

Gestión de Cambios

- 1 [Proceso: Gestión de Cambios](#)
- 2 [Objetivo](#)
- 3 [Alcance](#)
- 4 [Roles](#)
- 5 [Flujo del proceso](#)
 - 5.1 [Gestión de la demanda](#)
 - 5.2 [Ciclo de vida en cascada](#)
 - 5.3 [Ciclo de vida ágil](#)

? Archivo adjunto desconocido

Proceso: Gestión de Cambios

Objetivo

Modelar la forma en la que se responde a los requisitos cambiantes del negocio, asegurando que los cambios se registran y se evalúan con el objetivo de dar valor, reduciendo al mínimo el número de incidencias y las interrupciones del servicio.

Alcance

Este proceso se aplica a la gestión de los cambios en los productos o aplicativos centralizados que se gestionan en la STIC. Dichos productos siempre deben estar dados de alta en el sistema de gestión de la configuración (CMS) como elementos de configuración software de tipo aplicación.

Existen dos metodologías en la STIC para gestionar los cambios en los aplicativos centralizados:

- Metodología en cascada
- Metodología ágil

Roles

Independientemente de la metodología que se use, los roles que intervienen en este proceso son:

Rol	Descripción
Responsable de producto	Es el máximo responsable del producto desde el punto de vista de su gestión.
Responsable funcional	Es el responsable de la evolución del producto, indicando lo que tiene o no valor para el mismo.
Responsable de sistemas	Responsable de la o las plataformas en la que se despliega la aplicación.
Responsable de área	Responsable de la gestión económica de los productos y de la asignación de recursos a los evolutivos.
Proveedor	Responsable de acometer los desarrollos de los evolutivos de los productos y de corregir las posibles incidencias.
Proveedor sistemas	Responsable de construir las plataformas en las que se despliega la aplicación, así como de las posibles implantaciones de los evolutivos y correctivos en las mismas.
Oficina Técnica de Calidad (OCA)	Responsable de hacer la verificación de las entregas, tanto las que se despliegan, como las que no.

Flujo del proceso

Gestión de la demanda

El primer paso consiste en gestionar la demanda que crean los usuarios en formas de mejoras en ayudaDIGITAL. Hay una primera revisión por parte del nivel 1 de ayudaDIGITAL para cribar aquello que entienden que no es una mejora.

Todas esas mejoras llegan al Responsable funcional quien debe seleccionar entre aquellas que aportan valor a su producto y aquellas que no lo aportan, bien porque no vayan de acuerdo con la estrategia seguida por la STIC o simplemente porque aún siendo interesante, se le haya podido ocurrir antes a otro usuario. Por tanto, es necesario "aprobar" aquellas mejoras que aportan valor y clasificarlas en urgencia para que se acometan con mayor o menor premura. Tanto si una mejora se aprueba como si se rechaza, la solicitud del usuario final se cierra enviándole un mensaje diferente tanto si se aprueba como si se rechaza.

Existe la posibilidad de que una de esas mejoras realmente no lo sea o que lo sea pero esté creada para una aplicación incorrecta. En ese caso, Responsable de producto y/o Responsable funcional podrán escalar la mejora de manera que ayudaDIGITAL la retifique o la asigne a la aplicación correcta. En este último caso, volverán de nuevo a enviarla a la herramienta de gestión para que siga el flujo.

- El mensaje que se envía al usuario si se aprueba la mejora es

"Desde el equipo responsable de la aplicación le agradecemos su colaboración. Tras analizar su propuesta de mejora, consideramos que ésta aporta valor y va en consonancia con las líneas estratégicas marcadas. Por tanto, será incluida en próximas versiones. Un cordial saludo."

- El mensaje que se envía al usuario si se rechaza la mejora es

- Si se rechaza por ser no compatible con la estrategia de evolución

"Desde el equipo responsable de la aplicación te agradecemos tu colaboración. Analizada tu solicitud, sentimos comunicarte que no podemos atenderla en el momento actual, ya que no se ajusta a la estrategia diseñada para la evolución de la aplicación. Seguimos trabajando para dar respuesta a todas las necesidades de la organización y esperamos que las mejoras que se introduzcan faciliten tu labor. Un cordial saludo."

- Si se rechaza por tratarse de una funcionalidad ya detectada

"Desde el equipo responsable de la aplicación te agradecemos tu colaboración. Tras analizar tu solicitud, te comunicamos que tu necesidad ya ha sido detectada anteriormente y será incorporada en próximas versiones. Un cordial saludo."

El proceso consta de los siguientes pasos:

- Primer paso (Responsable de producto y/o Responsable funcional): determinar las mejoras viables y rechazar tanto las no viables como las repetidas y aquellas que entran en el sistema como mejoras y que realmente no lo son.

