

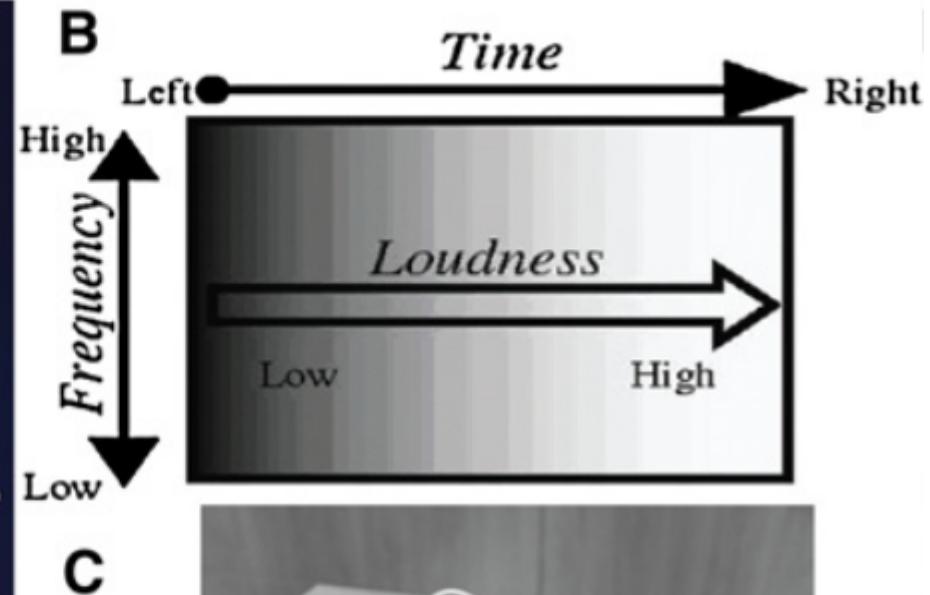
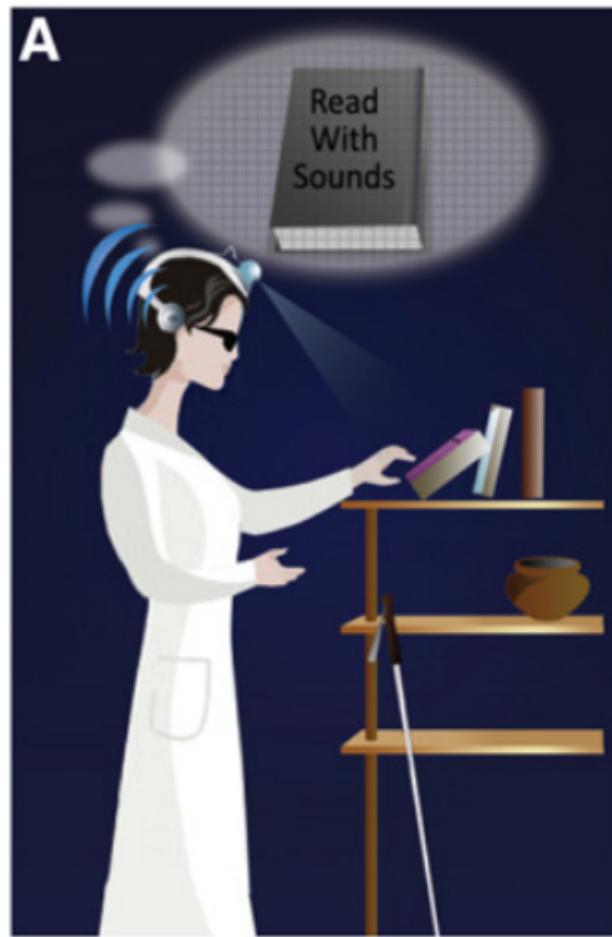
# Charla de seminario

Presentación de los resultados  
(en proceso de construcción)

# Idea detrás del proyecto

- Hay diversos mecanismos de sustitución sensorial.
- Los que reemplazan imagen por sonido son los que permiten mayor resolución y otras potencialidades.
- Con entrenamiento las personas (ciegas o videntes) aprender a reconocer patrones complejos.
- Nosotros queremos estudiar que pasa con el reconocimiento conceptual de geometría en este sistema de sustitución sensorial.

# La tecnología “vOICe”



# Que se puede “ver” con el vOICe

Natural Vision



VOICe SSD (25,344 pixels)  
Maximal Theoretical Resolution



VOICe SSD - Functional Resolution  
(Median Subject - 20/360 Acuity, 4575 pixels)

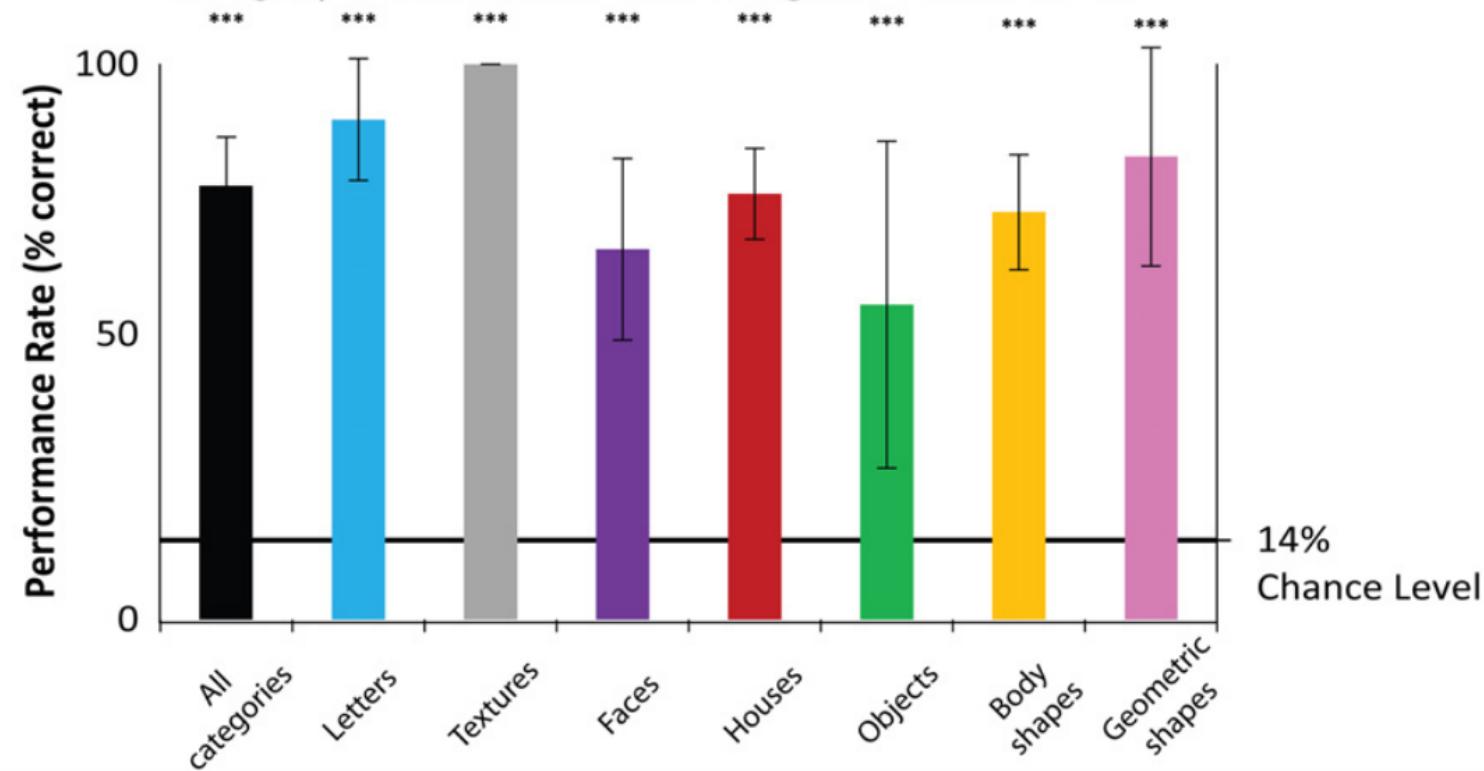


# Experimentos previos

E vOICe representations of the visual images

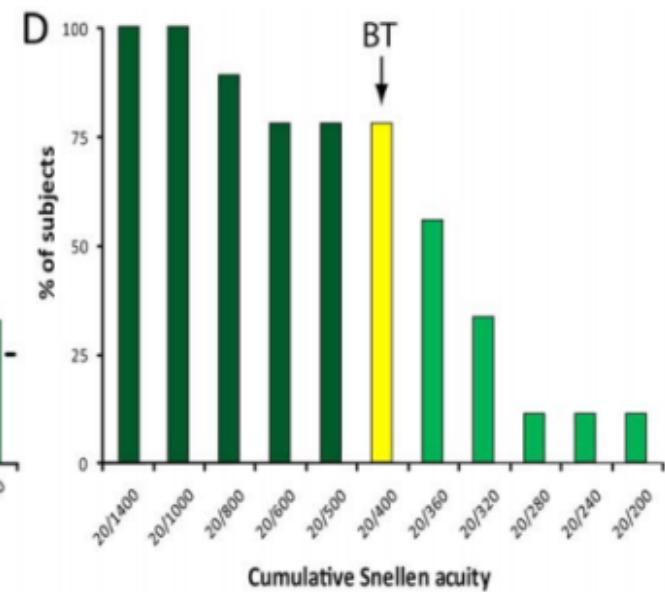
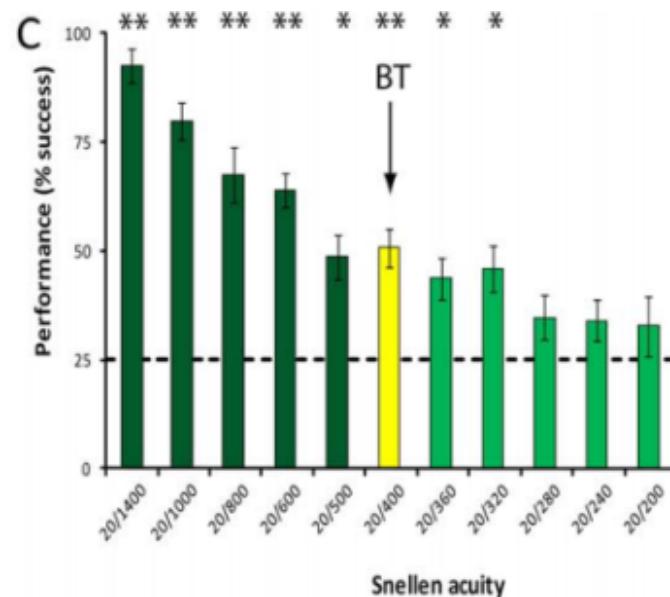


F Category classification task using the vOICe (n=6)

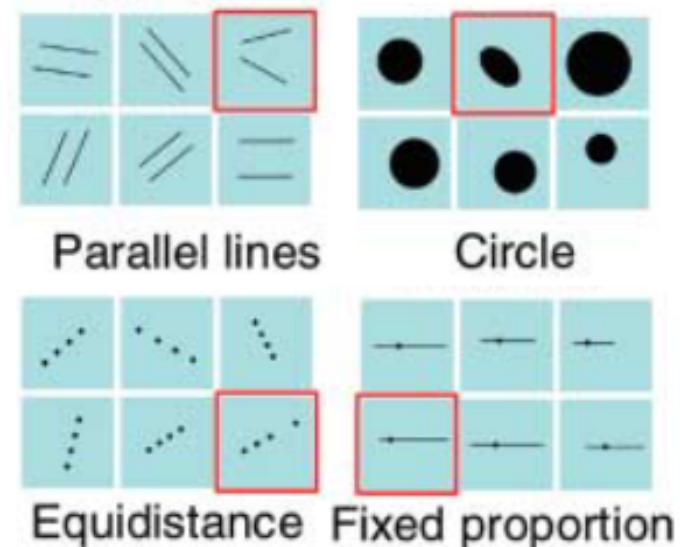
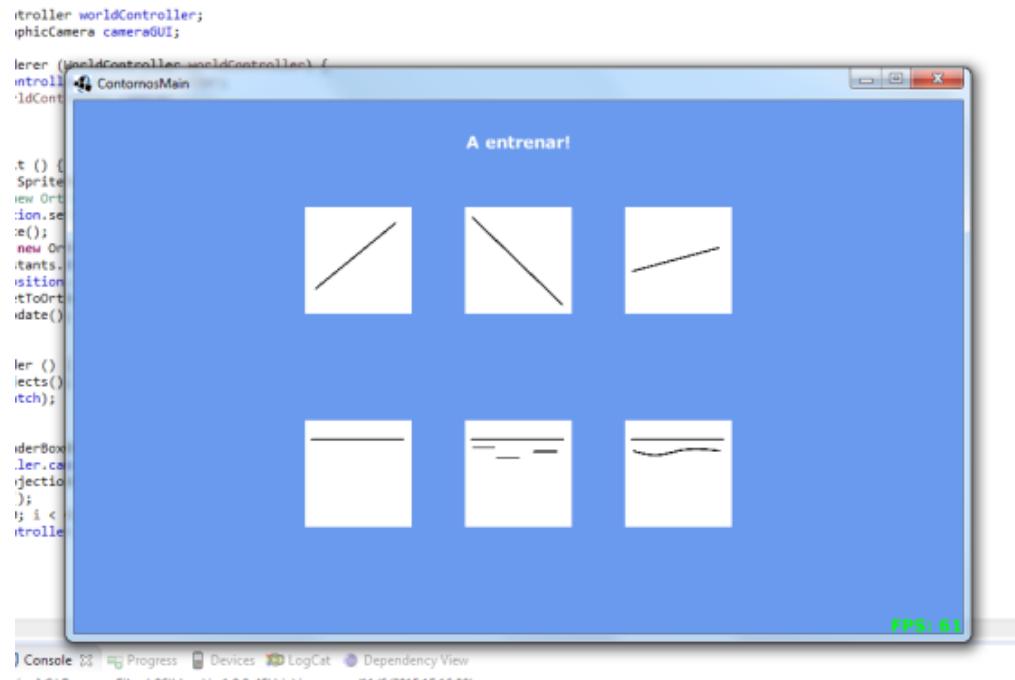


