



Smart Software Solutions

EQUIPO DEL PROYECTO:

ALBERTO GARCÍA HERNÁNDEZ

JUAN ABASCAL SÁNCHEZ

CARLOS OLIVARES SÁNCHEZ MANJAVACAS

DANIEL GONZÁLEZ DE LA HERA

CARLOS TORMO SÁNCHEZ

Grupo 1

DOBLE GRADO EN: INGENIERÍA INFORMÁTICA Y ADE

DIRECCIÓN DE PROYECTOS DEL DESARROLLO DEL SOFTWARE

Índice general

Índice general	1
1 Plan General de Configuración	3
I. Introducción	3
I.1. Propósito del plan	3
I.2. Alcance	3
I.3. Definiciones y acrónimos	3
II. Especificaciones de requisitos	5
II.1. Organización	5
II.2. Responsabilidades	5
II.3. Políticas, directivas y procedimientos aplicables	6
III. Actividades de gestión de configuración	6
III.1. Identificación de la configuración	7
III.2. Control de cambios en la configuración	13
A Estado Actual del Plan de Gestión de la Configuración	17
I. Definición de los EC hasta el momento	17
II. Definición de las relaciones de los actuales EC	18
II.1. Relaciones de sucesión	18
II.2. Relaciones de derivación	18
II.3. Relaciones de dependencia	18
II.4. Relaciones de variante	19
II.5. Relaciones de composición	19
III. Definición de las líneas base actuales	19
Índice de figuras	20
Índice de cuadros	20
Bibliografía	21

PLAN GENERAL DE CONFIGURACIÓN

I INTRODUCCIÓN

I.1 PROPÓSITO DEL PLAN

El Plan que a continuación se detalla, va dirigido tanto al personal desarrollador como al equipo de dirección. Con él se pretende dotar al proyecto de suficiente robustez a la hora de recopilar información acerca del estado del producto, así como a la hora de realizar un cambio. Los cambios son especialmente delicados en este, dado que existen elementos que requieren especial atención y cuidado a la hora de modificarlos. Así pues se pretende documentar cada línea base y cada cambio realizado según lo indicado más abajo cuando se detallen las actividades de gestión de configuración.

I.2 ALCANCE

El presente plan de GCS se aplicará al proyecto realizado por Smart Software Solutions para CARSAFETY, el cual se corresponde con un sistema de gestión de la seguridad para vehículos, compuesto por cinco subsistemas: control del punto ciego, alerta de cambio de carril, alerta de velocidad, llamada automática de emergencia y sistema pre-colisión

I.3 DEFINICIONES Y ACRÓNIMOS

A continuación aparecen las definiciones utilizadas en el presente plan de gestión de configuración.

Bibliotecas software: es un repositorio de documentación y de colecciones de software que sirve como soporte para ayudar en el desarrollo de un proyecto.

Ciclo de vida: es una secuencia estructurada y bien definida de las etapas necesarias para desarrollar un determinado producto software. Comité de Control de Cambios:] persona o conjunto de personas encargadas de supervisar y aprobar todos los cambios sugeridos.

Control de versiones:	se trata de la gestión de las diversas modificaciones realizados sobre los elementos del proyecto.
Elementos de configuración: Líneas base:	es la información creada como parte del proceso de un determinado proyecto. especificación o producto que ya se ha revisado formalmente, y sobre el que se ha llegado a un acuerdo. De esta manera, sirve como base para cualquier desarrollo posterior que se quiera realizar.
Petición de cambio:	solicitud que se presenta ante el CCC, que describe un cambio de cualquier tipo en el ciclo de vida natural del producto, o en aspectos relacionados.
Versión:	es el estado en el que se encuentra un proyecto en un momento determinado de su desarrollo.
Versión en desarrollo:	versión de un componente que todavía está sufriendo modificaciones y, por lo tanto, no está disponible para su uso.
Versión final:	versión de un componente que se encuentra disponible para el uso de usuarios finales.

A continuación, aparecen los acrónimos utilizados en el presente plan de gestión de configuración.

CCB:	Configuration Control Board. Comité de control de la configuración.
CI:	Configuration Item. Elemento bajo gestión de la configuración.
CM:	Configuration Management. Manejo de la gestión de la configuración.
SCM:	Software Configuration Management. Gestión de configuración del software.
SCMR:	SCM Responsible. Responsable del SCM.
SCMP:	Software Configuration Management Plan
SCR:	System/Software Change Request. Petición de cambio en el sistema/software.
CCC:	Comité de Control de Configuración.
CCR:	Responsable del CC.
EC:	Elemento de Configuración.
EVS:	Estudio de Viabilidad del Sistema
DAS:	Documento de Análisis del Sistema.
DCC:	Documento de Cálculo de Costes.
DDS:	Documento de Diseño del Sistema.
DHP:	Documento de Histórico del Proyecto.
DIS:	Documento de Implantación del Sistema.
IAS:	Implantación y Asimilación del Sistema.
IQS:	Informe Quincenal de Seguimiento.

