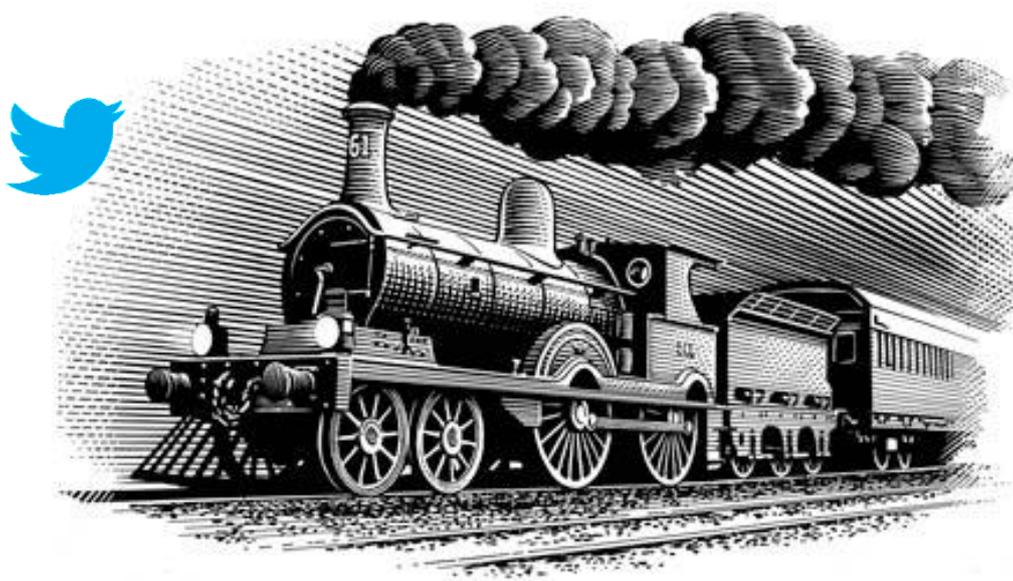


Trabajo Integrador Final: Ing. Industrial



Twitren

Autores:

Matías Anaya (46.344)

Juan Pablo Casal (46.258)

Ramiro Casares (46.363)

Nicolás Fontana (46.391)

Docente Guía: Ing. Maximiliano Catalano Dupuy

Año 2012

Resumen

La presente investigación analiza la factibilidad de desarrollar un sistema que permita predecir el tiempo de arribo de trenes de pasajeros, en un ramal de la provincia de Buenos Aires, Argentina, utilizando información aportada por los pasajeros del tren. Esta información es la geo localización y el timestamp de mensajes publicados en la red social Twitter, y emitidos desde las formaciones en movimiento.

Se desarrolló un modelo de simulación que contempla el tren en movimiento, y la emisión de Tweets desde el tren y los alrededores de la vía. Mediante la utilización de puestos de control, se junta la información de esas señales, para predecir la ubicación del tren y por ende su horario estimado de llegada a las distintas estaciones siguientes.

Se analizaron las variables claves a tener en cuenta para el éxito del modelo, entre las cuales se incluyen costos, difusión, tecnologías, etc.

Las distintas simulaciones realizadas permitieron calibrar el sistema, y determinar los valores críticos de las variables involucradas, que dimensionan el modelo y lo contrastan con los valores encontrados en la realidad.

Abstract

This investigation studies the feasibility of developing a system capable of predicting the time of arrival for passenger trains in Buenos Aires, Argentina, by using information provided by those passengers (location and timestamp of short messages posted by users in the social network Twitter, from within moving trains).

A simulation model was developed, which included the moving train, the emission of messages in Twitter by passengers, and from nearby people. Making use of checkpoints, the information from Twitter was processed to estimate the train's current location, and thus tell its estimated time of arrival to the subsequent stations in the system.

Different related-variables were studied, such as costs, awareness generation, technology, etc.

Different simulations carried out allowed us to tune the system, determine the critical values of involved variables, and size the model, while becoming aware of the existing gaps between these variables and real-world ones.

Índices

Índice General

INTRODUCCIÓN	1
TEMA	1
OBJETIVO	1
RESUMEN	1
DESARROLLO: CONCEPTOS	2
ESBOZO DEL PROBLEMA	2
SOBRE LAS TECNOLOGÍAS INVOLUCRADAS	2
<i>Trenes (Actualmente la Concesión TBA)</i>	2
<i>Twitter</i>	3
<i>Smartphones (Teléfonos Inteligentes)</i>	6
ENFOQUE Y METODOLOGÍA	6
FLUJOGRAMA DE NUESTRO PROCESO	8
1. <i>Front-End: Usuario</i>	8
Pre-Requisitos	8
Pasos a seguir	9
2. <i>Back-End: Acceso a la Información de Twitter</i>	10
Twitter Search API	10
Ejemplo de consulta	11
Consulta http	11
Devolución	11
Análisis	11
3. <i>Back-End: Procesamiento de la información</i>	12
4. <i>Front-End: Presentación de la Información Final al Usuario</i>	12
Página Web	12
Aplicación	13
Publicación en Twitter	13
Información a proveer	14
<i>Tiempos</i>	15
BENEFICIOS	17
COSTOS INVOLUCRADOS DEL PROYECTO	18
<i>Costos de Desarrollo</i>	18
<i>Costo de Instalación</i>	19
<i>Costo de Mantenimiento</i>	19
<i>Costo de Difusión</i>	20
Adopción de la tecnología	20
Otros Relacionados	21
FUENTES DE FINANCIAMIENTO	22
<i>Co-Branding</i>	22
<i>Sponsor</i>	24
<i>Financiamiento mediante concurso</i>	24
Realización de una página web:	24
Venta del producto:	24
<i>Autofinanciamiento</i>	24

DESARROLLO: SIMULACIÓN.....	26
SUPUESTO DEL ESTUDIO	26
DESARROLLO DEL MODELO DE SIMULACIÓN.....	27
<i>Tren</i>	27
Modelización del sistema de vías.....	27
Modelización del movimiento del tren	29
Implementación en Arena.....	29
Variables	30
Módulos.....	31
<i>Tweets</i>	35
Modelización de la generación de Tweets	35
Análisis estadístico de la situación actual	35
Implementación en Arena.....	39
<i>Procesamiento de datos</i>	41
Implementación en Arena.....	41
Implementación en Excel	44
Variables de Entrada	44
Variables de Salida	45
Datos a procesar	45
Indicadores de Exito	46
Key Performance Indicator	46
EXPERIMENTACIÓN.....	47
<i>Bases</i>	47
Parámetros comunes	47
<i>Resultados y Análisis</i>	48
Control	48
Variables de entrada.....	48
Resultado	48
Análisis.....	49
Ventana de tiempo para el contador	49
Variables de entrada.....	49
Resultado	49
Análisis.....	51
Error en la geo localización de la señal	52
Variables de entrada.....	52
Resultado.....	52
Análisis.....	55
Señal – Tweets emitidos desde el tren.....	56
Variables de Entrada.....	56
Resultados	56
Análisis.....	57
Ruido	57
Variables de entrada.....	57
Resultados	58
Análisis.....	59
Determinación del sentido.....	60
Variables de Entrada.....	60
Resultado	60
Análisis.....	60
<i>Simulación con variables reales</i>	61
Variables de entrada.....	61
Resultado	61
Análisis.....	63

CONCLUSIONES	64
FACTIBILIDAD TÉCNICA	64
CONSIDERACIONES PARA LA IMPLEMENTACIÓN	65
<i>Seteo óptimo del Sistema</i>	65
<i>Elección de los puestos de control</i>	66
<i>Relevamiento de parámetros para setear el sistema</i>	66
<i>Parametrización del sistema de vías</i>	67
<i>Información de salida</i>	67
FUENTES DE MEJORA	67
<i>Predicción del sentido</i>	67
<i>Filtro de Ruido</i>	68
<i>Memoria y contraste con el pasado</i>	69
LIMITACIONES DEL ESTUDIO	69
<i>Superposición de señales</i>	69
<i>Ruido desde estación</i>	69
<i>Circulación Irregular</i>	70
APÉNDICES	71
APÉNDICE I: HERRAMIENTAS PARA LA EXTRACCIÓN DE INFORMACIÓN DE TWITTER	71
<i>Json:</i>	71
<i>Python:</i>	71
Código en Python	71
APÉNDICE II: MUESTREO DE TIEMPO ENTRE TWEETS	72
APÉNDICE III: ANÁLISIS ESTADÍSTICO DE LOS TWEETS	73
<i>SPSS:</i>	73
<i>Q-Q Plot:</i>	74
<i>Distribución Exponencial</i>	74
<i>Box Plot</i>	75
<i>Stem-and-Leaf Plot</i>	76

Índice de Figuras

FIGURA 1: POPULARIDAD DE LAS DISTINTAS REDES SOCIALES EN AMÉRICA LATINA	4
FIGURA 2: CUENTAS DE TWITTER POR PAÍS (TOP 20).....	5
FIGURA 3: FLUJOGRAMA DE LA METODOLOGÍA	8
FIGURA 4: CONFIGURACIÓN DE GEO LOCALIZACIÓN EN TWITTER PARA BLACKBERRY	9
FIGURA 5: PANTALLA DE INICIO DE LA APLICACIÓN DE TWITTER EN BLACKBERRY.....	9
FIGURA 6: PROTOTIPO DE PRESENTACIÓN DE LA INFORMACIÓN.....	14
FIGURA 7: ESTACIONES A SIMULAR (ADAPTACIÓN DE INFORMACIÓN PROVISTA POR TBA)	28
FIGURA 8: SISTEMA DE COORDENADAS DEL MODELO	29
FIGURA 9: RECORRIDO DEL TREN EN ARENA	30
FIGURA 10: MÓDULO DEL MODELO	31
FIGURA 11: MÓDULO DEL MODELO	31
FIGURA 12: MÓDULO DEL MODELO	32
FIGURA 13: MÓDULO DEL MODELO	33
FIGURA 14: MÓDULO DEL MODELO	33
FIGURA 15: MÓDULO DEL MODELO	34
FIGURA 16: MÓDULO DEL MODELO	34
FIGURA 17: DISTRIBUCIÓN DE TWEETS EN EL ESPACIO.....	36
FIGURA 18: TIEMPO ENTRE TWITTEOS (SEGUNDOS).....	36
FIGURA 19: BOX PLOT.....	37
FIGURA 20: QQ PLOT	38
FIGURA 21: STEM-AND-LEAF PLOT - FRECUENCIA.....	38
FIGURA 22: MODELO DE GENERACIÓN DE TWEETS.....	39
FIGURA 23: MÓDULO DEL MODELO	39
FIGURA 24: MÓDULO DEL MODELO	41
FIGURA 25: MÓDULO DEL MODELO	42
FIGURA 26: MÓDULO DEL MODELO	43
FIGURA 27: MÓDULO DEL MODELO	44
FIGURA 28: DISTRIBUCIÓN DEL ERROR DEL EXPERIMENTO	49
FIGURA 29: VARIABLES DE ENTRADA	49
FIGURA 30: RESULTADOS	49
FIGURA 31: DISTRIBUCIÓN DEL ERROR DEL EXPERIMENTO	50
FIGURA 32: DISTRIBUCIÓN DEL ERROR DEL EXPERIMENT	50
FIGURA 33: DISTRIBUCIÓN DEL NÚMERO DE TWEETS EN EL CONTADOR	51
FIGURA 34: DISTRIBUCIÓN DEL ERROR DEL EXPERIMENTO	53
FIGURA 35: DISTRIBUCIÓN DEL ERROR DEL EXPERIMENTO	54
FIGURA 36: DISTRIBUCIÓN DEL ERROR DEL EXPERIMENTO	55
FIGURA 37: DISTRIBUCIÓN DEL ERROR DEL EXPERIMENTO	57
FIGURA 38: DISTRIBUCIÓN DEL ERROR DEL EXPERIMENTO	58
FIGURA 39: DISTRIBUCIÓN DEL ERROR DEL EXPERIMENTO	59
FIGURA 40: : DISTRIBUCIÓN DEL ERROR DEL EXPERIMENTO	62
FIGURA 41: MÁRGEN DE ERROR VS CANTIDAD DE TWEETS EN CONTADOR	62
FIGURA 42: TWEETS EMITIDOS ENTRE ESTACIONES VS. % DE TRENES.....	64

Índice de Tablas

TABLA 1: EJEMPLO DE DATOS EXTRAÍDOS	12
TABLA 2: POSICIÓN DE LAS ESTACIONES.....	28
TABLA 3: POSICIONES DEL TREN EN EL MODELO	30
TABLA 4: PARÁMETROS COMUNES	47
TABLA 5: VARIABLES DE ENTRADA	48
TABLA 6: RESULTADOS.....	48
TABLA 7: EVALUACIÓN DEL ERROR TEÓRICO MEDIO	51
TABLA 8: VARIABLES DE ENTRADA	52
TABLA 9: RESULTADOS.....	52
TABLA 10: MÁRGENES	55
TABLA 11: VARIABLES DE ENTRADA	56
TABLA 12: RESULTADOS.....	56
TABLA 13: VARIABLES DE ENTRADA	57
TABLA 14: RESULTADOS.....	58
TABLA 15: MÁRGENES	59
TABLA 16: VARIABLES DE ENTRADA	60
TABLA 17: RESULTADOS.....	60
TABLA 18: VARIABLES DE ENTRADA	61
TABLA 19: RESULTADOS.....	61
TABLA 20: CANTIDAD DE TWEETS POR USUARIO (TOP 8).....	68

Introducción

Tema

La presente investigación apunta a analizar la posibilidad de proporcionar información en tiempo real, sobre el estado del servicio de trenes en Buenos Aires, a partir de datos generados por los usuarios mediante la red social Twitter.

Objetivo

Estudiar la factibilidad técnica de la propuesta y detectar las consideraciones para su implementación.

Resumen

Se toma el ramal Retiro-Tigre de la línea Mitre del TBA como objeto de estudio. Se propone analizar la factibilidad de desarrollar un sistema que permita predecir el tiempo de arribo de los trenes, a las distintas estaciones, alimentándose de información aportada por los usuarios. Dicha información es la geo localización y el timestamp de los tweets (actualizaciones de estado publicadas en Twitter) emitidos desde un tren.

Se propone analizar las variables relacionadas a la implementación, para evaluar su viabilidad. Se buscará también identificar y dimensionar las variables críticas que podrían extrapolar este proyecto a otras líneas de trenes.

Esta propuesta nace de querer aprovechar el Trabajo Final de Ingeniería Industrial para impactar positivamente en la sociedad, creando algo que nosotros consideramos útil y de lo que el ITBA pueda estar orgulloso de ser asociado.

Desarrollo: Conceptos

Esbozo del problema

La presente investigación apunta a diseñar un sistema de geo localización de formaciones ferroviarias haciendo uso de datos generados por los usuarios de la misma. Estos datos se compilan de una red social popular (Twitter) y se procesan mediante un modelo diseñado, para predecir la ubicación de las formaciones. Se realizarán corridas del modelo en una aplicación de simulación, y se realizarán análisis de sensibilidad para estudiar la precisión del sistema, y los valores críticos de las variables involucradas, para asegurar un acertado funcionamiento del sistema.

Las razones que dan lugar a esta investigación, en épocas donde son populares y de bajo costos sistemas de localización satelital (GPS), es la falta de divulgación de esta información por parte de la empresa concesionaria de la línea, y el hecho de que los cronogramas de horarios no son respetados por diversas razones, resultando todo esto en un servicio poco puntual y confiable.

Dejando de lado las trabas y las razones que pudiera tener la empresa para no brindar tan valiosa información para los usuarios, es que se pensó en este proyecto, en el que muchos usuarios podrían aportar información para lograr una aproximación de la localización de los trenes, y así poder volver a tener una idea de los horarios de partida y llegada de las formaciones.

Sobre las tecnologías involucradas

Trenes (Actualmente la Concesión TBA)

El servicio de Trenes de Buenos Aires (TBA) en su recorrido Retiro-Tigre, transporta arriba de 42 millones de pasajeros por año¹ a una tarifa congelada y extremadamente baja (alrededor de 25 centavos de dólar, para recorrer arriba de 30 km).

Las formaciones que realizan el trayecto Retiro-Tigre, construidas por EMFER S.A, se componen de 5 vagones con capacidad para 64 personas sentadas², los cuales siempre van sobrecargados, por los que suelen haber, en promedio, unas 20 personas paradas por vagón. En promedio entonces se mueven alrededor de 450 personas por formación.

¹ Dato de TBA (www.tbanet.com.ar) El valor pertenece al año 2007, que es el último disponible por parte de TBA. Se supone un valor considerablemente mayor, por el crecimiento poblacional, y por el crecimiento en la cantidad de pasajeros que viajan sin pagar boleto (no registrados).

² Dato de EMFER S.A - <http://www.emfer.com.ar/trayectoria.htm>

Es de público conocimiento que la concesión no se encuentra en una situación favorable; TBA maneja otros ramales, además del mencionado, y recibe subsidios del estado mayores a los recaudados por la venta de boletos. Al 2010, se estimaba³ que por cada boleto vendido, el subsidio recibido por la empresa triplicaba el valor nominal del boleto. El estado de las formaciones no es bueno, las condiciones de viaje tampoco, y ocurren accidentes.

Diferentes factores vinculados a la condición de las formaciones y sus salidas de servicio por imposibilidad de reparación, provocan retrasos en las mismas; por lo que si bien existe un cronograma de horarios definido⁴, éste ya no es confiable, lo que conlleva una imposibilidad de saber con certeza cuándo pasará una formación, como para poder calcular una llegada a horario.

Twitter

Es el medio principal a ser utilizado al momento de recopilar datos. Twitter es una plataforma muy popular de micro-blogging en la que un usuario sube a su espacio un mensaje corto (140 caracteres) llamado tweet para compartir con sus seguidores. El usuario a su vez puede suscribirse a los mensajes de otros usuarios para visualizarlos en la página principal. Esto se llama “seguir” a otro usuario. De esta forma el usuario visualizarlos tweets de todos los usuarios a los que sigue en orden cronológico inverso.

Twitter fue creado en marzo 2006 y lanzado en julio del mismo año por Jack Dorsey en San Francisco, California. Desde ahí la red experimentó un crecimiento exponencial que continua hasta el día de hoy, expandiéndose a todos los países. En un estudio realizado por comScore, empresa líder en la medición del mundo digital, sobre las tendencias de uso de Internet en Latinoamérica, se reveló que en 2011 Twitter tuvo un crecimiento del 60% en cantidad de visitantes.

³ Diario La Nación. <http://www.lanacion.com.ar/1476083-gobierno-y-tba-el-fin-de-una-intima-relacion>

⁴ http://www.mitresarmiento.com.ar/Horarios_Retiro-Tigre.pdf

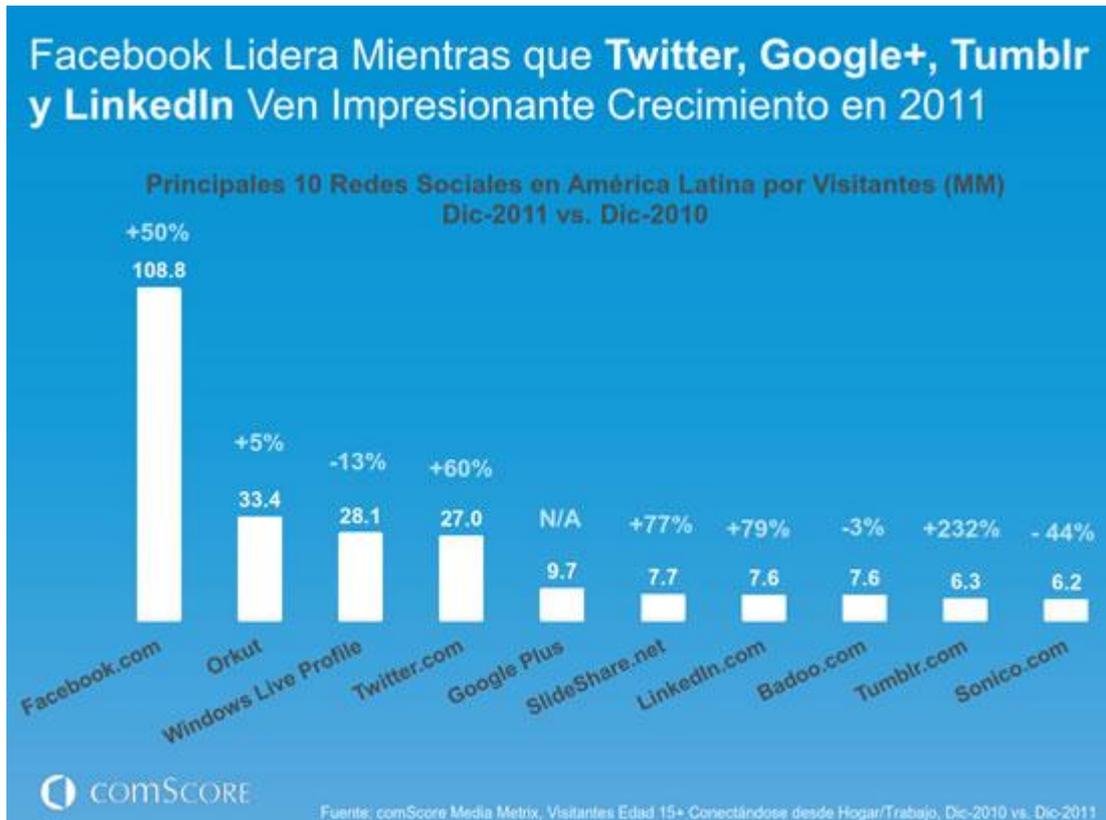


Figura 1: Popularidad de las distintas redes sociales en América Latina

Según datos del 2012⁵ existen en el mundo 500 millones de cuentas creadas en Twitter con un crecimiento de 1 millón de cuentas al día y se emiten alrededor de 340 millones de tweets por día. Estados Unidos lidera el ranking de los países con más cuentas creadas, aunque según los últimos estudios Twitter es un fenómeno mundial con gran penetración en Latinoamérica. El siguiente gráfico muestra los países con mayor número de cuentas creadas hasta el primer día del 2012.

⁵ Infographic Labs <http://www.bitrebels.com/social/twitter-2012-the-projected-stats-facts-infographic/attachment/twitter-2012-facts-infographic-1/>

Top 20 countries in terms of Twitter accounts

(accounts created before 01-01-2012)

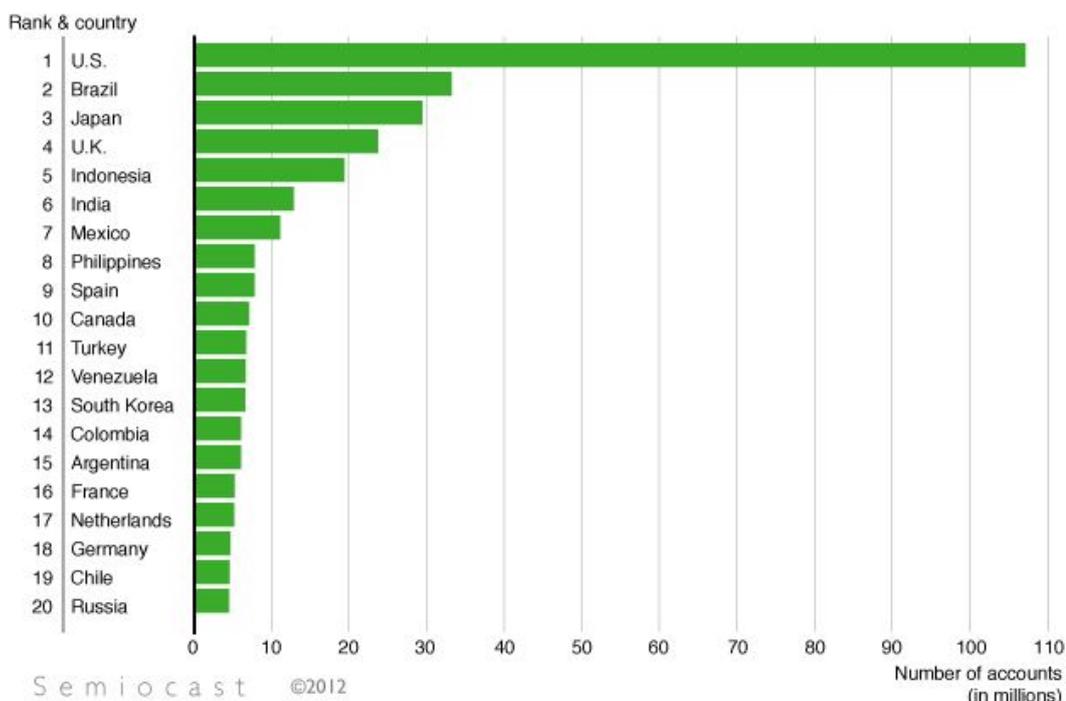


Figura 2: Cuentas de Twitter por país (Top 20)

Argentina, además de ser el quinceavo país con más cuentas creadas es uno de los países que registra mayor penetración de esta red social entre los usuarios de internet. Según datos de comScore, obtenidos de una investigación realizada a mediados de 2011, Argentina se ubica en el puesto 7 de países con mayor penetración de Twitter entre la población en línea, midiendo un 18%. Este dato demuestra el grado de adopción de esta red social en el país, siendo la principal vía de comunicación elegida por grandes personalidades como atletas y figuras del espectáculo.

