Venezolanos en Twitter: ¿Humanos, Bots o Ciborgs? Modelo de Clasificación

■ Josemy I. Duarte F.
email: duartejosemy@gmail.com
Facultad de Ciencias
Universidad Central de Venezuela
Caracas, Venezuela

■ Gabriel E. Rodríguez G.
email: gersaibot3688@gmail.com
Facultad de Ciencias
Universidad Central de Venezuela
Caracas, Venezuela

■ Jesus Lares
email: jesus.lares@gmail.com
Facultad de Ciencias
Universidad Central de Venezuela
Caracas, Venezuela

■ José R. Sosa B.
email: josersosa@gmail.com
Facultad de Ciencias
Universidad Central de Venezuela
Caracas, Venezuela

Fecha de Recepción: 15 de Agosto 2016
Fecha de Aceptación: 26 de Septiembre de 2016

RESUMEN

Twitter es una red social de microblogging que experimentó un aumento descomunal de popularidad en el 2009, convirtiéndola en una de las plataformas sociales más influyentes en la actualidad. Este fenómeno ocasionó que surgieran distintos tipos de cuentas que perjudican la interacción entre los usuarios esparciendo contenido spam, influyendo opiniones y realizando publicaciones con fines meramente publicitarios. Este tipo de usuarios son conocidos como bots, los cuales se caracterizan por tener un comportamiento

automático y programado para cumplir sus funciones. Sin embargo, los bots no son el único tipo de usuarios que pueden tener un comportamiento automatizado; existen usuarios humanos que pueden asistir cuentas bot o utilizar herramientas para programar parte de su comportamiento. Este tipo de usuarios se conoce como ciborgs. En la presente investigación se estudian las características inherentes al contenido y el comportamiento de los usuarios venezolanos de la plataforma de Twitter con el fin de detectar patrones que permitan clasificar a los usuarios en tres categorías: humanos, ciborgs y bots. Se desarrolló un modelo de Machine Learning entrenado a partir de conjuntos de tweets spam y no spam, y conjuntos de usuarios humanos, ciborgs y bots. Finalmente se evaluó el modelo, obteniendo un 86% de exactitud.

Palabras Clave—humano; ciborg; bot; twitter; Venezuela; machine learning; random forest; spam;

I. INTRODUCCIÓN

Twitter es una plataforma que ofrece servicios de microblogging. Es considerada una de las redes sociales con mayor influencia y penetración social a nivel global desde el descomunal aumento de popularidad que percibio dos años luego de su creación en el 2006, pasando de 475.000 usuarios en febrero de 2008 a más de 7.038.000 usuarios para febrero de 2009, experimentando un crecimiento anual de 1372% [1]. Este increíble aumento de popularidad posicionó a Twitter dentro del espectro mundial de comunicación y, hasta la fecha, contabiliza alrededor de 310 millones de usuarios activos por mes [2]. Sin embargo, a medida que este tipo de redes/medios sociales crecen suelen desarrollarse cierto tipo de fenómenos bastante comunes, tales como el contenido spam, las publicaciones o entradas con fines meramente publicitarios y, en especial, los usuarios bots. La situación expuesta anteriormente conlleva a plantearse interrogantes vitales para la gran mayoría de estudios, análisis y proyectos basados en esta plataforma: ¿Cuántos de los usuarios activos dentro de Twitter son realmente usuarios completamente humanos? ¿Es posible detectar de forma automática si un usuario es bot?. El objetivo de esta investigación es analizar características específicas asociadas al perfil de usuarios de Twitter y al contenido publicado por los mismos a partir de una muestra de datos extraídos de la red en cuestión, presentando una serie de medidas que permitan elaborar un modelo con el fin de categorizar a los usuarios en tres grupos básicos:

Humanos: Cuentas de twitter correspondiente a las personas e individuos comunes, con comportamiento irregular y generadores de contenido original.

Bots: Cuentas donde la generación de contenido se encuentra automatizado de acuerdo a ciertos parámetros.

Ciborgs: Cuentas mixtas donde la publicación de contenido es compartida entre humanos y bots.

II. INVESTIGACIONES RELACIONADAS

Desde su creación en el 2006 y gracias a su impacto social, Twitter ha sido objeto de estudio en múltiples áreas. La evolución de esta y otras plataforma de social media fue estudiada de forma reciente por Ferrara et al. [3]. Su trabajo también abordó el efecto que tienen los bots sobre estas plataformas, planteando un conjunto de consecuencias negativas de gran relevancia y la composición de distintos sistemas de detección de usuarios bots. El efecto de los bots sobre las redes sociales es también estudiado por Boshmaf et al. [4], cuyo trabajo se concentró en evaluar y

determinar cuan vulnerables son las redes sociales con respecto a la infiltración a gran escala por parte de los "socialbots": programas de computadora que controlan cuentas de redes sociales e imitan usuarios reales. Por otra parte, en cuanto a la interacción entre los usuarios humanos y usuarios bots, Murgia et al. [5] realiza una serie de experimentos preliminares, tomando como caso de uso al sitio web Stack Overflow, con el fin de determinar en qué medida un bot puede simular el comportamiento de un humano y cuál es la retroalimentación que recibe. El propósito y la influencia de los usuarios bot no necesariamente deben ser siempre considerados como un elemento perjudicial para una plataforma o red social, por lo tanto, la capacidad de discernir cuales cuentas automatizadas son benignas y cuales son malignas es fundamentalmente importante para modelos de clasificación de usuarios y sistemas de detección de bots. Dentro de esta perspectiva, Penna [6] realizó un estudio sobre el comportamiento en línea de tres tipos de cuentas de usuarios en Twitter: 1) personales, que corresponden a usuarios humanos; 2) asistidas, referente a las cuentas de corporaciones, y 3) bots. Aparte, realizaron un análisis estadístico sobre los perfiles de usuarios y crearon dos algoritmos de Machine Learnig basados en el comportamiento de las publicaciones: 1) un clasificador Bayesiano ingenuo y 2) un modelo de predicción probabilístico. En adición a estudios concentrados en la categorización de usuarios en Twitter, Chu et al. [7] determinaron las principales diferencias entre usuarios humanos, bots y ciborgs con respecto a los patrones de publicación, el contenido de los tweets y las características asociadas a los perfiles. Fundamentándose en los resultados obtenidos, propusieron un sistema de clasificación compuesto por 4 componentes: 1) un componente basado en la entropía, 2) un componente de detección de spam, 3) un componente de propiedades de las cuentas, y 4) un componente para la toma de decisiones.

III. RECOLECCIÓN DE LOS DATOS

Para la recolección de datos se utilizó un método de muestreo de datos ego-céntrico [8] aplicado en dos fases:

Contrucción del árbol de cuentas

Se eligieron ocho cuentas de usuarios consideradas de gran interés para la población venezolana, apuntando a que la mayoría de sus seguidores, y los seguidores de estos, estuvieran relacionados con Venezuela. Las cuentas utilizadas fueron: 1) @metro_caracas (Metro de Caracas), 2) @UNoticias (Periódico Ultimas Noticias), 3) @noticierovv (Noticiero Venevisión), 4) @trafficMIRANDA (Informes de tráfico del Estado Miranda), 5) @BcodeVenezuela (Banco de Venezuela), 6) @ifetren (Ferrocarriles Venezuela), 7) @SomosMovilnet (Operadora Movilnet) y 8) @MeridianoTV (Canal de TV Meridiano).

Para cada cuenta se extrajeron los IDs de los primeros 80 seguidores devueltos por el API de Twitter, repitiendo el proceso para cada uno de los seguidores obtenidos de forma recursiva, haciendo uso de un algoritmo conocido como Búsqueda por Anchura (Breadth-first search) [9] para la construcción del árbol de cuentas. El proceso se detuvo con un total de 110.000 cuentas.

Obtención de los timelines de usuarios

Se realizó un recorrido del árbol de cuentas construido en el paso anterior extrayendo todos los datos disponibles de los últimos 800 tweets del timeline de cada usuario mediante el API de Twitter. El proceso de extracción de datos siempre estuvo limitado por las restricciones establecidas por Twitter en el uso de su API. Estas limitaciones se

encuentran claramente expresadas la página oficial para desarrolladores de Twitter [10]. Debido a estas restricciones, la recolección de datos tomó 1 mes y 2 semanas en 2 máquinas dedicadas 24/7 a la extracción de los datos de Twitter.

