

**DISEÑO DE BASES DE DATOS: LABORATORIO 3**  
**ANÁLISIS DE EVENTOS SOCIALES MASIVOS**

**ELÍAS GONZÁLEZ**  
**JOSÉ LATAPIATT**  
**IAN ORELLANA**

Profesora: Carolina Bonacic  
Ayudantes: Fabián Arismendi  
Miguel Cárcamo



# TABLA DE CONTENIDOS

<b>ÍNDICE DE FIGURAS.....</b>	<b>iv</b>
<b>CAPÍTULO 1. INTRODUCCIÓN .....</b>	<b>5</b>
<b>CAPÍTULO 2. OBJETIVOS.....</b>	<b>6</b>
2.1    OBJETIVO GENERAL . . . . .	6
2.2    OBJETIVOS ESPECÍFICOS . . . . .	6
<b>CAPÍTULO 3. MODELO UTILIZADO .....</b>	<b>7</b>
<b>CAPÍTULO 4. DESCRIPCIÓN DE LA APLICACIÓN.....</b>	<b>12</b>
<b>CAPÍTULO 5. DESCRIPCIÓN DE CONSULTAS SQL .....</b>	<b>13</b>
<b>CAPÍTULO 6. CONCLUSIÓN .....</b>	<b>15</b>
<b>CAPÍTULO 7. REFERENCIAS.....</b>	<b>16</b>

## ÍNDICE DE FIGURAS

3.1	Modelo Físico de bases de datos. . . . .	7
3.2	Entidades y atributos del modelo físico. Sección 1 . . . . .	8
3.3	Entidades y atributos del modelo físico. Sección 2 . . . . .	9

## **CAPÍTULO 1. INTRODUCCIÓN**

En la experiencia anterior, se hicieron correcciones sobre el modelo con el objetivo de distribuir de mejor forma los datos que se reciben de la API de twitter, además de crear tablas no consideradas para almacenar datos de utilidad.

En esta experiencia se busca mostrar una estadística real representada por los datos obtenidos, utilizando el modelo creado junto con las consultas actualizadas con respecto a la experiencia anterior.

## **CAPÍTULO 2. OBJETIVOS**

### **2.1 OBJETIVO GENERAL**

Crear vistas que muestren estadísticas con respecto a los eventos masivos de entretenimiento

### **2.2 OBJETIVOS ESPECÍFICOS**

- Especificar características de la aplicación.
- Describir y actualizar las consultas SQL realizadas.
- Mostrar estadísticas obtenidas en base al modelo.

## CAPÍTULO 3. MODELO UTILIZADO

Con el objetivo de producir una descripción base de la implementación de la base de datos, se muestra el modelo físico de bases de datos, el cuál expone además las relaciones de mayor importancia o relaciones base, donde es necesario almacenar ciertos datos puesto que el tipo de relación es de muchos a muchos.