Existen aplicaciones para teléfonos inteligentes, que permiten “twittear” desde dichas unidades, y una función interesante y clave para este proyecto, es la posibilidad de geo localizar los tweets emitidos. El teléfono entonces adjunta con el tuiteo un dato sobre la ubicación de la persona al momento de emitir el mensaje, el cual es conseguido por el teléfono ya sea por medio de GPS (en los casos que el teléfono posea) o por triangulación de antenas (en el resto de los teléfonos, y con una precisión menor)

Para teléfonos de tipo BlackBerry, incluyendo el modelo 8520 (único modelo fabricado en el país, y el más vendido debido a la restricción a las importaciones) existe

documentación que se explaya más sobre la precisión de su geo localización⁶. El método típico de localización mediante triangulación de antenas celulares tiene un error de entre 0,2 y 1 kilómetro de radio⁷.

Smartphones (Teléfonos Inteligentes)

Los Smartphones básicamente son teléfonos capaces de conectarse a internet, y hacer uso de dicha conectividad para hacer funcionar aplicaciones interactivas, buscar información, correos electrónicos, etc. El 24% de la población tiene teléfonos de este tipo en Argentina⁸.

El share de los teléfonos inteligentes en el mercado está definitivamente creciendo, por la necesidad cada vez mayor de las personas de estar comunicadas constantemente mediante distintas formas: redes sociales, mensajería instantánea, plataformas de micro-blogging como Twitter, etc. Tomaremos por ende como válida la tendencia creciente del share de los teléfonos inteligentes en el mercado, y más aún en el tramo ferroviario a ser estudiado, por el nivel adquisitivo mayor que presentan, en promedio, sus pasajeros.

Enfoque y Metodología

Nuevamente, la idea principal tras este proyecto es la utilización de tecnologías pre-existentes con el fin de brindarle a los pasajeros la posibilidad, mediante su propia participación, de tener una mejor idea que la que tienen hoy sobre dónde se encuentra un tren en un determinado momento, y por ende, cuánto podría tardar en llegar a su estación de partida.

La hipótesis básica en la que se basa nuestra investigación es el hecho de que si hay personas emitiendo tweets desde una formación en movimiento, entonces un muestreo de la cantidad de tweets en la porción de tierra por la cual está avanzando esa formación, devolvería un salto en la densidad de mensajes detectados, lo que podría pensarse como una nube móvil de tweets.

La metodología a utilizar consta de dos fases; la primera, es un análisis estadístico de una porción de vía ferroviaria, con el fin de determinar los parámetros que serán para nosotros indicativos de la presencia de un tren. Una vez conseguido esto, la segunda fase será la modelización del recorrido del tren mediante simulación en Arena, con el

6

http://docs.blackberry.com/en/developers/deliverables/644/GPS_and_BlackBerry_Maps_Development_Guide.pdf

⁷ Ver documentación con la especificación de los valores utilizados en http://docs.blackberry.com/en/developers/deliverables/644/GPS_and_BlackBerry_Maps_Development_Guide.pdf

⁸ <http://www.lanacion.com.ar/1473620-el-smartphone-otra-pasion-argentina>

fin de poder alterar los valores de las variables que resulten significativas en la primera fase (como podrían ser cantidad de personas tuiteando sobre el tren, cercanía de formaciones entre sí, sentidos de circulación, etc.). Esta alteración de las variables, debería repercutir directamente en los indicadores de la performance del modelo (porcentaje de trenes detectados, cantidad de falsos positivos, y otros que se estudiarán más adelante en la sección correspondiente a la simulación).

El parámetro obtenido en la primera fase será utilizado como determinante en el output del modelo realizado en la segunda fase, con el fin de determinar, frente a variaciones de las variables de entrada del modelo, en qué casos podríamos estar en condiciones de determinar la presencia de un tren exitosamente.

El siguiente modelo simplificado resume la metodología:

- Se determina tras los correspondientes muestreos, que de poder detectar X tuiteos en una coordenada dada, tendría que estar pasando un tren con un nivel de seguridad Y.
- Se simulan distintos recorridos de tren, en horario, con demoras, con distinta cantidad de gente tuiteando, y variando otros parámetros de importancia, y se evalúa en distintos puntos del recorrido el “modelo de detección” creado antes.
- Se contrasta el resultado del modelo en cada nodo, con la presencia o ausencia del tren en ese nodo en ese momento (lo que es sabido, ya que es propiedad de la simulación)
- Se verifican los valores necesarios de las variables críticas del modelo, para que las pruebas mencionadas en el paso anterior devuelvan coincidencias en todas, o la mayor cantidad de nodos posibles.

Flujograma de nuestro proceso

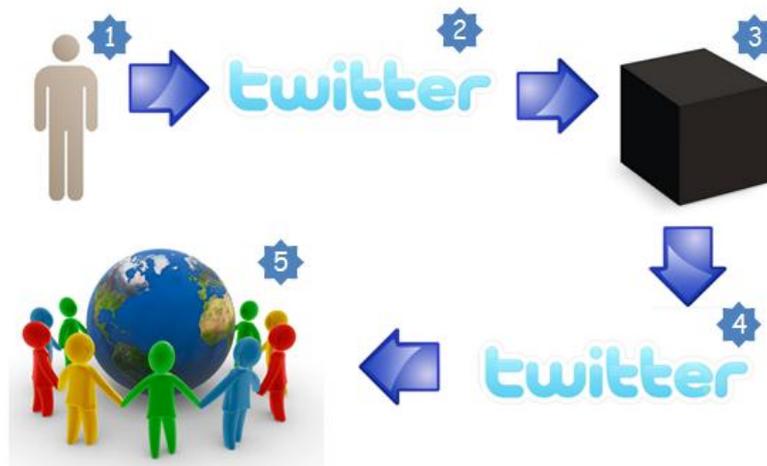


Figura 3: Flujograma de la metodología

El proceso tiene 5 pasos:

1. El usuario tuitea con la geo localización activada.
2. Twitter sube la información.
3. Extraemos la información de Twitter y la procesamos
4. Subimos esa información a Twitter
5. Los usuarios de Twitter pueden saber cuánto falta para el próximo tren

1. Front-End: Usuario

Esta sección describe el rol del usuario y sus requisitos para hacer uso de nuestra plataforma.

Pre-Requisitos

- Crear una cuenta en Twitter. *Se puede hacer una ingresando a twitter.com/signup y el único requisito es tener una cuenta de email.*
- Tener un teléfono celular disponible, compatible con la aplicación de Twitter. *Esto se cumple en varios BlackBerry, todos los iPhones y algunos equipos que utilizan Android.*
- Tener un plan de datos asociado al teléfono celular. Twitter y su aplicación para celulares utiliza una conexión a internet desde el celular para poder subir la información a sus servidores

- Instalar la aplicación de Twitter en el teléfono celular. *Se puede descargar ingresando a twitter.com desde el teléfono celular y siguiendo las indicaciones que ahí aparecen.*
- Configurar la aplicación para que los tweets incluyan Geo localización. *En la aplicación para el teléfono celular ir a “opciones > agregar ubicación a tweet” y marcar la opción.*



Figura 4: Configuración de geo localización en Twitter para BlackBerry

Pasos a seguir

1. Subir al tren
2. Abrir la aplicación de Twitter en su teléfono celular. Este paso será distinto de acuerdo al modelo de teléfono utilizado.

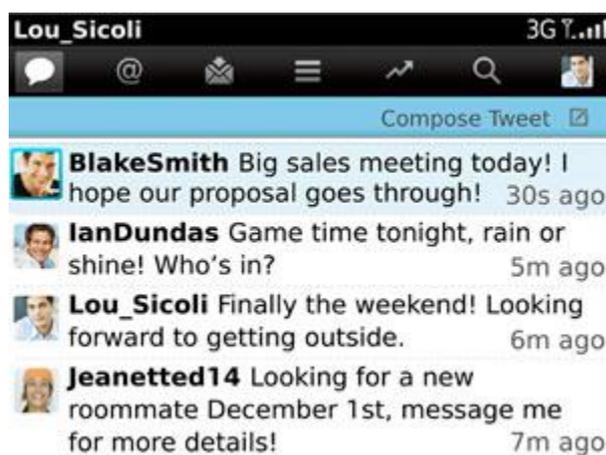


Figura 5: Pantalla de inicio de la aplicación de Twitter en BlackBerry

3. Redactar un Tweet nuevo. En general implica presionar el botón “redactar tweet”, aunque varía dependiendo de cómo lo tenga configurado cada usuario. Luego hay que darle contenido.
4. Activar la geo localización para el nuevo Tweet. Hay que presionar el símbolo que parece una mira, para que se fije la localización del teléfono.
5. Presionar el botón “Tweet”

2. Back-End: Acceso a la Información de Twitter

Consta de dos partes. La primera, en donde extraemos la información, se basa en realizar consultas sobre la página misma de Twitter, la cual posee una herramienta de búsqueda muy simple. Esta herramienta, denominada *Search API* (<https://dev.twitter.com/docs/api/1/get/search>) funciona mediante consultas simples (la sintaxis necesaria para cada tipo de consulta se encuentra documentada online) con las cuales uno puede pedirle desde tweets que contengan ciertas palabras, hasta tweets ubicados en cierta latitud y longitud. Lógicamente en nuestro caso utilizaremos esta última, sumándole la variable tiempo. Por ende, le estaríamos pidiendo a Twitter que nos devuelva cuales tweets fueron emitidos en cierto lugar en cierto tiempo.

Una vez recolectada la información, pasamos a la segunda etapa de este proceso. Aquí se debe moldear los datos de salida de la Twitter API para que sean procesados por nuestro algoritmo. Para el prototipado del mismo (el cual incluyo mucho manejo artesanal de datos y modelos) utilizamos un archivo `.json` (ver anexo) desde el cual debemos extraer la información útil, ya que como vemos en otro capítulo, obtenemos desde datos del usuario que lo creó, hasta datos del texto mismo que en el caso de estudio no son relevantes para el resultado final. Para obtener esto, utilizamos la herramienta Python (ver anexo), y mediante una sintaxis simple (ver anexo) obtenemos las variables deseadas, como tiempo y ubicación. Sin embargo, para la implementación real del sistema, estas etapas del proceso deben ser rediseñadas y los datos modelados acordes a la implementación final del algoritmo.

Twitter Search API

La base de Twitter es pública. Las consultas se hacen mediante consultas *http* a un servidor. Las mismas son anónimas y está limitada la cantidad que un determinado IP puede hacer por hora (De todas formas, existe otro mecanismo que se puede utilizar para extraer información de la base de Twitter sin limitación de cantidad, la cual debe ser utilizada al momento de implementar realmente el sistema).

La estructura básica de una consulta es '<http://search.twitter.com/search.json?q=>'. A esto hay que agregarle los parámetros de la consulta. En nuestro caso los parámetros van a ser los siguientes:

Geocode - Coordenadas (latitud y longitud) desde donde se emiten los tweets. La consulta devolverá únicamente resultados en esa área. Radio, En kilómetros, para transformar la localización en un área y que la consulta tenga sentido. Los Geocodes los obtendremos a partir de un mapa digital, como Google Maps, en el cual podemos observar claramente todo el tramo de vía, y elegir los puntos que consideremos tendrían que ser nuestras marcas de control, sabiendo sus respectivas latitudes y longitudes. La aplicación de Google Earth, a su vez permite graficar radios a partir de un punto dado. Con eso definiríamos los puntos y los radios a trabajar.

Page - El número de la página que queremos nos devuelva.

result_type - Recent

rpp - 100

since_id - Para evitar traer demasiado cada vez se utiliza el ‘&’ como separador de parámetros.

Ejemplo de consulta

Consulta http

http://search.twitter.com/search.json?q=*&geocode=-34.5173598913518,-58.48496675491333,2km

Devolución

```
{
  "completed_in": 0.196,
  "max_id": 199926456942465024,
  "max_id_str": "199926456942465024",
  "next_page": "?page=2&max_id=199926456942465024&q=*&geocode=-34.5173598913518%2C-58.48496675491333%2C2km",
  "page": 1,
  "query": "*",
  "refresh_url": "?since_id=199926456942465024&q=*&geocode=-34.5173598913518%2C-58.48496675491333%2C2km",
  "results": [
    {
      "created_at": "Tue, 08 May 2012 18:19:16 +0000",
      "from_user": "pancholimongi",
      "from_user_id": 76092294,
      "from_user_id_str": "76092294",
      "from_user_name": "Francisco Limongi",
      "geo": null,
      "location": "\u00dct: -34.507999,-58.492555",
      "id": 199926456942465024,
      "id_str": "199926456942465024",
      "iso_language_code": "it",
      "metadata": {
        "result_type": "recent"
      },
      "profile_image_url": "http://a0.twimg.com/profile_images/2006038341/image_normal.jpg",
      "profile_image_url_https": "https://si0.twimg.com/profile_images/2006038341/image_normal.jpg",
      "source": "&lt;a href=&quot;http://twitter.com/#!/download/iphone&quot; rel=&quot;nofollow&quot;&gt;Twitter for iPhone&lt;/a&gt;",
      "text": "@zoccerzone como vas francisco.limongi@gmail.com",
      "to_user": "zoccerzone",
      "to_user_id": 126501480,
      "to_user_id_str": "126501480",
      "to_user_name": "Paul Zabala",
      "in_reply_to_status_id": 199921932538355712,
      "in_reply_to_status_id_str": "199921932538355712",
      "created_at": "Tue, 08 May 2012 16:24:29 +0000",
      "from_user": "nattklein21",
      "from_user_id": 295268083,
      "from_user_id_str": "295268083",
      "from_user_name": "natalie klein",
      "geo": {
        "coordinates": [-34.5147,-58.4902],
        "type": "Point"
      },
      "id": 199897571504816128,
      "id_str": "199897571504816128",
      "iso_language_code": "es",
      "metadata": {
        "result_type": "recent"
      },
      "profile_image_url": "http://a0.twimg.com/profile_images/1344316192/277_normal.jpg",
      "profile_image_url_https": "https://si0.twimg.com/profile_images/1344316192/277_normal.jpg",
      "source": "&lt;a href=&quot;http://twitter.com/#!/download/iphone&quot; rel=&quot;nofollow&quot;&gt;Twitter for iPhone&lt;/a&gt;",
      "text": "Enferma en casa sola! Odio estooo",
      "to_user": null,
      "to_user_id": 0,
      "to_user_id_str": "0",
      "to_user_name": null
    }
  ],
  "results_per_page": 15,
  "since_id": 0,
  "since_id_str": "0"
}
```

Análisis

Cómo ya se explicó previamente este es un archivo JSON, y mediante la herramienta Python extraeremos las siguientes variables:

1. *created* - fecha de creación (fecha, hora, minutos y segundos)
2. *from_user_id* - identifica al usuario
3. *geo* - posición (latitud y longitud)
4. *type = point* - indica si es una carga móvil geo localizada
5. *source* - indica el dispositivo utilizado para twittear

El código anterior contiene, por ejemplo, los siguientes mensajes:

text	Pego la vuelta solo, con mi alma en pena, con besos perdidos en una sabana donde dejo la esencia este servidor. http://t.co/vC80qeRY
text	Como que ya record... 3 veces en esta estaci\u00f3n de M!3R6a! (@ Estaci\u00f3n Retiro [L\u00ednea Mitre]) http://t.co/aT1Amsft
text	Dale que me quiero ir a #Mardel #MdP
text	Zzzzzzzz (@ Estaci\u00f3n Retiro [L\u00ednea Mitre] w/ 4 others) http://t.co/cQL3plWv
text	Avanzo desde ultimo vagon amigoo (@ Estaci\u00f3n Retiro [L\u00ednea Mitre] w/ 2 others) http://t.co/8Lt00D9T
text	Retiro... @ Estaci\u00f3n Retiro [L\u00ednea Mitre] http://t.co/Nk3EHgTI
text	Que onda, regalan chori y coca en los furgones q siempre hay q viajar asi? Y eso q la mayor\u00eda no tienen bici. . . =S http://t.co/5KMLuTps
text	Luego del curso en la UAI regreso a casa por fin seeee flying @nodannino (@ Estaci\u00f3n Retiro [L\u00ednea Mitre] w/ 5 others) http://t.co/Bhysa9iU
text	Estoy en Estaci\u00f3n Retiro [L\u00ednea Mitre] (Ciudad de Buenos Aires) w/ 6 others http://t.co/CCdM2Tdv
text	Waiting for my train back home, late as usual (@ Estacion Retiro [L\u00ednea TIGRE]) http://t.co/MmKXuDID
text	RT @m_anchorena: @CBTP_WEB en mi caso somos 3 con pasaje pago y emitido...viajo igual con o sin entrada
text	Here we go again (at Estaci\u00f3n Retiro [L\u00ednea Mitre]) \u2014 http://t.co/naj6a6Sj
text	@CBTP_WEB en mi caso somos 3 con pasaje pago y emitido...viajo igual con o sin entrada
text	Otra vez a provincia (at Estaci\u00f3n Retiro [L\u00ednea Mitre]) \u2014 http://t.co/DFZRioiE

Tabla 1: Ejemplo de datos extraídos

3. Back-End: Procesamiento de la información

A la información extraída en el punto anterior se le realiza el análisis estadístico que reconoce donde están ubicados los trenes (si es que los datos enviados desde los mismos son los suficientes) y entrega el resultado del mismo. Luego, a éste resultado se le aplican las correcciones existentes debido a los errores, y se calcula cuando va a llegar el tren a la próxima estación. Todo este proceso esta detallado en el desarrollo de la simulación.

4. Front-End: Presentación de la Información Final al Usuario

Una vez recopilada la información de Twitter y procesada utilizando el algoritmo desarrollado, se presenta la misma al usuario. La comunicación se puede efectuar por diversas vías que se analizan a continuación.

Página Web

Una opción es crear una página en Internet en la cual se suba la información del estado del sistema ferroviario (como en una especie de blog) para que el usuario pueda consultar desde su casa o desde un Smartphone. La información estaría disponible para cualquier usuario, sin necesidad previa de inscripción a ninguna red particular.

Tiene la ventaja de ser sencillo y tener un costo relativamente bajo. También se lograría un display de la información atractivo para el usuario y fácil de entender, ya que podría incluir esquemas e imágenes. Esto podría ser utilizado incluso para incluir publicidad, como alternativa de recaudación de fondos para el mantenimiento del proyecto.

La mayor desventaja es que la mayoría de los smartphones no tiene la velocidad de conexión a Internet suficiente como para hacer que la consulta sea un trámite rápido y

cómodo. Esto llevaría a tener una página extremadamente simple, que ocupe la menor cantidad de espacio posible, lo que jugaría en contra del diseño atractivo esperado en una página web. La alternativa sería tener una versión para computadoras, y otra para dispositivos móviles, pero crecería la complejidad y los costos asociados.

Aplicación

Esta opción consta de crear una aplicación para que los usuarios instalen en sus smartphones y consulten de manera instantánea el estado del sistema. Para llevar a cabo esta opción se requiere contratar programadores que desarrollen la aplicación, contar con un administrador de contenidos, un servicio de hosting y personal para mantener todo el sistema funcionando, realizar el mantenimiento requerido y brindar soporte a los usuarios.

Esta opción es la más conveniente para el usuario debido a que puede realizar la consulta de manera rápida y cómoda y con una interfaz atractiva y fácil de entender (similar a un usuario consultando la página web desde una computadora fija). También nos otorgaría la posibilidad de acceder en un futuro a fuentes más certeras de información: una aplicación podría acceder al GPS de aquellos teléfonos que cuenten con uno, y otorgarnos para ese usuario una información de localización sin error alguno. También la aplicación podría presentar una opción simple, como un botón, que los usuarios puedan apretar una vez sobre el tren para aportarnos información directamente, y nosotros saber fehacientemente que es información proveniente de un tren.

Las desventajas no obstante son muy elevadas: por un lado el desarrollo y mantenimiento de la misma para todas las plataformas móviles existentes es bastante costoso. Por otro lado, es un paso más complejo el lograr adopción de una base de usuario a una aplicación, que al simple hecho de twittear con geo localización, que no requiere nada radicalmente distinto a lo que la mayoría de los usuarios realizan a diario.

Publicación en Twitter

Consiste en crear uno o más usuarios en Twitter para publicar las actualizaciones de la localización de las formaciones para que el usuario consulte cada vez que se quiera informar.

Esta opción presenta como ventaja la mayor robustez, ya que solo requiere conexión a Internet y que no se caigan los servidores de Twitter para estar en funcionamiento. También es la opción de menores costos y la que requiere menor desarrollo.

Entre sus desventajas se puede mencionar que solo sirve a usuarios que tengan la opción de consultar Twitter, y que la forma de presentar la información se limita a un texto corto.

Entre estas opciones se optó por publicar la información en Twitter. En primer lugar porque es la forma más sencilla y fácil de automatizar. Con un software adecuado se puede programar para que se disparen los tweets automáticamente, requiriendo solamente una persona que monitoree el sistema para detectar fallas y posibles ajustes.

Se descartó la posibilidad de subir la información a una página web porque el usuario menos beneficiado es el que tiene que realizar la consulta desde un dispositivo móvil como cualquier Smartphone, y es justamente el usuario que está en la estación con su celular el que mayor provecho sacaría de nuestro sistema.

La opción de crear una aplicación para que el usuario instale en su Smartphone se descarta, al menos en una primera instancia del proyecto, porque requiere una inversión considerable. Se podría llevar a cabo cobrando al usuario un pequeño importe para bajar la aplicación para obtener un servicio eficiente, y con ese ingreso pagar los costos de lanzamiento y mantenimiento e incluso obtener una ganancia. No obstante esto iría en contra de la adopción de Twitren, ya que como se mencionó antes, uno de los incentivos principales para twittear con geolocalización y aportar datos a la plataforma, es el poder hacer uso de la misma, y beneficiarse. El cobrar por una aplicación alejaría a muchos usuarios, que por ende no se verían impulsados a geolocalizar sus mensajes y aportarnos información. El fin de nuestro proyecto tampoco es obtener un beneficio económico, sino proveer un servicio al usuario de trenes, libre de costo haciendo uso de la información que él mismo publica en Internet. El análisis de la viabilidad de hacer de esto un proyecto rentable quedaría pendiente para una etapa posterior, no abarcada por nuestro estudio.

Información a proveer

Habiendo optado por utilizar Twitter como vía de comunicación al usuario, la información se presentará en forma de texto, pronosticando el horario de llegada de la próxima formación a la estación requerida.



Figura 6: Prototipo de presentación de la información

Este es un ejemplo de cómo verá el usuario nuestro producto final. Como se puede observar habrá una dirección de Twitter destinada a cada una de las estaciones y se publicará el horario de modo que si el sistema deja de funcionar por cualquier motivo

el usuario se dará cuenta fácilmente en lugar de quedarse con la información del último tweet.

El usuario tendrá la opción de *seguir* a la estación que le interese vía Twitter y que le lleguen a su dirección todas las actualizaciones que se van haciendo. Este método es poco práctico porque una persona que viaja en tren generalmente lo utiliza una vez en el día en cada dirección, y le estarían llegando a su espacio en Twitter actualizaciones a lo largo de todo el día. La alternativa es que el usuario entre a consultar el espacio de la estación que quiera donde se publican las actualizaciones. El método para realizar esto es el siguiente:

- Entrar en la aplicación de Twitter para Smartphone
- Ir a la solapa de búsquedas y entrar. La misma está identificada por el siguiente logo:
- Escribir en la barra de búsqueda la dirección de la estación en la cual el usuario se encuentra.
- La búsqueda arrojará como resultado los últimos tweets ordenados de más reciente a más antiguo.

Tiempos

Los tiempos de cada paso son fundamentales, ya que vamos a mostrar información minuto a minuto. Por ende, mientras más rápido sea el proceso completo, mayor precisión le vamos a brindar al servicio (y teniendo en cuenta que ya contamos con errores por el uso de la geo localización, deberíamos intentar reducir al mínimo cualquier error extra que pueda surgir). Es por esto que vamos a identificar, paso a paso, cuales son las demoras esperadas para completar el proceso y a dar un breve resumen de que es lo que se realizará en dicho paso.

1. El usuario emite un tweet

El proceso comienza cuando un usuario subido al tren, tweekea con la geo localización activada. Para esto debe haber elegido en la solapa “opciones”, “Agregar ubicación a tweet” y luego, cuando está por escribir el sistema actualiza automáticamente la localización (con su error ya explicado previamente). De esta forma, el usuario ya envió los datos necesarios a Twitter y sólo es cuestión de esperar a que el sistema los suba. Dado que los tests pueden ser realizados desde cualquier parte del tren, el tiempo que el usuario pueda demorar no cambia el output del proyecto.

2. Twitter sube la información

Este paso depende de 2 procesos fundamentales. La conexión a internet del Smartphone, y la velocidad de Twitter para subir esa información y tenerla disponible en la API para la posterior descarga.