# Experimentos previos

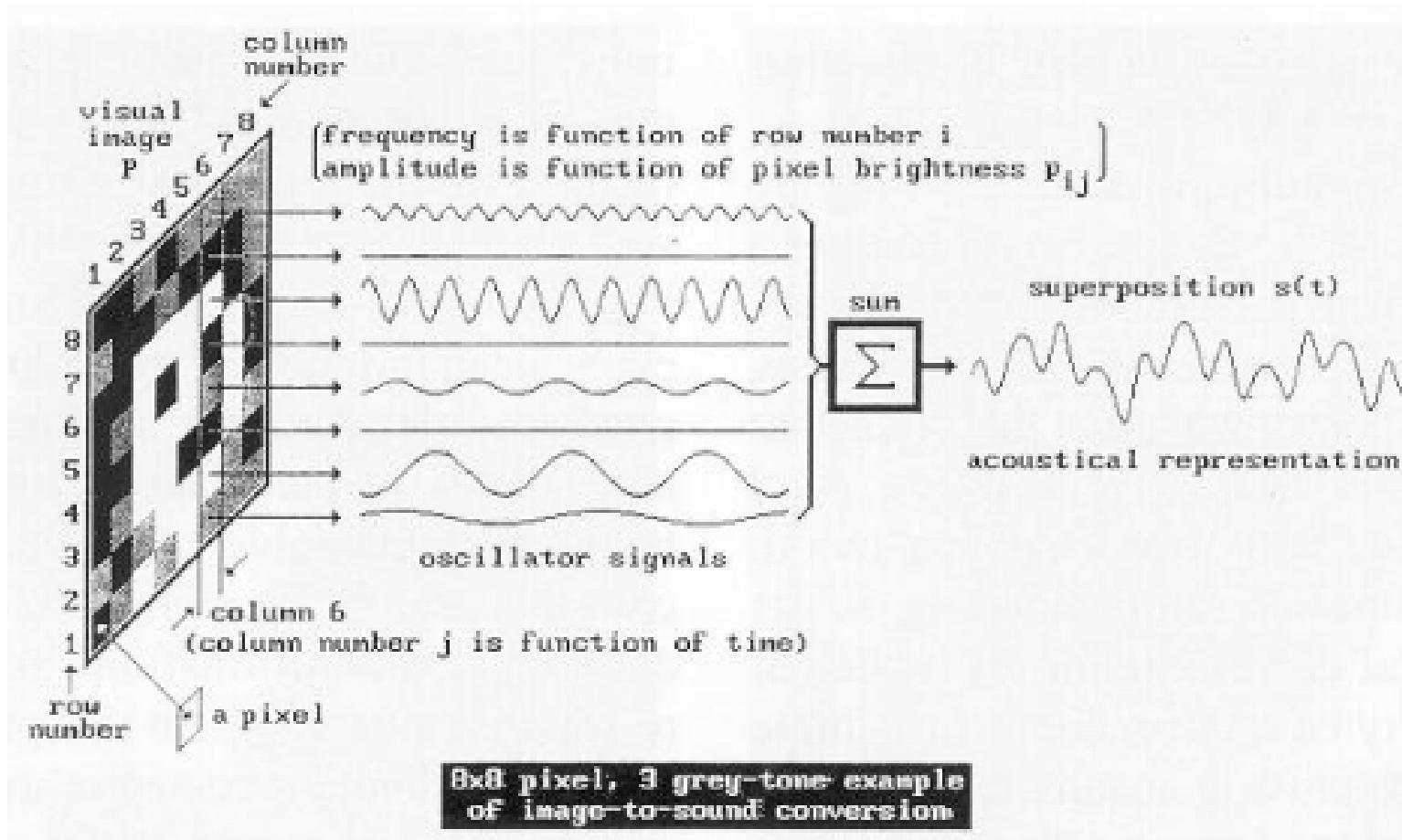
B



# Tipo de experimentos que queríamos hacer en un principio:



# Sistema de codificación del vOICe



# Limitaciones

- Esta pixelado! Eso trae los siguientes problemas:
  - Saltos en fase y amplitud al cambiar de pixel
  - Efectos de batido para frecuencias similares
- Nosotros queremos poder representar imágenes con alta resolución porque sino perdemos la capacidad de medir

# Necesitábamos armar un sistema de representación equivalente pero propio



# Necesitábamos crear patrones geométricos en forma parametrica

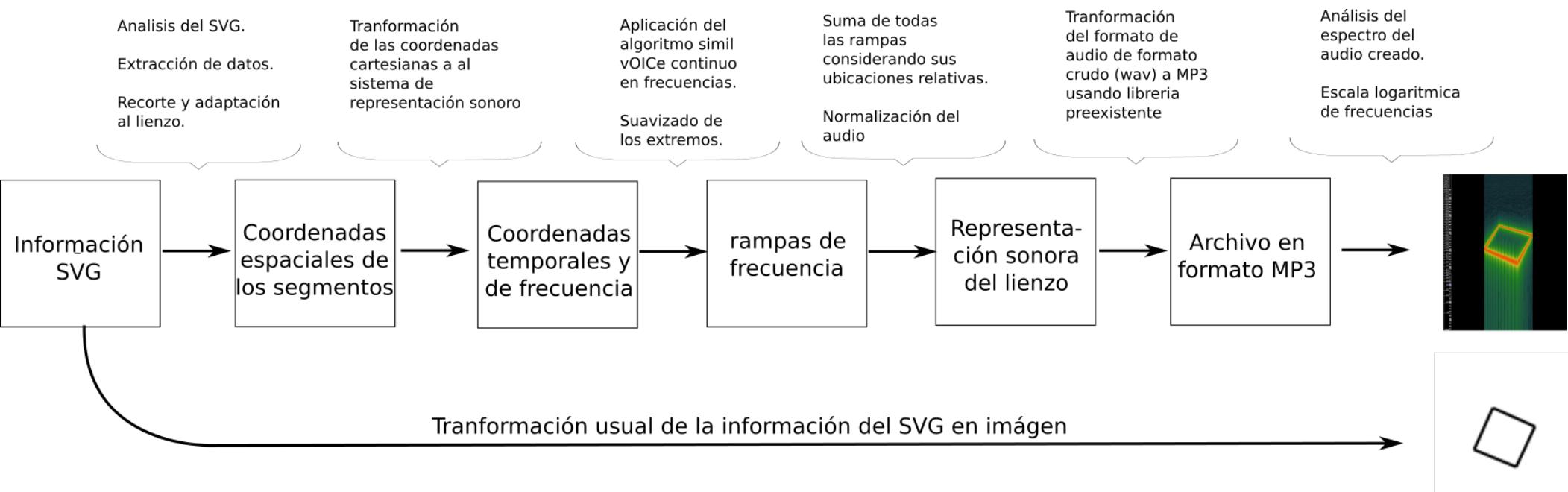
La solución que encontramos fue usar archivos SVG. Ventajas:

- Es un estándar en términos computacionales
- Guarda información de figuras geométricas en forma vectorial sin perdidas de información conceptual
- Se renderiza automáticamente en muchas plataformas
- Es legible por humanos!!!!

```
<!-- Este archivo es creado automaticamente por el generador de contenido del programa contornos version 1.1. Este elementos es el numero 678 de la serie 115 -->
<svg xmlns="http://www.w3.org/2000/svg" height="100" width="100">
<rect stroke-width="5" stroke="#ffffffff" fill="#ffffffff" height="100" width="100" y="0" x="0"/>
<line x1="69.44438" y1="51.12194" x2="56.66159" y2="79.83256" stroke-width="2" stroke="black" />
<line x1="56.6616" y1="79.83256" x2="27.950985" y2="67.04977" stroke-width="2" stroke="black" />
<line x1="40.733772" y1="38.33915" x2="27.950981" y2="67.04977" stroke-width="2" stroke="black" />
<line x1="69.44438" y1="51.12194" x2="40.73377" y2="38.33915" stroke-width="2" stroke="black" />
</svg>
```



# Proceso de transformación



# Algunas cuentas:

Código 1: Constantes y parámetros utilizados en el código que crea transforma la información geométrica en el estímulo sonoro.

---

```
// Define algunas constantes
public final static boolean logScale = true;
public final static boolean fixScale = true;
public final static float maxHeighth = 100;
public final static float frecMax = 4000;
public final static float frecMin = 100;
public final static float time = 5; // in secs
public final static boolean fixedTime = true; // indicate if the length of sound must be the
    variable time
public final static float secByPix = 5 / 100f; // indicate how many sec are represented by
    pixel.
public final static int fs = 44100; // hz of the sound
public final static float base = 10; // base of the log scale
```

---

# Algunas cuentas:

$$t = x \cdot \frac{T}{NroPixels} \quad (2)$$

$$Frec(y) = A \cdot b^y + B \quad (3)$$

con

$$Frec(Y_{max}) = A \cdot b^{y_{max}} + B$$

$$Frec(Y_{min}) = A \cdot b^{y_{min}} + B$$

donde se asume que  $y_{min} = 0$  por lo que reescalando con  $Y^* = Y/Y_{max}$  queda que

# Algunas cuentas:

$$Frec(y*) = A \cdot b^{y*} + B$$

$$Frec(Y*_{max}) = A \cdot b^1 + B$$

$$Frec(Y*_{min}) = A \cdot b^0 + B$$

de donde sale que

$$A = \frac{Frec(Y_{max}) - Frec(Y_{min})}{b^1 - b^0} \quad (4)$$

$$B = ((Frec(Y_{max}) + Frec(Y_{min})) - A \cdot (b^1 + b^0))/2 \quad (5)$$

$$Frec(Y) = A \cdot b^{Y/Y_{max}} + B \quad (6)$$

# Código que genera las rampas

Código 2: Código que genera una rampa de frecuencia que varía en forma continua. Cada una de estas rampas es la representación sonora de un segmento en la lógica del vOICe pero preservando la continuidad de la señal al cambiar en forma continua la frecuencia como se observa en la figura 6

---

```
/*
 * Create a sound that changes frequency in logarithmic form
 *
 *
 * @param freqi
 *         Initial value of frequency
 * @param freqf
 *         Final value of frequency
 * @param ti
 *         Time (in sec) initial
 * @param tf
 *         Time (in sec) final
 * @return
 */
private double[] createMusicRamp(double freqi, double freqf, double ti, double tf) {
    double dt = 1 / (double) fs; // This is the dt that passes between sample and sample
    long N = Math.round((tf - ti) * fs); // The number of samples that need to be created
    double[] freq = logspace(freqi, freqf, N, base); // Creates a logarithmic scale from base 10
    // from the initial frequency to the final
    for (int i = 0; i < freq.length; i++) { // Multiplies by 2pi to work with phase
        freq[i] = freq[i] * 2 * Math.PI;
    }
    // Integrates the instantaneous frequencies
    for (int i = 1; i < freq.length; i++) {
        freq[i] = freq[i - 1] + freq[i] * dt; // The first one leaves it as it is and then adds up to the last
    }
    // Now freq is the phase at each instant

    // We are going to calculate the cosine of the phase
    for (int i = 0; i < freq.length; i++) {
        freq[i] = Math.cos(freq[i]);
    }
    // It has already created the sound

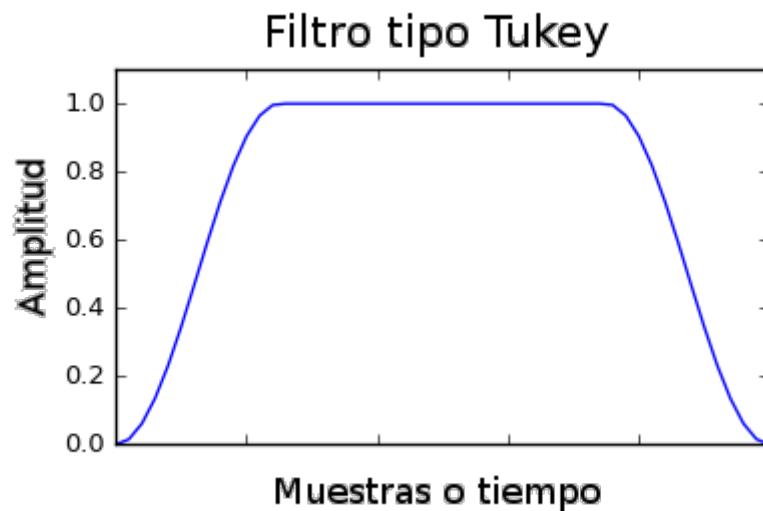
    freq = tukeywin(freq, 0.02);

    return freq;
}
```

---

# Aplicación de filtros

- Para evitar la inclusión de armónicos por culpa de los extremos de las rampas agregamos un filtro tipo tukey



# Código del tukey

Código 3: Código que aplica el filtro tipo Tukey (ver figura 7) para evitar la inclusión de armónicos por el efecto de los contornos.