OFE:	Oferta.
GConf:	Plan de Gestión de Configuración.
PGCal:	Plan de Gestión de Calidad.
PER:	Planificación y especificación de requisitos
DCS:	Documento de Construcción del Sistema.

II

ESPECIFICACIONES DE REQUISITOS

En este apartado se identifican las tareas de coordinación y gestión que serán necesarias para llevar a cabo la SCM. El SCMR será el encargado de realizar las siguientes actividades: definir ítems de configuración, definir un ambiente para llevar el control de cambios sobre estos ítems, definir el proceso de cambios, mantener la línea base del proyecto, controlar cambio importantes sobre la línea base del proyecto y auditar la estabilidad de la línea base.

Este apartado está compuesto por tres puntos: la organización del proyecto, las responsabilidades del SCRM, y las referencias a las políticas que se van a aplicar en este proyecto.

II.1

ORGANIZACIÓN

Debe existir contacto permanente y directo entre el personal desarrollador y el comité de control de cambios, de modo que las demoras en la tramitación de un cambio sean lo más cortas posible, de modo que los procesos tanto de mejora como de corrección no sean un trabajo tedioso. Tanto el comité de control de cambios como el resto de personal desarrollador deben prestar especial atención a los puntos en los que se ha estipulado que se van a establecer líneas base dentro del desarrollo.

II.2

RESPONSABILIDADES

Comité de control de cambios:	conjunto de personas encargadas de valorar las ventajas y los inconvenientes de las solicitudes de cambios que pueden afectar al proyecto, de tal manera que el impacto que puede producir dichos cambios sea mínimo. Estas personas deben evaluar las peticiones de cambio, aceptándolas o rechazándolas.
Responsable de GC:	es la persona encargada de la planificación de la configuración. Además es su responsabilidad definir las líneas base y asegurar su seguimiento. Como actividades secundarias, también participa en la implantación del producto con el cliente y es el responsable del control de cambios, es decir, debe reportar los cambios no autorizados e identificar y controlar los cambios en los CI. Es también el encargado de aprobar los cambios estructurales en la base de datos de configuración.
Bibliotecario:	es el encargado de establecer y mantener el software y la documentación de cada proyecto de acuerdo a un proceso documentado. Debe proveer a los desarrolladores copias sobre las líneas base del proyecto e informarles sobre los cambios que surjan en relación a los elementos de configuración.

**Resto personal
desarrollador:**

este equipo debe revisar y realizar observaciones sobre el SCM, ya que posteriormente deberá de implementar las actividades de acuerdo al plan. Deberán participar también en la solución a los problemas del SCM que sean de su competencia. Por último, deberán implementar las prácticas, procesos y procedimientos definidos en el plan del proyecto y en otros planes o documentos complementarios.

Responsable	Ocupación
Juan Abascal	SCMR
Adriana Lima	CC
Alberto García	Bibliotecario
Carlos Olivares Sanchez- Manjavacas, Carlos Tormo Sánchez, Irina Shayk	Equipo de desarrollo

Cuadro 1.1: Responsabilidades.

II.3

POLÍTICAS, DIRECTIVAS Y PROCEDIMIENTOS APLICABLES

Durante el proceso de documentación se y desarrollo de todo el proyecto se van a utilizar la herramienta GIT, almacenando el proyecto en un repositorio privado de la empresa BitBucket, permitiendo un control de versiones muy eficiente. Toda la documentación será llevada a cabo en el lenguaje LaTeX, por lo que utilizaremos también GIT para su control de versiones. Sin embargo, con el fin de facilitar el desarrollo de la documentación, se realizarán en Google Docs varias versiones, para posteriormente pasarlas a LaTeX.

- **Políticas de configuración de código fuente y documentación de usuario:** se utilizará un repositorio privado en la empresa BitBucket, que implementa un sistema GIT, para llevar a cabo un control de las versiones. Cada *commit* será explicativo, lo que ayudará a todo el equipo.
- **Política de almacenamiento:** se utilizará Google Drive junto con Google Docs para las primeras versiones de la documentación. Una vez el contenido del documento sea firme, será transcrito a LaTeX y almacenado junto con el código fuente en el repositorio de BitBucket.
- **Políticas de cambios:** los documentos únicamente podrán ser modificados por el responsable de la gestión de configuración (CCR) y solo cuando el Comité de Control de Cambios lo estime oportuno. Cualquier miembro del equipo podrá proponer una solicitud para cambiar o revisar cualquier parte del proyecto. Esta solicitud será comunicada al CCR y posteriormente se remitirá al CCC.
- **Política de confidencialidad:** todo los documentos relacionados con el proyecto, ya sean para uso interno del equipo o para el cliente tendrán un carácter confidencial.

