



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escola Tècnica
Superior d'Enginyeria
Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica
Universitat Politècnica de València

Desarrollo de un Customer Portal integrado con una herramienta de Service Desk

Trabajo Fin de Grado

Grado en Ingeniería Informática

Autor: Francisco Javier Piqueras Martínez

Tutor: Patricio Letelier Torres

2016/2017

Agradecimientos

Mi agradecimiento va dedicado al director de este trabajo, Patricio, por su paciencia y el tiempo que me ha dedicado directa e indirectamente. Me ha facilitado mucho el trabajo dándome consejos, facilitándome documentación que me ha sido de gran ayuda y corrigiéndome la memoria y probando una y otra vez el producto.

Siempre que he necesitado reunirme con él ha estado disponible y su amabilidad ha hecho que disfrutara al máximo en la elaboración del trabajo final de grado. Por todo esto: ¡gracias Patricio!

Además, quiero mostrar mi agradecimiento a Carlos del Fresno, desarrollador de TUNE-UP, de nuevo por su paciencia y por implementar y modificar los servicios que he requerido a lo largo del desarrollo del proyecto. Por todo esto: ¡gracias Carlos!

Dedicatoria

Quiero dedicar este trabajo a mucha gente que ha estado apoyándome durante todo este tiempo.

En primer lugar, a mis padres, por estar siempre a mi lado, apoyándome en todo momento. Sin ellos no hubiera conseguido nada.

También quiero nombrar al resto de mi familia, quien siempre se ha interesado por saber cuál era el estado en el que me encontraba dentro de la carrera.

Por último, y no por ello menos importante, a mis amigos de carrera y de estudio, de los que he recibido gran apoyo y ayuda desde 2013 que entramos hasta ahora.

Resumen

En el ámbito del desarrollo de software, además la actividad de desarrollo, también se requiere un mantenimiento de forma paralela, especialmente, cuando un producto o parte de éste ya ha sido implantado. Se necesita un canal de comunicación entre el cliente y los desarrolladores llamado gestor de incidencias. Este gestor de incidencias posibilitará que el usuario registre cualquier tipo de incidencia en un portal (Customer Portal), que llegará a manos del equipo de soporte (personal de Service Desk) que serán los encargados de categorizar esta incidencia (error, formación, duda, etc) y definir cuál será el flujo que ésta tiene que seguir para que se solucione.

La herramienta TUNE-UP es una herramienta que nos permite gestionar el proceso de desarrollo software basado en metodologías ágiles. Con ella, cubrimos tanto la parte de desarrollo como la del equipo de Service Desk. No obstante, no contamos con el Portal de incidencias necesario para registrarlas. El usuario o cliente tiene que ponerse en contacto con el equipo de soporte para comunicar una incidencia y éstos a su vez, la registrarán manualmente. De este modo, el usuario no es conocedor en ningún momento del camino que se lleva a cabo para su resolución y tampoco tiene un canal de comunicación directa en función de una incidencia.

El objetivo de este TFG es desarrollar un Customer Portal integrado con TUNE-UP. Así, los usuarios de un producto podrán gestionar sus tickets reportando incidencias, peticiones, dudas, etc. Estos tickets se gestionarán por el personal de Service Desk en un primer nivel de soporte a través de TUNE-UP y cuando corresponda se podrán escalar como nuevos requisitos, mejoras o correcciones de fallo para ser resueltas por el equipo de desarrollo. Gracias a esta integración existirá una trazabilidad explícita entre tickets reportados por usuarios y unidades de trabajo de desarrollo (cuando corresponda). Además, los usuarios de los productos podrán gestionar sus propios tickets y conocer el estado cuando estén pendientes de ser resueltos por el equipo de desarrollo.

El portal se desarrollará con la tecnología HTML5, CSS3 y AngularJS, un framework de JavaScript de código abierto, mantenido por Google, que se utiliza para crear y mantener aplicaciones web dinámicas. Los servicios de back-end requeridos para la realización del portal los ofrecerá el equipo de desarrollo de TUNE-UP.

Abstract

In the area of software development, in addition to development activity, maintenance is also required, especially when a product or a part of it has already been implanted. You need a communication channel between the client and the developers called incident manager. This incident manager will allow the user to register any type of incident in a portal (Customer Portal), which will come into the hands of the support team (Service Desk staff) who will be in charge of categorizing this incident (error, etc) and define the flow that it has to follow to be solved.

TUNE-UP is a tool that allows us to manage the software development process based on agile methodologies. With it, we cover both the development part and the Service Desk team. However, we do not have the Incidents Portal needed to register them. The user or client has to contact the support team to report any incident.

The goal of this TFG is to develop a Customer Portal integrated with TUNE-UP. Thus, the users of a product will be able to manage their tickets reporting incidents, requests, doubts, etc. These tickets will be managed by the Service Desk at a first level of support through TUNE-UP and, when necessary, may be escalated as new requirements, improvements or bug fixes to be resolved by the development team. Thanks to this integration there will be an explicit traceability between tickets reported by users and development work units. In addition, users of a product will be able to manage their own tickets and know their status when they are pending to be resolved.

The portal will be developed with HTML5, CSS3 and AngularJS technology, an open source JavaScript framework maintained by Google that is used to create and maintain dynamic web applications. The back-end services required for the realization of the portal will be offered by the TUNE-UP development team.

Tabla de contenidos

1.	Introducción.....	13
1.1	Motivación.....	13
1.2	Objetivo	14
1.3	Estructura del trabajo.....	14
2.	Service Desk.....	15
2.1	Introducción.....	15
2.2	Funciones.....	16
2.3	Componentes e integración con desarrollo	16
3.	Customer Portal.....	20
3.1	Introducción.....	20
3.2	Características de un Customer Portal	20
3.3	Integración de Customer Portal con Service Desk.....	21
4.	Herramientas Service Desk y Gestión de proyectos software.....	23
4.1	Objetivo del estudio	23
4.2	Herramientas de Service Desk.....	23
4.2.1	Zendesk.....	23
4.2.2	Desk.com.....	24
4.2.3	Tender.....	26
4.2.4	ServiceTonic	27
4.2.5	Resumen.....	29
4.3	Estudio de herramientas de gestión de proyectos.....	29
4.3.1	Jira	29
4.3.2	Ontime	32
4.3.3	Version One.....	34
4.3.4	Team concert.....	35
4.3.5	TUNE-UP Process	36
4.3.6	Resumen.....	39
5.	Integración de Customer Portal, Service Desk y Desarrollo	41
5.1	Introducción.....	41
5.2	Estructura del producto de desarrollo.....	41
5.3	Definición de unidades de trabajo.....	42
5.3.1	Unidades de trabajo de desarrollo	42

5.3.2	Unidades de trabajo del Service Desk.....	43
5.3.3	Unidades de trabajo del Customer Portal. Tickets.....	44
5.3.4	Protocolo de Integración	45
5.4	Resumen.....	46
6.	Caso Práctico con TUNE-UP Customer Portal	48
6.1	Introducción.....	48
6.2	TUNE-UP Process como herramienta de Service Desk	48
6.3	Descripción de la herramienta.....	48
6.3.1	Gestión de tickets.....	49
6.3.2	Gestión de la administración de contactos	52
6.3.3	Gestión de perfil de contacto.....	54
6.4	Proceso de desarrollo y tecnologías utilizadas.....	55
6.5	Arquitectura y organización del código.....	57
6.6	Manual de instalación	61
7.	Conclusiones y trabajo futuro	63
8.	Referencias.....	64



Tabla de ilustraciones

Ilustración 1. Portal de usuario y Service Desk.....	13
Ilustración 2. Service Desk como único punto de contacto	15
Ilustración 3. Estructura básica de los componentes del Service Desk	17
Ilustración 4. Esquema de interacción entre el Service Desk y Desarrollo.	19
Ilustración 5. Flujo de un ticket.....	21
Ilustración 6. Correspondencias entre tickets del Customer Portal y tickets del Service Desk.....	22
Ilustración 7. Ticket creado a partir de un correo.....	23
Ilustración 8. Bandeja de entrada del usuario de soporte	24
Ilustración 9. Priorización de tareas mediante Desk.com.....	24
Ilustración 10. Creación de foros y preguntas	25
Ilustración 11. Organización de la base de datos del conocimiento.....	25
Ilustración 12. Widget incrustado en la web del cliente	26
Ilustración 13. Agrupación de tickets Tender	26
Ilustración 14. Base de datos del conocimiento.....	27
Ilustración 15. Gestión de incidencias.....	27
Ilustración 16. Gestión de reglas de negocio.....	28
Ilustración 17. Base de datos del conocimiento	28
Ilustración 18. Comparativa de herramientas de Service Desk.....	29
Ilustración 19. Gestión de incidencias en Jira.....	30
Ilustración 20. Gestión de workflows en Jira	30
Ilustración 21. Priorización de tareas en Jira	30
Ilustración 22. Gestión de backlog en Jira con GreenHopper	31
Ilustración 23. Uso de Kanban + Scrum	32
Ilustración 24. Gestión del backlog en OnTime.....	33
Ilustración 25. Gestión de tareas en OnTime	33
Ilustración 26. Organización de tareas en Version One	34
Ilustración 27. Interacción con el usuario.....	35
Ilustración 28. Gestión del backlog en Team Concert.....	35
Ilustración 29. Personal Planner (PEP)	37
Ilustración 30. Gestor de UTs	38
Ilustración 31. Requeriments Manager (REM).....	38
Ilustración 32. Version contents & tracking.....	39
Ilustración 33. Comparativa de herramientas de desarrollo	39
Ilustración 34. Nodos de un producto	42
Ilustración 35. Diagrama de clases fallo.....	43
Ilustración 36. Diagrama de clases incidencia.....	44
Ilustración 37. Diagrama integración Customer Portal, Service Desk y Desarrollo	46
Ilustración 38. Proceso desde la creación de una incidencia hasta su resolución	47
Ilustración 39. Solicitud de ticket en TUNE-UP Customer Portal.....	49
Ilustración 40. Vista de tickets en TUNE-UP Customer Portal	50
Ilustración 41. Vista de tickets administrador en TUNE-UP Customer Portal	50
Ilustración 42. Detalles de ticket y paso de mensajes en TUNE-UP Customer Portal..	51
Ilustración 43. Detalles del ticket en TUNE-UP Process.....	51
Ilustración 44. Gestión de contactos en TUNE-UP Customer Portal.....	52

Ilustración 45. Alta de un nuevo contacto en TUNE-UP Customer Portal.....	53
Ilustración 46. Visualización de lista de contactos por cliente en TUNE-UP Process...	54
Ilustración 47. Perfil de contacto en TUNE-UP Customer Portal.....	55
Ilustración 48. Resumen UTs TUNE-UP Customer Portal.....	55
Ilustración 49. Proceso incremental e iterativo de desarrollo	56
Ilustración 50. Organización del código 1	57
Ilustración 51. Organización del código 2	58
Ilustración 52. Descarga del repositorio	62



1. Introducción

1.1 Motivación

En el mercado, existen herramientas para gestionar el trabajo del equipo de Service Desk y el equipo de desarrollo, así como herramientas de ticketing, que facilitan la gestión y seguimiento de incidencias en un producto software.

Una herramienta de Service Desk sirve para llevar a cabo una gestión de todas las incidencias que puedan surgir de un producto software, desde fallos y errores, hasta propuestas de mejora. Éste Service Desk se encarga de solucionar o proporcionar una solución al motivo de la incidencia para su posterior cierre. Es de vital importancia que cada organización disponga de un equipo de Service Desk que se encargue exclusivamente de esta tarea.

La motivación de este trabajo está en integrar un portal de usuario con una herramienta de Service Desk para tener, de una forma sencilla y práctica, toda la gestión de incidencias concentrada en una sola herramienta, sin necesidad de llamadas telefónicas, o de tener que atender al correo corporativo para la resolución de éstas.

La [Ilustración 1](#) muestra cómo sería ésta comunicación. Por la derecha se encuentra el usuario, quien da de alta la incidencia y por la izquierda el técnico de Service Desk que la recibe, cataloga y soluciona.



Ilustración 1. Portal de usuario y Service Desk

1.2 Objetivo

Los objetivos de este trabajo son:

- Establecer un mecanismo de comunicación entre los clientes usuarios y el equipo de Service Desk y desarrollo, desde la creación de tickets de incidencias por parte del usuario hasta el cierre del ticket por parte del técnico de Service Desk o de un desarrollador al cual ha sido escalado el ticket.
- Facilitar al cliente usuario un seguimiento del estado de sus tickets en tiempo real.
- Aplicar la integración en un caso práctico con el uso de una herramienta. Establecer mejoras a la herramienta TUNE-UP Process respecto de las funcionalidades necesarias para la integración.

Para definir mejor el trabajo que tiene que realizar cada equipo (Service Desk y desarrollo), se llevará a cabo la definición de un método de trabajo que sirva de guía para enlazar estos dos equipos de la forma más eficiente posible. Este método de trabajo no estará sujeto a ninguna herramienta, por tanto se podrá aplicar a cualquier equipo que sienta la necesidad de tener estos equipos conectados.

Aunque el método de trabajo definido es independiente de la herramienta que se utilice, en este trabajo, se ha optado por utilizar la herramienta TUNE-UP Process para poder tener una visión práctica de cómo llevar a cabo la integración.

1.3 Estructura del trabajo

En primer lugar, en el [capítulo 2](#) y [capítulo 3](#), se definirán los conceptos en los que se va a basar la memoria, que son Service Desk y Customer Portal, explicando el trabajo del equipo que compone cada uno de ellos y como se integrarían, en cuanto a proceso, ambas partes.

Una vez analizadas las actividades de los dos conceptos antes mencionados, en el [capítulo 4](#) se estudiarán las herramientas que hay en el mercado.

Después, en el [capítulo 5](#), se propondrá una metodología a seguir para tener una trazabilidad completa entre el Service Desk, desarrollo y Customer Portal.

Seguidamente, en el [capítulo 6](#), se definirán todos estos conceptos en la herramienta TUNE-UP Process y se propondrá un caso práctico consistente en una herramienta de ticketing que hará la función de Customer Portal, que se integra directamente con la herramienta de Service Desk.

Por último, en el [capítulo 7](#) y se hará una conclusión y se expondrá el trabajo futuro esperado.

2. Service Desk

2.1 Introducción

La gestión de servicios de TI (ITSM) se refiere a la totalidad de actividades (estructuradas en procesos) que son realizadas por una organización para planificar, diseñar, entregar, operar y controlar los servicios de tecnología de información ofrecidas a clientes. Así, se ocupa de la implementación de servicios de TI que satisfagan las necesidades de los clientes y es realizada por el proveedor de servicios a través de una mezcla apropiada de personas, procesos y tecnología de la información.

El objetivo del Service Desk es proporcionar un único punto de contacto (“SPOC”) para satisfacer las necesidades de comunicación tanto de los usuarios como del personal de TI, y también para satisfacer tanto los objetivos de los clientes como los de los proveedores de TI. Se habla de usuario como el usuario real de servicio mientras que el término cliente es referido a la entidad que está pagando por dicho servicio. En la [Ilustración 2](#) se refleja de forma clara esa dependencia de Service Desk como punto de contacto.



Ilustración 2. Service Desk como único punto de contacto

Gracias al Service Desk, se evitan problemas en las organizaciones. Por ejemplo, cuando un cliente genera una incidencia debe ser atendido de manera rápida y no estar llamando a diferentes lugares y siendo transferido a diferentes áreas hasta localizar a la persona indicada ya que esto ocasionará disgusto en ellos.

Service Desk dispone de un registro y administra todos los incidentes entre el servicio de negocio y los clientes manteniendo informados a éstos del estado de sus requerimientos.

2.2 Funciones

Son muchas las funciones que un Service Desk nos ofrece y gracias a ellas se puede hacer una clasificación de éstos en diferentes categorías (Help Desk, centro de llamadas, etc).

La primera función básica de un Service Desk es recibir llamadas. En la actualidad el teléfono es uno de los medios más utilizados para la comunicación, establecer un centro de atención de llamadas permite realizar un gran número de acciones con gran agilidad, lo que hace que no sea necesario desplazarse.

De forma paralela, un Service Desk llevará a cabo las siguientes tareas, que se llevarán a cabo de forma secuencial, consiguiendo así el circuito básico de funcionalidad. Estas tareas son:

- Registro y seguimiento de incidentes.
- Realizar una evaluación inicial sobre los requerimientos.
- Solucionar o escalar a otro nivel para su posterior resolución dichos requerimientos.
- Cierre de incidencias y comunicación a los clientes.

De este modo, gracias a Service Desk, el coordinador de una organización debe tener a su disposición recursos como:

- Toda la información necesaria sobre todos los problemas y dudas que los clientes tienen en las operaciones que realizan con su organización.
- Información relativa a las acciones que está llevando a cabo su organización a través de sus diferentes áreas para resolver problemas y dudas de sus clientes, que le permita verificar el estado de avance que se encuentra cada uno de ellos.
- Información histórica, para poder efectuar un análisis de los problemas y eventos que han sido reportados por sus clientes.

2.3 Componentes e integración con desarrollo

Service Desk es conocido comúnmente conocido como el primer nivel de soporte técnico. Los técnicos de este nivel son técnicos con conocimientos generales en cuanto a amplitud (aunque no necesariamente en cuanto a profundidad), de los problemas que se pueden presentar a los usuarios finales. En algunas organizaciones, existe un segundo nivel de soporte, un poco más especializado (por ejemplo, redes, hardware, etc), en el que escalar ciertos problemas si no pueden ser resueltos en el primer nivel.

Un Service Desk maneja sus tareas usando un sistema de solicitud por ticket, de esta forma, cuando los usuarios tienen algún problema, llenan un ticket de Service Desk (por teléfono o a través de un portal gestor de incidencias), que posteriormente es catalogado.

Además, los técnicos de Service Desk llevan a cabo otras tareas consistentes en la recolección y uso de datos. Todas las peticiones (que son registradas) proporcionan información valiosa que la organización puede aprovechar para el beneficio propio para tomar decisiones acerca del mejoramiento del soporte técnico (por ejemplo, programar más capacitaciones sobre ciertos aspectos).

Una estructura básica de los componentes del Service Desk sería la mostrada en la Ilustración 3.