Con respecto a la conexión, si el teléfono cuenta con conexión 3G o está conectado a una red vía Wifi, el proceso es mucho más rápido, por ende, vamos a tomar el peor escenario que es el de la Red Móvil de su proveedor (Movistar, Claro, Personal, etc.). En éste último caso, el twitteo puede demorar desde 5 a 30 segundos, siendo éste por ende un punto clave en la precisión del sistema, ya que no está dentro de nuestras posibilidades el poder acelerar este proceso (a menos que se ponga una red Wifi dentro de cada Tren de TBA).

En cuanto al tiempo que Twitter tarda desde que recibe la información, hasta que lo sube a la web, se podría considerar como “nulo” el tiempo de demora ya que en el proceso dura menos de 1 segundo.

Con esto, quedaría un tiempo total variable entre 5 y 30 segundos.

3. Extraemos la información de Twitter y la procesamos

Este punto, es crucial para la eficiencia y efectividad del servicio que se va a brindar ya que es aquel sobre el cual tenemos más influencia a la hora de aumentar la precisión. El tiempo de extracción y proceso no obstante se despreciará, ya que será un procedimiento automatizado, cuyas variables de influencia no son considerables.

Este paso se hará desde un servidor central, y dado que los códigos API que ya se analizaron no son para nada extensos (el hecho de que se extraigan en promedio cada 2 minutos con información sobre lapsos de ese orden los hace aún menores) no nos hará incurrir en tiempos elevados de procesamiento. El tiempo que lleva este paso podrá por ende ser despreciado, sin impactar los resultados del modelo.

4. Se sube la información a Twitter

Ya que este proceso va a ocurrir de manera automática, y el sistema que va a generar el output del proceso anterior va a estar conectado a internet mediante una conexión veloz, los tiempos pasan a ser nuevamente casi nulos ya que una vez generado el output es sólo cuestión de enviar el Tweet y

esperar a que Twitter lo suba. En este caso, corren los mismos supuestos que en el paso 2, pero extrayéndole la demora generada por la Red Móvil, y por ende quitándole casi el 100% del tiempo.

5. Los usuarios de Twitter pueden saber cuánto falta para el próximo tren

Al llegar a este paso la información ya está disponible para los usuarios; lo único que hace falta es que lo vean. Por ende acá depende nuevamente de la conexión a internet del usuario. No obstante, este paso ya no es de interés del proyecto, ya que la información postada por nuestro algoritmo en Twitter, o la plataforma que eligiéramos para realizar el delivery, tiene la correspondiente marca horaria del momento preciso de su emisión. Esto quiere decir que por más que un usuario tuviera una conexión extremadamente lenta al momento de buscar la información sobre el estado del servicio, cuando finalmente pueda llegar a la misma, podrá también ver que eso que está leyendo se emitió “n” minutos atrás, lo que le dará una noción de la precisión de la misma.

En resumen, si sumamos todos los tiempos del proceso, nos daremos cuenta que el factor de retraso principal está al momento de la emisión del tweet por parte del usuario, por lo que estaríamos trabajando valores medios de entre 5 y 30 segundos.

Beneficios

Como se mencionó en la descripción del contexto, el funcionamiento actual de los trenes de la empresa TBA no es el mejor. Varias veces por mes se producen desperfectos técnicos que hacen salir formaciones de circulación generando retrasos. En algunos casos estos retrasos son tan grandes que la gente acumulada en las estaciones no tiene más remedio que utilizar otro medio de transporte, como colectivos o taxis. Al movilizarse semejante masa de personas cualquiera de estos medios de transporte colapsa, haciendo que los usuarios lleguen tarde a sus respectivos compromisos.

Poder de decisión - Si un usuario tuviera a su disposición información en tiempo real sobre la ubicación de las formaciones en circulación, podría entonces decidir por otro medio de transporte antes de llegar a la estación, evitando esperar 40 minutos por un tren o competir con todo el flujo de personas que abandona la estación por un lugar en un colectivo o taxi.

Menor incertidumbre - El usuario también sufre por la incertidumbre de no saber si el próximo tren llegará en 10 minutos o en media hora ya que pocas veces anuncian la

demora por el altoparlante (puede ser porque el encargado de anunciarlo no tiene noticias de la ubicación del tren más cercano). Esta incertidumbre genera un gran malestar en el usuario, sobre todo cuando tiene que cumplir un horario en el trabajo o asistir a una reunión importante. Saber aproximadamente cuál es el retraso genera un alivio en el usuario y a la vez le permite reorganizarse si es necesario.

Calidad del viaje - Otra información que resultaría beneficiosa para el usuario del TBA es cuándo llegará el tren que sigue al que arribará en primer lugar a la estación. La descoordinación en los horarios de partida y en la velocidad media de los trenes puede generar que dos formaciones lleguen a la misma estación con diferencia de uno o dos minutos. Esto hace que se genere un amontonamiento de usuarios en el primer tren y que el segundo vaya muy poco cargado. De saber esto, usuarios que están holgados con el tiempo dejarían pasar la primer formación para ir más cómodos en la segunda, haciendo más eficiente el transporte de pasajeros.

Costos Involucrados del Proyecto

En este apartado vamos a describir y estimar los principales costos relacionados al proyecto. Empezando por el desarrollo del software (tanto para procesamiento de la información de Twitter como para el posterior vuelco del resultado sobre el mismo) y su instalación, pasando por la concientización del usuario (para generar los Tweets necesarios para el correcto funcionamiento) y finalizando con el mantenimiento de la plataforma.

Costos de Desarrollo

Los dividiremos en 3 partes distintas:

- a. Extracción de la información.
- b. Procesamiento de la información.
- c. Vuelco del resultado.

Es necesario crear un algoritmo para poder extraer la información de los Tweets generados con geo localización de forma automática para cada puesto de control. Luego se necesita poder procesar esta información lo más velozmente posible para poder así indicar la presencia de un tren. Por último se debe volcar esta información de nuevo a Twitter. Todos estos procesos se deben realizar de forma automática, y para ello es necesario desarrollar y mantener un algoritmo que lo haga posible.

Para esto se requieren personas con conocimientos informáticos. Puede ser desde alguien que haya estudiado Ingeniería Informática, Ingeniería en Sistemas, u otra carrera similar. La cantidad y la calidad de lo creado van a depender principalmente del tiempo esperado para lograr el resultado. Pero esto está lejos de ser una limitante en

nuestro caso ya que, como se mencionó anteriormente, hoy en día no contamos con los Tweets necesarios para que el proyecto se haga realidad, y el proceso de concientización llevaría un tiempo considerable (más de 6 meses como mínimo).

Es por esto que se opta por el método más económico que es el de contratación de 2 estudiantes de las carreras mencionadas en calidad de Pasantes. De esta manera no se deberán pagar costos adicionales como las cargas sociales, entre otros, y el sueldo a pagar va a ser considerablemente menor al de un profesional.

Con esto se calcula en base a la oferta que hoy tiene el mercado, un sueldo estímulo de \$ 2,500 mensuales para dos personas por aproximadamente 3 meses. Este apartado es el único en el cual se es obligatorio el desembolso de dinero ya que los conocimientos necesarios son ya demasiado elevados para poder ser realizados por alguien que no sea un profesional/profesional en curso.

Queda así un total de \$ 15,000 necesarios para el desarrollo completo del software.

Costo de Instalación

Por otro lado es necesario contar no sólo con el software, sino también con la infraestructura capaz de poder ejecutar y procesar el software a la mayor velocidad posible. Para esto se calcula la compra de una pc de alto rendimiento, CPU Intel Core i5, que hoy en día está valuado en unos US\$1.500. Quedando así la inversión inicial total de \$ 22.500⁹.

Costo de Mantenimiento

A medida que pase el tiempo, será necesario un mantenimiento ya que puede haber desde cambios en Twitter que requieran por ende tanto modificaciones al sistema de extracción como al vuelco del resultado final. Además, el software puede llegar a tener algún malfuncionamiento que impida el procesamiento de la información.

Esta clase de arreglos, se consideran necesarios pero simples a su vez, es por esto que nuevamente se opta por la forma más económica, pagando a una persona de perfil similar al descrito anteriormente, pero en este caso en la modalidad de “trabajo freelance”. En otras palabras, se ofrece el pago de \$1,000 para el mantenimiento del sistema ya que se calcula que no requerirá más de una hora por día para la persona interesada. Esto generaría un costo constante de unos \$ 1,000 / mes.

⁹ Los dólares fueron convertidos a pesos argentinos a una tasa de \$5 por dólar estadounidense, siguiendo la tendencia alcista de la moneda extranjera en los últimos años, y calculando un valor útil para un horizonte futuro cercano (6 meses adelante/1 año)

Costo de Difusión

Adopción de la tecnología

Este es uno de los mayores desafíos que presenta el proyecto, ya que los *tweets* suelen ser emitidos sin geolocalización ya que ésta no es una opción predeterminada, y requiere un seteo extra.

Para acompañar el proyecto, se volvería necesario tomar acciones relacionadas con publicidad/marketing, como para generar la promoción suficiente, y lograr que la gente, además de twittear, lo haga con geolocalización.

Se analizará el costo y la factibilidad de generar campañas de difusión en las estaciones, y dentro de las formaciones del ramal Retiro-Tigre. Dado que habría que difundir por un lado la instrucción para twittear con geolocalización, y por el otro generar incentivos para emitir mensajes durante el recorrido, se apuntaría a impulsar lo primero en las estaciones, y lo segundo dentro de las formaciones. Se buscará proceder de esta manera, ya que es deseable que las personas twiteen desde el tren, y menos desde las estaciones, para evitar la generación de “ruido” (ver más adelante en el estudio analítico, cómo los tweets estancos desde las estaciones pueden generar perturbaciones en el sistema).

Como ideas preliminares para la promoción, se crearía por un lado material gráfico claro y explicativo de la tecnología para colocar en las estaciones. Dado que es un lugar donde las personas pasan tiempo detenidas esperando el tren, se prestaría a ser leído. Explicada la tecnología, también se brindaría un instructivo gráfico de 3 pasos enseñando a emitir tweets geo localizados. Para dentro de las formaciones, ya se apuntaría más a hacer que la gente genere mensajes vía Twitter. Se utilizaría en este caso material gráfico que haga referencia al proyecto (mediante el logo que eventualmente se cree para el mismo) y con frases, ideas, palabras, que inviten a twittear. Dado que el tiempo dentro del tren suele ser de ocio (la realidad es que las condiciones de viaje no siempre posibilitan la realización de otras actividades como lectura, contemplación del paisaje, etc.) el teléfono celular suele ser el pasatiempo de la mayoría de los pasajeros. De hecho, basta con viajar en tren para ver que muchas de las personas que lo rodean a uno, están utilizando su teléfono celular. Es entonces razonable pensar que de colocar material como el recién mencionado para fomentar la utilización de Twitter, sería efectivo para aumentar la cantidad de tweets emitidos desde las formaciones en movimiento. La incorporación del logo del proyecto, haría a los pasajeros asociar Twitter con el servicio, lo que esperamos los haga twittear utilizando geo localización.

Otros Relacionados

He aquí el costo de la concientización del potencial usuario, para poder generar así la cantidad necesaria de Tweets con geo localización de personas viajando arriba del tren. Afortunadamente, hoy en día existen medios gratuitos para poder comunicar al potencial usuario del producto que se quiere brindar. Es por esto que se va a utilizar, al menos en una primera etapa, tanto Facebook como YouTube para promocionar el producto y explicar al interesado el funcionamiento del mismo.

Es extremadamente importante en esta etapa el no obviar nada, y dejar por ende sumamente clara la importancia y la necesidad de los Tweets (con geo localización activada) sin los cuales el proyecto no vale nada, y los resultados serían cuasi-nulos.

Para llevar a cabo la comunicación, se planea utilizar nuevamente la modalidad Free-lance. En este caso el perfil a buscar sería totalmente distinto, ya que los fines son completamente opuestos. Es por esto que se piensa ofrecer un total de \$ 1,600 mensuales + un extra variable de alrededor de \$600 por cumplimiento de objetivos para algún licenciado en las carreras de Marketing, Publicidad, Comunicación o afines para que realicen el trabajo de manera profesional y se logre el resultado esperado. Este proceso de comunicación se planea mantener durante 6 meses constantemente, revisando en forma periódica el impacto que la “publicidad” está teniendo sobre el usuario (será simple detectar si se está siendo efectivo o no, ya que el aumento de Tweets será evidente) para así también poder ver qué clase de campañas están teniendo éxito, focalizar el trabajo en ellas y otorgar (o no) el adicional por objetivos. Para estar más organizados, se planea plantear objetivos mensuales para tener al cabo de los 6 meses los testeos necesarios para tener en correcto funcionamiento el sistema completo. El adicional se otorgará en proporción a la cercanía del objetivo cumplido (si se consiguen la mitad de los testeos, se pagará la mitad del incentivo, corriendo el mismo criterio si se supera lo esperado). De esta forma, la persona contratada estará fuertemente motivada para cumplir los objetivos ya que una buena performance puede significar hasta más de un 50% de paga a fin de mes. El contrato se firmará por 6 meses, ya que se cree que lograr los objetivos en un plazo menor es demasiado complicado, y se cree que la continuidad del trabajo es importante.

Una vez transcurridos los 6 meses, se evaluará el status actual del Tweeteo de los usuarios del tren, y se decidirá si se es necesario continuar con la implementación de la comunicación. A su vez se evaluará el trabajo realizado por la persona contratada y se barajará la posibilidad de renovar por el plazo necesario, como también el buscar otro candidato para el mismo. Por ende, nos encontramos con un costo que puede variar al cabo de los 6 meses.

En definitiva, quedaría un costo total de \$ 9,600 para los primeros 6 meses, con un costo variable de \$ 600 / mes para los meses subsiguientes, atado a los resultados del mismo. En caso de cumplir con los objetivos en el plazo esperado, el costo total

asciende a \$ 13,200. De no cumplir con los tiempos esperados, el costo variaría desde \$9,600 ~ \$13,000 para los primeros 6 meses, con un costo adicional similar al anteriormente planteado de \$1,600/mes + variables por \$600/mes.

No se están teniendo en cuenta los costos de espacios publicitarios en las estaciones y en las formaciones, ya que se piensa poder lograr ya sea descuentos considerables, o la cesión de espacios gratuitos, dada la naturaleza no lucrativa del proyecto, y las externalidades positivas que genera para los usuarios, y por ende para la percepción general del servicio de trenes. Dichos costos podrían ser recién estimados a partir de una negociación con el concesionario del servicio de trenes. Podrían crearse escenarios analizando en detalle las posibilidades, pero eso excede el alcance de este proyecto y bien podría ser motivo de una línea de investigación futura.

En resumen, y a modo estimativo, queda un total de inversión de alrededor de \$40.000.

Cabe destacar que el dimensionamiento del proyecto de inversión relacionado no es el objetivo de este proyecto, por lo que no se realizaron análisis detallados y exactos. El proyecto de inversión podría ser una línea futura de investigación, sembrada por el presente proyecto.

Fuentes de Financiamiento

Si bien no es el objetivo de este trabajo el ahondar en los detalles de financiamiento, se plantean distintas posibilidades que tendrían que ser puntos de partida para proyectos de inversión futuros que utilicen esta tecnología como centro.

Se plantearán a continuación algunas de las posibilidades existentes, que variarán dependiendo del objetivo que se quiere lograr con el proyecto:

1. Co-Branding
2. Sponsoreo
3. Financiamiento mediante concurso.
4. Auto-financiamiento.

Co-Branding

El financiamiento consiste en aliarse con alguna marca interesada en brindar el servicio para beneficio propio. Se puede lograr ya que el fin del proyecto es el brindar un servicio a miles de usuarios, y por ende la imagen de la marca se verá reforzada en las áreas en las cuales quiera hacer hincapié. Con este objetivo, Twitren ofrece un sin fin de oportunidades debido a su innovador mecanismo, y su moderna utilización de los medios actuales de información. Por ende se pueden realzar características

dependiendo de la marca. A continuación se enumeran algunas de las más interesantes.

Puntualidad - “Twitren ofrece un servicio que le permite llegar al usuario a destino con mayor precisión y ser así una persona más puntual y profesional.” Con este fin, se puede hacer desde co-branding con marcas de relojes o hasta de indumentaria para la oficina.

Modernidad - “Twitren ofrece lo último en servicio para el usuario, de manera gratuita y con el más simple acceso.” Aquí, la alianza con sitios web, o empresas de servicio.

Tecnología - “Twitren ofrece un servicio tecnológico que utiliza la inteligencia artificial para brindar al usuario una excelente oportunidad mediante el simple uso de la web”. En este caso en particular, las negociaciones pueden ser muy favorables, ya que no sólo aplican las empresas de productos tecnológicos en sí, sino que inclusive las compañías de teléfono.

En todos los casos, se pueden crear acuerdos especiales, para que la marca involucrada obtenga algún beneficio extra.

Por ejemplo, si se realiza un acuerdo con una página web, se puede plantear la posibilidad anteriormente mencionada de hacer una página especial para el servicio, y que la misma sea una sección dentro de ésta página web. Por ende, si el usuario quiere chequear el estado vía internet (sin utilizar Twitter), deberá visitar la página web para poder acceder y así aumentaría de forma automática el tráfico de la misma. Por ende al final de cuentas, la página web estaría pagando por tráfico (algo que es muy común hoy en día dentro del mundo cibernético, inclusive es uno de los principales ingresos de la gigante Google) de una forma indirecta y menos invasiva que las típicas publicidades.

Otro ejemplo, se puede dar al realizar un acuerdo con una marca de teléfonos celulares (smartphones necesariamente) o una compañía telefónica, donde se puede plantear la posibilidad de darle a los usuarios de esa marca en particular un servicio “extra” utilizando la información de Twitren. Puede ser desde una aplicación que se baje desde la misma página, o venga pre-instalada con el teléfono, y con la cual se provea de la misma información pero quizás más cómodamente (mediante apretar sólo un botón, u otras).

Este tipo de acuerdos exceden el alcance del proyecto en sí, pero uno puede notar las miles de oportunidades que existen para la financiación mediante Co-Branding. Además, dependiendo del acuerdo que se realice, se pueden ayudar a Twitren con la difusión del mismo (utilizando los medios que la marca misma ya tiene), achicando los costos previstos.

Sponsor

Al igual que mediante Co-Branding, en este caso el lanzamiento de Twitren estaría acompañado por un tercero ajeno al proyecto en sí. A diferencia del caso anterior, el principal fin para el cual se realizaría la alianza, sería porque el interesado quiere brindarles un servicio gratuito a los pasajeros del Tren. Es por esto que se menciona al estado como principal parte, ya que es el primer ente que surge al pensar en brindar servicio a la comunidad. Desde ya que se sabe que el verdadero interés puede no ser el de brindar servicio, sino también el de realzar la imagen de quien lo financie. Este punto se deja a juicio de quien quiera llevar adelante el proyecto.

Financiamiento mediante concurso

Hoy en día existen concursos focalizado en nuevos emprendimientos, que brindan fondos de hasta U\$S 50.000 para la realización de proyectos. Es por esto que se puede presentar el proyecto para concursar por uno de los mismos. En estos casos por lo general se cede una parte del proyecto que pasa a ser propiedad del financiador mismo. Si se opta por esta posibilidad, para que el proyecto tenga mayores posibilidades de ganar el concurso se debe plantear el ingreso de dinero en alguna etapa del mismo.

De ser este el caso se plantean las siguientes posibilidades:

Realización de una página web:

Mediante la realización de una página web en donde se le brinde al usuario la información de Twitren, se estará creando una página en la cual miles de visitantes entren a diario. De esta forma se creará una cantidad determinada de tráfico diario muy valiosa. Una vez generado un buen caudal, se podrán realizar acuerdos con otras marcas para vender publicidad dentro de la página.

Venta del producto:

Una vez creado y puesto en funcionamiento el producto, el mismo tendrá un valor dentro del mercado. Por ende se puede proyectar vender el mismo a una cantidad de dinero suficiente como para que el precio de venta exceda la inversión realizada y generar así ganancias.

Autofinanciamiento

La última opción presentada, requiere el juntar el dinero necesario entre los socios que lleven a cabo el proyecto para así ser dueños del 100% del mismo y poder hacer con él lo que deseen. Pueden desde comenzar una empresa propia que brinde servicios varios a los usuarios, algunos gratuitos como el Twitren y otros pagos, hasta realizar lo propuesto anteriormente de vender el producto una vez que fue lanzado a un precio

que supere los costos iniciales. Esto nuevamente se deja a gusto de quien lleve a cabo el proyecto.

Desarrollo: Simulación

Supuesto del estudio

El objetivo del sistema a desarrollar es el de informar en tiempo real la situación del servicio de trenes. Indistintamente de cómo se presente la información al usuario final, el sistema debe identificar la posición de las formaciones de tren en las vías en estudio. Por el contexto en el que está inmerso el sistema, tiene a su disposición la siguiente información para sacar conclusiones:

- Par (posición, tiempo) de todos los tweets emitidos en los últimos cinco días.

Esos tweets tienen características fundamentales. La posición informada está relacionada con la posición real desde donde fue emitido el tweet más/menos un error, producto de la tecnología empleada para medir esa posición. Esto hace que no sea posible distinguir la fuente de emisión (tren o tierra) de todos los tweets basándose únicamente en su posición. De todas formas el error de la posición es acotado y por lo tanto el universo de tweets a analizar que pueden provenir de un tren también es acotado.

Para poder lograr el objetivo entonces se propone el siguiente supuesto:

“Un segmento de tierra finito tiene una distribución probabilística de cantidad de tweets geolocalizados en su interior en un determinado rango de tiempo. De la misma forma, un tren tiene una distribución probabilística de cantidad de tweets geolocalizados en su interior en un determinado rango de tiempo. Por lo que si se estudia la evolución en el tiempo de dicha distribución probabilística para un segmento de tierra sobre el cual pasa una vía de tren, la misma debería tener un comportamiento distinto dependiendo de si en ese intervalo de tiempo se encontraba un tren pasando por el mismo o no. Cuando pasa un tren, las distribuciones interfieren una con la otra, constructivamente, y cuando no pasa el tren, solo se encuentra la distribución del segmento de tierra.”

Si el supuesto anterior se cumple, bajo determinadas condiciones, es posible distinguir entre instantes de tiempo en el cual se encuentra interferencia de las distribuciones y por lo tanto hay un tren en ese segmento de tierra e instantes de tiempo en el cual no hay tren.

Este es el funcionamiento básico del sistema que pretendemos analizar y a su vez estudiar distintas técnicas para poder determinar el sentido de circulación del tren una vez que es detectado en un segmento de tierra.

De este planteo se desprende que debemos lograr que la gente emita tweets desde el tren y que el contraste con los tweets emitidos desde la tierra sea notable, para poder identificarlo.

Desarrollo del modelo de Simulación

Como se explicó en el planteo de la hipótesis, nuestro sistema busca detectar la presencia de un tren en un segmento de tierra finito, e intenta identificar su sentido. Para lograr esto recolecta tweets emitidos tanto por la tierra como por el tren, que a su vez avanza por un segmento de vía. El éxito del sistema depende de determinadas condiciones bajo las cuales se cumple el supuesto anteriormente planteado. Para determinar esas condiciones y ver el impacto que tiene cada una de las condiciones sobre el éxito, se simula el proceso.

Se buscan simular tres partes:

1. Un tren avanzando por un segmento de vías.
2. Tweets siendo emitidos desde el tren y desde la tierra con su par (posición, tiempo) debidamente calculados.
3. Un sistema de recolección de esos datos.

Cada una de estas partes tiene variables que en conjunto son las condiciones anteriormente comentadas.

Al mismo tiempo parte del algoritmo de decisión se integrará en la simulación para que sea más fácil procesar los datos de salida. Esos datos luego son procesados en un Excel.

Tren

Modelización del sistema de vías

Simulamos cuatro estaciones contiguas de un ramal de una línea de tren. Por las mismas circula un tren por vez. Este tren puede ingresar por cualquiera de los extremos del sistema y avanzar hacia el opuesto. El mismo frena en las estaciones y avanza a un ritmo continuo en los segmentos de vía. Una vez que el tren concluye la espera en la última estación, sale del sistema. El sistema permanece sin trenes dentro de él por un tiempo finito y luego el ciclo comienza de nuevo con un tren ingresando al sistema por alguno de sus extremos.

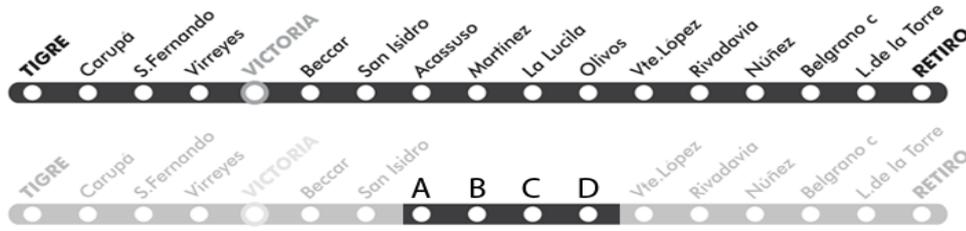


Figura 7: Estaciones a simular (adaptación de información provista por TBA)

Se eligió este esquema para la simulación para poder estudiar el comportamiento del algoritmo en una unidad de análisis general, para poder extrapolar las conclusiones al resto del ramal del recorrido. De todas formas las conclusiones quedan excluidas de los extremos del ramal, ya que allí el comportamiento de los trenes es distinto y por lo tanto el comportamiento del algoritmo en dichos extremos no es predecible a partir de este modelo.

