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\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Contenido

[1. Introducción 3](#_Toc99044234)

[1.1. Motivación 3](#_Toc99044235)

[1.2. Abstract 3](#_Toc99044236)

[1.3. Objetivos propuestos (generales y específicos) 3](#_Toc99044237)

[2. Metodología usada 4](#_Toc99044238)

[Otros modelos evaluados 4](#_Toc99044239)

[3. Tecnologías y herramientas utilizadas en el proyecto 6](#_Toc99044240)

[4. Estimación de recursos y planificación 7](#_Toc99044241)

[5. Análisis del proyecto 8](#_Toc99044242)

[Requisitos funcionales 8](#_Toc99044243)

[Requisitos no funcionales 9](#_Toc99044244)

[Diagrama Entidad-Relación 9](#_Toc99044245)

[Casos de Uso 11](#_Toc99044246)

[CU\_01 LOGIN 11](#_Toc99044247)

[CU\_02 Consultar TICKETs 12](#_Toc99044248)

[CU\_03 Creación TICKET 13](#_Toc99044249)

[CU\_04 Gestión del TICKET 14](#_Toc99044250)

[CU\_05 Adjuntar Comentarios al TICKET 15](#_Toc99044251)

[CU\_06 Gestión de Usuarios 16](#_Toc99044252)

[6. Diseño del proyecto 18](#_Toc99044253)

[7. Despliegue y pruebas 19](#_Toc99044254)

[8. Conclusiones 20](#_Toc99044255)

[9. Vías futuras 21](#_Toc99044256)

[Objetivos que se plantearon en la petición inicial y no se alcanzaron: 21](#_Toc99044257)

[Futuras mejoras que se incorporarán en el sistema: 21](#_Toc99044258)

[10. Bibliografía/Webgrafía 23](#_Toc99044259)

[Anexo A. Glosario 24](#_Toc99044260)

[Anexo N. Manual estilos proyectos JAVA 24](#_Toc99044261)

[IntelliJ IDEA, atajos de teclado 24](#_Toc99044262)

[Anexo N. Manual estilos proyectos DELPHI 24](#_Toc99044263)

***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

Memoria del proyecto fin de ciclo Desarrollo Aplicaciones Multiplataforma de Victor Garcia Velasco. La aplicación se llama *Clever Help Desk* y se trataría de un sistema de seguimiento de incidencias/asistencias/consultas orientado a un equipo de desarrollo de software, aunque se podría extender a otras áreas IT.

## Motivación

[TO-DO]

## Abstract

[TO-DO]

## Objetivos propuestos (generales y específicos)

[TO-DO]

# 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

Herramientas desarrollo:

* IntelliJ IDEA 2021.2.3 (Community Edition), Java+spring boot
* Delphi 10.4 Community Edition, Firemonkey para desarrollo multiplataforma (Windows y Android)
* Github
* Jira

Base de datos:

* Oracle 11XE

Herramientas apoyo:

* Microsoft Word
* Microsoft Excel
* Microsoft Onenote
* <https://www.diagrams.net/>

Equipo hardware:

* Lenovo T490 con software base Windows 11
* Sansung Galaxy Note 8

# Estimación de recursos y planificación

Texto

Descripción generada automáticamente

# Análisis del proyecto

En el proceso de análisis especificamos las características del sistema, se detalla el interface que se desarrollará y se indican las restricciones de este.

Empezaremos plasmando los requisitos funcionales y no funcionales, para posteriormente plasmar los diferentes diagramas que nos facilitarán el entendimiento conceptual, así como ver las relaciones e iteraciones entre los diferentes componentes que conforman el sistema.

