

## Práctica 4

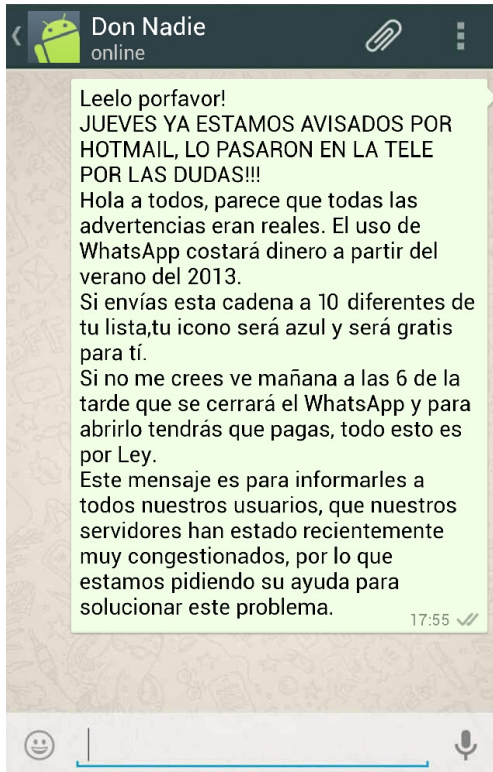
## Todo esto es por ley

**Fecha de entrega:** 11 de junio de 2014.

**Porcentaje en la calificación total de prácticas:** 10 % (es decir, 3 % de la nota final de la asignatura).

Esta práctica se divide en cuatro apartados. De estos, son obligatorios los tres primeros, con los que se obtendrá una calificación de APTO (5), que equivale a 0.15 puntos en la nota final de la asignatura. Para obtener la calificación de APTO (10) (es decir, 0.3 puntos en la nota final) se deberá realizar el cuarto apartado de la práctica.

Se valorará la corrección y la simplicidad de las reglas implementadas en *Datalog*.



¿Cómo? ¿No os enterasteis? No me extraña. Yo ya ni me acuerdo de la última vez que miré mi correo de *Hotmail*.

Para aquellos que os estéis lamentando de no tener el famoso icono azul, os alegrará saber que esto no fue más que un bulo, como otros tantos que circulan por la red. Apareció allá por abril de 2013, aunque no se propagó demasiado, pues resulta curioso que una empresa con unos servidores 'muy congestionados' anime a sus usuarios a reenviar mensajes en masa para 'solucionar este problema'.

Este es el tema de la presente práctica: bulos en la red. Partiremos de una base de datos *Datalog* que contiene información sobre los mensajes gestionados por un ficticio servicio de mensajería instantánea, y obtendremos información sobre los usuarios que propagan (o no) este tipo de bulos.

Supondremos que cada usuario del servicio de mensajería está identificado con un número. Por otro lado, cada mensaje también tendrá un identificador. Dispondremos de una relación *envia* con tres parámetros: el identificador del emisor, el identificador del mensaje enviado, y el identificador del receptor.

Por ejemplo, supongamos que el identificador del mensaje de la figura es el 27. Los siguientes hechos denotan que el usuario 1 envió este mensaje al usuario 3, y que este último lo reenvió al usuario 6:

```
envia(1, 27, 3).
```

```
envia(3, 27, 6).
```

Supondremos que ningún usuario se envía mensajes a sí mismo.

En el Campus Virtual encontrarás un fichero *Datalog* con una serie de hechos *envia*. El objetivo de la práctica es la implementación de reglas que obtengan la información que se indica en los siguientes apartados. Para entregar la práctica en el Campus Virtual **no tenéis que subir ningún fichero**; tan solo hay que copiar las reglas implementadas en el cuadro de texto que aparecerá a la hora de entregar la práctica.