III

ACTIVIDADES DE GESTIÓN DE CONFIGURACIÓN

A lo largo de este apartado se explicarán las actividades de Gestión de Configuración del Software (SCM) que se van a realizar durante el desarrollo del proyecto CARSEAFITY. Para ello, primero se hará una identificación de la configuración y finalmente se detallará el control de cambios de la configuración.

III.1

IDENTIFICACIÓN DE LA CONFIGURACIÓN

A continuación se identificará y se hará una descripción detallada de todos aquellos elementos que deban ser considerados Elementos de Configuración (EC).

III.1.1

SE ESTABLECE LA JERARQUÍA PRELIMINAR DEL PRODUCTO

En este apartado se estudia y muestra una primera visión de la estructura del sistema software. Esta estructura está definida por sistemas y subsistemas, así como por las relaciones entre cada uno de ellos. En la figura 1.1 podemos ver un esquema simplificado de esta estructura.

A continuación se describen los sistemas y subsistemas que entran en juego en el sistema software:

Subsistema <i>pre-crash</i>:	este subsistema se encarga del control de posibles colisiones y la disminución de la velocidad o parada del vehículo en caso de alta probabilidad de colisión.
Subsistema de alerta de velocidad:	este subsistema se encarga de alertar al conductor si la velocidad instantánea del vehículo supera la velocidad máxima permitida en la vía.
Subsistema de control de punto ciego:	este subsistema se encarga de advertir al conductor de la presencia de vehículos en el lateral trasero del vehículo (en el conocido como punto ciego).
Subsistema de control de la fatiga:	este subsistema se encarga de avisar al conductor en caso de que este pierda la atención al volante a causa de la fatiga o similar.
Subsistema de cambio de carril:	este subsistema es el encargado de comparar la trayectoria del vehículo con la de la vía y advertir al conductor si el vehículo se desvía de la trayectoria que debe seguir.
Subsistema de llamada de emergencia:	este subsistema se encarga de realizar una llamada a los servicios de emergencia en caso de que sea necesario.
Sistema de comunicación central:	este sistema se encarga de comunicar el resto de subsistemas con el sistema central de CARSEAFETY.
Sistema CARSEAFETY:	este sistema es el encargado de centralizar la información, mostrarla al conductor y tomar las decisiones sobre las medidas a tomar en cada caso.



Figura 1.1: Estructura de los sistemas y subsistemas

III.1.2

SELECCIÓN DE LOS ELEMENTOS DE CONFIGURACIÓN

En este apartado se identifican los Elementos de Configuración (EC). Éstos son las unidades que se deben poder definir y controlar de forma separada e independiente unos de otros. Así, se corresponden en este proyecto con los productos de las tareas de la metodología de Craig Larman [1] y , de forma complementaria, de Métrica3 [2].

Pueden verse en la tabla 1.2, en la que se indica el nombre del EC identificado.

NOMBRE

Documento de oferta

Documento de control de costes

Estudio de Viabilidad del Sistema

Continúa en la siguiente página

NOMBRE

Plan de Gestión de la Configuración

Plan de Calidad

Diagrama de casos de uso

Matriz de trazabilidad

Casos de uso de alto nivel

Priorización de casos de uso

Casos de uso en formato expandido

Arquitectura del sistema

Estimación

Planificación y seguimiento

Modelo de clases

Contratos de operación

Diagramas de secuencia

Gestión de cambios

Estándar de implementación

Ejecutable de implementación

Diagrama de estados

Reporte de pruebas

Presentación del sistema

Informes quinquenales de seguimiento

Cuadro 1.2: Elementos de Configuración.**III.1.3****SELECCIÓN DEL ESQUEMA DE IDENTIFICACIÓN**

Tras la identificación de los ECs del apartado anterior, es necesario escoger un esquema de identificación para poder referenciarlos a lo largo tanto del presente documento como del proyecto.

Para ello, hemos decidido usar una identificación no significativa. Los motivos para esta elección se fundamentan en dos aspectos básicos: por un lado, la fácil asignación de código identificativo; en segundo lugar, el uso de un medio electrónico para el desarrollo del proyecto, lo que suple el déficit de la identificación de identificativa.

Mediante el uso de hipervínculo se puede referenciar cada uno de los EC a pesar de que su nombre no sea identificativo.

Por ello, cada uno de los EC seleccionados en el apartado anterior, sus variantes y versiones serán identificados mediante cuatro dígitos anteceditos por las letras EC.