## Requisitos funcionales

1. Se deberá hacer LOGIN para poder utilizar la aplicación CLIENTE.
2. Hay tres niveles acceso ADMINISTRADOR, AGENTE, USUARIO, se deberá tener el acceso coherente al nivel asignado.
3. Se podrán consultar los tickets según el nivel acceso:
   1. ADMINISTRADOR y AGENTE, pueden ver todos.
   2. USUARIO, solo los tickets que haya abierto.
4. Creación tickets.
5. Asignar un agente a un ticket.
6. Cambiar estado/proyecto/tipo ticket.
7. Poder adjuntar comentarios de texto al ticket.
8. Guardar un pequeño histórico cronológico del ticket, cuando se creo/cambios estados/etc.
9. Generación de email con el número de ticket cuando se abra y se cierre.
10. Posibilidad de crear un ticket enviando un email a una dirección genérica de la aplicación.
    1. Solo creará el ticket si la dirección de email del remitente está registrada como usuario
    2. En caso de no reconocer el email como usuario devolverá un email indicando que no se ha podido crear el ticket indicando que se debe poner en contacto con el administrador
11. Poder dar de alta/baja usuarios (mantenimiento de usuarios)
12. Poder crear proyectos para agrupar los tickets por temática (mantenimiento de proyectos)

## Requisitos no funcionales

1. Poder utilizar la aplicación en S.O. Windows y Android mediante aplicación nativa, realizada con DELPHI/FIREMONKEY
2. Seguridad básica, guardado passwords encriptados y utilizar JWT para autentificación en el servidor REST
3. Creación servidor REST/JSON en JAVA utilizando el framework SPRING BOOT
4. Como BBDD se utilizará ORACLE, pero el lado servidor utilizará el estándar JPA con su implementación HIBERNATE de forma que permita la migración a cualquier otro gestor de BBDD relacional que lo soporte

## Diagrama Entidad-Relación

A continuación se plasma el modelo entidad-relación para facilitar la representación de datos, entidades y relaciones que formarán la base de datos.

Diagrama

Descripción generada automáticamente

Entidades:

* ***User***, usuarios que tendrán acceso a la aplicación
* ***Role***, perfiles de usuario, inicialmente serán: USUARIO, AGENTE y ADMINISTRADOR, aunque permitiría añadir nuevos roles
* ***Ticket***, son las peticiones/incidencias/etc que han abierto los usuarios
* ***Comment***, posibles comentarios que pueda ir introduciendo el usuario que abre el ticket o el agente de lo atiende
* ***Ticket History***, es un histórico cronológico que guarda los eventos más relevantes del ticket. Por ejemplo almacenará cuando se hizo un cambio de estado y quien lo hizo
* ***Project***, diferentes proyectos o áreas en las que luego podremos agrupar los diferentes tickets
* ***Master type***, diferentes tipos de ticket, por ejemplo: consulta, incidencia, extracción datos BD, etc.
* ***Master status***, los posibles estados que puede tener un ticket.

## Casos de Uso

A través de los *Casos de Uso* describiremos el comportamiento que tiene el sistema desde el punto de vista del usuario o los procesos que van a interactuar con él. A continuación detallamos cada caso de uso detectado junto con su diagrama para mejor comprensión.

Para facilitar la comprensión se muestra el diagrama Inicial que después se detalla para cada caso de uso junto:

Diagrama

Descripción generada automáticamente

### CU\_01 LOGIN

|  |  |  |  |
| --- | --- | --- | --- |
| **Identificador de Caso Uso** | | CU\_01 | |
| **Nombre** | | LOGIN | |
| **Descripción** | | Proceso de validación y acceso en el sistema | |
| **Actores** | | Usuario, Agente, Administrador | |
| **Secuencia normal** | | | |
| Actor | | | Software |
| 1. Introduce su email, password y pulsa “login” | | |  |
|  | | | 2. Se valida el email/password |
|  | | | 3. Se accede a la aplicación |
| Excepciones | | | Software |
| Usuario cancela el login | | | Se termina la ejecución |
| Usuario no registrado | | | Muestra mensaje |
| Usuario/password incorrecto | | | Muestra mensaje |
| Usuario/password en blanco | | | Muestra mensaje y posiciona en el campo vacío |
| **CU relacionados** |  | | |
| Precondición | 1. Debe estar dado de alta en la aplicación.  2. Debe rellenar los campos email y password | | |
| Post condición | El usuario puede acceder al sistema | | |

