Universidad Tecnológica de El Salvador

Compilación de investigaciones de tecnología 2017

Extracción de conocimiento a partir de texto
Investigador:
Ronny Adalberto Cortez Reyes

Aulas conectadas: sistema IoT para el registro de asistentes
Investigadores:
Omar Otoniel Flores Cortez
Verónica Idalia Rosa Urrutia

Colección Investigaciones | 78

Universidad Tecnológica de El Salvador

FOCUS DATA

Compilación de investigaciones de tecnología 2017
Extracción de conocimiento a partir de texto

Investigador:
Ronny Adalberto Cortez Reyes

Esta investigación fue subvencionada por la Universidad Tecnológica de El Salvador. Las solicitudes de información, separatas y otros documentos relativos a este estudio pueden hacerse a la siguiente dirección postal: Universidad Tecnológica de El Salvador, edificio *Dr. José Adolfo Araujo Romagoza*, Vicerrectoría de Investigación y Proyección Social, Dirección de Investigaciones, calle Arce y 19.ª avenida Sur, 1045, o a ronny.cortez@utec.edu.sv

San Salvador, 2018

005.52
C678c Cortez Reyes, Ronny Adalberto, 1987-
Compilación de investigaciones de tecnología / Ronny Adalberto
sv Cortez Reyes, Omar Otoniel Flores Cortez, Verónica Idalia Rosa
Urrutia. -- 1ª ed. -- San Salvador , El Salv. : Universidad Tecnológica
de El Salvador (UTEC), 2018.
228 p. : il. ; 23 cm. -- (Investigaciones ; nº. 78)

ISBN 978-99961-86-01-1

1. Procesamiento de palabras. 2. Procesamiento de textos.
(Computadores electrónicos). 3. Análisis de información. I. Flores
Cortez, Omar Otoniel, 1978-, coaut. II Rosa Urrutia, Verónica
Idalia, 1976-, coaut. III. Título.

BINA/jmh

PRIMERA EDICIÓN
150 ejemplares
Agosto, 2018

Impreso en El Salvador
Por Tecnoimpresos, S.A. de C.V.
19 Av. Norte, nº. 125, San Salvador, El Salvador
Tel.:(503) 2275-8861 • gcomercial@utec.edu.sv

Dedicatoria

Este trabajo está dedicado a mi madre y a mi hermano, que me han apoyado en todo momento y siempre me han animado a seguir adelante. También lo dedico a la memoria de mi padre y de mi abuelo, quienes me dieron los consejos y la educación que me han permitido alcanzar muchas metas y objetivos.

Agradecimientos especiales a Kevin Danilo Barrientos Larin por su gran aporte al desarrollo de esta investigación con su trabajo y constante dedicación.

Ìndice

Figuras

Tablas

Resumen

El trabajo "Extracción de conocimiento a partir de texto" tiene como objetivo aplicar técnicas de *data mining* para, a partir de un conjunto grande de textos, obtener información que permita extraer datos útiles para generar conceptos, ontologías y definiciones.

Las fuentes de información consultadas son artículos descargados de Scopus compuestas por *title, abstract, keywords* y *author keywords*; y de un conjunto de noticias de noviembre de 2017, compuestas por titular, resumen y cuerpo, proporcionado por la unidad "Focus Data" de *El Diario de Hoy* para la etapa de pruebas; y aprendizaje en el uso de *Word2Vec* y un conjunto de tuits de cuentas pertenecientes a algunos candidatos seleccionados basados en su trascendencia mediática, manteniendo representatividad tanto de candidatos a la Asamblea Legislativa como a concejos municipales; y con un balance entre los contendientes de diferentes partidos políticos para hacer análisis que incluyen relación entre las palabras, frecuencia de aparición en los textos y representación visual, tales como *heatmap*, dendrogramas, clústeres y nubes de palabras para mostrar los resultados obtenidos.

Las pruebas realizadas fueron pre-procesamiento de texto, aplicación de los modelos sobre cada una de las partes de los artículos, uso de nubes de palabras y otros métodos gráficos para determinar si el modelo es capaz de extraer información de útil y de calidad. Para los textos extraídos de los tuits, se hizo una limpieza propia del contenido.

Teniendo en cuenta nuestras necesidades y el contexto de las pruebas, los mejores resultados con *Word2Vec* se han obtenido utilizando el modelo que se ha generado sin aplicar *stemming* sobre los encabezados de las noticias.

Mediante los tuits, se pudo extraer información de los candidatos analizados, de sus propuestas e ideas; algunos de ellos con mayor movimiento en las redes sociales, tanto en cantidad de tuits como en el número de palabras contenidas en cada uno. Entre los usos de las cuentas, fueron de información, propuestas y noticias.

Palabras clave: *Word2Vec*, minería de textos, preprocesamiento de texto, análisis de clústeres, nube de palabras.

1. Introducción

1.1 *Objetivos*

1.1.1 *General*:

Aplicar técnicas de *data mining* para, a partir de un conjunto grande de textos, obtener una ontología que describa los conceptos de los que se habla en esos textos.

1.1.2 *Específicos:*

- Analizar y comprender el funcionamiento de *Word2Vec*.
- Desarrollar las bases de textos, a partir de artículos de Scopus, necesarias para las pruebas y extracción de conocimiento.
- Describir e interpretar los resultados obtenidos a partir de los modelos creados con Word2Vec y las ontologías conceptuales de los textos.
- Desarrollar una base de datos a partir de un conjunto de textos extraídos de la red social Twitter.
- Procesar y analizar los textos para extraer información.
- Representar los resultados mediante gráficos, tales como nubes de palabras, barras y relación de palabras.

1.2 *Problema identificado*

En los últimos años se ha generado una gran cantidad de textos en formato digital en diferentes plataformas, como por ejemplo redes sociales, correos, publicaciones científicas, foros, comentarios, periódicos, esto debido a que el número de usuarios con acceso a internet es cada vez mayor y con diferentes tipos de dispositivos,[1] y esta tendencia puede mantenerse debido a las nuevas tecnologías.[2]

1 https://ourworldindata.org/internet/
2 http://www.obs-edu.com/int/noticias/estudio-obs/en-2020-mas-de-30-mil-millones-de-dispositivos-estaran-conectados-internet

Es probable que continúe el aumento de internet y el crecimiento continuo del acceso en todo el mundo mediante diferentes tecnologías, y así cambie la comunicación y la forma en que accedemos a la información (Murphy & Roser, 2018).

Según el estudio elaborado por OBS (Open Broadcaster Software), el volumen de datos generados en 2014 se ha multiplicado. En un minuto, en internet se generan 4.1 millones de búsquedas en Google, se escriben 347 mil tuits, se comparten 3.3 millones de actualizaciones en Facebook, se suben 38 mil fotos a Instagram, se visualizan 10 millones de anuncios, se suben más de 100 horas de vídeo a YouTube, se escuchan 32 mil horas de música en *streaming*, se envían 34.7 millones de mensajes instantáneos por internet o se descargan 194 mil apps. En total, en un minuto se transfieren más de 1.570 terabytes de información.

El texto digital se ha convertido en una forma de intercambio de información y ha crecido a tal punto que cada vez es más difícil poder procesarla; no solamente localizarla rápida y eficientemente, sino también la extracción de conocimiento para ser utilizado en la toma de decisiones.

La extracción de conocimiento a partir de texto que describa los conceptos de los que se habla puede ser utilizada de diferentes maneras, como por ejemplo en el análisis de sentimientos, detección de riesgos, opiniones políticas, entre otros.

1.3 *Justificación*

Ante el incremento exponencial de datos en forma de texto y la dificultad de su análisis por medio de métodos tradicionales supervisados por humanos y que requieren una gran cantidad de tiempo y expertos que muchas veces son escasos, se hace necesario el conocimiento e implementación de técnicas de minería de datos que permitan la obtención de ontologías para describir los conceptos con el menor consumo de tiempo, en forma eficiente y con una reducción de costos.

Mediante técnicas de *data mining* se puede procesar grandes cantidades de texto sin la necesidad de que un experto dedique todo su tiempo a dicho proceso, obteniendo resultados en mucho menos tiempo. Para el proyecto, se ha decidido utilizar Word2Vec debido a sus ventajas, ya que es eficiente de entrenar proceso que consiste en introducir información al algoritmo para la creación de un modelo

preciso que permita responder preguntas correctamente la mayor parte del tiempo, está fácilmente disponible en línea con código y modelos preentrenados (Jurafsky & H. Martin, 2017).

Otra de las ventajas es que se encuentran implementaciones en diferentes lenguajes de programación y tiene muchas aplicaciones, tales como funciones de agrupación, clasificación de sentimientos, extracción de características semánticas entre palabras (Zhang, Xu, Su, & Xu, 2015).

Los trabajos anteriores para encontrar vectores de palabras basados en redes neuronales eran computacionalmente caros (memoria RAM, tiempo, cálculos en procesador).

Las densas representaciones vectoriales de palabras aprendidas por Word2Vec han demostrado notablemente que llevan significados semánticos y son útiles en una amplia gama de casos de uso, que van desde el procesamiento del lenguaje natural hasta el análisis de datos de flujo de la red. Quizás la propiedad más asombrosa de esto es que de alguna manera estas codificaciones de vectores captan con eficacia los significados semánticos de las palabras (Meyer, 2016).

Entre las redes sociales, los datos de Twitter constituyen una fuente rica disponible para capturar información sobre cualquier tema imaginable. Estos datos se pueden utilizar en diferentes aplicaciones, como encontrar tendencias relacionadas con una palabra clave específica, medir el sentimiento de la marca y recopilar comentarios sobre nuevos productos y servicios (Moujahid, 2014).

En el caso de las opiniones políticas, la red social Twitter nos permite ver de lo que se está hablando, conocer sobre historias y diferentes puntos de vista y extraer una gran cantidad de textos, ya sea de cuentas o de tuits. Todo ello por medio de una plataforma de desarrollo[3] que nos da acceso a la interfaz de programación de aplicaciones (API, siglas del inglés) para hacer los análisis de forma práctica y eficiente, que junto con las librerías disponibles para R adquiere un gran potencial.

Dada la inmensa cantidad de datos, una de las principales dificultades a las que nos enfrentamos con el análisis y los resultados obtenidos es la de su fiabilidad ya que se vuelve una tarea compleja fácilmente susceptible a fallos o variaciones basada en la experiencia de los especialistas y en el tiempo con el que se cuente.

3 https://developer.twitter.com/

2. Fundamentación: estado de la técnica y conceptos

En esta sección se describen el estado de la técnica sobre la extracción de ontologías y el uso de *Word2Vec* en el análisis de textos, las etapas involucradas, algunos conceptos importantes, modelos y parámetros y las técnicas utilizadas para la representación visual de los resultados, además se explica el funcionamiento de la API de Twitter.

La extracción de conocimiento a partir de textos es una tarea compleja, ya que en el análisis de textos deben ser tomados en cuenta muchos aspectos, tales como sarcasmo, variedad en la escritura, experiencias, continuidad en los diálogos y el hecho de que muchas veces no se sigue un formato común para la presentación de la información; aspectos que pueden ser complicados de identificar para la inteligencia artificial.

2.1 *Minería de textos*

En la actualidad, existe una gran cantidad de información digital en forma de texto, por ejemplo, en periódicos, foros, redes sociales, correos electrónicos, revistas, libros y otros más.

La proliferación del uso de dispositivos computacionales y de comunicación para la producción de información digital, y en particular en la producción de documentos textuales, ha generado la necesidad de desarrollar métodos, algoritmos y sistemas capaces de realizar el procesamiento automatizado de datos textuales estructurados, semiestructurados y no estructurados para su organización y consulta, y con ello el surgimiento de áreas de estudio de la información como la minería de texto (Contreras Barrera, 2014).

La minería de textos es el proceso de analizar colecciones de materiales de texto con el objeto de capturar los temas y conceptos clave y descubrir las relaciones ocultas y las tendencias existentes sin necesidad de conocer las palabras o los términos exactos que los autores han utilizado para expresar dichos conceptos. Una recuperación precisa de la información y su almacenamiento supone un reto importante, pero la extracción y administración de contenido de calidad, de terminología y de las relaciones contenidas en la información son procesos cruciales y determinantes (“IBM Knowledge Center”, s. f.).

La minería de texto surge como un enfoque particular del proceso de descubrimiento de conocimiento, específicamente, orientado al descubrimiento en fuentes textuales y no estructuradas (Rodríguez Blanco, Cuevas, & J, 2013). Se puede definir como "un proceso de descubrimiento de conocimientos potencialmente útiles, y no explícito, en una colección de textos, a partir de la identificación y exploración de patrones interesante" (Feldman, 1998).

Entre las herramientas desarrolladas para extraer información, y que intentan inferir relaciones que no aparecen de forma implícita en esa información, pueden citarse los siguientes: TextAnalyst, twURL, T-LAB, LexiQuest Mine, Text Miner, Weka y R (Botta-Ferret & Cabrera Gato, 2007).

Para cada artículo de texto, la minería de textos basada en la lingüística devuelve un índice de conceptos e información acerca de estos. De acuerdo con IBM Knowledge Center, esta información estructurada y desglosada puede combinarse con otros orígenes de datos para abarcar preguntas como las que a continuación se presentan.

- ¿Qué conceptos aparecen juntos?
- ¿A qué otras cosas están vinculados?
- ¿Qué categorías de nivel superior pueden crearse a partir de la información extraída?
- ¿Qué es lo que predicen los conceptos o las categorías?
- ¿Cómo predicen el comportamiento los conceptos o las categorías?

La combinación de minería de textos y minería de datos ofrece un punto de vista más amplio que el de los datos estructurados o el de los datos no estructurados por separado. Para IBM Knowledge Center, este proceso suele incluir los pasos siguientes:

- Identificar el texto en el que se va a realizar la minería. Preparar el texto para el proceso de minería. Si el texto existe en varios archivos, guarde los archivos en una misma ubicación. Para las bases de datos, determine el campo que contiene el texto.
- Minar el texto y extraer datos estructurados. Aplicar los algoritmos de minería de textos al texto de origen.

- Crear modelos de categoría y concepto. Identificar los conceptos clave para crear categorías. El número de conceptos que se devuelve de los datos no estructurados suele ser muy alto. Identificar los mejores conceptos y categorías para puntuar.
- Analizar los datos estructurados. Emplear técnicas de minería de datos convencionales, como el clúster, la clasificación y el modelado predictivo, con el objeto de descubrir las relaciones entre los conceptos. Fusionar los conceptos extraídos con otros datos estructurados para predecir comportamientos futuros basados en los conceptos.

Rochina (2017) afirma que la minería de textos comprende tres actividades fundamentales:

- Recuperación de la información: consiste en seleccionar los textos pertinentes.
- Extracción de la información incluida en esos textos mediante el procesamiento del lenguaje natural: hechos, acontecimientos, datos clave, relaciones entre ellos, etc.
- Minería de datos para encontrar asociaciones entre los datos clave previamente extraídos de entre los textos.

Estas actividades se dividen en las siguientes tres etapas fundamentales:

- Etapa de preprocesamiento: en esta etapa, los textos se transforman en algún tipo de representación estructurada o semiestructurada que facilite su posterior análisis. Es decir, el primer paso dentro de la minería de texto sería definir el conjunto (corpus) de documentos. Estos documentos deben ser representativos y seleccionarse aleatoriamente o mediante algún método de muestreo probabilístico. Se debe evitar en esta etapa la duplicación de documentos dentro del corpus. Con el corpus seleccionado y estructurado, debemos reconocer los *tokens* (unidades gramaticales más pequeñas, es decir, toda palabra que posee forma o estructura, una función y un significado), lo que implica representar el texto como una lista de palabras mediante una representación vectorial.

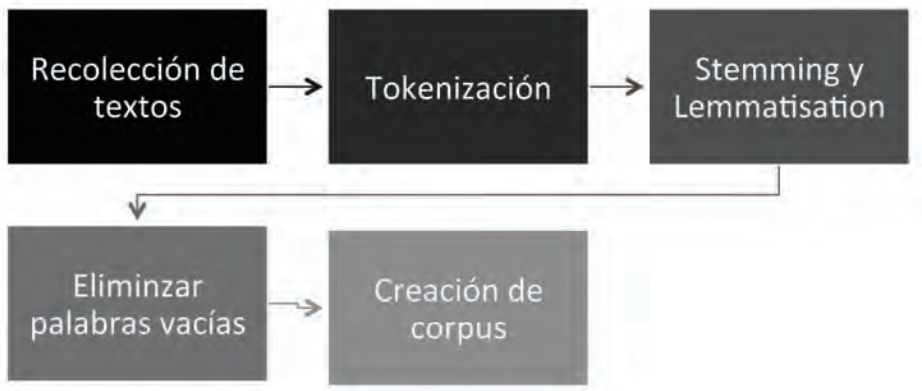
- Etapa de descubrimiento: en esta etapa, las representaciones internas se analizan con el objetivo de descubrir en ellas algunos patrones interesantes o nueva información.
- Etapa de visualización: es la etapa en la que los usuarios pueden observar y explorar los resultados.

Se puede notar que la descripción de las etapas anteriores tiene mucha relación con las técnicas clásicas utilizadas en la minería de texto.

2.2 *Preprocesamiento del texto*

En la etapa de preprocesamiento de texto, se debe realizar una serie de tareas teniendo en cuenta que el orden en que se apliquen puede variar de acuerdo con nuestras necesidades y que no siempre se usarán todas. Una posible opción es la mostrada en la figura 1. lemmatisation

Figura 1. Fases del preprocesamiento.



Fuente: diseño propio.

2.2.1 *Tokenización*

Es el proceso de separar una cadena de texto en palabras, frases, símbolos u otros elementos significativos llamados *tokens*. El objetivo de la *tokenización* es la exploración de las palabras en una oración. La

lista de tokens se convierte en entrada para el procesamiento posterior, como el análisis sintáctico o la minería de texto. La tokenización es útil tanto en la lingüística (donde es una forma de segmentación de texto) como en la informática, donde forma parte del análisis léxico (Kannan & Gurusamy, 2014).

La tokenización es muy importante, y casi siempre es de los primeros pasos. Esto se debe al hecho de que, a pesar de parecer algo sencillo, puede verse afectada por las decisiones que tomemos, por ejemplo: cómo tratar las letras mayúsculas, las siglas, los números, los signos de puntuación y algunas cadenas alfanuméricas que pueden verse afectadas.

2.2.2 Stemming *y* lemmatisation

Ambos son los métodos básicos de procesamiento de texto. Su objetivo es reducir las formas inflexionales y, a veces derivadas, las formas relacionadas de una palabra con una forma de base común (TextMiner, 2014).

Stemming es el proceso de combinar las formas variantes de una palabra en una representación común, la raíz. Por ejemplo, las palabras *presentación*, *presentado*, *presentación* podrían reducirse a una representación común: *present*. Este es un procedimiento ampliamente utilizado en el procesamiento de texto para la recuperación de información (IR), basado en la suposición de que plantear una consulta con el término *presentación* implica un interés en los documentos que contienen las palabras *presentación* y *presentado* (Kannan & Gurusamy, 2014).

Durante el proceso de *stemming* se pueden dar dos tipos de errores: el primero es que dos palabras con diferentes raíces sean llevadas a una misma y el segundo es que dos palabras que deberían tener una misma raíz no sean identificadas y sean siempre consideradas como dos palabras independientes.

Lemmatisation es el proceso que crea el conjunto de lemas (encabezado de una entrada de diccionario) de una base de datos léxica. Se concibe partiendo de las palabras de texto que se encuentran en un corpus y que conducen a las entradas de diccionario de lemas (Lehmann, 2017).

También lo podemos definir como el proceso mediante el cual las palabras de un texto que pertenecen a un mismo paradigma flexivo

o derivativo son llevadas a una forma normal que representa a toda una clase o conjunto de palabras por ejemplo si tomamos las palabras saltar, saltando, saltará y saltaron serán agrupadas como saltar al aplicar el proceso de *lemmatisation*. Esta forma normal, llamada *lema*, es típicamente la palabra utilizada como entrada en los diccionarios de lengua. Su importancia radica en el hecho de que, para acceso por contenido a bases de datos textuales, permite superar las limitaciones de una búsqueda simple de *strings*, haciendo que relaciones ocultas por la variabilidad morfológica de las palabras queden manifiestas. La *lemmatisation* mejora por lo tanto el recubrimiento (*recall*) aunque pueda ser a expensas de la precisión cuando diferentes conjugaciones morfológicas de una misma raíz están asociadas con conceptos distintos (Bassi, A., 2001).

La *lemmatisation* toma en consideración el análisis morfológico de las palabras. Para ello es necesario tener diccionarios detallados que el algoritmo puede recordar para vincular el formulario a su lema.

La principal diferencia es que un lema es la forma base de todas sus formas inflexionales. Sin embargo, la raíz puede ser la misma para las formas flexionales de diferentes lemas, proporcionando entonces ruido a nuestros resultados de búsqueda(García, Cabanilles, & Ramirez, 2017).

2.2.3 *Palabras vacías* (stop words)

Muchas palabras en los documentos se repiten frecuentemente, pero son esencialmente sin sentido, ya que se utilizan para unir palabras en una oración. Se entiende comúnmente que las palabras vacías no contribuyen al contexto o al contenido de los documentos textuales. Debido a su alta frecuencia de ocurrencia, su presencia en la minería de textos presenta un obstáculo para entender el contenido de los documentos (Kannan & Gurusamy, 2014).

En varias aplicaciones de la minería de datos, nos encontraremos con el problema de categorización de documentos (secuencias de palabras) por su tema. Típicamente, los temas se identifican encontrando las palabras especiales que caracterizan documentos sobre ese tema. La primera suposición podría ser que las palabras que aparecen con mayor frecuencia en un documento son las más significativas. Sin embargo, esa intuición es exactamente opuesta a la

verdad. Las palabras más frecuentes seguramente serán las palabras comunes, tales como *el* o *y*, que ayudan a construir ideas, pero no tienen ningún significado en sí mismas. De hecho, los varios cientos de palabras más comunes en inglés (palabras vacías) a menudo se quitan de los documentos antes de cualquier intento de clasificarlos (Leskovec, Rajaraman, & Ullman, 2014).

De hecho, los indicadores del tema son palabras relativamente raras. Sin embargo, no todas las palabras raras son igualmente útiles como indicadores. Hay ciertas palabras que aparecen raramente en una colección de documentos, pero no nos dicen nada útil. Por otro lado, algunas de estas palabras raras pueden decirnos algo sobre un tema en específico.

Las palabras vacías son las más comunes encontradas en cualquier lenguaje natural, que llevan muy poco o ningún contexto semántico significativo en una oración. Solo tienen importancia sintáctica que ayuda en la formación de la oración (Raulji & Saini, 2016).

No existe una lista de palabras vacías única, pues puede variar de acuerdo con el idioma, área de trabajo o palabras propias del momento en que se está trabajando con texto. Algunas veces, el proceso de eliminar las palabras vacías puede reducir los datos de texto y mejorar el rendimiento del sistema (Kannan & Gurusamy, 2014). Sin embargo, esto no siempre es así, ya que, si bien es cierto que puede reducir el tamaño de los textos y con ello la carga de procesamiento, también puede hacer que parte del texto pierda sentido o significado.

Por ejemplo, la frase "internet de las cosas" incluye dos palabras vacías que son *de* y *las* sin las cuales el significado será completamente diferente porque nos quedaríamos con dos palabras *internet* y *cosas*.

2.3 *Ontologías*

2.3.1 *Definición*

En la actualidad, coexisten dos usos del término *ontología*, que corresponden a dos ramas del saber, y, por tanto, le atribuyen características y propiedades distintas. El término *ontología* se origina en el campo de la Filosofía y de la Epistemología. Como ciencia, la Ontología es una rama de la Metafísica que se ocupa del estudio de la naturaleza de la existencia, de los seres y de sus propiedades transcendentales. En

Filosofía, por tanto, una ontología se considera como una explicación sistemática de la existencia (Pérez Hernández, 2002).

Se puede definir como la rama de la Filosofía que se ocupa de la naturaleza y organización de la realidad, es decir, de lo que "existe". En el campo de la inteligencia artificial, "lo que existe es aquello que puede ser representado" (Grela, Sauri, & Sellés, 2004).

En el contexto de las ciencias informáticas y de la información, una ontología define un conjunto de representaciones primitivas con las que modelar un dominio de conocimiento o discurso. Las representaciones primitivas son típicamente clases (o conjuntos), atributos (o propiedades) y relaciones (o relaciones entre miembros de la clase). Las definiciones de las representaciones primitivas incluyen información acerca de su significado y sus restricciones sobre su aplicación lógicamente consistente (Gruber, 2009).

Otras definiciones de ontología son las siguientes:

- Una ontología es una especificación explícita de una conceptualización, es decir, proporciona una estructura y contenidos de forma explícita que codifica las reglas implícitas de una parte de la realidad, independientemente del fin y del dominio de la aplicación en el que se usarán o reutilizarán sus definiciones (Grela et al., 2004).
- Una ontología define el vocabulario de un área mediante un conjunto de términos básicos y relaciones entre dichos términos, así como las reglas que combinan términos y relaciones que amplían las definiciones dadas en el vocabulario (Grela et al., 2004).
- En un sentido muy general, las ontologías son para la inteligencia artificial recursos construidos que permiten representar el conocimiento compartido y común sobre algo. Esta posibilidad de generar recursos compartibles, y la consecuencia natural de intercambiar la información en ellos almacenada, es lo que provoca que un concepto como el de *ontología* (antes privativo de la inteligencia artificial) se filtre en los entornos de trabajo de otros ámbitos, y en especial en lo relativo a la gestión de los recursos y herramientas del entorno digital (Arano, 2005).

2.3.2 *Tipos de ontologías*

Steve et al. (1998a: 1) distinguen tres tipos fundamentales de ontologías:

- Ontologías de un dominio, en las que se representa el conocimiento especializado pertinente de un dominio o subdominio, como la Medicina, las aplicaciones militares, la Cardiología o, en nuestro caso particular, la Oncología.
- Ontologías genéricas, en las que se representan conceptos generales y fundacionales del conocimiento como las estructuras parte/todo, la cuantificación, los procesos o los tipos de objetos.
- Ontologías representacionales, en las que se especifican las conceptualizaciones que subyacen a los formalismos de representación del conocimiento, por lo que también se denominan metaontologías (*meta-level* o *top-level ontologies*).

Según su dependencia y relación con una tarea específica desde un punto de vista, Guarino (1998a: 9) clasifica las ontologías en:

- Ontologías de alto nivel o genéricas: describen conceptos más generales.
- Ontologías de dominio: describen un vocabulario relacionado con un dominio genérico.
- Ontologías de tareas o técnicas básicas: describen una tarea, actividad o artefacto, por ejemplo, componentes, procesos o funciones.
- Ontologías de aplicación: describen conceptos que dependen tanto de un dominio específico como de una tarea específica, y generalmente son una especialización de ambas.

En el ámbito del procesamiento del lenguaje natural, las ontologías se están empleando para construir representaciones independientes de la lengua que puedan servir de punto de encuentro entre dos o más lenguas naturales. En este sentido, la ontología se considera como el repositorio de conceptos que establecen conexiones entre los símbolos de una lengua y sus referentes en el mundo o submundo (UoD) que se contempla. Es posible establecer un claro paralelismo entre la utilidad que las ontologías presentan para la traducción

automática y para la gestión terminológica, ya que, en los dos casos, las ontologías se crean para representar formal y explícitamente la estructura conceptual del lenguaje, aunque la finalidad última sea diferente (Pérez Hernández, 2002).

2.3.3 *Partes de las ontologías*

Según Gruber, las ontologías se componen de lo siguiente:

- Conceptos: son las ideas básicas que se intentan formalizar. Los conceptos pueden ser clases de objetos, métodos, planes, estrategias, procesos de razonamiento, etc.
- Relaciones: representan la interacción y el enlace entre los conceptos de un dominio. Suelen formar la taxonomía del dominio. Por ejemplo: subclase-de, parte-de, parte-exhaustiva-de, conectado-a, etc.
- Funciones: son un tipo concreto de relación donde se identifica un elemento mediante el cálculo de una función que considera varios elementos de la ontología. Por ejemplo, pueden aparecer funciones como asignar-fecha, categorizar-clase, etc. Aunque las funciones no siempre son aspectos calculables, sí pueden ser estructuras formadas de cierta relación que pueden ser usada en lugar de un término individual en una declaración.
- Instancias: se utilizan para representar objetos determinados de un concepto.
- Reglas de restricción o axiomas: son teoremas que se declaran sobre relaciones que deben cumplir los elementos de la ontología. Por ejemplo: "Si *A* y *B* son de la clase *C*, entonces *A* no es subclase de *B*", "Para todo *A* que cumpla la condición *B1*, *A* es *C*", etc. Los axiomas, junto con la herencia de conceptos, permiten inferir conocimiento que no esté indicado explícitamente en la taxonomía de conceptos.

2.4 WORD2VEC *para análisis de textos*

Hemos hablado sobre la minería de texto y sus funciones y también hemos definido el término *ontología*.

Word2Vec es un grupo de modelos relacionados que se utilizan para producir inserciones de palabras. Estos modelos son redes neurales de dos capas que son entrenadas para reconstruir contextos lingüísticos de palabras. Word2Vec toma como su entrada un gran corpus de texto y produce un espacio vectorial, típicamente de varios cientos de dimensiones, con cada palabra única en el corpus, siendo asignado un vector correspondiente en el espacio. Los vectores de palabras se sitúan en el espacio vectorial de manera que las palabras que comparten contextos comunes en el corpus se encuentran muy próximas entre sí en el espacio (Mikolov, Chen, Corrado, & Dean, 2013).

Así que, en lugar de una correlación uno a uno entre un elemento del vector y una palabra, la representación de una palabra se extiende a través de todos los elementos del vector, y cada elemento del vector contribuye a la definición de muchas palabras (Colyer, 2016).

En muchas tareas de procesamiento de lenguaje natural, las palabras son representadas a menudo por su puntaje TF-IDF.[4]

Si bien estas puntuaciones nos dan una idea de la importancia relativa de una palabra en un documento, no nos dan ninguna idea de su significado semántico. Word2Vec es el nombre atribuido a una clase de modelos de redes neuronales que, dado un corpus de entrenamiento no etiquetado, producen un vector para cada palabra en el corpus que codifica su información semántica (Minnaar, 2015).

Un vector de palabras es simplemente un vector de pesos. En una simple codificación 1-of-N cada elemento del vector está asociado con una palabra en el vocabulario. La codificación de una palabra dada es simplemente el vector en el que el elemento correspondiente se establece en uno, y todos los demás elementos son cero (Colyer, 2016).

Estos vectores son importantes por las dos razones siguientes:

- Podemos medir la semejanza semántica entre dos palabras, calculando la semejanza de coseno entre sus correspondientes vectores de palabras.
- Podemos utilizar estos vectores de palabras como características de varias tareas supervisadas de PNL, como la clasificación de

4 http://www.tfidf.com/

documentos, el reconocimiento de entidades nombradas y el análisis de sentimientos. La información semántica contenida en estos vectores los convierte en potentes funciones para estas tareas.

Una palabra es la unidad básica de datos discretos, definida como un elemento de un vocabulario indexado por 1,2……..v. Las palabras se representan como vectores basados en unidad que tienen un componente individual igual a uno y todos los demás componentes igual a cero. Por lo tanto, usando sobrescritos para denotar componentes, la V-esima palabra en el vocabulario es representada por un vector w de tal forma que wv = 1 y wu = 0 para u ≠ v. Un documento es una secuencia de *N* palabras denotadas por D = (w1, w2,……., wN), donde *wn* es la wenésima palabra en la secuencia. Un corpus es una colección de *M* documentos denotados por D = w1, w2…., wM Una palabra embebida W: palabras -> R^n es una función parametrizada que mapea palabras en algún lenguaje a vectores de alta dimensionalidad [quizás de 200 a 500 vectores] (Bussieck, 2017).

Lo dicho anteriormente lo podemos representar con el siguiente ejemplo:

W (“casa”) = (0.2, -0.6, 0.7, 0.5…….)
W (“cielo”) = (0.5, 0.2, -0.3, 0.6…….)

2.4.1 Skip-Gram

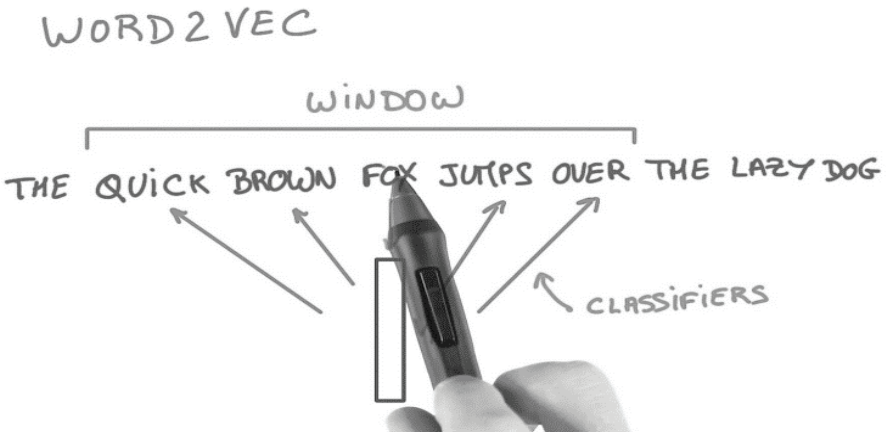
La idea principal detrás del modelo de *Skip-Gram* es la siguiente: toma cada palabra en un corpus grande (lo llamaremos *palabra de enfoque* o *central*); y también toma una por una las palabras que la rodean dentro de una “ventana” definida, para luego alimentar una red neuronal que, después del entrenamiento, predecirá la posibilidad de que cada palabra aparezca en la ventana alrededor de la palabra de enfoque (Barazza, 2017).

El tamaño de la ventana o contexto indica cuántas palabras antes y después de una palabra dada tomará en cuenta el algoritmo como contexto para el entrenamiento del modelo.[5] La ventana también se

5 https://www.kaggle.com/c/word2vec-nlp-tutorial/details/part-2-word-vectors

puede definir como la máxima distancia entre la palabra actual y la que se va a predecir dentro de una oración.[6]

Figura 2. Diagrama del método Skip-Gram

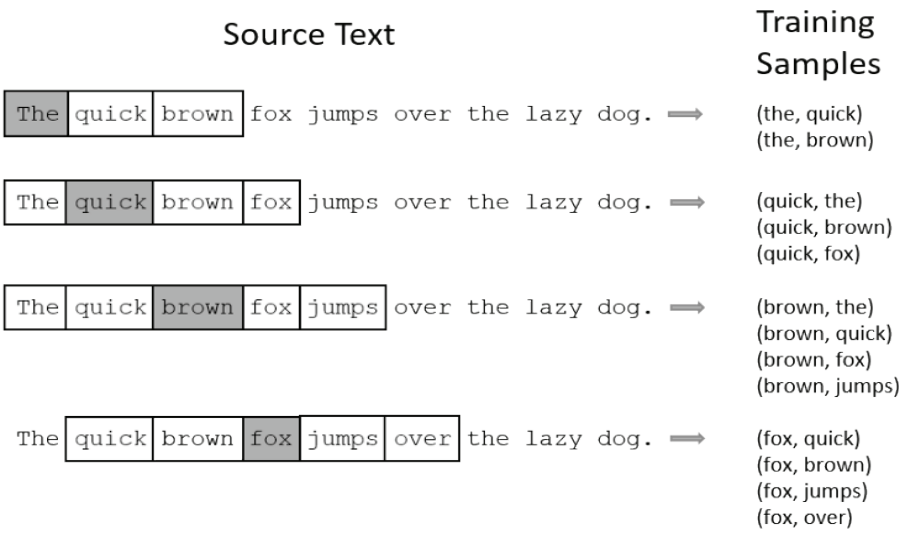


Copyrigth 2017 por Leonardo Barazza.

Si tenemos la oración *"The quick brown fox jumps over the lazy dog"*, los pasos que seguiremos son los representados en la imagen de abajo.

6 https://radimrehurek.com/gensim/models/word2vec.html

Figura 3. Pasos de *Skip-Gram*



Copyright 2016 por Chris McCormick.

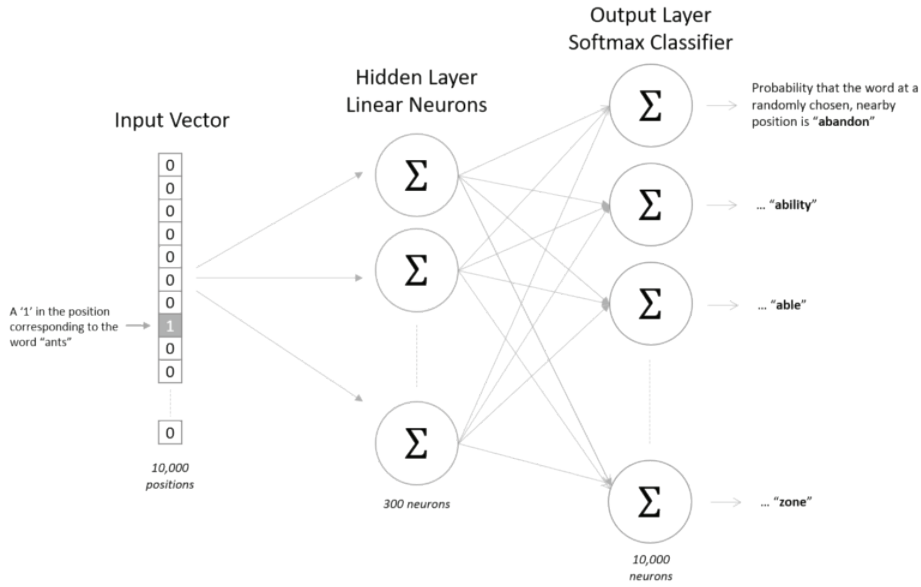
Se entiende que se debe alimentar una red neuronal con algunos pares de palabras, pero eso no se puede hacer simplemente usando como entrada los caracteres actuales. Es necesario poder representar las palabras en forma matemática, para que puedan ser procesadas; esto podría hacerse creando un vocabulario de todas las palabras de nuestro texto y luego codificar nuestra palabra como un vector de las mismas dimensiones que las de nuestro vocabulario. Cada dimensión puede ser pensada como una palabra en nuestro vocabulario. Así que tendremos un vector con todos los ceros y un 1 que representa la palabra correspondiente en el vocabulario. Esta técnica de codificación se denomina *codificación en caliente*.

Retomando el ejemplo anterior y separando cada palabra de la oración: *The, quick, brown, fox, jumps, over, the, lazy, dog*, la palabra *brown* estaría representada por el vector [0, 0, 1, 0, 0, 0 ,0, 0, 0]; y así se irían modificando, dependiendo de qué palabra se vaya tomando.

Esta representación ya puede ser utilizada como entrada de una red neuronal de dos capas. Será procesada por la segunda capa y dará como

resultado final una probabilidad, para que cada palabra del vocabulario aparezca en una posición elegida al azar de la palabra de enfoque o central.

Figura 4. Red neuronal para el modelo Skip-Gram

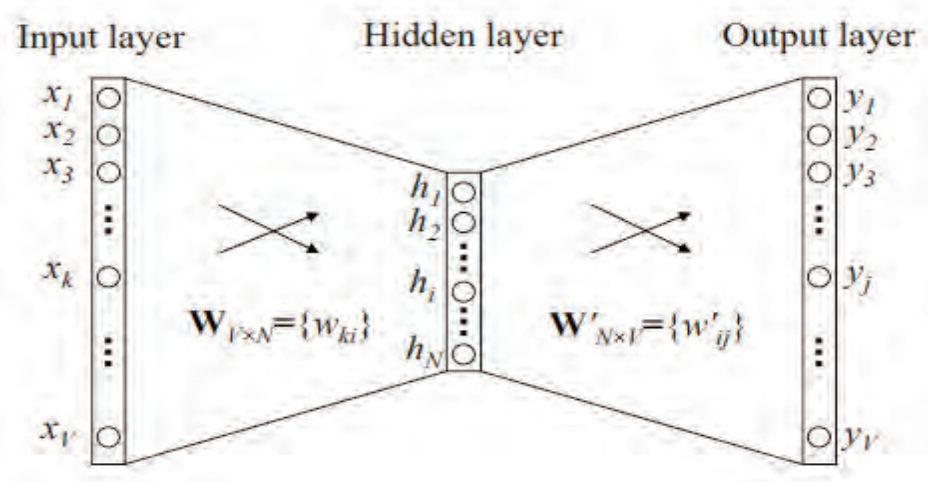


Copyright 2016 por Chris McCormick.

2.4.2 CBOW (continuous bag of words)

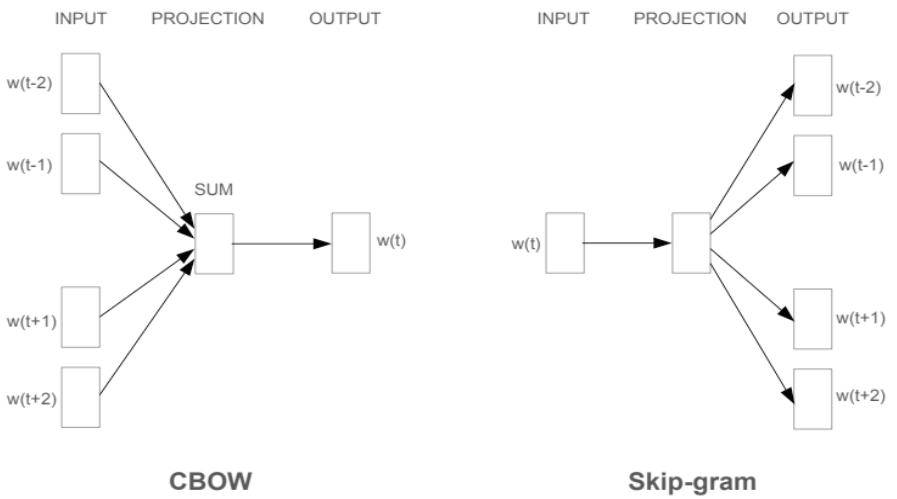
Si consideramos la versión más simple de CBOW, asumimos que solo hay una palabra tomada en cuenta por cada contexto, lo que significa que el modelo va a predecir una palabra objetivo dada una palabra de contexto, que es como un modelo *bigram* (Rong, 2014).

Figura 5. Modelo de CBOW con solo una palabra en el contexto (Rong, 2014)



En forma general, en el modelo CBOW, el contexto está representado por múltiples palabras para una determinada palabra de destino; el modelo predice la palabra actual a partir de una ventana de palabras contextuales circundantes.[7]

Figura 6. Comparación entre modelos



7 https://en.wikipedia.org/wiki/Word2vec

La arquitectura CBOW predice la palabra actual basada en el contexto, y el *Skip-Gram* predice las palabras circundantes dada la palabra actual (Mikolov et al., 2013).

En la figura 6 podemos ver una comparativa entre los modelos y de cómo se hace la predicción en cada uno de ellos.

Continuando con el ejemplo del zorro (*The, quick, brown, fox, jumps, over, the, lazy, dog*), dado que el objetivo de S*kip-Gram* es predecir palabras alrededor de un contexto dado, si consideramos una palabra antes y después de dicho contexto y *fox* como entrada, entonces los vectores asociados serán *brown* y *jump.*

En el caso de CBOW, si tenemos una palabra a la izquierda y una a la derecha, entonces *brown* y *jump* serán las entradas, y el vector asociado, *jump.*

2.4.3 *Muestreo negativo*

Es simplemente la idea de que solo actualizamos una muestra de palabras de salida por iteración. La palabra de destino de salida debe mantenerse en la muestra y se actualiza, y agregamos a esto algunas palabras (no objetivo) como muestras negativas. "Se necesita una distribución probabilística para el proceso de muestreo, y se puede elegir arbitrariamente... Uno puede determinar una buena distribución empíricamente" (Colyer, 2016).

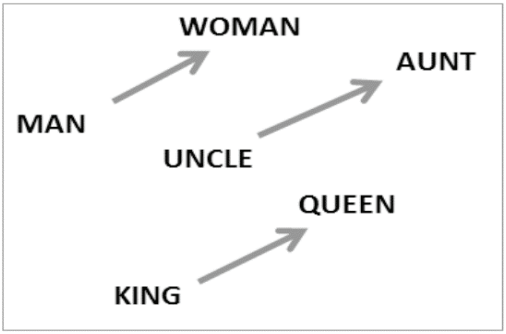
Según McCormick (2017), el muestreo negativo trata el problema del tremendo número de pesos que son actualizados por los millones de ejemplos que utilizamos como entrenamiento; que cada muestra de entrenamiento solo modifique un pequeño porcentaje de los pesos, en lugar de todos ellos.

Con el muestreo negativo, vamos a seleccionar al azar solo un pequeño número de palabras "negativas" para actualizar los pesos (en este contexto, una palabra "negativa" es aquella para la que queremos que la red produzca un 0 como salida). También se actualizan los pesos de nuestra palabra "positiva" (McCormick, 2016).

2.4.4 *Uso de los vectores*

Los vectores se pueden utilizar para responder a las analogías del tipo *A es a B como C es a algo que desconocemos*, por ejemplo, *tío es a tía como madre es a padre*, todo esto por medio del cálculo de la distancia de coseno.

Figura 7. Ejemplo de las relaciones de género que se pueden encontrar utilizando vectores



Una explicación de cómo se pueden utilizar los vectores la ofrece Mikolov (2013):

> Somewhat surprisingly, it was found that similarity of word representations goes beyond simple syntactic regularities. Using a word offset technique where simple algebraic operations are performed on the word vectors, it was shown for example that vector ("King") – vector ("Man") + vector ("Woman") results in a vector that is closest to the vector representation of the word Queen.

En la tabla 1 podemos ver algunos ejemplos de la relación antes mencionada.

Tabla 1. Ejemplos de relaciones entre palabras utilizando *word vectors*

Relationship	Example 1	Example 2	Example 3
France - Paris	Italy: Rome	Japan: Tokyo	Florida: Tallahassee
big - bigger	small: larger	cold: colder	quick: quicker
Miami - Florida	Baltimore: Maryland	Dallas: Texas	Kona: Hawaii
Einstein - scientist	Messi: midfielder	Mozart: violinist	Picasso: painter
Sarkozy - France	Berlusconi: Italy	Merkel: Germany	Koizumi: Japan
copper - Cu	zinc: Zn	gold: Au	uranium: plutonium
Berlusconi - Silvio	Sarkozy: Nicolas	Putin: Medvedev	Obama: Barack
Microsoft - Windows	Google: Android	IBM: Linux	Apple: iPhone
Microsoft - Ballmer	Google: Yahoo	IBM: McNealy	Apple: Jobs
Japan - sushi	Germany: bratwurst	France: tapas	USA: pizza

Recuperado de *The amazing power of word vectors.*

2.4.5 *Parámetros*

Para aprender un modelo implementando Word2Vec, este cuenta con una serie de parámetros que pueden ser modificados de acuerdo con nuestras necesidades y con la cantidad de información de que disponemos.

En el caso del lenguaje R,[8] podemos instalar la librería Word Vectors,[9] la cual en la sección de ayuda nos presenta la siguiente descripción acerca de los parámetros:

8 https://www.r-project.org/about.html
9 https://github.com/bmschmidt/wordVectors

Tabla 2. Parámetros de Word2Vec en R

Parámetro	Descripción
train file	Ruta del archivo de texto que será utilizado para entrenamiento. Los *tokens* son separados en espacios.
output file	Ruta del archivo de salida.
vectors	Número de vectores resultantes. Por defecto es 100. Más vectores usualmente significa más precisión, pero también más errores aleatorios, mayor uso de memoria y operaciones más lentas. Las elecciones sensatas están probablemente en el rango 100-500.
threads	Número de subprocesos por ejecutar durante el proceso de entrenamiento. El valor predeterminado es 1. Se puede usar número hasta el número de núcleos (virtuales) de su máquina para acelerar las cosas.
window	El tamaño de la ventana (en palabras) para usar en el entrenamiento.
classes	Número de clases para el agrupamiento *k-means*. No documentado/probado.
cbow	Si es 1, utiliza un modelo *bag of words* en lugar de S*kip-Gram*. El valor predeterminado es *false* (recomendado si se tiene poca experiencia con el modelo).
min count	Mínimo de veces en que una palabra debe aparecer para ser incluida en las muestras. Los valores altos ayudan a reducir el tamaño del modelo.
iter	Número de pases para hacer sobre el corpus en entrenamiento.
force	Si desea sobrescribir los archivos de modelos existentes.
negative samples	Número de muestras negativas que se deben tomar en el entrenamiento de S*kip-Gram*. 0 significa muestreo completo, mientras que los números más bajos dan un entrenamiento más rápido. Para grandes corpus, 2-5 puede ser utilizado; para corpus más pequeños, 5-15 es razonable.