Por otro lado, la descripción de los elementos de configuración constará del código identificativo, el nombre del EC, su descripción, la iteración en la que surgió o fue identificado, la fecha de creación y el código identificativo de la línea base. Puede verse un ejemplo de esta tabla descriptiva en la tabla 1.3.

Código Identificativo:	Código	Nombre:	Nombre del EC
Iteración:	Iteración	Fecha de creación:	dd/mm/aaaa
Descripción:	Descripción del EC		
Línea base:	Línea base del EC		

Cuadro 1.3: Ejemplo de la Descripción de un EC.

III.1.4

DEFINICIÓN DE RELACIONES

Tras definir los diferentes EC y saber como identificarlos, debemos definir e identificar las relaciones que existen entre ellos. Ello sirve para poder conocer qué elementos de configuración se ven afectados con los posibles cambios de otros elementos de configuración, y así poder delimitar los posibles impactos que se puedan producir.

En nuestro proyecto contemplamos varias posibles relaciones:

- Dependencia (DEP):** relación que se da cuando un EC tiene relación directa con otro EC. Es decir, si un EC cambia, el que tenga relación directa con él se verá afectado y podrá resultar modificado.
- Derivación (DER):** relación que se da cuando un EC requiere que haya terminado otro anterior.
- Sucesión (SUC):** relación de histórico de cambios sobre un EC de una revisión a otra.
- Variante (VAR):** relación de las distintas variaciones que hay sobre un mismo EC.
- Composición (COMP):** relación que se da cuando un EC está compuesto por un conjunto de ECs.

Una vez vistas las distintas relaciones que se pueden dar entre dos EC, pasamos a proponer un esquema de identificación de las mismas. Lo primero que se indicará, tal y como se ve en la figura 1.2, serán los identificadores de los dos EC relacionados separados por un guión. A continuación, y mediante la separación por otro guión, se indicará la abreviatura de la relación que los uno vista en el párrafo anterior.



Figura 1.2: Esquema de identificación de relación entre ECs.

En cuanto a las relaciones y al esquema mencionado, se deben tener varios aspectos en cuenta:

- Si la relación es bidireccional (véase Relación de Dependencia), el orden de los identificadores que se indique en el esquema de la figura 1.2 es indiferente.
- Si la relación es unidireccional, el orden de los elementos de configuración en el formato sí es significativo. En la Relación de Derivación (DER), EC1-EC2-DER implica que EC2 deriva de EC1. En el caso de la Relación de Sucesión (SUC), EC1-EC2-SUC implica que EC2 surge de una revisión de EC1, por lo que EC2 es el *hijo* de EC1.
- Si la relación es de Composición (COMP), se añadirán tantos identificadores como sean necesarios al esquema de la figura 1.2, siendo EC1 el EC que se compone de los EC2 a ECn.
- Si la relación es de Sucesión (SUC), EC1-EC2-SUC implica que EC1 es una versión actualizada de EC2. Véase que al ser un identificador no significativo, cada versión de un mismo EC tiene un identificador distinto.
- Si la relación es Variante (VAR), EC1-EC2-VAR implica que EC2 es una variante de EC1.

III.1.5

DEFINICIÓN Y ESTABLECIMIENTO DE LÍNEAS BASE

En este apartado se recogen las diferentes líneas base del proyecto. Como se puede ver en la tabla 1.4 para cada una de ellas recogemos su nombre, su descripción, su estado (que será o bien *abierta* o bien *cerrada*), la fecha y los identificadores de los EC que la componen.

Nombre: Nombre de la línea base			
Descripción: Descripción de la línea base			
Estado:	Abierta/Cerrada	Fecha de creación:	dd/mm/aaaa
EC1	Identificador EC1	EC2	Identificador EC2
EC3	Identificador EC3	ECn	Identificador ECn

Cuadro 1.4: Ejemplo de una Línea Base.

Las líneas base definidas en este proyecto son:

- Fase de Documentación
- Fase de Planificación y Especificación de requisitos
- Fase de Análisis de la primera iteración
- Fase de Diseño de la primera iteración
- Fase de Análisis de la segunda iteración
- Fase de Diseño de la segunda iteración
- Fase de Análisis de la tercera iteración
- Fase de Diseño de la tercera iteración
- Fase de Implementación

III.1.6

DEFINICIÓN Y ESTABLECIMIENTO DE BIBLIOTECAS DE SOFTWARE

A lo largo de este apartado se definirán las bibliotecas de software que serán usadas durante el proyecto. Así mismo, se establecerá su ubicación exacta dentro del proyecto. Así, y como se verá en este apartado, incluiremos bibliotecas de documentación, de soporte, de desarrollo, de *backup* y una maestra, entre otras.

