



**Universitat Autònoma  
de Barcelona**

[#ReadThis - Un servicio que lee el texto  
en las imágenes enviadas a través de  
Twitter](#)

**Mario Arias Escalona**

## Índice

Introducción .....	3
Descripción .....	3
Contexto .....	3
Necesidad .....	3
Información previa .....	3
¿Qué es Twitter? .....	3
¿Existen límites a la hora de utilizar las APIs? .....	4
¿Hay otras aplicaciones cliente/servidor basadas en Twitter? .....	4
Datos de interés .....	5
Objetivos .....	5
Metodología y planificación .....	6
Proceso de trabajo a desarrollar .....	6
Metodología de desarrollo .....	6
Planificación .....	7
Bibliografía .....	8
Webgrafía .....	8

## Introducción

En el presente documento se exponen de manera detallada los primeros pasos que se llevarán a cabo en la realización del proyecto “[#ReadThis - Un servicio que lee el texto en las imágenes enviadas a través de Twitter](#)”. En primer lugar, se contextualizarán y describirán los objetivos que se pretenden conseguir. En un segundo lugar se mostrará la planificación y metodología del proyecto, que se seguirá a lo largo de la realización del mismo.

## Descripción

### Contexto

El proyecto es una iniciativa del Centro de Visión por Computador. Se trata de una institución pública líder en investigación y desarrollo en su campo. La Generalitat y la UAB lo fundaron en el año 1995 con la finalidad de hacer una investigación de excelencia mediante la generación de conocimiento de calidad y la transferencia de tecnología hacia la sociedad, ofreciendo también valor añadido a las empresas. Desde su creación, el CVC integra el Servicio de Tratamiento de Imágenes, servicio científico-técnico de la UAB.

Está formado por más de 130 investigadores y especialistas procedentes de todo el mundo.

### Necesidad

Dado que se trata de un centro público de investigación y en consonancia con la comisión europea, la intención del CVC es la de seguir con la idea de “Open Science”. Representa un enfoque del proceso científico basado en el trabajo cooperativo y nuevas formas de distribución del conocimiento mediante tecnologías digitales.

Siguiendo en la línea anterior, desde el CVC se quieren activar procesos por los que se transfieran conocimientos científicos de esta institución a otras organizaciones, realizando así lo que se conoce como transferencia tecnológica.

*“La transferencia de tecnología es el conjunto de acciones orientadas a facilitar el rendimiento comercial en el mercado de las capacidades investigadoras y los resultados de investigación de las actividades de I+D que realizan las universidades, institutos, centros de investigación y empresas. Corresponde a una etapa del proceso global de valorización y posterior comercialización de la tecnología.”*

(OVTT, Centro Virtual de transferencia de tecnología, s.f.)

Es por esto por lo que nace la necesidad de crear un servicio que sea accesible por cualquier persona u organización, y que estos puedan ver las posibilidades que ofrecen los sistemas y algoritmos desarrollados en el CVC.

## Información previa

### ¿Qué es Twitter?

Twitter es una aplicación web gratis de microblogging, y por tanto reúne las ventajas de los blogs, las redes sociales y la mensajería instantánea. Esta forma de comunicación te permite conectar con personas afines a tus gustos o aficiones, y acceder de manera instantánea a las publicaciones que ellos realizan.

## ¿Cómo prestar un servicio a través de twitter?

A través de la API de la propia plataforma. Una API es una “llave de acceso” a funciones que nos permiten hacer uso de un servicio web provisto en este caso por Twitter, dentro de una aplicación web propia, de manera segura.

## ¿Qué se puede obtener con las APIs de Twitter?

Twitter ofrece tres APIs: **Streaming API**, **REST API** y **Search API** aplicables a necesidades diferentes.

**Streaming API** proporciona un conjunto de tweets a tiempo real. Se establece una conexión permanente por usuario con los servidores de Twitter y mediante una petición http se recibe un flujo continuo de tweets.

**Search API** suministra los tweets desde hoy hasta 7 días atrás.

**REST API** ofrece a los desarrolladores el acceso a los datos de Twitter. Todas las operaciones que se pueden hacer vía web son posibles realizarlas desde el API. Dependiendo de la operación requiere o no autenticación, con el mismo criterio que en el acceso web.

