

# TRABAJO FINAL DE MÁSTER

---





---

## SOFTWARE DE ADQUISICIÓN DE IMÁGENES Y RECONOCIMIENTO ÓPTICO DE CARACTERES PARA ANDROID

**Autor: Jaime Navarro Santapau**

**Consultores: Gregorio Robles Martínez, Roberto Gil Casas**

**Enero 2013**

- 
- **Planificación**
  - **Análisis**
  - **Diseño**
  - **Implementación**
  - **Documentación**
  - **Conclusiones y Futuros trabajos**
- 

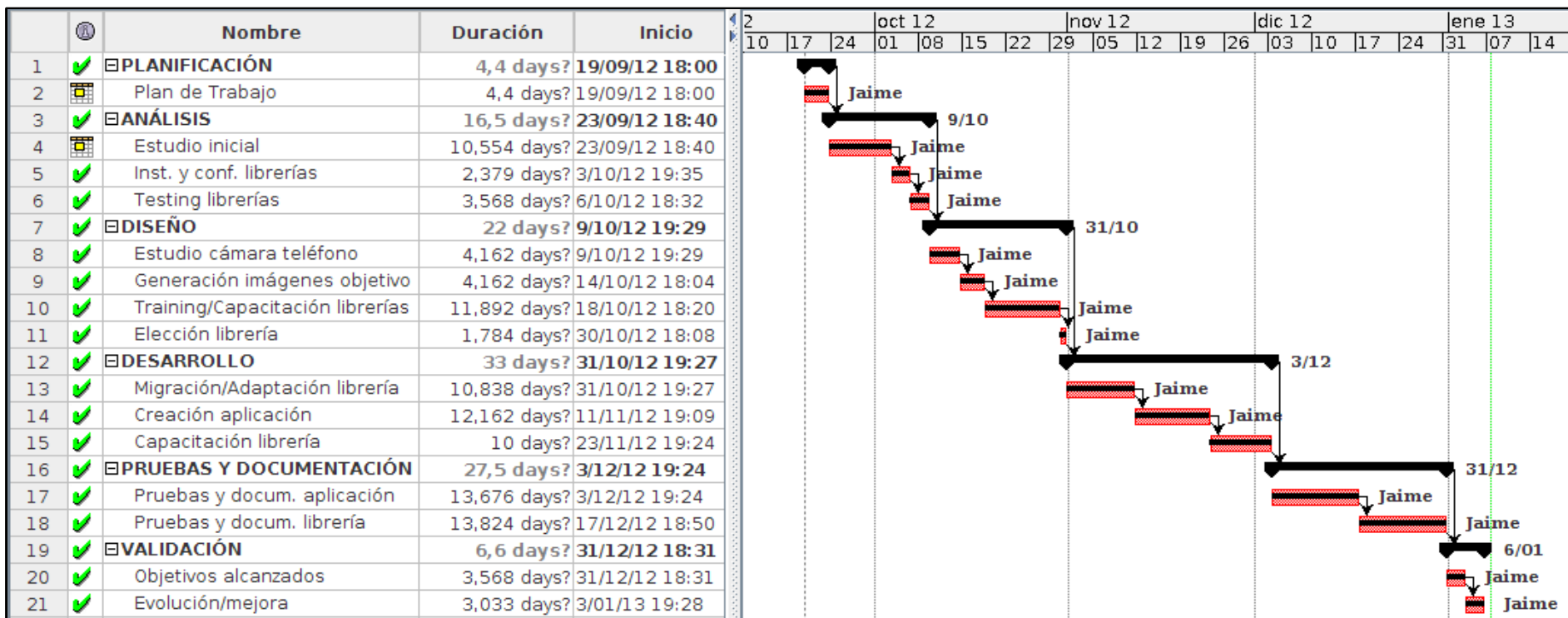


# Planificació



- **Objetivos del Trabajo Final de Máster**
  - Investigación de diferentes técnicas de OCR.
  - Estudio del estado del arte de las librerías OCR en el SL.
  - Investigación del funcionamiento de la cámara en teléfonos móviles con Android.
  - Estudio de la viabilidad de adaptación y capacitación de las librerías analizadas para nuevos patrones de referencia (Tickets).
  - Desarrollo e implementación de la librería para ser utilizada en teléfonos Android.
  - Desarrollo e implementación de una aplicación que haga uso de esta librería.

