
This is the **published version** of the bachelor thesis:

Fijo Moreno, Adrià; Oropesa Física, Ana, dir. Creación de plataforma ágil para la gobernanza de departamentos IT. 2022. (958 Enginyeria Informàtica)

This version is available at <https://ddd.uab.cat/record/264138>

under the terms of the  license

Creación de plataforma ágil para la gobernanza de departamentos IT

Fijo Moreno, Adrià

Resumen— Con el objetivo de facilitar la gestión de peticiones de una organización, este trabajo trata sobre la creación y personalización de un portal de autoservicio usando la plataforma de Jira proporcionada por Atlassian. A través del portal, los usuarios envían peticiones a la empresa y pueden ver el estado de dichas peticiones en cualquier momento. Estas peticiones son escaladas automáticamente a los Grupos Resolutores correspondientes, los cuales son los encargados de gestionarlas o denegarlas. El proyecto también consta de una parte de gestión de proyectos, siendo que si un Grupo Resolutor considera que una petición es lo bastante compleja, puede crear un proyecto fácilmente para gestionarla, además de poder definirse un presupuesto para el proyecto y hacer un seguimiento del mismo.

Palabras clave— Atlassian, Automatización, Centro de Atención al Usuario (CAU), Gestión de Proyectos, Incidencias, Jira, Peticiones, Portal Autoservicio, SPOC, SLA

Abstract— With the aim of facilitating the management of requests in an organization, this work is about the creation and customization of a self-service portal using the Jira platform provided by Atlassian. Through the self-service portal, users send requests to the company and they are able to see the status of these requests at any time. These requests are automatically escalated to the corresponding Resolutor Group, which is in charge of managing and denying said requests. The project also has a project management part, if a Resolutor Group considers that a request is complex enough, it can easily create a project to manage it, in addition a budget can be defined for the project and a monitoring of it.

Keywords— Atlassian, Automation, Issues, Jira, Project Management, Requests, Self Service Portal, SPOC, SLA, User Support Center (USC)

1 INTRODUCCIÓN

EL trabajo de fin de grado se ubica en la empresa Quabu Solutions SL [1], la cual se dedica a realizar diversos proyectos en el ámbito informático que tocan una gran variedad de temas como pueden ser el diseño web, proyectos de inteligencia artificial, o proyectos de transformaciones digitales de empresas.

Debido a la necesidad de Quabu Solutions de crear un proyecto que unificase todos los tipos de peticiones de una empresa cliente, se decidió hacer un trabajo de final de grado para cubrir dicha necesidad.

El proyecto trata sobre la creación de un portal SPOC (*Single Point of Contact*) para una empresa cliente, este portal consiste en unificar todos los canales de entrada y salida de información en un único punto, de manera que facilite la gestión de dicha información y agilice la gestión de la organización [2].

Dicho de otra manera, se trata de un portal a través del cual los clientes podrán mandar todo tipo de peticiones a la empresa, y esta podrá llevar a cabo dichas peticiones, y no solo eso, sino que la organización internamente podrá usar el portal para gestionar sus propios proyectos internos.

2 OBJETIVOS

En esta sección se explicarán los objetivos del proyecto. El objetivo general del mismo es conseguir una mayor facilidad a la hora de gestionar peticiones tanto fuera como dentro de la organización, concentrando todos los modos

- E-mail de contacto: adria.fijo@autonoma.cat
- Mención realizada: Tecnologías de la Información
- Trabajo tutorizado por: Ana Oropesa Física (DEIC)
- Curso 2021/22

de petición en una sola aplicación. Además, también hay cuatro objetivos específicos.

El primero de ellos es conocer la situación actual de la empresa cliente para poder conseguir un listado del alcance del proyecto, es decir, saber cuantos tipos de usuarios hay y qué tipo de peticiones se podrán realizar. Esto se hará a través de un estudio sobre la organización y estructura interna de la empresa, el cual incluirá toma de requerimientos, análisis de los requisitos y propuesta de mejoras, dicho estudio durará dos semanas.

El segundo de los objetivos específicos se trata de la creación de un borrador de la interfaz de la aplicación y la obtención de validación por parte de la empresa. La creación del borrador y la obtención de la aprobación de la empresa se espera que tarde una semana.

El tercer objetivo es el desarrollo de la aplicación en sí. Este objetivo es el que llevará más tiempo, aproximadamente seis semanas. Se trata de crear la aplicación para la empresa usando la *suite* de Atlassian. La parte de la interfaz es la más liviana ya que este objetivo es temporalmente posterior a la creación del borrador de la interfaz pero se deberá configurar todos los permisos, los distintos tipos de peticiones y los automatismos.

Finalmente, el último de los objetivos específicos es el de la verificación del objetivo general sobre la mayor facilidad de gestionar las peticiones dentro de la empresa mediante una encuesta a diferentes tipos de usuario dentro de ella, con el fin de saber la relevancia y el impacto del proyecto. Se espera una puntuación de más de 7 puntos sobre 10 sobre la facilidad. Esta acción se llevará a cabo con una duración de dos semanas.

3 PLANIFICACIÓN Y METODOLOGÍA

En esta sección se ve la planificación y metodología usadas para este proyecto.

3.1. Planificación

Para planificar el trabajo se ha decidido realizar un diagrama de Gantt, ya que es una herramienta muy útil para representar visualmente la planificación de un proyecto [3].

Las distintas fases del proyecto son cada uno de los objetivos específicos comentados en el apartado anterior más una última fase de realización del informe final del trabajo. Además se han incluido buffers de tiempo para imprevistos, marcados de color rosa, y en amarillo se muestran las fechas de entrega de los distintos informes del trabajo, por orden están: entrega del informe inicial, informe de seguimiento I, informe de seguimiento II y propuesta del informe final.

Para realizar el diagrama de Gantt se ha usado una herramienta que puede descargarse de forma gratuita llamada GanttProject [4]. El diagrama de Gantt lo podemos ver en el Anexo 1.

3.2. Metodología

La metodología que se ha escogido para la realización del trabajo de fin de grado se trata de Kanban. A la hora de ele-

gir una metodología, se escogieron las ágiles, ya que estas te permiten conseguir una mayor flexibilidad e inmediatez a la hora de amoldar el proyecto que las metodologías tradicionales [5].

Dentro de las metodologías ágiles se decidió trabajar con Kanban ya que es una metodología que ya se había usado con anterioridad y se ha considerado que es muy cómodo trabajar con ella debido a que ayuda a la organización del proyecto y es sencilla de implementar.

La metodología Kanban consiste en un tablero, las columnas del cual muestran las distintas fases por las que han de pasar las tareas para poder ser desarrolladas, generalmente los tableros Kanban constan de 3 columnas; “Por Hacer”, “En Proceso” y “Finalizado” [6]. Las tareas son representadas mediante tarjetas y se van desplazando de izquierda a derecha a través de las distintas columnas (fases) del tablero hasta ser realizadas.

Para utilizar la metodología Kanban se usará Jira Software de la *suite* Atlassian [7].

4 TOMA DE REQUISITOS

El primero de los objetivos específicos que se estableció fué el de conocer la situación actual de la empresa, para ello se hizo un proceso de toma de requisitos y en esta sección veremos los resultados del mismo.

4.1. Situación actual de la empresa

La empresa cliente para la cual se creará la aplicación es una multinacional que cuenta con más de 7.500 empleados y necesitan una plataforma para gestionar de manera ágil y unificada todos los tipos de peticiones que pueden hacer los empleados dentro de la empresa.

4.2. Necesidades del cliente

En la Tabla 1 se muestran las necesidades del cliente y una comparativa de las herramientas que se usarán para solucionarlas; Jira Software y Jira Service Management:

Necesidades cliente	Jira Sw	Jira Svc Mgmt
Crear proyectos	Sí	Sí
Flujos de trabajo	Sí	Sí
Informes básicos	Sí	Sí
Tableros ágiles	Sí	No
Planificación <i>backlog</i>	Sí	No
SLAs	No	Sí
Help Desk	No	Sí

TABLA 1: Necesidades del cliente

Jira Software es una herramienta que permite a equipos poder planificar, asignar, supervisar y gestionar el trabajo, así como elaborar informes al respecto [8]. Esta herramienta se usará para la parte de Gestión de Proyectos del trabajo.