Fuente: creación propia, tomando los datos de la ayuda proporcionada por la función en R.

2.5 *Visualización de los datos*

Con los vectores de palabras podemos hacer cálculos matemáticos y operaciones algebraicas, lo cual nos puede proporcionar una gran cantidad de información sobre los textos, sin embargo, puede ser difícil poder reconocer los resultados sin una representación visual.

Para ello existen diferentes técnicas que pueden ser implementadas en el lenguaje R. En esta sección se mencionan las utilizadas durante las pruebas.

2.5.1 *Nube de palabras*

Una nube de palabras, o nube de etiquetas, es una representación visual de las palabras que conforman un texto, en donde el tamaño es mayor para las palabras que aparecen con más frecuencia (Halvey & Keane, 2017).

Figura 8. Ejemplo de nube de palabras. El tamaño de las palabras representa la frecuencia de aparición en los textos, a mayor frecuencia mayor tamaño.

Fuente: nube obtenida a partir del modelo creado.

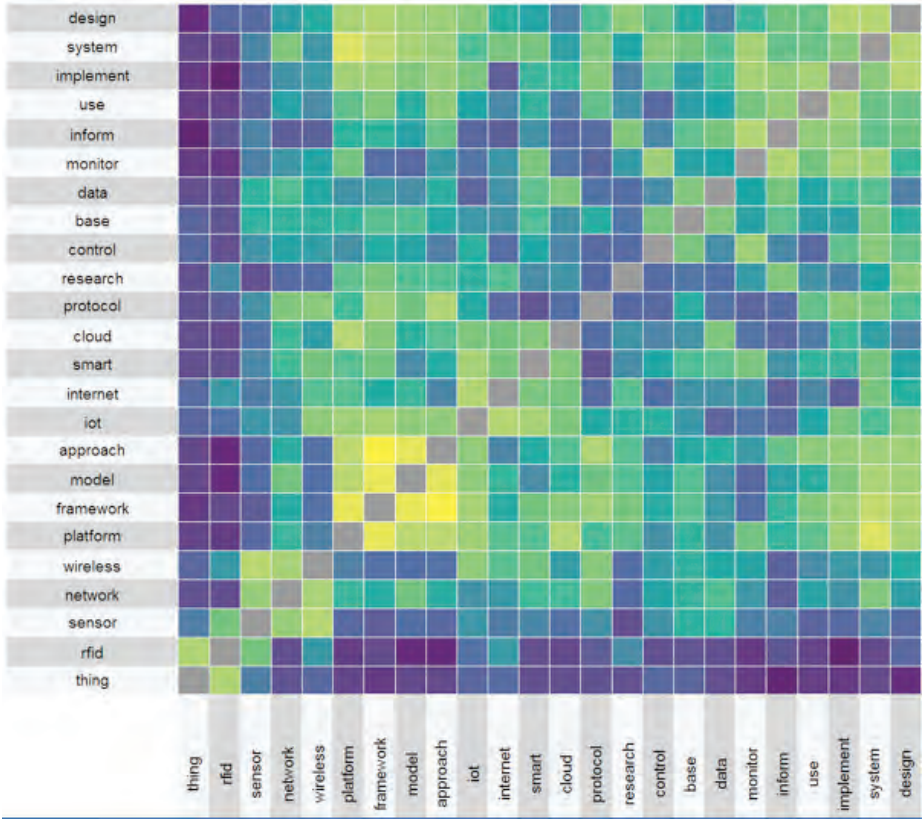
Usando las nubes de palabras y los modelos generados por Word2Vec se podrían agrupar palabras mediante clúster y generar un posible concepto.

2.5.2 *Similitud de coseno*

Es una medida de la similitud existente entre dos vectores en un espacio que posee un producto interior con el que se evalúa el valor del coseno del ángulo comprendido entre ellos. Esta función trigonométrica proporciona un valor igual a 1 si el ángulo comprendido entre los vectores es cero, es decir, si ambos apuntan a un mismo lugar. Cualquier ángulo existente entre los vectores, el resultado de la similitud de coseno arrojaría un valor inferior a uno. Si los vectores fuesen ortogonales, el coseno se anularía, y si apuntasen en sentido contrario, su valor sería -1. De esta forma, el valor de esta métrica se encuentra entre -1 y 1, es decir en el intervalo cerrado [-1,1].[10]

10 https://es.wikipedia.org/wiki/Similitud_coseno

Figura 9. Ejemplo de *heatmap*. Los colores en general representan la relación entre palabras, mientras los colores violetas indican que las palabras se encuentran distantes, y el amarillo, que se encuentran cercanas; como punto intermedio se encuentra el color verde.



Fuente: mapa obtenido a partir de la creación del modelo.

Esta distancia podemos representarla en R utilizando el paquete *superheat*,[11] el cual fue desarrollado para producir *heatmaps* personalizables y extensibles que actúan como una herramienta para la exploración visual de conjuntos de datos complejos. *Superheat* mejora el *heatmap* tradicional proporcionando una plataforma para visualizar una amplia gama de tipos de datos simultáneamente, agregando al

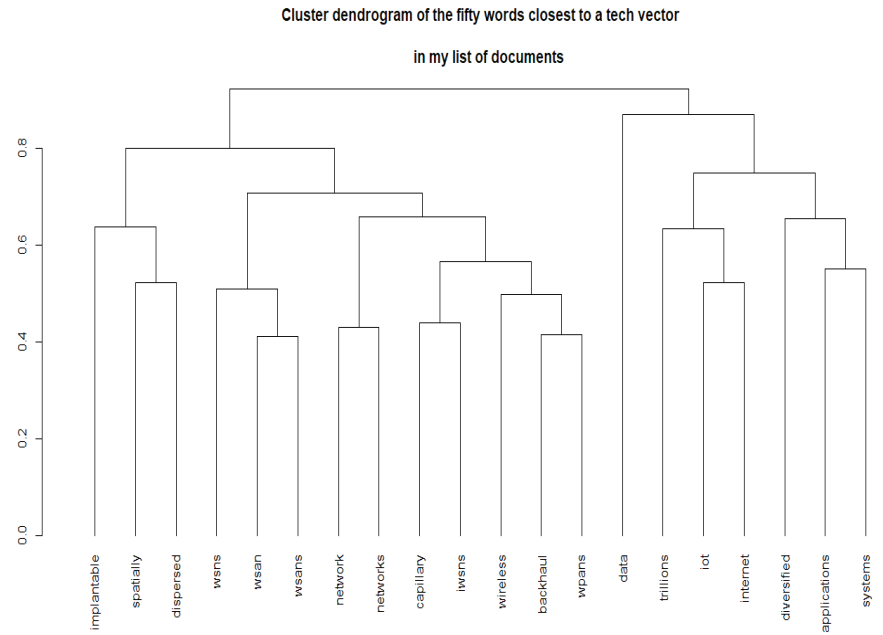
11 https://cran.r-project.org/web/packages/superheat/index.html

heatmap una variable de respuesta como un diagrama de dispersión, resultados de diferentes modelos como diagramas de caja, información de correlación como diagramas de barras, información de texto y más.[12]

2.5.3 *Análisis de clúster y dendrogramas*

El análisis de clúster (o análisis de conglomerados) es una técnica de análisis exploratorio de datos para resolver problemas de clasificación. Su fin consiste en ordenar objetos en grupos (conglomerados o clúster) de forma que el grado de asociación/similitud entre miembros del mismo clúster sea más fuerte que el grado de asociación/similitud entre miembros de diferentes clústeres. Cada clúster se describe como la clase a la que sus miembros pertenecen (Vicente Villardón, 2007).

Figura 10. Ejemplo de representación mediante dendrograma



Fuente: dendrograma obtenido a partir de la creación del modelo.

12 https://rlbarter.github.io/superheat/index.html

El análisis clúster es un conjunto de técnicas multivariantes utilizadas para clasificar a un conjunto de individuos en grupos homogéneos. Así pues, el objetivo es obtener clasificaciones (*clusterings*), teniendo, por lo tanto, el análisis un marcado carácter exploratorio.[13]

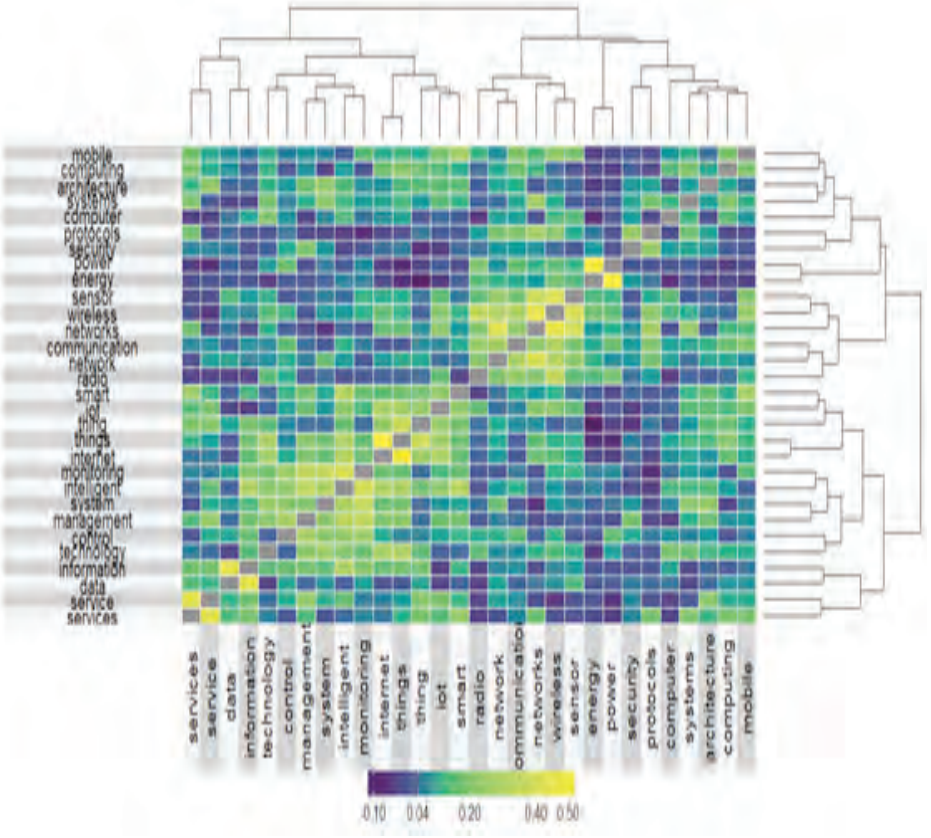
Podemos encontrarnos dos tipos fundamentales de métodos de clasificación: jerárquicos y no jerárquicos. En los primeros, la clasificación resultante tiene un número creciente de clases anidadas, mientras que en el segundo las clases no son anidadas (Vicente Villardón, 2007).

Para visualizar estos clústeres podemos hacer uso de los dendrogramas, los cuales son un tipo de representación gráfica o diagrama de datos en forma de árbol (del griego δένδρον, *déndron*, 'árbol'), que organiza los datos en subcategorías que se van dividiendo en otros hasta llegar al nivel de detalle deseado (asemejándose a las ramas de un árbol, que se van dividiendo en otras sucesivamente). Este tipo de representación permite apreciar claramente las relaciones de agrupación entre los datos e incluso entre grupos de ellos, aunque no las relaciones de similitud o cercanía entre categorías.[14]

En R tenemos la posibilidad de hacer una representación combinando los *heatmap* y los dendrogramas, con un resultado como el que se muestra en la figura 11.

13 https://www.uv.es/ceaces/multivari/cluster/CLUSTER2.htm
14 https://es.wikipedia.org/wiki/Dendrograma

Figura 11. Combinación de *heatmap* y dendrograma. La escala de colores nos indica la relación entre las palabras, mientras más oscuro es el color más distantes se encuentran, y mientras más claro (amarillo), *más cercanas.*



Fuente: creación propia.

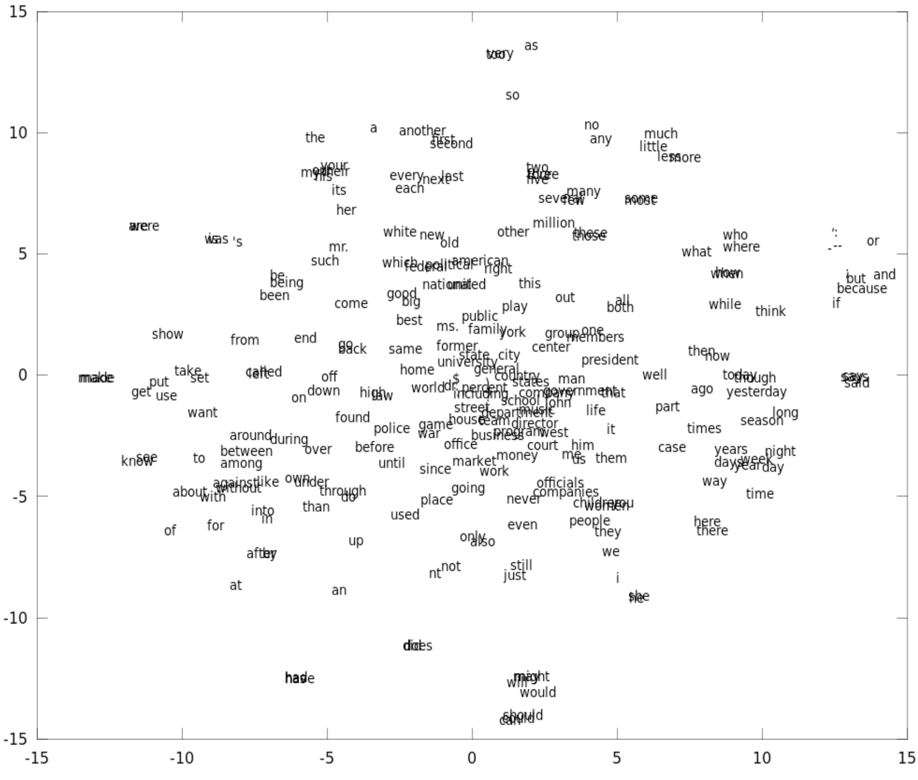
2.5.4 t-SNE

De acuerdo con Maaten (2008), t-SNE se puede describir como una manera de convertir un conjunto de datos de alta dimensión en una matriz de similitud de pares para visualizar la similitud de datos resultante; y capaz de capturar muy bien gran parte de la estructura local de la alta dimensionalidad, mientras revela estructura global como la presencia de clúster en varias escalas.

t-SNE es una técnica utilizada para la reducción de la dimensionalidad, que es particularmente adecuada para la visualización de conjuntos de datos de alta dimensión. La técnica se puede implementar mediante aproximaciones Barnes-Hut, lo que permite su aplicación en grandes conjuntos de datos del mundo real (Maaten, 2017).

En la mayoría de casos, t-SNE funciona fácilmente y además está implementado en diferentes lenguajes de programación, como es el caso de R.[15]

Figura 12. Representación de aprendizaje de palabras con t-SNE



Copyright 2014 por Nitish Srivastava, Jian Yao.

15 https://cran.r-project.org/web/packages/tsne/tsne.pdf

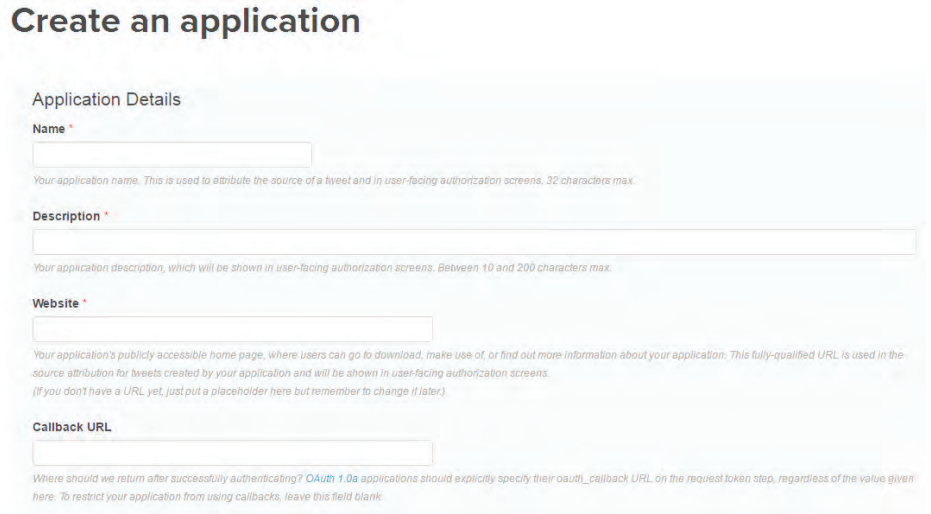
En la figura 12, se puede ver una representación embebida de palabras aprendidas de un modelo en dos dimensiones.

2.6 *Obteniendo la información de twitter*

R cuenta con una serie de librerías que nos permite recolectar tuits mediante una API creada desde el sitio web para desarrolladores de Twitter. Para crear la cuenta, debemos seguir los pasos listados a continuación:

1. Contar con una cuenta activa en Twitter a la cual previamente se haya agregado un número telefónico, pues es requisito para la creación de la API.
2. Ir a https://dev.twitter.com/apps en la sección *My apps*.
3. Creamos una nueva aplicación llenando todos los campos obligatorios.

Figura 13. Creación de la API desde el sitio web de desarrollo de Twitter



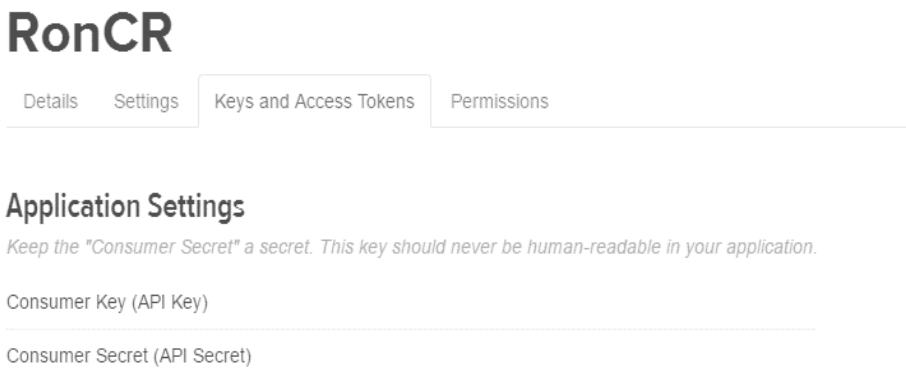
Fuente: generada a partir de la creación de la API.

Figura 14. API creada para el análisis de datos

Fuente: generada a partir de la creación de la API.

4. Obtenemos los siguientes datos que serán utilizados como conexión:
    a. Consumer Key (API Key)
    b. Consumer Secret (API Secret)
    c. Access Token
    d. Access Token Secret

Figura 15. Acceso a las credenciales para ser utilizadas en R



Fuente: generada a partir de la creación de la API.

Luego de crear la API, podremos tener acceso utilizando el código mostrado en la figura 16.

Figura 16. Uso de la clave secreta y el *token* para tener acceso a nuestra API desde R

```
#Estos son los datos de la API
api_key <-
api_secret

#Estos son los datos del token para hacer peticiones a nuestra API
token <-
token_secret <-

#Con los datos antes mencionados creamos una coneccion con Twitter
setup_twitter_oauth(api_key, api_secret, token, token_secret)
```

Fuente: generada a partir de la creación de la API.

3. Metodología

En esta sección se hace una breve descripción de los pasos seguidos para desarrollar el proyecto. Para lograr los objetivos planteados, el proyecto fue dividido en las siguientes fases principales: 1. definición del proyecto mediante reuniones con el tutor, 2. aprendizaje, 3. preparación de la base de datos y 4. desarrollo de pruebas y resultados.

1) Definición del proyecto

Parte importante es definir el alcance y los objetivos del proyecto, los resultados que se buscan y la forma en que se van a analizar. Para ello se iban evaluando constantemente los avances, teniendo en cuenta los objetivos planteados; también se tuvieron algunas reuniones con especialistas en el tema para solventar dudas o solicitar información pertinente.

2) Aprendizaje

Para lograr los resultados finales planteados en el proyecto, fue necesario adquirir conocimiento sobre el uso de R para análisis de texto, de

técnicas de visualización y su comprensión. Conceptos, implementación, ventajas, funcionamiento y uso de las bases de datos también fueron parte del conocimiento adquirido.

Se consultaron diferentes fuentes, como por ejemplo el sitio oficial de R, tutoriales de YouTube, tutoriales para cada una de las técnicas de visualización, Coursera[16] y Kaggle.[17] Para la parte de codificación, también se consultó una gran variedad de ejemplos que se encontraban almacenados en sitios como Github[18] y Stack Overflow.[19,20]

Unas de las etapas donde se encontraron dificultades fueron la de preparación del entorno en R y la de utilización del modelo generado. Al momento de instalar las librerías para el uso de Word2Vec, aparecieron algunos errores que tenían que ver con Rtool. Cabe mencionar que en la versión de R x64 3.3.2 y de RStudio 1.0.136 ya no se presentaron problemas.

En el caso de la parte enfocada al análisis de tuits, fue necesario conocer más sobre las etapas de limpieza de texto dado que, debido a la naturaleza del contenido, existe una mayor cantidad de caracteres que pueden afectar los resultados.

3) Preparación de base de datos

El desarrollo de las pruebas se dividió en dos partes:

Análisis de textos utilizando el algoritmo Word2Vec

Para el desarrollo de las pruebas y la comprensión de los modelos, era necesario construir una base datos de textos; y dado que eran pruebas de conceptos no era relevante la base que se utilizaría, por lo que se sondearon diferentes opciones, de las cuales se seleccionaron Scopus y Web of Science[21] porque proporcionan diferentes opciones para obtener la información.

16 https://www.coursera.org/
17 https://www.kaggle.com/
18 https://github.com/
19 https://stackoverflow.com/
20 Ver la sección "Enlaces a artículos y tutoriales consultados" donde se encuentra detallado todas las fuentes de información consultadas durante el proceso de aprendizaje.
21 http://wokinfo.com/media/pdf/qrc/webofscience_qrc_en.pdf

Inicialmente se había decidido que la información que se utilizaría de los artículos o textos de las bases de datos fuese *title* y *abstrac*, pero a medida se avanzó también se consideró utilizar las *author keywords* y las *index keywords*.[22]

Teniendo en cuenta lo anterior, la base que se utilizó finalmente fue Scopus, ya que luego de aprender el funcionamiento de cada una, conocer las ventajas y posibilidades, hacer pruebas y comparar los resultados, esta era la que mejores datos aportaba para nuestras pruebas de concepto.

Se seleccionó como tema para la búsqueda de artículos "*Internet of Things*"; las descargas se hicieron seleccionando los artículos por año, desde 2008 hasta 2017, con lo que se obtuvieron más de 9 mil ejemplares en archivos separados, con los cuales se formó una sola base de datos utilizada para crear el modelo con Word2Vec.

Para aplicar los modelos a una fuente de textos en español, se utilizó un conjunto de artículos proporcionados por la Unidad de Datos de *El Diario de Hoy*, que van desde el 1 de noviembre al 1 de diciembre de 2017; y que están compuestos por titular, resumen y cuerpo.

Análisis de tuits utilizando las librerías de R

Para el análisis de tuits, se extrajo la información de un conjunto de cuentas por petición de la Unidad de Datos de *El Diario de Hoy* bajo los siguientes criterios:

- Cuentas de candidatos a puestos de elección popular con alta trascendencia mediática.
- Mantener representatividad tanto de candidatos a la Asamblea Legislativa como a los concejos municipales.
- Mantener un balance entre los contendientes de diferentes partidos políticos.

22 https://www.elsevier.com/__data/assets/pdf_file/0007/69451/scopus_content_coverage_guide.pdf

Las cuentas utilizadas fueron las siguientes:

Tabla 3. Cuentas de Twitter por analizar

Nombre	Afiliación	Cuenta
Luis Rodríguez	FMLN-CD	@LRodriguez_SV
Milagro Navas	Arena	@Milagro_Navas
Roberto d'Aubuisson	Arena	@RDaubuisson
Lorena Peña	Arena	@Lorenagpeam
Miguel Pereira	FMLN	@MiguelPereiraSV
Will Salgado	Gana	@willsalgado
Jackeline Rivera	FMLN	@JackelineRA_
Ernesto Muyshondt	Arena	@EMuyshondt
Norman Quijano	Arena	@Norman_Quijano
Guillermo Gallegos	Gana	@GGallegos24
Milena de Escalón	Arena	@MilenaEscalon

Lista proporcionada por la Unidad de Datos de *El Diario de Hoy*
Fuente: creación propia.

4) Pruebas e implementación

Teniendo la base de datos, se procedió a utilizar Word2Vec con un modelo preentrenado por Google llamado *Google News* y otro generado a partir de dicha base de datos, se utilizaron las secciones de los artículos y representó la información con diferentes técnicas de visualización, como nubes de palabras, t-SNE, *heatmap* y dendrogramas.

Una de las dificultades que se encontró en esta etapa fue la extracción de la información de los modelos, ya que tienen un formato binario, lo que complicaba la extracción de la estructura de los vectores resultantes.

Se consultaron opciones hasta que finalmente se encontró un programa desarrollado en C, ejecutado en Linux, que hacía la conversión sin pérdida de datos (ver la sección "Pruebas" para más detalles del procedimiento seguido).

Con respecto a los análisis de las cuentas de Twitter, algunas dificultades encontradas fueron con el manejo de las librerías de R del idioma español, por ejemplo, las tildes, la lista de palabras vacías, los

caracteres especiales, además de todos los símbolos y caracteres propios de la red social.

4. Desarrollo

En esta sección se describe el proyecto, detallando los procedimientos seguidos durante sus etapas. Contiene los procesos desarrollados desde el aprendizaje hasta la presentación de los resultados finales, y se incluye una breve descripción de las principales herramientas utilizadas.

4.1 *Herramientas utilizadas*

Antes de empezar con la descripción de lo hecho durante el proyecto, es apropiado listar las herramientas utilizadas con una pequeña descripción de su función, que han sido divididas en dos partes: una orientada al desarrollo y otra a la edición y documentación.

Tabla 4. Lenguajes de programación y entornos de desarrollo

Nombre	Ámbito	Descripción
R	Lenguaje de programación	Lenguaje y un entorno para la informática estadística y los gráficos. Es un proyecto GNU que es similar al lenguaje S y el entorno que fue desarrollado en Bell Laboratories.[23]
RStudio	IDE	RStudio es un entorno de desarrollo integrado (IDE) para R. Incluye una consola, editor de resaltado de sintaxis que admite la ejecución directa de código, así como herramientas para trazar, la historia, la depuración y la gestión del espacio de trabajo.[24]

23 https://www.r-project.org/about.html
24 https://www.rstudio.com/products/rstudio/

C	Lenguaje de programación	El lenguaje de programación C fue ideado a principios de los setenta como un lenguaje de implementación de sistemas para el naciente sistema operativo Unix. Creado en una pequeña máquina como una herramienta para mejorar un escaso ambiente de programación, se ha convertido en uno de los lenguajes dominantes de hoy.[25]

Descripción de los lenguajes utilizados durante el desarrollo. La tabla ha sido creada a partir de la descripción disponible en los sitios web de cada lenguaje.
Fuente: creación propia.

Tabla 5. Herramientas utilizadas para la obtención de los datos, documentación y edición de textos

Nombre	Ámbito	Descripción
Scopus	Base de datos bibliográfica	Es la mayor base de datos de citas y resúmenes de literatura revisada por pares: revistas científicas, libros y actas de congresos. Ofrece una visión general de la producción mundial de investigación en diferentes campos.[26]
Sublime Text	Editor de texto	Sofisticado editor de texto para código, marcado y prosa, con interfaz de usuario elegante, características extraordinarias y un rendimiento increíble.
Microsoft Word	Documentación	Programa informático orientado al procesamiento de textos. Fue creado por la empresa Microsoft, y viene integrado predeterminadamente en el paquete ofimático denominado Microsoft Office.[27]
Twitter	Red social	Es un servicio de *microblogging*, con sede en San Francisco (California), con filiales en San Antonio (Texas) y Boston (Massachusetts) en Estados Unidos.[28]

Descripción de las herramientas de edición utilizadas. La tabla ha sido creada a partir de la descripción disponible en los sitios web de cada herramienta.
Fuente: creación propia.

25 http://csapp.cs.cmu.edu/3e/docs/chistory..html
26 https://www.elsevier.com/solutions/scopus
27 https://es.wikipedia.org./wiki/Microsoft_Word
28 https://es.wikipedia.org./wiki/Twittwe

4.2 *Base de datos*

En un principio, para la fase de aprendizaje, se planteó utilizar la base de datos de Web of Science,[29] pero para tener una comparativa también se decidió utilizar Scopus, llegando a la conclusión de que esta última base de datos ofrecía la flexibilidad de descarga y selección de atributos que más se adaptaba a nuestras necesidades.

En esta sección se detallan los procedimientos seguidos para descargar la información desde Scopus; y que se utilizó durante las pruebas.

La secuencia de pasos seguida para preparar los datos es la que se muestra en la figura 17.

Figura 17. Diagrama con los pasos seguidos para la creación de la base de datos

Fuente: creación propia.

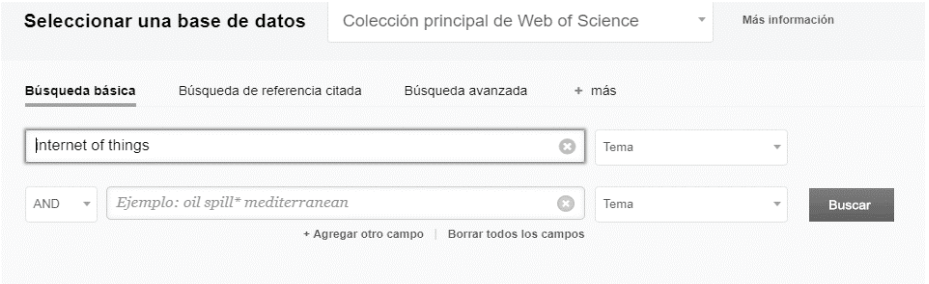
4.2.1 *Selección de base de datos*

La primera base para obtener la información fue Web of Science; se hicieron pruebas para verificar las opciones que brindaba y se pudo observar que no contaba con todas las facilidades que se requerían.

Contaba con la posibilidad de hacer búsquedas refinadas con base en diferentes criterios, como se muestra en la figura 18.

29 www.webofknowledge.com/

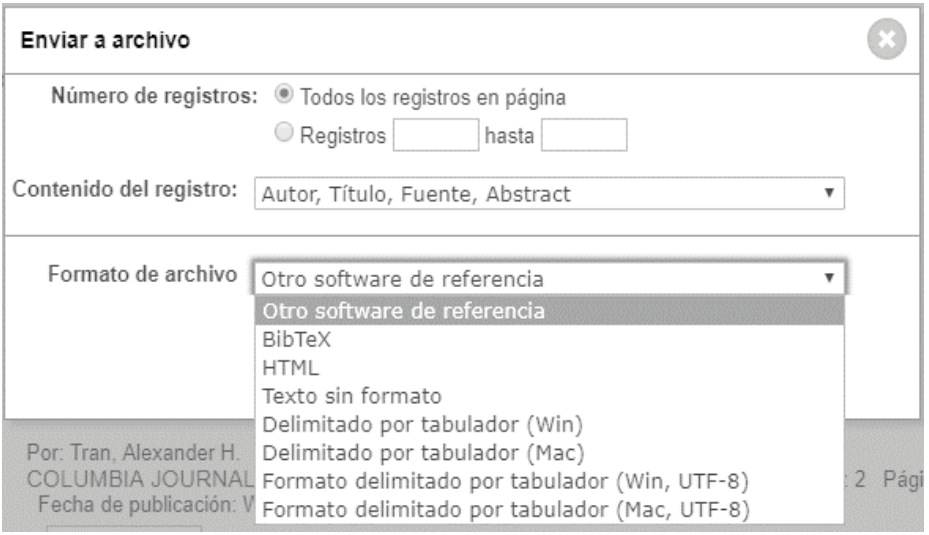
Figura 18. Búsqueda en la base de datos Web of Science



Fuente: https://www.scopus.com/freelookup/form/author.uri

Tanto los campos como los formatos soportados eran limitados, como se muestra en la figura 19.

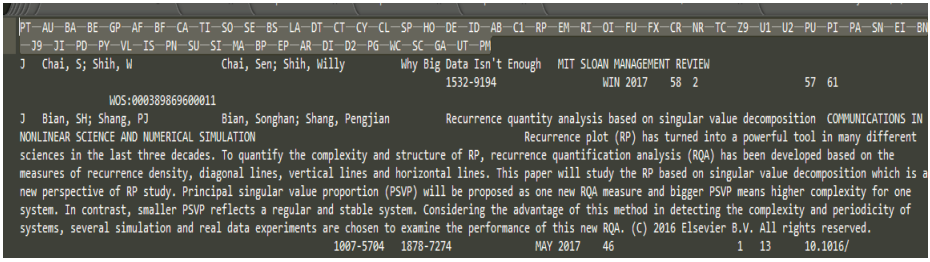
Figura 19. Opciones de descarga de Web of Science



Fuente: https://www.scopus.com/freelookup/form/author.uri

Al descargar la información, esta se guardaba en archivo de texto, ya que para las otras se debía tener cuentas activas. Estos archivos tenían información adicional que no se utilizarían para las pruebas; y algunas veces no se encontraban separados adecuadamente. Se puede ver la estructura en la figura 20.

Figura 20. Estructura de los datos descargados de Web of Science



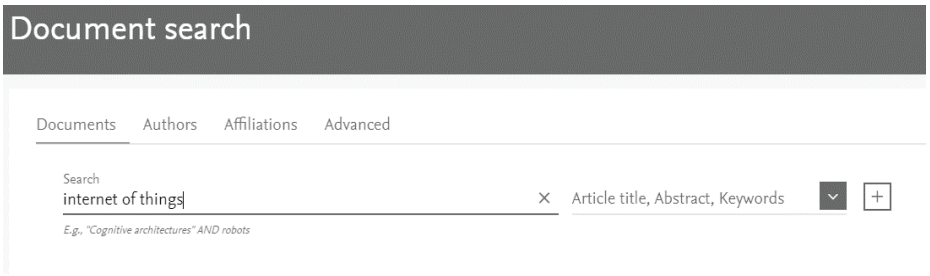
Fuente: creación propia.

Teniendo en cuenta lo anterior, se optó por cambiar a Scopus como fuente para descargar los datos que serían utilizados.

4.2.2 *Creación de la base de datos*

Se decidió seleccionar como tema de búsqueda *Internet of Things* y descargar documentos relacionados con este.

Figura 21. Búsqueda de documentos en Scopus

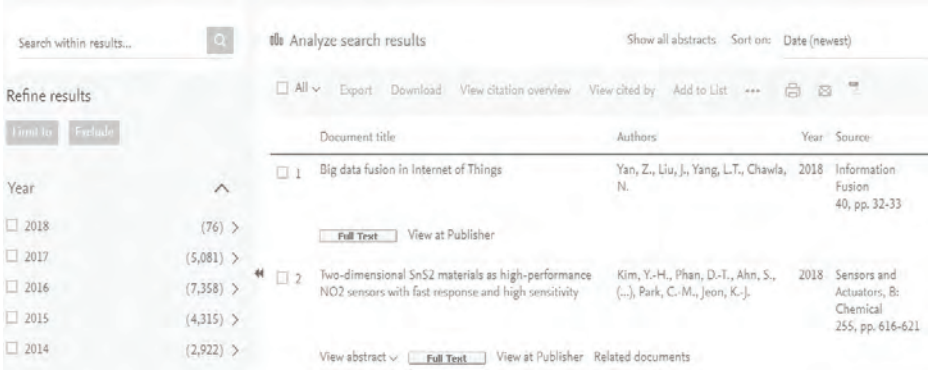


Fuente: https://www.scopus.com/freelookup/form/author.uri

La figura 21 nos muestra la plataforma de búsqueda de Scopus, la cual es muy intuitiva y nos permite hacer búsquedas más refinadas, por ejemplo, utilizando los operadores lógicos *And*, *Or* y *Not*, además de poder tener en cuenta diferentes secciones de las publicaciones.

Los resultados presentados son clasificados por diferentes aspectos, como, por ejemplo, año, autor, temática, tipo de documento y otros.

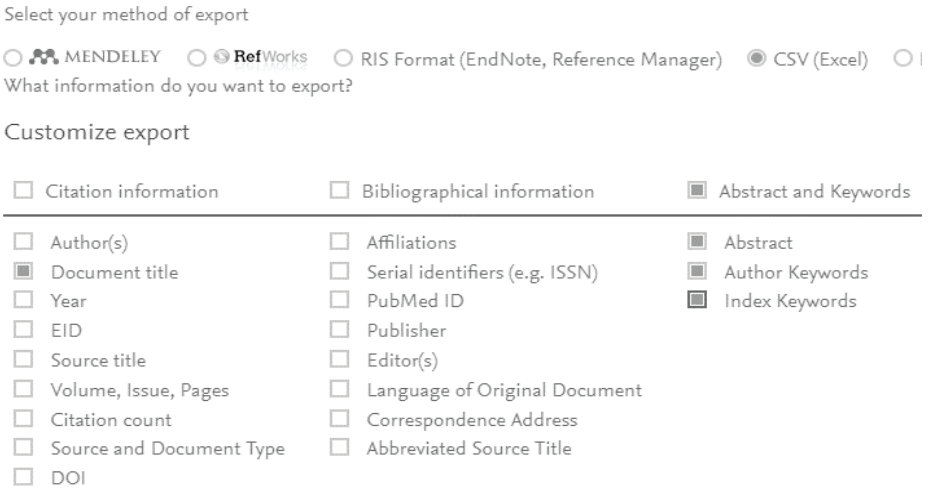
Figura 22. Resultados de la búsqueda, donde se detalla la información de los documentos y la clasificación por año



Fuente: https://www.scopus.com/freelookup/form/author.uri

Como se mencionó, una de las ventajas de Scopus es que permite descargar los datos de acuerdo con las necesidades, brindando diferentes formatos para la descarga de los archivos y pudiendo elegir todos los campos o solamente algunos de ellos. En nuestro caso utilizamos el tipo de archivo CSV y seleccionamos el título, el *abstract* y las *keywords*, aunque en un principio fueron solamente el título y el *abstract*.

Figura 23. Selección de los campos que se han de utilizar y el tipo de archivo en el que se descargará la información



Fuente: https://www.scopus.com/freelookup/form/author.uri

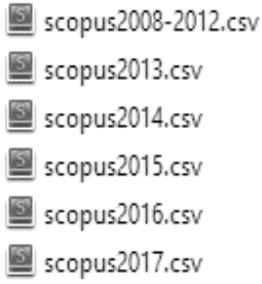
Uno de los detalles de Scopus es que solamente permite descargar 2 mil documentos a la vez, como se muestra en la figura 24.

Figura 24. Número de documentos permitidos por descarga

Fuente: https://www.scopus.com/freelookup/form/author.uri

Debido a esto, lo que se hizo fue seleccionar los artículos por años para tener un total de más de 9 mil documentos. De este modo se generaron archivos por cada año, como se muestra en la figura 25.

Figura 25. Archivos segmentados con la información descargada de Scopus



Fuente propia.

Para iniciar con las pruebas, era necesario fusionar todos los archivos para tener la información en uno solo, esto se hizo mediante una función en R que permitía leer los archivos de un directorio y luego ir agregando uno a uno a un *dataset*. Cabe mencionar que es posible agregar más archivos a futuro, para que la base de datos siga creciendo y de este modo se cuente con más datos; y así no se tendría que hacer modificación alguna al código.

Figura 26. Representación del *dataset* conteniendo todos los documentos descargados

Title	Link	Abstract
Application architectures for smart multi-device appli...	https://www.scopus.com/inward/record.uri?eid=2-s...	The growing number of connected devices a
Wireless access and mobility support for automated ...	https://www.scopus.com/inward/record.uri?eid=2-s...	The preconditions of storage pre-schedule ar
Intelligent condition monitoring and management for ...	https://www.scopus.com/inward/record.uri?eid=2-s...	Condition monitoring and management systen
Enabling the usage of sensor networks with service-...	https://www.scopus.com/inward/record.uri?eid=2-s...	Closing the gap between device-oriented ser
Complex event processing mechanism in internet of ...	https://www.scopus.com/inward/record.uri?eid=2-s...	The data and events from Internet of Things a
Demo: Uncovering device whispers in smart homes	https://www.scopus.com/inward/record.uri?eid=2-s...	As the Internet of Things finds its way into pri
Heavy minerals in the 2011 Tohoku-oki tsunami depo...	https://www.scopus.com/inward/record.uri?eid=2-s...	The 2011 Tohoku-oki tsunami left sand and m
Model checking of the reliability of publish/subscrib...	https://www.scopus.com/inward/record.uri?eid=2-s...	Recent studies show that Information Centric
The legal challenges of networked robotics: From th...	https://www.scopus.com/inward/record.uri?eid=2-s...	One of the reasons that future robots will enh
A cognitive management framework to support explo...	https://www.scopus.com/inward/record.uri?eid=2-s...	In this article, a cognitive management framev
A resource scheduling approach for media uploading...	https://www.scopus.com/inward/record.uri?eid=2-s...	Currently, more and more Internet of Things (I

Showing 1 to 13 of 9,361 entries

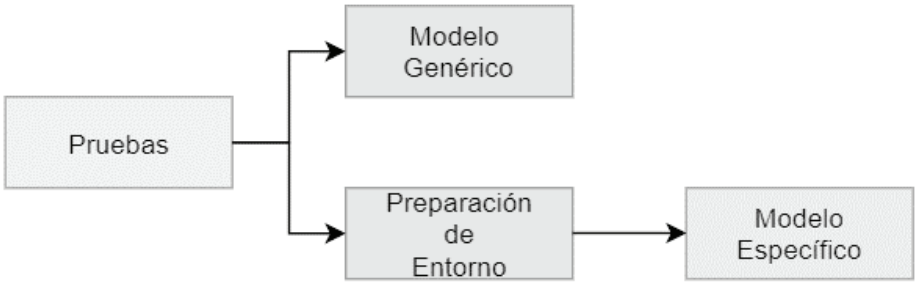
Fuente: creación propia

4.3 *Pruebas*

4.3.1 *Haciendo uso de* WORD2VEC

Para poder ver el funcionamiento de Word2Vec y cómo los resultados podrían variar de acuerdo con la información utilizada se realizaron diferentes pruebas, que se representan en el diagrama de la figura 27.

Figura 27. Diagrama con los pasos seguidos para las pruebas



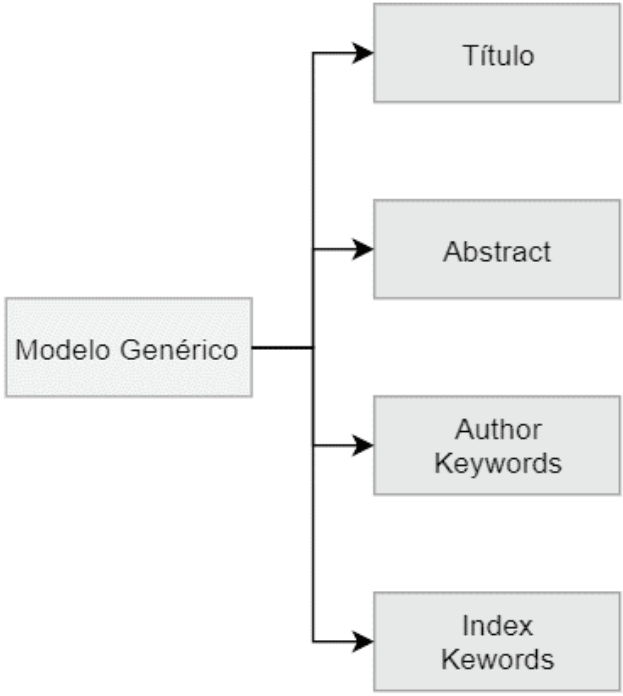
Fuente: creación propia

4.3.1.1 *Modelo genérico*

Como modelo genérico, se utilizó uno entrenado por Google llamado Google News,[30] el cual cuenta con una enorme cantidad de palabras. Por espacio de representación, en el caso de los *heatmap*, se han utilizado solamente las palabras más comunes, pero en las nubes de palabras se usa una mayor cantidad. Las pruebas desarrolladas en esta sección se detallan en la figura 28.

30 https://code.google.com/archive/p/word2vec/

Figura 28. Pruebas desarrolladas con el modelo genérico



Fuente: creación propia.

Estas pruebas se dividieron en dos partes: la primera haciendo preprocesamiento de texto sin usar el *stemming*, y la segunda, donde sí se incluye.

Para iniciar las pruebas, fue necesario extraer el modelo generado por Google; que era un archivo binario y pasarlo a texto, y de este modo extraer la información para comparar con la base de datos extraída de Scopus.

Fue necesario utilizar el programa Convert[31] desarrollado en C para poder hacer la conversión de archivos desde una terminal en Linux, como se muestra en la figura 29.

31 https://github.com/anotheremily/bin2txt

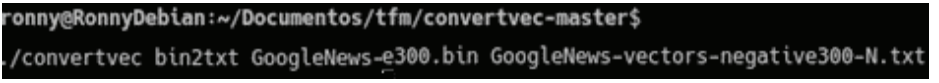
Figura 29. Parámetros necesarios para convertir los archivos binarios en texto

Fuente: creación propia.

Los parámetros necesarios son los siguientes:

- *Method:* permite seleccionar el tipo de conversión, ya sea de binario en texto o viceversa.
- *Input_path:* ruta de la ubicación del archivo origen.
- *Output_path:* destino y nombre del archivo resultante.

Figura 30. Ejemplo de uso de parámetros para la conversión de archivos



Fuente: creación propia.

La figura 30 nos muestra el archivo binario que contiene el modelo de Google y nos da como resultado un archivo de texto que contiene toda la información, y que puede ser utilizado en las pruebas siguientes. El tiempo de conversión puede variar de acuerdo con el tamaño del modelo y con los recursos con los que cuente el ordenador, tanto en memoria RAM como en microprocesador.

4.3.1.1.1 *Texto sin* stemming

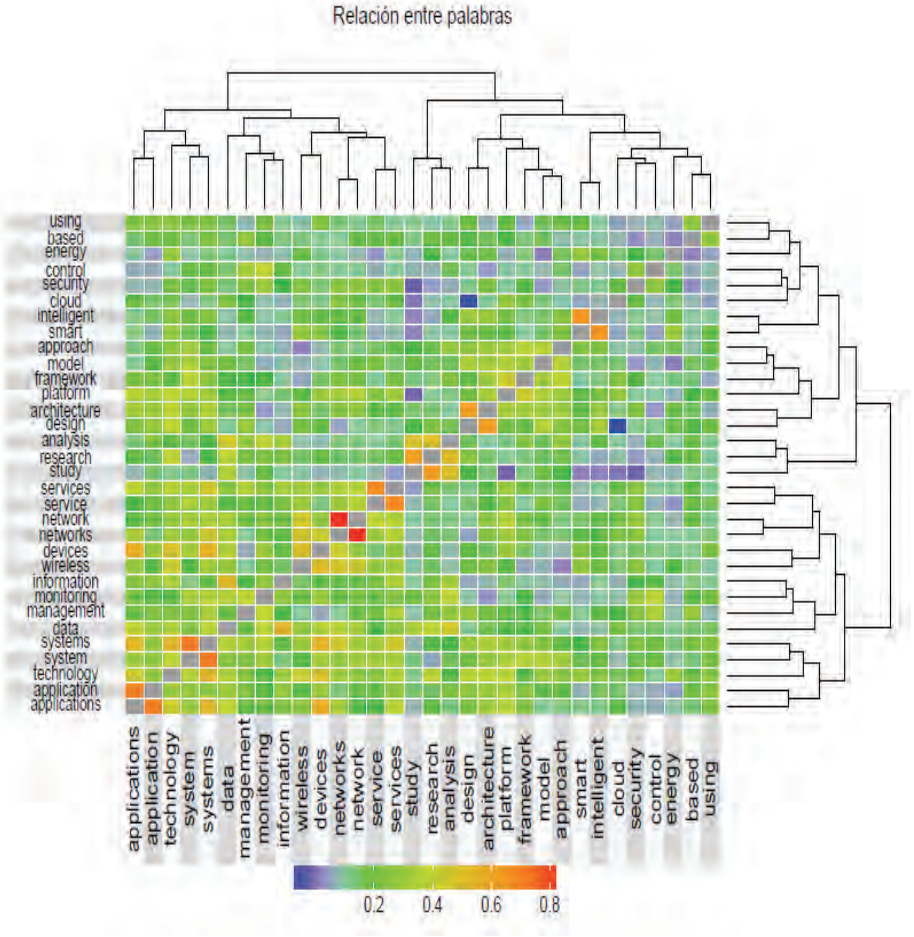
En la primera parte de las pruebas se preprocesó el texto eliminando puntuación, números, palabras vacías y llevando los textos a minúsculas sobre las partes seleccionadas de la base de datos.

