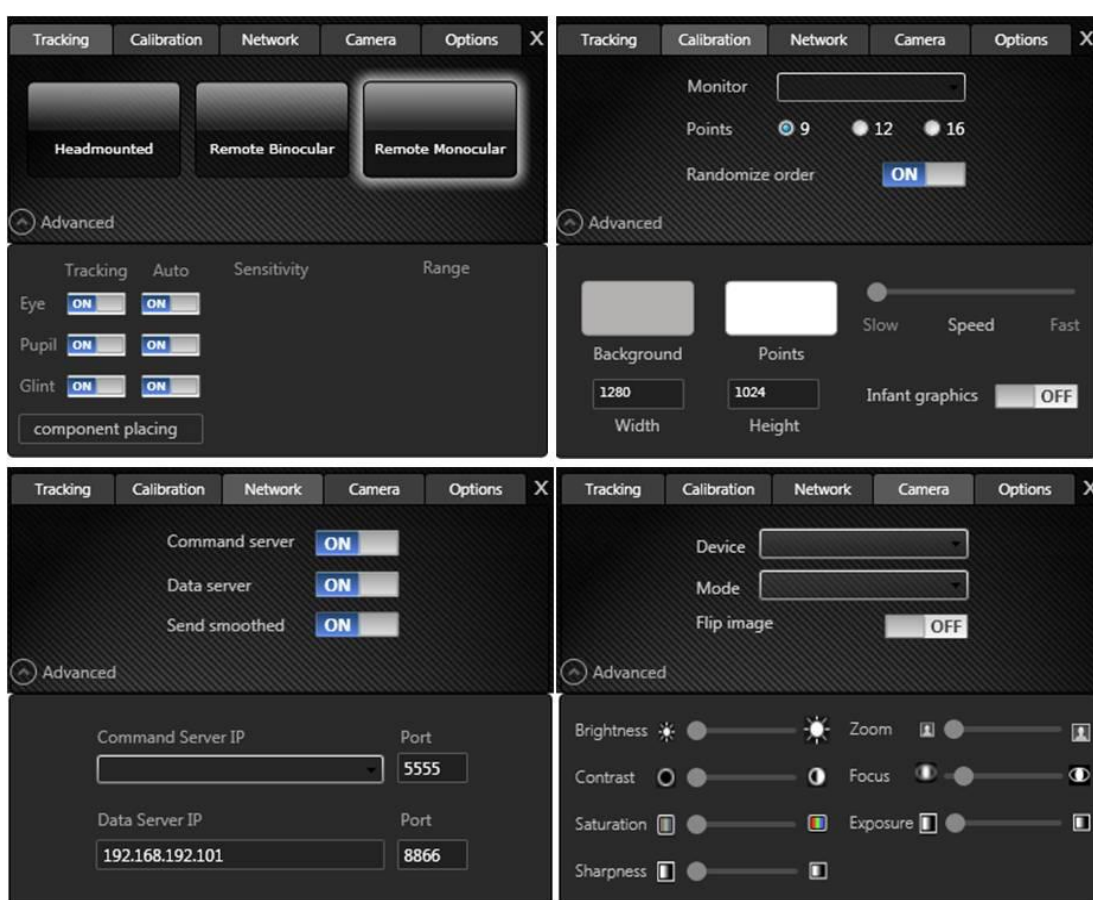
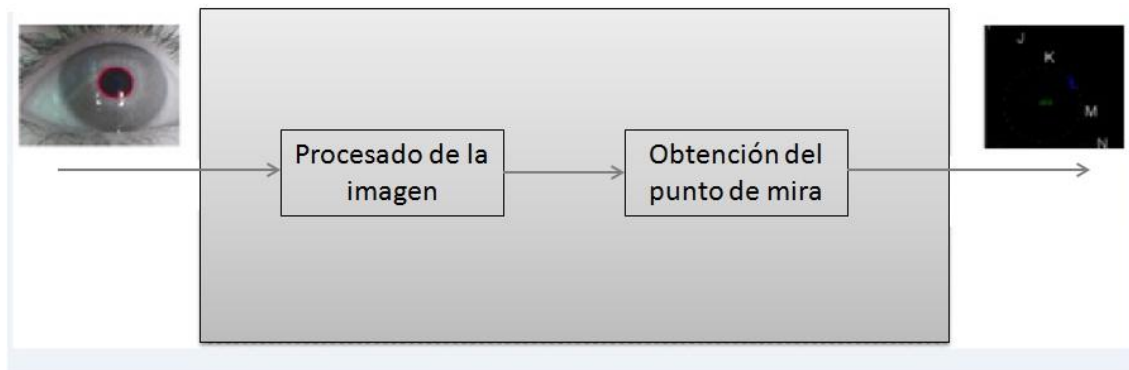


Figura 17. Detalle de los submenús del menú de configuración del *ITU GazeTracker*

Otra vez en la Figura 16, el botón *calibrate* sirve para calibrar el dispositivo, y el *start* para que empiece a funcionar después de realizar dicha calibración. El programa procesa la información de la forma que aparece en la Figura 18. Las etapas de este proceso se detallarán más adelante.

Con este programa, lo que se hará, en resumidas cuentas, es convertir el movimiento del ojo en una serie de coordenadas X e Y que se definen como el punto al cual se está mirando en el área de calibración utilizado.

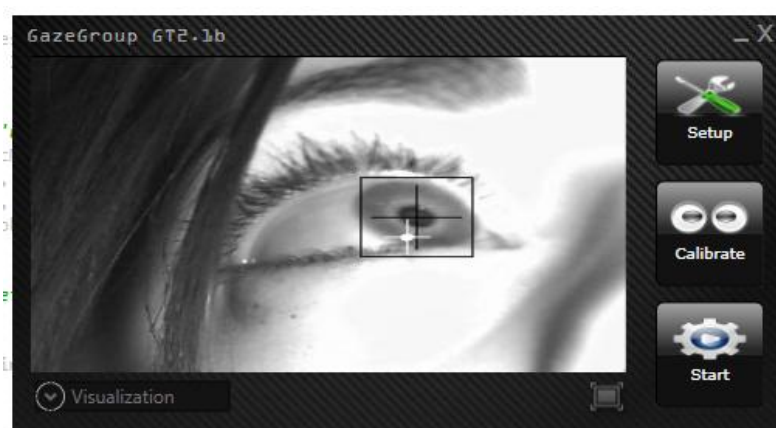


**Figura 18.** Pasos que sigue el programa *ITU Gaze Tracking*

### 3.3.2.2. *Gaze Tracker*: procesado de la imagen

Lo primero que se realiza para obtener las características del ojo es el procesado de la imagen que se recibe a través de la cámara. El software permite el rastreo (*tracking*) tanto de la pupila como de uno o dos reflejos de la córnea (*glints*) (San Agustín et al., 2010), que dependen de si el seguimiento es para un ojo (monocular), como en la Figura 19, o para los dos (binocular).

En la Figura 19, también se puede observar que el programa *ITU Gaze Tracker* ha identificado la pupila (cruz negra) y el reflejo de la córnea o glint (cruz blanca).



**Figura 19.** Ejemplo de uso del programa *ITU Gaze Tracker* para un solo ojo

El procesamiento de imagen se realiza con *OpenCV (Open Source Computer Vision)*, una librería de código abierto de la visión en tiempo real, desarrollada por Intel®, que tiene más de 2500 algoritmos optimizados.

Para estimar la mirada, lo primero que hay que tener en cuenta es que al ser un sistema basado en video (imágenes en movimiento capturadas por la cámara), el programa ha de pasar de un espacio de  $m$  coordenadas como son las características del ojo, a 3 coordenadas: eje X, eje Y, y el tiempo (Agustin, 2010).

$$\Phi : \mathbb{R}^m \rightarrow \mathbb{R}^3$$

**Ecuación 3**

La función  $\Phi$  que se representa en la Ecuación 3 representa ese paso, de forma genérica, de  $m$  a 3 coordenadas. Para hallar la función  $\Phi$  existen dos tipos de técnicas:

- Basadas en la interpolación. Se asume que la función  $\Phi$  es genérica y que mapea las características del ojo a partir de imágenes de la mirada. Esta es la técnica que usa el programa *ITU Gaze Tracker*, y por tanto será la que se aborde posteriormente.
- Basadas en la geometría. Usan un modelo del ojo para hallar el punto de mira y en algunos casos la dirección de la mirada (calculada como la intersección del vector de la mirada y el plano de la pantalla).

Dicha función  $\Phi$  es la que se encarga de estimar la mirada. Dependiendo de la naturaleza de los parámetros de la misma, existen cuatro tipos de calibración: la de la cámara (la lente), de la de la geometría (dependiendo de la localización y la orientación), la de los parámetros propiamente humanos, como la curvatura de la córnea y los parámetros del modelo, como los coeficientes de regresión de los métodos de interpolación. Los dos últimos tipos se estiman calibrando mirando a distintos puntos de la pantalla.

Esta calibración requiere un patrón de calibración predefinido. Si cambian algunos de los parámetros iniciales, es necesario repetir la calibración (Agustin, 2010).

#### 3.3.2.2.1. Métodos de interpolación

Este método requiere un sistema de calibración por puntos, generalmente 3x3 ó 4x4 puntos en la pantalla, y se asume que con estos puntos calibrados se puede simular el comportamiento del ojo en el resto de la pantalla. Cuanto mayor sea el número de puntos en la pantalla, mejor es la calibración del dispositivo. Es un método muy utilizado cuando se dispone poca o ninguna información acerca de la geometría del ojo.

La función genérica  $\Phi$  que modela el comportamiento del ojo, para este caso, puede simplificarse mediante la Ecuación 4.

$$\Phi = A \cdot E \quad \text{Ecuación 4}$$

Donde A son los coeficientes de la ecuación lineal y E las características del ojo usadas en regresión, que son prefijadas. Durante la calibración, se hallan las características del ojo para cada punto de cada imagen del mismo, con lo que se calcula el parámetro A.

El tipo de función más implementado para estimar  $\Phi$  es el polinomio lineal. Cuanto mayor es el grado del polinomio, el modelo se hace más preciso y también más complejo, por lo que normalmente se elige uno de segundo orden, como en este caso. En (Morimoto et al., 1999), por ejemplo, emplean un par de polinomios de segundo grado para mapear el vector del reflejo de la córnea,  $(x_{pc}, y_{pc})$  respecto al vector de coordenadas en la pantalla  $(x_s, y_s)$ , como se aprecia en la Ecuación 5.

$$\begin{aligned} x_s &= a_0 + a_1x_{pc} + a_2y_{pc} + a_3x_{pc}y_{pc} + a_4x_{pc}^2 + a_5y_{pc}^2 \\ y_s &= a_6 + a_7x_{pc} + a_8y_{pc} + a_9x_{pc}y_{pc} + a_{10}x_{pc}^2 + a_{11}y_{pc}^2 \end{aligned} \quad \text{Ecuación 5}$$

### 3.3.2.3. Obtención del punto de mira en la pantalla

La información acerca del punto de mira obtenida en el apartado anterior, necesita de un postprocesado para agrupar los puntos de mira obtenidos anteriormente en áreas de interés para ser utilizados en el siguiente paso. Para este fin, se utilizan algoritmos de *smoothing* (suavizado), que eliminan ese ruido producido por el *jitter*, o retardo, que se produce cuando el usuario intenta mirar a un punto. El ruido puede provenir de la imprecisión a la hora de obtener las características del ojo, o de los pequeños movimientos que puede realizar el ojo durante una fijación, por lo que el punto en una fijación puede no ser constante, siendo más significativo en los sistemas de bajo coste, como el que se presenta.