También se usará Jira Service Management, el cual sirve para recibir, hacer un seguimiento, gestionar y resolver fácilmente las solicitudes de los clientes de nuestro equipo. También permite a los clientes poder enviar solicitudes

mediante un *help center* personalizable y además facilita la clasificación de solicitudes de servicio, incidencias, problemas y cambios, organizando y priorizando estas solicitudes en un solo sitio [9].

Al trabajar con Jira hay tres opciones de *hosting* distintas, trabajar con Jira en el Cloud, Jira Server o Jira Data Center. Se usará Data Center debido a que es el que usa la empresa cliente. La principal diferencia entre estos tipos de *hosting*, es que en la versión Cloud, Atlassian se encarga de alojar tu despliegue de Jira mientras que en las versiones de Server y Data Center es necesario encargarse del alojamiento [10].

A continuación se detallan los tipos de usuarios y peticiones que necesita el cliente.

4.2.1. Usuarios

En el sistema hay dos tipos de usuarios distintos. Por un lado tenemos los empleados de la empresa que son los usuarios base, estos usuarios pueden hacer peticiones al portal de autoservicio y ver el estado de dichas peticiones.

Por otro lado tenemos los Grupos Resolutores, estos son usuarios o grupos de usuarios, que reciben las peticiones de los primeros y las gestionan. También pueden hacer peticiones al portal si así lo desean. Los Grupos Resolutores que se crearán para gestionar todas las peticiones son:

Helpdesk: Gestiona las incidencias y las peticiones relacionadas con la mejora de la organización.

Ciberseguridad: Gestiona las peticiones relacionadas con el acceso a los sistemas de la empresa.

Compras: Encargados de gestionar las peticiones de compra de activos hardware y software.

Sistemas: Gestiona peticiones de soporte técnico y los cambios en el software o productos de la empresa.

Laboral: Gestiona las peticiones relacionadas con los empleados, como nuevas incorporaciones o bajas.

Mantenimiento: Se encarga de gestionar las peticiones de limpieza de la empresa y sus instalaciones.

Cabe destacar en este apartado que el proyecto cuenta con una aplicación extra, la cual se trata de la Gestión de Proyectos y los Grupos Resolutores la usarán para crear proyectos si las peticiones de los usuarios lo requieren.

4.2.2. Tipos de peticiones

Existen diversas necesidades dentro de la empresa cliente y por eso necesitan diferentes tipos de peticiones para satisfacerlas.

En la Figura 1 podemos ver las peticiones que resolverá cada uno de los Grupos Resolutores, a continuación se explican brevemente en que consisten dichas peticiones y que necesidades satisfacen:

Solicitar nuevo Hardware: para que los empleados puedan solicitar, por ejemplo, ratones, teclados, etc.

Solicitar nuevo dispositivo móvil: si un empleado necesita un nuevo dispositivo móvil.

Solicitar Software: los empleados de la empresa cliente pueden solicitar softwares y la empresa estudiará la viabilidad de la propuesta.

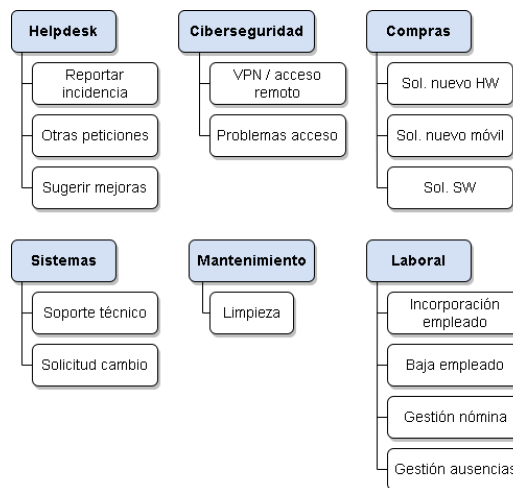


Fig. 1: Peticiones y Grupos Resolutores

VPN / acceso remoto: para que los empleados puedan solicitar una VPN si necesitan acceder a redes de la empresa desde casa.

Solicitud de cambio: si se necesita hacer algún cambio o actualización de algún software que use toda la empresa, como por ejemplo actualizar Jira, se creará esta petición y se valorará la viabilidad de la misma.

Problemas de acceso: si un empleado tiene problemas de acceso o permisos a alguno de los sistemas de la empresa en esta petición los puede especificar.

Soporte técnico: si el usuario necesita ayuda instalando algún software, configurando algún programa, etc.

Reportar una incidencia: si el usuario quiere reportar algún problema que ha tenido o visto relacionado con la organización o sus servicios.

Sugerir mejoras: los usuarios pueden proponer mejoras a la organización.

Incorporación nuevo profesional: solicitar acceso para nuevos empleados de la empresa.

Baja empleado: cuando un empleado abandona la empresa, en esta solicitud se especifican los aspectos que debe gestionar la organización, como por ejemplo si el empleado dispone de algún activo que devolver (ordenador de la empresa, etc).

Gestión de nómina: si el empleado quiere hacer alguna petición respecto a su nómina, por ejemplo informar de un cambio en su domiciliación bancaria.

Gestión de ausencias o vacaciones: si el usuario estará ausente del trabajo durante unos días concretos, aquí podrá especificar el motivo, ya sea personal, vacacional, etc.

Limpieza: peticiones relacionadas con las tareas de limpieza y mantenimiento de la empresa y sus instalaciones.

Otras peticiones: si el usuario no ha encontrado la solicitud que quería hacer, la puede especificar aquí.

4.3. Lógica del sistema

En esta subsección se explica el funcionamiento de la aplicación. Dicho funcionamiento está ilustrado en la Figura 2.

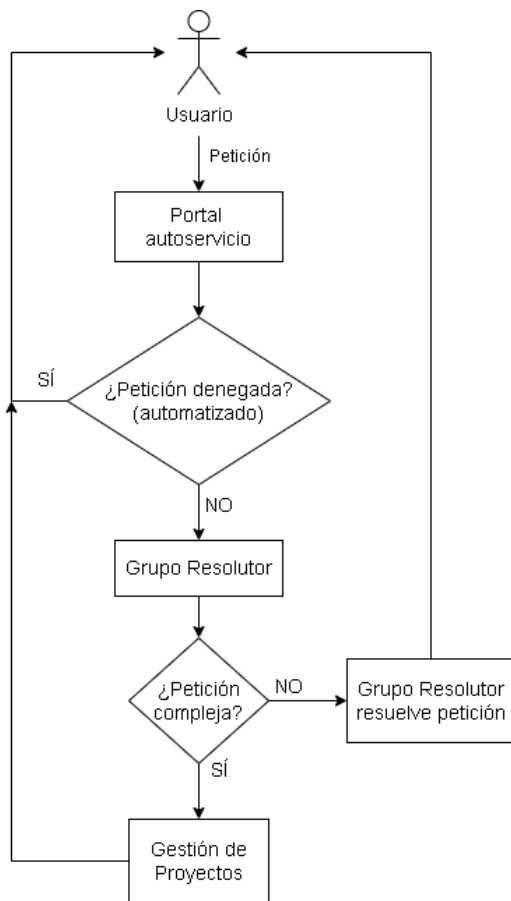


Fig. 2: Lógica de la aplicación

El usuario entra al portal de autoservicio y genera una petición. Esta petición se denegará o escalará a uno de los Grupos Resolutores de forma automática a través de un *script*. Si dicha petición es denegada, se informará al usuario.

Una vez la petición está debidamente escalada en su correspondiente Grupo Resolutor, este determinará si la petición es lo suficientemente compleja como para crear un proyecto para solventarla. En el caso que la petición no sea demasiado compleja, el Grupo Resolutor la resolverá.

Si la petición resulta ser lo suficiente compleja, se procederá a la fase “Gestión de Proyectos” la cual comprende todas las fases de la creación y gestión de un proyecto, desde definir un presupuesto hasta realizar la ejecución de dicho proyecto.

Tanto si la petición es resuelta por el Grupo Resolutor como si es necesaria la creación de un proyecto, se informará al usuario.

4.4. Requisitos

En este apartado se explican los requisitos, tanto funcionales como no funcionales del proyecto.