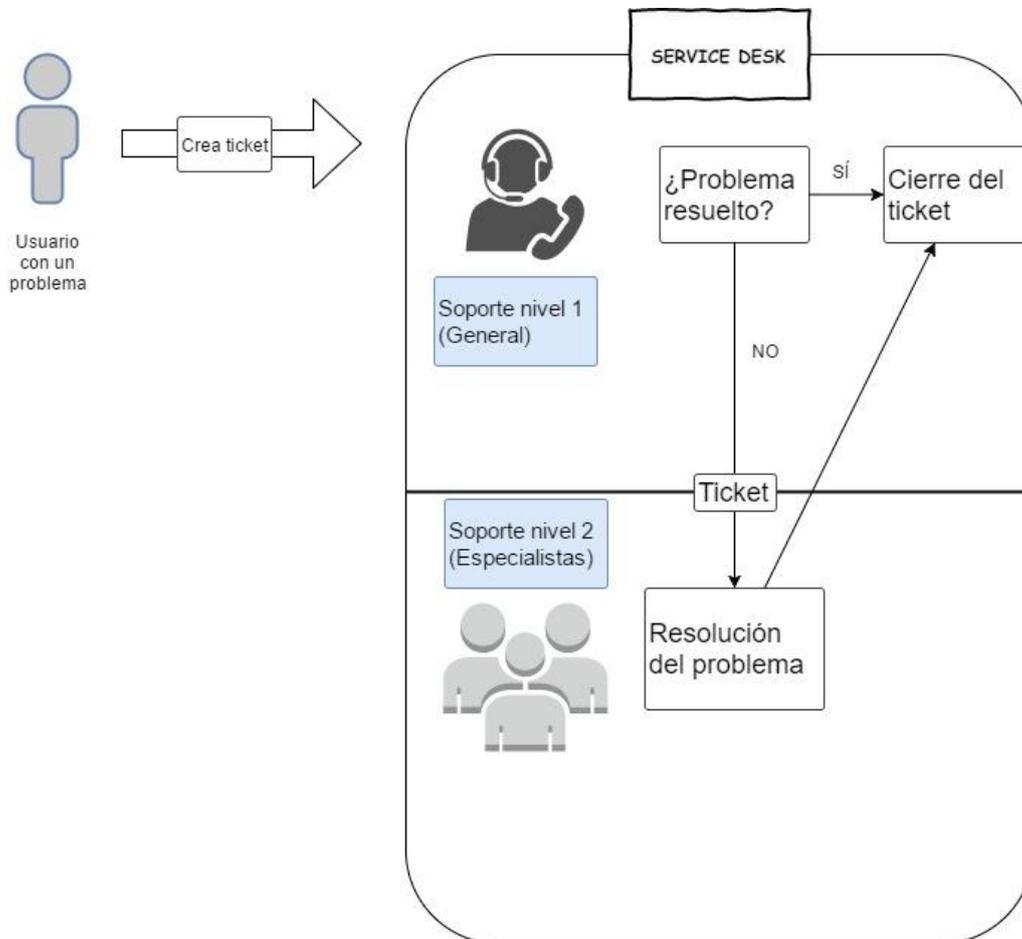


Ilustración 3. Estructura básica de los componentes del Service Desk

Para medir el éxito, se suelen considerar ciertos indicadores, que típicamente son:

- El porcentaje de solicitudes por ticket cerradas exitosamente.
- El porcentaje de solicitudes por ticket escaladas al nivel especializado.
- El tiempo que toma responder una solicitud por ticket y cerrarla.
- La satisfacción del usuario final (o cliente) con la cortesía, paciencia y ayuda de los técnicos.

En el contexto del desarrollo software, este nivel de soporte especializado, serían los desarrolladores, quienes resolverían el problema y devolverían el ticket al nivel uno para su posterior cierre y comunicación al cliente.

Estos desarrolladores, podrían ser encargados de mantenimiento exclusivamente, o desarrolladores en implementación. Esta estructura depende de la propia organización y del estado actual del producto (en desarrollo, en mantenimiento, etc).

Un ejemplo claro del circuito básico del Service Desk integrado con desarrollo sería el siguiente:

“El cliente de una empresa que se dedica a desarrollar un sistema de gestión hospitalaria abre un ticket porque no consigue dar una cita a un paciente. Transcurridos unos minutos, el técnico de soporte de Service Desk visualiza el nombre y la descripción detallada del ticket, quién evalúa el problema comprobándolo él mismo. Tras realizar la acción con los datos, en la situación que el cliente le ha reportado, se da cuenta que no es capaz de solucionar la incidencia, por lo que escala el ticket al siguiente nivel (desarrollo). Un desarrollador del equipo, visualiza el ticket y se pone a trabajar en él, identifica el error y lo soluciona. Tras ello, le devuelve el ticket al técnico de soporte de nivel uno indicándole cuál era el problema y cuál ha sido su resolución (además de ciertas indicaciones de funcionalidad para el cliente si son necesarias). El técnico de soporte cierra el ticket, comunicándole al cliente que el problema ha sido resuelto y los detalles de la resolución si se considera necesario. Seguidamente, se toma nota de los datos necesarios para los indicadores de éxito y se atiende al siguiente ticket.”

Dado el contexto del proyecto, a partir de este punto, se hablará del nivel de especialistas, como equipo de desarrollo y mantenimiento.

La comunicación entre el cliente, el Service Desk y es vital para garantizar el éxito del proyecto ya que entre los tres equipos fluye mucha información. Esta información tiene que estar recogida en una herramienta que sea fácil de utilizar y accesible para todos los equipos, de forma que si algún miembro del equipo tiene una duda o necesita consultar algo del proyecto, lo pueda hacer de una manera rápida y eficiente.

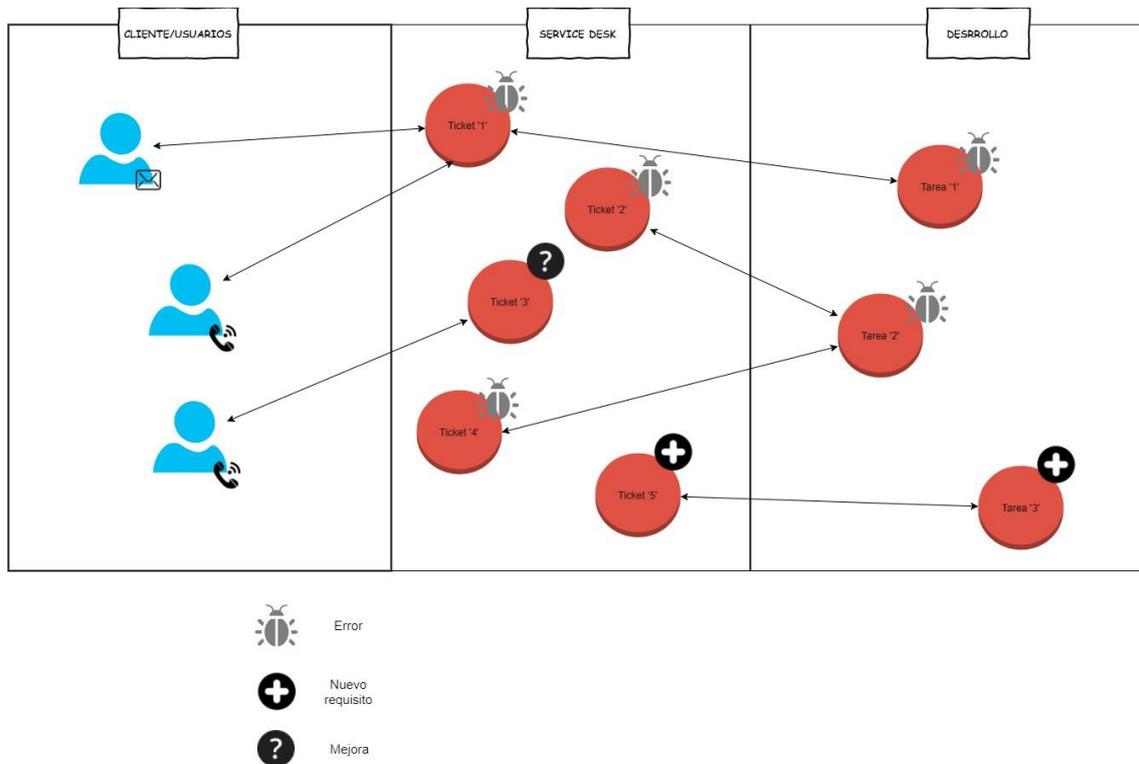


Ilustración 4. Esquema de interacción entre el Service Desk y Desarrollo.

En la **Ilustración 4** se puede apreciar cómo para un mismo ticket en soporte, se crea una única unidad de trabajo en desarrollo. Esto es dado ya que puede haber duplicidad de problemas, reportadas por diferentes usuarios, que deben estar informados del estado del él, pero que el proceso de resolución para los problemas es el mismo.

3. Customer Portal

3.1 Introducción

Un Customer Portal (Portal de Cliente) o Client Portal, es una puerta de enlace electrónica a una colección de archivos, servicios e información digitales, accesible a través de Internet.

En el contexto de la gestión de incidencias, este concepto se aplicaría a la puerta de enlace entre el cliente y el equipo de soporte (Service Desk) con el objetivo de dar de alta, modificar, visualizar y eliminar incidencias respectivas a un producto o sistema concreto. De esta forma, el cliente sería consciente en todo momento del estado de su ticket, siendo transparente para él el proceso interno que la organización lleva a cabo para la resolución del problema.

3.2 Características de un Customer Portal

Para satisfacer el objetivo de un portal de incidencias para un cliente, éste debe cumplir de forma parcial o total las características explicadas a continuación.

En primer lugar, éste debe permitir al usuario poder realizar las siguientes funciones:

- Dar de alta un ticket. El usuario podrá registrar nuevas incidencias a modo de tickets con un nombre claro. El ticket además, deberá tener una descripción detallada, incluyendo el escenario en el que el problema ha aparecido y los datos utilizados. Se deberá apoyar de imágenes o capturas de pantalla si éstas aportan información de valor para el entendimiento del problema. Además, si el portal lo permite, el usuario podrá catalogar (a la espera de la evaluación de los miembros del equipo de soporte técnico) la incidencia (fallo, duda, etc.).
- Visualización de tickets. Es importante que el cliente y usuario que ha reportado la incidencia tengan una visión clara del estado actual de cada uno de los tickets, permitiendo ver fecha de alta, fecha de cierre (si cabe), nombre del ticket, identificador y descripción.
- Contacto con soporte. Además, debe haber un canal de comunicación abierto dentro de cada ticket para las consultas de cualquiera de los dos lados, pues pueden haber dudas o aclaraciones que necesiten ser resueltas por ambos lados.
- Información de contacto. Es imprescindible que el usuario que reporta una incidencia tenga actualizada su información de contacto (correo electrónico, teléfono, etc.) pues en un determinado momento, un miembro puede necesitar ponerse en contacto con él para obtener información más precisa que la que se puede obtener por paso de mensajes.

Además, sería muy interesante que fuera una aplicación web, ya que dota al sistema de portabilidad, lo que hace que un cliente o usuario, pueda crear tickets con mucha comodidad, así como visualizar su estado sin necesidad de que éste tenga que realizar ningún tipo de instalación.

Otra característica muy importante, sería que los usuarios, cuando den de alta tickets vía telefónica, sean identificados para poder realizar el seguimiento por el portal del

mismo modo que si lo hubieran dado de alta a través de él. Se puede observar el flujo entero que puede seguir un ticket en la [Ilustración 5](#).

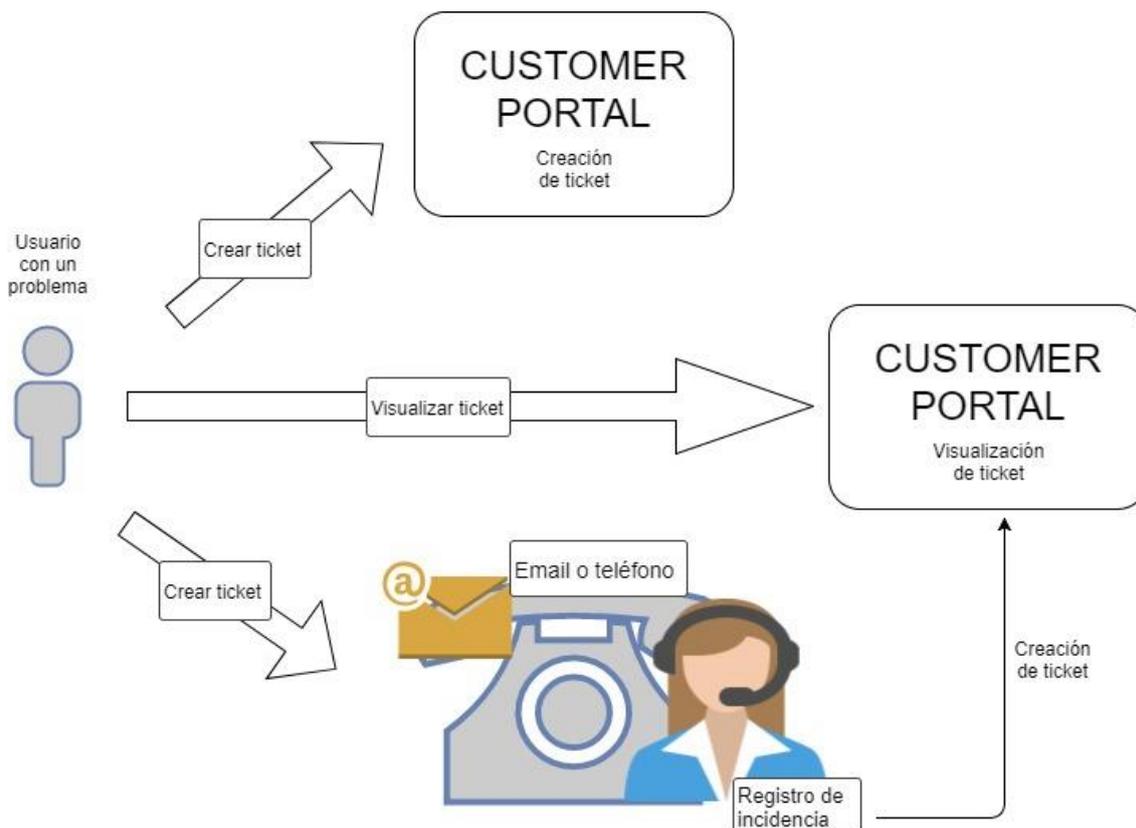


Ilustración 5. Flujo de un ticket

3.3 Integración de Customer Portal con Service Desk.

Por lo general, esta herramienta forma parte del sistema de Service Desk, ya que van cogidas de la mano. No obstante, se ha tratado de diferenciar ambas partes de ella, debido a que el objetivo del proyecto es desarrollar un portal web para una herramienta que no precisaba de éste.

Esta herramienta, de la que se habla más adelante, está preparada para poder usarse como herramienta de Service Desk. No obstante, todo queda de cara a la organización, el cliente o usuario en ningún momento es conocedor del estado de sus problemas y la resolución que se está llevando a cabo.

De este modo, dispondremos de dos herramientas, para el portal de usuarios, y para la gestión de los tickets. Como se ha comentado, el portal no es más que un canal de comunicación, por lo que todos los datos necesarios para su implementación son extraídos de la herramienta de Service Desk, filtrando los datos que queremos que sean transparentes para el usuario del portal.

Esto se puede observar en la [Ilustración 6](#).

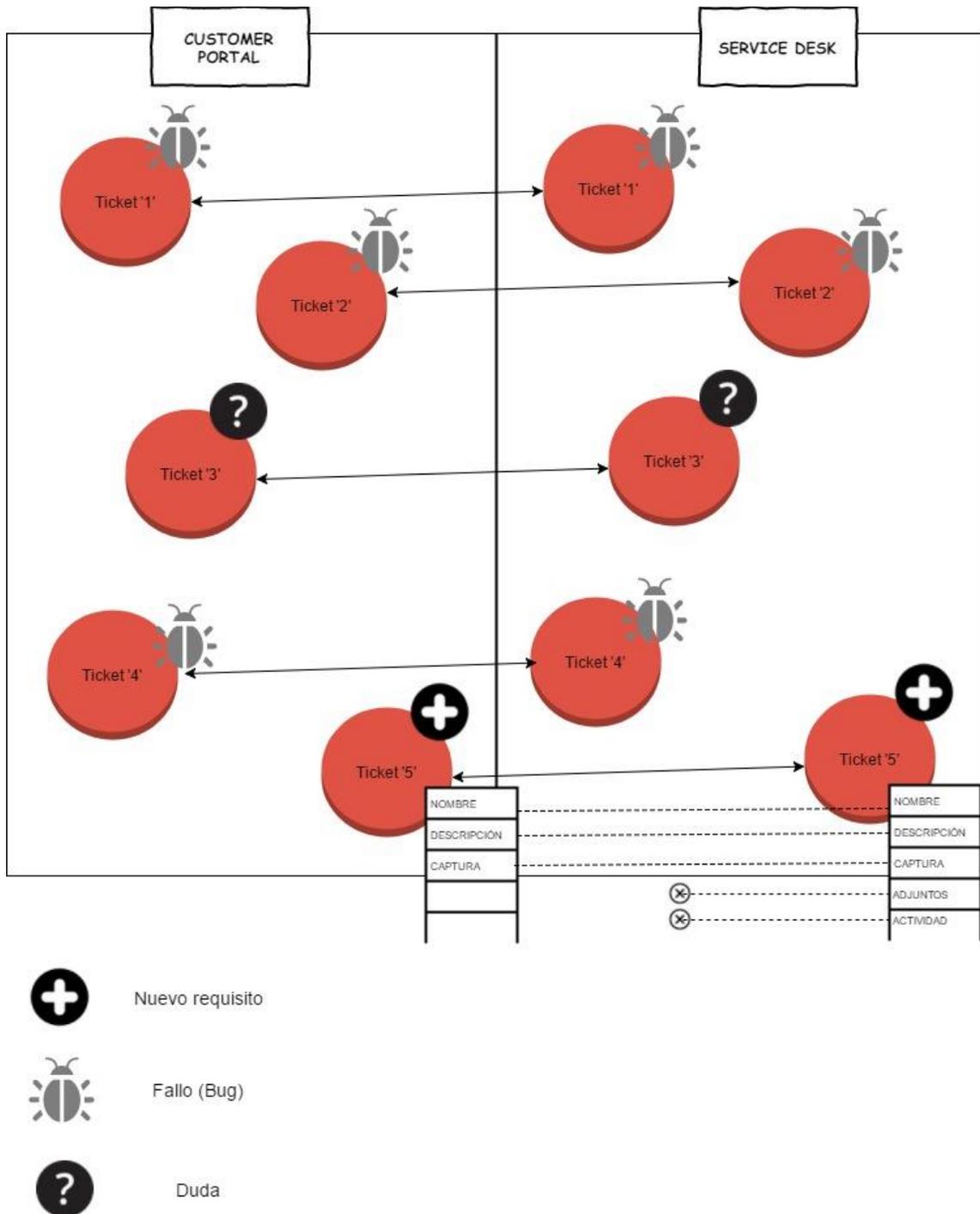


Ilustración 6. Correspondencias entre tickets del Customer Portal y tickets del Service Desk.

Como se puede observar, el ticket real está alojado en el Service Desk, mientras que el ticket mostrado en el Customer Portal no es más que una copia del ticket, filtrando toda la información que no es relevante para el usuario del portal.

Sin embargo, la creación de un ticket puede hacerse tanto en la herramienta de Service Desk como en el Customer Portal.

4. Herramientas Service Desk y Gestión de proyectos software.

4.1 Objetivo del estudio

En este apartado se va a realizar un estudio de las principales herramientas que se utilizan en la gestión de un Service Desk en el mercado. Además, también se va a realizar un estudio de las principales herramientas que se utilizan en la gestión de un proyecto de desarrollo.

Las características en que se centrará el estudio de herramientas de desarrollo son parecidas a las de las herramientas de Service Desk:

- Priorización del trabajo.
- Gestión de Workflows.
- Gestión de la duplicidad de trabajo.
- Integración de la estructura del producto.

4.2 Herramientas de Service Desk

4.2.1 Zendesk

Zendesk es una herramienta orientada a la gestión de un Service Desk, el cliente se puede comunicar con el equipo de soporte mediante diferentes canales: cuenta de correo, twitter, facebook, chat o desde su propia web, de esta forma, puede utilizar el canal que quiera y el equipo de soporte puede ver todas las incidencias en un solo lugar.