Por todo esto, es necesario un algoritmo de detección de movimientos que elimine estos inconvenientes, los algoritmos existentes son de este tipo:

- **Basados en velocidad.** Utilizan la velocidad del ojo para detectar cuando hay una fijación, basándose en que el ojo tiene una velocidad menor en una fijación que en un movimiento sacádico.
- **Basados en dispersión.** Estos algoritmos utilizan la posición del cursor para hallar la fijación, y se basan en que la dispersión es baja cuando hay fijación, para lo cual existe un umbral de dispersión, y cuando se baja de ese umbral, hay fijación.
- **Basados en área.** Estos algoritmos miran solo si hay interacción en determinadas áreas de interés, lo que hace que si detecta que está allí la fijación, la sitúa directamente en el centro de esta área.
- **Basados en velocidad + dispersión.** Este método tiene dos etapas: una de detección que combina velocidad y dispersión, y otra de *smoothing*, que calcula el promedio de las N fijaciones anteriores. Esta técnica es la que utiliza el *ITU Gaze Tracker*, y será detallada a continuación.

### 3.3.2.3.1. Algoritmos basados en velocidad y dispersión

Estos algoritmos, como se ha detallado anteriormente, constan de dos etapas básicas:

#### 1. Detección de fijaciones

Partiendo de la baja velocidad y la baja dispersión existente en una fijación, como se ha comentado anteriormente. Esta etapa tiene a su vez dos subetapas:

- **Etapada basada en velocidad**

Para cada punto de fijación  $i$ , con coordenadas  $(x_i, y_i)$ , se calcula su velocidad respecto al punto anterior, o sea, el punto  $i-1$ ,  $(x_{i-1}, y_{i-1})$ . La distancia entre ambos puntos se calcula como  $d = \|(x_i, y_i), (x_{i-1}, y_{i-1})\|$ , se calcula el ángulo  $\theta_i$ :

$$\theta_i = 2 \tan^{-1} \frac{d/2}{D} \quad \text{Ecuación 6}$$

Dónde  $D$  es la distancia del ojo a la pantalla, y  $d$  es la distancia entre el punto de fijación actual y el anterior. Con este ángulo, se puede calcular la velocidad angular de esa fijación como  $\dot{\theta}_i = \theta_i / \Delta t$ . Para poder saber si el valor  $\dot{\theta}_i$  hallado proviene de una fijación o es un movimiento sacádico, hay que definir un umbral,  $\dot{\theta}_{max}$ , de forma que si  $\dot{\theta}_i$  es menor que dicho umbral, es una fijación, y si es mayor, es un movimiento sacádico.

- **Etapada basada en la dispersión**

Para una secuencia de  $N$  puntos de fijación, se calcula la dispersión. Si dicha dispersión está por debajo de un umbral fijado, se detecta una fijación. Este algoritmo necesita dos parámetros básicos: el tiempo mínimo de duración de una fijación,  $t_{min}$ , y el umbral de dispersión,  $D_{max}$ .

Los valores de este  $t_{min}$  típicos están entre 100 y 200ms, por lo que una vez transcurrido este tiempo, se calcula la dispersión en la ventana anterior de  $N$  puntos y se compara con el umbral  $D_{max}$ . La máxima dispersión se considera que es 1º del ángulo de visión, pero este umbral puede verse afectado por los efectos del ruido. Para este algoritmo, se utiliza un  $t_{min}$  de 120ms, y se compara con el valor de  $D_{max}$  correspondiente a 1º de ángulo de visión.

Utilizando el algoritmo de dispersión se corrige un defecto que tiene el algoritmo de velocidad: presenta problemas cuando el valor  $\dot{\theta}_i$  correspondiente a un punto se acerca al valor máximo  $\dot{\theta}_{max}$ . Añadiendo el algoritmo de dispersión, se solventa este problema al rechazar fijaciones muy cortas que haya podido admitir el algoritmo de velocidad.

Para todo esto, hay que tener en cuenta que se necesitan dos parámetros prefijados: las dimensiones de la pantalla y la distancia del usuario a dicha pantalla. Los primeros se pueden obtener de las características gráficas del dispositivo, la distancia a la pantalla, en cambio, se asume de 60cm.



## 2. Etapa de suavizado o de *smoothing*

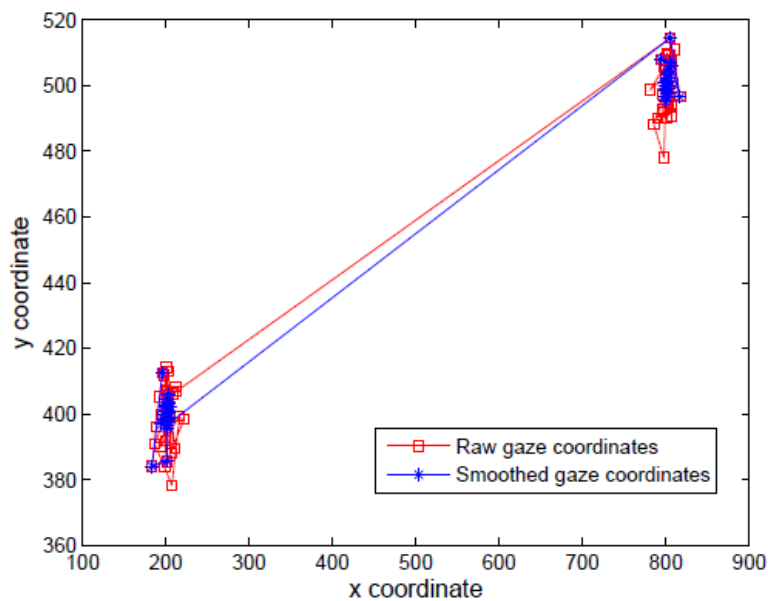
Es necesario suavizar el movimiento del cursor para eliminar el *jitter* y el ruido producido por pequeños temblores del ojo. Cada punto de fijación después del suavizado  $(x_s, y_s)$ , se calcula como:

$$x_s = \frac{1}{N} \sum_{i=0}^{N-1} x_i \quad \text{Ecuación 7}$$

$$y_s = \frac{1}{N} \sum_{i=0}^{N-1} y_i \quad \text{Ecuación 8}$$

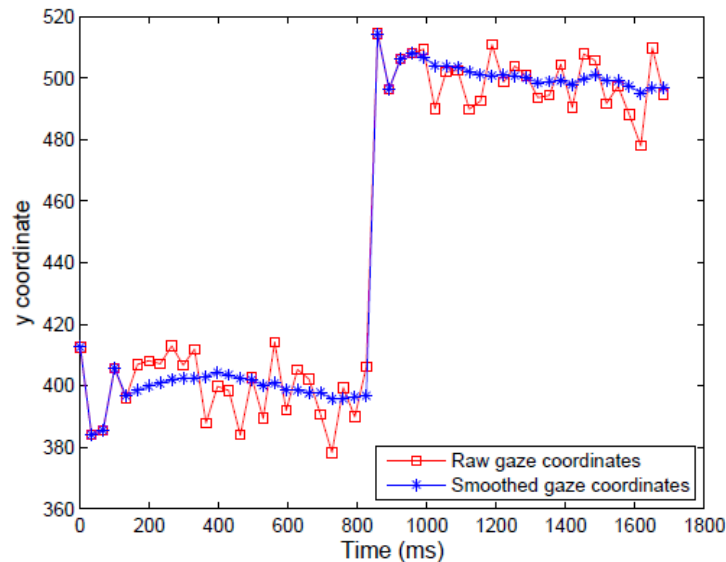
Como puede verse en la Ecuación 7 y la Ecuación 8, cada punto se calcula como un promedio de los N puntos anteriores (ventana de N puntos). En el caso de este software, se utiliza una ventana de 15 puntos para la etapa del suavizado (Agustin, 2010).

En la Figura 20, se representan en el eje de abscisas la coordenada x del punto y en el eje de ordenadas la coordenada y del mismo. Pueden verse en rojo los puntos antes del suavizado, y en azul después del mismo. Los puntos en rojo están más dispersos que los que están en azul, que tienden a concentrarse en un punto, debido al efecto de la etapa de suavizado. También pueden verse dos zonas en las cuales hay puntos, o sea, tenemos dos puntos de fijación claramente diferenciados.



**Figura 20.** Efecto de la etapa de suavizado. Adaptado de (Agustin, 2010)

En la Figura 21, se puede observar el efecto del suavizado en una de las coordenadas, concretamente la coordenada Y, respecto al tiempo. Existe un cambio de posición de fijación en torno a un tiempo de 800 ms. También puede verse como gracias al suavizado, se promedia el valor de la coordenada en dicho punto de fijación, haciendo el punto de mira estable.



**Figura 21.** Efecto del suavizado en la coordenada y respecto al tiempo. Adaptado de (Agustin, 2010)

### 3.3.3. Gaze Gestures

Una vez hayan sido obtenidas las coordenadas del punto de mira del ojo y de haber diferenciado entre lo que es y lo que no es fijación, se procederá a reconocer el gesto que el ojo trata de hacer, o sea, el reconocimiento de gestos.

Se define *Gaze Gestures* como gestos que se pueden realizar con la mirada, a partir de una sucesión de trazos (*strokes*) entre distintos puntos de la pantalla, o sea, puede definirse como una combinación de puntos hechos con la mirada en un determinado momento.

#### 3.3.3.1. Algoritmo de reconocimiento de los *Gaze Gestures*

Este algoritmo, convierte las anteriores coordenadas en la pantalla del dispositivo a lo largo del tiempo, en un gesto reconocido.

Tiene varios parámetros ajustables, como pueden ser las imágenes por segundo, que se fija por defecto en torno a 30, o las dimensiones de la pantalla del dispositivo en el que se ejecuta, que las toma del API de Windows®.

