



UNIVERSIDAD DE CASTILLA-LA MANCHA
ESCUELA SUPERIOR DE INFORMÁTICA

GRADO EN INGENIERÍA INFORMÁTICA
TECNOLOGÍA ESPECÍFICA DE COMPUTACIÓN

TRABAJO FIN DE GRADO

Sistema Inteligente para la Gestión y Optimización de Energía
basado en la Nube

Pablo Palomino Gómez

Julio, 2019



**UNIVERSIDAD DE CASTILLA-LA MANCHA
ESCUELA SUPERIOR DE INFORMÁTICA**

Departamento de Tecnologías y Sistemas de Información

TECNOLOGÍA ESPECÍFICA DE COMPUTACIÓN

TRABAJO FIN DE GRADO

**Sistema Inteligente para la Gestión y Optimización de
Energía basado en la Nube**

Autor(a): Pablo Palomino Gómez

Director(a): Luis Jiménez Linares

Director(a): Luis Rodríguez Benítez

Julio, 2019

Sistema Inteligente para la Gestión y Optimización de Energía basado en la Nube
© Pablo Palomino Gómez, 2019

Este documento se distribuye con licencia Creative Commons Atribución Compartir Igual 4.0. El texto completo de la licencia puede obtenerse en <https://creativecommons.org/licenses/by-sa/4.0/>.

La copia y distribución de esta obra está permitida en todo el mundo, sin regalías y por cualquier medio, siempre que esta nota sea preservada. Se concede permiso para copiar y distribuir traducciones de este libro desde el español original a otro idioma, siempre que la traducción sea aprobada por el autor del libro y tanto el aviso de copyright como esta nota de permiso, sean preservados en todas las copias.



TRIBUNAL:

Presidente: _____

Vocal: _____

Secretario: _____

FECHA DE DEFENSA: _____

CALIFICACIÓN: _____

PRESIDENTE

VOCAL

SECRETARIO

Fdo.:

Fdo.:

Fdo.:

Dedicatoria
(para siempre)

Resumen

(... versión del resumen en español ...)

El resumen debe ocupar como máximo una página y en dicho espacio proporcionará información crucial sobre el ‘*qué*’ (problemática que trata de resolver el TFG), el ‘*cómo*’ (metodología para llegar a los resultados) y los objetivos alcanzados.

Abstract

(... english version of the abstract ...)

Versión del resumen en inglés. En los trabajos cuyo idioma principal sea el inglés, el orden de Resumen y Abstract se invertirá.

AGRADECIMIENTOS

Aunque es un apartado opcional, haremos bueno el refrán «*Es de bien nacidos, ser agradecidos*» si empleamos este espacio es un medio para agradecer a todos los que, de un modo u otro, han hecho posible que el TFG «llegue a buen puerto». Esta sección es ideal para agradecer a familiares, directores, profesores, compañeros, amigos, etc.

Estos agradecimientos pueden ser tan personales como se desee e incluir anécdotas y chascarrillos, pero nunca deberían ocupar más de una página.

Pablo Palomino Gómez

ÍNDICE GENERAL

Índice de figuras	XVII
Índice de tablas	XIX
Índice de listados	XXI
Índice de algoritmos	XXIII
1. Introducción	1
1.1. Ejemplos de listas	1
1.2. Ejemplos de tablas	2
1.3. Ejemplos de figuras	3
1.4. Ejemplos de listados	4
1.4.1. Algoritmos con el paquete <code>algorithm2e</code>	4
1.5. Menús, paths y teclas con el paquete <code>menukeys</code>	5
2. Objetivo	7
3. Metodología	9
4. Resultados	11
5. Conclusiones	13
A. El primer anexo	15