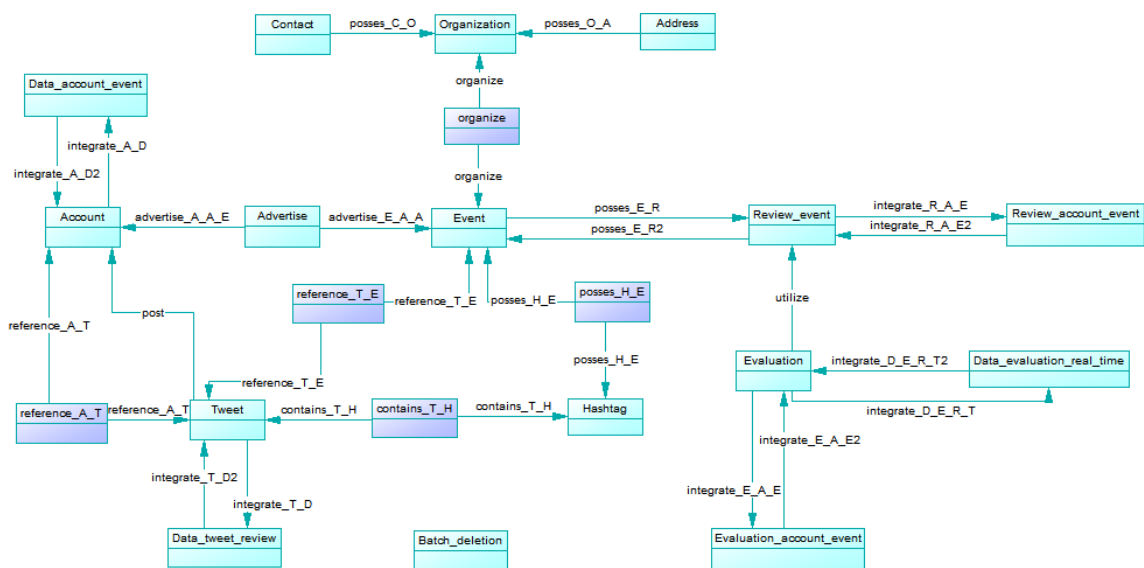


Figura 3.1: Modelo Físico de bases de datos.

Data_account_event	Contact	Address
<u>ID_DAE</u> ID_account location url_twitter_event url_official num_tweets num_photo_and_video num_following num_followers account_official date_creation_account date_update_DAE	<u>ID_contact</u> ID_organization telefonos_contact emails_contact country_contact date_update_contact	<u>ID_address</u> ID_organization national headquarters address_all_in date_update_address
	Organization	Advertise
	<u>ID_organization</u> name_organization url_organization trademarks date_update_organization	<u>ID_advertise</u> ID_account ID_event date_origin_advertise date_end_advertise
Account	Tweet	Review_event
<u>ID_account</u> ID_DAE user_twitter user_name date_ingress event	<u>ID_tweet</u> ID_account ID_DTR num_favorite num_retweet num_reply date_publish_tweet date_insert_BD	<u>ID_review_event</u> ID_event ID_RAE start_date_review ending_date_review num_of_mentions num_active_users passing_rate
Event	Evaluation	Hashtag
<u>ID_event</u> ID_review_event name_event type_of_event description event_location event_price_all_in num_turnout comuna_event start_date_event ending_date_event date_update_event	<u>ID_evaluation</u> ID_review_event ID_DETR ID_EAE date_start_evaluation date_end_evaluation num_mentions num_mentions_positive num_mentions_negative num_user_active evaluation_real_time date_update_evaluation	<u>ID_hashtag</u> date_publish_hashtag official num_mentions_hashtag date_update_hashtag hashtag_content
		Data_evaluation_real_time
		<u>ID_DETR</u> ID_evaluation

Figura 3.2: Entidades y atributos del modelo físico. Sección 1



Review_account_event	Data_tweet_review	Batch_deletion
<u>ID_RAE</u>	<u>ID_DTR</u>	<u>ID_Batch_deletion</u>
ID_review_event	ID_tweet	date_deletion
num_followers_start_review	geo	num_tweets_delete
num_followers_end_review	state	date_until_delete_tweets
date_update_RAE	geo_cord	num_account_delete
posses_H_E	city_tweet	date_until_delete_account
<u>ID_event</u>	comuna_tweet	num_evaluation_delete
<u>ID_hashtag</u>	content_tweet	date_delete_evaluation
organize	sensitivity_rank	
<u>ID_event</u>	date_update_DTR	Evaluation_account_event
ID_organization	reference_T_E	<u>ID_EAE</u>
contains_T_H	<u>ID_event</u>	ID_evaluation
<u>ID_hashtag</u>	<u>ID_tweet</u>	num_followers_start
<u>ID_tweet</u>	reference_A_T	num_followers_end
	<u>ID_account</u>	date_update_EAE
	<u>ID_tweet</u>	

Figura 3.3: Entidades y atributos del modelo físico. Sección 2

*Data\_account\_event*, es una entidad dirigida a soportar datos correspondientes a las cuentas asociadas a los eventos, entre ellos localización, número de tweets, número de seguidores, fechas en que comenzó a publicar y distingue un atributo exclusivo para la página oficial del evento, diferenciándola del que posee el evento en twitter.