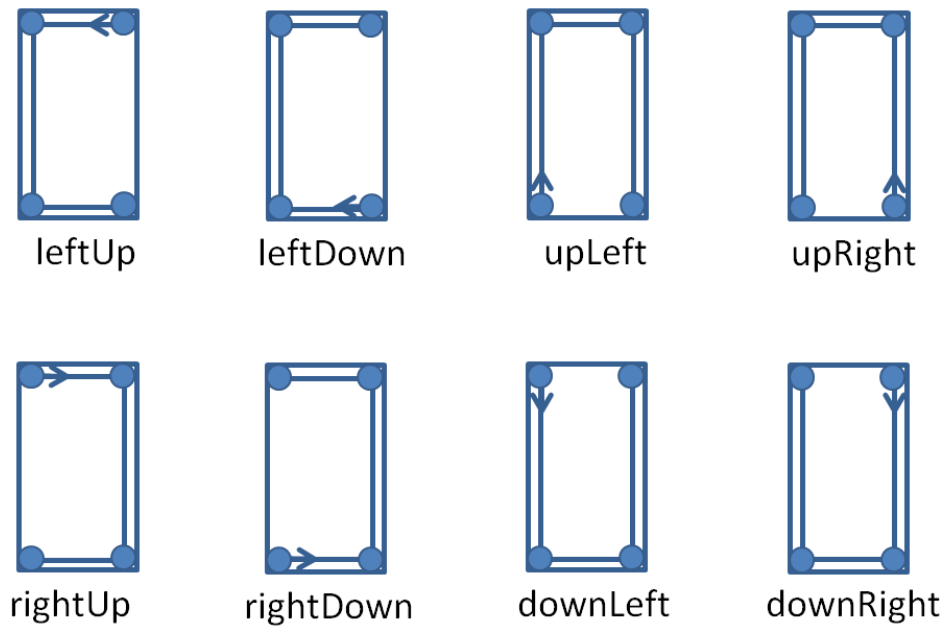
El algoritmo, lo primero que hace, es dividir la pantalla en N áreas de igual tamaño, para agrupar todos los puntos en función de la zona en la que estén. En este caso, serán 9 las áreas en las que se dividirá la pantalla del teléfono móvil, como puede verse en la Figura 22.



**Figura 22.** Áreas de la pantalla del móvil

Con lo cual, ahora los puntos se clasifican en 9 tipos (A,B,C,D,E,F,G,H,I), y un gesto se podría considerar como una combinación de estas letras en el orden definido. La detección de la secuencia se basa en el algoritmo de *Needleman-Wunsch*, visto en el capítulo 2.4.2, con valores de 3 para coincidencia, -2 para no coincidencia y 1 para g.

El algoritmo puede almacenar cualquier gesto que sea combinación de trazos entre los puntos correspondientes a cualquiera de las zonas, pero los elegidos son ocho gestos de tres trazos, concretamente gestos usados para el control del Smartphone son los descritos en la Figura 23. Al ser todos los gestos utilizados de tres trazos, y ser los gestos 2º, 4º, 6º y 8º el mismo que el anterior, pero hecho al revés, ayuda a que sean fácilmente recordados por el usuario.

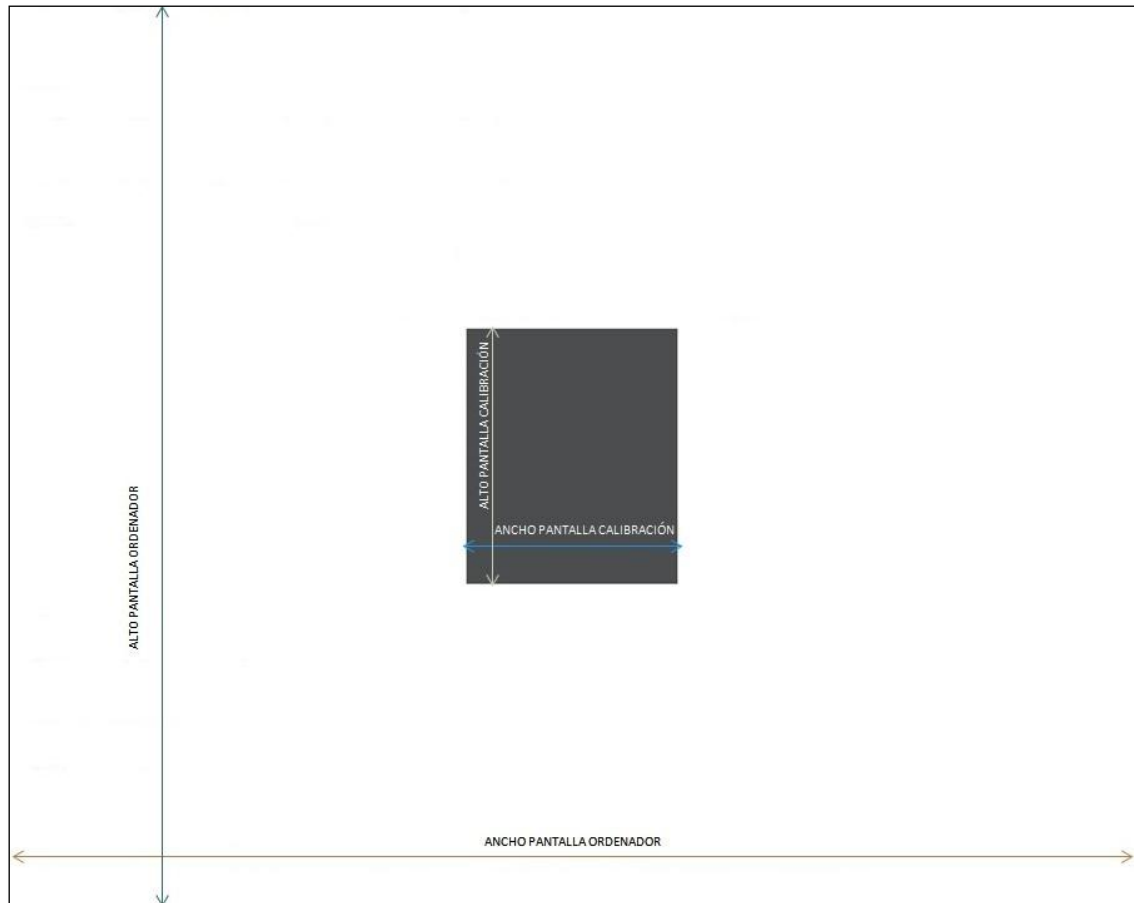


**Figura 23.** Lista de los *Gaze Gestures* definidos en el algoritmo

### 3.3.3.2. Adaptación del reconocimiento de *Gaze Gestures* al *Smartphone*

Toda la ejecución, tanto del programa de *Eye Tracking*, como el algoritmo de los *Gaze Gestures*, se lleva a cabo en un ordenador personal, y, aunque en el teléfono se estén realizando los gestos, la pantalla no tiene las mismas dimensiones que la del ordenador, por lo que es necesaria una adaptación previa tanto de la pantalla de calibración de *ITU Gaze Tracker*, como de las constantes que utiliza el algoritmo de los *Gaze Gestures*.

En el caso del *ITU Gaze Tracker*, en el menú de configuración, en la pestaña *Calibration*, existe la opción de cambiar el área de la pantalla (ver más detallado en 3.3.2.1), que se ajustará manualmente hasta que coincida con la del móvil, en este caso 240 píxeles de ancho, por 290 de alto, ya que el Smartphone que se ha utilizado para las pruebas es un *Samsung Galaxy Mini*.



**Figura 24.** Pantalla de calibración de *ITU Gaze Tracker* en el ordenador

En la Figura 24 puede apreciarse la pantalla del ordenador en el cual se está ejecutando tanto el programa de *Eye Tracking*, como el algoritmo de reconocimiento de *Gaze Gestures*. El algoritmo toma como origen de coordenadas por defecto el punto (0,0), que se corresponde con la esquina superior izquierda de la pantalla.

Como el algoritmo de *Gaze Gestures* trabaja por defecto con las coordenadas de la pantalla, al disminuir el área de calibración del *Eye Tracker*, hay que corregir este efecto en el algoritmo, o sea, ahora el nuevo punto de origen será la esquina superior izquierda, pero del área de calibración, lo que implica que a todos los puntos obtenidos del *Eye Tracker* hay que restar a las coordenadas X e Y un valor de offset, de valor la mitad de la diferencia entre el ancho de la pantalla del ordenador y el de la pantalla de calibración para el valor de X, y de la mitad de la diferencia entre el alto de la pantalla del ordenador y de la pantalla de calibración para Y.



# 4. Resultados

---

## 4.1. Inicialización del sistema

El sistema de bajo coste para el control de un teléfono, propuesto anteriormente en el apartado de diseño, ha de ser probado para demostrar su funcionamiento. Para ello, lo primero que se hará será configurar los parámetros iniciales que se requieran, tanto en el *Smartphone*, como en el programa de *Eye Tracking*, como, finalmente, ajustar los parámetros requeridos en los algoritmos de *Gaze Gestures*.

### 4.1.1. Configuración del teléfono

Lo primero que hay que tener en cuenta es que el teléfono con el cual se realizarán las pruebas ha de tener conexión a internet, ya sea Wi-Fi, o de datos vía 3G, para comunicarse con el ordenador. Ya una vez conectado a internet, hay que disponer de la aplicación *TeamViewer*, disponible en *Google Play* (plataforma de descarga de aplicaciones de *Android*), que es gratuita. También hay que tener en cuenta que hay que tener instalado en el ordenador sobre el que se ejecutará el programa de *Eye Tracking* y los algoritmos también la versión para PC de *TeamViewer*, para poder conectar teléfono y máquina.

Como puede verse en la Figura 14, la imagen de la izquierda se corresponde con la interfaz de conexión de *TeamViewer* a través de un *Smartphone*, y la de la derecha, con la del ordenador. Con introducir el ID y la contraseña (omitidos por seguridad) del ordenador en cuestión, se puede tener un acceso remoto desde el *Smartphone* (u otro dispositivo) al mismo.

### 4.1.2. Configuración del *Eye Tracker*

En el apartado 3.3.2.1 se puede observar al detalle todo el menú de configuración del programa *Gaze Tracker*, al que habrá que ajustarle los siguientes valores:

- En la pestaña *Tracking*, el tracking será de tipo monocular, y todos los parámetros estarán en modo auto.
- En la pestaña *Calibration*, se pondrá una calibración de nueve puntos en modo aleatorio, para evitar un aprendizaje de la secuencia de calibración. El área de calibración se ajustará a los valores de ancho y alto del área del teléfono.
- En la pestaña *Camera* se invertirá la imagen si fuera necesario, así como los parámetros de brillo, contraste, saturación,...

### 4.1.3. Configuración del algoritmo de reconocimiento de *Gaze Gestures*

El algoritmo de reconocimiento de los *Gaze Gestures*, como se comentó en el apartado 3.3.3, tiene una serie de parámetros ajustables, como pueden ser:

- El tamaño del área de calibración, que además de en el programa de *Eye Tracking*, hay que ajustarlo en el algoritmo.
- El umbral de puntuación del algoritmo de *Needleman-Wunsch* (ver apartado 2.4.2) a partir del cual la puntuación de una secuencia se considera que es un gesto.
- El tipo de realización del gesto.
- El tipo de experimento a realizar.
- El número de gestos a reconocer en el experimento y los gestos que lo forman.
- Definición de los gestos existentes.

## 4.2. Validación del control con *Gaze Gestures*

Para mostrar el funcionamiento del sistema de *Gaze Tracking* de bajo coste propuesto anteriormente (ver Figura 12) se realizarán una serie de experimentos con 20 sujetos, de edades comprendidas entre los 20 y los 50 años, de ambos sexos. Las pruebas que serán realizadas a los mismos, se describirán detalladamente en el siguiente apartado. Con estas pruebas se medirá la precisión, falsos positivos, rapidez del mismo y efecto del aprendizaje en su uso.

