



CIT@MEDICA

## Acta de constitución del proyecto

**CIT@MEDICA**

**Fecha: 29/09/2022**

## Tabla de contenido

Tabla de versionado.....	2
Cómo Representar los Requisitos.....	3
Cómo se Priorizan.....	3
Qué Métricas se Utilizarán para Controlar su Cumplimiento.....	4
Cómo se Analizará el Impacto de un Cambio y quién lo Autoriza.....	4
Cómo hacer el Seguimiento de los cambios en los Requisitos .....	4

## Tabla de versionado

Versión	Modificado por	Descripción	Fecha
<b>V0.1</b>	Diego Jesús Díaz López Juan Antonio Mena Vargas	<i>Versión inicial del documento</i>	03/10/2022
<b>V0.2</b>	Diego Jesús Díaz López	<i>Pequeña modificación sobre cómo representar los requisitos y su priorización. Actualizado cómo hacer el seguimiento de los cambios en requisitos.</i>	18/10/2022
<b>V0.3</b>	Diego Jesús Díaz López	<i>Modificado criterio de aceptación Unificado formateado</i>	23/10/2022
<b>V1.0</b>	Todos	<i>Aprobación del documento</i>	24/10/2022

---

## Cómo Representar los Requisitos

---

El equipo de desarrollo acordó una nomenclatura específica para representar los requisitos y llevar un orden de estos. El esquema básico de representación sería:

Task-YXX-vx.x: abc

(Descripción)

Donde YXX sería el identificador del requisito en cuestión, formado por una letra A, B o C correspondiente con los requisitos del producto, del proyecto y de la organización respectivamente y dos números enteros. Además abc sería el título del requisito. Seguidamente, se aportará una breve descripción de este. Finalmente, vx.x se corresponde con la versión del requisito en cuestión.

Los requisitos se categorizarán según si son funcionales o no funcionales, mantenimiento, seguridad u obsoletos mediante etiquetas asociadas a los mismos.

De cara a GitHub, se crearán ramas a partir de las issues con el formato autogenerado de la plataforma a través del botón “create Branch” en los detalles de la issue en concreto. Se añadirá al comienzo del nombre generado el prefijo “FEATURE/”, “BUGFIX/”, “PLAN/” si se corresponden con ramas en las que se implementa una task, un arreglo o un documento respectivamente.

---

## Cómo se Priorizan

---

Los requisitos se priorizan mediante un sistema de etiquetado por el que se diferencian tres niveles de prioridad; “muy alta”, “alta”, “media” y “baja”. El equipo llegará a un consenso sobre la prioridad de cada una de las tareas.

En el caso de que un requisito quede obsoleto, este no poseerá ninguna prioridad.

---

---

## Qué Métricas se Utilizarán para Controlar su Cumplimiento

---

Para controlar el cumplimiento de un requisito se estimarán una serie de tests que dicho requisito debe de cumplir para considerarlo como terminado. Estos tests deben en lo posible ejecutarse de forma automática.

Como métrica, por lo tanto se tendrán en cuenta cuántos de los tests previstos supera adecuadamente el requisito y cuántos de ellos se realizan de forma automática.

---

## Cómo se Analizará el Impacto de un Cambio y quién lo Autoriza

---

Tras la notificación de rechazo de un requisito implementado por parte del patrocinador, el equipo se reunirá en un plazo de 3 días para llevar a cabo un análisis en el que se evaluarán aquellos subsistemas que se vean afectados por el cambio que debe de hacerse. Tras este análisis, es el director de proyecto el encargado de autorizar la puesta en marcha del nuevo requisito.

---

## Cómo hacer el Seguimiento de los cambios en los Requisitos

---

Ante un cambio en un requisito, se creará una nueva issue con la versión correspondiente del requisito a través del repositorio en GitHub, así como la rama correspondiente marcada con el prefijo “FEATURE/”.

En el caso de que algún requisito requiera de modificación, se reestructurará el mismo en el documento de requisitos tras ser autorizado por el director de proyecto. Además, deberá estar al tanto de las nuevas versiones y aceptar las pull request con la rama de desarrollo “develop”.

---