- Segundo paso (Responsable de producto y/o Responsable funcional): ordenar la demanda aprobada por la urgencia con la que necesita ser acometida.
- Tercer paso (Responsable de producto): Acometer el desarrollo de las mejoras más urgentes y tener como objetivo la implantación de aquellas incidencias que están a la espera de ser solucionadas por parte del usuario. Llegados a este tercer paso, es cuando se distinguen dos formas de gestionar los productos:

- Ciclo de vida en cascada
- Ciclo de vida ágil: descomposición en historias de usuario y creación de sprints

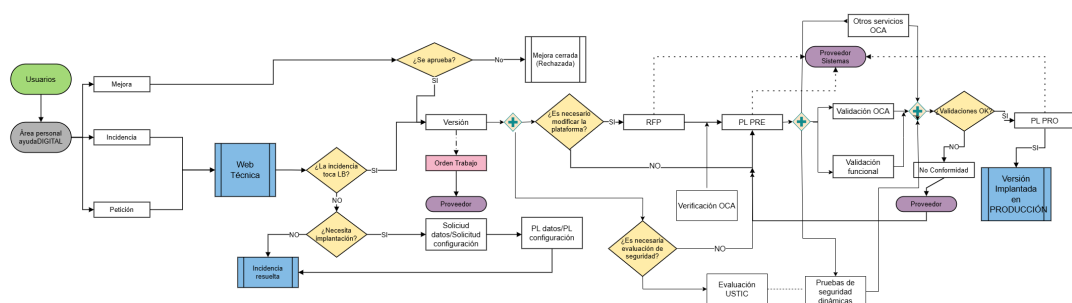
i Las incidencias que modifican la línea base del producto y que por tanto deben ser incluidas en la versión, se envían a la herramienta de gestión (JIRA) desde la de Operación (WT) por parte del Proveedor.

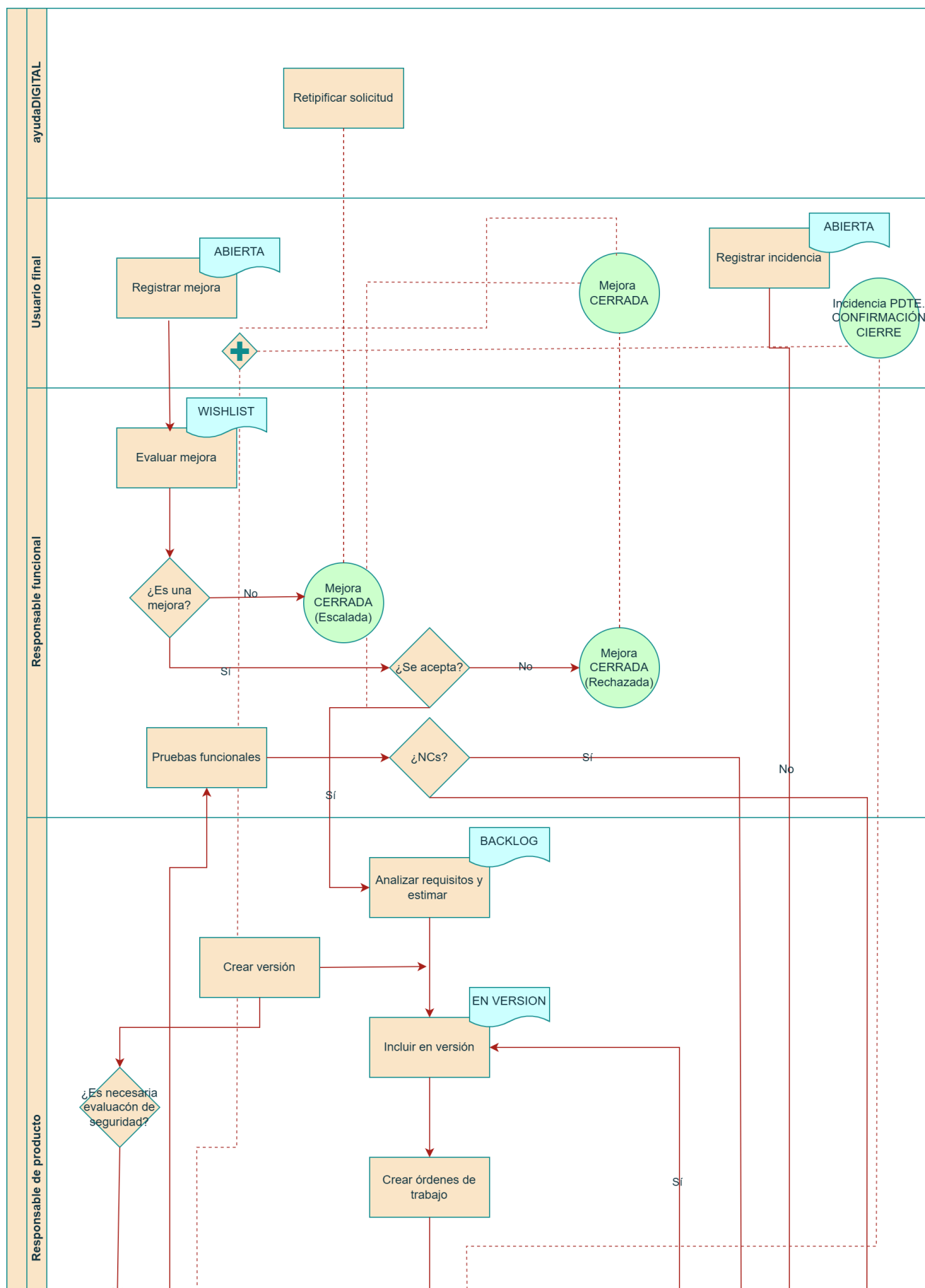
Llegan "preaprobadas" (Backlog), y pueden ser incluidas en la versión tanto por parte de Responsable de producto como por el Proveedor.