## 4.3. Descripción de las pruebas

Las pruebas que se realizarán a los usuarios de cara a probar el sistema con el uso de *Gaze Gestures*, pueden clasificarse en cuatro grandes grupos:

- Dependiendo de la forma de la realización del gesto.
- Por el tipo de uso del dispositivo, si es un gesto voluntario, medir su precisión, si es involuntario, ver si da muchos falsos positivos.
- Pruebas de rapidez: medir el tiempo de realización de un gesto.
- Pruebas para medir la influencia del aprendizaje en el uso del dispositivo.





**Figura 25.** Ejemplo de realización de un gesto en la pantalla de un móvil

### 4.3.1. Por tipo de la realización del gesto

El algoritmo de los *Gaze Gestures* admite dos formas diferentes para realizar el gesto: sin *dwell time* o con *dwell time*. *Dwell time*, o tiempo de fijación es una forma de marcar el comienzo y el final de un gesto marcando ambos puntos con una fijación más larga de la mirada.

Las pruebas sin *dwell time* consisten en la realización del gesto mirando a los puntos correspondientes, pero sin marcar ni el comienzo ni el fin. Por el contrario, los pasos para realizar el gesto en el caso de las pruebas con *dwell time* serán, primero, fijar la mirada en el punto inicial, cuando se escuche un pitido, quiere decir que se puede dar comienzo al gesto, que se realizará como en el caso de sin *dwell time*, y en el último punto, se volverá a fijar la mirada hasta escuchar otro pitido. Por ejemplo, en el gesto de la Figura 25, para realizarlo con *dwell time* habría que fijar la mirada en el área C, esperar el pitido, hacer el gesto y cuando se llegue al área I, fijar la mirada hasta que devuelva un segundo pitido.

### 4.3.2. Por tipo de uso

Otra forma de clasificar las pruebas que se realizarán a usuarios es midiendo cuánto es el sistema capaz de reconocer un gesto cuando es voluntario o si simplemente reconoce gestos cuando no son intencionados.

Las pruebas de *accuracy*, o de precisión consisten en la realización de una serie de gestos, en este caso la serie de ocho gestos que aparecen en la Figura 23, y medir el porcentaje de gestos que se han acertado. El algoritmo solo pasa a reconocer el gesto siguiente si ha habido algún gesto reconocido en el anterior, ya sea un acierto o un fallo, es decir, si al realizar el gesto el sistema no reconoce nada, se ha de repetir hasta que se reconozca uno, de acierto o fallo.

Las pruebas de *browsing* o de navegación, consisten en simular el uso normal del teléfono, como puede ser leer mensajes de texto, jugar a un juego, leer el correo, leer noticias, ver fotos,... Un sencillo pase de diapositivas de aplicaciones y fotos será observado por el usuario mientras el algoritmo funciona, detectando falsos positivos en el caso que se realizara un gesto, que evidentemente, sería involuntario.

### 4.3.3. Medida de la rapidez de un gesto

Con estas pruebas se pretende medir el tiempo que se tarda en hacer un determinado gesto, con lo cual se halla en cierta forma una medida de cuanto de molesto le es a un usuario hacer un gesto tanto sin como con *dwell time*. El gesto elegido para ambos casos es el denominado '*LeftUp*' de la Figura 23. Dicho gesto será cronometrado con una décima de precisión con un cronómetro manual.

### 4.3.4. Pruebas de aprendizaje

Estas pruebas consisten en la realización repetidas veces de las pruebas de *accuracy* y de rapidez de un gesto. En este caso, se ha decidido que el número de veces que se repitan sean cinco, en momentos diferentes, para finalmente ver qué efecto tiene el aprendizaje a la hora de usar *Gaze Gestures*, es decir, si se mejoran los resultados obtenidos si no es la primera vez que se usa el sistema.

## 4.4. Resultados obtenidos

A cada uno de los veinte usuarios se les hicieron los siguientes experimentos para probar el funcionamiento de los *Gaze Gestures* en un Smartphone en el prototipo de bajo coste diseñado anteriormente:

- Pruebas de *accuracy* tanto para la realización de gestos sin y con *dwell time*.
- Pruebas de *browsing* tanto para la realización de gestos sin y con *dwell time*.
- Tiempo de realización del gesto "*LeftUp*" (ver Figura 23) sin y con *dwell time*.
- Encuesta en la que se valora del 1 al 5 la experiencia de usuario en dicha tecnología.

De esos veinte usuarios, cinco de ellos realizaron los experimentos de aprendizaje, que son los siguientes:

- Cinco repeticiones de las pruebas de *accuracy* tanto para la realización de gestos sin y con *dwell time*.
- Cinco repeticiones del gesto "*LeftUp*" (ver Figura 23) sin y con *dwell time* con objetivo de medir el tiempo de realización del gesto.

Todas las tablas de resultados mostrarán la media de los mismos, su desviación estándar y su error estándar. Estos parámetros se calculan según la Ecuación 9, la Ecuación 10 y la Ecuación 11, siendo  $x_i$  cada uno de los elementos de la muestra,  $\bar{x}$  la media y  $n$  el número de elementos, que puede ser 20 ó 5, dependiendo del experimento.

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i \quad \text{Ecuación 9}$$

$$SD = \frac{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2}}{n - 1} \quad \text{Ecuación 10}$$

$$Error = SD/\sqrt{n} \quad \text{Ecuación 11}$$

#### 4.4.1. Experimento de *accuracy*

En la Tabla 1 pueden verse los resultados obtenidos para el experimento de *accuracy* para los veinte usuarios. Los gráficos de la Figura 26 han sido obtenidos a partir de los datos de la Tabla 1, obteniendo los valores medios y de error para ambos experimentos.

	<b>Sin dwell time</b>	<b>Con dwell time</b>
Sujeto 1	87,5	100,0
Sujeto 2	100,0	100,0
Sujeto 3	75,0	100,0
Sujeto 4	87,5	87,5
Sujeto 5	75,0	100,0
Sujeto 6	100,0	100,0
Sujeto 7	75,0	100,0
Sujeto 8	87,5	87,5
Sujeto 9	75,0	75,0
Sujeto 10	50,0	50,0
Sujeto 11	62,5	87,5
Sujeto 12	87,5	100,0
Sujeto 13	62,5	87,5
Sujeto 14	75,0	100,0
Sujeto 15	87,5	100,0
Sujeto 16	62,5	87,5
Sujeto 17	75,0	87,5
Sujeto 18	75,0	75,0
Sujeto 19	87,5	100,0
Sujeto 20	75,0	100,0

**Tabla 1.** Porcentaje de acierto para el experimento de *accuracy*, sin y con *dwell time*

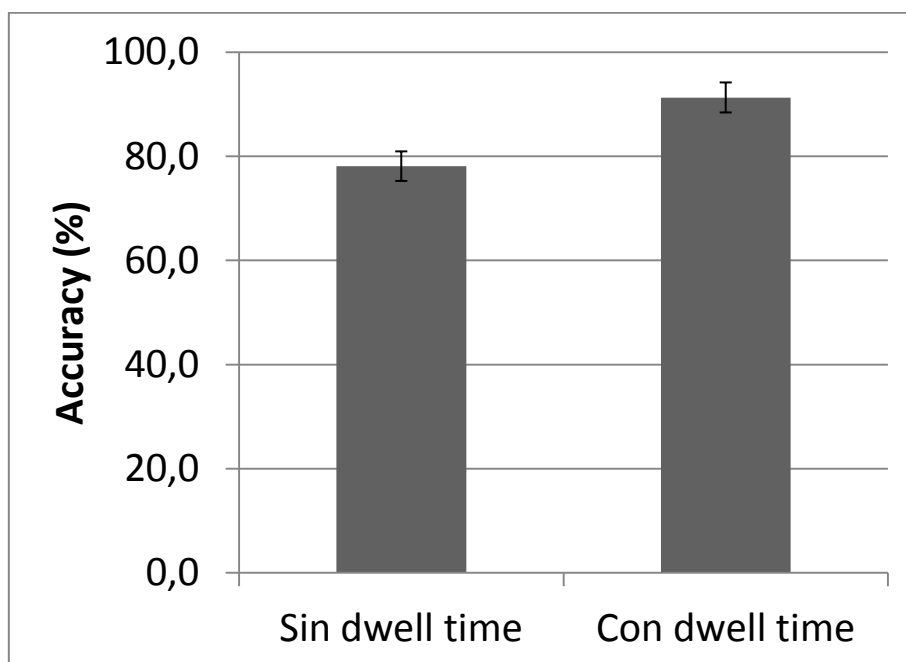


Figura 26. Media y error en el porcentaje de acierto para el experimento de *accuracy*

Lo primero que puede observarse en la Figura 26, es que la precisión (*accuracy*) media en porcentaje en la realización de los ocho gestos de la Figura 23, es mayor en el caso del experimento con *dwell time* que en el experimento que no tiene *dwell time*, siendo el valor en el caso en el que no tiene *dwell time* 78.1% y en el caso del experimento con *dwell time* de 91.2%.

Esto puede ser debido a que a la hora de realizar un gesto, en el caso de ser con *dwell time*, al existir esa señal que marca el inicio y el fin del gesto, ya se está asegurando con ello que el primer y el último punto del gesto sean el correcto, es decir, existe menos probabilidad de fallo al alinear una secuencia de siete puntos que de cinco puntos. Por ejemplo, en la Figura 25 puede observarse que la realización de un gesto (“*LeftUp*”) se corresponde con esta secuencia (C,B,A,D,G,H,I), pero como con *dwell time* estamos marcando el punto C y el punto I, ya solo quedaría alinear la secuencia (B,A,D,G,H).

#### 4.4.2. Experimento de *browsing*

En la Tabla 2 pueden verse los resultados para el experimento de *browsing* para los veinte usuarios que realizaron las pruebas. Estas pruebas constaban de la observación de un pase de diapositivas de 140 segundos de duración, las cuales, eran capturas de pantalla de los menús y aplicaciones típicas existentes en un *Smartphone*. El resultado obtenido en este experimento es el número de falsos positivos (gestos realizados de forma involuntaria) que se han dado mientras funcionaba el algoritmo sin y con *dwell time*.

	<b>Sin dwell time</b>	<b>Con dwell time</b>
Sujeto 1	0	0
Sujeto 2	0	0
Sujeto 3	0	0
Sujeto 4	1	0
Sujeto 5	1	0
Sujeto 6	1	0
Sujeto 7	1	0
Sujeto 8	0	0
Sujeto 9	0	1
Sujeto 10	0	0
Sujeto 11	0	0
Sujeto 12	0	0
Sujeto 13	0	0
Sujeto 14	0	0
Sujeto 15	1	0
Sujeto 16	0	0
Sujeto 17	0	0
Sujeto 18	0	0
Sujeto 19	0	0
Sujeto 20	0	0

**Tabla 2.** Número de falsos positivos en el experimento de *browsing*, sin y con *dwell time*

Lo primero que se debería tener en cuenta es el bajo número de falsos positivos. En la Tabla 2 puede observarse que uno de cada cuatro usuarios tiene un falso positivo en este experimento, mientras que en el caso con *dwell time* solo hubo un usuario que dio un falso positivo.