## ■ Diagrama de Gantt



## ■ Hitos establecidos al inicio del proyecto

ID	HITO	FECHA
H1	Final Análisis	09/10/12
H2	Final Diseño	31/10/12
H3	Final Implementación	03/12/12
H4	Final Pruebas y Documentación	31/12/12
H5	Entrega Proyecto	06/01/13





# Análisis



## ■ Análisis de motores OCR en SL

Proyecto	<u>WebSite</u>	Ultima Versión	Fecha publicación	Ultima actualización repositorio	Escrito en	Licencia
<b>GOOCR</b>	<a href="http://jocr.sourceforge.net/index.html">http://jocr.sourceforge.net/index.html</a>	0.49	24/09/10	31/05/12	C y C++	GPL v2
<b>JavaOCR</b>	<a href="http://sourceforge.net/projects/javaocr/">http://sourceforge.net/projects/javaocr/</a>	1.101	06/06/10	06/06/10	Java	BSD License
<b>Ocrad</b>	<a href="http://www.gnu.org/software/ocrad/">http://www.gnu.org/software/ocrad/</a>	0.21	10/01/11	22/01/12	C y C++	GPL v3
<b>Tesseract</b>	<a href="https://code.google.com/p/tesseract-ocr/">https://code.google.com/p/tesseract-ocr/</a>	3.02	01/11/12	03/11/12	C y C++	Apache License 2.0





## ■ Pruebas sobre motores OCR en SL

### Eficacia

	Imagen Test 1	Imagen Test 2	Imagen Test 3	Imagen Test 4
<b>Ocrad</b>	97,51%	98,81%	97,23%	98,90%
<b>GOOCR</b>	95,78%	96,23%	95,97%	96,78%
<b>Tesseract</b>	<b>98,34%</b>	<b>99,21%</b>	<b>98,86%</b>	<b>99,04%</b>
<b>JavaOCR</b>	86,38%	85,48%	87,48%	85,96%

### Adaptabilidad

	Adaptación nuevos caracteres	Adaptación nuevas tipografías	Adaptación palabras frecuentes	Lista de caracteres permitidos	Lista de caracteres prohibidos
<b>Ocrad</b>	NO	NO	NO	NO	NO
<b>GOOCR</b>	NO	SI	NO	NO	NO
<b>Tesseract</b>	<b>SI</b>	<b>SI</b>	<b>SI</b>	<b>SI</b>	<b>SI</b>
<b>JavaOCR</b>	SI	SI	NO	NO	NO



## ■ Pruebas sobre motores OCR en SL

	Imagen Test 1	Imagen Test 2	Imagen Test 3	Imagen Test 4
<b>Ocrad</b>	<b>0,75</b>	<b>0,82</b>	<b>0,79</b>	<b>0,75</b>
<b>GOOCR</b>	0,89	0,86	0,92	0,87
<b>Tesseract</b>	2,37	1,98	2,23	2,11
<b>JavaOCR</b>	2,12	1,82	2,25	2,17

Eficiencia (s)

Licencia

	Licencia	Tipo
<b>Ocrad</b>	GPL v3	Robusta
<b>GOOCR</b>	GPL v2	Robusta
<b>Tesseract</b>	<b>Apache License 2.0</b>	<b>Permisiva</b>
<b>JavaOCR</b>	<b>BSD License</b>	<b>Permisiva</b>

