

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[TO-DO]

Convocatoria: Curso 2021-2021 - Segundo Semestre

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

Documentos del proyecto: Enlace a la carpeta del Drive[TO-DO]



Contenido

1. INTRODUCCIÓN	3
1.1 MOTIVACIÓN	3
1.2 ABSTRACT	3
1.3 OBJETIVOS PROPUESTOS (GENERALES Y ESPECÍFICOS).....	3
2 METODOLOGÍA USADA	5
OTROS MODELOS EVALUADOS.....	5
3 TECNOLOGÍAS Y HERRAMIENTAS UTILIZADAS EN EL PROYECTO.....	7
4 ESTIMACIÓN DE RECURSOS Y PLANIFICACIÓN	9
5 ANÁLISIS DEL PROYECTO	12
REQUISITOS FUNCIONALES.....	12
REQUISITOS NO FUNCIONALES.....	13
DIAGRAMA ENTIDAD-RELACIÓN	13
CASOS DE USO	15
CU_01 LOGIN.....	16
CU_02 Consultar TICKETs	17
CU_03 Creación TICKET	18
CU_04 Gestión del TICKET	19
CU_05 Adjuntar Comentarios al TICKET	20
CU_06 Gestión de Usuarios	21
CU_07 Creación Proyectos.....	22
DIAGRAMA CLASES (CONCEPTUAL).....	23
6 DISEÑO DEL PROYECTO	25
PROTOTIPOS:.....	25
Formulario de Login	25
Formulario principal y Consulta de Tickets	26
Formulario consulta/gestión de un ticket.....	29
7 DESPLIEGUE Y PRUEBAS	30
8 CONCLUSIONES	31
9 VÍAS FUTURAS.....	32
OBJETIVOS QUE SE PLANTEARON EN LA PETICIÓN INICIAL Y NO SE ALCANZARON:.....	32
FUTURAS MEJORAS QUE SE INCORPORARÁN EN EL SISTEMA:	32
10 BIBLIOGRAFÍA/WEBGRAFÍA	34

ANEXO A. GLOSARIO.....	35
ANEXO N. MANUAL ESTILOS PROYECTOS JAVA.....	35
INTELLIJ IDEA, ATAJO DE TECLADO	35
ANEXO N. MANUAL ESTILOS PROYECTOS DELPHI	35

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

1. 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/peticiones+mejoras orientado a un equipo de desarrollo de software, aunque se podría extender a otras áreas IT. Esta aplicación se utilizará para tener un registro de incidencias/asistencias/consultas/peticiones+mejoras así como para su gestión y saber en que estado está cada item así como saber quien del equipo la ha atendido.

1.1 Motivación

[TO-DO]

1.2 Abstract

[TO-DO]

1.3 Objetivos propuestos (generales y específicos)

Mi objetivo principal es desarrollar una aplicación funcional multi-plataforma para poder abrir “tickets” (incidencias/asistencias/peticiones/etc) por parte de los usuarios y así los agentes poder atenderlas, modificar el status, hacer comentarios, etc. La gracia es que utilizo una tecnología que me permite, a partir de un mismo código fuente, generar una aplicación nativa WINDOWS y otro aplicación nativa ANDROID (también podría hacer la aplicación nativa para

MacOS e IOS pero por falta de recursos y tiempo no lo podré hacer). Esto lo conseguimos utilizando las librerías de FIREMONKEY en el IDE DELPHI.

Otro objetivo es que la aplicación utilizar el patrón de arquitectura MVC para ello me apoyaré en SPRING BOOT.

Más específicamente el sistema tendrá las siguientes funcionalidades:

- Posibilidad de hacer login (requerido para poder entrar en la aplicación)
- Generación de email con el número de ticket cuando se abra y se cierre
- Posibilidad de crear un ticket enviando un email a una dirección específica
- Formulario de navegación/consulta por los “tickets” abiertos (ver los que tengo “yo” asignados, ver los de otro agente, ver los tickets cerrados, etc)
- Formulario de creación/modificación “ticket” con los campos
 - nombre
 - descripción
 - tipo (consulta, asistencia, consulta, permisos, etc)
 - área funcional/proyecto -estado (registrado, trabajando, parado, terminado) - fecha apertura -fecha último cambio
 - fecha cierre
 - agente asignado
 - usuario que reporta el ticket
 - registros comentarios (pueden haber N y junto el comentario se guardará la fecha)
- Mantenimiento Usuarios, con tres perfiles: Administrador, agente y usuario
- Mantenimiento Áreas funcionales/Proyectos

2 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:



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



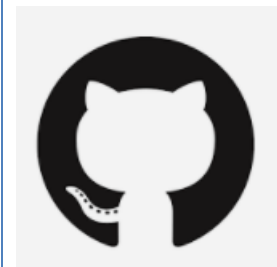
- Tener poco tiempo para realizar este proyecto (apenas 6 semanas)
- Los requerimientos ya están cerrados y claros
- Fácil de entender, planificar y realizar seguimiento

Otros modelos evaluados

En la evaluación de qué metodología iba a usar me he interesado por la utilización de las metodologías ágiles: *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.