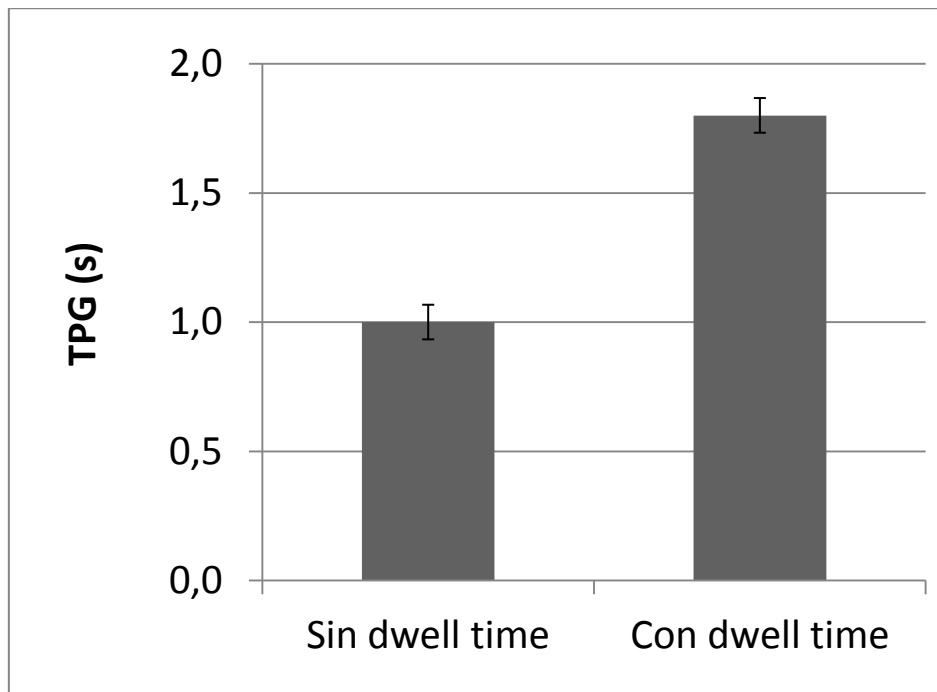
#### 4.4.3. Tiempo de realización de un gesto

En la Tabla 3 se encuentran los tiempos de realización del gesto “*LeftUp*” (ver Figura 23), en segundos, para los casos de sin y con *dwell time*, de los veinte sujetos que realizaron el experimento, con precisión de una décima de segundo. Con los datos de la Tabla 3 se ha realizado la Figura 27, en la cual se representa el tiempo medio de los experimentos y sus correspondientes barras de error, para poder compararlos fácilmente.

El tiempo de realización del gesto será denominado a partir de ahora TPG (*Time per Gesture*), para simplificar su uso. Este tiempo será medido en segundos con una décima de precisión, como se comentó anteriormente.

	<b>Sin dwell time</b>	<b>Con dwell time</b>
Sujeto 1	0,8	1,5
Sujeto 2	0,5	1,2
Sujeto 3	1,0	1,7
Sujeto 4	0,6	1,4
Sujeto 5	0,9	1,5
Sujeto 6	0,6	1,4
Sujeto 7	0,5	1,5
Sujeto 8	1,1	1,6
Sujeto 9	1,5	2,1
Sujeto 10	0,8	1,5
Sujeto 11	1,2	2,2
Sujeto 12	1,5	2,6
Sujeto 13	0,7	2,0
Sujeto 14	1,0	1,9
Sujeto 15	0,9	2,0
Sujeto 16	0,9	1,9
Sujeto 17	1,2	2,0
Sujeto 18	1,5	1,7
Sujeto 19	1,2	2,0
Sujeto 20	1,1	1,8

**Tabla 3.** Tiempo de la realización de un gesto, sin y con *dwell time*, en segundos



**Figura 27.** Media y error para el tiempo de la realización de un gesto, sin y con *dwell time*

Como puede apreciarse en la Figura 27, el tiempo medio de realización de un gesto, en el caso sin *dwell time* es de 1.0 s. En cambio, el tiempo de realización de un gesto con *dwell time* es mucho mayor, 1.8 s de media. Con esto cabe destacar que la realización de un gesto sin *dwell time* es mucho más rápida que con *dwell time*.

Este resultado era esperado, ya que para realizar el experimento con *dwell time* se requiere un tiempo de fijación, tanto para marcar el comienzo, como el final del gesto, tiempo que no es necesario a la hora de realizar el gesto sin *dwell time*.

#### 4.4.4. Efectos del aprendizaje

En este apartado, se analizarán los efectos que tiene el aprendizaje en el uso de gestos con el dispositivo móvil, tanto en el experimento de *accuracy* o precisión, como en el tiempo de realización de un gesto. Para ello se han hecho cinco repeticiones de cada experimento por cada uno de los cinco usuarios que lo realizaron, para los experimentos sin y con *dwell time*.

##### 4.4.4.1. Aprendizaje en el experimento de *accuracy*

En la Tabla 4 se puede ver los resultados del experimento de *accuracy* para el caso sin *dwell time*, para cada una de las cinco repeticiones, para los cinco usuarios. En la Tabla 5 se presentan los mismos resultados, pero para el experimento con *dwell time*. La media y los errores por intento para ambos casos está disponible en la Figura 28 en forma de gráfico, para facilitar su comparativa.

Intento	1	2	3	4	5
Sujeto 1	75,0	75,0	100,0	100,0	100,0
Sujeto 2	75,0	75,0	75,0	87,5	87,5
Sujeto 3	50,0	75,0	62,5	75,0	75,0
Sujeto 4	62,5	75,0	75,0	87,5	87,5
Sujeto 5	75,0	75,0	87,5	87,5	87,5

**Tabla 4.** Porcentaje de acierto para el aprendizaje en el experimento de *accuracy* sin *dwell time*

Intento	1	2	3	4	5
Sujeto 1	100,0	100,0	100,0	100,0	100,0
Sujeto 2	87,5	87,5	100,0	87,5	100,0
Sujeto 3	87,5	75,0	87,5	87,5	100,0
Sujeto 4	75,0	87,5	100,0	87,5	100,0
Sujeto 5	100,0	100,0	100,0	87,5	100,0

**Tabla 5.** Porcentaje de acierto para el aprendizaje en el experimento de *accuracy* con *dwell time*

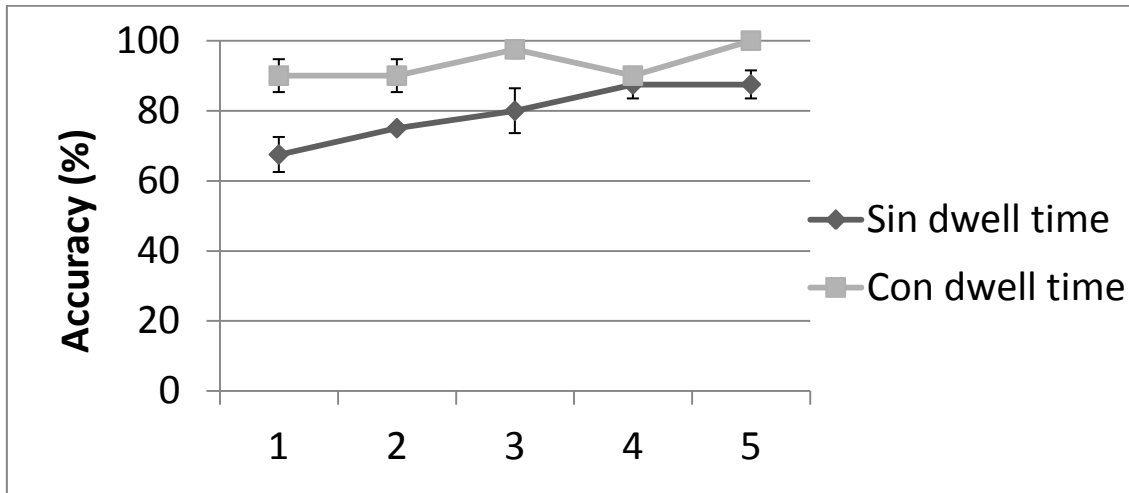


Figura 28. Porcentaje de acierto medio y errores para el experimento de *accuracy* con y sin *dwell time*

En la Figura 28 puede apreciarse la tendencia media a lo largo de los intentos. Aunque siempre son mejores los resultados para el caso del experimento con *dwell time*, ambos mejoran su porcentaje de precisión a lo largo del experimento. En el caso sin *dwell time*, se parte de un porcentaje medio del 67.5% en el primer intento para llegar con los mismos usuarios al 87.5% de acierto en el quinto intento. Lo mismo ocurre en el experimento con *dwell time*, partiendo de un 90% de media en el primer intento para finalizar en un 100% de acierto en el quinto intento. Se puede concluir que los resultados para el experimento de *accuracy* mejoran con aprendizaje.

#### 4.4.4.2. Aprendizaje en el tiempo de realización de un gesto

En la Tabla 6 se puede ver los resultados para el tiempo de realización de un gesto en el caso sin *dwell time*, para cada una de las cinco repeticiones para los cinco usuarios. En la Tabla 7 se presentan los mismos resultados, pero para el caso con *dwell time*. La media y los errores por intento para ambos casos está disponible en la Figura 29, para facilitar su comparativa.

Intento	1	2	3	4	5
Sujeto 1	0,5	0,6	0,5	1,0	0,6
Sujeto 2	0,9	0,8	0,7	0,7	0,7
Sujeto 3	0,8	1,0	0,8	0,7	0,9
Sujeto 4	1,1	1,2	1,0	1,0	0,8
Sujeto 5	1,0	0,9	0,9	0,8	0,8

Tabla 6. TPG en el aprendizaje en el experimento de *accuracy* sin *dwell time*

Intento	1	2	3	4	5
Sujeto 1	1,5	1,5	1,3	1,6	1,4
Sujeto 2	1,9	2,0	1,7	1,9	1,6
Sujeto 3	2,0	1,8	1,7	1,7	1,7
Sujeto 4	1,9	1,7	1,5	1,7	1,6
Sujeto 5	1,9	2,0	1,8	1,8	1,6