Evolución

	Fecha Ultima Release	Ultima Actualización Repositorio
<b>Ocrad</b>	10/01/11	22/01/12
<b>GOOCR</b>	24/09/10	31/05/12
<b>Tesseract</b>	<b>01/11/12</b>	<b>03/11/12</b>
<b>JavaOCR</b>	06/06/10	06/06/10



Diseño



- Adaptación a las Imágenes Objetivo





- Adaptación a las Imágenes Objetivo

### Palabras Frecuentes:

- **COMERCIO:**
- **TPV:**
- **APLIC.:**
- **MASTERCAD o VISA**
- **EUR**
- **Tran:**
- **Sec:**
- **Aut:**
- **Op:**
- **Resp:**
- **Fecha:**
- **Hora:**





- Adaptación a las Imágenes Objetivo

### Caracteres a detectar:

*A, B, C, D, E, F, G, H, I, J, K, L, M, N, Ñ, O, P, Q, R, S, T, U, V, W, X, Y, Z*

*a, c, e, h, n, o, p, r, s, t, u*

*0, 1, 2, 3, 4, 5, 6, 7, 8, 9*


*\* : / . ,*





- **Adquisición de las Imágenes Objetivo**

**Las características de la cámara principal:**

- **8 MP**
  - **3264x2448 píxeles**
  - **Autofocus**
  - **Flash LED**
  - **Geo-tagging**
  - **Detección de rostro y sonrisa**
  - **Foco táctil**
  - **Estabilizador de imagen**
  - **Vídeo 1080p@30fps**
- 



# Implementación





- **Implementación Librería:**
  - *Reutilización del código fuente de Tesseract C/C++*
  - *Reutilización del código fuente de Leptonica C/C++*
  - *Creación de una Interfaz JNI para Tesseract y Leptonica:*
    - ◆ *Basada en Tess-two y Tesseract-android-tools*
    - ◆ *Adaptación propia de la Interfaz JNI para Tesseract*
    - ◆ *Adaptación propia de la Interfaz JNI para Leptonica*
  - *Utilización del NDK de Android v8b:*
    - ◆ *Compilación Tesseract C/C++ para armeabi-v7a, armeabi, x86*
    - ◆ *Compilación Leptonica C/C++ para armeabi-v7a, armeabi, x86*
  - *Creación JAR mediante el SDK de Android v21.0.1*



- **Implementación Aplicación:**
  - ***Sistema de adquisición de imágenes***
    - ◆ *Basado en la aplicación Zxing*
    - ◆ *Configurable Flashes LED*
    - ◆ *Configurable AutoFocus*
  - ***Procesamiento a través de la librería***
    - ◆ *Basado en la aplicación Android-ocr y Eyes-free*
    - ◆ *Configurable para 4 patrones de referencia*
    - ◆ *Configurable Lista de caracteres presentes*
    - ◆ *Configurable Lista de caracteres no presentes*
    - ◆ *Configurable modo ejecución librería*
  - ***Adaptación del GUI a tres Idiomas***
  - ***Compilación mediante el SDK de Android v21.0.1***



- **Manual para la generación de un patrón de referencia:**
  - **Requisitos Imágenes de referencia:**
    - ◆ Mínimo 10 muestras de cada carácter
    - ◆ Los caracteres suficientemente espaciados
    - ◆ Las imágenes agrupadas por tipo de letra, estilo de fuente o tipografía
    - ◆ El formato de imagen a utilizar TIFF
  - **Imágenes Reales:**
    - ◆ Inconveniente: Calidad y nitidez de las imágenes
    - ◆ Inconveniente: Un conjunto amplio de imágenes (10 muestras de carácter)
    - ◆ Ventaja: Características de caracteres completamente reales
  - **Imágenes Simuladas:**
    - ◆ Inconveniente: Incertidumbre en la elección del tipo de fuente
    - ◆ Ventaja: Fácil de generar multitud de imágenes de muestra



