Ingeniería de Software

**Gestión de la Configuración (CM)**

***Definición de un Proceso de Control del Cambio***

MIEMBROS DEL GRUPO (L11:007.1.6):

* Diego Salvado Carrillo
* Mateo Senín López
* Pablo Suárez Garabal
* Anxo Trillo Santamaría

FECHA DE ENTREGA: 25/2/2024

| **CONTROL DE VERSIONES** | | |
| --- | --- | --- |
| **VERSION** | **FECHA** | **DESCRIPCIÓN DEL CAMBIO** |
| 1.0 | 02/02/2024 | Diseño del documento inicial. |
| 2.0 | 17/02/2024 | Aplicadas las modificaciones escogidas de entre las sugeridas por el grupo siguiente. |
| 3.0 | 25/02/2024 | Añadidos apartados tocantes a Scrum Poker y control de versiones |
|  |  |  |
|  |  |  |
|  |  |  |

**ÍNDICE**

[**1 Información sobre la práctica a realizar 1**](#_heading=h.oc87a5rdf8zv)

[1.1 Descripción de la práctica 1](#_heading=h.3znysh7)

[1.2 Descripción del grupo de trabajo 1](#_heading=h.4n9uqvxfzk6a)

[1.3 Seguimiento de la práctica 1](#_heading=h.c8l6l7mhl627)

[**2 DOCUMENTACIÓN DE LA PRÁCTICA 2**](#_heading=h.u0l1nq2cort)

[2.1 Descripción del proceso de control de cambios 2](#_heading=h.56whbcb5hwtu)

[2.2 Diagrama de actividades 3](#_heading=h.ulvbmqr2p4z)

[2.3 Definición de Actividades 4](#_heading=h.2s8eyo1)

[2.3.1 Elaborar RFC 4](#_heading=h.b1fk6c6ykyoe)

[2.3.2 Procesar RFC 4](#_heading=h.qdwf8rht6kzs)

[2.3.3 Procesar Archivado 5](#_heading=h.jy8xhq4oamt4)

[2.3.4 Planificación 6](#_heading=h.a739vg2869vm)

[2.3.5 Implementación 6](#_heading=h.p4jboja9hq2x)

[2.3.6 Verificación y Validación 7](#_heading=h.vvnn5kb4jk4x)

[2.3.7 Despliegue 8](#_heading=h.s7xkjpqu2nm0)

[2.3.8 Repliegue 9](#_heading=h.8m65gnco7euo)

[2.3.9 Cierre 10](#_heading=h.6quo29h40rum)

[2.4 Control de versiones mediante Git 11](#_heading=h.jwu2yhbem0js)

[2.5 Toma de decisiones con SCRUM Poker 13](#_heading=h.dal2zj1bc62)

[2.6 Plantillas del proceso 15](#_heading=h.e6q631wzscnt)

[2.6.1 Plantilla de RFC. 15](#_heading=h.26in1rg)

[2.6.2 Plantilla de Procesado. 17](#_heading=)

[2.6.3 Plantilla de planificación 19](#_heading=h.u8zstlve0a5o)

[2.6.4 Plantilla de verificación 21](#_heading=h.b1ryhfj8nq51)

[2.6.5 Plantilla de validación 22](#_heading=h.x7hibtklygs)

[**3 ANEXOS 23**](#_heading=h.35nkun2)

[3.1 Anexo 1.- Cambios propuestos por cada miembro del grupo 23](#_heading=h.2r4ta6ifz3f9)

[3.2 Anexo 2.- Bibliografía y material utilizado 23](#_heading=h.jsc5ypuvcgqz)

# Información sobre la práctica a realizar

## Descripción de la práctica

La práctica consiste en el campo de la gestión de configuración. En este documento se expondrá el ciclo del procedimiento de control de cambios, que está compuesto por una serie de actividades que, una vez estén terminadas, supondrán la implementación del cambio a realizar dentro del proyecto.

En primer lugar se describirán las actividades, así como los miembros relacionados con estas, el objetivo de cada una y su implicación en el proceso global.

Posteriormente, se detallarán el diseño y estructura que deben seguir las plantillas que se utilicen dentro del ciclo del control de cambios, especificando de una manera razonada los campos que contienen.

Finalmente, se expondrán diferentes sugerencias de cambio con respecto al proyecto tratado en la práctica anterior. Por cada miembro del grupo se deberá sugerir un nuevo requisito funcional y un nuevo requisito no funcional, así como mostrar un problema encontrado en el proyecto anterior.

## Descripción del grupo de trabajo

La práctica ha sido desarrollada por el Equipo 6 del Grupo 4 de prácticas, formado por los siguientes integrantes:

* Diego Salvado Carrillo
* Mateo Senín López
* Pablo Suárez Garabal
* Anxo Trillo Santamaría

## Seguimiento de la práctica

A continuación se especifica el tiempo invertido por parte de cada miembro del grupo en el desarrollo de la práctica:

* Diego Salvado Carrillo: 8 horas
* Mateo Senín López: 8 horas
* Pablo Suárez Garabal: 8 horas
* Anxo Trillo Santamaría: 8 horas