ÍNDICE DE FIGURAS

1.1. Ejemplo de figura	3
1.2. Ejemplo de subfiguras	3

ÍNDICE DE TABLAS

1.1. Ejemplo de uso de la macro <code>cline</code>	2
1.2. Ejemplo de tabla con especificación de anchura de columna	2

ÍNDICE DE LISTADOS

1.1. Código fuente en Java	4
1.2. Ejemplo de código C	4

ÍNDICE DE ALGORITMOS

1.1. Cómo escribir algoritmos	5
---	---

INTRODUCCIÓN

Este capítulo aborda la motivación del trabajo. Se trata de señalar la necesidad de la que surge, su actualidad y pertinencia. Puede incluir también un estado de la cuestión en la que se revisen estudios o desarrollos previos y en qué medida sirven de base al trabajo que se presenta.

A continuación se muestran algunos ejemplos para la inclusión de elementos en el documento.

1.1. EJEMPLOS DE LISTAS

A continuación se van a añadir algunos ejemplos que pueden emplearse al redactar la memoria.

Ejemplo de lista con *bullet* especial.

- * peras
- manzanas
- ♥ naranjas

Ejemplo de lista compacta (también se puede emplear el entorno para enumeraciones *compactenum*)

- peras
- manzanas
- naranjas

Ejemplo de lista en varias columnas.

- | | |
|-------------|--------------|
| 1. peras | 4. patatas |
| 2. manzanas | 5. calabazas |
| 3. naranjas | 6. fresas |

1.2. EJEMPLOS DE TABLAS

A continuación se incluyen algunos ejemplos de tablas hechas con \LaTeX y paquetes dedicados.

Tabla 1.1: Ejemplo de uso de la macro `cline`

7C0	hexadecimal
3700	octal
11111000000	binario
1984	decimal

Ejemplo de tabla en la que se controla el ancho de la celda.

Tabla 1.2: Ejemplo de tabla con especificación de anchura de columna

Día	Temp Mín (°C)	Temp Máx (°C)	Previsión
Lunes	11	22	Día claro y muy soleado. Sin embargo, la brisa de la tarde puede hacer que las temperaturas desciendan
Martes	9	19	Nuboso con chubascos en muchas regiones. En Cataluña claro con posibilidad de bancos nubosos al norte de la región
Miércoles	10	21	La lluvia continuará por la mañana pero las condiciones climáticas mejorarán considerablemente por la tarde

