Facultad de
Ciencias Sociales y
Tecnologías de la Información
Talavera de la Reina. UCLM

UNIVERSIDAD DE CASTILLA-LA MANCHA

TRABAJO FIN DE GRADO

Diagnostico de patologias mediante el analisis de movimiento via inteligencia artificial

Carlos Rincón González

Mes, Año

UNIVERSIDAD DE CASTILLA-LA MANCHA

TRABAJO FIN DE GRADO

Departamento tutor académico

Tecnología Específica

Diagnostico de patologias mediante el analisis de movimiento via inteligencia artificial

Carlos Rincón González

Félix Albertos Marco

David Carneros Prado

Mes , Año

*D*edicado a mi familia y a todos
aquellos que me apoyaron durante el camino

Yo, Carlos Rincón González con DNI 04235371J, declaro que soy el único autor del trabajo fin de grado titulado ” Diagnostico de patologias mediante el analisis de movimiento via inteligencia artificial” y que el citado trabajo no infringe las leyes en vigor sobre propiedad intelectual y que todo el material no original contenido en dicho trabajo está apropiadamente atribuido a sus legítimos autores.

Talavera de la Reina, a 19 de Diciembre de 2023

Fdo: Carlos Rincón González

Resumen

Esta plantilla puede modificarse para adaptarse a las particularidades de cada Proyecto, tanto en contenido como en formato, siempre y cuando se respete las directrices básicas indicadas en la guía de estilo y formato para la elaboración de TFG del Grado en Ingeniería Informática de la Facultad de Ciencias Sociales y Tecnologías de la Información de Talavera de la Reina.

Abstract

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Contents

List of Figures

List of Tables

Listings

HTML Lenguaje de marcas de hypertexto

HCI Human Computer Interaction

TFG Trabajo Final de Grado

Chapter 1

Introducción

El capítulo de Introducción debe describir el problema que se pretende resolver con el desarrollo del Trabajo Fin de Grado (TFG). Debe dar respuesta al qué sin especificar cómo se va a realizar, para lo cual se usarán el resto de los capítulos del documento. El lector de este documento debe tener claro el alcance del proyecto habiendo leído únicamente el capítulo de Introducción.

1.1 Motivación

Esta sección aborda la motivación del trabajo. Se trata de señalar la necesidad que lo origina, su actualidad y pertinencia. Puede incluir también un estado de la cuestión (o estado del arte) en la que se revisen estudios o desarrollos previos y en qué medida sirven de base al trabajo que se presenta.

En este capítulo debería introducirse el contexto disciplinar y tecnológico en el que se desarrolla el trabajo de modo que pueda entenderse con facilidad el ámbito y alcance del TFG. Puesto que un TFG no tiene que ser necesariamente un trabajo con aportes novedosos u originales, solo es necesario la inclusión de estado del arte cuando este contribuya a aclarar aspectos clave del TFG o se desee justificar la originalidad del trabajo realizado. Si la sección estado del arte es muy extensa, considera la opción de introducirla como un capítulo independiente.

1.2 Motivation

Health is a problem that never can scape us, wether we want it or not, we get sick from time to time, and illneses are a main cause of

In today's world health and its studies have improved a lot. Nevertheless, the improvement must

never stop, and helping this improvement is part of us as programmers and technicians. This improvement can often come as a part of studies and research on new technologies and innovations on old ones, and also it can come as the development of the structure behind the system itself. The field of computer sience is an ever developing field, and new tecnologies emerge every day giving us oportunities to make the world a better place. The new hotness is Artificial Inteligence, and we thing we can use it to develop new tools that can help us. In this case we are using a type of artificial inteligence that In this project we aim to tackle both fronts, we aim to develop a system that can help us recognize patterns in illnesses so they help specialists with their work, and also we want to create a tool for them to use so the system is less cluttered and

1.3 Redacción de la Memoria

Durante la realización de la memoria del TFG es importante tener presente respetar la guía de estilo de la institución. Por tanto, el empleo de plantillas para un sistema de procesamiento de textos (por ejemplo, Word o LaTeX) puede requerir su adaptación cuando la plantilla mencionada no haya sido suministrada en la institución a la que se dirige el trabajo.

Para redactar un trabajo académico de modo efectivo se deben tener presentes una serie de normas que ayuden a conseguir un resultado final que sea claro y de fácil lectura.

A la hora de redactar el texto se debe poner especial atención en no cometer plagio y respetar los derechos de propiedad intelectual. En particular merece gran atención la inclusión de gráficos e imágenes procedentes de Internet que no sean de elaboración propia. En este sentido se recomienda consultar el manual de la Universidad de Cantabria [1] en el que se explica de modo conciso cómo incluir imágenes en un trabajo académico.