Se elige simular cuatro estaciones y tres segmentos de vía, para así poder analizar el segmento de vía interior. Este segmento permite sacar conclusiones generales, ya que un tren que transita por él debe, necesariamente, haber transitado por otro segmento de vía contiguo y haber esperado en dos estaciones. Esto implica que ese tren tiene historia en el sistema y esa información es fundamental para contrastar el resultado del algoritmo con la realidad simulada.

Para poder llevar adelante la simulación se definen los siguientes parámetros:

- distancia(X,Y) = distancia entre dos estaciones contiguas X e Y [kilómetros].
- espera(Z) = tiempo de espera en una estación Z [segundos].
- desplazamiento(a,b) = tiempo necesario para recorrer el tramo de vías entre estaciones contiguas a y b a una velocidad constante [segundos].

Para la distancia (X,Y) se toma 1 km para todo par (X,Y). Es importante que el error de la geolocalización de los tweets emitidos desde el tren tiene de radio 1 km. Ya que vamos a estar midiendo tweets en el segmento intermedio de vías cuyos extremos están a 1 km del fin del sistema, la elección toma sentido. Esta decisión también implica simplicidad programática, la cual queda expuesta cuando se explica el sistema de coordenadas elegido para simular. El ramal retiro-tigre de la línea mitre tiene la siguiente distancia entre estaciones:

Station	Retiro	Lisandro de la Torre	Belgrano	Núñez	Rivadavia	Vicente Lopez	Olivos	La Lucila	
Km	0	1.7	2.4	2.9	3.6	5.4	6.3	7	
Estación	Martínez	Acassuso	San Isidro	Beccar	Victoria	Virreyes	San Fernando	Carupa	Tigre
Km	7.9	9.2	10	10.9	11.9	14.6	16.5	18.7	23.5

Tabla 2: Posición de las estaciones

Lo cual se traduce a una distancia promedio de 1.45 km y una mediana de 0.95 km, por lo que elegir una distancia estándar entre estaciones de 1 km se ajusta en cierta medida con la realidad del objeto de estudio.

El objetivo en esta parte de la simulación es imitar el movimiento del tren a lo largo de un tramo de su recorrido, ajustándose lo mejor posible a la realidad pero simplificando algunos aspectos para facilitar el estudio del impacto de su comportamiento.

Modelización del movimiento del tren

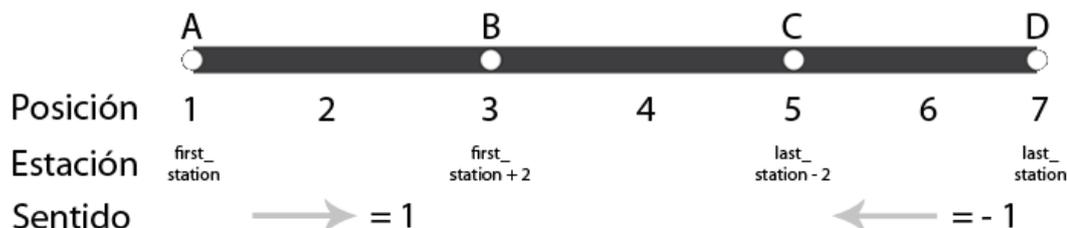


Figura 8: Sistema de Coordenadas del modelo

Como debemos simular el avance de un tren por un sistema de vías y estaciones, mientras se va almacenado el par (posición, tiempo) del tren, bajo determinadas circunstancias, se propone el sistema de coordenadas esquematizado en la figura.

Esto quiere decir que las estaciones tendrán posiciones impares y los segmentos de vía tendrán posiciones pares. Un tren avanzando de la estación A, hacia la B, tiene sentido positivo y su posición va de 1 a 3 mientras que un tren avanzando de B, hacia A, tiene sentido negativo (-1) y su posición disminuye de 3 a 1.

Implementación en Arena

El circuito que creamos en Arena para simular el recorrido del tren es el siguiente:

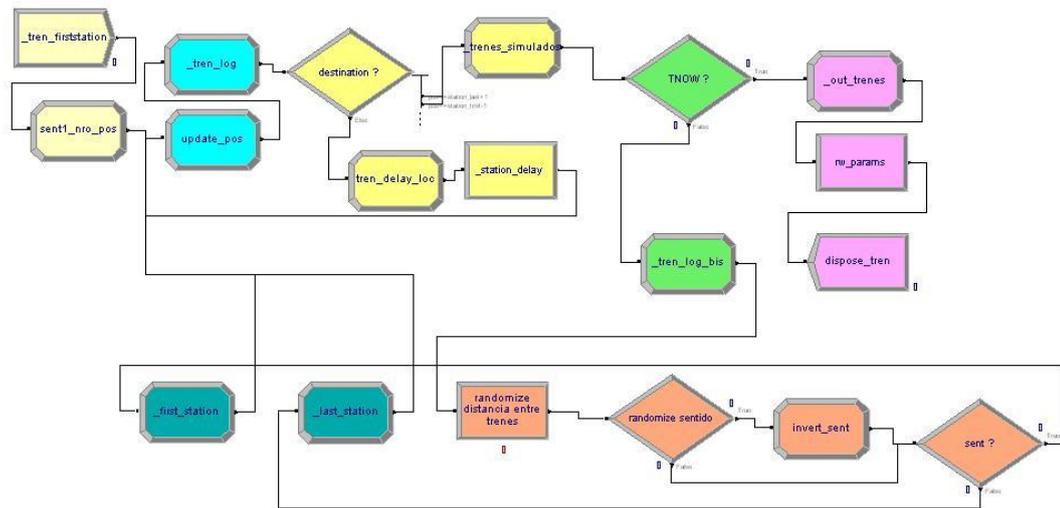


Figura 9: Recorrido del tren en Arena

Una entidad tren entra al circuito en el momento cero de la simulación. En la simulación se trata siempre de la misma entidad, pero se le asigna un número de identificación diferente cada vez que recorre todo el tramo de estudio (1 a 8). El tiempo que transcurre en las estaciones y recorriendo las vías está modelizado con demoras.

Variables

El tren puede adoptar una serie limitada de posiciones durante la simulación.

POS	Representa
-999	Una posición fuera del sistema
station_first-1	El estadio anterior a ingresar al sistema
station_first	La primera estación. Donde el tren inicia su recorrido y empieza a emitir tweets
station_first+1	El primer tramo de vía
station_last-1	El último tramo de vía
station_last	La última estación del recorrido desde la cual el tren puede emitir tweets
station_last+1	El estadio inmediatamente posterior a que un tren salga del sistema
[station_first,station_last]	Todo el recorrido del tren, alternando tramos de vía y estaciones

Tabla 3: Posiciones del tren en el modelo

En la mayoría de las corridas utilizamos station_first = 1 y station_last = 7. Esto nos permite tener 4 estaciones con tres tramos de vías intermedios. Utilizando estos valores, el tren creado arrancararía en la estación 0, inmediatamente ingresa a la estación 1 y esperaría un tiempo t determinado por el delay en esa estación. Luego ingresaría al tramo 2 y esperaría el tiempo correspondiente a ese tramo (lo que

representa el tren avanzando por las vías). Así repetiría su proceso hasta llegar a la estación 7, donde una vez concluida su espera sería sacado del circuito al pasar a la estación 8, donde luego se le asigna la $pos = -999$

Módulos

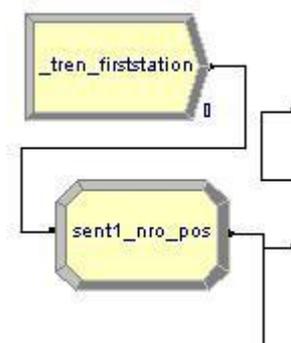


Figura 10: Módulo del modelo

El primer módulo de la simulación es un *create* que introduce la única entidad tren en el circuito. Antes de que la entidad ingrese al circuito de estaciones y tramos de vía se le asigna los atributos posición inicial y número de entidad tren. De esta forma, la entidad ingresa al circuito con la posición $station_first-1$ y el número de identificación 1 representado por el atributo *emit*. Al tener estas variables definidas la entidad tren está lista para ser registrada y recorrer el circuito.

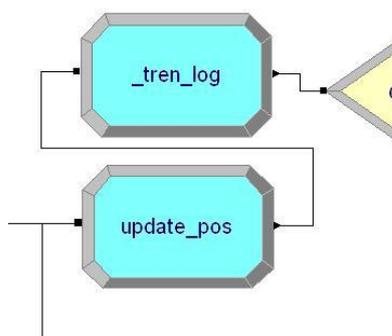


Figura 11: Módulo del modelo

Una vez ingresada al sistema la entidad pasa por dos módulos *assign* llamados *update_pos* y *_tren_log*. En el primero se le actualiza la posición, sumándole el sentido a la posición que ya tenía y en el segundo se asienta en una tabla la posición nueva y el sentido de circulación. Durante el primer recorrido el tren tiene automáticamente la dirección positiva, que va de la estación A a la D. La misma luego se actualiza cada vez que el tren sale del tramo de 4 estaciones.

A continuación, la entidad tren ingresa a un módulo *decide* en donde según la posición que tenga asignada permanecerá o no dentro del circuito. En el caso que tenga una posición mayor a la correspondiente a la última estación (estación D) o menor a la

correspondiente a la primera (estación A), la entidad irá al módulo `_trenes_simulados` donde se aumenta el conteo de trenes simulados y se actualizan algunas variables necesarias para la simulación en su conjunto. Caso contrario, la entidad se moverá hacia el módulo `tren_delay_loc` donde se le asignará un tiempo de demora para el próximo proceso en función de si se encuentra en una estación o un tramo de vías. Luego irá al siguiente módulo del tipo *delay* llamado `_station_delay`, donde permanecerá un lapso de tiempo que representa el tiempo frenado en la estación o de traslado en las vías.

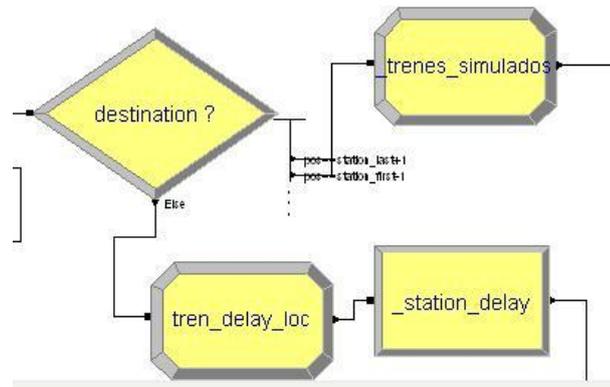


Figura 12: Módulo del modelo

De seguir dentro del circuito, la entidad pasará del módulo `_station_delay` al `update_pos` mencionado anteriormente donde se le asignará su nueva posición y hará nuevamente el recorrido descrito hasta aquí. De esta forma la entidad recorrerá el loop formado por los módulos `update_pos`, `_tren_log`, `destination?`, `tren_delay_loc` y `_station_delay` representando cada vuelta el avance del tren desde una estación a un tramo de vías o de un tramo de vías a una estación hasta que acabe el recorrido.

En el caso de que el tren salga del tramo, la entidad pasará por un módulo “decide” llamado `TNOW?` en el cual si el tiempo total transcurrido es mayor al tiempo límite de la simulación la entidad comenzará el recorrido hacia su salida definitiva de la simulación, situación que se explicará más adelante.

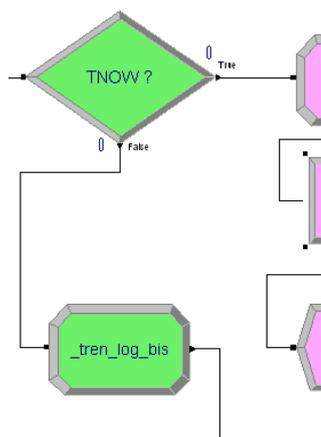


Figura 13: Módulo del modelo

Si aún no transcurrió el tiempo total de la simulación, la entidad pasará al módulo `_tren_log_bis`, donde se le asignará la posición `-999`, correspondiente al tren fuera del tramo de circulación, y se limpiará el valor de la variable `sent`.

Estando la entidad tren fuera del tramo de circulación se debe volver a introducir la misma en el tramo, esperando un tiempo al azar e ingresando por cualquiera de las dos estaciones de las puntas. Para modelizar esto se utilizan los siguientes módulos:

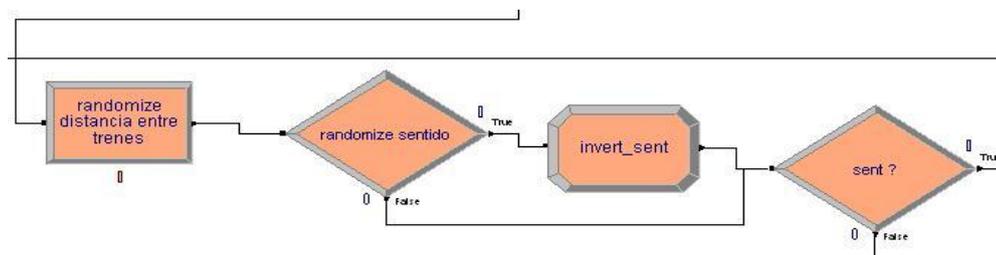


Figura 14: Módulo del modelo

El módulo `randomize distancia entre trenes` es del tipo “delay” y determina el tiempo que pasa entre la salida de un tren del tramo de estaciones y la entrada del siguiente. Este tiempo es una variable aleatoria con distribución uniforme que toma valores entre 0 y 20 minutos. `Randomize sentido` es un módulo del tipo decide en el que la salida se define aleatoriamente con un 50% de probabilidades de salir por cada rama. En una salida la entidad pasa por un módulo “Assign” llamado `invert_sent` en el cual la variable `sent` se multiplica por `-1` invirtiendo el sentido de circulación del tren. De la otra salida se va directamente al módulo “decide” llamado `sent?` en el cual dependiendo del valor de la variable `sent`, se va a la primera o la última estación del tramo para recorrerlo nuevamente.

Una vez que se definió el sentido de circulación del tren simulado a reingresar en el circuito, la entidad ingresara a uno de los siguientes módulos:

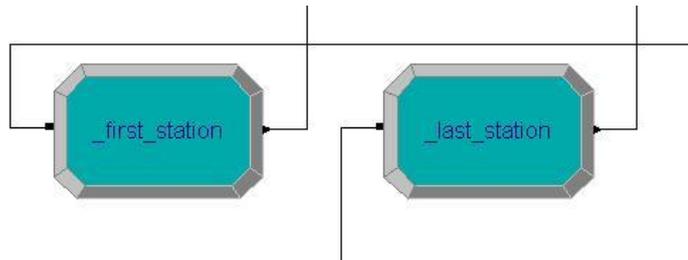


Figura 15: Módulo del modelo

El módulo `_first_station` le atribuye a la entidad la posición `station_first-1` que es la inmediatamente anterior a la posición 1 correspondiente a la primera estación. En caso de que el sentido del tren sea el contrario, la entidad entrará al módulo `_last_station` en el cuál se le atribuirá la posición `station_last+1`, correspondiente a la posición inmediatamente posterior a la última estación (posición 8). Habiendo adquirido la nueva posición la entidad vuelve a entrar al circuito que simula el paso por las estaciones, dirigiéndose al módulo `update_pos`.

Este proceso se lleva a cabo durante un tiempo determinado, hasta que se agota el tiempo establecido como tiempo límite de la simulación. Esta ocurrencia se detecta en el módulo de tipo *decide*, descrito anteriormente, llamado `TNOW?`

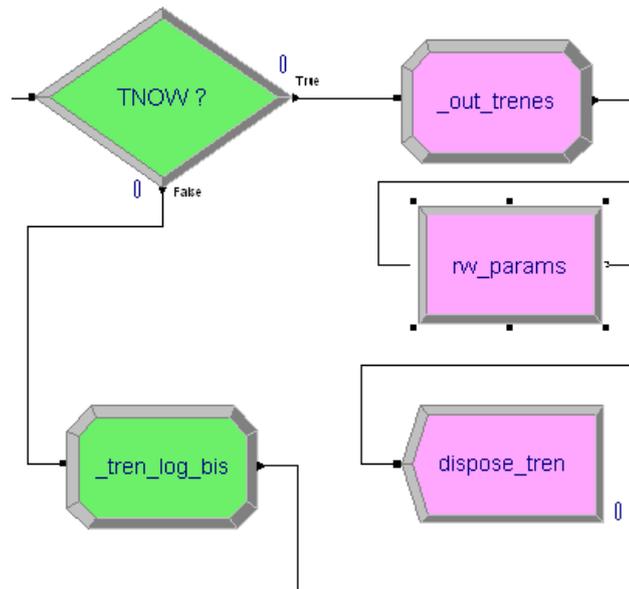


Figura 16: Módulo del modelo

Transcurrido el tiempo de la simulación la entidad pasa al módulo `_out_trenes` donde se le quita el valor a la variable `sent` (que indica el sentido de circulación del tren pudiendo ser 1 o -1) y se le asigna un valor unitario a la variable `trenes_out`. Luego pasa por el módulo "ReadWrite" llamado `rw_params` donde se graba el valor de todas las variables y finalmente llega a un módulo "Dispose" donde es eliminada.

Tweets

Modelización de la generación de Tweets

Habiendo esquematizado el movimiento del tren en el tramo bajo estudio, resta modelizar la generación de tweets tanto desde el tren como desde el suelo. La realidad presenta la particularidad que los tweets son generados por personas que o se encuentran en el tren o se encuentran en la tierra. El conjunto de personas que twitteen desde la tierra o desde el tren crean una distribución de tweets emitidos. Con eso en mente, la generación de tweets es modelada suponiendo que hay solo dos posibles emisores de tweets, la tierra y el tren, y que cada uno tiene una distribución aleatoria particular. Entonces lo que se hace es simular porciones de tierra y trenes en movimiento que emiten tweets acorde a su distribución asignada. Cada tweet emitido debe estar acompañado por su par (tiempo, posición) y ese es parte del desafío del modelo elegido. La tierra tiene posición acotada, mientras que el tren puede estar en distintas partes de un tramo de vías, dependiendo del momento. El modelo se encarga de asignarle a cada tweet emitido desde cualquiera de los dos entes generadores la posición acorde. El tiempo, que completa el par, viene dado por el momento en que fue emitido, que es función de la distribución aleatoria.

Análisis estadístico de la situación actual

Se realiza un análisis empírico estadístico para poder saber cómo se comportan los Tweets. En otras palabras, qué distribución estadística es la que mejor se ajusta y qué media/desvío tienen para poder así modelar y simular los Tweets a futuro. Para que los Tweets se ajusten lo mejor posible a la realidad, debemos encontrar y fundamentar la distribución que mejor se ajuste a los datos que hoy en día aparecen en Twitter.

Vamos a empezar por describir los datos que Twitter nos provee para un cierto tramo de vía:

Los datos vienen como ya los hemos descrito previamente y la fuente completa para este caso se encuentra en el *Apéndice Muestreo de Tiempo entre Tweets*. En esta sección vamos a mostrar su posición, y a utilizar el momento con día, hora, minutos y segundos para realizar el ajuste estadístico.

Su posición según Twitter (incluye al error previamente mencionado):

Estos son datos reales generados durante 3 días diferentes para un tramo de vía.

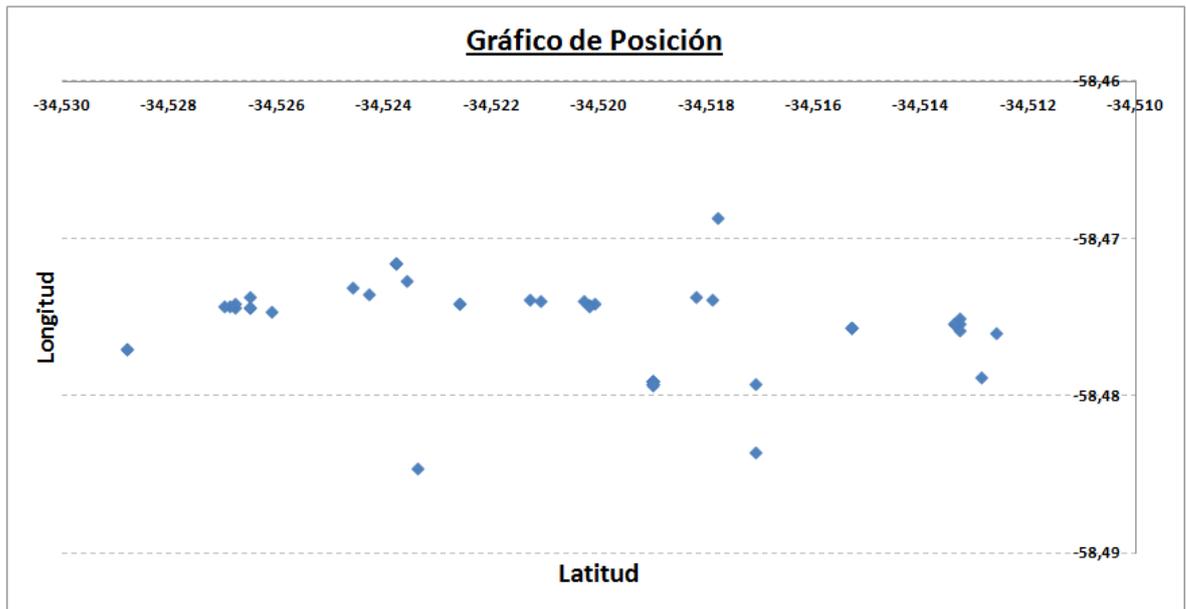


Figura 17: Distribución de Tweets en el espacio.

La variable que se pondrá en tela de juicio será el tiempo entre Twitteos debido a que la localización nos es un dato previamente calculado. Para conocer el comportamiento de los Tweets, se realiza un histograma tomando intervalos de 60 segundos, quedando de la siguiente forma:

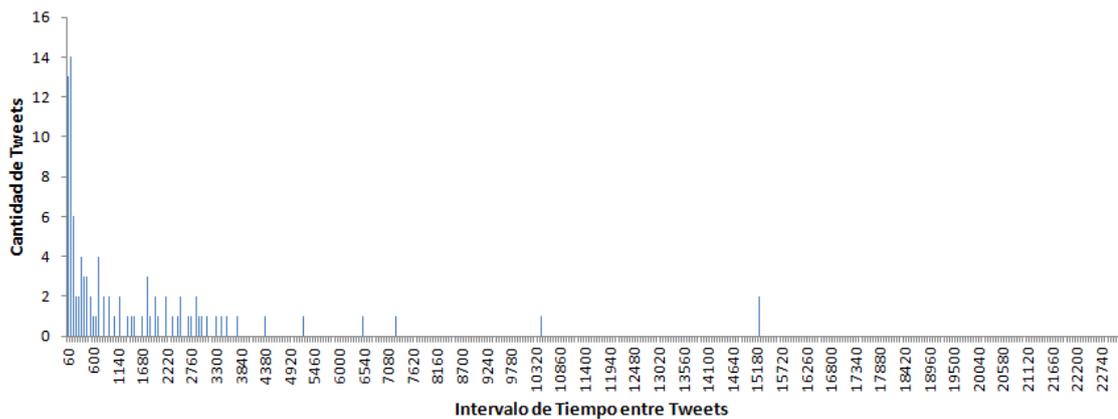


Figura 18: Tiempo entre Tweets (segundos)

Podemos notar a simple vista como la gran mayoría de los Tweets se acumulan por debajo de los 10 minutos (600 segundos) teniendo también valores más altos cuando tomamos un intervalo más próximo a cero.

Una vez que ya conocemos los datos, procedemos a realizar el análisis estadístico de los datos. Para ello se utiliza la herramienta de análisis de datos SPSS (ver anexo) que nos devolverá la distribución a la cual los datos mejor se ajustan. Luego, para estar seguros de que el resultado teórico se ajusta también a la práctica, se realizará un “Q-Q Plot” (ver anexo) de los datos sobre la distribución.

Los resultados muestran lo que ya se podía sospechar: que los testeos tienen una distribución exponencial (*ver anexo*). Lo curioso sucede al ver la media, que está en 30 minutos. Para poder entender mejor el porqué del resultado, se proceden a mostrar algunos de los gráficos más destacados.

Realizando un “Box plot” (*ver anexo*) podemos entender por qué la media nos da tanto mayor de lo que se esperaba, y es porque a pesar de tener todos los datos acumulados en los primeros intervalos de tiempo, también aparecen algunos espacios entre tweets de más de 6 horas que igualmente son parte de la muestra real de testeos.



Figura 19: Box Plot

Podemos ver cómo 5 valores caen por fuera, y que algunos de éstos están a una distancia considerable, por ende, la media se tiende a ser mayor.

