|  |
| --- |
| Universidad Politécnica de Madrid |
| Detección y resolución de un sudoku mediante realidad aumentada |
|  |

|  |
| --- |
| Carlos Morillo Lozano – M15329  22-6-2016 |

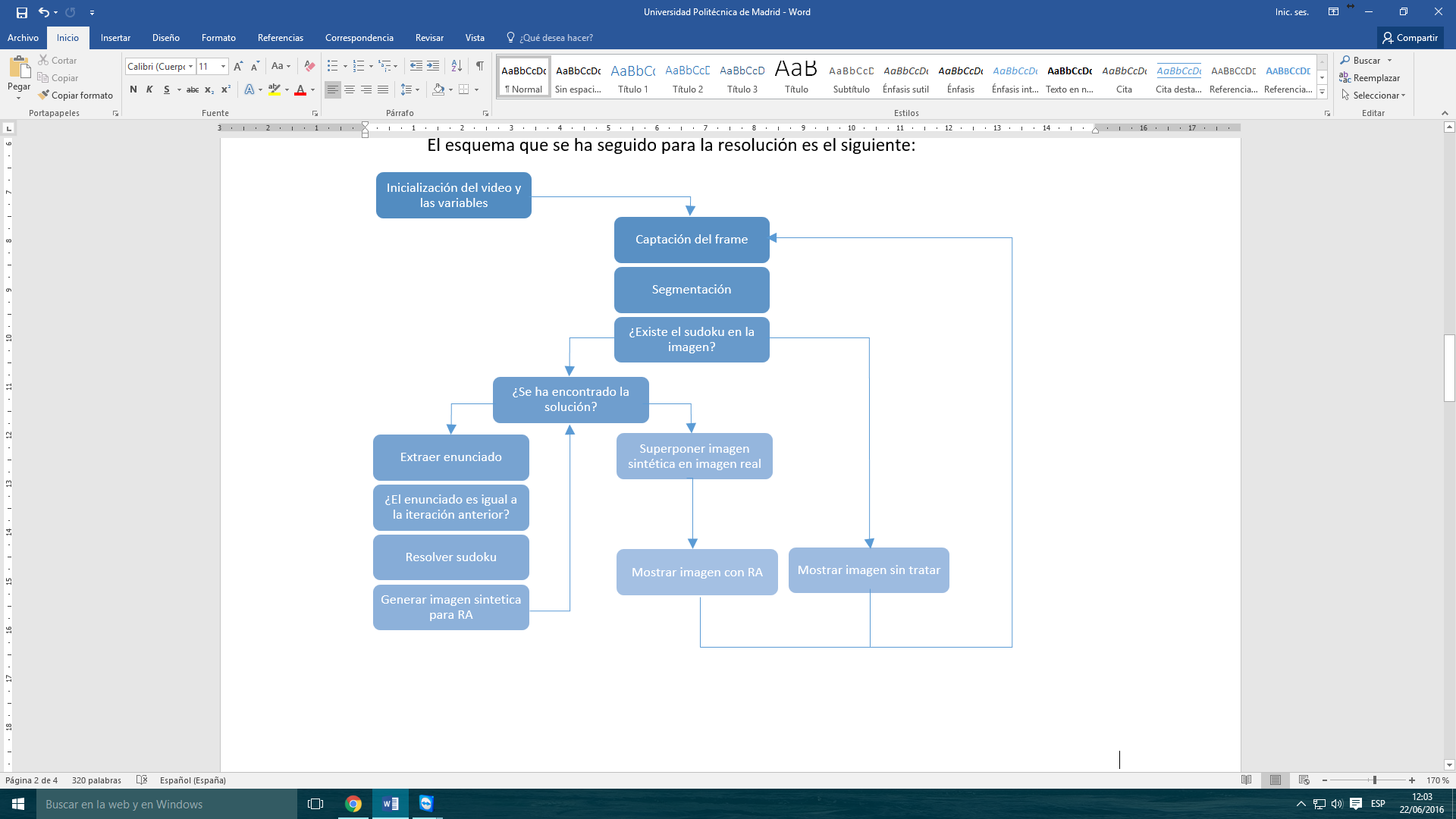
# Introducción

El trabajo consiste en la elaboración de un algoritmo capaz de detectar en tiempo real el enunciado de un sudoku, analizar su contenido y tras resolverlo, mostrar en pantalla sobre el sudoku real la imagen de su resolución en Realidad Aumentada.