Cuando se recibe un correo o cualquier otra petición de los diferentes canales, se crea un ticket asociado como se muestra en la [Ilustración 7](#).



<input type="checkbox"/>	SUBJECT	REQUESTER	REQUESTED	PRIORITY	GROUP	ASSIGNEE	UPDATED
<input type="checkbox"/>	Problems using the XR-50 Hypershot	Bobby Bob	3 minutes ago	-	-	-	3 minutes ago

Ilustración 7. Ticket creado a partir de un correo

Los tickets que se crean se pueden clasificar en cuatro tipos: duda, incidencia, problema y tarea. Una vez dado de alta, se puede dar prioridad al ticket, los tipos de prioridad son: urgente, alta, normal o baja. La prioridad del ticket se usa en todo el Service Desk de ayuda para generar vistas e informes y también se utiliza como condiciones y acciones en automatizaciones y disparadores.

El cliente puede crear comunidades de soporte para plasmar sus mejoras y preguntas sobre el producto.

Proporciona foros para crear una base de conocimientos de información de soporte para productos y servicios. Los usuarios finales pueden tener la posibilidad de comentar sobre los temas o crear nuevos temas.

Los tickets pueden compartirse para permitir la colaboración en la resolución de los mismos.

La creación o actualización de tickets en Zendesk genera eventos. Estos eventos se pueden usar para modificar automáticamente los tickets y enviar notificaciones de correo electrónico. Por ejemplo, cuando se crea un nuevo ticket, Zendesk envía una confirmación por correo electrónico a la persona que generó el ticket (el solicitante).

También se puede asignar automáticamente un ticket a un agente o grupo de soporte específico según la dirección de correo electrónico al que fue enviado, la organización a la cual pertenece el solicitante o las palabras clave incluidas en el mensaje de solicitud.

4.2.2 Desk.com

Desk.com aglutina todas las tareas que se asignan al Service Desk desde diferentes plataformas: correo electrónico, twitter, facebook, chat, teléfono.

Se organizan estas entradas en forma de tickets en la bandeja de entrada del usuario de soporte como se muestra en la [Ilustración 8](#).

	CUSTOMER		SUBJECT
	Larita Villas	travel	Missed my flight. Help!
	Linh Pham		Do you offer sailing lessons?
	Carlos Carballo	more info	Qué debo llevar en mi viaje?
	Nolan Gett	lodging	Hotels in Cabo San Lucas?
	Quinton Sary		My family and I are planning a vacation and ...
	David Hanson		Where is the best place to grab gear?

Ilustración 8. Bandeja de entrada del usuario de soporte

Se pueden priorizar las tareas mediante los siguientes estados: nuevo, abierto, pendiente, resuelto y cerrado como se muestra en la [Ilustración 9](#).

	CUSTOMER	SUBJECT		STATUS	UPDATED	ASSIGNED	PRIORITY	ACTIONS	
	Larita Villas	travel	Missed my flight. Help!	1	Open	4 hours ago	Diego Garcia	7	
	Linh Pham		Do you offer sailing lessons?	4	Pending	9 hours ago	Graham Murphy	5	
	Carlos Carballo	more info	Qué debo llevar en mi viaje?	1	New	2 months ago	Diego Garcia	4	
	Nolan Gett	lodging	Hotels in Cabo San Lucas?	2	Open	1 month ago	Hospitality Team	2	
	Quinton Sary		My family and I are planning a vacation and ...	1	Open	1 week ago	Matt Trifiro	6	

Ilustración 9. Priorización de tareas mediante Desk.com

Se pueden crear reglas de negocio para asignar las tareas a personas o para avisar a ciertas personas que la tarea se ha cerrado.

Se pueden crear foros, discusiones sobre un tema, dar de alta FAQs para que el cliente pueda consultar cualquier duda en el momento que crea oportuno, es decir, crear una base de conocimiento de acceso rápido como se muestra en la [Ilustración 10](#).

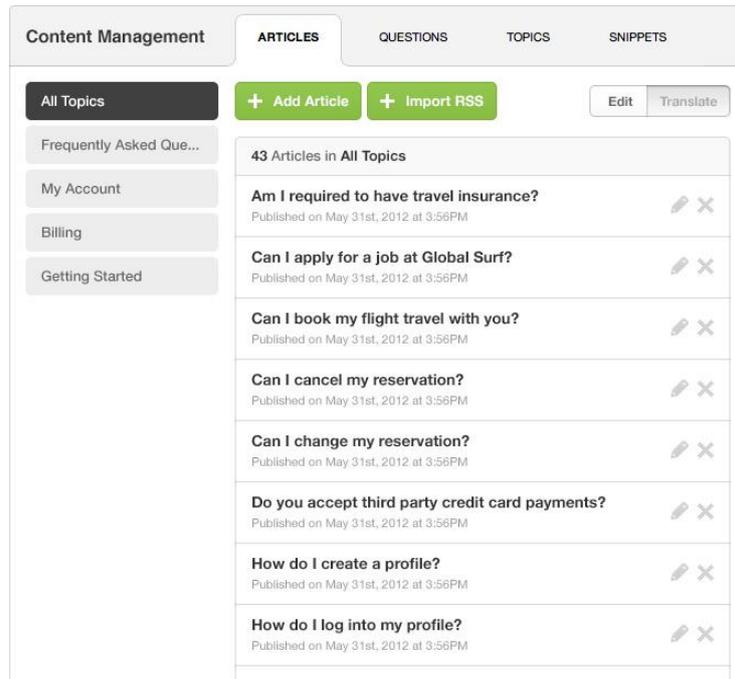


Ilustración 10. Creación de foros y preguntas

Además, se puede crear una base de datos del conocimiento a partir de la escritura de artículos del equipo de Service Desk, de forma, que todo el equipo tenga acceso a estos artículos. Se pueden organizar los artículos por temas como se ve en la [Ilustración 11](#).

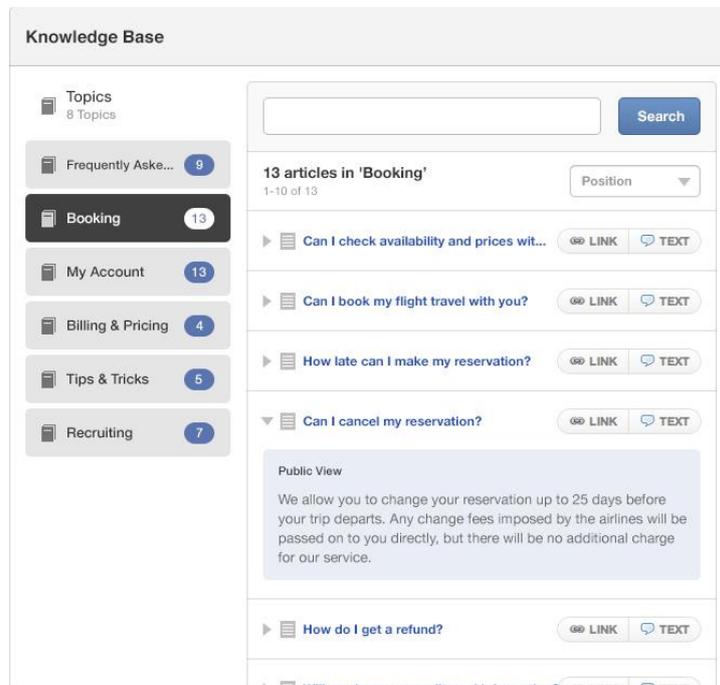


Ilustración 11. Organización de la base de datos del conocimiento

4.2.3 Tender

Tender es otra herramienta de gestión de Service Desk, se organiza a partir de una cuenta de correo, si el cliente tiene web, se puede insertar un widget para que pueda dar de alta incidencias desde su web como se muestra en la **Ilustración 12**.

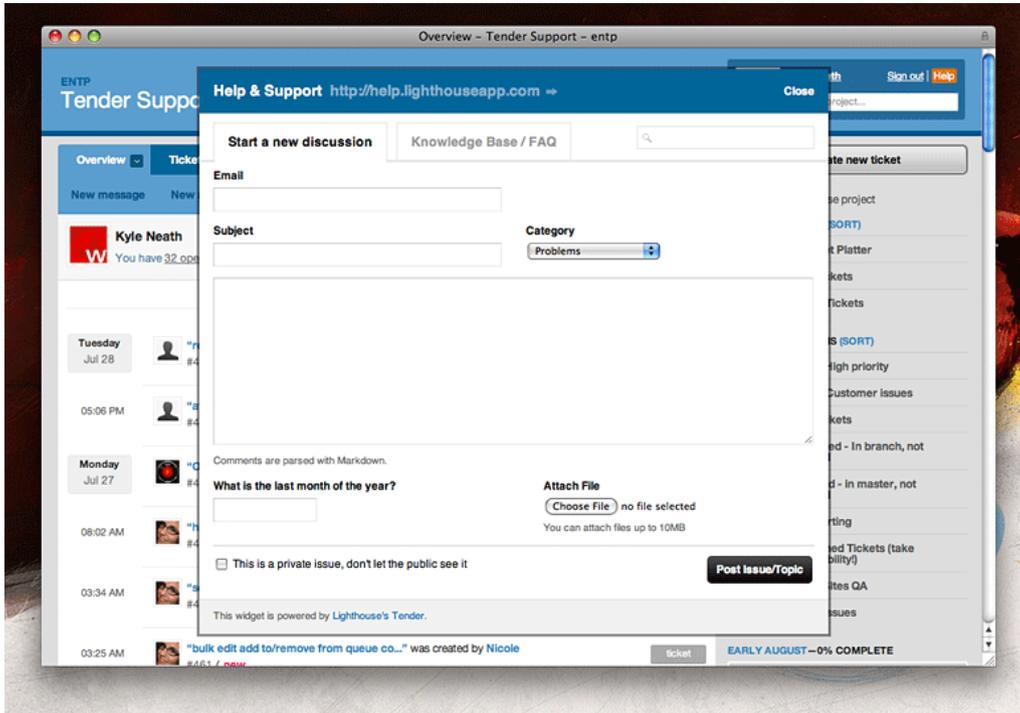


Ilustración 12. Widget incrustado en la web del cliente

Los tickets se pueden agrupar en categorías y se pueden organizar dependiendo del estado de la tarea como se muestra en la **Ilustración 13**.

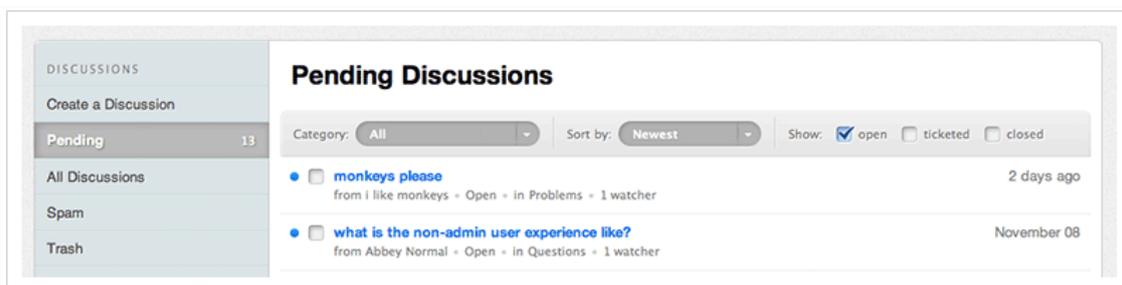


Ilustración 13. Agrupación de tickets Tender

Se puede crear una zona de ayuda al usuario con las preguntas más frecuentes, publicación de discusiones y las FAQs más destacadas. Esto se puede observar en la **Ilustración 14**.

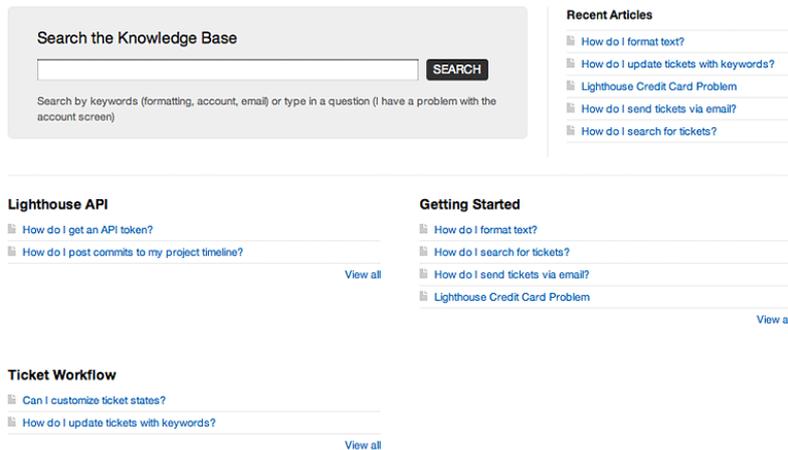


Ilustración 14. Base de datos del conocimiento

4.2.4 ServiceTonic

ServiceTonic es una herramienta de gestión de Service Desk, puede crear tickets a partir de emails enviados por el cliente.

Una vez se crea el ticket, se puede hacer un seguimiento mediante la gestión de estados de la petición, además de hacer una priorización de la misma como se muestra en la Ilustración 15.

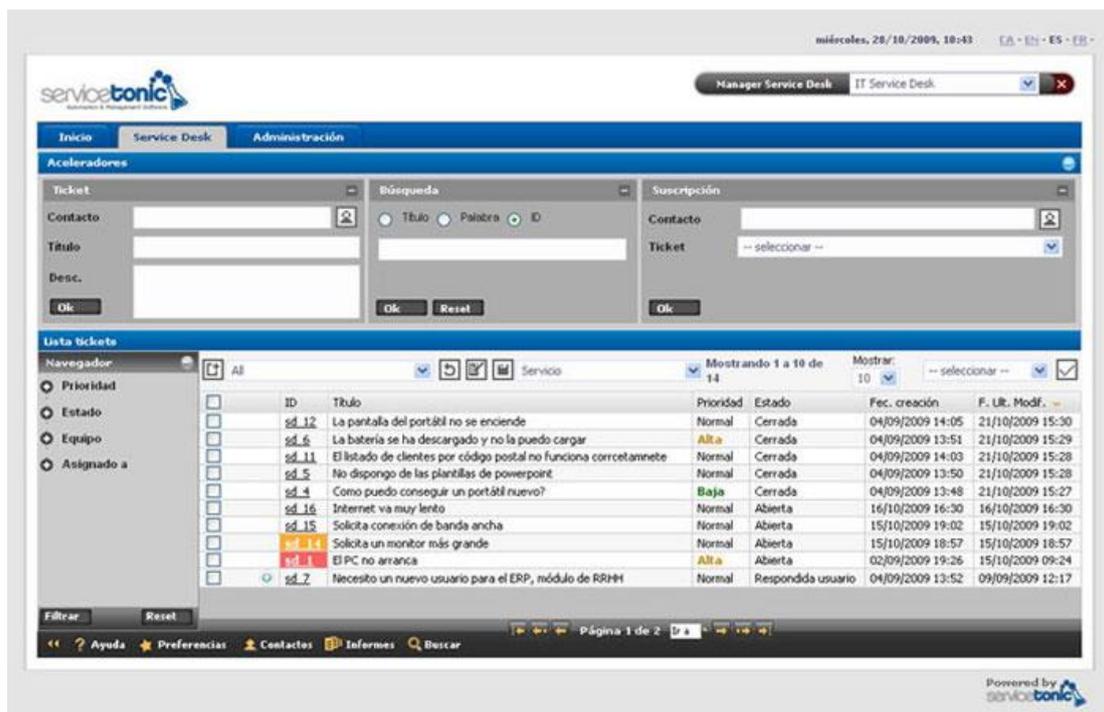


Ilustración 15. Gestión de incidencias

Se pueden definir reglas automatizadas para la asignación de trabajo y notificaciones como se muestra en la Ilustración 16.

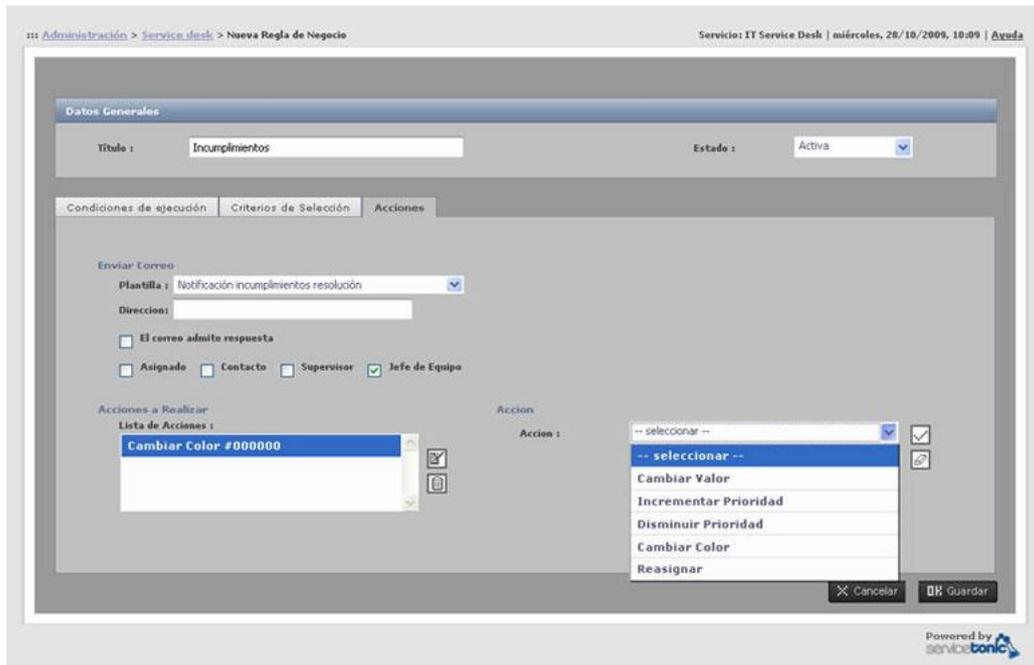


Ilustración 16. Gestión de reglas de negocio

Se puede integrar la base de datos del conocimiento con la gestión de incidencias, de forma que tanto el cliente como el equipo de Service Desk, pueda buscar, de una forma sencilla, las dudas que le surjan. La Ilustración 17 muestra la base de datos del conocimiento.