4.4.1. Requisitos funcionales

1. Los usuarios han de poder enviar peticiones a través del portal de autoservicio.
2. Los Grupos Resolutores han de recibir automáticamente una notificación cuando les hacen una petición.

3. Los Grupos Resolutores han de ser capaces de resolver las peticiones de los usuarios.
4. Los usuarios han de poder consultar a través del portal el estado de sus peticiones.
5. Los grupos resolutores han de ser capaces de crear y gestionar proyectos si así lo requiere un tipo específico de petición.
6. Tanto el portal como la gestión de peticiones, incidencias y proyectos se hará usando las plataformas de Jira Service Management y Jira Software.

4.4.2. Requisitos no funcionales

1. La interfaz del portal de autoservicio ha de ser fácil de usar para los usuarios.
2. La interfaz del portal debe adaptarse a la imagen corporativa de la empresa cliente.

4.5. Costes añadidos del proyecto

El proyecto está sujeto a dos grandes costes añadidos; el primero de ellos es que tanto Jira Software como Jira Service Management son de pago y requieren de licencia para ser usados.

El otro son las limitadas opciones de customización del portal de autoservicio. Se buscó en Atlassian Marketplace para ver si había algún *plugin* que ofreciese más opciones de personalización al portal, y se vió que todos los que podían ayudar son de pago.

5 CREACIÓN INTERFICIE PORTAL

Se creó un borrador a mano de la interfaz del portal, el cual fué aprobado por el equipo de Quabu y la empresa cliente. En la Figura 3 se puede ver dicho borrador.

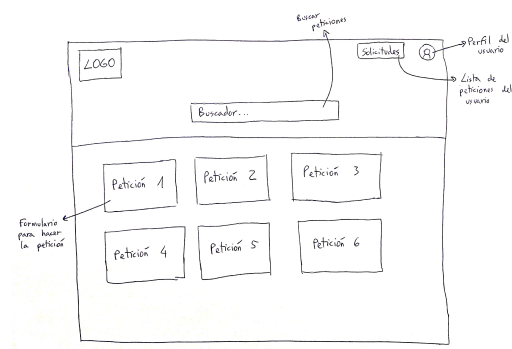


Fig. 3: Borrador interfaz portal

Debido a temas de confidencialidad, se decidió no usar el logo de la empresa cliente y crear un logo como ejemplo.

Para crear dicho logo se usó una página web que genera un logo aleatorio para tu negocio de forma gratuita, llamada Adobe Creative Cloud Express [11]. Mediante esta aplicación, se creó un logo lo más simple posible para una empresa ejemplo, la cual se llamó “TFG”.

Una vez creado el logo, se adaptó el *look & feel* del portal de autoservicio para que se adecuara a la imagen corporativa que se había creado. Sin embargo, como se comentó anteriormente, las opciones de customización del portal de autoservicio que ofrece Atlassian por defecto no son suficientes. Los portales creados y personalizados con las herramientas base, son muy similares entre sí, lo que no ayudaría a distinguir lo suficiente a la empresa de sus competidoras.

Por ello, se hizo un pequeño estudio para ver que *plugin* de Atlassian Marketplace se podía utilizar para personalizar el portal. Se encontraron tres que podían ser útiles, sin embargo, y como se comentó en la subsección de Costes añadidos del proyecto, los tres son de pago.

Se eligió basándose en el coste ya que no se disponía del tiempo suficiente para probarlos todos, el *plugin* elegido se llama Theme Extension for Jira Service Management [12].

Una vez escogido e instalado el *plugin*, se creó un proyecto en Jira del tipo “Service Desk”, el cual incorpora un portal de autoservicio por defecto, se llamó a dicho proyecto CAU (Centro de Atención al Usuario). Su función es actuar como portal para que los usuarios envíen peticiones, y escalar dichas peticiones a los Grupos Resolutores.

Acorde a la planificación, se tardó en crear el borrador de la interfaz de la aplicación una semana y usando el *plugin* comentado anteriormente, se gozaron de unas opciones de personalización mucho más extensas que con la aplicación base. Entre otras cosas, se pudo usar un tipo de distribución del contenido llamada *Cards* donde las distintas peticiones se muestran como tarjetas. También se editaron los colores del portal para que se adaptaran al color de nuestra organización de ejemplo.

En los Anexos 2 y 3 se muestra la interfaz del portal de autoservicio antes y después, respectivamente, de aplicar la personalización mediante el *plugin* comentado.

6 CONFIGURACIÓN DEL SISTEMA

En esta sección se detallan los pasos que se han ido siguiendo a la hora de configurar el portal de autoservicio.

6.1. Personalización de peticiones

Una vez creada la interfaz del portal de autoservicio, se empezó a trabajar en los tipos de peticiones que pueden realizar los usuarios.

Para ello se crearon Campos Personalizados, los cuales sirven para recolectar información que no está disponible en los campos por defecto del sistema [13]. Ocuparía mucho espacio detallar todos los campos usados para cada una de las distintas peticiones que hay en el portal, por lo tanto solo se verá el ejemplo de uno de los campos creados para la petición Incorporación de empleado.

El campo en cuestión se trata de una lista de selección múltiple donde el usuario ha de seleccionar que es lo que necesita el nuevo empleado (ordenador, dispositivo móvil, email corporativo...) para que así la empresa cliente lo pueda gestionar.

6.2. Configuración proyecto CAU

A continuación se procedió a la configuración del proyecto CAU que actúa como portal de autoservicio.

Para empezar se creó un esquema de tipos de incidencias, el cual agrupa tipos de incidencias para hacer más fácil la creación de las mismas y el mantenimiento del proyecto [14]. Como el proyecto CAU solo actúa como portal, este esquema solo tiene incidencias del tipo “Solicitud de Servicio” y del tipo “Incidente”, que son los dos tipos de peticiones que podrá crear un usuario a través del portal de autoservicio.

También se creó un flujo de trabajo específico para el proyecto CAU. Este consiste en la definición de un conjunto de estados y transiciones por las que pasa una petición durante su ciclo de vida [15]. En la Figura 4 hay un esquema del flujo de trabajo creado:

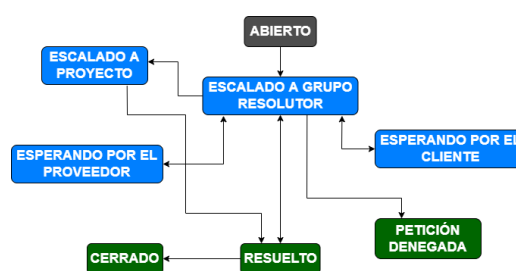


Fig. 4: Flujo de trabajo CAU

Otro campo que se editó fué el esquema de pantallas. Este campo permite elegir que pantalla se le mostrará al usuario de Jira cuando realice una operación relacionada con incidencias, hay tres tipos de estas operaciones; crear una incidencia, editar una incidencia y visualizar una incidencia [16]. Este campo no se tuvo que crear de cero, se usó el que ya estaba por defecto añadiéndole los campos personalizados creados con anterioridad para que se pudieran usar desde el portal de autoservicio.

Finalmente también se editó el esquema de prioridades, el cual permite asociar un proyecto en particular a un subconjunto de prioridades [17]. Se implementó el esquema para priorizar las incidencias propio de la empresa cliente.

El resto de campos de configuración de momento se dejó por defecto, aunque se volverá a ellos.

6.3. Configuración Grupos Resolutores

Para la creación de los Grupos Resolutores se crearon seis proyectos del tipo “Software”, a los cuales el proyecto CAU escalará los distintos tipos de peticiones para que ellos las gestionen.

Una vez creados hay que configurarlos para que se adapten a los requerimientos establecidos. Los pasos para realizar dicha configuración son los mismos que cuando se configuró el proyecto CAU aunque hay algunas diferencias.

Se creó un esquema de tipos de incidencias para los seis proyectos, ya que todos harán funciones similares. La diferencia con el esquema del proyecto CAU es que en este hay un nuevo tipo de incidencia, el tipo Tarea. Las incidencias de este tipo se usan para representar trabajo que necesita ser realizado y servirán para la Gestión de Proyectos.