Área personal ayudaDIGITAL

WT

JIRA







Ciclo de vida en cascada

La unidad de gestión es la **VERSIÓN**, entendida como un contenedor de alcance que hay que desarrollar e implantar en un tiempo determinado, con un coste asignado y con una calidad máxima.

El alcance de una versión está determinado por **MEJORAS** (lo lógico es incluir aquellas más urgentes) que hay que desarrollar por parte del Proveedor e **INCIDENCIAS** que constituyen errores de la versión del producto en producción que necesitan para ser corregidos una modificación del código y por tanto una implantación.

Veremos más adelante que también pueden formar parte del alcance de una versión fallos que se hayan encontrado en versiones anteriores antes de estar en Producción y que, por necesidades del negocio, no hayan podido ser corregidos en la versión en la que se han detectado y se haya decidido corregirlos en versiones posteriores. Estas incidencias detectadas en entornos no productivos se denominan No Conformidades y pueden por tanto, formar parte del alcance de una versión.

La versión necesita para poder comenzar tener al menos una mejora/incidencia/no conformidad en su alcance. Normalmente para que una versión se comience debe ser aprobada por el Responsable de área correspondiente, aunque existe la opción de que esa aprobación esté delegada en el Responsable de producto.

Una vez que la versión ha comenzado, el Responsable de producto va creando órdenes de trabajo al proveedor para completar las diferentes fases del ciclo de vida: requisitos, análisis, diseño, construcción e implantación.

Estas órdenes tienen establecidos una serie de entregables obligatorios que deben ser aportados por el proveedor y revisados y aceptados por el Responsable de producto.

Además, el Responsable de producto debe aceptar y justificar, si procede, las posibles desviaciones en coste y plazo de ejecución de cada una de esas órdenes de trabajo una vez finalizadas por parte del proveedor.

Cuando el código a desarrollar está listo, el proveedor deberá entregarlo en el repositorio correspondiente como se indica en la [Gestión de lanzamientos](#). Deberá crear una rama identificada con el código de la mejora. Puede ocurrir que el proveedor sea el que compile el código con lo que el procedimiento de entrega cambia. Una vez entregado el código, y también como se detalla en el proceso de [Gestión de lanzamientos](#), deberá solicitarse a la Oficina de Calidad la verificación de que el código es apto para el despliegue y posteriormente hacer la petición de lanzamiento correspondiente para que se despliegue en los entornos necesarios.

Una vez que se produce el despliegue en PRE, la rama de la mejora en el repositorio se cierra.

Cada mejora que se desarrolla debe ser validada antes de su subida a Producción, tanto por parte del Responsable funcional como en muchos casos por parte de la Oficina de Calidad. Cualquier fallo o defecto detectado pasa a registrarse como una No Conformidad (ver proceso de [Gestión de No Conformidades](#)), que deberá ser solucionada por el Proveedor, y que, como hemos comentado antes, puede formar parte del alcance de una versión.

De cara a la entrega de la NC en el repositorio, el flujo de Gitflow funciona igual que se ha indicado en las mejoras.

