

VERSIÓN 1.0

PROYECTO SCP

SISTEMA DE CONTROL DE PESO



Systems and Solutions

Proyecto SCP

Sistema de gestión de peso
Plan de Desarrollo Software y Documentación Técnica

Versión 1.0

Hoja de control del documento

Título	Plan de desarrollo de software y documentación técnica. Sistema de control de peso SCP		
Descripción	SCP es una aplicación en red desarrollada para facilitar el control de trazabilidad del transporte en las carreteras intradepartamentales, a través de registros de datos para su posible análisis.		
Autor	Víctor Paúl Téllez Pérez		
Responsable	Ing. Juan Gabriel Bermúdez Hernández		
Organización	System and Solutions.		
Categoría	Documentación técnica		
Palabras clave	Trazabilidad del transporte, software, SCP		
Versión	1.0	Fecha	19 de Noviembre de 2020
Estadísticas	2708480 bytes, 105562 caracteres, 59 páginas, 1859 párrafos, 16121 palabras		
Edición	1 revisiones		

Historial de revisiones

Fecha	Versión	Descripción	Autor
2020-11-16	1.0	Versión inicial	Víctor Paúl Téllez Pérez

Tabla de contenidos

1	Introducción	7
1.1	Propósito	7
1.2	Usuarios	7
1.3	Alcance.....	8
1.4	Resumen	8
1.5	Advertencias.....	9
1.6	Glosario, acrónimos y definiciones.....	9
2	Vista general del proyecto	11
2.1	Propósito, alcance y objetivos	11
2.2	Restricciones y suposiciones	12
2.2.1	Restricciones	12
2.2.2	Suposiciones	13
2.3	Aclaraciones previas	13
2.3.1	Especificación.....	14
2.3.2	Diseño y arquitectura.....	14
2.3.3	Programación	14
2.3.4	Prueba	14
2.3.5	Mantenimiento	15
2.4	Ciclo de vida.....	15
2.5	Normativa y protocolos aplicables al ciclo de vida	16
2.6	Metodología.....	18
2.7	Entregables del proyecto.....	22
3	Organización del proyecto.....	24
3.1	Recursos humanos	24
3.2	Jerarquía RRHH.....	25
4	Gestión de procesos en el proyecto	27
4.1	Pendiente	27
5	Infraestructura.....	28
6	Estructura física de la información	29
7	Estructura lógica de la información	33
7.1	Elementos	33
8	Diseño de la arquitectura del sistema	35
8.1	Descripción general.....	35
8.2	Ingeniería de requisitos	39
8.2.1	Requisitos funcionales.....	39
8.2.2	Requisitos organizacionales.....	40
8.2.3	Requisitos no funcionales.....	40
8.3	Implementación de la solución	40
9	Resumen de procesos del sistema	41
9.1	Actores y casos de uso. Modelos básicos	41
9.1.1	Actores.....	42

9.1.2	Casos de uso.....	42
9.2	Procesos	44
9.2.1	Procesos principales	44
9.2.2	Procesos auxiliares	44
10	Interfaces del sistema	45
10.1	Back-end	45
10.2	Front-end.....	46
11	Gestión de riesgos	47
12	Verificación de la calidad.....	49
12.1	Evaluación del nivel de calidad	50
13	Anexo gráfico.....	51
14	Anexo documental.....	58
14.1	Normativa aplicada.....	59
14.2	Bibliografía	59

Tabla de ilustraciones

Ilustración 1. Jerarquía metodológica en SCP	19
Ilustración 2. Ciclo SCRUM.....	20
Ilustración 3. Desarrollo secuencial y solapado.....	21
Ilustración 4. Disciplinas y fases RUP en SCP	23
Ilustración 5. Interacciones entre actores en SCP	27
Ilustración 6. Modelo simplificado SI SCP	37
Ilustración 7. Procesos GC según PMI	49
Ilustración 8. SCP Conexión. Mapa conceptual.....	51
Ilustración 9. SCP: Modelo de Datos.....	51
Ilustración 10. Registro de Vehículos.....	52
Ilustración 11. Registro. Empresa de transporte. Relaciones	52
Ilustración 12. Registro PERSONA. Relaciones	53
Ilustración 13. Registro Punto de control. Relaciones	53
Ilustración 14. Registro Interfaz Front-End. Relaciones.....	54
Ilustración 15. Registro Interfaz Front-End. Catalogos	54
Ilustración 16. Registro Interfaz Front-End. Pesadas.....	55
Ilustración 17. Registro Interfaz Front-End. Reportes	55
Ilustración 18. Registro Interfaz Front-End Detalle de Reportes.....	56
Ilustración 19. Diagrama UML genérico. Modelo de casos de usos	57

Esta página ha sido intencionadamente dejada en blanco.

1 Introducción

El presente documento (desde ahora, PDSDT) está destinado a describir el plan de desarrollo (en adelante PDS) de la aplicación denominada Sistema de Control de Peso (en adelante SCP) y sus especificaciones técnicas (en adelante DT). Este documento provee una visión global del enfoque de desarrollo propuesto y una descripción abreviada de las mencionadas especificaciones.

El proyecto ha sido realizado y descrito en PDSDT basándose en una estructura de tipo *Rational Unified Process* (RUP), que junto con el Lenguaje Unificado de Modelado UML constituye la metodología estándar más utilizada para el análisis, diseño, implementación y documentación de sistemas orientados a objetos. Por ello PDSDT utiliza la terminología propia de dicha metodología e incluye el modelo del negocio y el alcance de SCP, identifica los actores y casos de uso y esboza un plan de negocio para determinar qué recursos deben ser asignados a cada tarea.

El enfoque de desarrollo propuesto constituye una configuración del proceso RUP de acuerdo a las características de SCP, seleccionando los roles de los participantes, las actividades a realizar y los artefactos (entregables) que serán generados. PDSDT es, a su vez, uno de los artefactos RUP.

1.1 Propósito

El propósito de PDS es proporcionar la información necesaria para controlar el proyecto SCP y describir su funcionalidad, además de dotarlo de una base teórica susceptible de revisión, modificaciones y mejoras. El propósito de DT es describir con cierto nivel de detalle las soluciones técnicas adoptadas para implementar adecuadamente PDS de acuerdo a lo en él estipulado.

Téngase en cuenta que existe documentación adicional detallada, que incluye el código fuente y los diagramas UML completos empleados durante el proceso de desarrollo que no figura en PDSDT. Los objetivos esenciales son:

- Documentar textualmente de forma sucinta pero clara los elementos esenciales que configuran el proyecto SCP.
- Referenciar las fases esenciales del proyecto: análisis previo, desarrollo primario, pruebas funcionales, puesta en marcha y corrección de errores y/o disfuncionalidades.
- Informar sobre el estado actual de la ejecución del proyecto, situándolo con respecto a la planificación inicial.
- Evaluar el grado de ajuste del estado actual mencionado en el punto anterior y las posibilidades de conclusión exitosa del proyecto SCP en tiempo y forma.

1.2 Usuarios

PDSDT está concebido como herramienta de trabajo y referencia, y no como texto con finalidad divulgativa o para usuarios finales. La audiencia prevista para el mismo, por tanto, tiene un marcado perfil técnico. Los actores previstos para la consulta son:

- *Responsables SCP*: como herramienta de seguimiento y para control del proceso de desarrollo.
- *Desarrolladores*: como guía de actividades, calendario de fases y ejecución.
- *Auditores*: para verificar el ajuste de lo proyectado a la realidad de la implementación.
- *DBA*: para el manejo y control de los datos recopilados en las bases de datos.

13 Alcance

PDSDT no pretende abarcar la descripción exhaustiva de ninguno de los apartados PDS y DT. Para ello se establece como artefacto una posterior versión, cuya entrega está prevista a la finalización del desarrollo del proyecto SCP. El detalle de las iteraciones se describe en los planes individuales de cada una, que figurarán detalladamente en el artefacto final y que ahora se mencionan en 4.2. La base para la planificación de las citadas iteraciones reside en la estructura de subprocesos que se describe en DT (véase 9).

Dadas las características específicas del proyecto SCP (dependiente del desarrollo de SCP), la captura de información para ingeniería de requisitos no se ha realizado, como suele ser habitual, a través de un *stakeholder* representante del MTI (en adelante RMTI) sino que ha sido realizada a partir de las estipulaciones de los actores responsables, que detentan, en este caso concreto, la calidad de responsables también en el MTI. Por tanto, los requerimientos se han establecido en un primer momento mediante una estimación aproximada, y una vez comenzado el proyecto y durante la fase inicial se ha ido refinando a partir de las entregas de prototipos de artefactos. Está prevista la edición de sucesivas versiones de PDSDT (anteriores a la final) para adecuarse a tales circunstancias.

14 Resumen

De acuerdo a lo especificado, PDSDT se organiza en secciones según el siguiente esquema operativo, en el que cada apartado va precedido por el ámbito al que pertenece (PDS o DT):

- **PDS. Vista general del proyecto.** Proporciona una descripción del propósito, alcance y objetivos de SCP, estableciendo los artefactos que se producen y utilizan durante la ejecución del proyecto. Se incluye el modelo de ciclo de vida adoptado y la metodología de desarrollo.
- **PDS. Organización del proyecto.** Describe la estructura organizacional y funcional de los recursos humanos implicados en el desarrollo de SCP.
- **PDS. Gestión de procesos del proyecto.** Explica los costos temporales y la planificación estimada, define las fases e hitos de SCP y describe cómo se realiza su seguimiento.
- **PDS. Infraestructura.** Estipula los requisitos mínimos necesarios para poder implementar la solución propuesta de manera que se garantice el adecuado funcionamiento del conjunto.
- **DT. Estructura física de la información.** Describe la organización de las tablas y relaciones de la BD.
- **DT. Estructura lógica de la información.** Dadas las características específicas de la implementación de SCP, establece la significación de cada elemento semántico en el conjunto de datos almacenados.
- **DT. Diseño de la arquitectura del sistema.** Descripción de los patrones y *frameworks* utilizados y las razones para ello.
- **DT. Resumen de procesos del sistema.** Descripción de los procesos y subprocesos implicados y sus referencias al modelo de datos.

- **DT. Interfaces del sistema.** Descripción sucinta de los interfaces principales con mención expresa de los actores implicados.
- **DT. Componentes SW.** Descripción de los diferentes componentes software empleados y sus relaciones. Se incluyen las referencias a librerías externas y SW de terceros.
- **DT. Gestión de riesgos.** Descripción del plan de seguridad activa y proactiva.
- **DT. Verificación de la calidad.** Descripción de los mecanismos de certificación y validación de la información almacenada.

15 Advertencias

Por incluir los apartados correspondientes a PDS y DT en el mismo documento, cabe la posibilidad de que determinados apartados de PDSDT puedan prestarse a confusión o no resulten claros. La correcta interpretación del presente texto debe realizarse tomando en cuenta las llamadas y referencias cruzadas de unos apartados a otros, donde se contiene la información significativa. Por ejemplo, puesto que es imposible realizar una estimación de esfuerzo basada en casos de uso si no se conoce la estructura de datos, los procesos y los casos mismos, es conveniente, tal y como se establece en 4.1, consultar la información suministrada en 7 y 9.

16 Glosario, acrónimos y definiciones

Con el fin de clarificar la terminología utilizada en el presente documento, se adjunta un breve prontuario de acrónimos, siglas y términos de uso específico.

Baseline	Registro estático del estado de cada artefacto al final de cada iteración, estableciendo una versión o <i>release</i>
Bug	En el contexto de desarrollo, error descubierto en un programa
CDU	Caso de uso
RMTI	Actor que realiza el papel de representante del MTI o "cliente" (no en el sentido comercial de la palabra)
CMS	<i>Content Management System</i>
CRUD	<i>Create, Read, Update, Delete</i> . Acrónimo del proceso de gestión de registros en una base de datos
Changelog	Registro detallado de los cambios realizados en los artefactos
CBS	<i>Control de Basculas</i>
DP	<i>Data Person</i> .
PCT	Punto de control o estación de bascula
DRAE	Diccionario de la Real Academia Española
DT	Documentación Técnica
SCP	Sistema de control de Peso
S&S	System and Solutions
MTI	Ministerio de Transporte e Infraestructura
FW	<i>Framework</i> . Terminología de arquitectura de desarrollo
GC	Gestión de calidad

HW	<i>Hardware</i>
ISO	<i>International Organization for Standarization</i>
KISS	<i>Keep It Simple, Stupid.</i> Lema de trabajo que aboga por la implantación de la “navaja de Oc-kham” como principio esencial en la adopción de soluciones
LP	<i>Lean Programming.</i> Acrónimo de un tipo de metodología ágil
OS	<i>Operating System</i>
OSI	<i>Open Source Initiative.</i> También, según el contexto, puede aludir al modelo <i>Open System Interconnection</i> descrito por ISO/IEC 7498-1
PDS	Proyecto de Desarrollo de Software
PDSDT	Documento actual (Proyecto de Desarrollo de Software y Documentación Técnica)
PMBOK	<i>Project Management Body of Knowledge.</i> Estándar desarrollado por el <i>Project Management Institute</i> para gestionar los proyectos y medir su calidad y éxito
PMI	<i>Project Management Institute</i>
POO	Programación Orientada a Objetos. También conocido por sus siglas inglesas, OOP
RAD	<i>Rapid Application Development</i>
Red	Con mayúscula inicial, Internet
Release	Término inglés para <i>lanzamiento</i> o liberación de una versión de un producto SW o “artefac-to”, en terminología RUP
RRHH	Recursos humanos
RUP	<i>Rational Unified Process</i>
SaaS	<i>Software as a Service</i>
SC	Solicitud de cambio. En contexto de procesos (véase 9), Sistema de Captura
SCRUM	Metodología organizativa relacionada con los sistemas ágiles de desarrollo
SEO	<i>Search Engine Optimization</i>
SESC	<i>Software Engineering Standards Committee</i>
SGBD	Sistema Gestor de Bases de Datos. También se conoce como DBMS
SI	Sistema de Información
SPICE	<i>Software Process Improvement Capability Determination</i>
Stakeholder	Parte interesada en el proceso de desarrollo, como interno o como CL, en terminología SCRUM
SW	<i>Software</i>
UML	<i>Unified Modeling Language</i>
Workflow	O <i>flujo de trabajo</i> . Estudio de los aspectos operacionales de una actividad de trabajo: cómo se estructuran las tareas, cómo se realizan, cuál es su orden correlativo, cómo se sincroni-zan, cómo fluye la información que soporta las tareas y cómo se le hace seguimiento. Ter-minología propia de RUP
XP	<i>eXtreme Programming.</i> Acrónimo de un tipo de metodología ágil

2 Vista general del proyecto

21 Propósito, alcance y objetivos

El propósito del proyecto de desarrollo de *software* SCP, un proyecto que realice la recopilación de los datos de cada uno de los medios de transporte que llegue una báscula del Ministerio de Transporte e Infraestructura (Nicaragua), también posibilita la realización de otros proyectos de esta misma institución, en conjunto con la recopilación de los datos de peso de los medios de transporte, así mismo alguna activada que se realice después de la recopilación de los datos, como son Multas, Advertencias, Sanciones, etc.

El proyecto formara parte de las líneas de trabajo emprendidas en conjunto con otros desarrolladores, cuyo objetivo específico es la promoción, desarrollo y consolidación en Nicaragua de las sobre las vías de transporte terrestres, acuática y de construcción de todas las areas de nicaragua.

El proyecto SCP exige que se realicen las acciones oportunas para que un sistema lo más automatizado posible capture datos que estén al alcance del objeto a analizar: peso en bascula, datos del conductor, datos del medio de transporte y la actividad consecuente. Los datos deben poder, a posteriori, ser susceptibles de recibir un análisis cuantitativo y/o cualitativo que permita extraer conclusiones científicamente o económicamente válidas y, por tanto, generar nuevo conocimiento, a través de reportes.

Para cumplir adecuadamente con el cometido, las características requeridas por el sistema SCP deberán ser las siguientes:

- **Captura automatizada de información.**
 - Localización de fuentes de información primarias a partir de las cuales se obtendrán los datos.
 - Puesta a disposición de los datos capturados a los actores encargados de su consolidación y revisión.
 - Establecimiento del proceso de captura como tarea automatizada y repetitiva del peso obtenido en basculas.
 - Establecimiento de filtros previos que afinen el proceso de captura en función de la pertinencia de los contenidos localizados en las fuentes remotas de la Red.
- **Consolidación de la información localizada.**
 - Revisión del conjunto de capturas por parte de los actores encargados de ello, descartando la información capturada no relevante a los efectos del propósito de SCP.
 - Completado de la información considerada esencial en cada PCT localizada y que esté ausente en los datos primarios capturados de las ubicaciones remotas de la Red.
 - Almacenamiento de los datos localizados y revisados en un SGBD con garantías de seguridad, fiabilidad y posibilidades de recuperación.
 - Normalización de la información almacenada mediante revisiones que garanticen la recuperabilidad y homologabilidad de los datos y permitan su clasificación y ordenación.
- **Explotación de la información.**
 - Generación de un formato exportable para su uso por parte de los actores externos a SCP y pertenecientes a MTI, regido por normativa de intercambio lo más estándar

posible, adaptada a las peculiaridades de la estructura y arquitectura de datos de la aplicación.

- Creación de un mecanismo de exportación de los datos consolidados y normalizados a los actores externos intervinientes que formen parte de EX y estén habilitados para ello previa adecuada salvaguarda de identificación y validez de las solicitudes de acceso.
- Creación de un mecanismo de explotación directa de la información cuantificable de los datos coleccionados, disponible desde el apropiado interfaz de SCP y paralela a cualquier otra implementación que al respecto puedan llevar a cabo los oportunos actores de SCP.

22 Restricciones y suposiciones

Si bien no existe, como en el caso de los proyectos de carácter comercial o no académico, un conjunto de restricciones muy específicas, relacionadas con la infraestructura del MTI y su disponibilidad económica y temporal, la propia existencia de unos plazos de término para la entrega de la documentación, condiciona el desarrollo de SCP, así como el presupuesto derivado de la dotación económica que reciba S&S.

Este apartado 2.2 podrá ser modificado en sucesivas versiones de PDSDT de forma que se adapte a las condiciones evolutivas de SCP.

2.2.1 Restricciones

Derivan de las condiciones de adjudicación de la ayuda y de los recursos humanos, temporales y materiales disponibles por el Ministerio de transporte e infraestructura.