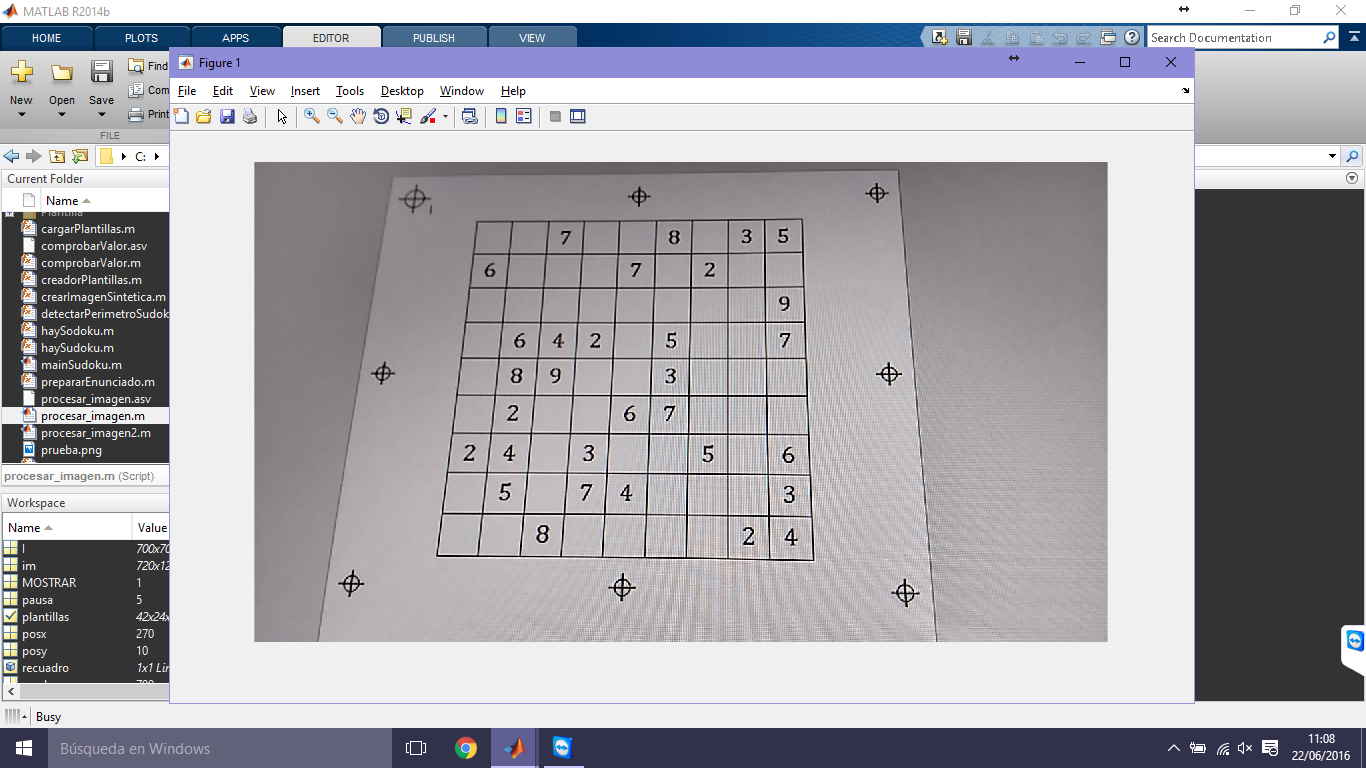
Para la resolución del problema habrá que dividir el trabajo en diferentes partes como: segmentación, para la detección del sudoku; relaciones de transformación, para conocer la orientación del pasatiempo; comparación con las plantillas, para la identificación de los números; resolución del sudoku y superposición de la solución.

# Resolución del problema

El esquema que se ha seguido para la resolución es el siguiente:

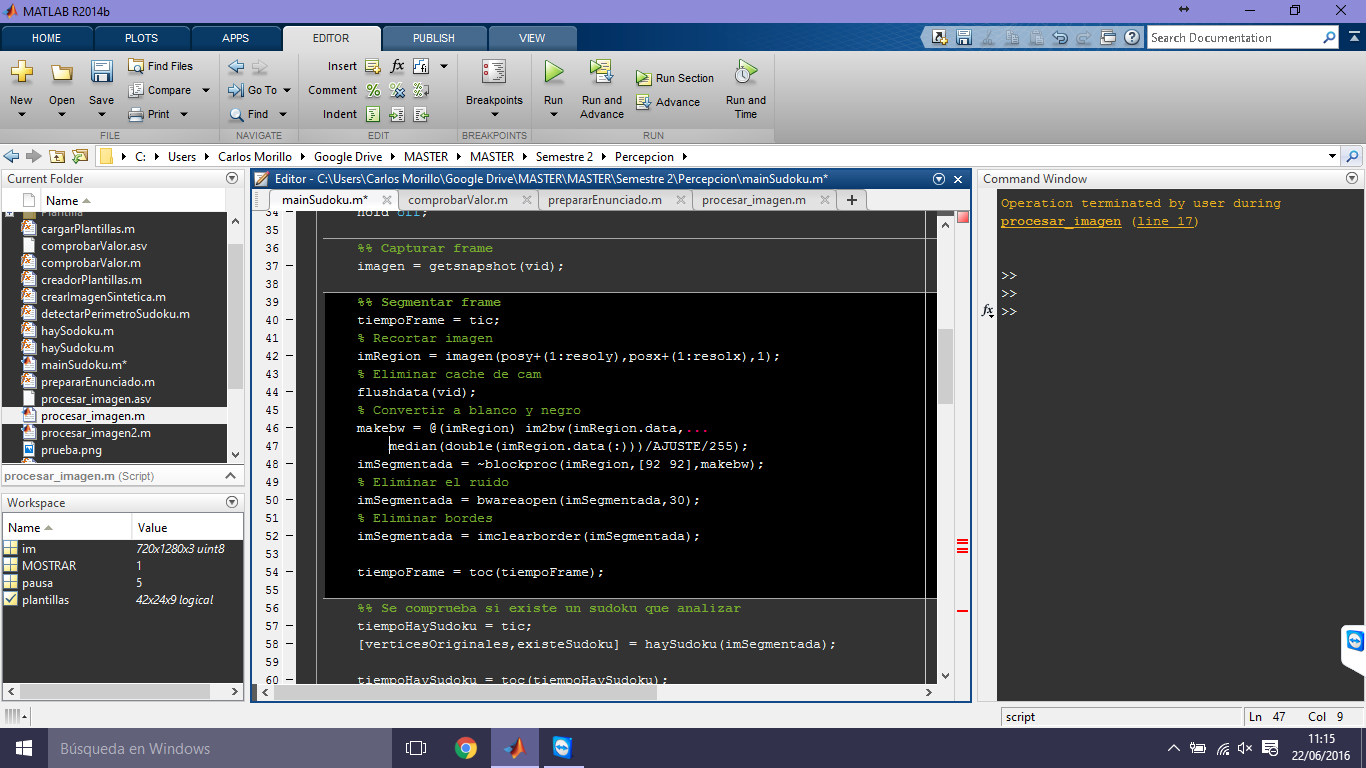


Para poder detallar los pasos realizados con mayor exactitud, se mostrará cada paso aplicando el algoritmo a una imagen de muestra como la que se muestra a continuación:



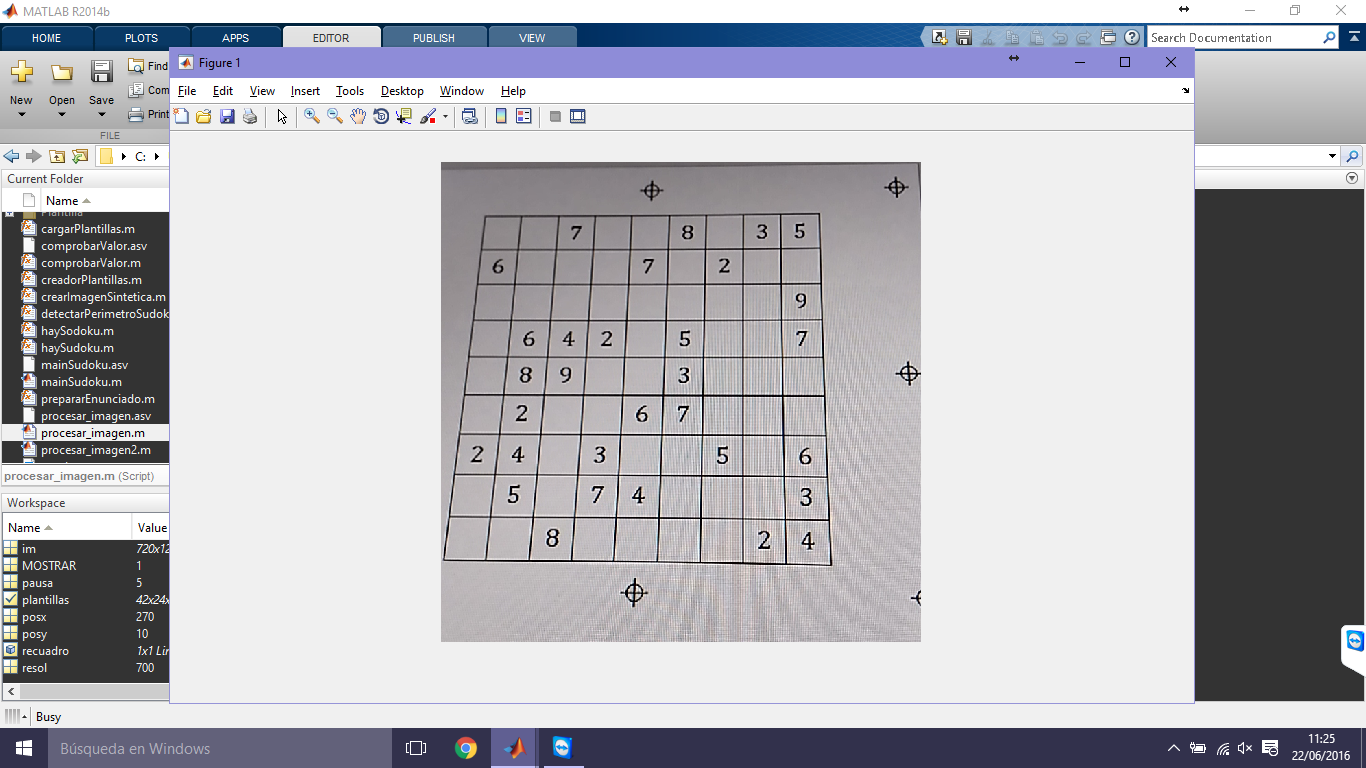
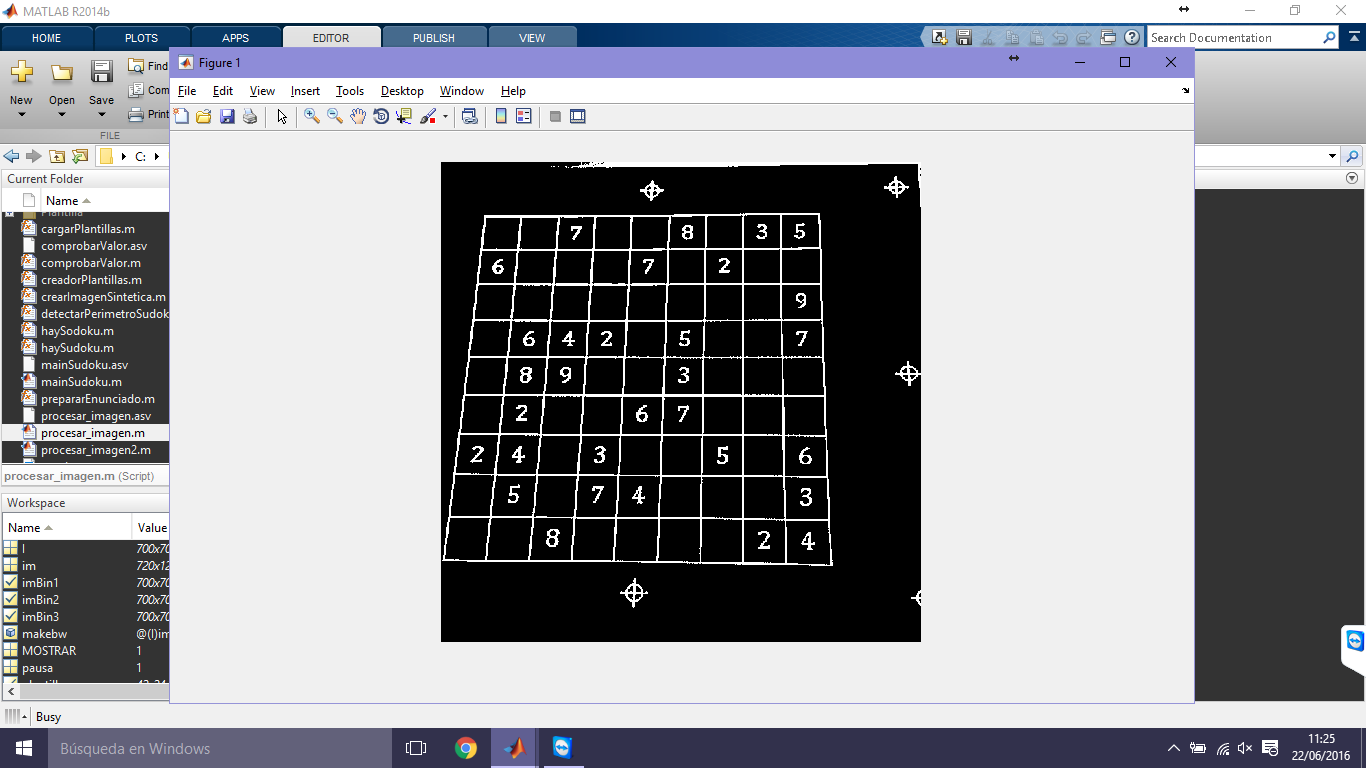
## Segmentación inicial

La segmentación consta de varios pasos, en primer lugar, se recorta el área central de la imagen, eliminando los bordes y permitiendo tener una vista centrada del sudoku. A continuación, dicha imagen se convertirá a blanco y negro, después se filtrará el ruido y finalmente se eliminarán los objetos que estén en contacto con el borde de la imagen.