1.4 Estructura del Documento

Este capítulo suele incluir una sección que indica la estructura (capítulos y anexos) del documento y el contenido de cada una de las partes en que se divide. Por tanto, las secciones que suelen acompañar este capítulo son:

- Motivación. Responde a la pregunta sobre la necesidad o pertinencia del trabajo.
- Objetivo. Determina de modo claro el propósito del trabajo descrito que puede desglosarse en subobjetivos cuando el objetivo principal se puede descomponer en módulos o componentes. Es muy importante definir el objetivo de modo apropiado. El Capítulo Objetivos de esta guía explica cómo definir el objetivo.

[1] Guía de Imágenes: https://web.unican.es/buc/Documents/Formacion/guia_imagenes.pdf

- Antecedentes o Contexto disciplinar/tecnológico. También puede denominarse Estado del Arte cuando se trata de comentar trabajos relacionados que han abordado la cuestión u objetivo que se plantea.
- Estructura del documento. Resumen de los capítulos y anexos que integran el documento.

1.5 Abreviaturas y Acrónimos

Un TFG que utiliza muchas abreviaturas y acrónimos puede añadir esta sección dónde se muestra el conjunto de abreviaturas y acrónimos y su significado.

Chapter 2

Objetivos

Para hacer un planteamiento apropiado de los objetivos se recomienda utilizar la Guía para la elaboración de propuestas de TFG en la que se explica cómo definir correctamente los objetivos de un TFG.

2.1 Objetivo General

Introduce y motiva la problemática (i.e ¿cuál es el problema que se plantea y por qué 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/la estudiante 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 en un trabajo de esta naturaleza.

En primer lugar, se debe distinguir entre dos tipos de objetivo:

- 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.
- 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 intensificación cursada.

En el ámbito de la memoria del TFG se tiene que definir el primer tipo de objetivo, mientras que el segundo tipo es el que se añade en el Capítulo de Conclusiones y que justifica las competencias específicas de la intensificación alcanzadas y/o reforzadas con la realización del trabajo.

La categoría del objetivo planteado justifica modificaciones en la organización genérica de la memoria del trabajo. 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.

2.2 Objetivos Específicos

Generalmente, el objetivo general puede ser descompuesto en varios objetivos más específicos que se pretenden alcanzar. En esta sección se enumeran y describen cada uno de ellos.

Junto con la definición de estos objetivos 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 un trabajo 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 trabajo. 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.

Chapter 3

Objectives

3.1 O1: Obtención de los datos de movimiento a partir de una cámara RGB

Vamos a tener en cuenta los distintos tipos de camaras a utilizar para poder procesar los datos, estas camaras tienen distintas capacidades y caracteristicas. Vamos a analizar las diferencias entre ellas y vamos a ver la forma en la que podemos hacer uso de sus datos. En nuestro caso vamos a usar camaras RGB, las cuales no tienen percepción de la profundidad por si mismas, por lo cual tenemos que investigar la forma de transcribir la profundidad a ellas. No solo es relevante la camara que usemos ya que el software que se use para procesar lo visto por la cmaara tambien es relevante. Por ello vamos a investigar las distintas formas de captura de datos y su procesado. ## O2: Validación del sistema de obtención de datos para el análisis de movimiento en entornos clínicos

En contraste con otros tipos de camara nuestra elección, la camara RGB, no es una camara con validación cientifica previa. Por esto tenemos que investigar como se puede hacer el proceso de validación de la camara y los datos que obtenemos de ella. Los datos necesitan ser validados ya no solo cientificamente

3.2 O3: Definición de los datos a utilizar y su generalización para el entrenamiento de modelos de IA

3.3 O4: Comparación de diferentes métodos para el entrenamiento de modelos de IA para el diagnóstico de la patología del caso de estudio inicial

3.4 O5: Desarrollo de una herramienta para el entrenamiento de modelos de IA para el caso de estudio inicial

3.5 O6: Validación del sistema de diagnóstico para la patología del caso de estudio inicial

Chapter 4

Metodología

En este apartado se deben indicar las metodologías empleadas para planificación y desarrollo del TFG, así como explicar de modo claro y conciso cómo se han aplicado dichas metodologías.

4.1 Guía Rápida de las Metodologías de Desarrollo del Software

A continuación, se incluye una guía rápida que puede ser de gran utilidad en la elaboración de este capítulo.

4.1.1 Proceso de Desarrollo de Software

El proceso de desarrollo de software se denomina también ciclo de vida del desarrollo del software (SDLC, Software Development Life-Cycle) y cubre las siguientes actividades:

- Obtención y análisis de requisitos (requirements analysis). Es la definición de la funcionalidad del software a desarrollar. Suele requerir entrevistas entre los ingenieros de software y el cliente para obtener el 'qué' y 'cómo'. Permite obtener una especificación funcional del software.
- Diseño (SW design). Consiste en la definición de la arquitectura, los componentes, las interfaces y otras características del sistema o sus componentes.
- Implementación (SW construction and coding). Es el proceso de codificación del software en un lenguaje de programación. Constituye la fase en que tiene lugar el desarrollo de software.