Los resultados obtenidos en cada una fueron los siguientes:

- **Título**

La primera representación de los resultados fue mediante un *heatmap* donde podemos ver la relación entre las palabras, teniendo en cuenta la distancia de coseno que hay entre ellas, pasando del azul al rojo a medida que se van acercando.

Figura 31. Heatmap con los resultados sobre el título de las publicaciones. Los colores indican la relación entre palabras, siendo las de tonalidad azul las más lejanas entre ellas, y las rojas, las más cercanas.



Fuente: creación propia.

En la figura 31 podemos ver, además, la relación de las palabras mediante dendrogramas que agrupan las palabras similares, pudiendo de este modo hacernos una primera idea de posibles conceptos.

Debido a que no se ha aplicado *stemming*, las palabras aparecen en sus diferentes formas; y podemos notar que los singulares y plurales de una misma palabra se encuentran muy cercanos y aparecen con una tonalidad roja, como es el caso de *application* y *applications, system* y *systems, network* y *networks.*

Se puede apreciar también la relación entre sinónimos, como es el caso de *smart* e *intelligent, control* y *security, model* y *approach.*

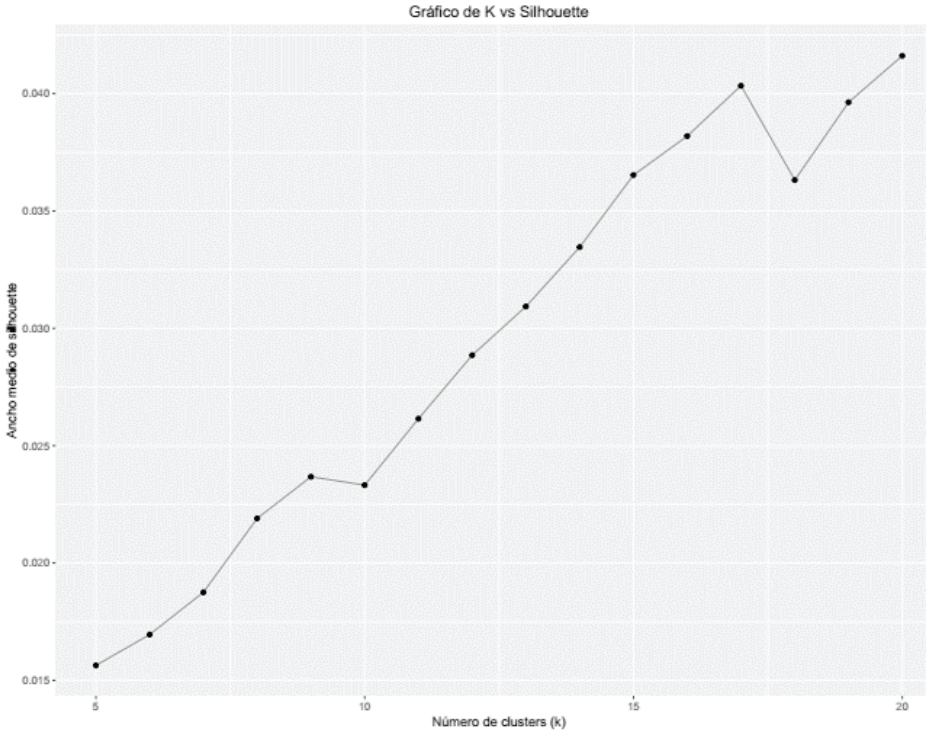
Se dan otras relaciones, como por ejemplo *wireless, devices* y *network*, que podrían dar origen a dispositivos inalámbricos junto con servicios. Podemos decir que los dispositivos inalámbricos brindan servicios.

A partir del modelo, se generaron nubes de palabras de cuyo conjunto se podrían obtener algunos conceptos desde la validación de clúster con el método Silhouette[32,33] para ver cuántos podemos utilizar.

32 https://en.wikipedia.org/wiki/Silhouette_(clustering)

33 http://scikit-learn.org/stable/auto_examples/cluster/plot_kmeans_silhouette_analysis.html

Figura 32. Resultados de evaluación de clústeres con el método Silhouette

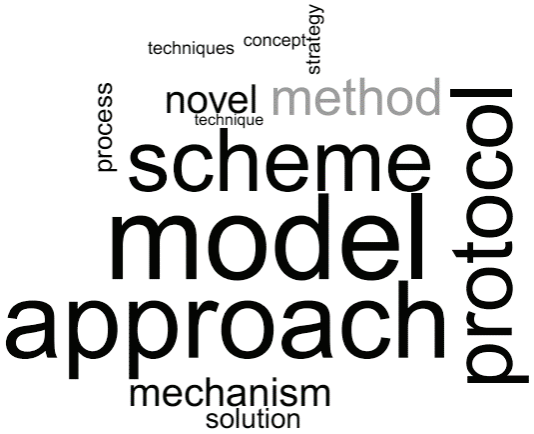


Fuente: creación propia.

Teniendo en cuenta los resultados que se muestran en la figura 32, se decidió formar 19 nubes de palabras o clúster.

De las nubes de palabras obtenidas, 12 tenían un conjunto de palabras con las que se podría generar un concepto o una idea. Algunas de ellas carecían de sentido. Se debe tener en cuenta que para estas pruebas se utilizó un modelo genérico contra una base de datos técnica de un tema en específico.

Figura 33. Ejemplo de nube de palabra con buenos resultados



(El tamaño de las palabras indica la frecuencia con que ocurren en los textos.)
Fuente: creación propia.

La figura 33 nos muestra un ejemplo de nube de palabras con elementos que pueden dar origen a un concepto, teniendo como centro del clúster la palabra *método*. De esta nube, se puede extraer el concepto de un proceso o técnica utilizada para desarrollar una estrategia siguiendo un protocolo.

Algunas nubes formadas no permiten extraer una idea, ya sea por las palabras que se han agrupado o por su cantidad, como es el caso de la mostrada en la figura 34.

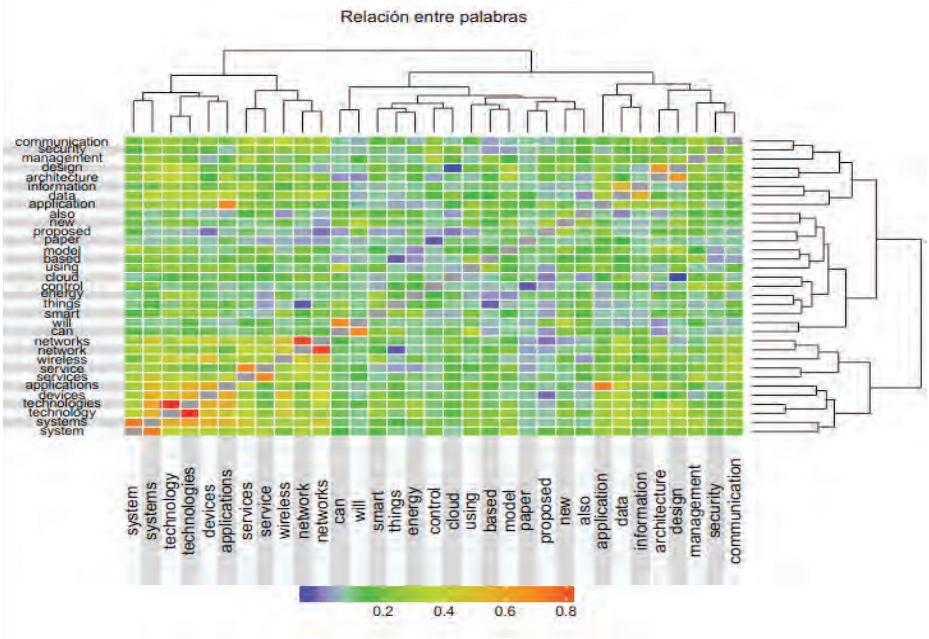
Figura 34. Nube de palabras con poco o nulo sentido

Fuente: creación propia.

• ***Abstract***

La siguiente prueba se llevó a cabo utilizando el *abstract* de las publicaciones. En la mayoría de casos este es mucho más largo que el título, brindando una mayor variedad de palabras. Los resultados obtenidos se muestran en la figura 35.

Figura 35. *Heatmap* con los resultados de usar el *abstract*. Los colores indican la relación entre palabras, siendo las de tonalidad azul las más lejanas entre ellas, y las rojas, las más cercanas.



Fuente: creación propia.

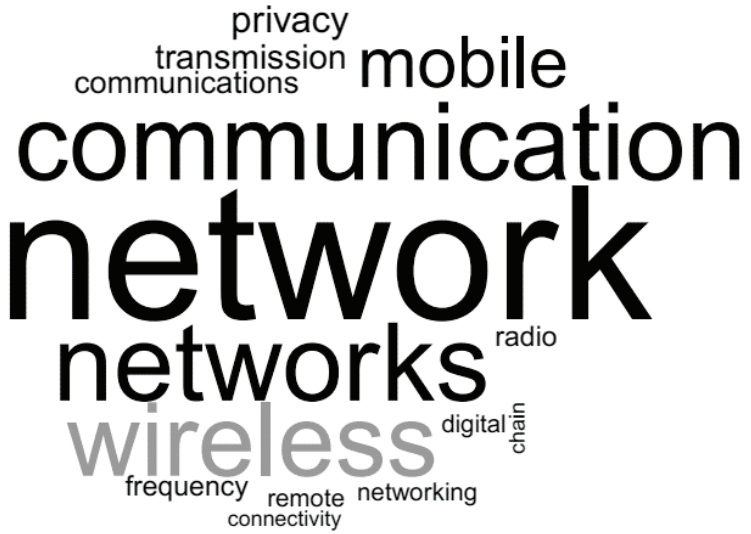
Se puede apreciar que en el *abstract* existe una mayor variedad de palabras y que algunas siguen apareciendo, como es el caso de *technology*, *network*, *wireless* y *model*.

Aparecen además otras nuevas. Como, por ejemplo, *proposed*, *paper* y *new* aparecen juntas formando grupos debido a que son mencionadas constantemente en conjunto.

Las palabras *communication, security, management, design, architecture* e *information* forman un grupo que involucra el concepto de *administración de arquitecturas seguras* para la comunicación y transferencia de información.

Utilizando el *abstract*, 14 de las 19 nubes que se generaron pueden dar origen a ideas y conceptos. Cabe mencionar que, debido a la cantidad mayor de texto y de variedad de palabras, estas aparecen con más variantes, puesto que no se ha aplicado *stemming*; y también aparecen algunas palabras vacías.

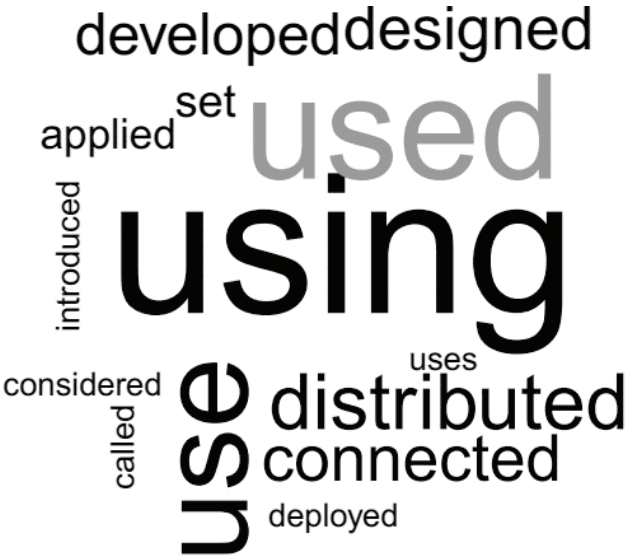
Figura 36. Nube de palabras a partir del abstract. El tamaño de las palabras indica la frecuencia con que ocurren en los textos. A mayor frecuencia, mayor tamaño.



Fuente: creación propia.

La figura 36 nos muestra una nube de palabra que da origen al concepto de *comunicación inal*ámbrica, mientras que la figura 37 nos muestra una nube donde aparece la palabra *use* con sus variantes.

Figura 37. Nube donde se muestra la palabra *use* con algunas variaciones. El tamaño de las palabras indica la frecuencia con que ocurren en los textos. A mayor frecuencia, mayor tamaño.

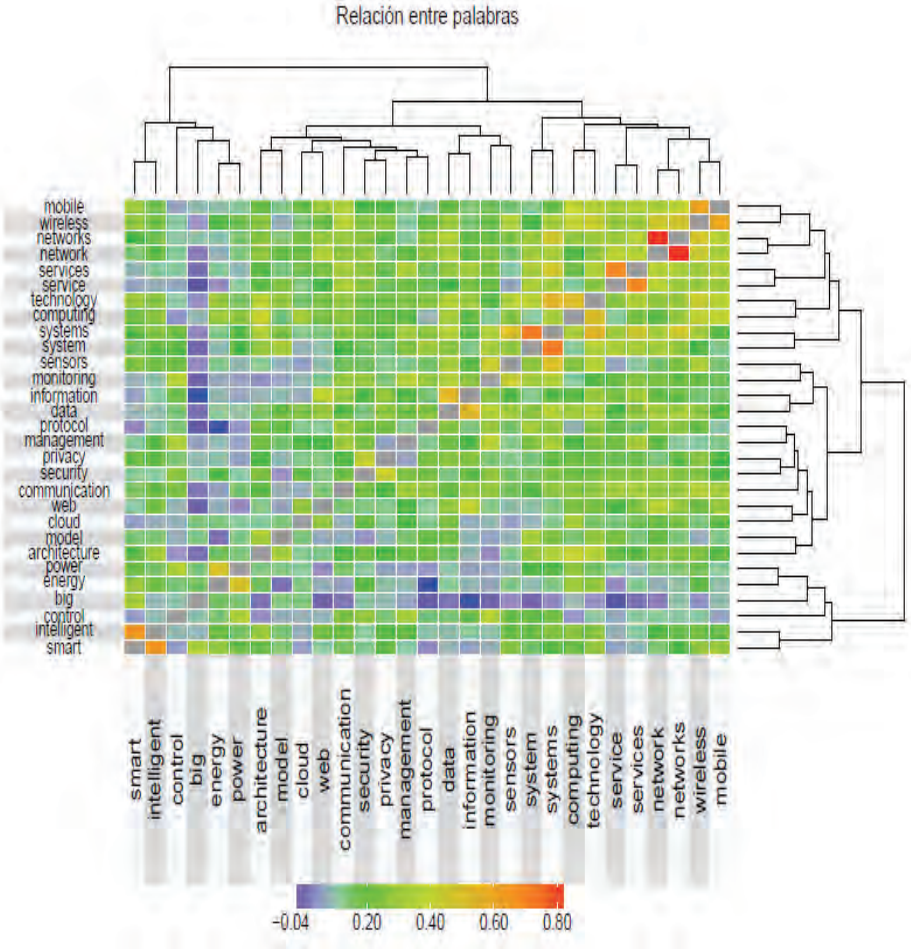


Fuente: creación propia.

- ***Author keywords***

Se desarrollaron pruebas con la sección *Author keywords* y los resultados se muestran en la figura 38.

Figura 38. Heatmap con los resultados de las pruebas sobre *author keywords*. Los colores indican la relación entre palabras, siendo las de tonalidad azul las más lejanas entre ellas, y las rojas, las más cercanas.



Fuente: creación propia.

Podemos ver que aparecen palabras relacionadas como posibles sinónimos, por ejemplo, *data* e *information*, *energy* y *power*, *web* y *cloud*, además la agrupación de *architecture, model, cloud* y *web* puede expresar la idea de que *cloud* y *web* son una arquitectura.

Utilizando las *author keywords*, 15 de las 19 nubes están compuestas por grupo de palabras que pueden dar origen a un concepto. Cabe mencionar que, por ser tan específicas, no aparecen palabras vacías y en las nubes aparecen menos variaciones de las palabras; lo opuesto a lo ocurrido con el *abstract*.

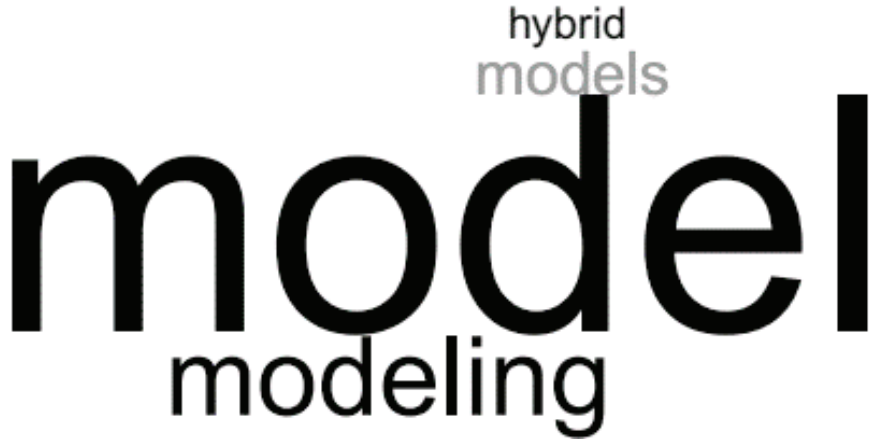
Figura 39. Nube de palabras que resulta del uso de las *author keywords*. El tamaño de las palabras indica la frecuencia con que ocurren en los textos. A mayor frecuencia, mayor tamaño.

Fuente: creación propia.

La figura 39 nos muestra una nube donde aparecen diferentes palabras, dando origen al concepto de *gestión* con todas las palabras completamente diferentes sin variaciones en plural, singular u otros.

Aparecen además algunas nubes con poco contenido para dar origen a un concepto o idea, como se muestra en la figura 40.

Figura 40. Nube de palabras con poco contenido. El tamaño de las palabras indica la frecuencia con que ocurren en los textos. A mayor frecuencia, mayor tamaño.

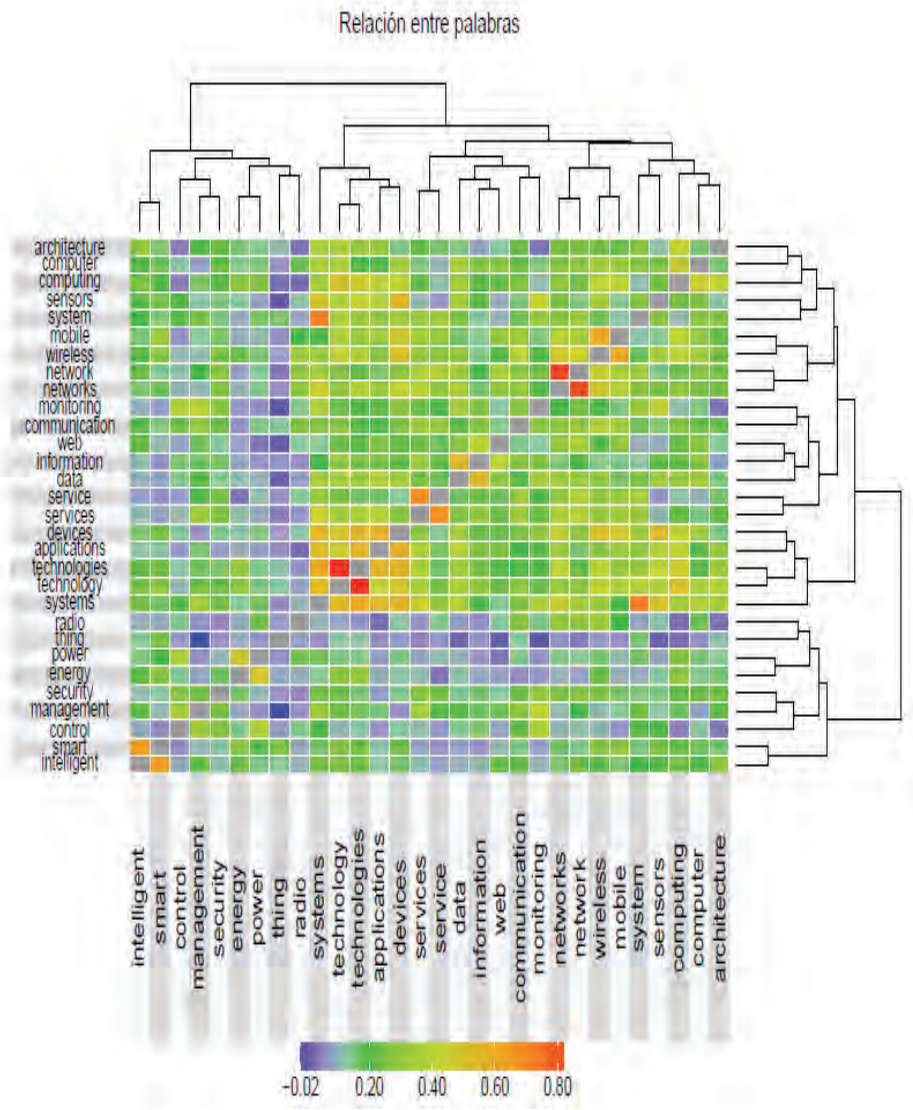


Fuente: creación propia.

- ***Index keywords***

La última prueba realizada fue con las *index keywords*. Existen diferencias entre las *author keywords* y las *index keywords* estas, llegando al punto que en algunos casos uno de los campos está vacío, por lo que era necesario confirmar si las nubes de palabras también presentarían nuevos conceptos.

Figura 41. Heatmap resultante de usar las *index keywords*. Los colores indican la relación entre palabras, siendo las de tonalidad azul las más lejanas entre ellas, y las rojas, las más cercanas.



Fuente: creación propia.

Los resultados visuales son muy parecidos a los de *author keywords*, pero las nubes de palabras han sido distintas, dando como resultado un total de 13, de 19, con información de importancia. Similar al caso anterior, las nubes formadas son bastante específicas, aunque aparecen un poco más de palabras con variaciones, como muestra la figura 42.

Figura 42. Nube de palabras donde aparecen variaciones de *technology*.
El tamaño de las palabras indica la frecuencia con que ocurren en los textos.
A mayor frecuencia, mayor tamaño.

Fuente: creación propia.

Tomando como referencia que se han generado 19 nubes de palabras para cada prueba, la tabla 5 muestra un resumen de los resultados obtenidos:

Tabla 6. Resultado de las nubes de palabras con información de importancia en cada prueba

Sección	Resultados
Título	0.63
Abstract	0.74
Author Keywords	0.79
Index Keywords	0.68

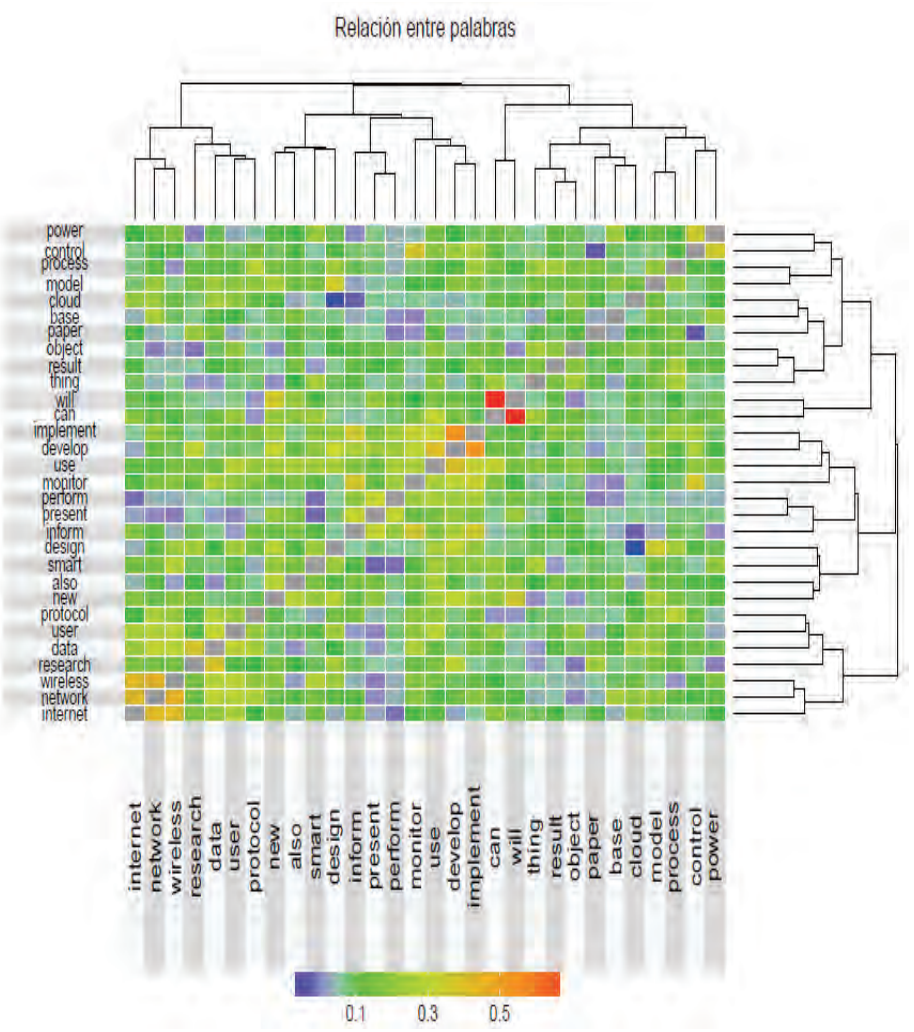
Para obtener los resultados detallados en la tabla, se dividió el total de nubes creadas con el modelo de *Word2Vec* entre aquellas que se consideró por parte del especialista que tenían sentido o contenido de importancia.
Fuente: creación propia.

Se puede observar que los mejores resultados se obtuvieron con las *author keywords*, con nubes más específicas y con una menor cantidad de palabras con variaciones. Muy deferente de lo ocurrido con el *abstract*, donde aparecen palabras con más variaciones y además algunas nubes que están formadas por palabras vacías.

4.3.1.1.2 *Texto con* stemming

Se pudo observar en las pruebas anteriores que muchas palabras aparecían con variaciones, por ejemplo, plural, singular y otras; por lo que en esta sección se aplicó *stemming* para poder generar conceptos con la raíz de las palabras. Se seleccionó solamente *abstract* y *author keywords*, ya que fueron las que mejor resultado obtuvieron.

Figura 43. Resultado de aplicar *stemming* sobre *abstract*. Los colores indican la relación entre palabras, siendo las de tonalidad azul las más lejanas entre ellas, y las rojas, las más cercanas.

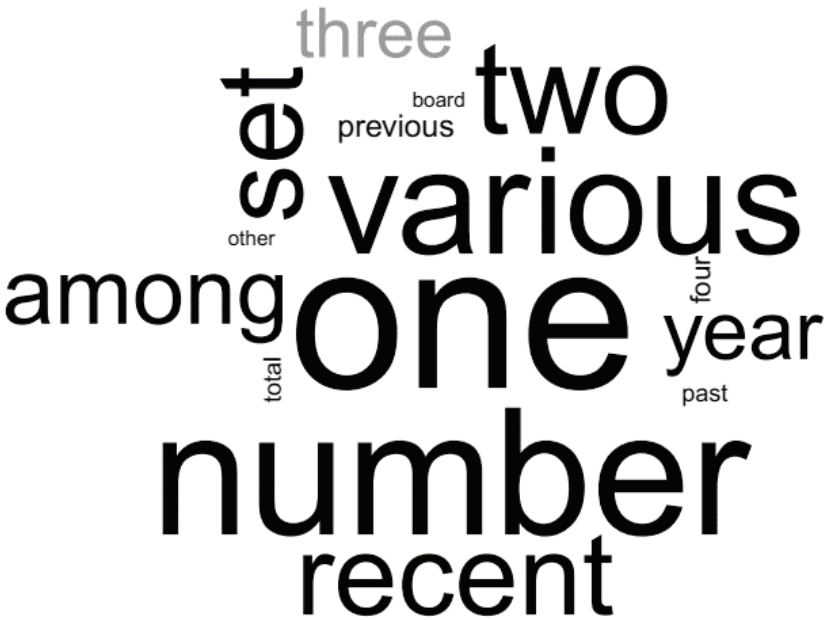


Fuente: creación propia.

En la nueva nube de palabras, siguen apareciendo algunas palabras vacías; pero solamente las palabras en su raíz, por lo que la escala de la distancia entre las palabras ha disminuido. Siguen apareciendo palabras agrupadas con bastante relación, como es el caso de *wireless, network* e *internet, develop* e *implement, model* y *process.*

De las 19 nubes de palabras, 16 están formadas de modo que pueden dar origen a un concepto, sin embargo, al aplicar *stemming* parece que se ha perdido información porque las nuevas nubes hacen referencia a cosas más genéricas y no tan técnicas como en las pruebas anteriores.

Figura 44. Nube de palabras que muestra tiempo y números. El tamaño de las palabras indica la frecuencia con que ocurren en los textos. A mayor frecuencia, mayor tamaño.

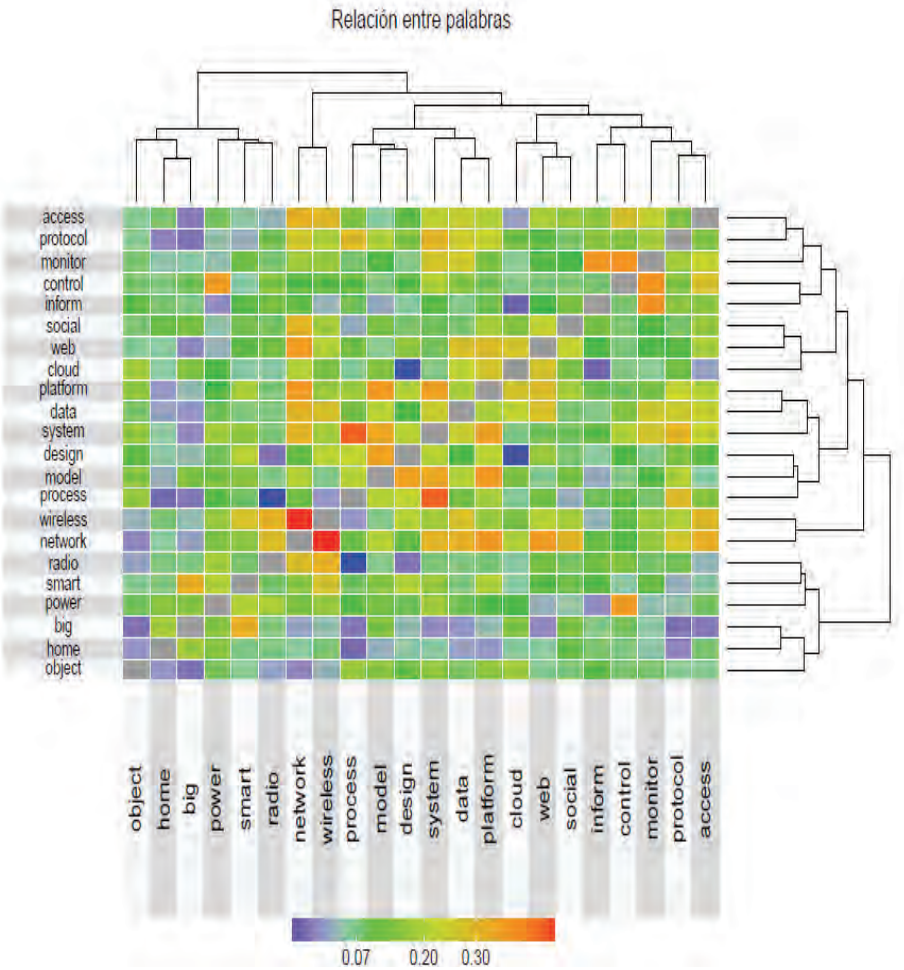


Fuente: creación propia.

- ***Author Keywords***

  Los nuevos clústeres formados son diferentes a los anteriores, están formados por palabras bastante específicas.

Figura 45. Resultado de aplicar *stemming* sobre las *author keywords*. Los colores indican la relación entre palabras, siendo las de tonalidad azul las más lejanas entre ellas, y las rojas, las más cercanas.



Fuente: creación propia.

El resultado para las *Author Keywords* fue de 13 nubes de palabras con información de importancia sobre 19, nuevamente estas nubes hacen mayor referencia a cosas generales que a textos técnicos.

La tabla 7 nos muestra un resumen de los resultados obtenidos:

Tabla 7. Resultados de aplicar *stemming*

Sección	Resultados
Título	0.47
Abstract	0.84
Author Keywords	0.68
Index Keywords	0.74

Para obtener los resultados detallados en la tabla, se dividió el total de nubes creadas con el modelo de *Word2Vec* entre aquellas que se consideró por parte del especialista que tenían sentido o contenido de importancia.
Fuente: creación propia.

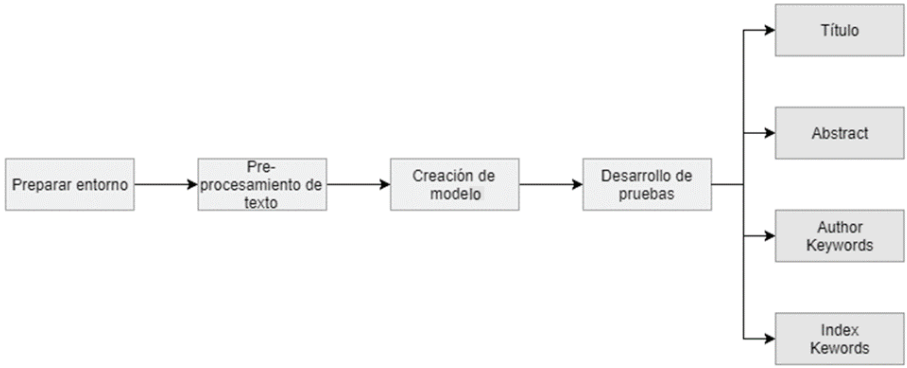
Se puede ver que el título ha sido más afectado, pues disminuyó el número de nubes de palabras con información de importancia; y a nivel general, todas las pruebas dieron como resultado conceptos o información más genéricos.

4.3.1.2 *Modelo específico*

El modelo empleado en las pruebas anteriores fue creado a partir de textos cotidianos, no de un entorno técnico o de una rama en específico, por lo que surgieron algunas nubes de palabras y relaciones que hacían referencia a este tipo de situaciones.

El diagrama de la figura 46 muestra los pasos seguidos para tener un modelo propio.

Figura 46. Procedimiento para creación de un modelo *Word2Vec*



Fuente: creación propia.

4.3.1.2.1 *Preparando el entorno*

Para crear un modelo propio, fue necesario hacer uso del paquete Word Vectors[34] en R, también se hace uso del paquete DevTools;[35] en algunas versiones de R es necesario instalar RStan[36] para que DevTools funcione adecuadamente.

4.3.1.2.2 *Preprocesamiento de texto*

Antes de crear el modelo, fue necesario hacer preprocesamiento de texto debido a que algunos caracteres podían afectar los resultados, por ejemplo, los signos de puntuación, otros caracteres extranjeros y el manejo del guion medio (-).

34 https://github.com/bmschmidt/wordVectors
35 https://cran.r-project.org/web/packages/devtools/index.html
36 https://github.com/stan-dev/rstan/wiki/Installing-RStan-on-Windows

Para obtener los mejores resultados posibles, en el preprocesamiento se incluyó lo siguiente:

- Convertir el texto de la base de datos (*title, abstract, author keywords* e *index keywords*) a Corpus.
- Pasar todo el texto a minúsculas.
- Sustituir el guion medio "-" por guion bajo "_".
- Eliminar todos los signos de puntuación, excepto el guion bajo "_".
- Eliminar los apóstrofes.
- Eliminar los números y espacios en blanco.

4.3.1.2.3 *Creación del modelo*

Teniendo el entorno en R preparado y el texto preprocesado, los parámetros fueron los siguientes:

- ***threads* = 3:** indica el número de núcleos que se han de usar en la PC donde se va a crear el modelo.
- ***vectors* = 500:** número de atributos que se van a generar. De acuerdo con *papers*, entre 300 y 500 es recomendable.
- ***Window* = 10**: número de palabras que se utilizarán desde la base.
- Por defecto, el modelo se crea con Skyp-Gram; y es el que se debe utilizar.
- ***negative_samples*:** de acuerdo con *papers*, 5-15 para *dataset* pequeños, y 2-5 para grandes, se utilizan 5 como valor del parámetro en la creación de nuestro modelo.

La línea de comandos en R es la que se muestra en la figura 47.

Figura 47. Línea de comandos para generar el modelo.

```
model = train_word2vec("ScopusAllData_.txt",output="ScopusAllData_vectorsNeg_.bin",
                       threads = 3,vectors = 500,window=10,
                       negative_samples = 5, min_count = 10)
```

Fuente: creación propia.

- **Pruebas sobre el modelo**

Luego de crear un modelo con *Word2Vec*, se procedió a realizar algunas pruebas para ver el funcionamiento, por ejemplo palabras que se encuentran cercanas a una específica, lo cual podemos hacer de dos formas: *closest to*, que toma la palabra como 1 y va disminuyendo el valor a medida que una palabra se aleja; y *nearest to*, que toma la palabra como 0 (cero) la palabra de referencia y va incrementando la distancia a medida que se va alejando.

Figura 48. Resultados de utilizar *closest to*

```
                word similarity to model[["mobile"]]
mobile                                     1.0000000
handsets                                   0.5326911
telephones                                 0.4705289
phones                                     0.4683068
android                                    0.4024062
pda                                        0.3980577
smart_phone                                0.3821995
telephone                                  0.3807276
ubiquitous                                 0.3739598
telecommunication                          0.3736325
```

Fuente: creación propia.

Figura 49. Resultados de utilizar *nearest to*

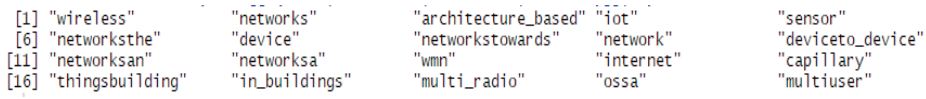
```
mobile       handsets     telephones       phones       android          pda
 0.000          0.467         0.529        0.532         0.598        0.602
smart_phone  telephone    ubiquitous telecommunication
 0.618          0.619         0.626        0.626
```

Fuente: creación propia.

Tanto en la figura 48 como en la 49, podemos ver que existen unas palabras cerca de *mobile* que pueden ser sinónimas o generar una idea entre todas.

Un aspecto interesante de la función *nearest to* es que, mediante un grupo de palabras y cierta cantidad que se encuentren cerca, podemos generar agrupamientos para extraer una temática, como se muestra en la figura 50.

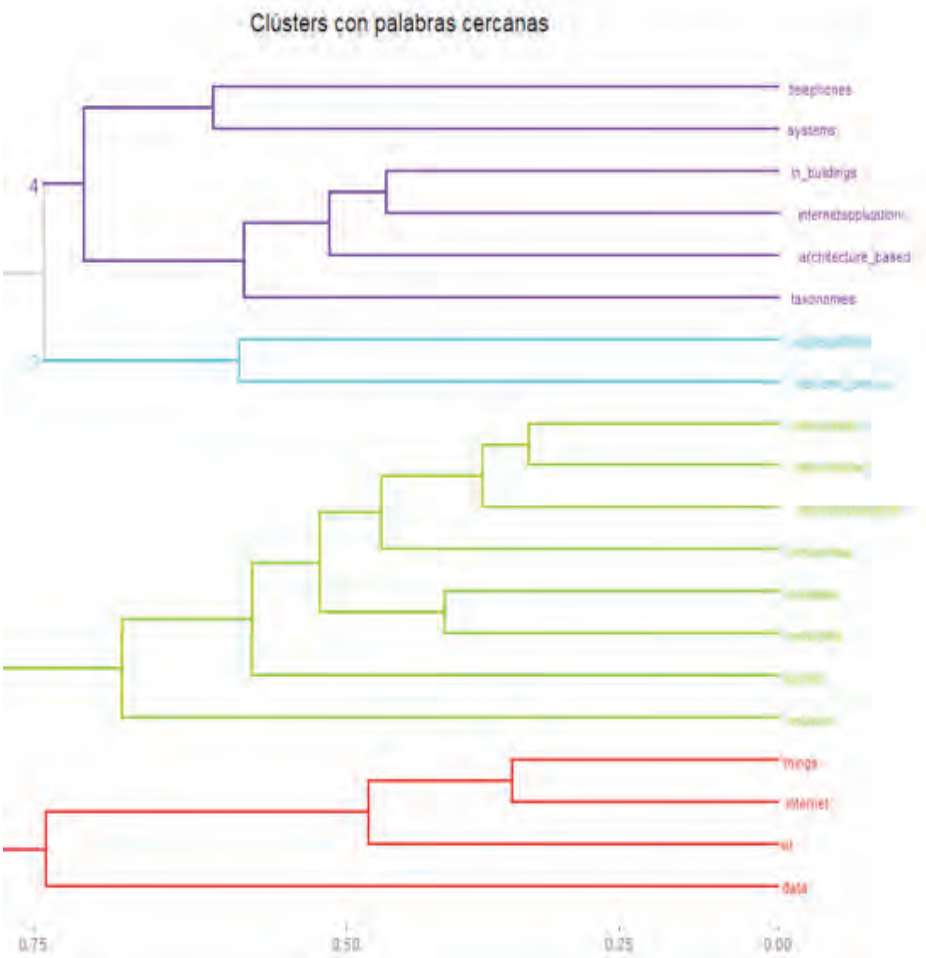
Figura 50. Grupo de palabras formado con la función *nearest to* y que toma como palabras base *iot*, *wireless, networks* y *device*



Fuente: creación propia.

El grupo de palabras de la ilustración 50 nos puede dar una idea de un concepto, sin embargo, no representa una relación entre ellas, los niveles o una ontología propiamente dicha. Para ello, es necesario hacer uso de dendogramas que nos muestren una mejor representación visual.

Figura 51. Clústeres formados a partir de una lista de palabras



Fuente: creación propia.

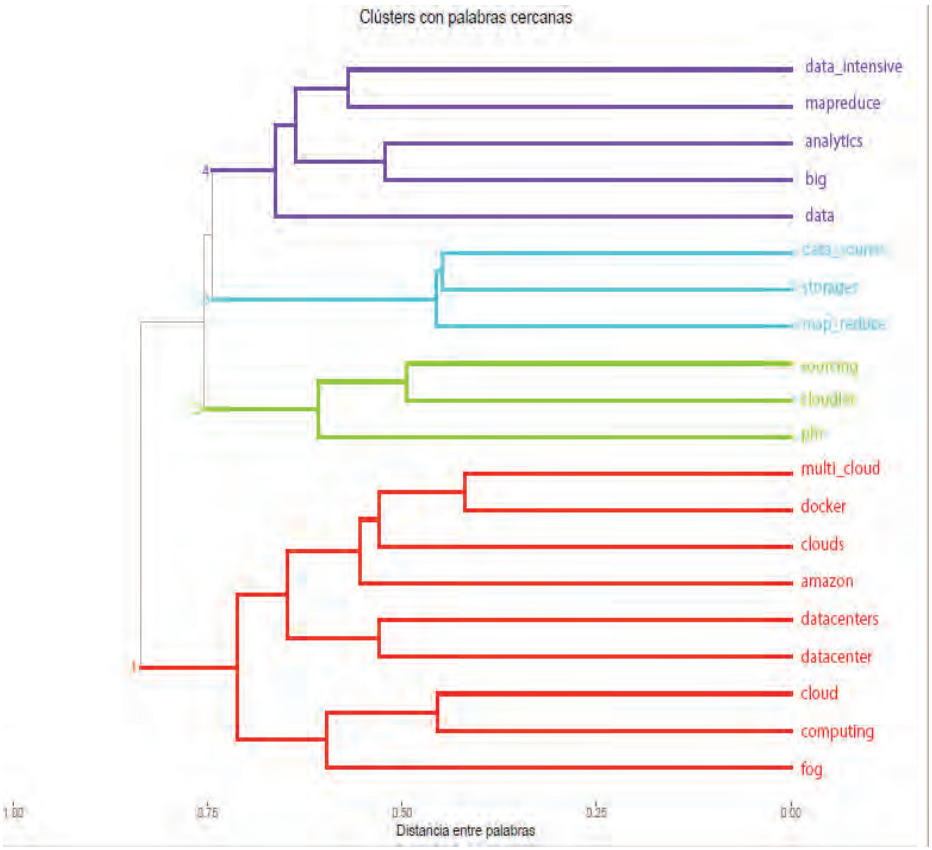
En la figura 51 podemos ver cuatro grupos formados a partir de un conjunto de palabras cercanas a una lista que puede ser definida (el número de palabras cercanas también puede ser decidido). Para este ejemplo se han seleccionado 20.

De los cuatro clústeres, tres proporcionan información y niveles para formar conceptos o ideas. El primero, en color rojo, hace referencia

propiamente a *Internet of Things* y a las siglas utilizadas *iot*; el segundo, en color verde, representa los términos de red inalámbrica y sensores; y el cuarto, en color púrpura, representa a los teléfonos como sistemas donde se ejecutan aplicaciones.

Como segunda prueba, se presenta los clústeres formados con las palabras *cloud computing* y *big data,* con los resultados mostrados en la figura 52.

Figura 52. Dendrograma formado a partir de las frases *cloud computing* y *big data*



Fuente: creación propia.

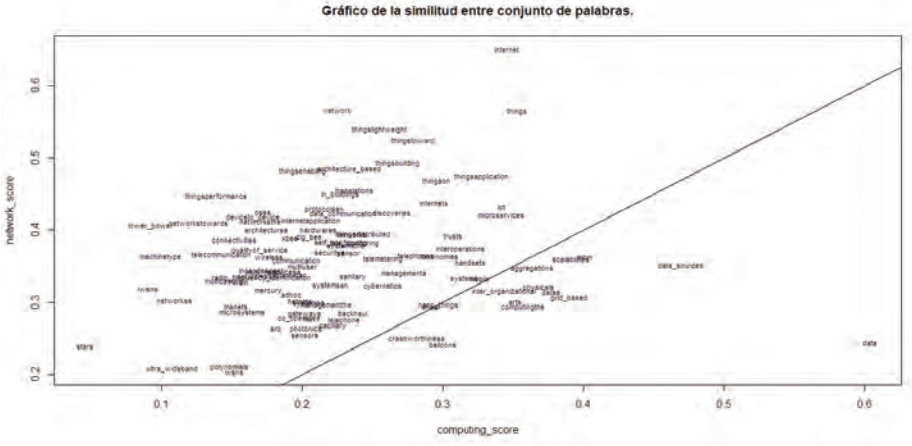
En el dendrograma podemos ver un clúster bien marcado que da origen al concepto de *big data*, incluyendo elementos que están involucrados con el concepto, por ejemplo Docker,[37] *data center* y Amazon, compañía esta última que tiene bastante relación con *big data*.

Se puede observar, además, que las frases *data centers*, en plural y singular, están conectadas en un mismo nivel. Durante el procesamiento de texto no se aplicó *stemming* para poder obtener la mayor cantidad de información posible; lo mismo se ha hecho con las palabras vacías.

Otra representación de la información es mediante un gráfico que muestre la similitud entre palabras, siendo aquellas más distantes las que menos tienen en común.

La figura 53 está formada a partir de dos grupos de palabras base: 1. *cloud, computing, big* y *data,* tratando de hacer referencia al *cloud computing*, y 2. *internet, thing, network* y *connection*, y con el término *Internet of Things*. Se puede observar que el grupo está marcado por una línea que funciona como frontera o límite entre las frases o palabras que están cercanas a ambos términos o las que se alejan dando origen a posibles antónimos.