## ¿Existen límites a la hora de utilizar las APIs?

La respuesta es sí, y se deberá tener muy en cuenta a la hora de crear una aplicación web que interactue con la misma. Las limitaciones mas destacades son:

- \***1.000 actualizaciones en total por día**, desde cualquier web y cualquier dispositivo (Web, móvil, teléfono, API, etc)
- \***250 mensajes directos totales por día**, desde cualquier web y cualquier dispositivo (Web, móvil, teléfono, API, etc)
- \***150 solicitudes de la API por hora**
- \***Las imágenes recibidas están codificadas únicamente en JPG.**

Tras **24 horas** la mayoría de estas limitaciones de uso de la API se revocan y nuestra cuenta vuelve a funcionar correctamente.

En el caso que se desee construir sistemas que hagan un uso intensivo de la API de Twitter, estaría la posibilidad de registrar la aplicación. Los límites de acceso al API sin registro como hemos visto anteriormente son 150 solicitudes por hora, mientras que para aplicaciones registradas en la "whitelist" (previo pago) podrían llegarse a hacer 20.000 solicitudes por hora.

## ¿Hay otras aplicaciones cliente/servidor basadas en Twitter?

Al tener una API abierta, existen infinidad de aplicaciones que interactúan con Twitter. Para el tema que nos concierne que es el tratamiento de imágenes existen algunas aproximaciones de proyectos que aplican algoritmos OCR como, por ejemplo:

**Reverse OCR** - Reverse OCR dibuja líneas aleatorias hasta que el software de reconocimiento óptico de caracteres piense que parece una cierta palabra.

También podemos encontrar bots que utilizan algoritmos de reconocimiento para cuestiones como saber cuánta gente cuelga fotos cuando tiene poca batería en el móvil.

**Lowbatterymuch – GitHub**

Actualmente no existe ningún BOT que sea capaz de utilizar múltiples algoritmos de reconocimiento de imagen a partir de una misma cuenta. Si bien encontramos proyectos de muy diversa índole, en el campo del tratamiento de imágenes todavía queda mucho por hacer.

### Datos de interés

Los tuits con imágenes reciben un **18% más de clics**, un 89% más “me gusta” y un 150% más RTs.

Las empresas que usan Twitter para dar servicio al cliente **incrementan la satisfacción hasta en un 19%**.

Un **60% de los consumidores** espera que las marcas respondan a sus preguntas en Twitter en menos de una hora, pero el promedio es una hora y 24 minutos.

El **76% de los consumidores** recomendarían una marca después de una interacción amigable.

El **77% de los usuarios** de Twitter tienen un sentimiento positivo con respecto a una marca cuando ésta les contesta un tuit.

Fuente: [www.brandwatch.com](http://www.brandwatch.com)

## Objetivos

Este proyecto se trata de una prueba de concepto que abarcara la creación de un sistema que sea capaz, a través de una tecnología concreta del CVC, de procesar imágenes a través de Twitter. En este piloto probablemente se utilizará la detección de texto en imágenes. Los objetivos del proyecto son los siguientes.

1. Buscar información relevante para determinar la necesidad y la construcción del piloto.
2. Analizar la información recopilada, seleccionándola y organizándola detalladamente, determinando así el alcance y las necesidades del sistema.
3. Diseñar el sistema con la información y las especificaciones previamente concretadas.
4. Desarrollar un sistema que sea capaz de procesar una imagen procedente de Twitter.
5. Creación de documentación de las diferentes partes del sistema.
6. Realización de pruebas unitarias y pruebas de carga.

## Metodología y planificación

### Proceso de trabajo a desarrollar

En primer lugar, habrá una búsqueda de información de todos los puntos relevantes del proyecto tales como: tecnologías a utilizar, API de Twitter, Servidores, lenguajes de programación más adecuados en cada módulo, bases de datos, encapsulación de algoritmos en otros lenguajes, Web Services, deploy en la nube, etc....

Una vez recolectada toda la información, se procederá al diseño de la arquitectura del sistema. Con toda la información recopilada y con los conocimientos técnicos del equipo de desarrollo, se diseñará de manera conceptual cada parte del sistema, siendo este diseño modificable a lo largo del proyecto, ya que, si se encuentra alguna incompatibilidad a la hora de desarrollar un determinado sistema, se deberá hacer un reajuste en el desarrollo, estimando el alcance del cambio y procediendo a que este cause el menor impacto posible.