- Pruebas (testing and verification). Verificación del correcto funcionamiento del software para detectar fallos lo antes posible. Persigue la obtención de software de calidad. Consisten en pruebas de caja negra y caja blanca. Las primeras comprueban que la funcionalidad es la esperada y para ello se verifica que, ante un conjunto amplio de entradas, la salida es correcta. Con las segundas se comprueba la robustez del código sometiéndolo a pruebas cuya finalidad es provocar fallos de software. Esta fase también incorpora las pruebas de integración en las que se verifica la interoperabilidad del sistema con otros existentes.
- Documentación (documentation). Persigue facilitar la mejora continua del software y su mantenimiento.
- Despliegue (deployment). Consiste en la instalación del software en un entorno de producción y puesta en marcha para explotación. En ocasiones implica una fase de entrenamiento de los usuarios del software.
- Mantenimiento (maintenance). Su propósito es la resolución de problemas, mejora y adaptación del software en explotación.

4.1.2 Metodologías de Desarrollo Software

Las metodologías son el modo en que las fases del proceso software se organizan e interaccionan para conseguir que dicho proceso sea reproducible y predecible para aumentar la productividad y la calidad del software.

Una metodología es una colección de:

- Procedimientos: indican cómo hacer cada tarea y en qué momento,
- Herramientas: ayudas para la realización de cada tarea, y
- Ayudas documentales.

Cada metodología es apropiada para un tipo de proyecto dependiendo de sus características técnicas, organizativas y del equipo de trabajo. En los entornos empresariales es obligado, a veces, el uso de una metodología concreta (p. ej. para participar en concursos públicos). El estándar internacional ISO/IEC 12270 describe el método para seleccionar, implementar y monitorear el ciclo de vida del software.

Mientras que unas intentan sistematizar y formalizar las tareas de diseño, otras aplican técnicas de gestión de proyectos para dicha tarea. Las metodologías de desarrollo se pueden agrupar dentro de varios enfoques según se señala a continuación.

- Metodología de Análisis y Diseño de Sistemas Estructurados (SSADM, Structured Sys-

tems Analysis and Design Methodology). Es uno de los paradigmas más antiguos. En esta metodología se emplea un modelo de desarrollo en cascada (waterfall). Las fases de desarrollo tienen lugar de modo secuencial. Una fase comienza cuando termina la anterior. Es un método clásico poco flexible y adaptable a cambios en los requisitos. Hace hincapié en la planificación derivada de una exhaustiva definición y análisis de los requisitos. Son metodologías que no lidian bien con la flexibilidad requerida en los proyectos de desarrollo software. Derivan de los procesos en ingeniería tradicionales y están enfocadas a la reducción del riesgo. Emplea tres técnicas clave:

  - Modelado lógico de datos (Logical Data Modelling),
  - Modelado de flujo de datos (Data Flow Modelling), y
  - Modelado de Entidades y Eventos (Entity EventModelling).

- Metodología de Diseño Orientado a Objetos (OOD, Object-Oriented Design). Está muy ligado a la OOP (Programación Orientada a Objetos) en que se persigue la reutilización. A diferencia del anterior, en este paradigma los datos y los procesos se combinan en una única entidad denominada objetos (o clases). Esta orientación pretende que los sistemas sean más modulares para mejorar la eficiencia, calidad del análisis y el diseño. Emplea extensivamente el Lenguaje Unificado de Modelado (UML) para especificar, visualizar, construir y documentar los artefactos de los sistemas software y también el modelo de negocio. UML proporciona una serie diagramas de básicos para modelar un sistema:

  - Diagrama de Clases (Class Diagram). Muestra los objetos del sistema y sus relaciones.
  - Diagrama de Caso de Uso (Use Case Diagram). Plasma la funcionalidad del sistema y quién interacciona con él.
  - Diagrama de secuencia (Sequence Diagram). Muestra los eventos que se producen en el sistema y como este reacciona ante ellos.
  - Modelo de Datos (Data Model).

- Desarrollo Rápido de Aplicaciones (RAD, Rapid Application Developmnent). Su filosofía es sacrificar calidad a cambio de poner en producción el sistema rápidamente con la funcionalidad esencial. Los procesos de especificación, diseño e implementación son simultáneos. No se realiza una especificación detallada y se reduce la documentación de diseño. El sistema se diseña en una serie de pasos, los usuarios evalúan cada etapa en la que proponen cambios y nuevas mejoras. Las interfaces de usuario se desarrollan habitualmente mediante sistemas interactivos de desarrollo. En vez de seguir un modelo de desarrollo en cascada sigue un modelo en espiral (Boehm). La clave de este modelo es el desarrollo continuo que ayuda a minimizar los riesgos. Los desarrolladores deben definir las carac-

terísticas de mayor prioridad. Este tipo de desarrollo se basa en la creación de prototipos y realimentación obtenida de los clientes para definir e implementar más características hasta alcanzar un sistema aceptable para despliegue.

