MEMORIA ESCRITA DEL PROYECTO

CFGS Desarrollo de Aplicaciones Multiplataforma

**Clever Help Desk**

**Autor:** Victor Garcia Velasco

**Tutor:** Mario Gago

**Fecha de entrega:** dd/mm/aaaa\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Convocatoria:** Segundo Semestre – 2022

**GitHub del proyecto:** <https://github.com/victorgv/CleverHelpDesk>

**Documentos del proyecto:**Enlace a la carpeta del Drive\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Índice de contenidos

[1. Introducción 3](#_Toc95911256)

[1.1. Motivación 3](#_Toc95911257)

[1.2. Abstract 3](#_Toc95911258)

[1.3. Objetivos propuestos (generales y específicos) 3](#_Toc95911259)

[2. Metodología usada 4](#_Toc95911260)

[3. Tecnologías y herramientas utilizadas en el proyecto 5](#_Toc95911261)

[4. Estimación de recursos y planificación 6](#_Toc95911262)

[5. Análisis del proyecto 7](#_Toc95911263)

[6. Diseño del proyecto 8](#_Toc95911264)

[7. Despliegue y pruebas 9](#_Toc95911265)

[8. Conclusiones 10](#_Toc95911266)

[9. Vías futuras 11](#_Toc95911267)

[10. Bibliografía/Webgrafía 12](#_Toc95911268)

***En la normativa de proyectos vigente encontrarás una breve descripción de cada uno de estos apartados para saber qué información debes incluir en ellos***

# Introducción

## Motivación

## Abstract

## Objetivos propuestos (generales y específicos)

# Metodología usada

Debido al tiempo limitado y a que los requisitos están claros desde un principio he decidido utilizar el *Modelo en Cascada* para la realización de este proyecto de software. En este modelo de desarrollo clásico dividimos el ciclo de vida del proyecto en estas 5 fases:

Diagrama

Descripción generada automáticamente

Las principales motivaciones que me han hecho elegir este modelo son:

* Tener poco tiempo para realizar este proyecto (apenas 6 semanas)
* Los requerimientos ya están cerrados y claros
* Fácil de entender, planificar y realizar seguimiento

## Otros modelos evaluados

En la evaluación de qué metodología iba a usar me he interesado por la utilización de las metodologías agiles: *Scrum* y *Programación Extrema (XP)*, en ambos casos es necesario una retroalimentación por parte del cliente e ir realizando ciclos incrementales y refinando en cada uno los requisitos (tanto de ciclos anteriores como añadiendo nuevos requisitos). Evidentemente no voy a tener un cliente que me revise las entregas de cada ciclo, y la realización de varios ciclos aportando en cada entrega valor funcional al producto me habría llevado más tiempo, este ha sido mi principal motivo para descartar estas metodologías.

# Tecnologías y herramientas utilizadas en el proyecto

# Estimación de recursos y planificación

# Análisis del proyecto

# Diseño del proyecto

Diseño detallado:

* Diagramas de clases
* Diagramas de secuencia
* Especificación de métodos
* Modelos relacionales
* Diseño interfaces

Elementos a tener en cuenta:

* Como se hará la persistencia de los datos
* Seguridad
* Manejo de errores
* Iteracción y presentación

# Despliegue y pruebas

# Conclusiones

# Vías futuras

# Bibliografía/Webgrafía

* <https://docwiki.embarcadero.com/>
* <http://spring.io/>
* <https://openwebinars.net> Curso de Spring Core

# Anexo A. Manual estilos proyectos JAVA

dd

## IntelliJ IDEA, atajos de teclado

|  |  |  |
| --- | --- | --- |
| Acción | Atajo teclado | Comentario |
| Formatear código | CTRL+ALT+L | También botón derecho en navegador ficheros seleccionando *Reformat code*  Otra opción más avanzada CTRL+ALT+SHIFT+L |
| Acciones sobre código | ALT+ENTER | Para añadir imports o resolver errores |
| Generar getters, setters, constructors, etc | ALT+INSERT | También botón derecho sobre el código opción *Generate* |
| Duplicar línea | CTRL+D |  |

# Anexo B. Manual estilos proyectos DELPHI

Ss