Figura 53. Similitud entre palabras

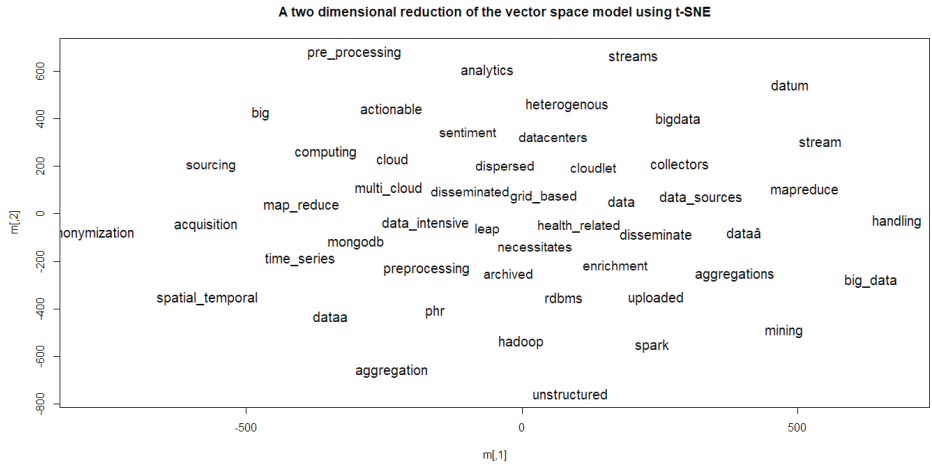


Fuente: creación propia.

37 https://aws.amazon.com/es/docker/

Se puede ver que ambos términos se encuentran completamente separados; pero un grupo de palabras se encuentra cerca de la frontera, lo que indica que tienen relación.

Figura 54. Representación binimensional usando t-SNE



Fuente: creación propia.

La ilustración 54 muestra un cuadrante donde la distancia entre palabras indica su relación.

4.3.1.2.4 *Comparación de resultados*

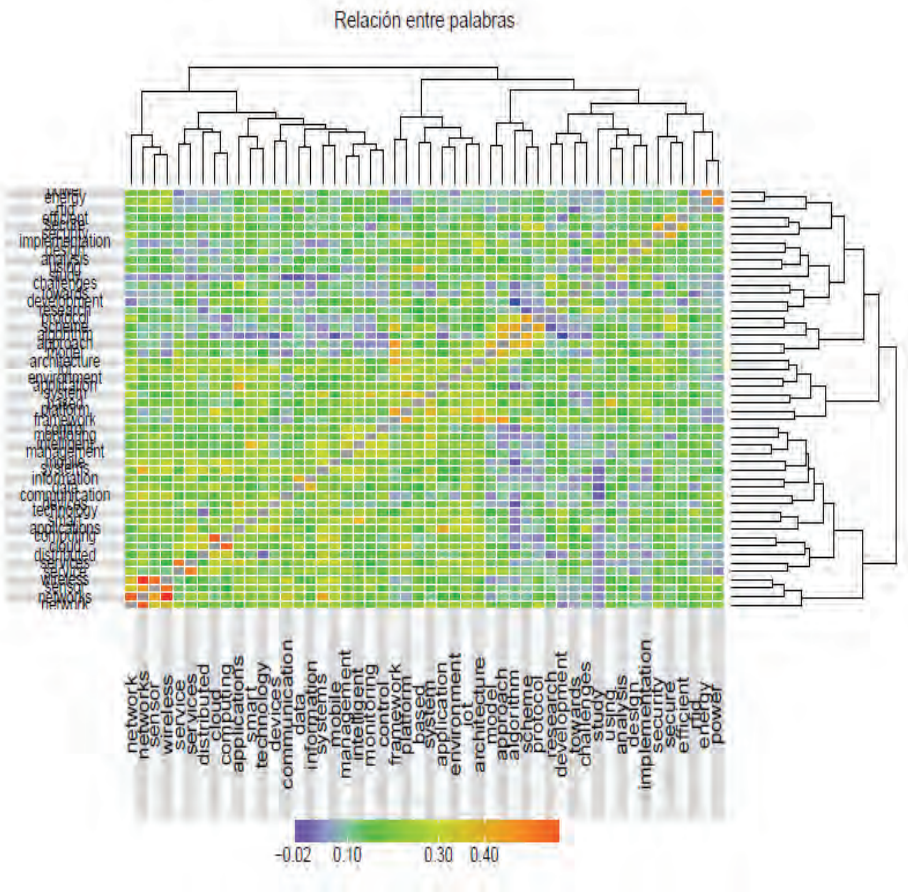
Para verificar si existe una variación en los resultados, tanto en la cantidad de nubes de palabras con un sentido como en su calidad, se realizaron las mismas pruebas que con el modelo genérico.

4.3.1.2.4.1 Sin stemming

Title

En la figura 55, se puede observar que la distancia entre palabras ha disminuido en comparación con las del título, utilizando el modelo genérico, sin embargo, no aparecen palabras genéricas y las estructuras son más interesantes.

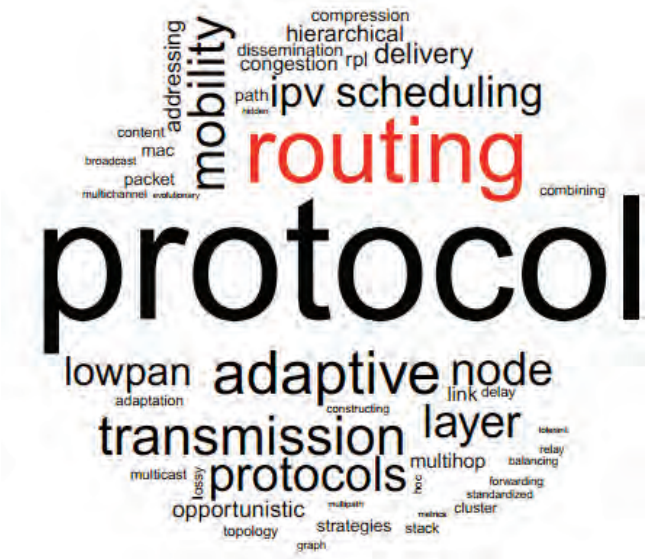
Figura 55. Resultados de aplicar el modelo sobre el título



Fuente: creación propia.

De las nubes de palabras generadas, 17 de 19 están formadas por palabras que dan origen a un posible concepto; y se puede ver que son conceptos mucho más técnicos o relacionados con el tema elegido al principio como base de datos.

Figura 56. Ejemplo de nube de palabras con Title

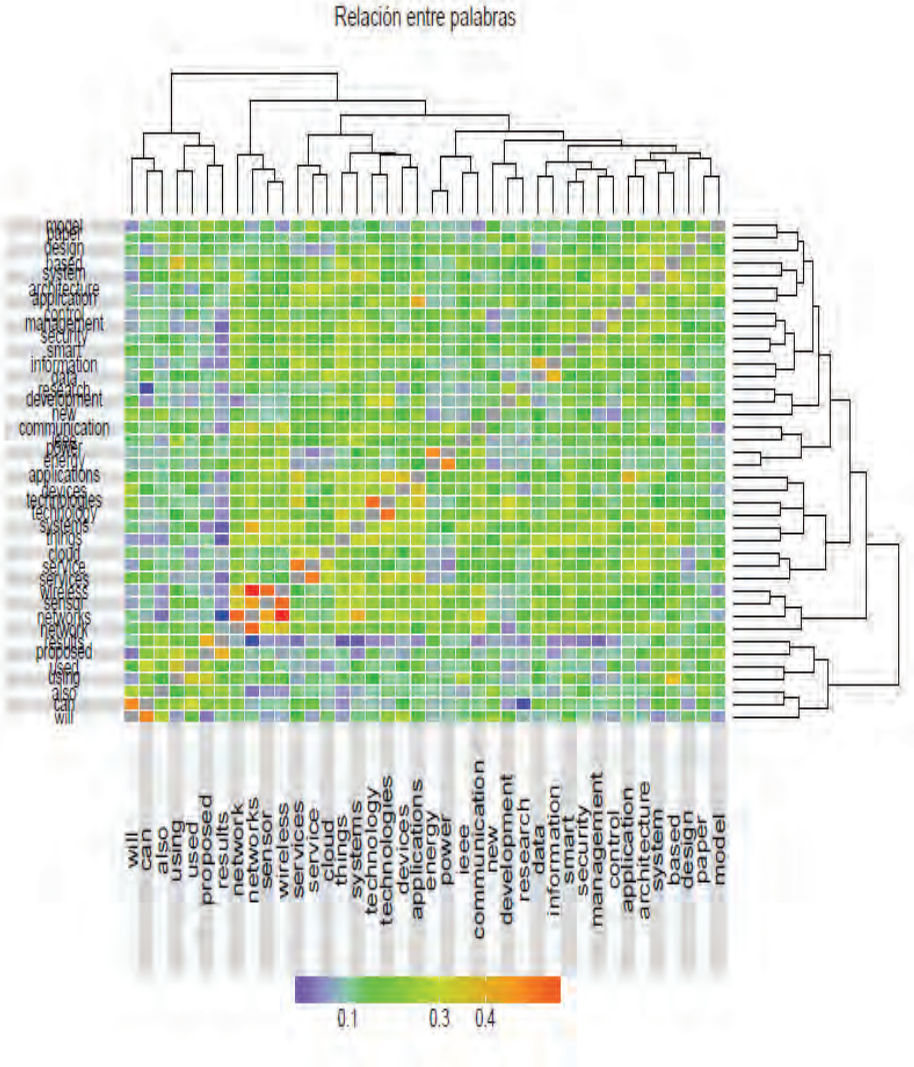


Fuente: creación propia.

Pruebas con el *abstract*

Al utilizar el *abstract*, aparecen nuevamente palabras vacías; pero esta vez se encuentran agrupadas al contrario del modelo genérico en el que aparecían mezcladas, lo que hace ver que son diferentes a las demás palabras, pero similares o relacionadas entre ellas. Esto se puede ver en la figura 57.

Figura 57. *Heatmap* con los resultados de aplicar el modelo sobre *Abstract*. Los colores indican la relación entre palabras, siendo las de tonalidad azul las más lejanas entre ellas, y las rojas, las más cercanas.



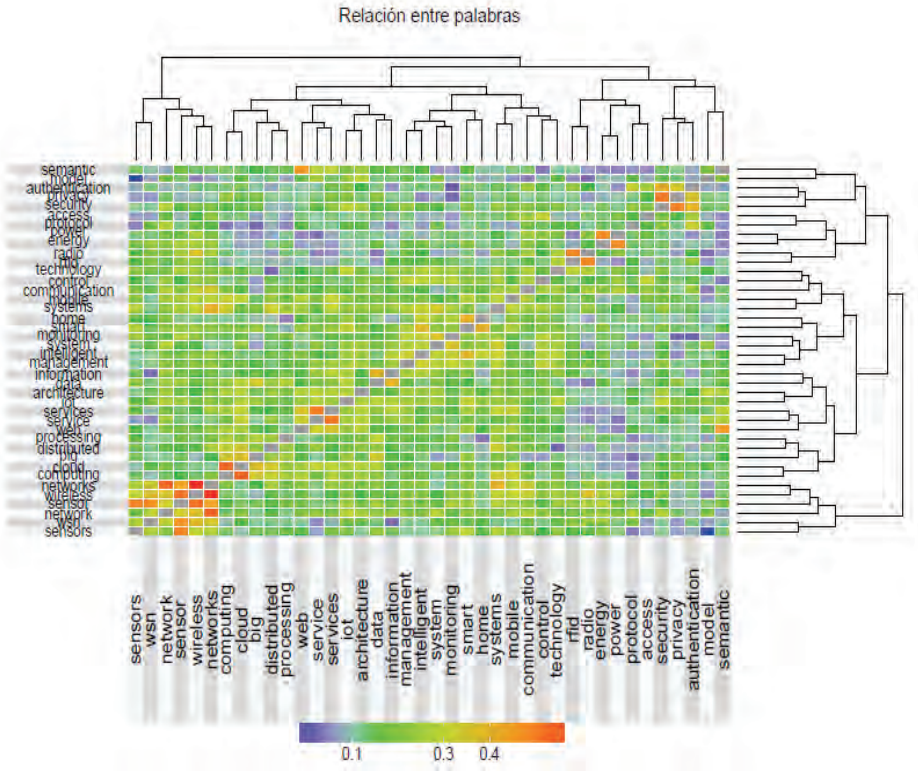
Fuente: creación propia.

Utilizando el *Abstract*, 15 de las 19 nubes de palabras han generado información de importancia. Aparecieron palabras vacías en las nubes y también se muestran algunas palabras de uso general, lo que es aceptable debido a que en esta sección los autores tienen la libertad de una mayor expresión y un lenguaje más amplio y con una extensión mucho más larga que los títulos.

Author keywords

Con las *author keywords*, como se muestra en la figura 58, aparecen palabras con bastante relación, por ejemplo, *sensors* y *WSN (Wireless Sensor Network)*, lo cual es interesante porque pueden ser tomadas como sinónimas, o que una incluye a la otra, siendo la *WSN* un tipo de sensor específico. Otras palabras parece que basan su relación en el hecho de que siempre aparecen unidas, como *wireless networks, web service, cloud computing* y *semantic mode*, por lo que juntas pueden tener un significado diferente que cuando están separadas.

Figura 58. *Heatmap* con los resultados de aplicar el modelo sobre *author keywords*



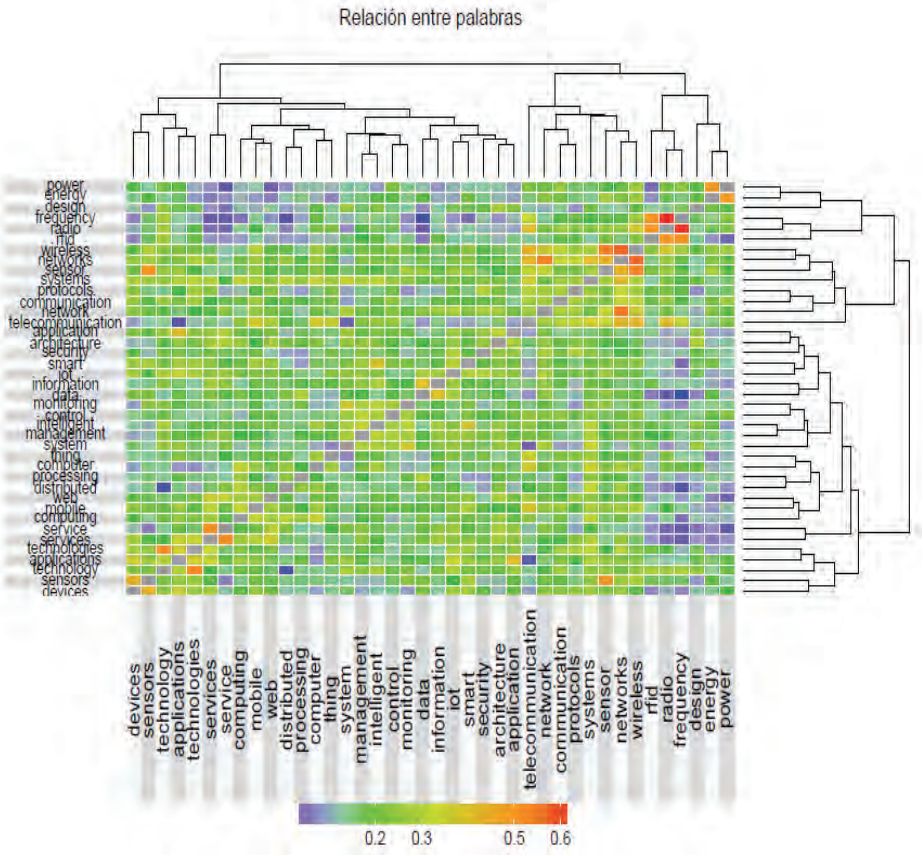
Fuente: creación propia.

Los resultados con las nubes de palabras han sido 15 de 19 con una estructura de la cual se puede extraer información con significado.

Index keywords

La última prueba desarrollada fue utilizando las *index keywords*. En la figura 59, podemos visualizar una relación muy interesante entre *techonology*, *service* y *techonologies*, donde las dos últimas aparecen en un mismo nivel de agrupamiento; y la primera como segundo nivel, a pesar de que es una variación de la primera. Además, si bien algunas palabras se repiten con las *author keywords*, varía la relación entre ellas.

Figura 59. *Heatmap* con los resultados de aplicar el modelo sobre las *index keywords*. Los colores indican la relación entre palabras, siendo las de tonalidad azul las más lejanas entre ellas, y las rojas, las más cercanas.



Fuente: creación propia.

En esta parte, se obtuvieron 17 nubes con estructura para obtener información interesante. Es necesario mencionar que, además de la cantidad de nubes con significado que han ido surgiendo, la calidad representada también ha mejorado, reduciendo la aparición de palabras genéricas.

Para tener una comparativa de los resultados obtenidos en esta sección, estos han sido detallados en la tabla 8 para poder ver cómo se comporta cada una de las secciones de los textos.

Tabla 8. Comparativa de los resultados obtenidos con las diferentes secciones de los textos

Sección	Resultados
Título	0.47
Abstract	0.84
Author Keywords	0.68
Index Kewords	0.74

Para obtener los resultados detallados en la tabla, se dividió el total de nubes creadas con el modelo de *Word2Vec* entre aquellas que se consideró por parte del especialista que tenían sentido o contenido de importancia.
Fuente: creación propia.

4.3.1.2.4.2 *Pruebas con* stemming

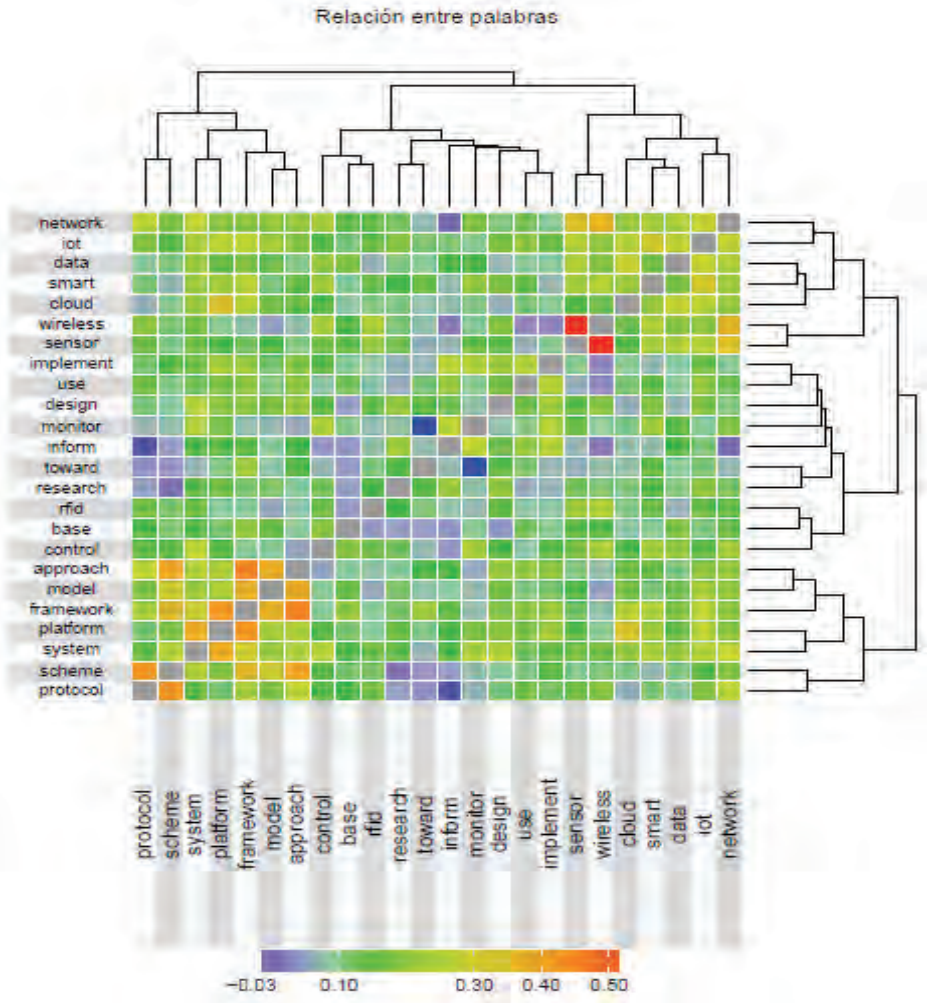
Con el modelo genérico se pudo observar que al aplicar *stemming* los resultados podrían variar, puesto que en el *title* y en las *author keywords* hubo una pequeña disminución en la cantidad de nube de palabras con sentido, mientras que en el *abstract* y en las *index keywords,* para comparar los resultados, se aplicó también sobre el texto que se debía comparar con el modelo que se creó.

Title

En el título ha aparecido una menor cantidad de palabras relacionadas; nuevamente algunas de ellas pueden ser utilizadas como sinónimos.

Aparecen unas palabras con valores muy cercanos a cero lo cual indica que tienen una relación casi nula. Se puede apreciar la representación en la figura 60.

Figura 60. Heatmap con los resultados de hacer uso de *title* con *stemming*. Los colores indican la relación entre palabras, siendo las de tonalidad azul las más lejanas entre ellas, y las rojas, las más cercanas.



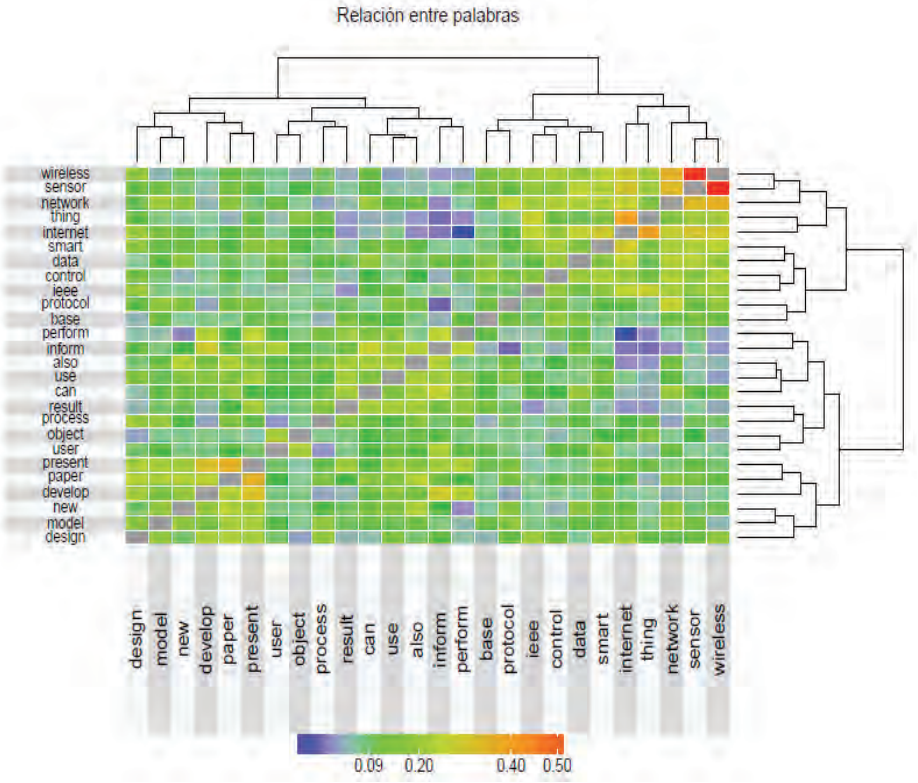
Fuente: creación propia.

De las nubes de palabras, solamente 11 de las 19 agrupan palabras que pueden ser de utilidad para extraer información; algunas de ellas un poco menos en comparación de las otras.

Abstract

En esta etapa también algunas palabras vacías aparecen juntas y otras que pueden dar origen a conceptos. Podemos tener una idea de las relaciones por medio de la figura 61.

Figura 61. Heatmap resultante de aplicar el modelo sobre *abstract* utilizando *stemming*. Los colores indican la relación entre palabras, siendo las de tonalidad azul las más lejanas entre ellas, y las rojas, las más cercanas.



Fuente: creación propia.

Para esta etapa, el resultado fue 17 de las 19 nubes de palabras con información de interés.

Los resultados finales de las pruebas pueden apreciarse en la tabla 9.

Tabla 9. Resultado en cada una de las pruebas

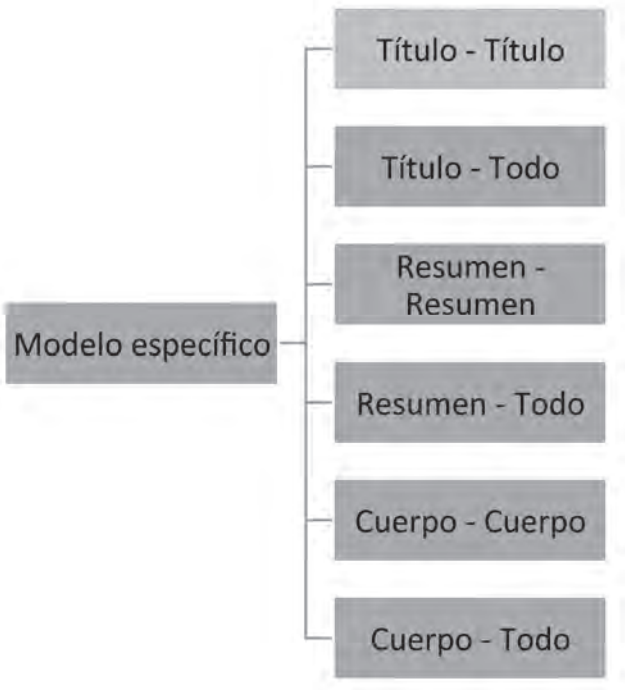
Sección	Resultados
Título	0.58
Abstract	0.89
Author Keywords	0.63
Index Keywords	0.63

Para obtener los resultados detallados en la tabla, se dividió el total de nubes creadas con el modelo de *Word2Vec* entre aquellas que se consideró por parte del especialista que tenían sentido o contenido de importancia.
Fuente: creación propia.

4.3.1.3 *Pruebas sobre los artículos proporcionados por la unidad de datos de* El Diario de Hoy

Luego del aprendizaje sobre el uso de *Word2Vec*, y viendo las ventajas de crear un modelo propio sin hacer uso del *stemming*, y eliminando las palabras vacías, se procedió a hacer el análisis de los artículos proporcionados por la Unidad de Datos de *El Diario de Hoy*; y se desarrollaron las pruebas que aparecen en la figura 62.

Figura 62. Pruebas desarrolladas con los artículos proporcionados por la Unidad de Datos de *El Diario de Hoy*



Fuente: creación propia.

En las pruebas, se utilizó el texto de cada parte de los artículos para la creación de modelos independientes y un modelo conteniendo todas las partes. De igual manera que las etapas anteriores, se generaron 19 nubes de palabras que fueron evaluadas por los miembros de la Unidad de Datos de *El Diario de Hoy*, seleccionando aquellas que tuvieran importancia periodística, es decir, que permitieran generar ideas de lo que se habla, obteniendo los siguientes resultados:

Tabla 10. Resultados obtenidos con las diferentes partes de los textos

Titular-titular	0.79
Titular-all	0.84
Resumen-resumen	0.58
Resumen-All	0.68
Cuerpo-cuerpo	0.68
Cuerpo-All	0.63

Para obtener los resultados detallados en la tabla, se dividió el total de nubes creadas con el modelo de *Word2Vec* entre aquellas que se consideró por parte del especialista que tenían sentido o contenido de importancia.
Fuente: creación propia.

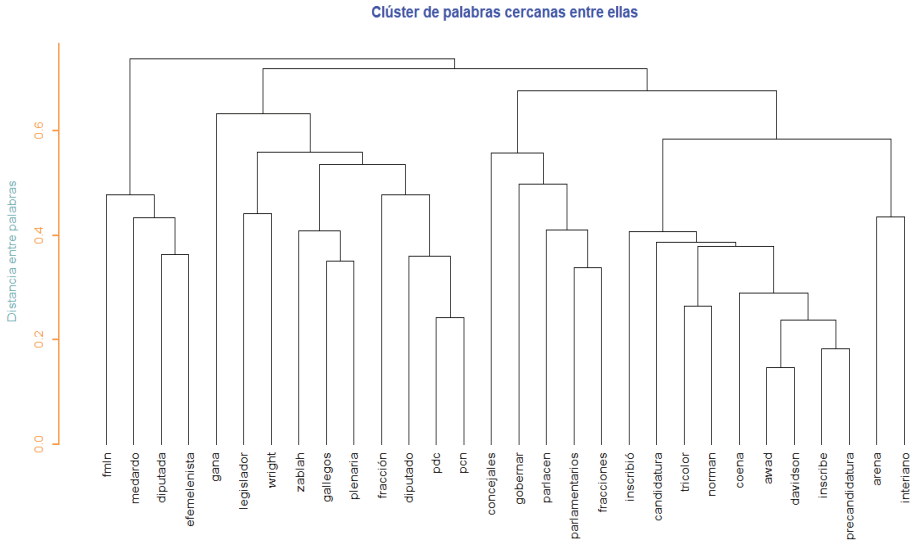
Los mejores resultados obtenidos, para tener una idea de lo que se habla en los artículos, han sido a partir de los textos proporcionados por los titulares, mejorando un poco al mezclar todos los textos y los resultados más bajos han sido utilizando el cuerpo de los textos.

Figura 63. Nube de palabras creada a partir del texto de los titulares. El tamaño de las palabras indica la frecuencia con que ocurren en los textos. A mayor frecuencia, mayor tamaño.

Fuente: creación propia.

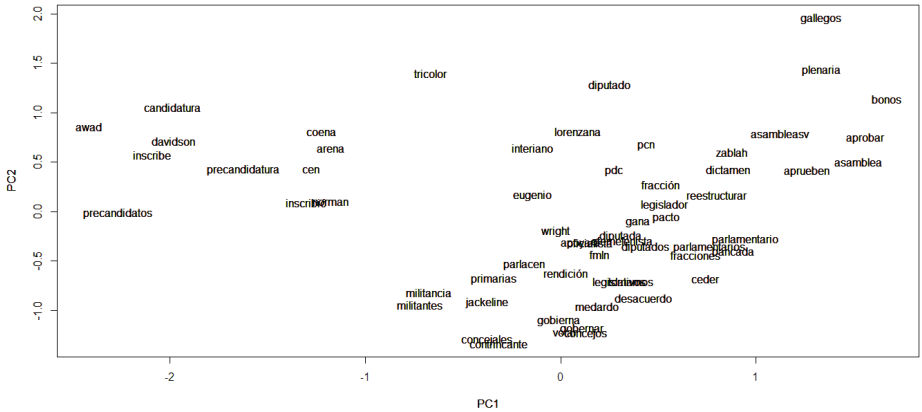
Otra forma de extraer la información de los textos es utilizando clústeres que nos permiten ver la relación entre las palabras m*ás frecuentes de los textos y la* representación en dos dimensiones, con la ubicación de las palabras de acuerdo con la distancia o relación que existe entre ellas.

Figura 64. Clúster que representa la relación entre las palabras más comunes de los textos



Fuente: creación propia a partir del modelo.

Figura 65. Representación de la relación entre palabras en dos dimensiones



Fuente: creación propia a partir del modelo.

4.3.3 *Pruebas con las cuentas de Twitter*

Debido a que los tuits vienen con una gran cantidad de caracteres que pueden no ser relevantes para su análisis, es necesario un proceso de limpieza. En este punto se hicieron varias pruebas hasta lograr que el texto quedase completamente libre de elementos innecesarios.

Algunos de los problemas más comunes incluyen la presencia de URL, ya que muchas se encuentran incompletas y las expresiones regulares no las detectaban todas, por lo que se hizo una combinación de expresiones regulares que abarcaras todas las posibles formas de URL. Convertir todas las palabras a minúsculas también presentaba algunos errores, debido a eso se utilizó una función que detectara y eliminara los errores.

Puesto que las palabras utilizadas por las personas al momento de redactar los tuits son muy variadas, no todas se incluyen en las listas disponibles en R, por lo que es necesario agregar otras listas para completarla y obtener mejores resultados.

Además, se hizo una limpieza propia del contenido de los tuits de la forma siguiente:

1. Eliminar las entidades de retuit.
2. Eliminar los signos de arroba (@) y el texto irrelevante.
3. Eliminar todos los símbolos no numéricos o que no estén en el idioma inglés.
4. Eliminar los *hashtags*.

Finalmente se eliminaron los números y los signos de puntuación se sustituyeron por espacios en blanco. En caso de hacer búsquedas en español, además de una nueva lista de palabras vacías, es necesario eliminar tildes y caracteres especiales, para que no haya problemas en la visualización de los resultados.

Figura 66. Librería utilizada para eliminar tildes y caracteres especiales del idioma español

```
#Just for Spanish text
library(stringi)
tweets.corpus <- stri_trans_general(tweets.corpus, "Latin-ASCII")
tweets.corpus <- Corpus(VectorSource(tweets.corpus))
```

Fuente: creación propia a partir del modelo.

No se implementó el *stemming* para la reducción de sus palabras a sus raíces, ya que en este proceso se puede perder información. En caso de que se quiera hacer una aplicación dedicada a la búsqueda de un tema en específico, se puede personalizar el proceso para ciertas palabras.

4.3.3.1 *Representación de los datos*

Para la extracción de conocimiento a partir de los tuits, es necesario valerse de diferentes técnicas de representación gráfica; como consecuencia, luego de la etapa de procesamiento y limpieza, se utilizaron tres técnicas para cada cuenta, que se detallan a continuación.

4.3.3.2 *Frecuencia*

El texto cuenta con una frecuencia muy variada de las palabras; algunas aparecen solamente una vez y otras un número mayor a cien veces, por lo que se representan solamente las 20 que más se repiten. No se ha definido un número mínimo de repeticiones, sino que se seleccionan aquellas que aparecen con mayor frecuencia.

4.3.3.3 *Relación entre palabras*

Para la relación de palabras y las siguientes gráficas, es necesario obtener una *term-document matrix* (TDM), o matriz de espacio vectorial, que se genera a partir de un corpus creado con el texto procesado de los tuits. La TDM es un objeto muy importante en el análisis de texto; es la matriz de documento de términos porque nos permite almacenar una biblioteca completa de texto dentro de una matriz única. Esto puede usarse para el análisis y para buscar documentos; forma la base de los motores de búsqueda, análisis de temas y clasificación [filtrado de *spam*] (Das, 2017).

Figura 67. Ejemplo de matriz de términos generada a partir del corpus que contiene los textos de los tuits preprocesados

```
<<TermDocumentMatrix (terms: 11, documents: 8)>>
Non-/sparse entries: 14/74
Sparsity           : 84%
Maximal term length: 12
Weighting          : term frequency (tf)
Sample             :
             Docs
Terms         11 12 13 14 15 16 17 18
  artificial   0  1  1  1  1  1  1  1
  center       0  0  0  0  0  0  0  0
  charles      0  0  0  0  0  0  0  0
  company      0  0  0  0  0  0  0  0
  congrats     0  0  0  0  0  0  0  0
  intelligence 0  1  1  1  1  1  1  1
  invents      0  0  0  0  0  0  0  0
  nigeria      0  0  0  0  0  0  0  0
  onu          0  0  0  0  0  0  0  0
  syst         0  0  0  0  0  0  0  0
```

Fuente: creación propia a partir del modelo.

4.3.3.4 *Nube de palabras*

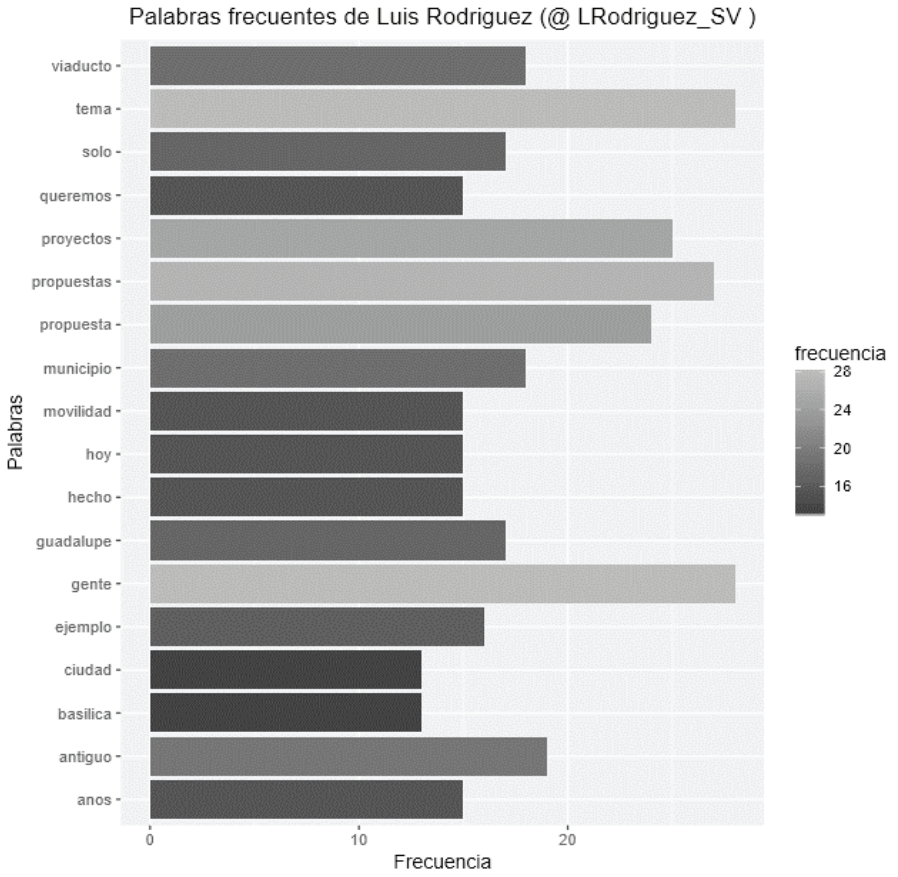
Las nubes de palabras permiten visualizar la información de los textos en forma clara y sencilla con un mayor número de palabras, cuyo tamaño nos indica la frecuencia con que aparecen en los textos. Mientras mayor sea el tamaño, mayor será la frecuencia.

4.3.3.5 *Resultados por cuentas*

@LRodriguez_SV

De la cuenta de @LRodriguez_SV, se obtuvieron 427 tuits. La gráfica de frecuencia nos muestra que las palabras más utilizadas son las relacionadas con proyectos que tienen una repetición de palabras de entre 16 y 28 veces, siendo *gente* y *tema* las que más se repiten.

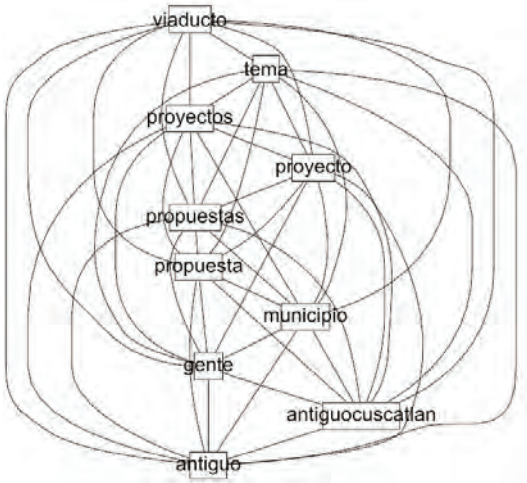
Figura 68. Palabras más frecuentes utilizadas por Luis Rodríguez



Fuente: creación propia a partir del modelo.

Mediante la representación de la relación entre las palabras más utilizadas, se confirma que de lo que más se habla es de los proyectos y propuestas para la gente de Antiguo Cuscatlán. El tema que sobre sale es "Viaducto frente a la basílica de Guadalupe", que es mencionado de diferentes formas en los textos de los tuits.

Figura 69. Relación entre las palabras más utilizadas por Luis Rodríguez



Fuente: creación propia a partir del modelo.

La nube de palabras muestra, además del tema "Viaducto frente a la basílica de Guadalupe", otros que incluyen el tráfico, *runners* y deportes, los habitantes y el diseño del palacio de las artes.

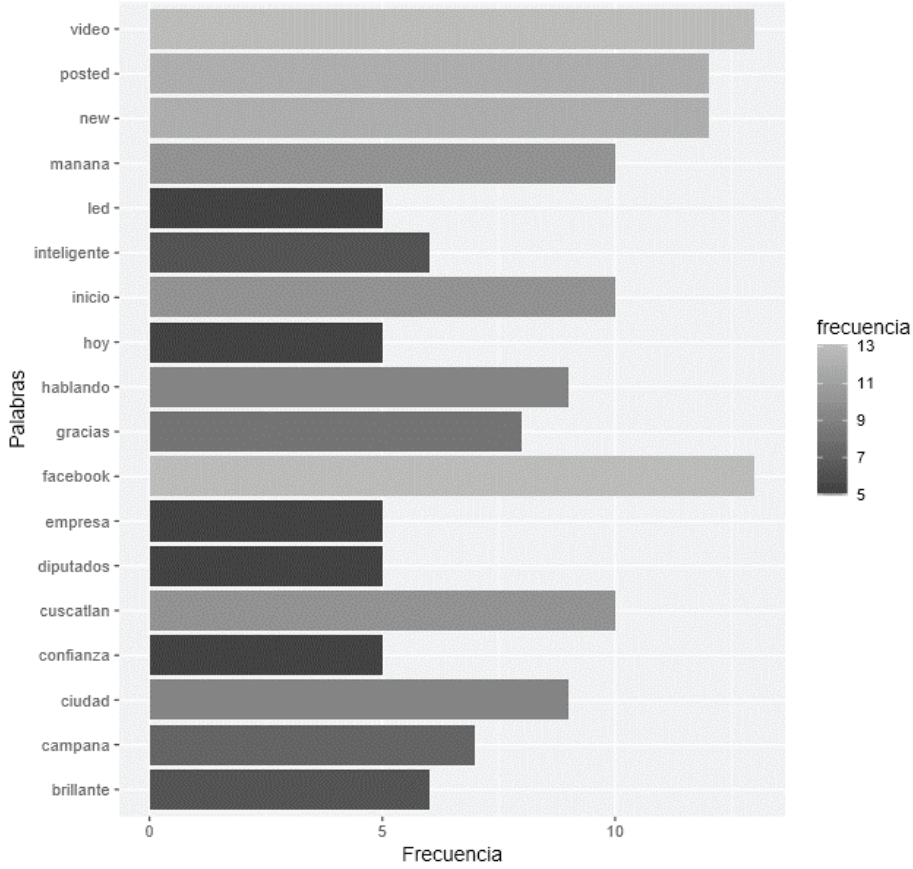
Figura 70. Nube con las palabras más utilizadas por Luis Rodríguez

Fuente: creación propia a partir del modelo.

@milagro__navas

De la cuenta de @milagro__navas, se obtuvieron 76 tuits. La gráfica de frecuencia nos muestra que las palabras más utilizadas son las relacionadas con videos, con una repetición de palabras de entre 5 y 13 veces, siendo entre estas *Facebook* y *video* las que más se repiten.

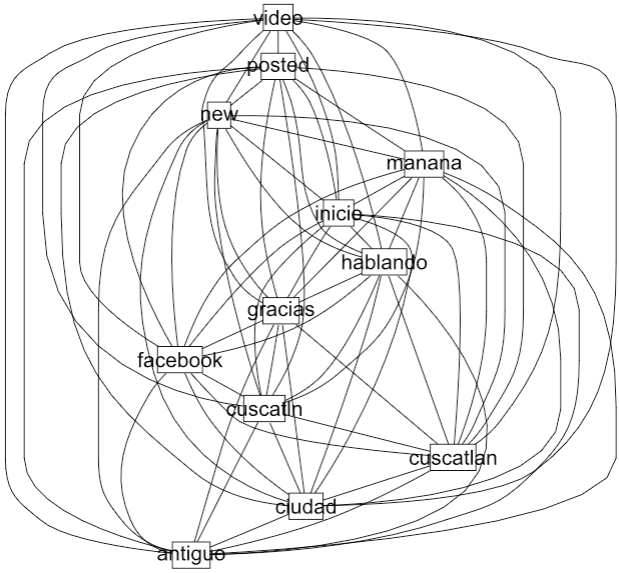
Figura 71. Palabras más frecuentes utilizadas por Milagro Navas



Fuente: creación propia a partir del modelo.

En la figura se constata que las palabras más utilizadas son *Facebook* y *video*. Mediante la representación de la relación entre las palabras más utilizadas, se confirma que la alcaldesa utiliza la red para compartir enlaces con Facebook. Además, se menciona el proyecto de convertir a la ciudad de Antiguo Cuscatlán en una ciudad inteligente.

Figura 72. Relación entre las palabras más utilizadas por Milagro Navas



Fuente: creación propia a partir del modelo.

La nube de palabras nos permite visualizar, mediante el tamaño de las palabras, aquellas que han sido utilizadas un mayor número de veces, apareciendo, adicional a lo antes mencionado, el tema de la colocación de luminarias LED.

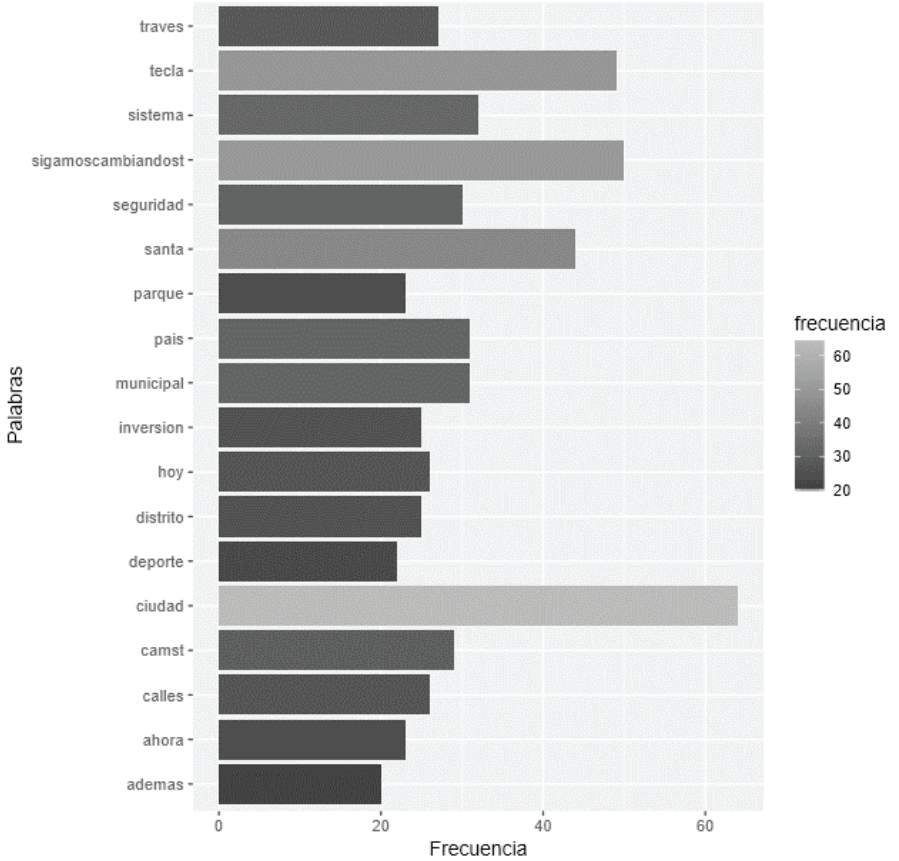
Figura 73. Nube con las palabras más utilizadas por Luis Rodríguez

Fuente: creación propia a partir del modelo.

@rdaubuisson

De la cuenta de @rdaubuisson, se obtuvieron 626 tuits. La gráfica de frecuencia nos muestra que las palabras más utilizadas son las relacionadas con la ciudad, con una repetición de palabras de entre 20 y 60 veces, siendo *tecla, sistema, seguridad* y el *hashtag sigamoscambiandost* las que más se repiten.

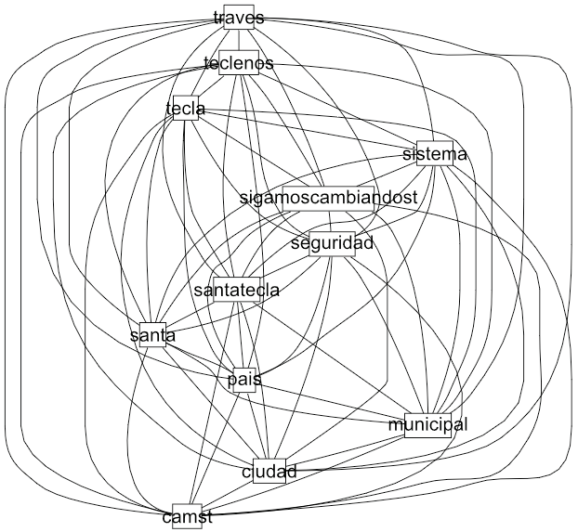
Figura 74. Palabras más frecuentes utilizadas por Roberto d'Aubuisson



Fuente: creación propia a partir del modelo.