3 Tecnologías y herramientas utilizadas en el proyecto

Herramientas desarrollo:

	Herramienta	Descripción y uso en este proyecto
	IntelliJ IDEA 2021.2.3 (Community Edition)	<p>IDE para desarrollo de la parte servidor para la creación de un servidor <i>REST/JSON</i> al que se podrá conectar la aplicación cliente.</p> <p>El lenguaje de programación utilizado es <i>JAVA</i> junto con las librerías <i>SPRING BOOT</i>.</p>
	Delphi 10.4 Community Edition	<p>IDE desarrollo para la parte cliente, apoyado con su librería para desarrollo multiplataforma <i>FIREMONKEY</i>. con un paradigma claro <u>un único código fuente para varias plataformas</u>. En este proyecto se creará el programa cliente para Windows y Android. También permitiría crearlo para MacOS e IOS pero es necesario tener un "Mac" para hacer la compilación y no dispongo de él.</p>
	Github	<p>El repositorio del proyecto se alojará en Github, que a su vez utiliza Git como sistema de versionado y control de versiones.</p>

Base de datos:

- Oracle 11XE

Herramientas apoyo:

- Microsoft Word

- Microsoft Excel
- Microsoft Onenote
- teamgantt.com – aparte de utilizarlo para representar el diagrama Gantt también lo utilizo para controlar el estado de las tareas así como voy actualizando en el tiempo para poder comparar la “foto” inicial del proyecto (planificación) con la “foto” al finalizar el mismo.
- <https://www.diagrams.net/>

Equipo hardware:

- Lenovo T490 con software base Windows 11
- Samsung Galaxy Note 8
- NAS QNAP 473

4 Estimación de recursos y planificación

Para la estimación y ver el coste en horas que podría suponer el proyecto he definido las diferentes tareas en base los requisitos funcionales, análisis (punto 5 de esta memoria) y la metodología elegida. Para ser estrictamente correcto faltaría una última fase de mantenimiento, donde se podría incluir el arranque en producción, apoyo tras arranque y un posible contrato de mantenimiento temporal, no he incluido esta fase porque no se realizará.

Por otro lado también he planificado una fase adicional “otros” para las tareas de preparación de la memoria y video de presentación del proyecto.

El detalle de tareas queda así:

Fase	ID	Actividad	Estimación (horas)
Análisis	A01	Definición Requisitos y Estimación Proyecto	12
Análisis	A02	Módelado diagramas y detallado (modelo ER, diagrama clases, casos uso)	15
Diseño	D01	Especificación de funcionalidades, detalle de clases, revisión modelo BD, etc	12
Diseño	D02	Prototipos interfaces gráficas	5
Codificación	C01	BackEnd. Creación del core de la aplicación spring boot	17
Codificación	C02	BackEnd. Clases para el modelo de datos	8
Codificación	C03	BackEnd. Controller	7
Codificación	C04	BackEnd. Servicio y lógica negocio	15
Codificación	C05	BackEnd. Bean Envío emails	5
Codificación	C06	BackEnd. Bean Recepción emails y procesado	13
Codificación	C07	Cliente. Core a la aplicación multiplataforma (windows / android)	10
Codificación	C08	Cliente. Login	5
Codificación	C09	Cliente. Formulario consulta "tickets"	14
Codificación	C10	Cliente. Formulario detalle/creación/modificación ticket. Funcionalidad eliminar. Gestión del ticket. Adjuntar/consultar comentarios. Ver histórico.	29
Codificación	C11	Cliente. Gestión Usuarios	5
Codificación	C12	Cliente. Gestión Proyectos	5
Pruebas	P01	Realización pruebas de verificación y validación	15
Otros	O01	Memoria (primera entrega 28 marzo) y revisión planificación	15
Otros	O02	Memoria revisión y finalización	11
Otros	O03	Creación diapositivas video	8
Otros	O04	Creación video presentación proyecto	8
Total (horas)			234

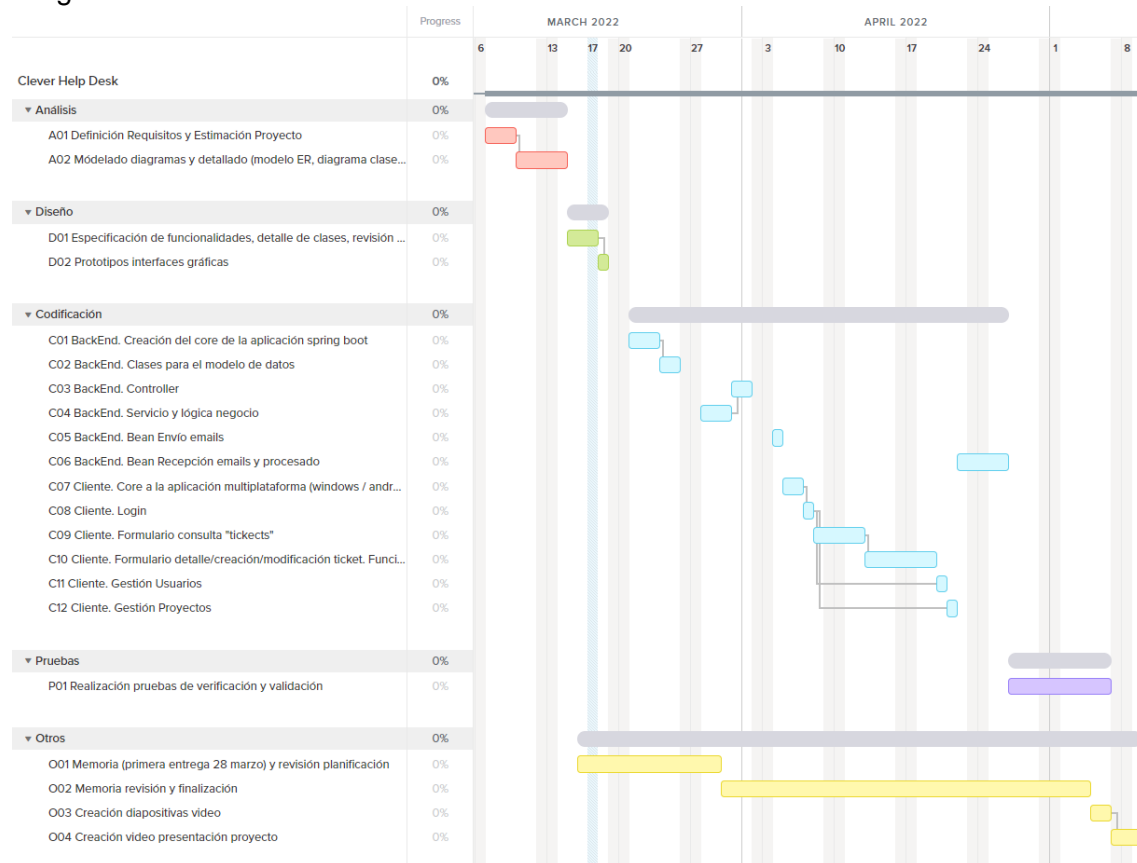
En cuanto a planificación de recursos humanos y de tiempo, yo seré el único trabajador dentro de este proyecto por lo que todas las tareas recaerán sobre mí. No contemplo coger días de vacaciones, ni festivos ya que el calendario es muy ajustado, he estimado que necesitare invertir unas 25 horas semanales (para simplificar tomaremos 5h L-V) empezando el lunes 7 marzo y terminando el 9 mayo.