Se creó también un flujo de trabajo para los seis proyectos, con estados similares al del proyecto CAU. También se les asignó a dichos proyectos el esquema de prioridades que usa la organización cliente. El resto de configuraciones de momento se dejaron por defecto ya que no es necesario personalizarlas todavía.

6.4. Automatización y SLAs

Una vez creada la configuración básica de los proyectos, se procedió a la creación de reglas de automatización y a la definición de SLAs (*Service Level Agreement*), o Acuerdos a Nivel de Servicio. Los SLA definen el nivel de servicio que espera un cliente de un proveedor, estableciendo las métricas a través de las cuales se mide el servicio y si hay penalización si no se cumplen los niveles de servicio preestablecidos [18].

De las reglas de automatización que se tuvieron que crear se diferencian dos tipos; las por defecto y las creadas mediante un *plugin*. Las primeras se pueden crear desde la misma configuración de un proyecto del tipo “Service Desk” con una herramienta llamada Automatización.

Sin embargo, al usar el tipo de *hosting* Data Center en el proyecto, las opciones de automatización que existen por defecto son bastante limitadas, así que para la creación de reglas más complejas, se necesitó usar un *plugin* llamado Automation for Jira - Data Center and Server [19] el cual amplía las opciones de automatización.

6.4.1. Automatización por defecto

La primera regla de automatización que se creó identifica desde el proyecto CAU cuando se crea una petición y actualiza su estado a “Escalado a Grupo Resolutor”. De esta forma, cuando el cliente crea una petición desde el portal de autoservicio, al cabo de un rato verá que el estado de la misma cambia a “Escalado a Grupo Resolutor”.

A continuación se creó el *script* que realmente hace que las peticiones creadas en el portal se autoescalen a los proyectos de los Grupos Resolutores. Para hacer esto, se necesitó otro *plugin* de Atlassian Marketplace llamado ScriptRunner for Jira [20], el cual permite la creación de *scripts* personalizados utilizando el lenguaje de programación de Apache Groovy.

Se usó el *plugin* para la definición de *Post Functions*. Las *Post Functions* son funciones asociadas a flujos de trabajo, las cuales realizan acciones automatizadas cuando una incidencia transiciona a un nuevo estado. Jira por defecto ya proporciona una serie de *Post Functions*, pero este *plugin* amplía el catálogo de las mismas y además permite usar *scripts* para hacerlas mucho más complejas.

Se crearon seis *Post Functions* distintas, una para cada Grupo Resolutor. Estas funciones se ejecutan automáticamente cuando una petición actualiza su estado a “Escalado a Grupo Resolutor” (esto se hace de forma automática con la regla de automatización anterior). Cada una de las funciones tiene un pequeño *script*, el cual comprueba de que tipo de petición se trata y en función del tipo se clona a un proyecto Grupo Resolutor u otro. En la Figura 5 se puede ver un esquema del funcionamiento:

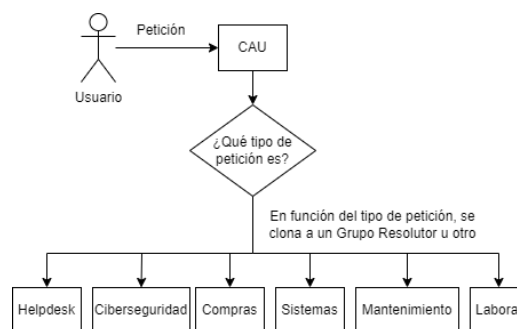


Fig. 5: Escalado peticiones Grupos Resolutores

Ahora se clonan las peticiones del portal de autoservicio a los proyectos Grupo Resolutor. Sin embargo si alguien que forma parte del Grupo Resolutor se pone a trabajar en una incidencia y la resuelve, esto no se ve reflejado en la incidencia del proyecto CAU, lo que significa que para el usuario el estado sigue siendo “Escalado a Grupo Resolutor” cuando la incidencia ya está resuelta.

Para ello se creó otra regla de automatización en la cual se especificó que si se transiciona una incidencia enlazada (al clonar una incidencia se crea un enlace de tipo *cloned by*), se actualice el estado de la incidencia original.

6.4.2. Plugin automatización

Como se comentó anteriormente, las opciones de automatización que proporciona Jira Data Center por defecto no posibilitan la creación de todas las reglas y funcionalidades que se querían implementar, así que a partir de aquí se usó el *plugin* Automation for Jira - Data Center and Server.

Mediante el *plugin* se creó una regla para sincronizar los comentarios entre las incidencias originales y las clonadas en los Grupos Resolutores. En la Figura 6 se puede ver dicha funcionalidad:



Fig. 6: Sincronización comentarios

Esta funcionalidad es muy importante ya que si un usuario escribe un comentario en su petición desde el portal, por ejemplo que no está satisfecho con la resolución de la misma, este comentario solo se vería reflejado en el ticket original (el del proyecto CAU). Mediante esta regla, el comentario se clona en el ticket correspondiente al Grupo Resolutor para que el responsable del mismo pueda hacer algo al respecto. Y lo mismo viceversa, si el responsable del ticket escribe algún comentario en la incidencia del Grupo Resolutor, dicho comentario se clona a la incidencia original y el usuario lo puede ver desde el portal. Además, se habilitó la opción para que cuando esta regla se ejecute envíe un correo electrónico y así notificar a los interesados.

Se creó otra regla de automatización la cual sincroniza las prioridades entre las incidencias clonadas y las originales. Esta regla es necesaria ya que aunque por defecto al crear una petición desde el portal a esta se le asigna una prioridad, el encargado de resolver dicha petición puede razonar que esta prioridad no es adecuada y que es, por ejemplo, más

prioritaria de lo que se estipuló. Dicho responsable actualiza la prioridad de la petición, y esta prioridad también se ve actualizada en la petición original.

El objetivo que se busca al crear estas reglas es que los Grupos Resolutores no tengan que acceder a las incidencias originales (las del proyecto CAU), sino que desde el proyecto Grupo Resolutor puedan gestionar todas las incidencias, cambiar la prioridad de las mismas y asignárselas a ellos o a quién crean correspondiente dentro del Grupo Resolutor.

Para cumplir con este objetivo se creó otra regla de automatización. Esta detecta cuando un usuario crea una petición en el portal de autoservicio y envía un correo electrónico con los campos de dicha petición a los miembros del Grupo Resolutor a los que va dirigida y así notificarles.

Para implementar esta funcionalidad se configuró un servidor de correo SMTP para enviar el correo saliente, ya que por defecto Jira no lo proporciona. Un servidor SMTP sirve para transportar correo electrónico de un destinatario a otro y funciona a través de un proceso de autenticación: si el servidor SMTP confirma las identidades del remitente y del destinatario, el envío se realiza [21].

Para ello se creó una cuenta de outlook llamada *empresatfg@outlook.es* ya que Microsoft proporciona un servicio SMTP.

Aún queda una tarea por realizar para cumplir el objetivo de que los Grupos Resolutores no tengan que ver las incidencias del proyecto CAU. Dicha tarea se verá al final de la siguiente subsección ya que tiene que ver con los SLAs.

6.4.3. Service Level Agreements (SLAs)

Se crearon dos SLAs distintos. El primero establece el tiempo de resolución de las distintas peticiones e incidencias que los usuarios crearán a través del portal, estipulando diferentes tiempos en función de si es un incidente o una petición, y también en base a la prioridad. Estos tiempos están establecidos por la empresa cliente. En la Tabla 2 se muestran los tiempos del SLA:

Prioridad \ Tipo	Incidencia	Petición
Baja	4h	1d
Normal	2h	12h
Alta/Crítica	1h	4h

TABLA 2: Tiempo SLA Resolución

El segundo de los SLAs que se crearon especifica el tiempo máximo que puede transcurrir antes de cerrar una incidencia resuelta. Se estableció que las incidencias resueltas pueden seguir abiertas durante 24 horas antes de ser cerradas, ya que a lo mejor el cliente la quiere reabrir porque no está satisfecho con la resolución.

Finalmente se creó una regla de automatización usando el *plugin* de Automation para cerrar de forma automática las incidencias resueltas cuando queden 30 minutos para que venza el SLA que se especificó anteriormente. Esto es necesario ya que es posible que a los responsables de las incidencias se les olvide cerrarlas una vez resueltas y mediante esta regla la empresa no incumple ningún SLA.