Mediante la relación entre palabras, podemos observar que de lo que más se habla es sobre el sistema de seguridad y la apuesta por los deportes. Finalmente podemos mencionar que se está haciendo énfasis en los cambios llevados a cabo durante la gestión del alcalde.

Figura 75. Relación entre las palabras más utilizadas por Roberto d'Aubuisson



Fuente: creación propia a partir del modelo.

La nube de palaras nos permite visualizar otras palabras que han sido utilizadas en torno los temas de cambios, seguridad y deportes, agregando información para comprender de lo que se habla en los tuits.

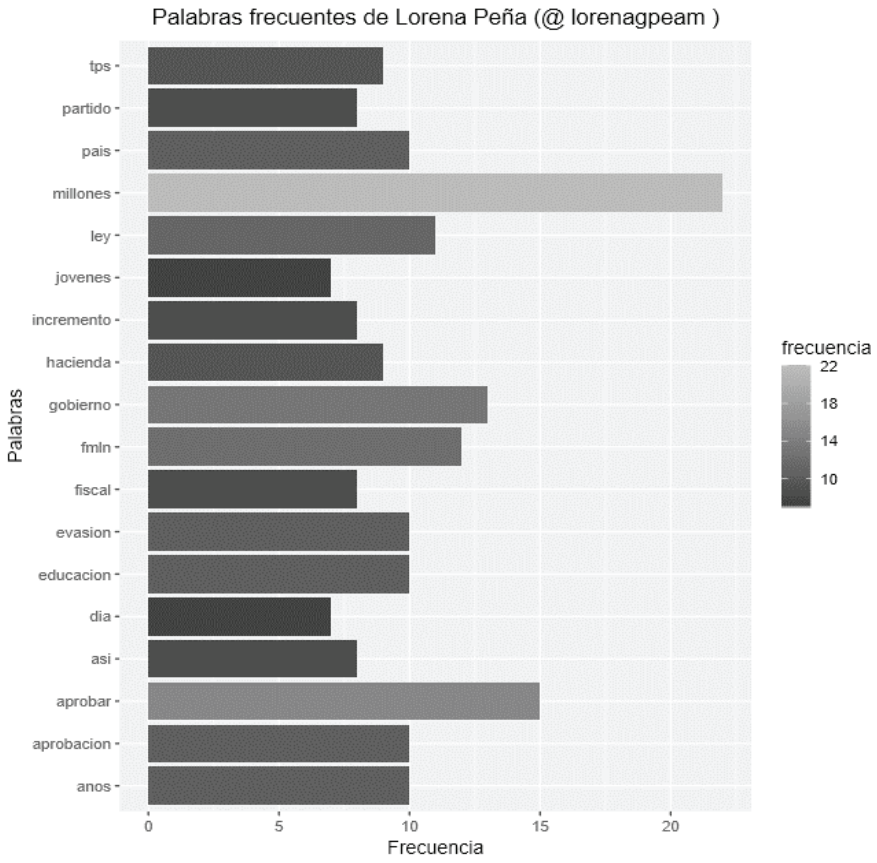
Figura 76. Nube con las palabras más utilizadas por Roberto d'Aubuisson

Fuente: creación propia a partir del modelo.

@lorenagpeam

De la cuenta de @lorenagpeam, se obtuvieron 292 tuits. La gráfica de frecuencia nos muestra que las palabras más utilizadas son las relacionadas con el gobierno de su partido, con una repetición de palabras de entre 10 y 22 veces, siendo *millones* y *aprobar* las que más se repiten. La palabra *millones* está relacionada con *montos* en diferentes formas, como impuestos, distribución de fondos y la aprobación del presupuesto general de la nación.

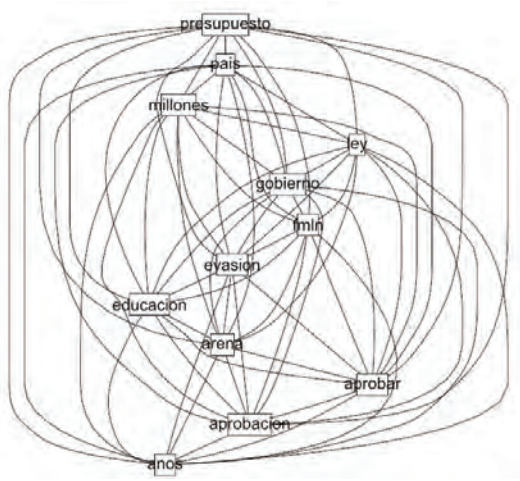
Figura 77. Palabras más frecuentes utilizadas por Lorena Peña



Fuente: creación propia a partir del modelo.

Mediante la relación de palabras, se puede observar que el tema central o más utilizado en los textos es la aprobación del presupuesto general de la nación, aparecen además menciones de su partido político, vía *hashtags* o en forma directa.

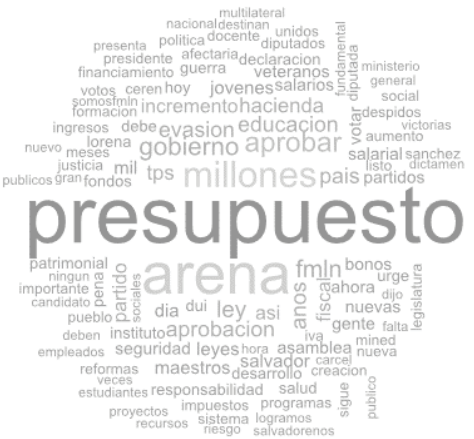
Figura 78. Relación entre las palabras más utilizadas por Lorena Peña



Fuente: creación propia a partir del modelo.

La nube de palabras muestra como palabra central y más utilizada *presupuesto*, y en torno a ella se encuentran otras que nos permiten tener una idea más completa de lo que se está mencionando, como, por ejemplo, partidos políticos, recursos, leyes y campaña electoral.

Figura 79. Nube con las palabras más utilizadas por Lorena Peña

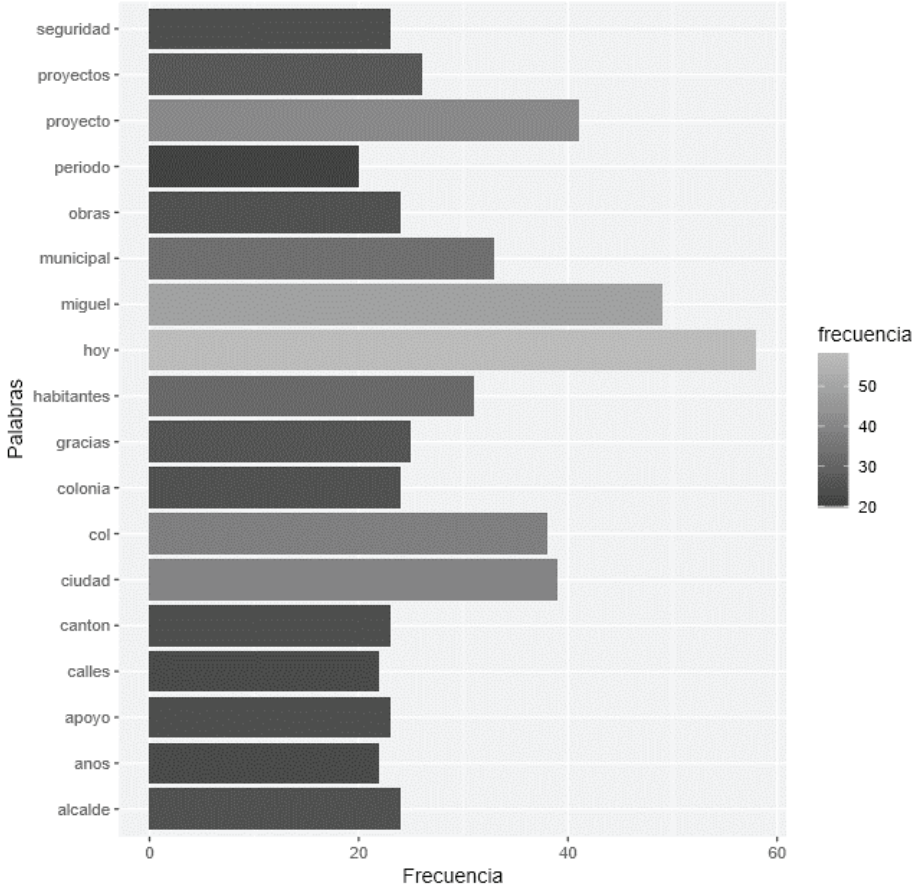


Fuente: creación propia a partir del modelo.

@Miguelpereirasv

De la cuenta de @Miguelpereirasv, se obtuvieron 454 tuits. La gráfica de frecuencia nos muestra que las palabras más utilizadas son las relacionadas *San Miguel* y *proyectos*, con una repetición de entre 20 y 50 veces. La palabra *hoy* aparece con una mayor repetición debido a que los mensajes o tuits hablan sobre actividades que se van realizando a diario, como por ejemplo visitas, reuniones, inauguraciones, entre otras.

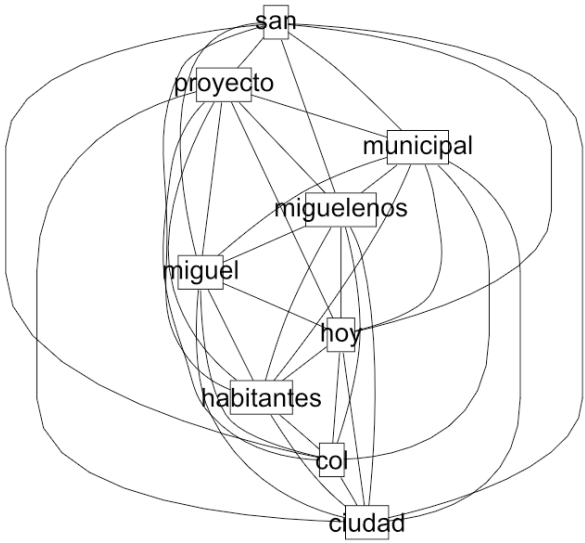
Figura 80. Palabras más frecuentes utilizadas por Miguel Pereira



Fuente: creación propia a partir del modelo.

Se observa, en la relación de palabras, que en general los textos hacen referencia a diferentes proyectos llevados a cabo en San Miguel y a la población de dicho departamento. La principal actividad o uso de la cuenta, ha sido la de informar sobre las actividades desarrolladas.

Figura 81. Relación entre las palabras más utilizadas por Miguel Pereira



Fuente: creación propia a partir del modelo.

La nube de palabras muestra como palabra central y más utilizada *migueleños*; y nos presenta una lista de palabras relacionadas con los proyectos, como por ejemplo, construcciones, inversiones, becas, vigilancia, seguridad y otros.

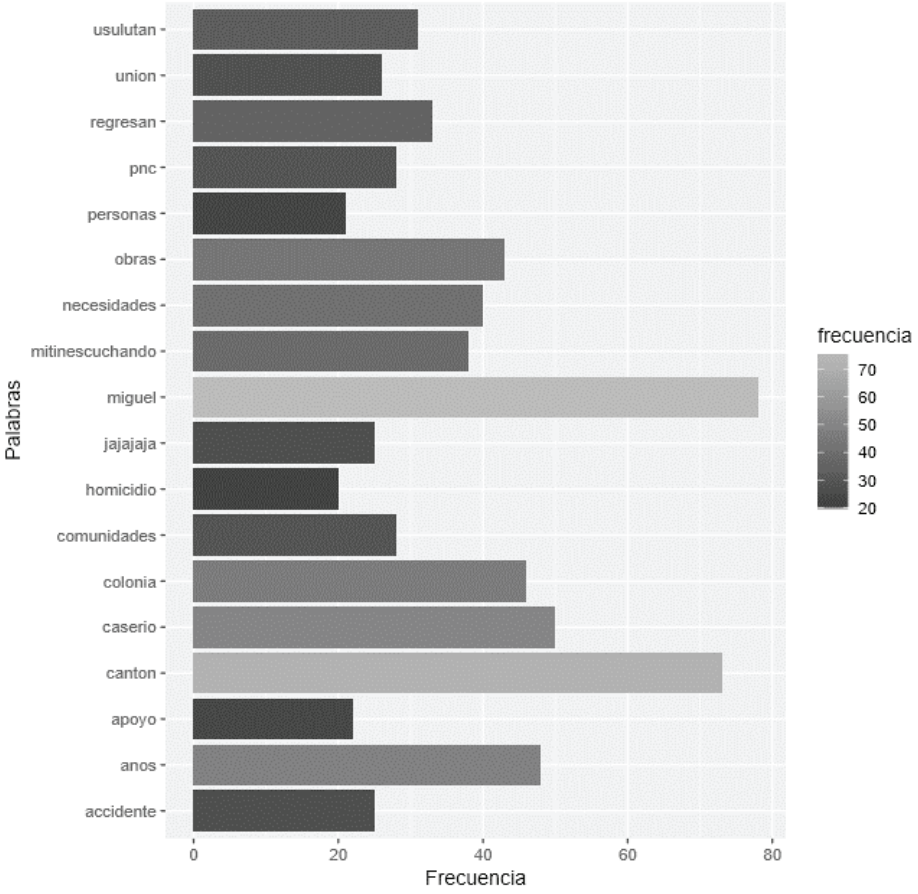
Figura 82. Nube con las palabras más utilizadas por Miguel Pereira

Fuente: creación propia a partir del modelo.

@WILLSALGADO

De la cuenta de @WILLSALGADO, se obtuvieron 965 tuits. La gráfica de frecuencia nos muestra que las palabras más utilizadas son las relacionadas *San Miguel, cantones, caseríos* y *colonias*, con una repetición de palabras de entre 20 y 70 veces. *Usulután* también es repetido, esto debido a que en los textos se notifica sobre diferentes acontecimientos ocurridos en dicho departamento.

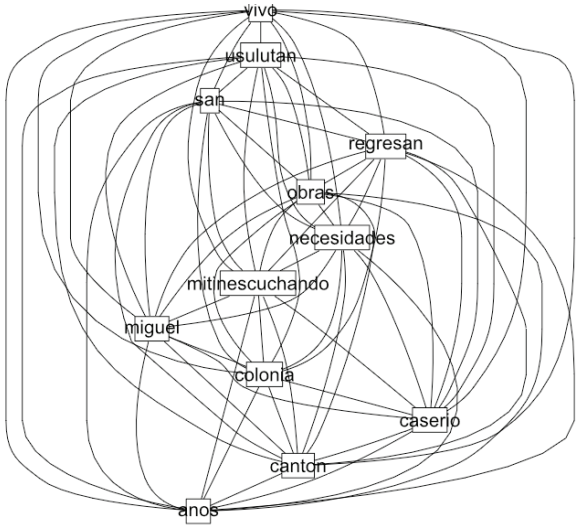
Figura 83. Palabras más frecuentes utilizadas por Will Salgado



Fuente: creación propia a partir del modelo.

Se observa, en la relación de palabras, que en general los textos hacen referencia a notificación de diferentes eventos ocurridos en Usulután y a mítines llevados a cabo en diferentes ciudades para escuchar las necesidades locales.

Figura 84. Relación entre las palabras más utilizadas por Will Salgado



Fuente: creación propia a partir del modelo.

La nube de palabras muestra *vivo* con una mayor repetición, ya que en los textos se hace referencia a videos presentados en vivo, además podemos observar que la principal actividad de esta cuenta ha sido informativa.

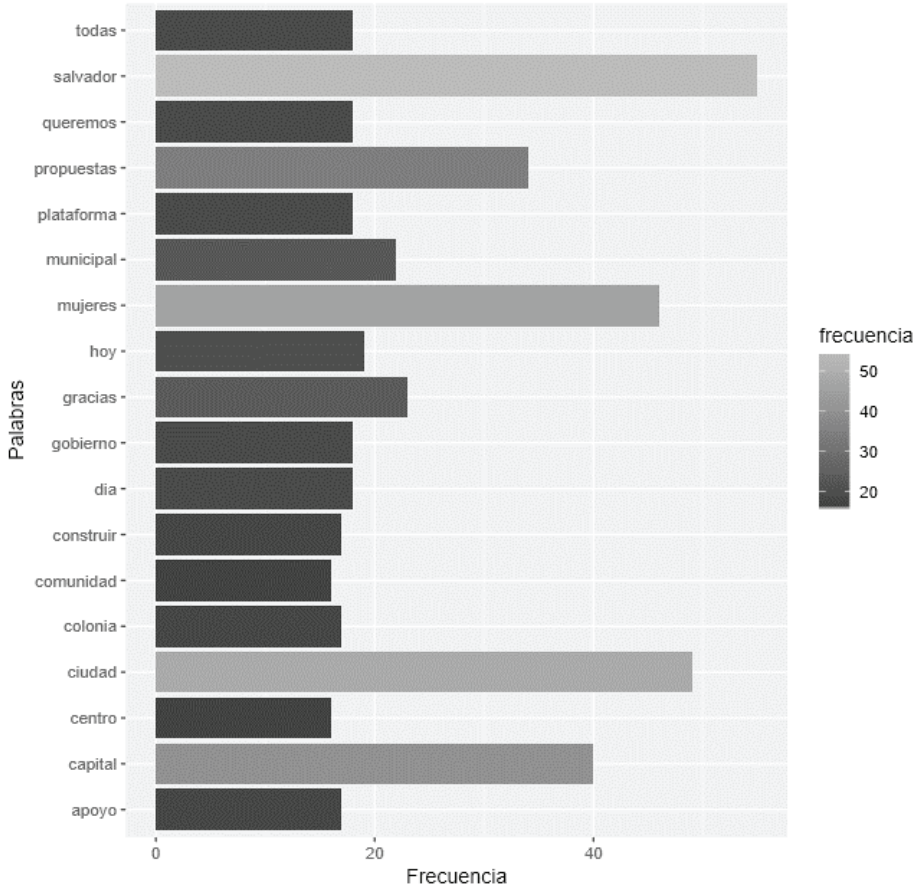
Figura 85. Nube con las palabras más utilizadas por Will Salgado

Fuente: creación propia a partir del modelo.

*@JackelineRA_*

De la cuenta de @JackelineRA_, se obtuvieron 432 tuits. La gráfica de frecuencia nos muestra que las palabras más utilizadas son las relacionadas *San Salvador, capital, propuestas* y *ciudad mujer*, con una repetición de palabras de entre 20 y 50 veces.

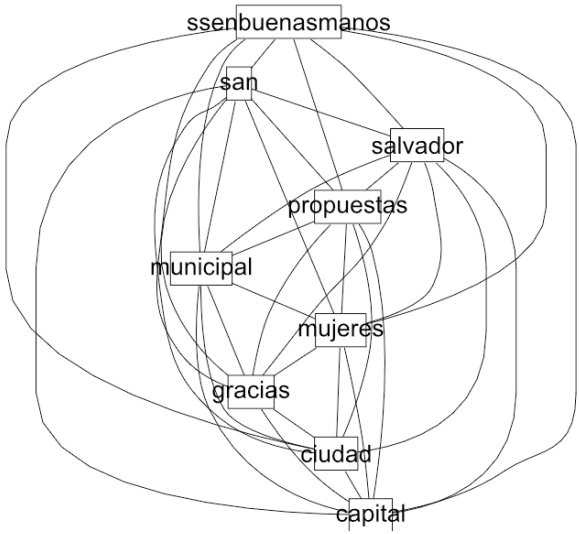
Figura 86. Palabras más frecuentes utilizadas por Jackeline Rivera



Fuente: creación propia a partir del modelo.

Se observa, en la relación de palabras, que posiblemente la cuenta está siendo utilizada para hablar de las propuestas de campaña, teniendo como eje central el tema de Ciudad Mujer en la capital y otros temas para la ciudadanía en general.

Figura 87. Relación entre las palabras más utilizadas por Jackeline Rivera



Fuente: creación propia a partir del modelo.

La nube de palabras confirma los temas de San Salvador y de Ciudad Mujer como centrales, mencionando también otras ideas orientadas al desarrollo y motivación de la juventud, a la comunidad, a las familias y a los vecinos.

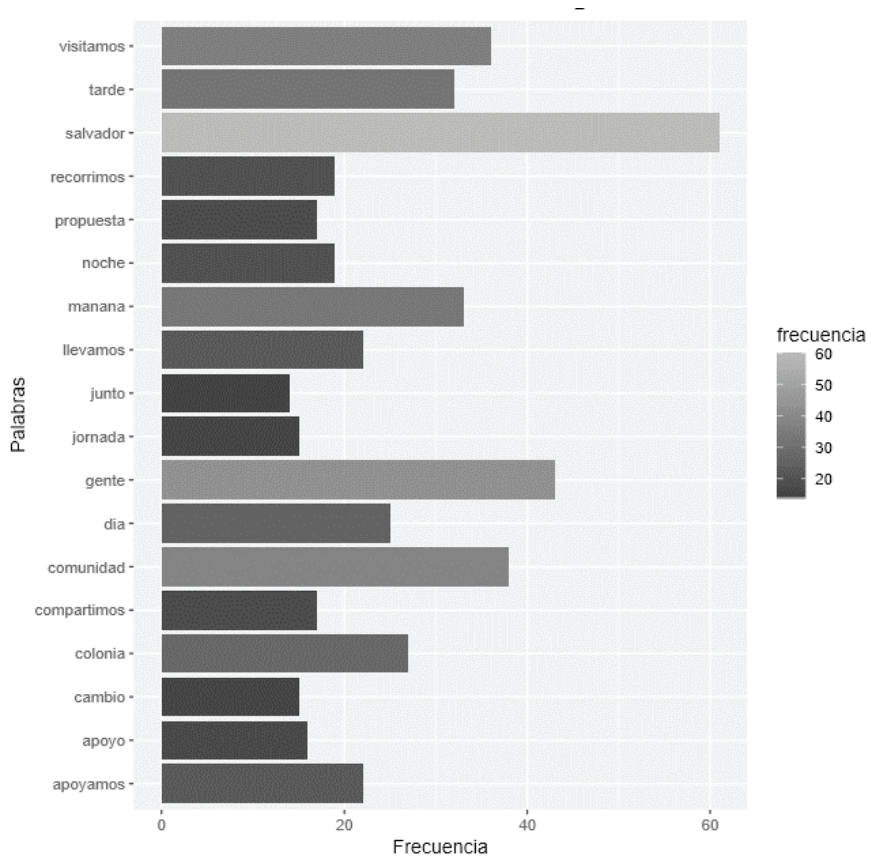
Figura 88. Nube con las palabras más utilizadas por Jackeline Rivera

Fuente: creación propia a partir del modelo.

@emuyshondt

De la cuenta de @emuyshondt, se obtuvieron 228 tuits. La gráfica de frecuencia nos muestra que las palabras más utilizadas son las relacionadas *San Salvador, visitas, comunidad* y *colonias*, con una repetición de palabras de entre 20 y 60 veces. Las palabras *visita* y *mañana* aparecen juntas en muchas ocasiones debido a que en los textos se habla sobre actividades desarrolladas a diario en diferentes localidades.

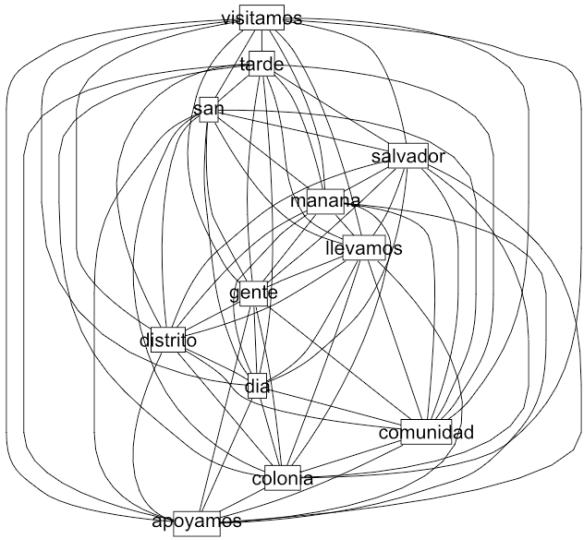
Figura 89. Palabras más frecuentes utilizadas por Ernesto Muyshondt



Fuente: creación propia a partir del modelo.

La conexión entre las palabras utilizadas se muestra en la gráfica de relación, donde se puede observar que se habla sobre diferentes actividades llevadas a cabo, como, por ejemplo, visitas y apoyos de diferentes tipos, tanto por la mañana como por la tarde, es decir, que la cuenta está siendo utilizada como medio informativo.

Figura 90. Relación entre las palabras más utilizadas por Ernesto Muyshondt



Fuente: creación propia a partir del modelo.

La nube de palabras muestra las diferentes actividades mencionadas en la cuenta, como por ejemplo la entrega de materiales, jornadas médicas, recorridos de las comunidades, canchas, entre otras. El tamaño de las letras nos permite observar que lo más repetido es *San Salvador*, *visitas*, *comunidad* y *gente*.

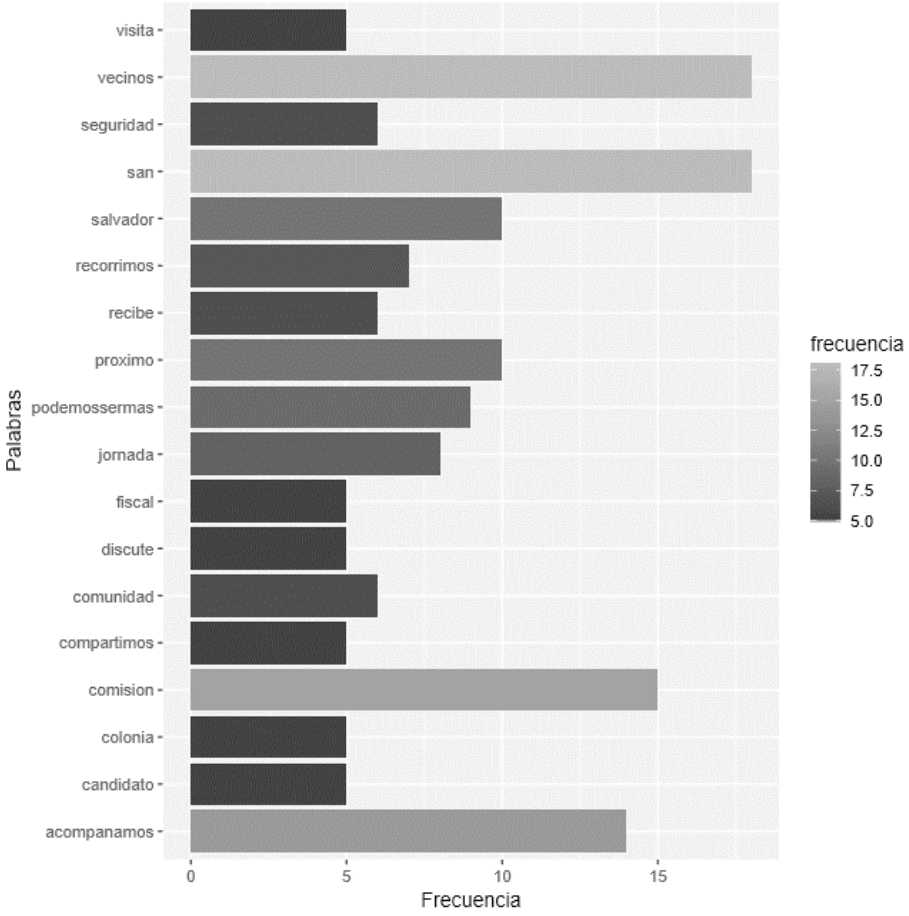
Figura 91. Nube con las palabras más utilizadas por Ernesto Muyshondt

Fuente: creación propia a partir del modelo.

@norman_quijano

De la cuenta de @norman_quijano, se obtuvieron 95 tuits y una repetición de palabras de entre 5 y 18 veces, una menor cantidad comparada con las otras cuentas. La gráfica de frecuencia nos muestra que las palabras más utilizadas son *San Salvador, vecinos* y *jornada*.

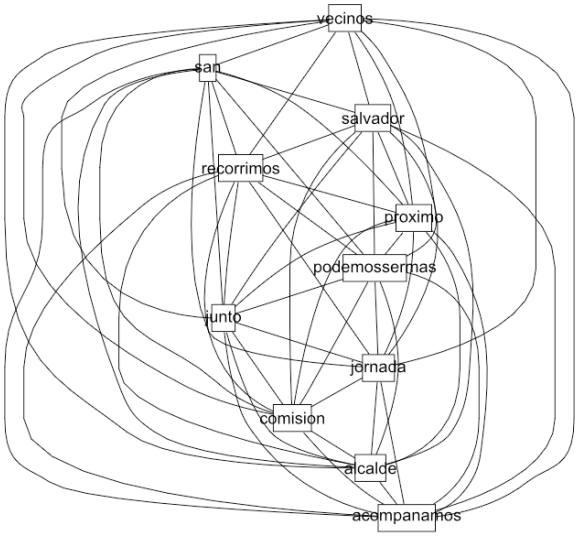
Figura 92. Palabras más frecuentes utilizadas por Norman Quijano



Fuente: creación propia a partir del modelo.

Las conexiones entre las palabras nos indican que aquellas más utilizadas aparecen juntas, en muchas ocasiones haciendo referencia a visitas y recorridos que se llevaron a cabo en diferentes lugares.

Figura 93. Relación entre las palabras más utilizadas por Norman Quijano



Fuente: creación propia a partir del modelo.

La nube de palabras muestra las palabras *alcalde* y *junto* con mayor frecuencia al compararla con las otras, esto es debido a que en los textos se hace referencia a actividades desarrolladas junto con otras personalidades del ámbito político, estas actividades incluyen visitas y recorridos en comunidades.

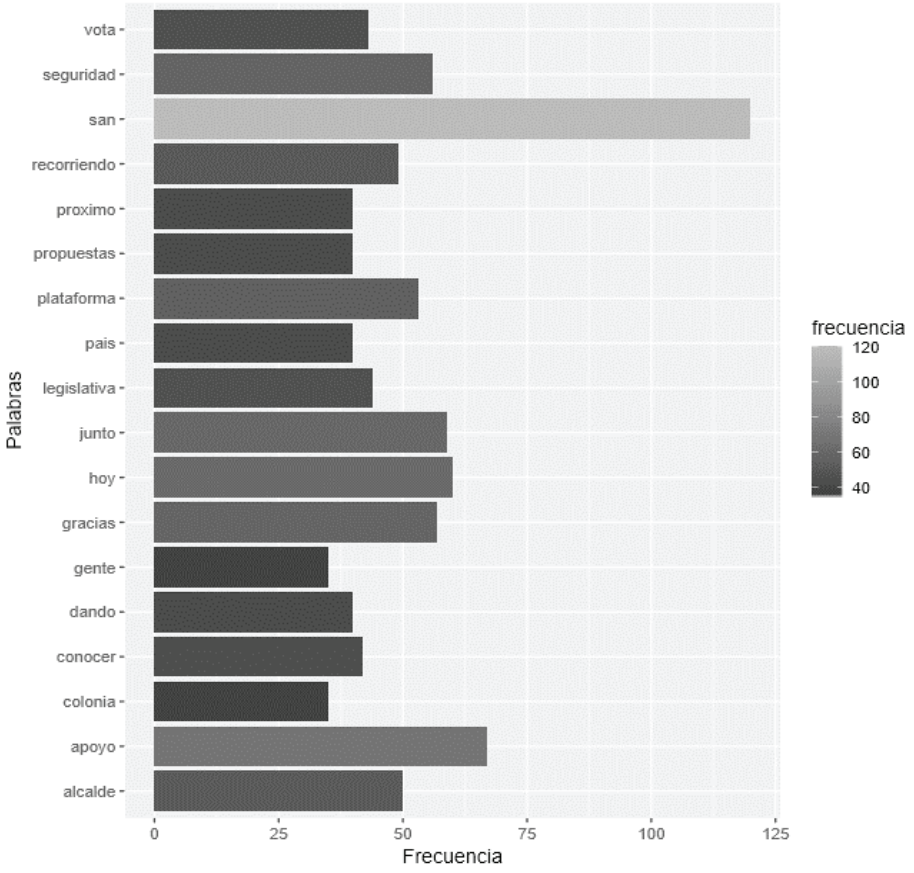
Figura 94. Nube con las palabras más utilizadas por Norman Quijano

Fuente: creación propia a partir del modelo.

@GGallegos24

De la cuenta de @GGallegos24, se obtuvieron 803 tuits. La gráfica de frecuencia nos muestra que las palabras más utilizadas son *apoyo* y *seguridad*, con una repetición de palabras entre 40 y 120 veces. El término *apoyo* es utilizado en los textos tanto en forma de ofrecimiento como de recibido y la palabra seguridad como propuesta del candidato.

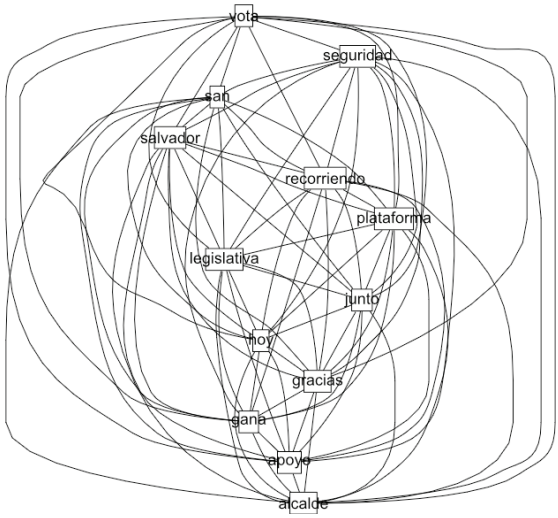
Figura 95. Palabras más frecuentes utilizadas por Guillermo Gallegos



Fuente: creación propia a partir del modelo.

La relación entre las palabras nos muestra que en los textos se habla sobre la petición de votos y la presentación de una propuesta legislativas enfocadas en la seguridad.

Figura 96. Relación entre las palabras más utilizadas por Guillermo Gallegos



Fuente: creación propia a partir del modelo.

Mediante la nube de palabras podemos observar que el tema de seguridad también está relacionado con grupos delictivos, y, como idea principal, además se muestran otras como *seguridad* y *educación*. Los textos recolectados para la cuenta muestran que esta ha sido utilizada como medio para dar a conocer las propuestas y para solicitar votos.

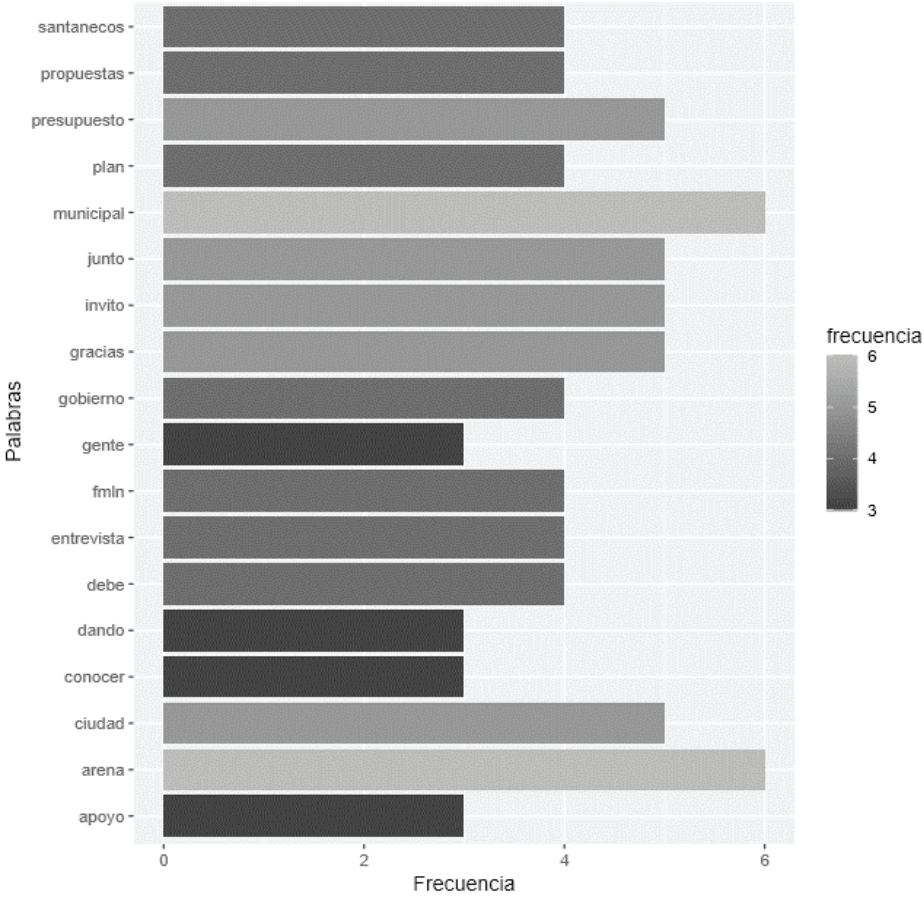
Figura 97. Nube con las palabras más utilizadas por Guillermo Gallegos

Fuente: creación propia a partir del modelo.

@Milena_Escalon

De la cuenta de @Milena_Escalon, se obtuvieron 62 tuits. La gráfica de frecuencia nos muestra que las palabras más utilizadas cuentan con una frecuencia similar, con una repetición de entre 3 y 6 veces. Teniendo en cuenta la frecuencia con que aparecen las palabras más utilizadas y la cantidad de tuits, podemos decir que la cuenta ha tenido menor uso al compararla con las otras.

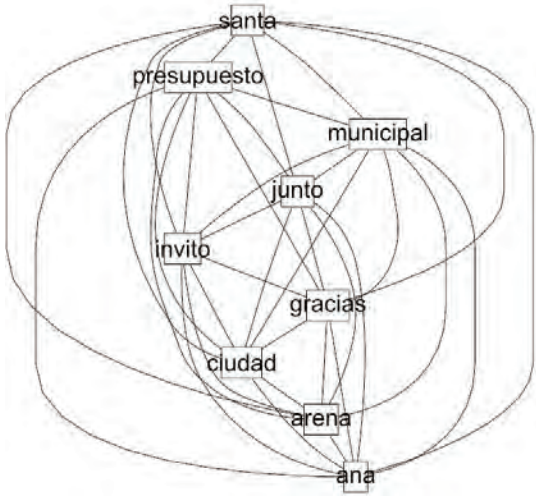
Figura 98. Palabras más frecuentes utilizadas por Milena de Escalón



Fuente: creación propia a partir del modelo.

La relación entre las palabras nos muestra que los temas centrales de los textos es la ciudad de Santa Ana y el presupuesto.

Figura 99. Relación entre las palabras más utilizadas por Milena de Escalón



Fuente: creación propia a partir del modelo.

Debido a que la cantidad de texto extraído de la cuenta es menor, la nube de palabras tiene menos elementos comparados a las otras cuentas, pero nos presenta la idea central enfocada en Santa Ana, dos partidos políticos y el presupuesto.

Figura 100. Nube con las palabras más utilizadas por Milena de Escalón

Fuente: creación propia a partir del modelo.

5. Conclusiones y trabajos futuros

5.1 *Conclusiones*

El análisis de texto es una tarea que requiere de diferentes pruebas para obtener buenos resultados y para extraer información útil, estos varían de acuerdo con el contexto, el idioma y la cantidad de información con la que se cuenta. De todas las etapas, la de mayor importancia es la de preprocesamiento; y es a la que más tiempo se le debe dedicar porque se debe depurar todo aquello que no sea de utilidad.

R es una herramienta poderosa que cuenta con una gran variedad de librerías para el análisis de texto desde diferentes fuentes y la visualización de resultados, esto último es muy importante, ya que sin representación gráfica el texto procesado, en algunas ocasiones, carece de sentido o relevancia, por lo que es altamente recomendable seleccionar adecuadamente los gráficos de acuerdo con las necesidades y con la idea que se quiera mostrar con la información, como se hizo para conocer la frecuencia de las palabras, relaciones, temas y nubes de palabras.

5.1.1 *Uso de* WORD2VEC

Con los resultados obtenidos, se ha identificado que *Word2Vec* es una poderosa herramienta con la cual se puede extraer mucha información a partir de textos donde se tiene en cuenta el contexto de cada una de las palabras, aun utilizando un modelo genérico a partir de una inmensa cantidad de textos que puede abarcar una variedad de términos, palabras y situaciones, como el caso de Google News, o creando nuestro propio modelo con información específica o textos orientados a una rama en particular.

Si bien Word2Vec nos permite hacer diferentes tipos de análisis sobre los textos y poder aplicar operaciones algebraicas para aprender ontologías, es necesario hacer uso de técnicas de visualización para analizar los resultados; y que puedan ser más comprensibles e identificar estructuras gramaticales fácilmente.

Algunas de las herramientas, como los *heatmap* y los dendrogramas, nos permitieron visualizar la relación entre las palabras, y a partir de los niveles de los clústeres formar estructuras o descubrir conceptos; otras, como es el caso de las nubes de palabras, nos permiten observar una mayor cantidad de palabrasagrupadas para tener una idea de lo que se habla en los textos y de la calidad de dichas relaciones.

Tabla 11
Resultados de las pruebas obtenidas basados en el número de nubes de palabras con información de interés

Sin *stemming*		
Sección	Modelo genérico	Modelo creado
Título	0.63	0.89
Abstract	0.74	0.79
Author keywords	0.79	0.79
Index keywords	0.68	0.89
Con *stemming*		
Título	0.47	0.58
Abstract	0.84	0.89
Author keywords	0.68	0.63
Index keywords	0.74	0.63

La base de textos extraída de Scopus nos ha proporcionado diferentes resultados, así como el hecho de aplicar *stemming* en la etapa de preprocesamiento. Se puede notar que para nuestro contexto los mejores resultados se han obtenidos a partir del modelo que se creó con la base de datos Scopus; y que en el caso del *abstract* (0.84 con el modelo genérico y 0.89 con el modelo creado), los resultados mejoran, ocurriendo lo opuesto con el *title* y el *index keyword*, esto puede ocurrir debido a que en el *abstract* se tiene una mayor cantidad de información, palabras en diferentes tiempos y mayor flexibilidad que en los otros campos.

En los resultados no solamente basta el número de nubes de palabras donde existe una cantidad de texto relacionado, sino que se debe evaluar su calidad, como es el caso de los textos proporcionados por la Unidad de Datos de *El Diario de Hoy*, donde las nubes fueron seleccionadas con base en su importancia periodística, es decir, aquellas que dieran información de utilidad para conocer y comprender los temas de las noticias; como consecuencia, los mejores resultados obtenidos fueron con los titulares de las noticias, ya que permitieron extraer ideas concretas.

El cuerpo de las noticias también fue de utilidad para formar ideas y conceptos utilizando otros tipos de representación visual, como es el caso de los clústeres y las gráficas en dos dimensiones. Debido a la cantidad de texto que compone esta parte, nos permite generar relaciones de diferentes niveles, que podemos ir seleccionando para ampliar el alcance o definición de los conceptos.

5.1.2 *Análisis de tuits*

El análisis de texto de tuits se vuelve aún más complicado que el texto proveniente de otras fuentes con una estructura más rígida o que pasan por un filtro antes de ser publicados, debido a la libertad con que las personas los escriben, como, por ejemplo, el uso de abreviaturas, escritura incorrecta, uso de términos propios de cada país y la inclusión de caracteres especiales como por ejemplo símbolos, URL, entidades de *retweet*, entre otros.

Algunas técnicas de procesamiento, como el *stemming*, puede generar pérdida de información, por lo que, en caso de que se trabaje con temas específicos donde tengamos una idea previa de los términos más utilizados, es posible crear un diccionario compuesto por aquellas palabras que más se repiten y sus variaciones, como por ejemplo los *hashtags*.

Se puede ver, mediante el análisis de las cuentas seleccionadas, que Twitter puede ser utilizado de diferentes maneras, por ejemplo, como medio informativo, para presentación de propuestas, campañas políticas o para el seguimiento de proyectos. También se pudo observar que algunas tenían un mayor movimiento que otras, tanto en número de tuits como en cantidad de palabras utilizadas, indicando así su presencia en los medios digitales y selección de estas plataformas como complemento a los medios tradicionales (Televisión, radio, periódicos, boletines y otros).

5.2 *Trabajo futuro*

El trabajo futuro puede ser dividido en dos partes, teniendo en cuenta su importancia y el tiempo que requieren para su implementación, siendo aquellas a corto plazo las más importantes y que requieren modificaciones menores sobre el trabajo desarrollado; y aquellas a largo plazo las que pueden dar origen a nuevos proyectos o que requieren mayor tiempo para poder ser desarrolladas.

5.2.1 *A corto plazo*

Las tareas a corto plazo que pueden mejorar los resultados obtenidos son las siguientes:

- Generar una base de datos de mayor tamaño: se debe tener en cuenta que, mientras mayor sea la cantidad de texto que componen las bases de datos, mejor será el entrenamiento de los modelos generados, por lo que es recomendable utilizar una mayor cantidad de artículos para evaluar nuevamente los resultados obtenidos.
- Preprocesamiento de texto: se pueden desarrollar pruebas modificando el preprocesamiento y refinando las expresiones regulares para eliminar ciertos caracteres y mantener aquellos que pueden ser de utilidad, para que las palabras utilizadas tengan mayor peso. Ocurrió que en algunos pasos las palabras sufrían modificaciones durante esta etapa debido a que estaban unidas por símbolos, como es el caso del guion medio "-" o la comilla simple "'". En el caso de los tuits, se puede hacer un diccionario con palabras más utilizadas por las cuentas o de los temas que se analizan para poder centrarse en los otros términos y extraer ideas más concretas.
- Modificar los parámetros de *Word2Vec*: los parámetros seleccionados pueden generar impacto en los resultados, por lo que se pueden modificar como es el caso del número de vectores, la ventana y los ejemplos utilizados.
- Ngramas: a partir de los textos descargados, se pueden generar n-gramas y hacer pruebas con diferentes valores, por ejemplo,

entre 2 y 3 gramas, ya que algunas palabras por separado pueden aportar cierta información o tener diferente significado cuando aparecen juntas, como es el caso de *big data, cloud computing, San Miguel, San Salvador* y nombres de personas, entre otros, que juntos pueden dar origen a nuevos conceptos.

5.2.2 *A largo plazo*

El objetivo del proyecto era obtener una ontología que describa los conceptos de los que se habla en los textos de la base seleccionada, sin embargo, *Word2Vec* es una herramienta que puede ser utilizada para otras alternativas; y lo realizado hasta el momento puede dar origen a nuevas aplicaciones, como las siguientes:

- Extracción de *keywords:* mediante las nubes de palabras o de los clústeres se pueden extraer las palabras clave de los textos, esto se logra analizando los documentos en forma independiente y analizando las nubes resultantes.
- Resúmenes: con la información representada en forma gráfica, se puede generar una mejor idea de lo que se está hablando en los textos, por ejemplo, se descargan las publicaciones de un año o un mes u otra unidad de tiempo y se analizan los resultados. De este modo se puede saber cuáles son conceptos de los que se está hablando y hacer un resumen de la información.
- Englobar y extraer temáticas: similar a los resúmenes, se puede dar origen a temáticas con base en los resultados obtenidos.
- Taxonomías: con información de diferentes bases, se podría generar una taxonomía o clasificación de los textos en diferentes tipos, como, por ejemplo, novelas, ensayos, anuncios, libros, periódicos y artículos especializados.

En el caso del análisis de tuits, se pueden llevar a cabo las siguientes actividades:

- Análisis de sentimientos.
- Posicionamiento de marcas.
- Seguimiento de campañas políticas.
- Aceptación de productos por la población.
- Comparativa entre marcas.
- Recomendaciones.
- Análisis de tendencias.
- Análisis de influencias.

6. Referencias

Arano (2005). "Los tesauros y las ontologías en la Biblioteconomía y la Documentación". Recuperado 23 de agosto de 2017 de http://www.upf.edu/hipertextnet/numero-3/tesauros.html

Barazza, L. (2017, febrero 18). *"How does Word2Vec's Skip-Gram work?"* Recuperado 28 de agosto de 2017 de https://becominghuman.ai/how-does-word2vecs-skip-gram-work-f92e0525def4

Bassi, Alejandro. (2001, Marzo 15). "Lematización basada en análisis no supervisado de corpus". Recuperado 23 de agosto de 2017 de https://users.dcc.uchile.cl/~abassi/ecos/lema.html

Botta-Ferret, E., & Cabrera Gato, J.E. (2007, agosto 6). "Minería de textos: una herramienta útil para mejorar la gestión del bibliotecario en el entorno digital". Recuperado 22 de agosto de 2017 de http://bvs.sld.cu/revistas/aci/vol16_4_07/aci051007.html

Bussieck, J. (2017, agosto 22). *"Demystifying Word2Vec"*. Recuperado 26 de agosto de 2017 de http://www.deeplearningweekly.com/blog/demystifying-word2vec

Colyer, A. (2016, abril 21). *"The amazing power of word vectors"*. Recuperado 29 de agosto de 2017 de https://blog.acolyer.org/2016/04/21/the-amazing-power-of-word-vectors/

Contreras Barrera, M. (2014). *Minería de texto: una visión actual* (Vol. 17). Distrito Federal, México: Universidad Nacional Autónoma de México. Recuperado de http://www.redalyc.org/pdf/285/28540279005.pdf

Feldman, R. (1998). *"Text mining at the term level"*. *Principles of Data Mining and Knowledge Discovery*, *In: Żytkow J.M., Quafafou M. (eds) Lecture Notes in Computer Science* (vol 1510).