# DOCUMENTACIÓN DE LA PRÁCTICA

## Descripción del proceso de control de cambios

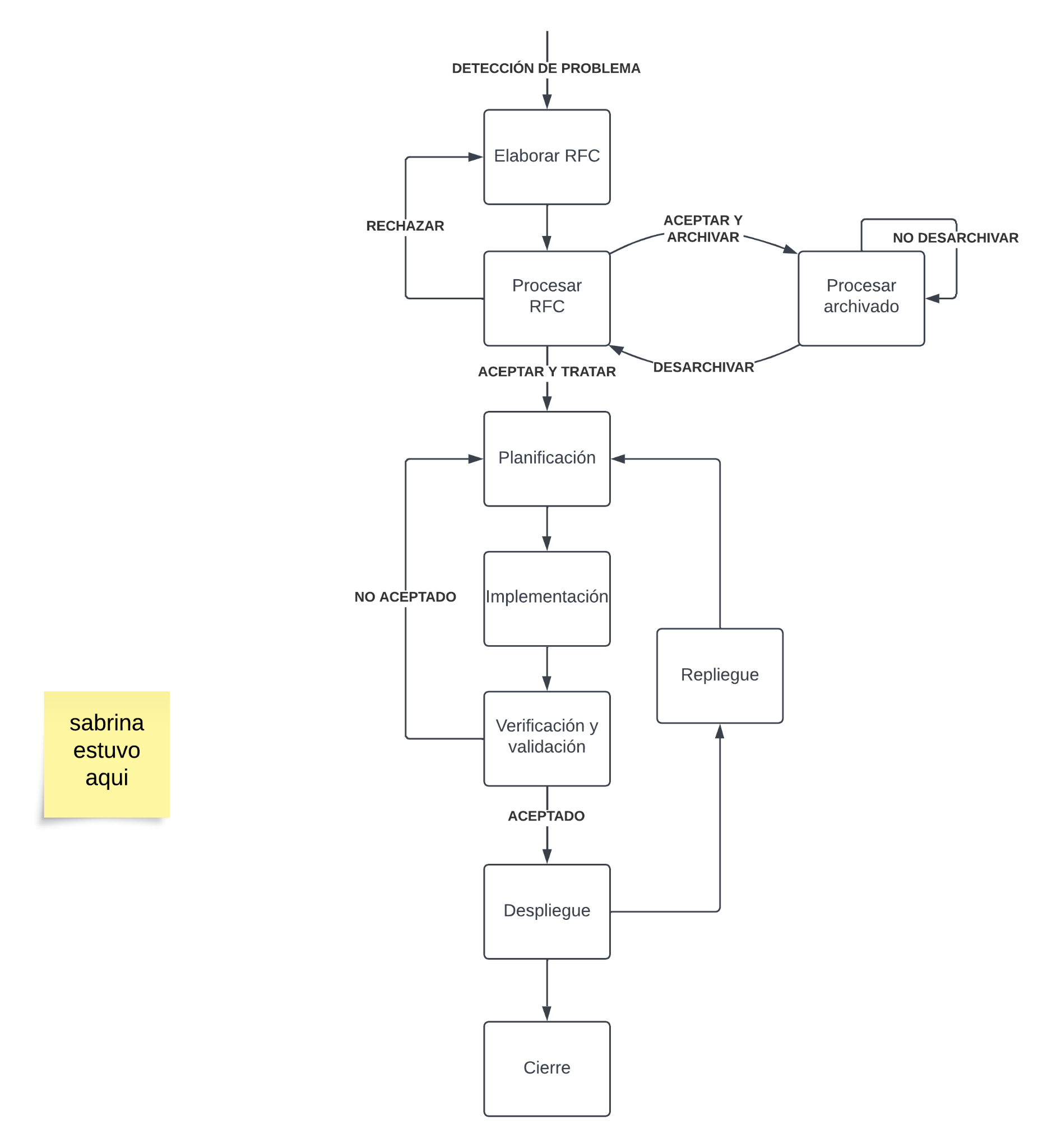
El proceso de control de cambios normalmente se inicia debido a la identificación de un problema. Comienza con una propuesta de cambio (RFC: Request For Change) por parte de la persona que ha identificado el problema.

Posteriormente debemos realizar un procesado de la RFC sugerida, valorando la necesidad y en caso de considerar el cambio necesario la prioridad del tema tratado.

Si se rechaza se informará al cliente y el flujo de procesado de solicitudes de cambios continúa; si se acepta, se deberá planear el tratamiento y la temporización del desarrollo de la solución al problema.

Una vez tratado todo esto los siguientes pasos serán la aplicación de dicha solución y análisis de los resultados de esta; completando el proceso de cambio.

## Diagrama de actividades

******

## 

## Definición de Actividades

### Elaborar RFC

* + - 1. Descripción

Se obtienen las quejas y recomendaciones de los clientes de la fase de detección de problemas. Uno o varios de nuestros empleados se encargan de condensar esa información en una RFC, detallando las razones de queja y su impacto.

* + - 1. Involucrados

Los involucrados en el proceso de elaboración de los RFC son el personal de servicio al cliente o solución de quejas. Los clientes no son parte del proceso de elaboración del RFC, sino del paso previo que es la notificación de un error o mejora, raramente a cargo de la empresa de modo directo sino del usuario.

* + - 1. Entradas requeridas

Entrada de información sobre un problema o posible mejora del sistema.

* + - 1. Productos del trabajo

Plantilla cubierta con información sobre el cambio a realizar, el área a la que atañe y la razón del reporte subida al repositorio en la carpeta /rfc/nuevos.

* + - 1. Criterios de entrada

Una cantidad considerada suficiente de sugerencias de cambio llegan a la empresa.

* + - 1. Criterios de salida

Se ha identificado la intención del reporte del usuario, se ha terminado de redactar el contenido sobre la plantilla de RFC y se ha subido al repositorio git en la carpeta /rfc/nuevos.

### Procesar RFC

2.3.2.1. Descripción

Uno o varios de nuestros empleados revisan la lista de solicitudes de cambio de la que se dispone. Después de su revisión y categorización dependiendo del nivel de urgencia del cambio puede marcarlo como apto para tratamiento, rechazarlo en caso de considerarse trivial o innecesario para la experiencia final del producto, o archivarlo en caso de ser una petición válida pero no de urgencia inmediata o de relevancia suficiente como para ser tratada por delante de otras solicitudes.

2.3.2.2 Involucrados

El personal encargado de revisar las solicitudes que hayan sido enviadas.

2.3.2.3 Entradas requeridas

Documento RFC donde se describen las solicitudes de cambio.