Hay dos tipos de órdenes de trabajo:

- Órdenes de trabajo con crédito disponible: el Responsable de Producto es quien determina el techo de coste o esfuerzo en HBS requerido para la orden. El Proveedor se limita a, teniendo ese techo de coste asignado, planificar las fechas en las que va a hacer el trabajo. Hay diferentes tipos de órdenes de trabajo con crédito disponible:
 - OTEVS (Estudio de viabilidad del sistema). Son las únicas órdenes de trabajo que pueden crearse con la versión sin haber comenzado, incluso pueden crearse para más de una versión. Sus incurridos no se tienen en cuenta en el cómputo de las HBS de la versión.
 - OTR (Requisitos)
 - OTInmediata
 - OTEquipo (para equipos dedicados que trabajan con esta metodología)
 - PST (Petición soporte Transición): son peticiones de ayuda del área de Proyectos al Proveedor de sistemas
- Órdenes de trabajo con estimación: como su propio nombre indica, es necesario que el Proveedor estime el coste/esfuerzo de la orden y que planifique las fechas en las que podrá ser realizada. Esa estimación deberá ser aprobada por el Responsable de producto, y será realizada por el Proveedor tantas veces hasta que sea finalmente aprobada. Hay diferentes tipos de órdenes de trabajo con estimación:
 - OTA (Análisis)
 - OTD (Diseño)
 - OTC (Construcción)
 - OTAD (Análisis y diseño)
 - OTADC (Análisis, diseño y construcción)
 - OTImpl (Implantación)

El Responsable de producto puede además solicitar servicios a la Oficina de Calidad asociados a esa versión. Estos servicios son:

- Servicios de Testing Temprano (STT):
Revisión de la documentación aportada en las fases ASI y DSI. Se verifican casos de uso, actores, subsistemas, requisitos, precondiciones, casos de pruebas y la trazabilidad entre ellos.

Se valida que tanto en el análisis como en el diseño se cumpla la normativa específica de la organización, así como mantener una trazabilidad completa en el modelo propuesto.

La petición debe hacerse por parte del Responsable de producto al menos un mes antes de la fecha prevista para el comienzo de las pruebas y deberá incluir el modelado del aplicativo en Enterprise Architect incluyendo los ficheros correspondientes, bien en la fase de análisis para una primera revisión y/o bien en la fase de diseño para una revisión final y completa:

Entrada:

- Archivo EAP

Salidas:

- Informe sobre la evaluación del testing temprano.
- Alta de "No conformidades" y propuestas de mejoras detectadas durante la revisión.

- Servicio de Verificación de entregas (SVE)

Una vez que se haya finalizado la construcción de las mejoras y llegue el momento de desplegar en un entorno, habrá que tener en cuenta si es la primera versión del producto que se va a desplegar. En este caso, habrá sido necesario construir la plataforma en la que se va a hacer el despliegue. (Ver [Proceso de Gestión de plataformas](#)).

Una vez que la plataforma o plataformas en las que hay que realizar el despliegue están construidas, es necesario hacer la petición de lanzamiento para el despliegue en esas plataformas y en los entornos en los que se quiera implantar la versión del producto. (Ver [Proceso de Gestión de lanzamientos](#)).

Mediante este servicio, se garantiza el cumplimiento de la normativa de entregas publicada por el Área de Gobernanza, así como la calidad del código del producto.

Este servicio es absolutamente necesario para poder gestionar el lanzamiento de una versión a menos de que se trate de un lanzamiento urgente (PLU), en cuyo caso el servicio de la OCA podrá hacerse con posterioridad al lanzamiento.

El proceso incluye la revisión de los siguientes puntos:

- Compilación de fuentes.
- Ejecución de test unitarios.
- Empaquetado de binarios.
- Análisis estático de Calidad.

Entradas:

- Software (código fuente)

Salida:

- Informe sobre la revisión, aceptación o rechazo de la entrega.

- Servicio de validación funcional (SVF)

Esta validación permite determinar si se ha construido el software deseado, que cumple con los requerimientos, y por lo tanto que es candidato a desplegarse en un entorno productivo. Esto ayudará al Servicio Andaluz de Salud a tomar decisiones relacionadas con la versión del producto entregado por el proveedor.

Parte de la ejecución de este Servicio incluye el lanzamiento de pruebas asociadas a versiones anteriores con el fin de garantizar que no se han producido regresiones en la entrega de código. Estas pruebas estarán agrupadas en un plan diseñado para tal fin.

El Responsable de producto deberá realizar la petición al menos un mes antes de la fecha prevista para el comienzo de las pruebas y deberá incluir la siguiente información:

- Entorno
- URLs de las aplicaciones afectadas
- Alcance
- Plan de pruebas (EA o GIT)
- Credenciales de acceso y juego de datos para la ejecución de las pruebas

Entradas:

- Alcance de la versión
- Identificación entorno a probar.
- Plan de pruebas en EA o Git.
- Usuarios de acceso y juego de datos para las pruebas

Salidas:

- Alta de "No conformidades" detectadas durante la verificación funcional en JIRA. En las NC estarán adjuntas las evidencias correspondientes.
- Informe de resultados de las ejecuciones y listado de las NC.