---

```
/*
 * Aplica una funcion tipo tukeywin (que suaviza los extremos)
 *
 * @param freq
 *      Es el array de datos de entrada
 * @param d
 *      es el parametro de cuanto suavizar. Si es un numero menor que uno asume que es el
 *      porcentaje (por unidad), si es mayor que es numero de frames
 *      a suavizar
 * @return Devuelve el input suavizado
 */
private double[] tukeywin(double[] freq, double d) {
    int framesMaximos;

    if (d >= 1) { // recupera cuantos frames tiene que suavizar
        framesMaximos = (int) d;
    } else {
        framesMaximos = (int) (freq.length * d);
    }

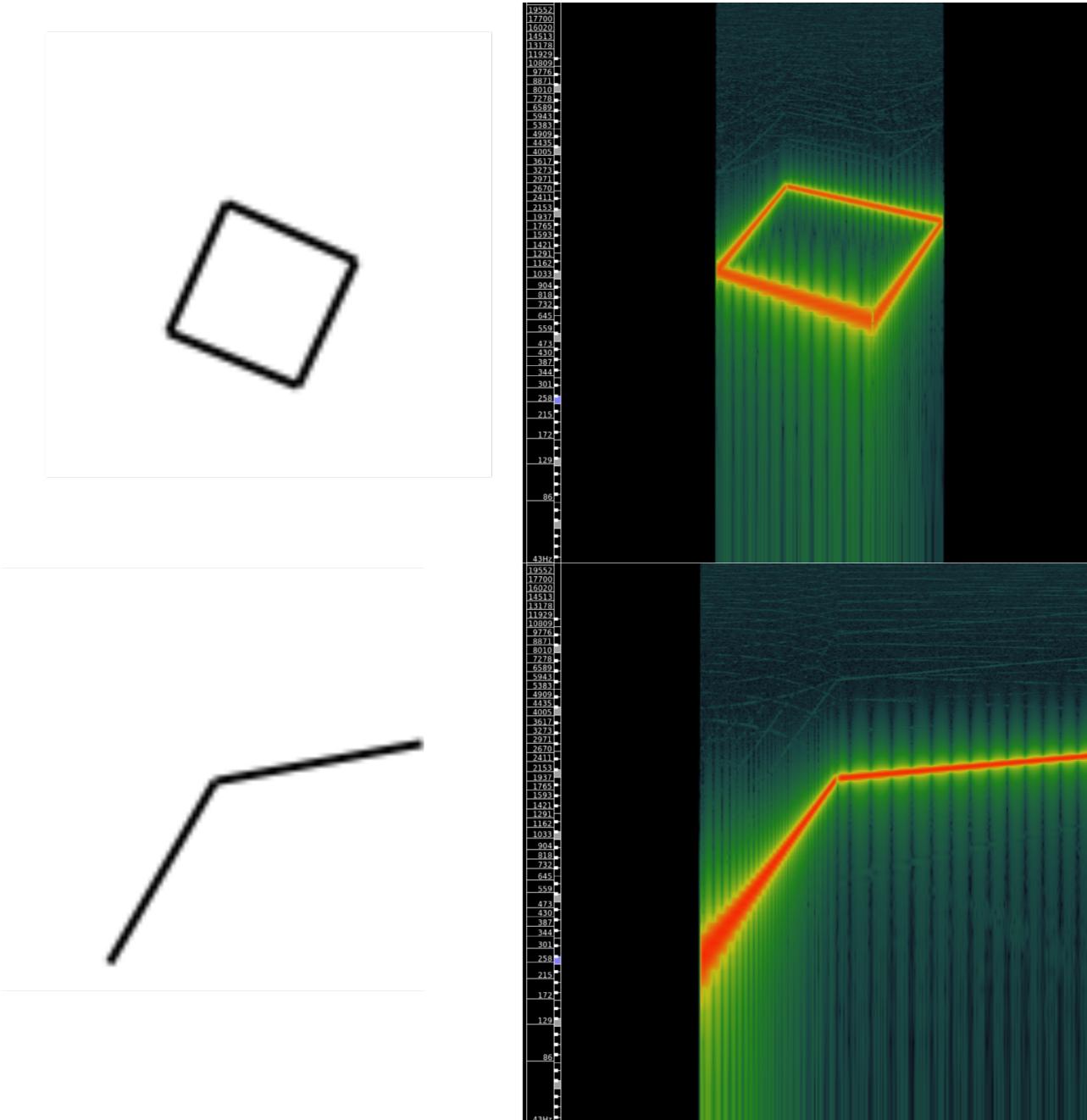
    if (framesMaximos != 0) {

        // suaviza los frames del inicio
        for (int i = 0; i < framesMaximos; i++) {
            double fase = (double) i / framesMaximos * Math.PI;
            double factor = (-Math.cos(fase) + 1) / 2;
            freq[i] = freq[i] * factor;
        }

        // suaviza los frames del final
        for (int i = 0; i < framesMaximos; i++) {
            double fase = (double) i / framesMaximos * Math.PI;
            double factor = (-Math.cos(fase) + 1) / 2;
            freq[freq.length - 1 - i] = freq[freq.length - 1 - i] * factor;
        }
    }
    return freq;
}
```

---

# Ejemplo de resultados obtenidos

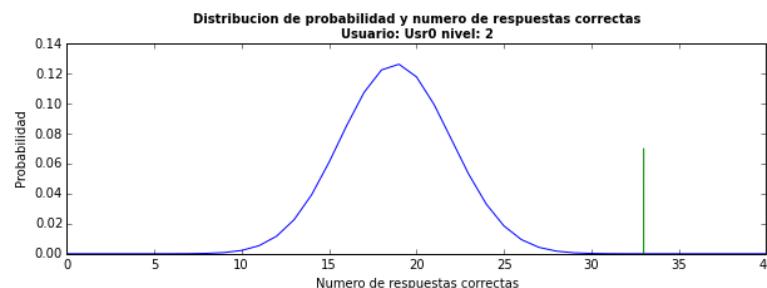


# Con el código que representa la geometría nos propusimos los siguientes experimentos

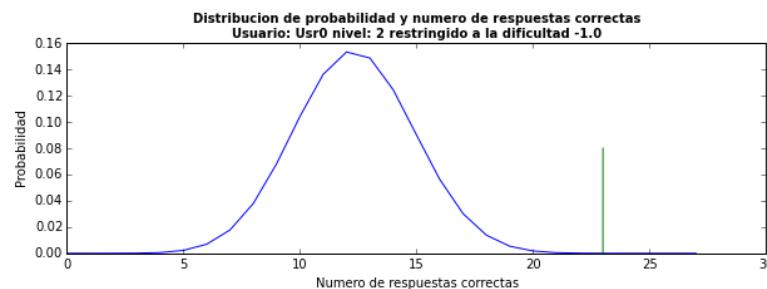
- Experimentos preliminares donde descubrimos que el planteo funcionaba y los usuarios podían distinguir cosas (era importante pero no servía como resultado). Descubrimos una alta dependencia con la orientación
- Experimento donde quisimos probar la dependencia con la orientación
- Experimento donde quisimos probar el efecto del entrenamiento y una eventual transferencia

# Pruebas preliminares

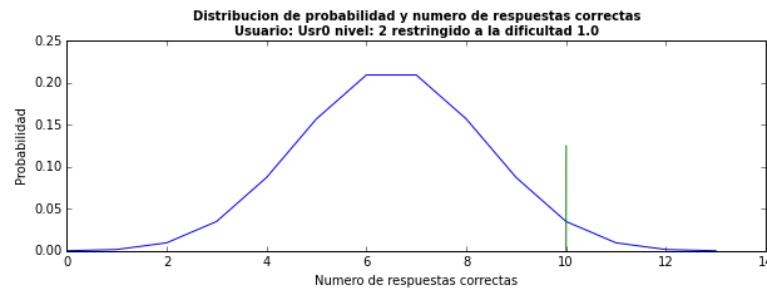
- Esto así no va al informe! Pero no tengo otras figura de esto



haciendo el analisis para la dificultad -1.0



haciendo el analisis para la dificultad 1.0

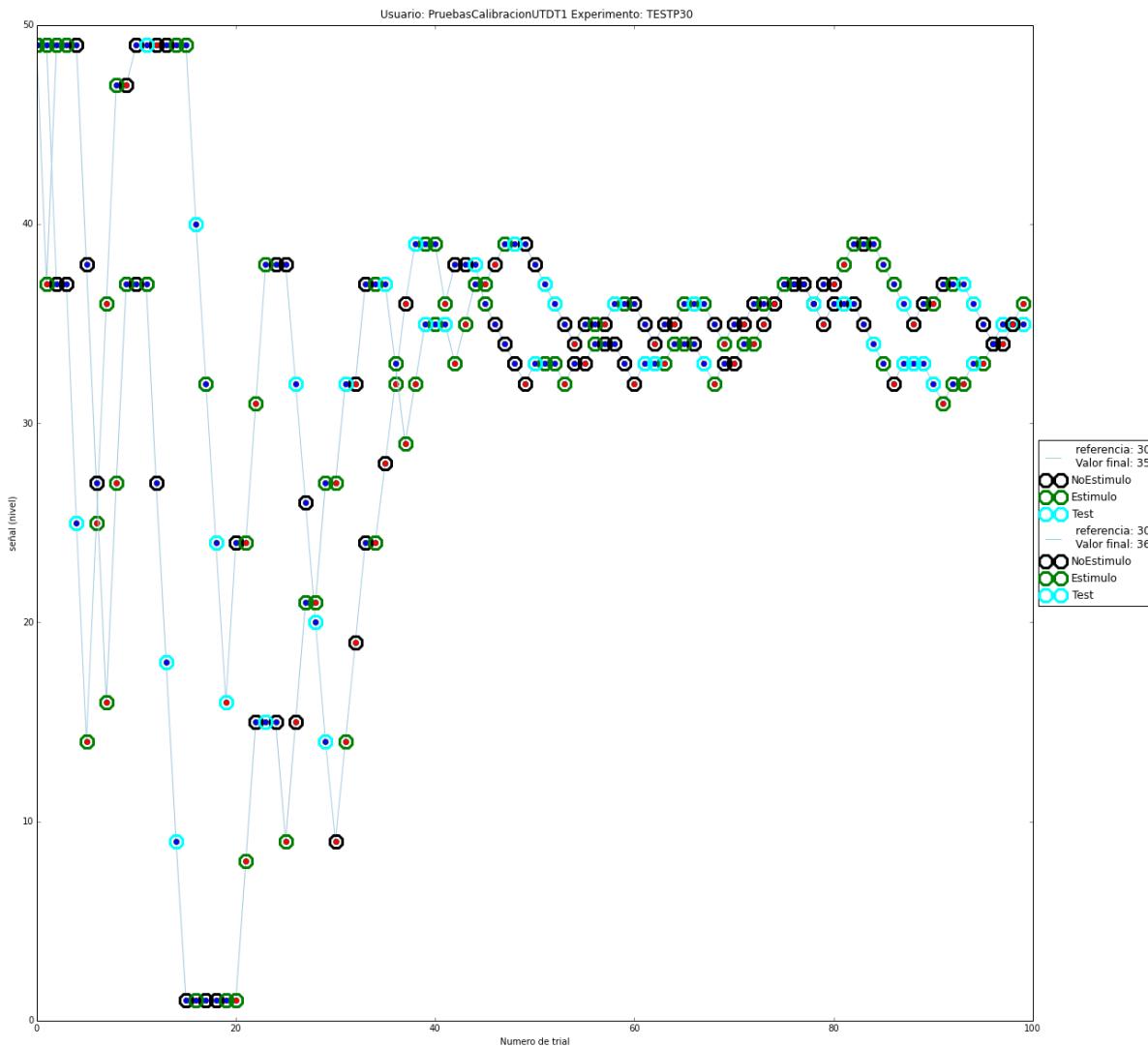


# Primer experimento

- Tuvo dos partes, la primera, la piloto que la hice solo en dos sujetos (uno yo), buscamos ver con cuidado la dependencia con la orientación
- La segunda donde trajimos sujetos al labo y buscamos reconfirmar los resultados anteriores en forma valida para presentar resultados.
- Ademas esta segunda parte sirvio como prueba de concepto de lo que seria el procedimiento para el siguiente experimento.

# StairCase....

## Sin graficos lindos, contar en pizarron



# Primer experimento

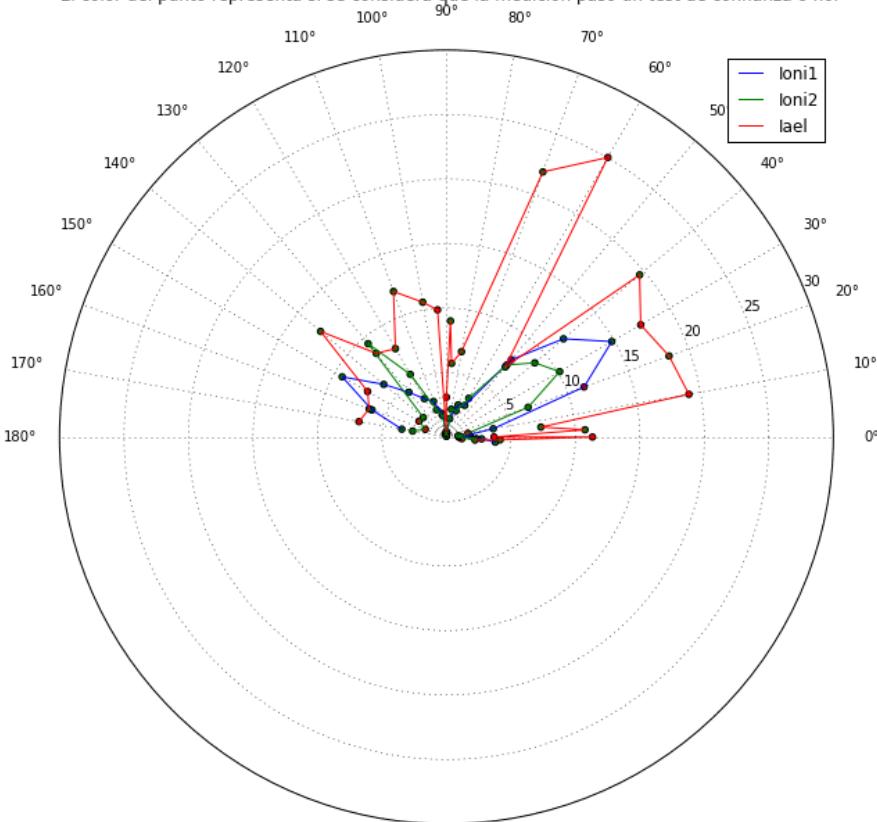
## diseño del experimento

- Hicimos una secuencia super detallada de paralelas orientadas en diferentes angulos, con mas densidad en los ejes.