2.3.2.4 Productos del trabajo

Plantilla de Procesado cubierta y subida al repositorio en la carpeta procesado, y el rfc movido de /rfc/nuevos o /rfc/archivados a la carpeta pertinente en función a la decisión tomada, /rfc/en\_curso en caso de aceptarlo, /rfc/archivados en caso de archivarlo y /rfc/historial en caso de rechazarlo.

2.3.2.5 criterios de entrada

Debe haber alguna solicitud de cambio registrada.

2.3.2.6 Criterios de salida

Decisión sobre la relevancia del RFC tomada y rfc movido a una de las carpetas /rfc/en\_curso, /rfc/archivados o /rfc/historial.

### Procesar Archivado

* + 1. Descripción

El RFC designado para archivamiento se había guardado en el archivo con una prioridad asignada a opinión de su relevancia por nuestro personal, esta prioridad se usará durante la acción de desarchivado para elegir el RFC a entrar en etapa de planificación.

* + 1. Involucrados

El personal encargado de revisar las RFCs enviadas y gestionar su estado.

* + 1. Entradas requeridas

Notificación de encontrarse en proceso de análisis de un RFC de prioridad menor que alguno de los RFC que se encuentran archivados.

* + 1. Productos de trabajo

Nuevo cambio seleccionado para iniciar la fase de planificación, introducir en el archivo o borrar en caso de que las circunstancias en las que se archivó dicho RFC ya no existan.

Notificación vía correo electrónico al equipo de planificación conteniendo el RFC.

* + 1. Criterios de entrada

Haber recibido la notificación correspondiente a haber completado el procesado de un RFC de prioridad menor a uno del archivo.

* + 1. Criterios de salida

Se ha enviado una notificación vía correo electrónico con el RFC seleccionado al equipo de gestión de solicitudes RFC en caso de volver a ponerse en circulación; caso contrario, se reintroduce el RFC en la cola. En una tercera opción se ha borrado el RFC al no ser necesario.

### 2.3.4 Planificación

2.3.4.1 Descripción

Se evalúan los pasos necesarios, así como el ciclo de vida más indicado para completar la modificación propuesta por el cambio y se establece una temporización conforme a ellos. El resultado de la actividad se refleja en un documento para su posterior consulta, el cual tendrá como cabecera su CID (String de caracteres ÚNICA que identifican el cambio)

2.3.4.2 Involucrados

Personal asignado a la planificación de cambios en el producto

2.3.4.3 Entradas requeridas

Plantilla de procesado de RFC.

2.3.4.4 Productos del trabajo

Documento con la planificación de los cambios a realizar (plantilla de planificación) subido al repositorio en la carpeta /planificacion con la etiqueta indicando su versión.

2.3.4.5 Criterios de entrada

Recepción de la plantilla de procesado, llegando esta por asignación directa (Aceptar y tratar), o cogiendo el cambio a realizar con mayor prioridad entre las peticiones archivadas.

2.3.4.6 Criterios de salida

Plantilla de planificación con los pasos y tiempos en los que se realizará el cambio subida al repositorio git en la carpeta /planificacion.

### 2.3.5 Implementación

2.3.5.1 Descripción

Se realizan los cambios según la planificación acordada en la etapa anterior, pudiendo este proceso implicar una modificación en el código o la documentación, entre otros.

2.3.5.2 Involucrados

Personal de la empresa, concretamente los directores del proyecto y el personal de desarrollo.

2.3.5.3 Entradas requeridas

Documento con la planificación de los cambios a realizar y el producto en un estado apto para el desarrollo.

2.3.5.4 Productos del trabajo

Programa o solución con los cambios sugeridos implementados en una nueva rama en la carpeta /implementacion con una etiqueta indicando su versión. No se han realizado comprobaciones a fondo ni pruebas exhaustivas del funcionamiento todavía.

2.3.5.5 Criterios de entrada

Recepción de una notificación acorde a que el proceso de planificación de un cambio ha terminado y se puede comenzar la fase siguiente.

2.3.5.6 Criterios de salida

Se ha completado la implementación de los cambios sugeridos en la carpeta /implementacion del repositorio git, se envía una notificación al equipo encargado de verificación y validación para realizar una evaluación del resultado.

### 2.3.6 Verificación y Validación

#### 2.3.6.1 Verificación

2.3.6.1.1 Descripción

Se verifica que la implementación sea correcta a un nivel funcional, es decir, que el modo en que desempeña sus tareas no tenga fallos.

2.3.6.1.2 Involucrados

El personal encargado de comprobar la correcta funcionalidad de la implementación

2.3.6.1.3 Entradas requeridas

Correo/informe del equipo de Implementación, explicando los cambios hechos, el comportamiento que se desea y demás información que el equipo de implementación considere relevante.

2.3.6.1.4 Productos de trabajo

Plantilla de verificación, conteniendo el veredicto y una descripción al respecto subida a la carpeta /verificacion del repositorio.

Notificación al personal de validación para que inicien la fase de validación en caso de verificación satisfactoria.

2.3.6.1.5 Criterios de entrada

Recepción de una notificación vía correo electrónico de la necesidad de verificar el correcto funcionamiento de un bloque de código.

2.3.6.1.6 Criterios de salida

Se ha cubierto correctamente la plantilla de verificación con los datos respectivos sobre las conclusiones tomadas y se ha subido al repositorio git en la carpeta /verificacion.

Se ha enviado una notificación al remitente adecuado (personal de codificación si se han detectado errores o de validación si no).

#### 2.3.6.2 Validación

2.3.6.2.1 Descripción

Se valida que la implementación sea correcta al nivel de los requisitos acordados, es decir, que cumpla su rol de acuerdo al reporte de error o solicitud de cambio enviado en primer lugar.

2.3.6.2.2 Involucrados

El personal, que puede ser una persona o un grupo de personas, encargado de aceptar o denegar la implementación.

2.3.6.2.3 Entradas requeridas

Notificación del equipo de verificación.