Semana	Horas trabajo
07-mar	25
14-mar	25
21-mar	25
28-mar	25
04-abr	25
11-abr	25
18-abr	25
25-abr	25
02-may	25
Total (horas)	225

Ya podemos ver un ligero desvío entre las horas previstas (234h) y las horas efectivas que podré realizar (225), este pequeño descuadre lo intentaremos corregir añadiendo algún día extra de trabajo (sábado o domingo).

Y para completar esta planificación he estimado una fechas y dependencias entre tareas para que sean abordadas, las plasmo en el diagrama Gantt a continuación para tener una foto inicial del proyecto. Este dia diagrama lo voy actualizando de forma que al finalizar el proyecto podremos comparar como queda y que desviaciones hemos tenido.

Diagrama Gantt Inicial:



5 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

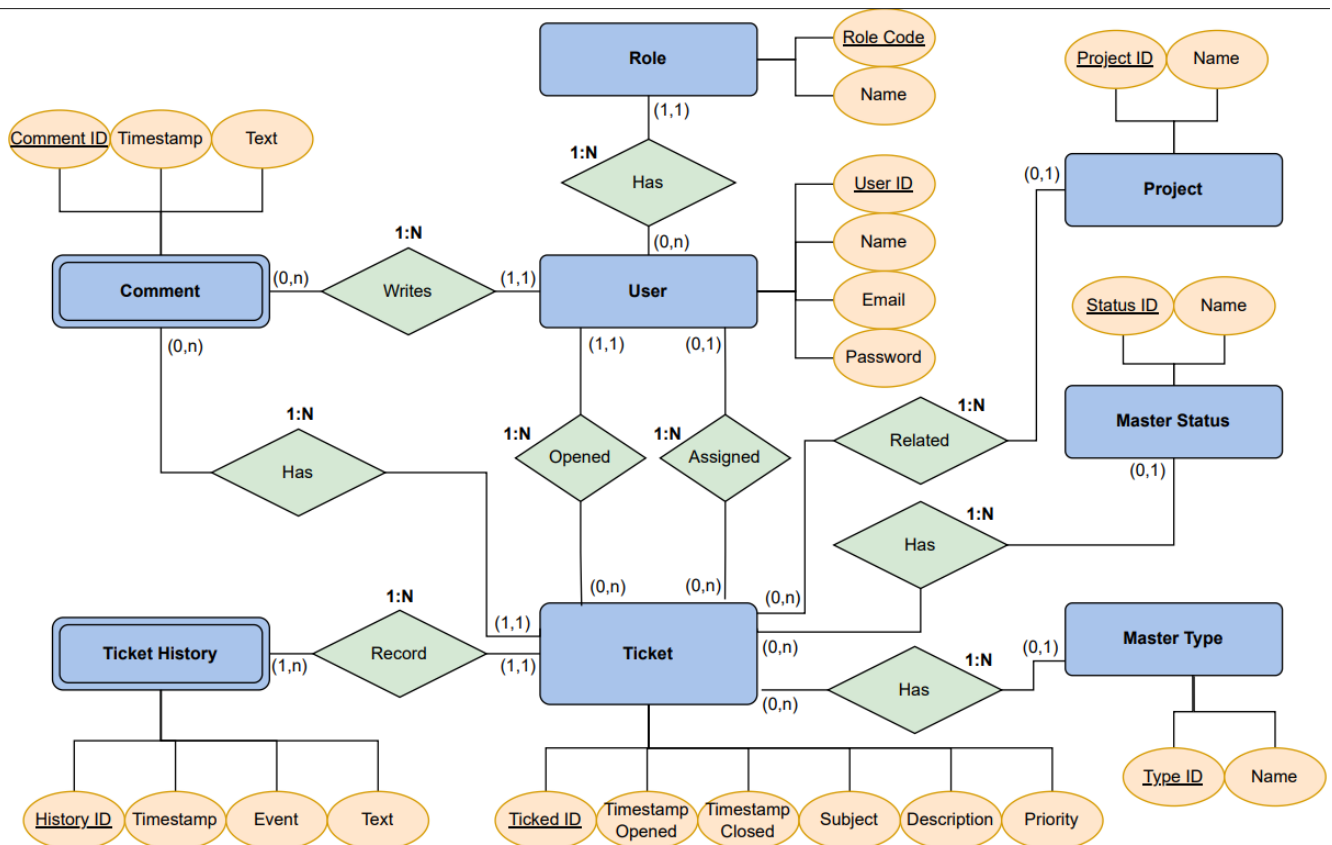
1. Se deberá hacer LOGIN para poder utilizar la aplicación CLIENTE.
2. Hay tres niveles acceso ADMINISTRADOR, AGENTE, USUARIO, se deberá tener el acceso coherente al nivel asignado.
3. Se podrán consultar los tickets según el nivel acceso:
 - a. ADMINISTRADOR y AGENTE, pueden ver todos.
 - b. USUARIO, 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.
 - a. Solo creará el ticket si la dirección de email del remitente está registrada como usuario
 - b. 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 autenticación 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.



