



ESCUELA SUPERIOR DE INGENIERÍA

Titulación

Título del Proyecto Fin de Carrera

Curso 20xx-20yy

Nombre-autor Apellido1 Apellido2

Cádiz, xx de yy de zzzz





# ESCUELA SUPERIOR DE INGENIERÍA

## Titulación

Título del Proyecto Fin de Carrera

DEPARTAMENTO: Ingeniería Informática.

DIRECTOR DEL PROYECTO: Nombre-director Apellido1  
Apellido2.

AUTOR DEL PROYECTO: Nombre-autor Apellido1  
Apellido2.

Cádiz, xx de yy de zzzz

Fdo.: Nombre-autor Apellido1 Apellido2



# Índice general

<b>1. Introducción</b>	<b>5</b>
1.1. Motivación . . . . .	8
1.2. Objetivos . . . . .	8
1.3. Alcance . . . . .	8
1.4. Visión general . . . . .	8
1.5. Glosario . . . . .	8
1.5.1. Acrónimos . . . . .	8
1.5.2. Definiciones . . . . .	8
<b>2. Estado del arte</b>	<b>9</b>
<b>3. Desarrollo del calendario</b>	<b>11</b>
3.1. Fases . . . . .	11
3.2. Diagrama de Gantt . . . . .	11
<b>4. Descripción general del proyecto</b>	<b>13</b>
4.1. Perspectiva del producto . . . . .	13
4.1.1. Entorno del producto . . . . .	13
4.1.2. Interfaz de usuario . . . . .	13
4.2. Funciones . . . . .	13
4.3. Características del usuario . . . . .	13
4.4. Restricciones generales . . . . .	13
4.4.1. Control de versiones . . . . .	13
4.4.2. Lenguajes de programación y tecnologías . . . . .	13
4.4.3. Herramientas . . . . .	13
4.4.4. Sistemas operativos y hardware . . . . .	13
<b>5. Desarrollo del proyecto</b>	<b>15</b>
5.1. Modelo de ciclo de vida . . . . .	15
5.2. Requisitos . . . . .	15
5.2.1. Funcionales . . . . .	15
5.2.2. De información . . . . .	15
5.2.3. Reglas de negocio . . . . .	15
5.2.4. Interfaz . . . . .	15

## *Índice general*

5.2.5. No funcionales . . . . .	15
5.3. Análisis del sistema . . . . .	15
5.3.1. Casos de uso . . . . .	15
5.3.2. Modelo conceptual de datos del dominio . . . . .	15
5.3.3. Diagramas de secuencia . . . . .	15
5.4. Diseño del sistema . . . . .	15
5.5. Implementación . . . . .	15
5.6. Pruebas y validación . . . . .	15
<b>6. Resumen</b>	<b>17</b>
<b>7. Conclusiones y trabajo futuro</b>	<b>19</b>
7.1. Conclusiones . . . . .	19
7.2. Trabajo futuro . . . . .	19
<b>8. Agradecimientos</b>	<b>21</b>
<b>A. Manual de instalación</b>	<b>23</b>
<b>B. Manual de usuario</b>	<b>25</b>
<b>C. Manual del desarrollador</b>	<b>27</b>
<b>Bibliografía</b>	<b>29</b>

# Índice de figuras

1.1. Logotipo de la Universidad de Cádiz . . . . .	6
--	---

## *Índice de figuras*



# Índice de tablas

1.1. Diferencias entre SOA y EDA . . . . .	7
--	---



# 1. Introducción

Esta plantilla para el procesador de documentos LyX (<http://www.lyx.org>) está preparada para la redacción de una memoria de PFC (Proyecto Fin de Carrera) de las titulaciones de Ingeniería Técnica en Informática de Gestión (ITIG), Ingeniería Técnica en Informática de Sistemas e Ingeniería en Informática (II) de la UCA (Universidad de Cádiz). Téngase en cuenta que el contenido de dicha memoria podrá variar según la temática y el tutor del PFC.

Los acrónimos aparecen dentro de un recuadro porque los hemos insertado como código  $\LaTeX$ . Para insertar código  $\LaTeX$  en un fichero .lyx debe pulsarse Control+L. Como puede verse en el párrafo anterior, cada vez que utilicemos un acrónimo podemos especificarlo como código  $\LaTeX$ . De este modo,  $\LaTeX$  se encargará de definirlo automáticamente la primera vez que aparezca en el texto. Para ello, debe definirse su acrónimo en la sección correspondiente dentro de este capítulo.

## Formato y Estilo

Los tres iconos de formato que presenta el procesador LyX son «cambiar énfasis» (sería equivalente a cursiva), «cambiar versalitas» y «aplicar último» (último formato). Si se desean **otros formatos**: *seleccionar la palabra > botón derecho > Estilo del texto > Personalizado*.

## Etiquetas

Cada vez que se añada un capítulo, sección, tabla, figura, etc. debería asociársele una etiqueta para que pueda ser referenciado posteriormente.

Para insertar una etiqueta: *Insertar>Etiqueta*. Podría seguirse algún tipo de nomenclatura para los nombres de las etiquetas. Por ejemplo, las etiquetas de figuras comenzarán por «fig:», los capítulos por «cap:», etc.

Para referenciar una etiqueta en el texto: *Insertar>Referencia Cruzada*.

## 1. Introducción



Figura 1.1.: Logotipo de la Universidad de Cádiz

## Referencias

El autor de esta plantilla, Prof. Juan Boubeta Puig, es miembro del Grupo UCASE de Ingeniería del Software (TIC-025) [1]. En Boubeta-Puig et al. [2] se presenta una de sus publicaciones en congresos españoles. Nótese que las referencias deben ser guardadas en el fichero «referencias.bib» y deben ser incluidas en el documento .lyx con *Insertar>Cita*. Además, para evitar que la referencia sea separada de la palabra anterior, debe incluirse un espacio protegido ( ) pulsando Control+Espacio. Todas las referencias deben tener fecha, incluidas las referencias web.