- La recopilación de datos para la codificación del sistema de parte del equipo técnico de S&S.
- Colaboración de parte del equipo de MTI para brindar la debida información que se solicite.
- Previamente a la finalización, un grupo de actores deberá realizar el proceso de análisis de la información custodiada por SCP.
- Por las condiciones MTI como la entidad que hará uso del sistema, dará acceso a un grupo humano con la adecuada formación para acometer la tarea de actores necesarios en SCP con perfiles muy específicos, pero no existe un excesivo margen de maniobra para retribuir la colaboración de dichos actores, de manera que SCP puede verse sometido a picos de productividad y rendimiento no planificados, derivados de la compatibilidad de tareas por parte de los mencionados actores.
- Al tratarse de una entidad gubernamental dependiente del presupuesto general de la república, SCP debe utilizar recursos HW y/o SW compatibles.
- Por necesidad de cumplir adecuadamente los plazos estipulados SCP debe estar a pleno rendimiento y completamente funcional para permitir la explotación de los datos coleccionados con antelación en un periodo de 3.5 meses una vez aprobado este PDSDT.
- Para que la validez de los estudios cuantitativos y/o cualitativos derivados del uso de los datos coleccionados transferidos desde SCP a los actores oportunos sea efectiva, la información recopilada deberá abarcar de la forma más exhaustiva posible un conjunto normas cuya significación permita elevar las conjeturas derivadas de la investigación operativa de los datos a la categoría de conclusiones científicamente contrastables.

2.2.2 Suposiciones

- SCP supone la existencia de fuentes primarias de información disponibles en la Red, de las que es posible extraer un conjunto relevante de datos que tratar. La no existencia de dichas fuentes ya es, de por sí, significativa.
- Dada la naturaleza y el enfoque que se adopta como solución proyectual para SCP (véase 5), todos los actores que intervengan en el proceso de consolidación y almacenamiento de datos deben disponer del HW y la conectividad adecuados para acceder a la Red.
- SCP no necesita una ubicación física o “domicilio” para desarrollarse, pudiendo intervenir los actores desde cualquier punto, lo que por una parte supone una evidente ventaja, pero por otra puede ser contraproducente, al no existir un “set” HW/SW estandarizado que se empleará como herramienta de trabajo por parte de los actores.
- Se supone que todos los actores implicados en los procesos de grabación, normalización, verificación y revisión aludidos en 4 disponen de la adecuada formación, no necesariamente completa en ambos aspectos pero sí obligatoriamente en el primero, tanto en DP y CBS como en el manejo de herramientas ofimáticas simples, no siendo posible para el plan de SCP contar con una fase de formación genérica de sus actores, si bien sí la hay, en formato cíclico, para la específica.

23 Aclaraciones previas

La norma IEEE Std 610 define software como “programas, procedimientos y documentación y datos asociados, relacionados con la operación de un sistema informático”. PDSDT utiliza tal definición como referencia y, por tanto, considera la existencia de tres componentes en el conjunto de desarrollo.

- **Programas.** Conjuntos de instrucciones que proporcionan la funcionalidad deseada cuando son ejecutadas por el ordenador. Están escritos usando lenguajes específicos que los ordenadores pueden leer y ejecutar, tales como lenguaje ensamblador, Basic, C, C#, Visual Basic ... Los programas también pueden ser generados usando generadores de programas.
- **Datos.** Elementos significativos portadores de la información que se pretende tratar mediante el empleo de programas.
- **Documentos.** Imprescindibles para que los actores dispongan de una explicación de cómo usar los programas (manuales de usuario y de operatoria) y de cómo intervenir sobre ellos para ajustar su funcionalidad.

Según la clasificación de SW recogida en el documento “Ingeniería de Software: metodologías y ciclos de vida” (INTECO 2009)², la aplicación de SCP puede considerarse como integrante de las siguientes categorías:

- **Software de aplicaciones:** se usan para proveer servicios a clientes y ejecutar negocios de forma más eficiente. El software de aplicaciones puede ser un sistema pequeño o uno grande integrado.
- **Software de gestión:** el proceso de la información comercial constituye la mayor de las áreas de aplicación del software. Los sistemas discretos (por ejemplo: nóminas, cuentas de haberes-débitos, inventarios, etc.) han evolucionado hacia el software de sistemas de información de gestión (SIG) que accede a una o más bases de datos que contienen información comercial. Las aplicaciones en esta área reestructuran los datos existentes pa-

ra facilitar las operaciones comerciales o gestionar la toma de decisiones.

- **Software basado en web:** las páginas web buscadas por un explorador son software que incorpora instrucciones ejecutables y datos.

La **ingeniería de software** dispone de una definición de IEEE SESC (1990)³ del siguiente tenor: “Software engineering is the application of a systematic, disciplined, quantifiable approach to the design, development, operation, and maintenance of software, and the study of these approaches; that is, the application of engineering to software”.

En la ingeniería de software, la **ingeniería de requisitos** comprende todas las tareas relacionadas con la determinación de las necesidades o de las condiciones a satisfacer para un SW nuevo o modificado, tomando en cuenta los diversos requerimientos de las partes interesadas, que pueden entrar en conflicto entre ellos. El propósito de la ingeniería de requisitos es hacer que los mismos alcancen un estado óptimo antes de alcanzar la fase de diseño en el proyecto. Los buenos requisitos deben ser medibles, comprobables, sin ambigüedades o contradicciones, etc. La norma 830-1988 de IEEE regula los procedimientos de la ingeniería de requisitos.

El estándar IEEE-1074 para los procesos de vida del software describe el conjunto de actividades y procesos obligatorios para el desarrollo y mantenimiento de SW. Tiene como objetivo establecer un marco común para el desarrollo de modelos para el proceso de construcción. EF estructura su proceso vital en fases.

2.3.1 Especificación

Escribir detalladamente el SW en desarrollo, en una forma matemáticamente rigurosa. Cubierta mediante PDSDT.

2.3.2 Diseño y arquitectura

Describir las pautas o patrones a los que se adecua el mecanismo de funcionamiento de la aplicación, incorporando consideraciones de la implementación tecnológica y obteniendo un modelo cercano a la programación orientada a objetos. Cubierta mediante PDSDT.

2.3.3 Programación

Reducir un diseño a código ejecutable en un ordenador.

2.3.4 Prueba

Verificar que SW realiza correctamente las tareas indicadas en la especificación del proyecto

² INTECO. *Ingeniería de Software: metodologías y ciclos de vida*. Laboratorio Nacional de Calidad del Software. Instituto Nacional de Tecnologías de la Comunicación. 2009.

³ IEEE. *IEEE Standard Glossary of Software Engineering Terminology*, IEEE std 610.12-1990, 1990.

2.3.1 Mantenimiento

Corregir errores y mejorar SW para solventar fallos descubiertos y tratar con nuevos requisitos. Puede ser de cuatro tipos: *perfectivo* (mejorar la calidad interna de los sistemas), *evolutivo* (incorporaciones, modificaciones y eliminaciones necesarias en un producto software para cubrir la expansión o cambio en las necesidades del usuario), *adaptativo* (modificaciones que afectan a los entornos en los que el sistema opera, por ejemplo, cambios de configuración del HW, SW de base, gestores de base de datos, comunicaciones) y *correctivo* (corrección de errores).

Estas fases se corresponden, de acuerdo a lo normalizado en *Ilustración 4. Disciplinas y fases RUP en SCP* (ver 2.7), respectivamente, con lo expresado en la siguiente matriz.

FASES IEEE-1074	FASES RUP
Especificación	Inicial
Diseño y arquitectura	Elaboración
Programación	Construcción
Prueba	Transición
Mantenimiento	Transición

24 Ciclo de vida

A los efectos PDSDT, se entenderá por **ciclo de vida** o **paradigma** el conjunto de fases por las que pasa el sistema que se está desarrollando desde que nace la idea inicial hasta que el software es retirado o remplazado (muere). Las fases mencionadas se desarrollan de forma sucesiva, y pueden estar compuestas por tareas panificables. Cada fase se construye agrupando tareas (actividades elementales), que pueden compartir un tramo determinado del tiempo de vida de un proyecto. La agrupación temporal de tareas impone requisitos temporales correspondientes a la asignación de recursos (humanos, financieros o materiales).

Según el modelo de ciclo de vida, la sucesión de fases puede ampliarse con bucles de *feedback*, de manera que lo que conceptualmente se considera una misma fase se pueda ejecutar más de una vez a lo largo de un proyecto, recibiendo en cada pasada de ejecución aportaciones a los resultados intermedios que se van produciendo (retroalimentación).

Un concepto adicional, que se usa en PDSDT, es el de **entregables**. Por tales deben entenderse los productos intermedios que generan las fases. Pueden ser materiales o inmateriales (documentos, SW). Los entregables permiten evaluar la marcha del proyecto mediante comprobaciones de su adecuación o no a los requisitos funcionales y de condiciones de realización previamente establecidos.

El modelo de ciclo de vida adoptado en el desarrollo de SCP es el de **prototipos**, propio de la metodología ágil utilizada para todo el proceso completo (ver 2.6). Según los principios de este modelo, el RMTI a menudo define un conjunto de objetivos generales para SW, pero no identifica los requisitos detallados de entrada, proceso o salida. En otros casos, el responsable del desarrollo del software puede no estar seguro de la eficiencia de un algoritmo, de la calidad de adaptación de un sistema operativo, o de la forma en que debería tomarse la interacción hombre-máquina. En estas y en otras muchas situaciones, un paradigma de construcción de prototipos puede ofrecer el mejor enfoque.

El ciclo de construcción de prototipos comienza con la recolección de requisitos. El desarrollador y RMTI encuentran y definen los objetivos globales para el software, identifican los requisitos conocidos y las áreas del esquema en donde es obligatoria más definición. Entonces aparece un diseño rápido. El diseño rápido se centra en una representación de esos aspectos del software

que serán visibles para RMTI. El diseño rápido lleva a la construcción de un prototipo. El prototipo lo evalúa RMTI y se utiliza para refinar los requisitos del SW a desarrollar. La iteración ocurre cuando el prototipo se pone a punto para satisfacer las necesidades de RMTI, permitiendo al mismo tiempo que el desarrollador comprenda mejor lo que se necesita hacer.

Entre las ventajas, citadas por INTECO 2009, figuran:

- Ofrece visibilidad del producto desde el inicio del ciclo de vida con el primer prototipo. Esto puede ayudar al RMTI a definir mejor los requisitos y a ver las necesidades reales del producto. Permite introducir cambios en las iteraciones siguientes del ciclo. Permite la realimentación continua del RMTI.
- El prototipo es un documento vivo de buen funcionamiento del producto final. El RMTI reacciona mucho mejor ante el prototipo, sobre el que puede experimentar, que no sobre una especificación escrita.
- Este modelo reduce el riesgo de construir productos que no satisfagan las necesidades de los usuarios.

El documento citado también describe algunos inconvenientes de este tipo de ciclo, pero, dadas las características específicas de SCP, el equipo de desarrollo ha concluido que eran poco aplicables en este caso concreto y, por tanto, su impacto en el resultado final es despreciable. Tales inconvenientes pueden describirse como sigue:

- Es posible que el conjunto procesual de desarrollo sea lento. Además, puede llevar a que se realicen fuertes inversiones en un producto desechable ya que los prototipos se descartan. Ambas contrariedades carecen de relevancia en el caso SCP, puesto que tanto la inversión está predeterminada desde el primer momento, y la relación con RMTI es con intermediarios, es decir un tanto indirecta.
- Con este modelo pueden surgir problemas con RMTI, que ve funcionando versiones del prototipo, pero puede desilusionarse porque el producto final aún no ha sido construido. El desarrollador puede caer en la tentación de ampliar el prototipo para construir el sistema final sin tener en cuenta los compromisos de calidad y de mantenimiento que tiene con el RMTI. No es el caso de SCP, por tratarse de un equipo meramente de desarrollo por así decirlo, un prototipo representaría un eslabón más del equipo de diseño.

25 Normativa y protocolos aplicables al ciclo de vida

La norma ISO/IEC 12207 define un modelo de ciclo de vida como un marco de referencia que contiene los procesos, actividades y tareas involucradas en el desarrollo, operación y mantenimiento de un producto software, y que abarca toda la vida del sistema, desde la definición de sus requisitos hasta el final del uso.

Esta norma agrupa las actividades que pueden llevarse a cabo durante el ciclo de vida del SW en procesos principales, de apoyo y organizativos. Cada proceso del ciclo de vida está dividido en un conjunto de actividades y cada una de ellas en un conjunto de tareas. Con afán de simplificación y adecuación a las necesidades específicas del proyecto SCP, el equipo de desarrollo resumió la estructura de la mencionada norma, dejando solamente los procesos y actividades de aplicación en el caso. Por ejemplo, la actividad de adquisición tiene el sentido ya que SCP se trata de un trabajo personalizable, una “venta” de software.

Téngase en cuenta que esta agrupación de procesos no responde a un mecanismo secuencial, donde uno precede a otro, sino que pueden ejecutarse de forma simultánea. Por ejemplo, las

actividades de solución de problemas y mejora van unidas al tiempo de operación.

A continuación, se describe la jerarquía de procesos y tareas tal y como se implanta en SCP.

- **Procesos principales:** dan servicio a las partes más importantes del ciclo. Los principales son:
 - **Suministro.** Asimilada a la compra del *HW* necesario para llevar a término el proyecto.
 - **Desarrollo.** Creación del *SW* para la funcionalidad de la aplicación final.
 - **Operación.** Incluye el proceso de captura automatizado, la consolidación y grabación de datos, su normalización y verificación, además de las tareas propias de administración del aplicativo (parámetros, opciones por defecto,...).
 - **Mantenimiento.** Tareas propias de la administración del sistema. Garantizan el adecuado funcionamiento del OS empleado en el servidor o alojamiento web, del motor SGBD y de otros elementos comunes de infraestructura, imprescindibles para el funcionamiento correcto del proyecto.
- **Procesos de apoyo:** contribuyen a la puesta en funcionamiento de otros procesos como parte esencial de los mismos, con un propósito bien definido y al éxito y calidad del proyecto. Son empleados y ejecutados por otros procesos según sus necesidades. Los procesos de apoyo son:
 - **Documentación.** Tanto la propia del diseño del proyecto de SW y de su proyecto matriz como los manuales de usuario. Realizada a medias entre el equipo de desarrollo y los actores implicados en verificación y validación.
 - **Gestión de la configuración.** Asimilada a las tareas de administración del sistema.
 - **Verificación.** Comprobación de la calidad de los datos capturados automáticamente y completado de los mismos con información obtenida por cualquier medio válido (preferentemente, disponible de forma irrestricta en la Red).
 - **Validación. o Normalización.** Adecuación de la información contenida en los datos ya validados a la reglamentación que sobre los mismos exista (clasificaciones,).
 - **Auditoría.** Realizada por actores externos a SCP, preferiblemente por aquellos que dieron seguimiento a todo el proceso de desarrollo.
 - **Solución de problemas.** Ajuste fino del funcionamiento de SW aplicativo, de oficio o a petición de cualquier actor implicado en su uso.
- **Procesos organizativos:** se emplean para establecer e implementar una infraestructura construida por procesos y personal asociado al ciclo de vida, y para mejorar de forma cíclica el conjunto. Los más destacados son:
 - **Gestión.** Coordinación de los equipos formados por los grupos de actores tal y como se describen en 3. Comunicaciones de alcance global para el conjunto de recursos humanos implicados. Relaciones con el ministerio de transporte e infraestructura.
 - **Infraestructura.** Gestión del suministro de HW/OS/SW imprescindible para la implementación y la implantación. Monitorización periódica y sistemática del

estado del conjunto servidor (HW/OS/SW).

- **Mejora.** Llevada a cabo a partir de sugerencias en reuniones periódicas de tipo *sprint* según la metodología SCRUM/XP. Implementada por el equipo de desarrollo y supervisada por el *Product Owner* de SCP.
- **Formación.** Específica para los actores implicados en verificación y validación. No incluye, por tratarse de un conocimiento previo necesario, preparación en manejo de herramientas ofimáticas.

26 Metodología

Definido por el DRAE como “conjunto de métodos que se siguen en una investigación científica o en una exposición doctrinal”, en el contexto del DBA utilizamos el término *metodología* para hacer referencia al grupo normativizado de procedimientos para abordar las tareas que componen las fases del ciclo de vida. Cada una de esas tareas puede ser abordada y resuelta de múltiples maneras, con distintas herramientas y utilizando distintas técnicas, y justamente al conjunto de todos esos instrumentos para acometer su realización, conjunto que se rige por unos determinados protocolos, es a lo que el presente documento alude cuando se habla de metodología.

Se suele establecer en la actualidad una distinción entre las *metodologías tradicionales* y las *ágiles*. Cada una de ellas tiene aplicación dependiendo de las características del proyecto abordado. En el caso de SCP, el equipo de desarrollo optó por las denominadas “ágiles”.

Entre las tradicionales, desechadas para usarlas de base del trabajo en SCP desde el primer momento, figuran algunas como el *Desarrollo de Sistemas de Jackson* (JSD), muy propia de los años 80 del pasado siglo, o, de las mismas fechas, la *Ingeniería de la Información* o la británica *Structured System Analysis and Design Method* (SSADM). El Ministerio de Administraciones Públicas promueve *METRICA* (actualmente en su versión 3) como estándar para el trabajo de organismos oficiales.

Por contraste con los sistemas anteriores, caracterizados por la rigidez de sus procedimientos y la sobreabundancia de entregables documentales, el **desarrollo ágil** de SW utiliza un proceso iterativo como base para abogar por un punto de vista más ligero y más centrado en las personas que en el caso de las soluciones tradicionales. Los usos ágiles emplean retroalimentación en lugar de planificación, como principal mecanismo de control. La retroalimentación se canaliza por medio de pruebas periódicas y frecuentes versiones del software. De hecho, se trata de metodologías aplicables a numerosos tipos de actividades humanas, y no solamente al desarrollo de SW.

Hay muchas variantes de los procesos ágiles, pero en EF el equipo de desarrollo ha procurado ceñirse a dos modelos esenciales: **eXtreme Programming (XP)** y **Lean Programming**. En ambos casos, es posible (aunque el desarrollo de la pertinente argumentación queda fuera del alcance del presente documento) hacer compatibles tales modelos con el estándar ISO/IEC 15504 o SPICE, considerado uno de los más fiables para el establecimiento de mejoras en el complejo conjunto de especificaciones de la ingeniería de software.