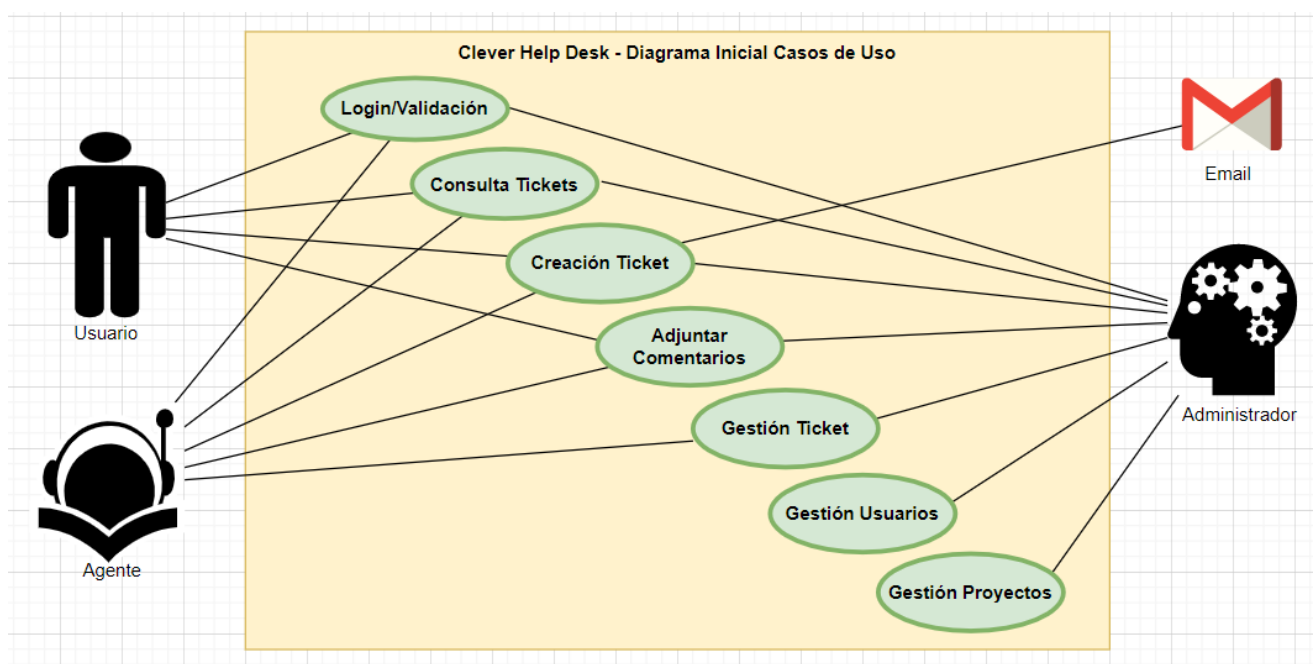
Entidades:

- **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

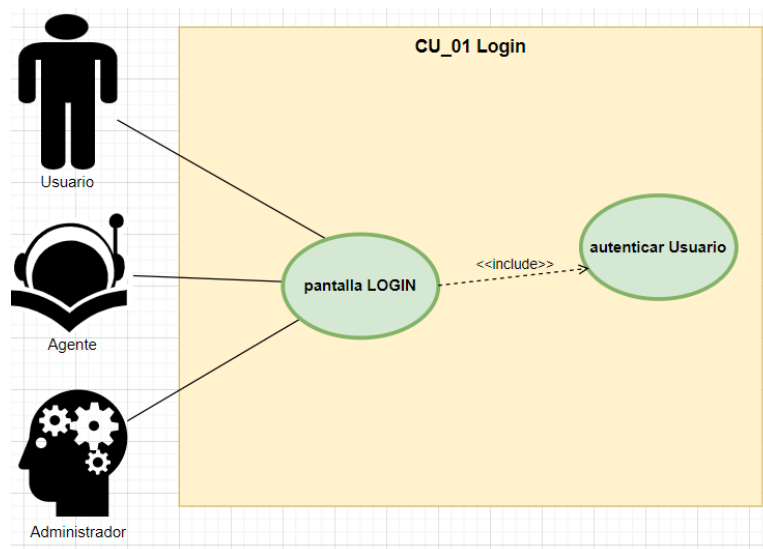
A través de los *Casos de Uso* describiremos el comportamiento que tiene el sistema desde el punto de vista del usuario o los procesos que van a interactuar con él. A continuación detallamos cada caso de uso detectado junto con su diagrama para mejor comprensión.