IV. PREPARACIÓN DE LOS DATOS

El proceso de preparación de los datos recolectado estuvo constituido por las siguientes etapas:

Calculo de la muestra de datos

De acuerdo al Ministerio del Poder Popular para la Comunicación e Información de Venezuela (MINCI) [11], al cierre del año 2015 Venezuela contaba con al menos 16 millones de personas con acceso a internet. Es natural pensar que no todas y cada una de las personas con acceso a internet en Venezuela poseen cuentas en la plataforma de Twitter. Para ajustar aún más la cifra, segun el estudio realizado por Statista (Digital Market Outlook) [12] referente al número de usuarios activos en Twitter para mayo del 2016, el último país entre las primeras once naciones con más usuarios en la plataforma de Twitter es España, con aproximadamente 7.52 millones de usuarios. Razón por la cual es sensato considerar que en Venezuela existe una cantidad inferior a 7.52 millones de usuarios.

Sin embargo, debido a la carencia de información certera sobre esta cifra, se decidió realizar los cálculos y evaluaciones asumiendo que Venezuela cuenta con, como máximo, 10 millones de usuarios activos en Twitter, la cual es una cantidad incluso superior a la presentada por México, el cual ocupa el 8vo lugar entre los países con más usuarios dentro de la plataforma.

Para establecer una cantidad de usuarios que pueda definirse como representativa basada en el número de usuarios que se asumen activos en Twitter para Venezuela (N), se realizó el cálculo para el tamaño de la muestra [13] asumiendo la constante 0.5 como la desviación estándar (σ) de la población, un nivel de confianza (Z) de 99% que deriva en el valor 2.58 y un límite aceptable de error muestral (e) establecido en 5%.

$$\frac{N\sigma^2 Z^2}{e^2(N-1)+\sigma^2 Z^2} \approx 665.595$$

Resultando en, aproximadamente, 666 como cantidad de usuarios suficiente para satisfacer los criterios definidos para el cálculo de la muestra. Sin embargo, para esta investigación se decidió escoger a 1.000 usuarios, provenientes de los 110.000 timelines recaudados, como muestra representativa para cada categoría de usuarios, reduciendo el límite del error muestral a 4% y obteniendo un total de 3.000 cuentas de Twitter para conformar el set de datos inicial.

Cada una de estos usuarios se clasifico manualmente por un componente humano entre los tres grupos previamente definidos. Para cada categoría de usuarios, el 80% (800 usuarios) se utilizó como set de datos de entrenamiento y el 20% restante (200 usuarios) fue utilizado para probar la eficacia del modelo.

Clasificacion manual de usuarios

Para el proceso de clasificación manual de cada usuario se realizó el siguiente conjunto de actividades:

Se visitó la página principal del usuario (http://twitter.com/username).

Se revisaron las características asociadas al perfil del usuario, tales como la

cantidad de tweets publicados, número de seguidores, número de amigos, número de publicaciones favoritas, fecha de creación de la cuenta (en caso de ser pública), imagen de perfil por defecto y respuestas a publicaciones. Se tomó en cuenta la coherencia del contenido publicado con respecto al perfil general del usuario. Por ejemplo: Si una cuenta tiene un perfil que refiere a un humano pero posee una extraña cantidad de publicaciones con fines publicitarios; ó si publica contenido sin sentido semántico, para ambos casos se dice que existe incoherencia con respecto a su perfil.

La frecuencia de respuestas a otras publicaciones sospechosamente automatizadas también es un factor de relevancia para determinar la categoría a la que corresponde una cuenta.

Se inspeccionó el timeline del usuario para examinar características adicionales como, por ejemplo, los dispositivos de publicación.

Un usuario es clasificado como humano si se obtuvo evidencia de que el contenido publicado es inteligente, original, coherente y similar al contenido que podría publicar un humano. Por otra parte, un usuario es clasificado como bot si: el contenido publicado carece de originalidad, existe una cantidad excesiva de publicaciones automáticas, cuenta con una cantidad anormal de tweets duplicados, y si la cantidad de seguidores y amigos es exageradamente alta para un corto periodo de tiempo. Por último, un usuario es clasificado como ciborg si no puede clasificarse como humano pero tiene suficiente contenido original como para suponer que se trata de una cuenta asistida (refiérase tanto a una cuenta bot asistida por un humano o una cuenta de un humano con cierto grado de automatización).

Creación de conjuntos de Tweets spam y no spam

Se elaboraron dos conjuntos de datos de forma manual a partir de tweets que cumplían características específicas: uno correspondiente a aquellos tweets que fueron catalogados como spam, y otro correspondiente a los tweets clasificados como no spam. Se consideraron como spam aquellos tweets provenientes de cuentas bots, con enlaces externas maliciosas o con publicidad no deseada. Algunas cuentas bots "avanzadas" esconden los tweets spam entre tweets no spam; este tipo de tweets fueron ignorados. Se consideró como no spam a los tweets provenientes de usuarios humanos, sin enlaces externos o archivos multimedia. Como medida conservadora, el set de datos no spam no contiene tweets de bots o ciborgs.

Consideraciones

Los usuarios cuyos timelines que estaban protegidos al momento de la recolección de datos fueron excluidos del estudio, así como también aquellos usuarios cuyo conteo de publicaciones era inferior 100, ya que se consideraron como cuentas con poca actividad para realizar su respectiva categorización.

V. COMPONENTES DE EVALUACIÓN

El proceso para la definición del modelo de clasificación de usuarios en tres categorías (humanos, ciborgs y bots) resultó en la creación de tres componentes primarios:

Componente de caracteristicas del timeline

Muchas características asociadas a las cuentas de usuarios de Twitter poseen un alto nivel descriptivo. En este componente se extraen y agrupan las características de interés para el modelo de clasificación, tales como: el número de publicaciones desde la creación de la cuenta, número de seguidores, numero de amigos, numero de publicaciones favoritas, año de registro de la cuenta,

verificación del perfil, imagen de perfil, numero de listas, funcionalidad de geolocalización, entre otras.

Componente detector de spam

El componente detector de spam examina el contenido de los tweets para detectar spam. Basado en el set de tweets spam y no spam creados en la fase de preparación de los datos, se procedió al entrenamiento de un modelo de bosques aleatorios (Random Forest) [14] debido a que demostró poseer una mayor taza de acierto que otros modelos en otros estudios [15], al igual que en las pruebas realizadas haciendo uso de un algoritmo Bayesiano. Para efectos de comparación se pueden apreciar los resultados de dichas pruebas en la Tabla I.

TABLA . RANDOM FOREST VS BAYESIANO

	Algoritmos	
	Bayesiano	*Random Forest*
Precisión	0.764	0.811

Para el entrenamiento del bosque aleatorio se utilizó el método de bolsa de palabras (Bags of Words) [16] para la extracción de características de los tweets. Tanto el entrenamiento del modelo, como la extracción de las características de los tweets fueron realizados mediante el conjunto de funcionalidades ofrecidas por el proyecto Apache Spark para Machine Learning (ML).

Juez

Para la implementación del juez de usuarios se optó igualmente por la utilización de un modelo Random Forest, principalmente debido a su eficacia en procesos de clasificación en casos que involucran más de dos categorías, además de ser capaz de manejar una gran cantidad de características o variables, pudiendo descartar aquellas que no proporcionen suficiente información para discriminar entre las categorías; y de haber demostrado buenos resultados en estudios relacionados [7]. Este juez utiliza la lista de características respectivas de cada usuario para realizar su predicción, juzgándolo como humano, bot o ciborg.