### 1 El icono azul de *WhatsApp*.

Suponiendo que el mensaje mostrado en la figura anterior tiene como identificador el número 27, define las reglas necesarias para definir una relación *icono\_azul(Usuario)* que contenga aquellos usuarios que han reenviado dicho mensaje a diez personas o más. Utiliza, para ello, *group\_by*. El resultado de consultar esta relación ha de ser el siguiente:

```
DES> icono_azul(Usuario)
{
  icono_azul(21)
}
```

### 2 Reenvía este mensaje o sufrirás.

El mensaje con identificador 3 amenaza a aquellos usuarios que lo reciban y no lo reenvíen a ninguna otra persona con introducir un virus en el sistema que cambie el tipo de letra de todos los programas a Comic Sans. Sin utilizar *group\_by*, escribe una relación *comic\_sans(Usuario)* que contenga los usuarios del servicio que sufrirán esta horrible maldición.

```
DES> comic_sans(Usuario).
{
  comic_sans(29),
  comic_sans(37)
}
```

### 3 Reenvía este mensaje y algo bueno te pasará.

El mensaje con identificador 14 reza lo siguiente:

*Si envías este mensaje al menos a 10 personas que a su vez lo reenvíen a otras 3 personas, tu icono de WhatsApp se volverá de color violeta, tendrás buena suerte para toda la vida y aparecerá un arco iris triple en el cielo.*

Escribe una relación *arco\_iris\_triple(Usuario)* que contenga aquellos afortunados usuarios que verán el arco iris triple. Para verlo es necesario enviar el mensaje a un número  $N$  de personas ( $N \geq 10$ ) y que al menos 10 de esos  $N$  usuarios reenvíen el mensaje a 3 o más personas (que pueden incluir a la que envió el mensaje inicialmente).

```
DES> triple_arco_iris(Usuario)
{
  triple_arco_iris(91)
}
```

*Indicación:* Utiliza una relación auxiliar `reenvio_doble(Origen, Mensaje, Destino)` que contenga los mensajes enviados entre `Origen` y `Destino` que a su vez hayan sido reenviados por `Destino` tres veces o más.

#### 4 Bill Gates y el timo de la pirámide.

El año 2002 nos sorprendió con el siguiente bulo en forma de correo:

*Estimados Amigos, Por favor no tome esto por una carta basura.*

*Bill Gates está compartiendo su fortuna. Si usted ignora esto se arrepentirá después. Microsoft y AOL son ahora las mayores compañías de Internet y en un esfuerzo por asegurarse de que Internet Explorer permanezca como el programa más usado, Microsoft y AOL están realizando una prueba beta de e-mail. Si usted reenvía este e-mail a sus amigos, Microsoft puede rastrearlo y lo hará (si usted es usuario de Microsoft Windows) en un periodo de tiempo de una semana. Por cada persona a la que usted remita este e-mail Microsoft le pagará \$245.00, por cada persona que reenvíe el mensaje que usted le envió, Microsoft le pagará \$243.00 y por cada tercera persona que lo reciba, a usted se le pagará \$241.00.*

*En dos semanas, Microsoft se pondrá en contacto con usted para pedirle su dirección y entonces le enviará un cheque. [...]*

Supongamos que nuestro sistema de mensajería contiene este mensaje con el identificador 19. Para simplificar supondremos que un usuario ganaría una cantidad de \$245.00 por cada persona a la que envíe el mensaje, y esa misma cantidad por cada uno de los reenvíos de estas últimas personas, y esa misma cantidad por cada uno de los reenvíos de los reenvíos, etc. sin límite en la longitud de la cadena de reenvíos. Escribe las reglas de una relación `fortuna_usuario(Usuario, CantidadDinero)` que indique la cantidad de dinero que supuestamente recibiría un usuario por los mensajes con identificador 19 que envíe directa e indirectamente (es decir, a través de otros usuarios que lo reenvían). En esta cantidad **no** se contabilizan las cadenas de reenvíos que involucren al usuario original.

Por ejemplo, dados los siguientes envíos:

```
envia(7, 19, 4).
envia(7, 19, 5).
envia(4, 19, 1).
envia(1, 19, 3).
envia(1, 19, 7).
```

Se debe obtener el siguiente resultado:

```
DES> fortuna_usuario(7, Suma)
{
  fortuna_usuario(7,980)
}
```

*Indicación:* Define una relación auxiliar que relacione cada usuario con los usuarios a los que les envía el mensaje nº 19, bien sea directamente, bien a través de una cadena de reenvíos.