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



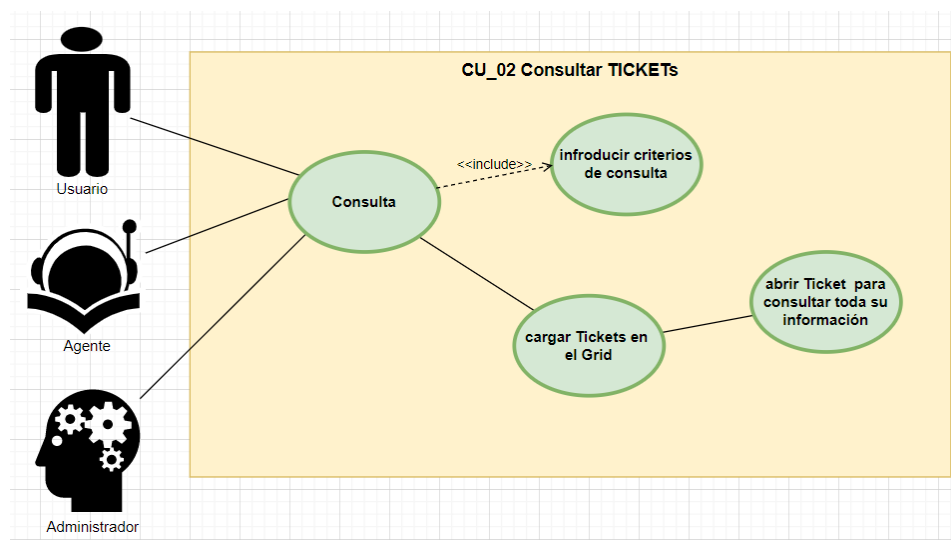
CU_01 LOGIN

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



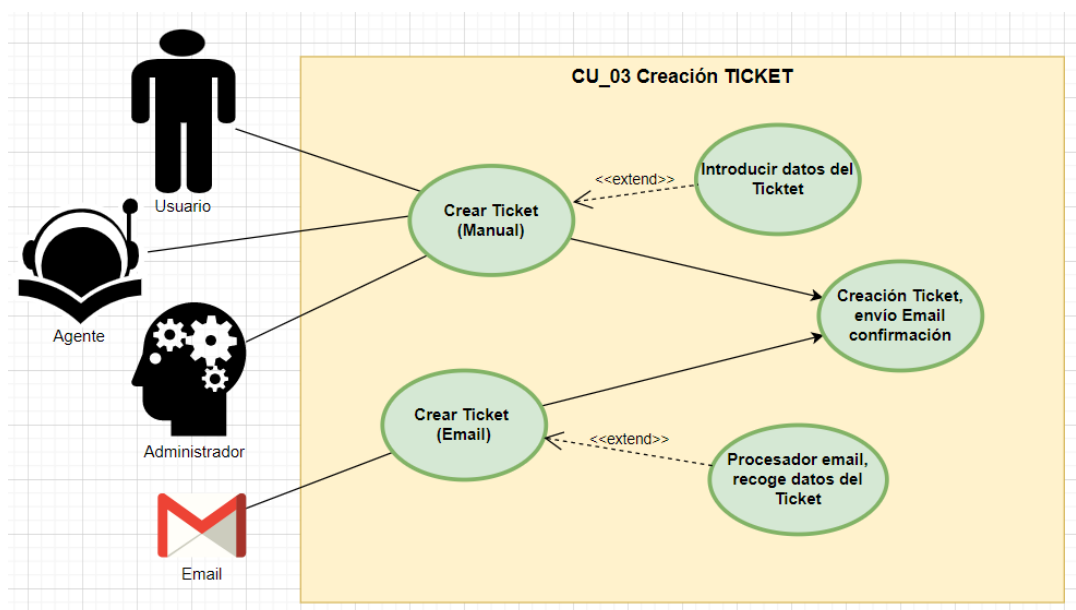
CU_02 Consultar TICKETs

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 rango 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	