- Metodologías Ágiles. "[…] envuelven un enfoque para la toma de decisiones en los proyectos de software, que se refiere a métodos de ingeniería del software basados en el desarrollo iterativo e incremental, donde los requisitos y soluciones evolucionan con el tiempo según la necesidad del proyecto. Así el trabajo es realizado mediante la colaboración de equipos auto-organizados y multidisciplinarios, inmersos en un proceso compartido de toma de decisiones a corto plazo. Cada iteración del ciclo de vida incluye: planificación, análisis de requisitos, diseño, codificación, pruebas y documentación. Teniendo gran importancia el concepto de"Finalizado" (Done), ya que el objetivo de cada iteración no es agregar toda la funcionalidad para justificar el lanzamiento del producto al mercado, sino incrementar el valor por medio de "software que funciona" (sin errores). Los métodos ágiles enfatizan las comunicaciones cara a cara en vez de la documentación. […]"}[1]

4.2 Proceso de Testing

Se debe indicar qué tipo de pruebas se han realizado, por ejemplo las siguientes:

- Pruebas modulares (pruebas unitarias). Su propósito es hacer pruebas sobre un módulo tan pronto como sea posible. Las pruebas unitarias que comprueban el correcto funcionamiento de una unidad de código. Dicha unidad elemental de código consistiría en cada función o procedimiento, en el caso de programación estructurada y cada clase, para la programación orientada a objetos. Las características de una prueba unitaria de calidad son: automatizable (sin intervención manual), completa, reutilizable, independiente y profesional.
- Pruebas de integración. Pruebas de varios módulos en conjunto para comprobar su interoperabilidad.
- Pruebas de caja negra.
- Beta testing.
- Pruebas de sistema y aceptación.
- …

[1] Fuente: Wikipedia

4.3 Marco Tecnológico

En esta sección se enumeran las tecnologías y herramientas utilizadas en la elaboración del TFG. A continuación, se citan algunos ejemplos.

4.3.1 Herramientas CASE (Computer Aided Software Engineering)

Las herramientas CASE están destinadas a facilitar una o varias de las tareas implicadas en el ciclo de vida del desarrollo de software. Se pueden dividir en la siguientes categorías:

- Modelado y análisis de negocio.
- Desarrollo. Facilitan las fases de diseño y construcción.
- Verificación y validación.
- Gestión de configuraciones.
- Métricas y medidas.
- Gestión de proyecto. Gestión de planes, asignación de tareas, planificación, etc.

4.3.2 IDE (Integrated Development Environment)

- Notepad++: https://notepad-plus-plus.org/
- Visual Studio Code: https://code.visualstudio.com/
- Atom: https://atom.io/
- GNU Emacs: https://www.gnu.org/s/emacs/
- NetBeans: https://netbeans.org/
- Eclipse: https://eclipse.org/
- QtCreator: https://www.qt.io/ide/
- jEdit: http://www.jedit.org/

4.3.3 Depuración

- GNU Debugger: https://www.gnu.org/s/gdb/

4.3.4 Repositorios y control de versiones

- Git: https://git-scm.com/
- Mercurial: [https://www.mercurial-scm.org/
- Github: https://github.com/
- Bitbucket: https://bitbucket.org/
- SourceTree: https://www.sourcetreeapp.com/

4.3.5 Documentación

- LaTeX: https://www.latex-project.org/
- Markdown: https://markdown.es/
- Doxygen: https://www.doxygen.nl/
- DocGen: http://mtmacdonald.github.io/docgen/docs/index.html
- Pandoc: http://pandoc.org/

4.3.6 Gestión y Planificación de Proyectos

- Trello: https://trello.com/
- Jira: https://es.atlassian.com/software/jira
- Asana: https://asana.com/
- Slack: https://slack.com/
- Basecamp: https://basecamp.com/
- Teamwork Proyects: https://www.teamwork.com/project-management-software
- Zoho Projects: https://www.zoho.com/projects/

Chapter 5

Resultados

En los que se describen cómo se ha aplicado el método de trabajo para el caso concreto del TFG, incluyendo aquellos elementos (modelos, diagramas, especificaciones, etc.) más importantes y relevantes que se quieran hacer notar.

5.1 Resultados del TFG

Este apartado debe explicar cómo el empleo de la metodología permite satisfacer tanto el objetivo principal como los específicos planteados en el TFG así como los requisitos exigidos (según exposición en capítulo Objetivos).

Chapter 6

Conclusiones

En este capítulo se debe incluir el 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. Incluirá obligatoriamente la justificación de las competencias de la tecnología específica cursada por el estudiante que se han adquirido durante el desarrollo del TFG

6.1 Revisión de los Objetivos

En esta sección se deberá revisar en qué grado se han completado los objetivos fijados al principio del proyecto. Se deberá también indicar las posibles desviaciones de los objetivos fijados, así como de la planificación, y tratar de justificar tales desviaciones.

