

Procedimiento Nuevo Requerimiento

Servicio de Soporte

Empresa: CMPC



Autor: Luis Sandoval Medina

Versión del documento: 1.0

Fecha Documento: 05/07/2021



Procedimiento Nuevo Requerimiento



Contenido

1	Introducción	2
2	Procedimieno Nuevo Requerimiento	3
3	Procedimiento Informar a QA tickets en curso	6
4	ANEXO: Excel de paso a producción	7

1 Introducción

El presente documento describe el procedimiento administrativo relacionado a la creación de un nuevo requerimiento, en el se detallan los pasos desde la creación de la rama reléase RE en bitbucket, hasta el cierre de esta post paso a producción y aprobación por el usuario final.

También se incluye un breve procedimiento de comunicación de la información de requerimientos y sus fechas desde la Jefe de Servicio a la Jefe del área de Calidad, a fin de proyección de nuevas liberaciones y coordinación de la carga laboral.

2 Procedimiento Nuevo Requerimiento

A continuación, en la *Ilustración 1*, se presenta el diagrama del procedimiento administrativo de un requerimiento de 40 horas o más, para requerimientos internos y de 100 horas o mas para requerimientos de proveedores.

Los actores de este procedimiento son:

- (1) Desarrollador.
- (2) QA.
- (3) Bitbucket
- (4) Usuario
- (5) Jefe de Proyecto CMPC