*Account* contiene los datos específicos de un elemento de estudio, como lo son su twitter y su nombre, además posee un dato tipo *boolean* para identificar si corresponde a una cuenta oficial de un evento.

*Tweet* se encarga de contar las veces que un tweet fue retweeteado, compartido o añadido a favoritos, acompañado de su fecha de publicación y .a fecha en que fue añadida a la base de datos. Posee dos tipos de relación con *Account*, puesto que una cuenta puede publicar varios tweets (relación uno a muchos), pero por otro lado, una cuenta puede ser referenciada por varios tweets y a la vez referenciar a muchos otros.

*Event* captura datos y características del evento, en su mayor parte son los que no se obtienen de twitter, tales como comuna, tipo de evento o descripción. Además añade fechas de inicio y término si es un evento que se lleva a cabo por más de un día.

*Hashtag* almacena información importante de los hashtag, como su contenido, fecha, hora y número de veces que fue mencionado.

*Address* y *Contact*, como su nombre lo indica, guardan información acerca de la dirección física de la organizadora del evento y el contacto por cualquier medio con esta.

*Batch\_deletion* la cantidad de tweets que son borrados y la fechas útiles para evaluar cuando un tweet debe ser eliminado de la base de datos. *reference\_A\_T* es una relación que contiene los identificadores de una cuenta y un tweet que realiza. Esta relación hace referencia a que una cuenta puede referenciar a distintos tweets, y a la vez ser referenciada por varios tweets.

*reference\_T\_E* indica que un tweet puede referenciar a varios eventos y un evento puede referenciar a varios tweets. Almacena tanto la identificación del evento como la del tweet.

*posses\_H\_E*, indica que un evento puede poseer varios hashtag que lo referencien, y a la vez

*contains\_T\_H* hace referencia a la que en un tweet se pueden incluir varios hashtag, y a la vez, un hashtag puede ser referenciado en muchos tweets. Guarda la identificación del tweet y el hashtag utilizado.

*organize* ya que una organización puede organizar varios eventos, y a la vez un evento puede estar a cargo de varias organizaciones, esta relación se encarga de guardar las identificaciones tanto de la organización como del evento. *Organización* contiene los datos principales de la organización que dirige el evento, nombre, dirección de su página web y marcas asociadas.

*Advertise*, es una tabla que conecta *Account* y *Event*, y almacena las fechas de comienzo y fin de publicidad de un evento por parte de una cuenta.

*Evaluation* contiene información sobre el análisis realizado del evento en un plazo determinado, además, algunas evaluaciones servirán para persistir información en tiempo real en caso de falla de sistema, contiene los datos que se usarán para crear las estadísticas de cierto evento de acuerdo a sus menciones positivas o negativas.

*Review\_event* posee la información final del análisis realizado para determinado evento, para esto maneja atributos como la cantidad total de las menciones positivas y negativas.

*Evaluation\_acount\_event* mantiene el número de seguidores a un evento en un tiempo determinado.

*Data\_evaluation\_real\_time* mantiene información sobre eventuales evaluaciones en tiempo real.