Tabla 7. TPG en el aprendizaje en el experimento de *accuracy* con *dwell time*



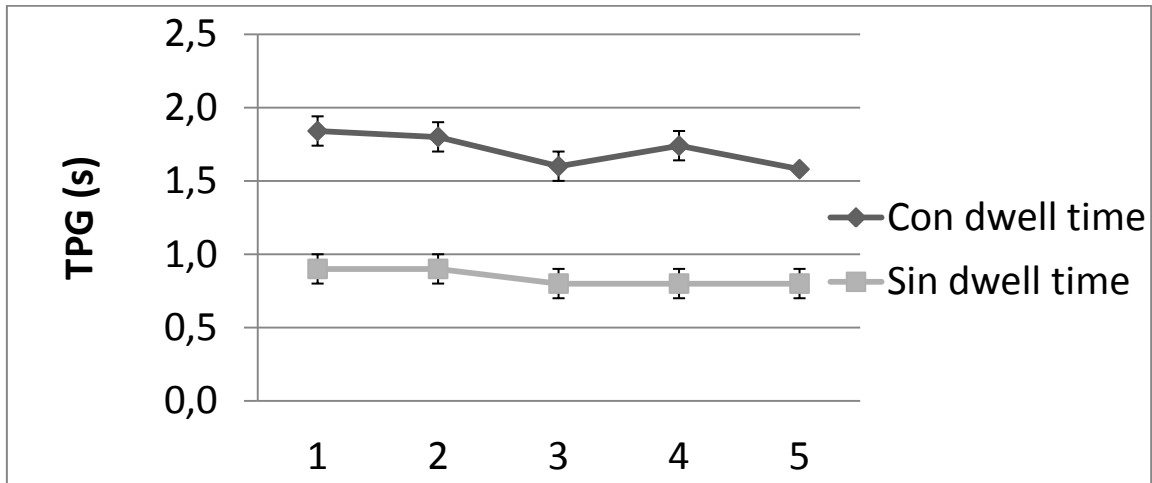


Figura 29. Comparativa de los efectos de aprendizaje para el tiempo por gesto con y sin *dwell time*

En la Figura 29 puede verse que el tiempo de realización de un gesto disminuye a lo largo de los cinco intentos para la misma persona, aunque el tiempo de realización de un gesto sin *dwell time* sigue siendo del orden de la mitad que uno hecho con *dwell time* a lo largo de las sucesivas realizaciones. De acuerdo a la Figura 29, en el caso sin *dwell time*, se pasa de un TPG medio de 0.9 s en el primer intento a un TPG medio de 0.8 s en el quinto intento. En la misma línea, en el caso con *dwell time*, se pasa de un TPG medio de 1.8 s en el primer intento a uno de 1.6 s en el quinto, es decir, una mejora mayor que en el caso sin *dwell time* aunque siga siendo un valor de más del doble que en el caso sin *dwell time*.

#### 4.4.5. Opinión de los usuarios

Al final del experimento, fue realizada una encuesta a cada uno de los veinte usuarios que hicieron los experimentos de *accuracy*, *browsing*, y de tiempo por gesto. En ella, se valoran para los dos sistemas propuestos (con y sin *dwell time*) cuatro aspectos básicos sobre su funcionamiento, valorados del 1 al 5:

- Si es fácil o no de usar (1, muy difícil; 5, muy fácil).
- Si es un sistema rápido (1, muy lento; 5, muy rápido).
- Si es cansado (1, poco cansado; 5 muy cansado).
- Si es preciso (1, poco preciso; 5, muy preciso).

En la Tabla 8 se muestran las opiniones de los veinte usuarios acerca del sistema sin *dwell time*, y en la Tabla 9 se muestran los mismos valores, pero para el experimento sin *dwell time*. En la Figura 30 se muestra la comparativa para ambos sistemas en media y error sobre lo opinado por los sujetos que realizaron los experimentos.

	Fácil de usar	Rápido	Cansado	Preciso
Sujeto 1	4	4	4	2
Sujeto 2	5	5	4	3
Sujeto 3	4	3	3	3
Sujeto 4	5	4	1	3
Sujeto 5	4	2	4	3
Sujeto 6	5	3	4	3
Sujeto 7	2	3	4	3
Sujeto 8	4	4	3	3
Sujeto 9	4	4	3	3
Sujeto 10	4	4	3	3
Sujeto 11	4	4	4	3
Sujeto 12	2	3	5	4
Sujeto 13	4	4	3	4
Sujeto 14	5	3	4	3
Sujeto 15	2	1	5	2
Sujeto 16	4	5	3	3
Sujeto 17	3	4	2	3
Sujeto 18	3	3	3	3
Sujeto 19	5	5	1	3
Sujeto 20	3	3	2	2

**Tabla 8.** Opinión del 1 al 5 de los usuarios acerca del sistema sin *dwell time*

	Fácil de usar	Rápido	Cansado	Precisión
Sujeto 1	3	2	5	4
Sujeto 2	3	3	5	5
Sujeto 3	2	2	4	5
Sujeto 4	2	1	5	5
Sujeto 5	3	1	3	5
Sujeto 6	2	1	5	4
Sujeto 7	4	3	3	4
Sujeto 8	4	2	4	5
Sujeto 9	3	2	4	3
Sujeto 10	5	5	2	3
Sujeto 11	3	3	4	3
Sujeto 12	4	2	5	4
Sujeto 13	4	3	3	4
Sujeto 14	2	2	5	5
Sujeto 15	1	1	5	3
Sujeto 16	5	5	4	5
Sujeto 17	4	1	5	4
Sujeto 18	4	4	3	4
Sujeto 19	3	3	3	3
Sujeto 20	4	4	1	3

**Tabla 9.** Opinión del 1 al 5 de los usuarios acerca del sistema con *dwell time*

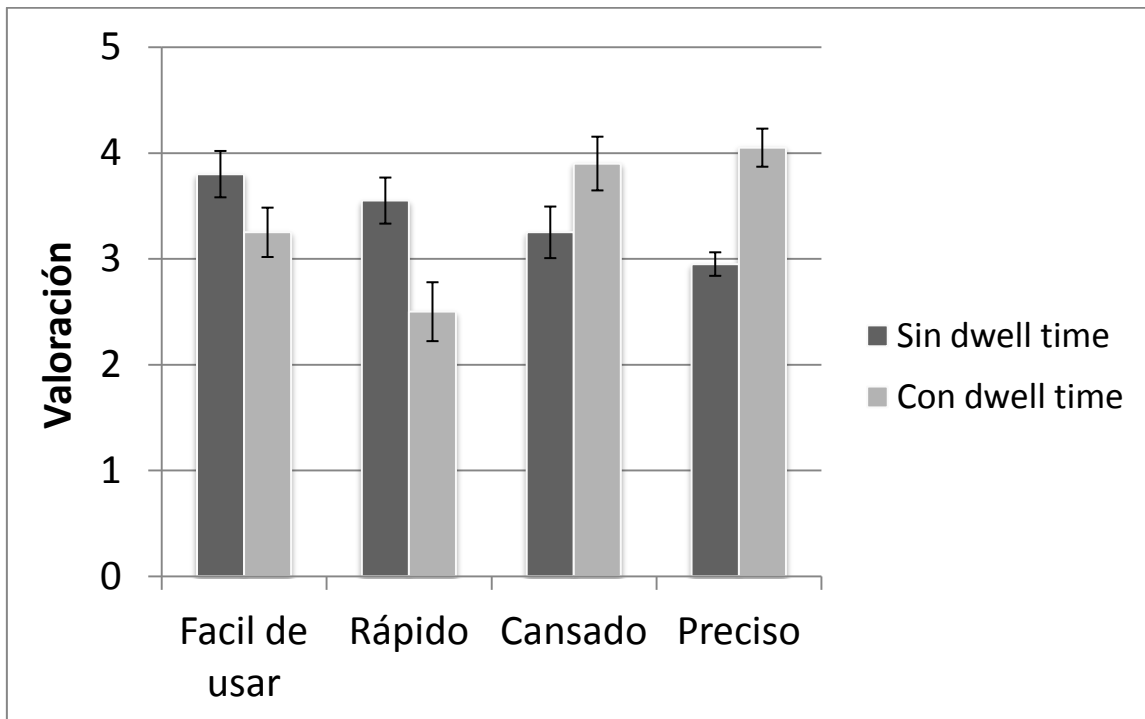


Figura 30. Gráfica comparativa de la opinión del 1 al 5 de los usuarios acerca del sistema con y sin *dwell time*

En la Figura 30 puede verse la comparativa, en media, de la opinión de los usuarios en cada sistema. Puede observarse que los usuarios hacen la siguiente valoración del mismo:

- El sistema sin *dwell time* es más fácil de usar que el que no tiene *dwell time*.
- El sistema sin *dwell time* es más rápido que el que no tiene *dwell time*.
- El sistema con *dwell time* es más cansado que el que no tiene *dwell time*.
- El sistema con *dwell time* es más preciso que el que no tiene *dwell time*.

Puede verse también, que según los usuarios, el punto fuerte del sistema sin *dwell time* respecto al sistema con *dwell time* es la rapidez, y el punto fuerte del sistema con *dwell time* respecto al sin *dwell time* es la precisión. También hay que destacar un aspecto positivo: el usuario que nunca ha utilizado esta tecnología la considera, en líneas generales, fácil de usar, a la vez que tiene un lado negativo, que consideran que es cansada.



# 5. Conclusiones y trabajo futuro

---

## 5.1. Conclusiones

A lo largo de la historia de los ordenadores primero, y luego en otros dispositivos como las tabletas, teléfonos móviles, pantallas de información, cajeros automáticos, etc. ha sido siempre necesaria tener una capacidad de interacción con los mismos. A la vez que surgieron las primeras máquinas, surgieron también las primeras interfaces, como tarjetas perforadas, teclados, o posteriormente, ratones o interfaces táctiles.

La llegada al mercado de los teléfonos móviles inteligentes, también llamados *Smartphone*, y las tabletas, abrieron la puerta a la interfaz táctil en estos sistemas, limitada hasta ahora en su uso en cajeros automáticos o pantallas de información. Pero el uso táctil moderno es, en su mayor parte a base de gestos: para desbloquear la pantalla, acceder al correo, acceder y ampliar fotografías, acceso a redes sociales,...

Por otra parte, aparece un problema en este tipo de interacción: necesita las manos. Existen casos en los cuales no se puede o no es conveniente hacer uso de las mismas, ya sea por que el usuario tiene algún tipo de discapacidad o lesión temporal, o por el simple hecho de no poder usarlas, por tenerlas ocupadas en ese momento. Estas situaciones también aparecen en contextos donde no se puede utilizar el control por voz: ambientes muy ruidosos o por el contrario que requieren silencio.

Teniendo en cuenta los dos hechos anteriores: el uso de gestos en *Smartphones* y que no sean necesariamente táctiles, abre una nueva alternativa: la interacción en un *Smartphone* a través de gestos hechos con los ojos (*Gaze Gestures*), para lo cual es necesaria la tecnología de *Eye Tracking* o seguimiento de los ojos. Para poder usar dicha tecnología, hace falta una serie de elementos *hardware* como cámara IR y luces IR, que han sido integradas en un soporte de fabricación casera. También es necesaria una parte *software*, como el programa de *Eye Tracking* y los algoritmos de reconocimiento de los *Gaze Gestures*, que se ejecutarán en un ordenador.

Lo primero que se ha llevado a cabo en el proyecto ha sido el diseño e implementación del prototipo de bajo coste en el que se han realizado las pruebas, siguiendo una serie de restricciones ya comentadas, para, acto seguido, probar en el mismo su funcionamiento con ayuda del software para *Eye Tracking* a través de la interacción por gestos o *Gaze Gestures*.

A la vista de lo comentado anteriormente, cabe destacar primero, que el objetivo de este proyecto ha sido demostrar que es posible realizar un sistema que use la tecnología de *Gaze Tracking* a través de gestos visuales o *Gaze Gestures*, de bajo coste, más que encontrar el mejor sistema que utilice esta tecnología. A continuación se pasarán a exponer tanto las ventajas como los inconvenientes del sistema propuesto.