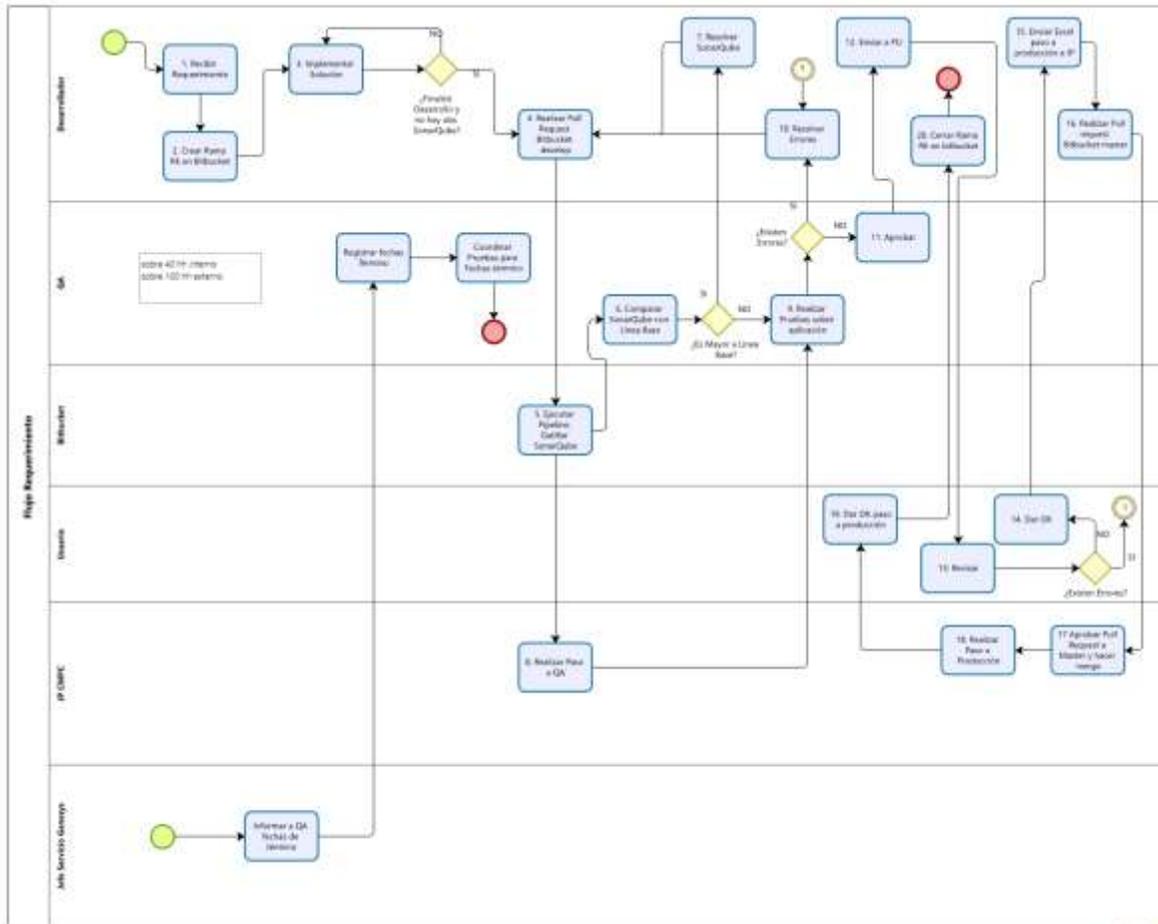


Ilustración 1: Procedimiento Nuevo Requerimiento

2.2 Descripción Procedimiento Nuevo Requerimiento.

- 1. Recibir Requerimiento:** el proceso se inicia cuando un **desarrollador** recibe para implementar, un requerimiento de 40 horas o más, si es Genesys y 100 horas o más, si es u proveedor.
- 2. Crear Rama RE en Bitbucket:** en este momento el **desarrollador** deberá crear una rama tipo feature¹ en bitbucket, considerando que el nombre debe estar compuesto por RE + <número ticket>. Esta rama debe ser creada desde la rama develop.
- 3. Implementar Solución:** El **desarrollador** una vez que ya cuenta con la rama feature en local, creada desde la rama develop, deberá comenzar a implementar la solución asociada al requerimiento.
- 4. Realizar Pull Request Bitbucket develop:** una vez que el **desarrollador** finalizó la implementación y confirmó que no existen observaciones de SonarQube, el **desarrollador** deberá realizar un Pull Rquest a la

¹ Se deberá crear tantas ramas tipo feature, como repositorios participen de la solución a ser desarrollada.

rama develop, realizando el correspondiente merge, esto gatillará en bitbucket la ejecución del pipeline (ver siguiente paso). En caso de que el desarrollo aún no finalice se deberá continuar en el paso 3, implementando la solución.

5. **Ejecutar Pipeline, Gatillar SonarQube:** una vez realizado el commit de los códigos a la rama develop, automáticamente **Bitbucket** ejecutará el pipeline, gatillando la ejecución de SonarQube quien generará un reporte de observaciones de mejores prácticas y vulnerabilidad.

6. **Comparar SonarQube con Línea Base:** una vez ejecutado el pipeline y ejecutado SonarQube, corresponderá al **área de QA** asegurar que la aplicación no posee mas observaciones de SonarQube, que las registradas en la línea base, en caso de que existan mas vulnerabilidades el desarrollador deberá resolver estas en el siguiente paso.

7. **Resolver SonarQube:** el **desarrollador** resolverá las vulnerabilidades adicionales (no presentes en la línea base), existan en la aplicación. Una vez resueltas las vulnerabilidades, **el desarrollador** deberá nuevamente realizar el paso **4. Realizar Pull Request Bitbucket develop**.

8. **Realizar Paso a QA:** una vez ejecutado el pipeline en el paso 5, además de la comparación con la línea base de sonarqube por parte de QA en el paso 6, **el jefe de proyecto de CMPC**, deberá realizar la instalación de la aplicación (solución implementada por el desarrollador), en el o los servidores de QA involucrados. Con el paso a servidores de QA realizado por el Jefe de proyecto de CMPC y una comparación exitosa de QA en el paso 6 asociada a SonarQube, QA podrá realizar las pruebas de la aplicación en el siguiente paso.

9. **Realizar pruebas sobre la aplicación:** el **área de QA** realizará las pruebas necesarias a los códigos modificados por el desarrollador y generará la evidencia de pruebas que entregará al desarrollador en caso de errores.

10. **Resolver Errores:** una vez revisada la aplicación por el área QA en el **paso 9**, será **el desarrollador** quien deberá resolver los errores reportados en caso de que existan, una vez resueltos los errores **el desarrollador** deberá llevar a cabo el paso **4. Realizar Pull Request Bitbucket Develop**.

11. **Aprobar:** en caso de que las pruebas realizadas por el área QA en el paso 9 resulten exitosas, **el área QA** realizará la aprobación de la aplicación informando de esto, a través, de un correo al desarrollador.

12. **Enviar a PU:** una vez que **el desarrollador** recibió la aprobación desde el área de QA, este enviará la aplicación a pruebas de usuario, informando al usuario respectivo que ya puede probar la aplicación indicando las formas de acceder y detalles de lo que debe probar, si fuese necesario.