- Vídeo demostración



# Ejecución

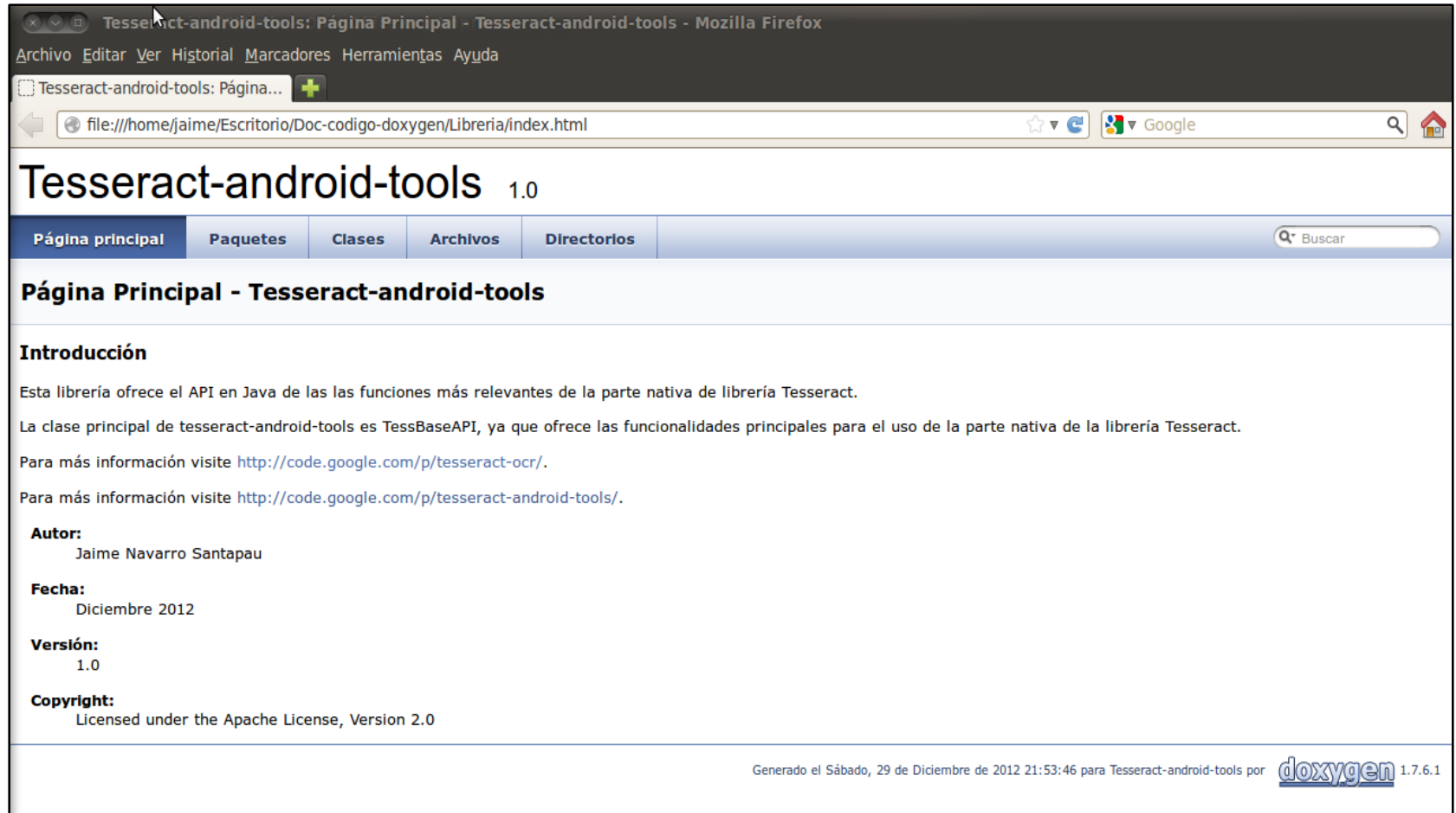
## Patrón Español



# Documentación



## Documentación Librería - Doxygen



Tesseract-android-tools: Página Principal - Tesseract-android-tools - Mozilla Firefox

