

Universidad de Alcalá

Escuela Politécnica Superior

Grado en Ingeniería Informática

Trabajo Fin de Grado

Análisis de datos en redes sociales: Caso de estudio aplicado en
Twitter

Autor: David Márquez Mínguez

Tutor: Juan José Cuadrado Gallego

2021

UNIVERSIDAD DE ALCALÁ

ESCUELA POLITÉCNICA SUPERIOR

Grado en Ingeniería Informática

Trabajo Fin de Grado

**Análisis de datos en redes sociales: Caso de estudio aplicado en
Twitter**

Autor: David Márquez Mínguez

Tutor: Juan José Cuadrado Gallego

Tribunal:

Presidente: Name of the tribunal president

Vocal 1º: Name of the first vocal

Vocal 2º: Name of the second vocal

Fecha de depósito: X de X de 2021

A nuestros alumnos pasados, presentes y futuros...

“Empieza haciendo lo necesario, luego haz lo posible y de pronto empezarás a hacer lo imposible.”

Francisco de Asís

Agradecimientos

A todos los que la presente vieron y entendieron.

Inicio de las Leyes Orgánicas. Juan Carlos I

Aqui va la parte de agradecimientos.....

Resumen

Resumen..... correo de contacto: David Márquez Mínguez <david.marquez@edu.uah.es>.

Palabras clave: Trabajo fin de /grado, L^AT_EX, soporte de español e inglés, hasta cinco....

Abstract

Abstract..... contact email: David Márquez Mínguez <david.marquez@edu.uah.es>.

Keywords: Bachelor final project , L^AT_EX, English/Spanish support, maximum of five....

Resumen extendido

Con un máximo de cuatro o cinco páginas. Se supone que sólo está definido como obligatorio para los TFGs y PFCs de UAH.

Índice general

Resumen	ix
Abstract	xi
Resumen extendido	xiii
Índice general	xv
Índice de figuras	xvii
Índice de tablas	xix
Índice de listados de código fuente	xxi
Índice de algoritmos	xxiii
Lista de acrónimos	xxiii
Lista de símbolos	xxiii
1 Ejemplo práctico: Minería de textos con R	1
2 Presupuesto	3
Bibliografía	5
Apéndice A Manual de usuario	7
A.1 Introducción	7
A.2 Manual	7
A.3 Ejemplos de inclusión de fragmentos de código fuente	7
A.4 Ejemplos de inclusión de algoritmos	9
Apéndice B Herramientas y recursos	11
Apéndice C Versiones	13

Índice de figuras

Índice de tablas

Índice de listados de código fuente

A.1	Ejemplo de código fuente con un <code>lstinputlisting</code> dentro de un <code>codefloat</code>	8
A.2	Ejemplo de código fuente con estilo <code>Cnice</code> , de nuevo con un <code>lstinputlisting</code> dentro de un <code>codefloat</code>	8
A.3	Ejemplo de código fuente con estilo <code>Cnice</code> , modificado para que no aparezca la numeración.	9
A.4	Ejemplo con colores usando el estilo <code>Ccolor</code>	9

Índice de algoritmos

A.1	How to write algorithms	10
A.2	IntervalRestriction	10

Capítulo 1

Ejemplo práctico: Minería de textos con R

Twitter es actualmente una dinámica e ingente fuente de contenidos, que dada su popularidad e impacto, se ha convertido en la principal fuente de información para estudios relacionados con Social Media Analytics.

Capítulo 2

Presupuesto

Blah, blah, blah.

Bibliografía

Apéndice A

Manual de usuario

A.1 Introducción

Blah, blah, blah...

A.2 Manual

Pues eso.

A.3 Ejemplos de inclusión de fragmentos de código fuente

Para la inclusión de código fuente se utiliza el paquete `listings`, para el que se han definido algunos estilos de ejemplo que pueden verse en el fichero `config/preamble.tex` y que se usan a continuación.

Así se inserta código fuente, usando el estilo `CppExample` que hemos definido en el `preamble`, escribiendo el código directamente :

```
#include <stdio.h>

// Esto es una funcion de prueba
void funcionPrueba(int argumento)
{
    int prueba = 1;

    printf("Esto_es_una_prueba_ [%d] [%d]\n", argumento, prueba);
}
```

O bien insertando directamente código de un fichero externo, como en el ejemplo [A.1](#), usando `\lstinputlisting` y cambiando el estilo a `Cbluebox` (además de usar el entorno `codefloat` para evitar `pagebreaks`, etc.).

Listado A.1: Ejemplo de código fuente con un `lstinputlisting` dentro de un `codefloat`

```
#include <stdio.h>

// Esto es una función de prueba
void funcionPrueba(int argumento)
{
    int prueba = 1;

    printf("Esto es una prueba [%d][%d]\n", argumento, prueba);
}
```

O por ejemplo en matlab, definiendo settings en lugar de usar estilos definidos:

```
%
% add_simple.m - Simple matlab script to run with condor
%
a = 9;
b = 10;

c = a+b;

fprintf(1, 'La suma de %d y %d es igual a %d\n', a, b, c);
```

O incluso como en el listado A.2, usando un layout más refinado (con los settings de <http://www.rafalinux.com/?p=599> en un `lststyle Cnice`).

Listado A.2: Ejemplo de código fuente con estilo `Cnice`, de nuevo con un `lstinputlisting` dentro de un `codefloat`

```
1  #include <stdio.h>
2
3  #define LOOP_TIMES 5
4
5  int main(int argc, char* argv[])
6  {
7      int i;
8
9      for (i = 1; i < LOOP_TIMES; i++)
10         puts("Hola mundo!");
11 }
```

Y podemos reutilizar estilos cambiando algún parámetro, como podemos ver en el listado A.3, en el que hemos vuelto a usar el estilo `Cnice` eliminando la numeración.