En el caso de la *programación extrema (XP)*, las fases se realizan en pasos muy cortos (o “continuos”) con respecto al anterior. Se crean pruebas automatizadas para proveer metas concretas al desarrollo. Después se programa el código, que será completo cuando todas las pruebas se superan sin errores. El diseño y la arquitectura emergen a partir de la refactorización del código, y lo realizan los propios desarrolladores. El sistema, incompleto, pero funcional, se despliega para su

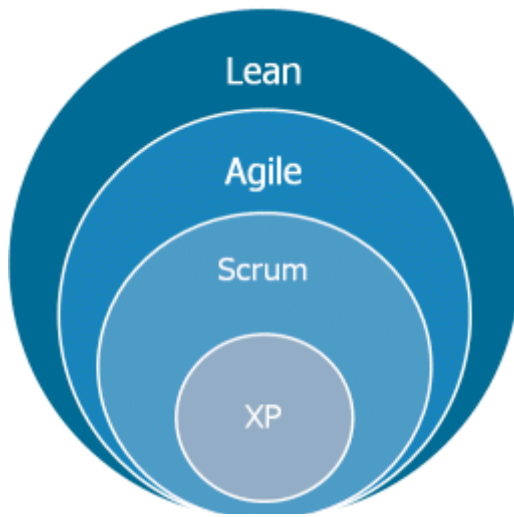


Ilustración 1. Jerarquía metodológica en EF

demostración al RMTI a través de los usuarios directos (al menos uno de los cuales pertenece al equipo de desarrollo).

Esta metodología es muy conveniente para SCP por cuanto, como se verá en 3, existe un grupo de usuarios finales, que pueden considerarse parte del RMTI, que trabajan como revisores y grabadores de datos capturados y actúan como parte interesada en el ciclo de desarrollo.

Lean Software Development, también conocido como *Lean Programming* es un conjunto de técnicas que engloban una metodología de desarrollo ágil de software orientado a conseguir exactamente lo que necesita el RMTI. Es una evolución del Método Toyota de Producción aplicado al desarrollo y que esta muy de moda en-

tre los equipos de desarrollo en *startups*. Principalmente consiste en ciclos de evolución de SW incrementales en los que se posponen las decisiones lo más posible hasta haber obtenido un feedback del cliente y así reaccionar lo más rápido y eficazmente a sus necesidades. Se fundamenta en tener un equipo potente y comprometido y el principio de aprendizaje continuo sobre el producto final.

El desarrollo basado en LP se debe su nombre a la palabra inglesa que puede traducirse por “esbelto”, pero también por “apoyado”. Ambas cosas coinciden en una metodología cimentada a su vez sobre las buenas prácticas industriales (de Toyota, en concreto)⁴. LP basa su estructura en siete importantes principios, muy relacionados con los ciclos diarios de desarrollo que se popularizaron tras la publicación del libro *Microsoft Secrets*⁵, que hacía público el sistema de desarrollo incremental con ciclos diarios utilizado por la célebre empresa norteamericana. Estos principios, a los que SCP se adhiere, son:

- Eliminar todo aquello que no sea útil para el desarrollo: burocracia innecesaria, retrasos en entregas, código redundante,...
- Aprendizaje continuo y ampliado. De los errores también se aprende.
- No adoptar prematuramente decisiones que comprometan el desarrollo.
- Reaccionar a la máxima velocidad posible ante modificaciones, ampliaciones o cambios de rumbo en el proyecto y –cómo no- ante los *bugs*.
- Potenciar el trabajo en equipo y la cohesión entre los diferentes agentes y roles que interactúan entre sí en SCP.
- Elevar la integridad conceptual a la categoría de rumbo principal del proyecto. Este término se emplea en SCP para aludir a dos cuestiones:
 - Cada artefacto debe funcionar a la perfección en coordinación con los demás, es decir, que la ejecución de un código no puede realizarse correctamente a costa de otro.
 - Cada actor que intervenga como verificador o supervisor debe asumir (por cuanto

se verá en lo especificado en 8) que la calidad de los datos que custodia depende exclusivamente de su escurpulooso cumplimiento de las normas de revisión y grabación

- Ver siempre SCP como un conjunto único, un único proyecto, de forma que nunca se pierda la perspectiva global, que ha de garantizar tanto la integridad conceptual como el aprendizaje continuo y las respuestas rápidas.

Para el trabajo efectivo, SCP adopta una técnica “suavizada” de desarrollo **SCRUM**, un modelo de referencia que define un conjunto de prácticas y roles, y que puede tomarse como punto de partida para definir el proceso de desarrollo que se ejecutará durante un proyecto. Presentado por Ken Schwaber a mediados de la década de los noventa del pasado siglo en una conferencia internacional sobre POO, recoge el estudio de dos investigadores japoneses que habían detectado las pautas de funcionamiento básicas en grandes empresas industriales como Xerox, 3M, Canon y otras. El nombre alude a un tipo de puesto de jugador en las tácticas del deporte del rugby.

SCRUM se basa en la adopción de **roles** por parte de quienes intervienen en el proceso de desarrollo (tanto de la parte RMTI como del equipo que construye el proyecto). Cada actor interviene según las pautas del rol que se le ha asignado. Los procesos se realizan empleando **artefactos** y los hitos de desarrollo se denominan **eventos**. La gestión del proyecto se realiza mediante reuniones cíclicas cuyo contenido es referenciado documentalmente de carácter periódico en cuanto a temporalidad y cíclico en cuanto a contenido.

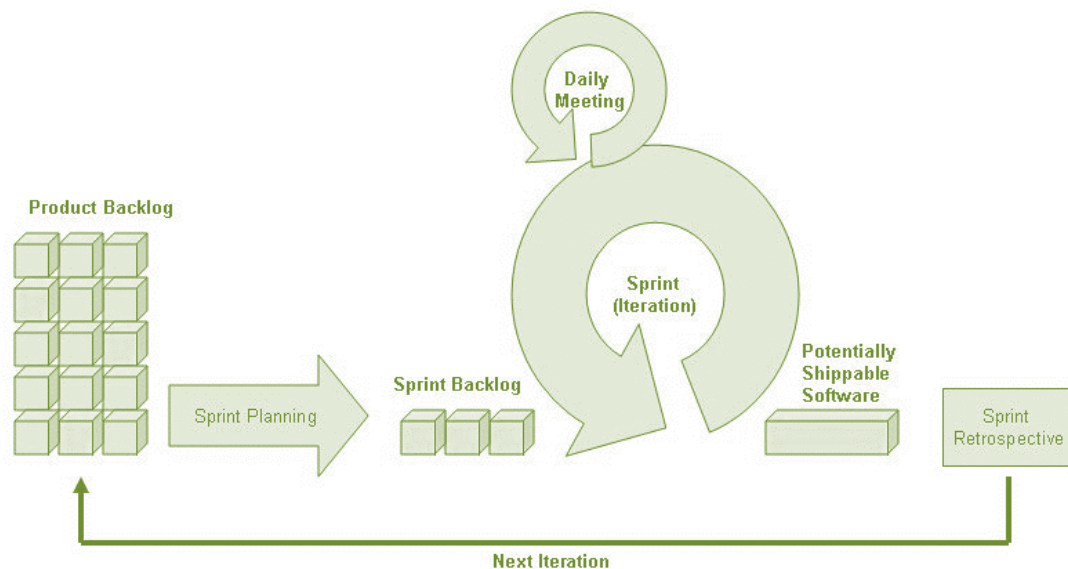


Ilustración 2. Ciclo SCRUM

⁴ A pesar de lo que comúnmente se cree, los que comenzaron a popularizar el concepto *lean* no fueron los japoneses, sino los laboratorios del MIT, en especial tras la publicación de su célebre libro “La máquina que cambió el mundo” (1990). Sin embargo, fue el matrimonio formado por Mary y Tom Poppendieck los primeros en atribuir el concepto al SW, a través de su obra *Lean Development Software: An Agile Toolkit for Software Development Managers (The Agile Software Development Series)* (2003, Addison-Wesley, EEUU).

⁵ M. Cusmano y M. Selby. *Microsoft Secrets*. EEUU: HarperCollins Business, 1997.

SCRUM implica, por parte del equipo de desarrollo, la asunción de unos principios, a los que SCP, por las circunstancias concretas de su generación (es un proyecto de recopilación de datos cuya finalidad es lanzar alertas de cuando se incumple los reglamentos de vías terrestres del MTI), se adhiere de forma natural. Dichos principios esenciales son:

- Adoptar una estrategia de desarrollo incremental, en lugar de la planificación y ejecución completa del producto. Muy conveniente, por cuanto el objetivo final concreto de SCP es “territorio inexplorado todavía, pero se tiene una breve esencia”, y es importante no dar por sentados los planteamientos iniciales. SCP debe “tener cintura” para lidiar con frecuentes cambios de orientación que pueden llegar a afectar al paradigma.
- Basar la calidad del resultado más en el conocimiento tácito de las personas en equipos auto organizados, que en la calidad de los procesos empleados. De ahí la importancia de la relación académica inherente a los recursos humanos que constituyen el personal de trabajo.
- Solapamiento de las diferentes fases del desarrollo, en lugar de realizar una tras otra en un ciclo secuencial o de cascada. La limitación de tiempo, la premura de un *back-end* inexorable, tanto más cuanto, como se ha mencionado, ningún miembro del equipo puede disponer de dedicación exclusiva, aconsejan acelerar los procesos y simultanearlos.

Un principio adicional, no obstante que clave, en el planteamiento SCRUM es la asunción como pauta de desarrollo de que durante un proyecto RMTI puede cambiar de idea sobre lo que necesita o desea obtener, alterando así los requerimientos iniciales e influyendo en la disciplina de la ingeniería de requisitos, y que los desafíos impredecibles no pueden ser fácilmente enfrentados de una forma predictiva y planificada. Adhiriéndose plenamente a ello, SCP adopta una aproximación pragmática, aceptando que el problema no puede ser completamente entendido o definido, y centrándose en maximizar la capacidad del equipo de entregar rápidamente y responder a requisitos emergentes. Los “saltos” evolutivos se llevan a término en ciclos, denominados **Sprints**.

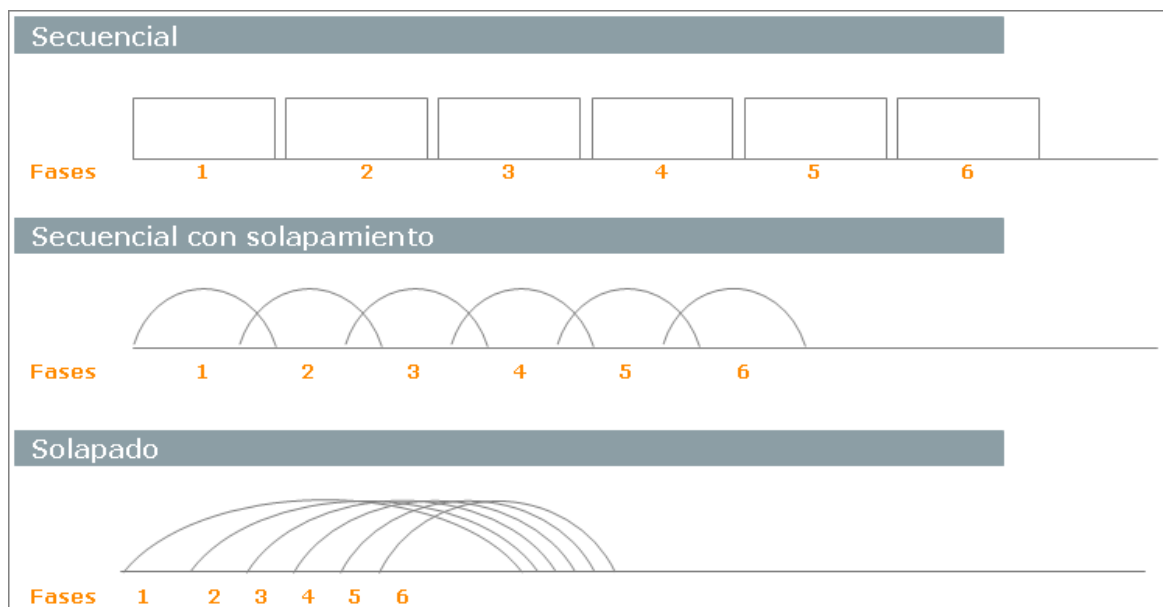


Ilustración 3. Desarrollo secuencial y solapado

SCP no implementa el modelo SCRUM completo (es posible obtener documentación adicional al respecto en Palacio 2015⁶) por las características de los recursos humanos implicados (carecen en su totalidad de dedicación plena al proyecto), que imposibilitan un plan de reuniones tan estricto como el modelado original requiere. La plasmación modélica de SCP utiliza los siguientes roles:

ROL	ACTOR	CARACTERÍSTICAS
Scrum Master	Desarrollador jefe	Facilita la aplicación de SCRUM y gestionar cambios. Coordina con Team.
Product Owner	Supervisor jefe	Representa a los <i>stakeholders</i> (interesados externos o internos).
Team	Desarrolladores	Realizan los artefactos determinados en las Reuniones.

Las reuniones en SCRUM siguen una pauta poco flexible, y por ello SCP ha simplificado su jerarquía reduciéndolas a una por ciclo (***Sprint Planning Meeting***), que, por razones de premura y organizativas, suelen realizarse una vez cada quince días, en vez de con periodicidad mensual, como suele ser la tónica común.

El ***Product Backlog***, que es un conjunto de requisitos de alto nivel priorizados que definen el trabajo a realizar, viene en SCP predeterminado desde el inicio por la matriz aceptada de requerimientos, y, como determina el método, se altera en función de los resultados del ***Sprint Backlog*** para adaptar el conjunto de características a la evolución de las necesidades de RMTI, a las especificidades del trabajo de los diferentes actores de SCP que también son cambiantes según se lleve a cabo el desempeño de los roles concretos y a los límites temporales y presupuestarios.

27 Entregables del proyecto

A continuación, se indican y describen cada uno de los *artefactos* que han sido, son o serán generados y utilizados por el proyecto y que constituyen los ***entregables***. Esta lista describe la configuración RUP desde la perspectiva de artefactos, adaptada al método SCRUM modificado propio de SCP.

Es preciso destacar, no obstante, que, de acuerdo a la filosofía RUP (y de cualquier procedimiento iterativo e incremental), todos los artefactos son objeto de modificaciones a lo largo del tiempo de desarrollo, con lo cual sólo al término de un proceso es posible disponer de una versión definitiva y completa de ellos. Sin embargo, el resultado de cada iteración, el ***Sprint Backlog*** y los hitos del proyecto están enfocados a conseguir un cierto grado de completitud y estabilidad de los artefactos.

⁶ J. Palacio. *Gestión de proyectos Scrum Manager (Scrum Manager I y II). Versión 2.5.1*. España, 2015.

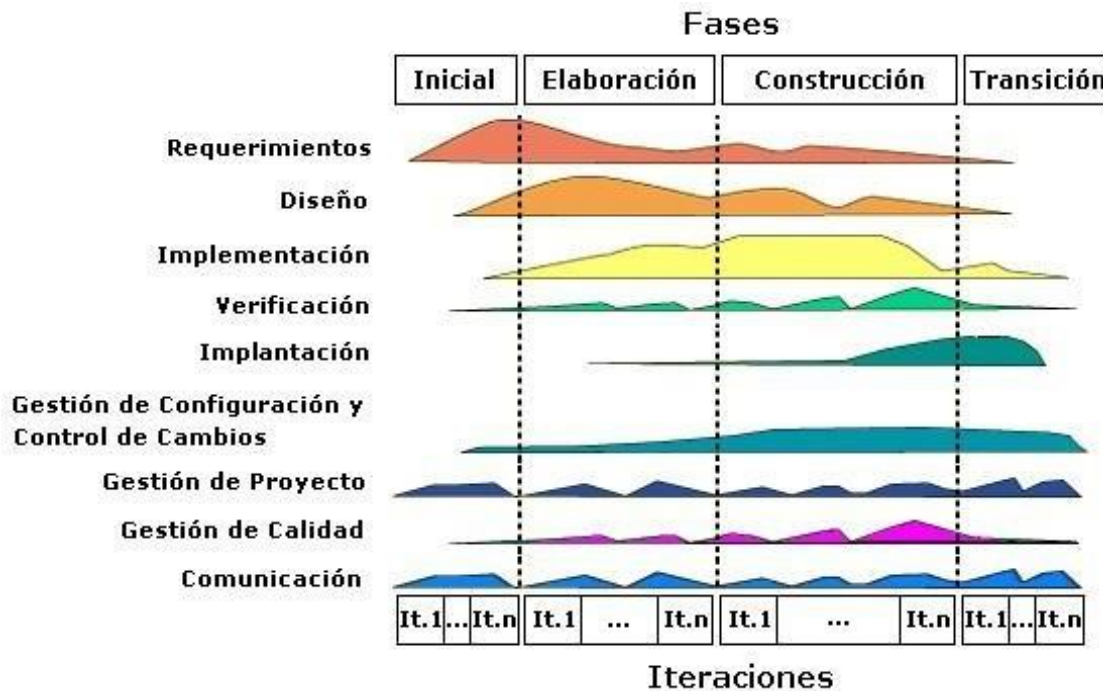


Ilustración 4. Disciplinas y fases RUP en SCP

La estructura de procesos y secuencias que el principio RUP impone agrupa las tareas que se realizan en **disciplinas** (en el diagrama, los elementos que figuran a la izquierda), y la flecha temporal se basa en la realización de **iteraciones** de las tareas reunidas en **fases**. En SCP, y puesto que, como ha quedado establecido ya desde 1, el *constructo* lógico del proyecto se apoya en una planificación *Rational Unified*, a la vez que sigue principios SCRUM, LP y XP, los entregables se ciñen a lo estipulado en el modelo, de manera que tanto RMTI como el propio *Team* y los diferentes *Managers* puedan realizar un seguimiento detallado que permita aplicar los principios LP sobre el aprendizaje, la rapidez de reacción y la integridad. El diagrama representa como una línea gráfica con picos y valles la intensidad de las actividades prevista desde el principio para el despliegue de SCP. Los entregables se agrupan tal como figura en la siguiente tabla.

DISCIPLINA	ENTREGABLE	PREVISIÓN	OBSERVACIONES
Gestión del proyecto	PDS	Noviembre 2020	Incluido en PDSDT. En 3
Diseño (Modelado de negocio)	Casos de uso agrupados	Noviembre 2020	Incluido en PDSDT. En 4
Diseño (Modelado de negocio)	Modelo de datos	Noviembre 2020	Parcialmente incluido en PDSDT (modelos físico y lógico). En 6 y 7
Requerimientos (Requisitos)	Restricciones y suposiciones	Noviembre 2020	Incluido en PDSDT. En 2
Requerimientos (Requisitos)	Matriz de requisitos	Noviembre 2020	Incluido en PDSDT
Implementación	Modelo de implementación	Noviembre 2020	Incluido en PDSDT. En 8
Verificación	Modelo de control de errores	Noviembre 2020	Incluido en PDSDT. En 10

Implantación	Sistema	Pendiente	Instalado en servidor
Implantación	Modelos de interfaces	Noviembre 2020	Incluido en PDSDT. En 10
Implantación	Manuales de usuario	Pendiente	No incluidos en PDSDT
Gestión de calidad	Matriz de PMBOK	Noviembre 2020	Incluido en PDSDT. En 13
Comunicación	Publicación de artículos y organización de <i>workshop</i>	Pendiente	No incluidos en PDSDT

Nótese que no todas las disciplinas implican la existencia de entregables y que, por la propia naturaleza del proyecto SCP, algunos dependen de la participación de RMTI.