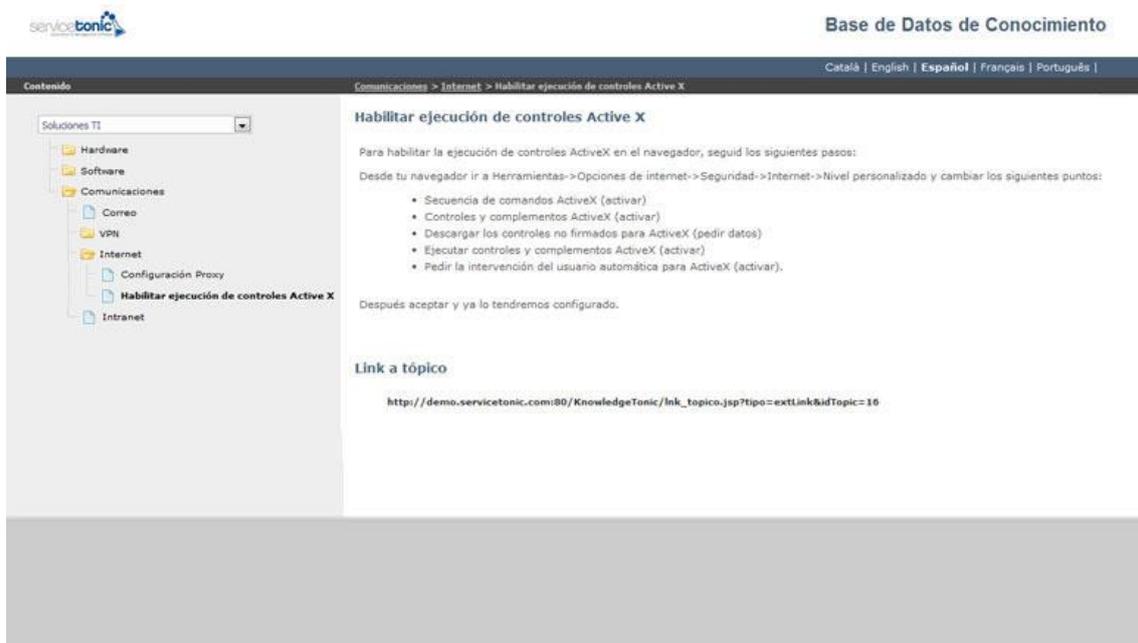


Ilustración 17. Base de datos del conocimiento

Se pueden vincular las incidencias con los problemas y con la base de datos del conocimiento, de forma que al encontrar una solución temporal para un problema, ésta se traspasará a los tickets vinculados. Al cerrar un problema, se notificará a los tickets vinculados traspasando la solución.

4.2.5 Resumen

El resumen del estudio realizado es el que se muestra en el siguiente cuadro resumen.

Los símbolos tienen el siguiente significado:

- ✓ → La aplicación lo permite.
- ✗ → La aplicación no lo permite.
- ! → No está completo en la aplicación.

	Service Desk			
	Zendesk	Desk.com	Tender	ServiceTonic
Gestionar la duplicidad de incidencias	✓	✓	✓	✓
Identificación del cliente que ha reportado la incidencia	✓	✓	✓	✓
Base de datos del conocimiento	✗	!	!	✓
Identificación del ítem de desarrollo relacionado con el ítem de service desk	✗	✗	✗	✗
Conocimiento de la versión que se va a entregar la solución del problema	✗	✗	✗	✗
Priorización de las incidencias	✓	✓	✓	✓
Gestión de Workflows	!	!	!	✓
Integración de la estructura del producto	✗	✗	✗	✗

Ilustración 18. Comparativa de herramientas de Service Desk

4.3 Estudio de herramientas de gestión de proyectos

4.3.1 Jira

Jira es una herramienta de gestión de proyectos que permite planificar las tareas, priorizarlas y construir la solución deseada (ver [Ilustración 19](#)).

Es una de las herramientas de gestión más extendidas entre las empresas, muchos equipos la utilizan para capturar, asignar y supervisar el trabajo: desde errores a nuevas funcionalidades, historias y requerimientos, hasta tareas o peticiones.

JIRA viene configurada con unos tipos de tareas, también se pueden definir tipos de tareas propios y campos para gestionar la información más importante para el equipo.

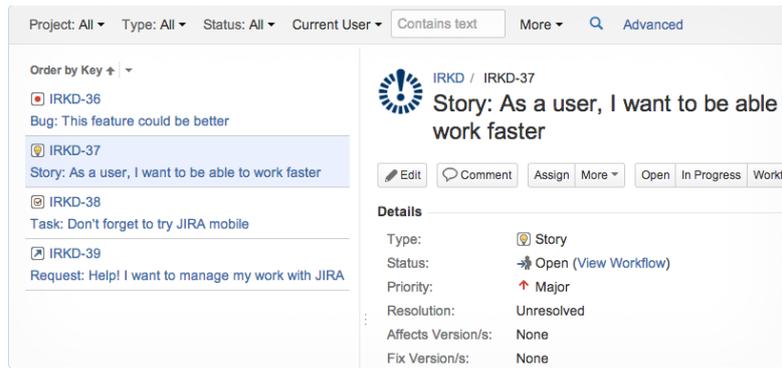


Ilustración 19. Gestión de incidencias en Jira

Se pueden crear workflows diferentes dependiendo del ítem de trabajo, estos workflows son manuales y se pueden adaptar al funcionamiento de cualquier empresa, en la Ilustración 20 se muestra un ejemplo de workflow definido en Jira.

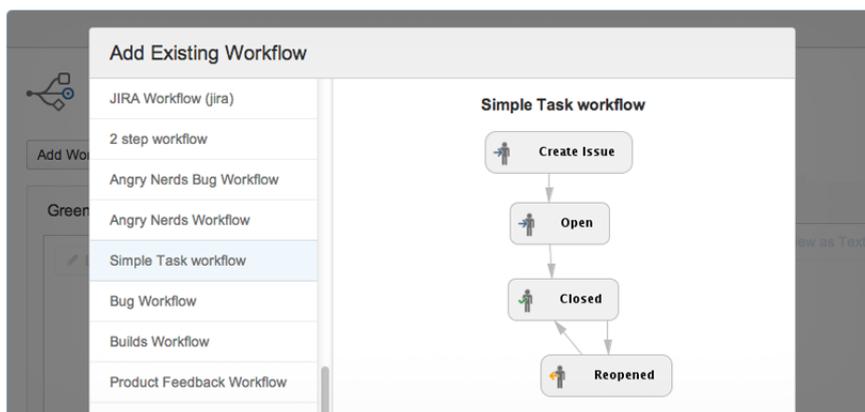


Ilustración 20. Gestión de workflows en Jira

En la Ilustración 21 se muestra la organización del trabajo mediante la priorización de tareas.

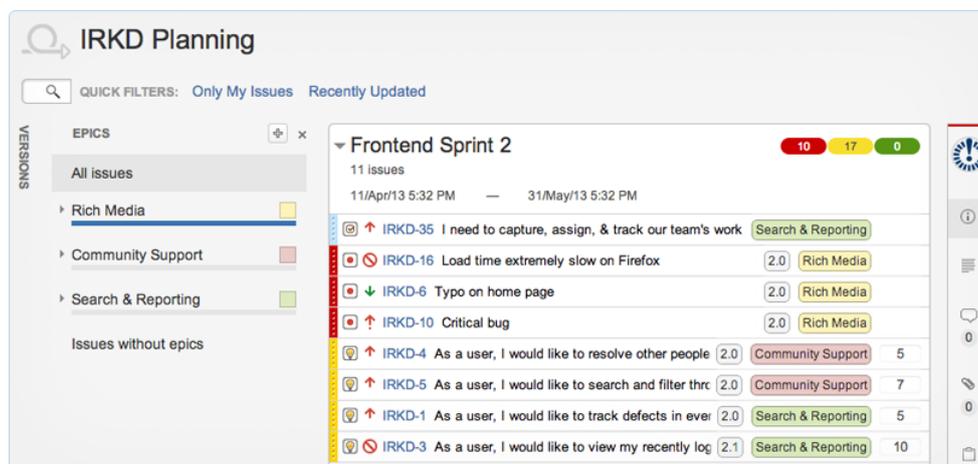


Ilustración 21. Priorización de tareas en Jira

Jira ofrece plugins para la gestión de proyectos ágiles, como es GreenHopper (ver Ilustración 22) que permite la planificación de iteraciones (Scrum) y gestionar su flujo (Lean/Kanban).

Se puede planificar y gestionar el backlog de cada iteración de sus proyectos, así como organizar y priorizar las tareas simplemente arrastrándolas hacia arriba o abajo en el backlog.

The screenshot shows the Jira interface for a project named 'GreenHopper Team Scrum'. The top navigation bar includes 'Dashboards', 'Projects', 'Issues', and 'Agile'. The main content area is divided into two sections. On the left is the 'Backlog' view, which displays a list of issues with their titles and estimates. The issues are: GHS-5077 (3), GHS-5286 (5), GHS-5285 (5), GHS-5107 (5), GHS-5085 (8), and GHS-5087 (5). Below the backlog is a 'Sprint Marker' section showing 6 issues and a total estimate of 21. On the right is the 'Details' view for issue GHS-5077. The description reads: 'Currently GH does some things to protect itself in upgrade tasks but not enough. We need to track' followed by a bulleted list: 'when a upgrade tasks runs', 'what upgrade task', 'what version of JIRA at the time', and 'what version of GH at the time'. Below the list, it says 'and a page to reveal this to admins'.

Ilustración 22. Gestión de backlog en Jira con GreenHopper

Permite aplicar los principios básicos de Kanban al proceso Scrum y mejorar el flujo de historias a través del tablero como se ve en la Ilustración 23.



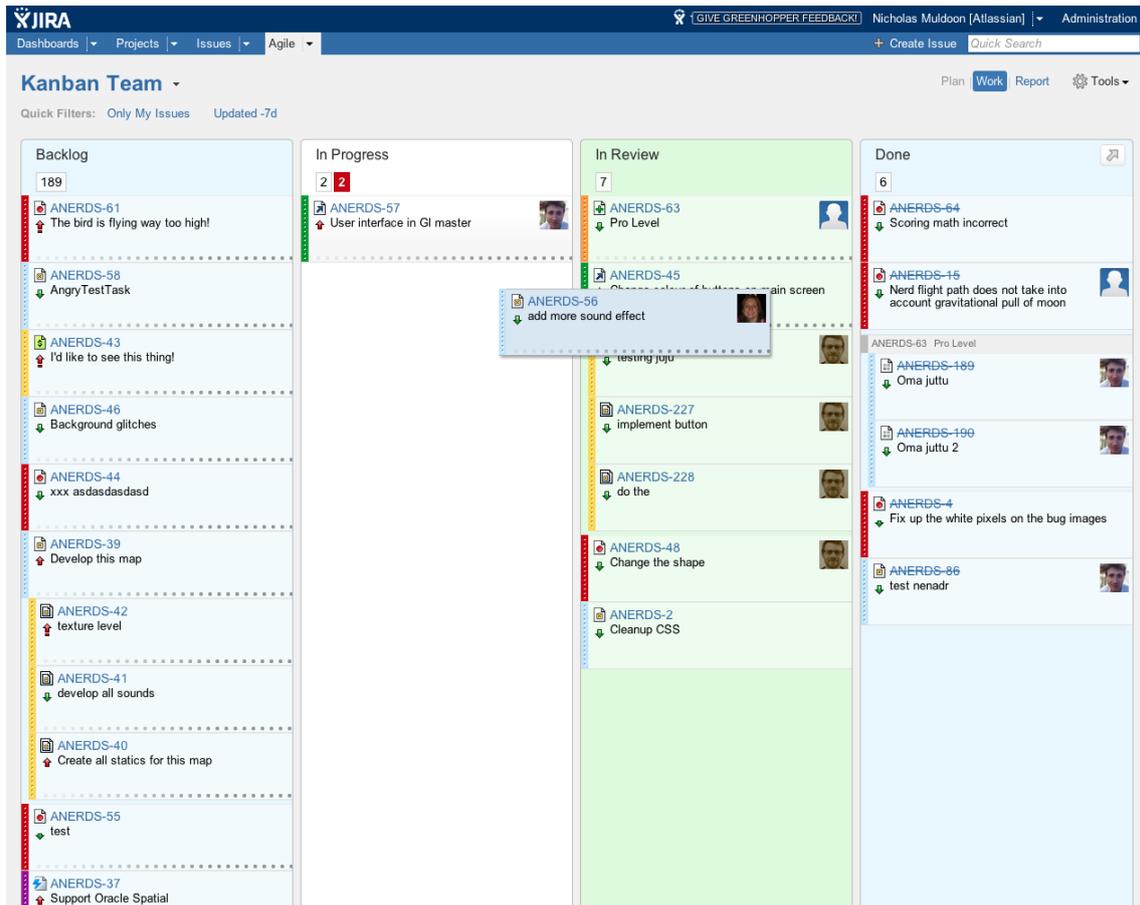


Ilustración 23. Uso de Kanban + Scrum

4.3.2 Ontime

Ontime es una herramienta de gestión de proyectos de metodología ágil, que permite gestionar el backlog y gestionar los sprints, en la Ilustración 24 se muestra un ejemplo de gestión del backlog.

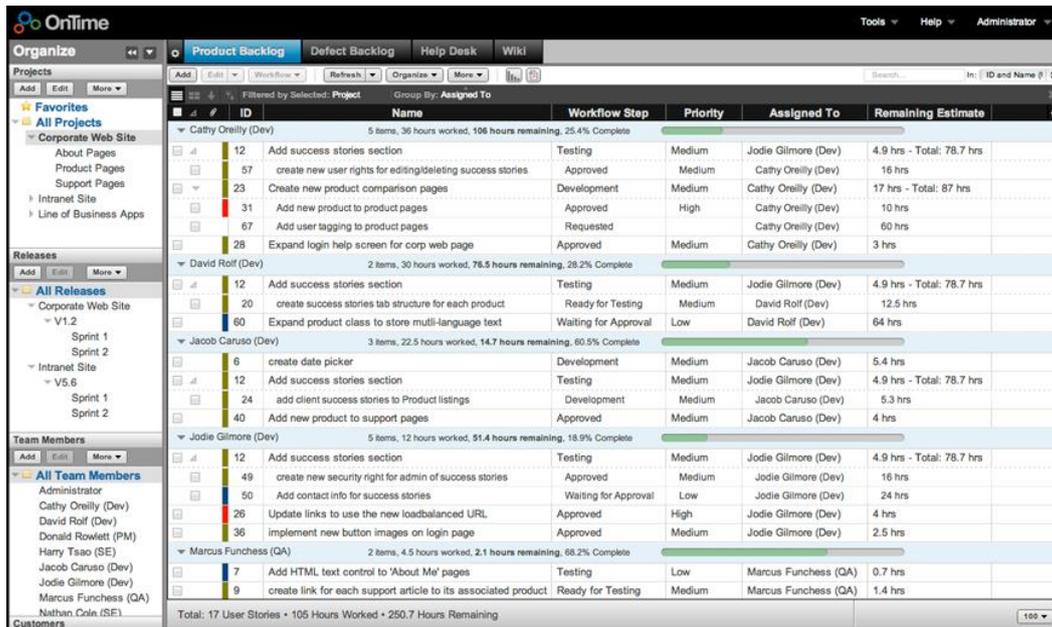


Ilustración 24. Gestión del backlog en OnTime

Tiene una organización del proyecto en árbol y las tareas se asocian a cada una de las ramas del árbol.

Se pueden gestionar las tareas por prioridad, asignar a personas del equipo y gestionar el ciclo de vida de las mismas como se muestra en la Ilustración 25.

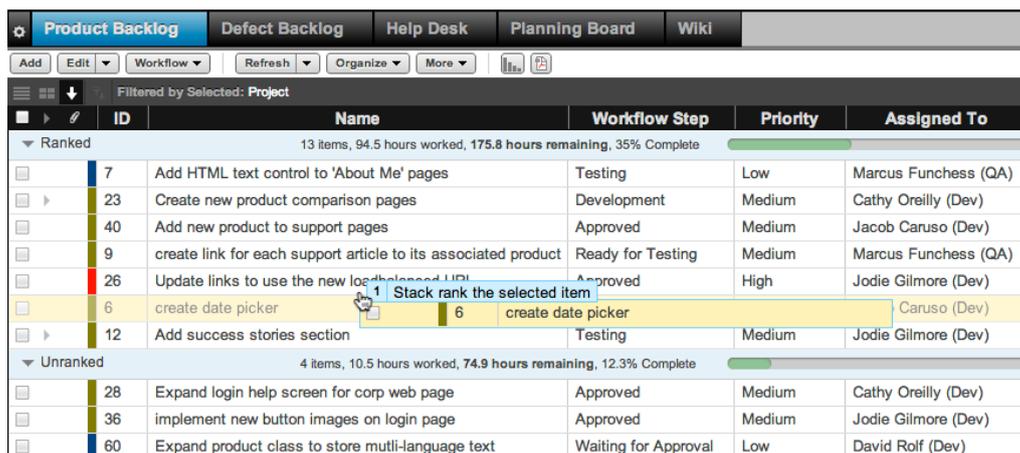


Ilustración 25. Gestión de tareas en OnTime

Se pueden gestionar Workflows manualmente.

Existe una integración con la herramienta OnTime Help Desk, pero no permite enlazar tareas de Service Desk con tareas de desarrollo.



4.3.3 Version One

Es un gestor de proyectos ágiles que permite al equipo llevar a cabo un ciclo de vida ágil, como Scrum, Kanban, Lean, XP e híbridos. Se organizan las tareas en un Kanban como se muestra en la [Ilustración 26](#).

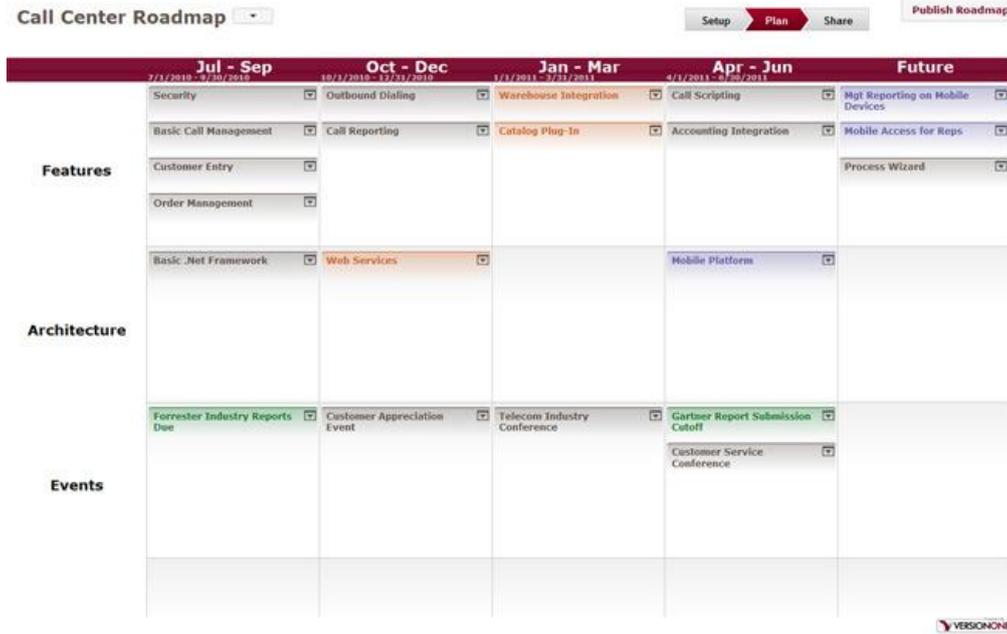


Ilustración 26. Organización de tareas en Version One

Se puede gestionar el backlog, planificar sprints y gestionar los workflows adaptándolos al equipo.

Tiene una gestión de ideas dirigida al cliente, de forma que el cliente puede comentar sobre alguna funcionalidad, colaborar o priorizar la misma. No es una gestión de Help Desk, es una opción que se le da al cliente para que pueda opinar sobre las tareas que se están realizando o incluir alguna idea nueva que necesite. La [Ilustración 27](#) muestra como vería el cliente esta funcionalidad:.