VI. ANÁLISIS DE LOS DATOS

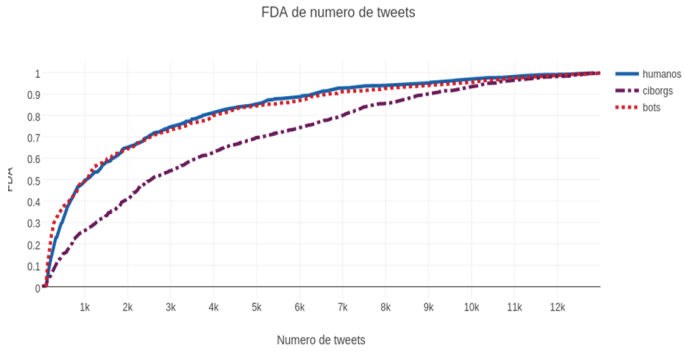


Figura 1

Se desarrollaron distintos programas y herramientas para visualizar de forma útil los datos de extraídos, los cuales fueron obtenidos completamente en formato JSON. Cabe destacar que, con el propósito de extraer información útil de la data analizada, en la mayoría de los casos se utilizó una Función de Distribución Acumulada (FDA) para representar el porcentaje de elementos (eje y) que cumplen la condición establecida por la característica evaluada (eje x).

A continuación se presentan algunas observaciones interesantes sobre la data procesada:

Volumen de Tweets

La Figura 1 presenta la Función de Distribución Acumulada del número de tweets para cada usuario perteneciente a las tres categorías de humanos, ciborgs y bots.

Se puede notar que, para el 50% de usuarios humanos y bots (y=0.5) presentan una cantidad menor o igual a

aproximadamente 1.000 tweets. En cambio, el 50% de los usuarios ciborgs (y=0.5), presentan una cantidad menor o igual a aproximadamente 2.000 tweets. Esto quiere decir que los usuarios ciborg realizan en su mayoría más publicaciones que los usuarios bot y los usuarios humanos. Este volumen considerable de publicaciones se atribuye a los propósitos comerciales que suelen tener este tipo de cuentas. Aunque cierta parte de la gestión de estas cuentas es realizada por empleados, la mayoría de las publicaciones son realizadas por herramientas automatizadas. Por otro lado, las cuentas de usuarios humanos y bots tienen un comportamiento similar en cuanto al conteo de sus publicaciones. Sin embargo, el volumen de tweets de ambos grupos no necesariamente pudo ser generado en el mismo instante de tiempo. Los períodos de activación de las cuentas bots nivelan el volumen de publicaciones con respecto al generado por comportamiento el constante de los usuarios humanos.

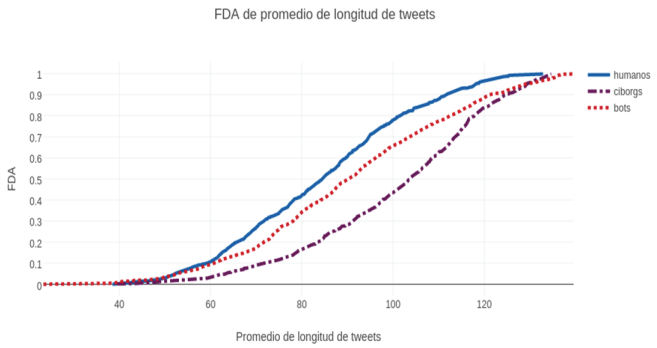


Figura 2
Longitud de Tweets

Una característica insigne de Twitter es el límite de caracteres permitidos para cada tweet. Actualmente, este límite establecido en 140 caracteres no contempla los caracteres relacionados al contenido multimedia o a los nombres de usuarios (respuestas y menciones). En la Figura 2 se calcula la FDA de la longitud promedio de los tweets para cada usuario de las distintas categorías. La mayoría de los usuarios humanos solamente utilizan la cantidad de caracteres necesaria para expresar sus ideas u opiniones. En contraparte a este resultado, los usuarios ciborgs aprovechan al máximo la cantidad de caracteres límite con el fin de incluir toda la información posible en sus publicaciones, en su mayoría con fines publicitarios. Los bots obtienen un resultado intermedio en esta medida ya que sus publicaciones dependen fundamentalmente del tipo contenido que se dedique a publicar cada cuenta.

Uso de enlaces

Se evaluó la frecuencia con la que se encontraron enlaces externos en el contenido de los tweets publicados por los distintos tipos de usuarios. Como se puede observar en la Figura 3, los bots tienden a incluir enlaces en sus publicaciones con mayor frecuencia respecto a los otros usuarios. Este comportamiento tiene como propósito redirigir a los usuarios a las páginas de interés para el administrador de la cuenta. En muchos casos, los bots suelen incluir más de un enlace en cada tweet. Los ciborgs siguen de cerca a los bots respecto a la proporción de enlaces publicados en sus tweets. Un gran número de ciborgs suele integrar su timeline con fuentes RSS o actualizaciones de blogs, generando tweets con títulos de artículos seguidos por enlaces a la página web que ofrece el resto de la información. Los humanos poseen la menor cantidad de enlaces externos por tweet publicado, debido a que generalmente sus publicaciones describen lo que está haciendo, pensando o lo que sucede a su alrededor, lo cual es descrito en su mayoría con solo texto, sin ningún tipo de enlace a otros sitios web.

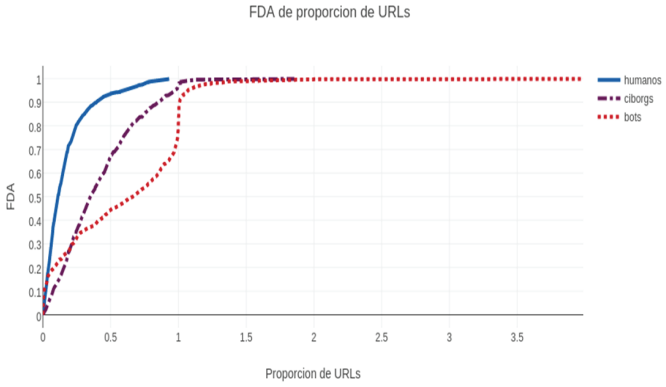


Figura 3

Contenido spam

En la Figura 4 se evalúan los resultados obtenidos durante la detección de contenido spam, basado en lo descrito en los componentes de evaluación, los cuales demuestran una clara diferencia de comportamiento entre las tres categorías de usuarios. Se puede destacar a los usuarios bots como los mayores generadores de contenido spam. Este resultado está relacionado al propósito que suele tener este tipo de cuentas con respecto a la generación de contenido no deseado y publicidad engañosa. Sin embargo, cabe destacar que no todas las cuentas de este tipo (con comportamiento y funciones automatizadas) generan contenido spam dentro de Twitter. Luego se encuentran los ciborgs que, debido a su naturaleza hibrida, poseen una proporción intermedia entre los humanos y los bots.

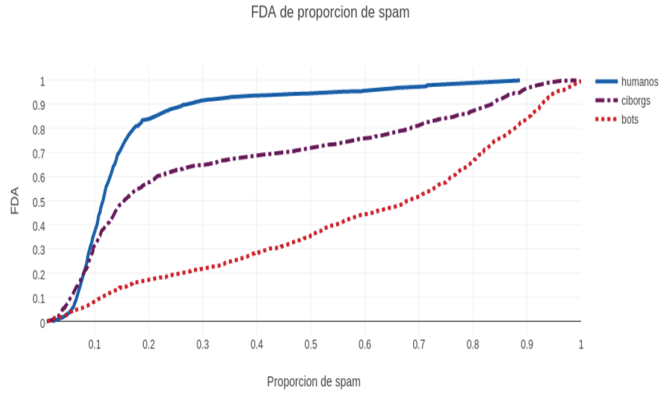


Figura 4

Dispositivos de publicación

Se realizó un proceso de agrupación sobre las diversas fuentes disponibles para la publicación de tweets, de donde resultaron tres categorías. El uso del sitio web oficial de Twitter es la única fuente categorizada como “Uso web”. Cualquier publicación proveniente del cliente oficial de twitter para dispositivos móviles (Blackberry, Windows Phone, iOS, Android, entre otros) se catalogaron como “Uso móvil”. Por último, cualquier publicación proveniente de una fuente no incluida en las dos categorías anteriores (SmarTV, TweetDeck, RSS, entre otros) se catalogaron como “Uso de Terceros”, refiriéndose al uso del API de Twitter desde terceros para la publicación de los tweets.