- Servicio de validación de la accesibilidad (SVA)

La accesibilidad tiene como objetivo lograr que las páginas web/aplicaciones móviles sean utilizables por el máximo número de personas, independientemente de sus conocimientos o capacidades personales e independientemente de las características técnicas del equipo utilizado para acceder a la Web o Aplicación.

Todos los sitios webs y aplicaciones para dispositivos móviles desarrollados para la STIC deberán ser accesibles para sus personas usuarias y, en particular, para las personas mayores y personas con discapacidad, de modo que sus contenidos sean perceptibles, operables, comprensibles y robustos. La accesibilidad se tendrá presente de forma integral en el proceso de diseño, gestión, mantenimiento y actualización de contenidos de los sitios web y las aplicaciones para dispositivos móviles.

El Responsable de producto deberá realizar la petición al menos un mes antes de la fecha prevista para el comienzo de las pruebas y deberá incluir la siguiente información:

- Entorno
- Casos de uso
- Credenciales de acceso para las pruebas

- En el caso de aplicaciones móviles APK de la misma.

Entradas:

Entorno de **PRE**:

- APK de la aplicación (SVA-Móvil)
- URL web (SVA-Web)
- Casos de uso
- Credenciales de acceso para las pruebas (Ej: usuario/contraseña, certificado digital, etc)

Entorno de **PRO**:

- Tener publicada la aplicación en el Play Store, Apple Store, etc. (SVA-Móvil)
- URL web (SVA-Web)
- Casos de uso
- Credenciales de acceso para las pruebas (Ej: usuario/contraseña, certificado digital, etc)

Salidas:

- Informe resultados Accesibilidad
- Alta de "No conformidades" detectadas durante la verificación de accesibilidad. En ellas estarán adjuntas las evidencias correspondientes. En el caso de realizar el servicio en el entorno de PRO, sólo se dará el informe de resultados para que sea el Responsable de producto quien registre las incidencias o mejoras oportunas.

- Servicio de validación de la usabilidad (SVU)

La usabilidad persigue que la experiencia del usuario sea lo más satisfactoria posible al interactuar con un producto o servicio, en nuestro caso una web o una aplicación.

La usabilidad es una cualidad abstracta que no puede ser medida directamente y es difícilmente cuantificable. Por ello, se descompone habitualmente en "atributos" que si pueden ser medidos utilizando las denominadas "pruebas de usabilidad", las cuales se aplican sobre el producto software para determinar si es o no usable.

Por tanto, aunque en primera instancia pueda parecer que la experiencia de usuario, y por tanto la usabilidad, son conceptos meramente subjetivos, existen pruebas empíricas para su medición. En el caso de una web o aplicación digital, podemos medir los objetivos de usabilidad en términos de eficacia, eficiencia y satisfacción:

- **Eficacia.** El usuario logra lo que quiere. Se mide en función del número de errores que comete en el proceso.
- **Eficiencia.** Lo logra sin esfuerzo. Se mide en función del tiempo y clics empleados en el proceso.

- **Satisfacción.** La experiencia le reporta satisfacción. Se mide de forma subjetiva a través de preguntas al usuario y será relativa, ya que dependerá de los objetivos planteados y de su comparación con otras experiencias similares.

El Responsable de producto deberá realizar la petición al menos un mes antes de la fecha prevista para el comienzo de las pruebas y deberá incluir la siguiente información:

- Entorno
- Casos de uso
- Credenciales de acceso para las pruebas

Entradas:

- Identificación entorno a probar
- Casos de uso

Salidas:

- Informe resultados usabilidad
- Servicio de pruebas dinámicas de seguridad (SPSD)

Es una vía paralela para la detección de vulnerabilidades de seguridad que ya hace la USTIC mediante la [evaluación de seguridad](#) correspondiente.

En [Servicios de la OCA](#) se muestran videos de sesiones formativas para estos servicios.

Cuando se hayan hecho todas las posibles implantaciones para una versión, es el Responsable de producto quien considera que la versión está implantada, lo que supondrá la implantación de todas las mejoras, incidencias y no conformidades que constituyan el alcance de la versión. Esto no implica que la versión se pueda finalizar, puesto que es posible que queden órdenes de trabajo sin acabar o incluso que fuera necesaria la creación de una orden de trabajo para que el proveedor impartiera formación. Una vez que todas las órdenes de trabajo están cerradas, la versión se puede finalizar.

La implementación de este proceso se hace en [JIRA](#). Accediendo al enlace podrás ver el manual en el que se detallan los tipos de entidades, los estados y las acciones por estado, así como los roles que pueden ejecutarlas.