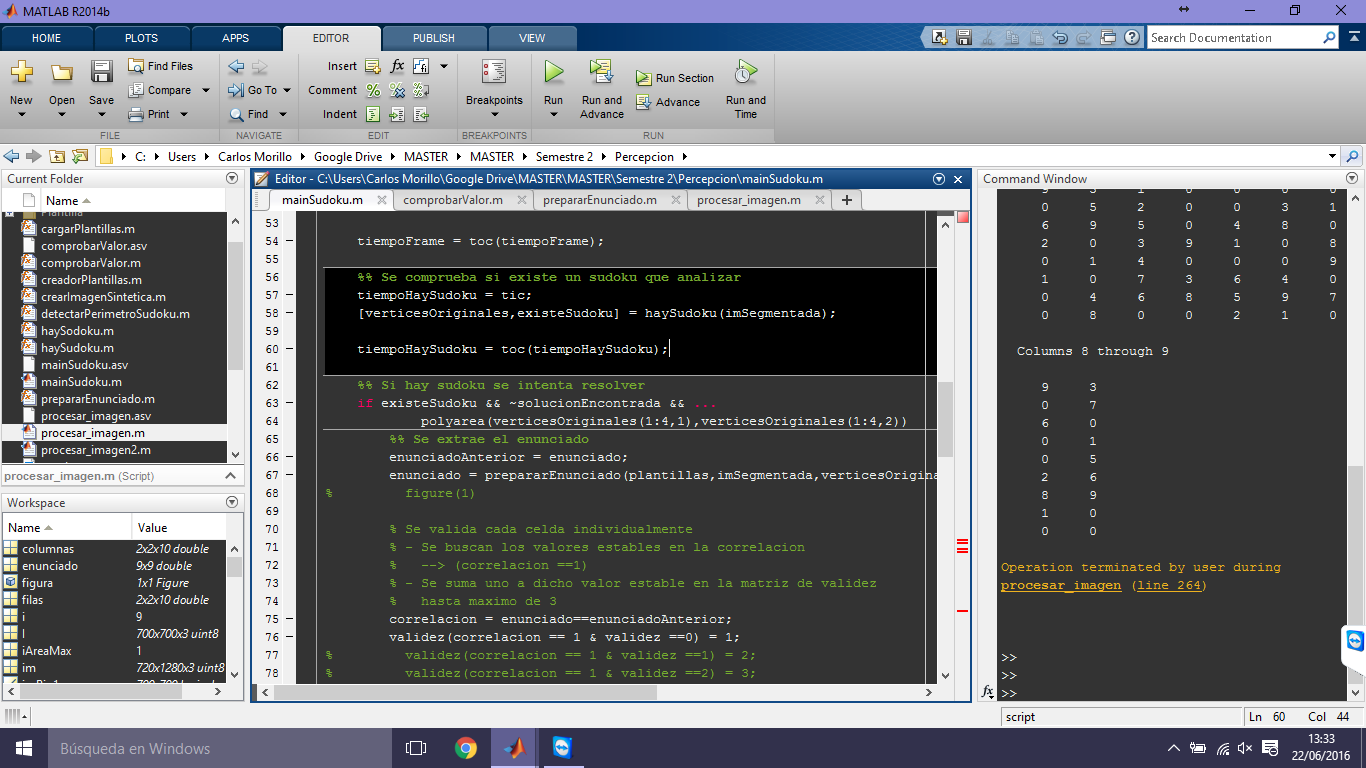
Dado que las características de iluminación no son constantes en las distintas partes de la imagen, se ha implementado una binarización por bloques utilizando la función *blockproc.* Lo que permite este método es realizar una misma función repetidas veces en una imagen aplicándose mediante un barrido por bloques del tamaño deseado. De esta forma en cada bloque el algoritmo ajustará las condiciones de iluminación a cada caso sin crear grandes diferencias entre un extremo muy oscuro de la fotografía y otro muy iluminado. Por el contrario, aumenta el tiempo de procesado, pero los resultados son excelentes.