Tal como se puede apreciar en la Figura 5, los humanos prefieren el uso de los dispositivos móviles para la publicación de sus mensajes, relegando al sitio web de Twitter como la segunda opción y el uso desde terceros como la alternativa menos utilizada. Los bots, en total contraste respecto a los humanos, tienen como medio predilecto de publicación a las fuentes terceras, debido a las capacidades de automatización que muchas de ellas ofrecen. Respecto a los ciborgs se puede observar un comportamiento parecido al de los humanos, destacando un mayor uso en las fuentes de terceros, compartiendo una

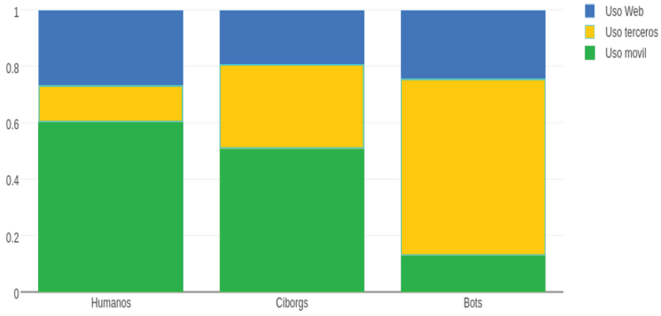


Figura 5

ligera similitud con los bots, demostrando su naturaleza heterogénea.

Menciones a otros usuarios

Un usuario es capaz de mencionar a otro usuario específico dentro de una publicación con el propósito de compartir contenido de una forma más directa. En la Figura 6 se señala el comportamiento de las tres categorías de usuario con respecto a esta funcionalidad. Los usuarios humanos y ciborgs presentan gran similitud pero se se diferencian drásticamente de los usuarios bots, esto puede explicarse de la siguiente forma: 1) los humanos tienden a interactuar con mayor frecuencia con otros individuos dentro de la plataforma que los usuarios de las otras categorías, 2) las cuentas ciborgs suelen pertenecer a compañías, marcas registradas o proveedores de servicio ya que requieren un componente humano para atender y dar respuesta las exigencias de sus seguidores, y 3) la interacción constante y fluida con otros usuarios es difícil de automatizar por lo que muchas cuentas bots carecen de esta capacidad.

Respuestas a publicaciones

Una publicación es considerada como una “respuesta” si comienza con el nombre de usuario (@username, por ejemplo) de la persona a quien va dirigida dicha respuesta.

En la Figura 7 se puede notar una separación considerable entre las tres categorías de usuarios en relación a esta funcionalidad de Twitter. Los usuarios humanos destacan con la mayor proporción de respuestas entre sus publicaciones, debiéndose esto a que la mayoría de la interacción de este tipo en Twitter es realizada precisamente por usuarios humanos. Solamente una cantidad minúscula y excepcional de bots realizan respuestas a otros usuarios. Por otra parte, los usuarios ciborg se encuentran entre los humanos y los bots, el cual es el comportamiento esperado para este tipo de cuentas mixtas.

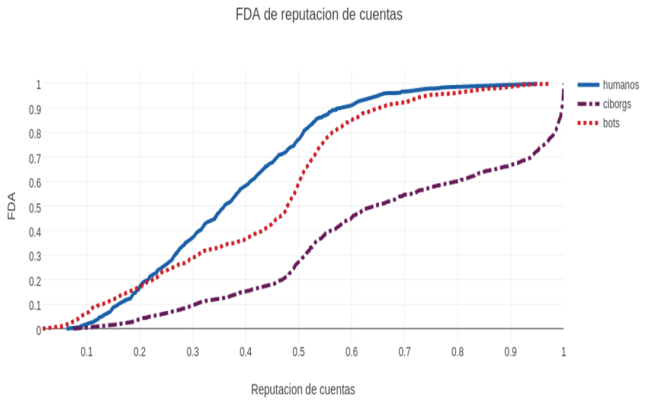


Figura 6
Relacion entre seguidores y amigos

Dentro de la plataforma de twitter, cada usuario tiene la capacidad de seguir distintas cuentas (amigos) y de ser seguido por otros usuarios (seguidores). Para medir la relación entre la cantidad de seguidores y la cantidad de amigos de un usuario calculamos el valor de reputación de una cuenta, el cual definimos como:

$$Reputaciónde cuenta = \frac{seguidores}{seguidores + amigos}$$

Los valores de reputación más altos (cercanos a uno) corresponden a los usuarios seguidos por muchas cuentas pero que siguen a pocos usuarios, como también a los usuarios con muy pocos seguidores pero con una cantidad considerable de amigos. Este último comportamiento se presenta claramente para la categoría de usuarios humanos en la Figura 8, donde muestra la FDA de la reputación de los usuarios de cada grupo. Por otro lado, la categoría ciborg presenta el comportamiento opuesto, donde a partir del percentil 30 los usuarios tienen una reputación igual o mayor a 0.5

Entropía de los grupos

El valor entrópico calculado permite medir y representar el nivel de irregularidad que tiene un usuario con respecto a la frecuencia de sus publicaciones. Los usuarios humanos, en su gran mayoría, tienden a comportarse de forma irregular en contraste con las publicaciones programadas periódicamente de los usuarios ciborg y bot. Para la creación de la Figura 9, se normalizaron los valores entrópicos originales entre el valor de entropía máximo y mínimo de la base de conocimiento, teniendo como resultado la entropía relativa. Se puede observar como el grupo de usuarios humanos se diferencia en su mayoría de los usuarios ciborgs y bots, los cuales tienden a solaparse.

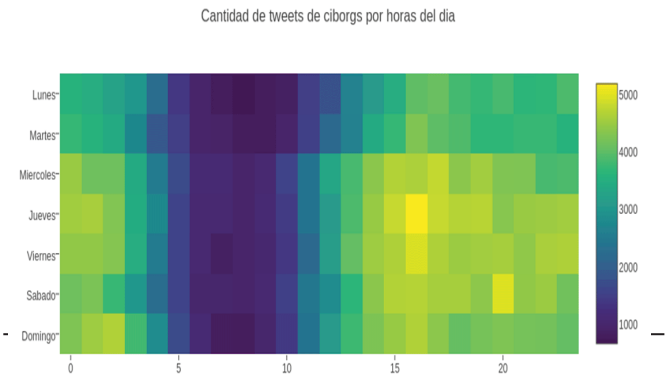


Figura 7

Horarios de alto tráfico

Dentro del estudio se determinó que la hora y el día en el que un tweet es publicado permiten diferenciar de una forma interesante a los usuarios de cada grupo. En la Figura 10, Figura 11 y Figura 12 se puede visualizar cuales son los momentos de cada día, de los siete días de la semana, en los que cada grupo presenta mayor actividad de publicación. Se puede resaltar que los usuarios humanos tienen fuertes picos de actividad entre las 00:00 horas y las 04:00 horas, en especial el día domingo.

Figura 11. Volumen de publicaciones de ciborgs por hora en la semana (GMT +00:00) Por otra parte, los usuarios ciborg tienen un amplio periodo de actividad entre aproximadamente las 14:00 horas y las 02:00 horas en todos los días de la semana. Los usuarios bots

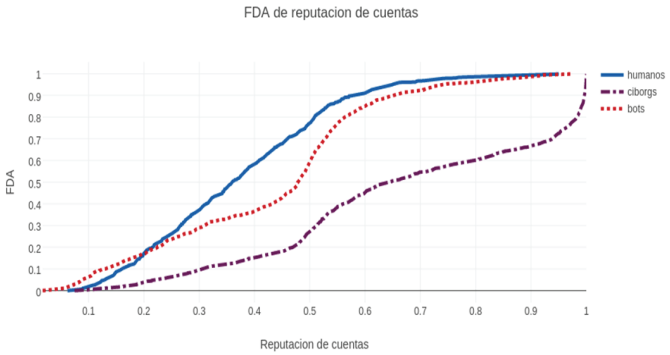


Figura 8

		Clasificados			Total	Precisión
		Humanos	*Bots*	*Ciborgs*		
Reales	Humanos	180	8	12	200	0,9
	Bots	0	183	17	200	0,915
	Ciborgs	15	29	156	200	0,78
					avg	0,865

poseen un comportamiento similar a los ciborg, concentrando su actividad en los días miércoles, jueves y viernes; pero con un decrecimiento considerable en la cantidad de sus publicaciones los días lunes y martes. Los tres grupos de usuarios generan la menor cantidad de volumen por día en el periodo comprendido por las 06:00 horas y las 10:00 horas.