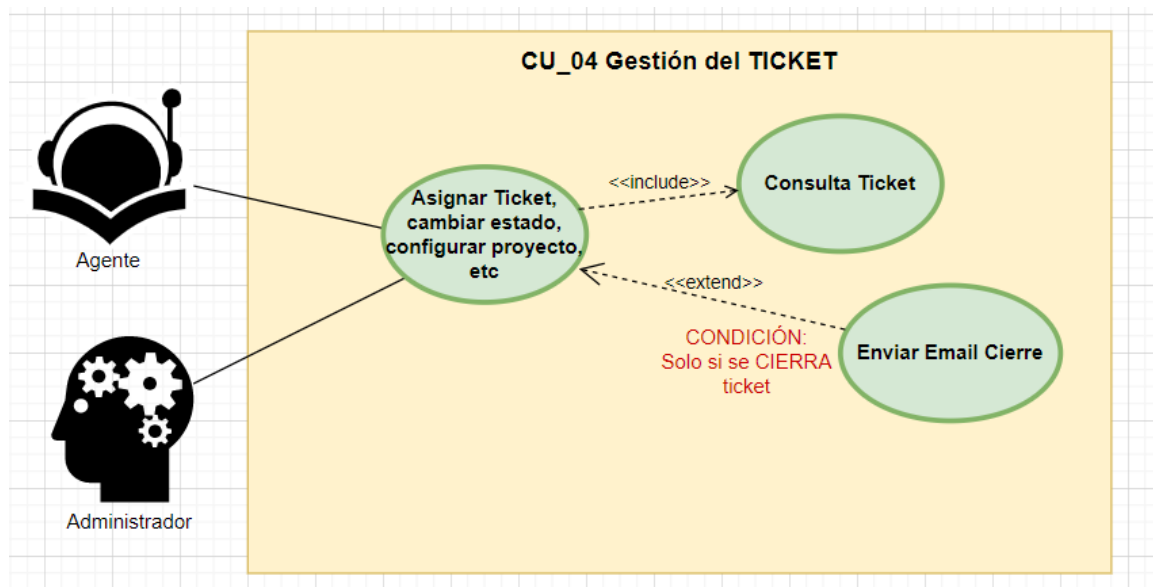
CU_03 Creación TICKET

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



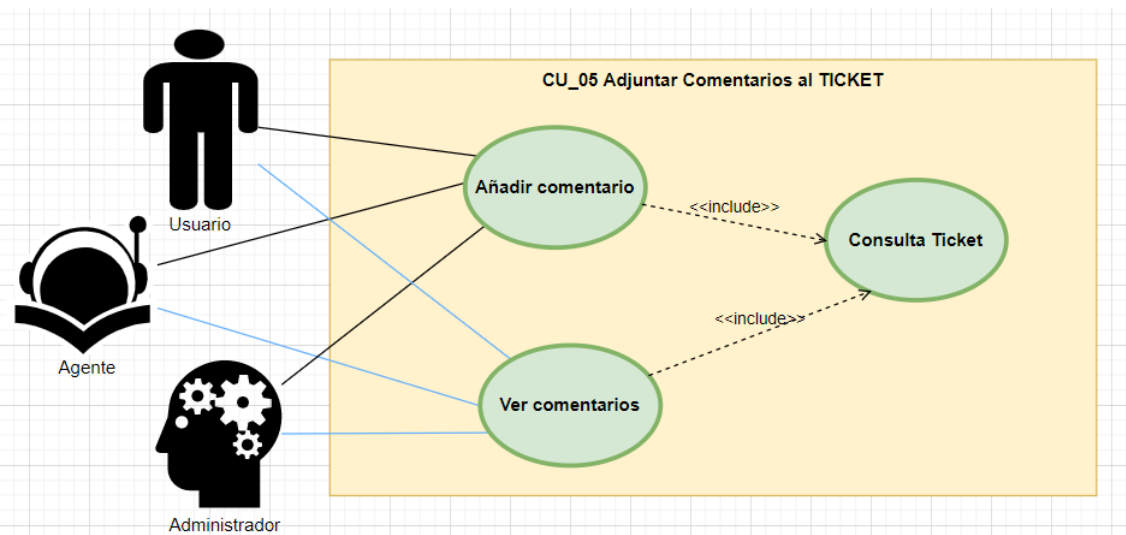
CU_04 Gestión del TICKET

Identificador de Caso Uso	CU_04
Nombre	Gestión del TICKET
Descripción	Cuando un AGENTE o ADMINISTRADOR 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 vez 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



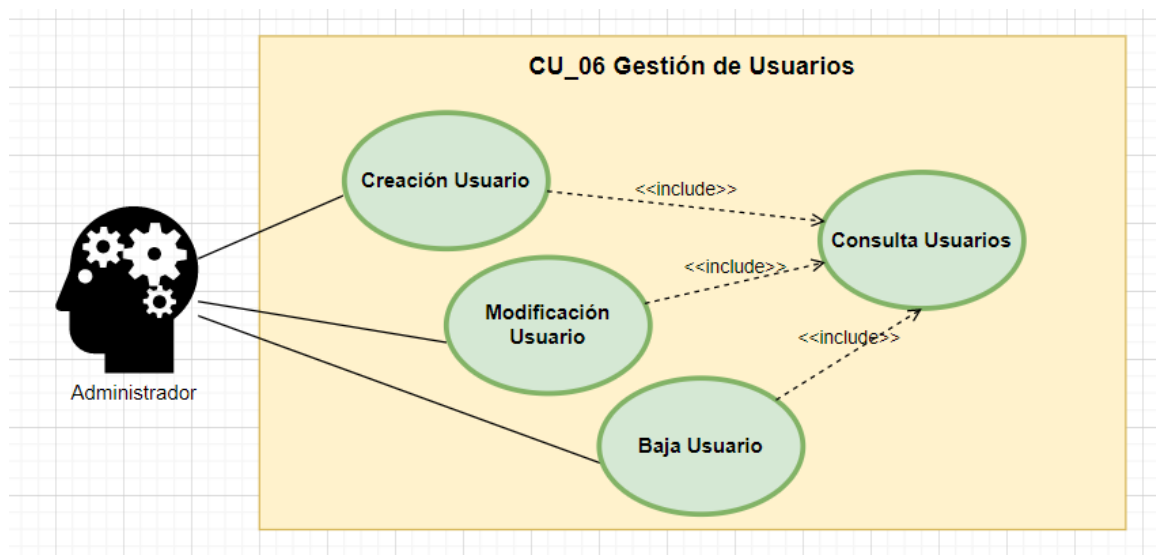
CU_05 Adjuntar Comentarios al TICKET

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



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, rellendo 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



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. Se 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