Archivo Editar Ver Historial Marcadores Herramientas Ayuda

Tesseract-android-tools: Página... +

file:///home/jaime/Escritorio/Doc-codigo-doxygen/Libreria/index.html

Google

# Tesseract-android-tools 1.0

Página principal Paquetes Clases Archivos Directorios

Buscar

## Página Principal - Tesseract-android-tools

### Introducción

Esta librería ofrece el API en Java de las las funciones más relevantes de la parte nativa de librería Tesseract.

La clase principal de tesseract-android-tools es TessBaseAPI, ya que ofrece las funcionalidades principales para el uso de la parte nativa de la librería Tesseract.

Para más información visite <http://code.google.com/p/tesseract-ocr/>.


Para más información visite <http://code.google.com/p/tesseract-android-tools/>.

**Autor:**  
Jaime Navarro Santapau

**Fecha:**  
Diciembre 2012

**Versión:**  
1.0

**Copyright:**  
Licensed under the Apache License, Version 2.0

Generado el Sábado, 29 de Diciembre de 2012 21:53:46 para Tesseract-android-tools por  1.7.6.1

## Documentación Aplicación - Doxygen



OCR Configurable: Pagina Principal - OCR Configurable - Mozilla Firefox

Archivo Editar Ver Historial Marcadores Herramientas Ayuda

OCR Configurable: Pagina Princi... +

file:///home/jaime/Escritorio/Doc-codigo-doxygen/Aplicacion/index.html

# OCR Configurable 1.0

Página principal Paquetes Clases Archivos Directorios

Buscar

## Pagina Principal - OCR Configurable

### Introducción

Esta aplicación realiza el procesado OCR de imagenes mediante la librería Tesseract. Para más información visite <http://code.google.com/p/tesseract-ocr/>.

- Permite configurar ciertos parámetros para optimizar la adquisición de las imagenes:
  - Autoenfoque.
  - Luz delantera.
- Permite configurar ciertos parámetros del motor Tesseract:
  - Procesar las imágenes mediante un patrón específico realizado por nosotros, que debemos situar en `"/mnt/sdcard/Android/data/edu.uoc.ocr/files/tessdata"` con el nombre `man.traineddata`.
  - Procesar las imágenes mediante alguno de los patrones generales descargados desde <http://tesseract-ocr.googlecode.com/files/>.
  - Definir la lista de caracteres permitidos.
  - Definir la lista de caracteres prohibidos.
  - Seleccionar el modo de ejecución de la librería Tesseract.

**Autor:**  
Jaime Navarro Santapau

**Fecha:**  
Diciembre 2012

**Versión:**  
1.0

**Copyright:**  
Licensed under the Apache License, Version 2.0

Generado el Sábado, 29 de Diciembre de 2012 21:56:44 para OCR Configurable por [doxygen](#) 1.7.6.1



# Conclusiones y Futuros Trabajos





- **Las características más importantes del software:**
  - *Adaptabilidad en el proceso de adquisición de las imágenes*
  - *Adaptabilidad en el procesado OCR*
  - *Interfaz gráfico traducido a diferentes idiomas*
  - *Documentación del código fuente*
  - *Creación de una guía para la generación de nuevos patrones de referencia*
- **Futuros Trabajos:**
  - *Sistema de adquisición de imágenes*
    - ◆ *Métodos adquisición y Restricción Hardware*
  - *Mejora del patrón de referencia*
    - ◆ *Limitación Tipografías*
    - ◆ *Estudio de los tickets más comunes*
    - ◆ *Acceso a Terminales TPV*



**Gracias por su atención...**