CARACTERÍSTICA	PESO	CARACTERÍSTICA	PESO
Horas de publicación	0,1957	Días de publicación	0,0504
Promedio de spam	0,0935	Año de registro	0,0494
Proporción de respuestas	0,0904	Proporción de menciones	0,0481
# de listas	0,0831	Proporción de seguidores	0,038
Reputación	0,0777	Proporción de enlaces	0,0348
Publicaciones por móvil	0,0522	# de favoritos	0,0332

VIII. CONCLUSIÓN

Para obtener el mejor modelo se genera una tabla de parámetros que Spark podrá utilizar para crear el modelo del Random Forest. Spark se encarga de realizar una validación cruzada usando el set de datos de entrenamiento, el cual está constituido por el 80% de los datos del set inicial, utilizando las distintas configuraciones disponibles entre los parámetros de la tabla y retornando aquel modelo cuya configuración de parámetros haya proporcionado la mayor exactitud durante la evaluación.

El set de datos de prueba, representado por el 20% de datos restante, es utilizado para calcular la matriz de confusión y la exactitud del modelo. Se tomó la medida de calcular la exactitud del modelo y la matriz de confusión con un set de datos completamente independiente para garantizar robustez y confianza en los resultados, aun cuando el modelo fue realizado con una validación cruzada.

En la Tabla II se presentan los resultados de ejecución del Juez sobre los usuarios de Twitter categorizados manualmente en los tres grupos definidos. En ella las filas denotadas como “Reales” son la categoría real a la cual pertenecen los usuarios. La columna “Clasificados” expresa el resultado del juez. Por ejemplo, en la intersección de la fila y la columna “Humanos” se señala que 180 usuarios que son humanos, fueron clasificados correctamente como humanos. En cambio, en la intersección de la fila “Bots” y la columna “Ciborgs” se muestra que 17 usuarios que son “Bots” fueron clasificados erróneamente como “Ciborgs”. En general, se obtuvo una exactitud de 86.5%.

La Tabla III señala la relevancia medida en pesos de las características más importantes utilizadas por el Random Forest para la clasificación de los usuarios. La relevancia de cada característica indica que tan importante es la misma para el modelo, lo que quiere decir que, a mayor relevancia, más diferenciable es un usuario al ser evaluado por dicha característica. El cálculo de los pesos fue realizado en base al índice Gini [17] para el cual, en cuanto mayor sea la medida, más variabilidad aporta la característica independiente que está siendo evaluada. El cálculo consiste en los siguientes dos pasos:

- Importancia de la característica j = suma (sobre los nodos en los cuales se encuentra repartida la característica j) de la ganancia de información, donde la misma es escalada por el número de instancias pasadas a través del nodo.
- Normalización de las importancias del árbol para igualar su suma a uno (1).

Se puede observar que ninguna característica es determinante de forma absoluta, sino que es la suma de todas ellas permite realizar una buena categorización.

En la actualidad, muchos estudios que involucran el análisis de la interacción de los individuos en las redes sociales se pueden interpretar como indicadores de la realidad social. Bien sea porque estudian la forma en la que se expresan en las redes como un indicador de la opinión general, o por referirse a las relaciones entre usuarios de la red social relacionadas de alguna forma con la vida real, en cualquier caso no se puede negar la paridad por la que se suelen realizar estos estudios.

El desarrollo de esta investigación intenta contribuir con ese tipo de estudios, ofreciéndoles un modelo que les permita descartar aquellos tipos de usuarios que quizás no aporten información valiosa a la investigación o que generen cierto ruido en los datos. Igualmente, su uso no queda limitado al descrito anteriormente, sino al ingenio del investigador y a cualquier función que le pueda encontrar a lo expuesto en esta investigación.
Cada usuario de Twitter es libre de publicar el contenido que desee (texto, imágenes, videos, URLs a otras páginas…), permitiéndole expresarse libremente. Luego del estudio realizado, es posible afirmar que el contenido de las publicaciones hechas por cada tipo de usuario, sea humano, bot o ciborgs, suele diferir entre cada categoría. Sea por que los humanos con frecuencia publican mensajes más cortos que los bots o ciborgs, o porque los bots incluyen con mayor frecuencia URLs en sus publicaciones, o la cantidad de spam presente en las publicaciones de los bots respecto a los otros… No se puede negar que el contenido de las publicaciones sugiere que existen diferencias entre cada categoría.

Tomando en cuenta que el contenido publicado por los usuarios realmente marcaba una diferencia entre ellos, se procedió a estudiar el comportamiento. En este caso, también se observaron diferencias entre las categorías. Se observó por ejemplo, que los humanos suelen realizar sus publicaciones desde dispositivos móviles con mayor frecuencia que los bots, que suelen preferir publicar desde aplicaciones de terceros. También que la frecuencia de respuestas a las publicaciones al igual que las horas de actividad, son características que definen diferencias bien demarcadas entre los distintos tipos de usuario.

Luego de estudiar el contenido de las publicaciones y el comportamiento de los usuarios (humanos, bots y ciborgs) en Twitter, se observó que existían una serie de características que podrían ser útiles para la clasificación automática de las distintas categorías de usuarios. Fueron estas características las utilizadas para el entrenamiento y evaluación de un modelo Random Forest que logró una precisión de un 86%, cuya debilidad se puede decir son los ciborgs, especulando que esto es debido a su naturaleza hibrida.

Cabe destacar que las tecnologías elegidas para el desarrollo del modelo (Apache Spark) hacen viable la evaluación de grandes cantidades de datos de forma distribuida y paralela, debido a los principios en los cuales está fundamentado.

IX. REFERENCIAS

M. McGiboneyn, "Twitter's Tweet Smell Of Success", 2009. [En línea]. Disponible: http://www.nielsen.com/us/en/insights/news/2009/twitters-tweet-smell-of-success.html.

Twitter, Inc., "Empresa | About", 2016. [En línea]. Disponible: https://about.twitter.com/es/company.

E. Ferrara, O. Varol, C. Davis, F. Menczer, y A. Flammini, “The rise of social bots”, en Communications of the ACM, vol. 59, iss. 7, Eds. ACM New York, pp. 96-104, 2016.

Y. Boshmaf, I. Muslukhov, K. Beznosov, y M. Ripeanu, “The socialbot network: when bots socialize for fame and money”, en Proceedings of the 27th Annual Computer Security Applications Conference, pp. 93-102, 2011.

A. Murgia, D. Janssens, S. Demeyer, and B. Vasilescu, “Among the Machines: Human-Bot Interaction on Social Q&A Websites” en Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems, pp. 1272-1279, 2016.

G. Penna, “Reality Mining in Twitter”, M.S.c degree, Dept. Comp., Imperial College London, London, GB, 2012.

Z. Chu, S. Gianvecchio, H. Wang, S. Jajosia, “Detecting Automation of Twitter Accounts: Are You a Human, Bot, or Cyborg?“, en IEEE Transactions on Dependable and Secure Computing, vol.9, no. 6, pp. 811-824, 2012.

R. Hanneman, “Introduction to social network methods”, University of California, Riverside, CA, 2005. [En línea]. Disponible: http://faculty.ucr.edu/~hanneman/.

E. F. Moore. “The shortest path through a maze”, en Int. Symp. on Th. of Switching, pp. 285-292, 1959.

Twitter, Inc., "API Rate Limits | Twitter Developers", 2016. [En línea]. Disponible: https://dev.twitter.com/rest/public/rate-limiting.

MINCI, “Venezuela amplía acceso a servicios de Internet”, [En línea]. Disponible: http://minci.gob.ve/2016/06/venezuela-amplia-acceso-a-servicios-de-internet/.