Así, las bibliotecas establecidas se ubicarán todas ellas dentro de la siguiente ruta:

```
1 $PATH:$ /User/carseafty/
```

Puede verse un esquema del sistema de librerías en la figura 1.4. Las bibliotecas de nuestro proyecto serán:

Documentación: contendrá la documentación del proyecto. Esta documentación estará recogida en cuatro subgrupos diferentes: la documentación relativa a la oferta y los costes, la documentación de Estudio de Viabilidad del Sistema, la documentación de calidad y la documentación de configuración.

La ubicación de cada una de estas carpetas será:

```
1 $PATH:$ /User/carseafty/documentation/offer
2 $PATH:$ /User/carseafty/documentation/evs
3 $PATH:$ /User/carseafty/documentation/quality
4 $PATH:$ /User/carseafty/documentation/configuration
```

Desarrollo: esta biblioteca recoge el trabajo desarrollado por el equipo de desarrollo. El código y documentación aquí almacenado está en proceso, pero no acabado. Una vez estén finalizados, pasarán a la biblioteca de producción. Esta biblioteca está dividida en las siguientes bibliotecas, correspondientes a los subsistemas del proyecto:

```
1 $PATH:$ /User/carseafty/development/precrash
2 $PATH:$ /User/carseafty/development/velocity
3 $PATH:$ /User/carseafty/development/laneChange
4 $PATH:$ /User/carseafty/development/blindSpot
5 $PATH:$ /User/carseafty/development/ecall
6 $PATH:$ /User/carseafty/development/fatigue
7 $PATH:$ /User/carseafty/development/comunication
```

Producción: esta biblioteca contiene las líneas base creadas y acabadas. La ubicación exacta es:

```
1 $PATH:$ /User/carseafty/production
```

Maestra: esta biblioteca contiene la versión actual del cliente. La ubicación exacta es:

```
1 $PATH:$ /User/carseafty/production/master
```

Proves: esta carpeta contiene las pruebas realizadas sobre las distintas versiones del proyecto. Cada una de las versiones probadas tendrá una subcarpeta dentro de:

```
1 $PATH:$ /User/carseafty/proves
```

Backup: almacena una copia de seguridad. La ubicación exacta es:

```
1 $PATH:$ /User/carseafty/buckup
```

Como se indicó en el apartado [II.3](#), se utilizará BitBucket para el almacenamiento de la documentación y el código. Así, se especificarán los permisos de cada una de las bibliotecas en función de la tabla [??](#). Cabe destacar que si un miembro del equipo tiene permisos sobre una biblioteca de nivel superior, ello implica que tiene permisos sobre todas las bibliotecas de niveles inferiores.

III.2

CONTROL DE CAMBIOS EN LA CONFIGURACIÓN

III.2.1

PROCEDIMIENTO APLICABLE

A lo largo de esta sección detallamos las actividades de solicitud, evaluación, aprobación e implementación de los cambios solicitados y realizados en los elementos de la configuración. El procedimiento descrito a continuación es el que se utilizará cada vez que se precise introducir un cambio en el sistema.

Este proceso se describe en la figura [1.3](#). Atendiendo a la misma, los pasos que se deben seguir son los siguientes:

1. Cuando un posible cambio en un EC es identificado, se debe generar una instancia del Informe de Solicitud de Cambios (véase [III.2.2](#)). Este debe recoger toda la información necesaria para justificar dicho cambio. Así mismo, es necesario que sea lo más detallado posible. Este informe será colocado en la carpeta de *configuration*, dentro de la carpeta de *changes* y, más concretamente, en la subcarpeta de *inProgress*. La ruta concreta sería:

```
1 $PATH:$ /User/carseafty/documentation/configuration/changes/inProgress
```

2. Una vez realizado el Informe de Solicitud de Cambio y registrada la petición, se debe revisar si esta está completa o no. Esta revisión será en parte prescindible si el formulario es electrónico y no permite el envío del mismo en caso de estar incompleto. Tras realizar esta comprobación, se debe evaluar la solicitud. Esta evaluación debe tener en cuenta los ECs afectados y el posible impacto de los cambios solicitados. Así, el Comité de Control de la Configuración debe decidir si aceptar o no la petición correspondiente.
3. En caso de que el CGC decida rechazar el cambio, la petición (el ISC) deberá ser trasladado a la carpeta:

```
1 $MOVE TO:$ /User/carseafty/documentation/configuration/changes/evaluated
```

Además, se creará un Informe de Certificación de Cambios que se almacenará en la misma carpeta indicando el motivo de la denegación.