3 Organización del proyecto

3.1 Recursos humanos

La estructura organizativa de SCP depende de dos principales entidades S&S y MTI cabe consultar la pertinente disposición del personal responsable del seguimiento así mismo los detalles sobre actores “externos” que intervengan en el procesado analítico de los datos consolidados en SCP. El perfil del personal asignado se agrupa en **equipos** de la siguiente manera.

Supervisión

Cargo: Supervisor principal. Tareas: Coordinación y dirección. Especialidades: Administración del sistema actual, Transito por vías terrestres, Análisis relacional.

Cargo: Supervisor secundario. Tareas: Coordinación y dirección. Especialidades: Administración del sistema actual, Transito por vías terrestres, Análisis relacional.

Desarrollo y administración del sistema

Víctor Paúl Téllez Pérez. VPTP. Cargo: Desarrollador. Tareas: Desarrollo de aplicaciones. Especialidades: Conocimientos Informáticos, Humanidades Digitales, Informática, Desarrollo.

Cargo. Desarrollador. Tareas: Desarrollo de aplicaciones. Especialidades: Conocimientos Informáticos, Humanidades Digitales, Informática, Desarrollo.

Revisión

Juan Gabriel Bermúdez Hernández. JGBH. Cargo: Investigador Informático. Ing. De sistemas Tareas: Coordinación de equipo. Datos. Especialidades: director informático, Analista de datos, Coordinación.

Grabación y verificación

Cargo: BDA colaborador. Tareas: Datos. Especialidades: Gestión de datos, Analista de datos, modelador y diseñador de bases de datos.

Dadas las específicas condiciones de SCP, su ausencia de objetivo comercial o lucrativo y su vocación científica, el **perfil** profesional y/o académico de los diferentes miembros de los grupos se adecua a las necesidades propias de la ejecución de las tareas que les son asignadas.

EQUIPO	PERFIL	ESPECIALIDADES
Supervisión		
Desarrollo	Ingeniero de sistemas	
Administración del sistema		SW y/o Telecomunicaciones
Revisión		.
Grabación	Ingeniero de sistemas o Ingeniero de computación (especialidad en gestión de datos).	
Verificación	Gestión de datos, Analista de datos, modelador y diseñador de bases de datos.	.

Si bien cae fuera del alcance de PDSDT abundar sobre la existencia de un mecanismo de relación laboral como vínculo entre los distintos miembros de los diferentes grupos y el proyecto, sí que cabe señalar que, los *stakeholder* carecen de remuneración dineraria por su participación en SCP y pueden solamente establecer una dedicación parcial, mientras que los restantes perfiles sí acceden a retribuciones mediante contratos especiales con S&S. Ello redunda, evidentemente, en beneficio de la intensidad de la dedicación de dichos perfiles, no obstante lo cual las limitaciones presupuestarias derivadas de RMTI (en este caso, el propio MTI) no permiten ampliar el horario de trabajo, pero se deben de realizar las entregas a tiempo y forma establecido en el plan de trabajo. A los efectos organizativos dentro de SCP, la **matriz de atribución de roles, actores, equipos y dedicación** puede consultarse en la siguiente tabla.

EQUIPO	ACTOR	ROL	DEDICACIÓN	REMUNERACIÓN	ASIGNACIÓN
Supervisión	Supervisor	Product Owner	Parcial	No	
Desarrollo	Desarrollador	Team	Total	Si	VPTP
Administración del sistema	Administrador	Team Manager	Parcial	No	
Revisión	Revisor	Team	Total	Si	JGBH
Grabación	Grabador verificador	Team	20	Si	
Verificación	Grabador verificador	Team	20	Si	

32 Jerarquía RRHH

La fluidez del funcionamiento de los diferentes equipos depende enteramente de una bien planificada estructura jerárquica que permita a unos actores realizar tareas que perfeccionen las llevadas a cabo por otros. De esa forma se refuerza la cohesión entre los miembros, se mantiene la integridad conceptual y se facilita la eliminación de tareas inútiles como comunicaciones en bucle o mediante escalafones complejos. El entramado de rangos y comunicaciones se recogen en la siguiente tabla y el esquema adjunto. Se incluye el actor “no humano” servidor por hallarse implicado un actor perteneciente a los equipos RRHH.

ACTOR	→ ACCIÓN →	ACTOR
Administrador	Informa de cambios de estado en el servicio	Supervisor
Administrador	Informa de cambios de estado en el servicio	Grabador verificador
Administrador	Informa de cambios de estado en el servicio	Revisor
Administrador	Informa de cambios de estado en el servicio	Desarrollador
Administrador	Monitoriza estado del servicio	Servidor
Administrador	Realiza optimizaciones específicas	Servidor
Administrador	Realiza optimizaciones periódicas	Servidor
Desarrollador	Informa sobre novedades en funcionamiento de SCP	Grabador verificador
Desarrollador	Informa sobre novedades en funcionamiento de SCP	Revisor
Desarrollador	Informa sobre novedades en funcionamiento de SCP	Supervisor
Desarrollador	Informa sobre novedades en funcionamiento de SCP	Administrador
Desarrollador	Solicita ajustes u optimizaciones del servicio	Administrador
Desarrollador	Valida calidad de manuales de usuario	Revisor
Grabador verificador	Informa de errores en programación	Desarrollador
Grabador verificador	Solicita modificaciones en valores de referencia	Revisor
Grabador verificador	Verifica y graba	Servidor
Revisor	Corrige errores de verificación y/o grabación	Grabador verificador
Revisor	Elabora y entrega manuales de usuario	Grabador verificador
Revisor	Elabora y entrega manuales de usuario	Supervisor
Revisor	Informa de errores en programación	Desarrollador
Revisor	Informa de errores en verificación y/o grabación	Grabador verificador
Revisor	Informa de errores en verificación y/o grabación	Supervisor
Revisor	Pide autorización para modificar valores de referencia	Supervisor
Supervisor	Autoriza modificar valores de referencia	Revisor
Supervisor	Informa de errores en programación	Desarrollador
Supervisor	Pide modificaciones en funcionalidad de SCP	Desarrollador
Supervisor	Solicita optimizaciones en prestaciones	Administrador
Supervisor	Valida calidad final de datos	Revisor

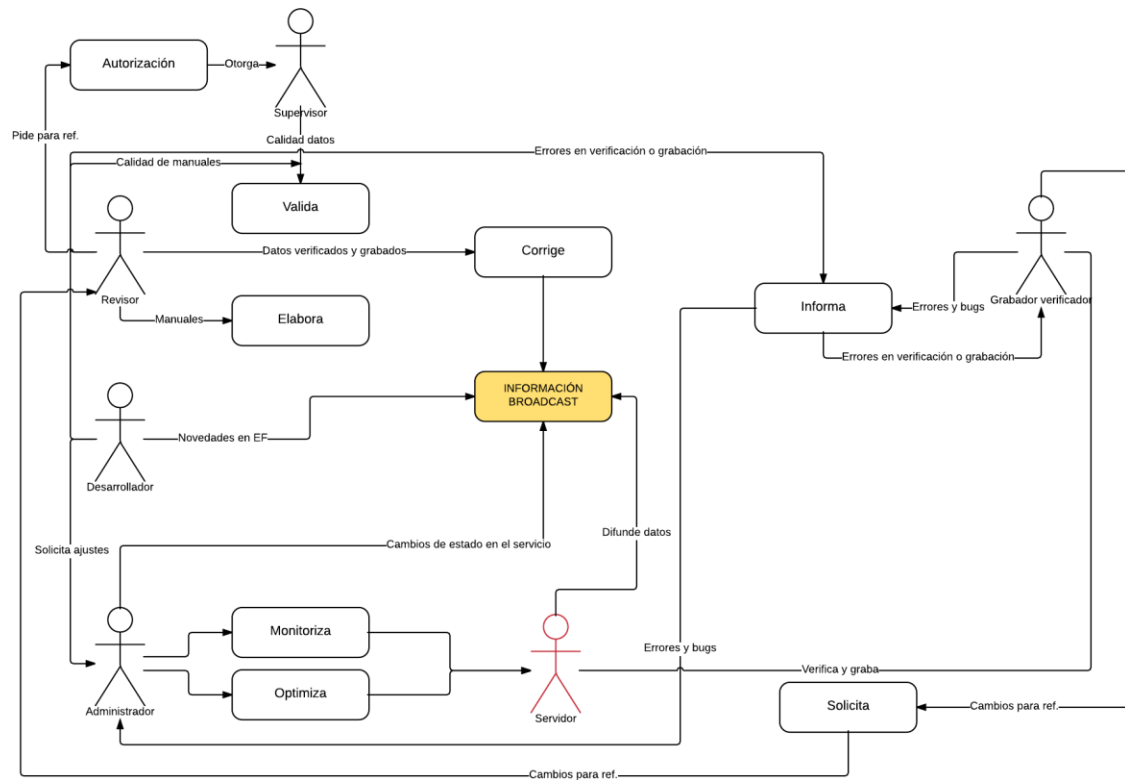


Ilustración 5. Interacciones entre actores en SCP

El elemento en color amarillo del diagrama ("Información *broadcast*") representa las interacciones en las que un único actor realiza una acción o entrega un artefacto cuyos destinatarios son la totalidad de los integrantes de los equipos de trabajo.

4 Gestión de procesos en el proyecto

41 Pendiente

5 Infraestructura

La modalidad de trabajo cliente/servidor, adoptada como solución funcional para SCP tal como se describe en 8, implica el uso del siguiente HW por parte de los diferentes actores del sistema.

Lado cliente

Equipo informático con cualquier sistema operativo capaz de soportar conexiones a Internet (OSX, Windows, Linux,...). Mínimo 4 Gb RAM. EL espacio de almacenamiento local es irrelevante.

Navegador basado en Webkit o Blink (Google Chrome 40 ó superior, Safari, Firefox... o bien Microsoft Edge).

Conexión de banda ancha a la red (xDSL, FTTH, FTTC...).

Velocidad de red mínima 5 MB.

Lado servidor

Equipo de marca indiferente, con resiliencia eléctrica y almacenamiento.

RAID. Sistema operativo Indiferente, con acceso *root*.

SGBD SQLServer

Conexión de alta velocidad a Internet.

Dominio Internet de tipo “.com”

Alternativa

Equipo informático con cualquier sistema operativo capaz de soportar conexiones a Internet (OSX, Windows, Linux,...). Mínimo 4 Gb RAM. EL espacio de almacenamiento local es irrelevante.

6 GB de ram

512 De almacenamiento.

Conexión de banda ancha a la red

(xDSL, FTTH, FTTC...).

Velocidad de red mínima 10 MB.

De acuerdo con los requerimientos solicitados por S&S, que se expresan en 8, y para el correcto funcionamiento del sistema SCP, *Product Owner/Supervisor* esta en derecho de declinar esta propuesta y está obligado a realizar una contra propuesta que se ajuste a la posibilidad de adquisición y la modalidad que mejor convenga según su presupuesto.

6 Estructura física de la información

Según se expresa en el análisis de requisitos y en la arquitectura consecuentemente adoptada (ver 8), existirá una capa física de estructura de los datos, que se corresponde con la configuración de tablas en el SGBD empleado, y otra lógica, que ajusta las peculiaridades de los requerimientos a lo determinado en la anterior. En realidad, la capa lógica hace las veces de interfaz entre los requerimientos y el sistema implementado en SCP.

La organización responde a las suposiciones basadas en la información brindada con anterioridad y pequeños flujos de trabajos que se han observación en algunos puntos

Tabla: Personas					
Campo	Tipo	Nulos	Clave	Por defecto	Extra
persona_id	integer		PRI		auto_increment
first_name	varchar(50)		IND	NULL	
second_name	varchar(50)		IND	NULL	
first_last_name	varchar(50)			NULL	
second_last_name	varchar(50)			NULL	
Identificacion	varchar(50)				
email	varchar(100)	YES		NULL	
foto	varchar(max)	YES		NULL	
Fecha de nacimiento	date	YES		NULL	
genero	int	YES		NULL	
celular	int	YES		NULL	
telefono	Int	YES		NULL	
Codigo_de_pais	varchar(10)	YES		NULL	

Índices

Nombre	Tipo	Campos
PRIMARY	PRIMARY	persona_id

Tabla: Vehículos					
Campo	Tipo	Nulos	Clave	Por defecto	Extra
Vehiculo_id	int		PRI	NULL	auto_increment
Vehiculo_color	varchar(20)	YES		NULL	
Num_chasis	varchar(50)		UNIQUE	0000-00-00 00:00:00	
Num_motor	varchar(50)		UNIQUE	0000-00-00 00:00:00	
Num_placa	varchar(50)		UNIQUE	NULL	
Toneladas_Max	double			NULL	
Ejes	int			NULL	
Marca	varchar(50)	YES		NULL	
Modelo	varchar(50)	YES		NULL	
Num_circulacion	varchar(50)			NULL	

Índices

Nombre	Tipo	Campos
PRIMARY	PRIMARY	Vehiculo_id
UNIQUE	UNIQUE	Num_chasis
UNIQUE	UNIQUE	Num_motor
UNIQUE	UNIQUE	Num_placa

Tabla: Empresas Transporte					
Campo	Tipo	Nulos	Clave	Por defecto	Extra

Empresa_transp_id	int		PRI		auto_increment
nombre	varchar(200)				
Cellphone	int			NULL	
country_phone_code	varchar(10)	YES		NULL	
Persona_id	int		FK persona		

Índices	Nombre	Tipo	Campos
PRIMARY		PRIMARY	Empresa_transp_id
FORENGKEY		FORENGKEY	Persona_id

Tabla: Conductores					
Campo	Tipo	Nulos	Clave	Por defecto	Extra
Conductor_id	int		PRI		auto_increment
Person_id	int		FK persona		
Num_licencia	Varchar(50)			NULL	
Categorías_vehiculos	Varchar(50)			NULL	
Nacionalidad	Varchar(50)			NULL	
Poliza_id	int		FK poliza		
Empresa_transp_id	int				

Índices	Nombre	Tipo	Campos
PRIMARY		PRIMARY	Conductor_id
FORENGKEY		FORENGKEY	Persona_id
FORENGKEY		FORENGKEY	Poliza_id
FORENGKEY		FORENGKEY	Empresa_transp_id

Tabla: Pólizas					
Campo	Tipo	Nulos	Clave	Por defecto	Extra
Poliza_id	int		PRI		auto_increment
Num_Poliza	varchar(30)		UNIQUE		
Num_seguro	varchar(30)		UNIQUE		
Compania	varchar(50)			NULL	
Fecha_vencimiento	Datetime			NULL	
Tipo_seguro_id	Int		FK tiposeguro		

Índices	Nombre	Tipo	Campos
PRIMARY		PRIMARY	Poliza_id
UNIQUE		UNIQUE	Num_Poliza
UNIQUE		UNIQUE	Num_seguro
FORENGKEY		FORENGKEY	Tipo_seguro_id

Tabla: Tipos de seguro					
Campo	Tipo	Nulos	Clave	Por defecto	Extra
Tipo_seguro_id	int		PRI		auto_increment
Tipo_seguro	varchar(50)			null	

Índices	Nombre	Tipo	Campos
PRIMARY		PRIMARY	Tipo_seguro_id

Tabla: Puntos de control					
Campo	Tipo	Nulos	Clave	Por defecto	Extra
Punto_id	int		PRI		auto_increment
Nombre_Punto	varchar(50)		UNIQUE		
Bascula_id	int		FK Basculas		
Ubicacion	varchar(250)				
cellphone	int	YES		NULL	
number_phone	Int	YES		NULL	
Persona_id	int		FK persona		

Índices

Nombre	Tipo	Campos
PRIMARY	PRIMARY	Punto_id
UNIQUE	UNIQUE	Nombre_Punto
FORENGKEY	FORENGKEY	Persona_id
FORENGKEY	FORENGKEY	Bascula_id

Tabla: Basculas					
Campo	Tipo	Nulos	Clave	Por defecto	Extra
Bascula_id	int		PRI		auto_increment
Modelo	varchar(50)		UNIQUE		
Capacidad	varchar(250)				
Unidad de medida	varchar(5)	YES		NULL	
Marca	varchar(50)	YES		NULL	
Foto	varchar(250)				

Índices

Nombre	Tipo	Campos
PRIMARY	PRIMARY	Bascula_id
UNIQUE	UNIQUE	Modelo

Tabla: Países					
Campo	Tipo	Nulos	Clave	Por defecto	Extra
Pais_id	int		PRI		auto_increment
Nombre	varchar(50)		UNIQUE		

Índices	Nombre	Tipo	Campos
PRIMARY		PRIMARY	Pais_id

Tabla: Departamentos					
Campo	Tipo	Nulos	Clave	Por defecto	Extra
Departamento_id	int		PRI		auto_increment
Nombre	varchar(50)		UNIQUE		
Pais_id	int		FK paises		

Índices	Nombre	Tipo	Campos
PRIMARY		PRIMARY	Departamento_id
FORENGKEY		FORENGKEY	Pais_id

Tabla: Municipios					
Campo	Tipo	Nulos	Clave	Por defecto	Extra
Municipio_id	int		PRI		auto_increment
Nombre	varchar(50)		UNIQUE		
Departamento_id	int		FK Departamento		
Índices	Nombre	Tipo	Campos		
PRIMARY		PRIMARY	Municipio_id		
FORENGKEY		FORENGKEY	Departamento_id		

Tabla: Usuarios.					
Campo	Tipo	Nulos	Clave	Por defecto	Extra
Usuario_id	int		PRI		auto_increment
Persona_id	int		FK persona		
Nivel_id	int		FK niveles		
Usuario	Varchar(50)				
Contraseña	Varchar(15)				
Índices	Nombre	Tipo	Campos		
PRIMARY		PRIMARY	Usuario_id		
FORENGKEY		FORENGKEY	Persona_id		
FORENGKEY		FORENGKEY	Nivel_id		

Tabla: Niveles.					
Campo	Tipo	Nulos	Clave	Por defecto	Extra
Nivel_id	int		PRI		auto_increment
Nivel	int		FK persona		
Nombre	int		FK niveles		
Descripcion	Varchar(50)				
Índices	Nombre	Tipo	Campos		
PRIMARY		PRIMARY	Usuario_id		
FORENGKEY		FORENGKEY	Persona_id		
FORENGKEY		FORENGKEY	Nivel_id		

Tabla: Pesadas.					
Campo	Tipo	Nulos	Clave	Por defecto	Extra
Pesada_id	int		PRI		auto_increment
Punto_id	int		FK punto_control		
Usuario_id	int		FK usuario		
Conductor_id	int		FK conductore		
Vehiculo_id	int		FK Vehiculo		
Peso	Double				0
Tara	Double				0
Destino	Varchar(250)				
Índices	Nombre	Tipo	Campos		
PRIMARY		PRIMARY	Pesada_id		
FORENGKEY		FORENGKEY	Punto_id		

FORENGKEY	FORENGKEY	Usuario_id
FORENGKEY	FORENGKEY	Conductor_id
FORENGKEY	FORENGKEY	Vehiculo_id

No se incluyen las tablas que dependen de la estructura interna de funcionamiento de SQL y que no están relacionadas con la operatoria de SCP.