Diagrama

Descripción generada automáticamente

### CU\_02 Consultar TICKETs

|  |  |  |  |
| --- | --- | --- | --- |
| **Identificador de Caso Uso** | | CU\_02 | |
| **Nombre** | | Consultar TICKETs | |
| **Descripción** | | Formulario principal donde se podrán buscar los TICKETs según el perfil y abrir para ver el contenido | |
| **Actores** | | Usuario, Agente, Administrador | |
| **Secuencia normal** | | | |
| Actor | | | Software |
| 1. Lanzar consulta por diferentes criterios (abiertos por mí, con un texto determinado, abiertos en una fecha X, etc) | | |  |
|  | | | 2. Volcará en un GRID los TICKETs que cumplen con los criterios de consulta |
| 3. El usuario abre un TICKET para ver su contenido | | |  |
|  | | | 4. Se mostrará en una ventana o pestaña nueva para poder visualizar toda la información |
| Excepciones | | | Software |
| La consulta no devuelve información | | | El GRID se muestra sin ningún registro |
| Se controla el rengo de fechas para no permitir consultas en rangos de fechas mayores a 366 días | | | El sistema muestra una ventana de aviso “el rango máximo de consulta es 1 año” y cancela la consulta |
| **CU relacionados** | **CU\_01** | | |
| Precondición | 1. debe estar registrado el email  2. estar logeado | | |
| Post condición |  | | |

Diagrama

Descripción generada automáticamente

### CU\_03 Creación TICKET

|  |  |  |  |
| --- | --- | --- | --- |
| **Identificador de Caso Uso** | | CU\_03 | |
| **Nombre** | | Creación de un TICKET | |
| **Descripción** | | Proceso de creación ticket | |
| **Actores** | | Usuario, Agente, Administrador y Email | |
| **Secuencia normal** | | | |
| Actor | | | Software |
| 1a. EMAIL: se recibe un email | | |  |
| 1b. USUARIO: crea un ticket desde el formulario rellenando el título y la descripción de la petición | | |  |
|  | | | 2. Se crea el registro, con el título y descripción |
|  | | | 3. Se emite un email con el número de ticket como confirmación |
| Excepciones | | | Software |
| EMAIL: dirección email no existe en el sistema, no está registrado como usuario | | | Se emite un email de respuesta indicando que no está registrando y debe ponerse en contacto con el administrador |
| USUARIO: cancela la creación del TICKET | | | No se graba el registro y se vuelve a la pantalla de consulta |
| **CU relacionados** | **CU\_01, CU\_02** | | |
| Precondición | 1. debe estar registrado el email  2. Si es usuario, agente o administrador debe estar logeado | | |
| Post condición | Usuario vuelve a la pantalla de consulta tickets | | |

Diagrama

Descripción generada automáticamente

### CU\_04 Gestión del TICKET

|  |  |  |  |
| --- | --- | --- | --- |
| **Identificador de Caso Uso** | | CU\_04 | |
| **Nombre** | | Gestión del TICKET | |
| **Descripción** | | Cuando un AGENTE o ADMINSTRADOR trabaja sobre el ticket, debe configurar a que proyecto pertenece y de que tipo es. Puede cambiar el estado. | |
| **Actores** | | Agente y Administrador | |
| **Secuencia normal** | | | |
| Actor | | | Software |
| 1. Acceso al formulario gestionar ticket. | | |  |
|  | | | 2. Se abre el formulario |
| 3. Se modifican los campos (agente que atiende el ticket, proyecto, tipo ticket, estado, Cerrar ticket, etc) y se pulsa “grabar” | | |  |
|  | | | 4. Se confirman los cambios y se vuelve a la pantalla de consulta. En el histórico se registrará como log los cambios (por ejemplo cada ver que se cambia el estados, quien y cuando provoco la modificación) |
|  | | | 5. En el caso de cierre de un ticket se emitirá un email |
| Excepciones | | | Software |
| Intentar cerrar el ticket o pasar al estado “FINALIZADO” sin haber rellenado los campos proyecto o tipo de ticket | | | Se mostrará un aviso indicando que no se puede finalizar sin completar los campos indicados |
| **CU relacionados** | **CU\_01, CU\_02** | | |
| Precondición | 1. el agente o administrador debe estar logeado  2. el ticket debe estar creado previamente | | |
| Post condición | Usuario vuelve a la pantalla de consulta tickets | | |