Statista, “Number of active Twitter users in leading markets as of May 2016”, 2016, [En línea]. Disponible: http://www.statista.com/statistics/242606/number-of-active-twitter-users-in-selected-countries/.

S. Smith, “Determining Sample Size: How to Ensure You Get the Correct Sample Size”, 2013, [En línea]. Disponible en: https://www.qualtrics.com/blog/determining-sample-size/.

T. K. Ho, “Random decision forests. In Document Analysis and Recognition”, en Proceedings of the Third International Conference, vol. 1, pp. 278-282, 1995.

M. Mccord y M. Chuah, “Spam detection on twitter using traditional classifiers”, en International Conference on Autonomic and Trusted Computing, pp. 175-186, 2011.

Z. S. Harris, “Distributional structure” Word, vol. 10, no. 2-3, pp. 146-162, 1954.

C. Gini, “Variabilità e mutabilità” Reimprimido en Pizetti, E.; Salvemini, T., eds. 1955. Memorie di metodologica statistica, Libreria Eredi Virgilio Veschi, Roma, IT , 1912.

La arquitectura de un sistema de software es resultado de numerosas decisiones de diseño sobre los compo- nentes y estructura adecuados para proveer las funcionalidades y propiedades requeridas. Las tácticas de arquitectura son una forma destacada de abordar requisitos no-funcionales (RNFs), pero parece existir poco trabajo metodológico que formalice el uso de tácticas para combinar componentes para abordar RNFs específicos. Este artículo describe una revisión sistemática de la literatura (RSL) sobre el uso metodológico combinado de RNFs, tácticas de arquitectura,

ask is what you get: Understanding architecturally significant functional requirements," In 2015 IEEE 23rd International Requirements Engineering Conference (RE), pp. 86–95, 2015.

IV-B.PI-2: ¿Es posible que la relación entre requisitos y arquitectura de software permita seleccionar componentes?

[20]J. Eckhardt, A. Vogelsang, and D. M. Fernández, "Are "non-functional" requirements really non-functional?" 38th International Conference on Software Engineering, At Austin, Texas, 2016.

[21]M. Galster, A. Eberlein, and M. Moussavi, "Comparing methodologies for the transition between software requirements and architectures," Systems, Man and Cybernetics, 2009. SMC 2009. IEEE International Conference on, no. October, pp. 2380–2385, 2009.

[22]J. Cleland-Huang, A. Czauderna, and E. Keenan, "A persona-based approach for exploring architecturally significant requirements in agile projects," Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 7830 LNCS, pp. 18–33, 2013.

[23]R. Ferrari, O. Sudmann, C. Henke, J. Geisler, W. Schafer, and N. Mad- havji, "Requirements and systems architecture interaction in a prototy- pical project: Emerging results," Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 6182 LNCS, pp. 23–29, 2010.

[24]C. Lopez and H. Astudillo, "Explicit Architectural Policies to Satisfy NFRs Using COTS," Satellite Events at the MoDELS 2005 Conference, pp. 227–236, 2005.

[25]H. Astudillo, J. Pereira, and C. López, "Identifying "interesting" compo- nent assemblies for nfrs using imperfect information," Third European Workshop, EWSA 2006, pp. 204–211, 2006.

[26]P. R. Anish, B. Balasubramaniam, J. Cleland-Huang, R. Wieringa, M. Daneva, and S. Ghaisas, "Identifying Architecturally Significant Functional Requirements," 2015 IEEE/ACM 5th International Workshop on the Twin Peaks of Requirements and Architecture, pp. 3–8, 2015. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7184705

[27]P. R. Anish and B.Balasubramaniam, "A knowledge- assisted framework to bridge functional and architecturally significant requirements," Proceedings of the 4th International Workshop on Twin Peaks of Requirements and Architecture - TwinPeaks 2014, pp. 14–17, 2014. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2593861.2593864

[28]J. Cleland-huang and D. Schmelzer, "Dynamically Tracing Non- Functional Requirements through Design Pattern Invariants," Procee- dings of the 2nd International Workshop on Traceability in Emerging Forms of Software Engineering TEFSE, no. JANUARY 2003, 2003.

[29]S. Kim, “A quantitative and knowledge-based approach to choosing security architectural tactics,” Ad Hoc and Ubiquitous Computing, vol. 18, no. 1/2, pp. 45–53, 2015.
[30]M. Mirakhorli, J. Carvalho, J. Cleland-Huang, and P. Mader, “A domain-centric approach for recommending architectural tactics to satisfy quality concerns,” in TwinPeaks 2013 - 3rd International Workshop on the Twin Peaks of Requirements and Architecture (@ICSE 2013). Ieee, jul 2013, pp. 1–8. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6617352
[31]J. Ryoo, P. Laplante, and R. Kazman, “Revising a security tactics hierarchy through decomposition, reclassification, and derivation,” Pro- ceedings of the 2012 IEEE 6th International Conference on Software Security and Reliability Companion, SERE-C 2012, pp. 85–91, 2012.
[32]N. B. Harrison and P. Avgeriou, “How do architecture patterns and tactics interact? A model and annotation,” Journal of Systems and Software, vol. 83, no. 10, pp. 1735–1758, 2010. [Online]. Available: http://dx.doi.org/10.1016/j.jss.2010.04.067
[33]G. Pedraza-Garcia, H. Astudillo, and D. Correal, “A methodological approach to apply security tactics in software architecture design,” 2014 IEEE Colombian Conference on Communications and Computing, COLCOM 2014 - Conference Proceedings, 2014.
[34]M. Mirakhorli and J. Cleland-Huang, “A pattern system for tracing architectural concerns,” Proceedings of the 18th Conference on Pattern Languages of Programs - PLoP ’11, pp. 1–10, 2011. [Online].
Available: http://dl.acm.org/citation.cfm?doid=2578903.2579143 [35]M. Mirakhorli, H.-M. Chen, and R. Kazman, “Mining Big Data for Detecting, Extracting and Recommending Architectural Design Concepts,” 2015 IEEE/ACM 1st International Workshop on Big Data Software Engineering, pp. 15–18, 2015. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7166053
[36]K. Yskout and R. Scandariato, “Change Patterns : Co-evolving Requi- rements and Architecture Katholieke Universiteit Leuven Department of Computer Science Change Patterns : Co-evolving Requirements and Architecture,” no. August, 2010.
[37]R. Noel, G. Pedraza-García, and H. Astudillo, “An exploratory compa- rison of security patterns and tactics to harden systems,” Proceedings of the 11th Workshop on Experimental Software Engineering (ESELAW 2014), ser. CibSE, 2014.
[38]M. Mirakhorli and J. Cleland-Huang, “Detecting, Tracing, and Monitoring Architectural Tactics in Code,” IEEE Transactions on Software Engineering, vol. XX, no. X, pp. 1–1, 2015. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7270338
[39]J. Ryoo, P. Laplante, and R. Kazman, “A methodology for mining security tactics from security patterns,” Proceedings of the Annual Hawaii International Conference on System Sciences, pp. 1–5, 2010.
[40]M. Kassab, G. El-Boussaidi, and H. Mili, “A quantitative evaluation of the impact of architectural patterns on quality requirements,” Software Engineering Research,Management and Applications, vol. 377, pp. 173– 184, 2012.
[41]A. V. Uzunov, K. Falkner, and E. B. Fernandez, “Decomposing distri- buted