Cabe destacar que lo usual es crear un calendario con los

horarios de la empresa, para que si por ejemplo un usuario crea una petición en el portal un sábado a las 3 de la madrugada, el SLA empieza a contar el tiempo a partir del horario de trabajo de la empresa y no venza sin que nadie haya estado trabajando para poder solucionar dicha petición. Sin embargo la empresa cliente ofrece un servicio a sus usuarios 24/7 así que se implementaron los SLA con el calendario por defecto, el cual no tiene en cuenta horarios ni festivos.

Queda por solucionar el hecho que por defecto los SLA solo se muestran en los proyectos del tipo “Service Desk”, por lo que no se visualizarán desde los Grupos Resolutores ya que son proyectos del tipo “Software”. Los usuarios que actúan como Grupos Resolutores tendrían que abrir la incidencia original para ver el tiempo que le queda al SLA para vencer, lo cual no resultaría demasiado cómodo.

Para solucionar este inconveniente se utilizó otro *plugin* de Atlassian Marketplace llamado Time to SLA [22], el cual amplía mucho las opciones de configuración de los SLAs y permite a todo tipo de proyectos tener los suyos propios.

Usando este *plugin*, desde los proyectos Grupos Resolutores se pueden ver los SLA de los distintos tickets y así los usuarios que los resuelven no tienen que buscar la incidencia original para asegurarse de cumplir los SLA.

De esta forma se consiguió que los usuarios de los Grupos Resolutores no tengan ninguna necesidad de acceder a las incidencias originales del proyecto CAU ya que con todas las funcionalidades que se han añadido, pueden trabajar directamente sobre las incidencias clonadas.

6.5. Creación de Usuarios y Grupos

A parte del usuario administrador, el cual tiene acceso a las aplicaciones de Jira Software y Jira Service Management, se crearon 8 usuarios nuevos para probar las distintas funcionalidades que se requieren de la aplicación creada.

Se crearon seis usuarios para gestionar cada usuario uno de los proyectos Grupo Resolutor. A estos usuarios se les dio acceso a la aplicación de Jira Software ya que la necesitarán para poder ver y gestionar las incidencias clonadas en los proyectos Grupo Resolutor.

Además se creó otro usuario usando el correo personal, el cual también tiene acceso a la misma aplicación comentada anteriormente. Este usuario se usó para testear que efectivamente les llega una notificación al correo electrónico a los responsables de los Grupos Resolutores cuando se les hace una petición.

Se decidió gestionar de esta forma las licencias (usar una de Jira Service Management y siete de Jira Software) ya que las primeras son prácticamente el doble de caras que las segundas. Para este trabajo no hubiese afectado ya que solo se crearon unos pocos usuarios, sin embargo en corporaciones grandes como la empresa cliente, este tipo de gestión puede ahorrar mucho presupuesto.

También se creó un usuario base del portal de autoservicio, el cual no tiene permisos para usar ninguna de las dos aplicaciones de Jira y solo tiene acceso al portal. Este tipo de usuarios se pueden crear manualmente como se hizo aquí, o cualquiera que acceda al portal se puede crear una cuenta con las mismas características.

A continuación se crearon dos esquemas de permisos, uno para el proyecto CAU y otro para los 6 proyectos de los Grupos Resolutores. Un esquema de permisos permite aplicar diferentes niveles de acceso a grupos, roles e individuos en un proyecto específico. Habilita o deshabilita acciones específicas para los usuarios de un proyecto [23].

Se editó el apartado de Usuarios y Roles en cada uno de los proyectos. Se usaron dos roles, el rol de “Administradores”, el cual puede editar, eliminar y gestionar los proyectos, y el rol de “Service Desk Team”, que no puede editar proyectos, pero si ver las incidencias creadas y gestionarlas.

En el proyecto CAU como Administrador de proyectos se añadió el usuario administrador que es el que se ha usado durante todo el trabajo para realizar todas las configuraciones. Como rol de “Service Desk Team” no se asignó a ningún usuario ya que los usuarios que actúan como Grupos Resolutores no deberían ver las incidencias originales y solo trabajar en las clonadas.

En los proyectos Grupo Resolutor se añadió como Administrador también el usuario administrador y como Service Desk Team a los usuarios correspondientes que se quiere que actúen como Grupo Resolutor y resuelvan las incidencias pertinentes de cada uno de los proyectos.

6.6. Gestión de Proyectos

Como se comentó en la sección de Toma de Requisitos, los usuarios pueden realizar ciertas peticiones en el portal que son significativamente más complejas que otras, y requieren de la creación de un proyecto para ser solventadas.

Para ello se creó en los flujos de trabajo de los proyectos Grupo Resolutor un estado llamado Escalado a Proyecto, al cual los Grupos Resolutores transicionarán las incidencias o peticiones que consideren que son lo bastante complejas.

Además, se crearon seis reglas de automatización, una para cada proyecto Grupo Resolutor, las cuales serán las que creen un proyecto a partir de una incidencia cuando estas actualicen su estado a Escalado a Proyecto.

Para la creación de un proyecto de forma automática, estas reglas envían un *webhook* a la *rest api* de Jira. Un *webhook* se trata de una solicitud HTTP POST, mientras que la *rest api* de Jira se trata de una aplicación que proporciona una forma de interactuar con el servidor de Jira de forma remota, el cuál proporciona funcionalidades como la que interesa a este trabajo, que es la creación de proyectos [24]. En la Figura 7 se muestra el proceso de creación de un nuevo proyecto automáticamente:

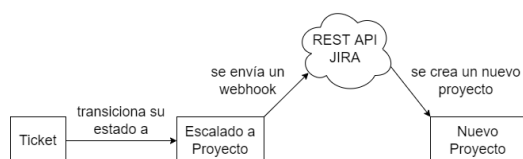


Fig. 7: Creación nuevo proyecto

Estos *webhooks* enviados cuentan con todos los datos que necesita un proyecto para ser creado, como el nombre de dicho proyecto o quién es su administrador. Al crear los proyectos cuando una incidencia transiciona su estado, se pueden usar variables con información de dicha incidencia

para la creación del proyecto, así que datos como por ejemplo el responsable de la incidencia, se usaron para asignar el administrador del proyecto.

Se usó también un *plugin* llamado Tempo Budgets para planificar y gestionar el coste de los proyectos de la empresa cliente [25]. A continuación se explica un ejemplo del uso de este *plugin* para la Gestión de Proyectos:

Primero un usuario crea una petición en el portal de auto-servicio del tipo Solicitud de Cambio la cual se trata de una nueva funcionalidad para el Jira de la empresa que se tiene que desarrollar. El Grupo Resolutor correspondiente decide que la petición es bastante compleja así que actualizan el estado del ticket para que se cree un proyecto de forma automática usando la regla comentada con anterioridad.

A continuación usando el *plugin* de Tempo Budget se creó un folio llamado Ejemplo Presupuesto para el proyecto creado, en el cual se define el tipo de moneda (EUR), la fecha de inicio y finalización, y el proyecto al que afecta, en este caso se escogió el proyecto creado a partir de la solicitud de cambio creada por el usuario en el portal.

Dicho *plugin* permite planificar el coste de los recursos humanos y otros costes (gastos no relacionados con salarios). Se asignó al proyecto un usuario, y se definió que trabajaría en ese proyecto el 100 % de tiempo que este durase, en este caso 184 horas (8 horas los días entre semana), especificando también cuánto cobra ese empleado.

Cada vez que el usuario trabaja en el proyecto, ha de imputar las horas trabajadas, y de forma automática se calcula cuanto equivalen esas horas en costes reales del proyecto. También se planificó un coste al inicio del proyecto en referencia a la licencia de un supuesto *plugin* y además un coste adicional referente a un gasto inesperado.

El *plugin* permite ver una comparativa del coste planificado y el coste real del presupuesto establecido. El Anexo 4 muestra el gráfico generado sobre los costos esperados y los reales. Se puede ver que al inicio del proyecto hay un gasto equivalente a la licencia de un plugin, además de ir subiendo el coste progresivamente debido al salario del trabajador. También se puede ver un punto en el gráfico en el que los costes reales del proyecto superan a los planificados debido al gasto inesperado que se añadió.