2.3.6.2.4 Productos del trabajo

Emisión de una decisión acerca del cambio propuesto que puede ser aceptada o denegada detallada en un documento en la carpeta /validacion del repositorio.

2.3.6.2.5 Criterios de entrada

Plantillas de verificación completamente cubiertas, conteniendo el veredicto y una descripción al respecto.

2.3.6.2.6 Criterios de salida

El acta de aceptación o negación debe estar correctamente completada y firmada.

### 2.3.7 Despliegue

2.3.7.1 Descripción

Los cambios se publican en una nueva versión del producto, que se distribuye al mercado con el propósito de bien enmendar los errores que han sido notificados a la empresa o bien de cubrir las sugerencias de los RFC.

Nótese que no hay modo de saber si la solución enviada va a ser completamente efectiva con total seguridad. Si bien se cuentan con procesos de verificación y validación, el proceso no es perfecto y la solución enviada puede bien ser defectuosa, generar nuevas sugerencias y problemas o ser exitosa pero no solucionar todos los problemas presentes, mostrando la necesidad de procesos de repliegue y de que el proceso de análisis de cambios sea cíclico.

2.3.7.2 Involucrados

Trabajadores encargados de testear la calidad del software nuevamente desarrollado y mecanismo de publicación.

2.3.7.3 Entradas requeridas

Versión del ejecutable que incluye las funcionalidades que implementan el cambio propuesto.

2.3.7.4 Productos del trabajo

Informe de rendimiento donde se recoja si las salidas y funcionamiento del ejecutable son correctas o erróneas, así como datos relativos a la complejidad temporal o espacial de las nuevas funcionalidades.

2.3.7.5 Criterios de entrada

La versión del ejecutable debe incluir todas las funcionalidades necesarias para implementar el cambio propuesto.

2.3.7.6 Criterios de salida

El informe de rendimiento debe estar rellenado de forma correcta.

### 2.3.8 Repliegue

2.3.8.1 Descripción

En caso de que el producto final tenga algún tipo de error o que la nueva versión enviada a los clientes no se adapte a los requisitos se procederá a dar vuelta atrás y eliminar la nueva versión defectuosa del mercado. Debemos tener en cuenta que el proceso por el cual consideramos que la versión pública no es apta y se debe retirar es diferente al proceso de notificar errores y sugerir cambios, en el sentido de que eliminar la versión más novedosa implica volver a trabajar en el cambio mal implementado en lugar de empezar a tratar nuevas solicitudes.

Como aclaración sobre la utilidad de esta función en el proceso, replegar nos permite hacer que, si una solución que no está dando los resultados deseados o falla directamente, se regrese al paso de planificación, planteando un modo de solucionarlo que se espere que tenga éxito sobre el previo. Si bien se podría omitir este paso y asumir que el proceso debería regresar a la fase de identificación de problemas, también debemos tener en cuenta que hemos enviado un producto defectuoso o que no cumple las necesidades previstas (entendiendo las necesidades como solucionar un problema o sugerencia, no como el propósito del producto). Esto deberá ser sacado del mercado o arreglado con algún tipo de parche, en lugar de avanzar y esperar a un RFC que sugiera su mal funcionamiento.

2.3.8.2 Involucrados

El personal encargado de la gestión del estado de los cambios.

2.3.8.3 Entradas requeridas

CID de la actualización fallida junto a una descripción del fallo detectado.

2.3.8.4 Productos del trabajo

Modificación de la plantilla de procesado correspondiente al CID recibido, cambiando el estado a ERR\_PROD y concatenando la descripción del nuevo fallo detectado a la descripción original de la petición de cambio.

2.3.8.5 Criterios de entrada

Haber recibido una notificación de error en el periodo de inicio de despliegue (4 horas).

2.3.8.6 Criterios de salida

Plantilla de procesado modificada para su inmediata resolución.

### 2.3.9 Cierre

2.3.9.1 Descripción

Final del proceso de gestión de cambios. Una vez el proceso se considere exitoso y no se necesite una fase de repliegue, se puede dar por terminado y se elimina la solicitud de cambio de la circulación dentro de la empresa. Una vez terminado se puede regresar a la fase de tratado de RFC.

2.3.9.2 Involucrados

El personal encargado de la gestión del estado de los cambios.

2.3.9.3 Entradas requeridas

Notificación de haber completado el despliegue del cambio.

2.3.9.4 Productos de trabajo

Clausura de la plantilla de planificación del cambio.

2.3.9.5 Criterios de entrada

Haber recibido la notificación correspondiente a haber desplegado el cambio.

Haber pasado un período sin ocasionar errores (4h).

2.3.9.6 Criterios de salida

Cerrar correctamente la plantilla de planificación.