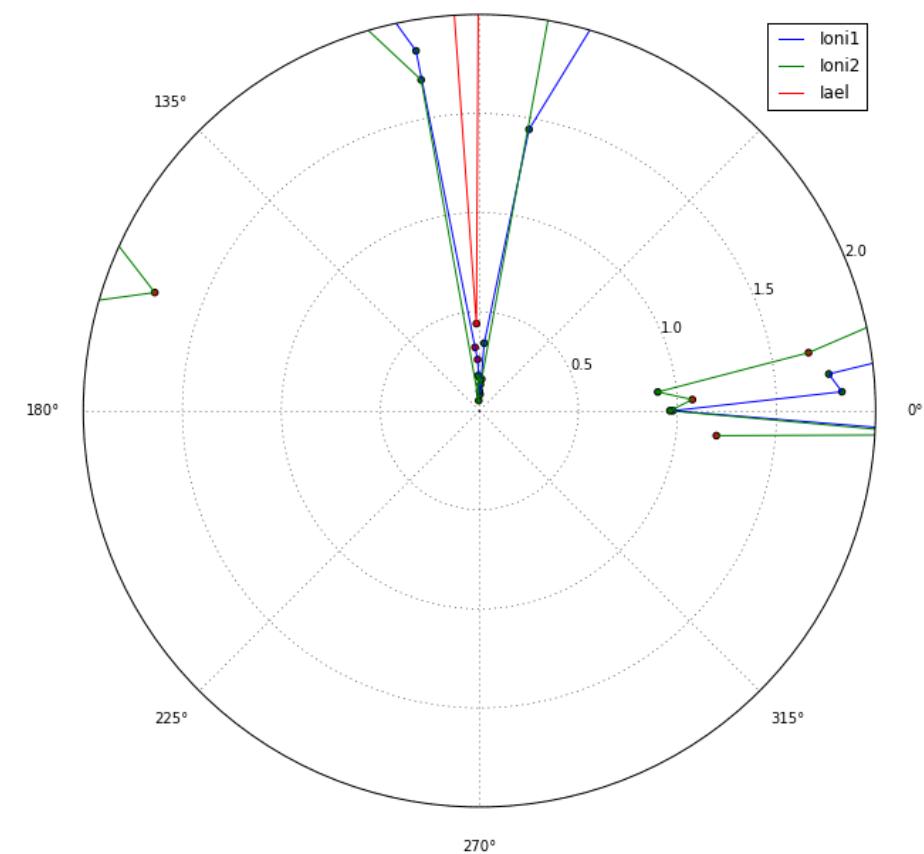
# Primer experimento

## Resultados

Separación angular mínima (en  $^{\circ}$ ) que se puede detectar en función de la orientación de las rectas 'pseudoparalelas'.  
El color del punto representa si se considera que la medición pasó un test de confianza o no.

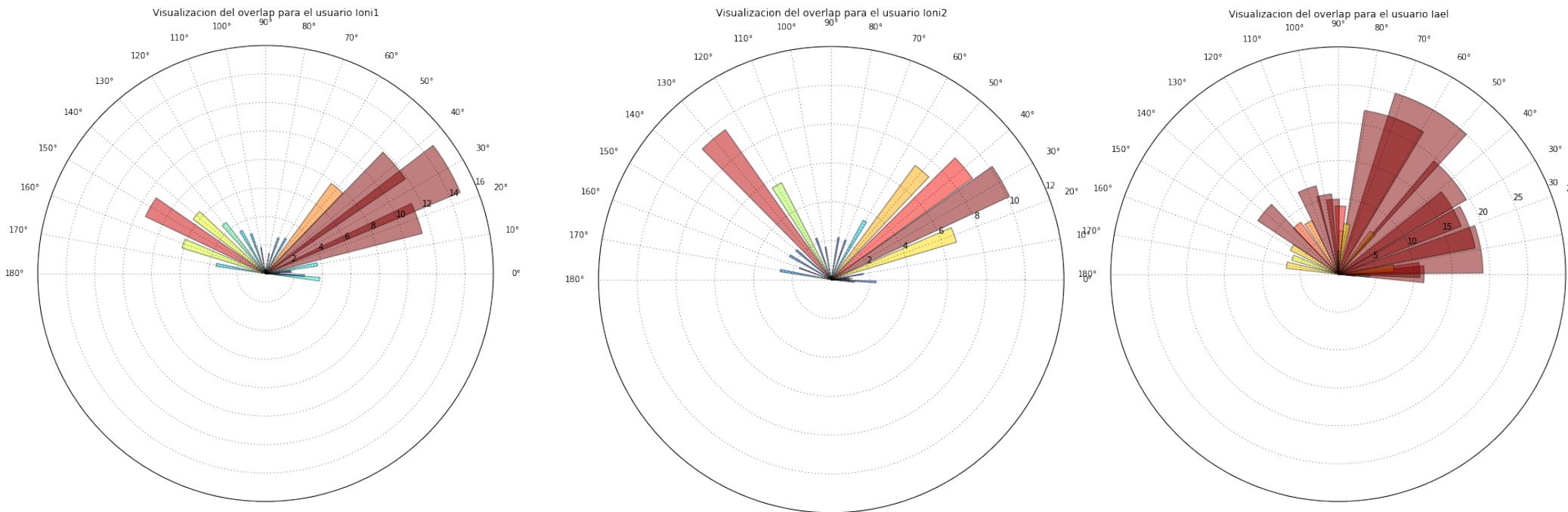


Separación angular mínima (en  $^{\circ}$ ) que se puede detectar en función de la orientación de las rectas 'pseudoparalelas'.  
El color del punto representa si se considera que la medición pasó un test de confianza o no.



# Primer experimento

## Resultados



# Primer experimento

## Diseño experimental parte II

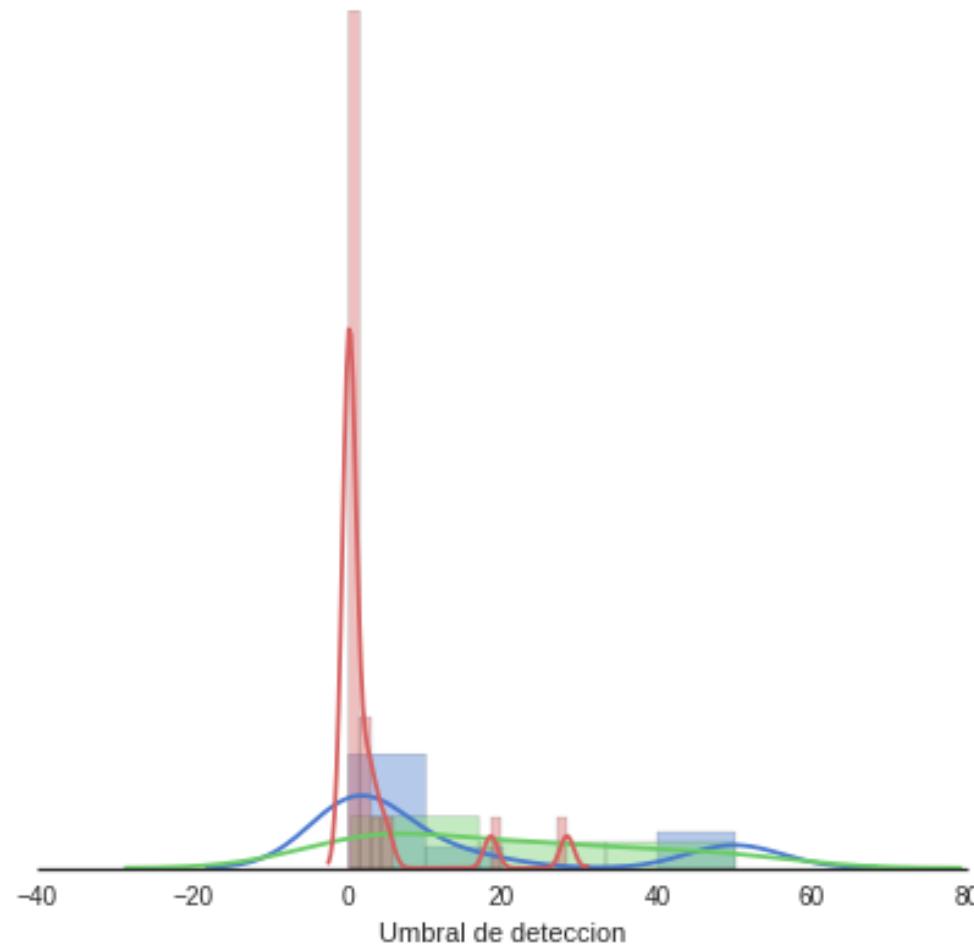
- Probamos con tres orientaciones, en angulos y en paralelismo.
- Las orientaciones eran, 0, 30 y 90º
- Se escuchaban los estímulos y el usuario tenía que responder entre las categorías “agudo”, “recto” “obtuso”, o entre dos segmentos que se juntaban hacia un lado o hacia otro según que nivel fuera

# Primer experimento

## Resultados Parte II

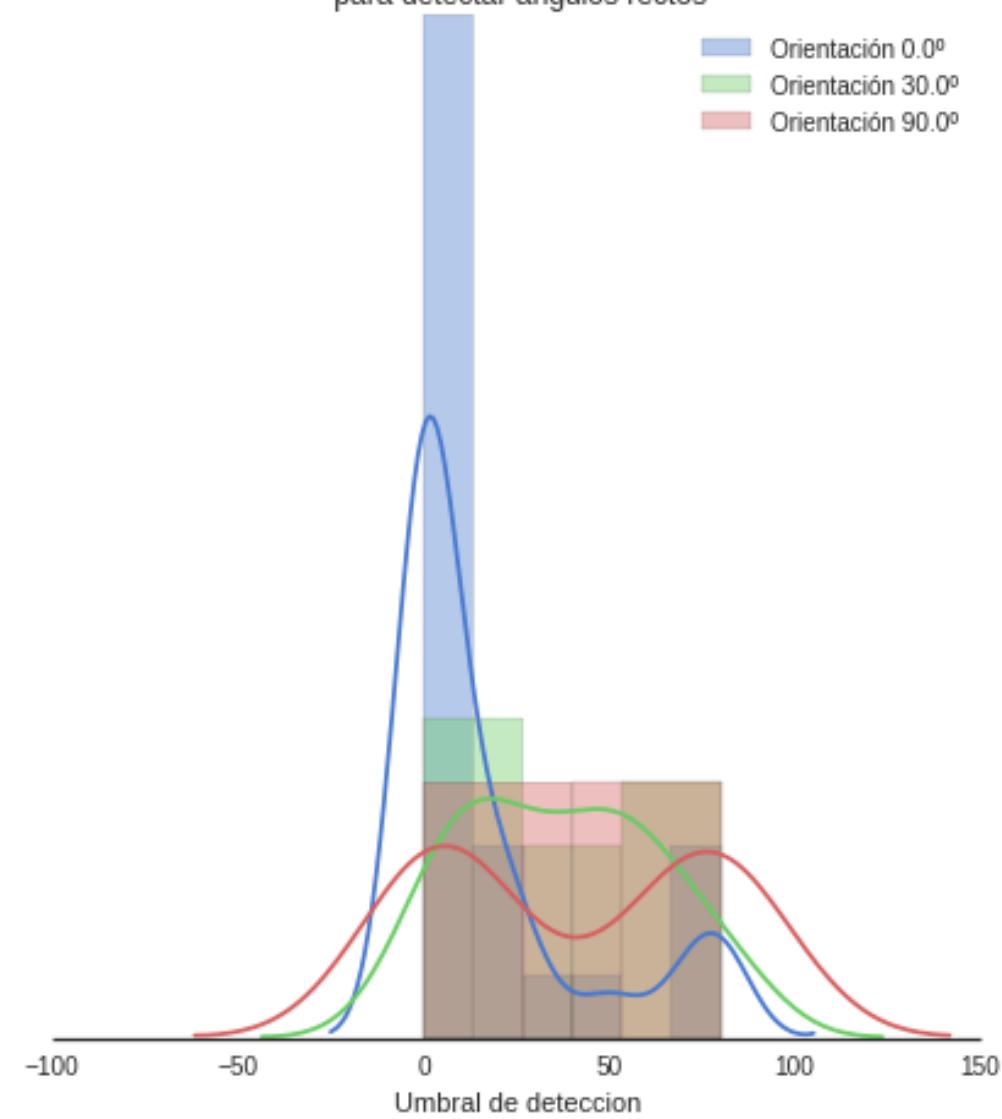
Distribucion de umbrales de deteccion (n=24)  
para detectar paralelismo

Orientación 0.0°  
Orientación 30.0°  
Orientación 90.0°



Distribucion de umbrales de deteccion (n=24)  
para detectar angulos rectos

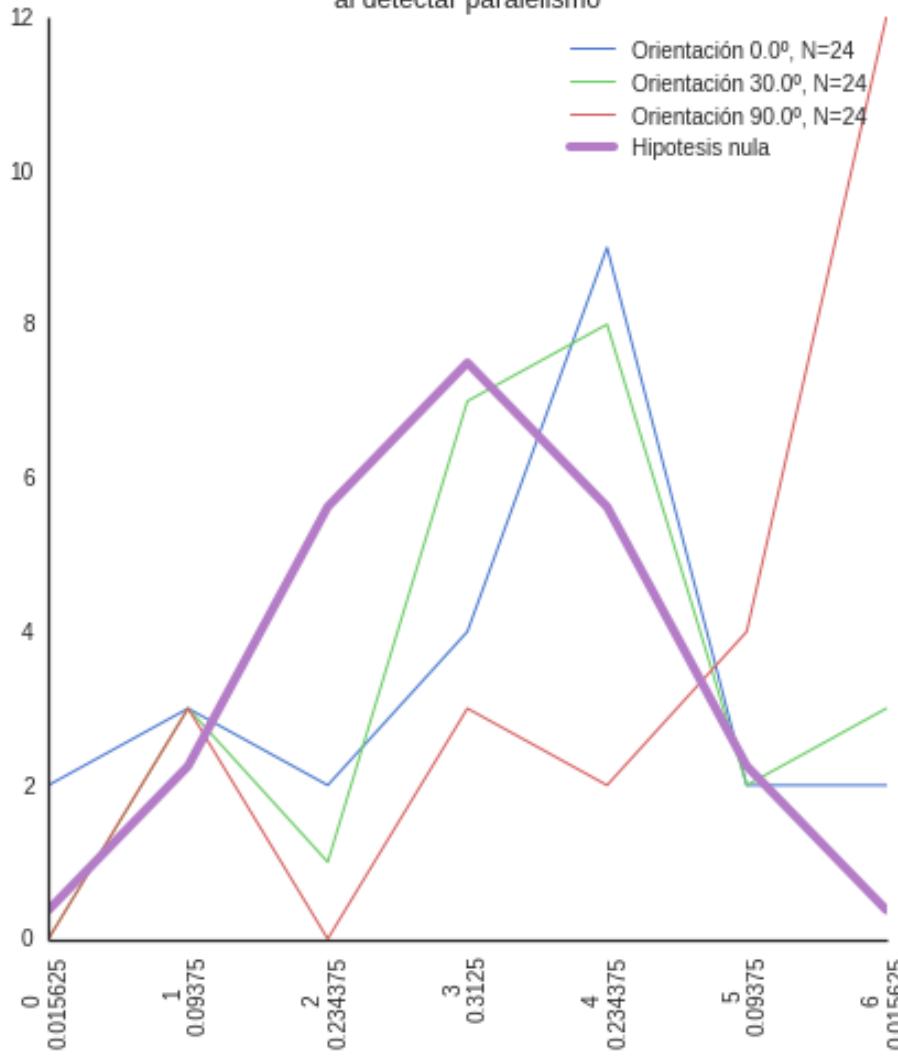
Orientación 0.0°  
Orientación 30.0°  
Orientación 90.0°