García, C.; Cabanilles, J.P., & Ramírez, B. (2017, febrero 24). *"What is the difference between stemming and lemmatization?"* Recuperado 21 de agosto de 2017 de https://blog.bitext.com/what-is-the-difference-between-stemming-and-lemmatization/

Grela, L.; Sauri, E., & Sellés, A. (2004, junio 24). "Ontologías en documentación". Recuperado 23 de agosto de http://personales.upv.es/ccarrasc/doc/2001-2002/ontologias/INICIO.htm

Gruber, T. (2009). *"Ontology (Computer Science) - definition in Encyclopedia of Database Systems"*. Recuperado 23 de agosto de 2017 de http://tomgruber.org/writing/ontology-definition-2007.htm

Halvey, M., & Keane, M.T. (2017, mayo 12). *"An Assessment of Tag Presentation Techniques"*. Recuperado 29 de agosto de 2017 de http://www2007.org/htmlposters/poster988/

IBM Knowledge Center (s. f.). Recuperado 20 de agosto de 2017 de https://www.ibm.com/support/knowledgecenter/es/SS3RA7_16.0.0/com.ibm.spss.ta.help/textmining/shared_entities/tm_intro_tm_defined.htm

Jurafsky, D., & H. Martin, J. (2017, agosto 28). *"Speech and Language Processing"*. Recuperado 7 de septiembre de 2017 de https://web.stanford.edu/~jurafsky/slp3/

Kannan, S., & Gurusamy, V. (2014, octubre). *"Preprocessing Techniques for Text Mining"*. Recuperado 21 de agosto de 2017 de https://www.researchgate.net/publication/273127322_Preprocessing_Techniques_for_Text_Mining

Lehmann, C. (2017, septiembre 28). *"Lexicography"*. Recuperado 21 de agosto de 2017 de http://www.christianlehmann.eu/ling/ling_meth/ling_description/lexicography/index.html?http://www.christianlehmann.eu/ling/ling_meth/ling_description/lexicography/lemmatization.html

Leskovec, J.; Rajaraman, A., & Ullman, J. (2014). *"Mining of Massive Datasets"*. Recuperado de http://www.mmds.org/

Maaten, L. van der (2017, diciembre 15). "t-SNE". Recuperado 30 de agosto de 2017 de https://lvdmaaten.github.io/tsne/

McCormick, C. (2016, abril 19). *"Word2Vec Tutorial - The Skip-Gram Model"*. Recuperado 28 de agosto de 2017 de http://mccormickml.com/2016/04/19/word2vec-tutorial-the-skip-gram-model/

Meyer, D. (2016, julio 31). *"How exactly does Word2Vec work?"* Recuperado 25 de agosto de 2017 de http://www.1-4-5.net/~dmm/ml/how_does_word2vec_work.pdf

Mikolov, T.; Chen, K.; Corrado, G., & Dean, J. (2013). *"Efficient Estimation of Word Representations in Vector Space"*. *arXiv:1301.3781 [cs]*. Recuperado de http://arxiv.org/abs/1301.3781

Minnaar, A. (2015, abril 12). *"Word2Vec tutorial Part 1: The Skip-Gram Model"*. Recuperado 26 de agosto de 2017 de http://mccormic-

kml.com/assets/word2vec/Alex_Minnaar_Word2Vec_Tutorial_Part_I_The_Skip-Gram_Model.pdf

Moujahid, A. (2014, julio 25). *"An Introduction to Text Mining using Twitter Streaming API and Python"* // Adil Moujahid // Data Analytics and more. Recuperado 16 de febrero de 2018 de http://adilmoujahid.com/posts/2014/07/twitter-analytics/

Murphy, J., & Roser, M. (2018). "Internet". Recuperado 16 de febrero de 2018 de https://ourworldindata.org/internet

Pérez Hernández, M.C. (2002). "Explotación de los córpora textuales informatizados para la creación de bases de datos terminológicas basadas en el conocimiento". *Estudios de Lingüística del Español (ELiEs)*, *18 (2002)*. Recuperado de http://elies.rediris.es/elies18/

Raulji, J., & Saini, J. (2016, septiembre). Stop-Word Removal Algorithm and its Implementation for Sanskrit Language. International Journal of Computer Applications, 150(2):15-17. Recuperado de http://www.ijcaonline.org/archives/volume150/number2/raulji-2016-ijca-911462.pdf

Rong, X. (2014). *"Word2Vec Parameter Learning Explained". arXiv:1411.2738 [cs]*. Recuperado de http://arxiv.org/abs/1411.2738

Srivastava, N., & Yao, J. (2014, febrero 4). Learning distributed word representations. Recuperado 21 de marzo de 2018 de http://www.cs.toronto.edu/~yaojian/csc321/assignment1.html

TextMiner (2014, julio 18). *"Dive Into NLTK, Part IV: Stemming and Lemmatization – Text Mining Online"*. Recuperado 17 de enero de 2018 de http://textminingonline.com/dive-into-nltk-part-iv-stemming-and-lemmatization

Vicente Villardón, J.L. (2007). *Introducción al análisis de clúster*. Departamento de Estadística, Universidad de Salamanca. 22p. Recuperado a partir de http://benjamindespensa.tripod.com/spss/AC.pdf

Zhang, D.; Xu, H.; Su, Z.; & Xu, Y. (2015). *"Chinese comments sentiment classification based on Word2Vec and SVM"*. *Expert Systems with Applications*, *42*(4), 1857-1863.

7. Enlaces con artículos y tutoriales consultados durante el aprendizaje

https://www.coursera.org/learn/text-mining-analytics

https://www.upf.edu/hipertextnet/numero-3/tesauros.html#2

https://www.youtube.com/watch?v=ERibwqs9p38

https://www.youtube.com/watch?v=thLzt3D-A10

https://www.youtube.com/watch?v=TsEGsdVJjuA

https://www.youtube.com/watch?v=BD8wPsr_DAI

https://www.youtube.com/watch?v=9x9JHJapHxA

http://bookworm.benschmidt.org/posts/2015-10-25-Word-Embeddings.html

https://medium.com/@mukulmalik/word2vec-part-1-fe2ec6514d70

https://rlbarter.github.io/superheat/basic-usage.html

https://www.coursera.org/learn/ml-clustering-and-retrieval/lecture/yyegc/distance-metrics-cosine-similarity

https://stackoverflow.com/questions/21474388/colorize-clusters-in-dendogram-with-ggplot2

http://yamano357.hatenadiary.com/entry/2015/11/04/000332

https://github.com/bmschmidt/wordVectors/blob/master/R/word2vec.R

http://developers-club.com/posts/258983/

https://rpubs.com/lmullen/nlp-chapter

https://ufal.mff.cuni.cz/mlnlpr13

https://www.kaggle.com/c/word2vec-nlp-tutorial/details/part-1-for-beginners-bag-of-words

https://rlbarter.github.io/superheat-examples/Word2Vec/

http://stackoverflow.com/questions/27324292/convert-word2vec-bin-file-to-text

http://googletrends.github.io/data/

https://github.com/rlbarter/superheat-examples

https://conda.io/docs/test-drive.html

https://radimrehurek.com/gensim/install.html

https://www.kaggle.com/c/word2vec-nlp-tutorial

http://mccormickml.com/2016/04/27/word2vec-resources/

https://plot.ly/ggplot2/ggdendro-dendrograms/

https://www.datasciencecentral.com/profiles/blogs/find-out-what-celebrities-tweet-about-the-most-1

http://www.sthda.com/english/wiki/text-mining-and-word-cloud-fundamentals-in-r-5-simple-steps-you-should-know

https://sites.google.com/site/miningtwitter/questions/talking-about/wordclouds/comparison-cloud

https://stackoverflow.com/questions/15224913/r-add-title-to-wordcloud-graphics-png

https://stackoverflow.com/questions/33855851/unable-to-rename-column-in-r

http://www.sthda.com/english/wiki/ggsave-save-a-ggplot-r-software-and-data-visualization

https://www.stat.berkeley.edu/~paciorek/computingTips/Saving_graphics_as_pdf_file.html

http://www.cookbook-r.com/Graphs/Output_to_a_file/

https://moderndata.plot.ly/create-colorful-graphs-in-r-with-rcolorbrewer-and-plotly/

Universidad Tecnológica
de El Salvador

Compilación de investigaciones de tecnología 2017
Aulas conectadas: sistema IoT para el registro de asistentes

Investigadores:
Omar Otoniel Flores Cortez
Verónica Idalia Rosa Urrutia

Esta investigación fue subvencionada por la Universidad Tecnológica de El Salvador. Las solicitudes de información, separatas y otros documentos relativos a este estudio pueden hacerse a la siguiente dirección postal: Universidad Tecnológica de El Salvador, edificio *Dr. José Adolfo Araujo Romagoza*, Vicerrectoría de Investigación y Proyección Social, Dirección de Investigaciones, calle Arce y 19.ª avenida Sur, 1045, o a omar.flores@utec.edu.sv.

San Salvador, 2018

Tabla de ilustraciones

Resumen

"El internet de las cosas (IoT) es una palabra pegadiza para decir dispositivos electrónicos embebidos conectados al internet, tales dispositivos están empotrados dentro de todo tipo de objetos de uso diario, permitiendo el fácil control o monitoreo de estado de estos objetos a través del internet gracias al dispositivo electrónico dentro de ellos" (Tollervey, 2017). En otras palabras, el IoT es un área de aplicación que combina electrónica, telecomunicaciones e informática, y que se ocupa de dotar a "cosas" de inteligencia automatizada para la conexión de estas a la red de internet, para que puedan enviar o recibir información por sí mismas; sistema que suple la necesidad de que la información y las funcionalidades ofrecidas por la multitud de objetos inteligentes, sensores y actuadores, que se prevé estén embebidos en el entorno en un futuro no muy lejano, sea accesible de manera sencilla e integral. Dentro del IoT, se encuentra un área de aplicación denominada *edificios inteligentes* (*smart city*) cuyo fin es el aplicar las técnicas IoT a entornos inmóticos (oficinas, comercios, escuelas, etc.) y domóticas (casas, apartamentos, residencias, etc.). Lo anterior implica que los sistemas electrónicos embebidos están en casi todo objeto que nos rodean, lo que implica que estamos envueltos por dispositivos computacionales que son únicos e identificables en el internet.

El presente es un informe sobre la investigación aplicada "Aulas conectadas: aplicación del IoT para el registro de asistentes", desarrollada durante el año 2017 dentro del plan de la Vicerrectoría de Investigación y Proyección Social de la Universidad Tecnológica de El Salvador (Utec). La investigación propone un nuevo conocimiento científico dentro de la aplicación de tecnologías y técnicas del IoT en la solución de una situación problemática específica: el registro automatizado de los asistentes a un evento, recinto, aula, a una conferencia, etc., cuyo objetivo principal fue el diseño, desarrollo y la validación de una plataforma o sistema IoT de bajo costo, para el registro automatizado vía internet de datos de asistentes a un recinto, basados en la información recogida por sensores electrónicos inalámbricos dispuestos en la entrada del lugar, y que sea una herramienta de apoyo a las labores de registro de datos de

usuarios y fortalecimiento de la seguridad de las personas que ingresan a la institución.

El sistema, fruto de esta investigación aplicada, fue diseñado en los siguientes bloques funcionales: 1. *hardware* electrónico, 2. *firmware* plataforma IoT y 3. *front-end* del usuario. En cada uno de estos se procuró el uso de una herramienta tecnológica de última generación y bajo coste. Para el primer bloque se utilizó el microcontrolador ESP8266, plataforma NodeMCU y el lector RFID; en el segundo se programó bajo los servicios de Google Api Script; y para el tercero, hojas de cálculo de Google Drive. Como resultado final, se obtuvo un sistema completo que permite que cada asistente, al acercar su tarjeta o carné al lector del *hardware* electrónico, pueda ser registrado al ingresar al aula, automáticamente esta información es enviada al internet y registrada en una base de datos para su visualización por el personal respectivo.

Los conocimientos científicos y las técnicas de IoT que aporta esta investigación son de gran ayuda en el planteamiento, diseño y la implementación de sistemas telemáticos de bajo coste, que permiten el monitoreo y control vía internet. Además, sus aplicaciones van mucho más allá de la realizada en este informe y son un fundamento muy importante en el desarrollo de ambientes inteligentes.

1. Introducción

1.1 *Problema investigado*

Actualmente el uso de sistemas electrónicos automatizados aplicados en la solución de tareas cotidianas o repetitivas es una necesidad; optimizar procesos y recursos económicos y humanos es prioritario para toda institución moderna. Específicamente, aquellas que ofrecen servicios a grupos de usuarios de forma grupal, tales como escuelas universidades, salas de eventos, o hasta instituciones que poseen una planta de recurso humano grande, se encuentran ante una tarea casi inevitable: el registro y monitoreo de las personas que asisten a la institución, evento, clase, concierto, congreso, conferencias, etc.

La situación descrita se vuelve critica en ambientes académicos, escuelas o universidades, lugares donde el registro de la asistencia de los estudiantes es necesario, por efectos de monitoreo, control y avaluación, más aún cuando el campus es extenso, se tienen muchas aulas y diferentes horarios de clases.

Al realizar un recorrido de campo por diferentes universidades del entorno nacional, y hacer una consulta sobre la metodología de registro de asistencia, encontramos que estas tareas de control de asistencias son actualmente realizadas haciendo un conteo visual y "a mano", además, utilizando personal académico para la realización de recorrido a pie por cada una de las aulas del campus.

En el caso de este estudio específico, que es de la Utec, uno de los procesos de índole administrativo que más demandan tiempo y esfuerzo por parte del cuerpo académico de cada escuela y decanato es el conteo de los estudiantes asistentes a las sesiones de clases en los diferentes horarios. Este es un proceso que se está desarrollando de forma manual en su recolección y en ocasiones hasta en su procesamiento y notificación.

Este proceso manual, actualmente se desarrolla en las siguientes dos etapas:

1. Una persona realiza una visita personal, aula por aula, a cada una de las clases en un bloque de horarios pertenecientes a una escuela o facultad, esto implica que esta persona debe hacer su recolección,

en ocasiones, en todos los edificios del campus; se presenta al aula y procede a contar a los asistentes sentados dentro, o en ocasiones le pregunta directamente al docente: "¿Cuántos le han llegado?".

2. La persona, al terminar de recolectar el número de asistentes de cada clase de ese bloque horario, procede a notificar personalmente a un administrador, para que este acceda a un computador donde ingresa manualmente cada número de asistentes por clase a un sistema informático o base de datos que puede ser consultado posteriormente por los directores y decano respectivos.

Algunas características detectadas en este proceso son las siguientes:

- Se realiza en promedio 7 veces de lunes a sábado, 44 veces en una semana; y solo se registra la cantidad de asistentes.
- Las personas encargadas de la recolección de datos solo realizan un conteo simple de los asistentes, y no necesariamente cuentan solo estudiantes, ya que si en el aula esta algún acompañante de un verdadero estudiante es posible que sea contado.
- La recolección del dato puede o no realizarse en un momento o muy temprano o muy tarde en la clase, lo que implica que algunos estudiantes o aún no han llegado o ya se retiraron del aula, implicando que no se recoge un dato exacto de los asistentes a clase.
- El personal recolector en ocasiones interrumpe al docente en su exposición para realizar el conteo o al directamente preguntarle por el número de asistentes, esto en ocasiones genera incomodidad en algunos docentes y estudiantes.
- Se destinan dos y hasta tres personas a todo el proceso de conteo en un mismo horario, pudiendo estas destinar su tiempo a otras actividades.
- Los administradores que deben ingresar los datos al sistema informatizado, y hasta los mismos directores y decanos que reciben la información, en ocasiones no están disponibles en el momento específico para realizar su función, por lo que la entrega de los datos puede ser retrasada involuntariamente y por eso no estar disponibles de inmediato.

A partir de lo anterior, se plantea el siguiente árbol de problemas:

- Efectos directos
  - Recolección inexacta de los datos de asistentes a un evento/clase o recinto.
  - Error en el procesamiento de los datos de los asistentes a un evento/clase o recinto.
  - Mal inversión del tiempo del recurso humano destinado a estas tareas.
  - No se cuenta con un listado de asistentes, solo con una cantidad de asistentes.
- Efectos indirectos
  - Interrupciones involuntarias durante las clases debido al proceso manual de conteo de los asistentes.
  - Ingreso de personas ajenas a la institución o a la clase.
  - Pérdida de reputación y credibilidad en procesos administrativos de la institución.
- Causas directas
  - El conteo es realizado por más de una persona en distintos rangos de tiempo y produce apreciaciones diferentes.
  - Digitación incorrecta al introducir los datos al sistema.
  - El individuo, como tal, se cuenta como un número y no como una persona con información significativa que está registrada en la base de datos de la institución.
- Causas indirectas
  - Costos de la implementación.
  - Usuarios que no hacen buen uso del sistema.
  - Oposición al cambio tradicional de procesos.

1.2 *Justificación*

La investigación aplicada es una forma de llevar soluciones de alto nivel científico a los problemas de la sociedad actual, y más cuando su implementación permite brindar beneficios colaterales.

En la actualidad, las instituciones de servicio de cualquier índole están obligadas a mejorar sus procesos internos y de atención al usuario o cliente. Este mejoramiento implica muchas acciones e implementaciones

en variadas áreas de la institución, desde la administración y la producción hasta el servicio al cliente, ya sean estas mejoras de índole humana, logística, técnica y de infraestructura. Estamos frente a una revolución en los procesos y técnicas en las instituciones. La revolución llamada *Industria 4.0*, o cuarta revolución industrial, se refiere a llevar técnicas de automatización a los procesos dentro de las instituciones (Goasduff, 2016), con el objetivo de ayudar a la eficiencia de estas tareas o procesos, buscando el aumento de productividad y ahorro de costos. Bajo las perspectivas de la Industria 4.0, en un futuro próximo las instituciones que no se adapten y automaticen sus procesos estarán condenadas a ser reconocidas como empresas obsoletas y atrasadas, tecnológicamente hablando (Vuksanović, 2017).

Registrar la asistencia exacta en un centro educativo permite llevar un mayor control de la población estudiantil, con lo que se puede sacar ciertos estadísticos en la toma de decisiones a corto y largo plazo. La asistencia a un recinto donde se efectúa una actividad, ya sea de carácter voluntario u obligatorio, puede ayudar a un sinnúmero de procesos, tener el número de los asistentes a dicha actividad hasta asignar una nota. La realidad salvadoreña cada día demanda una mayor atención en el área de la seguridad, por ejemplo, una institución como la Utec, que es la universidad privada más grande de El Salvador en cuanto a población estudiantil, se ve en la necesidad de disponer de un mecanismo que proporcione un control de quienes ingresan a la institución, y que además sea un claro mensaje a personas ajenas de que el monitoreo de estudiantes es una labor permanente dentro de la institución. En la actualidad, una propuesta a bajo costos para la identificación de usuarios, empleados y asistentes, son los sistemas de magnetización por medio de identificación por radiofrecuencia (RFID, por sus siglas en ingles), por lo que el trabajo de esta investigación un sistema de bajo coste que permita tener en cada aula un dispositivo electrónico que registre a cada estudiante que ingresa a un aula mediante el uso de un carné RFID similar a los utilizados por el cuerpo docente hora-clase, y que además el mismo sistema notifique en tiempo real, vía acceso a un sitio web, la cantidad de asistentes a los administradores, directores y decanos responsables, así como el listado de la identificación de quienes han asistido.

En un recinto o edificio que se utilice para realizar concentraciones o eventos, específicamente de índole académica como los que realiza la Utec, es de suma importancia para sus procesos administrativos y operativos tener un registro de la cantidad de estudiantes que asisten a cada una de las actividades que se realizan a diario. Este dato es muy importante para los administradores, ya que a partir de este se pueden tomar medidas y acciones sobre el funcionamiento de la institución, y así mejorar o corregir el servicio a la población estudiantil.

En el ámbito salvadoreño, no existen aplicaciones de este tipo en ninguna institución educativa, sin embargo, sí están implementados en diferentes empresas sistemas de control de asistencia para empleados con tarjetas RFID estos sistemas registran la hora de entrada y salida del empleado y las almacenan vía alámbrica en una base de datos local en el departamento de recursos humanos de la institución. Un sistema para registro de estudiantes con capacidad de notificar por internet a diferentes personas cada segmento de horario de clases, no existe en el ambiente educativo local. En el mercado comercial de El Salvador existen empresas (representantes de marcas y revendedores) dedicadas a implementar este tipo de soluciones, que a la larga son de costo elevado y no permiten una flexibilidad de adaptación a cada institución, ya que son sistemas cerrados diseñados y construidos por terceros. Cabe destacar que un sistema con las características planteadas por este trabajo no está disponible en el mercado (Ejje, 2017).

La propuesta de esta investigación tiene los siguientes bloques o etapas funcionales que se han de diseñar e implementar:

- Usuario: con su tarjeta RFID debidamente programada, con su número único de identificación o número de carné.
- Circuito electrónico: con la función de escaneo y decodificación de la información de la tarjeta, además de permitir la conexión y transferencia de datos a internet.
- Plataforma IoT: *software* en internet diseñado para la captura, inserción de datos y presentación, esto alojado en un servicio de bajo costo.

Algunos beneficios de la presente investigación aplicada son los siguientes:

- Generación de conocimiento científico nuevo en técnicas y procedimientos.
- Brindar una solución de automatización con tecnología de punta y a bajo costo.
- Mejoras en el proceso de conteo de asistentes: ahorro en tiempo de realización de la tarea, adecuada utilización del personal académico, monitoreo y accesibilidad de datos en tiempo real, comodidad para los asistentes y encargados de grupo de clase, sensación de modernidad de la institución.
- Ventaja competitiva y prestigio tecnológico a la institución.

Cabe destacar que la investigación aplicada debe proveer conocimientos, técnicas y soluciones prácticas a problemáticas de las sociedades, instituciones, industrias, empresas, etc., por lo que las universidades deben de ser un ente generador de ideas y diseños científicos; y la industria deberá proveer el campo de experimentación. Este binomio en El Salvador ha estado un poco olvidado, y por eso que con este proyecto se pretende dar un aliciente a un futuro de colaboradores que conlleven un crecimiento académico, científico y social en el país.

Además, los resultados de esta propuesta de investigación aplicada se estima que podrán ser replicados y de beneficio no solo para la empresa estudiada, sino también para muchas otras de la industria, como mercados, supermercados, almacenadoras, importadores, exportadores, etc. Otro aspecto importante es que esta investigación pretende ser el inicio de una línea investigativa que puede derivar en otros proyectos afines, además de ser la base para realizar implementación social institucional para centros educativos del país.

1.3 *Objetivos del estudio*

En el desarrollo de esta investigación se plantearon los objetivos siguientes:

1.3.1 *Objetivo general*

Implementar un sistema IoT de bajo costo para el registro automatizado vía internet de datos de asistentes a un recinto, como apoyo a las labores de registro de datos y fortalecimiento de la seguridad de las personas que ingresan a la institución.

1.3.2 *Objetivos específicos*

- Aportar y divulgar nuevo conocimiento científico, teórico y práctico, sobre el diseño e implementación de sistemas de IoT eficientes y de bajo costo en la automatización de tareas.
- Diseñar y construir un circuito electrónico basado en un sistema microcontrolador que sea capaz de registrar datos de un usuario que utilice tarjetas RFID en su ingreso a un evento/clase o recinto, además, que permita enviar estos datos, cantidades y nombres a través de una red local inalámbrica.
- Diseñar un *firmware* exclusivo para el funcionamiento del prototipo electrónico, escrito en lenguaje C++, que sea capaz de controlar el prototipo propuesto.
- Construir y acoplar el prototipo diseñado en un aula y un recinto para la validación de resultados.

2. Marco teórico

2.1 *Internet de las cosas*

El IoT es un concepto un poco abstracto, pero que ha estado ganando bastante popularidad en años recientes. La idea que intenta representar queda bastante bien ilustrada por su nombre: cosas cotidianas que se conectan a internet, pero en realidad se trata de mucho más que eso. En un sentido amplio, ya no solo tenemos objetos que hacen algo, tenemos objetos con algún tipo de procesamiento electrónico

computacional, es decir, computadoras que encienden la luz, mantienen el aire acondicionado de un edificio, transportan personas o materiales, preparan nuestro café, se encargan de la seguridad de nuestro hogar, en resumen, un sinfín de tareas que están siendo automatizadas y conectadas a internet mediante sistemas electrónicos computacionales dedicados al IoT (Tollervey, 2017).

Para entender cómo funciona el IoT, debemos también comprender que sus fundamentos no son en lo absoluto nuevos. Desde hace unos 30 años se viene trabajando con la idea de hacer un poco más interactivos todos los objetos de uso cotidiano. El IoT potencia objetos que antiguamente se conectaban mediante circuito cerrado, como comunicadores, cámaras, sensores y demás; y les permite comunicarse globalmente mediante el uso de la red de redes.

Si tuviéramos que dar una definición del IoT, probablemente lo mejor sería decir que se trata de una red que interconecta objetos físicos valiéndose del internet. Los objetos mencionados se valen de sistemas embebidos, o lo que es lo mismo, *hardware* especializado que le permite no solo la conectividad a internet, sino que además programa eventos específicos en función de las tareas que le sean dictadas remotamente (Torres, 2014).

IoT es la revolución tecnológica que ha cambiado al mundo a corto plazo, pero que influirá más de forma drástica en la vida diaria, en la opinión de los analistas de mercado, los emprendimientos locales más destacados y los hogares inteligentes. Ejemplos sobresalientes de esto son zapatos deportivos que cuentan kilómetros, collares para perros que informan a sus dueños sobre la localización de estos, jarras que mandan mensajes cuando alguien se excedió con la ingesta de alcohol, medias que ponen pausa en la película o serie al momento en que el espectador se quedó dormido, y millones de otros *gadgets*, útiles o no tanto, que por ahora solo están en la imaginación o en los laboratorios de los inventores del mañana (Shutterstock, 2016).

2.1.1 *Internet de las cosas en el presente*

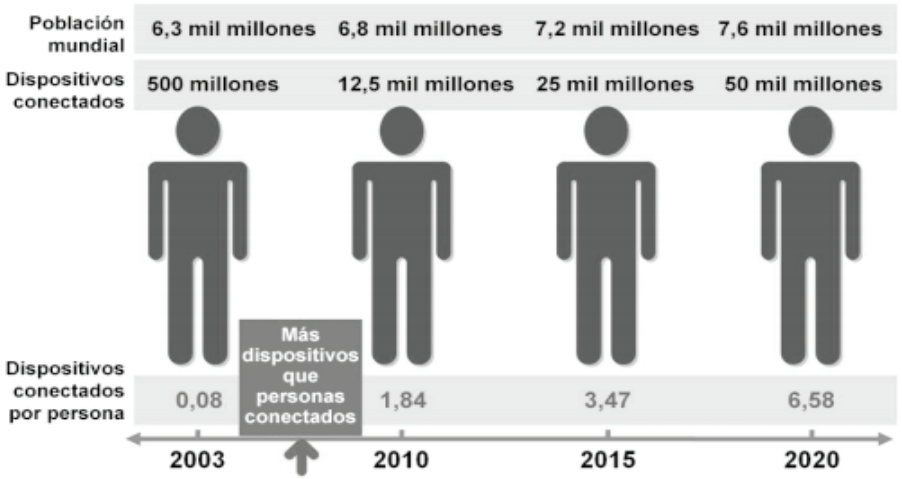
Antes de analizar el estado actual del IoT, es importante ponerse de acuerdo con una definición. Según el Grupo de Soluciones Empresariales de Internet (IBSG, siglas del inglés) de Cisco, IoT es sencillamente el

punto en el tiempo en el que se conectaron a internet más "cosas u objetos" que personas. (Cisco, 2011).

En 2003, había aproximadamente 6.3 mil millones de personas en el planeta, y había 500 millones de dispositivos conectados a internet. Si dividimos la cantidad de dispositivos conectados entre la población mundial, el resultado indica que había menos de un dispositivo (0.08) por persona. De acuerdo con la definición de Cisco IBSG, el IoT aún no existía en 2003 porque la cantidad de cosas conectadas era relativamente escasa, dado que apenas comenzaba la invasión de los dispositivos omnipresentes, como los *Smartphone*.

El crecimiento explosivo de los *Smartphone* y de las *tabletas* se elevó a 12.5 mil millones en 2010 la cantidad de dispositivos conectados a internet, en tanto que la población mundial aumentó a 6.8 mil millones, por lo que el número de dispositivos conectados por persona es superior a 1 (1.84, para ser exactos). Si se desglosan aún más estas cifras, Cisco IBSG estima que el IoT inicio en algún punto entre 2008 y 2009 (ver figura 1). Actualmente, el IoT está firmemente encaminada, según lo demuestra el avance de iniciativas como Planetary Skin de Cisco, la matriz inteligente y los vehículos inteligentes (Dave Evans, 2011).

Figura 1. Internet de las cosas "nació" entre los años 2008 y 2009 (Cisco, 2011)



Fuente: Cisco IBSG, abril de 2011

2.1.2 *Internet de las cosas en el futuro*

Estamos en el mes de octubre y ya casi finalizando el 2017, sin embargo, se pronostica que, a finales de este año, más de 8.400 dispositivos estarán conectados y en funcionamiento en todo el mundo, generando un volumen de negocio que alcanzará los 2 billones de dólares; lo que supone un incremento de casi un tercio con respecto a 2016, en un mercado que llegará a las 20.400 millones "cosas" conectadas a finales de 2020. A nivel de regiones del mundo, China, Norteamérica y Europa Occidental son las tres que dirigen el empleo de las cosas conectadas a internet, representando en su conjunto el 67 % de toda la base de IoT instalada en 2017. El segmento correspondiente a consumo es el más extenso, con 5.200 millones de unidades conectadas a la red en 2017, lo que representa el 63 % del número total de aplicaciones en uso, seguido del sector empresarial, que emplearán 3.100 millones de cosas conectadas a internet a finales de este año (Gartner, 2017).

Aparte de la industria de la automoción, las aplicaciones que entrarán mayormente en uso por parte de los consumidores serán televisores digitales; en tanto que los contadores eléctricos inteligentes y las cámaras de seguridad en establecimientos comerciales serán las "cosas" que más emplearán los negocios. "Los servicios IoT son fundamentales para el despegue del mercado de los dispositivos de internet de las cosas", ha señalado Denise Rueb, directora de investigación de Gartner, quien ha añadido que el gasto total en servicios IoT, correspondiente a los mercados profesional, de consumo y de servicios de conectividad alcanzará los 273.000 millones de dólares a finales de 2017. Según esta experta, los servicios estarán dominados por tecnología IoT operativa y orientada a los profesionales; una categoría en la que los profesionales de empresas de canal asistirán a los negocios en la implementación, el diseño y la operatividad de sistemas IoT (Tendencias, 2017)

2.1.3 *Aplicaciones de* IoT

Dentro del área de aplicación de la IoT, se encuentran múltiples y diversos campos en los cuales los sistemas de este encuentran cabida,

sin embargo, se mencionan algunos según una publicación realizada por Intel (Intel, 2017), que son los siguientes:

- Automotriz: cuando se lo vincula con IoT, el automóvil convierte los datos en una perspectiva que permite actuar tanto dentro del automóvil como en el mundo que lo rodea.
- Energía: mediante IoT, los innumerables dispositivos de la red eléctrica pueden compartir información en tiempo real para distribuir y manejar la energía en forma más eficiente.
- Atención médica: desde dispositivos en ropas de uso clínico hasta tabletas para servicios de emergencia y equipos quirúrgicos sofisticados, IoT está transformando los servicios de salud.
- Fabricación inteligente: la tecnología de IoT permite que las fábricas de hoy liberen la eficacia operacional, optimicen la producción y aumenten la seguridad de los trabajadores.
- Comercio minorista: para los comerciantes minoristas, IoT ofrece oportunidades ilimitadas que aumentan la eficacia de la cadena de suministro, desarrollan nuevos servicios y rediseñan la experiencia del cliente.
- Edificios inteligentes: IoT está dando respuesta a los costos crecientes de la energía, la sustentabilidad y la conformidad con códigos conectando, administrando y asegurando los dispositivos que recopilan datos de los sistemas centrales.
- Casas inteligentes: desde reconocer su voz hasta saber quién está en la puerta principal, la tecnología IoT está convirtiendo en realidad el sueño de una casa inteligente segura.
- Transporte inteligente: desde automóviles conectados o de autoconducción hasta sistemas de logística y transporte inteligentes, IoT puede salvar vidas, reducir el tráfico y minimizar el impacto de los vehículos en el ambiente.

2.1.4 *Componentes de un Sistema* IoT

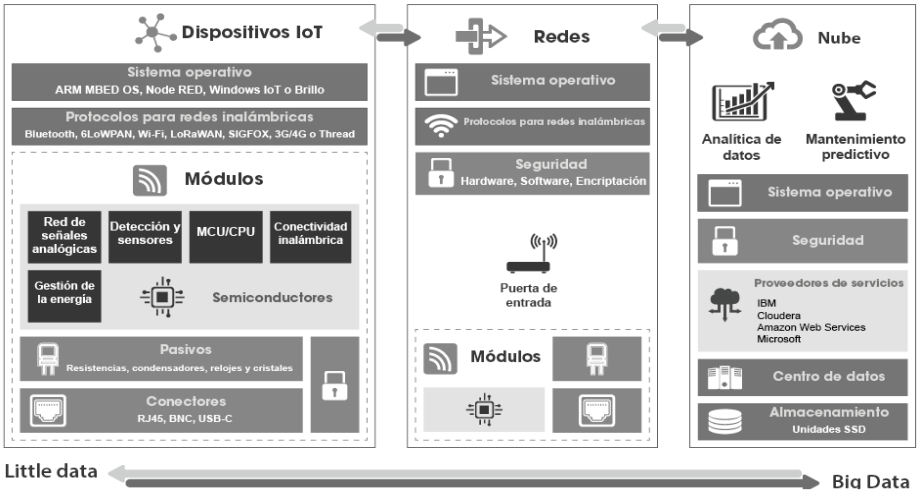
El IoT es un sistema de dispositivos informáticos interrelacionados, máquinas mecánicas y digitales, objetos, animales o personas que cuentan con identificadores únicos y con la capacidad de transferir datos a través de una red sin requerir de humano a humano o humano

interactuando con computadoras. Un sistema IoT completo integra cuatro componentes: sensores/dispositivos, conectividad, procesamiento de datos y una interfaz de usuario.

A continuación, se lista lo que un sistema IoT completo necesita:

- ***Hardware*,** como sensores o dispositivos. Estos recopilan datos del entorno (por ejemplo, un sensor de humedad) o realizan acciones en el entorno (por ejemplo, cultivos de riego).
- **Conectividad.** El *hardware* necesita una forma de transmitir toda esa información a la nube (por ejemplo, enviar datos de humedad) o necesita una forma de recibir comandos de la nube (por ejemplo, regar los cultivos ahora). Para algunos sistemas IoT, puede haber un paso intermedio entre el *hardware* y la conexión a la nube, como una puerta de enlace o enrutador.
- ***Software*.** Este *software* está alojado en la nube y es responsable de analizar los datos que recopila de los sensores y toma decisiones (por ejemplo, saber, a partir de datos de humedad, que simplemente llovió y luego decirle al sistema de riego que no se encienda hoy).
- **Interfaz de usuario.** Para hacer que todo esto sea útil, es necesario que los usuarios interactúen con el sistema IoT (por ejemplo, una aplicación web con un tablero que muestre tendencias de humedad y permita a los usuarios activar o desactivar manualmente los sistemas de riego).

Figura 2. Arquitectura generalizada de un sistema IoT (RS, 2017)



Las plataformas de IoT son el *software* de soporte que conecta todo en un sistema de IoT. Una plataforma IoT facilita la comunicación, el flujo de datos, la administración de dispositivos y la funcionalidad de las aplicaciones (ver figura 2).

Cuando varios dispositivos envían estos pequeños datos, a través de una red a la nube, se pueden monitorizar, y así, con el tiempo, la cantidad de datos será más grande. Con frecuencia esto se conoce como *big data*; y es aquí cuando el IoT se hace inteligente. Los *big data* permiten analizar miles o millones de puntos de datos con el fin de aprender, entender o controlar algo más a fondo.

2.1.5 *Ciudades inteligentes*

Una *smart city*, o ciudad inteligente, se puede describir como aquella que aplica las tecnologías de la información y de la comunicación con el objetivo de proveerla de una infraestructura que garantice lo siguiente:

- Un desarrollo sostenible.
- Un incremento de la calidad de vida de los ciudadanos.

- Una mayor eficacia de los recursos disponibles.
- Una participación ciudadana activa.

Por lo tanto, son ciudades que son sostenibles en lo económico, social y medioambiental. La *smart city* nace de la necesidad de mantener una armonía entre estos aspectos. Se prevé que en el 2050 un 85 % de la población mundial vivirá en ciudades o centros urbanos (Educa, 2014). Por lo que se prevé que en las siguientes décadas los núcleos urbanos tengan que afrontar un número creciente de problemas ligados a este número elevado de personas, situaciones como estas:

- El abastecimiento energético.
- Las emisiones de dióxido de carbono.
- La planificación del tráfico automovilístico.
- La provisión de bienes y materias primas.
- La prestación de servicios sanitarios y de seguridad a los residentes de enormes y masificados centros de población.

Hay diferentes parámetros por los que se valora más a una ciudad que otra. Para ello se consideran diez dimensiones que son clave: gobernanza, planificación urbana, gestión pública, tecnología, medio ambiente, proyección internacional, cohesión social, movilidad y transporte, capital humano y economía. Las cinco ciudades que de alguna manera cumplen con lo anterior son las siguientes:

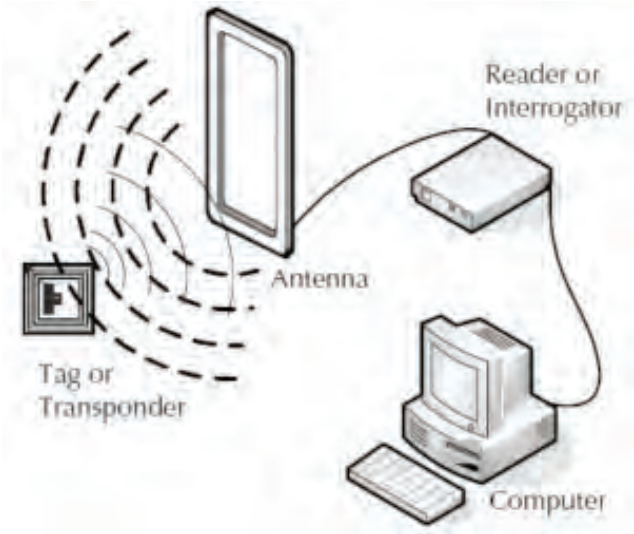
- Tokio: es la ciudad que mejor situada está en el *ranking* de 2013, con el primer puesto en capital humano y gestión pública. Sin embargo, en cohesión social ha quedado muy relegada sobre todo por el terremoto de Fukushima y el posterior tsunami.
- Londres: mantiene niveles altos en casi todas las dimensiones, y destaca especialmente en proyección internacional y tecnología. Sin embargo, en gestión pública y cohesión social tiene valores relativamente bajos.
- Nueva York: es la ciudad más poblada de Estados Unidos y la segunda aglomeración urbana del continente después de Ciudad de México. Es una de las ciudades más importantes en cuanto a capital humano y economía del mundo.

- Zúrich: se trata de la principal ciudad de Suiza, y es el motor financiero y centro cultural del país. Destaca en las dimensiones medio ambiente, y movilidad y transporte.
- París: es el destino turístico más popular del mundo, superando los 40 millones de turistas extranjeros al año. Sobresale en proyección internacional, tecnología, y movilidad y transporte.

2.2 *Identificación por radiofrecuencia*

La identificación por radiofrecuencia, o RFID (siglas de *Radio Frequency Identification*), es una tecnología de identificación remota e inalámbrica en la cual un dispositivo lector, vinculado con un equipo de cómputo, se comunica a través de una antena con un transponedor (también conocido como *tag* o etiqueta) mediante ondas de radio. En la figura 3 se pueden apreciar los elementos que componen un sistema de comunicación por RFID. Esta tecnología, que existe desde los años 40, se ha utilizado y se sigue utilizando para múltiples aplicaciones, incluyendo casetas de peaje, control de acceso, identificación de ganado y tarjetas electrónicas de transporte.

Figura 3. Elementos en un sistema de comunicación RFID
(*Albanian Times*, 2017)



2.2.1 *Aplicaciones y ventajas de la tecnología* RFID

El sistema de comunicación por RFID tiene muchas aplicaciones (Actum, 2017), ya que por su bajo costo son cada vez más utilizados en diversos campos. A continuación, se presenta un breve listado de algunas aplicaciones.

- Los sistemas de baja frecuencia se usan para la identificación de animales, seguimiento de materiales, llave de automóviles, identificación industrial.
- Los *tags* o etiquetas RFID de alta frecuencia se utilizan en bibliotecas, seguimiento de libros, de palés, control de acceso, seguimiento de equipaje o de ropa.
- Las etiquetas de UHF se utilizan comúnmente de forma comercial en seguimiento de palé y envases, camiones y remolques en envíos o en sistemas de distribución o antirrobo.
- Los transponedores RFID, de microondas, se utilizan en el control de acceso en vehículos y peajes de autopistas

Algunas de las ventajas de la tecnología RFID sobre el código de barras se enlistan a continuación:

- No requiere una línea de visión.
- No requiere de intervención humana (ideal para automatizar).
- Distancias de lectura de 1 a 10 metros.
- Lectura simultánea de múltiples artículos (protocolo anticolisión).
- Hasta 500 lecturas por minuto (5 veces más rápido que un código de barras).
- No le afectan los ambientes sucios.
- Capacidad de lectura y escritura.

El principal inconveniente de esta tecnología es que en ocasiones la lectura de datos es defectuosa cuando las etiquetas RFID están inmersas en materiales líquidos o metálicas. Otro inconveniente que presentan es que, si utilizamos dos lectores a la vez para una misma tarjeta RFID, esta no podrá dar una información correcta, ya que los dispositivos lectores cruzarán sus ondas; y la tarjeta no es capaz de responder a dos consultas

simultáneas. Los estándares de RFID abordan cuatro las siguientes áreas fundamentales:

- Protocolo en la interfaz aérea: especifica el modo en el que etiquetas RFID y lectores se comunican mediante radiofrecuencia.
- Contenido de los datos: especifica el formato y semántica de los datos que se comunican entre etiquetas y lectores.
- Certificación: prueba que los productos deben cumplir para garantizar el desarrollo de los estándares y pueden comunicarse con otros dispositivos de distintos fabricantes.
- Como en otras áreas tecnológicas, la estandarización en el campo de RFID se caracteriza por la existencia de varios grupos de especificaciones competidores. Por una parte, está ISO, y por otra, Auto-ID Centre (conocida desde octubre de 2003 como EPCglobal, de EPC, Electronic Product Code). Ambas comparten el objetivo de conseguir etiquetas de bajo coste que operen en UHF.

2.3 Hardware *para sistemas* IoT

Implementar un sistema de IoT requiere un componente de *hardware* fuertemente integrado en el entorno físico, capaz de interactuar con este, percibir su estado y transmitir información acerca de él; y un componente de *software* adecuado para gestionar la información generada y actuar sobre el *hardware* anteriormente mencionado.

El *hardware* utilizado en los sistemas IoT incluye dispositivos para un panel de control remoto, dispositivos para control, servidores, un dispositivo de enrutamiento o puente y sensores. Estos dispositivos administran las tareas y funciones clave, como la activación del sistema, las especificaciones de acción, la seguridad, la comunicación y la detección, para respaldar objetivos y acciones específicos.

2.3.1 *Sensores*

El *hardware* más importante en IoT podría ser sus sensores. Un sensor se usa en la detección mediante diversos dispositivos de medición pasivos

y activos. Aquí hay una lista de algunos de los dispositivos de medición utilizados en IoT:

- acelerómetros,
- sensores de temperatura,
- magnetómetros,
- sensores de proximidad,
- giroscopios,
- sensores de imagen,
- sensores acústicos,
- sensores de luz,
- sensores de presión,
- sensores RFID,
- de gas,
- sensores de humedad y
- sensores de micro flujo.

2.3.2 *Procesador: tarjeta* Raspberry Pi

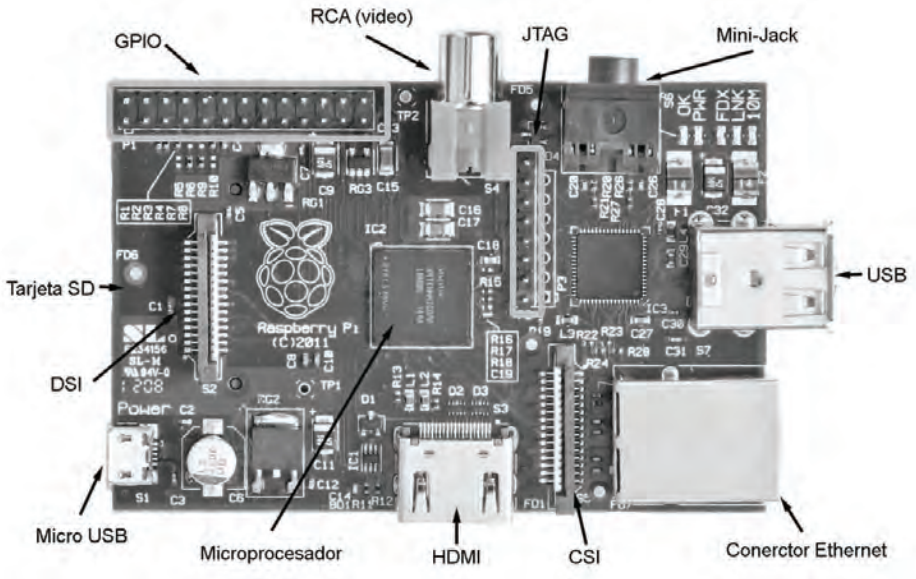
Para mucho es conocido que uno de los pesos pesados en cuanto a ordenadores de placa reducida es el Arduino. Es el dispositivo más barato y sencillo de implementar cuando se necesite desarrollar un entorno de IoT. Por otro lado, es la alternativa básica, con todo lo que ello conlleva. Dispone de un *hardware* menos potente que sus competidores y algunas limitaciones en cuanto al *software* y el sistema operativo.

El segundo dominador del mercado de estas placas de desarrollo es Raspberry Pi (González, 2016). Es un pequeño ordenador capaz de alojar un sistema operativo y con mejores prestaciones que Arduino. Su precio superior (aunque no mucho) hace que debamos considerar el uso de este dispositivo o de un Arduino en función de las necesidades *software* que tenga nuestro sistema.

Raspberry Pi sería ideal para instalar un entorno domótico en un hogar, gestionando información de distintos sensores e interactuando con elementos de la vivienda. Estos sensores podrían estar conectados directamente en la Raspberry Pi o a alguna placa auxiliar que hiciera de intermediario (por ejemplo, un Arduino).

Con el éxito de estas tecnologías, han surgido empresas y productos alternativos que ofrecen variedad para escoger el dispositivo más adecuado para el sistema de IoT personal.

Figura 4. Componentes de una tarjeta Raspberry Pi (Zona Maker, 2016)



En cuanto al *hardware*, no se indica expresamente si es libre o con derechos de marca. En su web oficial explican que disponen de contratos de distribución y venta con dos empresas, pero al mismo tiempo cualquiera puede convertirse en revendedor o redistribuidor de las tarjetas Raspberry Pi, por lo que se entiende que es un producto con propiedad registrada, manteniendo el control de la plataforma, pero permitiendo su uso libre tanto a nivel educativo como particular. El *software* que utiliza es de código abierto, se basa en GNU/Linux, siendo su sistema operativo oficial una versión adaptada de Debian, denominada *Raspbian*, aunque permite usar otros sistemas operativos, incluido una versión de Windows 10. En todas sus versiones incluye un procesador Broadcom, una memoria RAM, una GPU, puertos USB, HDMI, Ethernet (el primer modelo no lo tenía), 40 pines GPIO y un conector para cámara (Zona Maker, 2016).