The screenshot shows a web interface for an idea titled "Customer Calendar". At the top left, it indicates "Votes: 5" with a "Vote for" button. The author is "Ferris H. | about a month ago | Call Center". A red sidebar on the right contains the text "Cast your vote or comment ..." and a "return to idea list" link. The main content area contains the idea's description: "It would be really nifty if Customers had an online calendar where they could enter expected call volume. In the existing system, I have to send an email to let you know when we're running a sale or promotion, or when our office is closed. Sometimes that information conflicts with previous information, and we have to call to sort out the schedule. With an online Calendar, I can update our resource needs as our schedule changes." Below the description is a "Comments" section with two entries: "I don't understand. Can you give an example of how this might work?" by John H. and "Outsourcing seems the way to go here. Any ROI calculations yet?" by Daphne B. There is a text input field and a "Submit" button at the bottom of the comments section.

Ilustración 27. Interacción con el usuario

4.3.4 Team concert

Rational Team concert es la herramienta de IBM que se utiliza para gestionar proyecto ágiles. Permite organizar el trabajo, priorizarlo y asignarlo a los recursos, como se muestra en la Ilustración 28.

The screenshot displays a "Sprint Backlog" interface. At the top, it shows "14 items: 10 open, 4 closed | Ends in: 8 days". Below this is a "Plan Details" section with an "Edit" button. The main area is divided into columns for "Planned Items", "Links", "Snapshots", "Dashboard", and "Notes". A "View As" dropdown is set to "Taskboard". The backlog is organized by user, with sections for "Bob Smith" and "Jane Williams". Each user's section has columns for "Story", "Open", "In Progress", and "Resolved". Bob Smith's items include "As a user, I want a calendar view", "Visual design elements for calendar UI" (2 days), "Calendar widget design" (2 days), and "Competitive analysis". Jane Williams' items include "As a user, I want to schedule", "Calendar: database development", "Develop calendar widget UI", and "Confirm permissions".

Ilustración 28. Gestión del backlog en Team Concert.

Permite gestionar el backlog del producto y cada uno de los sprints, así como los principales actores.

Se puede evaluar el riesgo de un sprint, así como planificar el trabajo diario de cada persona del equipo.

Se puede configurar el proceso de trabajo, así como definir los diferentes roles para adecuar la herramienta al trabajo del equipo.

4.3.5 TUNE-UP Process

En este apartado se resume TUNE-UP Process con información extraída de la [Referencia 7](#) [Análisis de Impacto de Requisitos en un proceso de desarrollo centrado en Pruebas de Aceptación].

TUNE-UP Process es el nombre de la herramienta de apoyo para la aplicación efectiva de la metodología TUNE-UP. La herramienta está formada por cuatro módulos principales: Personal Planner (PEP), Gestor de UTs, Version Contents & Tracking (VCT) y Requirements Manager (REM). A continuación, se describe brevemente cada uno de estos módulos.

Personal Planner (PEP)

Es el punto de entrada al espacio de trabajo del agente. Este trabajo corresponde a las actividades que tenga asignadas en las UTs no terminadas. En el Kanban el agente puede determinar fácilmente en el PEP qué actividades tiene pendientes o en progreso (activas o pausadas). Además, puede distinguir aquellas actividades que tienen pendientes o en proceso otros miembros del equipo y que, o bien estarían por llegarle, o ya ha finalizado, y han continuado en actividades posteriores en el workflow.

El PEP ofrece una variedad de facilidades de filtrado y ordenamiento de información para que el agente pueda determinar las prioridades de su trabajo y realizar una correcta elección.

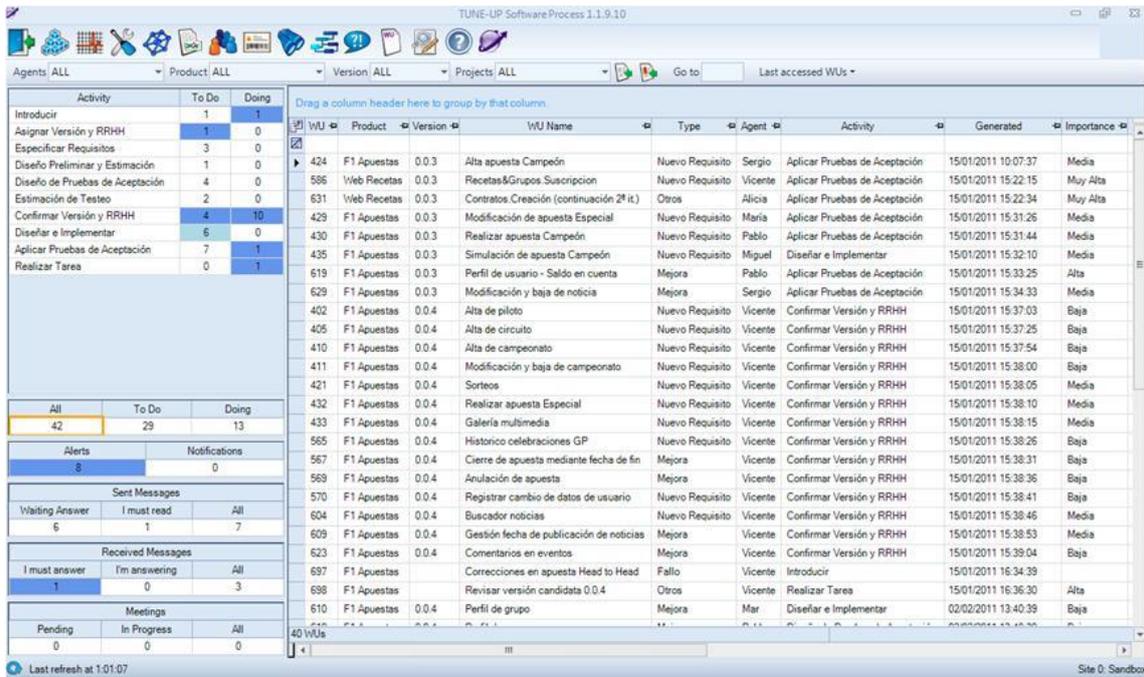


Ilustración 29. Personal Planner (PEP)

Work unit manager

Cuando el agente decide la UT y la actividad en la que desea trabajar, accede con ella al Gestor de UTs.

En la ficha de la parte superior del gestor de UTs se muestran los datos generales de la UT, y en la parte inferior, un conjunto de pestañas con información más específica (ver Ilustración 30).

En la pestaña Tracking se muestra la lista de actividades del workflow por las que ha pasado la unidad de trabajo. Cada actividad está asociada a un registro de seguimiento que incluye: la actividad, el agente que la desarrolla, el estado en el que se encuentra, etc. Además, en esta pestaña, con los botones Record, Pause y Finish, el agente puede controlar el registro de tiempos, activando, pausando o finalizando la actividad.

Al finalizar una actividad, la UT continúa su workflow o puede a petición del usuario dar un salto a otra actividad del workflow. Otras facilidades para gestionar una UT y que se encuentran en otras pestañas del gestor de UTs son: documentación, mensajes para comunicarse con otros agentes en el contexto de la UT, reuniones para registrar tiempo y acuerdos alcanzados, relaciones con otras UTs, detalle de los tiempos registrados y estimados para cada actividad, e información de las partes del producto que son afectadas por la UT.



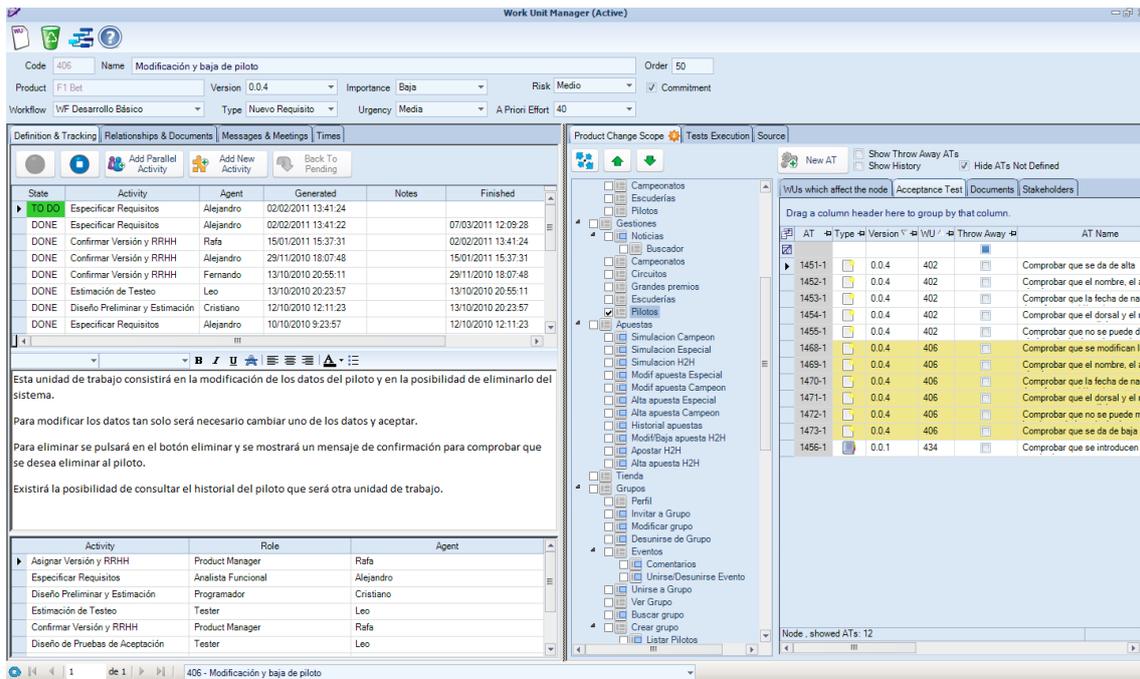


Ilustración 30. Gestor de UTs

Requirements manager (REM)

El REM es una de las innovaciones que ofrece TUNE-UP en cuanto a gestión del producto y sus requisitos. Los requisitos del sistema se definen en forma de árbol, al seleccionar un nodo, se mostrarán todas las UTs que están relacionadas con ese nodo (ver Ilustración 31).

Finalmente cada nodo en la pestaña 'Stakeholders' puede tener especificados los stakeholders para los cuales dicho nodo es relevante.

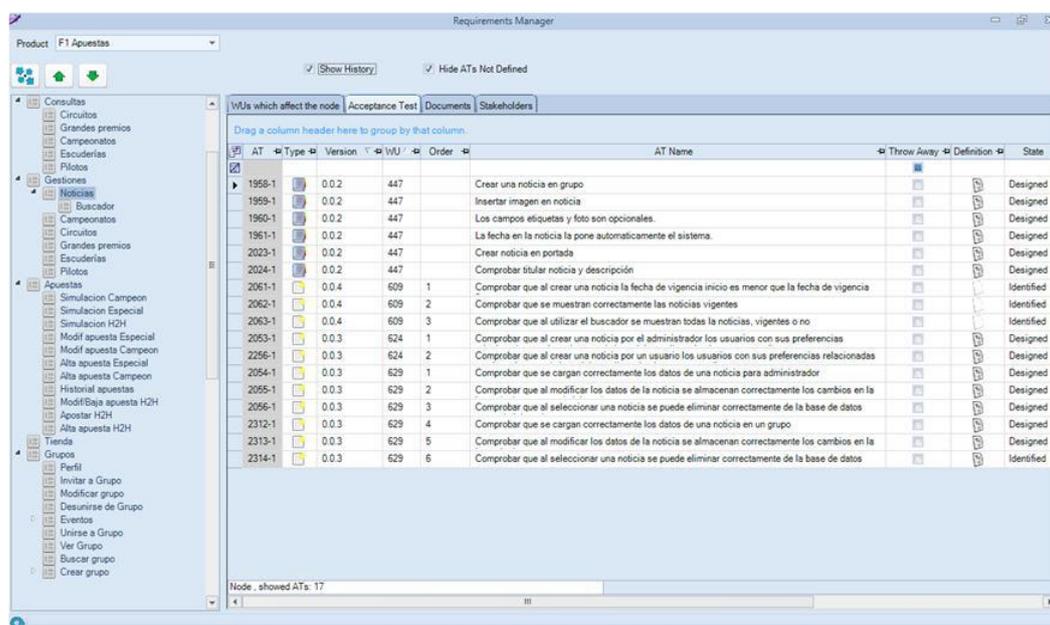


Ilustración 31. Requeriments Manager (REM)

Version contents and tracking (VCT)

En este módulo se gestiona la planificación y el seguimiento de las versiones. En la [Ilustración 32](#) se muestra el contenido de la pestaña 'Agent Workload', en ella se puede conocer en cualquier momento la holgura simple de los agentes respecto de sus actividades en una versión del producto.

Ilustración 32. Version contents & tracking

4.3.6 Resumen

Este es el cuadro resumen del estudio realizado de las cinco herramientas:

	Desarrollo				
	Jira	Ontime	Version One	Team concert	TuneUp
Trabajo pendiente ordenado por prioridad	✓	✓	✓	✓	✓
Identificar en qué partes del software tiene más problemas el usuario	✓	✓	✓	✓	✓
Base de datos de conocimiento	!	!	!	!	!
Gestión de Workflows	!	!	!	!	✓
Priorización de incidencias	✓	✓	✓	✓	✓
Integración de la estructura del producto	✗	✗	✗	✗	✓

Ilustración 33. Comparativa de herramientas de desarrollo

Como se puede ver en la [Ilustración 33](#), en todas ellas se puede asignar una priorización de tareas y organizar el trabajo en base a esta priorización.



La base de datos del conocimiento se puede configurar con todas las herramientas pero de una forma difícil de gestionar y de visualizar.

Con lo que respecta a la integración de la estructura del producto con las tareas que realiza el equipo, solo lo tiene TUNE-UP Process.

Analizando los estudios realizados, se pueden sacar las siguientes conclusiones:

- En ninguna herramienta hay una gestión conjunta del trabajo del Service Desk con el de desarrollo.
- Solo en Tune Up se tiene una visión en árbol de la estructura del producto, facilitando así ver la información de cada módulo funcional con el trabajo asociado.
- Las herramientas de Service Desk están centradas solo en la gestión del Service Desk, no hacen referencia a la gestión de desarrollo a no ser que se integre con alguna herramienta de gestión del desarrollo.
- Ninguna de ellas tiene una gestión de la base de datos del conocimiento tal y como ITIL lo define.
- Todas tienen gestión de workflows pero el avance en el workflow se hace manual, solo Tune Up tiene una gestión de workflows automática.

Para tener la integración entre el equipo de desarrollo y el equipo de Service Desk, se necesita adaptar una herramienta de desarrollo al flujo de trabajo de Service Desk ya que las herramientas de Service Desk son demasiado sencillas para adaptarlas al flujo de trabajo de desarrollo.

Por esta razón, en los próximos capítulos se va a definir una metodología de trabajo para unir los dos equipos y, además, se hará una simulación en la herramienta TUNE-UP Process.

5. Integración de Customer Portal, Service Desk y Desarrollo

5.1 Introducción

En este apartado se va a plantear una propuesta de integración de todos los componentes, de forma que se pueda ver cuál sería el flujo completo en cuanto a la gestión de incidencias para un producto de desarrollo software.

Habrá que tener en cuenta lo expuesto en los planteamientos anteriores sobre las funcionalidades del Service Desk con respecto al equipo de desarrollo y la relación existente entre el Customer Portal y el Service Desk. Se planteará la solución usando como base el Service Desk.

La comunicación entre el cliente, el Service Desk y desarrollo es vital para garantizar el éxito del proyecto ya que entre los tres equipos fluye mucha información. Esta información será captada en primera instancia por el Customer Portal a través de la información introducida por el usuario. A partir de esta información, en segunda instancia, el personal de Service Desk la cataloga, le asigna una prioridad, importancia y riesgo. Además, crea una dependencia entre los tickets y los nodos del producto afectados. Finalmente, resolverá la incidencia, o la escalará al nivel de desarrollo si procede.

Por su parte, desarrollo puede obtener la información del cliente final, mediante la consulta de los ítems que se han dado de alta en el Customer Portal y que posteriormente, los miembros del equipo de soporte han revisado.

De la misma forma, el usuario puede realizar un seguimiento del ticket desde que lo da de alta hasta que éste se cierra.

5.2 Estructura del producto de desarrollo.

La estructura del producto es una forma de organizar sus requisitos funcionales. Ésta se puede representar mediante un grafo acíclico cuyos nodos son funcionalidades del sistema.

Cuando un miembro del equipo de Service Desk supervisa un ticket creado por un usuario, éste creará una dependencia entre el ticket y uno (o varios) de los nodos de la estructura del producto.

Por lo tanto, es muy importante que la herramienta de desarrollo y el Service Desk compartan la misma estructura de producto.

Un ejemplo de estructuración de un producto software sería el mostrado en la siguiente figura.

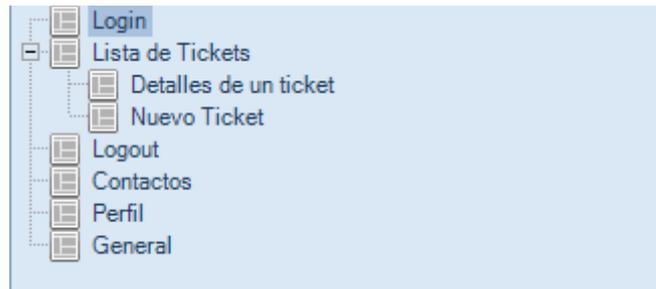


Ilustración 34. Nodos de un producto

A cada nodo, se asociarían las distintas unidades de trabajo (bien sean tickets o unidades de trabajo de desarrollo) pertinentes.

5.3 Definición de unidades de trabajo

En este caso, no trataremos de seguir ninguna metodología de desarrollo en concreto, debido a que la herramienta en que se basa este proyecto, permite la parametrización para cada proyecto que puede estar basado en diferentes metodologías.

Las unidades de trabajo, también conocidas como historias de usuario, reflejan cada uno de los ítems en los que un equipo, en un producto determinado debe trabajar.

Cada una de las unidades de trabajo debe estar tipificada, según la parametrización concreta de cada proyecto. No obstante, existe una clasificación mínima de unidades de trabajo.

5.3.1 Unidades de trabajo de desarrollo

El Service Desk, debe parametrizar para cada proyecto los tipos de unidades de trabajo. Para facilitar el trabajo y permitir una puesta en producción de la herramienta desde el momento cero, se ha definido una clasificación mínima de unidades de trabajo basada en ITIL para proyectos de desarrollo:

- **Nuevo requisito:** Es una funcionalidad que no existe en el producto que requiere una nueva implementación.
- **Mejora:** Implica una acción sobre el producto existente para cambiar el funcionamiento.
- **Corrección del fallo.** Un fallo es un comportamiento no deseado del producto, visible desde el punto de vista externo de éste, no detectada en la fase de desarrollo. Por ejemplo, se presenta un fallo cuando el usuario o el equipo de testeo realiza una acción y aparece un mensaje de excepción no manejada o un resultado no deseado.

Dentro de la corrección del fallo, una vez desarrollo lo analiza, éste se puede catalogar de dos formas distintas:

- **Defecto:** es la imperfección que causa el fallo. El defecto puede ser visible o no en el exterior del producto, es decir, puede tratarse de una anomalía interna del producto. Un defecto puede existir en el producto y no manifestarse como fallo.

Por ejemplo, es un defecto tener una variable no inicializada, lo cual puede causar un fallo.

- Error: es lo que causó la introducción del defecto en el producto. El error siempre es cometido directamente o indirectamente por una acción o no acción humana. Por ejemplo, es un error no realizar la inicialización de una variable en el mismo instante de su declaración.