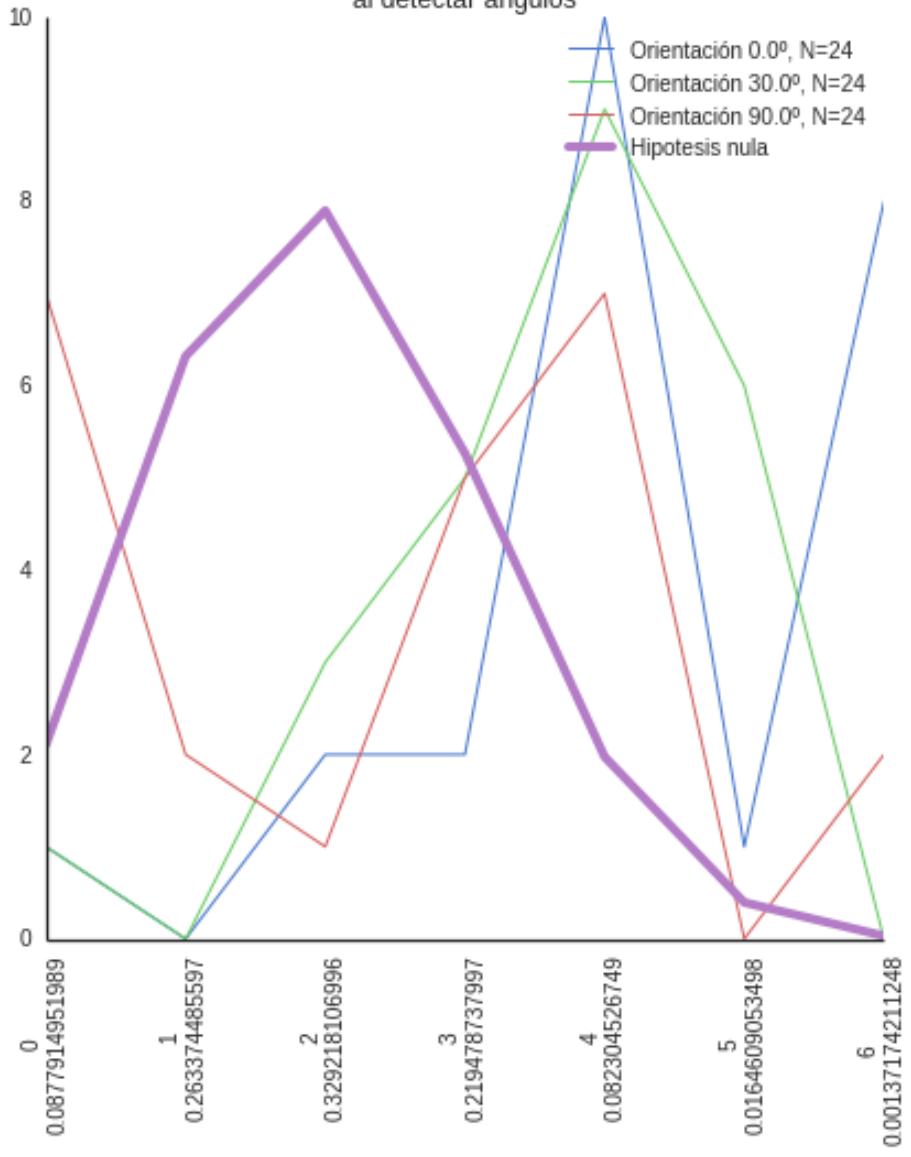
# Primer experimento

## Resultados Parte II

Histograma de aciertos obtenidos en los ultimos 6 trials  
y sus correspondientes pValue  
al detectar paralelismo

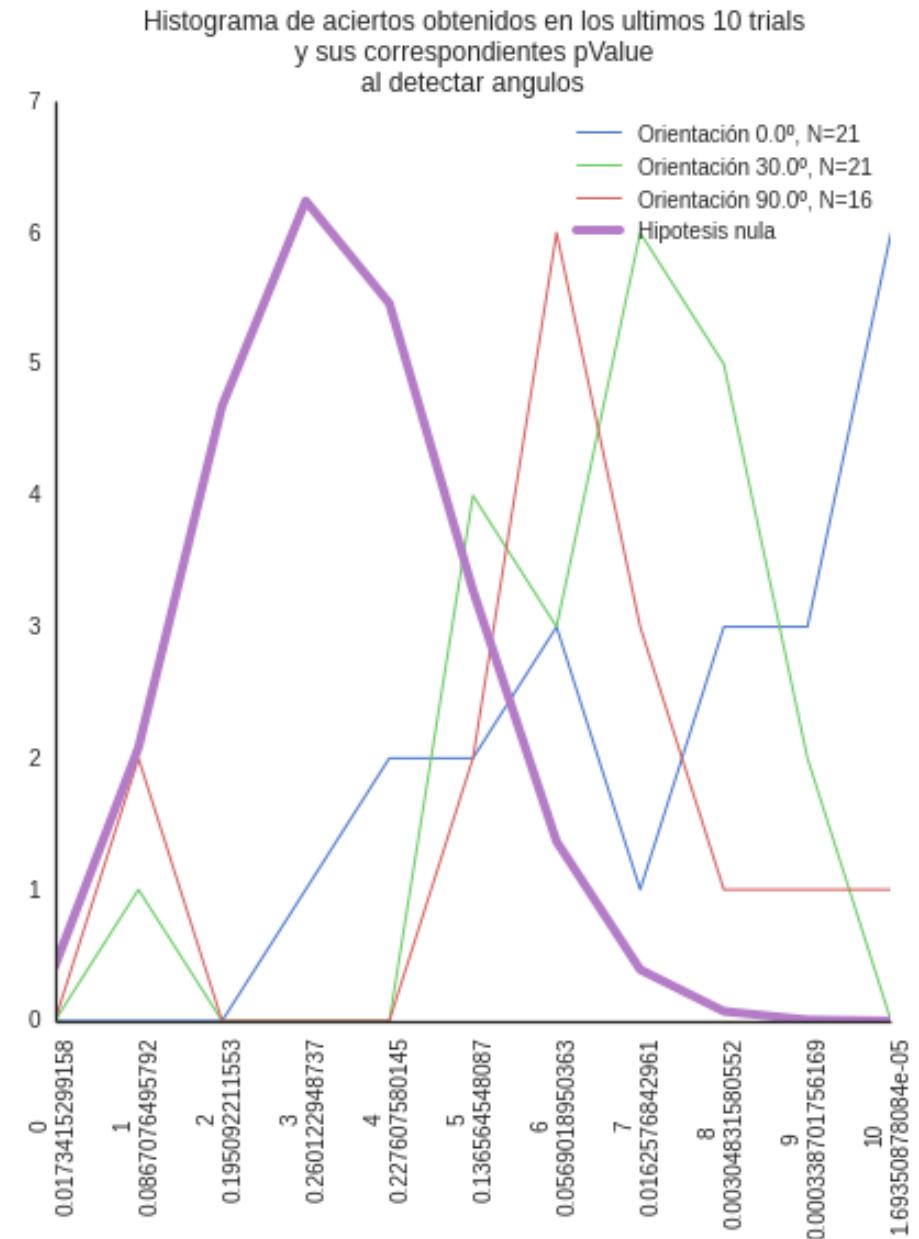
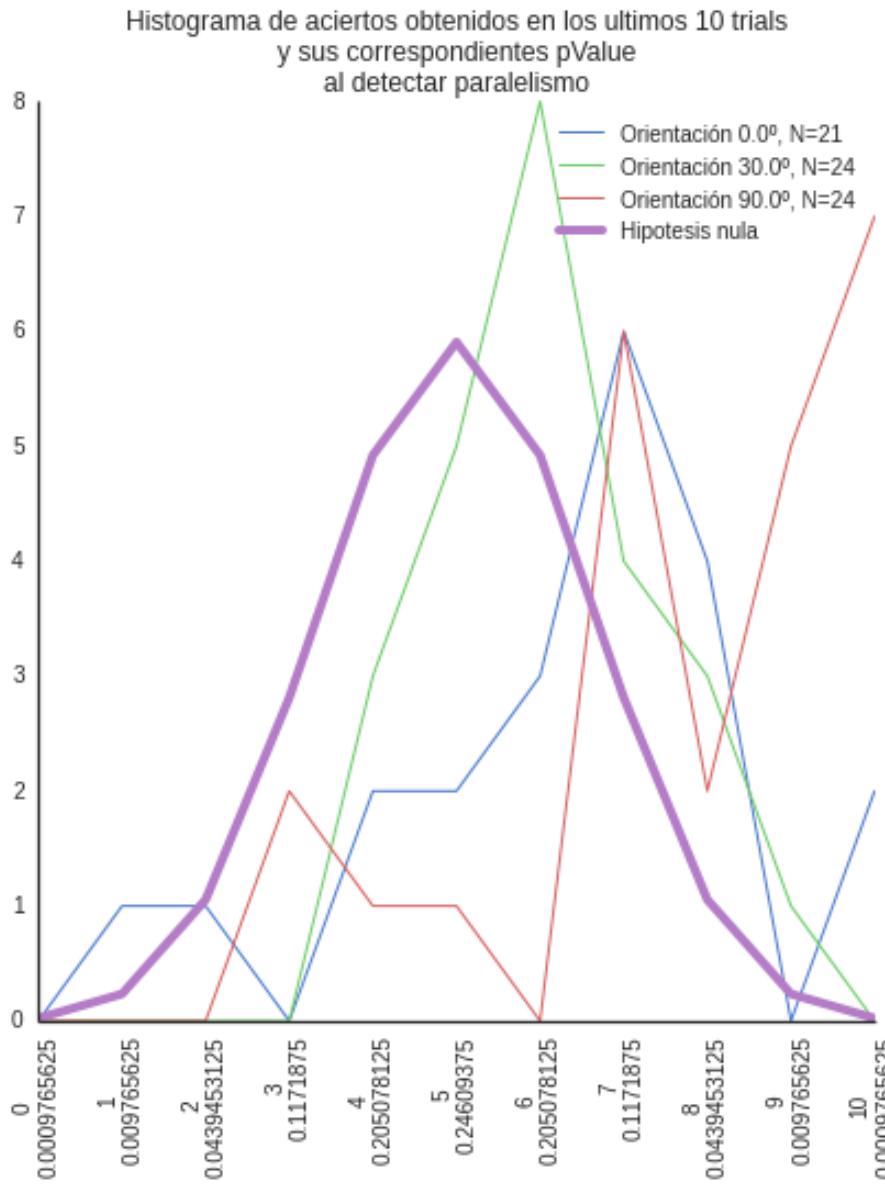


Histograma de aciertos obtenidos en los ultimos 6 trials  
y sus correspondientes pValue  
al detectar angulos