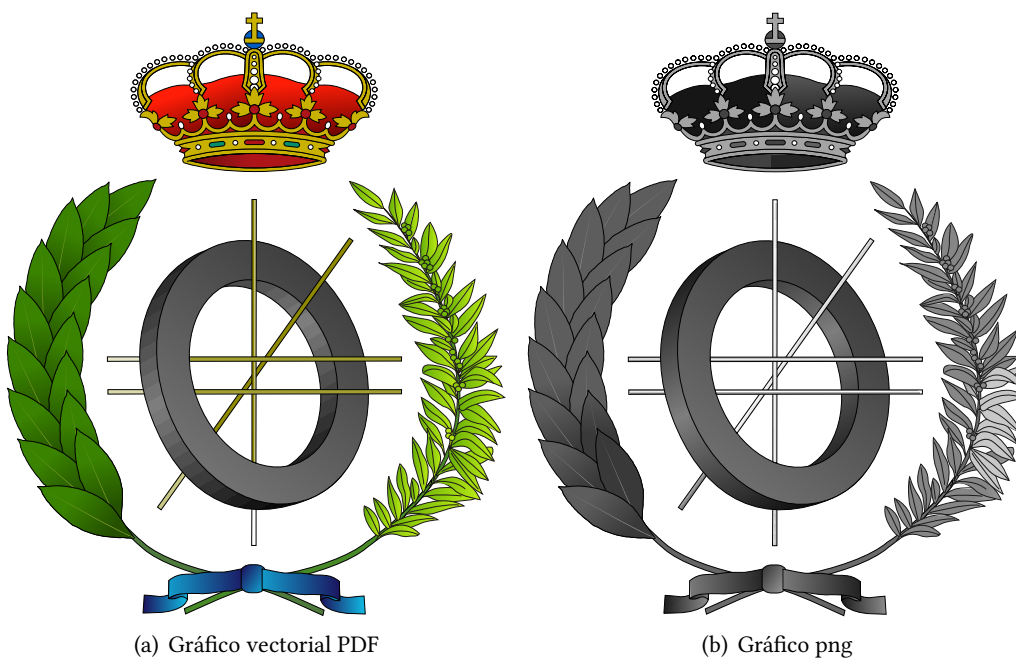
1.3. EJEMPLOS DE FIGURAS

En esta sección se añaden ejemplos de muestra para la inclusión de figuras simples y subfiguras.



Figura 1.1: Figura vectorial del escudo de la ESI

Ejemplo de figuras compuestas por subfiguras.



(a) Gráfico vectorial PDF

(b) Gráfico png

Figura 1.2: Ejemplo de inclusión de subfiguras en un mismo entorno

1.4. EJEMPLOS DE LISTADOS

Ejemplos más representativos de inclusión de porciones de código fuente.

Listado 1.1: Ejemplo de código fuente en lenguaje Java

```

1 // @author www.javadb.com
2 public class Main {
3 // Este método convierte un String a
4 // un vector de bytes
5
6 public void convertStringToByteArray() {
7
8 String stringToConvert = "This_String_is_15";
9 byte[] theByteArray = stringToConvert.getBytes();
10 System.out.println(theByteArray.length);
11 }
12
13 // argumentos de línea de comandos
14 public static void main(String[] args) {
15 new Main().convertStringToByteArray();
16 }
17 }

```

Otro ejemplo.

Listado 1.2: Ejemplo de código C

```

1 // Este código se ha incluido tal cual está
2 // en el fichero LATEX
3 #include <stdio.h>
4 int main(int argc, char* argv[]) {
5 puts(";Hola_mundo!");
6 }

```

Ejemplo de entrada por consola.

```
$ gcc -o Hola_HolaMundo.c
```

1.4.1. Algoritmos con el paquete algorithm2e

Como ya se ha comentado en los textos científicos relacionados con las TIC¹ (Tecnologías de la Información y Comunicaciones) suelen aparecer porciones de código en los que se explica alguna función o característica relevante del trabajo que se expone. Muchas veces lo que se quiere ilustrar es un algoritmo o método en que se ha resuelto un problema abstrayéndose del lenguaje de programación concreto en que se realiza la implementación. El paquete `algorithm2e`² proporciona un entorno `algorithm` para la impresión apropiada de algoritmos tratándolos como objetos flotantes y con muchas flexibilidad de personalización. En el algoritmo 1.1 se muestra cómo puede emplearse dicho paquete. En este curso no se explican las posibilidades del paquete más en profundidad ya que excede el propósito del curso. A todos los interesados se les remite a la documentación del mismo.

¹Por supuesto en un TFG o tesis de una Escuela de Informática.

²<https://osl.ugr.es/CTAN/macros/latex/contrib/algorithm2e/doc/algorithm2e.pdf>

Algoritmo 1.1: Cómo escribir algoritmos

```

Datos : este texto
Resultado : como escribir algoritmos con  $\text{\LaTeX}$ 2e
1 inicialización;
2 while no es el fin del documento do
3   leer actual;
4   if comprendido then
5     ir a la siguiente sección;
6     la sección actual es esta;
7   else
8     ir al principio de la sección actual;
9   end
10 end

```

1.5. MENÚS, PATHS Y TECLAS CON EL PAQUETE MENUKEYS

Cada vez es más usual que los trabajos en ingeniería exijan el uso de software. Para poder especificar de modo elegante el uso menús, pulsación de teclas y directorios se recomienda el uso del paquete `menukeys`.³ Este paquete nos permite especificar el acceso a un menú, por ejemplo:

Herramientas » Órdenes » PDFLaTeX

También un conjunto de teclas. Por ejemplo: `Ctrl` + `↑` + `T`

O un directorio: `C:\user\LaTeX\Ejemplos`

Aunque este paquete permite muchas opciones de configuración de los estilos aplicados, no es necesario hacerlo para obtener unos resultados muy elegantes.