13. **Revisar:** una vez que el desarrollador informó al usuario respectivo de que la aplicación está disponible para pruebas, será **el usuario** quien deberá revisar que desde su visión no existan errores.

14. **Dar OK:** En caso de que no existan errores **el usuario** enviará un OK al desarrollador quien continuará en el **paso 15**, pero en caso de existir errores estos los comunicará al desarrollador quien deberá continuar en el **paso 10. Resolver errores**.

15. **Enviar Excel paso a producción a JP:** con el ok a los cambios realizados a la aplicación, recibido desde el usuario, será **el desarrollador** quien solicite el paso a producción al JP, enviando la **Excel de paso a producción definida por CMPC**².

² Para mas detalles ver punto 4. Anexo: Excel de paso a producción, de este documento.

16. **Realizar Pull Request Bibucket master:** en paralelo a la actividad 14, **el desarrollador** deberá realizar pull request a la rama master, ingresando el mail del jefe de proyecto a fin de que se reciba una alerta de esta actividad. A diferencia de la actividad 4, en esta ocasión **el desarrollador** no deberá realizar la aceptación y merge del pull request.

17. **Aprobar Pull Request a Master y hacer merge:** una vez recibido por el JP la información de Excel de paso a producción enviada por el desarrollador en el paso 14 y realizado el pull request por parte del desarrollador en el paso 15, **el Jefe de proyecto de CMPC** podrá aprobar el pull request y realizar merge, para que los datos lleguen a la rama master.

18. **Realizar paso a producción:** corresponderá al **Jefe de proyecto de CMPC** una vez aprobado el Pull Request en el paso 16, que realizó previamente el desarrollador en el paso 15, realizar la instalación de los fuentes en el servidor de producción y una vez realizado informar tanto al desarrollador como al usuario de esta actividad.

19. **Dar OK paso a producción:** **el usuario** deberá revisar lo instalado en el servidor, y dar el ok respectivo, ok que enviará al Jefe de proyecto de CMPC y al desarrollador, con esto se dará el requerimiento por cerrado.

20. **Cerrar rama RE en Bitbucket:** **el desarrollador** una vez recibido el OK del usuario en producción, procederá a cerrar la rama RE creada en el paso 2 de este procedimiento, dando el requerimiento por cerrado.

3 Procedimiento Informar a QA tickets en curso

Este procedimiento, es complementario al anterior y se lleva a cabo una vez a la semana los días lunes, permite que QA disponga de información oportuna para su organización interna, al conocer las fechas futuras y todos los requerimientos en carpeta, con esto podrá medir su carga de trabajo.

El procedimiento se encuentra incorporado en la **Ilustración 1** y está compuesto por tres pasos.

- A. **Informar a QA fechas de término:** el **Jefe del servicio de Soporte**, enviará al área QA, un listado consolidado con todos los tickets de más de 40 horas, incluyendo como información, al menos los siguientes datos:
 - a. **Id de requerimiento.**
 - b. Nombre del requerimiento.
 - c. Fecha de Inicio.
 - d. Fecha de paso a QA.
 - e. Fecha comprometida de paso a producción.
 - f. Responsable del requerimiento (desarrollador).
 - g. Jefatura directa del desarrollador.
 - h. Usuario involucrado.
 - i. Cantidad de horas del requerimiento.
 - j. Sistema relacionado.
- B. **Registrar Fechas de Término:** el **área de QA** recibirá la información proporcionada por el Jefe de Servicio, y coordinará sus actividades y carga de trabajo para recibir la carga que se menciona en el listado de requerimientos recibida.
- C. **Coordinar pruebas para fecha de término:** el **área de QA** asignará un responsable de realizar las pruebas a quien informará la fecha y los antecedentes necesarios para que el responsable se

contacte con el desarrollador respectivo y pueda ser parte de las pruebas en el procedimiento de nuevo requerimiento ya descrito en el ítem: **2**, de este documento.

4 ANEXO: Excel de paso a producción

CMPC ha definido dos Excel de paso a producción, una para BD y otra para Aplicación, a continuación, se adjuntan ambos formatos.

Excel paso a producción BD



Especificacion PP
BD.xls

Excel paso a producción API



Especificacion PP
API.xlsx