6.2 Presupuesto

Si el TFG consiste en el desarrollo e implementación de un prototipo, la memoria debe incluir el coste del prototipo considerando tanto el hardware como los recursos humanos necesarios para su desarrollo.

Cuando se tiene en cuenta la puesta en marcha de un proyecto de ingeniería, la planificación y presupuesto que se realizan de modo previo a su ejecución son críticos para gestionar los recursos que permitan alcanzar los objetivos de calidad, temporales y económicos previstos para el proyecto. Es muy importante que todas las justificaciones aportadas se sustenten no solo en juicios de valor sino en evidencias tangibles como: historiales de actividad, repositorios de código y documentación, porciones de código, trazas de ejecución, capturas de pantalla, demos, etc.

6.3 Competencias Específicas de Intensificación Adquiridas y/o Reforzadas

Se deberán listar aquellas competencias de la intensificación que hayan sido adquiridas y/o reforzadas con el desarrollo de este TFG, incluyendo su justificación.

A. Uso de la Plantilla

A continuación se presenta la estructura y utilización de esta plantilla.

A.1 Configuración

La configuración de la plantilla se realiza a través del fichero *config.yaml*. Este es un fichero *yaml*, cuyos campos se describen en la Tabla A.1.

Table A.1: Valores del fichero *config.yaml*

clave	valor
Cite	Citas bibliográficas a incluir en la bibliografía no referenciadas en el texto
Cotutor	Nombre y apellidos del co-tutor académico
Csl	Fichero csl con el formato de las referencias
Department	Departamento del tutor académico
Language	Lenguaje de la plantilla [english|spanish]
Month	Mes de defensa del TFG
Name	Nombre del autor del TFG
Technology	Tecnología específica
Title	Título del TFG
Tutor	Nombre del tutor académico
Year	Año de defensa del TFG

Un ejemplo de configuración de este fichero se muestra en el Listado A.1.

Listing A.1: Ejemplo de fichero de configuración config.yaml

```
1 Cite: @Gutwin2010GoneBut, @Rekimoto1997PickDrop
2 Cotutor: John Deere
```

```
3  Csl: input/resources/csl/acm-sig-proceedings.csl
4  Department: Ciencias de la Computación
5  Language: spanish
6  Month: Agosto
7  Name: Johny Anston
8  Technology: Sistemas de Información
9  Title: Una Aplicación para Resolver Problemas Genéricos
10 Tutor: Adam Smith
11 Year: 2023
```

A.2 Estructura de Directorios

La plantilla tiene la estructura mostrada en la Figura A.1.

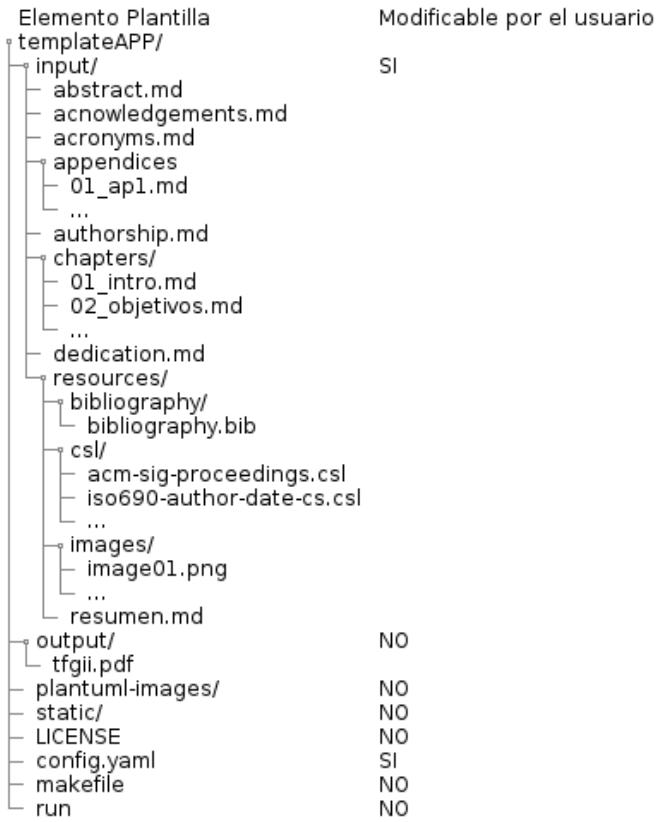

Figure A.1: Estructura de directorios de la plantilla del TFG

El documento generado por la plantilla es **output/tfgii.pdf**. Este documento se sobreescribe cada vez que se compila la plantilla. Es este documento, compilar se refiere a procesar los

elementos de la plantilla para generar el documento *pdf*.