## **CAPÍTULO 4. DESCRIPCIÓN DE LA APLICACIÓN**

El observatorio es capaz de realizar una serie de acciones necesarias para mostrar estadísticas acerca de los eventos masivos de entretenimiento:

- El sistema es capaz de captar tweets, distribuyendo en las tablas del modelo propuesto, todos los valores requeridos como fecha de emisión, contenido positivo o negativo, evento, etc.
- El sistema puede recibir la inclusión de un hashtag, asociarlo a un evento y buscar las veces que ha sido utilizado.
- El sistema provee un formulario que permite ingresar un nuevo evento con sus características generales con objetivo de que el usuario tenga conocimiento de los detalles del evento como lugar, costo de entradas, etc.
- Luego de ingresar eventos, el sistema los posiciona en un ranking ordenados según su numero de menciones en tweets o hashtags.
- El sistema debe clasificar los tweets según su intención positiva o negativa hacia cualquier aspecto del evento. El usuario puede acceder a estos datos los cuales el sistema presentará clasificados en número de menciones positivas y negativas, y sus respectivos porcentajes los cuales serán representados por gráficos de torta.
- El sistema debe mostrar distintos detalles de los eventos incluyendo lugar donde se realiza, valores de las entradas, fecha, hora, empresa organizadora y tipo de evento (datos de información convencionales respecto a cada evento).
- El usuario tiene la posibilidad de comparar cada aspecto anteriormente descrito entre los eventos, esto incluye las distintas versiones del mismo evento.

## CAPÍTULO 5. DESCRIPCIÓN DE CONSULTAS SQL

En esta sección se exhibirán consultas SQL realizadas en la aplicación utilizando Ruby on Rails, junto a una descripción de su utilidad para completar los objetivos de la aplicación.

- Obtener eventos más populares:

```
@ranks = ReviewEvent.joins(:event).order(num_of_mentions: :desc)
      .pluck(:name_event, :num_of_mentions);
```

Obtiene el número de menciones de la tabla ReviewEvent, utilizándolo como criterio de orden descendente, exhibiendo los tweets con más menciones.

- Seleccionar el tipo de evento

```
c = Event.where('type\_of\_event = ?', '<TIPO>')
```

Obtiene la tabla con los eventos que pertenecen al tipo de evento seleccionado por el usuario.

- Tweets por periodo de tiempo

```
d = Tweet.where('date\_publish\_tweet > ? AND
date\_publish\_tweet < ?', datetime1, datetime2)
```

Obtiene de la tabla tweets, aquellos datos que cumplan con las restricciones de tiempo ingresadas. Esta consulta entrega los datos útiles para usuarios que se interesan en la cantidad de tweets emitidos durante cierto periodo, y en el caso de requerir la variación de volumen a lo largo del tiempo, se obtienen el nombre del evento y la fecha y hora en que el tweet fue emitido.

- Obtener cantidad de tweets positivos o negativos del evento

```
@goodvsevilsingle = ReviewEvent.joins(:event).where('name_event = ?',  
'<nombre>').pluck(:name_event, :num_mentions_negative_sum,  
:num_mentions_positive_sum).first
```

Se realiza un join entre las tablas event y ReviewEvent, luego se entregan las estadísticas del número de menciones positivas y negativas.

## **CAPÍTULO 6. CONCLUSIÓN**

En esta experiencia se logra detallar las consultas sql y adaptarla al formato que próximamente será añadido a la aplicación. Con el conocimiento adquirido a lo largo del semestre se fue capaz de adquirir Tweets, almacenarlos en una base de datos con una distribución que corresponde a las utilidades que necesitamos, y poder realizar consultas para obtener estadísticas que implementan los requisitos de la aplicación propuesta.

## **CAPÍTULO 7. REFERENCIAS**

Twitter Developers (2014). GET statuses/firehose. [ONLINE] Available at:  
<https://dev.twitter.com/streaming/reference/get/statuses/firehose>. [Last Accessed e.g. 31 August 11].

2014 Twitter, Inc. Twitter API limits. [ONLINE] Available at:  
<https://support.twitter.com/articles/160385-twitter-api-limits#> [Last Accessed e.g. 31 August 11].

Active Record Query Interface [ONLINE] Available at:  
[http://guides.rubyonrails.org/active\\_record\\_querying.html](http://guides.rubyonrails.org/active_record_querying.html) [Last Accessed e.g. 31 August 11].