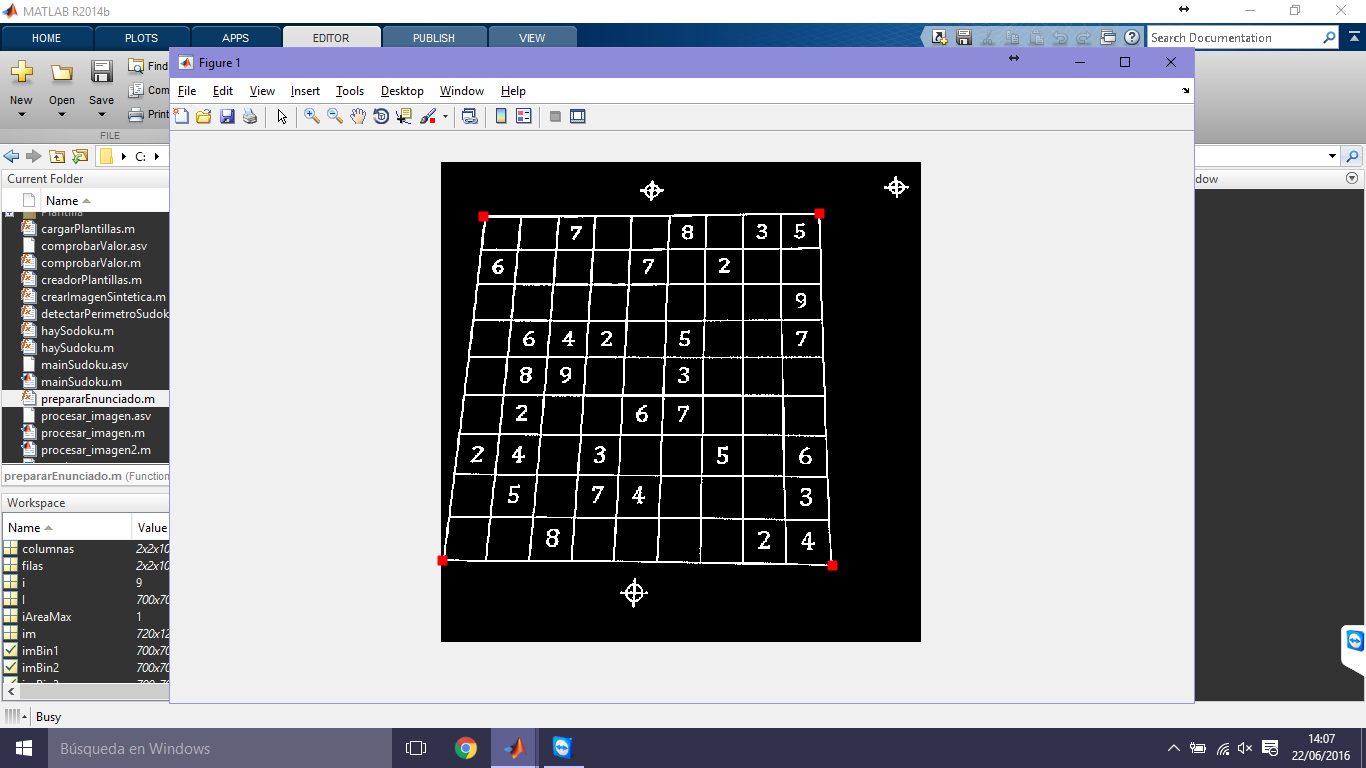
Tras el algoritmo de segmentación, la imagen se habrá transformado de la siguiente forma:

## Detección del sudoku

El siguiente paso será detectar si hay o no sudoku en la imagen para evitar realizar cálculos en caso de que no lo haya. Para eso se ha desarrollado una función que determina el objeto más grande en la imagen y si cumple unos requisitos de tamaño, calcula la posición de sus esquinas basándose en los valores de las diagonales del cuadrado que forman´.