Para finalizar esta sección, se usó una funcionalidad que viene incorporada en Jira que son las Pizarras. Estas sirven para mostrar distintos indicadores de los proyectos y las puede usar la dirección para ver y medir el rendimiento de su organización.

En el Anexo 5 hay un ejemplo de un diagrama de flujo acumulado generado con esta funcionalidad, el cual muestra los estados de las incidencias creadas a través del portal de autoservicio.

7 CREACIÓN DE ENCUESTA

Para completar el último de los objetivos específicos, se creó una encuesta usando los Formularios que proporciona Google [26], para medir si se había cumplido el objetivo general que se especificó al inicio del proyecto.

Inicialmente esta encuesta estaba pensada para ser respondida por los empleados de la empresa cliente, sin em-

bargo por diversos motivos no se pudo hacer de esta forma. Se decidió pasar el formulario a 13 empleados de Quabu Solutions SL ya que ya tenían cierta familiaridad trabajando con Jira.

La encuesta consta de dos preguntas, la primera es obligatoria y se responde con una puntuación del 0 al 10, siendo 10 un “Perfecto” y 0 un “Desastre”. Las dos preguntas se pueden ver en el Anexo 6.

Para realizar la encuesta se crearon usuarios en la aplicación para todos los miembros del equipo de Quabu Solutions. Además, se publicó la ip y un puerto de la máquina virtual donde está alojada la aplicación de Jira para que cualquiera se pudiese conectar desde cualquier sitio. Finalmente y para que los usuarios no se tuviesen que conectar a la aplicación a través de una ip, se usó una herramienta llamada No-IP la cual permite crear dominios para redireccionar conexiones. Usando esta herramienta se creó un host llamado tfg.hopto.org el cual redirecciona las conexiones hacia la ip de la máquina [27].

Se esperaba conseguir mínimo una puntuación de 7 puntos sobre 10 en la primera pregunta para asegurar el cumplimiento del objetivo del proyecto y se obtuvo un resultado equivalente a un 9’1, por lo que se cumplió con el objetivo. En el Anexo 7 se pueden las puntuaciones de cada usuario.

Las sugerencias de mejora proporcionadas por los encuestados se comentan en la sección de Líneas Futuras.

8 CONCLUSIONES

Para empezar hay que responder a la pregunta de si se ha conseguido satisfacer el objetivo general y los cuatro objetivos específicos que se establecieron al inicio del proyecto.

El primero era conocer el alcance del proyecto a realizar (usuarios, permisos, tipos de peticiones, lógica de la aplicación, etc). Dicho objetivo se cumplió ya que se hizo una fase de toma de requisitos en la que se definió toda la aplicación.

El segundo y el tercero trataban respectivamente de crear el borrador de la interfaz del portal y desarrollar la aplicación. El tercer objetivo fué el que más trabajo y tiempo ha requerido, pero finalmente se ha obtenido una aplicación con los requisitos que se definieron previamente así que estos dos objetivos también se cumplieron.

Finalmente el cuarto de los objetivos específicos era verificar el objetivo general mediante una encuesta. Los resultados de la encuesta indican que el objetivo se ha cumplido ya que se ha superado la nota mínima requerida de 7.

Uno de problemas que ha surgido durante el desarrollo de la aplicación ha sido el coste monetario ya que se usan en total cinco *plugins* de pago. Esto se ha podido contrarrestar un poco estructurando el proyecto de tal manera que solo se necesite una licencia de Jira Service Management, las cuales son prácticamente el doble de caras que las de Jira Software. Además a la hora de seleccionar los *plugins* se ha hecho una comparativa de precios, para obtener la funcionalidad requerida de la forma más económica posible.

El otro gran problema que surgió fué el de la configuración del servidor SMTP. Inicialmente se creó una cuenta de gmail ya que Google ofrece un servicio SMTP para enviar correo, sin embargo los mensajes eran detectados como

spam y Google bloqueó el servicio de la cuenta. Usando outlook se consiguió solucionar.

También comentar que tanto las aplicaciones utilizadas (Jira Software y Jira Service Management) como la metodología usada (Kanban) han estado acertadas y con ellas se ha podido realizar el trabajo previsto cómodamente. Sin embargo si se tuviera que repetir el proyecto, probablemente se usaría Jira con el tipo de *hosting* en el Cloud ya que cuenta con una mayor comunidad de usuarios y en la actualidad las soluciones en el Cloud parece que son el futuro.

Para terminar las conclusiones hace falta recalcar la importancia que ha tenido para el proyecto el hecho de disponer de buffers de tiempo para imprevistos. En cualquier proyecto pueden surgir problemas o retrasos, y si no se cuenta con tiempo extra es posible que no se pueda entregar el proyecto a tiempo. En este proyecto se tuvieron que usar estos buffers para imprevistos en las fases de Desarrollo de la app y en la Creación de la encuesta.

9 LÍNEAS FUTURAS

En esta sección se comentan una serie de mejoras, detectadas gracias a la encuesta realizada, que se podrían incluir en el proyecto para mejorar la calidad del mismo.

La primera de todas es usar Confluence para documentar como se creó la aplicación y como se implementaron las diferentes actualizaciones de la misma. De esta forma si un equipo quiere buscar documentación sobre cierta funcionalidad de la aplicación, la puede encontrar fácilmente, además de ayudar al futuro mantenimiento de la misma.

Otra de las mejoras que se podrían añadir se trata de redirigir la url del centro de asistencia al portal de autoservicio. Por defecto Jira crea un centro de asistencia, en el cual hay enlaces a los diferentes portales de autoservicio que se hayan creado, pero como en este proyecto solo hay un portal, esa página no tiene ninguna utilidad.

También sería interesante de cara al *manager* de servicio tener pizarras que mostrasen la carga de trabajo por técnico y la media de satisfacción de los usuarios con la aplicación.

Finalmente otras mejoras que se podrían realizar serían incluir más detalle en los estados de las peticiones escaladas (el usuario ve que su ticket está en el estado “Escalado a Grupo Resolutor” pero no sabe más) y añadir más campos a la hora de realizar ciertas peticiones al portal de autoservicio (que algunas peticiones estuviesen más detalladas).

AGRADECIMIENTOS

Primero agradecer a la tutora del trabajo de final de grado, Ana Oropesa, por el gran trabajo de tutelaje que ha hecho, dando ideas y recomendaciones, y sobre todo, guiándome en este proceso.

También agradecer enormemente a toda la empresa de Quabu Solutions SL, ya que sin ellos este trabajo no habría sido posible, en especial a Adrià Garrofé y Mauricio Méndez que son los que más me han ayudado.