software architectures for the determination and incorporation of security and other non-functional requirements," 2013 22nd Australian Software Engineering Conference, pp. 30 – 39, 2013.
[42]L. Liu, Z. Ma, R. Qiu, H. Chen, and W. Shao, "An approach to integrating non-functional requirements into uml design models based on nfr-specific patterns," 2012 12th International Conference on Quality Software, pp. 132 – 135, 2012.
[43]A. Sanchez, A. Aguiar, L. S. Barbosa, and D. Riesco, "Analysing tactics in architectural patterns," Software Engineering Workshop (SEW), 2012 35th Annual IEEE, pp. 32 – 41, 2012.
[44]A. Parvizi-Mosaed, S. Moaven, J. Habibi, and A. Heydarnoori, "Towards a tactic-based evaluation of self-adaptive software architecture availabi- lity," 26th Software Engineering and Knowledge Engineering (SEKE), Vancouver, Canada, 2014.
[45]A. Alebrahim, S. Fassbender, M. Filipczyk, M. Goedicke, and M. Heisel, "Towards systematic selection of architectural patterns with respect to quality requirements," Proceedings of the 20th European Conference on Pattern Languages of Programs, pp. 40:1–40:20, 2015.
[46]M. Kassab and G. El-Boussaidi, "Towards quantifying quality, tactics and architectural patterns relations," The 25th International Conference on Software Engineering and Knowledge Engineering, 2013.
[47]E. B. Fernandez, H. Astudillo, and G. Pedraza-García, "Revisting architectural tactics for security," Software Architecture. Springer In- ternational Publishing, pp. 55–69, 2015.
[48]A. Alebrahim and M. Heisel, "Towards developing secure software using problem-oriented security patterns," Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 8708, pp. 45–62, 2014. [Online]. Available: http://www.scopus.com/inward/record.url?eid=2-s2.0-84921767147&partnerID=40&md5=150d6224e36164aa2e111145368942ef
[49]J. Ryoo, B. Malone, P. A. Laplante, and P. Anand, "The use of security tactics in open source software projects," IEEE Transactions on Reliability, vol. PP, no. 99, pp. 1 – 10, 2015.
[50]A. E. Sabry, "Decision model for software architectural tactics selection based on quality attributes requirements," International Conference on Communications, management, and Information technology (ICC- MIT'2015), 2015.
[51]R. Torres and H. Astudillo, "From early architectural decisions to self- discovery components," CLEI, vol. 58, 2011.
[52]C. López and H. Astudillo, "Multidimensional catalogs for systematic exploration of component-based design spaces," Advanced Software Engineering: Expanding the Frontiers of Software Technology, pp. 32– 46, 2006.
[53]L. Chung and K. Cooper, "Matching , Ranking , and Selecting Com- ponents : A COTS-Aware Requirements Engineering and Software Ar- chitecting Approach," International Workshop on Models and Processes for the Evaluation of COTS Components at ICSE, pp. 41 – 44, 2004.
[54]M. Ibe, M. Vogel, B. Schindler, and A. Rausch, "Create: A co- modeling approach for scenario-based requierements and component- based architectures." International Conference on Software Engineering Advances, pp. 220 – 227, 2013.
[55]A. Vescan and C. S¸ erban, "A fuzzy-based approach for the multilevel component selection problem," Hybrid Artificial Intelligent Systems, pp. 463 – 474, 2016.

IV-B1 Discusión

IV-C. P1-3: ¿En qué forma pueden los RNF's relacionarse con las tácticas de arquitectura?

V. AMENAZAS DE VALIDACIÓN

[56]T. Marewa, J.-S. Leeb, and D.-H. Baea, “Tactics based approach for integrating non-functional requirements in object-oriented analysis and design,” Journal of Systems and Software, vol. 82, no. 10, pp. 1642– 1656, 2009.
[57]M. A. Khan and S. Mahmoodb, “A graph based requirements clustering approach for component selection,” Advances in Engineering Software, vol. 54, pp. 1–16, 2012.
[58]P. Potena, “Composition and tradeoff of non-functional attributes in software systems: Research directions,” Proceedings of the the 6th Joint Meeting of the European Software Engineering Conference and the ACM SIGSOFT Symposium on The Foundations of Software Engineering, pp. 583–586, 2007.
[59]L. Zhu and I. Gorton, “Uml profiles for design decisions and non- functional requirements,” Proceedings of the Second Workshop on SHA- ring and Reusing Architectural Knowledge Architecture, Rationale, and Design Intent, 2007.
[60]C. Tibermacine, S. Sadou, C. Dony, and L. Fabresse, “Component- based specification of software architecture constraints,” Proceedings of the 14th International ACM Sigsoft Symposium on Component Based Software Engineering, pp. 31–40, 2011.
[61]A. Mahmoud and G. Williams, “Detecting, classifying, and tracing non- functional software requirements,” Requirements Engineering, vol. 21, no. 3, pp. 357–381, 2015.
[62]N. Afreen, A. Khatoon, and M. Sadiq, “A taxonomy of software’s non-functional requirements,” Proceedings of the Second International Conference on Computer and Communication Technologies, vol. 1, pp. 47–53, 2016.
[63]M. Yahlali and A. Chouarfia, “Towards a software component assembly evaluation,” IET Software, vol. 9, no. 1, pp. 1 – 6, 2015.
[64]F. Silva, M. Lucena, and L. Lucena, “STREAM-AP : A Process to Systematize Architectur- al Patterns Choice Based on NFR,” in TwinPeaks 2013 - 3rd International Workshop on the Twin Peaks of Requirements and Architecture, no. i, 2013, pp. 27–34.
[65]T. Ruiz-López, J. L. Garrido, S. Supakkul, and L. Chung, “A pattern approach to dealing with nfrs in ubiquitous systems,” 25th International Conference on Advanced Information Systems Engineering, CAiSE, vol. 998, 2013.
[66]D. Ameller, O. Collell, and X. Franch, “Architech: Tool support for nfr- guided architectural decision-making,” 2012 20th IEEE International Requirements Engineering Conference (RE), 2012.
[67]D. Ameller and X. Franch, “How do software architects consider non- functional requirements: A survey,” Conference: Requirements Engi- neering: Foundation for Software Quality, 16th International Working Conference, REFSQ 2010, Essen, 2010.
[68]R. Bouaziz and B. Coulette, “Applying security patterns for component based applications using uml profile,” Computational Science and En- gineering (CSE), 2012 IEEE 15th International Conference on, pp. 186 – 193, 2012.
[69]C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, and A. Wesslén, Experimentation in software engineering. Springer Science and Business Media, 2012.

VI. CONCLUSIONES

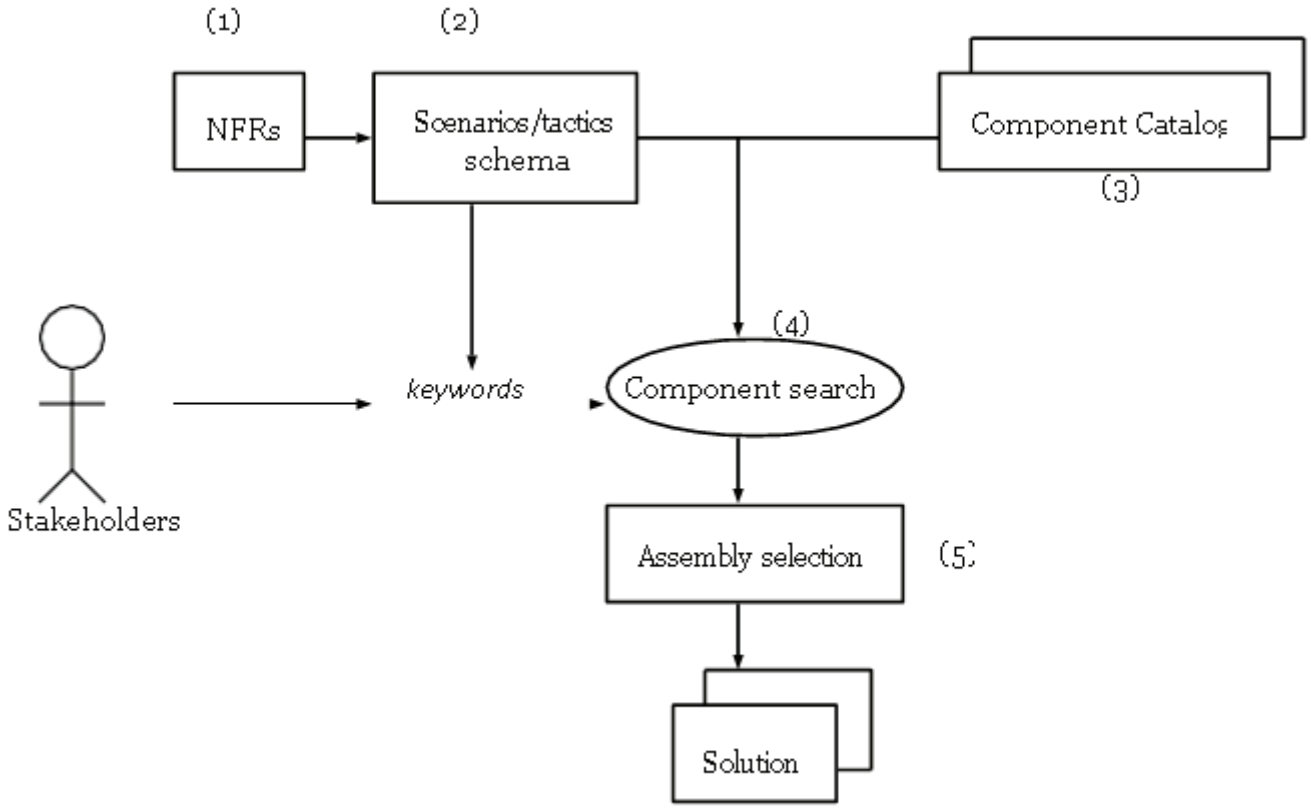
y componentes de software. Se definió preguntas de investigación sobre (1) selección de componentes utilizando RNFs, (2) selección de componentes utilizando la relación entre RNFs y arquitectura, y (3) relación entre RNFs y tácticas. Se encontró 1964 artículos en directorios conocidos, luego filtrados a 59 artículos directamente relevantes al tema de estudio. Un examen detallado de estos artículos ilustra que las tácticas son útiles como concepto intermedio para identificar componentes desde RNFs, pero hasta ahora hay sólo un grupo pequeño de propuestas específicas.