4. En caso de que el CGC decida aceptar la petición, deberá redactarse el ICC, indicando los aspectos antes detallados, y se almacenarán en la carpeta:

```
1      $PATH:$ /User/carseafty/documentation/configuration/changes/inProgress
```

5. A continuación, si el cambio ha sido aceptado, se debe diseñar y planificar el cambio para decidir cuándo llevar a cabo su implementación. Dicha implementación será ejercida tanto en el elemento de la configuración sobre el que se solicitó el cambio como en los EC que estén relacionados con éste y cuyo cambio afecte.
6. Una vez el cambio sea realizado, se moverán ambos archivos (el ICC correspondiente y el ISC del cambio aplicado) a la carpeta:

```
1      $MOVE TO:$ /User/carseafty/documentation/configuration/changes/evaluated
```

III.2.2

FORMATO DEL INFORME DE SOLICITUD DE CAMBIOS

Puede encontrarse el Informe de Solicitud de Cambios en esta URL: <https://carlososm.typeform.com/to/cuwCDI>.

III.2.3

FORMATO DEL INFORME DE CERTIFICACIÓN DE CAMBIOS

Puede encontrarse el Informe de Certificación de Cambios en esta URL: <https://carlososm.typeform.com/to/DVPDc6>.

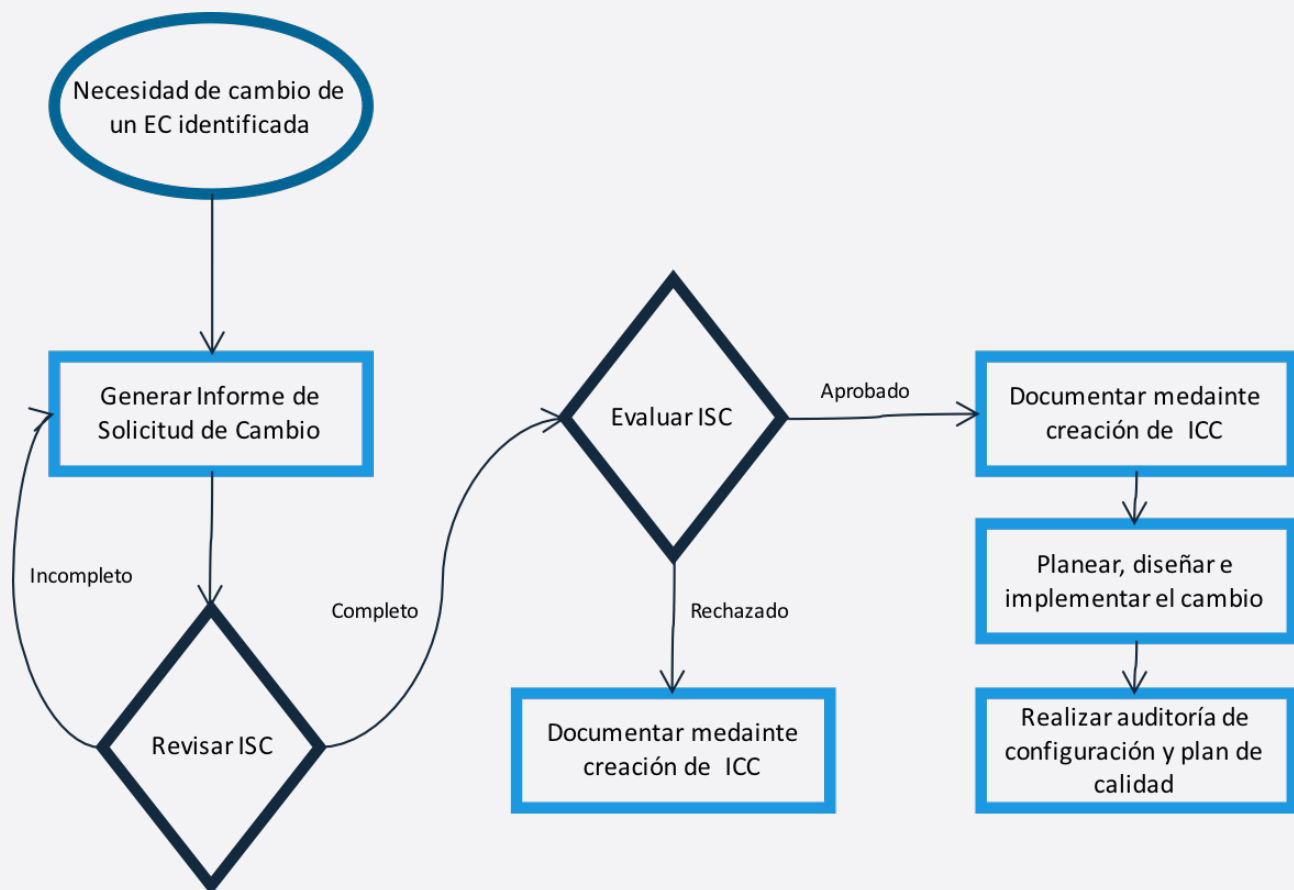


Figura 1.3: Procedimiento General de Control de Cambios.