Las carpetas y ficheros modificables por el usuario y que contienen la información propia del TFG son las siguientes (marcadas como SI en la figura A.1):

- input
- config.yaml

El resto de ficheros y directorios no deben de ser modificados por el usuario para el correcto funcionamiento de la plantilla.

A.2.1 Carpeta input

La carpeta *input* contiene los ficheros que corresponden con las partes del TFG.

En primer lugar están los ficheros correspondientes al **abstract**, **acknowledgements**, **acronyms**, **authorship** y **dedication**. En estos, a excepción del **acronyms**, sólo hay que introducir el texto correspondiente.

El fichero **acronyms** tiene un formato yaml. En el, hay que introducir los pares *acrónimo* y *valor* separados por ":" y un espacio. Un ejemplo sería el del Listado A.2.

Listing A.2: Ejemplo de definición de acrónimos

```
1 HTML: Lenguaje de marcas de hypertexto
2 HCI: Human Computer Interaction
```

También se encuentran en el directorio *input* los directorios **appendices** y **chapters**. Contienen, respectivamente, los apéndices y los capítulos del *TFG*, que corresponden con ficheros en *mark-down*. En estos ficheros hay que tener en cuenta lo siguiente:

- El fichero debe comenzar con un encabezado de primer nivel
- Los ficheros se ordenan en el documento según su orden alfabético
- Sólo se tienen en cuenta los ficheros que comienzan con dos dígitos numéricos, p.ej. 01
- Si un fichero no comienza con dos dígitos numéricos, es ignorado y no se incluye a la hora de generar el documento

A.2.2 Carpeta resources

El directorio *resources* contiene recursos de utilidad para la generación del documento, como imágenes o bibliografía.

La bibliografía se recoge en el fichero **bibliography/bibliography.bib**. Es un fichero en formato *bibtex* [**bibtex?**] que contiene los elementos bibliográficos. Por ejemplo, en el Listado A.3 se encuentra definido el elementos bibliográfico *bibtex*.

Listing A.3: Ejemplo de definición de elemento bibliográfico

```
1 @misc{bibtex,
2   title = {BibTeX Format Description},
3   howpublished = {\url{https://www.bibtex.org/Format/}},
4   note = {Accessed: 2023-09-28}
5 }
```

El directorio **csl** contiene ficheros con diferentes formatos de bibliografía. Finalmente, el directorio **images** contiene las imágenes a utilizar en el documento.

A.3 Elementos del Documento

Si bien la plantilla permite utilizar cualquier elemento de *Markdown* y *pandoc*, hay una serie de elementos que pueden ser de especial utilidad de cara a la realización de un TFG:

- Referencias bibliográficas
- Figuras
- Tablas
- Listados de código
- Notas al pie de página
- Acrónimos
- PlantUML
- Referencias dentro del Documento

A.3.1 Citas Bibliográficas

La plantilla utiliza la bibliografía que se recoge en el fichero correspondiente. Para incluir una cita a un elemento bibliográfico, hay que añadir en el documento su referencia precedida del símbolo @. Por ejemplo, para añadir la cita al elemento biliográfico *bibtex* se haría de la forma indicada en el Listado A.4.

Listing A.4: Ejemplo de cita de elemento bibliográfico

```
1 Es un fichero en formato *bibtex* @bibtex.
```

A.3.2 Figuras

Para incluir figuras lo hacemos añadiendo una imagen en la cual especificamos el elemento *label*, que sirve para referenciarla. Por ejemplo, el código del Listaqdo A.5 produce como resultado la Figura A.2.

Listing A.5: Ejemplo de definición de una figura

```
1 ![Logo de HTML5\label{anexo:ejemploFiguraResultado}](HTML5_sticker.png){width
    ↪ =50%}
```

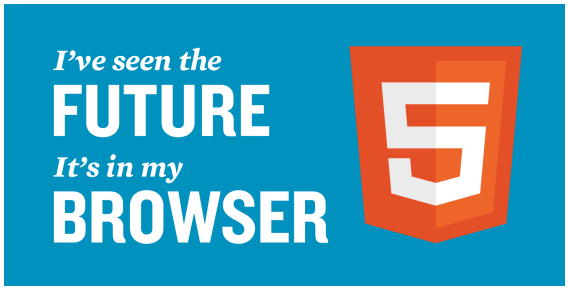

Figure A.2: Logo de HTML5

A.3.3 Tablas

Para definir una tabla se utiliza la sintaxis mostrada en el Listado de Código A.6, cuyo resultado se muestra en la Tabla A.2.

Listing A.6: Ejemplo de definición de una tabla

```
1 | Right | Left | Default | Center |
2 |------:|:-----|---------|:------:|
3 | 12 | 12 | 12 | 12 |
4 | 123 | 123 | 123 | 123 |
5 | 1 | 1 | 1 | 1 |
```

Table A.2: Tabla de ejemplo

Right	Left	Default	Center
12	12	12	12
123	123	123	123
1	1	1	1

A.3.4 Notas al Pie de Página

Para la definición de notas al pie de página se utiliza el código mostrado en el Listado A.7, cuyo resultado se muestra al pie de esta página[1].