2.3.3 *Procesador: tarjeta* NodeMCU

La placa de desarrollo NodeMCU está basada en el popular chip que revolucionó el Wi-Fi en sistemas embebidos, el microcontrolador ESP8266 (GeekFactory, 2017). Con este sencillo módulo se puede realizar el prototipo de cualquier sistema para el IoT en cuestión de horas. El concepto es muy similar al de Arduino, es decir, un microcontrolador conectado a través de un puente USB-Serial que interactúa con un *software* en la PC.

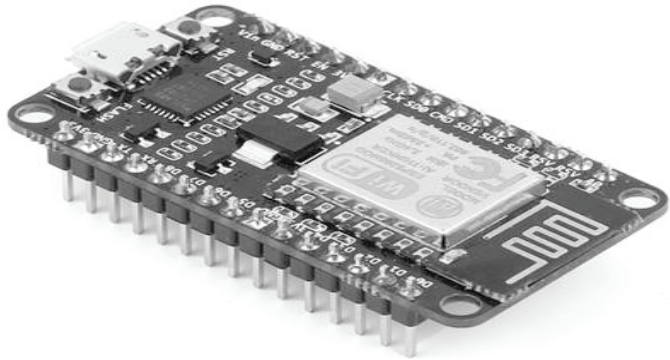
El ESP8266 en el NodeMCU es más que un simple circuito integrado para Wi-Fi. Se trata de un SoC (*system on a chip*) que integra en una sola pieza de silicio un procesador de aplicaciones con la electrónica necesaria para la comunicación por RF (Wi-Fi). Esta placa permite aprovechar el procesador que está dentro del ESP8266 y realizar *software* que corre en él, no solamente usarlo como un puente entre un microcontrolador y la red. Esta placa viene cargada con el *firmware* NodeMCU, sin embargo, también puede usarse como una excelente plataforma para desarrollar, evaluar y experimentar otros firmwares para el ESP8266. Dentro de las características técnicas de esta placa de desarrollo se pueden mencionar las siguientes:

- Procesador principal: ESP8266
- Protocolo inalámbrico 802.11 b/g/n
- TCP/IP integrado
- Potencia de salida +19.5dBm en modo 802.11b
- Sensor de temperatura integrado
- Corriente en espera: < 10uA

La capacidad de conectarse a una red, su pequeño tamaño y su precio tan asequible han hecho que este dispositivo sea muy utilizado en el desarrollo de sistemas IoT. La posibilidad de conectar el mundo físico de los sensores con el de internet y de "la nube" abre un gran abanico de posibilidades para el diseño de proyectos con ESP8266 para el IoT (Martín, 2017). El ESP8266EX se suele utilizar en una placa que incluye casi todo lo necesario. Las placas más básicas ESP-01, ESP-03, ESP12, etc., necesitan de un adaptador de USB a interfaz serie TTL. Pero las nuevas placas NodeMCU, Adafruit Huzzah, Witty Cloud, SparkFun

ESP8266 Thing o las WeMos D1 Mini y Pro sí incluyen todo lo necesario para comenzar a desarrollar sobre ellas. A pesar de que estas últimas pueden ser algo más costosas, es muy recomendable utilizarlas para la realización de prototipos.

Figura 5. Tarjeta NodeMCU con microcontrolador ESP8266 (Martín, 2017)



2.4 *Plataforma de* software *para sistemas* IoT

En palabras simples, el propósito de cualquier dispositivo IoT es conectarse con otros dispositivos y aplicaciones IoT (principalmente basados en la nube) para transmitir información, usando protocolos de transferencia de internet. La brecha entre los sensores del dispositivo procesador y las redes de datos locales debe ser zanjada por una plataforma IoT, que son las aplicaciones de *back-end*[38] para dar sentido a la gran cantidad de datos generados por el *hardware* de sensores; y que permite la conectividad entre el *hardware* de sensores con las aplicaciones de interfaz al usuario u otro dispositivo en el otro extremo del sistema. Hay muchas plataformas de IoT disponibles hoy en día que ofrecen la opción de implementar aplicaciones de IoT con relativa facilidad técnica

38 *Front-end* y *back-end* son términos de informática que se refieren a la separación de intereses entre una capa de presentación y una capa de acceso a datos, respectivamente (wikipedia.com).

y en muy poco tiempo. Se deben considerar varios aspectos para elegir la plataforma adecuada, considerando que existe una distinción clave entre las plataformas de internet de las cosas y las plataformas de IoT de consumo. Así que hay que considerar si la aplicación es industrial (como la de petróleo y gas y la de fabricación o gestión de activos), o está destinada a consumidores [como aplicaciones domésticas inteligentes o dispositivos portátiles] (Mclelland, Leverege, 2016).

Las plataformas IoT industriales y las de consumo pueden diferir significativamente debido a sus diferentes necesidades. Para las primeras, una falla en el sistema puede ser extremadamente importante, tal vez costando millones de dólares o incluso vidas. Para las segundas, una falla podría ser simplemente un inconveniente para el usuario final. E incluso en segmentos industriales o de consumo, las aplicaciones pueden tener necesidades de plataforma muy diferentes. Sin embargo, a pesar de la gran variación en las aplicaciones IoT, hay algunos elementos comunes que es crítico considerar cuando se evalúa la mejor plataforma IoT para su aplicación.

Para seleccionar la mejor plataforma de IoT, se debe considerar lo siguiente:

- La estabilidad de la plataforma: con tantas plataformas en el mercado, es probable que algunas fallen. Es importante elegir una plataforma que pueda funcionar durante varios años, de lo contrario la inversión podría desperdiciarse si el proveedor de la plataforma se retira. Se debe preguntar sobre clientes actuales y pasados. Si no tienen ninguno, probablemente no sea una buena señal.
- La escalabilidad y flexibilidad de la plataforma: las necesidades actuales van a cambiar con el tiempo. Por lo tanto, hay que asegurarse de que la plataforma funcione a pequeña escala y recién esté comenzando, pero que también funcionará cuando (con suerte) a gran escala.
- Además de ser escalable, la plataforma debe ser lo suficientemente flexible como para mantenerse al día con tecnologías, protocolos o características que cambian rápidamente. Las plataformas flexibles son a menudo las que se basan en estándares abiertos y se comprometen a mantener el ritmo con la evolución de los protocolos, estándares y tecnologías de IoT.

- También es importante que la plataforma sea independiente de la red. Esto significa que puede integrarse y trabajar con todos los principales sistemas tecnológicos, en lugar de estar atado en un único proveedor.
- El trabajo anterior del proveedor de la plataforma: como se mencionó anteriormente, las aplicaciones de IoT pueden variar mucho. Si el proveedor de la plataforma ha realizado un trabajo anterior que es similar a nuestra aplicación, ese es un buen indicador de que pueden satisfacer nuestras necesidades específicas. Sin embargo, hay que señalar que no es necesario que sea una coincidencia exacta. Si se está construyendo una aplicación de agricultura inteligente, por ejemplo, se puede buscar un caso de uso con características similares. Esa sería una aplicación que también involucra cientos/miles de sensores que generan datos, una conectividad similar (como LPWAN) y análisis de datos aplicados para crear ideas útiles.
- El modelo de fijación de precios y su caso comercial: hay que asegurarse de que el proveedor de la plataforma sea transparente en sus precios, que muestren una tarifa introductoria y luego aumenten significativamente cuando realmente se vaya a registrar.
- Además, ¿cómo van a vender? Si están haciendo un modelo de suscripción, tiene sentido pagar una suscripción para el servicio de la plataforma IoT, ya que puede ajustar los costos a los precios. Sin embargo, si está vendiendo *hardware*, podría tener más sentido buscar una opción de plataforma con una licencia inicial para que pueda incluir eso en los costos de desarrollo del producto.

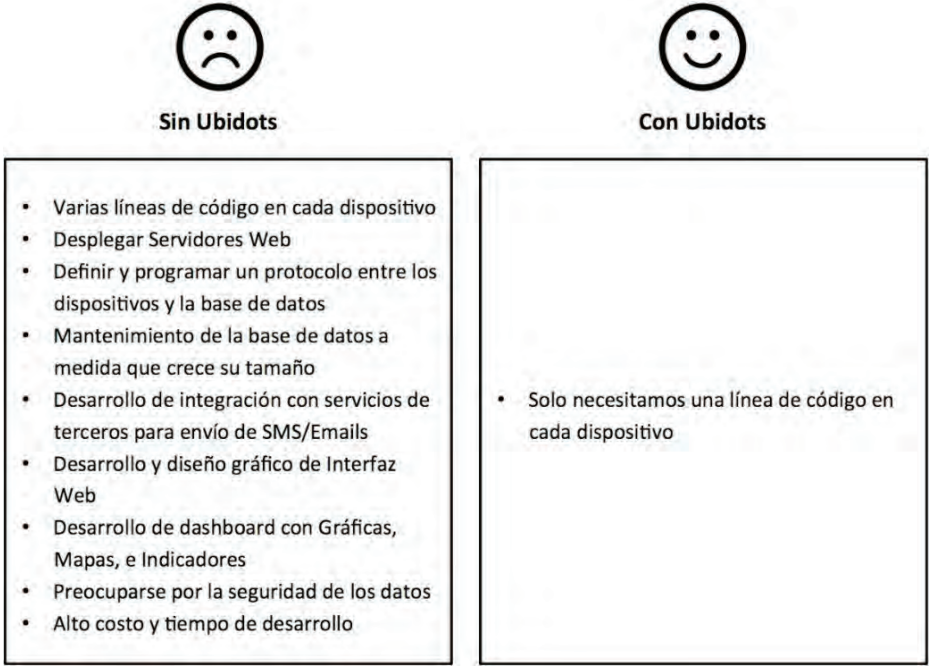
El valor del IoT está en los datos. Los datos pueden proporcionar información procesable sobre las operaciones o actividades cotidianas simples para reducir las ineficiencias o mejorar las experiencias. Debe buscar analíticas descriptivas básicas, visualización, diagnósticos, análisis predictivo y quizás incluso herramientas de aprendizaje automático. Además, hay que asegurarse de preguntar quién posee los datos. Si la respuesta no es simple (“Usted posee los datos generados por sus productos”), esta es una gran señal de advertencia porque, una vez más, el valor del IoT está en los datos.

2.4.1 Ubidots

Ubidots es un servicio en la nube que nos permite almacenar e interpretar información de sensores en tiempo real, haciendo posible la creación de aplicaciones para el IoT de una manera fácil, rápida y divertida (Ubidots, 2014). Gracias a esta herramienta, podremos ahorrarnos tiempo y dinero al momento de desarrollar aplicaciones como sistemas de telemetría GPS, sistemas para monitoreo de temperatura, aplicaciones para contar vehículos en una calle, etc.

Una gran ventaja de Ubidots es que ofrece un plan gratis, con el cual podemos realizar prototipos y aplicaciones cien por ciento funcionales. En la siguiente gráfica se ilustra el ahorro en tiempo y esfuerzo al crear una aplicación de IoT con la plataforma Ubidots, o sin ella.

Figura 6. Características de la plataforma Ubidots (Ubidots, 2014)



Para comenzar, tendremos que conectar nuestros dispositivos al API de Ubidots. Pero antes de ello veamos algunos conceptos de la plataforma:

- *Data source:* una fuente de datos se refiere a los datos generados por un dispositivo único. Cada *data source* puede tener uno o más sensores o variables. Por ejemplo, en una aplicación de transporte un vehículo sería un *data source*, y sus variables serían *Velocidad*, *GPS* o *RPM*.
- Variable: es un conjunto de datos que cambia en el tiempo. Por ejemplo, las variables de un *data source* llamado *Refrigerador* serían *Temperatura* y *Humedad*.
- *Value:* es el valor medido por el sensor en un instante de tiempo determinado.
- *Event:* los eventos son acciones que se pueden tomar según el valor de as variables. Por ejemplo, es posible configurar un evento para recibir un SMS si la velocidad de un vehículo es mayor que 100 kph.

Ubidots permite interactuar con cada uno de estos elementos de una manera programática, es decir, estos elementos pueden ser creados, modificados o eliminados mediante *hardware* o *software* a través de un API.

2.4.2 Google App Script IoT

Es una plataforma IoT inteligente que ayuda a obtener información útil para las empresas a partir de la red de dispositivos mundiales (Google Cloud, 2017). Google Cloud IoT es un conjunto de servicios completamente administrado e integrado que permite conectar, administrar e transferis datos a gran escala, así como de forma fácil y segura, a partir de dispositivos repartidos por todo el mundo. También ayuda a procesar, analizar y ver datos en tiempo real, implementar cambios operacionales y llevar a cabo las acciones que creamos pertinentes.

Google Cloud IoT es un conjunto completo de servicios integrados que permiten obtener información empresarial útil en tiempo real a partir de los datos de los ispositivos distribuidos por todo el mundo. Cloud IoT Core recopila datos de los dispositivos, que más tarde se

publican en Cloud Pub/Sub para su análisis. También permite realizar un análisis *ad hoc* mediante Google BigQuery, obtener datos analíticos y aplicarlos fácilmente a los sistemas de aprendizaje automático a través de Cloud Machine Learning Engine, o ver los datos del IoT resultantes con los completos informes y paneles de control de Google Data Studio.

Se puede obtener información útil sobre la eficacia operativa de los dispositivos como nunca antes. Gracias a la plataforma, se podrán administrar dispositivos distribuidos por todo el mundo y llevar a cabo actualizaciones del *firmware*. Google Cloud IoT es compatible con una gran variedad de sistemas operativos integrados y funciona perfectamente con Android Things. Además, funciona de forma nativa con dispositivos de los principales fabricantes, como Intel y Microchip, y es capaz de realizar acciones en tiempo real para implementar cambios mediante flujos de trabajo de Cloud Functions de forma automática.

La plataforma sin servidores que se ha creado para Cloud IoT permite librarse de la necesidad de desarrollar y mantener una infraestructura costosa para el análisis de los datos de IoT. Conecta fácilmente con la nube y aloja todos los dispositivos distribuidos por todo el mundo a través del puente de protocolos de Cloud IoT Core con balanceo de carga automático y escala horizontal. Se puede entonces disfrutar de la tranquilidad de saber que la seguridad de las conexiones de los dispositivos está garantizada gracias a los protocolos estándares del sector, y reduce el coste total de propiedad al no tener que cumplir requisitos de inversión de capital o de mantenimiento continuo (Google Cloud, 2017).

Específicamente, una aplicación de la plataforma de Google es el Apps Script, que no es más que un lenguaje de programación basado en JavaScript, y con un entorno de desarrollo y ejecución en la nube. Dicho de otra forma, es un lenguaje basado en JavaScript que, en vez de ejecutarse en local, se ejecuta en los servidores de Google; y que no está pensado para interactuar con el usuario, sino que está enfocado a procesar información.

Si bien tiene limitaciones, dispone de un abanico impresionante de posibilidades que no ofrece ningún otro lenguaje. Permite acceder a correos electrónicos, filtrarlos, borrarlos o reenviarlos, permite acceder a documentos de Google Drive: hojas de cálculo, carpetas, documentos de texto; crear, modificar y leer ficheros a placer, además de servir páginas

web y crear webs con contenido dinámico en que el HTML de la web se comunique con el código de Google App Script como si fuera PHP.

A nivel profesional, su gran potencia radica en que resulta muy fácil y rápido generar herramientas internas para verificar datos, generar alarmas, analizar información, buscar patrones y, en general, extraer el grano de la paja. (Bordas, 2017).

3. Metodología

3.1 *Método*

La investigación fue desarrollada usando el método multimodal-experimentación sin hipótesis, el cual es "un método empleado en casos donde la investigación tiene por objeto el provocar determinados fenómenos que no se presentan usualmente en la naturaleza y cuyo conocimiento puede ser interesante o importante en el avance de la ciencia y la tecnología" (Cegarra, 2004).

Específicamente en esta investigación, y a partir de la revisión teórica, se seleccionaron diversas herramientas y componentes electrónicos para el diseño de un prototipo electrónico que se apegara a los objetivos del estudio.

3.2 *Tipo de estudio*

Este estudio fue del tipo investigación aplicada tecnológica o investigación técnica. Según Cegarra (2004), este tipo de investigación "tiende a la resolución de problemas o al desarrollo de ideas, a corto o mediano plazo, dirigidas a conseguir innovaciones, mejoras en procesos o productos, incrementos de calidad y productividad, etc.".

3.3 *Sujeto de estudio*

Para la verificación del diseño propuesto, se realizó la implementación del diseño fruto de esta investigación aplicada dentro de un ambiente académico real en el campus central de la Utec.

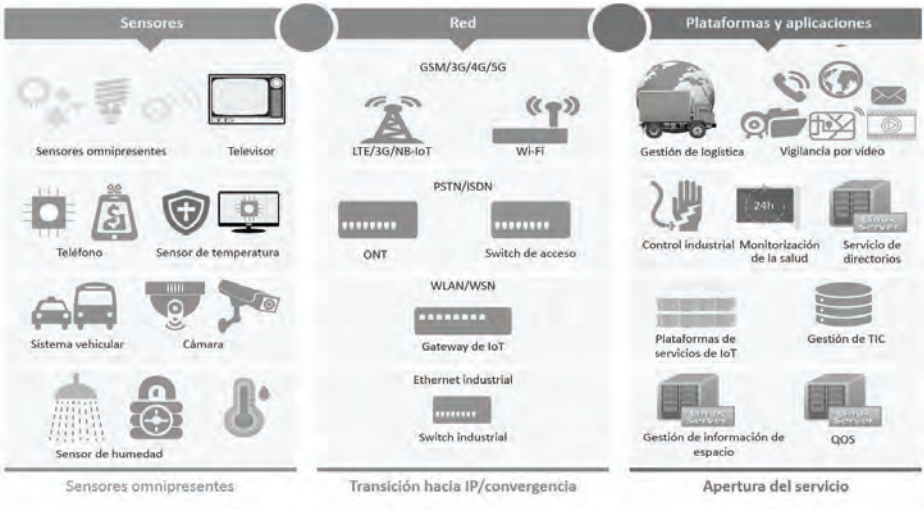
3.4 *Diseño del sistema*

3.4.1 *Arquitectura*

El primer paso para la implementación de un sistema de IoT es el diseño de la arquitectura de bloques del sistema, tomando en consideración el estado del arte, se establecieron las siguientes etapas funcionales necesarias para la implementación de un sistema de IoT:

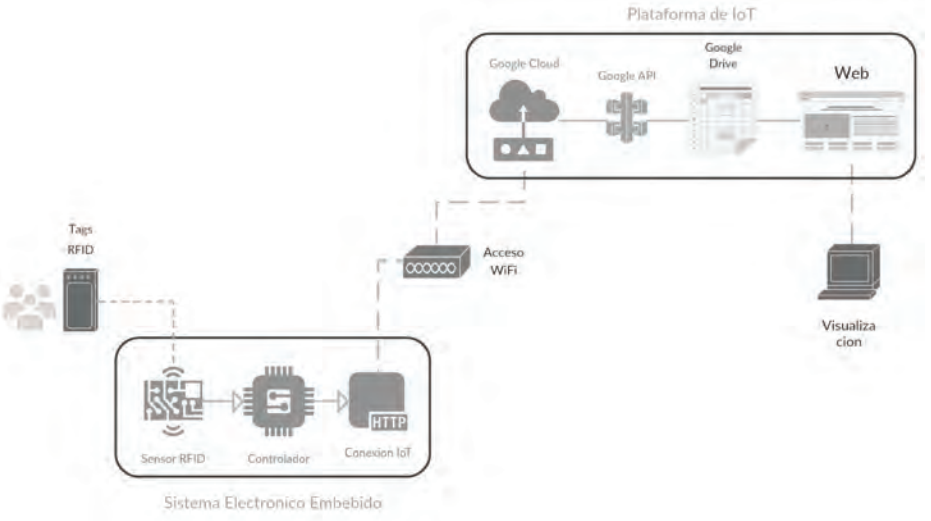
- Sensores: lectura de magnitudes físicas y conversión de señal.
- Procesamiento electrónico: normalmente implementado con un microcontrolador, que incluye memoria para almacenamiento de código ejecutable o *firmware*, además de una unidad de procesamiento central capaz de decodificar y controlar las acciones del sistema.
- Conectividad: dispositivo electrónico para la conexión con algún tipo de red alámbrica o inalámbrica para acceder al internet, por ejemplo, Wi-Fi, Bluetooth, GSM, banda ancha, etc.
- Plataforma IoT: servicio informático o nube en internet donde se almacenarán y procesará la información recibida desde el dispositivo electrónico.
- Aplicaciones/Visualización: se refiere a la etapa o página web donde se pondrán a disposición de un usurario, la información producida por los sensores y por el procesador electrónico en el otro extremo del sistema.

Figura 7. Etapas generales de un sistema IoT: Sensores/Electrónica, Red y Plataforma/Aplicaciones (Huawei, 2017)



A partir de lo anterior, se diseñó una arquitectura (figura 8) donde se pueden apreciar las etapas que se implementaron para el sistema IoT fruto de esta investigación aplicada. En cada etapa o bloque funcional se determinó el uso de componentes que cumplieran con ciertos aspectos basados en los objetivos de la investigación, como tecnologías recientes, vasta documentación técnica, bajo costo, bajo consumo eléctrico y que fueran accesibles en el mercado local.

Figura 8. Arquitectura diseñada para del sistema IoT por implementar.



Fuente propia.

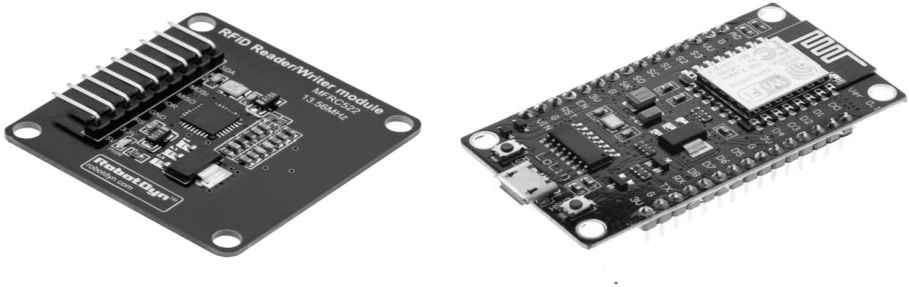
3.4.2 *Elección de componentes de* hardware

Tomando como base lo descrito anteriormente y la información recabada a partir del estado de la técnica del capítulo 2, se seleccionaron los siguientes componentes o herramientas tecnológicas para la implementación de cada etapa del sistema IoT propuesto:

- Sensores: se utilizará tecnología RFID, tarjetas y lectores electrónicos con interface digital, específicamente del modelo MFRC522 con frecuencia de trabajo 13.56 MHz (figura 9). Cabe destacar que a cada estudiante del grupo de prueba se le entregó una tarjeta, o *tag* RFID, la cual fue grabada previamente con el número de carné correspondiente único de cada estudiante.
- Procesamiento electrónico: se utilizó el microcontrolador ESP8266 junto con la placa de desarrollo NodeMCU, para la implementación de todo el procesamiento electrónico y de *firmware* que gobernará los sensores y el procesamiento de las señales producidas (ver figura 10).

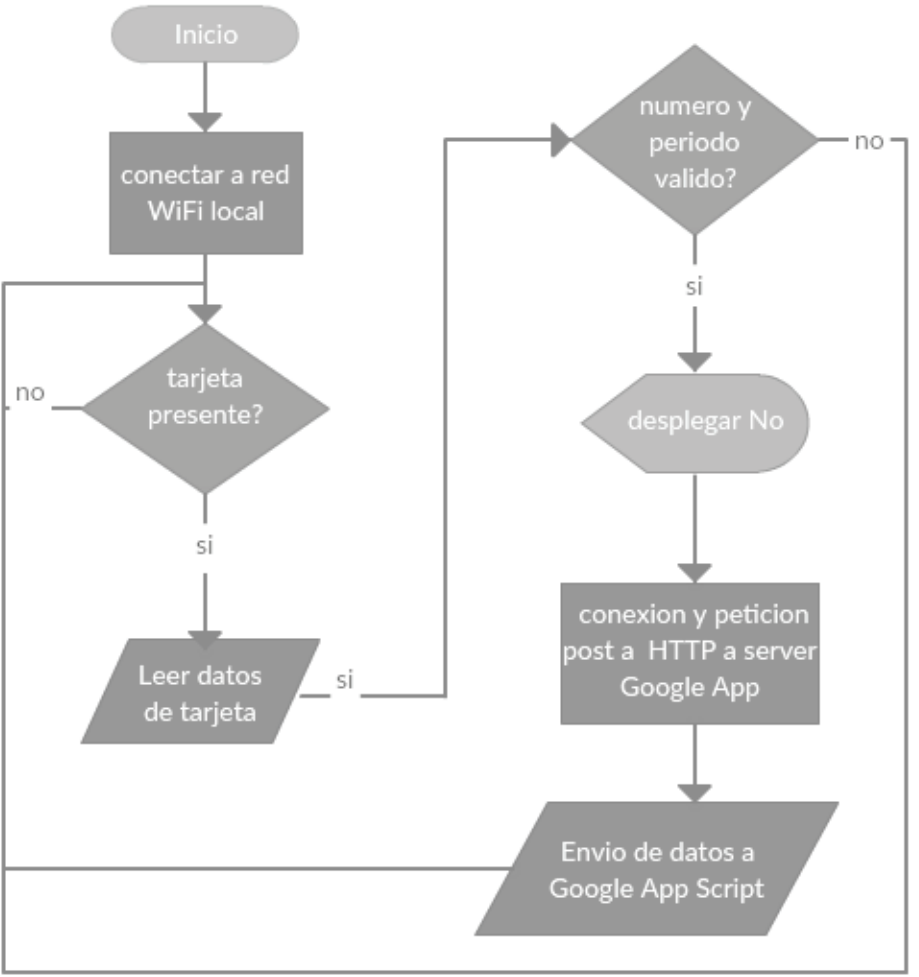
Figura 9. Grabador y tarjetas RFID por utilizar (Ebay, 2017)

Figura 10. Principales componentes electrónicos utilizados: (izq.) sensor RFID, (der.) tarteja NodeMCU. (Aliexpress, 2017)



- Conectividad: la placa NodeMCU, además del microcontrolador ESP8266, ya incluye un transceptor Wi-Fi capaz de funcionar como cliente y conectarse al internet por medio un *access point* a través de una red inalámbrica disponible en el sitio de implementación.

Figura 11. Diagrama de flujo implementado en el *firmware* del microcontrolador del sistema electrónico embebido.

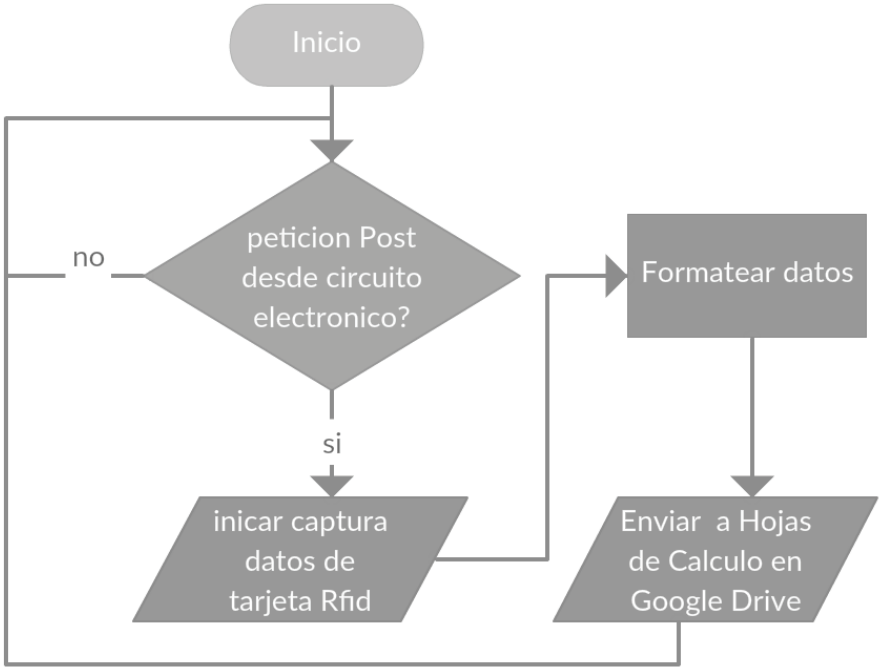


Fuente: imagen propia.

La etapa denominada *plataforma IoT* del sistema diseñado se basa en los servicios de Google, específicamente se utilizan el Google App Script y el hojas de cálculo de Google Drive. Primero se ha diseñado un código *script* que es almacenado y ejecutado en los servidores en la nube de

Google; es un programa en lenguaje de programación Java que se encarga de recibir, mediante protocolo HTPS, los datos de la tarjeta escaneada provenientes del circuito electrónico, que a su vez son enviados a una hoja de cálculo en Google Drive para su almacenamiento y visualización. En la figura 12 se muestra el diagrama de tareas diseñado para el *script*.

Figura 12. Diagrama de flujo para el *script* para la plataforma IoT de Google.



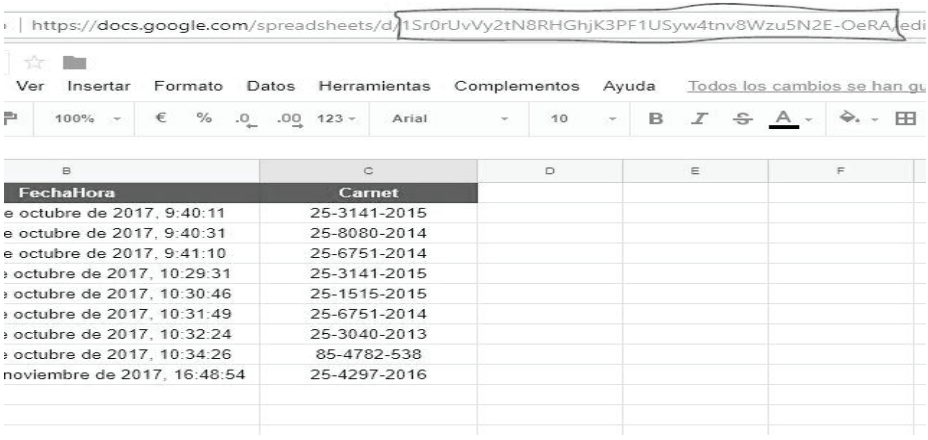
Fuente: imagen propia.

3.4.1.1 *Configuración plataforma* IoT

Como se mencionó, se debe diseñar un *script* para ser ejecutado en el servidor de IoT de Google que trabaje según lo descrito en el flujograma de la figura 13. Los pasos que han de seguirse, para la configuración de los servicios, son los siguientes:

- *Abrir una cuenta en los servicios de Google*. Cabe destacar que es un servicio gratuito; y con esto se tendrá acceso a muchas funciones que ofrece dicha compañía.
- *Crear una hoja de cálculo*. Dentro de las aplicaciones Google, se debe crear una hoja nueva donde se almacenarán los datos colectados por el sensor RFID dentro del circuito electrónico embebido en el aula. Se debe crear una hoja por cada circuito/aula que se monitoreará.
- *Obtener el identificador web de la hoja*. Dentro de la dirección URL de la hoja de cálculo, se debe copiar el identificador único de la hoja, que es la cadena de caracteres dentro de los separadores *d/* y */edit*, esta dirección será utilizada en el *script* a diseñar en la nube de Google.
- *Crear el* script. Este componente del sistema, es el código que realizará el enlace entre el circuito electrónico embebido y la hoja de cálculo en Google Drive, para programación del Script se accede al editor desde la misma hoja de cálculo creada, en el menú *Herramientas* y luego en la opción *Editor*, dentro de esta opción se encuentra el editor donde se deberá escribir el código del *script* (figura 14). El código que se ha de programar se puede ver en el anexo 2; se modifica la línea correspondiente con el identificador único de la hoja de cálculo por llenar, recordando que se deberá crear un *script* por hoja/aula que se debe monitorear. Luego de esto, ya se tiene configurado el lado IoT de nuestro sistema; y solo resta poner a prueba el sistema, construyendo e instalando el circuito electrónico, y escanear las tarjetas de los usuarios al ingresar al aula.

Figura 13. Obtención del identificador único de la hoja de cálculo.



Fuente propia.

Figura 14. Edición del *script* en el Google App.

Fuente propia.

4. *Resultados*

Los frutos de esta investigación son los siguientes:

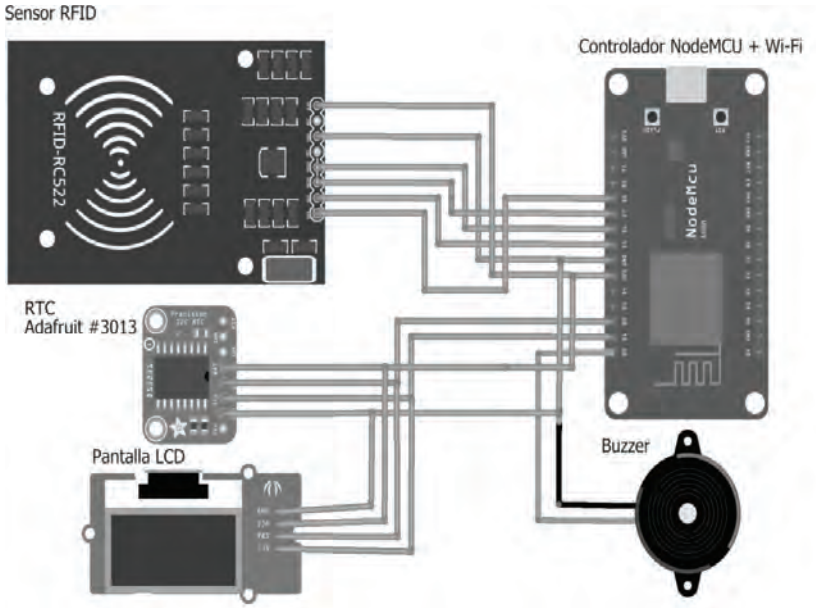
- El diseño de circuito electrónico junto con su *firmware* específico capaz de realizar la lectura del carné RFID de cada estudiante que ingresa al aula o recinto; y de enviar estos datos a internet vía conexión Wi-Fi.
- La creación de un código *script* dentro de los servicios de Google App Script, programa que corre en internet y que está pendiente de recibir la información generada por el circuito electrónico cada vez que un estudiante escanea su carné al entrar.
- Aplicación o página Web, junto con la configuración de hojas de cálculo de Google Drive para la visualización en tiempo real de la información del carné RFID escaneado.

4.1 *Circuito electrónico embebido*

Según la elección de elementos y herramientas descritas en la metodología de la investigación, se diseñó un circuito electrónico de conexión para el sistema embebido encargado de capturar la información dentro del carné RFID escaneado. Como se mencionó, el circuito se basa en la plataforma de desarrollo NodeMCU y en el microcontrolador ESP8266, lo cual permite un diseño minimalista pero eficiente técnicamente, esto gracias a las diversas características descritas en capítulos anteriores, como su inclusión, en la misma plataforma, de un chip para el manejo del acceso a internet vía conexión Wi-Fi. En la figura 15 se puede apreciar el circuito diseñado para la etapa de captura de datos, procesamiento y envió de datos al internet.

El microcontrolador a cargo de todos los procesos internos del circuito electrónico fue programado usando un *firmware* escrito para la función específica requerida por el sistema, dicho programa se escribió usando lenguaje de programación C y basado en el algoritmo básico pero eficaz de tres funciones: captura, procesamiento, conexión y envío a Google App Script, esto cada vez que se presenta una nueva tarjeta RFID.

Figura 15. Circuito electrónico diseñado para la captura, procesamiento y conexión vía Wi-Fi del sistema IoT.



Fuente propia.

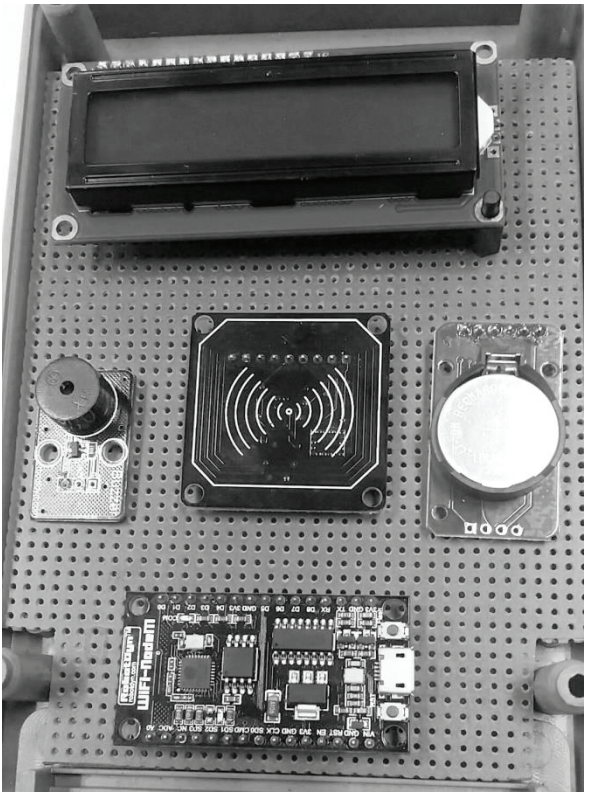
Las funciones principales del circuito electrónico embebido son: lectura de numero único identificador de la tarjeta escaneada, procesamiento de este número para determinar el carné del estudiante, validación de información correcta en tiempo y formato, conexión a internet vía Wi-Fi y envió al *script* en la nube de Google. La construcción del circuito fue realizada con éxito en una tarjeta de circuito impreso diseñada para el caso (ver figura 16); y para su instalación final se usó una carcasa plástica (ver figura 17).

4.2 *Aplicación* IoT *y visualización de datos*

Para la etapa del *firmware* que se debe ejecutar en el microcontrolador, se diseñó el código fuente, que se puede apreciar en el anexo 1; y para el *software* que se ejecutará en la nube de Google para la conexión entre el circuito y la etapa de presentación, se diseñó un *script* cuyo código

puede verse en el anexo 2. En la etapa de visualización, se utilizó la herramienta de Google Sites para montar un sitio web para acceso desde un navegador con https://sites.google.com/a/mail.utec.edu.sv/labeleutec/bj301

Figura 16. Implementación en circuito impreso del circuito electrónico embebido

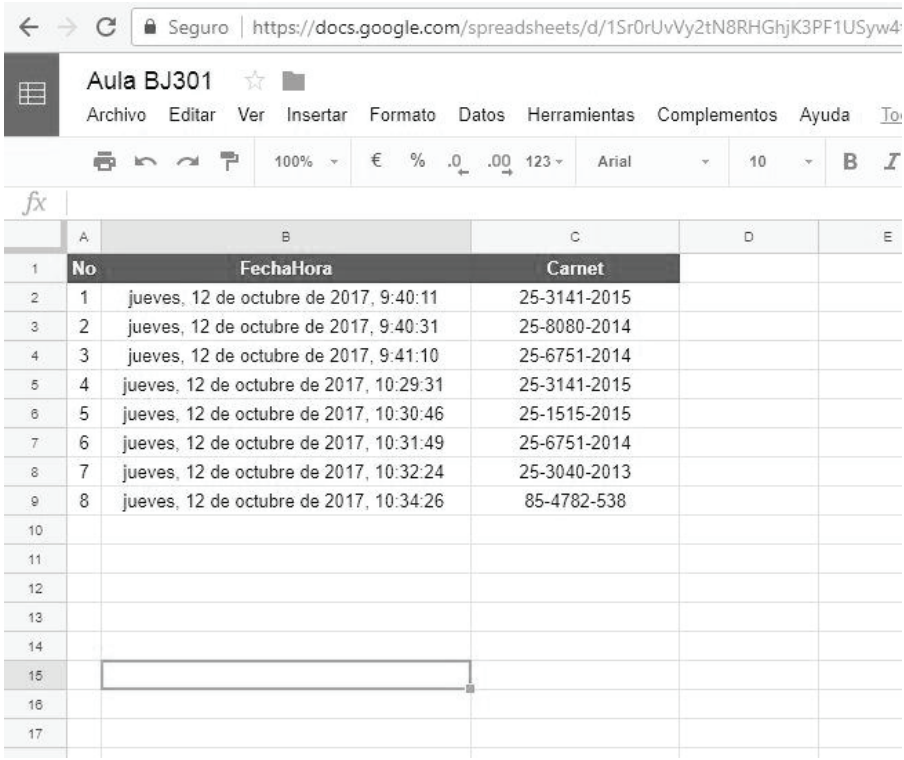


Fuente propia.

Figura 17. Vista del circuito electrónico en carcaza plástica

Fuente propia.

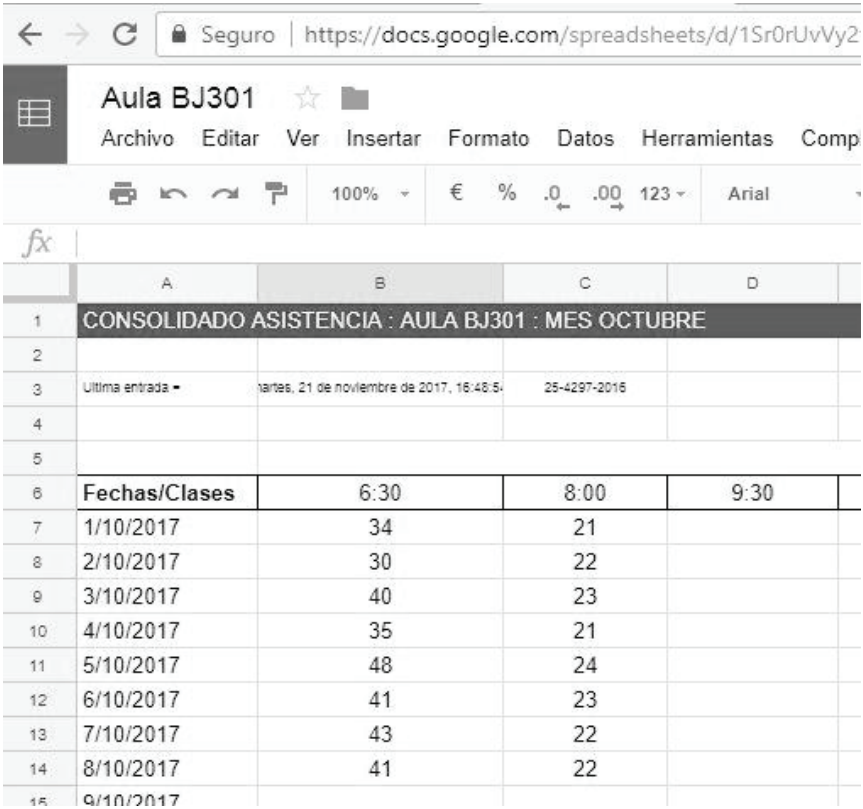
Figura 18. Captura de pantalla de la visualización de los datos de un aula en hoja de cálculo de Google Drive



	A	B	C	D	E
1	No	FechaHora	Carnet		
2	1	jueves, 12 de octubre de 2017, 9:40:11	25-3141-2015		
3	2	jueves, 12 de octubre de 2017, 9:40:31	25-8080-2014		
4	3	jueves, 12 de octubre de 2017, 9:41:10	25-6751-2014		
5	4	jueves, 12 de octubre de 2017, 10:29:31	25-3141-2015		
6	5	jueves, 12 de octubre de 2017, 10:30:46	25-1515-2015		
7	6	jueves, 12 de octubre de 2017, 10:31:49	25-6751-2014		
8	7	jueves, 12 de octubre de 2017, 10:32:24	25-3040-2013		
9	8	jueves, 12 de octubre de 2017, 10:34:26	85-4782-538		
10					
11					
12					
13					
14					
15					
16					
17					

Fuente propia

Figura 19. Captura de la hoja resume para un aula específica



	A	B	C	D
1	CONSOLIDADO ASISTENCIA : AULA BJ301 : MES OCTUBRE			
2				
3	Ultima entrada =	hartes, 21 de noviembre de 2017, 16:48:54	25-4297-2016	
4				
5				
6	**Fechas/Clases**	6:30	8:00	9:30
7	1/10/2017	34	21	
8	2/10/2017	30	22	
9	3/10/2017	40	23	
10	4/10/2017	35	21	
11	5/10/2017	48	24	
12	6/10/2017	41	23	
13	7/10/2017	43	22	
14	8/10/2017	41	22	
15	9/10/2017			

Fuente propia.

Figura 20. Captura de pantalla del sitio web implementado

No	FechaHora	Carnet
1	jueves, 12 de octubre de 2017, 9:40:11	25-3141-2015
2	jueves, 12 de octubre de 2017, 9:40:31	25-8080-2014
3	jueves, 12 de octubre de 2017, 9:41:10	25-6751-2014
4	jueves, 12 de octubre de 2017, 10:29:31	25-3141-2015
5	jueves, 12 de octubre de 2017, 10:30:46	25-1515-2015
6	jueves, 12 de octubre de 2017, 10:31:49	25-6751-2014
7	jueves, 12 de octubre de 2017, 10:32:24	25-3040-2013
8	jueves, 12 de octubre de 2017, 10:34:26	85-4782-538

(Fuente propia)

5. Conclusiones

A modo de conclusión global del trabajo realizado, el sistema de IoT diseñado para el registro automatizado de las personas que ingresan a un aula cumple con el objetivo general planteado; y se convierte en una herramienta tecnológica de bajo costo que sirve de apoyo en las labores logísticas de una institución donde controlar el número de asistentes sea una tarea periódica y muy importante dentro de su quehacer administrativo.

Específicamente, a partir lo de realizado con cada una de las labores en esta investigación aplicada, se pueden formular las conclusiones siguientes:

- Con esta investigación se ha podido aportar nuevo conocimiento científico, de manera que se ha mostrado nuevas formas de utilizar un sistema de IoT, como en la solución de problemas de automatización, en el monitoreo y control remoto de procesos

con herramientas tecnológicas recientes de bajo costo y eficientes, tales como el microcontrolador ESP8266 y la plataforma Google, asequibles en el entorno local.

- El uso de los componentes mencionados permitió el diseño y construcción de un circuito electrónico embebido eficiente y de bajo costo, que cumple con la función de permitir el escaneo de una tarjeta RFID, leer su información interna única, decodificarla y enviarla vía Wi-Fi a internet. Este circuito además es de fácil configuración y reproducción para una producción e implementación masiva, por ejemplo, en un campus universitario.
- Para el *software* producido, se diseñaron dos soportes: un *firmware* para el microcontrolador ESP8266 y un *script* para ser almacenado y ejecutado en internet dentro de los servicios de la plataforma Google. El uso de *software* de desarrollo de licenciamiento libre y fuente abierta Arduino C y el Google App Script permitió el cómodo desarrollo de los códigos completamente funcionales en ambos casos, lo cual demuestra que se pueden realizar desarrollos de *software* con herramientas disponibles sin costo alguno.
- A manera de validación de lo anterior, se implementó el prototipo de sistema IoT construido en un aula de la Utec, cuyo funcionamiento fue óptimo y logró las expectativas estimadas: el monitoreo de forma inmediata, desde cualquier punto con acceso a internet, de la información del alumno que ingresa a esa aula.

Como conclusión, se ha de mencionar que este no es el final de la investigación aplicada en el área de internet de las cosas, hay muchas líneas y aplicaciones que se deben desarrollar y un vasto campo de investigación por delante.

6. Recomendaciones

El equipo de trabajo recomienda continuar con el apoyo a investigaciones de este tipo, ya que es un campo poco explorado dentro del entorno académico, comercial e industrial del país, tomando como base que el IoT es un área que se vislumbra que tiene mucho potencial dentro de nuestro ámbito.

El equipo de trabajo propone que se tome como base el conocimiento científico técnico aportado por este trabajo, en el desarrollo de sistemas IoT como el planteado, en instituciones similares a la Utec. Específicamente se recomienda trabajar de la mano con el Ministerio de Educación; con posibles alianzas en sus centros educativos a escala nacional, para la puesta en marcha de aplicaciones como fruto de esta investigación.