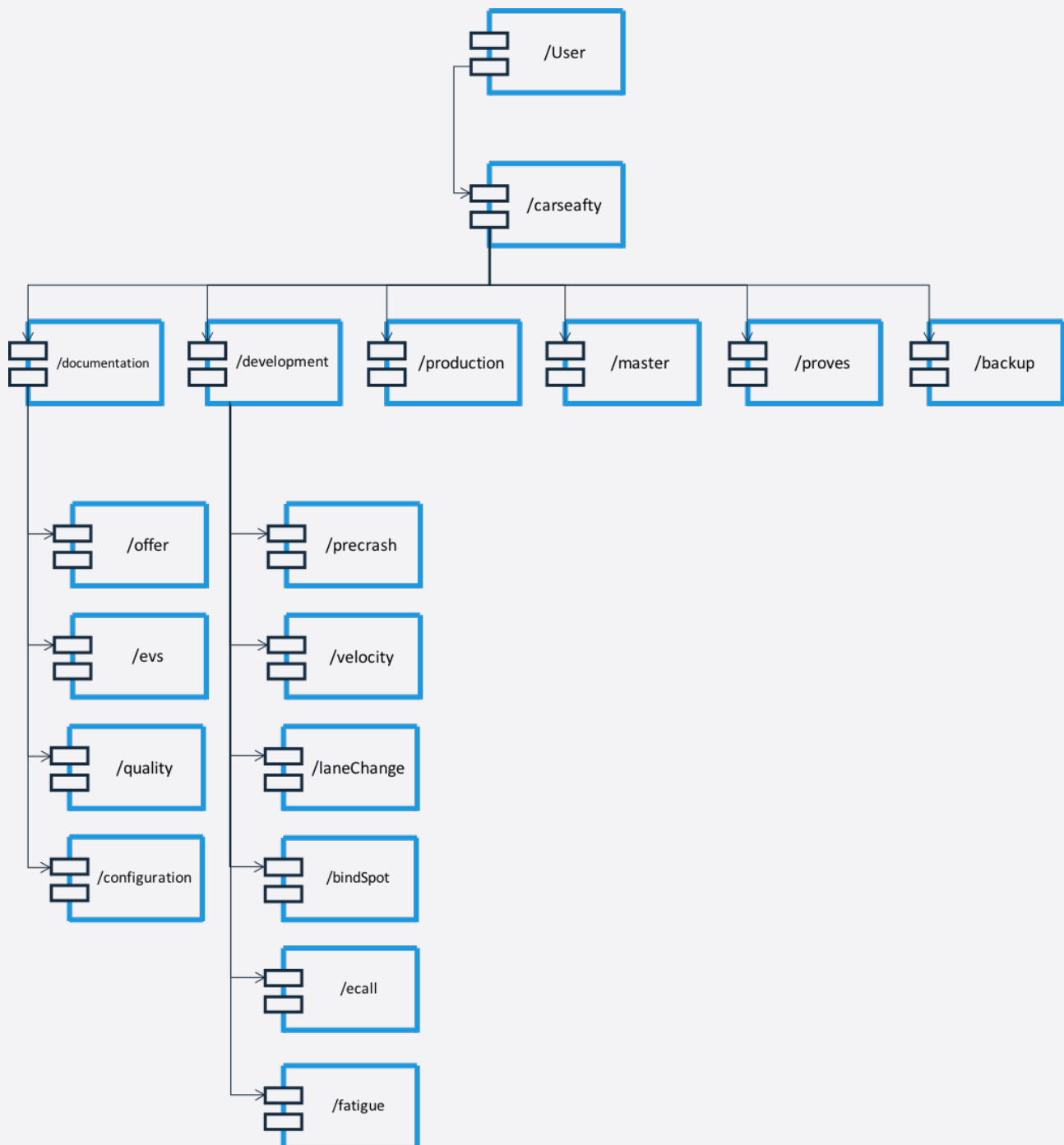


Figura 1.4: Esquema de identificación de relación entre ECs.