7 Estructura lógica de la información

A continuación, se suministra la información básica sobre la estructura lógica del sistema de información de SCP. En PDSDT se usa el término *sistema de información* (en adelante, SI) con el mismo sentido descrito por la Encyclopaedia Britannica: "*an integrated set of components for collecting, storing, and processing data and for delivering information, knowledge, and digital products*".

7.1 Elementos

Por razones de terminología propia de SCP, en PDSDT no se emplea el término *entidad* en el sentido propio de una descripción de un modelo de datos, porque podría plantear confusiones con uno de los modelos de registro utilizados. En su lugar, se usa **elemento**.

El ámbito y el alcance, así como su nomenclatura, de los elementos del SI se delimitan de acuerdo a las siguientes especificaciones:

- **Empresa de Transporte EMPT.** Se entiende como empresa de transporte (según DRAE) "cualquier corporación, compañía, institución, etc., tomada como persona jurídica que se encarga de la logística de cargas". En este caso será considerado como tal cualquier organismo con entidad legal que se dedique al rubro de transporté de cargas pesadas así se utilizará en el contexto de Sistema de control de Peso, bien como un origen de datos HTML, a partir de cuyos resultados el sistema explora la Red en búsqueda de información sobre pesos y cargas de diferentes medios de transportes que transiten las diferentes vías terrestres al pasar en un punto control así mismo el medio de carga deberá estar vinculado a un conductor con una empresa se transporte.
- **Personas PER.** Se entiende como persona en el contexto DT a quienes interactúan como actores de una acción en el sistema por ejemplo (usuario, Conductor, responsable de punto de control).
- **Vehículos VEH.** Se entiende como vehículo en el contexto DT cualquier medio de transporte que logre movilizar a un apersona u objeto a un destino distante. Su uso "natural" es para transportar cargas.
- **Conductor CON.** Se entiende como conductor a toda persona que maneja el mecanismo de dirección de un vehículo o, simplemente, los mandos cuando carece de la misma. en términos mercantiles, comerciales o de negocio.
- **Póliza POL.** La póliza de seguro es un contrato entre una persona u entidad, como asegurado y la compañía de seguros, que establece los derechos y obligaciones de ambos, en relación al seguro contratado. Relaciones entre elementos.
- **Tipos de Seguros TS.** Se trata de tipos de seguro a la multiplicidad de riesgos que pueden ser tratados mediante un contrato de seguro.
- **Punto de Control PDC.** Los puntos de control son definidos por EL CODEX como

un punto crítico de control (PCC) como "una etapa donde se puede aplicar un control y que sea esencial para evitar o eliminar un peligro a la inocuidad del alimento o para reducirlo a un nivel aceptable". En el contexto de SCP un punto de control es un lugar sobre la ruta de tránsito en el cual se chequeará la el peso del vehículo al cual se le aplicará control.

- **Basculas BSC.** Se define como Instrumento para medir pesos, generalmente grandes, que consiste en una plataforma donde se coloca lo que se quiere pesar, un sistema de palancas que transmite el peso a un brazo que se equilibra con una pesa, y un indicador que marca el peso. Este será el principal hito del sistema que se encarga de recopilar la principal información del sistema de información.
- **Usuarios USR.** Este elemento tiene el derecho de usar de una cosa ajena con unas limitaciones determinadas.
- **Niveles LVL.** Los niveles brindan acceso a determinadas áreas de un SI.
- **Pesadas PSD.** Son iteraciones del proceso de pesaje de los vehículos en el cual se recopila la mayor parte de la información que será posteriormente analizada para realizar una toma de decisión.

Los datos obtenidos se atienen al modelo expresado en las imágenes diagramáticos E/R, que se muestran en 14. En esencia, el SI agrupa la información de acuerdo al siguiente criterio (el símbolo \rightarrow significa *se relaciona con*):

Empresa de transporte

- Empresa de transporte \rightarrow Conductor:
- Empresa de transporte $[1 \rightarrow \infty]$. Vehículos.
- Empresa de transporte $[1 \rightarrow \infty]$. Pesadas.
- Empresa de transporte $[1 \rightarrow \infty]$. Puntos de control.
- Empresa de transporte \rightarrow Personas:
- Empresa de transporte $[1 \rightarrow 1]$. La persona encargada.
- Empresa de transporte $[1 \rightarrow 1]$. Localidad u nacionalidad.

Vehículos

- Vehículos \rightarrow Conductor:
- Vehículos $[1 \rightarrow 1]$. Conductor.
- Vehículos $[1 \rightarrow 1]$. Póliza.
- Vehículos $[1 \rightarrow \infty]$. Pesadas.

Personas

- Personas \rightarrow Empresas de Transporte:
- Personas $[1 \rightarrow 1]$. Representante de empresa
- Personas \rightarrow Puntos de control:
- Personas $[1 \rightarrow \infty]$. Usuarios
- Personas $[1 \rightarrow 1]$. Representante de Punto de control.
- Personas \rightarrow Conductor:
- Personas $[1 \rightarrow 1]$. Conductor.

- Dependencia [1→1]. La persona (derecha) es la principal en una relación descendente de dependencia de la persona.

Polizas

- Polizas → Tipos de seguros.
- Polizas [1→1]. Tipo de seguro que ejerce la poliza.

Conductor

- Conductor → Conductor.
- Conductor [1→∞]. Polizas.

Puntos de control

- Puntos de control → Basculas
- Puntos de control [1→1]. Basculas.
- Puntos de control → persona.
- Puntos de control [1→1]. Persona.
- Puntos de control [1→1]. Municipio.
- Puntos de control [1→∞]. Pesadas.

Usuarios

- Usuarios → persona
- Puntos de control [1→1]. persona.
- Puntos de control [1→1]. Nivel.

8 Diseño de la arquitectura del sistema

Para la fase de diseño, y de acuerdo con el análisis de ingeniería de requisitos a realizar a partir de las especificaciones planteadas por MTI, SCP se concibió como un aplicativo, adherido al paradigma **cliente-servidor** siguiendo un modelo **SaaS** —si bien, obviamente, sin los condicionantes económicos—, realizado sobre una estructura a modo de consultas al servidor entendiéndolo a la manera en que DocForge¹⁰ lo define: "un conjunto de código fuente o bibliotecas que proporcionan una funcionalidad común a toda una clase de aplicaciones".

8.1 Descripción general

Por facilidad de implementación, VB es el lenguaje de desarrollo elegido, sobre una plataforma Windows. Durante la fase de diseño se realizaron investigaciones con diversas distribuciones *framework* bien conocidas y utilizadas para desarrollos FrontEnd y Backend, como *Bootstrap*, *Swaggers* o *MVC*.

La prensa *online* especializada anda ciertamente dividida en cuanto a la posibilidad de usar ASP.net con WebForms como estructura pero si está de acuerdo en utilizar la estructura en capas , con posturas a menudo encontradas y, hasta cierto punto, escasamente razonadas en numerosas ocasiones. Sin embargo, es evidente que ASP.net con WebForms ofrece todo un complejo mundo de soluciones TI, basado en la exposición de sus rutinas mediante mecanismos de *hooking* y filtrado, y facilita, como parte de su diseño interno, la realización tareas y

operaciones tan esenciales como:

- Completa y razonablemente segura gestión de usuarios.
- Excelente motor de plantillas, basado en la abstracción de las mismas en un conjunto denominado "tema", que permite su explotación coordinada y estética y funcionalmente coherente.
- Mecanismo sólido para el registro de errores.
- Complementos de paquetes para la gestión de formularios y la validación de datos.
- Consistente grupo de entidades y clases para la manipulación de objetos.
- Uso de *hooks*, clases abstractas, prototipos y mecanismos de tipo *plugin* que permiten ampliar o modificar la funcionalidad básica.
- Sistema semiautomatizado de ejecución planificada de código.
- Robusto y eficiente motor de *caching*, esencial para obtener buenos resultados en cuanto a tiempos de operación con el equipo de trabajo.

Adicionalmente, al tratarse de un Sistema soportado por Microsoft acogido a la distribución bajo licencia GPLv2 (o posterior), resulta fácilmente mantenible, reuniendo todo el esfuerzo conjunto de una amplísima comunidad de desarrolladores y usuarios, de manera que la observación colectiva permite una rápida actuación frente a cualquier vulnerabilidad detectada. Además se cuenta con amplia documentación HTML que es un lenguaje de marcado que se utiliza para el desarrollo de páginas de Internet. Se trata de la sigla que corresponden a HyperText Markup Language, es decir, Lenguaje de Marcas de Hipertexto

Por todo lo expresado, la decisión sobre el ASP.net con web Forms se deba emplear resultó claramente favorable. Y desde el principio, de acuerdo con lo dicho sobre el mantenimiento de un aceptable nivel de seguridad, se trató de NO EMPLEAR NINGÚN MECANISMO QUE INTERFERIESE CON LA ESTRUCTURA INTERNA Y LA FUNCIONALIDAD NATIVA DEL MTI, de manera que fuese sencillo y exento de trabajo adicional o de riesgos (perder lo realizado, abrir "puertas traseras" o recuperar vulnerabilidades que se creían eliminadas) actualizar el núcleo del sistema sin mermar por ello su rendimiento y las garantías de aislamiento de peligros externos. En 14 se muestran imágenes que ilustran adecuadamente la estructura funcional en capas.

A partir de lo recogido en 7, es posible expresar en forma simplificada la estructura del SI SCP en la imagen que se muestra a continuación, a modo de diagrama relacional entre elementos.

¹¹ Multiple (wiki). "Web application framework". Docforge. Consultado 2015-01-19.

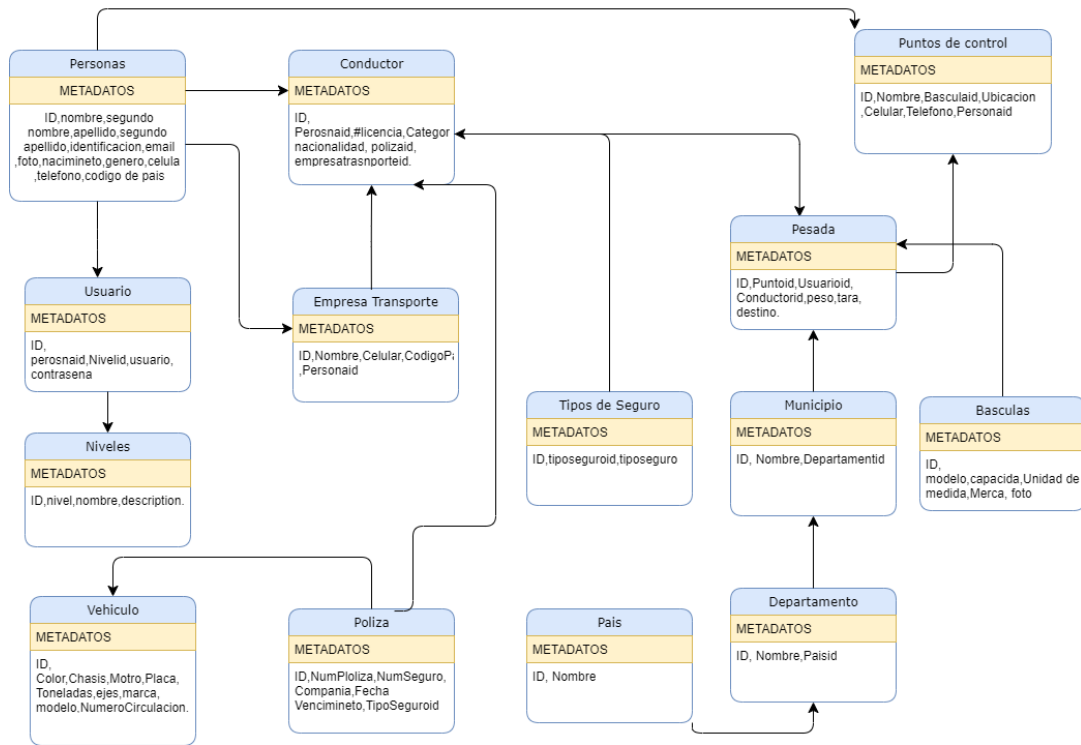


Ilustración 6. Modelo simplificado SI SCP

A partir de ella, se pueden establecer las relaciones entre los diversos dominios de información presentes en los requisitos, tal y como se exhiben en las siguientes tablas.

ELEMENTO	OBTIENE	DE
Pesadas	Fuente de información	Elementos
Personas	Catálogo	Insert
Vehiculo	Catálogo	Insert
Poliza	Catálogo	Insert
Departamento	Catálogo	Insert
Pais	Catálogo	Insert
Municipio	Catálogo	Insert
Conductor	Catálogo	Insert
Puntos de control	Catálogo	Insert
Basculas	Catálogo	Insert

SCP está diseñado bajo estructura ya se han realizado al menos dos proyectos de este mismo rubro y envergadura para transformarlo en un modelo lo más universal posible que permita realizar desarrollos sobre el.

Un FW consiste en un conjunto de convenciones -como dónde deben guardarse los archivos- así como de bibliotecas y herramientas -pongamos por caso, una capa de abstracción de base de datos- que nos permiten fácilmente comenzar a trabajar en una aplicación. En resumen, proporciona los medios por los cuales un programa puede ser construido partiendo de cero, a partir del esquema de base de datos hasta su *front-end*.

Existen en la Red proyectos que basan su funcionamiento en implementar un mecanismo MVC (*Model, View, Controller*) sobre ASP.net, que indudablemente hacen gala de una gran calidad y que usan una metodología perfectamente adaptada a los principios más adecuados de ingeniería del software. Pero fueron descartados para SCP porque hacían excesivamente complejo el desarrollo y obligaban a “forzar la máquina”, teniendo que realizar adaptaciones y modificaciones en el núcleo principal (*core*) de nuestra aplicación, lo que suponía renunciar a algunas de sus ventajas:

- Una base de datos con un esquema organizativo flexible y muy sólido.
- La facilidad para gestionar tareas en las caras de servidor y cliente, y asumiendo roles de administrador o usuario.

Los elementos y puntos fuertes de ASP.net como Modelo de capas que a continuación se relacionan se incorporan a SCP como núcleo fundamental sobre el que construir la arquitectura SW.

- Comunidad de desarrollo, soporte y documentación. Microsoft dispone de una gran comunidad y un foro de soporte activo. Ya se trate de un problema con la instalación, la configuración, desarrollo o cualquier otra cosa, normalmente cualquier problema se resuelva satisfactoriamente con la ayuda de la comunidad de apoyo.
- Gestión de usuarios. La seguridad no es, en el caso de aplicaciones abiertas a su uso en Internet, un problema menor.
- Pantallas de administración. ASP.net ofrece todo un interfaz, que permite crear páginas propias para poder hacer frente a una amplia serie de requisitos.
- Funcionamiento CRUD. Supone un enorme ahorro de “energía programadora” y de tiempo contar con un muy contratado mecanismo que controla absolutamente todo el proceso de dar de alta, de baja, modificar o leer cualquier registro.
- Completísimo juego de funciones internas perfectamente aprovechables. Incluyen desde la gestión de la internacionalización a través de un avanzado control hasta la “desinfección” preventiva de textos, bloqueando de esa forma posibles ataques de tipo *SQL injection* y variantes.
- Administración de medios. se ocupa de la carga de archivos y la gestión de los medios audiovisuales.
- Extensibilidad y escalabilidad. Desde el punto de vista del desarrollo, extensibilidad y escalabilidad constituyen las dos piedras de toque esenciales para adoptar una decisión sobre la infraestructura o el paradigma a emplear. Todos sabemos que lo que empieza como un pequeño proyecto acaba ampliándose ad infinitum.

- Plantillas. La presentación o *front-end* no es un mero adorno. Mediante una peculiar arquitectura basada en funciones y con extenso uso de retrollamadas.

82 Ingeniería de requisitos

Dadas las peculiaridades y el ambiente de SCP, el actor Supervisor se encarga de la especificación de requisitos. Según expresaba bajo conocimientos preliminares se trataba de explotar la función recopilación de la información en las diferentes vías del país. Es un proyecto que también se enfoca a mejorar la sociedad facilitando el acceso y la reutilización de contenidos digitales, y en este sentido plantea como uno de sus resultados desarrollar una plataforma web que proporcionará información de interés al usuario. Por tanto, se imponía desde un primer momento desarrollar una aplicación accesible desde la Red capaz de cubrir las necesidades que se planteaban.

En el caso de SCP, por tanto, no se han utilizado metodologías al uso, tales como las entrevistas, talleres, lluvia de ideas o contrato formalizado como fuente de especificaciones de requerimientos, sino un proceso analítico a partir de lo mencionado. Dicho proceso ha sido acometido por JGBH y VPTP, en conjunto con los partícipes cuyo cargo es de *Supervisor*, y ha sido concretado en este documento, en el cual se ha elaborado la estructura lógica y física del proyecto SCP, y que no se adjunta en el presente PDSDT.

Mediante la consulta a los Supervisores se ha realizado la **matriz de requisitos y procesos** que a continuación se recoge bajo supuestos esta información, según su clasificación. Téngase en consideración que dicha matriz incluye referencias a los **procesos** y subprocesos del sistema, que están especificados en 9. De acuerdo a la importancia o la trascendencia a los propósitos de SCP, los requisitos se han clasificado en *funcionales*, *organizacionales* y *no funcionales*. En todos los casos, la información que de ellos depende puede ser o no de carácter **crítico**. La **atribución** de un requisito a determinado actor o proyecto explicita para quién debe realizarse la solución propuesta en cada caso y qué elemento es responsable de su verificación final.