### Metodología de desarrollo

La metodología que se seguirá en este proyecto es una metodología ágil, basada en sprints. Los sprints tendrán una duración de dos semanas, intentando que la planificación de estos sea cada vez más precisa y acotada.

En una primera fase el objetivo es conseguir una visión clara de lo que se necesita, se va a hacer, fabricar o conseguir. Se tendrán en cuenta riesgos y compensaciones, qué es posible y qué es factible.

Una vez superada esta primera fase se estará en disposición de realizar una lista de tareas. Se trata de una lista de todo lo que debe hacerse para convertir la visión en realidad. Esta lista existe y evoluciona a lo largo del proceso, es el mapa o la hoja de ruta del proyecto. En cualquier momento, la lista de tareas pendientes es la única y definitiva vista panorámica de todo lo que se puede hacer, por orden de prioridades.

Cuando se tenga una lista de tareas habrá que ir ítem por ítem haciendo una estimación de cuánto tiempo llevara realizar cada uno de ellos. La estimación no se hará en horas ya que esto es contrario a las metodologías ágiles. Las tareas se medirán en: **Pequeñas, Medianas y Grandes**

### Proceso iterativo

Con todos los elementos anteriores están cerrados, será el momento de realizar el sprint planning. Será el momento de marcar las tareas a realizar durante las próximas 2 semanas. Al finalizar el sprint, habrá que hacer una pequeña retrospectiva de todo lo realizado en este ciclo. Habrá que analizar la acotación de las estimaciones y el resultado obtenido. Una vez finalizado esto volveríamos a empezar otro ciclo siguiendo todos los pasos anteriores.

Para la realización de esta metodología se utilizará GIT para el control de versiones y Trello como tablero de tareas.

## Planificación

Al tratarse de una metodología ágil la planificación no será exacta, ya que la ventaja que nos proporciona este método es la facilidad con la que se puede reestructurar el orden en que se realizan las diferentes partes del proyecto. De todas formas, a nivel global a continuación se puede ver una planificación global donde se especifican las diferentes fases por las que ir avanzando el proyecto

❖ Sprint 0 (Estudio de la arquitectura, preparación de GIT y de Tablero SCRUM)

❖ Sprint planning (lista de pequeñas tareas abordables en el sprint).

❖ Sprints ( 8 Sprints, con el último de 1 semana)

❖ Sprint review (revisión del sprint)

❖ Semana de parón

❖ Sprint de reléase (informe final, pruebas de carga, test, etc)

❖ Milestones (entrega de informes)

❖ Defensa TFG

## Octubre

L	M	Mi	J	V	S	D
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31					

## Diciembre

L	M	Mi	J	V	S	D
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

## Noviembre

L	M	Mi	J	V	S	D
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30			

## Enero

L	M	Mi	J	V	S	D
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

# Febrero

L	M	Mi	J	V	S	D
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28				

## Bibliografía

- [1] Amir Shevat, « **Designing Bots: Creating Conversational Experiences 1st Edition**», *May 2017*
- [2] James Shore, «**The Art of Agile Development**», *2007*
- [3] James Goodwill, Aleksa Vukotic, «**Apache Tomcat 7**», 3 de septiembre de 2011
- [4] Ekim Kaya, «**Bot Business 101: How to start, run & grow your Bot / AI business**», *2007*
- [5] Martin Kalin, «**Java Web Services: Up and Running: A Quick, Practical, and Thorough Introduction 2nd Edition**», *2009*
- [6] Sam Ruby, Leonard Richardson, «**RESTful Web Services**», *2007*

## Webgrafía

<http://ec.europa.eu/research/index.cfm> - Comision europea

<http://www.ovtt.org> – Observatorio virtual de transferencia tecnologica

<https://qz.com> -Quartz media

<https://www.brandwatch.com/es/> - Tendencias en brandwatch

<https://blog.twitter.com/> Blog de twitter

<https://developer.twitter.com/> Twitter developers

<https://blog.trello.com/> - Blog de Trello

<https://github.com/> Repositorio GitHub.