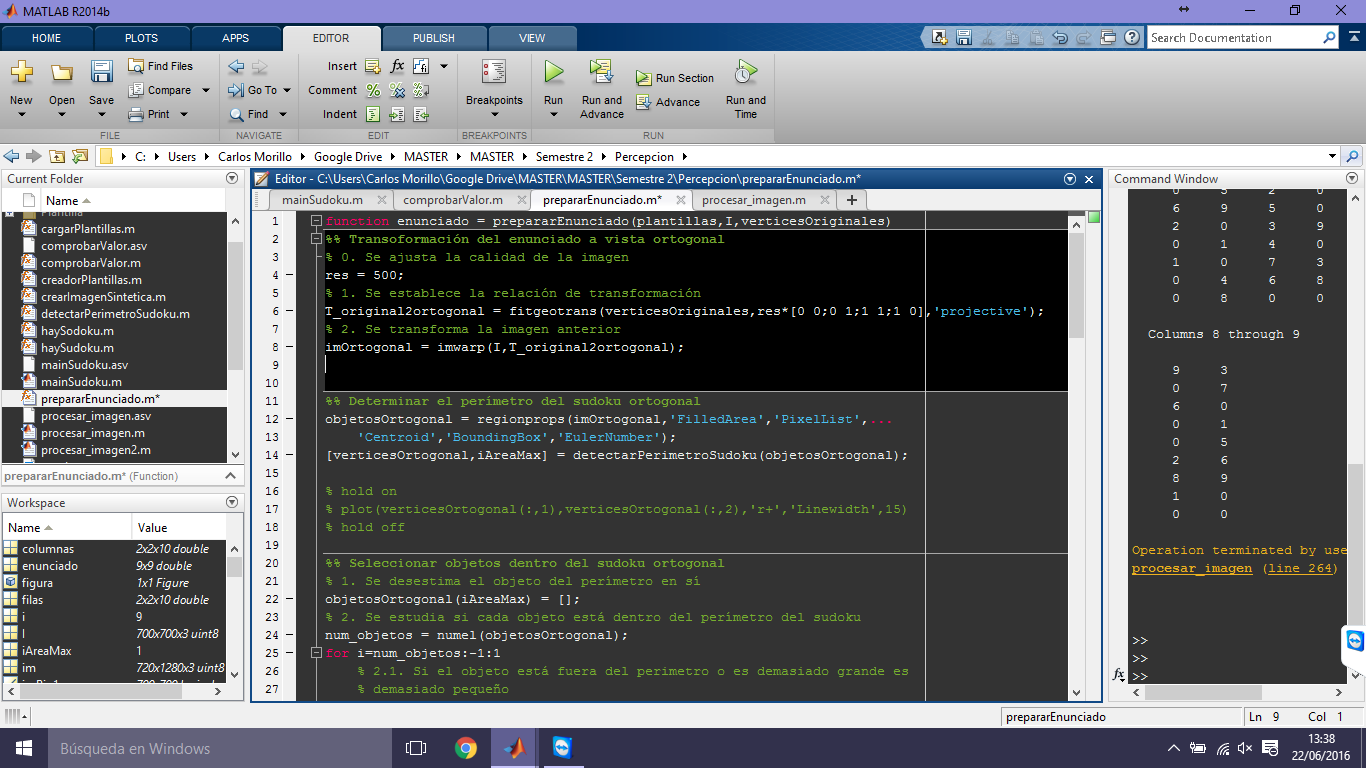


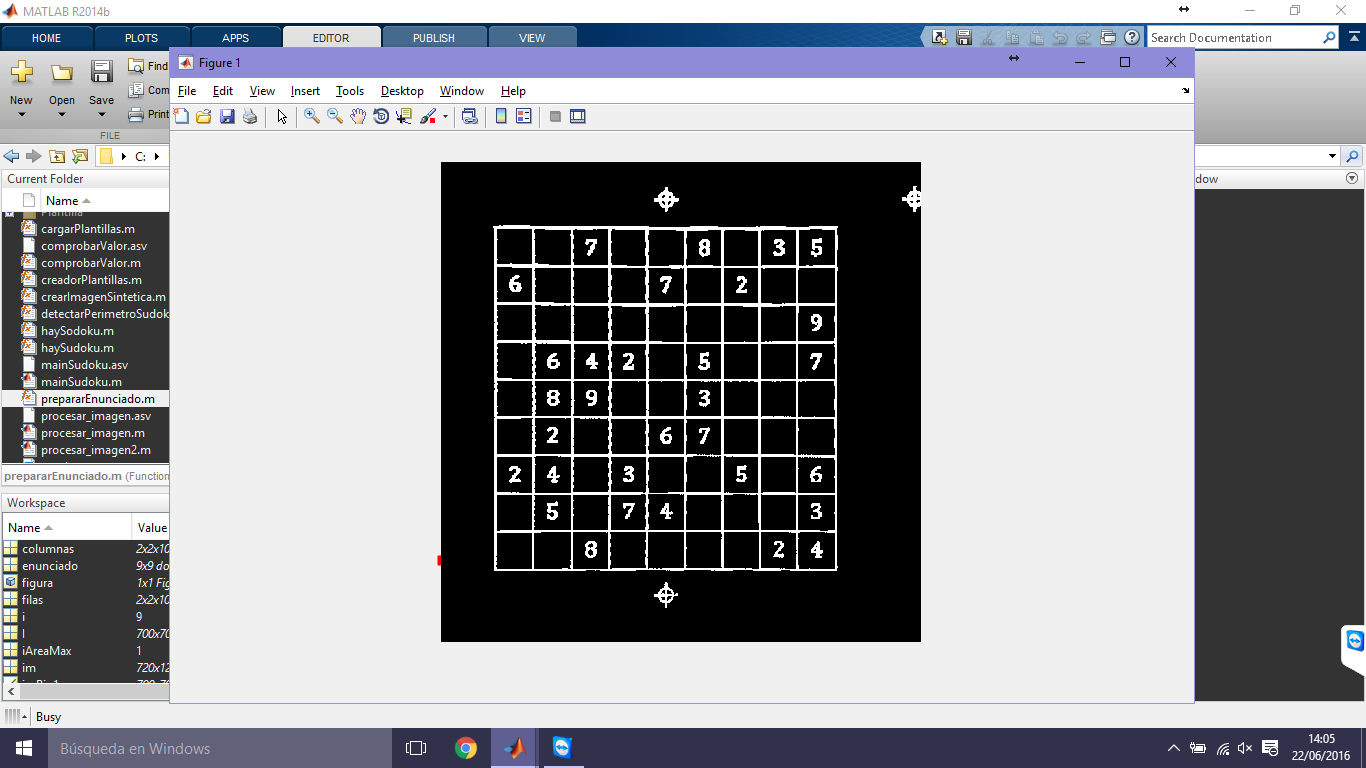


## Extraer enunciado

### Transformación

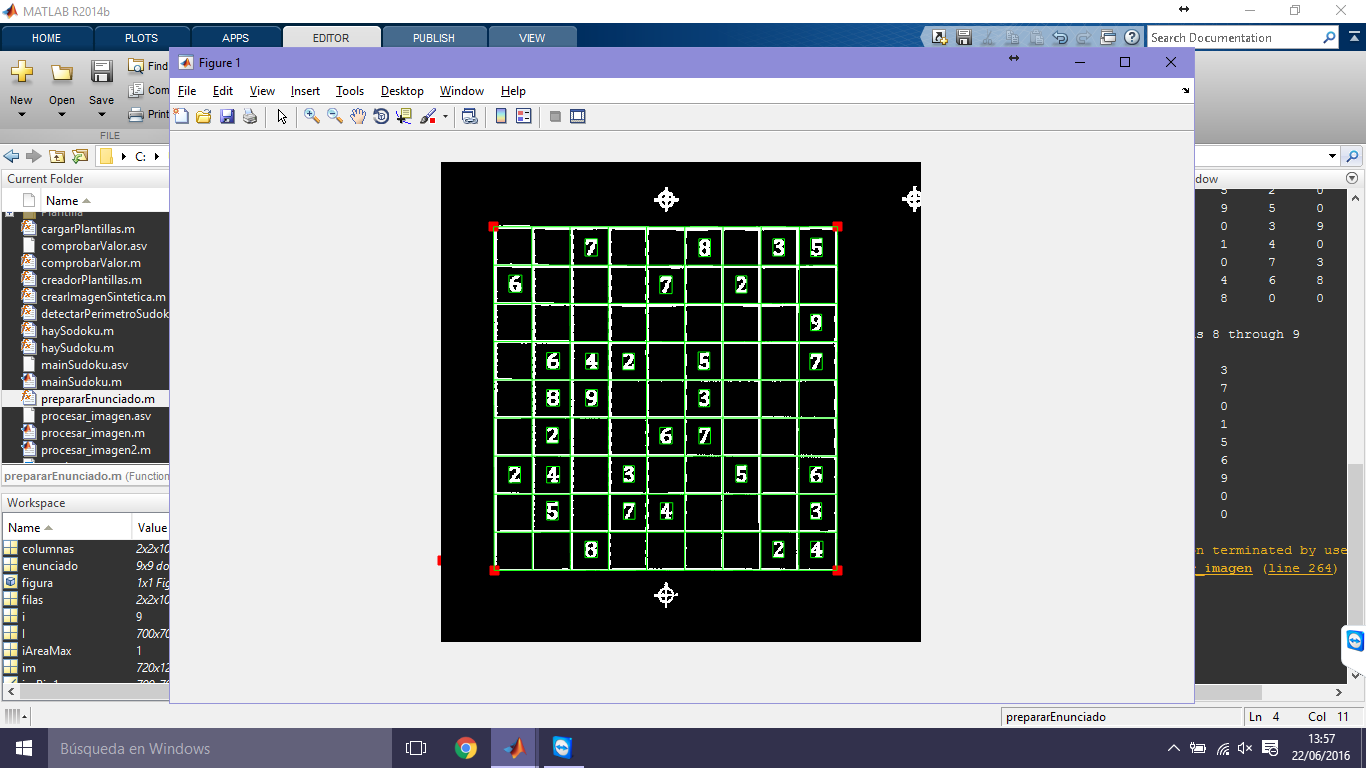
Este apartado consta de varios pasos, en primer lugar se transforma la imagen original a una vista ortogonal del sudoku. Para la transformación se usan como entrada los vértices previamente calculados y los vértices de un cuadrado normalizado de una resolución en pantalla de 500px (para no perder definición en el sudoku).