Listado A.3: Ejemplo de código fuente con estilo `Cnice`, modificado para que no aparezca la numeración.

```
#include <stdio.h>

#define LOOP_TIMES 5

int main(int argc, char* argv[])
{
    int i;

    for (i = 1; i < LOOP_TIMES; i++)
        puts("Hola mundo!");
}
```

Ahora compila usando `gcc`:

```
$ gcc -o hello hello.c
```

Y también podemos poner ejemplos de código *coloreado*, como se muestra en el [A.4](#).

Listado A.4: Ejemplo con colores usando el estilo `Ccolor`

```
#include <stdio.h>

#define LOOP_TIMES 5

int main(int argc, char* argv[])
{
    int i;

    for (i = 1; i < LOOP_TIMES; i++)
        puts("Hola mundo!");
}
```

Finalmente aquí tenemos un ejemplo de código shell, usando el estilo `BashInputStyle`:

```
#!/bin/sh

HOSTS_ALL="gc000 gc001 gc002 gc003 gc004 gc005 gc006 gc007"

for h in $HOSTS_ALL
do
    echo "Running [$*] in $h..."
    echo -n " "
    ssh root@$h $*
done
```

A.4 Ejemplos de inclusión de algoritmos

En la versión actual (abril de 2014), empezamos a usar el paquete `algorithm2e` para incluir algoritmos, y hay ajustes específicos y dependientes de este paquete tanto en `config/preamble.tex` como en `cover/extralistings.tex` (editadlos según vuestras necesidades).

Hay otras opciones disponibles (por ejemplo las descritas en <http://en.wikibooks.org/wiki/LaTeX/Algorithm>), y podemos abordarlas, pero por el momento nos quedamos con `algorithm2e`.

Incluimos dos ejemplos directamente del manual: uno sencillo en el algoritmo A.1, y otro un poco más complicado en el algoritmo A.2.

Data: this text

Result: how to write algorithm with L^AT_EX2_ε

initialization;

while *not at end of this document* **do**

 read current;

if *understand* **then**

 go to next section;

 current section becomes this one;

else

 go back to the beginning of current section;

Algoritmo A.1: How to write algorithms

Data: $G = (X, U)$ such that G^{tc} is an order.

Result: $G' = (X, V)$ with $V \subseteq U$ such that G'^{tc} is an interval order.

begin

$V \leftarrow U$

$S \leftarrow \emptyset$

for $x \in X$ **do**

$NbSuccInS(x) \leftarrow 0$

$NbPredInMin(x) \leftarrow 0$

$NbPredNotInMin(x) \leftarrow |ImPred(x)|$

for $x \in X$ **do**

if $NbPredInMin(x) = 0$ **and** $NbPredNotInMin(x) = 0$ **then**

 AppendToMin(x)

1 **while** $S \neq \emptyset$ **do**

REM remove x from the list of T of maximal index

2 **while** $|S \cap ImSucc(x)| \neq |S|$ **do**

for $y \in S - ImSucc(x)$ **do**

 { remove from V all the arcs zy : }

for $z \in ImPred(y) \cap Min$ **do**

 remove the arc zy from V

$NbSuccInS(z) \leftarrow NbSuccInS(z) - 1$

 move z in T to the list preceding its present list

 {i.e. If $z \in T[k]$, move z from $T[k]$ to $T[k - 1]$ }

$NbPredInMin(y) \leftarrow 0$

$NbPredNotInMin(y) \leftarrow 0$

$S \leftarrow S - \{y\}$

 AppendToMin(y)

 RemoveFromMin(x)

Algoritmo A.2: IntervalRestriction

Apéndice B

Herramientas y recursos

Las herramientas necesarias para la elaboración del proyecto han sido:

- PC compatible
- Sistema operativo GNU/Linux [?]
- Entorno de desarrollo Emacs [?]
- Entorno de desarrollo KDevelop [?]
- Procesador de textos \LaTeX [?]
- Lenguaje de procesamiento matemático Octave [?]
- Control de versiones CVS [?]
- Compilador C/C++ gcc [?]
- Gestor de compilaciones make [?]

Apéndice C

Versiones

En este apartado incluyo el historial de cambios más relevantes de la plantilla a lo largo del tiempo.

No empecé este apéndice hasta principios de 2015, con lo que se ha perdido parte de la información de los cambios importantes que ha ido sufriendo esta plantilla.

- Mayo 2015:
 - Hay disponible un `make bare` para que deje los capítulos mondos y lirondos y se pueda escribir desde casi cero sin tener que andar borrando manualmente.
- Abril 2015:
 - Ahora manejamos masculino/femenino en algunos sitios (el/la, autor/autora, alumno/alumna, del/de la, ...). Hay que definir variable con el género del autor (todavía queda pendiente lo de los tutores y tal). NOT FINISHED!!
- Enero 2015:
 - Solucionado el problema (gordo) de compilación del `anteproyecto.tex` y el `book.tex`, debido al uso de paths distintos en la compilación de la bibliografía. El sistema se ha complicado un poco (ver `biblio\bibliography.tex`).
 - Añadido un (rudimentario) sistema para generar pdf con las diferencias entre el documento en su estado actual y lo último disponible en el repositorio (usando `latexdiff`).
- Diciembre 2015:
 - Separada la compilación del anteproyecto de la del documento principal. Para el primero se ha creado el directorio `anteproyecto` donde está todo lo necesario.

Universidad de Alcalá
Escuela Politécnica Superior



ESCUELA POLITECNICA
SUPERIOR



Universidad
de Alcalá