## Control de versiones mediante Git

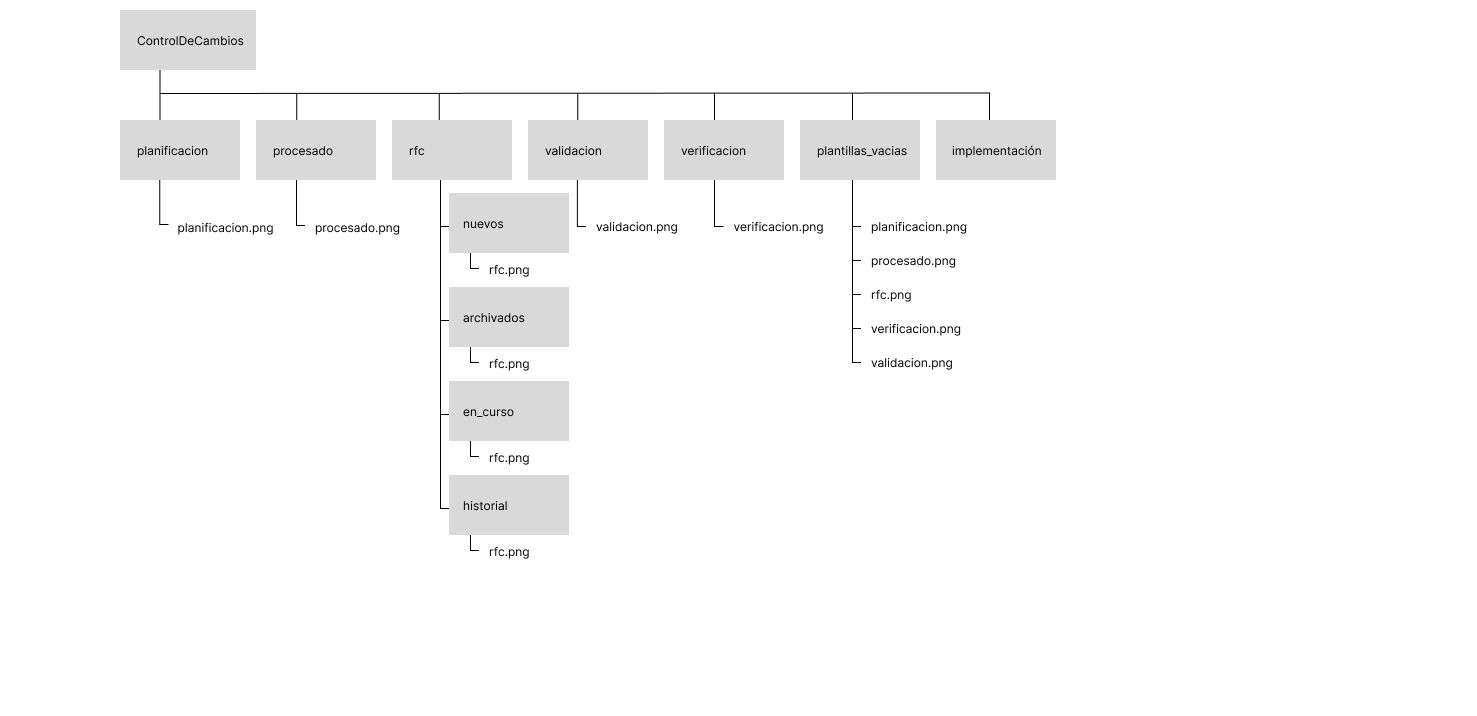
* + 1. Definición y necesidad

Dentro de un proyecto, la existencia de esfuerzos coordinados implica la necesidad de su sincronización o de buscar un modo de acoplar los avances de los diferentes miembros para obtener el producto final.

El control de versiones nace con el objetivo de mantener un registro de los cambios realizados en los documentos del proyecto. De este modo, no solo una modificación puede revertirse o se pueden consultar estados anteriores de documentos, sino que se pueden consultar también diferencias entre versiones y mezclar documentos con modificaciones realizadas por diferentes miembros del proyecto.

Git y su interfaz Github son las herramientas que usaremos para el control de versiones.

* + 1. Estructura del repositorio



* Plantillas\_vacias

La carpeta de plantillas vacías contiene las imágenes de cada una de las plantillas sin completar. Todos los equipos asignados a actividades en las cuales se necesite el uso de una plantilla deben tener permisos de acceso a esta; sin embargo solo los directores o administradores del proyecto deberían tener la capacidad de modificar, añadir o borrar archivos al directorio.

* Implementacion

En la carpeta de implementación se incluye un enlace dinámico al producto en un estado que permita su modificación por parte del equipo de desarrollo. Tendrán acceso únicamente el equipo de implementación (pudiendo modificar, añadir y borrar) y de verificación y validación (pudiendo sólo acceder para la lectura).

Se entiende que al inicio del proceso de cambios esta carpeta contendrá el producto original, y durante el proceso de implementación se creará una nueva rama sobre la que se realizarán los cambios, permitiendo al equipo de verificación y validación hacer comparaciones directas y ubicar los cambios.

* rfc

Contiene las plantillas completadas en la actividad de Elaboración de RFC. Debe tener acceso a esta carpeta todo el equipo, con la intención de permitir la visualización y modificación a todos los miembros.

* + Contiene una carpeta de nuevos, accesible por el equipo de elaboración de RFC y de procesado con permisos de crear, eliminar y modificar elementos. Se compone de los RFC creados pero que no han recibido tratamiento por ningún otro grupo.
  + Contiene una carpeta de archivados, accesible por el equipo de elaboración de RFC y de procesado con los mismos permisos que la previa. Se compone de los RFC que han sido marcados por el equipo de procesado como archivados y que necesitan procesarse de nuevo para poder avanzar en la cadena.
  + Contiene una carpeta de en\_curso, accesible por todo el proyecto, y que marca los RFC que se encuentran en tratamiento en cualquier paso del proceso de gestión de cambios entre la elaboración de los RFC y la publicación.
  + Contiene una carpeta de historial, accesible por todo el proyecto y que contiene los RFC que ya han sido solucionados previamente por el equipo y han sido publicados.

De este modo el trayecto de las plantillas de RFC será la carpeta de nuevas, opcionalmente la de archivados, la carpeta de en\_curso y finalmente la carpeta de historial de haber sido exitoso.

* procesado

Contiene las plantillas de procesado correspondientes a la actividad de Procesado de RFC, las cuales indican el tratamiento de las RFC de la actividad anterior. Debe tener acceso a esta carpeta el equipo de procesado con el fin de añadir, eliminar y modificar archivos y el equipo de planificación, con el fin de obtener la plantilla de mayor prioridad.

* planificacion

Contiene las plantillas de planificación correspondiente a la actividad de Planificación de las modificaciones, las cuales responden a un RFC procesado como aceptación e implicarán una implementación posterior. Debe tener acceso a esta carpeta el equipo de planificación, con permisos completos, y el equipo de implementación con permisos de visualización para acceder a las instrucciones dadas.