### Selección de objetos dentro del sudoku

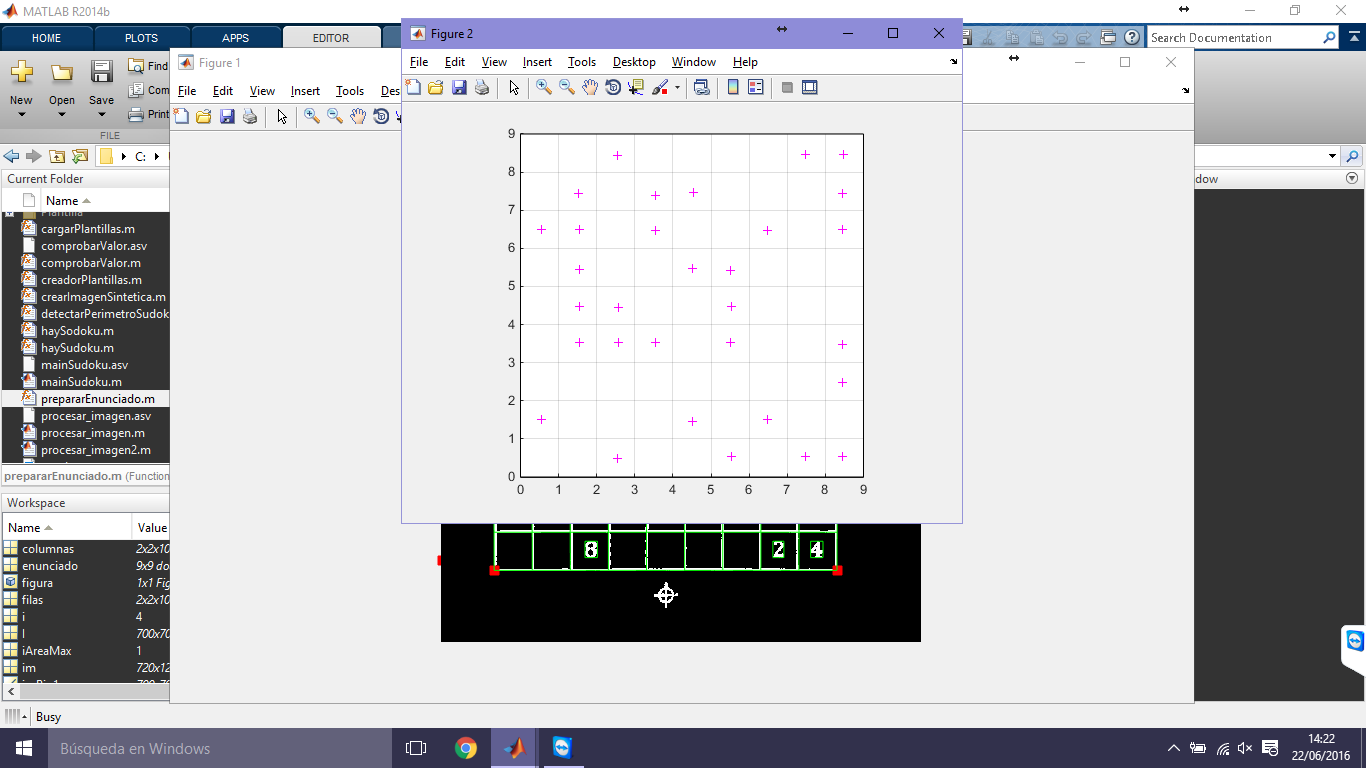
Se recalculan las esquinas del nuevo sudoku transformado y se desechan todos los objetos que no estén en el interior del mismo gracias a la función *inpolygon*.



### Filtrado de objetos no deseados

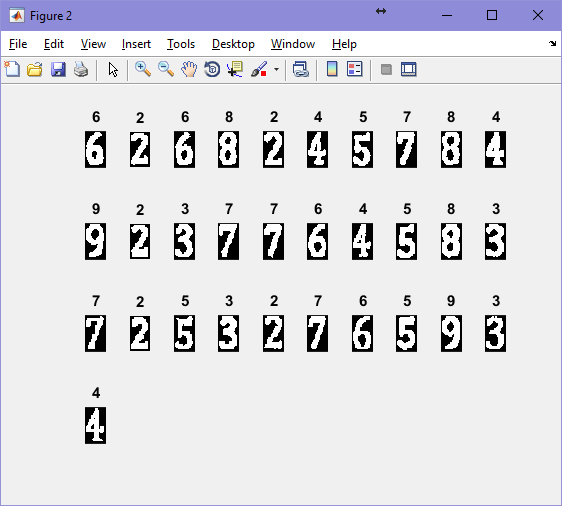
Cabe la posibilidad de que se detecte ruido procedente de las líneas interiores del sudoku, cuando esto pasa, el área de dicho objeto suele ser muy elevada o muy baja o si no, su centroide se encuentra lejos del centro de la casilla que se supone debería ocupar el objeto bueno.