Diagrama

Descripción generada automáticamente

### CU\_05 Adjuntar Comentarios al TICKET

|  |  |  |  |
| --- | --- | --- | --- |
| **Identificador de Caso Uso** | | CU\_05 | |
| **Nombre** | | Adjuntar comentarios al TICKET | |
| **Descripción** | | Se podrán añadir comentarios al TICKET y se podrán consultar cronológicamente | |
| **Actores** | | Usuario, Agente y Administrador | |
| **Secuencia normal** | | | |
| Actor | | | Software |
| 1. Acceso al formulario comentarios. | | |  |
|  | | | 2. Se abre el formulario |
| 3. Se introduce un comentario (debajo el último que ya hubiese) y se pulsa “grabar” | | |  |
|  | | | 4. Se confirman la inserción |
| Excepciones | | | Software |
| Intentar introducir un comentario en blanco | | | Se mostrará un aviso indicando que no se puede insertar un comentario vacío |
| **CU relacionados** | **CU\_01, CU\_02** | | |
| Precondición | 1. el agente o administrador debe estar logeado  2. el ticket debe estar creado previamente | | |
| Post condición | Usuario vuelve a la pantalla de consulta tickets | | |

Diagrama

Descripción generada automáticamente

### CU\_06 Gestión de Usuarios

|  |  |  |  |
| --- | --- | --- | --- |
| **Identificador de Caso Uso** | | CU\_06 | |
| **Nombre** | | Gestión Usuarios | |
| **Descripción** | | Pequeño mantenimiento donde se podrán dar de alta y baja los usuarios en el sistema, además de otras tareas básicas como resetear el password | |
| **Actores** | | Administrador | |
| **Secuencia normal** | | | |
| Actor | | | Software |
| 1. Acceso al formulario | | |  |
|  | | | 2. Se muestran los usuarios del sistema junto su perfil |
| 3. El administrador crea un usuario, rellenando sus datos | | |  |
|  | | | 4. Se confirma la inserción y el usuario puede entrar en el sistema con el password que le ha asignado el administrador |
| 5. Se da de baja un usuario | | |  |
|  | | | 6. Se confirma y a partir de ese momento el usuario no podrá acceder al sistema |
| Excepciones | | | Software |
| Se intenta crear un usuario que ya existe | | | Se mostrará un aviso indicando que no se puede duplicar el usuario |
| Se dejan en blanco campos obligatorios | | | Se avisa y se coloca sobre el campo faltante |
| **CU relacionados** | **CU\_01** | | |
| Precondición | 1. el administrador debe estar logeado  2. solo puede entrar si se tiene perfil ADMINISTRADOR | | |
| Post condición | El usuario creado puede entrar en el sistema | | |