Por otro lado, el Q-Q plot entrega los siguientes resultados:

Gráfico Q-Q Exponencial de VAR00001

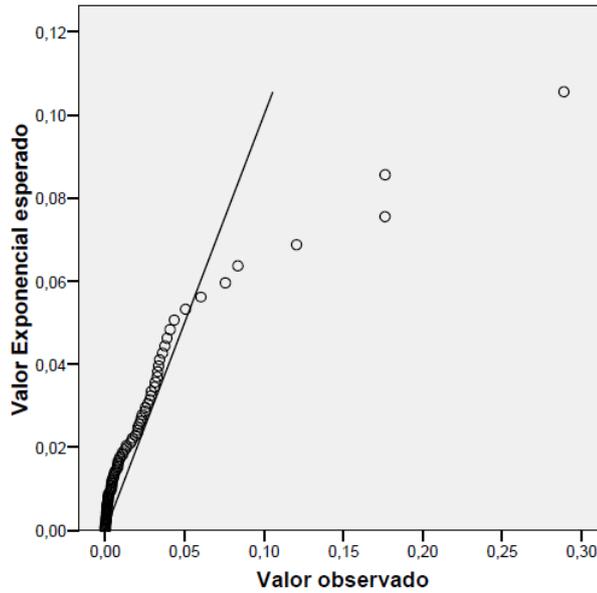


Figura 20: QQ Plot

Aquí podemos ver cómo la gran mayoría de puntos está sobre la recta, demostrando lo bien que se ajusta la muestra de Twitteos a una distribución exponencial descrita previamente. Esto avala los resultados anteriores.

Además, se realiza un “Stem-and-Leaf Plot” (*ver anexo*) para tener una razón más para estar seguros de que la exponencial es la mejor distribución.

```
VAR00001 Stem-and-Leaf Plot
Frequency      Stem & Leaf
    56,00      0 . 0000000001111112344456789
     9,00      1 . 3&&&
    13,00      2 . 0258&&
     9,00      3 . 13&&
     2,00      4 . &
     1,00      5 . &
     1,00      6 . &
     6,00 Extremes      (>=,076)
```

```
Stem width:  ,0100000
Each leaf:   2 case(s)
```

& denotes fractional leaves.

Figura 21: Stem-and-Leaf Plot - Frecuencia

En este caso, vemos algo muy similar a lo que veíamos con el histograma, que la cantidad de datos está distribuida de tal forma que se asemeja a una distribución exponencial.

Por último, para asegurarnos de que los números no se equivocan, contrastamos contra la teoría misma de la distribución exponencial para ver si la muestra de tweets puede caer dentro de este tipo de distribución probabilística. Como bien sabemos, la distribución exponencial se utiliza para describir “el tiempo hasta que se produce un determinado suceso, donde cada suceso es independiente del otro”, descripción que encaja perfectamente con la clase de datos que tenemos en nuestra muestra.

Por todas las razones descritas previamente, podemos estar seguros de que el tiempo entre testeos sigue una distribución exponencial y que ésta tiene una media de 30 minutos.

Implementación en Arena

Para llevar a cabo dicho proceso construimos el siguiente circuito en Arena:

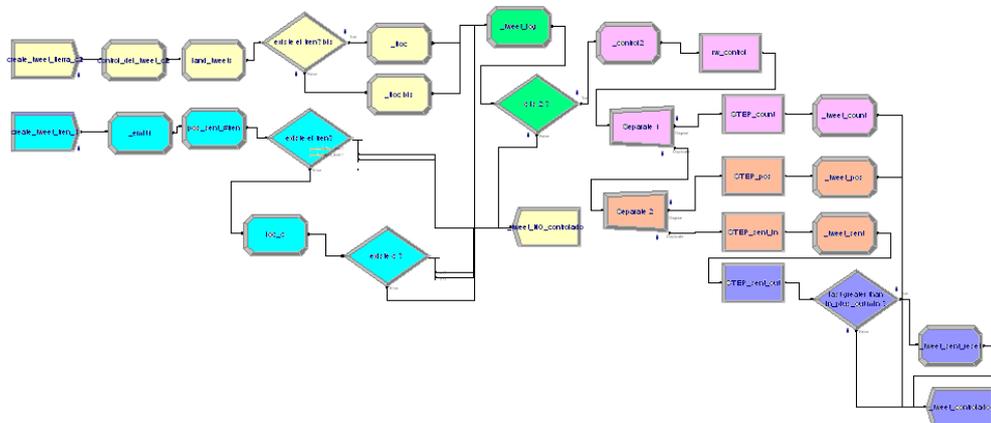


Figura 22: Modelo de generación de Tweets

Como se puede observar en el esquema, existen dos módulos “create” que generan entidades. La que se encuentra más arriba genera las entidades que representan tweets generados desde tierra que forman parte de lo denominado ruido.

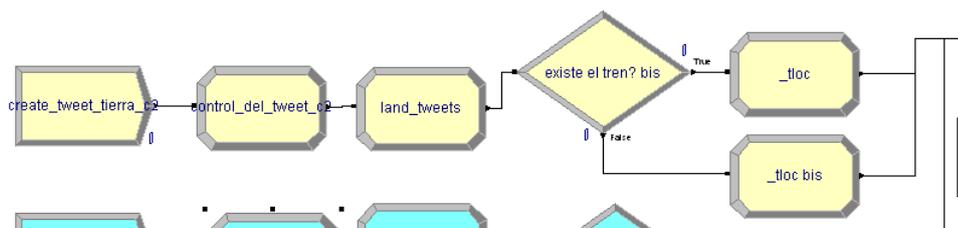


Figura 23: Módulo del modelo

El módulo llamado `create_tweet_tierra_cs` genera los tweets enviados desde tierra que caen dentro de nuestro puesto de control bajo estudio, el C2. Los genera de forma aleatoria siguiendo una distribución de probabilidad exponencial con una constante determinada. Las entidades generadas luego van a un `assign` donde se le atribuye el valor 2 a la variable `c`, significando que el tweet cayó dentro del área de control del puesto 2. Acto seguido pasan por `land_tweets`, otro módulo `assign` que asigna a la entidad un conjunto de atributos como posición del tren al momento de caer el tweet y locación del tweet, dada por la posición del mismo más el error que abarca un kilómetro de distancia en ambos sentidos. Luego entra en un “decide” en el que si el tren está presente en el momento, la entidad va a módulo `_tloc` donde se le atribuye la locación exacta del tren, dada por la siguiente fórmula:

$$\text{tloc} = \text{tpos} - 1 * \text{vias}(\text{tpos}) * \text{sent} + ((\text{TNOW} - \text{tren_log}(1,3)) / \text{tren_log}(1,4)) * \text{vias}(\text{tpos}) * \text{sent} * 2$$

El primer término de la ecuación, `tpos` representa la posición del tren que puede ser una estación o un tramo de vías. El segundo término multiplica la variable `vias(tpos)`, que puede tomar valor 1 si el tren está en las vías o 0 si el tren está en la estación, por el sentido. De esta forma si el tren se encuentra circulando en sentido positivo en un tramo de vías, al estarse restando de la posición, el término le restará una unidad al valor `tpos`, haciendo que en cualquier caso el tren figure en la estación en la que se encuentra o la inmediatamente anterior cuando está viajando. El último término de la ecuación tiene como objetivo sumar, en el caso de que el tren esté viajando de una estación a la otra, el tramo recorrido. Al estar multiplicada la variable `vias(tpos)` y el sentido, la locación del tren se ubicará más adelante (o más atrás según el sentido del avance) solo en el caso en que el tren no esté en la estación. La parte del término determina por $((\text{TNOW} - \text{tren_log}(1,3)) / \text{tren_log}(1,4))$ representa la proporción del tramo entre estaciones recorrido al caer el tweet. Por simplicidad consideramos que el tren viaja a la misma velocidad durante todo el tramo sin acelerar ni frenar en ningún momento, resultando en una trayectoria lineal entre estaciones. La resta $((\text{TNOW} - \text{tren_log}(1,3))$ determina el tiempo que transcurrió entre que el tren dejó la estación y el presente y el valor `tren_log(1,4)` es el tiempo total que tarda el tren en ir desde la estación predecesora a la siguiente. Si el tren acaba de salir de la estación la diferencia entre `TNOW` y `tren_log(1,3)` será mínima, haciendo que el término tome un valor cercano a 0. Si el tren está a punto de llegar a la estación siguiente, la diferencia será cercana a `tren_log(1,4)` haciendo que el valor del término sea cercano a 1.

Si el tren no se encuentra dentro del tramo en estudio la entidad pasa por el bloque `_tloc bis`, donde se le asigna la posición del tren fuera del tramo.

Simultáneamente, otro módulo “create” genera entidades representando los tweets que se emiten desde el tren.

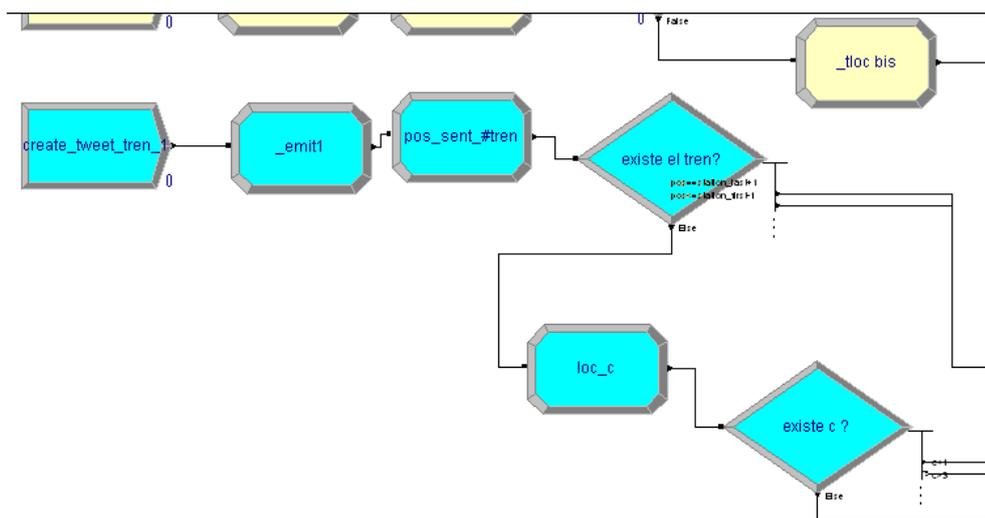


Figura 24: Módulo del modelo

Al ser generada, la entidad se dirige al módulo `_emit1` donde se le asigna un número de identificación. Inmediatamente después entra al bloque `pos_sent_#tren`, donde se le atribuye la posición, el sentido y el número del tren desde el cuál se emitió. Desde ahí la entidad se dirige al “decide” llamado `existe el tren?` en el cuál lógicamente se descarta si el tren no se encuentra dentro del tramo de vías estudiado. En caso de que exista el tren, a la entidad tweet se le asigna en `loc_c` la locación exacta donde cayó. Esto se logra determinando la posición exacta del tren en el momento en que partió el tweet haciendo uso de la ecuación del recuadro descripta anteriormente y sumando un desvío aleatorio determinado por $NORM(0,2*(error/3))$. De esta forma queda asignada la posición exacta donde cayó el tweet representada por la variable `loc`. También se define en qué puesto de control cayó el tweet, dado por `c`. Si la posición de `c` no coincide con alguna de las de nuestro estudio (1,2 y 3), la entidad automáticamente se descarta, caso contrario, habiendo adquirido los atributos mencionados, la entidad se mezcla con las provenientes desde la tierra y juntas ingresan en la parte del modelo en la que se aplican los algoritmos para el control.

Procesamiento de datos

Implementación en Arena

En el módulo “assign” llamado `_tweet_log` como bien lo menciona el nombre se le carga a la entidad tweet una serie de valores que más tarde se usarán en el análisis. Estas variables son el tiempo de arribo al circuito, la localización del tren, y el sentido en el que avanza el mismo. Una vez que se le grabaron estos valores, a la entidad se la hace pasar por un “decide”. Si el valor de `c` es igual a 2, la entidad prosigue con su recorrido. Si no es 2 significa que el tweet cayó fuera del rango que abarca el puesto de control en estudio y se elimina inmediatamente.

Luego la entidad tweet llega a un módulo assign llamado `_control2` el cual actúa como un contador. Cada vez que pasa un elemento por ahí el contador sube en una unidad y al mismo tiempo se le atribuye como número identificador a la entidad.

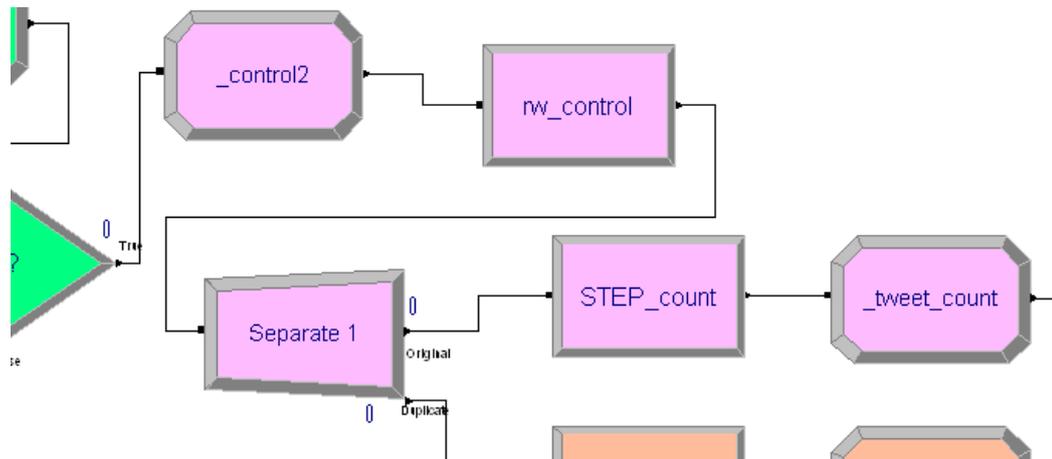


Figura 25: Módulo del modelo

Una vez que pasa por `_control2` la entidad llega al módulo de tipo “readwrite” en el que se vuelca toda la información recolectada hasta el momento en un documento de Excel. Luego se llega al primer bloque “separate” en el que se duplica la entidad y se hace salir una por cada lado. El objetivo de duplicar y separar los tweets es poder agrupar los mismos de diferentes maneras para obtener datos distintos sobre el mismo tren. En el primer caso las entidades avanzan hacia el módulo “delay” llamado `STEP_count` en el que se demoran un tiempo determinado antes de pasar al bloque `_tweet_count`. Esta demora, que es la primera de todo el circuito, hace que los datos se agrupen durante el período de tiempo que dura el `STEP` para luego pasar por `_tweet_count` que actúa como si fuera una salida del rango de detección del algoritmo ya que hace que el conteo de tweets disminuya en una unidad. De esta forma cada vez que se pide información al sistema para analizarla uno de los datos de salida será el conteo de tweets durante el lapso de duración de `STEP_count`.

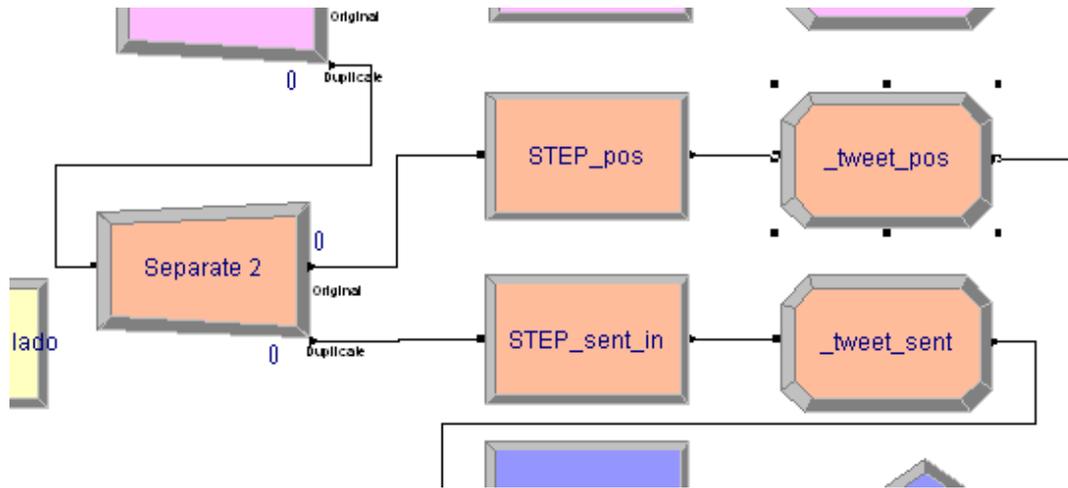


Figura 26: Módulo del modelo

La entidad duplicada que no se dirigió hacia el bloque STEP_count, seguirá su camino hacia el módulo Separate 2 donde se volverá a duplicar y separar en dos caminos diferentes. De esta forma se logrará tener la misma entidad con los mismos atributos avanzando sobre tres “delays” diferentes para ser analizadas de distinta forma. Los módulos STEP_pos y STEP_sent_in harán que cada consulta al sistema arroje resultados con una agrupación de tweets determinada. La diferencia entre los tipos de agrupación de tweets está en el delay que se le da a las entidades antes de salir del sistema. El módulo STEP_count es útil para determinar si se detecta tren o no ya que agrupa los tweets de los últimos 2 minutos. La segunda correspondiente a STEP_pos sirve para determinar la posición del tren en el caso de ser detectado, agrupando un menor número de tweets (aquellos de los últimos 30 segundos) para no tomar en cuenta tweets que se generaron muy lejos de la posición actual, mientras que la última determinada por STEP_sent_in y luego STEP_sent_out se utiliza para detectar el sentido de circulación de la formación. El módulo _tweet_pos actúa de la misma forma que el _tweet_count, desagrupando la entidad que pase por el mismo.

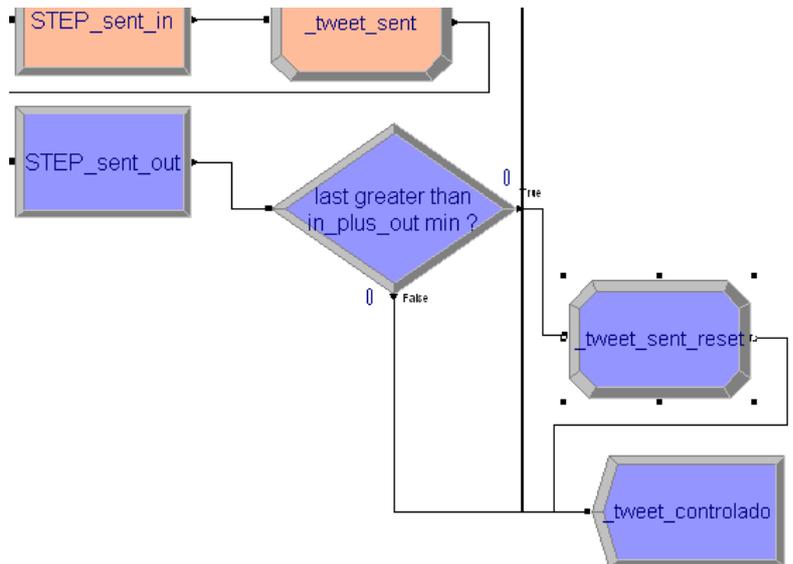


Figura 27: Módulo del modelo

Para determinar el sentido de circulación del tren se deben obtener dos datos por consulta para compararlos y llegar a una conclusión. Para lograr eso se ubicaron dos “delays” en serie que agrupan tweets en tiempos diferentes. El módulo STEP_sent_out corresponde cronológicamente a la última agrupación. Luego en el “decide” llamado last greater than in_plus_out min ? se compara el número de identificación de cada grupo de tweets y en caso de ser iguales se resetea el valor del sentido de circulación del tren en el bloque _tweet_sent_reset. Finalmente ambas entidades se dirigen al módulo “dispose” donde son eliminadas del circuito.

Implementación en Excel

La simulación devuelve en un Excel dos tipos de información. Uno representa las variables de entradas del modelo y el otro el resultado de la simulación.

Variables de Entrada

1. error = error de geo localización de los tweets [km]
2. tweets_tren = tiempo entre tweets emitidos desde el tren [segundos]
3. tweets_tierra = tiempo entre tweets emitidos desde tierra [segundos]
4. step_count_length = periodo de tiempo en el cual se agregan los tweets a un contador para detectar la presencia del tren [segundos]
5. step_pos_length = periodo de tiempo en el cual se agregan los tweets a un contador para determinar la posición del tren [segundos]
6. step_sent_length_in = distancia mínima en tiempo, entre dos tweets para que se puedan utilizar para detectar el sentido de un tren [segundos]

7. `step_sent_length_out` = distancia máxima en tiempo entre dos tweets para que se puedan utilizar para detectar el sentido de un tren [segundos]
8. `station_first` = primer estación a simular.
9. `station_last` = última estación a simular.
10. `trenes_create1` = entidades que representan trenes a crear con sentido positivo al comienzo de la simulación.
11. `trenes_create2` = entidades que representan trenes a crear con sentido negativo al comienzo de la simulación.
12. `tiempo_final` = tiempo máximo hasta el cual el sistema seguirá ingresando los trenes al tramo en estudio.
13. `trenes_simulados` = cantidad de trenes que circularon por el tramo de estudio a lo largo de la simulación. (Esta variable es esclava del `tiempo_final`)
14. `count_limite` = límite de tweets a contar en un step para indicar presencia de tren en ese instante.
15. `pos_limite` = límite de tweets a contar en un step para predecir la posición del tren en ese instante.

Estas variables de entrada son suficientes para describir el modelo o situación simulada.

Variables de Salida

Datos a procesar

1. `c2_id` = número de tweet en ingresar al puesto de control 2.
2. `t_create` = tiempo en el cual fue emitido.
3. `tloc` = localización del tren cuando ese tweet fue emitido
4. `sentido` = sentido del tren cuando ese tweet fue emitido
5. `count_log(c,1)` = conteo de tweets en el control 2 en ese instante.
6. `pos_log(c,1)` = posición promedio de los tweets en el contador en ese instante.
7. `pos_log(c,2)` = cantidad de tweets en ese contador en ese instante.
8. `cent_log(c,1)` = posición del tweet más antiguo en el contador.
9. `cent_log(c,2)` = posición del tweet más reciente en el contador.

10. $\text{cent_log}(c,3)$ = tiempo de emisión del tweet más antiguo en el contador.
11. $\text{cent_log}(c,4)$ = tiempo de emisión del tweet más reciente en el contador.
12. #tren = referencia para identificar el tren que circulaba en ese momento por el sistema.

Con toda esta información se procede a calcular los indicadores de éxito.

Indicadores de Éxito

1. $\text{count_log} > \text{count_limite}$ - indica si en ese instante el contador en el puesto de control es mayor que un límite pre-establecido. En caso de ser afirmativo, indicaría presencia del tren.
2. Avg. Tweet Pos - Tren Pos - indica la diferencia en posiciones entre donde detectamos el tren y donde estaba en un instante dado. La detección del tren se construye en base a el promedio de la ubicación de X cantidad de tweets en Y cantidad de segundos. Número positivos indica que el tren esta por detrás de nuestra predicción y números negativos indican que el tren está por delante de nuestra predicción.
3. Slope, 2 tw separated by X sec ($t_0 > X > t_1$) - devuelve la pendiente de una regresión lineal realizada entre dos tweets separados x segundos. Si la pendiente es positiva indica que el tren avanza. Por el contrario, si la pendiente es negativa, indica que el tren retrocede.
4. $\text{slope} * \text{sent}$ - multiplica la pendiente de la regresión por el sentido real del tren. En caso de que este número fuese negativo indica que el sentido de slope es distinto al sentido del tren y por lo tanto la predicción está equivocada.
5. Por último se construye un indicador por #tren, el cual informa para cada #tren, si se lo detecto, si se analizó su posición y si se predijo su sentido. Analiza esas tres instancias por separado y contabiliza la cantidad de instantes en los cuales se logró predecir las tres cosas en simultáneo.

Con estas variables de salida del sistema se construyen los KPI's del algoritmo.

Key Performance Indicator

1. **%count** - % de trenes que fueron detectados
2. **%pos** - % de trenes a los cuales se les identificó una posición
3. **%sent** - % de trenes a los cuales se les identificó un sentido
4. **%detectados** - % de trenes en los cuales existe un instante en el que simultáneamente se detectó un tren, su posición y su sentido.

5. **wrong_sent** - % de instantes en los que se identificó un sentido y este estaba equivocado.
6. **gráfico de distribución del error de la predicción de la posición del tren** - Error en metros (lo que anteriormente se definió como *Avg. Tweet Pos - Tren Pos*, que indica la diferencia en posiciones entre donde detectamos el tren y donde estaba en un instante dado) vs Cantidad (de “evaluaciones” que devolvieron ese resultado en abscisas, ya que cada vez que se detecta un tweet, se inicia el análisis que termina devolviendo ese parámetro de error.)

Experimentación

Bases

Se diseñaron una serie de experimentos para poder analizar el comportamiento de nuestro algoritmo predictor y del sistema en general a partir de la variación de ciertos parámetros. Se procedió buscando tres niveles de entendimiento:

1. Apreciación general de la respuesta del sistema a distintos impulsos.
2. Seteo de las variables del algoritmo para optimizar sus resultados.
3. Conclusiones generales, buscando identificar factores de éxito.