Palabras Clave — Revisión sistemática de literatura, com- ponentes de software, requisitos no funcionales, tácticas de arquitectura.

La arquitectura de software es un enlace entre los objetivos definidos en el negocio por los stakeholders y el resultado final del sistema, el cual en la mayoría de los casos, los requisitos y los intereses de los stakeholders se ven reflejado en los Requisitos No Funcionales (RNFs). Según [1], la complejidad de un sistema de software está determinada en parte por su funcionalidad y por su la globalidad de los requisitos. Estos RNFs juegan un rol crítico durante el desarrollo del sistema, sirviendo como criterio de selección para alternativas de decisiones de diseño hasta la implementación. Es en este punto donde Chen et al. [2] ha puesto en debate el concepto de requisitos de arquitectura significativos, con el objetivo de analizar si realmente todos los RNFs son realmente útiles al momento de tomar decisiones de arquitectura. Hasta el momento, una de las mejores maneras utilizadas para la toma de decisiones de diseño en sistemas de software son las tácticas de arquitectura. Una táctica de arquitectura es una decisión de diseño que influencia el cumplimiento de satisfacer los atributos de calidad obtenidos desde RNFs [3].

Por otro lado, la ingeniería de software basada en com- ponentes (CBSE, siglas en inglés) promueve el desarrollo y construcción de sistemas de software desde componentes de software ya existentes. El desarrollo de componentes se puede ver como una entidad reusable que va acorde a la evolución del sistema y adaptable a las necesidades de los stakeholders. La motivación del uso de la ingeniería de software basada en componentes nace desde la naturaleza del negocio con el objetivo de incrementar la eficencia y efectividad, el tiempo de salida de productos de software al mercado, reducir costos, entre otros [4].

Dicho lo anterior, nos hemos percatado



que para los ar- quitectos de software y los stakeholders las decisiones que se toman en un sistema de software son muy críticas. Por un lado, el arquitecto de software se preocupa que cualquier decisión de diseño que se tome no afecte las funcionalidades del sistema, pero por otro lado, los stakeholders ponen más énfasis en que las necesidades del negocio se lleven a cabo los más pronto posible. Es aquí donde la correcta selección de componentes debe ser un factor relevante a la hora de considerar decisiones de diseño a nivel de arquitectura de software, satisfaciendo las necesidades de los stakeholders y lograr los atributos de calidad definidos en los requisitos.

Con el objetivo de abordar esta interrogante, este artículo describe una Revisión Sistemática de la Literatura (RSL) usando las directrices de Kitchenham et al. [5] para conocer los trabajos, perspectivas e iniciativas en la comunidad donde se aborde la selección de componentes desde RNFs mediante el uso de tácticas de arquitectura. Hemos desarrollado un protocolo de revisión obteniendo como resultado final 59 artículos que nos han ayudado a comprender el contexto actual de nuestra interrogante.

El resto del artículo se organiza de la siguiente manera: la sección II describe aquellos trabajos que han abordado nuestra interrogante, desde cualquier punto de vista, mediante una RSL realizada previamente; la sección III explica la meto- dología usada en nuestra revisión describiendo las preguntas de investigación, el proceso de búsqueda, los criterios de inclusión/exclusión y las estrategias de extracción y síntesis de datos; la sección IV expone análisis y resultados de la ejecución del protocolo respondiendo a la preguntas de investigación; la sección V

AGRADECIMIENTOS

REFERENCIAS

detalla las amenazas de validez; y la sección VI presenta conclusiones y trabajo futuro.

En esta sección discutiremos sobre los trabajos que existen en la comunidad que se relacionan con búsquedas sistemáticas de literatura sobre la obtención de componentes de software desde RNFs mediante el uso de tácticas de arquitectura. En primera instancia, no hemos podido encontrar algún trabajo que proponga una revisión sistemática sobre nuestro objeto de estudio. No obstante, si dividimos los temas, encontramos el trabajo de Vale et al. [4], donde realiza un estudio muy profundo sobre la evolución de CBSE. Como conclusiones del estudio, los autores se percatan que el interés en la comunidad sobre el reuso de componentes ha ido en aumento debido a las nuevas exigencias del mercado en relación al desarrollo de aplicaciones en el corto tiempo. Por otro lado, un estudio realizado por Morisio et al. [6] refleja, desde la práctica, 15 proyectos en donde se han utilizado el enfoque de componentes de software, llegando a la conclusión que los componentes tienen un gran impacto en definiciones de alto de nivel, específicamente en la integración y testing. Por último, hemos analizado el trabajo de Breivold et al. [7] donde se aborda la evolución del software. Para lo anterior, los autores realizan una revisión sistemática de la literatura para ver cómo afectan los requisitos en la evolución de la arquitectura de software en distinto niveles.

En resumen, en la literatura hemos encontrado trabajos que abordan desde diferentes puntos vista la interrogante de componentes y RNFs, pero la inclusión de tácticas de arquitectura es un tema que no ha sido abordado aún. Lo anterior nos motiva a realizar nuestra RSL con el objetivo de ver qué propuestas existen en la literatura que respondan en un cierto grado a nuestra interrogante de investigación.

En esta sección vamos a describir el enfoque metodológico utilizado para realizar la RSL. Como resultado final, hemos obtenido un número significativo de artículos relevantes, los cuales serán resumidos en la siguiente sección.

Para realizar la revisión, hemos adoptado las directrices definidas por Kitchenham et al. [5], para conducir RSL en ingeniería de software. Estas directrices son utilizadas para identificar, evaluar e interpretar todos los trabajos relevantes en base a una pregunta de investigación, área de interés o un fenómeno de interés. Siguiendo las directrices, se debe definir un protocolo de revisión para la RSL y éste se debe seguir durante la revisión. El protocolo que nosotros hemos definido contiene la creación de la preguntas de investigación, el proceso de búsqueda, definición de criterios de inclusión y exclusión, estrategia de extracción y síntesis de datos y la ejecución. La Figura 1 resume lo anteriormente mencionado. La ejecución del protocolo fue realizada entre las fechas 13 de Enero y 20 de Julio del 2016 en las siguientes librerías digitales: Scopus, IEEE Xplorer, SciendeDirect y ACM. La razón por la cual se seleccionaron estas librerías es debido a su gran potencial y relevancia en la comunidad en relación a las conferencias y revistas que poseen.

Figura 1: Protocolo de búsqueda