El proceso de Gestión de productos está ligado al de [Gestión del conocimiento](#), ya que el Responsable de producto debe velar porque todo el conocimiento asociado a su producto sea:

- Centralizado en un único sitio.

- Accesible para todo el que lo necesite con una fácil gestión de acceso.
- Usado como gestor documental para los entregables de las órdenes de trabajo que acomete el proveedor.
- Estructurado de modo similar en todos los productos.
- Integrado con la herramienta en la que se gestiona el producto.

La gestión del conocimiento se hace mediante espacios en CONFLUENCE, de modo que cualquier producto siempre va a tener asociado un espacio en el que se custodie toda la información que afecte a su producto y del que el Responsable de producto va a ser administrador. Todos los espacios tendrán una misma estructura y cada vez que se cree una versión del producto se creará una página dentro del espacio que se actualiza automáticamente con la información de JIRA, estructurada con la información de gestión (OTs de la versión y calendario), alcance y entregables. Cualquier otra información relacionada con la versión se podrá incluir en esa página. Además, el uso de esta herramienta de gestión de conocimiento permitirá:

- Fomentar el trabajo en equipo en el espacio con posibilidad de edición en línea.
- Estandarizar la entrega de los entregables de las órdenes de trabajo.
- Tener un histórico de versiones de la documentación.
- En el futuro permitirá sustituir los “documentos” por páginas.
- Fomentar la comunicación en base a notificaciones “@usuario”.

Comunicación de los cambios (**Changelog de aplicaciones**): puedes ver una guía para saber cómo publicar las novedades de tu producto [aquí](#).

Ciclo de vida ágil

En esta gestión del ciclo de vida, las mejoras ya aprobadas (así como las incidencias y las no conformidades) se descomponen en historias de usuario que constituyen el backlog del producto. Esas historias de usuario se priorizan e incluso se les da un peso (bien por puntos de historia o por horas estimadas) y se incluyen en un sprint, entendido como un miniproyecto de no más de un mes (ciclos de ejecución muy cortos -entre una y cuatro semanas), cuyo objetivo es conseguir un incremento de valor en el producto que estamos construyendo.

Una vez que se tienen las historias de usuario, el equipo puede descomponerlas en subtareas que van acometiendo y que pueden visualizarse en JIRA en una pizarra. Una vez que todas las subtareas de una historia de usuario estén cerradas, la historia podrá resolverse y pasar a ser validada.

Las historias de usuario deben tener siempre una entidad origen. Lo habitual es que sea una mejora, pero también puede ser una incidencia, una no conformidad o un bug (son los errores detectados por la Oficina de Calidad como se ve más tarde).

Es importante destacar que sólo pueden tener una entidad origen en el caso de que ésta sea una mejora, una incidencia o una no conformidad. En el caso de los bugs, es posible que una historia de usuario tenga como origen más de uno, siempre teniendo en cuenta que la mejora origen de todos los bugs que se agrupan en una misma HU debe ser la misma.

Es decir:

HU<- BUG<-Test<-HU<-MEJORA

o

HU<- BUG<-Test<-HU<-NC<-MEJORA

La MEJORA de la que parte todo debe ser la misma.

Si una historia de usuario no tiene identificada su entidad origen no podrá incluirse en un sprint.

Por otra parte, las historias de usuario se clasifican en los siguientes tipos:

- No definida
- Funcional
- Técnica
- Bugs y desuda técnica
- Operación

Cuando se crea una historia de usuario siempre por defecto el tipo es no "No definida". Mientras no se clasifique la historia de usuario no podrá ser incluida en un sprint.

Es posible establecer relaciones de dependencia en historias de usuario.