En particular se responden las siguientes preguntas:

- ¿Cómo debo setear las variables del algoritmo para optimizar sus resultados bajo una situación real particular? Aquí se trabaja con las variables *step_count_length*, *step_pos_length*, *step_sent_in*, *step_sent_out*, *pos_limite* y *count_limite*.
- ¿Cuántos tweets debe emitir un tren para que el sistema funcione? Esta es una pregunta compleja ya que tiene muchas partes. Primero se debe definir que es *funcionar* y lo hacemos en base a los KPI antes comentados. Luego se intenta responder la pregunta directamente, para lo que se determinaron las condiciones de ruido y demás realidades al día de hoy.

Parámetros comunes

Todos los experimentos se realizaron bajo las mismas premisas básicas. Es decir, hay ciertas variables de entrada y condiciones de simulación comunes a todos ellos. A continuación se resumen estas:

Station_first	Station_last	Trenes_create1	Trenes_create2
1	7	1	0

Tabla 4: Parámetros comunes

Como se explicó anteriormente, esto quiere decir que habrá solo un tren en el sistema por vez, que arrancara con sentido positivo. El sistema estará compuesto por 4 estaciones y 3 tramos de vías entre las estaciones. El objeto de estudio será el puesto de control ubicado en el tramo de vía #2, entre la estación #2 y la #3. Esta posición en nuestro sistema se denomina station #4 y a la cual también se hace referencia indistintamente como puesto de control #2.

A continuación, en la sección Resultados y Análisis, se exponen los distintos experimentos que llevan adelante los análisis recién mencionados. Se verá para cada experimento, que se resumen las variables de entrada utilizadas en el modelo de Arena para correr la simulación, y a continuación se adjunta el gráfico de la distribución del error en la predicción de la posición del tren.

Resultados y Análisis

Control

Se simulo una situación de control para poder contrastar los resultados del resto de los experimentos. Se utilizaron condiciones denominadas *perfectas*. Es decir, sin ruido, sin error en la geo localización de los tweets y con mínimo tiempo entre tweets emitidos desde el tren.

Variables de entrada

#corrida	error_tweet [Km]	tweets_tren [s]	tweets_tierra [s]	step_count_length [s]	step_pos_length [s]	step_sent_length_in [s]	step_sent_length_out [s]	tiempo_final	trenes_simulados
0	-	0.5	-	10	10	30	10	65	198

Tabla 5: Variables de entrada

Resultado

#corrida	%detectados	%wrong_sent
0	100	0

Tabla 6: Resultados

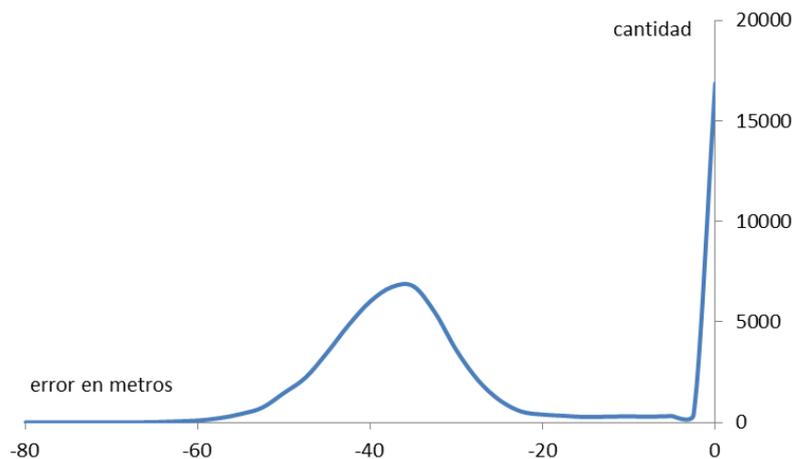


Figura 28: Distribución del error del experimento

Análisis

De los resultados se observa que el algoritmo tiene error en la predicción de la posición inclusive bajo condiciones perfectas. Esta es consecuencia de las decisiones de diseño. Para calcular la posición del tren se promedia la posición de las señales emitidas por el mismo, en una ventana de tiempo, `step_length_pos`. Por lo que si el tren se encuentra en movimiento la posición inicial y la final, junto con todas las intermedias serán promediadas. Esto resulta en un desfase entre la posición promedio de las señales y la posición de la última señal, o posición real del tren, inclusive bajo condiciones perfectas.

Ventana de tiempo para el contador

Variables de entrada

#corrida	error_tweet [Km]	tweets_tren [s]	tweets_tierra [s]	step_count_length [s]	step_sent_length_in [s]	step_sent_length_out [s]	tiempo_final	trenes_simulados
1	-	2	-	10	5	5	40	123

Figura 29: Variables de entrada

Resultado

#corrida	step_pos_length [s]	error_prediccion_maximo [m]
1a	2	-13
1b	20	-138
1c	60	-377
1d ¹⁰	60	-378

Figura 30: Resultados

¹⁰ La corrida d se hizo sin espera en las estaciones.

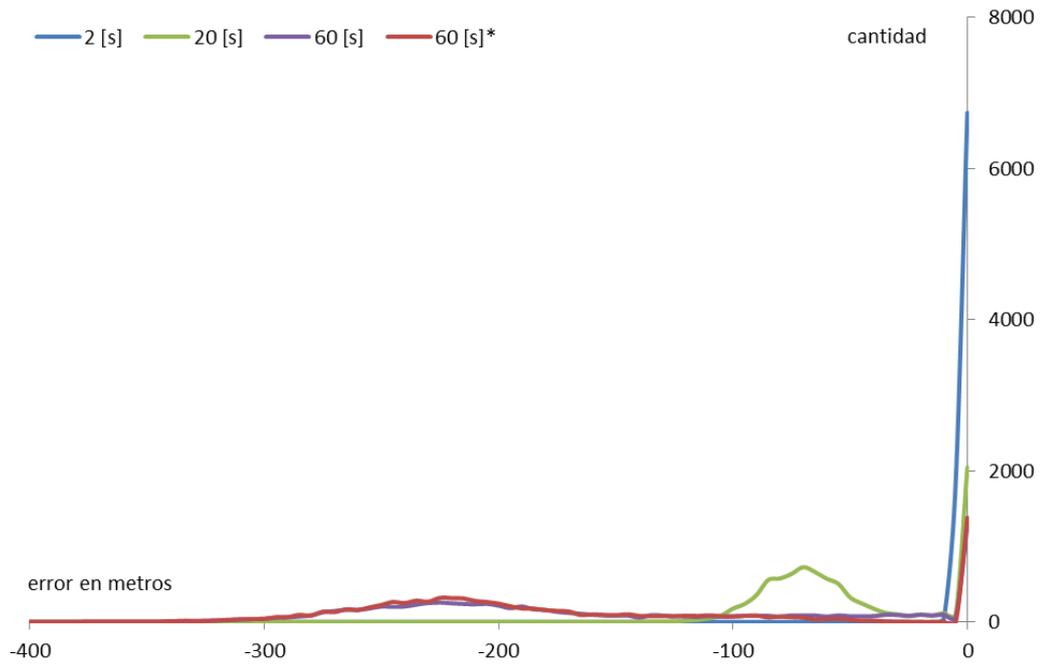


Figura 31: Distribución del error del experimento

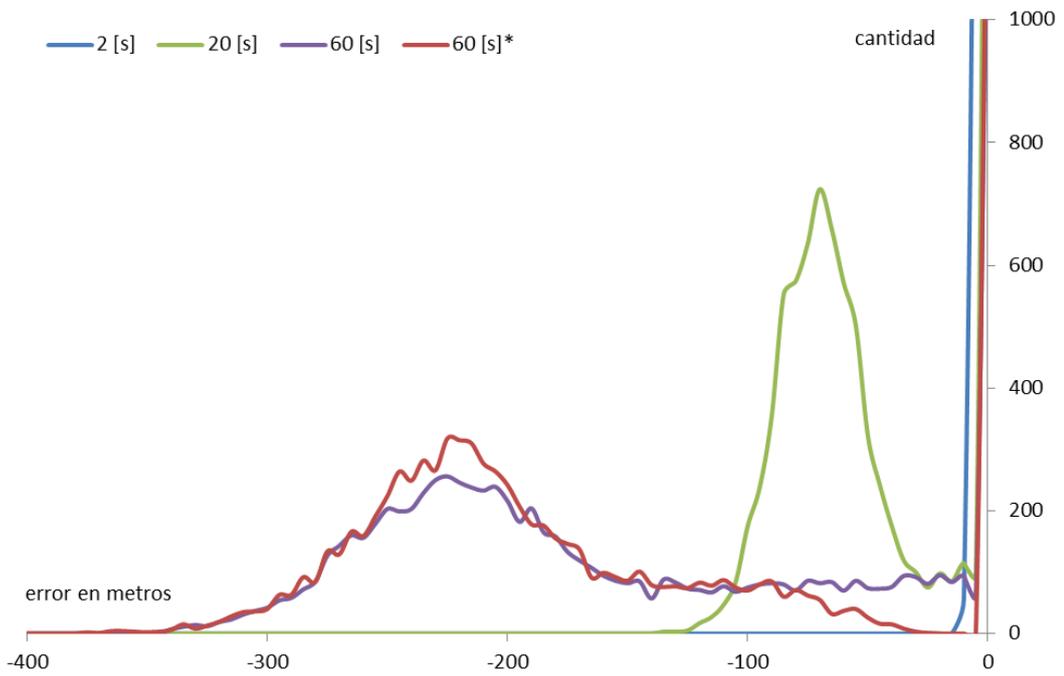


Figura 32: Distribución del error del experimento

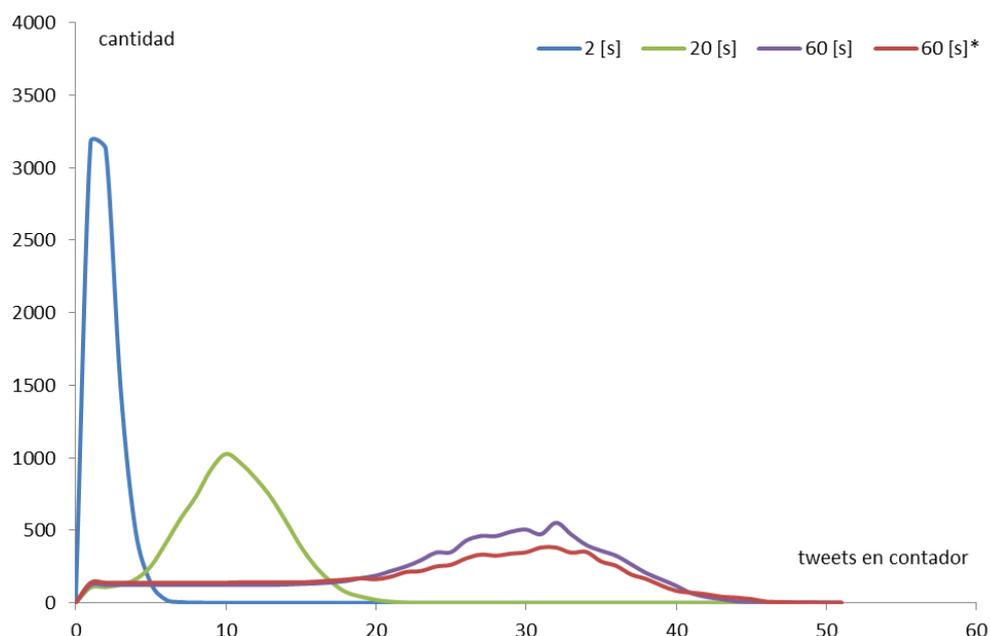


Figura 33: Distribución del número de Tweets en el contador

Análisis

Como era de esperar, al agrandar la variable de estudio, aumenta el promedio de tweets en el contador por cada vez que se lo revisa. La media en el contador se acerca a $(step_pos_length) / tweets_tren$

Por otro lado, el comportamiento de la media del error en la predicción de la posición responde al siguiente esquema:

$$error_prediccion_teorico = (step_pos_length * velocidad_de_avance_del_tren) / 2$$

$$velocidad_de_avance_del_tren = 8.333333 \text{ m/s}$$

#corrida	error_prediccion_teorico [m]
1a	-8.3
1b	-83.3
1c	-250

Tabla 7: Evaluación del error teórico medio

Al no haber error en la geo localización de la señal (condiciones *perfectas*), y al estar promediando señales en una ventana de tiempo, el algoritmo predecirá la posición del tren con un retraso de hasta $(step_pos_length * velocidad_de_avance_del_tren)$. Debido a que el tren está emitiendo señal constantemente ese retraso va a tender a $(step_pos_length * velocidad_de_avance_del_tren)/2$, como se plantea en la tabla de arriba. El desfase entre el valor teórico y el práctico se debe a la distribución aleatoria con la cual las señales son emitidas.

Se realizó una corrida sin tiempo de espera en las estaciones para ver como afectaba que los contadores arranquen sin 30 segundos de señales emitidas. Como se aprecia en las curvas del grafico de resultados, el peso de este factor no es relevante para el estudio.

Observación – Al alargar el `step_pos_length`, agrego inexactitud en la media de la predicción de la posición del algoritmo, a cambio de tener más datos dentro del contador.

Error en la geo localización de la señal

Variables de entrada

#corrida	tweets_tren [s]	tweets_tierra [s]	step_count_length [s]	step_pos_length [s]	step_sent_length_in [s]	step_sent_length_out [s]	tiempo_final	trenes_simulados
2	1	-	-	30	60	5	40	124

Tabla 8: Variables de entrada

Resultado

Se acomoda el `pos_limite` para compensa el error de la geo localización de la señal.

#corrida	error [km]	pos_limite	%detectados	%wrong_sent	media [m]	desvío [m]
2a	0.05	28	100	0	-120	21
2b	0.25	27	100	4	-120	26
2c	0.5	23	100	14	-113	57
2d	1	21	100	31	-99	140
2e	2	18	100	43	-84	274
2f	4	11	100	49	-66	542

Tabla 9: Resultados

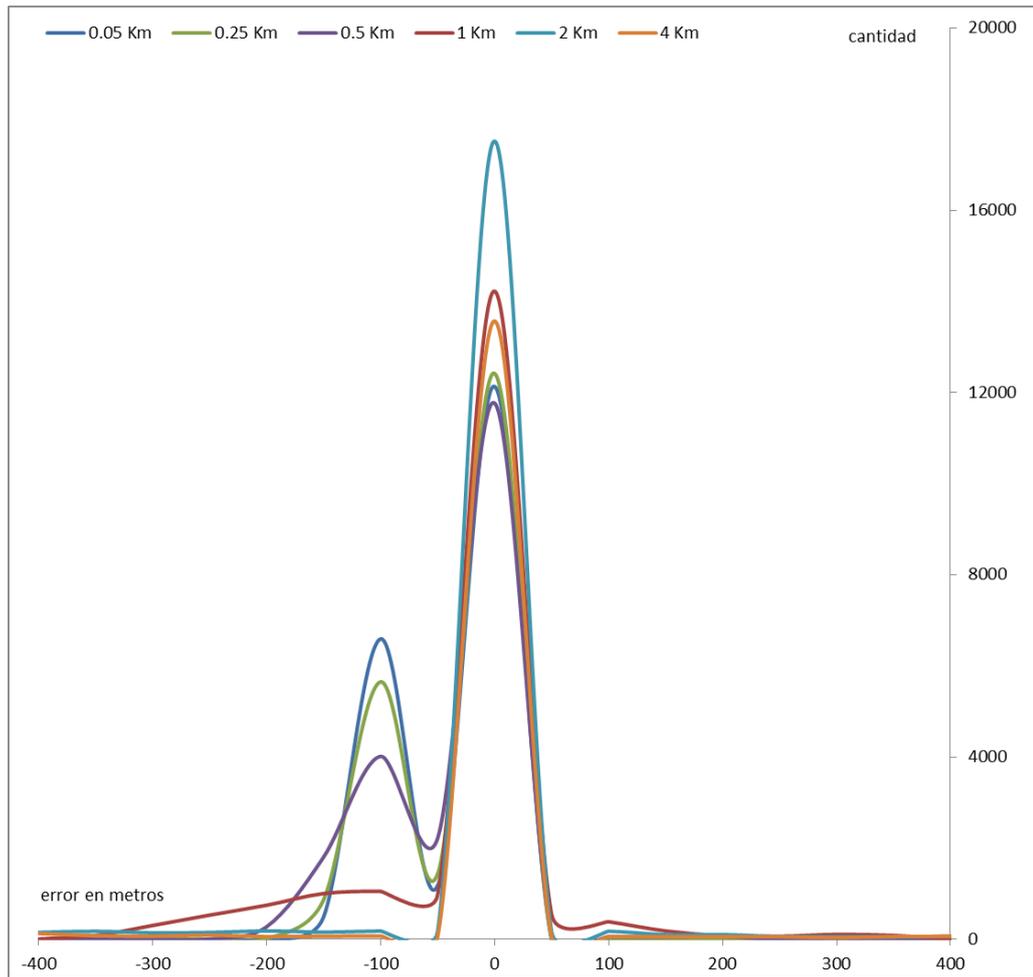


Figura 34: Distribución del error del experimento

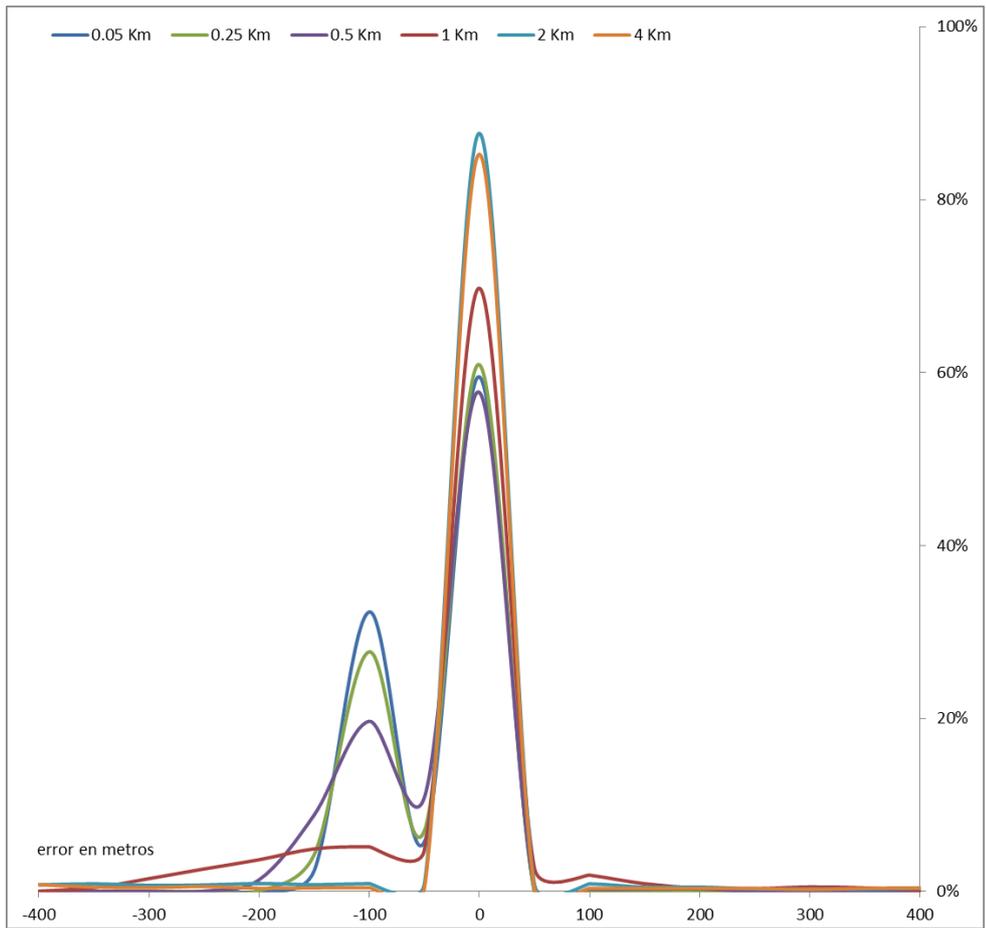


Figura 35: Distribución del error del experimento

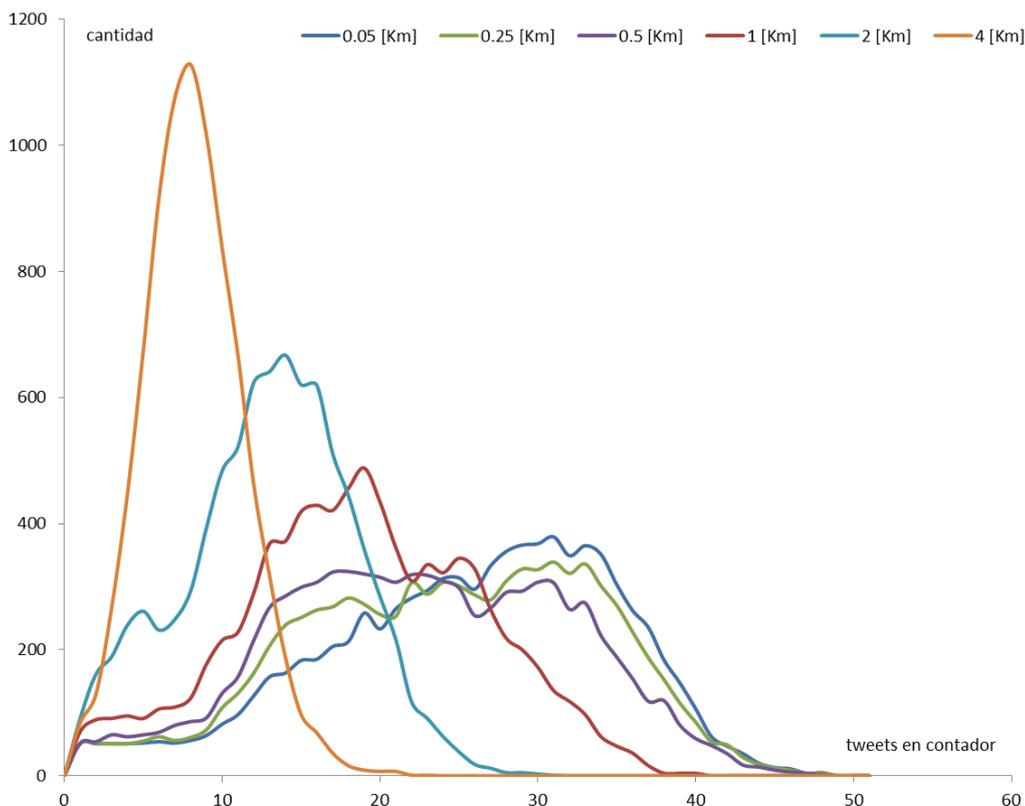


Figura 36: Distribución del error del experimento

Análisis

Al aumentar el error de la geo localización de la señal emitida ocurren dos fenómenos. Primero, disminuye la media de cantidad de tweets en el contador al mismo tiempo que disminuye el desvío de del misma. Esto se traduce en un aumento en el desvío del error de predicción de la posición, ya que hay menos datos para promediar y esos datos llevan mayor error. En segundo lugar, el aumento del error de la geo localización de la señal aplasta el retraso en la localización del tren (el que es producto de promediar tweets) contra la curva principal y por eso se ve un aumento en el pico principal. Sin embargo lo que está pasando es destructivo para la predicción del algoritmo ya que ahora el rango de error donde entra el 99% de los casos es mucho mayor. Esto último queda expuesto en la siguiente tabla.

#corrida	% de casos	Límite inferior	Límite superior
2a	99.96	-150	0
2b	99.88	-150	0
2c	>99	-200	50
2d	>99	-350	300
2e	>99	-450	450
2f	>99	-950	900

Tabla 10: Márgenes

Observación – El error de la geo localización de la señal influye fuertemente en la cantidad de información que los contadores obtienen. Su efecto sobre la predicción de la posición es menos relevante directamente, ya que de haber suficiente información en los contadores, se contrarresta.