Listing A.7: Ejemplo de definición de una nota a pie de página

```
cuyo resultado se muestra al pie de esta página[^anexo:ejemploNotaPie].

[^anexo:ejemploNotaPie]: Ejemplo de nota al pie de página.
```

A.3.5 Acrónimos

Para hacer referencia a un acrónimo, por ejemplo HTML, hay que utilizar el código mostrado en el Listado A.8. Previamente, el acrónimo debe de haber sido definido, tal y como se muestra en el Listado A.2.

Listing A.8: Ejemplo de definición de referencia a un acrónimo

```
Para hacer referencia a un acrónimo, por ejemplo [HTML](#HTML), hay ...
```

A.3.6 PlantUML

Esta plantilla permite la inserción de diagramas en *PlantUML* [**plantuml?**]. En el Listado A.9 se muestra un ejemplo básico de un diagrama de objetos en *PlantUML*, cuyo resultado se muestra en la Figura A.3.

Listing A.9: Ejemplo de código plantuml básico

```
@startuml
skinparam dpi 300
object Object01
object Object02
object Object03
object Object04
object Object05
object Object06
object Object07
object Object08
```

[1] Ejemplo de nota al pie de página.

```
11
12 Object01 <|-- Object02
13 Object03 *-- Object04
14 Object05 o-- "4" Object06
15 Object07 .. Object08 : some labels
16 @enduml
```

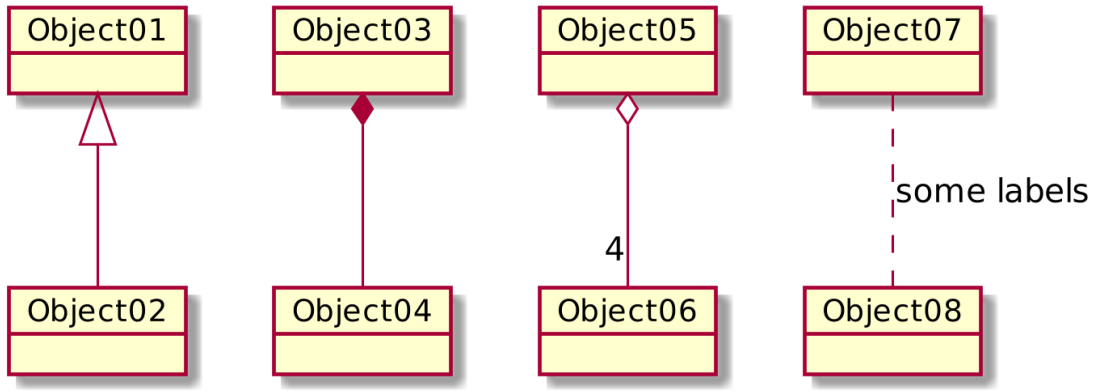

Figure A.3: Ejemplo de plantuml básico

También se puede dar formato, colores y formas, a los diagramas generados con PlantUML, tal y como se muestra en la Figura A.4.

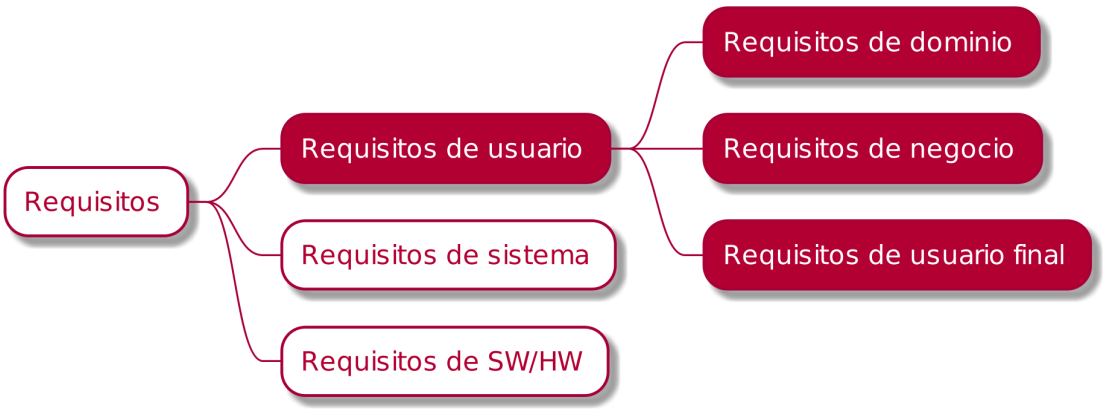

Figure A.4: Ejemplo de plantuml con formato

A.3.7 Referencias dentro del Documento

Para hacer una referencia a un capítulo dentro del documento se indica entre llaves el texto del enlace, seguido entre paréntesis del nombre del capítulo precedido del carácter *'#'*. El nombre

del capítulo se indica en minúsculas y se sustituyen los espacios por el carácter *'-'*. Por ejemplo, si se quiere referenciar al capítulo *"Estructura de Directorios"* se utilizará el siguiente código mostrado en la Figura A.10.