La definición en un diagrama de clases sería la siguiente:

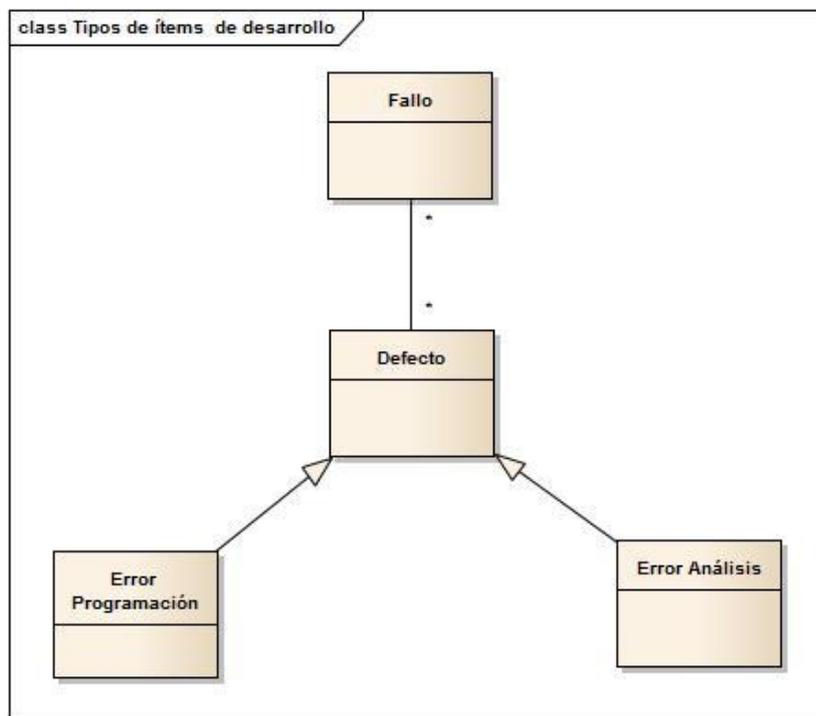


Ilustración 35. Diagrama de clases fallo

5.3.2 Unidades de trabajo del Service Desk

Como se ha comentado en el apartado de desarrollo, el Service Desk debe parametrizar para cada proyecto los tipos de unidades de trabajo. Para facilitar el trabajo y permitir una puesta en producción de la herramienta desde el momento cero, se ha definido una clasificación mínima de unidades de trabajo basada en ITIL para proyectos de soporte:

- Incidencia: Una incidencia puede descubrir un problema pero el incidente no se convierte en un problema directamente ya que se presenta cuando se desconoce las causas que provocan la interrupción o la desviación de la operativa normal.
- Problema: es la causa desconocida de uno o más eventos que provocan la interrupción o la desviación de la operativa normal, una vez conocidas las causas, se convierte en un error conocido.
- Error: Puede dar lugar a tres situaciones.
 - Solución conocida por haberse producido anteriormente y estar documentada.
 - Solución temporal que permite restaurar el servicio a sabiendas de que no es la solución definitiva.
 - Desconocimiento de la solución, pasando a ser un problema.

- Petición de nuevo requisito (RFC – Request for Change): La solicitud no estaba contemplada en los requisitos y modifica la estructura del producto.

La representación en forma de un diagrama de clases sería la siguiente:

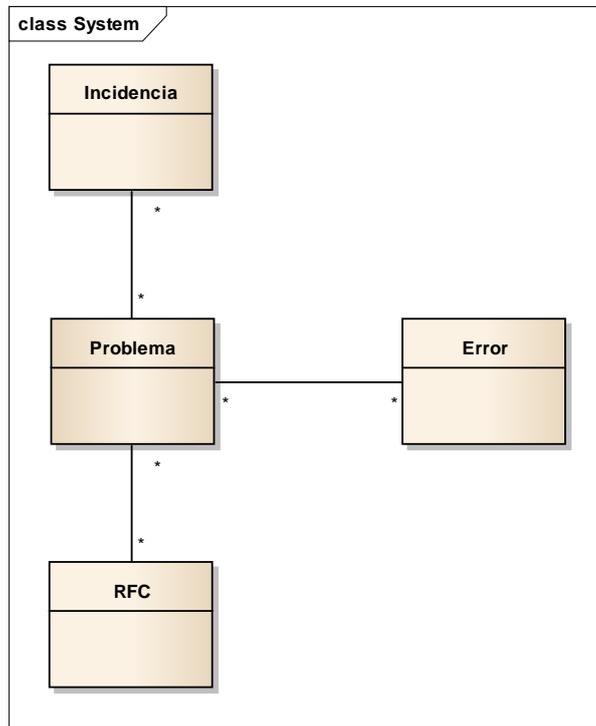


Ilustración 36. Diagrama de clases incidencia

5.3.3 Unidades de trabajo del Customer Portal. Tickets.

En el caso del Customer Portal, cada uno de los ítems de trabajo son denominados tickets.

Estos tickets consisten en la creación de una incidencia y deben ser definidos de acuerdo a la siguiente tabla:

NOMBRE	El ticket tendrá un nombre breve, conciso y que determine el asunto de la incidencia.
DESCRIPCIÓN	En este apartado se explica, de forma más extensa, el motivo de la incidencia. En caso de ser problemas con el producto, se deberá aportar la información necesaria para que los miembros del Service Desk sean capaces de entenderla, así como contextualizarlo, es decir, comentar los datos introducidos por el usuario al realizar la acción causante del problema, así como el resultado devuelto.
ADJUNTOS	Para apoyar la descripción de la incidencia.

Seguidamente, cada ticket se categorizará en primera instancia por el usuario, siendo revisado seguidamente por el técnico de soporte para que finalmente, pueda crear la unidad de trabajo pertinente en desarrollo si es necesario.

Además, cada ticket contendrá los siguientes datos:

- **Identificación:** Número clave para la localización de un ticket en un momento determinado.
- **Fecha de apertura:** Fecha en que el ticket fue creado.
- **Fecha de cierre (si está cerrado):** Fecha en que el ticket fue resuelto.
- **Producto:** Proyecto de soporte sobre el cual se reporta la incidencia.
- **Solicitante:** Usuario solicitante del ticket.
- **Cliente:** Cliente al que el solicitante pertenece.

5.3.4 Protocolo de Integración

La fuente de datos para el Service Desk y para desarrollo es completamente independiente, las unidades de trabajo de cada una son diferentes, y no tienen relación entre ellas.

Sin embargo, como se ha comentado, la herramienta de gestión de incidencias y desarrollo comparten la misma estructura de proyecto, por lo que ambas tendrán una dependencia directa con el nodo o nodos correspondientes del proyecto de desarrollo. Es importante saber que una unidad de trabajo en desarrollo puede estar relacionada con ninguna (si pertenece a los requisitos iniciales y es creada desde desarrollo), a una UT del Service Desk (si se ha reportado alguna incidencia) o a varias (si se han reportado más de una cuyo problema es el mismo).

Por otro lado, los tickets del Customer Portal no son más que una visión filtrada de las unidades de trabajo del Service Desk, que permiten al cliente ver el estado de sus tickets haciendo la gestión de la organización transparente para él.

Los tickets pueden ser creados desde el Customer Portal o desde el propio Service Desk.

En el primer caso, la integración de ambas herramientas creará directamente la unidad de trabajo en el Service Desk, permitiendo al usuario su inmediata visualización, a pesar de no haber sido supervisada por un técnico de soporte aún, quien supervisará y modificará los datos de ésta (si conviene), viendo el cliente los cambios producidos en su lista de tickets.

En el segundo, los tickets pueden ser creados desde el Service Desk (por ejemplo, si un usuario hace una llamada telefónica por un problema que se le ha presentado). No obstante, se requerirá una identificación del usuario solicitante, para que pueda visualizar el estado del ticket en su portal como si se tratara de un ticket que ha creado a través de él.

Finalmente, desde el Service Desk se solucionará la incidencia pasando a una actividad terminal, por lo que el ticket quedará cerrado, o se creará una UT en desarrollo para que el problema sea solucionado siguiendo el flujo de trabajo que se esté utilizando en ese contexto (no necesariamente se creará en desarrollo esta UT si ya existe); una vez la unidad de trabajo quede finalizada en desarrollo, el técnico de soporte dará por finalizadas las UTs relativas a dichas incidencias, por lo que los tickets quedarán cerrados, completando de esta forma el proceso que sigue cualquier incidencia.



Teniendo en cuenta lo expuesto en el [apartado 2.3](#), en el [apartado 3.3](#) sobre integración y todo lo expuesto en este, el diagrama propuesto resultante quedaría del siguiente modo.

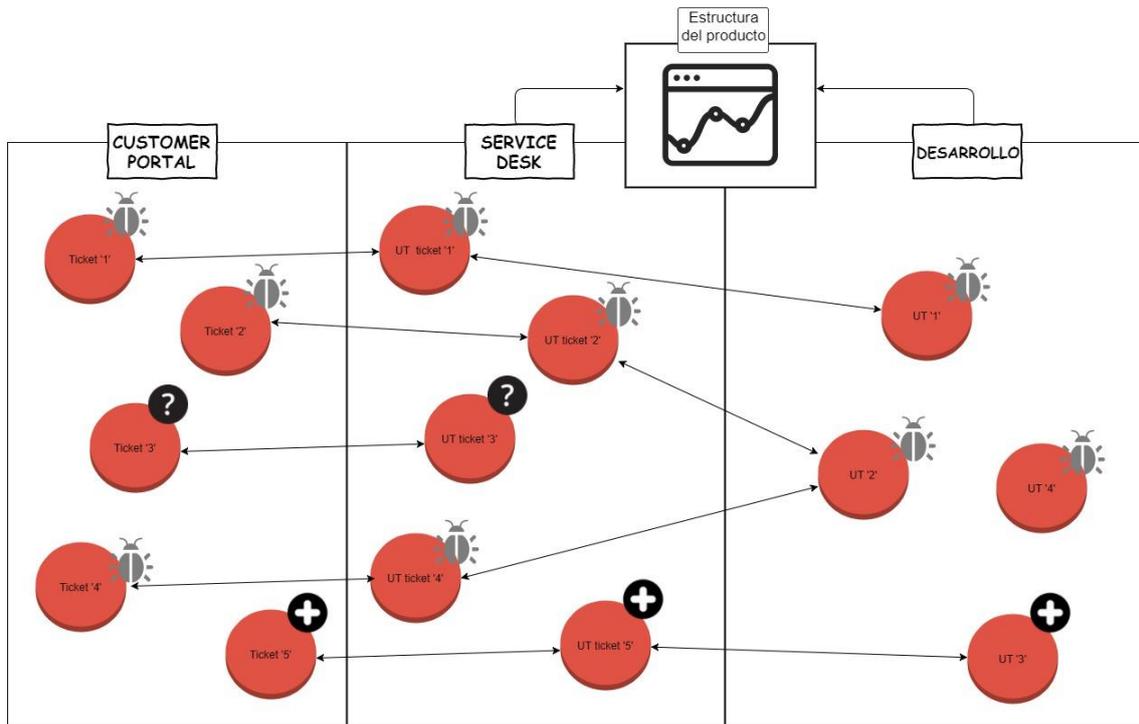


Ilustración 37. Diagrama integración Customer Portal, Service Desk y Desarrollo

5.4 Resumen

Como conclusión del apartado, se presenta el siguiente diagrama ([Ilustración 38](#)) del proceso seguido desde la creación de una incidencia hasta su resolución.

Esta propuesta de integración es necesaria para el entendimiento del [apartado 6](#), en el que se expone un proyecto de integración, consistente en el desarrollo de un Customer Portal, que se integra con TUNE-UP (que actúa como Service Desk) para la gestión de incidencias.

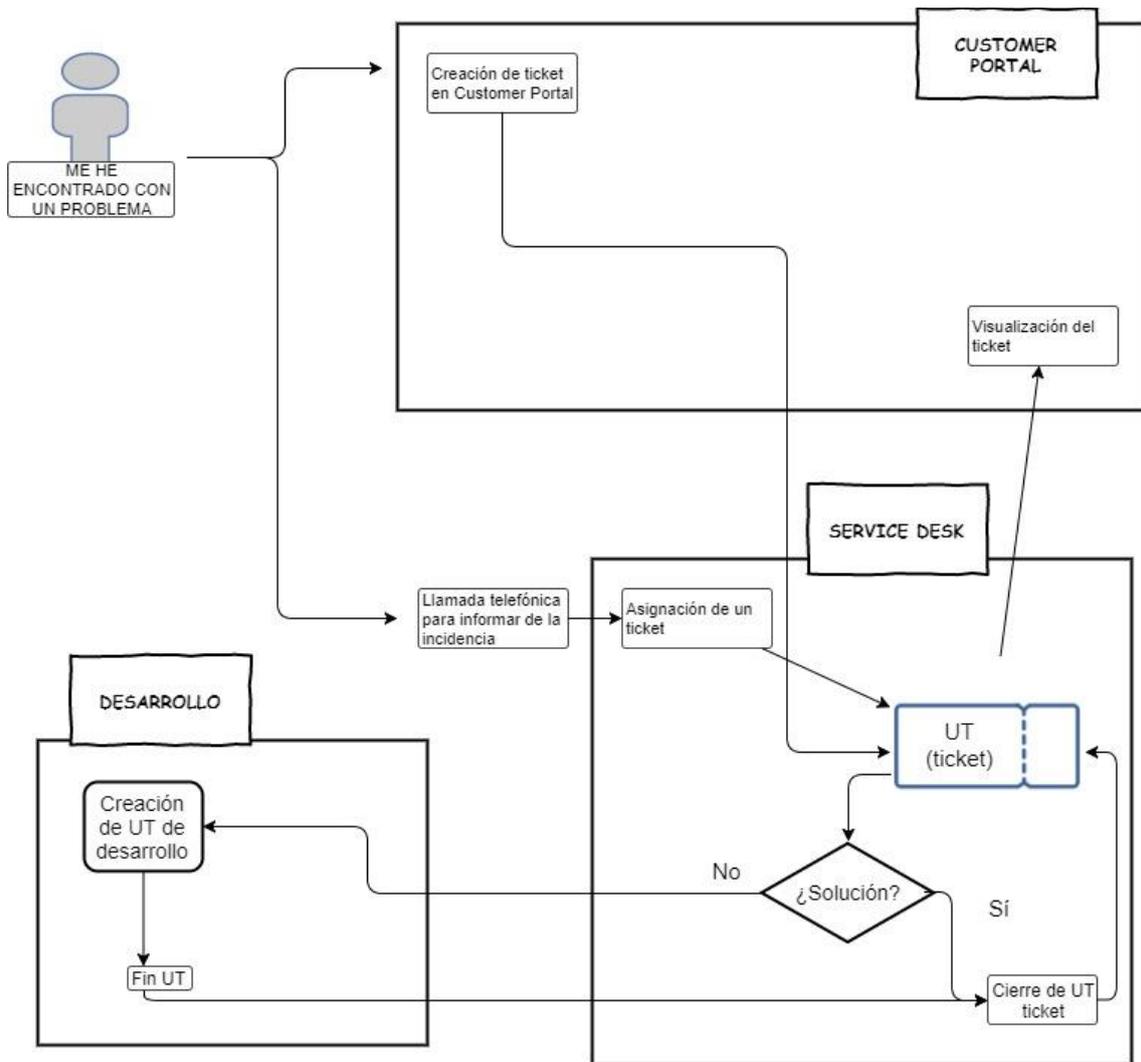


Ilustración 38. Proceso desde la creación de una incidencia hasta su resolución

6. Caso Práctico con TUNE-UP Customer Portal

6.1 Introducción

Como resultado de la propuesta de integración, se ha desarrollado un caso práctico que se explica a continuación.

Este caso consiste en la implementación de un portal de usuario (Customer Portal) que permite gestionar tickets de incidencias sobre un producto concreto, que se integra de forma directa, mediante servicios web, con una herramienta de Service Desk (TUNE-UP).

TUNE-UP Software Process es el nombre de la herramienta de apoyo para la aplicación efectiva de la metodología TUNE-UP. La herramienta está formada por cuatro módulos principales: Personal Planner (PEP), Gestor de UTs (WUM), Version Contents & Tracking (VCT) y Requirements Manager (REM). Cada uno de estos módulos está descrito en el [apartado 4.3.5](#) de esta memoria.

6.2 TUNE-UP Process como herramienta de Service Desk

Para llevar a cabo esta propuesta de integración, se ha utilizado la herramienta TUNE-UP como Service Desk y como herramienta de desarrollo.

Para este propósito, se ha creado un nuevo producto completamente independiente al producto de desarrollo que, únicamente, comparte la estructura del producto como se ha mostrado en apartados anteriores. Cada una de las partes (Service Desk y desarrollo) tiene sus propias unidades de trabajo.

La relación entre ambos productos consiste en que, para cada incidencia, se creará una unidad de trabajo (llamado ticket) en el producto de Service Desk, donde el técnico de nivel uno correspondiente revisará y, posteriormente, solucionará o escalará a desarrollo (mediante la creación de una unidad de trabajo en el producto de desarrollo) dicha incidencia.

Con lo anteriormente descrito, se sigue la propuesta de integración planteada en el [apartado 2.3](#) mediante la creación de dos productos independientes en la herramienta TUNE-UP Process.

6.3 Descripción de la herramienta

TUNE-UP Customer Portal es una herramienta de seguimiento de incidencias, orientada al cliente final de un producto de desarrollo, cuya función principal es poder dar de alta tickets (incidencias), y hacer seguimiento del proceso que se está llevando a cabo desde su apertura hasta su cierre (fin de la incidencia). Además, los tickets cerrados también podrán consultarse, permitiendo obtener la historia de los tickets abiertos a lo largo del ciclo de desarrollo o mantenimiento de un producto.

La herramienta permite crear una lista de contactos para cada cliente, en la que se podrán dar de alta todos los usuarios de éste, que permitirá el ingreso en el sistema con sus credenciales para poder realizar las funciones correspondientes al perfil asignado.

El sistema diferenciará entre dos tipos de usuarios: normales y administradores. Los usuarios administradores tienen funcionalidades adicionales con respecto al resto de usuarios.

6.3.1 Gestión de tickets

El Customer Portal facilita el seguimiento de tickets por parte del usuario, sin necesidad de que éste tenga que enviar un correo electrónico o realizar una llamada al equipo de soporte para abrir una incidencia.

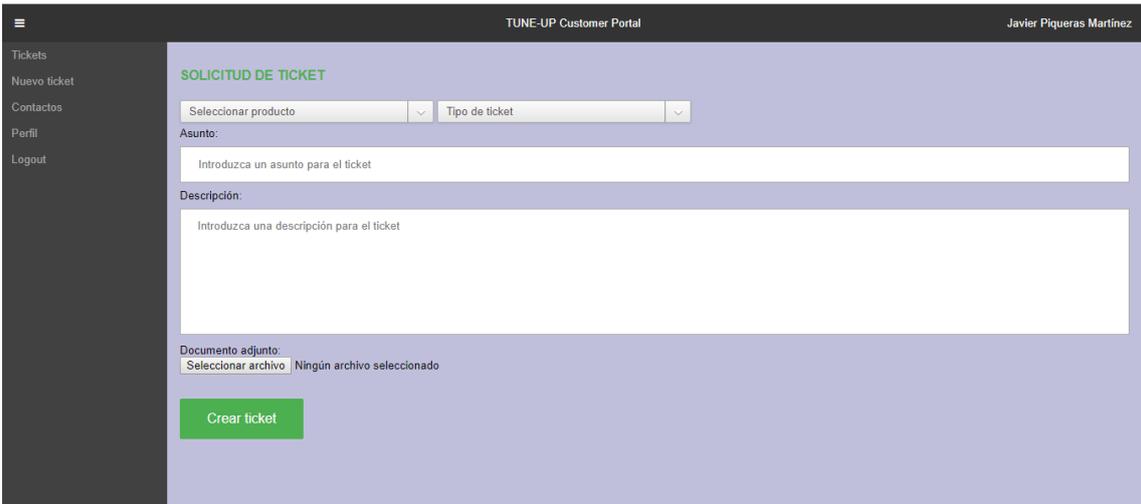
Una vez se ha creado el ticket, el usuario y el técnico de soporte intercambiarán mensajes en los que podrán aclarar las dudas que surjan durante su resolución.