Señal – Tweets emitidos desde el tren

Variables de Entrada

#corrida	Error [km]	tweets_tie rra [s]	step_coun t_length [s]	step_pos_l ength [s]	step_sent _length_in [s]	step_sent _length_o ut [s]	tiempo_fi nal
3	-	-	-	60	1	30	40

Tabla 11: Variables de entrada

Resultados

#corrida	tweets_tren [s]	pos_limite	%detectados	media [m]	desvío [m]
3a	1	51	100	-201	76
3b	2	25	100	-196	78
3c	5	10	100	-191	77
3d	10	3	100	-153	83
3e	30	-1	88	-94	89

Tabla 12: Resultados

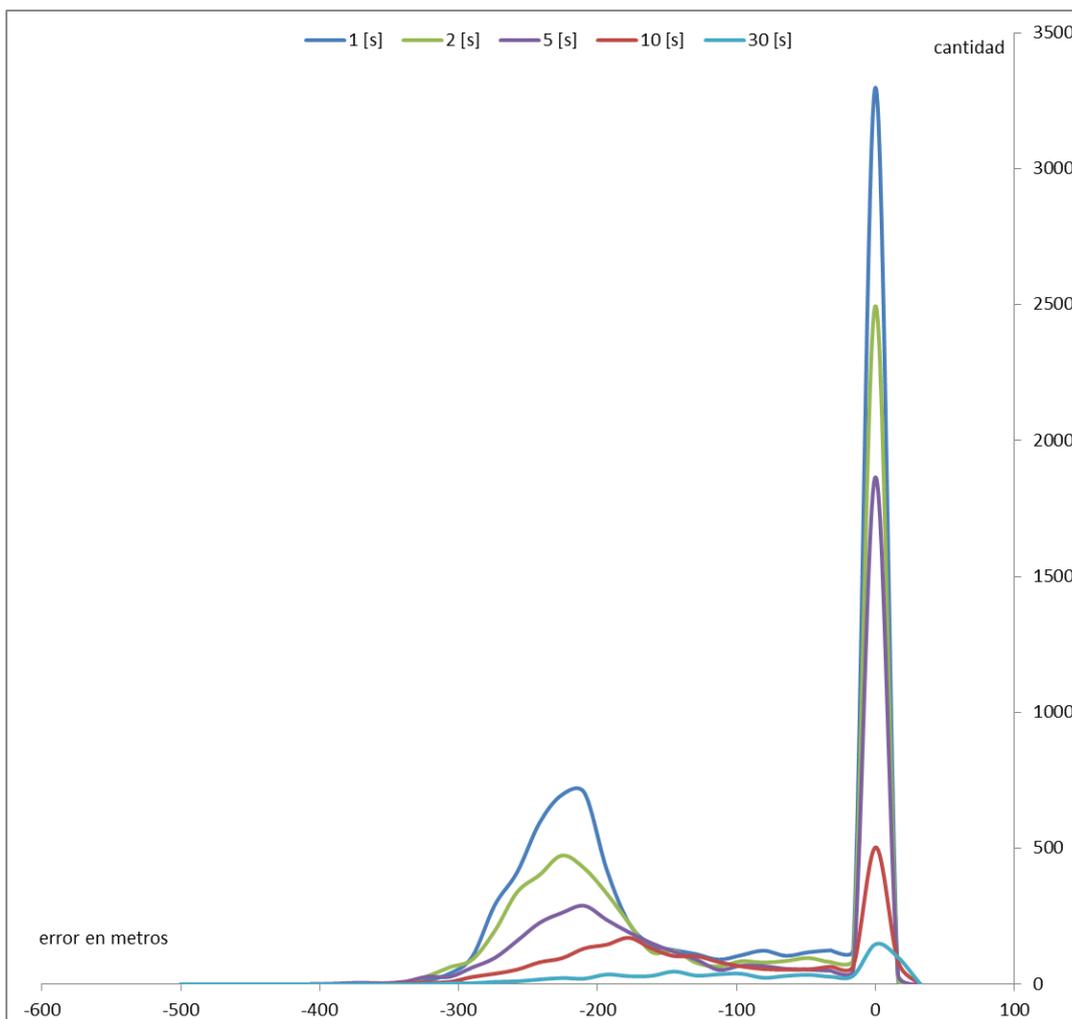


Figura 37: Distribución del error del experimento

Análisis

Observación - La potencia de la señal afecta directamente la cantidad de información disponible y es clave para el correcto funcionamiento del algoritmo. No afecta de otra manera relevante.

Ruido

Variables de entrada

#corrida	Error [km]	tweets_t ren [s]	step_cou nt_lengt h [s]	step_pos _length [s]	step_sen t_length _in [s]	step_sen t_length _out [s]	tiempo_f inal [horas]	trenes_si mulados
4	-	5	120	60	1	30	40	132

Tabla 13: Variables de entrada

Resultados

#corrida	tweets_terra [s]	pos_limite	%detectados	media [m]	desvío [m]
4a	5	19	100	-143	334
4b	50	9	100	-163	116
4c	250	8	100	-177	94
4d	900	6	100	-168	89
4e	1800	8	100	-179	82
4f	infinito	7	100	-171	83

Tabla 14: Resultados

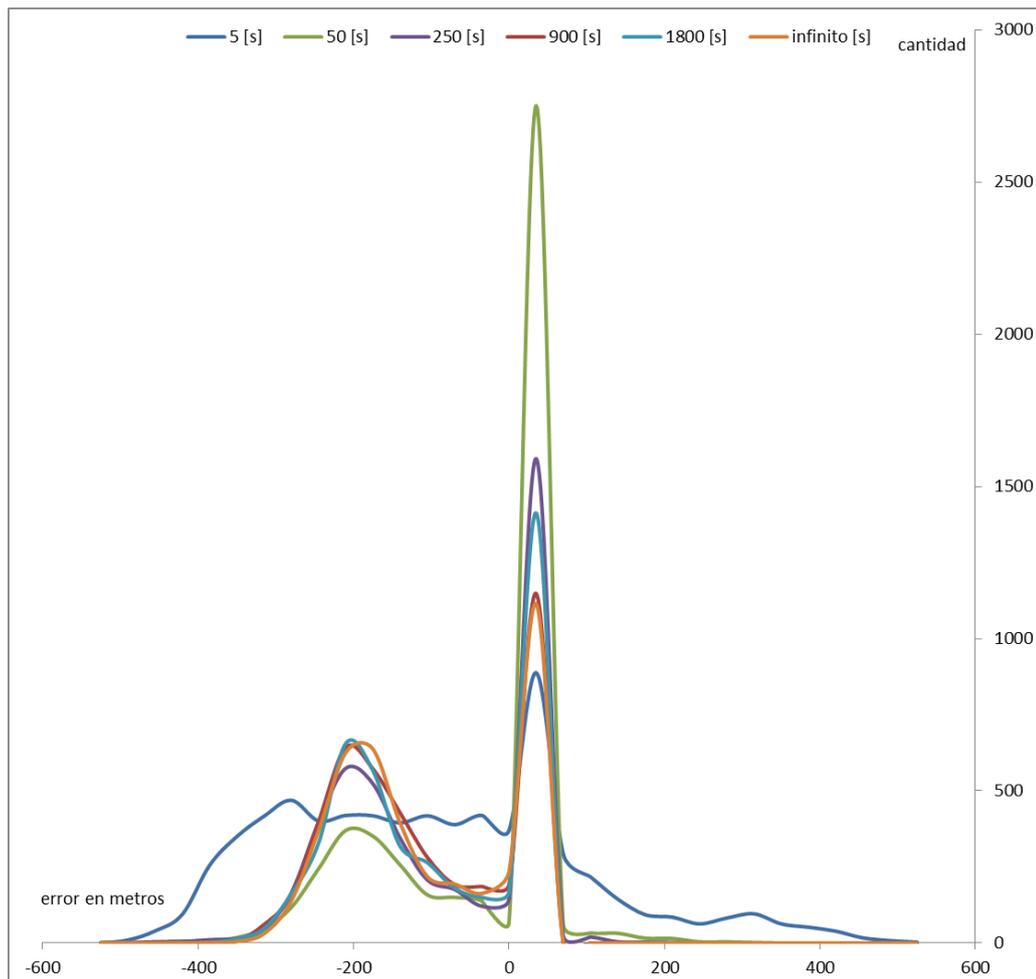


Figura 38: Distribución del error del experimento

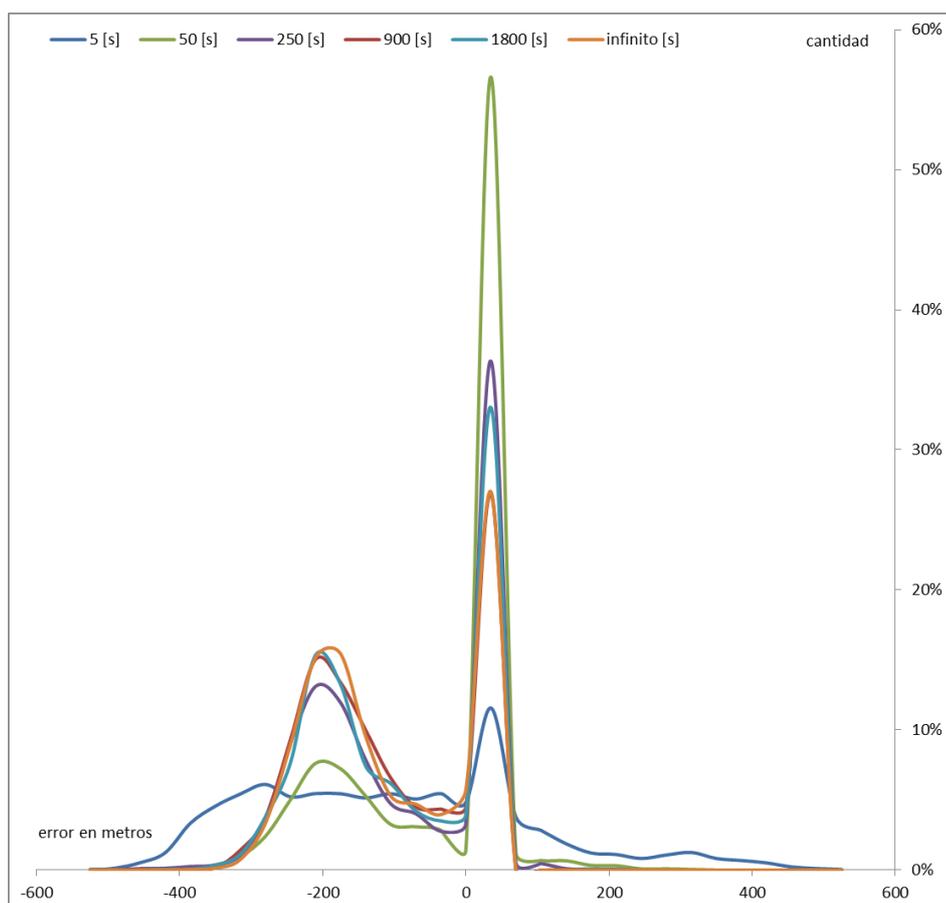


Figura 39: Distribución del error del experimento

Análisis

El aumento del ruido genera respuestas muy similares a las del aumento del error de la geo localización de la señal. Es decir, a mayor ruido, la media del error de la predicción de la posición tiende hacia el centro (cero metros) mientras que su desvío se dispara. Sin embargo aparece otro fenómeno llamativo. Si el ruido aumenta, más se concentra el error en la predicción, como es de esperarse, hasta el punto en el cual el ruido y la señal son llamativamente parecidos, y en ese momento, el error en la predicción de la posición se vuelve inconsistente. Por otra parte, el aumento en el ruido afecta el rango de error de predicción dentro del cual caen el 99% de los casos, como se ve en la siguiente tabla.

#corrida	% de casos	Límite inferior [m]	Límite superior [m]	Total [m]
4a	> 99	-1505	1120	2625
4b	> 99	-350	210	560
4c	> 99	-350	105	455
4d	> 99	-315	35	350
4e	> 99	-315	35	350
4a	> 99	-280	35	315

Tabla 15: Márgenes

*Observación – La relación ruido/señal tiene un límite que no debe superarse, ya que de lo contrario se vuelve destructivo para el sistema. Señal*10 < ruido, es una estimación valida con la información disponible.*

Determinación del sentido

La lógica que afecta este indicador es la misma que para la predicción de la posición, pero al tener un parámetro de más, se procede a analizar su efecto.

Variables de Entrada

#corrida	Error [km]	tweets_tr en [s]	step_count_length [s]	step_pos_length [s]	tiempo_final [horas]	trenes_sismulados
22-26	1	5	120	30	5	17

Tabla 16: Variables de entrada

Resultado

#corrida	step_sent_length (in,out)	%wrong	%detectados
22	(0,30)	48%	100%
23	(60,30)	25%	100%
24	(120,30)	11%	100%
25	(180,30)	9%	100%
26	(360,30)	20%	24%

Tabla 17: Resultados

Análisis

Al modificar el step_sent_lenght_in se observa la siguiente lógica. A medida que se aumenta, disminuye el %wrong, resultado lógico ya que la probabilidad de que se inviertan los sentidos de dos tweets con 120 segundos de diferencia es baja. Pasado cierto límite, se deja de detectar trenes, ya que estaríamos pidiendo más tiempo entre dos tweets del que es físicamente posible. Este parámetro por lo tanto depende fuertemente del error.

Observación – Este parámetro hay que regularlo para que sea probable encontrar un tweet en la ventana (in, out). De lo contrario el %detectados caerá. Para error de geo localización de 1km, (120, 30) es una elección valida.

Simulación con variables reales

Variables de entrada

#corrida	Error [km]	tweets_tierra [s]	step_count_length [s]	step_pos_length [s]	step_sent_length_in [s]	step_sent_length_out [s]	tiempo_fi nal [horas]	trenes_simulados
5	1	1800	120	60	120	30	40	

Tabla 18: Variables de entrada

Resultado

#corrida	Tweets_tren	%detectados	%wrong_sent	media [m]	desvío [m]
5a	5	100	16	-59	307
5b	10	100	17	-45	321
5c	15	94	18	-39	326
5d	30	67	15	-34	361
5e ¹¹	5	100	16	-27	304
5f ¹²	5	100	3	-60	130
5g ¹³	10	100	17	-94	344

Tabla 19: Resultados

¹¹ Esta corrida se realizó con un step_pos_length = 30 segundos.

¹² Esta corrida se realizó con un step_pos_length = 30 segundos y un error_tweet = 0.25 km

¹³ Esta corrida se realizó con un step_pos_length = 120 segundos

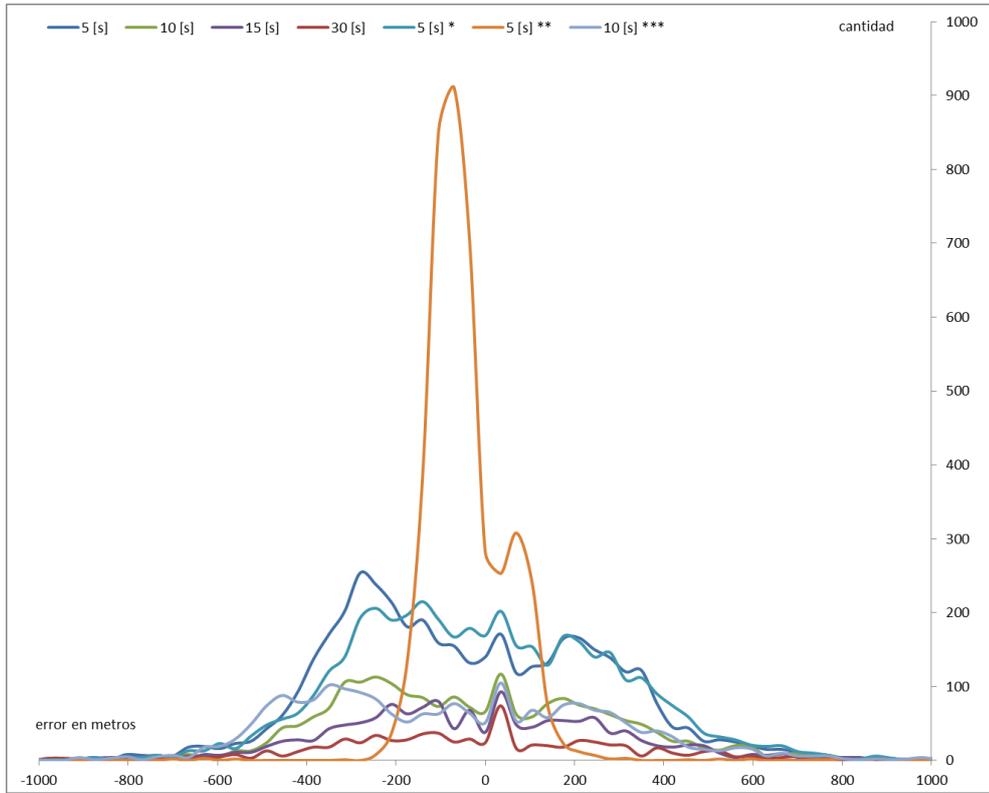


Figura 40: : Distribución del error del experimento

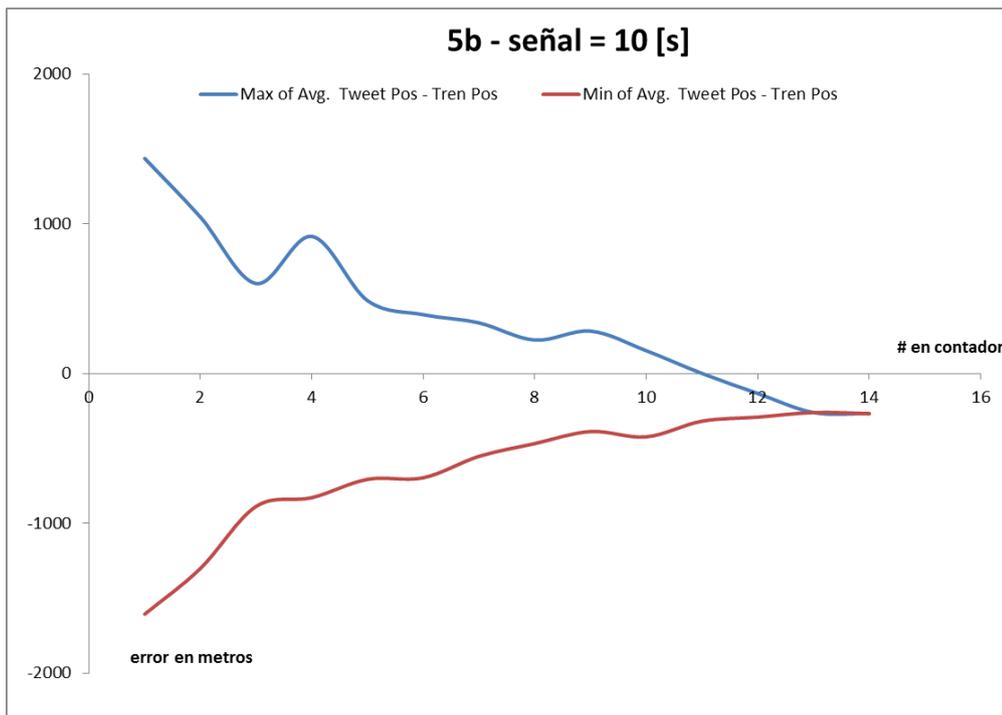


Figura 41: Margen de error vs Cantidad de Tweets en contador

Análisis

Si se lograra un señal = 10 [s] emitida desde el tren, y se fijara pos_limite = 4, eso devuelve un 97% de trenes detectados con un error real acotado en ± 500 metros. Lo que hace realmente una diferencia a gran escala es la disminución del error a 250m, como se ve en la corrida 5f.

Conclusiones

Factibilidad técnica

De la experimentación propuesta, se desprenden observaciones que permiten analizar situaciones y requisitos para el funcionamiento del sistema planteado. Estas están detalladas a continuación.

Con una señal emitida desde el tren de 10 [s] los resultados obtenidos (97% de trenes detectados y un error de posición ± 500 metros) son aceptables. Estos se dan bajo condiciones que no creemos vayan a cambiar significativamente en el corto plazo (ruido de 1800 [s] y error de geo localización de la señal de 1 kilometro). Esto implica que un tren debe emitir por lo menos diez tweets en un tramo de dos kilómetros entre estaciones. La distribución de este parámetro según nuestra simulación es la siguiente:

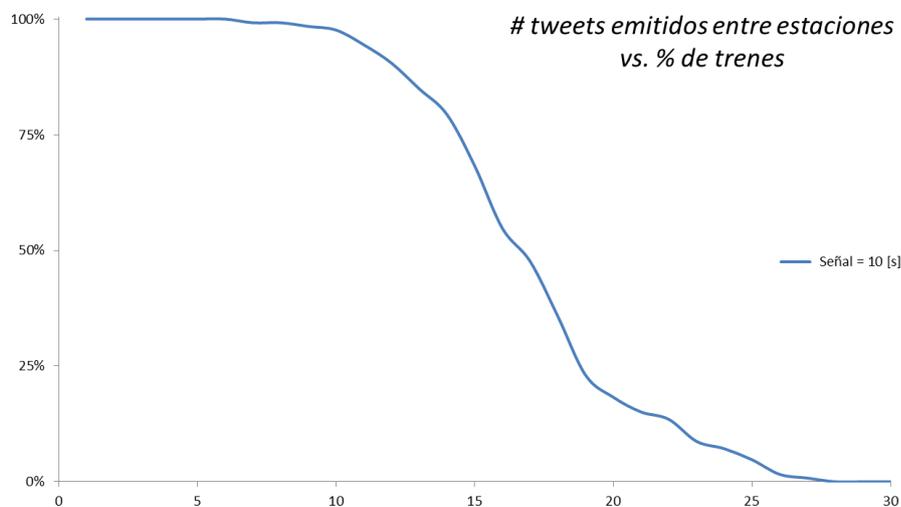


Figura 42: Tweets emitidos entre estaciones vs. % de trenes

Entonces, para lograr la señal que nos permite obtener resultados aceptables en el sistema, debemos conseguir 10 tweets entre estaciones. Dado que una formación transporta en promedio aproximadamente 400 personas por tramo, el requerimiento de señal se traduce a que 2.5% de los pasajeros emitan un tweet entre tramos. En la Argentina se estima que la penetración de Twitter es del 20% por lo que el requerimiento se puede suponer se encuentra dentro de parámetros aceptables. Estos datos habría que cruzarlos con penetración de Twitter en teléfonos móviles y patrones de uso de Twitter en la Argentina para obtener conclusiones definitivas.

Por otra parte tenemos que bajo estas condiciones, los trenes son localizados ± 500 metros. Esto en el peor de los casos (si hay una estación dentro de esos 500 metros) se corresponde con 60 segundos en vías + 30 segundos en estación. Por lo que el error de localización es de ± 90 segundos.

Al mismo tiempo, concluimos que uno de los efectos más positivos que una variable real puede tener sobre nuestros resultados es la disminución del error de localización de la señal. La misma hoy es de 1 kilómetro, pero si se viese reducida a 250 metros, la cota de error se vería reducida a ± 200 metros (60 segundos). Además de ello, el porcentaje de sentidos predichos incorrectamente pasaría de 17% a 3% y la señal que estaría emitiendo el tren sería considerablemente más robusta. Mirando hacia el futuro, donde las tecnologías de los Smartphone mejoran y el parque de celulares se ve renovado, esto es justamente lo que va a ocurrir, ya que se va a transicionar de la geo localización de la señal por triangulación de torres celulares hacia el uso del GPS interno.

Por último queda remarcar que el indicador de sentido del tren, tal cual como está diseñado, devuelve resultados sub-óptimos y debe ser objeto de futuros estudios para su mejora.

Consideraciones para la implementación

Para implementar el sistema en el mundo real hace falta realizar varias operaciones y estudios. A continuación se explicaran cuales son ellos y lo que creemos son las mejores prácticas para llevarlos a cabo.

Seteo óptimo del Sistema

Para una determina situación real (ruido, señal y error) hay un set de parámetros óptimos para el algoritmo. Según lo analizado anteriormente, estos se calculan de la siguiente manera.

Count_limite: se coloca acorde al nivel de ruido, para que todo set de tweets que cumpla el count_limite en un step_count_length tenga una determinada probabilidad de estar compuesto íntegramente de ruido. Ese porcentaje representa los falsos positivos del sistema. Es decir el porcentaje de veces en las que predecimos un tren inexistente. Creemos que un 1% de falsos positivos es un porcentaje aceptable.

Step_count_length: se calcula en conjunto con el count_limite, acorde al nivel de ruido, según lo estipulado en las instrucciones de seteo de esa variable.

Pos_limite: la regla de decisión para esta variable es maximizar hasta que se llegue a un % de detectados aceptable. Al aumentar disminuyen los trenes detectados, pero

aumenta la precisión de la detección, por lo tanto es cuestión de fijar uno de los dos y aceptar el otro.

Step_pos_length: Este parámetro hay que setearlo considerando que si se aumenta, disminuirá el desvío del error debido a que habrá más tweets para promediar y eso contrarresta la variabilidad de la normal, pero a su vez se está promediando tweets de posiciones más distantes y eso implica una media del error más alta. Entre 30 y 60 segundos resulta aceptable empíricamente.

Step_in: Aquí lo que se hace es se plantea un %wrong aceptable y de él se desprende el step_in de acuerdo al error de la situación actual. Si se exige demasiado este parámetro, baja el %detectados.

Step_out: Esta variable se fija en base a la situación de tweets_tren. Ya que la misma determina la ventana en la cual se va a estar buscando un tweet, y a distintos tweet_tren hay que modificarla para lograr el mismo % detectados.

Elección de los puestos de control

Los puestos de control representan la superficie de tierra donde se buscarán tweets. Por como está diseñada la API de Twitter, los mismos se deben parametrizar como circunferencias, con un centro y un radio. De todas formas se puede combinar la información de varias circunferencias, para lograr todo tipo de formas. Por diseño de nuestro algoritmo es conveniente buscar información en todo punto que se encuentra a menos de 1km (error de geo localización) de un segmento de vía. Ya que esto implicaría cantidades inmensas de trabajo parametrizando toda la vía, por simplicidad se recomienda establecer un puesto de control en el punto medio entre dos estaciones sobre las vías. Si el radio de análisis es pequeño, impactará negativamente en el resultado del algoritmo, ya que habrá menos información sobre la cual trabajar. Un puesto de control no debe estar levantando tweets sobre más de un tramo de vías, entendiéndose por tramo de vías a la porción que se encuentra entre dos estaciones contiguas.