8.2.1 Requisitos funcionales

REQUISITO	ATRIBUCIÓN	PROCESO	CRÍTICO
Recopilar información sobre los pesos de los equipos de transporte que es de suma importancia para MTI.	SCP	Búsqueda ¹³ Grabación	Sí
Agregar a la información de Empresa de transporte que puedan ser usadas para automatizar el proceso de recogida de información.	SCP	Grabación	Sí
Automatizar la recogida de información sobre Conductores mediante la copia de documentos personales.	SCP	Captura	Sí
Automatizar la recogida de información sobre vehículos mediante la observación de la información en los objetos.	SCP	Captura	Sí
Búsqueda de los datos de empresa de para referenciarlos a un vehículo.	SCP	Filtrado	Sí
Completar la metainformación pertinente para la información sobre el pesaje automáticamente capturada y seleccionada.	SCP	Grabación	Sí
Facilitar la transferencia de la información recopilada y verificada a actores externos a SCP para su posterior uso en tareas de análisis.	SCP	Exportación	Sí
Facilitar el acceso para consulta, monitorización y análisis (a menor escala que en el caso del realizado por los actores de SCP) a la información recopilada.	SCP	Filtrado	No

8.2.1 Requisitos organizacionales

REQUISITO	ATRIBUCIÓN	PROCESO	CRÍTICO
Poder realizar clasificaciones taxonómicas sobre la información y metainformación recopilada	SCP	Grabación	Sí
Poder explotar la información y metainformación recopilada desde un punto de vista de su cobertura temporal	SCP	Grabación	Sí
Poder explotar la información y metainformación recopilada desde un punto de vista de su cobertura geográfica	SCP	Grabación	Sí

8.2.1 Requisitos no funcionales

REQUISITO	ATRIBUCIÓN	PROCESO	CRÍTICO
Garantizar la seguridad de acceso y de conservación de la información recopilada	Desarrollador	Administración	Sí
Garantizar la coherencia de la metainformación recopilada mediante los vocabularios adecuados.	SCP Desarrollador	Desarrollo	Sí
Utilizar tecnología que aúne a la vez simplicidad y potencia bajo la filosofía KISS	Desarrollador	Administración	Sí
Permitir que el sistema de filtrado automático disponga de un mecanismo de aprendizaje basado en procedimientos heurísticos.	Desarrollador	Filtrado	No
Garantizar la disponibilidad de acceso a los equipos de trabajo, con un mínimo de “tres nueves” ¹⁴	Desarrollador Administrador	Administración	No

83 Implementación de la solución

De acuerdo con lo expresado en 8.1, SCP adopta como arquitectura la construcción de un artefacto sustentado en un modelo de capa utilizado como FW Bootstrap. No obstante, es necesario precisar que hay tres formas de construir la estructura de la aplicación con las citadas premisas.

Interviniendo sobre el núcleo funcional (core) de Microsoft core .net y adaptándolo a las necesidades de SCP. Descartado desde un principio para poder conseguir el mantenimiento de seguridad y funcionalidad derivado de las actualizaciones periódicas de Microsoft, como se ha dicho en 8.1, sin necesidad de acciones específicas.

Utilizando un *plugin*. Un *plugin* o complemento es una aplicación que se relaciona con otra para aportarle una función nueva y generalmente muy específica. Esta aplicación adicional es ejecutada por la aplicación principal e interactúan por medio de un API. o formato externo de la aplicación sobre ellos construida. El API está disponible por completo mediante tres sistemas:

- **Hooks.** O “ganchos”. Permiten al *plugin* incorporarse al sistema como un elemento más sin necesidad de modificar la estructura core interna.
- **Acciones.** Son causadas por eventos específicos que ocurren en SCP, como publicar un *post*, cambiar de plantilla (“tema”), o mostrar una página en el panel de administración.
- **Filtros.** Son funciones a través de las cuales ASP.net transmite datos en ciertos puntos de la ejecución, justo antes de emprender alguna acción con esos datos (como agregarlos al SGBD o enviarlos a la pantalla del navegador).

¹² Recuérdese que, en el caso de EF, coincide con el *Product Owner* de la filosofía SCRUM, lo que le faculta para decidir los obje-

tivos finales en función de su pertinencia para la obtención de resultados en EX.

En 14 figura una representación gráfica del *stack* o pila de capas de funcionamiento de SCP. A través de dicha representación es posible conocer la forma en que interactúan los diferentes elementos de un tema con el núcleo operativo. A continuación, se muestra una matriz de los subsistemas Microsfto y Webs utilizados desde SCP para implementar su funcionalidad.

SUBSISTEMA WP	FUNCIONALIDAD EF
Ajax	de funciones ejecutadas en cliente Control
CURL	Captura automatizada de datos
Database	Acceso directo a SGBD. Mantenimiento
Mail	Comunicación con usuarios (seguridad)
REST	Exportación de datos. Ajax (funciones ejecutadas en cliente)
ltextSharp	Manejo y construcción de documentos.
Sanitization	Seguridad frente a intrusiones
Usuarios	Acceso granular a funcionalidades EF. Seguridad de acceso

La estructura detallada de la solución arquitectónica del tema figura en 11.

9 Resumen de procesos del sistema

Dado el limitado alcance de PDSDT en la versión actual (1.0), en el presente apartado se han simplificado los casos de uso y la descripción de los principales procesos se exhibe agrupada por funcionalidad y extensibilidad.

Como ha quedado determinado en 8.2, el conjunto completo del proyecto SCP incluye dos grandes bloques de trabajo, que determinan, en gran medida, el ritmo y el formato de su despliegue:

- Por una parte, un “motor” que posibilite la extracción y el almacenamiento de la información sobre los pesos obtenidos en SCP.
- Por otra, dicho “motor” debe poder exportar los registros conseguidos de forma que sea factible su explotación en plataformas diseñadas para el análisis de grandes volúmenes de datos mediante técnicas de descubrimiento de relaciones y patronizado de estructuras reticulares.

9.1 Actores y casos de uso. Modelos básicos

Se denomina caso de uso (CDU en adelante) a una descripción de las actividades que deberán realizarse para llevar a cabo algún proceso. Los personajes o entidades que participan se denominan *actores*. En el ámbito de la ingeniería del software, la expresión alude a la secuencia de interacciones que deben realizarse entre un sistema y sus actores en respuesta a un evento que inicia un actor sobre el propio sistema.

¹³ La disponibilidad es usualmente expresada como un porcentaje del tiempo de funcionamiento en un año dado. La regla de los “tres nueves” implica que el sistema deberá estar disponible todo el tiempo salvo 99,9% = 43.8 minutos/mes u 8,76 horas/año.

9.1.1 Actores

La lista de actores que intervienen en EF se muestra a continuación.

NOMBRE	DESCRIPCIÓN
Administrador	Usuario que gestiona el SO del servidor y el funcionamiento interno de SCP
Revisor	Usuario que verifica la correcta aplicación de la normativa sobre coherencia de la información. Tiene derechos de modificación de la información almacenada por <i>Grabador verificador</i> .
Grabador verificador	Valida la información capturada por SCP. Descarta aquella que pueda haber sido filtrada como adecuada por error. Tiene derechos de modificación de la información de la cual es responsable.
Sistema	Sistema de software SCP
Supervisor	Usuario que valida la información, autoriza sus modificaciones y evalúa el éxito de SCP. Tiene derechos de modificación de la información almacenada por <i>Grabador verificador</i>
Usuario externo	Usuario con un nivel de autenticación mínimo (identificación, sin derechos de acceso a back-end)
Usuario no perteneciente a RRHH	Usuario que no se ha identificado frente al sistema. No puede acceder a front-end ni a back-end

9.1.2 Casos de uso

En el contexto de EF, y teniendo en cuenta que, tal y como se ha establecido desde 1.3, el alcance del presente artefacto PDSDT es limitado, se expresan los casos de forma genérica, adaptándolos a los principales procesos del sistema.

Nombre	Captura de información
Descripción	SCP capta registros de tipo entidad que contengan información sobre Conductores, Vehículos, Personas, Puntos de Control, países, Departamentos, municipios, empresa de transporte
Actores	Administrador, Grabador verificador
Precondiciones	Deben existir las tablas de la base de datos normalizada y relacionada para su debida recepción.
Flujo normal	SCP consulta SGBD usando la cadena de conexión para realizar la selección y solicitud de datos determinados.
Flujo alternativo	SCP dispara una condición de error usando en caso que no esté la información
Postcondiciones	Captura de información debe disponer de una lista de datos para seleccionar.
Referencias	Lectura de Datos

Nombre	Lectura de Datos
Descripción	SCP lee la lista de registros que estan disponibles en la base de datos previamente cargados.
Actores	Usuario externo.
Precondiciones	Debe existir la lista de datos o registros.
Flujo normal	SCP lee una solicitud. Mediante cadena de conexión extrae el resultado.
Flujo alternativo	SCP dispara una condición de error usando consultas o procedimientos de almacenados.
Postcondiciones	Filtrado deba disponer de un contenido analizable.
Referencias	Filtrado

Nombre	Filtrado
Descripción	En dependencia de lo que se solicite con los parámetros enviados, serán recepcionados por los procedimientos de almacenados y este mismo retornara los valores especificados.
Actores	Usuario externo, Grabador verificador
Precondiciones	Debe existir la información relevante. Debe poder ser transformada desde su formato original en texto.
Flujo normal	SCP lee la información relevante. Evalúa el contenido mediante aplicación de algoritmo de filtro. Transfiere resultados a Aprendizaje.
Flujo alternativo	SCP dispara una condición de error usando en caso que no esté la información

Postcondiciones	Debe disponer de un contenido en formato texto.
Referencias	Captura de información, Análisis

Nombre	Análisis
Descripción	SCP usa la información filtrada. Si procede de Filtrado como válida, aplica un post análisis para tomar decisiones para poder aplicarle una sanción a un conductor o en caso contrario, aplica el mismo algoritmo y la clasifica como valida.
Actores	Sistema, Revisor
Precondiciones	Debe existir la información evaluada.
Flujo normal	SCP lee la información evaluada. Aplica algoritmo de clasificación y la clasifica.
Flujo alternativo	SCP dispara una condición de error usando en caso que no esté la información
Postcondiciones	Grabación debe disponer de respuesta completa proporcionada por Captura de información y del resultado del filtro y de la clasificación.
Referencias	Grabación

Nombre	Grabación
Descripción	SCP recoge la información transferida por análisis. Añade la información. Graba un registro de tipo pesada en SGBD. Atribuye dicho registro al actor Grabador verificador que grabó la información de la entidad origen.
Actores	Sistema, Revisor, Grabador verificador
Precondiciones	Debe existir la información capturada, filtrada y aprendida. Debe existir la información asociada.
Flujo normal	SCP lee la información capturada, filtrada y analizada. Graba una pesada en estado correcta. Y retorna si es válida o no.
Flujo alternativo	SCP dispara una condición de error usando en caso que no esté la información.
Postcondiciones	Grabador verificador debe disponer de un registro con la información capturada.
Referencias	Análisis, Verificación

Nombre	Verificación
Descripción	Grabador verificador lee registros grabados por Grabación asignados a su usuario de SCP. Verifica que la información capturada es válida.
Actores	Grabador verificador, Sistema
Precondiciones	Debe existir un registro guardado asignado al usuario SCP y procedente de un punto de control que recopile la información actora.
Flujo normal	Grabador verificador lee registro guardado. Verifica que el trabajo sea posible de automatizar, si los registros son correctos ya no habrá intervención del actor verificador.
Flujo alternativo	Grabador verificador constata que el registro guardado no cumple las condiciones para ser válido. Lo marca como "Deleted".
Postcondiciones	Grabador verificador debe disponer de un registro con la información capturada y la información.
Referencias	Grabación, Revisión

Nombre	Revisión
Descripción	Revisor lee registro grabado para evaluar comprueba la adecuación de la metainformación grabada a la normativa sobre taxonomía y metainformación (vocabularios controlados). En caso de no adecuación a norma, utiliza flujo alternativo.
Actores	Grabador verificador, Revisor, Supervisor
Precondiciones	Debe existir un registro guardado asignado a Grabador verificador.
Flujo normal	Revisor lee registro grabado por Grabador verificador. Verifica validez de datos.
Flujo alternativo	Revisor constata que la información grabada en el registro guardado no se adecua a la normativa sobre información (vocabularios controlados).
Postcondiciones	Revisor debe disponer de un registro grabado.
Referencias	Verificación, Borrado, Supervisión

Nombre	Supervisión
Descripción	Revisor recibe un <i>feedback</i> del Supervisor. Evalúa si se trata de una actuación errónea o hay que modificar el vocabulario controlado o las normas de verificación.
Actores	Grabador verificador, Revisor, Supervisor
Precondiciones	Debe existir una <i>feedback</i> de Supervisor.
Flujo normal	Supervisor recibe un <i>feedback</i> de Revisor. Evalúa positivamente la solicitud. Realiza los cambios en el vocabulario controlado o modifica las normas
Flujo alternativo	Supervisor recibe un <i>feedback</i> de Revisor. Evalúa negativamente la solicitud. Envía mensaje por mensajería interna a Revisor y a Grabador verificador para que se efectúe Borrado.
Postcondiciones	Supervisor debe disponer de un <i>feedback</i> .
Referencias	Revisión.

En 14 se incluye un diagrama global y genérico del modelo de casos de uso.

92 Procesos

Los procesos de SCP se agrupan en **principales**, derivados de los CDU, y **auxiliares**, necesarios para el funcionamiento del sistema. Cada proceso se corresponde con un elemento dentro del conjunto de interfaces.

9.2.1 Procesos principales

Generados a partir de los casos de uso. Su equivalencia dentro de los sistemas de SCP se expresa en la siguiente tabla.

Captura de información	PROCESO	SISTEMA
Captura de información	Captura	SCP
Lectura de Datos	Captura	SCP
Filtrado	Filtrado	SCP
Análisis	Filtrado	SCP
Grabación	Grabación	SCP
Verificación	Grabación	SCP Admin
Revisión	Grabación	SCP Admin
Supervisión	Grabación	SCP Admin

9.2.2 Procesos auxiliares

Generados a partir de las necesidades derivadas de los casos de uso y de los requisitos. Se citan en la siguiente tabla.

PROCESO	DESCRIPCIÓN
Ayuda contextual	Adaptada a cada pantalla del interfaz back-end. Disponible para todos los usuarios.
Detalles de pesadas	Mostrar el historial de las pesadas realizadas.
Avisos administrativos	Emitidos por Supervisor o Administrador.
Control de calidad y Actividad	Evaluación automática de errores formales, estadísticas de trabajo y sesiones por usuario A través de un Dashboard.
Exportación de documentos	Exportación de datos en distintos formatos Xls, Pdf u otro en específico que solicite el MTI.
Mantenimiento	Realización de tareas de optimización, inicialización y mejora de los datos guardados en SGBD.
Ajuste	Modificación y control de los objetivos numéricos finales de SCP. Control de otros parámetros como unidades de medidas y monedas.
Consultas	Generación de <i>queries</i> SQL para uso en front-end. Disponible solo para Administrador.

10 Interfaces del sistema

Como ha quedado establecido en 9, existen bloques de interfaces que permiten acceder a las funciones de aplicación SCP. No todos los interfaces se corresponden con tareas. Por ejemplo, los procesos *Captura* y *Historial*, al ser ejecutados por el proceso del sistema SCP precisan disponer de una “cara” en pantalla o en consola. En realidad, sí que existe, a efectos de depuración, una salida estándar para ambos casos se utilizaran interfaz basada en Web Forms, pero no forma parte de una especificación formal así que el diseño es libre.

En esta descripción de interfaces NO SE INCLUYE INFORMACIÓN sobre elementos propios del sistema (gestión de usuarios, plugins, apariencia, actualizaciones,...) por considerarlo fuera del alcance del presente artefacto PDSDT.

INTERFAZ	DESCRIPCIÓN
Back-end	Interfaz administrativo. Permite operaciones CRUD, ajustes y monitorización del estado del proyecto
Front-end	Interfaz público ¹⁶ . Permite búsquedas de datos sobre todos los tipos de registro y consultas de carácter estadístico.

10.1 Back-end

Se ciñe estrictamente al interfaz diseñado por el grabador verificador, pero añade pantallas de catálogos que servirán para agregar valores que se utilizaran en otras tareas automatizadas del sistema, alterando algunas de las informaciones parametrizadas. Dentro del back-end se pueden distinguir dos tipos de pantalla:

- **Pantallas CRUD.** Se usan para la gestión y el mantenimiento de todos los tipos de *custom posts*: *Basculas*, *personas*, *Empresas de transporte*, *municipios*, *vehículos* y el de uso exclusivo de Administrador *consultas*.
- **Pantallas informativas.** Se usan para informar a los usuarios sobre datos que pueden ser de su interés: *Dashboard*, *Actividad*.

Las pantallas CRUD incluyen, básicamente, un completo formulario con los diversos tipos de campos que almacenan la información y la información de cada registro. Esos campos pueden ser:

- Campos de texto no editables. Solo muestran información.
- Campos de texto editables. Longitud limitada. Pueden contener mecanismos de verificación para ajuste a normas y vocabularios controlados.
- Campos de fechas. Incluyen un gráfico de calendario.

- Campos autorreferenciales. Muestran una lista de registros ya guardados en el sistema.
- Campos selectores. Botones de radio, casillas de verificación y desplegables. Permiten selecciones simples o múltiples.

Cada formulario CRUD incluye un mecanismo de verificación de tipo de dato por razones de seguridad e integridad de la información. Algunos formularios (como el de Cedula) incluyen procesos de supervisión o verificación externa. Todos los formularios incluyen mecanismos de supervisión de la calidad de ciertos campos, según las reglas de integridad y las normativas.

10.2 Front-end

En el momento de elaborar la versión actual de PDSDT, se encuentra en fase pre desarrollo y no ha sido comenzado. Para la elaboración se ha seguido una plantilla basada en el FW CSS3 + HTML5 + JavaScript Bootstrap 5.0, por la simplicidad de sus elementos, basados en un formato de bloques de color planos, sin simulación de relieve (no *skeumórficos*). Básicamente se agrupa en 4 grandes bloques.

- Pantalla de Dashboard. Simple *splash* de conexión. Con información resumida sobre las estadísticas de grabación de los distintos tipos de PCs y sus etiquetas más significativas, así como sobre el número de URIs HTML activas. Además, información gráfica sobre:
 - Grado de pesadas.
 - Porcentaje de pesajes buenos y malos registrados en el SCP de los vehículos.
 - Registros de múltiples ciudades más activas de las zonas donde se ubican los puntos de control.
- Menú Ayuda. Incluye información resumida sobre SCP, equipo de trabajo y estado del proyecto SCP.
- Menú Documentos. Incluye información sobre SCP y otro material de carácter técnico.
- Menú Catálogos. Incluye tres submenús.
 - **Vehículos.** Registra datos de un vehículo.
 - **Conductor.** Datos de un conductor.
 - **Empresa de transporte.** Datos de la empresa de transporté.
 - **Puntos de control.** Datos de puntos de control.
 - **Usuarios.** Manejo de usuarios.
- **Configuraciones:** mostrara datos parametrizados.
- **Reportes**
 - Tabla simple exportable como XLS, CSV, PDF o al portapapeles del SO, ordenable y buscable.