El usuario del portal podrá realizar las siguientes funcionalidades:

Apertura de un ticket

Como se puede observar en la [Ilustración 39](#), desde el dashboard principal de la aplicación, el usuario puede abrir un ticket desde el menú lateral “Nuevo ticket”.

Se deberán proporcionar los siguientes datos relativos al ticket: producto, tipo de ticket, nombre descriptivo y descripción del ticket. Además, se podrá añadir un fichero adjunto donde se ejemplifique el motivo de la incidencia.



The screenshot displays the 'SOLICITUD DE TICKET' form in the TUNE-UP Customer Portal. The interface includes a dark sidebar with navigation options: Tickets, Nuevo ticket, Contactos, Perfil, and Logout. The main content area has a light purple background and contains the following elements:

- Header: 'TUNE-UP Customer Portal' and user name 'Javier Piqueras Martínez'.
- Form Title: 'SOLICITUD DE TICKET'.
- Product Selection: A dropdown menu labeled 'Selección de producto'.
- Ticket Type Selection: A dropdown menu labeled 'Tipo de ticket'.
- Subject Field: A text input labeled 'Asunto:' with the placeholder 'Introduzca un asunto para el ticket'.
- Description Field: A larger text input labeled 'Descripción:' with the placeholder 'Introduzca una descripción para el ticket'.
- Attachment Section: Labeled 'Documento adjunto:', it includes a 'Seleccionar archivo' button and the text 'Ningún archivo seleccionado'.
- Action Button: A green button labeled 'Crear ticket'.

Ilustración 39. Solicitud de ticket en TUNE-UP Customer Portal

Seguimiento de tickets

Desde la opción “Tickets” del menú lateral, accedemos a una tabla resumen de todas las incidencias que el usuario en cuestión ha creado, pudiendo filtrar por producto.

En ella, podremos acceder a la siguiente información: producto, identificador de ticket, nombre de ticket, fecha de creación, fecha de cierre (si la tiene) y estado. Además, la tabla se puede ordenar de forma dinámica según los criterios de cada columna.

Por defecto sólo se muestran los tickets con estado “Abierto”, pudiendo desmarcar el check de “Sólo abiertos” para poder visualizarlos todos. La siguiente figura muestra un ejemplo de la aplicación visualizando la tabla como usuario normal.

Producto	Id	Nombre	Creado en v	Cerrado en	Estado
ACME Soporte	5968	Fallo 4	24/07/2017	-	Abierto
ACME Soporte	5716	Fallo en scroll 2	05/04/2017	-	Abierto
ACME Soporte	5573	Otro ticket	24/02/2017	-	Abierto
ACME Mantenimiento Continuo	5315	Mejorar rendimiento	28/11/2016	-	Abierto

Ilustración 40. Vista de tickets en TUNE-UP Customer Portal

En el caso de ser un usuario administrador, éste tendrá acceso al seguimiento de todos los tickets pertenecientes a usuarios del mismo cliente. Como se puede observar en la [Ilustración 40](#), el desplegable “Contacto” nos permite filtrar los tickets por usuario, siendo éste diferente al usuario administrador que ha ingresado en el sistema.

De esta forma, si en el desplegable se selecciona la opción “All”, se tendrá acceso a todos los tickets. Además, se añadirá de forma dinámica una nueva columna a la tabla que indicará el usuario que ha solicitado el ticket.

Producto	Id	Nombre	Creado en v	Cerrado en	Estado
ACME Soporte	5666	Como recuperar imágenes subidas por mi	31/03/2017	-	Abierto

Ilustración 41. Vista de tickets administrador en TUNE-UP Customer Portal

Visualización detallada de ticket

Al hacer clic en cualquier ticket de la tabla, se observan los detalles del ticket en cuestión. Estos detalles son: Nombre del ticket, Estado del ticket, Solicitante del ticket, Producto del ticket, Fecha de creación del ticket (y de cierre si la hay) y descripción del ticket.

Intercambio de mensajes con el técnico de soporte

Desde esa misma pantalla del sistema, debajo de la descripción detallada del ticket, se podrán enviar nuevos mensajes (y contestar) al técnico de soporte, relacionados con la incidencia abierta. De este modo, se consigue una comunicación constante entre un lado y el otro para aclarar cualquier duda relativa al ticket. Podemos observar esta funcionalidad en la siguiente figura.



Ilustración 42. Detalles de ticket y paso de mensajes en TUNE-UP Customer Portal

Como se ha comentado en la introducción del [apartado 6.2](#), el portal estará integrado con la herramienta de Service Desk, en la que en todo momento, el técnico de soporte observará los tickets que se han creado desde el portal. Además si un ticket es creado desde el Service Desk, el solicitante también tendrá acceso a su seguimiento.

En la [Ilustración 43](#), se puede observar la visualización del ticket de la [Ilustración 42](#) desde el Service Desk.

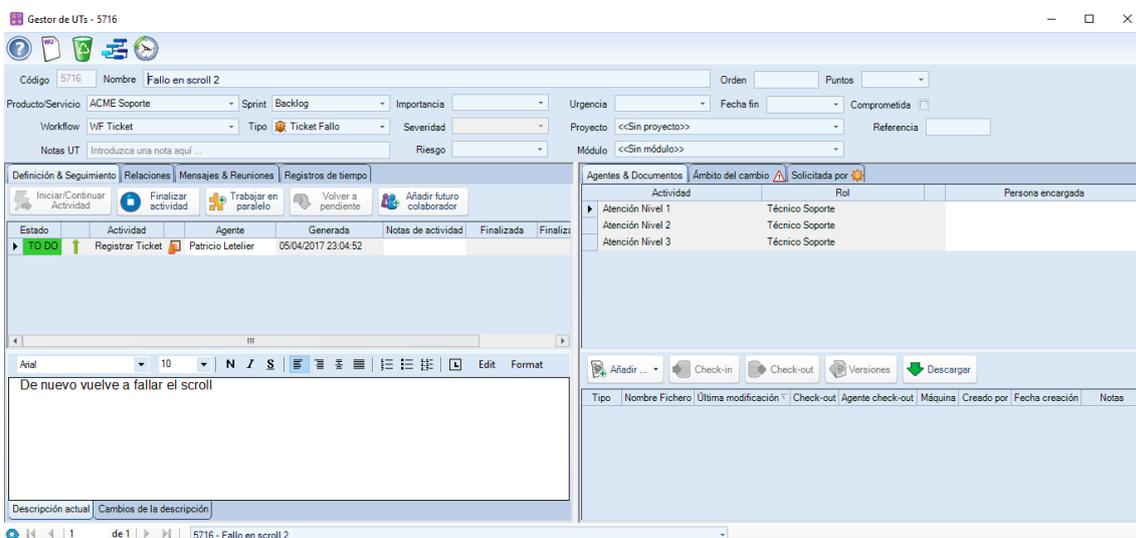


Ilustración 43. Detalles del ticket en TUNE-UP Process

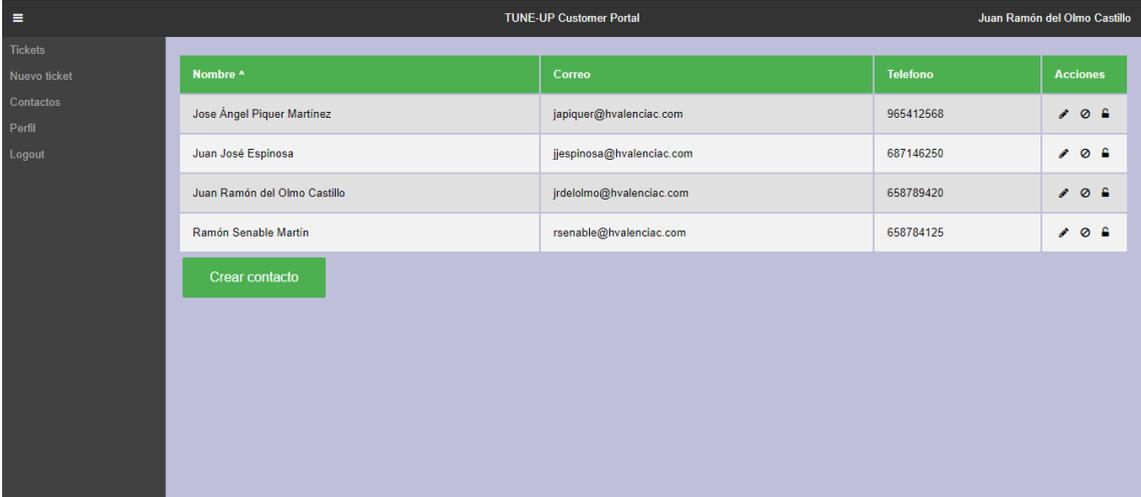
Desde aquí, el técnico de soporte ya puede asignarle al ticket una importancia, severidad, riesgo, urgencia, orden, etc... Así como crear una dependencia con la estructura del proyecto pertinente y crear una unidad de trabajo en el producto de desarrollo si es necesaria para la resolución del ticket.

6.3.2 Gestión de la administración de contactos

La gestión de la administración de contactos es una de las funcionalidades del sistema que únicamente podrá realizar un usuario administrador. En éste módulo, el usuario gozará de las siguientes funcionalidades.

Visualización de la lista de contactos

El administrador podrá ver la lista de contactos pertenecientes al cliente del que forma parte, así como observar los datos de contacto de cada uno de ellos. En la siguiente figura se muestra la tabla (también ordenable de forma dinámica por sus columnas como todas las tablas del sistema) que muestra todos datos de cada usuario (nombre, correo electrónico y teléfono).



Nombre ^	Correo	Telefono	Acciones
Jose Ángel Piquer Martinez	japiquer@hvalenciac.com	965412568	  
Juan José Espinosa	jjespinosa@hvalenciac.com	687146250	  
Juan Ramón del Olmo Castillo	jrdelolmo@hvalenciac.com	658789420	  
Ramón Senable Martín	rsenable@hvalenciac.com	658784125	  

Crear contacto

Ilustración 44. Gestión de contactos en TUNE-UP Customer Portal

Creación de nuevos usuarios

Además, como se puede observar, desde el botón “Crear contacto”, se podrán dar de alta nuevos contactos para el propio cliente en los que se indicará el nombre, dirección de correo electrónico y teléfono. Además también se le podrá dar al usuario derechos de administrador si se desea.

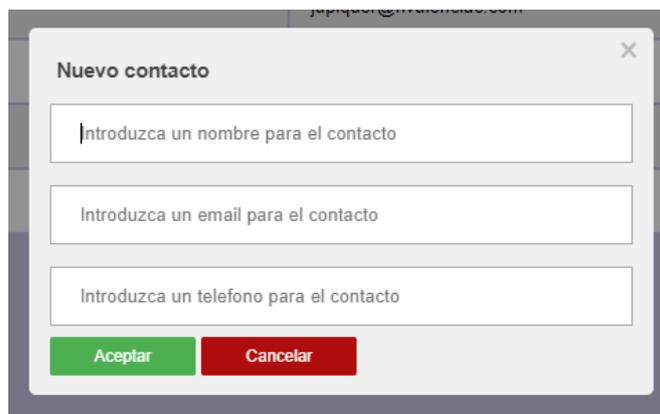


Ilustración 45. Alta de un nuevo contacto en TUNE-UP Customer Portal

Cuando un nuevo contacto es creado, se le asignará por defecto el mismo correo como contraseña.

Como se puede observar en la **figura**, hay tres acciones que se podrán realizar sobre un contacto: editar los datos, desactivar contacto y restaurar contraseña.

Editar datos de contacto

Desde el icono mostrado, se podrán modificar todos los datos de un usuario tal y como se le da de alta (con el mismo formulario).

Desactivar contacto

Desactivar un contacto quiere decir que éste deja de tener acceso al sistema, permaneciendo todos sus datos en la base de datos del sistema, así como los tickets en los que está involucrado.

Restaurar contraseña

Cuando se restaura la contraseña de un contacto por el motivo que sea, ésta pasará a ser la contraseña por defecto (el correo electrónico), pudiendo ser modificada a posteriori por el propio usuario.

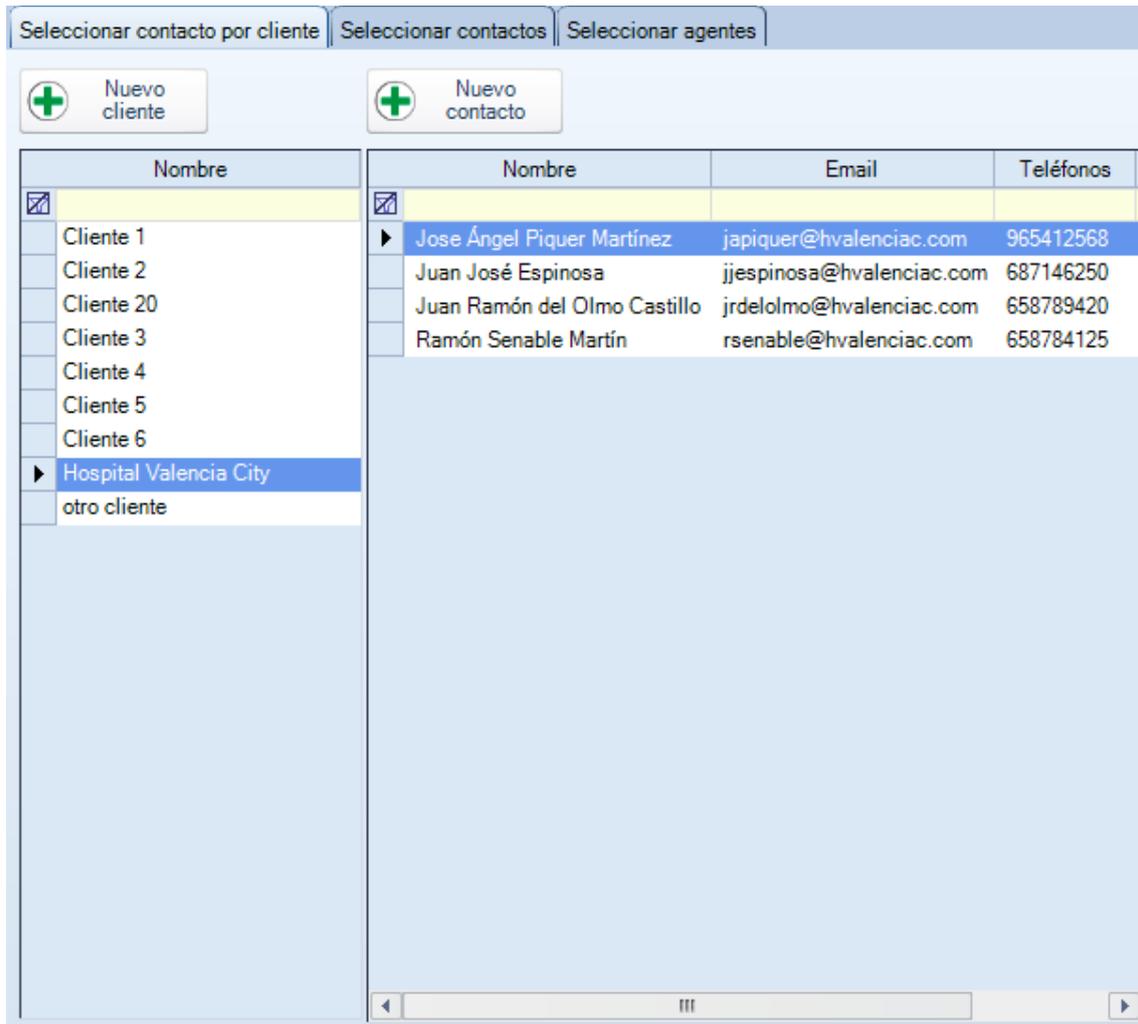


Ilustración 46. Visualización de lista de contactos por cliente en TUNE-UP Process

La figura anterior muestra el estado de los contactos desde el Service Desk, donde un contacto puede ser dado de alta desde la herramienta por un miembro del equipo de soporte.

6.3.3 Gestión de perfil de contacto

Finalmente, desde el módulo de perfil, un usuario puede modificar sus propios datos, así como cambiar su contraseña por la que él quiera como muestra la siguiente figura.

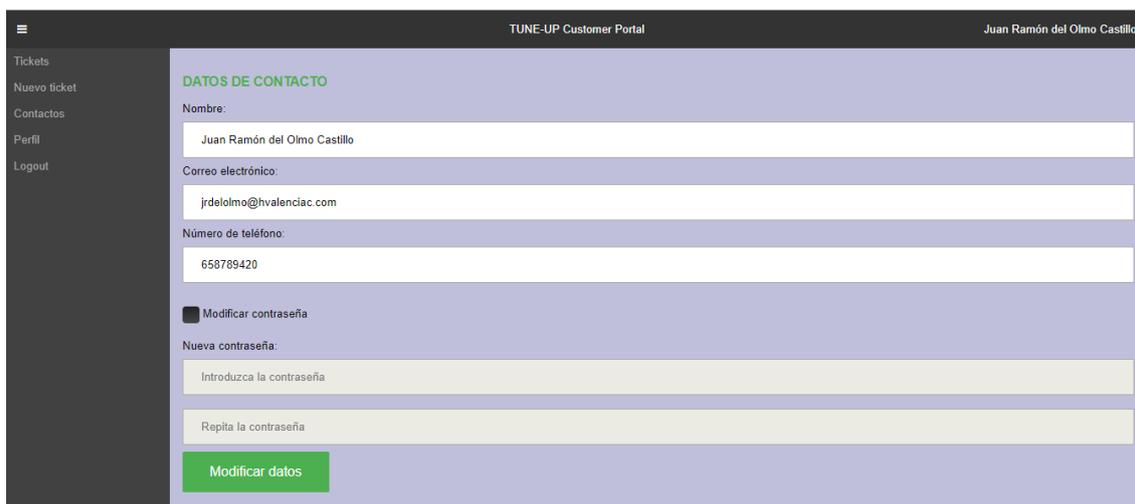


Ilustración 47. Perfil de contacto en TUNE-UP Customer Portal

Como se puede observar, el usuario decidirá si quiere modificar su contraseña o no.

6.4 Proceso de desarrollo y tecnologías utilizadas.

El proyecto ha sido desarrollado a lo largo de seis meses, en los que se han invertido un total aproximado de trescientas horas.

Su desarrollo ha sido ágil, haciendo uso de la herramienta TUNE-UP para descomponer el trabajo en unidades de trabajo y controlar el flujo de trabajo de cada una de ellas. Se observa en la [Ilustración 48](#) la descomposición del proyecto en las distintas unidades de trabajo.