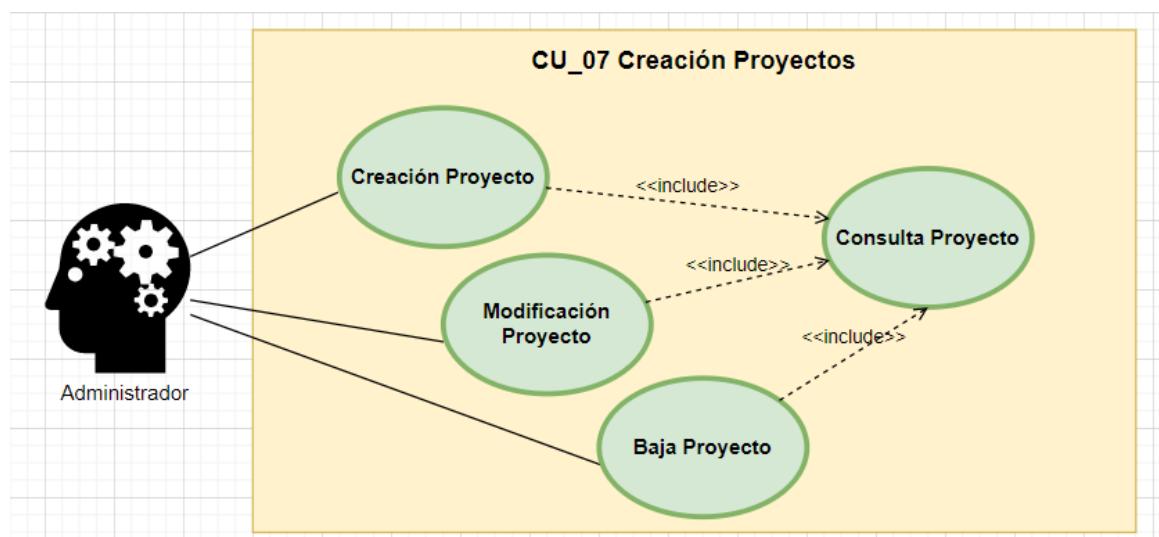
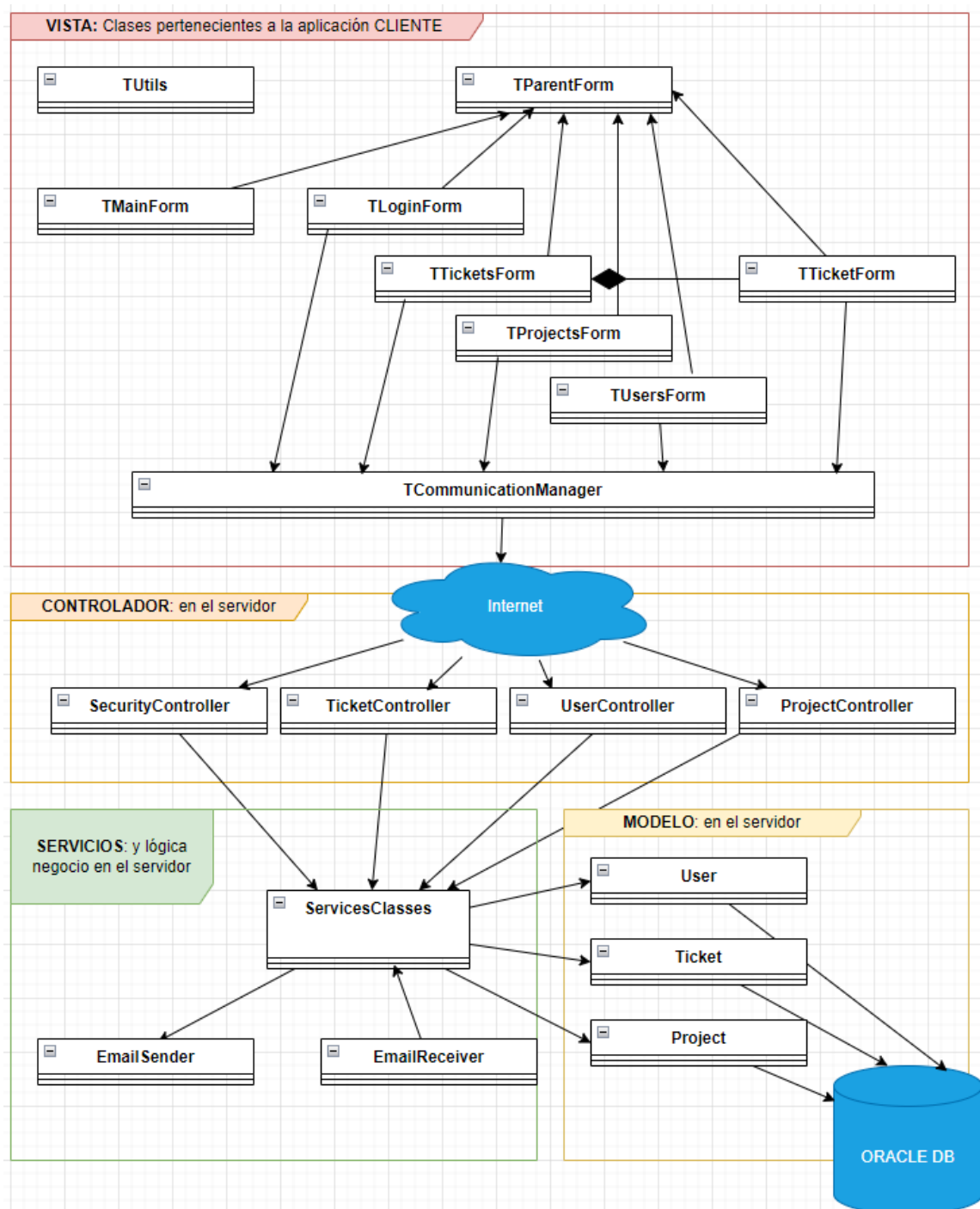