³<https://osl.ugr.es/CTAN/macros/latex/contrib/menukeys/menukeys.pdf>

CAPÍTULO 2

OBJETIVO

Introduce y motiva la problemática (i.e. *¿cuál es el problema que se plantea y porqué es interesante su resolución?*)

Debe concretar y exponer detalladamente el problema a resolver, el entorno de trabajo, la situación y qué se pretende obtener. También puede contemplar las limitaciones y condicionantes a considerar para la resolución del problema (lenguaje de construcción, equipo físico, equipo lógico de base o de apoyo, etc.). Si se considera necesario, esta sección puede titularse *Objetivos del TFG e hipótesis de trabajo*. En este caso, se añadirán las hipótesis de trabajo que el alumno pretende demostrar con su TFG.

Una de las tareas más complicadas al proponer un TFG es plantear su Objetivo. La dificultad deriva de la falta de consenso respecto de lo que se entiende por *objetivo* de un trabajo de esta naturaleza. En primer lugar se debe distinguir entre dos tipos de objetivo:

1. La *finalidad específica* del TFG que se plantea para resolver una problemática concreta aplicando los métodos y herramientas adquiridos durante la formación académica. Por ejemplo, «*Desarrollo de una aplicación software para gestionar reservas hoteleras on-line*».
2. El *propósito académico* que la realización de un TFG tiene en la formación de un graduado. Por ejemplo, la *adquisición de competencias específicas de la especialización cursada*.

En el ámbito de la memoria del TFG se tiene que definir el primer tipo de objetivo, mientras que el segundo tipo de objetivo es el que se añade al elaborar la propuesta de un TFG presentada ante un comité para su aprobación. Este segundo tipo de objetivo no debe incluirse en el apartado correspondiente de la memoria y en todo caso puede valorarse su satisfacción en la sección de resultados y conclusiones.

Un objetivo bien planteado para el TFG debe estar determinado en términos del «*producto final*» esperado que resuelve un problema específico. Es por tanto un sustantivo que debería ser *concreto y medible*. El Objetivo planteado puede pertenecer una de las categorías que se indica a continuación:

- *Diseño y desarrollo de «artefactos»* (habitual en las ingenierías),
- *Estudio* que ofrece información novedosa sobre un tema (usual en las ramas de ciencias y humanidades), y
- *Validación de una hipótesis* de partida (propio de los trabajos científicos y menos habitual en el caso de los TFG).

Estas categorías no son excluyentes, de modo que es posible plantear un trabajo cuyo objetivo sea el diseño y desarrollo de un «artefacto» y éste implique un estudio previo o la validación de alguna hipótesis para guiar el proceso. En este caso y cuando el objetivo sea lo suficientemente amplio puede ser conveniente su descomposición en elementos más simples hablando de *subobjetivos*. Por ejemplo, un programa informático puede descomponerse en módulos o requerir un estudio previo para plantear un nuevo algoritmo que será preciso validar.

La descomposición de un objetivo principal en subobjetivos u objetivos secundarios debería ser natural (no forzada), bien justificada y sólo pertinente en los TFG de gran amplitud.

Junto con la definición del objetivo del TFG se puede especificar los *requisitos* que debe satisfacer la solución aportada. Estos requisitos especifican *características* que debe poseer la solución y *restricciones* que acotan su alcance. En el caso de TFG cuyo objetivo es el desarrollo de un «artefacto» los requisitos pueden ser *funcionales* y *no funcionales*.