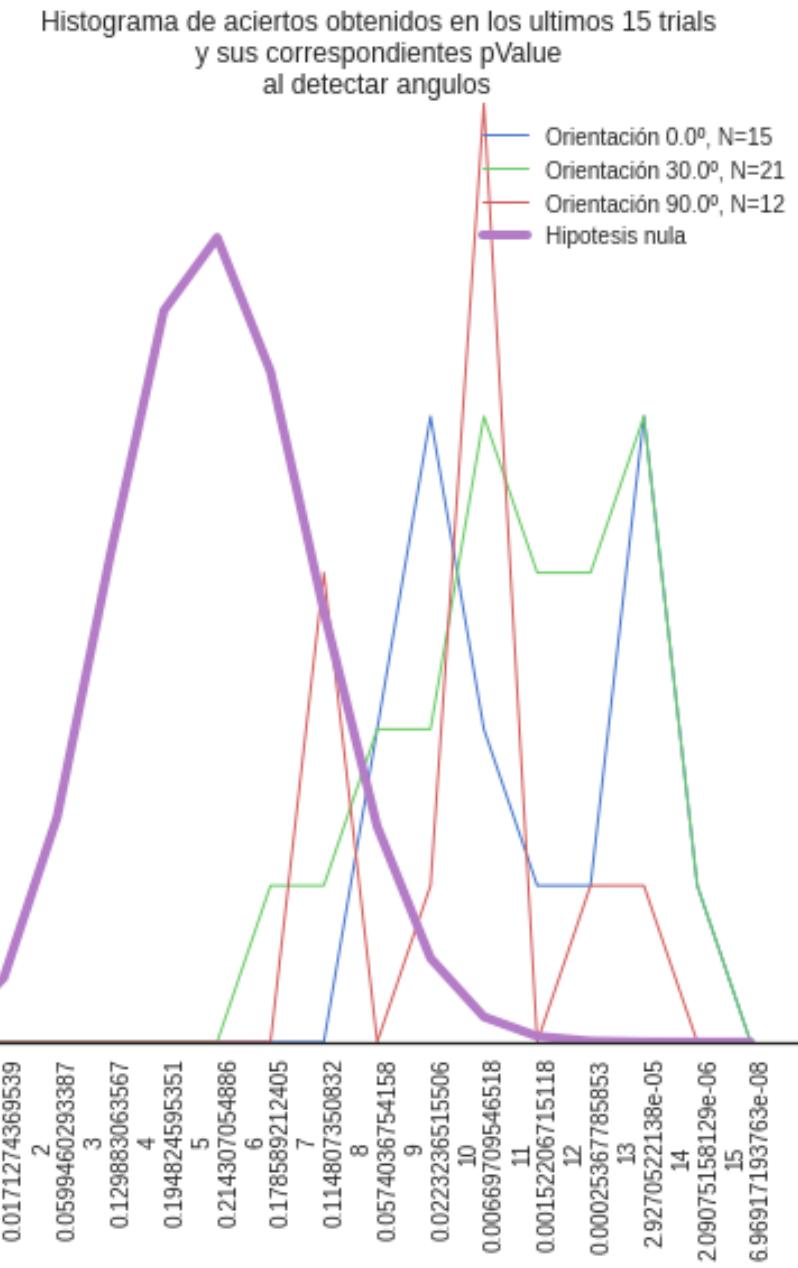
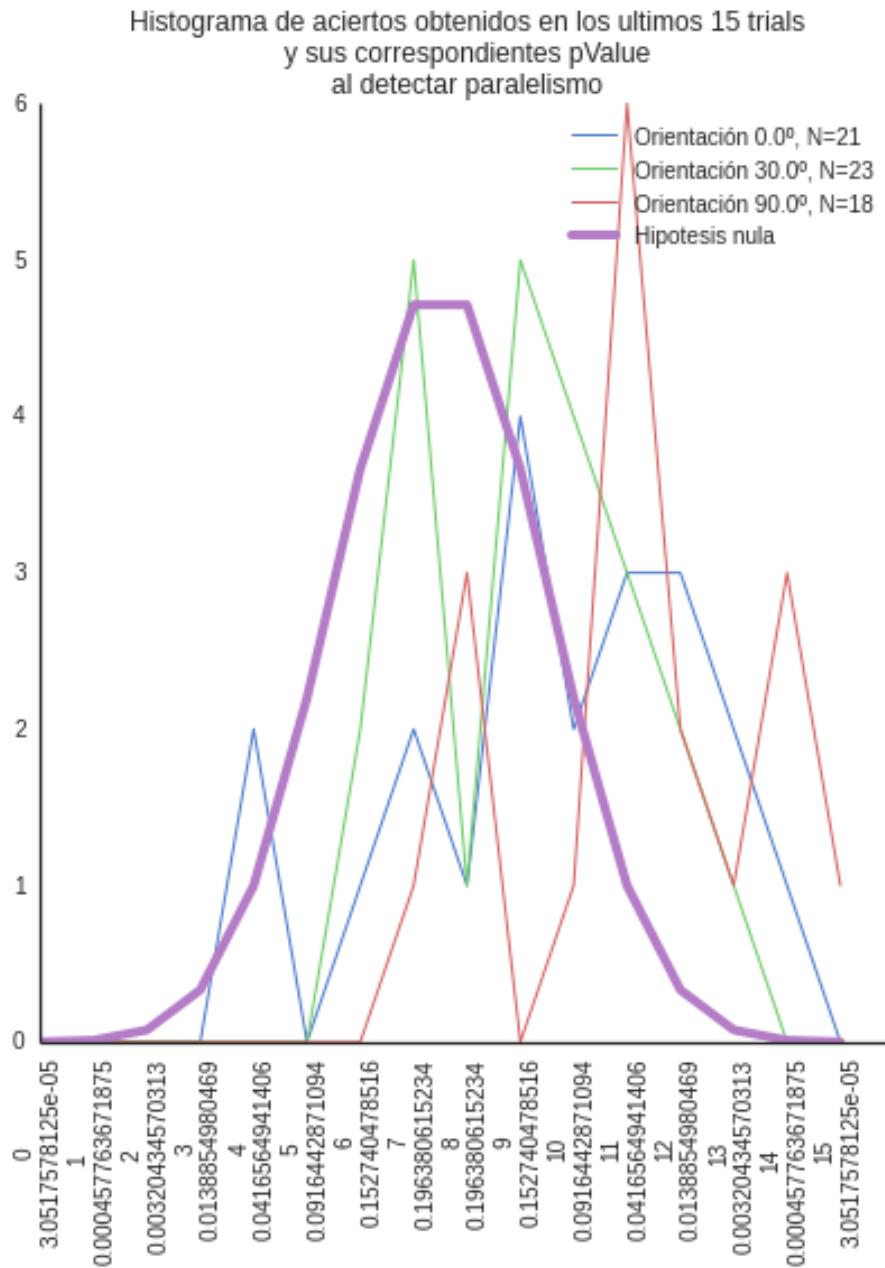
# Primer experimento

## Resultados Parte II



# Primer experimento

## Resultados Parte II

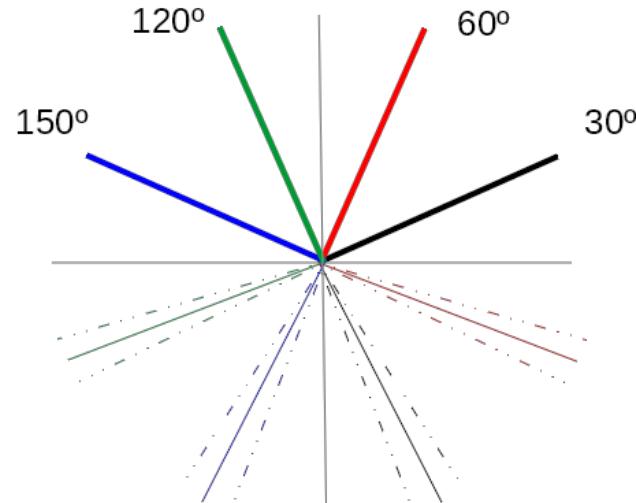


# Segundo experimento

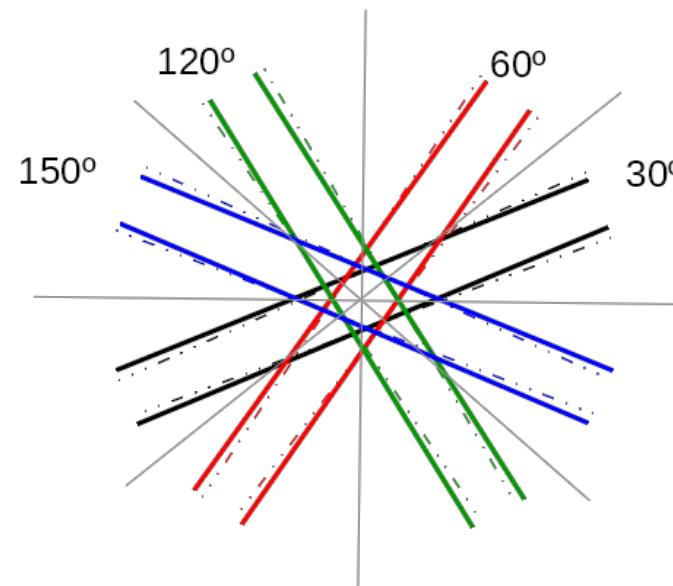
## Diseño conceptual

- Elegimos angulos de 30, 60 , 120 y 150 grados
- Hicimos un test previo
- Entrenamos a algunos sujetos
- **Hicimos un test posterior y comparamos**

— Lado fijo  
— Estimulo neutro  
··· Estimulo señal  
— Ejes Simetria

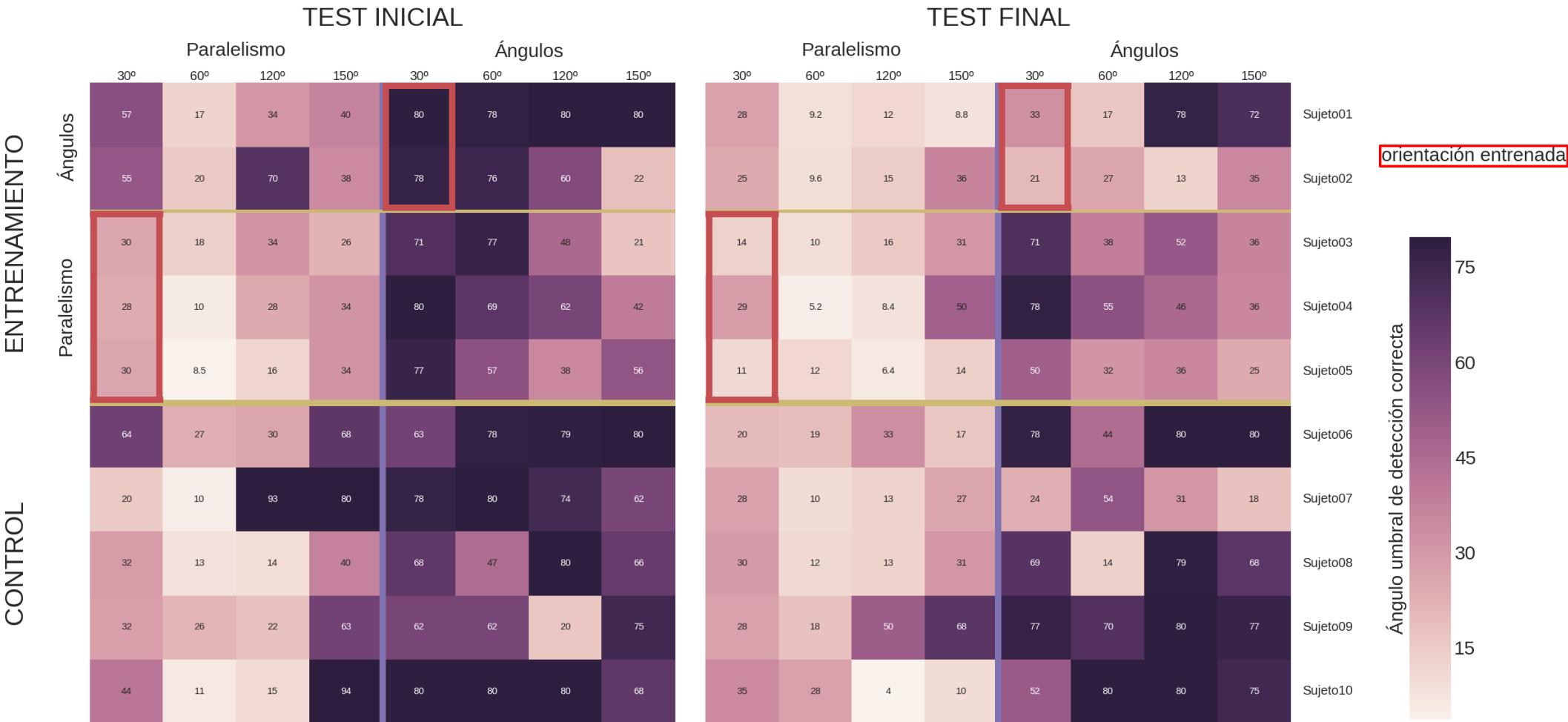


— Estimulo neutro  
··· Estimulo señal  
— Ejes Simetria



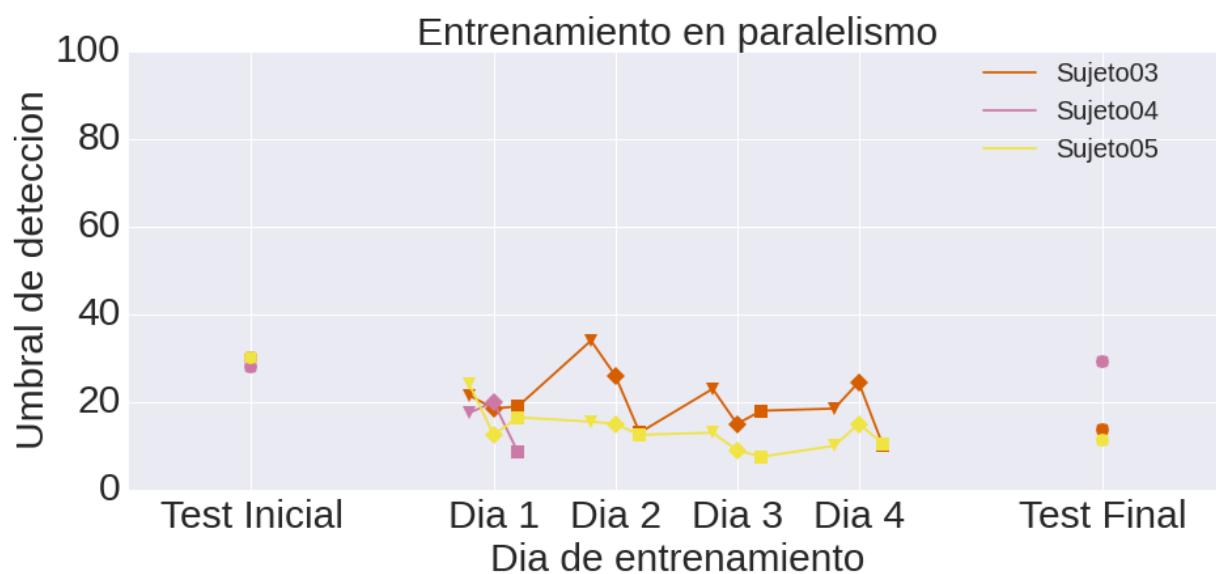
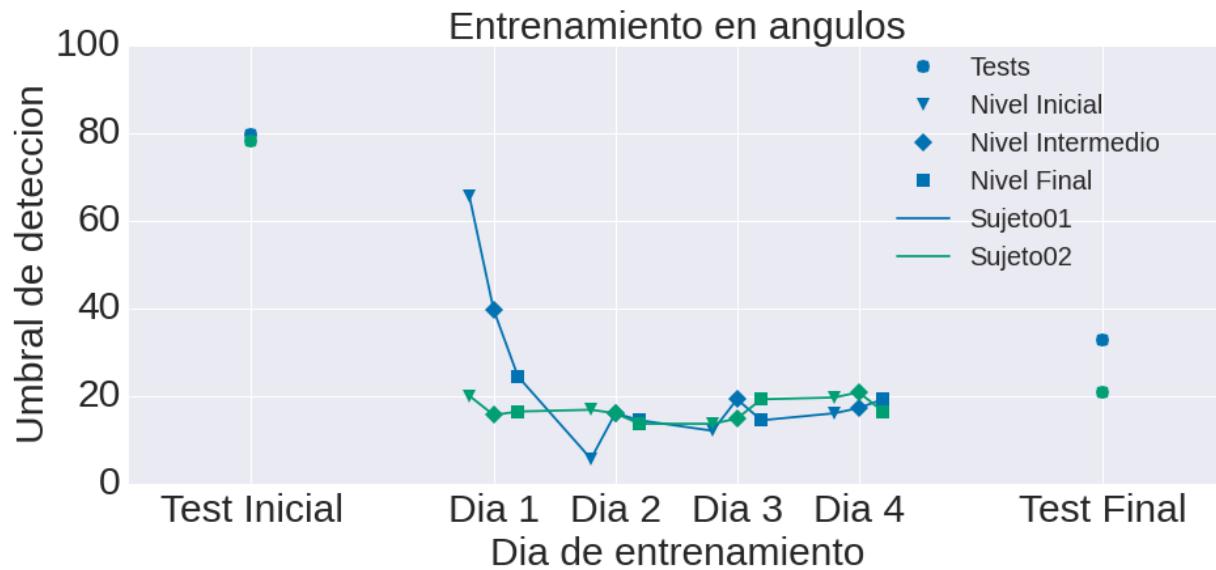
# Segundo experimento