Listing A.10: Referencia dentro del documento

```
1 Referencia a la [estructura de directorios](#estructura-de-directorios)
```

A.4 Creación del Documento

La creación del documento se puede realizar utilizando un contenedor en *Docker* o de forma nativa en *Linux*[2]. A continuación, se explica cómo funciona cada una de estas aproximaciones, así como sus requisitos.

A.4.1 Ubuntu

Ejecutar 'make' en el directorio raíz de la plantilla.

Hay que tener instalado el siguiente software:

hacer script que compruebe dependencias

- texlive-latex-base
- texlive-lang-spanish
- texlive-xetex
- make
- pandoc-plantuml-filter

También crear el siguente directorio, copiar las fuentes y actualizarlas:

```
$ sudo mkdir -p /usr/share/fonts/truetype/msttcorefonts/
$ sudo cp util/fonts/* /usr/share/fonts/truetype/msttcorefonts/
$ sudo fc-cache -f -v
$ sudo texhash
```

Finalmente, hay que instalar pandoc, pero en su versión 2.19.2-1. Si se utiliza cualquier otra versión es muy probable que se tengan problemas, por ejemplo a la hora de generar imágenes con *plantuml*.

```
$ sudo dpkg -i util/sw/pandoc-2.19.2-1-amd64.deb
```

[2] Se ha probado que funciona correctamente en la distribución *Ubuntu 22.04.1*.

A.4.2 Docker

Ejecutar ‘make docker’ en el directorio raíz de la plantilla.

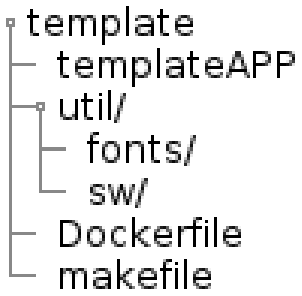

Figure A.5: Estructura de directorios raíz de la plantilla

En el caso de utilizar Docker, el único prerequisito es tenerlo instalado.

B. Anexo 1 - Metodologia de trabajo Scrum

We are going to use an Agile workflow based on scrum. Im going to be the developer and my turor is going to be the product owner.

This workflow consists of bi weekly or monthly sprints in wich we are going to be able to stablish the tasks for the sprint. This workflow wont have more meetings than the sprint meetings due to the fact that i'm the only person working on the project, therefore there is no need to do daily meetings fo stablish or split the work for the day. This is an adaptation of the clasic agile workflow in with we would have daily, sprint and project meetings in wich we would discuss the different aspects of the project.

For this workflow we are going to use the - *Explicar porque no hay dailyes (es adaptado para solo 1 desarrollador)* - ** Explicar el porque de GitHub - **** Añadir product backlog inicial

B.1 Sprint 1 (20/09/2023 - 04/10/2023) -

Sprint meeting - We stablish the bases of the project and the work methodology, also we stablish a final name and theme. **Sprint Tasks** - Define the basics of the methodology and decide on a interface and a form to aply it - Create the github project and create the product backlog

B.2 Sprint 2 (04/10/2023 - 22/11/2023)

Sprint meeting - Defining the project objectives and bulletpoints **Sprint Tasks** - Define and finalize the objectives of the project - Create the constitution project document. - Creation of the project based on Felix's template

B.3 Sprint 3 (22/11/2023 - 05/12/2023) - Introduction and state of the art.

Sprint meeting - We start discusions about the project information itself and how to make the project´s memory **Sprint Tasks** - Research about the state of the art regarding the different methods of image recognition **Sprint Retrospective** - We revised the progress and we saw that the Introduction was not started and the State of the art was starting to be researched but needed much more development.

B.4 Sprint 4 (05/12/2023 - 19/12/2023) - Deeper dive on Image recognition.

Sprint meeting - We agreed that the state of the art needed more develpement and we set ourselves on the path to research more about it - Also we have to stablish the motivations and objectives more clearly and develop a clear explanation of them in the project memory. **Sprint Tasks** - Research of the state of the art (4) - Further explanation of the introduction (2) - Rellenar las bases de la plantilla - Config.yaml (Datos, idioma) - Instrucciones en el el pdf del output (anexo1) - Enviar esto a sprint 2/1 - traducir esto a ingles - **Sprint Retrospective**

B.5 Sprint template (19/12/2023 -) - Titulo del sprint (resumiendo el objetivo)

- **Sprint meeting**
  - Revision del sprint anterior
  - Nuevas tareas
  - Establecer objetivo del sprint
- **Sprint Tasks**
  - Tareas al sprint backlog (Peso)
- **Sprint Retrospective**
  - Burndown Chart del sprint