* verificacion y validacion

Contienen las plantillas de verificación y validación respectivamente, las dos son accesibles por los equipos de verificación y validación con permisos completos. Debido a que este es el paso anterior al despliegue no hay equipos posteriores con permisos de acceso.

Política de versiones:

El número de versión incrementa cada vez que se vuelve a iniciar un proceso para un mismo cambio, pudiendo añadir dígitos precedidos de un punto para denotar modificaciones de particular interés dentro de una misma versión (“v1.1”, “v2.3”, …).

Política de ramas:

El proyecto poseerá una rama principal, que se corresponde con el proyecto en su estado actual. Una vez un RFC sea aceptado y realizada su planificación, se enviará al equipo de implementación, quien creará una nueva rama sobre la que realizar los cambios. Una vez acabada la modificación a realizar, el equipo de validación y de verificación revisarán el resultado, y en caso de ser aprobado, pasará a la fase de despliegue, donde se unirá la rama de modificaciones con la rama principal. Pasado el tiempo establecido para la posibilidad de repliegue se borrará la rama de modificación.

## Toma de decisiones con SCRUM Poker

* + 1. Necesidad

Dentro de nuestro planteamiento de control de cambios, al igual que en cualquier proyecto, se necesita un protocolo para la toma de decisiones y estimaciones sobre costos, tiempo etc.

Si bien esto puede ser decidido en base a la idea de una única persona, esto empobrece el resultado del producto al tomar una única visión de los problemas y aislarse en su opinión, además de que tener a diferentes miembros del grupo haciendo estimaciones da a todos ellos una visión general de la dificultad para ciertas tareas o partes del desarrollo.

El SCRUM Poker es un método de obtención de estimaciones, en el sentido de que implica a todos los miembros dentro de la reunión a tomar una posición en la decisión, así como a observar las ajenas y discutirlas entre iguales.

* + 1. Definición de SCRUM Poker

El planteamiento de las reuniones es simple. Se considera un problema, meta a alcanzar, idea, etc. Es necesario que tenga un parámetro que necesite estimación, pudiendo referirse a los costes en tiempo, dinero o personal.

Una vez se define el problema y la métrica a valorar se distribuyen “cartas”. Estas cartas son unas tarjetas que ilustran el valor que pueden tomar los distintos elementos en intervalos estáticos dentro de los límites de los valores posibles.

De este modo, si el parámetro a valorar es el tiempo que va a llevar al equipo manejar una operación X, las cartas representan el tiempo que puede llevar la operación, ofreciendo tiempos exactos separados en un intervalo.

Una vez tenemos estas cartas, los miembros de la reunión deben seleccionar la estimación que cada uno ve apropiada, sin haberse hecho pública la decisión todavía. Una vez deciden, todas las cartas se muestran, quedando a la vista la estimación del valor que hace cada miembro de la reunión.

Una vez revelada la posición de cada uno se debe llegar a un consenso o, en su defecto, a una mejora en la convergencia de las respuestas. Una medida para realizar esto puede ser la exposición de los puntos de vista de las posiciones extremas, lo que pretende inclinar las opiniones del equipo o bien hacia arriba o hacia abajo en la escala, acercando las estimaciones de cada miembro a un punto común.

Tras la exposición se repiten las votaciones, habiendo establecido antes una media de discrepancias “adecuada” para terminar la votación. Una vez alcanzada esta discrepancia mínima se usa una fórmula para calcular el veredicto de la reunión, siendo normalmente el tiempo Pert:

Este método nos permite no sólo implicar a todos los miembros, sino obtener estimaciones de cada uno de los miembros que no se hallen modificadas por las de otros. Por otra parte, permite observar al resto de los miembros el proyecto visto desde los extremos, diversificando la forma de ver el trabajo.

* + 1. Reuniones e implementación

Dentro del proyecto de control de cambios existen varios puntos donde es favorable la aplicación de SCRUM Poker, como pueden ser:

* Votación sobre la prioridad de una modificación.

Durante la fase de procesado de las RFC se necesita establecer una medida de la prioridad que puede tener una modificación con respecto a otras sobre el producto.

Si bien hay modos de calcular vagamente la prioridad que unas pueden tener sobre otras, la estimación de su importancia es una toma de decisiones importante para el producto.

* Votación sobre el tiempo que durará una implementación.

Durante la fase de planificación se necesita establecer una medida sobre el tiempo que llevará al equipo el desarrollo de las modificaciones.

La fase de planificación en sí consiste en una serie de toma de decisiones sobre la solución que se va a aplicar para solucionar el problema o sugerencia de cambio. Como último paso se añade una estimación del costo en tiempo y dinero que puede llevar dicha modificación.

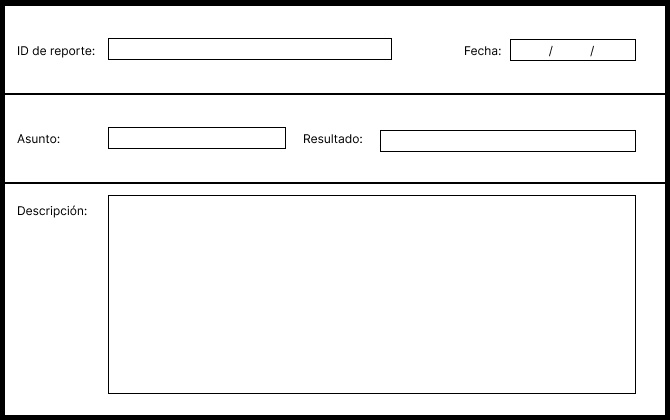
* Votación sobre los costes de una implementación

Complementario a la estimación del tiempo que lleva la implementación está la del costo de los cambios. Del mismo modo que se necesita hacer una estimación del tiempo que puede durar la implementación del proyecto, es necesario que el equipo se sitúe en la cifra que consideran idónea como presupuesto de su implementación.