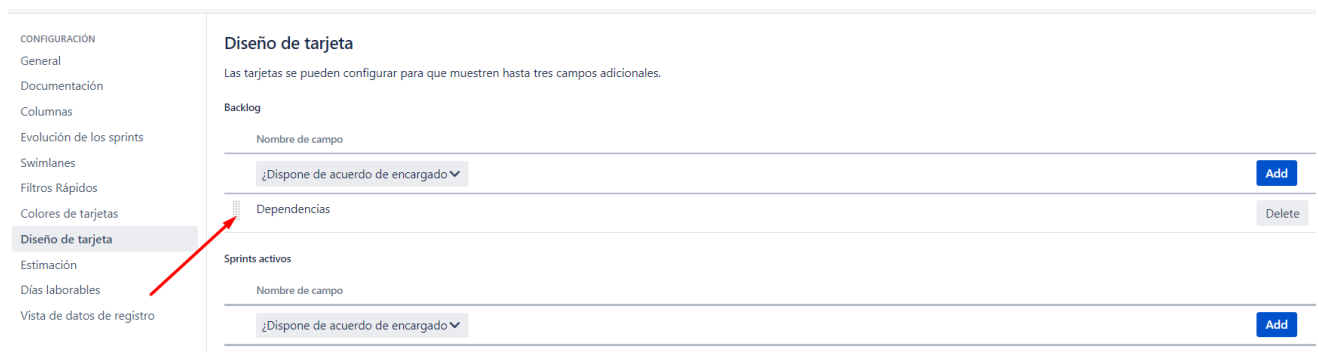
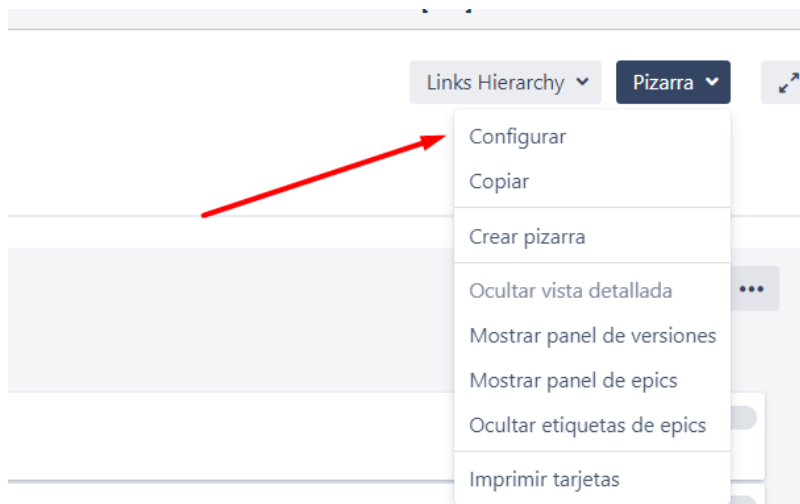
The screenshot shows the Jira user story details page. At the top, there is a toolbar with buttons: Editar, Añadir comentario, Pizarra Ágil, Más (dropdown), Probar, Editar, Paralizar, Cancelar, and Time to SLA (dropdown). The 'Más' dropdown menu is open, showing options: Clasificar al inicio, Clasificar al final, Adjuntar archivos, Adjuntar Captura de Pa..., Crear subtaska, Enlace externo (highlighted with a red arrow), Etiquetas, and Eliminar. The main content area is divided into 'Detalles' and 'Adjuntos'. Under 'Detalles', the 'Tipo' is 'Historia de usuario', 'Versión(es) Afectada(s)' is 'Ninguno', 'Etiquetas' is 'Ninguno', 'Puntos de Historia' is '1', 'Sprint' is 'AGESCON Sprint 1, PIZARRA', and 'Tipo HU' is 'No definido'. Under 'Adjuntos', there is a dashed box with a cloud icon and the text 'Suelte los archivos, pegue una captura de pantalla para adjuntarlos o [explorar](#).'.

Pulsando dos veces en Incidencia Jira

The screenshot shows the Jira 'Enlace' (Link) page. The page title is 'Enlace'. On the left sidebar, there is a section 'Incidencia Jira' with links 'Página de Confluence' and 'Enlace Web'. The main content area has a heading 'Seleccione una incidencia de Jira para enlazarla a esta incidencia'. Below this, there is a dropdown menu 'Esta incidencia' with the value 'Depende de'. Below the dropdown, there is a text input field 'registros' with a dropdown arrow. Below the input field, there is a link 'o [buscar una incidencia](#)'. Below the link, there is a text input field with the placeholder text 'Empiece a ingresar texto para buscar en las incidencias que ha visto recientemente'. Below the text input field, there is a 'Comentario' section with a rich text editor. The rich text editor has a toolbar with buttons: Estilo (dropdown), B (Bold), I (Italic), U (Underline), A (Text color), A (Background color), Link (dropdown), Unlink, Bulleted list, Numbered list, and a plus sign for more options. Below the toolbar, there is a text area for the comment.

Podrán incluirse todo tipo de enlaces entre la historia de usuario y cualquier registro de Jira, teniendo en cuenta la restricción en la relación "originada por" indicada anteriormente.

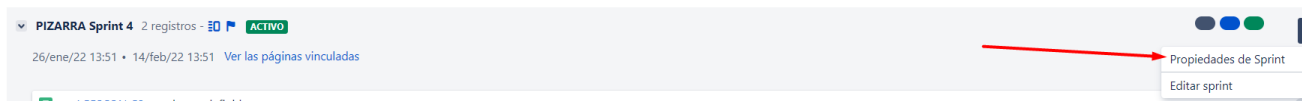
Esta funcionalidad podrá mostrarse en las pizarras (incluyendo en la configuración de las mismas que sea visible el campo DEPENDENCIAS). Puede mostrarse en el trabajo pendiente, en las historias en el sprint activo o en ambos.