El filtrado por tamaño es inmediato, sin embargo, para descartar objetos por centroide, hay que realizar una transformación. Dicha transformación pasará el sudoku ortogonal a un plano de 9x9, de esta forma, si los centroides se encuentran muy próximos a los valores enteros (1,2,3,4…) en cualquiera de los ejes x o y, significará que el objeto no está centrado en su casilla y que por lo tanto no es un número del propio enunciado. La siguiente imagen muestra dicho plano de 9x9.



### Obtención de los valores de los números

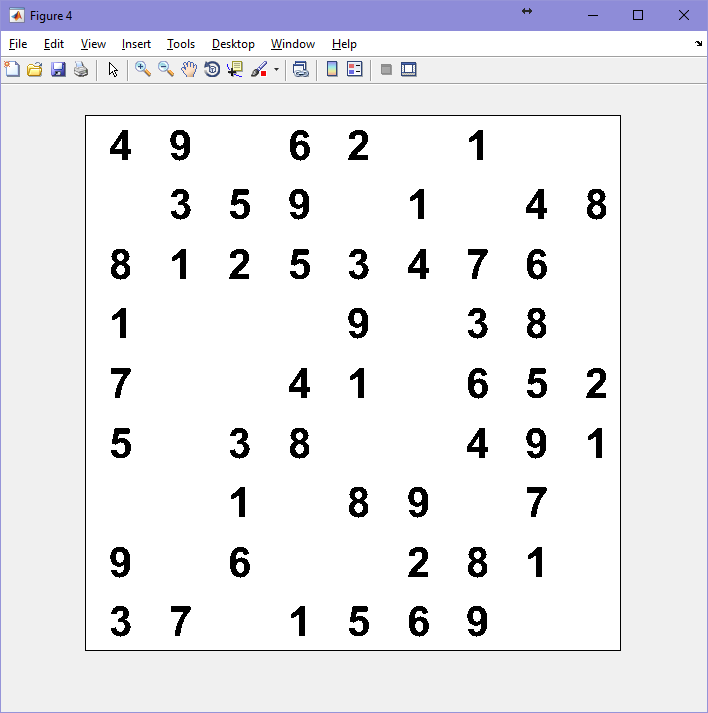
Para obtener el valor de cada número del enunciado se emplea una comparación de la imagen recortada de cada número con 9 plantillas usando la función *corr2*. La plantilla con mayor índice de correlación será el valor que se le dé al número del enunciado. Para evitar falsos positivos el valor final del número no se dará hasta que el algoritmo cumpla 2 repeticiones detectando el mismo número. Dado que se conoce el centroide de cada uno de los números, de la matriz 9x9 de la imagen anterior también se puede determinar la casilla que ocupa cada valor.



Conociendo los números y las casillas que ocupan se puede generar una matriz enunciado que pueda leer el programa Matlab y operar directamente con ella. Aquellos números donde no hay ningún valor se rellenarán con un ‘cero’.

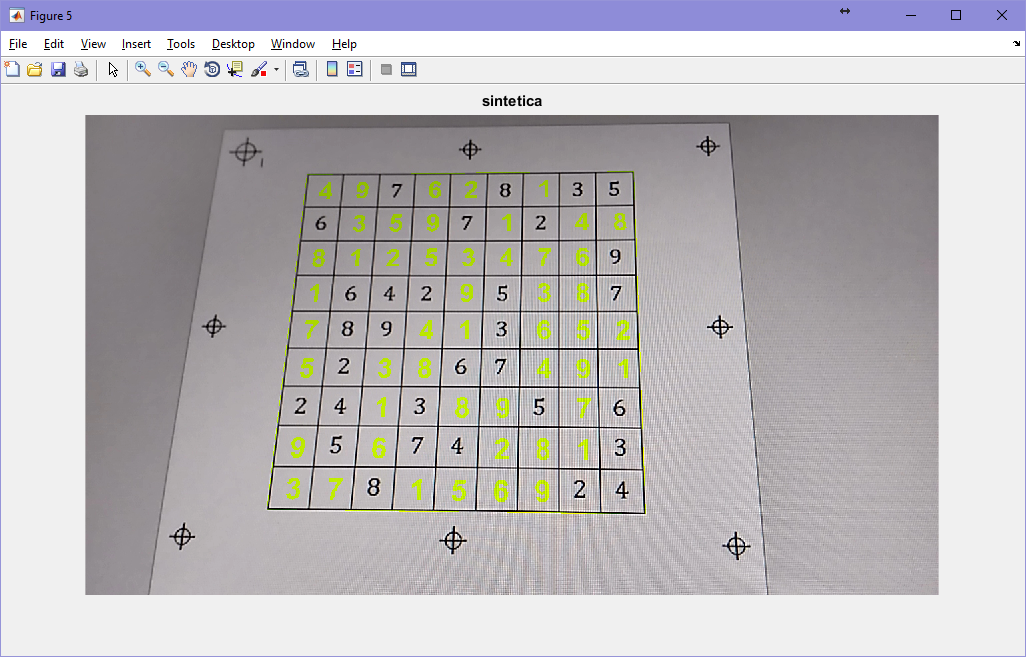
## Resolver sudoku

Una vez que ya se ha obtenido un enunciado fiable de la imagen obtenida por pantalla, se ejecuta el algoritmo de resolución del sudoku que emplea recurrencia para resolverlo. La salida obtenida se transformará a una imagen sintética cuadrada de resolución 500px, que se guardará en memoria.



## Superponer imágenes con Realidad Aumentada

Finalmente, dado que se conocen las posiciones de las esquinas tanto del sudoku en la imagen inicial como en la imagen sintética, la transformación se hace igual que al ortogonalizar la imagen en la segmentación, solo que esta vez se multiplican las imágenes para obtener dicha superposición.



# Caso: El sudoku ya ha sido resuelto

En este caso no se resolverá de nuevo para evitar tiempo de cálculo. Se realizará únicamente la superposición dado que el ángulo del sudoku real ha podido variar.

# Caso: No hay sudoku en escena

En este caso se realiza la segmentación y la comprobación de si hay o no sudoku. En caso negativo se mostrará directamente la imagen captada por la cámara.