## Plantillas del proceso

### Plantilla de RFC.

#### Modelo

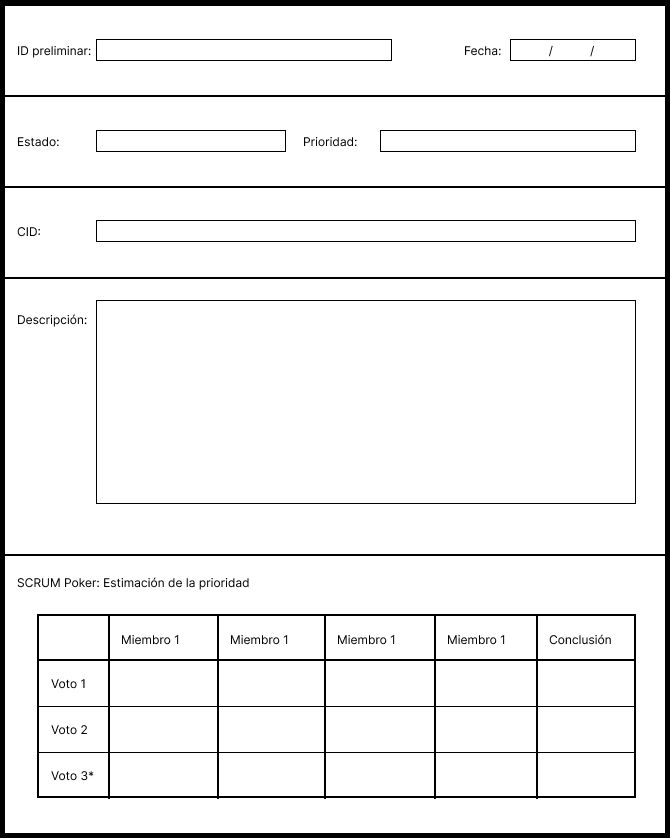
****

#### Justificación de sus campos

* Aclaración: Debido a que es la primera fase del proceso, y especialmente a que posteriormente se clasificará el RFC según su prioridad, los campos referentes a esta pueden ser más descriptivos en el RFC.
* ID de reporte: Código único con el cual diferenciar la plantilla de otra del mismo tipo.
* Fecha: Fecha en la que se entrega la plantilla.
* Asunto: Breve descripción de la sugerencia o problema, asociándolo con el departamento al cual pueda atañer la resolución de este.
* Resultado: Breve descripción del estado que provoca que dicha sugerencia no esté resuelta, sea el fallo total de alguna de las partes del producto, poca efectividad de una de sus partes, riesgos para los clientes etc. Este campo se usará principalmente para diferenciar una sugerencia de mejora (me gustaría que el producto hiciese A), con una solicitud de cambio que aluda a malas experiencias o fallos (el producto A no funciona correcta o favorablemente).
* Descripción: Texto de mayor densidad explicando la experiencia del producto que provoca la necesidad de cambios.

### Plantilla de Procesado.

#### Modelo

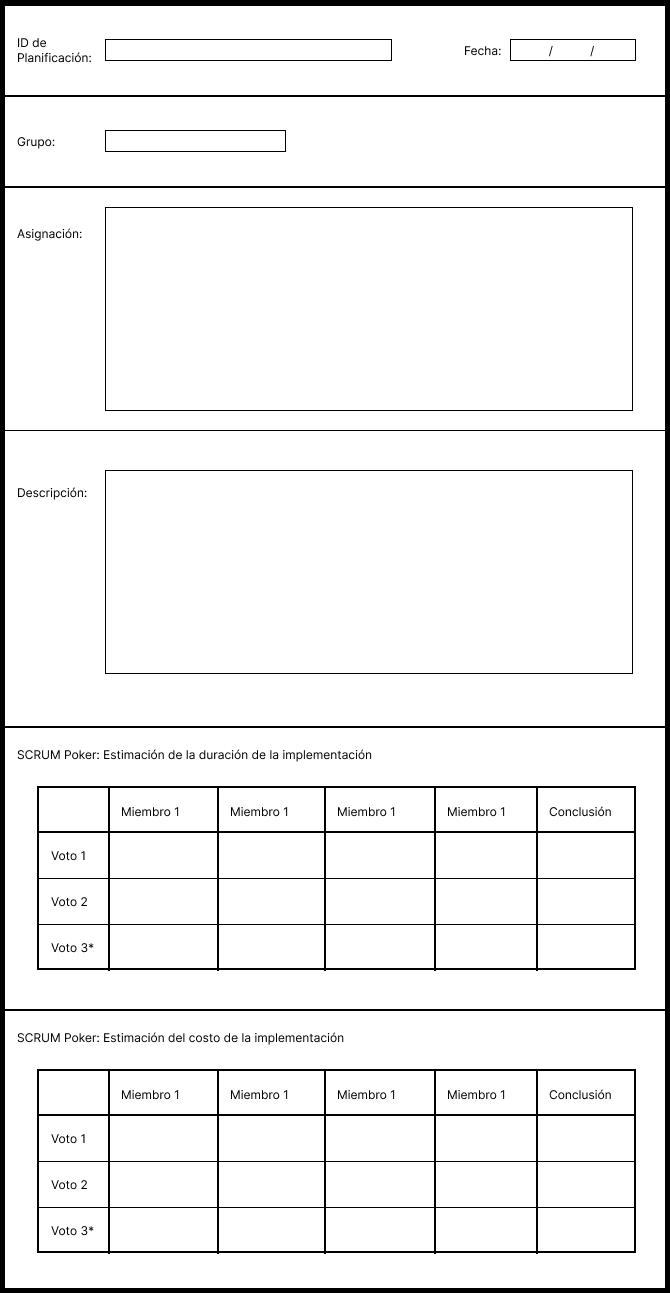
****

#### Justificación de sus campos

* ID: Cadena de caracteres que identifica de forma única el documento generado con la plantilla.
* Fecha: Fecha en la que se rellena la plantilla.
* Estado: Campo que indica el estado del procesado del cambio. Puede tomar los valores [Aceptado, Denegado, Archivado].
* Prioridad: Número decimal del 1 al 10 que estima la urgencia con la que debe ser tratado el cambio.
* CID: Cadena de caracteres que identifica el cambio de forma única que sólo se asigna una vez es aceptado.
* Descripción: Campo de texto que contiene una descripción del cambio.
* SCRUM Poker: Campos de la tabla destinada a las votaciones dentro de la estimación de valores del proceso. En este caso contamos con una tabla dedicada a la estimación de la prioridad de una RFC. Los campos verticales representan los votos de cada miembro y la conclusión de la votación. Los horizontales representan cada una de las votaciones, siendo la tercera opcional.