La gestión de los costes de un sprint se hace mediante una especie de proyecto (en JIRA la entidad se llama subproyecto ágil) al que se da un horizonte temporal y en el que se van sumando los costes de los diferentes sprints creados en el tiempo de ejecución del proyecto. La gestión de costes en un sprint se controla mediante la creación en la reunión de planning de una OTAgil, que es una orden de trabajo de crédito disponible en la que se controlan los costes del equipo durante el tiempo que dure el sprint y que debe estar relacionada con el subproyecto en el que se quieran controlar los costes de ese equipo.

En el [manual de JIRA](#) podrás encontrar una guía detallada de las historias de usuario, de las subtarefas ágiles y de los errores técnicos así como de la configuración de las pizarras.

Es importante destacar que cuando se crea un sprint y a lo largo de su duración pueden informarse propiedades del mismo, por parte del Responsable de producto, Responsable funcional y del Proveedor



Las propiedades que pueden informarse son:

- Capacidad: número entero entre 1 y 100
- Satisfacción DT: número entero entre 1 y 10 que representa la satisfacción del equipo de desarrollo
- Satisfacción PO: número entero entre 1 y 10 que representa la satisfacción del product owner
- Satisfacción STIC: número entero entre 1 y 10 que representa la satisfacción del Responsable de producto

En la configuración de la pizarra aparecerá una sección denominada Evolución sprints donde aparecerán los datos de las propiedades de los sprints que hayan sido informados.

En la metodología ágil se contempla el papel de la Oficina de Calidad en cada uno de los sprints. Normalmente el miembro de OCA que está trabajando con el equipo ágil realiza una serie de tareas que van destinadas a verificar todo lo que conlleva la entrega que se hace en el sprint. Este trabajo se modela en el Servicio de verificación de sprint (SVS).

A través de este servicio, el equipo de la Oficina de Calidad realiza una labor de acompañamiento y detección temprana de errores. El objetivo es agilizar la solución a los posibles problemas detectados en fase de desarrollo, y ofrecer una una visión del estado en que se encuentra el proyecto en cada etapa, para facilitar el aseguramiento de un producto de calidad que logre la satisfacción del usuario final.

El alcance de este servicio incluye:

- Verificación del repositorio de código.
- Verificación y validación de la calidad estática de código.
- Revisión de las pruebas (.feature).
- Organización del plan de pruebas para su ejecución en el entorno de pruebas.
- Verificación del cumplimiento del DoR.
- Verificación del cumplimiento del DoD.

Entradas:

- Taller concepción del producto (al inicio del proyecto en esta metodología)
- Entrevistas y acuerdos en la STIC (al inicio del proyecto en esta metodología)
- Manual usuario
- Historias de usuario
- Casos de prueba en archivos .feature

- Código en el repositorio correspondiente
- Entornos para las pruebas

Salidas:

- Informe de verificación.
- Alta en JIRA de los Bugs detectados durante la ejecución de las pruebas en el entorno de desarrollo.

Los bugs o fallos encontrados por la Oficina de Calidad en un sprint deben ser analizados por el proveedor y en caso de que realmente se trate de fallos deberán ser corregidos. Para ello deberán crearse historias de usuario originadas por esos bugs (siempre con la misma mejora origen), de manera que cuando dichas historias sean validadas, deberán validarse por la Oficina de Calidad que la resolución de los bugs ha sido satisfactoria.

Puedes encontrar [aquí](#) el ciclo de vida de un bug.

Cuando es necesario hacer un despliegue, es necesario crear una versión como las que se crean en la metodología en cascada.

En cuanto a la Gestión del Conocimiento en este tipo de proyectos, se hace de manera ligeramente diferente. En el espacio de cada producto debe haber una página denominada "Metodología ágil" en la que se debe crear una página con el "Definition of Done" y "Definition of Ready" siguiendo las plantillas definidas. Además por cada sprint que se cree, será necesario crear una página con la planificación del sprint (plantilla Sprint Planning) y otra de retrospectiva (con la plantilla del mismo nombre) el día que se haga la review. Es aconsejable usar una macro de calendario de CONFLUENCE para controlar la disponibilidad del equipo durante el sprint (vacaciones, formación, dedicaciones parciales, etc.)

Dentro del espacio se deja libertad al equipo para ir colocando prototipos o documentación funcional relativa a las historias de usuario.

Ayuda relacionada:

- [Enlace](#)
- [Enlace](#)