	5936	Instalación en servidor de TUNE-UP
	5935	Mejorar el look & feel
	5783	Mensajes desde o hacia contactos
	50	5749 Añadir documentos adjuntos al ticket en el momento de su creación
	40	5748 Descripción detallada de ticket
	30	5731 Perfil de contacto
	10	5730 Gestión contactos
	15	5728 Mejoras en visualización lista de Tickets
		5688 Mantener filtros al refrescar la pestaña del navegador
	20	5669 Mostrar el nombre de la instalación como marca en el portal
	30	5668 Logout
	10	5617 Login del contacto
	60	5616 Enviar email de confirmación de creación del ticket
	7	5615 Crear un nuevo ticket
	50	5614 Ofrecer ordenamiento por columnas y filtrado
	25	5613 Visualizar tickets usuario administrador
	20	5612 Primera Versión de Mis Tickets

Ilustración 48. Resumen UTs TUNE-UP Customer Portal

El desarrollo ha sido iterativo e incremental, es decir, como en cualquier otro producto software, durante el periodo de implementación se han ido identificando fallos que han sido posteriormente corregidos, así como propuesta de mejoras que han sido implementadas. En la Ilustración 49 se puede observar ese proceso iterativo entre el desarrollador (que implementaba la unidad de trabajo) y el “tester” (que probaba cada una de las pruebas de aceptación que tenía una unidad de trabajo), en el que, cuando una prueba de aceptación no cumplía los requisitos funcionales, era devuelta al desarrollador para que éste la implementara de nuevo con la finalidad de corregir el fallo.

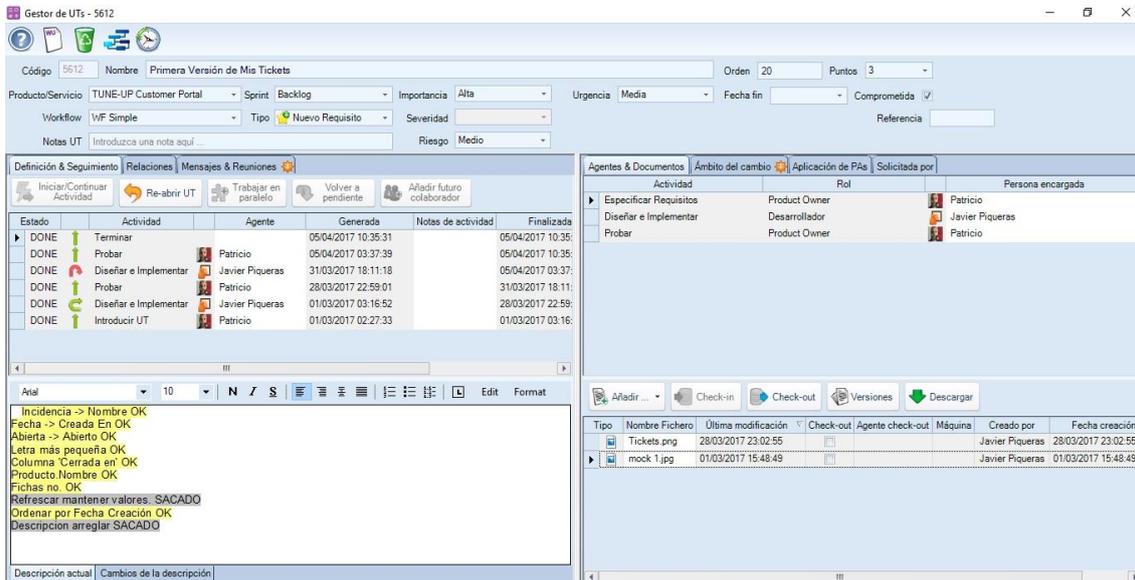


Ilustración 49. Proceso incremental e iterativo de desarrollo

Como se ha comentado, en la Ilustración 49 se aprecia como el flujo de trabajo ha pasado de “Diseñar e Implementar” a “Probar”. A continuación, el agente encargado de probar encuentra errores o fallos que no cumplen los requisitos funcionales estipulados, por lo tanto devuelve la unidad de trabajo a “Diseñar e Implementar”, para que de nuevo el agente encargado de la tarea, corrija los errores y finalmente, seguir el flujo normal que debe llevar la UT hasta llegar al estado terminal “Terminar”.

Para realizar las pruebas del sistema, se disponía de un servidor alojado en el servicio de alojamiento VPSie (Referencia 8) en CentOS 7, en el que se realizaban despliegues de un estado estable del producto, en el que el “tester” probaba las unidades de trabajo en un entorno mucho más similar al del usuario final. Las pruebas de desarrollo se desplegaban en un servidor local (localhost) de la propia máquina donde se desarrollaba.

Cuando el producto ya cogió cierto volumen y empezó a ser funcional, se migró todo el sistema desplegado en VPSie al servidor de TUNE-UP (Referencia 9), donde se fueron haciendo los siguientes despliegues de las versiones finales del producto.

El entorno tecnológico que se ha utilizado las siguientes tecnologías: AngularJS, HTML5, CSS3, PHP y bower.

AngularJS es un framework MVC (modelo-vista-controlador) de JavaScript para el Desarrollo Web Front End que permite crear aplicaciones SPA (Single-Page Applications).

HTML5 (HyperText Markup Language, versión 5) es la quinta revisión importante del lenguaje básico de la World Wide Web, HTML.

Hojas de estilo en cascada (o CSS, siglas en inglés de Cascading Stylesheets) es un lenguaje de diseño gráfico para definir y crear la presentación de un documento estructurado escrito en un lenguaje de marcado.

Bower es una herramienta de control de dependencias que se encarga de instalar las versiones adecuadas necesarias para el proyecto de forma dinámica, sin necesidad de que estas estén guardadas en el repositorio del proyecto.

PHP (acrónimo recursivo de PHP: Hypertext Preprocessor) es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML.

De forma adicional, el proyecto ha sido guardado en un repositorio privado de GitHub [<https://github.com/fracima2/PortalHelp>] (Referencia 2), controlando las distintas versiones que éste adquiriría con ayuda de la herramienta Git. En él se han realizado un total de 80 commits.

6.5 Arquitectura y organización del código

El código sigue la siguiente Estructura (Ilustración 49).

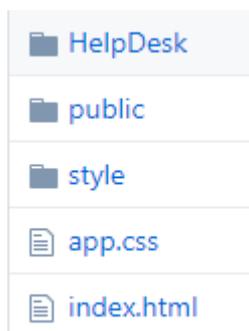


Ilustración 50. Organización del código 1

En la carpeta “HelpDesk” se ha implementado el producto interno, al cual solo tienen acceso los usuarios dados de alta en el sistema. Por otro lado, en la carpeta “public” se ha desarrollado el sistema de Login, al cual tienen acceso todos los usuarios.

Dentro de la carpeta “HelpDesk” se ha organizado el código separando en diferentes carpetas los distintos controladores, directivas, scripts de php, servicios, vistas y hojas de estilos. Esto se puede ver en la Ilustración 51.

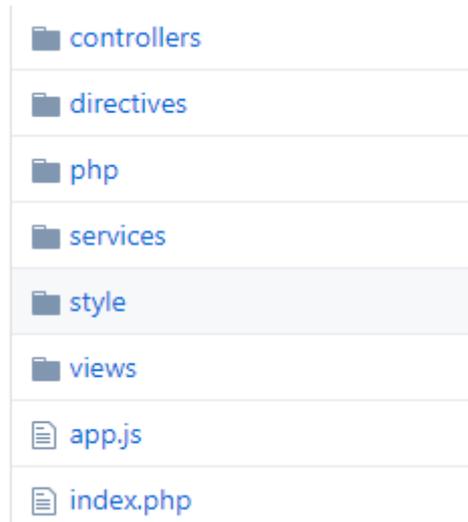


Ilustración 51. Organización del código 2

Una de las carpetas de mayor importancia del producto es la carpeta “services”, donde se encuentran todos los scripts encargados de hacer llamadas a los diferentes servicios de TUNE-UP. En ellos se reciben y se envían todos los datos necesarios para que éste producto pueda interoperar con TUNE-UP sin problemas.

La documentación de los servicios está en la api de TUNE-UP ([Referencia 10](#)).

Como ejemplo de la vista se muestra el siguiente código HTML para su visualización a modo de ejemplo que muestra la vista principal del sistema ([Ilustración 40 - Vista de tickets en TUNE-UP Customer Portal](#)).

```

<!--
DROPDOWNS-
-->

<div class="row dropdowns">
  <div class="inlinedisplay">
    <span>Producto: </span>
    <select id="soflow" ng-model="productChosen">
      <option value="">All</option>
      <option value="{{product.Id}}" ng-repeat="product in
products">{{ product.Nombre }}</option>
    </select>
  </div>
  <div class="inlinedisplay" ng-show="contactInf.Administrador">
    <span>Contacto: </span>
    <select id="soflow" ng-model="userChosen">
      <option value="">All</option>
      <option value="{{client.Id}}" ng-repeat="client in
clients">{{client.Nombre}}</option>
    </select>
  </div>
  <div class="checkboxes inlinedisplay">

```

```

        <div class="inlinedisplay">
            <span>Solo abiertos: </span>
            <div class="squaredThree inlinedisplay">
                <input type="checkbox" value="" ng-
model="showOnlyOpened" id="squaredThree" name="check" checked />
                <label for="squaredThree"></label>
            </div>
        </div>
        <!--<div class="inlinedisplay">
            <span>Vista tickets: </span>
            <div class="slideOne inlinedisplay">
                <input type="checkbox" value="" ng-
model="vistaTickets" id="slideOne" name="check" checked />
                <label for="slideOne"></label>
            </div>
        </div-->
    </div>
</div>
<!--VISTA TABLA-->
<div class="table-div-tickets" ng-if="!vistaTickets">
    <table class="table-tickets">
        <thead>
            <tr>
                <th ng-show="productChosen === '' || !productChosen"
class="date-width">
                    <a class="head-table" href="#" ng-
click="orderByField='NombreProducto'; reverseSort =
!reverseSort">Producto</a>
                    <span ng-show="orderByField ==
'NombreProducto'"><span ng-show="!reverseSort">^</span><span ng-
show="reverseSort">v</span></span>
                </th>
                <th ng-show="userChosen === '' || !userChosen"
class="date-width">
                    <a class="head-table" href="#" ng-
click="orderByField='SolicitadaPor'; reverseSort =
!reverseSort">Creado por</a>
                    <span ng-show="orderByField ==
'SolicitadaPor'"><span ng-show="!reverseSort">^</span><span ng-
show="reverseSort">v</span></span>
                </th>
                <th>
                    <a class="head-table" href="#" ng-
click="orderByField='IdUT'; reverseSort = !reverseSort">Id</a>
                    <span ng-show="orderByField == 'IdUT'"><span ng-
show="!reverseSort">^</span><span ng-

```

```

show="reverseSort">v</span></span>
    </th>
    <th>
        <a class="head-table" href="#" ng-
click="orderByField='Nombre'; reverseSort = !reverseSort">Nombre</a>
        <span ng-show="orderByField == 'Nombre'"><span
ng-show="!reverseSort">^</span><span ng-
show="reverseSort">v</span></span>
    </th>
    <!--<th>
        <a class="head-table" href="#" ng-
click="orderByField='Descripcion'; reverseSort =
!reverseSort">Descripción</a>
        <span ng-show="orderByField ==
'Descripcion'"><span ng-show="!reverseSort">^</span><span ng-
show="reverseSort">v</span></span>
    </th-->
    <th class="date-width">
        <a class="head-table" href="#" ng-
click="orderByField='FechaCreacion'; reverseSort =
!reverseSort">Creado en</a>
        <span ng-show="orderByField ==
'FechaCreacion'"><span ng-show="!reverseSort">^</span><span ng-
show="reverseSort">v</span></span>
    </th>
    <th class="date-width">
        <a class="head-table" href="#" ng-
click="orderByField='FechaCierre'; reverseSort =
!reverseSort">Cerrado en</a>
        <span ng-show="orderByField ==
'FechaCierre'"><span ng-show="!reverseSort">^</span><span ng-
show="reverseSort">v</span></span>
    </th>
    <th class="state-width">
        <a class="head-table" href="#" ng-
click="orderByField='Abierta'; reverseSort = !reverseSort">Estado</a>
        <span ng-show="orderByField == 'Abierta'"><span
ng-show="!reverseSort">^</span><span ng-
show="reverseSort">v</span></span>
    </th>
</tr>
</thead>
<tbody>
    <tr ng-repeat="ticket in tickets |
orderBy:orderByField:reverseSort"
ng-if="(!showOnlyOpened || ticket.Abierta)

```

```

(ticket.IdProducto == productChosen || productChosen == '' ||
!productChosen)"
    ng-click="irATicket(ticket)">
    <td ng-if="productChosen===' ' || !productChosen"
class="sz-small ">{{ ticket.NombreProducto }}</td>
    <td ng-if="userChosen===' ' || !userChosen"
class="sz-small ">{{ ticket.SolicitadaPor }}</td>
    <td class="sz-small ">{{ ticket.IdUT }}</td>
    <td class="sz-small "><b>{{ ticket.Nombre }}</b></td>
    <!--<td class="sz-small ">{{
ticket.Descripcion.trim() }}</td-->
    <td>{{ ticket.FechaCreacion | date: 'dd/MM/yyyy'
}}</td>
    <td ng-if="ticket.FechaCierre !== null">{{
ticket.FechaCierre | date: 'dd/MM/yyyy' }}</td>
    <td ng-if="ticket.FechaCierre === null"> - </td>
    <td ng-if="ticket.Abierta" class="sz-small
">Abierto</td>
    <td ng-if="!ticket.Abierta" class="sz-small
">Cerrado</td>
    </tr>
</tbody>
</table>
</div>

```

6.6 Manual de instalación

Para la correcta instalación del sistema, en un sistema operativo CentOS 7, se deben seguir los siguientes pasos:

- Creación de un servidor Apache. En primer lugar, se debe realizar una limpieza del “yum”, se actualiza y se instala:

```
sudo yum clean all
```

```
sudo yum -y update
```

```
sudo yum -y install httpd
```

- Para su configuración, en primer lugar, se deben abrir los puertos 80 y 443 ejecutando los comandos:

```
sudo firewall-cmd --permanent --add-port=80/tcp
```

```
sudo firewall-cmd --permanent --add-port=443/tcp
```

- Y recargar el cortafuegos:

```
sudo firewall-cmd --reload
```

- Finalmente, para la ejecución del servidor y su posterior manejo, se hará uso de los siguientes comandos:

```
sudo systemctl start httpd
```

```
sudo systemctl enable httpd
```

```
sudo systemctl status httpd
```

```
sudo systemctl stop httpd
```

- A continuación, una vez el servidor Apache se ha instalado, se debe descargar el código fuente del proyecto desde el repositorio de GitHub <https://github.com/frapima2/PortalHelp>. Posteriormente, ubicarlo en la carpeta del servidor Apache.

Clone or download ▾

Ilustración 52. Descarga del repositorio

- Activar servidor Apache.
- Acceder a la dirección del servidor para disfrutar del producto.

7. Conclusiones y trabajo futuro

Este proyecto ha conseguido integrar dos herramientas completamente independientes, que hacen uso de tecnologías diferentes.

Gracias a esto, se ha conseguido que, un usuario final de un sistema, sea capaz de reportar incidencias (abriendo tickets), que de forma automática crean una unidad de trabajo en el primer nivel del equipo de soporte, quién evalúa la incidencia y la soluciona o escala a otro nivel si procede. Además, el usuario puede realizar un seguimiento del ticket en todo momento y puede tener un canal de contacto dentro de cada ticket con el técnico de soporte en cuestión que les servirá para resolver dudas que puedan tener entre ellos. El usuario administrador podrá realizar funciones como gestionar los contactos pertenecientes a su propio cliente que el resto de usuarios no podrán realizar.

De este modo, se cumplen los objetivos establecidos previos al inicio del desarrollo del sistema.

Dada la aplicación de las nuevas tecnologías en el usuario final, por una parte, y dada la necesidad de desarrollo de proyectos en el entorno de la interoperabilidad entre sistemas, he considerado muy importante facilitar el uso de dichas tecnologías (en relación a la gestión de incidencias) así como facilitar también su integración con una herramienta de Service Desk de forma rápida, transparente y ágil, permitiendo al usuario poder dar de alta nuevas peticiones de servicio como hacer un seguimiento y trazabilidad de ellas.

Este proyecto me ha ayudado mucho a ampliar mucho más mis conocimientos sobre las diferentes metodologías ágiles y su aplicación en el ámbito de las incidencias software, así como corroborar los ya obtenidos anteriormente en la asignatura de Proceso de Software. Además, dado que los servicios REST los implementaba el desarrollador de TUNE-UP, he aprendido a trabajar en dos equipos completamente separados, en los que se negocia como es mejor abordar el desarrollo del sistema mediante la petición del desarrollo de servicios.

Se espera una implantación futura en distintas empresas donde la herramienta TUNE-UP está en uso. Así como su implantación en parte de la asignatura de PSW (proceso de software) y PIN (proyecto de ingeniería del software), ya que enseñaría a los alumnos, esta parte del proceso que hasta ahora no se ha podido ejemplificar.

Como mejora futura, considero necesario mejorar la interfaz gráfica de usuario por un UX designer, de forma que ésta pueda estar a la altura del resto de productos competentes.

8. Referencias

1. How to Install Apache on CentOS 7. (14/septiembre/2017) - <https://www.liquidweb.com/kb/how-to-install-apache-on-centos-7/>
2. Repositorio de caso práctico. (14/septiembre/2017) - <https://github.com/frapima2/PortalHelp>
3. Wikipedia. (14/septiembre/2017) - <https://es.wikipedia.org/>
4. Auxi Carlos Alberola. Máster en Ingeniería del Software. Trabajo fin de Máster. Departamento DSIC, UPV. Año 2013. “Integración de Service Desk con Desarrollo de Software basándose en ITILy métodos ágiles”.
5. Flowchart Marker & Online Diagram Software. (14/septiembre/2017) - <https://www.draw.io>
6. AngularJS – Superheroic JavaScript MVW framework. (14/septiembre/2017) - <https://angularjs.org/>
7. Análisis de Impacto de Requisitos en un proceso de desarrollo centrado en Pruebas de Aceptación.
8. SSD VPS Cloud Hosting | Windows Linux | (14/septiembre/2017) - <https://vpsie.com/>
9. TUNE-UP Customer Portal (14/septiembre/2017) - <http://cliente.tuneupprocess.com/CustomerPortal/app>
10. TUNE-UP Api (14/septiembre/2017) – <http://cliente.tuneupprocess.com/webapi/help>
11. HTML Tutorial w3schools(14/septiembre/2017) - <https://www.w3schools.com/html/default.asp>
12. CSS Tutorial s3schools (14/septiembre/2017) - <https://www.w3schools.com/css/default.asp>
13. AngularJS Tutorial w3schools (14/septiembre/2017) - <https://www.w3schools.com/angular/default.asp>
14. JavaScript Tutorial w3schools (14/septiembre/2017) - <https://www.w3schools.com/js/default.asp>
15. JSON Tutorial w3schools (14/septiembre/2017) - https://www.w3schools.com/js/js_json_intro.asp
16. Web templates w3schools(14/septiembre/2017) - https://www.w3schools.com/w3css/w3css_templates.asp
17. Código y librerías de AngularJS 1.6.6 (14/septiembre/2017) - <https://code.angularjs.org/1.6.6/>