Relevamiento de parámetros para setear el sistema

Como se propuso en el análisis del algoritmo, para lograr maximizar su potencia, se debe setear acorde a los parámetros de la realidad, siendo estos, la cantidad de señal, ruido y error de geo localización. Al ser los primeros dos, de naturaleza variable lo que se propone hacer es medirlos constantemente y reajustar el algoritmo cuando sea necesario. Para medir el ruido creemos que es suficiente tomando muestras de superficies de tierra adyacentes a los puestos de control, que sumen la misma cantidad de metros cuadrados que abarca un puesto de control, y que se encuentran a más de 1 km de las vías. Procesando estos datos se puede sacar un estimativo del nivel de ruido

que puede estar sufriendo el puesto de control adyacente. Hay que ser especialmente cuidadoso en el caso del ramal retiro tigre de no estar tomando como datos adyacentes, superficies de agua, ya que la densidad de tweets a encontrar allí no es representativa de lo que sucede en las cercanías de las vías. Una vez calculado el ruido, se le puede restar esa distribución a la encontrada dentro de los puestos de control, para así estimar la potencia de la señal. Con estos dos parámetros se puede regular constantemente el algoritmo y obtener los mejores resultados.

Parametrización del sistema de vías

A lo largo de nuestro estudio siempre se simplificó la vía, como si fuera un tramo recto, y la posición de los tweets tuviesen una sola coordenada de posición. Esto no sucede en la realidad, ya que los segmentos de vía son curvos y los tweets tienen latitud y longitud. Convertir estas coordenadas a un sistema de una sola coordenada implica una complejidad mayor, pero debe ser realizado al momento de implementar el sistema. Escapa a nuestro trabajo el análisis de la forma más eficiente de hacer esto, pero es un tema a ser tenido en cuenta, ya que probablemente implique la parametrización de todo el ramal de vías.

Información de salida

Al momento de presentarle información al usuario, hay que tomar la decisión de qué tipo de error va a conllevar la misma. Si nos guiamos por lo expuesto en el análisis, encontramos que nuestras predicciones estaban erradas con respecto a la realidad, pero su error estaba acotado. Entonces si se conoce las cotas de error de nuestra predicción, lo más apropiado sería ajustar convenientemente la información a presentar para que el usuario final se vea favorecido para esta información extra. Es decir, de toda predicción sabemos que el tren puede estar como máximo x metros por delante, hasta y metros por detrás. Lo apropiado sería agregarle a todas las predicciones x metros, por lo que todos los trenes estarían $x + y$ metros por detrás. De esa forma el usuario nunca se perdería un tren. Como contrapartida esto implica que es más probable que el usuario esté esperando un tren, y que nuestras predicciones sean generalmente equivocadas. Consideramos este es un pequeño precio a pagar, comparado al beneficio que obtiene el usuario.

Fuentes de mejora

Predicción del sentido

En lo que hace al algoritmo predictor, se mencionó anteriormente que los resultados devueltos por el predictor del sentido de circulación del tren son sub-óptimos. En las corridas en situación real rondaban el 15% de equivocación. Ese número se podría

bajar corrigiendo el `step_sent_lenght(in,out)`. La mejora sería incremental y se sacrificaría % de detectados. Sin embargo si se diseñase el indicador para que promedie la posición de cierto número de tweets en cada extremo de búsqueda, en lugar de tomar uno por cada extremo, la sensibilidad al error de geo localización de la señal se vería disminuida. Este es el mismo concepto que se aplicó a la predicción de la posición. En él, se promedian las posiciones en una ventana de tiempo, para contrarrestar la dispersión debida al error de geo localización. De implementarse este método en la predicción del sentido, se daría un cambio de paradigma en el mismo y estimamos los resultados sería exponencialmente mejores, permitiendo encima achicar el `step_sent_lengh(in,out)` y aumentar el % de detectados.

Filtro de Ruido

Primero, el ruido, definido como los tweets emitidos desde la tierra, es una fuente de variabilidad y error de nuestro sistema. Haciendo un análisis más profundo sobre cómo está compuesto el mismo y qué patrones sigue, se pueden desarrollar técnicas para filtrarlo y disminuirlo considerablemente. Una alternativa es analizar la fuente de esos tweets:

Usuario	# de Tweets
"HernanLondon"	16
"ernestoarriaga"	9
"GboSaul"	5
"HernanFSanz"	4
"MoreUrcelay"	3
"Fenomena"	3
"PochiMaidana"	3
"Doble69"	3

Tabla 20: Cantidad de Tweets por usuario (Top 8)

Esto es extraído de los mismos cien tweets que se utilizaban anteriormente para determinar el nivel de ruido actual. Se observa que un pequeño grupo de usuarios está generando la mayoría del ruido, y lo está haciendo a intervalos de tiempos que hacen poco probable que sean emitidos desde un tren (distintas horas a lo largo del día). Siguiendo esta lógica, es posible que estos usuarios vivan en el área en estudio o bien trabajen allí. Si se decidiese filtrar estos usuarios de los datos que ingresan al algoritmo se podría reducir considerablemente el ruido. Este análisis por supuesto debería ser extrapolado a todos los segmentos de vía en estudio.

Por otro lado, los datos extraídos de Twitter traen consigo un dato llamado "geo:null". La geo localización de un tweet puede estar dispuesta de dos maneras distintas. La primera es a través de antenas celulares y GPS propio de un equipo móvil como se explicó anteriormente. La segunda es que un usuario determine su localización en su perfil, de una forma genérica, la cual es estática. Este segundo tipo de localización, no

implica presencia física del emisor en esa posición y no es calculado y actualizado en tiempo real, por lo que puede ser filtrado del ruido también. Esta técnica reduce en un 50% los datos analizados en el análisis del ruido. Si se implementan efectivamente estos dos filtros, de los 100 tweets de ruido con los que se realizó el estudio de situación actual, solo quedan 8, lo cual es un impacto brutal.

Memoria y contraste con el pasado

Finalmente notamos que el análisis de la posición de los trenes y el sentido de los mismos se hace instante a instante, sin memoria alguna de lo predicho anteriormente. Este punto es uno de los más flojos del modelo y creemos que se deberían analizar estrategias para inyectarle memoria al proceso y agregar datos del pasado, para determinar el futuro. Dicha lógica escapa el estudio en cuestión, pero creemos que es una gran oportunidad de mejora en el futuro.

Limitaciones del estudio

Superposición de señales

Una de las limitaciones más evidentes del método de estudio elegido es que desconocemos la respuesta del algoritmo con dos o más trenes circulando por el mismo segmento de vías, o inclusive dentro de nuestro sistema de 4 estaciones. Esto fue una limitación que se eligió conscientemente por dos razones. Para comenzar, la complejidad que implicaba identificar a qué tren estábamos detectando en cada instante hacía casi imposible controlar la potencia y precisión del algoritmo. Por otro lado, se supuso que el valor del sistema propuesto radica en identificar la falta de trenes y no la presencia de más de uno. Es decir, buscamos generar valor pudiendo identificar esas situaciones en las cuales un usuario va a estar 30 o más minutos esperando en una estación, como para que no lo haga y desde allí agregar valor.

Ruido desde estación

Otro factor no contemplado en la simulación, es el ruido propio de las personas en las estaciones. Las dos fuentes principales de tweets en nuestro modelo fueron la tierra por un lado, con una densidad baja de emisión, y la señal (tweets desde tren) por el otro, con una densidad mayor que la anterior. Una realidad que no fue modelizada por ende, es el ruido proveniente de las personas esperando en las estaciones, que será bastante más alto al modelizado para la tierra. Esto podría ser un factor de error al momento de la implementación. Se tendría que disuadir a las personas de tweetear desde las estaciones, lo que podría hacerse con efectivas campañas gráficas que distraigan a las personas.

Circulación Irregular

El modelo no contempla tampoco un fenómeno bastante común que es la circulación no regular de las formaciones. En más de una ocasión este se detiene entre dos estaciones, se va fuera de servicio por algún imperfecto terminando el recorrido en una estación dada, u otros imprevistos no tenidos en cuenta en el modelo, que asumió un tren viajando constantemente entre estaciones.

Apéndices

Apéndice I: Herramientas para la extracción de información de Twitter

Json:

“JSON, acrónimo de JavaScript Object Notation, es un formato ligero para el intercambio de datos. JSON es un subconjunto de la notación literal de objetos de JavaScript que no requiere el uso de XML. La simplicidad de JSON ha dado lugar a la generalización de su uso, especialmente como alternativa a XML en AJAX. Una de las supuestas ventajas de JSON sobre XML como formato de intercambio de datos en este contexto es que es mucho más sencillo escribir un analizador sintáctico (parser) de JSON. En JavaScript, un texto JSON se puede analizar fácilmente usando el procedimiento eval(), lo cual ha sido fundamental para que JSON haya sido aceptado por parte de la comunidad de desarrolladores AJAX, debido a la ubicuidad de JavaScript en casi cualquier navegador web.”
<http://es.wikipedia.org/wiki/JSON>

Python:

“Python es un lenguaje de programación de alto nivel cuya filosofía hace hincapié en una sintaxis muy limpia y que favorezca un código legible. Se trata de un lenguaje de programación multiparadigma ya que soporta orientación a objetos, programación imperativa y, en menor medida, programación funcional. Es un lenguaje interpretado, usa tipado dinámico, es fuertemente tipado y multiplataforma. Es administrado por la Python Software Foundation. Posee una licencia de código abierto, denominada Python Software Foundation License, que es compatible con la Licencia pública general de GNU a partir de la versión 2.1.1, e incompatible en ciertas versiones anteriores. <http://es.wikipedia.org/wiki/Python>

Código en Python

```
import os
import simplejson
count = 0
while os.path.exists("datos.%d.csv" % count)
:
    count += 1
fd = file("datos.%d.csv" % count, "w")
print "Cargando el archivo..."
data = simplejson.loads(file("datos.json").read().strip())
count = 0
print "Escribiendo...."
fd.write("\t".join(['iso_language_code', 'from_user_id_str', 'text', 'from_user_name',
'from_user', 'geo', 'created_at']) + "\n")
```

```

for result in data["results"]:
    row = []
    for column in ['iso_language_code', 'from_user_id_str', 'text', 'from_user_name',
'from_user', "geo", 'created_at']:
        if column == "geo":
            geo = result.get("geo", {})
            if geo is None:
                geo = {}
            x, y = geo.get("coordinates", [0.0, 0.0])
            row.append("%.12f" % x)
            row.append("%.12f" % y)
        else:
            row.append(result.get(column))
    fd.write("\t".join(row).encode("utf-8") + "\n")
    if count % 100 == 0: print "Escribiendo %d...." % count
    count++
fd.close()

```

Apéndice II: Muestreo de Tiempo entre Tweets

Como se mencionó, se utilizó el siguiente código API de una porción aleatoria de tierra en el trayecto Retiro-Tigre, para obtener la distribución exponencial con parámetro 1800 segundos, que se utilizó como entrada en el modelo:

```

{"completed_in":0.17,"max_id":209387914327621633,"max_id_str":"209387914327621633","next_page":"?page=2&max_id=209
387914327621633&q=*&geocode=-34.5173598913518%2C-
58.48496675491333%2C2km","page":1,"query":"*","refresh_url":"?since_id=209387914327621633&q=*&geocode=-
34.5173598913518%2C-58.48496675491333%2C2km","results":[{"created_at":"Sun, 03 Jun 2012 20:55:43
+0000","from_user":"CabralAndrea1","from_user_id":"298834447","from_user_id_str":"298834447","from_user_name":"Andrea
Cabral","geo":null,"location":"San Vicente, Bs.As","id":"209387914327621633","id_str":"209387914327621633","iso_language_code":"pt","metadata":{"result_type":"recent"}
,"profile_image_url":"http://a0.twimg.com/profile_images/2168668499/DSC08349_normal.JPG","profile_image_url_https":"
https://si0.twimg.com/profile_images/2168668499/DSC08349_normal.JPG","source":"&lt;a
href=&quot;http://blackberry.com/twitter&quot; rel=&quot;nofollow&quot;&gt;Twitter for
BlackBerry\u00ae&lt;/a&gt;","text":"Hay dias que te dan ganas de mandar todo a la mierda
#contandohasta100","to_user":null,"to_user_id":"0","to_user_id_str":"0","to_user_name":null,{"created_at":"Sun, 03 Jun 2012
20:55:17
+0000","from_user":"ernestoarriaga","from_user_id":"180076191","from_user_id_str":"180076191","from_user_name":"Ernesto
Arriaga","geo":{"coordinates":[-34.5132,-
58.4754],"type":"Point"},"id":"209387806567567360","id_str":"209387806567567360","iso_language_code":"en","metadata":{"res
ult_type":"recent"},"profile_image_url":"http://a0.twimg.com/profile_images/1532748731/Arr_normal.JPG","profile_image_
url_https":"https://si0.twimg.com/profile_images/1532748731/Arr_normal.JPG","source":"&lt;a
href=&quot;http://twitter.com/#!/download/iphone&quot; rel=&quot;nofollow&quot;&gt;Twitter for
iPhone&lt;/a&gt;","text":"@ramdellano
uuu","to_user":"ramdellano","to_user_id":"376216070","to_user_id_str":"376216070","to_user_name":"Ramiro de
Llano","in_reply_to_status_id":"209387582918897664","in_reply_to_status_id_str":"209387582918897664"},"created_at":"Sun,
03 Jun 2012 20:54:49
+0000","from_user":"ernestoarriaga","from_user_id":"180076191","from_user_id_str":"180076191","from_user_name":"Ernesto
Arriaga","geo":{"coordinates":[-34.5132,-
58.4754],"type":"Point"},"id":"209387686115545088","id_str":"209387686115545088","iso_language_code":"es","metadata":{"res
ult_type":"recent"},"profile_image_url":"http://a0.twimg.com/profile_images/1532748731/Arr_normal.JPG","profile_image_
url_https":"https://si0.twimg.com/profile_images/1532748731/Arr_normal.JPG","source":"&lt;a
href=&quot;http://twitter.com/#!/download/iphone&quot; rel=&quot;nofollow&quot;&gt;Twitter for
iPhone&lt;/a&gt;","text":"Quien Hace el 2do Goooooo!!!!!! De BOCA ??????????????
rt","to_user":null,"to_user_id":"0","to_user_id_str":"0","to_user_name":null,{"created_at":"Sun, 03 Jun 2012 20:48:17

```

```
+0000","from_user":"estear্তু94","from_user_id":297779647,"from_user_id_str":"297779647","from_user_name":"Esterban
Arturo","geo":{"coordinates":[-34.5280,-
58.4937],"type":"Point"},"id":209386041126305793,"id_str":"209386041126305793","iso_language_code":"es","metadata":{"res
ult_type":"recent"},"profile_image_url":"http://a0.twimg.com/profile_images/1906911791/image_normal.jpg","profile_imag
e_url_https":"https://si0.twimg.com/profile_images/1906911791/image_normal.jpg","source":"&lt;a
href=&quot;http://twitter.com/#!/download/iphone&quot; rel=&quot;nofollow&quot;&gt;Twitter for
iPhone&lt;/a&gt;","text":"@CattoPazz jajajaja hay que imponer la alcurnia yungayina ps
xD","to_user":"CattoPazz","to_user_id":69392212,"to_user_id_str":"69392212","to_user_name":"Carolina Roa
Werner","in_reply_to_status_id":209374523588476929,"in_reply_to_status_id_str":"209374523588476929"},"created_at":"Sun,
03 Jun 2012 20:47:18
+0000","from_user":"mcbclara","from_user_id":94159478,"from_user_id_str":"94159478","from_user_name":"\u2728Mar\u00e
da Clara\u2728","geo":null,"location":"-34.513366,-
58.482369","id":209385796183130114,"id_str":"209385796183130114","iso_language_code":"es","metadata":{"result_type":"re
cent"},"profile_image_url":"http://a0.twimg.com/profile_images/2190655491/image_normal.jpg","profile_image_url_https":
"https://si0.twimg.com/profile_images/2190655491/image_normal.jpg","source":"&lt;a
href=&quot;http://twitter.com/#!/download/iphone&quot; rel=&quot;nofollow&quot;&gt;Twitter for
iPhone&lt;/a&gt;","text":"Definitivamente, los domingos me ponen de mal
humor","to_user":null,"to_user_id":0,"to_user_id_str":"0","to_user_name":null},"created_at":"Sun, 03 Jun 2012 20:40:21
+0000","from_user":"ernestoarriaga","from_user_id":180076191,"from_user_id_str":"180076191","from_user_name":"Ernesto
Arriaga","geo":{"coordinates":[-34.5132,-
58.4754],"type":"Point"},"id":209384047728799744,"id_str":"209384047728799744","iso_language_code":"es","metadata":{"res
ult_type":"recent"},"profile_image_url":"http://a0.twimg.com/profile_images/1532748731/Arr_normal.JPG","profile_image_
url_https":"https://si0.twimg.com/profile_images/1532748731/Arr_normal.JPG","source":"&lt;a
href=&quot;http://twitter.com/#!/download/iphone&quot; rel=&quot;nofollow&quot;&gt;Twitter for
iPhone&lt;/a&gt;","text":"Que Jugador. De. BOCA. Hace el primer GoooIIIIIIII.
Hoy????????","to_user":null,"to_user_id":0,"to_user_id_str":"0","to_user_name":null},"created_at":"Sun, 03 Jun 2012
20:39:34
+0000","from_user":"mcbclara","from_user_id":94159478,"from_user_id_str":"94159478","from_user_name":"\u2728Mar\u00e
da Clara\u2728","geo":null,"location":"-34.513366,-
58.482369","id":209383847652106240,"id_str":"209383847652106240","iso_language_code":"es","metadata":{"result_type":"re
cent"},"profile_image_url":"http://a0.twimg.com/profile_images/2190655491/image_normal.jpg","profile_image_url_https":
"https://si0.twimg.com/profile_images/2190655491/image_normal.jpg","source":"&lt;a
href=&quot;http://instagr.am&quot; rel=&quot;nofollow&quot;&gt;instagram&lt;/a&gt;","text":"Yo soy tu amiga fiel
http://t.co/jpfz8gMB","to_user":null,"to_user_id":0,"to_user_id_str":"0","to_user_name":null},"created_at":"Sun, 03 Jun
2012 20:39:30
+0000","from_user":"ernestoarriaga","from_user_id":180076191,"from_user_id_str":"180076191","from_user_name":"Ernesto
Arriaga","geo":{"coordinates":[-34.5132,-
58.4754],"type":"Point"},"id":209383831759896576,"id_str":"209383831759896576","iso_language_code":"en","metadata":{"res
ult_type":"recent"},"profile_image_url":"http://a0.twimg.com/profile_images/1532748731/Arr_normal.JPG","profile_image_
url_https":"https://si0.twimg.com/profile_images/1532748731/Arr_normal.JPG","source":"&lt;a
href=&quot;http://twitter.com/#!/download/iphone&quot; rel=&quot;nofollow&quot;&gt;Twitter for
iPhone&lt;/a&gt;","text":"@mguadalupev
ok","to_user":"mguadalupev","to_user_id":343809198,"to_user_id_str":"343809198","to_user_name":"M Guadalupe
Vazquez","in_reply_to_status_id":209382825617653760,"in_reply_to_status_id_str":"209382825617653760"},"created_at":"Sun
, 03 Jun 2012 20:36:48 (...)
```

Apéndice III: Análisis estadístico de los tweets

SPSS:

<http://www-01.ibm.com/software/analytics/spss/>

“SPSS es un programa estadístico informático muy usado en las ciencias sociales y las empresas de investigación de mercado. Originalmente SPSS fue creado como el acrónimo de *Statistical Package for the Social Sciences* aunque también se ha referido como "Statistical Product and Service Solutions" (Pardo, A., & Ruiz, M.A., 2002, p. 3).

Sin embargo, en la actualidad la parte SPSS del nombre completo del software (IBM SPSS) no es acrónimo de nada. [1](#)

Como programa estadístico es muy popular su uso debido a la capacidad de trabajar con bases de datos de gran tamaño. En la versión 12 es de 2 millones de registros y 250.000 variables. Además, de permitir la recodificación de las variables y registros según las necesidades del usuario. El programa consiste en un módulo base y módulos anexos que se han ido actualizando constantemente con nuevos procedimientos estadísticos. Cada uno de estos módulos se compra por separado.

Actualmente, compete no sólo con software licenciados como lo son SAS, MATLAB, Statistica, Stata, sino también con software de código abierto y libre, de los cuales el más destacado es el Lenguaje R.”

Q-Q Plot:

http://www.dm.uba.ar/materias/analisis_de_datos/2008/1/teoricas/Teor5.pdf

“Un gráfico Cuantil-Cuantil permite observar cuan cerca está la distribución de un conjunto de datos a alguna distribución ideal ó comparar la distribución de dos conjuntos de datos. “

http://es.wikipedia.org/wiki/Gr%C3%A1fico_Q-Q

“En estadística, un gráfico Q-Q ("Q" viene de cuantil) es un método gráfico para el diagnóstico de diferencias entre la distribución de probabilidad de una población de la que se ha extraído una muestra aleatoria y una distribución usada para la comparación. Una forma básica de gráfico surge cuando la distribución para la comparación es una distribución teórica. No obstante, puede usarse la misma idea para comparar las distribuciones inferidas directamente de dos conjuntos de observaciones, donde los tamaños de las muestras sean distintos.”

En un gráfico Q-Q se representa en el eje X la probabilidad acumulada para cada uno de los valores estudiados, y en el eje Y la probabilidad que le correspondería a ese valor si se tratase de la distribución con la cual se quiere comparar. Luego, cuanto más se acerque la línea de puntos representada a una recta, más se asemejarán los datos a la distribución.

Distribución Exponencial

http://es.wikipedia.org/wiki/Distribuci%C3%B3n_exponencial

“En estadística la distribución exponencial es una distribución de probabilidad continua con un parámetro $\lambda > 0$ cuya función de densidad es:

$$f(x) = \begin{cases} \lambda e^{-\lambda x} & \text{para } x \geq 0 \\ 0 & \text{de otro modo} \end{cases}$$

Su función de distribución es:

$$F(x) = P(X \leq x) = \begin{cases} 0 & \text{para } x < 0 \\ 1 - e^{-\lambda x} & \text{para } x \geq 0 \end{cases}$$

Donde e representa el número e .

El valor esperado y la varianza de una variable aleatoria X con distribución exponencial son:

$$E[X] = \frac{1}{\lambda}, \quad V(X) = \frac{1}{\lambda^2}$$

La distribución exponencial es un caso particular de distribución gamma con $k = 1$. Además la suma de variables aleatorias que siguen una misma distribución exponencial es una variable aleatoria expresable en términos de la distribución gamma.”

Mientras que la distribución de Poisson describe las llegadas por unidad de tiempo, la distribución exponencial estudia el tiempo entre cada una de estas llegadas. Además la distribución exponencial se utiliza para describir el tiempo que transcurre hasta que se produce un fallo siempre y cuando se cumpla la condición de que la probabilidad de que el fallo se produzca en un instante no depende del tiempo transcurrido. Por otro lado, esta distribución tiene aplicaciones en fiabilidad y en la teoría de la supervivencia.

Box Plot

En Estadística descriptiva, un “Box plot” es un gráfico basado en cuartiles que muestra y describe de forma entendible un grupo de datos numéricos a través de 5 números principales:

1. La observación más pequeña.
2. El menor cuartil.
3. La mediana
4. El cuartil superior
5. La observación más grande.

De ésta forma también se pueden conocer cuales datos podrían ser considerados como “outliers”.

Stem-and-Leaf Plot

<http://en.wikipedia.org/wiki/Stemplot>

<http://math.about.com/library/weekly/aa051002a.htm>

El diagrama “Stem-and-Leaf Plot” es un tipo de gráfico, similar a un histograma pero que entrega una información más detallada. ¿Cómo hace esto? Resumiendo la “forma” que un grupo de datos tiene al dividir los mismos en 2 columnas. La primera llamada “Stem” o “rama” que va a la izquierda, en la cual se anotan los valores más altos de cada número, y a la derecha se ubican las “hojas” o “Leafs” en dónde se anotan todos los valores contenidos por éstos números altos (por ejemplo, si tenemos los valores 112, 154 y 165, el “stem” sería igual a 1, y las hojas serían 12 , 54 y 65). Al organizar los datos de esta forma, queda un diagrama que tiene la forma de una rama con hojas (de ahí su nombre).

Por ejemplo, de éste set de datos:

44 46 47 49 63 64 66 68 68 72 72 75 76 81 84 88 106

Queda el siguiente diagrama:

```

4 | 4 6 7 9
5 |
6 | 3 4 6 8 8
7 | 2 2 5 6
8 | 1 4 8
9 |
10 | 6
    
```

Estos diagramas se suelen utilizar cuando hay grandes cantidades de datos, y se quieren conocer sus características.