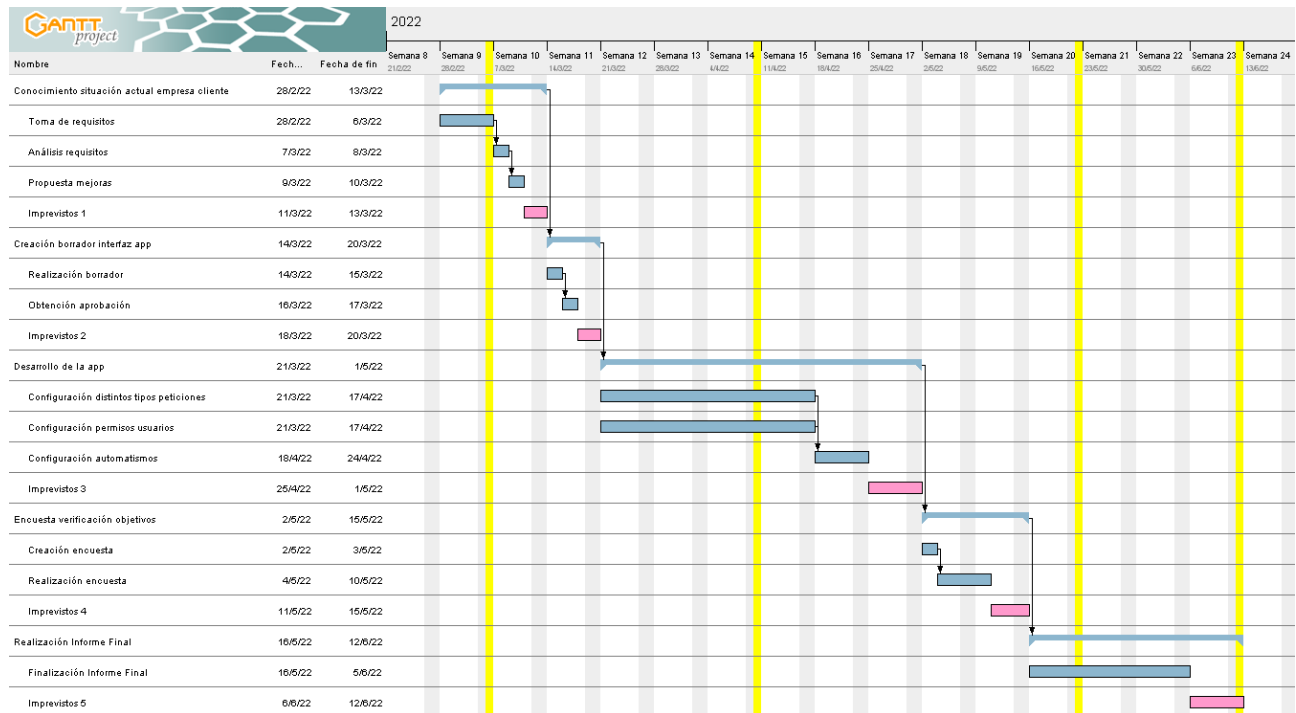
Finalmente, pero no menos importante, agradecer a toda la familia y amigos sobre todo por el apoyo durante estos meses que ha durado el trabajo.

REFERENCIAS

- [1] Quabu. (2021, 1 Febrero). Quabu Home Page - Quabu [Online]. Disponible en: <https://quabu.eu/>
- [2] C. Ayala. (2021, 25 Mayo). SPOC - Punto único de Contacto [Online]. Disponible en: <https://www.servicetonic.com/es/service-desk/spoc-punto-unico-de-contacto/>
- [3] E. Meardon. (2021, 31 Marzo). ¿Qué es un diagrama de Gantt? [Online]. Disponible en: <https://www.atlassian.com/es/agile/project-management/gantt-chart>
- [4] GanttProject. (2021, 1 Octubre). GanttProject - Free Project Management Application [Online]. Disponible en: <https://www.ganttproject.biz/>
- [5] S. Garrido. (2021, 9 Diciembre). Qué son las metodologías ágiles y cuáles son sus ventajas empresariales [Online]. Disponible en: <https://www.iebschool.com/blog/que-son-metodologias-agiles-agile-scrum/>
- [6] Kanban Tool. (2014, 17 Noviembre). Metodología Kanban [Online]. Disponible en: <https://kanbantool.com/es/metodologia-kanban>
- [7] Atlassian. (2011, 25 Octubre). Jira | Software de seguimiento de proyectos e incidencias [Online]. Disponible en: <https://www.atlassian.com/es/software/jira>
- [8] Atlassian. (2019, 31 Enero). Resumen de Jira | Productos, proyectos y alojamiento [Online]. Disponible en: <https://www.atlassian.com/es/software/jira/guides/getting-started/overview>
- [9] Atlassian. (2017, 28 Junio). What is Jira Service Management? | Jira Service Management Cloud [Online]. Disponible en: <https://support.atlassian.com/jira-service-management-cloud/docs/what-is-jira-service-management/>
- [10] Contegix. (2021, 3 Febrero). Comparing Atlassian Cloud vs. Server vs. Data Center [Online]. Disponible en: <https://www.contegix.com/resources/library/atlassian-cloud-vs-server>
- [11] Adobe. (2021, 2 Mayo). Free Random Logo Generator | Adobe Creative Cloud Express [Online]. Disponible en: <https://www.adobe.com/express/create/logo/random>
- [12] Atlassian. (2022, 10 Febrero). Theme Extension for Jira Service Management | Atlassian Marketplace [Online]. Disponible en: <https://marketplace.atlassian.com/apps/1220300/theme-extension-for-jira-service-management?hosting=datacenter&tab=overview>
- [13] Atlassian. (2021, 10 Mayo). Adding custom fields | Administering Jira applications [Online]. Disponible en: <https://confluence.atlassian.com/adminjiraserver/adding-custom-fields-1047552713.html>
- [14] Atlassian. (2016, 30 Agosto). What are issue type schemes? [Online]. Disponible en: <https://support.atlassian.com/jira-cloud-administration/docs/what-are-issue-type-schemes/>
- [15] Atlassian. (2020, 23 Junio). Working with workflows [Online]. Disponible en: <https://confluence.atlassian.com/adminjiraserver/working-with-workflows-938847362.html>
- [16] Atlassian. (2020, 23 Junio). Associating a screen with an issue operation [Online]. Disponible en: <https://confluence.atlassian.com/adminjiraserver/associating-a-screen-with-an-issue-operation-938847289.html>
- [17] Atlassian. (2020, 23 Junio). Associating priorities with projects [Online]. Disponible en: <https://confluence.atlassian.com/adminjiraserver/associating-priorities-with-projects-939514001.html>
- [18] S. Overby, L. Greinner y L. Paul. (2017, 5 Julio). What is an SLA? Best practices for service-level agreements [Online]. Disponible en: <https://www.cio.com/article/274740/outsourcing-sla-definitions-and-solutions.html>
- [19] Atlassian. (2022, 6 Abril). Automation for Jira - Data Center and Server | Atlassian Marketplace [Online]. Disponible en: <https://marketplace.atlassian.com/apps/1215460/automation-for-jira-data-center-and-server?hosting=datacenter&tab=overview>
- [20] Atlassian. (2022, 28 Marzo). ScriptRunner for Jira | Atlassian Marketplace [Online]. Disponible en: <https://marketplace.atlassian.com/apps/6820/scriptrunner-for-jira?hosting=cloud&tab=overview>
- [21] B. Tejedor. (2017, 20 Marzo). ¿Qué es un Servidor SMTP? Qué significa y cómo usarlo para enviar email - Mailjet [Online]. Disponible en: <https://es.mailjet.com/blog/news/servidor-smtp/#queues>
- [22] Atlassian. (2022, 11 Abril). Time to SLA | Atlassian Marketplace [Online]. Disponible en: <https://marketplace.atlassian.com/apps/1211843/time-to-sla?hosting=datacenter&tab=overview>
- [23] Atlassian. (2021, 20 Julio). JIRA Permissions General Overview [Online]. Disponible en: <https://confluence.atlassian.com/jirakb/jira-permissions-general-overview-625902332.html>
- [24] Atlassian. (2017, 9 Noviembre). REST APIs [Online]. Disponible en: <https://developer.atlassian.com/server/jira/platform/rest-apis/>
- [25] Atlassian. (2022, 12 Abril). Tempo Budgets | Atlassian Marketplace [Online]. Disponible en: <https://marketplace.atlassian.com/apps/1211259/tempo-budgets?hosting=datacenter&tab=overview>
- [26] Google. (2014, 17 Octubre). Google Forms: solución gratuita para crear formularios online [Online]. Disponible en: <https://www.google.es/intl/es/forms/about/>
- [27] NO IP. (2021, 10 Septiembre). DNS Dinámico Gratis [Online]. Disponible en: <https://www.noip.com/es-MX>