7. Referencias

Actum (2017). *"Actum"*. Recuperado de https://www.actum.es/preguntas-frecuentes/%C2%BFqu%C3%A9-significa-rfid

Aisemberg, D. (2010, septiembre 29). "Tecnozona". Retrieved from http://www.tecnozona.com/zona_de_los_que_opinan/el-abc-de-rfid-las-etiquetas-del-futuro-2/

Albanian Times (2017, Agosto). *"Global RFID System Market 2017"*. Retrieved from http://www.albaniantimes.com/2017/08/02/global-rfid-system-market-2017-2022/

Aliexpress. (2017, Abril 05). *Aliexpress*. Retrieved from https://www.aliexpress.com/

Arduino (2017). "Arduino.cc". Retrieved from https://store.arduino.cc/usa/mkr-gsm-1400

Batem, J.; Cortés, C.; Cruz, P., & Paz Penagos, H. (2009). "Diseño de un protocolo de identificación por radiofrecuencia (RFID) propietario para una aplicación específica". *Escuela Colombiana de Ingeniería Julio Garavito*, 329-330.

Bordas, F. (2017, abril). *"Google Apps Scripts"*. Retrieved from http://googleappscriptsweb.blogspot.com/2015/02/que-es-google-apps-script.html

Carriots (2017). *"Carriots.com"*. Retrieved from https://www.carriots.com/que-es-carriots

Cayssials, R. (2014). *Sistemas embebidos en FPGA.* Buenos Aires: Alfaomega Grupo Editor.

Cisco (2011, abril 15). Retrieved from https://www.cisco.com/c/dam/global/es_mx/solutions/executive/assets/pdf/internet-of-things-iot-ibsg.pdf

Cobo, J.G. (2017, septiembre 27). "Hardware Libre". Retrieved from https://www.hwlibre.com/arduino-mkr-wan-1300-arduino-mkr-gsm-1400-las-nuevas-placas-iot-del-proyecto-arduino/

Dave Evans (2011). *Internet de las cosas.* Cisco IBSG.

Ebay (2017). Retrieved from www.ebay.com

Educa, E. (2014). *Endesa Educa*. Retrieved from http://www.endesaeduca.com/Endesa_educa/recursos-interactivos/smart-city/

Ejje (2017). Retrieved from http://www.ejje.com/hardware/

Galeano, G. (2013). *Programación de sistemas embebidos en C: teoría y práctica aplicada a cualquier microcontrolador.* México: Alfaomega Grupo Editor.

Gartner (2017). *"Internet of Things"*. Retrieved from https://www.gartner.com/technology/research/internet-of-things/

Geek Factory (2017). "Geek Factory". Retrieved from https://www.geekfactory.mx/tienda/radiofrecuencia/nodemcu-esp8266-tarjeta-wifi/

Goasduff, L. (2016, mayo 15). *"gartner"*. Recuperado en Septiembre 02, 2017, desde https://grupogaratu.com/que-es-y-que-aporta-la-industria-4-0/

González, A.L. (2016). "Hardware de Intenet de las Cosas". *Altrantech360*.

Google Cloud, P. (2017, octubre 19). *"Google Cloud Platform"*. Retrieved from https://cloud.google.com/terms/?hl=es

Huawei (2017). *Hacia la creación de un mundo IoT fiable y gestionado.* Madrid: INCIBE.

Intel (2017). "Intel". Retrieved from https://www.intel.la/content/www/xl/es/internet-of-things/industry-solutions.html

Torres (2014, octubre 20). "Hipertextual". Retrieved from https://hipertextual.com/archivo/2014/10/gramofon-actualizacion/

Lajara Vizcaíno, J.R., & Pelegrí Sebastiá, J. (2015). *Sistemas integrados con Arduino.* México: Alfaomega grupo editor.

Martín, G. (2017). "Programarfacil.com". Retrieved from https://programarfacil.com/esp8266/proyectos-con-esp8266-iot/

Mclelland, C. (2016). *"Blog Leverege"*. Retrieved from https://www.leverege.com/blogpost/what-is-an-iot-platform

Mclelland, C. (2016). *"Leverege"*. Retrieved from https://www.leverege.com/blogpost/how-to-choose-the-best-iot-platform

Opertek (2017, abril 6). "Opertek.com". Retrieved from https://www.opertek.com/plataforma-predix/

Osswald, M. (2014, febrero 6). *"Hanson Inc"*. Recuperado Agosto 10, 2015, desde http://www.hansoninc.com

RS (2017). Retrieved from http://es.rs-online.com/web/generalDisplay.html?id=i/iot-internet-of-things

RS (n.d.). *RS*. Retrieved from http://es.rs-online.com/web/generalDisplay.html?id=i/iot-internet-of-things

Shutterstock (2016, agosto 12). "Infobae Tendencias". Retrieved from https://www.infobae.com/tendencias/2016/08/12/internet-de-las-cosas-la-revolucion-tecnologica-que-cambiara-al-mundo/

Singh, S. (2017, marzo 8). *"Top 20 IoT Platforms in 2018"*. Retrieved from https://internetofthingswiki.com/top-20-iot-platforms/634/

Temboo, I. (2016). *"Temboo"*. Retrieved from https://temboo.com/

Tendencias (2017). "Internet de las cosas". *ComputerWorld*.

Tollervey, N. (2017). *Programming with micropython.* California: O'Reilly Media.

Toro, L.M. (n.d.). *SISTEMAS DE IDENTIFICACIÓN POR RADIOFRECUENCIA.*

tutorialspoint (2017). *"tutorialspoint"*. Retrieved from https://www.tutorialspoint.com/internet_of_things/internet_of_things_hardware.htm

Ubidots (2014). *"Ubidots.com"*. Retrieved from https://ubidots.com/docs/es/get_started/introduccion.html

Velosa, A. (2016, Julio 21). *"Market Guide for IoT Platforms"*. Recuperado 05 Septiembre, 2017, from https://www.gartner.com/doc/3380746/market-guide-iot-platforms

Volt, C. (2016, noviembre 12). *"RogerBit.com"*. Retrieved from http://rogerbit.com/wprb/index.php/2016/11/12/latinoamerica-puede-ser-lider-en-internet-de-las-cosas-y-vencer-desigualdad/

Vuksanović, D. (2017). Industry 4.0: the future concepts and new visions of factory of the future development. *Advanced engineering systems*, 293-298.

Zona Maker (2016). *Zona Maker*. Recuperado desde https://www.zonamaker.com/raspberry/intro-raspberry

8. Anexos

Anexo 1. Código microcontrolador

```
/*************************************************************/
/* ************************************************
 * *** ***
 * Sistema RFID para monitoreo asistencia–
 * Firmware para el NodeMCU y sensor RFID
 *
 USO:
 - En cada aula / alumno debe acercar su tarjeta al entrar

 * *** ***
 * ******************************************************
 *
 * ************************************************
 * * Original por: *
 * * Autor: Omar Otoniel FLores *
 * * Mail: otoniel.flores@mail.utec.edu.sv *
 * * Licencia: GNU General Public License v3 or later *
 * ************************************************
 */
/*************************************************************/
/* datos de acceso Hoja de Datos BJ301 en Google Drive
 * spreadsheet's unique sharing key = 1Sr0rUvVy2tN8RHGhjK3PF1USyw4tnv8Wzu5N
2E-OeRA
 * nombres de las hojas: Resumen Registro
 * URL de pla aplicacion web del scrip = https://script.google.com/macros/s/AKfycbyfb_
UxP0N4kvQfFVLfnxgbWAj0PbFjmBbCKSL5vGY-T2YfpTo/exec
 */

/*************************************************************
 * Librerias por utilizar *
 *************************************************************/
/* acceso WiFi */
#include <ESP8266WiFi.h>
#include <WiFiClient.h>
/* acceso LCD I2C */
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
//LiquidCrystal_I2C lcd(0x3F, 16, 2); //verde
LiquidCrystal_I2C lcd(0x27, 16, 2); //axul
/* acceso RTC ds3231 */
#include <TimeLib.h>
#include <RtcDS3231.h> //https://github.com/Makuna/Rtc
RtcDS3231<TwoWire> Rtc(Wire); //conectado a puerto I2C
/* acceso Google Drive */
```

```
#include “HTTPSRedirect.h”
/* acceso sensor RFID */
#include <SPI.h>
#include <RFID.h>
/* conexion RFID con NodeMCU */
// MFRC522 ---- NodeMCU
// VCC----5V/3.3
// GND----GND
// RST----N.C.
// SDA----D8
// MOSI---D7
// MISO---D6
// SCK----D5

/*****************************************************************
 * Definicion de Objetos, Constantes y Variables *
 *****************************************************************/
/* credenciales acceso WiFi */
const char WIFI_SSID[] = “testIOT”; //”AsuncionLab”;
const char WIFI_PSK[] = “123456789”; //”labeleutec8992”;
//const char WIFI_SSID[] = “UTEC”;
//const char WIFI_PSK[] = “”;
//WiFiClient client;
/* credenciales acceso Google Drive */
// The ID below comes from Google Sheets.
// Towards the bottom of this page, it will explain how this can be obtained
const char* GScriptId = “AKfycbyfb_UxP0N4kvQfFVLfnxgbWAj0PbFjmBbCKSL5vGY-
T2YfpTo”;
const char* host = “script.google.com”;
const char* googleRedirHost = “script.googleusercontent.com”;
const int httpsPort = 443;
HTTPSRedirect client(httpsPort);
// Prepare the url (without the varying data)
String url = String(“/macros/s/”) + GScriptId + “/exec?”;
//sonido Buzzer
int frequency=2000; //Specified in Hz
int buzzPin=D0; //GPIO16
int timeOn=350; //specified in milliseconds
int timeOff=350; //specified in millisecods
//a utilizar en la lectura RFID
bool flag;
/* variables */
//manejo del RTC
byte actualHour , actualMinute , actualsecond ;
int actualyear ;
byte actualMonth , actualday , actualdayofWeek ;
char dateString[11];
char timeString[9];
String dia;
```

```
#define countof(a) (sizeof(a) / sizeof(a[0]))
const int timeZone = -6;
// Remote site information
const char http_site[] = "time.sodaq.net"; // sin HTTP://
const int http_port = 80;
long epocchF;
String c = "";
//manejo RFID
RFID rfid(15, 20); // objeto de la clase
unsigned int serNum[4];
unsigned int nuidPICC[4];

/*************************************************************
 * Setup *
 *************************************************************/
void setup()
{
 //**apagar led builtin
 pinMode(LED_BUILTIN, OUTPUT);
 digitalWrite(LED_BUILTIN, HIGH);
 //***rtc
 Rtc.Begin();
 Rtc.Enable32kHzPin(false);
 Rtc.SetSquareWavePin(DS3231SquareWavePin_ModeNone);
 //***lcd
 lcd.begin();
 lcd.clear();
 lcd.backlight();
 //analogWrite(10, 250);
 lcd.home();
 lcd.print("Iniciando...");
 Serial.begin(9600);
 //***conexion Wifi
 conWiFi();
 //***sincronizarRTCdesdeWEB
//sincroRTC();
 //**modulo RFID
 SPI.begin();
 rfid.init();
 lcd.clear();
}
/*************************************************************
 * Programa Principal LOOP *
 *************************************************************/

void loop()
{
 if (rfid.isCard()){leerTag();} //si se detecta una tarjeta frente al sensor.
 mostrarReloj();
```

```
}

/*** Funciones ***/

/*****************************************************/
/*** Leer el Tag y subir la info ***/
/*****************************************************/
void leerTag()
{

 rfid.readCardSerial();
 if (rfid.serNum[0] != nuidPICC[0] || rfid.serNum[1] != nuidPICC[1] || rfid.serNum[2]
!= nuidPICC[2] || rfid.serNum[3] != nuidPICC[3] )
 {
 for (byte i = 0; i < 4; i++)
 {
 nuidPICC[i] = rfid.serNum[i];
 }
 unsigned long carnet = (rfid.serNum[0]*1) + (rfid.serNum[1]* 256) + (rfid.serNum[2]*
65536) + (rfid.serNum[3]* 16777216);
 String Strin_carnet = String(carnet);
 String carne = Strin_carnet.substring(0, 2) + "-" + Strin_carnet.substring(2, 6) + "-" +
Strin_carnet.substring(6);
 //Serial.println(carne);
 lcd.clear();
 lcd.home();
 lcd.print("...Carnet No...");
 lcd.setCursor(2,1);
 lcd.print(carne);
 tone(buzzPin, frequency);
 delay(timeOn);
 noTone(buzzPin);
 digitalWrite(D0,1);
 delay(timeOff);

 // ******* //
 // postear en Google Spreedsheets
 flag = false;
 for (int i=0; i<5; i++){
 int retval = client.connect(host, httpsPort);
 if (retval == 1) {
 flag = true;
 break;
 }
 else
 Serial.println("Fallo en conexion…reintentando");
 }
 if (!client.connected()){
 Serial.println("Conectando…");
```

```
client.connect(host, httpsPort);
}
String urlFinal = url + "carnet=" + carne;
client.printRedir(urlFinal, host, googleRedirHost);
lcd.clear(); lcd.print("Guardando..."); lcd.setCursor(0,1);
for (int x=0; x<6; x++)
{
lcd.print(".");
delay(200);
}
lcd.clear();lcd.print ("***registrado***");
delay (1000);
lcd.clear();
//fue posteado
// ************************* //

}else{
lcd.clear();
lcd.print(" !Ya Registrada! ");
tone(buzzPin, 100);
delay(timeOn);
noTone(buzzPin);digitalWrite(D0,1);
delay(timeOff);
}
digitalWrite(D0,1);
for (byte i = 0; i < 4; i++)
{
Serial.print (rfid.serNum[i]);
}
Serial.println("");
rfid.halt();
delay(500);
lcd.clear();

}
/*****************************************************/
/*****************************************************/

/*****************************************************/
/*** Muestra el Reloj en la LCD ***/
/*****************************************************/
void mostrarReloj()
{
// acceder al RTC
RtcDateTime now = Rtc.GetDateTime();
switch (now.DayOfWeek())
{
case 0:
dia = "DOM";
```

```
 break;
 case 1:
 dia = “LUN”;
 break;
 case 2:
 dia = “MAR”;
 break;
 case 3:
 dia = “MIE”;
 break;
 case 4:
 dia = “JUE”;
 break;
 case 5:
 dia = “VIE”;
 break;
 case 6:
 dia = “SAB”;
 break;
 default:
 break;
 }
  snprintf_P(dateString, countof(dateString), PSTR(“%02u/%02u/%04u”), now.Day(), now.Month(), now.Year() );
  snprintf_P(timeString, countof(timeString), PSTR(“%02u:%02u:%02u”), now.Hour(), now.Minute(), now.Second() );

 //lcd.clear();
 //lcd.home();
 lcd.setCursor(0,0);
 lcd.print(“BJ301 “);
//lcd.print(“xxxxxxxx”);
 lcd.println(timeString);
 lcd.setCursor(0,1);
 lcd.print(dia);
 lcd.setCursor(6,1);
 lcd.print(dateString);
 delay(333);
}
/*****************************************************/
/*****************************************************/
 //mostrar en pantalla
 //actualizar via Web / cada dia 12 noche

/*****************************************************/
/*** SincronizarRTC desde sodaq.net WEB ***/
/*****************************************************/
void sincroRTC()
{
```

```
 //peticion conectar con web page
 if ( !getPage() )
 {
 Serial.println("fallo, reinicie!");
 }
 Serial.println("....");
 // se recibieron datos...
 while (client.available()== 0) {}
 c = client.readString();

 c.remove(0, 175);
 c.trim();
 //Serial.println("");
 //Serial.println(epocchF);
 epocchF = c.toInt();
 time_t t = epocchF + timeZone * SECS_PER_HOUR;
 // en este estructura t queda la hora
 // digital clock display of the time
 //Serial.println(hour(t));
 //Serial.print(day(t));Serial.print(month(t)); Serial.print(year(t));
 RtcDateTime currentTime = RtcDateTime(year(t),month(t),day(t),hour(t),minute(t),sec
ond(t)); //define date and time object
 Rtc.SetDateTime(currentTime); //configure the RTC with object
 lcd.clear();
}
// funcion para realizar peticion web
bool getPage() {

 // Attempt to make a connection to the remote server
 if ( !client.connect(http_site, http_port) ) {
 return false;
 }

 // Make an HTTP GET request
 client.println("GET /index.html HTTP/1.1");
 client.print("Host: ");
 client.println(http_site);
 client.println("Connection: close");
 client.println();

 return true;
}
/*************************************************/
/*** Conectarse red WiFi***/
/*************************************************/

// Attempt to connect to WiFi
void conWiFi()
{
```

```
lcd.clear(); lcd.home();
lcd.print("Buscando WiFi");
lcd.setCursor(0,1);
// Set WiFi mode to station (client)
WiFi.mode(WIFI_STA);
// Initiate connection with SSID and PSK
WiFi.begin(WIFI_SSID, WIFI_PSK);
// Blink LED while we wait for WiFi connection
while ( WiFi.status() != WL_CONNECTED ) {
lcd.print (".");
delay(100);
}
lcd.clear(); lcd.home();
lcd.print("Conectado !!!");
lcd.setCursor(0,1); lcd.print(WIFI_SSID);
delay(500);

}
/*****************************************************/
/*** Fin del Código ***/
/*****************************************************/
```

Anexo 2. Código Script Google App

```
// Modify by: Omar Otoniel Flores
// based on work: Akshaya Niraula
// ON: 2017.


// This method will be called first or hits first
function doGet(e){
 Logger.log("--- doGet ---");

 var carnet = "";
 //value = "";

 try {

 // this helps during debuggin
 if (e == null){e={}; e.parameters = {carnet:"test"};}

 carnet = e.parameters.carnet;
 //value = e.parameters.value;

 // save the data to spreadsheet
 save_data(carnet);


 return ContentService.createTextOutput("Wrote:\n carnet: " + carnet );
```

```
 } catch(error) {
 Logger.log(error);
 return ContentService.createTextOutput("oops...." + error.message
 + "\n" + new Date()
 + "\ncarnet: " + carnet);
 }
}

// Method to save given data to a sheet
function save_data(carnet){
 Logger.log("--- save_data ---");


 try {
 var dateTime = new Date();
 //var timeTime = dateTime.toLocaleTimeString();


 // Paste the URL of the Google Sheets starting from https thru /edit
 // For e.g.: https://docs.google.com/..../edit
   //var ss = SpreadsheetApp.openByUrl("https://docs.google.com/spreadsheets/d/---
Your-Google-Sheet-ID--Goes-Here---/edit");
 var ss = SpreadsheetApp.openByUrl("https://docs.google.com/spreadsheets/d/1Sr0rUv
Vy2tN8RHGhjK3PF1USyw4tnv8Wzu5N2E-OeRA/edit");
 var summarySheet = ss.getSheetByName("Resumen");
 var dataLoggerSheet = ss.getSheetByName("Registros");


 // Get last edited row from DataLogger sheet
 var row = dataLoggerSheet.getLastRow() + 1;


 // Start Populating the data
 dataLoggerSheet.getRange("A" + row).setValue(row -1); // ID
 dataLoggerSheet.getRange("B" + row).setValue(dateTime); // marca de fecha
 //dataLoggerSheet.getRange("C" + row).setValue(timeTime); // marca de tiempo
 dataLoggerSheet.getRange("C" + row).setValue(carnet); // carnet


 // Update summary sheet
 summarySheet.getRange("B3").setValue(dateTime); // Last modified date
 summarySheet.getRange("C3").setValue(carnet); // Count
 }

 catch(error) {
 Logger.log(JSON.stringify(error));
 }

 Logger.log("--- save_data end---");
}
```

Breve hoja de vida de los investigadores

Ronny Adalberto Cortez. Actualmente es investigador a tiempo completo en el área de Tecnología, y asistente de docente en el área de redes en la Universidad Tecnológica de El Salvador en donde también cursó la carrera de Ingeniería en Sistemas y Computación, la cual completó con un CUM de 9.0. Mediante una beca obtenida con el programa Eureka SD, cursó el Máster Universitario Oficial en Ciencia de Datos e Ingeniería de Computadores con Especialidad en Ciencia de Datos y Tecnologías Inteligentes en la Universidad de Granada.

También obtuvo una beca con el programa Erasmus Student Exchange Program para estudiar en Mondragon Unibertsitatea en el País Vasco, en donde trabajó en el proyecto *"Comparison of classification solutions in the field of technology watch for automatic content categorization"*, cuyo objetivo fue identificar y analizar alternativas que proporcionan las tecnológicas de inteligencia artificial y lenguaje natural para la clasificación o categorización de contenido en formato de texto en el campo de la vigilancia tecnológica, formando las bases para la especialización en el análisis de textos para la extracción de información. Los temas estudiados incluyen los relacionados con inteligencias artificiales orientadas a la minería de datos, procesamiento, clasificación y agrupación de información y análisis de datos no numéricos.

Omar Otoniel Flores Cortez. Investigador y docente titular del departamento de Electrónica de la Universidad Tecnológica de El Salvador (Utec), en las áreas de Programación de plataformas para sistemas embebidos, robótica y domótica educativa. Enfocando actualmente en investigaciones aplicadas en el área de internet de las cosas y sus aplicaciones dentro y fuera del aula, además de redactar artículos para publicaciones acerca de estos temas.

Ingeniero electricista (Universidad de El Salvador). Maestría en Docencia. Postgrado en Robótica (UNIR/2015). Posgrado en Campos Virtuales para la Práctica Educativa (Utec/2015). Summer School: IoT for developed countries (ICTP, Italia/2017). Actualmente es estudiante de doctorado en Informática (U. de Alicante). Ha realizado el Curso en

Investigación Científica (ASI/2016). Participó en el congreso Concapan IEEE/Nicaragua 2017; ha publicado el libro texto *Aprende Arduino* (ISBN 978-99961-0-346-9); la investigación "Aplicación IoT monitoreo panel solar" (ISBN 978-99961-48-62-0); y el artículo *"Blue energy study in Central America - Journal of Renewable and Sustainable Energy"* (ISSN -1941-7012).

Es colaborador investigador de la investigación "Energia Azul" / Universidad de Granada - Utec - UCA (Estadía Doctoral - María del Mar Fernández). Ha recibido subsidios para la investigación "Aplicación IoT monitoreo panel solar" por la Utec.

Verónica Idalia Rosa de Rivera. Candidata a doctora en Informática de la Universidad de Alicante, España. Máster en *Visual Analytics* y *Big Data* (2014) de la Universidad de La Rioja, España. Maestría en Docencia Universitaria (2009) e Ingeniería en Sistemas y Computación (2001) de la Universidad Tecnológica de El Salvador (Utec). Participó en el Taller: Seguridad Informática, WALC Utec (2017). Participó en el Premio de Investigación Científica y/o Tecnología en Educación Superior y Centros de Investigación 2017, Modalidad Póster. Participó como ponente en el Congreso de Investigación "Auprides 2016: Universidad, Empresa y Estado para la innovación y el Desarrollo Sostenible". Ha sido reconocida como ponente invitada en el marco de la celebración del "Día del Estudiante de Informática UGB-2016". Tiene diploma de participación en el taller Introducción a la dirección de proyectos de investigación (2016). Tienen diplomado en Android, recibió el taller Formación de Tutores (Sensibilización). Tiene diplomado en Gestión Educativa.

Ha recibido las siguientes distinciones: placa de reconocimiento por haber sido seleccionada la mejor estudiante egresada de la carrera de Ingeniería en Sistemas y Computación, otorgada por la Asociación Salvadoreña de Ingenieros y Arquitectos (Asia) en julio 19 de 2000; "Docente Distinguida" Utec 2003 y 2017; "Docente Investigadora" Utec 2016, 2017 y 2018.

Colección Investigaciones 2003-2018

Publicación	Nombre	ISBN
2003	Historia de la Economía de la Provincia del Salvador desde el siglo XVI hasta nuestros días. Primer Tomo Siglo XVI Jorge Barraza Ibarra	99923-21-12-1 (v 1) 99923-21-11-3 (Edición completa)
Diciembre 2003	Recopilaciones Investigativas. Tomos I, II y III	SIN ISBN
2004	Historia de la Economía de la Provincia del Salvador desde el siglo XVI hasta nuestros días. Segundo Tomo Siglos XVII y XVIII Jorge Barraza Ibarra	99923-21-14-8 (v 2) 99923-21-11-3 (Edición completa)
2004	Historia de la Economía de la Provincia del Salvador desde el siglo XVI hasta nuestros días. Tercer Tomo Siglo XIX Jorge Barraza Ibarra	99923-21-15-6 (v 3) 99923-21-11-3 (Edición completa)
2005	Historia de la Economía de la Provincia del Salvador desde el siglo XVI hasta nuestros días. Cuarto Tomo Siglo XIX Jorge Barraza Ibarra	99923-21-31-8 99923-21-11-3 (Edición completa)
2006	Historia de la Economía de la Provincia del Salvador desde el siglo XVI hasta nuestros días. Quinto Tomo Siglo XX Jorge Barraza Ibarra	99923-21-39-3 (v 5) 99923-21-11-3 (Edición completa)
2009	Recopilación Investigativa. Tomo I	978-99923-21-50-8 (v1)
2009	Recopilación Investigativa. Tomo II	978-99923-21-51-5 (v2)
2009	Recopilación Investigativa. Tomo III	978-99923-21-52-2 (v3)
Enero 2010	Casa Blanca Chalchuapa, El Salvador. Excavación en la trinchera 4N. Nobuyuki Ito	978-99923-21-58-4
Marzo 2010	Recopilación Investigativa 2009. Tomo 1	978-99922-21-59-1 (v.1)
Marzo 2010	Recopilación Investigativa 2009. Tomo 2	978-99922-21-60-7 (v.2)
Marzo 2010	Recopilación Investigativa 2009. Tomo 3	978-99922-21-61-7 (v.3)
Octubre 2010	Obstáculos para una investigación social orientada al desarrollo. Colección Investigaciones José Padrón Guillen	978-99923-21-62-1
Febrero 2011	Estructura familia y conducta antisocial de los estudiantes en Educación Media. Colección Investigaciones n.º 2 Luis Fernando Orantes Salazar	

Febrero 2011	Prevalencia de alteraciones afectivas: depresión y ansiedad en la población salvadoreña. Colección Investigaciones n.º 3 José Ricardo Gutiérrez Ana Sandra Aguilar de Mendoza	
Marzo 2011	Violación de derechos ante la discriminación de género. Enfoque social. Colección Investigaciones n.º 4 Elsa Ramos	
Marzo 2011	Recopilación Investigativa 2010. Tomo I	978-99923-21-65-2 (v1)
Marzo 2011	Recopilación Investigativa 2010. Tomo II	978-99923-21-65-2 (v2)
Marzo 2011	Recopilación Investigativa 2010. Tomo III	978-99923-21-67-6 (v3)
Abril 2011	Diseño de un modelo de vivienda bioclimática y sostenible. Fase I. Colección Investigaciones n.º 5 Ana Cristina Vidal Vidales Luis Ernesto Rico Herrera Guillermo Vásquez Cromeyer	
Noviembre 2011	Importancia de los indicadores y la medición del quehacer científico. Colección Investigaciones n.º 6 Noris López de Castaneda	978-99923-21-71-3
Noviembre 2011	Memoria Sexta Semana del Migrante	978-99923-21-70-6
Mayo 2012	Recopilación Investigativa 2011. Tomo I	978-99923-21-75-1 (tomo 1)
Mayo 2012	Recopilación Investigativa 2011. Tomo II	978-99923-21-76-8 (tomo 2)
Mayo 2012	Recopilación Investigativa 2011. Tomo III	978-99923-21-77-5 (tomo 3)
Abril 2012	La violencia social delincuencial asociada a la salud mental en los salvadoreños Colección Investigaciones n.º 7 Ricardo Gutiérrez Quintanilla	978-99923-21-72-0
Octubre 2012	Programa psicopreventivo de educación para la vida efectividad en adolescentes Utec-PGR. Colección Investigaciones Ana Sandra Aguilar de Mendoza Milton Alexander Portillo	978-99923-21-80-6

Noviembre 2012	Causas de la participación del clero salvadoreño en el movimiento emancipador del 5 de noviembre de 1811 en El Salvador y la postura de las autoridades eclesiales del Vaticano ante dicha participación. Claudia Rivera Navarrete	978-99923-21-88-1
Noviembre 2012	Estudio Histórico proceso de independencia: 1811-1823. José Melgar Brizuela	978-99923-21-87-4
Noviembre 2012	El Salvador insurgente 1811-1821 Centroamérica. César A. Ramírez A.	978-99923-21-86-7
Enero 2012	Situación de la educación superior en El Salvador. Colección Investigaciones n.º 1 Carlos Reynaldo López Nuila	
Febrero 2012	Estado de adaptación integral del estudiante de educación media de El Salvador. Colección Investigaciones n.º 8 Luis Fernando Orantes	
Marzo 2012	Aproximación etnográfica al culto popular del Hermano Macario en Izalco, Sonsonate, El Salvador. Colección Investigaciones n.º 9 José Heriberto Erquicia Cruz	978-99923-21-73-7
Mayo 2012	La televisión como generadora de pautas de conducta en los jóvenes salvadoreños. Colección Investigaciones n.º 10 Edith Ruth Vaquerano de Portillo Domingo Orlando Alfaro Alfaro	
Mayo 2012	Violencia en las franjas infantiles de la televisión salvadoreña y canales infantiles de cable. Colección Investigaciones n.º 11 Camila Calles Minero Morena Azucena Mayorga Tania Pineda	
Junio 2012	Obrajes de añil coloniales de los departamentos de San Vicente y La Paz, El Salvador. Colección Investigaciones n.º 14 José Heriberto Erquicia Cruz	

Junio 2012	San Benito de Palermo: elementos afrodescendientes en la religiosidad popular en El Salvador. Colección Investigaciones n.º 16 José Heriberto Erquicia Cruz Martha Marielba Herrera Reina	978-99923-21-80-5
Julio 2012	Formación ciudadana en jóvenes y su impacto en el proceso democrático de El Salvador. Colección Investigaciones n.º 17 Saúl Campos	
Julio 2012	Factores que influyen en los estudiantes y que contribuyeron a determinar los resultados de la PAES 2011. Colección Investigaciones n.º 12 Saúl Campos Blanca Ruth Orantes	978-99923-21-79-9
Agosto 2012	Turismo como estrategia de desarrollo local. Caso San Esteban Catarina. Colección Investigaciones n.º 18 Carolina Elizabeth Cerna Larissa Guadalupe Martín José Manuel Bonilla Alvarado	
Agosto 2012	Conformación de clúster de turismo como prueba piloto en el municipio de Nahuizalco. Colección Investigaciones n.º 19 Blanca Ruth Gálvez García Rosa Patricia Vásquez de Alfaro Juan Carlos Cerna Aguiñada Óscar Armando Melgar.	
Septiembre 2012	Mujer y remesas: administración de las remesas. Colección Investigaciones n.º 15 Elsa Ramos	978-99923-21-81-2
Octubre 2012	Responsabilidad legal en el manejo y disposición de desechos sólidos en hospitales de El Salvador. Colección Investigaciones n.º 13 Carolina Lucero Morán	978-99923-21-78-2
Febrero 2013	Estrategias pedagógicas implementadas para estudiantes de Educación Media y el Acoso Escolar (*bullying*). Colección Investigaciones n.º 25 Ana Sandra Aguilar de Mendoza	978-99923-21-92-8

Marzo 2013	Representatividad y pueblo en las revueltas de principios del siglo XIX en las colonias hispanoamericanas. Héctor Raúl Grenni Montiel	978-99961-21-91-1
Marzo 2013	Estrategias pedagógicas implementadas para estudiantes de educación media. Colección Investigaciones n.º 21 Ana Sandra Aguilar de Mendoza	978-99923-21-92-8
Abril 2013	Construcción, diseño y validez de instrumentos de medición de factores psicosociales de violencia juvenil. Colección Investigaciones José Ricardo Gutiérrez Quintanilla	978-99923-21-95-9
Mayo 2013	Participación política y ciudadana de la mujer en El Salvador. Colección Investigaciones n.º 20 Saúl Campos Morán	978-99923-21-94-2
Mayo 2013	Género y gestión del agua en la mancomunidad La Montañona, Chalatenango, El Salvador. Normando S. Javaloyes Laura Navarro Mantas Ileana Gómez	978-99923-21-99-7
Junio 2013	Libro Utec 2012 Estado del medio ambiente y perspectivas de sostenibilidad. Colección Investigaciones José Ricardo Calles Hernández	978-99961-48-00-2
Julio 2013	Guía básica para la exportación de la flor de loroco desde El Salvador hacia España, a través de las escuelas de hostelería del país vasco. Álvaro Fernández Pérez	978-99961-48-03-3
Agosto 2013	Proyecto Migraciones Nahua-pipiles del Postclásico en la cordillera del Bálsamo. Colección Investigaciones n.º 24 Marlon V. Escamilla William R. Fowler	978-99961-48-07-1
Agosto 2013	Transnacionalización de la sociedad salvadoreña, producto de las migraciones. Colección Investigaciones n.º 25 Elsa Ramos	978-99961-48-08-8

Septiembre 2013	La regulación jurídico penal de la trata de personas especial referencia a El Salvador y España. Colección Investigaciones Hazel Jasmin Bolaños Vásquez	978-99961-48-10-1
Septiembre 2013	Estrategias de implantación de clúster de turismo en Nahuizalco. Colección Investigaciones n.º 22 Blanca Ruth Gálvez Rivas Rosa Patricia Vásquez de Alfaro Óscar Armando Melgar Nájera	978-99961-48-05-7
Septiembre 2013	Fomento del emprendedurismo a través de la capacitación y asesoría empresarial como apoyo al fortalecimiento del sector de la Mipyme del municipio de Nahuizalco en el departamento de Sonsonate. Diagnóstico de gestión Colección Investigaciones n.º 23 Vilma Elena Flores de Ávila	978-99961-48-06-4
Septiembre 2013	Imaginarios y discursos de la herencia afrodescendiente en San Alejo, La Unión, El Salvador. Colección Investigaciones n.º 26 José Heriberto Erquicia Cruz Martha Marielba Herrera Reina Wolfgang Effenberger López	978-9961-48-09-5
Septiembre 2013	Memoria Séptima Semana del Migrante	978-99961-48-11-8
Septiembre 2013	Inventario de las capacidades turísticas del municipio de Chiltiupán, departamento de La Libertad. Colección Investigaciones n.º 33 Lissette Cristalina Canales de Ramírez Carlos Jonatan Chávez Marco Antonio Aguilar Flores	978-99961-48-17-0
Septiembre 2013	Condiciones culturales de los estudiantes de educación media para el aprendizaje del idioma Inglés. Colección Investigaciones n.º 35 Saúl Campos Morán Paola María Navarrete Julio Aníbal Blanco	978-99961-48-22-4

Septiembre 2013	Recopilación Investigativa 2012. Tomo I	978-99923-21-97-3
Septiembre 2013	Recopilación Investigativa 2012. Tomo II	978-99923-21-98-0
Noviembre 2013	Infancia y adolescencia como noticia en El Salvador. Camila Calles Minero	978-99961-48-12-5
Diciembre 2013	Metodología para la recuperación de espacios públicos. Ana Cristina Vidal Vidales Julio César Martínez Rivera	978-99961-48-4-9
Marzo 2014	Participación científica de las mujeres en El Salvador. Primera aproximación. Camila Calles Minero	978-99961-48-15-6
Abril 2014	Mejores prácticas en preparación de alimentos en la micro y pequeña empresa. Colección Investigaciones n.º 29 José Remberto Miranda Mejía	978-99961-48-20-0
Abril 2014	Historias, patrimonios e identidades en el municipio de Huizúcar, La Libertad, El Salvador. Colección Investigaciones n.º 31 José Heriberto Erquicia Martha Marielba Herrera Reina Ariana Ninel Pleitez Quiñonez	978-99961-48-18-7
Abril 2014	Evaluación de factores psicosociales de riesgo y de protección de violencia juvenil en El Salvador. Colección Investigaciones n.º 30 José Ricardo Gutiérrez	978-99961-48-19-4
Abril 2014	Condiciones socioeconómicas y académicas de preparación para la de los estudiantes de educación media. Colección Investigaciones n.º 32 Saúl Campos Paola María Navarrete	978-99961-48-21-7
Mayo 2014	Delitos relacionados con la pornografía de personas menores de 18 años: especial referencia a las tecnologías de la información y la comunicación con medios masivos. Colección Investigaciones n.º 34 Hazel Jasmín Bolaños Miguel Angel Boldova Carlos Fuentes Iglesias	978-99961-48-16-3

Junio 2014	Guía de buenas prácticas en preparación de alimentos en la micro y pequeña empresa	
Julio 2014	Perfil actual de la persona migrante en El Salvador. Utec-US COMMITTE	978-99961-48-25-5
Septiembre 2014	Técnicas de estudio. Recopilación basada en la experiencia docente. Flavio Castillo	978-99961-48-29-3
Septiembre 2014	Valoración económica del recurso hídrico como un servicio ambiental de las zonas de recarga del río Acelhuate. Colección Investigaciones n.º 36 José Ricardo Calles	978-99961-48-28-6
Septiembre 2014	Migración forzada y violencia criminal una aproximación teórica practica en el contexto actual. Colección Investigaciones n.º 37 Elsa Ramos	978-99961-48-27-9
Septiembre 2014	La prevención del maltrato en la escuela. Experiencia de un programa entre alumnos de educación media. Colección Investigaciones n.º 38 Ana Sandra Aguilar de Mendoza	978-99961-48-26-2
Septiembre 2014	Percepción del derecho a la alimentación en El Salvador. Perspectiva desde la biotecnología. Colección Investigaciones n.º 39 Licda. Carolina Lucero	978-99961-48-32-3
Diciembre 2014	El domo el Guegüecho y la evolución volcánica. San Pedro Perulapán (Departamento de Cuscatlán), El Salvador. Primer Informe. Colección Investigaciones n.º 41 Walter Hernández Guillermo E. Alvarado Brian Jicha Luis Mixco	978-99961-48-34-7
Enero 2015	Publicidad y violencia de género en El Salvador. Colección Investigaciones n.º 40 Camila Calles Minero Francisca Guerrero Morena L. Azucena Hazel Bolaños	978-99961-48-35-4

Marzo 2015	Imaginario colectivo, movimientos juveniles y cultura ciudadana juvenil en El Salvador. Colección Investigaciones n.º 42 Saúl Campos Morán Paola María Navarrete Carlos Felipe Osegueda	978-99961-48-37-8
Mayo 2015	Estudio de buenas prácticas en clínica de psicología. Caso Utec. Colección Investigaciones n.º 44 Edgardo Chacón Andrade Sandra Beatriz de Hasbún Claudia Membreño Chacón	978-99961-48-40-8
Junio 2015	Modelo de reactivación y desarrollo para cascos urbanos. Colección Investigaciones n.º 48 Coralia Rosalía Muñoz Márquez	978-99961-48-41-5
Junio 2015	Niñas, niños, adolescentes y mujeres en la ruta del migrante. Colección Investigaciones n.º 54 Elsa Ramos	978-99961-48-46-0
Julio 2015	Historia, patrimonio e identidades en el Municipio de Comasagua, La Libertad, El Salvador. Colección Investigaciones n.º 49 José Heriberto Erquicia Cruz Martha Marielba Herrera Reina	978-99961-48-42-2
Agosto 2015	Evaluación del sistema integrado de escuela inclusiva de tiempo pleno implementado por el Ministerio de Educación de El Salvador. (Estudio de las comunidades educativas del municipio de Zaragoza del departamento de La Libertad). Colección Investigaciones n.º 43 Mercedes Carolina Pinto Benítez Julio Aníbal Blanco Escobar Guillermo Alberto Cortez Arévalo Wilfredo Alfonso Marroquín Jiménez Luis Horaldo Romero Martínez	978-99961-48-43-9
Agosto 2015	Aplicación de una función dosis-respuesta para determinar los costos sociales de la contaminación hídrica en la microcuenca del Río Las Cañas, San Salvador, El Salvador. Colección Investigaciones n.º 45 José Ricardo Calles Hernández	978-99961-48-45-3

Octubre 2015	El derecho humano al agua en El Salvador y su impacto en el sistema hídrico. Colección Investigaciones n.º 50 Sandra Elizabeth Majano Carolina Lucero Morán Dagoberto Arévalo Herrera	978-99961-48-49-1
Octubre 2015	Análisis del tratamiento actual de las lámparas fluorescentes, nivel de contaminantes y disposición final. Colección Investigaciones n.º 53 José Remberto Miranda Mejía Samuel Martínez Gómez John Figerald Kenedy Hernández Miranda	978-99961-48-48-4
Noviembre 2015	El contexto familiar asociado al comportamiento agresivo en adolescentes de San Salvador. Colección Investigaciones n.º 52 José Ricardo Gutiérrez Quintanilla Delmi García Díaz María Elisabet Campos Tomasino	978-99961-48-52-1
Noviembre 2015	Práctica de prevención del abuso sexual a través del funcionamiento familiar. Colección Investigaciones n.º 55 Ana Sandra Aguilar de Mendoza María Elena Peña Jeé Manuel Andreu Ivett Idayary Camacho	978-99961-48-53-8
Diciembre 2015	Problemas educativos en escuelas de Cojutepeque contados por los profesores y profesoras. Escuela de Antropología. Julio Martínez	
Febrero 2016	Desplazamiento interno forzado y su relación con la migración internacional. Colección Investigaciones n.º 56 Elsa Ramos	978-99961-48-56-9
Marzo 2016	Monografía Cultural y socioeconómica del cantón Los Planes de Renderos. Colección Investigaciones n.º 57 Saúl Campos Paola Navarrete Carlos Osegueda Julio Blanco Melissa Campos	978-99961-48-60-6

Abril 2016	Modelo de vivienda urbana sostenible. Colección Investigaciones n.º 58 Coralia Rosalía Muñoz Márquez	978-99961-48-61-3
Mayo 2016	Recopilación de Investigaciones en Tecnología 2016: Colección Investigaciones n.º 59 Internet de las cosas: Diseño e implementación de prototipo electrónico para el monitoreo vía internet de sistemas de generación fotovoltaico. Omar Otoniel Flores Cortez German Antonio Rosa Implementación de un entorno de aprendizaje virtual integrando herramientas de *E-learning* y CMS. Marvin Elenilson Hernández Carlos Aguirre *Big data*, análisis de datos en la nube. José Guillermo Rivera Verónica Idalia Rosa Urrutia	978-99961-48-62-0
Julio 2016	Aplicación de buenas prácticas de negocio (pequeña y mediana empresa de los municipios de San Salvador, Santa Tecla y Soyapango en El Salvador.) Colección Investigaciones n.º 46 Vilma de Ávila	978-99961-48-44-6
Julio 2016	Afectaciones psicológicas en estudiantes de instituciones educativas públicas ubicadas en zonas pandilleriles. Colección Investigaciones n.º 60 Edgardo R. Chacón Manuel A. Olivar Robert David MacQuaid Marlon E. Lobos Rivera	978-99961-48-67-5
Octubre 2016	Los efectos cognitivos y emocionales presentes en los niños y las niñas que sufren violencia intrafamiliar. Colección Investigaciones n.º 61 Ana Sandra Aguilar Mendoza	978-99961-48-69-9
Noviembre 2016	Historia, patrimonio e identidad en el municipio Puerto de La Libertad, El Salvador. Colección Investigaciones n.º 62 José Heriberto Erquicia Cruz Paola María Navarrete Gálvez	978-99961-48-70-5

Febrero 2017	El comportamiento agresivo al conducir asociado a factores psicosociales en los conductores salvadoreños. Colección Investigaciones n.º 63 José Ricardo Gutiérrez Quintanilla Óscar Williams Martínez Marlon Elías Lobos Rivera	978-99961-48-72-9
Marzo 2017	Relaciones interétnicas: afrodescencientes en Centroamérica. Colección Investigaciones n.º 64 José Heriberto Erquicia Rina Cáceres	978-99961-48-73-6
Abril 2017	Diagnóstico de contaminiación atmosférica por emisiones diésel en la zona metropolitana de San Salvador y Santa Tecla. Cuantificación de contaminantes y calidad de combustibles. Colección Investigaciones n.º 65 José Remberto Miranda Mejía Samuel Martínez Gómez Yonh Figerald Kenedy Hernández Miranda René Leonel Figueroa Noé Aguirre	978-99961-48-75-0
Mayo 2017	Causas y condiciones del incremento de la migración de mujeres salvadoreñas. Colección Investigaciones n.º 66 Elsa Ramos	978-99961-48-76-7
Junio 2017	Etnografía del volcán de San Salvador. Colección Investigaciones n.º 67 Saúl Campos Morán Paola María Navarrete Carlos Felipe Osegueda	978-99961-48-77-4
Agosto 2017	Modelo de e-Turismo cultural aplicando tecnología *m-Learning*, georreferencia, visitas virtuales y realidad aumentada para dispositivos móviles. Colección Investigaciones n.º 68 Elvis Moisés Martínez Pérez Melissa Regina Campos Solórzano Claudia Ivette Rodríguez de Castro Ronny Adalberto Cortez Reyes Rosa Vania Chicas Molina Jaime Giovanni Turcios Dubón	978-99961-48-80-4

Octubre 2017	Influencia de la tradición oral, la cocina que practican los pueblos indígenas y las variantes dialectales en la conservación y difusión de la lengua náhuat pipil. Colección Investigaciones n.º 69 Morena Guadalupe Magaña de Hernández Jesús Marcos Soriano Aguilar Clelia Alcira Orellana Mercedes Carolina Pinto Julio Aníbal Blanco José Ángel García Tejada	978-99961-48-84-2
Noviembre 2017	Propuesta de políticas públicas frente al perfil demográfico de El Salvador Carolina Lucero Morán Guiomar Bay Saúl Campos Morán Lucía del Carmen Zelaya de Soto	978-99961-48-87-3
Noviembre 2017	El estado de las competencias de desarrollo de la mujer en la zona de La Libertad Ana Sandra Aguilar de Mendoza	978-99961-48-88-0
Diciembre 2017	Conocimiento financiero y económico entre estudiantes universitarios: un estudio comparativo entre El Salvador y Puerto Rico Modesta Fidelina Corado Roberto Filánder Rivas Ronald Hernández Maldonado	978-99961-48-89-7
Enero 2018	Situación actual del manejo de las aguas ordinarias en lotificaciones y parcelaciones habitacionales de la zona rural de El Salvador. Un análisis de cumplimiento técnico y legal aproximado Alma Carolina Sánchez Fuentes María Teresa Castellanos Araujo Ricardo Calles Hernández Erick Abraham Castillo Flores	978-99961-48-91-0
Abril 2018	Aplicación de herramientas *big data* al Viceministerio de Vivienda y Desarrollo Urbano del Ministerio de Obras Públicas de El Salvador Verónica Idalia Rosa José Guillermo Rivera	978-99961-48-97-2
Mayo 2018	Diagnóstico de necesidades de capacitación del personal de empresas del sector turismo del municipio de La Libertad Carlos Rolando Barrios López Blanca Ruth Gálvez Rivas	978-99961-48-98-9

Junio	Etnografía de Santa María Ostuma: tierra de la piña, leyendas y tradiciones Carlos Felipe Osegueda Osegueda Miguel Ángel Hernández Vásquez Georgina Sulamita Ordóñez Valle Francisco Enrique Santos Alvarenga Josué Mauricio López Quintana Miguel Ángel Rodas Ramírez	978-99961-48-99-6
Julio	El *ombudsman* de las audiencias de los medios de comunicación en El Salvador: factibilidad y aceptación Camila Calles Minero Leida Monterroza Matute	978-99961-86-00-4

Este libro se terminó de imprimir
en el mes de agosto de 2018
en los talleres de Tecnoimpresos, S.A. de C.V.
19ª. Av. Norte N.º 125,
ciudad de San Salvador, El Salvador, C.A.

En esta ocasión, la Universidad Tecnológica de El Salvador presenta dos investigaciones en el área de tecnología: "Extracción de conocimiento a partir de texto" y "Aulas conectadas: sistema IoT para el registro de asistentes". Ambas muestran la utilización de herramientas tecnológicas para la obtención de resultados de investigación, además analizan la aplicación de esos resultados en dos ámbitos: las aulas y el análisis de texto.

La Colección Investigaciones tiene el objetivo de evidenciar el trabajo científico de la Universidad Tecnológica de El Salvador ante la comunidad científica nacional e internacional, y la sociedad.

No hay enseñanza sin investigación ni investigación sin enseñanza
Pablo Freire

ISBN 978-99961-86-01-1

Vicerrectoría de Investigación y Proyección Social
Calle Arce y 19ª avenida Sur n.° 1045, edificio *Dr. José Adolfo Araujo Romagoza,*
San Salvador, El Salvador, (503) 2275 1013 / 2275 1011