Diagrama

Descripción generada automáticamente

CU\_07 Creación Proyectos

|  |  |  |  |
| --- | --- | --- | --- |
| **Identificador de Caso Uso** | | CU\_07 | |
| **Nombre** | | Creación Proyectos | |
| **Descripción** | | Pequeño mantenimiento donde se podrán dar de alta Proyectos. Estos proyectos permitirán agrupar los TICKETs en sus áreas funcionales | |
| **Actores** | | Administrador | |
| **Secuencia normal** | | | |
| Actor | | | Software |
| 1. Acceso al formulario | | |  |
|  | | | 2. Se muestran los proyectos que hay creados |
| 3. El administrador crea un proyecto, rellenando sus datos | | |  |
|  | | | 4. Se confirma la inserción y a partir de ese momento se podrá utilizar para asignar en los TICKETs |
| 5. De modifica o da de baja | | |  |
|  | | | 6. Se confirma los cambios |
| Excepciones | | | Software |
| Se intenta crear un usuario que ya existe | | | Se mostrará un aviso indicando que no se puede duplicar el usuario |
| Se dejan en blanco campos obligatorios | | | Se avisa y se coloca sobre el campo faltante |
| **CU relacionados** | **CU\_01** | | |
| Precondición | 1. el administrador debe estar logeado  2. solo puede entrar si se tiene perfil ADMINISTRADOR | | |
| Post condición | El proyecto creado ya está disponible para utilizar | | |

Diagrama

Descripción generada automáticamente

# 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

[TO-DO]

Texto

Descripción generada automáticamente

## Objetivos que se plantearon en la petición inicial y no se alcanzaron:

## Futuras mejoras que se incorporarán en el sistema:

A continuación se exploran las próximas funcionalidades que añadiríamos en futuras versiones, lo primero será arrancar en un entorno productivo, con esta primera versión y las sugerencias de los usuarios seguro que llegamos a un sistema muy completo.

* ***Gestión de la retención de la información***, es muy importante tratar este tema desde las fases iniciales del proyecto para evitar que en el tiempo crezca descontroladamente el tamaño de las bases de datos. En este caso tendríamos un proceso nocturno que eliminaría registros “obsoletos” y se podría configurar por proyecto y/o tipo tarea, por ejemplo podríamos tener una configuración así:
  + Retención tablas LOGs: 1 año
  + Retención tablas TICKETs y sus relacionadas: 5 años (a partir de la fecha de cierre)
  + Retención ficheros adjuntos: 2 años (a partir de la fecha de cierre)
  + Etc.
* ***Posibilidad de adjuntar ficheros***, ya sea imágenes, .PDF, .DOC u cualquier otro fichero interesante para el tratamiento del TICKET. Se establecerá un límite de tamaño fichero máximo 25Mb.
* ***Integrar con el LDAP corporativo***, aportando las siguientes ventajas:
  + Simplifica al usuario la utilización de los sistemas ya que con un usuario y password acceden a todos los sistemas (que tienen permiso) de la empresa
  + Unifica la política de seguridad, por ejemplo, password de mínimo 10 letras, con mínimo 1 letra maýuscula, mínimo 1 letra numérica, etc. Otro punto importante es la obligación de cambiar el password pasados los 30 días.
  + Mejora la seguridad, por ejemplo, el usuario se bloquea después de N intentos fallidos, se caduca si pasan más de 30 días sin utilizar el sistema, etc.
  + Cuando un usuario causa baja en la empresa automáticamente se dá de baja en LDAP impidiendo que pueda entrar en las aplicaciones.

# Bibliografía/Webgrafía

* <https://www.diagrams.net/> aplicación para modelado de diagramas
* <https://openwebinars.net> - curso de Spring Core
* <https://www.youtube.com/watch?v=iFcDoP6jEeE> Casos de uso y diagramas de casos de uso | | UPV
* <https://www.youtube.com/watch?v=orvAkFFWo5o> Diagrama de casos de uso | | UPV
* <https://www.youtube.com/watch?v=JioEGJIlg88> Diagrama de clases | | UPV
* <https://docwiki.embarcadero.com/RADStudio/Alexandria/en/Multi-Device_Applications_Index> Ayuda DELPHI/FIREMONKEY para desarrollo aplicaciones multiplataforma
* <http://spring.io/> Documentación Spring

# Anexo A. Glosario

A continuación se incluye la terminología y siglas utilizadas en la elaboración de esta memoria.

# Anexo N. 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 N. Manual estilos proyectos DELPHI

Ss