## Resultados



# Segundo experimento

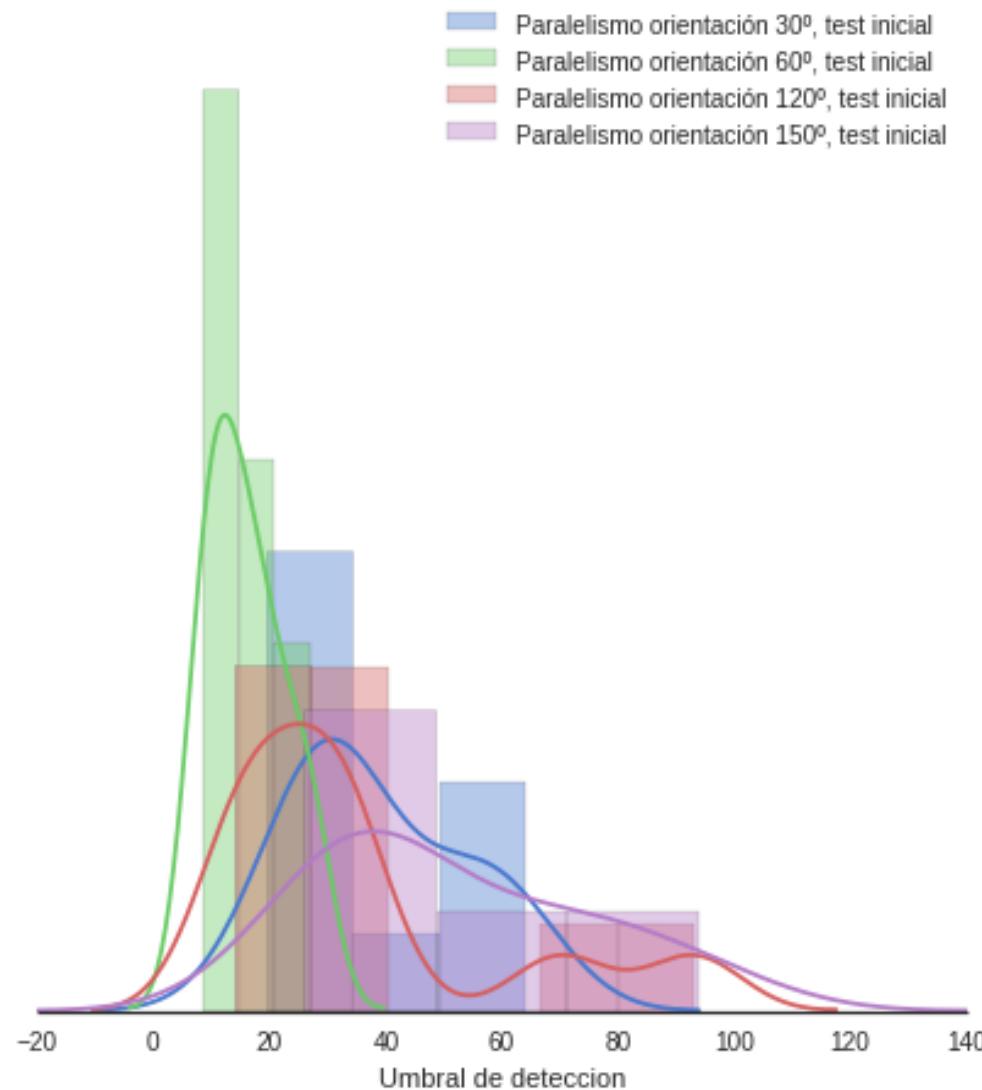
## Resultados



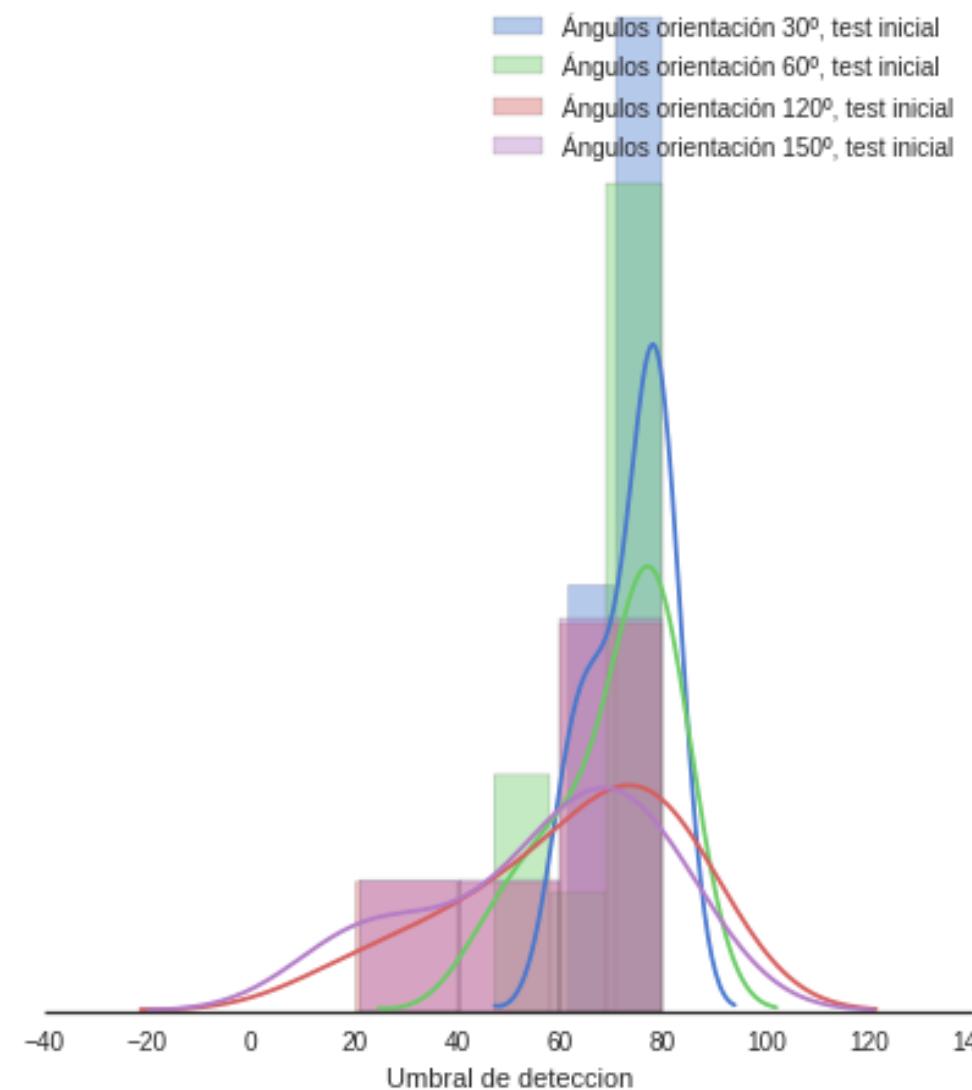
# Segundo experimento

## Resultados

Distribucion de umbrales de detección (n=10)



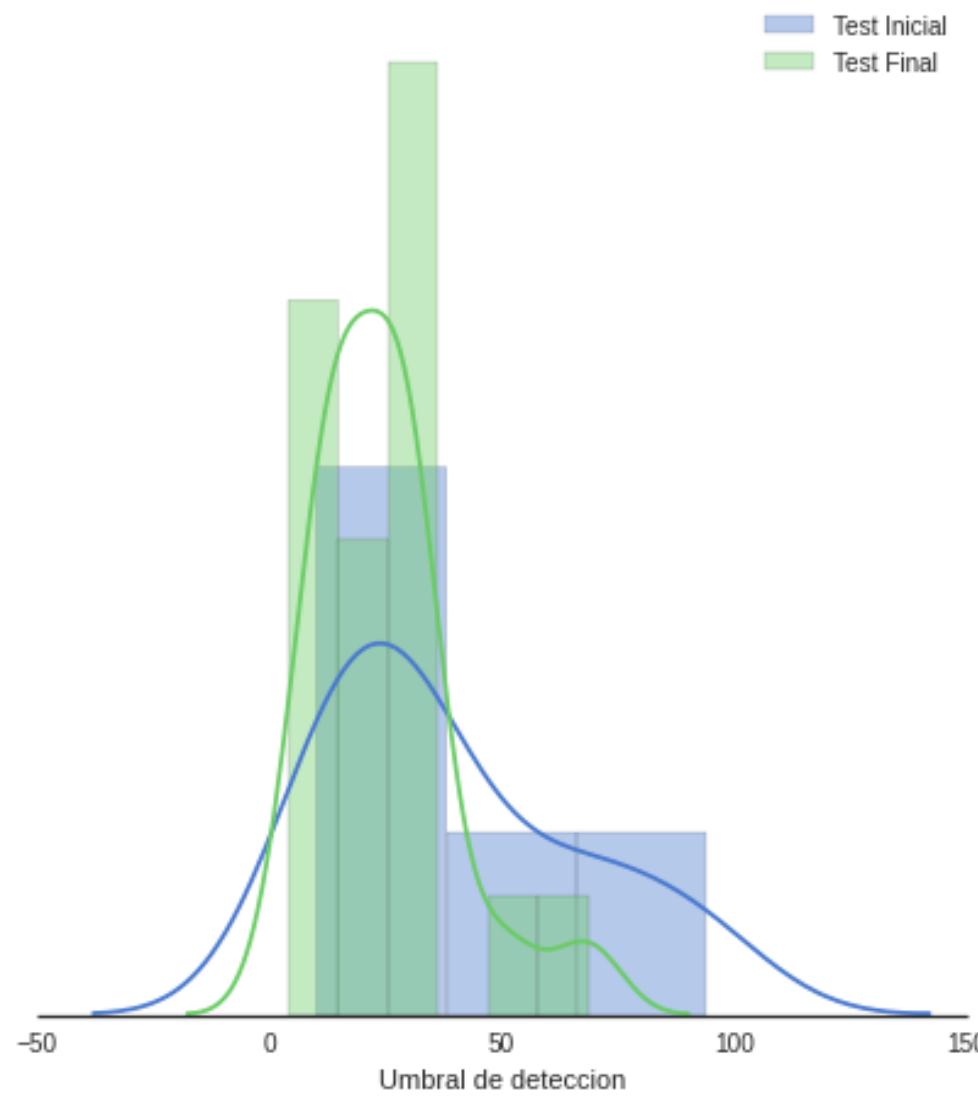
Distribucion de umbrales de detección (n=10)



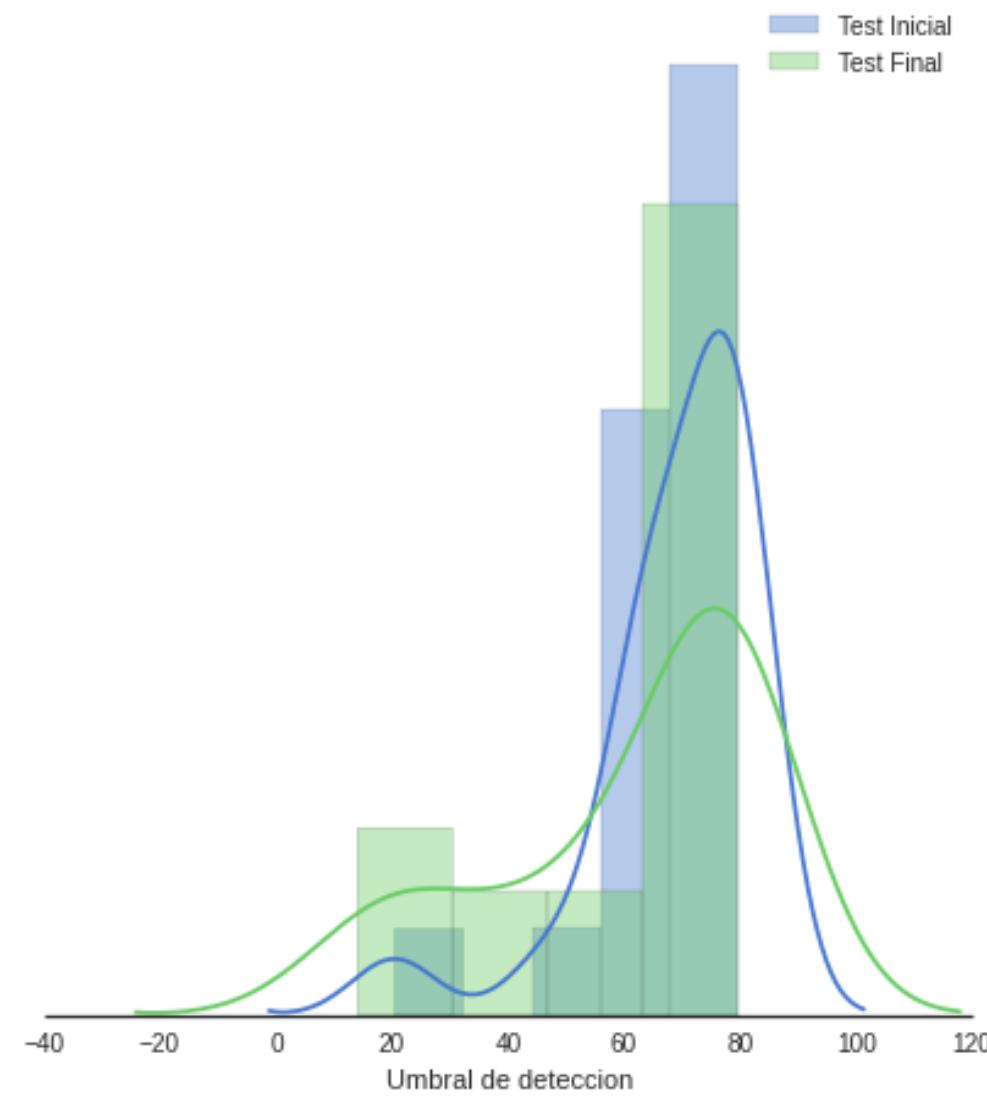
# Segundo experimento

## Resultados

Distribucion de umbrales de deteccion  
test de paralelismo en sujetos control (n=5)