## Ventajas

Primero, hay que señalar que es de las primeras veces que se crea un dispositivo de este tipo de manera experimental, ya que de manera comercial no existen aún dispositivos de *Gaze Tracking* para teléfonos móviles tipo *Smartphone*, siendo un problema a la hora de encontrar tecnología de este tipo.

El sistema que se ha propuesto en este trabajo, tiene la gran ventaja de tener un coste aproximado de unos 40€, como se dijo anteriormente, y es de realización relativamente casera, es decir, se puede implementar uno de similares características, siempre que se tengan en cuenta las recomendaciones que se indican en el punto 3.1.

A continuación, se presentan los precios de algunos de los dispositivos de *Gaze Tracking* para PC disponibles en el mercado, ya que como se ha comentado anteriormente, no existen aún dispositivos para *Smartphone*.

Sistema	Precio
ERICA	\$8000
EyeTech TM3	Entre \$6480 y \$14280
VisionKey	\$4995
SeeTech	\$8000
The Eyegaze Communication System	Entre \$7,250 y \$13,500
VISIOBOARD	20000€
ECOpoint	\$7250
TE-9100 Nursing System	\$29800
Tobii	13800€

Tabla 10. Precios de diferentes sistemas de *Eye Tracking*<sup>7</sup>

Como puede observarse en la Tabla 10, los precios de los dispositivos existentes en el mercado oscilan entre los \$7250 hasta los \$29800, siendo todos ellos mucho más caros que el dispositivo propuesto en esta memoria.

Este hecho facilitaría el acceso a un sistema de *Gaze Tracking* a cualquier persona, siendo muy necesario su uso en personas con movilidad reducida, ya que la interacción con un dispositivo de este estilo siempre ha sido manual.

## Inconvenientes

El principal inconveniente de este prototipo es que tanto el programa de *Eye Tracking*, como los algoritmos de reconocimiento de los *Gaze Gestures* no se ejecutan en el *Smartphone* en cuestión, sino en un ordenador, con lo que este sistema depende de estar conectado al mismo tanto por la cámara, como por el programa *TeamVlewer*, como se comentó en el apartado 3.2.

<sup>7</sup> [http://wiki.cogain.info/index.php/Eye\\_Trackers](http://wiki.cogain.info/index.php/Eye_Trackers)

El control del dispositivo es limitado, ya que al estar todo el sistema integrado en el soporte, depende de la posición de la cabeza para su funcionamiento, ya que hay que estar colocado en una posición determinada para que el dispositivo capte los movimientos oculares.

A pesar de estos inconvenientes, los resultados de los experimentos que se han hecho en este trabajo apoyan la hipótesis de que es posible controlar un teléfono móvil a través de gestos oculares utilizando una tecnología al alcance de cualquier bolsillo.

## 5.2. Trabajo futuro

El trabajo futuro en este campo puede ir orientado a salvar los dos inconvenientes antes citados: integrar todo el funcionamiento del dispositivo en el teléfono y hacer que el mismo no dependa de la posición de la cabeza a la hora de realizar un gesto.

El primer caso consistiría en crear una aplicación *Android* o *iOS* (sistema operativo de *iphone*) de *Eye Tracking*, similar al *ITU Gaze Tracker* (visto en 3.3.2), con la salvedad de que esta se ejecutaría directamente en el *Smartphone*. Además, también habría que convertir los códigos del algoritmo de reconocimiento de los *Gaze Gestures* para que pudieran ser compilados en *Android*, así podrían ejecutarse directamente en el teléfono móvil y evitar la dependencia del sistema con un ordenador adicional. La capacidad multicore de los teléfonos actuales podría permitir la ejecución en tiempo real.

El segundo caso podría salvarse si el sistema fuera de tipo *head mounted*, es decir, estuviera colocado sobre la cabeza, como en unas gafas o una gorra. Una forma de hacerlo sería colocar la cámara y las luces, además de una alimentación para las mismas, en unas gafas, similares a las que se proponen en *Project Glass* de Google<sup>8</sup>. Este proyecto va enfocado a crear unas gafas de realidad aumentada, las cuales sirven para que con los ojos, mediante *Eye Tracking*, y con comandos de voz se pueda obtener información de un *Smartphone* sin usar las manos, dando la capacidad de realizar acciones tales como leer el correo, acceder a la agenda, etc.

Como puede verse, su objetivo no es otro que obtener información de un dispositivo utilizando los ojos aunque en un principio solo con *Eye Tracking*, pudiendo ser las gafas que se utilizan en *Project Glass* un ejemplo para salvar el inconveniente de la posición de la cabeza en dispositivo implementado. En esta línea, se han propuesto sistemas de tipo *head mounted* como en (Abbott and Faisal, 2012), cuyo sistema de *Eye Tracking* es independiente de la posición de la cabeza.

Finalmente, hay que destacar que el control por gestos oculares puede combinarse con otros métodos clásicos como el control táctil o por voz, para mejorar la velocidad o la eficiencia de la interacción mediante esta combinación multimodal.

---

<sup>8</sup> <https://plus.google.com/+projectglass>





# Referencias

---

- Abbott, W. W., and Faisal, A. A. (2012). Ultra-low-cost 3D gaze estimation: an intuitive high information throughput compliment to direct brain-machine interfaces. *Journal of neural engineering* 9.
- Agustin, J. S. (2010). Off-the-shelf gaze interaction. Ph.D. thesis. IT University of Copenhagen.
- Dileep, G., and Bobby, J. (2007). The HTM Learning Algorithms. *Machine Learning* 1037, 1–44.
- Drewes, H. (2010). Eye Gaze Tracking for Human Computer Interaction. Ph.D. thesis. Universität München.
- Gaikwad, S. K., Gawali, B. W., and Yannawar, P. (2010). A Review on Speech Recognition Technique. *International Journal of Computer Applications IJCA* 10, 24–28.
- Grossberg, S., Srihasam, K., and Bullock, D. (2012). Neural dynamics of saccadic and smooth pursuit eye movement coordination during visual tracking of unpredictably moving targets. *Neural networks : the official journal of the International Neural Network Society* 27, 1–20.
- Hawkins, J., and Blakeslee, S. (2004). *On Intelligence*. Times Books.
- Hiley, J. B., Redekopp, A. H., and Fazel-Rezai, R. (2006). A low cost human computer interface based on eye tracking. *Conf Proc IEEE Eng Med Biol Soc* 1, 3226–3229.
- Jaimes, A., and Sebe, N. (2007). Multimodal human–computer interaction: A survey. *Computer Vision and Image Understanding* 108, 116–134.
- Mollenbach, E. (2010). Selection strategies in gaze interaction. Ph.D. thesis. Loughborough University.
- Moore, R. J., and Churchill, E. F. (2011). Computer Interaction Analysis: Toward an Empirical Approach to Understanding User Practice and Eye Gaze in GUI-Based Interaction. *Computer Supported Cooperative Work (CSCW)* 20, 497–528.
- Morimoto, C. H., Koons, D., Amit, A., Flickner, M., and Zhai, S. (1999). Keeping an eye for HCI. in XII Brazilian Symposium on Computer Graphics and Image Processing (Cat. No.PR00481) (IEEE Comput. Soc), 171–176.
- Needleman, S. B., and Wunsch, C. D. (1970). A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of molecular biology* 48, 443–53.

- Qvarfordt, P., and Zhai, S. (2005). Conversing with the user based on eye-gaze patterns. in Proceedings of the SIGCHI conference on Human factors in computing systems - CHI'05 (New York, New York, USA: ACM Press), 221–230.
- Rehm, M., and Leichtenstern, K. (2011). Gesture-based mobile training of intercultural behavior. *Multimedia Systems* 18, 33–51.
- Rozado, D. (2011). Analysis and extension of hierarchical temporal memory for multivariable time series. Ph.D. thesis. Universidad Autónoma de Madrid.
- Rozado, D., Agustin, J. S., Rodriguez, F. B., and Varona, P. (2012a). Gliding and saccadic gaze gesture recognition in real time. *ACM Transactions on Interactive Intelligent Systems (TiiS)* 1, 10.
- Rozado, D., Rodriguez, F. B., and Varona, P. (2012b). Extending the bioinspired hierarchical temporal memory paradigm for sign language recognition. *Neurocomputing* 79, 75–86.
- Rozado, D., Rodriguez, F., and Varona, P. (2011). Gaze Gesture Recognition with Hierarchical Temporal Memory Networks. *Lecture Notes in Computer Science* 6691, 1–8.
- San Agustin, J., Skovsgaard, H., Mollenbach, E., Barret, M., Tall, M., Hansen, D. W., and Hansen, J. P. (2010). Evaluation of a low-cost open-source gaze tracker. in Proceedings of the 2010 Symposium on Eye-Tracking Research & Applications - ETRA'10 (New York, New York, USA: ACM Press), 77–80.
- Santis, A. De, and Iacoviello, D. (2009). Robust real time eye tracking for computer interface for disabled people. *Comput Methods Programs Biomed* 96, 1–11.
- Smith, W. M., and Warter, J. (1960). Eye movement and stimulus movement; new photoelectric electromechanical system for recording and measuring tracking motions of the eye. *Journal of the Optical Society of America* 50, 245–250.
- Smith, W. M., and Warter, P. J. (1959). Photoelectric Technique for Measuring Eye Movements. *Science* 130, 1248–1249.
- Stark, L., Vossius, G., and Young, L. R. (1962). Predictive Control of Eye Tracking Movements. *IRE Transactions on Human Factors in Electronics* HFE-3, 52–57.
- Téglás, E., Gergely, A., Kupán, K., Miklósi, Á., and Topál, J. (2012). Dogs' Gaze Following Is Tuned to Human Communicative Signals. *Current Biology* 22, 209–212.
- Wobbrock, J., Rubinstein, J., Sawyer, M., and Duchowski, A. (2007). Gaze-based Creativity Not Typing but Writing: Eye-based Text Entry Using Letter-like Gestures. in The 3rd Conference on Communication by Gaze Interaction - COGAIN 2007.

Yang, X., and Chen, G. (2009). Human-Computer Interaction Design in Product Design. *2009 First International Workshop on Education Technology and Computer Science 2*, 437–439.

Zhu, Z., and Ji, Q. (2004). Eye and gaze tracking for interactive graphic display. *Machine Vision and Applications 15*, 139–148.



# Glosario de acrónimos

---

**HCI:** Human Computer Interaction

**PDA:** Personal Digital Assistant

**SMS:** Short Message Service

**HTM:** Hierarchical Temporal Memory

**IR:** Infrared

**TPG:** Time per Gesture



# Publicaciones

---

**Título:** Controlling a Smartphone Using Gaze Gestures as Input Mechanism

**Autores:** D. Rozado, T. Moreno, J. San Agustín, F.B. Rodriguez, P. Varona

Enviado a **Human-Computer Interaction** en Octubre de 2012. En estado de revisión.

**HUMAN-COMPUTER INTERACTION**, revista indexada en el JCR, factor de impacto 1.476, Q1 en la categoría de COMPUTER SCIENCE, THEORY & METHODS (puesto 15 de 99 revistas en esta categoría en 2011).