APÉNDICE

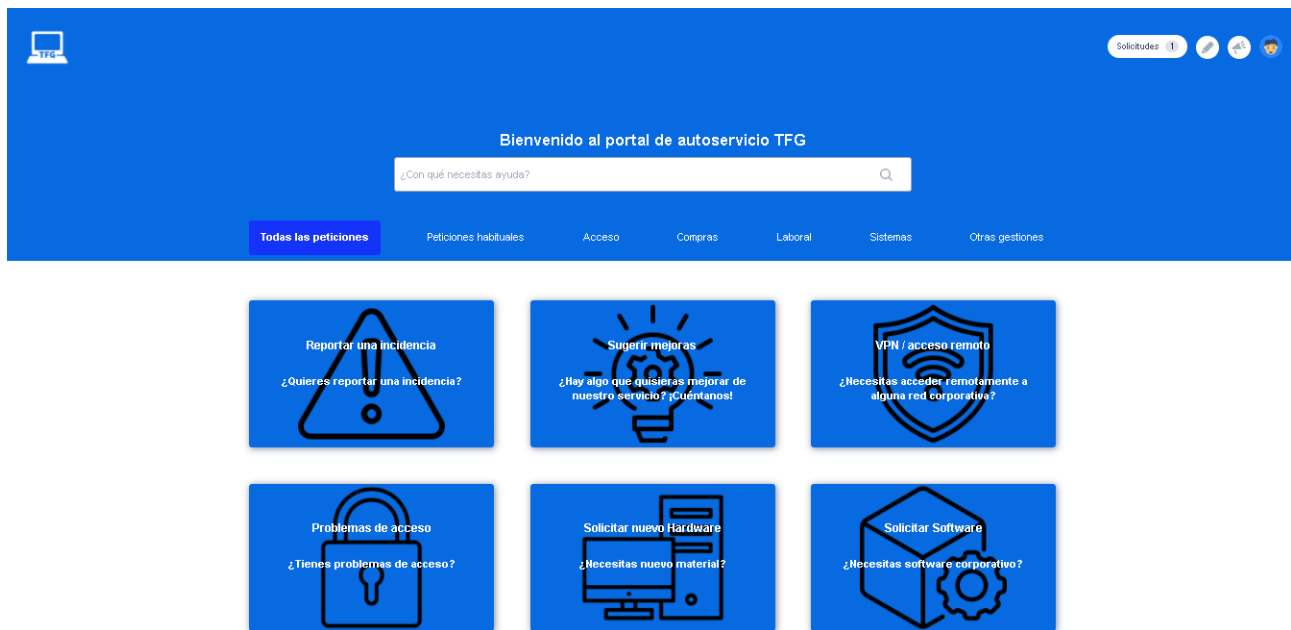
A.1. Diagrama de Gantt



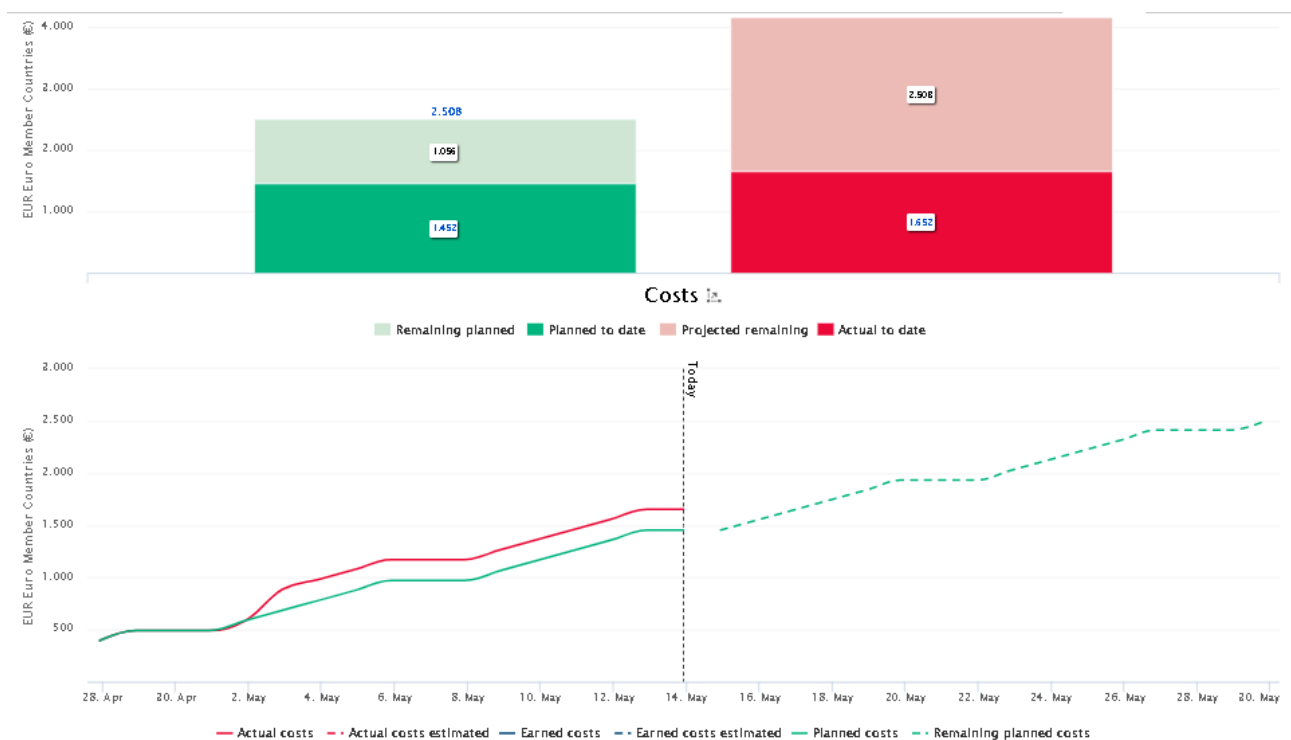
A.2. Interfaz portal de autoservicio antes



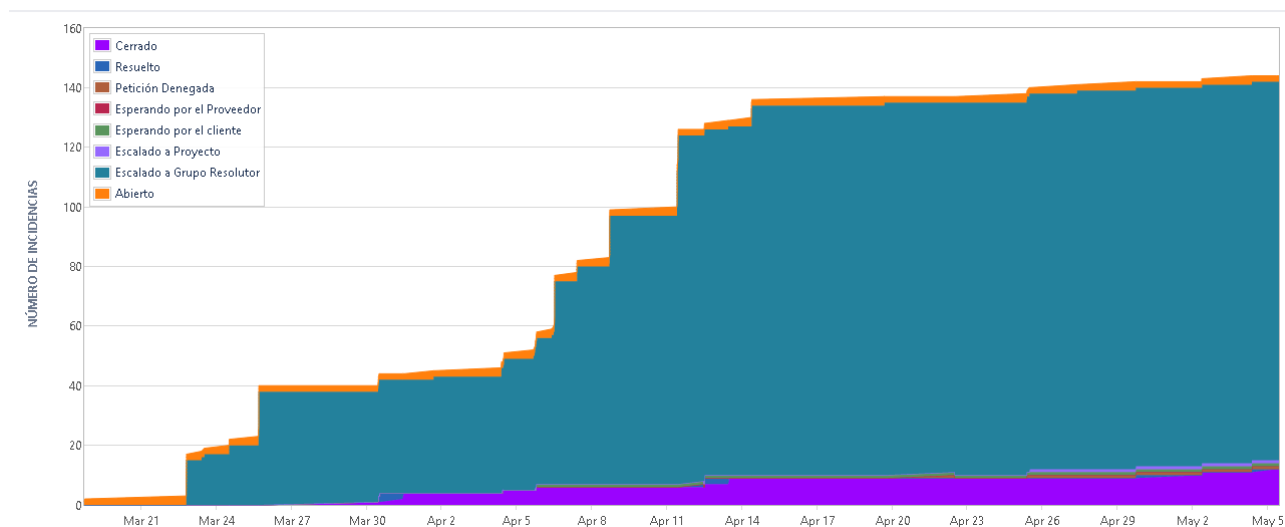
A.3. Interfaz portal de autoservicio después



A.4. Comparativa costes tempo budgets



A.5. Diagrama de flujo estados incidencias



A.6. Preguntas encuesta

Puntúa del 0 al 10, siendo 0 la peor nota y 10 la mejor, la aplicación creada teniendo en cuenta si esta ha conseguido cumplir el objetivo de facilitar la gestión de peticiones de la empresa:

0	1	2	3	4	5	6	7	8	9	10
☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐

¿Se te ocurre alguna mejora para esta aplicación?

Tu respuesta

A.7. Resultados encuesta

Puntúa del 0 al 10, siendo 0 la peor nota y 10 la mejor, la aplicación creada teniendo en cuenta si esta ha conseguido cumplir el objetivo de facilitar la gestión de peticiones de la empresa:



13 respuestas

