



UNIVERSIDAD NACIONAL DE EDUCACIÓN A DISTANCIA

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA

Proyecto de Fin de Grado en <poner la titulación>

## **Implementación del Patrón de Diseño de Reusabilidad de GUIs.**

<NOMBRE DEL ESTUDIANTE>

Dirigido por: JOSÉ MANUEL CUADRA TRONCOSO

Curso: 2019-2020: <1ª o 2ª> Convocatoria





## **Implementación del Patrón de Diseño de Reusabilidad de GUIs.**

Proyecto de Fin de Grado de modalidad oferta general

Realizado por: <nombre del estudiante>

Dirigido por: José Manuel Cuadra Troncoso

Tribunal calificador

Presidente: D/D<sup>a</sup>.

Secretario: D/D<sup>a</sup>.

Vocal: D/D<sup>a</sup>.

Fecha de lectura y defensa:

Calificación:



# Agradecimientos

<Quisiera dedicar ...>



# Resumen

<Este proyecto ...>

# Abstract

<This project ...>



# Índice general

<b>1. Introducción general y objetivos</b>	<b>1</b>
1.1. Motivación y objetivos . . . . .	1
1.2. Estructura de la memoria . . . . .	4
<b>2. Conclusiones y trabajos futuros</b>	<b>5</b>
2.1. Conclusiones . . . . .	5
2.2. Trabajos futuros . . . . .	5
<b>A. &lt;Título Anexo A&gt;</b>	<b>9</b>
A.1. <Primera sección anexo> . . . . .	9
A.1.1. <Primera subsección anexo> . . . . .	9



# Nomenclatura

POO Programación Orientada a Objetos, página 1



# Índice de figuras

1.1. Ejemplo de figura con un pie largo . . . . .	2
---	---



# Índice de tablas

1.1. Ejemplo de tabla . . . . .	1
---------------------------------	---





# Capítulo 1

## Introducción general y objetivos

<Intro ... >

### 1.1. Motivación y objetivos

<Ejemplo biblio Eckel (2003); Gamma et al. (2005)>

<Ejemplo referencia a sección 1.2 y a subsección de otro capítulo, que en este caso es un anexo A.1.1. **Para que funcionen correctamente las referencias a otros capítulos al exportar a pdf, debe estar abierto el archivo maestro, abrir/editar el archivo a exportar desde él y exportar desde el maestro.**>

<Ejemplo referencia a figura 1.1>

<Ejemplo referencia a tabla 1.1>

<Ejemplo de nomenclatura (POO)>

<Ejemplo de listado de código, ver listado 1.1, lo mejor para utilizarlo es copiar el recuadro y cambiar el código contenido, así conservaremos las opciones que se han establecido, como: mostrar números de línea, quebrar líneas largas. etc. Para ver las opciones y poder modificarlas click derecho sobre el listado y elegir Configuración. Para cambiar el listado contenido copiar del IDE el trozo de código que se desee y pegarlo dentro del listado con Ctrl+May+v, en el menú de Lyx Editar->Pegado especial->Texto simple.>

1	2	3	4
aaaaaa	aaaaaa	aaaaaa	aaaaaa
aaaaaa	aaaaaa	aaaaaa	aaaaaa

Cuadro 1.1: Ejemplo de tabla



Figura 1.1: Ejemplo de figura con un pie largo para mostrar el uso del ítem del menú de Lyx Insertar->Título breve. La utilidad del título breve se aprecia en la lista de figuras, es ahí donde aparece, si no, aparecería todo el pie en la lista, también es conveniente para títulos de capítulos secciones y demás largo que ocupen más de una línea en la lista de índice correspondiente.(Para insertar salto de línea en pie o en enumeración Ctrl-Enter)

La figura o tabla hay que insertarla dentro de un flotante.

Siempre hay que Insertar->Etiqueta de la figura (fig:Ejemplo-de-figura), tabla etc, para referenciarla desde el texto.

Listado de código 1.1: Código de edit

```
1  public final boolean edit() {
2
3      ArrayList<String> campos = cargaCampos();
4
5      Scanner in = new Scanner( System.in );
6
7      /* La variable resp sera true si el usuario acepta la edicion
8      */
9
10     boolean resp = false;
11     int numero_total_opciones = campos.size() +
        opciones_no_campos;
12     System.out.println("\n" + toString()); // muestra el
        registro
13
14     editLoop: while (true) {
15         editMensaje();
16         System.out.println("Seleccione un número (1- " +
            numero_total_opciones + "): ");
17         int opcion;
18         if ( in.hasNextInt() ) {
19             opcion = in.nextInt();
20             in.nextLine();
21         } else {
22             System.out.println("\nOPCION NO VALIDA. POR FAVOR
                , INTRODUCIR UN NUMERO.");
23             in.nextLine();
24             continue;
25         }
26         campos = procesaOpcion(campos, opcion, in);
27         if(salir) {
28             if(aceptar)
29                 resp = true;
30                 salir = false;
31                 aceptar = false;
32                 break editLoop;
33         }
34
35     }
36     return resp;
37 }
```

## 1.2. Estructura de la memoria

La memoria de esta proyecto se estructura en los siguientes capítulos:

1. Introducción general y objetivos
2. <añadir los demás capítulos>
3. Conclusiones y trabajos futuros

<Comentar los capítulos>

## **Capítulo 2**

### **Conclusiones y trabajos futuros**

#### **2.1. Conclusiones**

<....>

#### **2.2. Trabajos futuros**

<....>



# Bibliografía

- Eckel, B. (2003). *Thinking in Patterns: Problem-Solving Techniques using Java*.  
<http://mindview.net/Books/TIPatterns/>.
- Gamma, E., Helm, R., Johnson, R., and Vlissides, J. (2005). *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley, Boston, MA.





# **Anexo A**

## **<Título Anexo A>**

**<...>**

### **A.1. <Primera sección anexo>**

#### **A.1.1. <Primera subsección anexo>**