Diagrama Clases (conceptual)

En esta fase de análisis también es importante modelar un diagrama de clases para tener una aproximación mejor alcance, en este caso muestro un diagrama más conceptual ya que en la parte de diseño lo detallaré. Para facilitar el entendimiento he dividido en varios bloques, teniendo los 3 del patrón arquitectura MVC que utilizo en este proyecto más uno adicional de SERVICIOS y lógica de negocio.

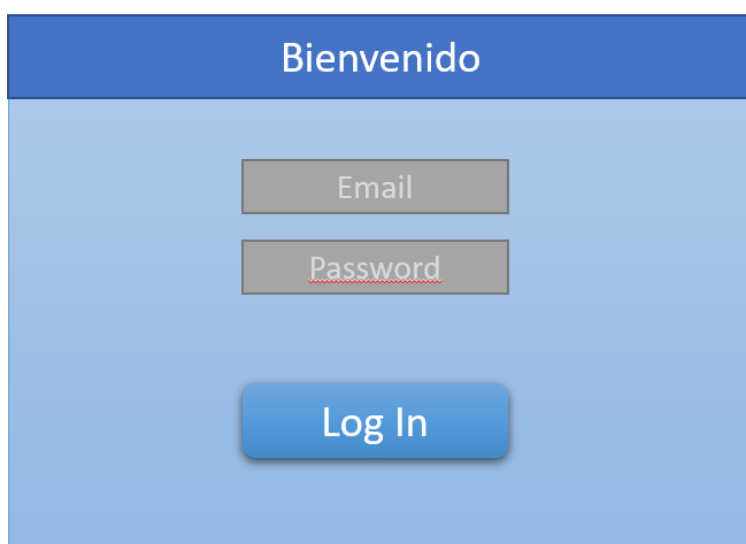


6 Diseño del proyecto

Prototipos:

En este apartado vamos a crear los prototipos de las UI de usuario.

Formulario de Login




The image shows a login form prototype. It has a blue header bar with the text "Bienvenido". Below the header, there are two input fields: "Email" and "Password". The "Password" field has a red underline. Below the input fields, there is a blue button with the text "Log In".

Consta del campo Email y Password, una vez rellenados se pulsará el botón Log In para que valide y se acceda al programa si es correcto.

Formulario principal y Consulta de Tickets

Consta de dos visualizaciones: con el menú sin desplegar y con el menú desplegado.

Prototipo sin desplegar el menú:





 Clever Help Desk			
123	CERRADO	01/02/2022	Petición permisos accesos
22213	ABIERTO	04/03/2022	Error al generar informe personal
3234	ASIGNADO	09/03/2022	Datos incorrectos en el cálculo cierre mensual
3235	CERRADO	10/03/2022	Deadlock en el PL reinspección equipajes
3236	ABIERTO	19/03/2022	Lentitud cálculo turnos MAD
3237	ABIERTO	19/03/2022	No se reciben los télex de VY

En el grid de datos aparecerán 4 columnas de información que se cargarán en función de los filtros que se hayan seleccionado desde el menú desplegable.

Las columnas son:

- Identificador único del ticket
- Estado del ticket
- Fecha apertura del ticket, por espacio en el prototipo no muestro las horas pero el formato de fecha será DD/MM/YYYY HH:MI
- Título del ticket

Según el perfil del usuario podrá acceder solo a sus tickets o podrá acceder a todos los tickets abiertos.

-  selector para poder filtrar por un tipo de estado concreto
-  selector para poder filtrar por un tipo de ticket concreto
-  selector para poder los tickets de un proyecto
-  selector para poder filtrar los tickets asignados a un agente

Formulario consulta/gestión de un ticket

Formulario donde vamos a poder ver toda la información del ticket así como poder gestionarlo y modificarlo si tenemos perfil de AGENTE o ADMINISTRADOR.

1234 – Título p.e. Error comunicación web service			
ESTADO ▼	Abierto	25/03/2022 11:11	Cerrado 25/03/2022
Tipología ▼		AGENTE ASIGNADO ▼	
*** DESCRIPCIÓN *** se detecta en el web Service de ALTEA cuando hay más de 300 pax en los logs guarda un error buffer overflow, parece que solo aparece en vuelos de UX			
Comentarios +	VICTOR	26/03/2022	Se detecta el fallo, se corrige y prueba en desarrollo
	PABLO	25/03/2022	Cliente reclama incidencia
	VICTOR	25/03/2022	En los logs del servicio se puede obtener más información

Se divide en dos partes, la zona superior donde está el ID y título del ticket así como los campos necesarios para su gestión (estado, tipo, agente y descripción principal).

En la zona inferior encontramos los posibles comentarios que se hayan escrito sobre este ticket.

7 Despliegue y pruebas

8 Conclusiones

9 Vías futuras

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

[TO-DO]

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 mayúscula, 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.

•

10 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=JioEGJlq88> 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 <i>Reformat code</i> 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 <i>Generate</i>
Duplicar línea	CTRL+D	

Anexo N. Manual estilos proyectos DELPHI

Ss