# PRESUPUESTO

## 1) Ejecución Material

- Compra de ordenador personal (Software incluido)..... 2000.00 €
- Alquiler de impresora láser durante 6 meses ..... 50.00 €
- Material de oficina ..... 170.00 €
- Material para el soporte ..... 40.00 €
- Total de ejecución material..... 2260.00 €

## 2) Gastos generales

- 16 % sobre Ejecución Material..... 361.60 €

## 3) Beneficio Industrial

- 6 % sobre Ejecución Material..... 135.60 €

## 4) Honorarios Proyecto

- 1000 horas a 15 € / hora..... 15000.00 €

## 5) Material fungible

- Gastos de impresión ..... 100.00 €
- Encuadernación ..... 20.00 €

## 6) Subtotal del presupuesto

- Subtotal Presupuesto..... 17877.20 €

## 7) I.V.A. aplicable

- 21% Subtotal Presupuesto..... 3754,21 €

## 8) Total presupuesto

- Total Presupuesto ..... 21631,41 €

Madrid, Enero de 2013

El Ingeniero Jefe de Proyecto

Fdo.: Tamara Moreno Martín

Ingeniero Superior de Telecomunicación



## PLIEGO DE CONDICIONES

Este documento contiene las condiciones legales que guiarán la realización, en este proyecto, de un **prototipo de Gaze Tracking de bajo coste**. En lo que sigue, se supondrá que el proyecto ha sido encargado por una empresa cliente a una empresa consultora con la finalidad de realizar dicho sistema. Dicha empresa ha debido desarrollar una línea de investigación con objeto de elaborar el proyecto. Esta línea de investigación, junto con el posterior desarrollo de los programas está amparada por las condiciones particulares del siguiente pliego.

Supuesto que la utilización industrial de los métodos recogidos en el presente proyecto ha sido decidida por parte de la empresa cliente o de otras, la obra a realizar se regulará por las siguientes:

### Condiciones generales

1. La modalidad de contratación será el concurso. La adjudicación se hará, por tanto, a la proposición más favorable sin atender exclusivamente al valor económico, dependiendo de las mayores garantías ofrecidas. La empresa que somete el proyecto a concurso se reserva el derecho a declararlo desierto.

2. El montaje y mecanización completa de los equipos que intervengan será realizado totalmente por la empresa licitadora.

3. En la oferta, se hará constar el precio total por el que se compromete a realizar la obra y el tanto por ciento de baja que supone este precio en relación con un importe límite si este se hubiera fijado.

4. La obra se realizará bajo la dirección técnica de un Ingeniero Superior de Telecomunicación, auxiliado por el número de Ingenieros Técnicos y Programadores que se estime preciso para el desarrollo de la misma.

5. Aparte del Ingeniero Director, el contratista tendrá derecho a contratar al resto del personal, pudiendo ceder esta prerrogativa a favor del Ingeniero Director, quien no estará obligado a aceptarla.

6. El contratista tiene derecho a sacar copias a su costa de los planos, pliego de condiciones y presupuestos. El Ingeniero autor del proyecto autorizará con su firma las copias solicitadas por el contratista después de confrontarlas.

7. Se abonará al contratista la obra que realmente ejecute con sujeción al proyecto que sirvió de base para la contratación, a las modificaciones autorizadas por la superioridad o a las órdenes que con arreglo a sus facultades le hayan comunicado por escrito al Ingeniero Director de obras siempre que dicha obra se haya ajustado a los preceptos de los pliegos de condiciones, con arreglo a los cuales, se harán las modificaciones y la valoración de las diversas unidades sin que el importe total pueda exceder de los presupuestos aprobados. Por consiguiente, el número de unidades que se consignan en el proyecto o en el presupuesto, no podrá servirle de fundamento para entablar reclamaciones de ninguna clase, salvo en los casos de rescisión.

8. Tanto en las certificaciones de obras como en la liquidación final, se abonarán los trabajos realizados por el contratista a los precios de ejecución material que figuran en el presupuesto para cada unidad de la obra.

9. Si excepcionalmente se hubiera ejecutado algún trabajo que no se ajustase a las condiciones de la contrata pero que sin embargo es admisible a juicio del Ingeniero Director de obras, se dará conocimiento a la Dirección, proponiendo a la vez la rebaja de precios que el Ingeniero estime justa y si la Dirección resolviera aceptar la obra, quedará el contratista obligado a conformarse con la rebaja acordada.

10. Cuando se juzgue necesario emplear materiales o ejecutar obras que no figuren en el presupuesto de la contrata, se evaluará su importe a los precios asignados a otras obras o materiales análogos si los hubiere y cuando no, se discutirán entre el Ingeniero Director y el contratista, sometiéndolos a la aprobación de la Dirección. Los nuevos precios convenidos por uno u otro procedimiento, se sujetarán siempre al establecido en el punto anterior.

11. Cuando el contratista, con autorización del Ingeniero Director de obras, emplee materiales de calidad más elevada o de mayores dimensiones de lo estipulado en el proyecto, o sustituya una clase de fabricación por otra que tenga asignado mayor precio o ejecute con mayores dimensiones cualquier otra parte de las obras, o en general, introduzca en ellas cualquier modificación que sea beneficiosa a juicio del Ingeniero Director de obras, no tendrá derecho sin embargo, sino a lo que le correspondería si hubiera realizado la obra con estricta sujeción a lo proyectado y contratado.

12. Las cantidades calculadas para obras accesorias, aunque figuren por partida alzada en el presupuesto final (general), no serán abonadas sino a los precios de la contrata, según las condiciones de la misma y los proyectos particulares que para ellas se formen, o en su defecto, por lo que resulte de su medición final.

13. El contratista queda obligado a abonar al Ingeniero autor del proyecto y director de obras así como a los Ingenieros Técnicos, el importe de sus respectivos honorarios facultativos por formación del proyecto, dirección técnica y administración en su caso, con arreglo a las tarifas y honorarios vigentes.

14. Concluida la ejecución de la obra, será reconocida por el Ingeniero Director que a tal efecto designe la empresa.

15. La garantía definitiva será del 4% del presupuesto y la provisional del 2%.

16. La forma de pago será por certificaciones mensuales de la obra ejecutada, de acuerdo con los precios del presupuesto, deducida la baja si la hubiera.

17. La fecha de comienzo de las obras será a partir de los 15 días naturales del replanteo oficial de las mismas y la definitiva, al año de haber ejecutado la provisional, procediéndose si no existe reclamación alguna, a la reclamación de la fianza.

18. Si el contratista al efectuar el replanteo, observase algún error en el proyecto, deberá comunicarlo en el plazo de quince días al Ingeniero Director de obras, pues transcurrido ese plazo será responsable de la exactitud del proyecto.

19. El contratista está obligado a designar una persona responsable que se entenderá con el Ingeniero Director de obras, o con el delegado que éste designe, para todo relacionado con ella. Al ser el Ingeniero Director de obras el que interpreta el proyecto, el contratista deberá consultarle cualquier duda que surja en su realización.

20. Durante la realización de la obra, se girarán visitas de inspección por personal facultativo de la empresa cliente, para hacer las comprobaciones que se crean oportunas. Es obligación del contratista, la conservación de la obra ya ejecutada hasta la recepción de la

misma, por lo que el deterioro parcial o total de ella, aunque sea por agentes atmosféricos u otras causas, deberá ser reparado o reconstruido por su cuenta.

21. El contratista, deberá realizar la obra en el plazo mencionado a partir de la fecha del contrato, incurriendo en multa, por retraso de la ejecución siempre que éste no sea debido a causas de fuerza mayor. A la terminación de la obra, se hará una recepción provisional previo reconocimiento y examen por la dirección técnica, el depositario de efectos, el interventor y el jefe de servicio o un representante, estampando su conformidad el contratista.

22. Hecha la recepción provisional, se certificará al contratista el resto de la obra, reservándose la administración el importe de los gastos de conservación de la misma hasta su recepción definitiva y la fianza durante el tiempo señalado como plazo de garantía. La recepción definitiva se hará en las mismas condiciones que la provisional, extendiéndose el acta correspondiente. El Director Técnico propondrá a la Junta Económica la devolución de la fianza al contratista de acuerdo con las condiciones económicas legales establecidas.

23. Las tarifas para la determinación de honorarios, reguladas por orden de la Presidencia del Gobierno el 19 de Octubre de 1961, se aplicarán sobre el denominado en la actualidad "Presupuesto de Ejecución de Contrata" y anteriormente llamado "Presupuesto de Ejecución Material" que hoy designa otro concepto.

### **Condiciones particulares**

La empresa consultora, que ha desarrollado el presente proyecto, lo entregará a la empresa cliente bajo las condiciones generales ya formuladas, debiendo añadirse las siguientes condiciones particulares:

1. La propiedad intelectual de los procesos descritos y analizados en el presente trabajo, pertenece por entero a la empresa consultora representada por el Ingeniero Director del Proyecto.

2. La empresa consultora se reserva el derecho a la utilización total o parcial de los resultados de la investigación realizada para desarrollar el siguiente proyecto, bien para su

publicación o bien para su uso en trabajos o proyectos posteriores, para la misma empresa cliente o para otra.

3. Cualquier tipo de reproducción aparte de las reseñadas en las condiciones generales, bien sea para uso particular de la empresa cliente, o para cualquier otra aplicación, contará con autorización expresa y por escrito del Ingeniero Director del Proyecto, que actuará en representación de la empresa consultora.

4. En la autorización se ha de hacer constar la aplicación a que se destinan sus reproducciones así como su cantidad.

5. En todas las reproducciones se indicará su procedencia, explicitando el nombre del proyecto, nombre del Ingeniero Director y de la empresa consultora.

6. Si el proyecto pasa la etapa de desarrollo, cualquier modificación que se realice sobre él, deberá ser notificada al Ingeniero Director del Proyecto y a criterio de éste, la empresa consultora decidirá aceptar o no la modificación propuesta.

7. Si la modificación se acepta, la empresa consultora se hará responsable al mismo nivel que el proyecto inicial del que resulta el añadirla.

8. Si la modificación no es aceptada, por el contrario, la empresa consultora declinará toda responsabilidad que se derive de la aplicación o influencia de la misma.

9. Si la empresa cliente decide desarrollar industrialmente uno o varios productos en los que resulte parcial o totalmente aplicable el estudio de este proyecto, deberá comunicarlo a la empresa consultora.

10. La empresa consultora no se responsabiliza de los efectos laterales que se puedan producir en el momento en que se utilice la herramienta objeto del presente proyecto para la realización de otras aplicaciones.

11. La empresa consultora tendrá prioridad respecto a otras en la elaboración de los proyectos auxiliares que fuese necesario desarrollar para dicha aplicación industrial, siempre que no haga explícita renuncia a este hecho. En este caso, deberá autorizar expresamente los proyectos presentados por otros.

12. El Ingeniero Director del presente proyecto, será el responsable de la dirección de la aplicación industrial siempre que la empresa consultora lo estime oportuno. En caso contrario, la persona designada deberá contar con la autorización del mismo, quien delegará en él las responsabilidades que ostente.