## Figuras

Toda figura debe estar referenciada en el texto, es decir, no debería existir una figura a la que no se haga mención en el texto. Un ejemplo de esto puede verse en la Figura 1.1. Nótese que se ha incluido una referencia a la figura correspondiente con *Insertar>Referencia Cruzada* y se ha utilizado un espacio protegido entre la palabra «Figura» y su referencia.

Para más opciones de tamaño... de la figura, botón derecho sobre la figura insertada.

## Listados

Una opción es incluir directamente el código en el documento utilizando *verbatim*:

```
epService.getEPRuntime().sendEvent(evento);
```

Tabla 1.1.: Diferencias entre SOA y EDA

SOA	EDA
Basada en servicios demandados, normalmente síncronos.	Basada en eventos en tiempo real, normalmente asíncronos e impredecibles.
Clientes y servidores débilmente acoplados.	Clientes y servidores desacoplados.
Petición/respuesta uno-a-uno ( <i>pull</i> ).	Publicación/suscripción muchos-a-muchos ( <i>push</i> ).
Bidireccional, pero no garantiza la entrega.	Unidireccional, entrega de evento garantizada.
Ideal para procesamiento secuencial y aplicaciones compuestas en las que están disponibles todos los sistemas asociados y servicios.	Ideal para procesos paralelos, eventos (disparadores y uniones) y manejo de excepciones entre sistemas desconectados.
Usa metadatos de interfaces.	Usa metadatos de descriptores de eventos.
El cliente dirige flujos.	El receptor determina el flujo.
Cerrado a una entrada imprevista una vez que el flujo ha comenzado.	Puede reaccionar a nuevas entradas externas mientras el proceso esté activo.

L<sup>A</sup>T<sub>E</sub>X también permite incluir listados desde ficheros de texto, ajustando automáticamente las líneas de código a la configuración especificada. Puede verse un ejemplo en el Listado 1.1.

Listado 1.1: Envío de un evento a Esper

```
epService.getEPRuntime().sendEvent(evento);
```

## Tablas

A continuación, se presenta un ejemplo sobre cómo insertar y referenciar una tabla (véase la Tabla 1.1).

## Más información

Para más información consúltese la ayuda de L<sup>A</sup>T<sub>E</sub>X en el menú *Ayuda*.

## *1. Introducción*

### **1.1. Motivación**

### **1.2. Objetivos**

### **1.3. Alcance**

### **1.4. Visión general**

### **1.5. Glosario**

#### **1.5.1. Acrónimos**

**PFC** Proyecto Fin de Carrera

**UCA** Universidad de Cádiz

#### **1.5.2. Definiciones**

Para incluir definiciones en la memoria se puede utilizar el tipo «descripción». Nótese que si el concepto a definir está compuesto por más de una palabra, estas palabras deben ser separadas por un espacio protegido (Control+Espacio).

**Evento complejo** Un evento que es una abstracción de otros eventos.

## **2. Estado del arte**

## *2. Estado del arte*



## **3. Desarrollo del calendario**

### **3.1. Fases**

### **3.2. Diagrama de Gantt**

### *3. Desarrollo del calendario*

## **4. Descripción general del proyecto**

### **4.1. Perspectiva del producto**

#### **4.1.1. Entorno del producto**

#### **4.1.2. Interfaz de usuario**

### **4.2. Funciones**

### **4.3. Características del usuario**

### **4.4. Restricciones generales**

#### **4.4.1. Control de versiones**

#### **4.4.2. Lenguajes de programación y tecnologías**

#### **4.4.3. Herramientas**

#### **4.4.4. Sistemas operativos y hardware**

#### *4. Descripción general del proyecto*

## **5. Desarrollo del proyecto**

### **5.1. Modelo de ciclo de vida**

### **5.2. Requisitos**

#### **5.2.1. Funcionales**

#### **5.2.2. De información**

#### **5.2.3. Reglas de negocio**

#### **5.2.4. Interfaz**

#### **5.2.5. No funcionales**

### **5.3. Análisis del sistema**

#### **5.3.1. Casos de uso**

#### **5.3.2. Modelo conceptual de datos del dominio**

#### **5.3.3. Diagramas de secuencia**

### **5.4. Diseño del sistema**

### **5.5. Implementación**

### **5.6. Pruebas y validación**

## *5. Desarrollo del proyecto*

## **6. Resumen**

## 6. *Resumen*



## **7. Conclusiones y trabajo futuro**

### **7.1. Conclusiones**

### **7.2. Trabajo futuro**

## *7. Conclusiones y trabajo futuro*

## **8. Agradecimientos**

## 8. *Agradecimientos*

## **A. Manual de instalación**

## *A. Manual de instalación*

## **B. Manual de usuario**

## *B. Manual de usuario*



## **C. Manual del desarrollador**

### *C. Manual del desarrollador*

# Bibliografía

- [1] Grupo UCASE de Ingeniería del Software (Jan 2014), <https://ucase.uca.es/>
- [2] Boubeta Puig, J., Ortiz, G., Medina Bulo, I.: Procesamiento de Eventos Complejos en Entornos SOA: Caso de Estudio para la Detección Temprana de Epidemias. In: Actas de las VII Jornadas de Ciencia e Ingeniería de Servicios. pp. 63–76. Servizo de publicacións da Universidade da Coruña, A Coruña, Spain (Sep 2011)