¹⁷ Según el esquema oficial de URIs de IANA. Véase <http://www.iana.org/assignments/uri-schemes/uri-schemes.xhtml>.

11 Gestión de riesgos

Se entiende por riesgo en el contexto PDSDT cualquier acontecimiento futuro o no predecible que pueda afectar de forma negativa a SCP. No todos los riesgos tienen la misma probabilidad de ocurrir, ni el mismo impacto, por lo que a la hora de tratarlos DT opta por seguir una pauta establecida como normal.

En SCP se clasifican los riesgos de acuerdo a la matriz que a continuación se expone, basada en los estudios de Pressman²⁰, clasificándolos por su tipología de acuerdo a las fases expresadas en 4.2.

TIPO	ELABORACIÓN	CONSTRUCCIÓN	TRANSICIÓN	MANTENIMIENTO
Técnicos	Requisitos	Diseño Implementación Interfaz	Verificación	Control de calidad
Del proyecto	RRHH Tiempo Requisitos	RRHH Tiempo	RRHH Tiempo	
Del negocio		Apoyo Presupuesto	Presupuesto	Presupuesto

Como base de trabajo para la gestión de riesgos, SCP usa la **estrategia proactiva** (no reactiva) con los siguientes principios:

- Se realiza un estudio de los riesgos potenciales (conocidos y desconocidos).
- Se miden las posibilidades de que se conviertan en reales y se cuantifican los problemas que puedan surgir.
- Se redacta una lista priorizada de los riesgos detectados.
- Se realiza el plan de gestión de riesgos, es decir, se planifica el proyecto para evitar los riesgos o minimizarlos.
- En caso de no dar resultado se dispondrá de planes de contingencia.
-

En cuanto al plan de gestión de riesgos, se abordan de acuerdo a la clasificación propuesta por Boehm²¹ (modelo en espiral).

VALORACIÓN DEL RIESGO	CONTROL DEL RIESGO
Identificación	Planificación de la gestión
Análisis de las causas	Resolución
Priorización según gravedad	Monitorización del resultado de la solución adoptada

La estrategia proactiva adoptada sigue unos **modelos preventivos** basados en los **principales factores de riesgo**, que se identifican a continuación. Para los restantes sucesos imponderables y con implicaciones de riesgo, Supervisor, en coordinación con Administrador y Desarrollador, previa reunión, adoptarán las medidas de corrección oportunas en cada caso.

²⁰ PRESSMAN, R. S. *Ingeniería del Software: un enfoque práctico*. 2005, McGraw-Hill

²¹ BOHEM B. ["A Spiral Model of Software Development and Enhancement"](#). *IEEE Computer*, IEEE, 1988.

FACTOR RIESGO	ACCIONES PREVENTIVAS
Pérdida de información	Copias de seguridad automatizadas en servidor, con ciclo incremental diario, a partir de las herramientas ofrecidas por SO y SQL Server.
Pérdida de código	Copias de seguridad automatizadas en servidor, con ciclo incremental diario, a partir de las herramientas ofrecidas por SO. Uso del recurso proporcionado por la herramienta Git con uso de repositorios privados.
Acceso no deseado a información	Uso de niveles de usuarios para asegurar la identificación de cualquier usuario externo.
Falta de competencia adecuada de miembros de RR.HH.	Revisión de vinculación laboral y/o académica con los miembros afectados. Renovación de plantilla.
Extracción de información no autorizada	Durante proceso Exportación. Aseguramiento de las transacciones queden registros de los usuarios que se han logueado.

12 Verificación de la calidad

Para la verificación de la calidad se recurre al modelo popularizado por PMBOK, en una versión simplificada donde solamente se toman en cuenta determinados parámetros, tanto desde el punto de vista MTI como desde el de Desarrollador.

El *Project Management Institute* (PMI) distingue tres procesos en la gestión de la calidad (GC) de un proyecto, que SCP procura seguir.

Planificación [GCP]: Identificación de los requisitos de calidad y/o normas del proyecto, documentando la manera en que se demostrará el cumplimiento de los mismos.

Aseguramiento [GCA]: Auditoría de los requisitos de calidad y los resultados obtenidos a partir de medidas de control de calidad, a fin de garantizar que se utilicen definiciones operacionales y normas adecuadas. También implica el perfeccionamiento y la mejora del proceso con el fin de eliminar las actividades que no agreguen valor.

Control [GCC]: Monitorización y registro de los resultados de la ejecución de las actividades de calidad, a fin de evaluar el desempeño y recomendar cambios necesarios. Permite validar que los entregables y el trabajo cumplen con los requisitos especificados.

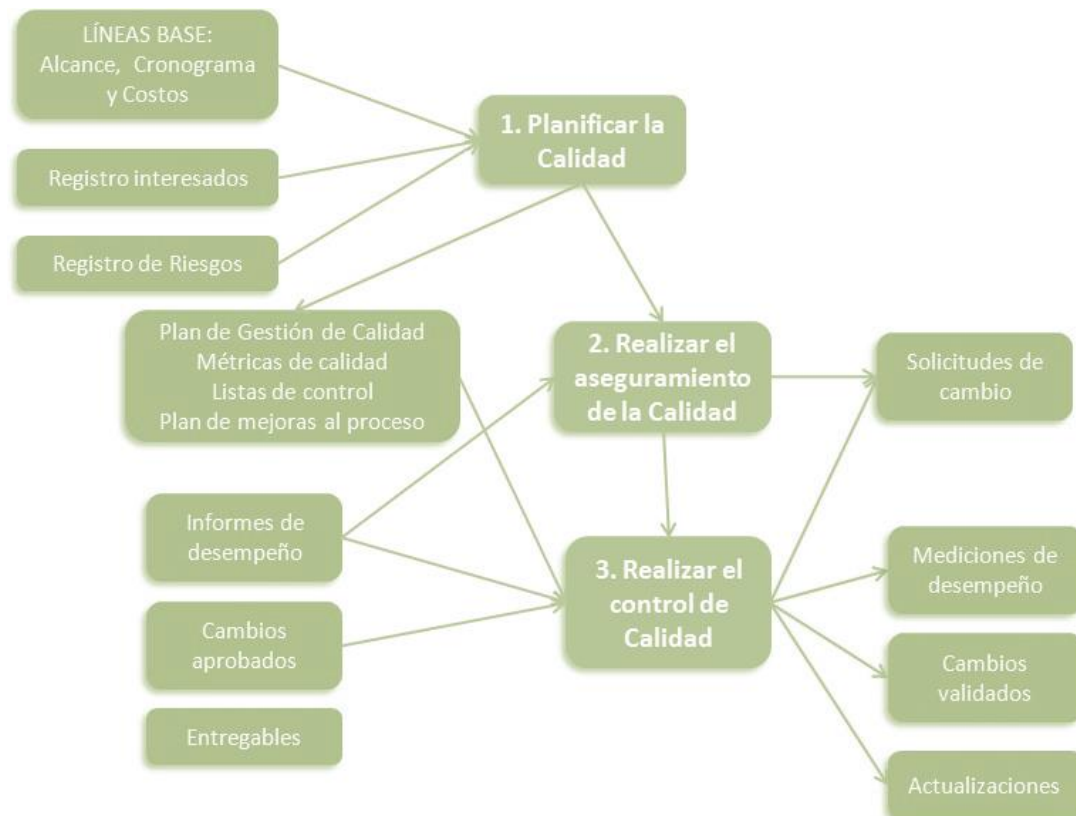


Ilustración 7. Procesos GC según PMI

SCP considera que GC es una herramienta esencial para medir el grado de éxito, y por ello implementa el núcleo de los tres procesos mencionados a través de diversos mecanismos. GCP está presente tanto en el propio diseño del proyecto como en la arquitectura y la funcionalidad interna de SCP. GCA se ha implementado mediante una política de información continua sobre la calidad y

el estado del proyecto. GCC utiliza un conjunto de métricas para obtener una valoración final del nivel de calidad.

Además de los establecidos por PBMOK, y dada la naturaleza informativa y de análisis de SCP, además de la cualidad del proyecto de ser un SI, existen condicionantes de calidad adicionales. En la siguiente tabla se detallan las acciones de verificación que se realizan para culminar cada proceso.

PROCESO	ACCIONES		EN PDSDT
Planificación	1	Cumplimiento de restricciones y suposiciones. Se verifica que todos los puntos establecidos se cumplen	2.2
	2	Cumplimiento de requisitos. Se verifica que todos los requisitos iniciales se cumplen	8.2
	3	Cumplimiento del plan de fases. Se verifica que los entregables cumplen el plan establecido.	4.2.1
	4	Cumplimiento de tabla de esfuerzo. Se verifica que la grabación alcanza las cifras estipuladas	4.1
Aseguramiento	5	SCRUM. Reuniones de Sprint. Se discute sobre la información de errores individual. Se discute sobre grado de cumplimiento de esfuerzo.	2.6
Control	6	Información en tiempo real. Grado de cumplimiento de esfuerzo. En interfaz back-end. Accesible a todos los usuarios.	10.1
	7	Información en tiempo real. Número de errores por actor. En interfaz back-end. Accesible a todos los usuarios.	10.1
	8	Suministro de información. Errores formales individuales por actor. En interfaz back-end. Accesible a todos los usuarios.	10.1
Condicionantes adicionales	10	Disponibilidad SW. Se verifica y monitoriza el ancho de banda y el acceso al servidor durante el tiempo de duración del proyecto	5
	11	Calidad SW. Se verifica la calidad del código entregado mediante herramientas de métrica estándar	11

12.1 Evaluación del nivel de calidad

Este apartado estará disponible al finalizar la codificación de SCP.

13. Anexo gráfico

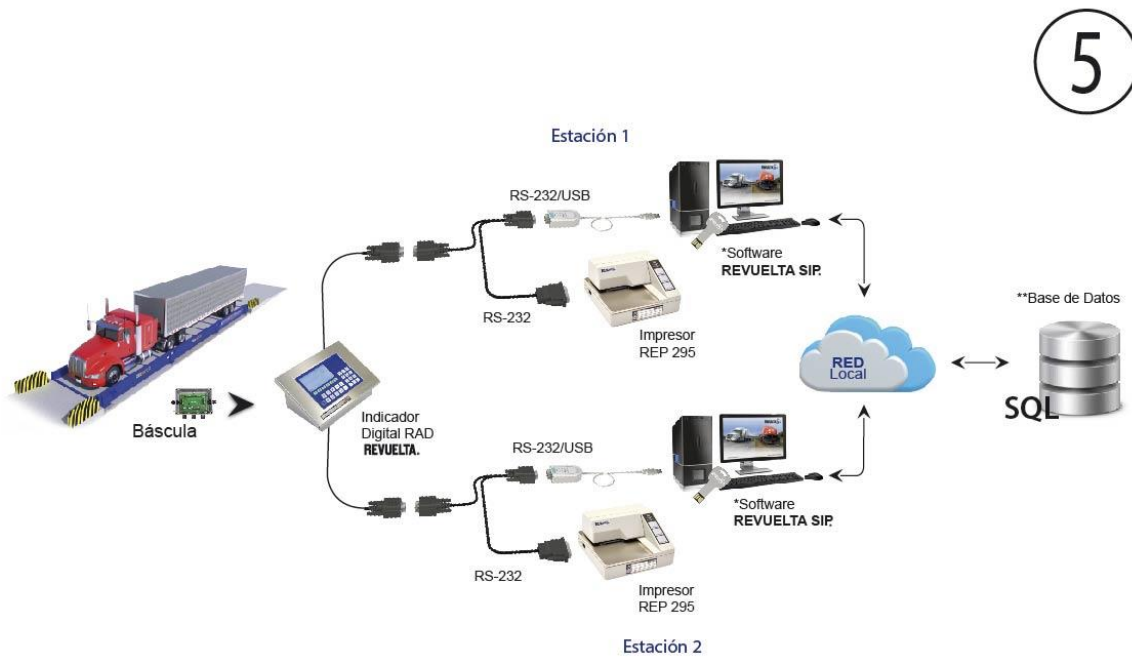


Ilustración 8. SCP lógica de conexión multi estación (los equipos pueden variar se tomó a medida de ejemplo)

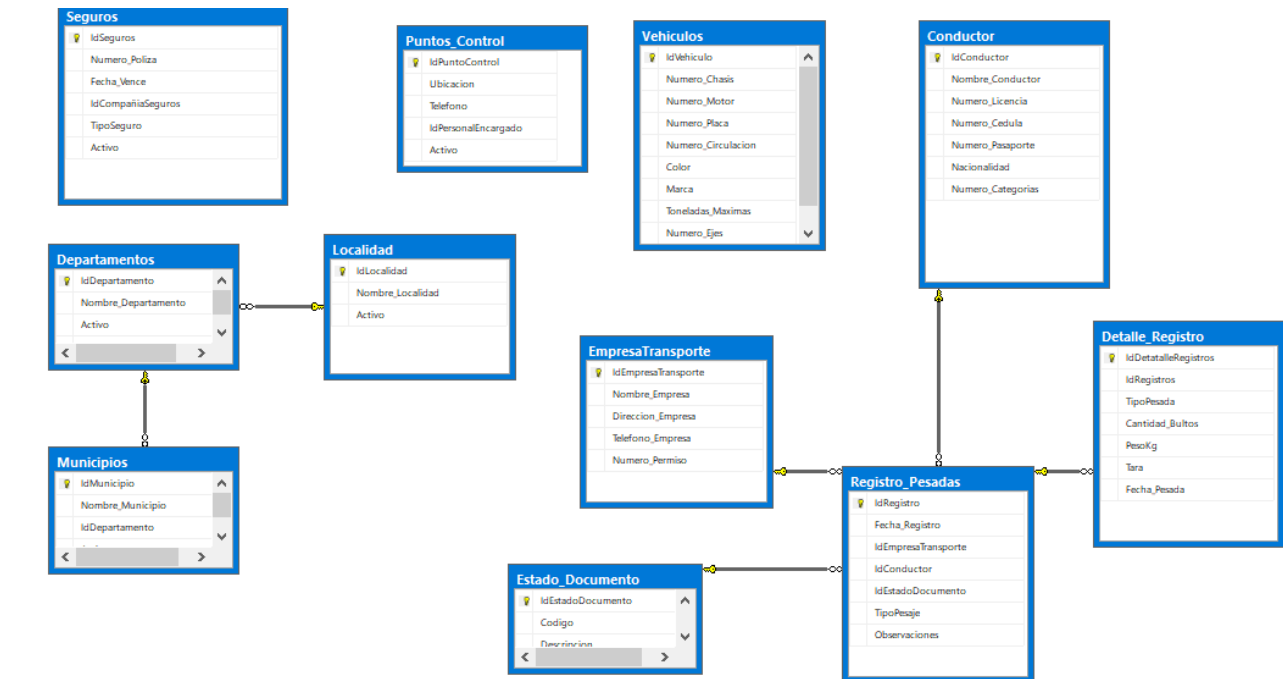


Ilustración 9. SCP. Modelo de datos

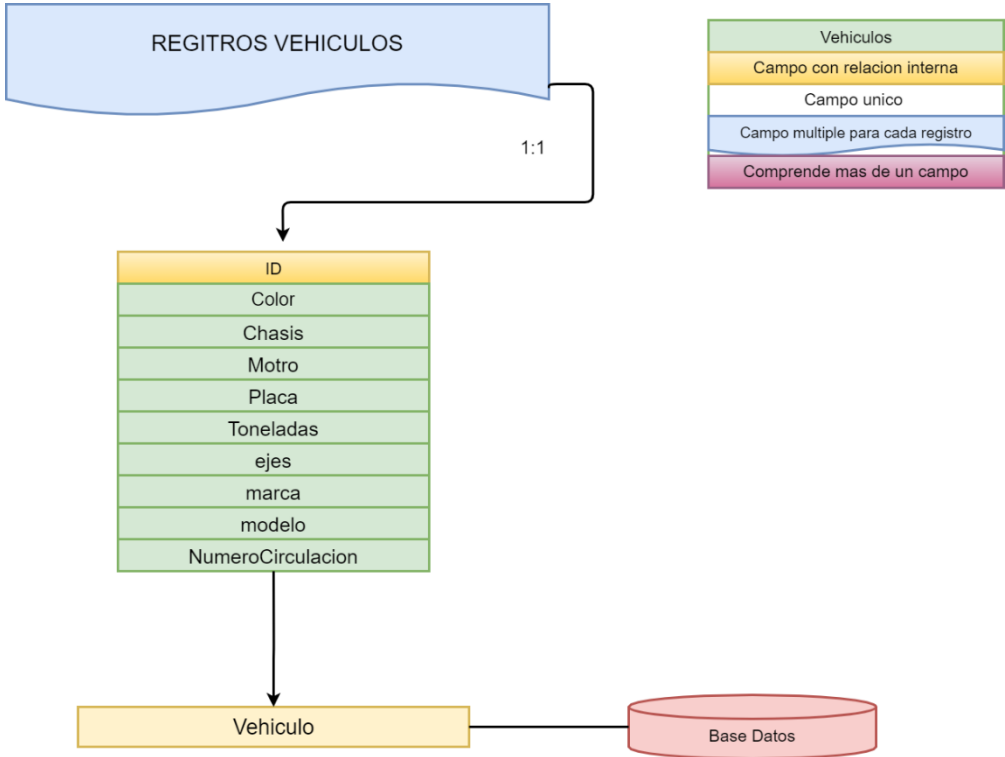


Ilustración 10. Registro Vehículo. Relaciones

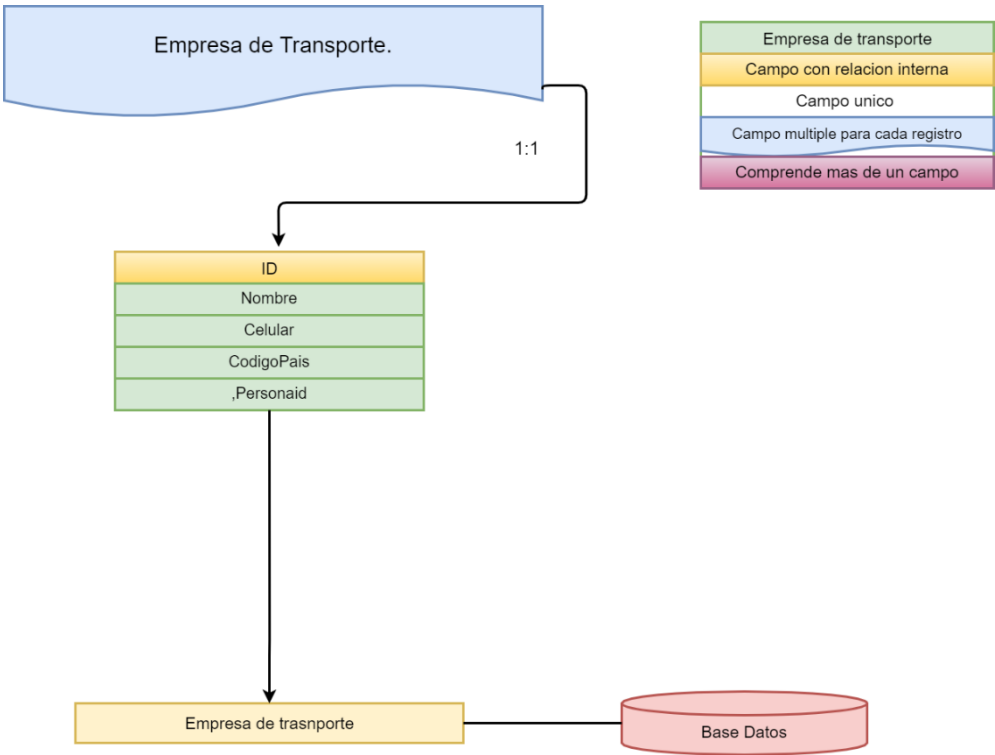


Ilustración 11. Registro Empresa de Transporte. Relaciones.