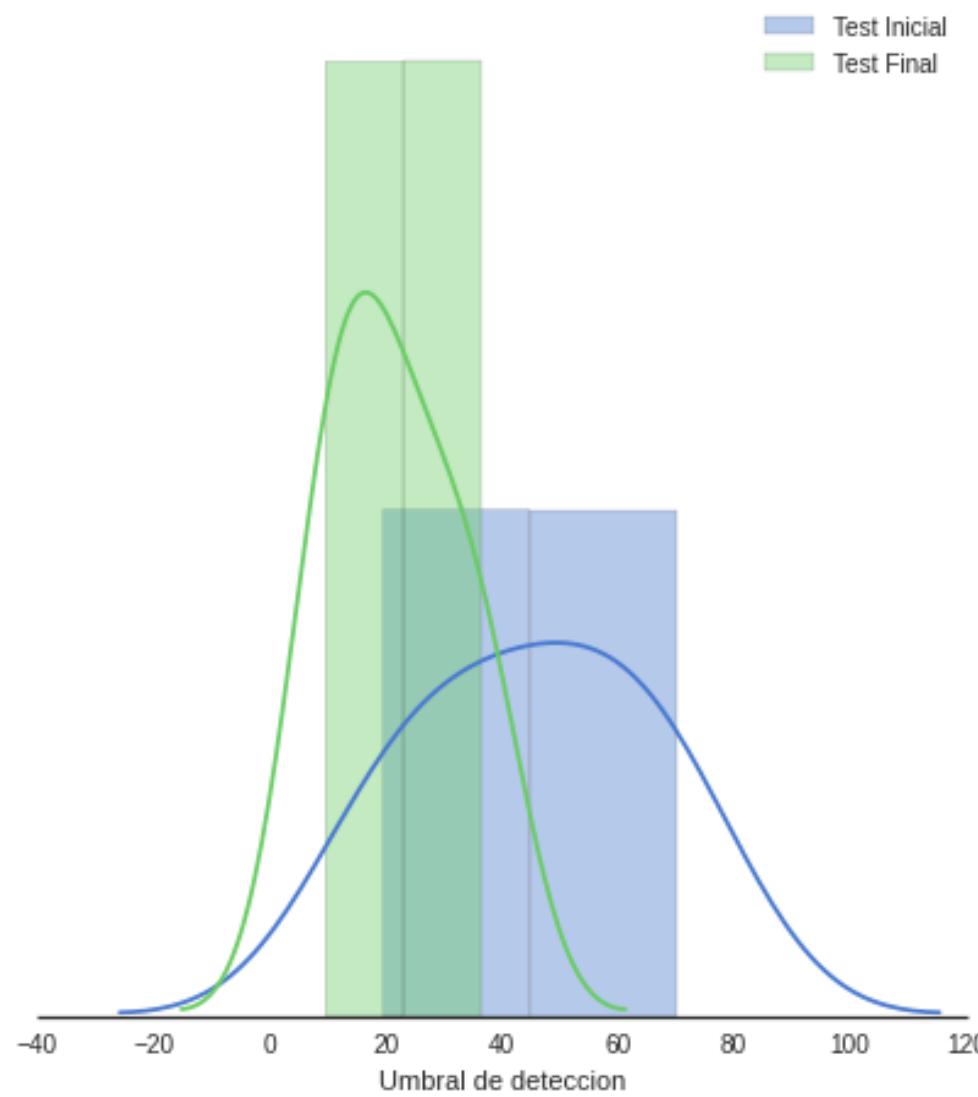
Distribucion de umbrales de deteccion  
test de ángulos en sujetos control (n=5)



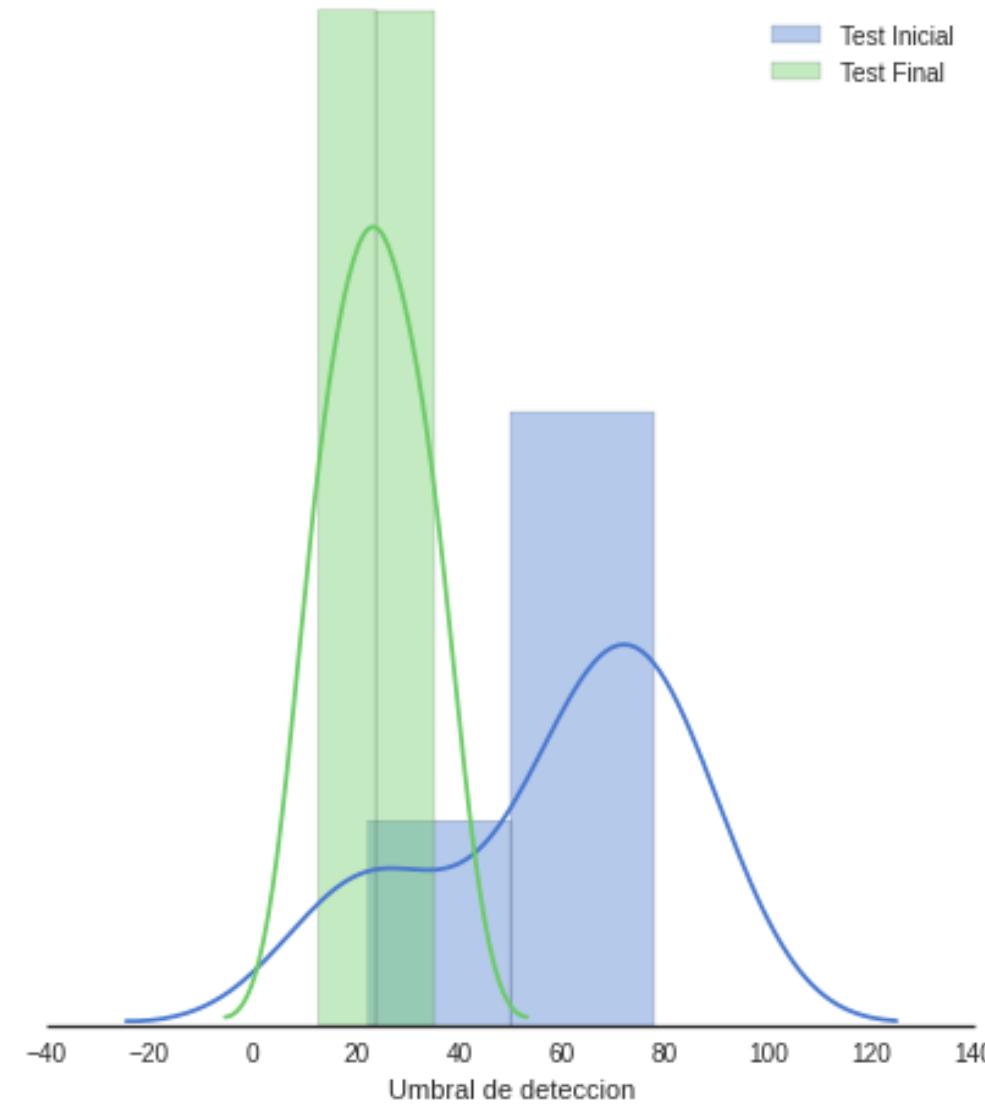
# Segundo experimento

## Resultados

Distribucion de umbrales de deteccion  
test de paralelismo en sujetos entrenados en angulos (n=2)



Distribucion de umbrales de deteccion  
test de ángulos en sujetos entrenados en angulos (n=2)

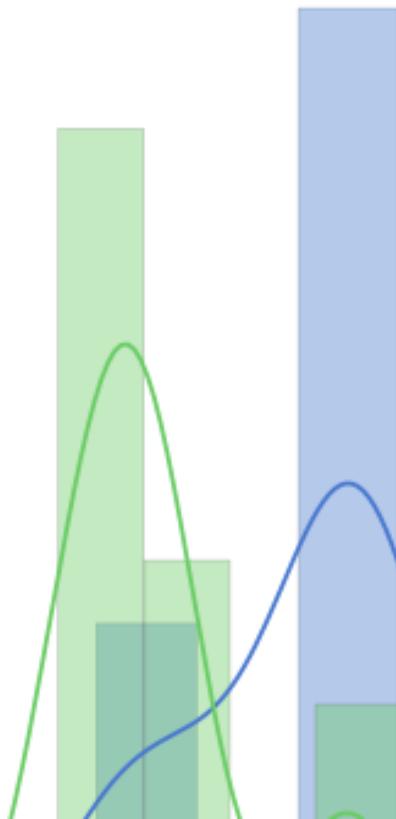


# Segundo experimento

## Resultados

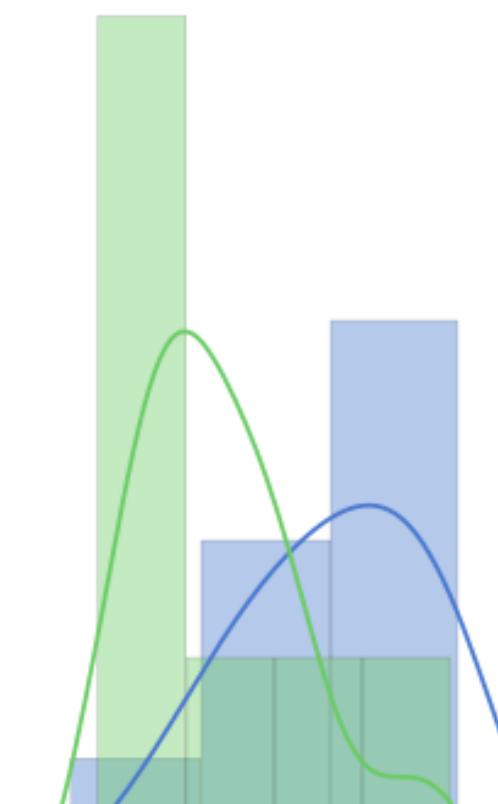
Distribucion de umbrales de deteccion  
test de paralelismo en sujetos entrenados en paralelismo (n=3)

Test Inicial  
Test Final



Distribucion de umbrales de deteccion  
test de ángulos en sujetos entrenados en paralelismo (n=3)

Test Inicial  
Test Final



# Segundo experimento

## Resultados

Tabla 2: Resultados obtenidos para los ajustes de la pendiente y el efecto feedback considerando diferentes conjuntos de datos. Se observa que el único caso donde se encontró un efecto significativo es en la mejora global del entrenamiento unificando todos los datos correspondientes las categorías de entrenamiento diferente y en la mejora intrasesión para el caso de paralelismo.

\* El usuario 1 presentó un comportamiento marcadamente diferente durante el día 1, por lo que se analizó los datos excluyendo estos datos

\*\* Datos excluyendo día 1 del sujeto 1

Criterio utilizado	$\beta$	$\alpha$	std err	t	P >  t
<b>Efecto del entrenamiento</b>					
Todos los datos medidos	-0.8219	0.346	-2.374	0.022	
Excluyendo usuario 1*	-0.1361	0.244	-0.558	0.580	
Excluyendo día 1	-0.1292	0.353	-0.365	0.717	
Excluyendo día 1 usuario 1*	-0.2385	0.213	-1.117	0.270	
Angulos (todos los datos)	-1.1916	0.646	-1.844	0.079	
Angulos sin día 1 sujeto 1*	0.3126	0.225	1.389	0.181	
Paralelismo	-0.5731	0.322	-1.779	0.087	
<b>Efecto del entrenamiento dentro de cada sesión</b>					
Angulos**	0.7143	0.910	0.785	0.442	
Paralelismo	-3.4167	1.324	-2.580	<b>0.016</b>	
<b>Efecto de la ausencia de feedback</b>					
Angulos**	1.5714	1.577	0.997	0.332	
Paralelismo	1.0278	2.332	0.441	0.663	
<b>Efecto de los tests iniciales y finales</b>					
Paralelismo $\alpha_i$	9.3021	4.210	2.209	<b>0.035</b>	
Paralelismo $\alpha_f$	5.4191	4.375	1.239	0.225	
Paralelismo $\alpha_{cte}$	20.0312	2.303	8.698	0.000	
Angulos** $\alpha_i$	64.9951	3.235	20.092	<b>0.000</b>	
Angulos** $\alpha_f$	8.9307	3.070	2.909	<b>0.008</b>	
Angulos** $\alpha_{cte}$	13.8049	1.950	7.079	0.000	

# Segundo experimento

## Resultados

Tabla 3: Estudios estadísticos realizados para comparar mediciones en busca de efectos de dificultad, aprendizaje o transferencia.

Comparacion realizada	estadistico	p-value	significativo	comentario
<b>Comparaciones generales de las orientaciones usando el test inicial</b>				
ttest angulos vs paralelismo	-6.74	0.000	SI	
anova angulos vs orientacion	5.999	0.00201	SI	Detecta una diferencia en P60
anova angulos sin P60 vs orientacion	1.504	0.240	NO	
anova paralelismo vs orientacion	2.087	0.119	NO	
<b>Comparaciones entre sujetos entrenados y no entrenados previo al entrenamiento</b>				
anova paralelismo vs control vs angulos	2.389	0.098	NO	Son equivalentes
<b>Analisis del efecto del entrenamiento en P30</b>				
<b>Comparación en la performance en la orientación P30 post entrenamiento</b>				
anova: control, angulos y paralelismo	2.244	0.176	NO	El entrenamiento no hace efecto
anova: control, angulos y paralelismo***	9.464	0.013	SI	Excluyendo al sujeto 4***.
ttest: paralelismo*** y control	-3.979	0.010	SI	
ttest: angulos y control	-0.457	0.666	NO	
ttest: angulos y paralelismo***	7.484	0.017	SI	
<b>Comparación en la performance en otras orientaciones de paralelismo post entrenamiento</b>				
anova: control, angulos y paralelismo***	1.150	0.333	NO	No parece haber transferencia
<b>Analisis del efecto del entrenamiento en A30</b>				
<b>Comparación en la performance en la orientación A30</b>				
anova: control, angulos y paralelismo	2.81	0.126	NO	No parece haber efecto
<b>Analisis del efecto del control</b>				
ttest pareado inicial vs final	2.512	0.016	SI	Se observa mejora en el control
ttest sin parear inicial vs final	1.913	0.059	NO	solo al parear las mediciones.