ESTADO ACTUAL DEL PLAN DE GESTIÓN DE LA CONFIGURACIÓN

I DEFINICIÓN DE LOS EC HASTA EL MOMENTO

En este apartado se incluyen los elementos de configuración detectados en el proyecto hasta el momento actual. A medida que se avance en el desarrollo del proyecto y que se vayan generando nuevos elementos de la configuración, se irán añadiendo. Así, puede verse en la tabla [A.1](#) los EC generados hasta el momento:

ENTREGA	FECHA	IDENTIFICADOR
Documento de oferta 1	03/03/17	EC0001
Documento de control 1 de costes	03/03/17	EC0002
Estudio de Viabilidad del Sistema 1	24/02/17	EC0003
Documento de oferta 2	24/03/17	EC0004
Documento de control 2 de costes	24/03/17	EC0005
Estudio de Viabilidad del Sistema 2	24/03/17	EC0006
Plan de Calidad	07/04/17	EC0007
Casos de uso de alto nivel	07/04/17	EC0008
Diagrama de casos de uso	07/04/17	EC0009
Matriz de trazabilidad	07/04/17	EC0010
Priorización de casos de uso	07/04/17	EC0011

Continúa en la siguiente página

ENTREGA	FECHA	IDENTIFICADOR
Estimación	07/04/17	EC0012
Planificación y seguimiento	07/04/17	EC0013
Informe quinquenal de seguimiento 1	07/04/17	EC0014
Plan de Gestión de la Configuración	07/04/17	EC0015

Cuadro A.1: Elementos de Configuración identificados hasta el momento.

II DEFINICIÓN DE LAS RELACIONES DE LOS ACTUALES EC

Entre los EC descritos en el apartado anterior se detectan las siguientes relaciones.

II.1 RELACIONES DE SUCESIÓN

- EC0001-EC0004-SUC
- EC0002-EC0005-SUC
- EC0003-EC0006-SUC

II.2 RELACIONES DE DERIVACIÓN

- EC0001-EC0003-DER
- EC0002-EC0004-DER
- EC0004-EC0006-DER
- EC0005-EC0006-DER
- EC0006-EC0009-DER
- EC0008-EC0012-DER
- EC0008-EC0011-DER

II.3 RELACIONES DE DEPENDENCIA

- EC0006-EC0009-DEP
- EC0009-EC0008-DEP
- EC0008-EC0012-DEP
- EC0008-EC0011-DEP

II.4 RELACIONES DE VARIANTE

No existen relaciones de variante en estos momentos.

II.5 RELACIONES DE COMPOSICIÓN

No existen relaciones de composición en estos momentos.

III DEFINICIÓN DE LAS LÍNEAS BASE ACTUALES

Por último, definimos las líneas base que se han creado hasta el momento. Éstas líneas base pueden estar ya finalizadas o que todavía queden elementos de configuración por incluirse. Es decir, a medida que se vayan incluyendo elementos de configuración en el apartado primero de este anexo se tendrán que ir incluyendo en las líneas base correspondientes. Dichas líneas base puede que ya estén creadas o puede que haya que crear una nueva. Las líneas base generadas hasta el momento son las reflejadas en las tablas [A.2](#) y [A.3](#).

Nombre: Línea base de Documentación

Descripción: Línea base de la documentación perteneciente a la aplicación de Métrica 3 [\[2\]](#).

Estado: Abierta	Fecha de creación: 05/02/2017
EC0004 Documento de Oferta 2	EC0005 Documento de control de costes 2
EC0006 Estudio de la Viabilidad del Sistema 2	EC0007 Plan de Calidad
EC0015 Plan de Gestión de la Configuración	

Cuadro A.2: Línea Base Documentación actual.

Nombre: Línea Base Planificación y Especificación de requisitos actual

Descripción: Elementos de configuración pertenecientes a la fase de planificación y especificación de requisitos de la metodología Craig-Larman [\[1\]](#).

Estado: Abierta	Fecha de creación: 27/03/2017
EC0009 Diagrama de casos de uso	EC0008 Casos de uso de alto nivel
EC0010 Matriz de trazabilidad	EC0011 Priorización de casos de uso
EC0012 Estimación	EC0013 Planificación y seguimiento

Cuadro A.3: Línea Base Planificación y Especificación de requisitos actual.

Índice de figuras

1.1. Estructura de los sistemas y subsistemas	8
1.2. Esquema de identificación de relación entre ECs.	10
1.3. Procedimiento General de Control de Cambios.	15
1.4. Esquema de identificación de relación entre ECs.	16

Índice de cuadros

1.1. Responsabilidades.	6
1.2. Elementos de Configuración.	9
1.3. Ejemplo de la Descripción de un EC.	10
1.4. Ejemplo de una Línea Base.	11
A.1. Elementos de Configuración identificados hasta el momento.	18
A.2. Línea Base Documentación actual.	19
A.3. Línea Base Planificación y Especificación de requisitos actual.	19

BIBLIOGRAFÍA

- [1] X. F. Grau and M. I. S. Segura, “Desarrollo orientado a objetos con uml,” Facultad de Informática - UPM; Escuela Politécnica Superior - UC3M, vol. 1, 2008.
- [2] U. C. I. de Madrid, Revisión de Métrica Versión 3. Versión subida en www.aulaglobal.uc3m.es.