Al redactar el objetivo de un TFG se debe evitar confundir los medios con el fin. Así es habitual encontrarse con objetivos definidos en términos de las *acciones* (verbos) o *tareas* que será preciso realizar para llegar al verdadero objetivo. Sin embargo, a la hora de planificar el desarrollo del trabajo si es apropiado descomponer todo el trabajo en *hitos* y estos en *tareas* para facilitar dicha *planificación*.

La categoría del objetivo planteado justifica modificaciones en la organización genérica de la memoria del TFG. Así en el caso de estudios y validación de hipótesis el apartado de resultados y conclusiones debería incluir los resultados de experimentación y los comentarios de cómo dichos resultados validan o refutan la hipótesis planteada.

METODOLOGÍA

En este capítulo se debe detallar las metodologías empleadas para planificación y desarrollo del trabajo, así como explicar de modo claro y conciso cómo se han aplicado dichas metodologías.

RESULTADOS

En esta sección se describirá la aplicación del método de trabajo presentado en el capítulo 3 en este caso concreto, mostrando los elementos (modelos, diagramas, especificaciones, etc.) más importantes. Este apartado debe explicar cómo la metodología satisface los objetivos y requisitos planteados.

CONCLUSIONES

En este capítulo se realizará un juicio crítico y discusión sobre los resultados obtenidos. Si es pertinente deberá incluir información sobre trabajos derivados como publicaciones o ponencias, así como trabajos futuros, solo si estos están planificados en el momento en que se redacta el texto. Además incluirá obligatoriamente la explicación de cómo el trabajo realizado satisface las competencias de la tecnología específica cursada.

EL PRIMER ANEXO

En los anexos se incluirá de modo opcional material suplementario que podrá consistir en breves manuales, listados de código fuente, esquemas, planos, etc. Se recomienda que no sean excesivamente voluminosos, aunque su extensión no estará sometida a regulación por afectar esta únicamente al texto principal.

Bibliografía Esta sección, que si se prefiere puede titularse «Referencias», incluirá un listado por orden alfabético (primer apellido del primer autor) con todas las obras en que se ha basado para la realización del TFG en las que se especificará: autor/es, título, editorial y año de publicación. Solo se incluirán en esta sección las referencias bibliográficas que hayan sido citadas en el documento. Todas las fuentes consultadas no citadas en el documento deberían incluirse en una sección opcional denominada «Material de consulta», aunque preferiblemente estas deberían incluirse como referencias en notas a pie de página a lo largo del documento.

Se usará método de citación numérico con el número de la referencia empleada entre corchetes. La cita podrá incluir el número de página concreto de la referencia que desea citarse. Debe tenerse en cuenta que el uso correcto de la citación implica que debe quedar claro para el lector cuál es el texto, material o idea citado. Las obras referenciadas sin mención explícita o implícita al material concreto citado deberían considerarse material de consulta y por tanto ser agrupados como «Material de consulta» distinguiéndolas claramente de aquellas otras en las que si se recurre a la citación.

Cuando se desee incluir referencias a páginas genéricas de la Web sin mención expresa a un artículo con título y autor definido, dichas referencias podrán hacerse como notas al pie de página o como un apartado dedicado a las «Direcciones de Internet».

Todo el material ajeno deberá ser citado convenientemente sin contravenir los términos de las licencias de uso y distribución de dicho material. Esto se extiende al uso de diagramas y fotografías. El incumplimiento de la legislación vigente en materia de protección de la propiedad intelectual es responsabilidad exclusiva del autor del trabajo independientemente de la cesión de derechos que este haya convenido. De este modo será responsable legal ante cualquier acción judicial derivada del incumplimiento de los preceptos aplicables. Así mismo ante dicha circunstancia los órganos académicos se reservan el derecho a imponer al autor la sanción administrativa que se estime pertinente.

Índice temático Este índice es opcional y se empleará como índice para encontrar los temas tratados en el trabajo. Se organizará de modo alfabético indicando el número de página(s) en el que se aborda el tema concreto señalado.