### Plantilla de planificación

#### Modelo



#### 2.4.3.2 Justificación de sus campos

* ID: código único que identifica el problema que se está procesando.
* Fecha (dd/mm/aaaa): fecha de emisión de la plantilla.
* Grupo: identificador del grupo de trabajadores que lleva a cabo el análisis del problema.
* Fechas límite parciales: fecha de finalización límite para cada parte del trabajo realizado por los trabajadores.
* Encargados de cada parte: parte de la que se tiene que encargar cada trabajador, que será entregada antes de la fecha límite de cada entrega.
* SCRUM Poker: Campos de tabla destinadas a las votaciones dentro de la estimación de valores del proceso. En este caso contamos con una tabla dedicada a la estimación del tiempo que puede durar la fase de implementación y otra dedicada a estimar el coste que va a tener la misma. Los campos verticales representan los votos de cada miembro y la conclusión de la votación. Los horizontales representan cada una de las votaciones, siendo la tercera opcional.

### Plantilla de verificación

#### Modelo

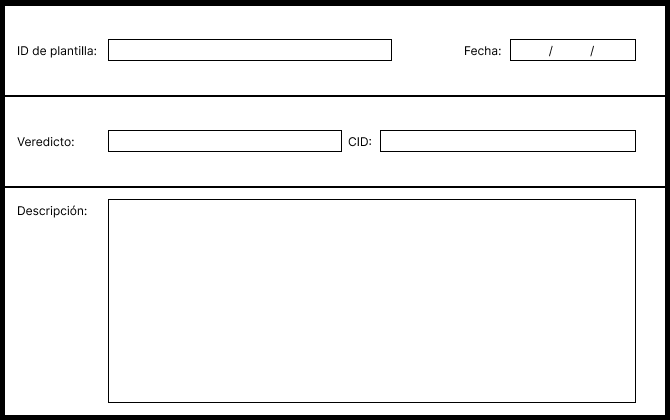
#### 

#### Justificación de sus campos

* ID: Cadena de caracteres que identifica de forma única el documento generado con la plantilla
* Fecha (dd/mm/aaaa) de emisión de la plantilla.
* Nombre del equipo o personal que ha completado la plantilla de validación.
* Objetivos cumplidos que estaban pendientes de ser completados.
* Objetivos no cumplidos
* Descripción: Descripción de los cambios realizados en caso de un veredicto exitoso, y descripción del fallo detectado en caso de error.
* Veredicto de la funcionalidad del cambio, indicando si se ha solucionado el problema de forma correcta, o en cambio los posibles fallos.
* Listo: Carácter que identifica si la implementación está lista para validarse, o si por el contrario deben corregirse o completarse otros errores.

### Plantilla de validación

#### Modelo



#### Justificación de sus campos

* ID: Cadena de caracteres que identifica de forma única el documento generado con la plantilla.
* Fecha (dd/mm/aaaa) de emisión de la plantilla.
* Veredicto de la funcionalidad del cambio, indicando si se ha solucionado el problema de forma correcta, o en cambio los posibles fallos.
* CID: Cadena de caracteres de identificación del cambio realizado.
* Descripción: Descripción del cambio realizado en caso de un veredicto exitoso y descripción del fallo detectado en caso de error.

# ANEXOS

## Anexo 1.- Cambios propuestos por cada miembro del grupo

Diego Salvado Carrillo:

* Propuesta de requisito no funcional: Opciones de conexión P2P
* Propuesta de requisito funcional: Permitir la opción de que (si así lo desean) los administradores de 2 o más grupos puedan crear un grupo temporal de mayor tamaño para actividades compartidas.
* Propuesta problema en documentos de línea base: Sobrecarga en los servidores por el paso de mensajes

Mateo Senín López:

* Propuesta de requisito no funcional: Cliente web
* Propuesta de requisito funcional: Permitir la localización de los miembros de un grupo vía GPS (los miembros a localizar reciben una solicitud de ubicación que pueden rechazar)
* Propuesta problema en documentos de línea base: Ver reseñas, un requisito fundamental para alcanzar uno de los objetivos del proyecto, está evaluado como el menos prioritario de todos.

Pablo Suárez Garabal:

* Propuesta de requisito no funcional: Seguridad de la app y prevención de acoso en chat de usuarios.
* Propuesta de requisito funcional: Solicitud de rutas turísticas o bares según geolocalización ordenándolas por distancia y calidad de puntuaciones en reseñas.
* Propuesta problema en documentos de línea base: La urgencia e importancia de “Reportar incidencia con usuario” es demasiado baja como para que “Solucionar reporte” tenga una importancia más alta, teniendo en cuenta que para que haya una solución a una incidencia o problema tiene que haber alguien que reporte la incidencia o problema.

Anxo Trillo Santamaría:

* Propuesta de requisito no funcional: Autenticación en dos pasos para los clientes de móvil y evitar multicuentas.
* Propuesta de requisito funcional: Solicitar creación de puntos de interés
* Propuesta problema en documentos de línea base: Las funciones de solución de reporte están escritas de modo que no hay funciones reales para que un administrador tome medidas al respecto

## Anexo 2.- Bibliografía y material utilizado