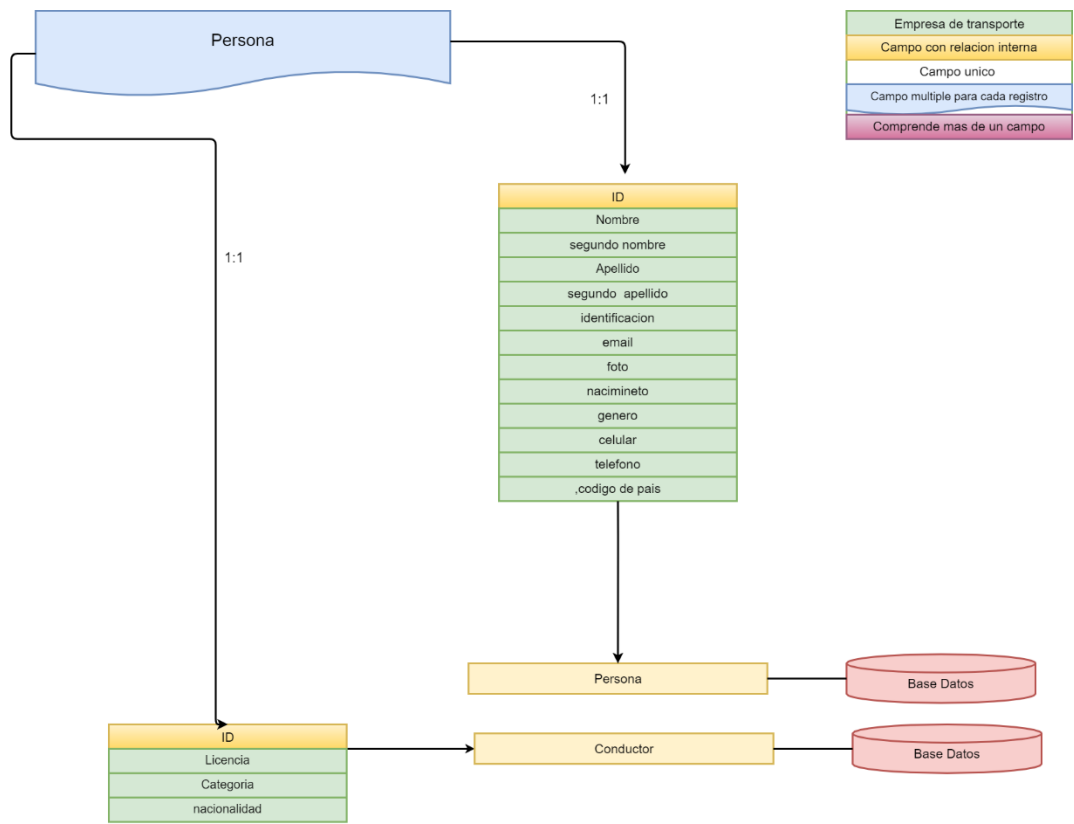


Ilustración 12. Registro PERSONA. Relaciones

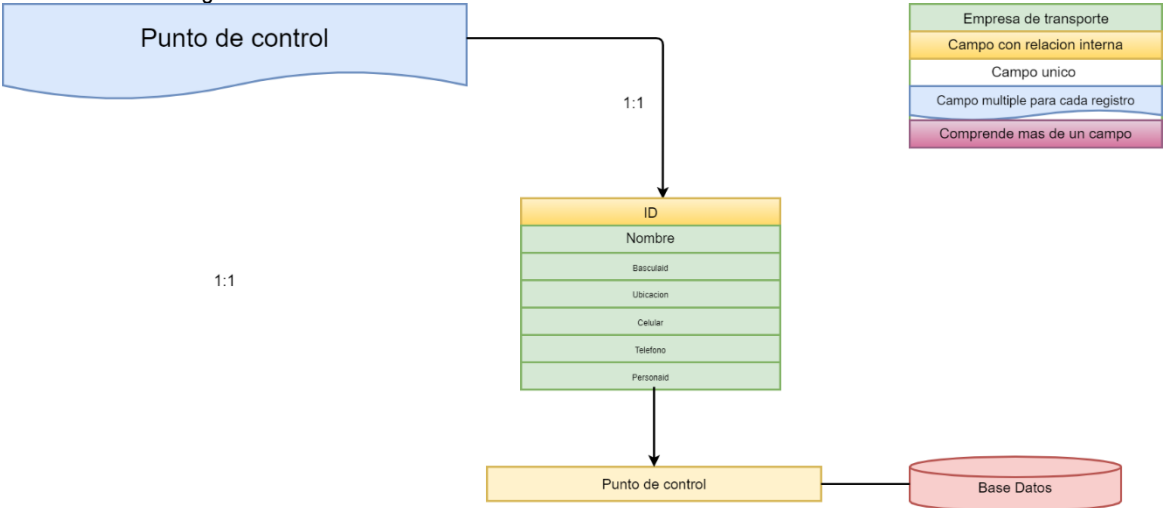


Ilustración 13. Registro punto de control . Relaciones

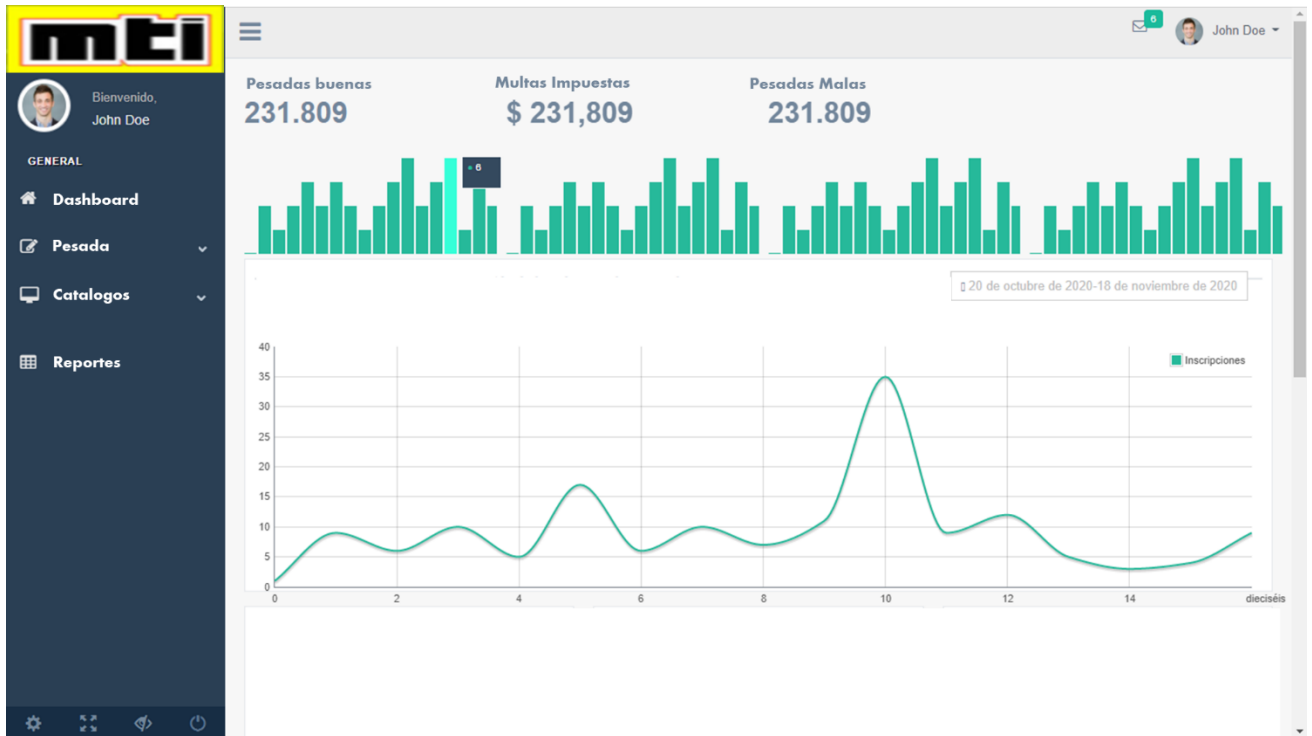


Ilustración 14. Interfaz Front-end Dashboard

The screenshot displays the MTI Front-end Catalogs interface. The sidebar on the left includes a user profile for John Doe and a menu with options: Dashboard, Catalogos, Persona, Vehiculos, Conductor, Empresa Transporte, Punto de control, Usuario, Pesada, and Reportes. The main content area is titled 'Elementos de formulario' and contains a form for creating or editing a catalog item. The form includes fields for 'Nombre *', 'Apellido *', 'segundo nombre / Inicial', 'Género' (with 'Masculino' and 'Hembra' buttons), 'Fecha de nacimiento *' (with a date picker), 'Correo electrónico', and a 'Seleccione' dropdown menu. At the bottom right, there are three buttons: 'Cancelar', 'Reiniciar', and 'Enviar'.

Ilustración 15. Interfaz front-end Catalogos

Pesada standard

Ticket No: 00001

12/12/2020 08:00:00 PM

Conectado

Peso Bruto
2500 KG
Comprobado Desenfrenado

Peso Correcto !!!

Placa:

Empresa:

Conductor:

Producto:

Costo Unit:

Costo Neto:

Bascula:

Bruto:

Tara:

Neto:

Observaciones:

Cancelar Reiniciar Enviar Ticket Eliminar

Acciones masivas (2 registros seleccionados)

<input checked="" type="checkbox"/>	121000040	23 de mayo de 2014 11:47:56 PM	121000210 ↑	John Blank L	Pagado	\$ 7.45	Ver
<input checked="" type="checkbox"/>	121000039	23 de mayo de 2014 11:30:12 PM	121000208 ↑	John Blank L	Pagado	\$ 741.20	Ver
<input type="checkbox"/>	121000038	24 de mayo de 2014 10:55:33 PM	121000203 ↑	Mike Smith	Pagado	\$ 432.26	Ver
<input type="checkbox"/>	121000037	24 de mayo de 2014 10:52:44 PM	121000204	Mike Smith	Pagado	\$ 333.21	Ver

Ilustración 16. Interfaz front-end Pesadas

Reporteria SCP

Buscar... ¡Vamos!


MTI Exportar PDF


Reportes

Estos son distintos reportes que podrán Ser exportados a distintos formatos dentro el SCP.

glyphicon-glyphicon-asterisco	glyphicon-glyphicon-plus	glyphicon-glyphicon-euro	glyphicon-glyphicon-eur	glyphicon-glyphicon-menos	glifo glifo-nube	glyphicon-glyphicon-sobre	glyphicon-glyphicon-lápiz
glyphicon-glyphicon-glass	glyphicon-glyphicon-música	glyphicon-glyphicon-search	glyphicon-glyphicon-corazón	glyphicon-glyphicon-star	glyphicon-glyphicon-star-empty	glyphicon-glyphicon-usuario	glyphicon-glyphicon-film
glyphicon	glyphicon	glyphicon	glyphicon	glyphicon	glyphicon	glyphicon	glyphicon

Ilustración 17. Interfaz front-end Reportes





Bienvenido,
John Doe

GENERAL

Dashboard
 Pesada
 Catalogos
 Reportes

DataTables tiene la mayoría de las funciones habilitadas de forma predeterminada, por lo que todo lo que necesita hacer para usarlo con sus propias tablas es llamar a la función de construcción: `$.dataTable();`

mostrar 10 entradas
 Buscar:

	Nombre	Posición	Oficina	Años	Fecha de inicio	Salario
<input type="checkbox"/>	Tiger Nixon	sistema arquitecto	Edimburgo	61	2011/04/25	\$ 320,800
<input type="checkbox"/>	Garrett Winters	Contador	Tokio	63	2011/07/25	\$ 170,750
<input type="checkbox"/>	Ashton Cox	Autor técnico junior	San Francisco	66	2009/01/12	\$ 86 000
<input type="checkbox"/>	Cedric Kelly	Desarrollador Senior de Javascript	Edimburgo	22	2012/03/29	\$ 433,060
<input type="checkbox"/>	Airi Satou	Contador	Tokio	33	2008/11/28	\$ 162,700
<input type="checkbox"/>	Brielle Williamson	Especialista en integración	Nueva York	61	2012/12/02	\$ 372,000
<input type="checkbox"/>	Herrod Chandler	Asistente de ventas	San Francisco	59	2012/08/06	\$ 137,500
<input type="checkbox"/>	Rhona Davidson	Especialista en integración	Tokio	55	2010/10/14	\$ 327,900
<input type="checkbox"/>	Colleen Hurst	Desarrollador Javascript	San Francisco	39	15/09/2009	\$ 205,500
<input type="checkbox"/>	Sonya Frost	Ingeniero de software	Edimburgo	23	2008/12/13	\$ 103,600

Mostrando 1 a 10 de 57 registros
 Anterior 1 2 3 4 5 6 próximo

Impresión
 Generar PDF

Ilustración 18. Interfaz front-end Detalle de reportes

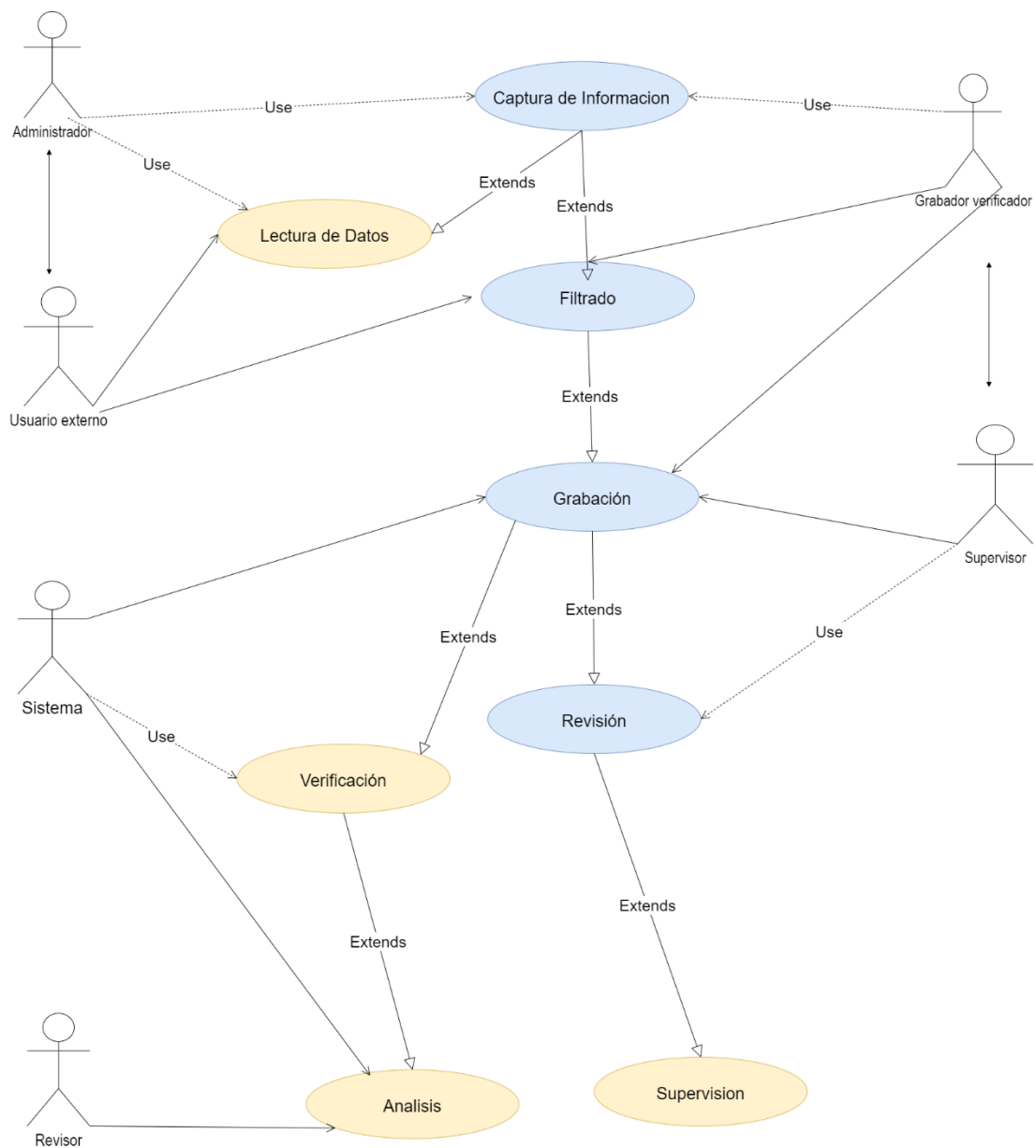


Ilustración 19. Diagrama UML genérico. Modelo de casos de uso

13.1 Normativa aplicada

La confección del artefacto PDSDT ha seguido la normativa sugerida y especificada en ISO/IEC 12207.

13.2 Bibliografía

A. Piscitelli. "Humanidades digitales y nuevo normal educativo". *TELOS (Cuadernos de Comunicación e Innovación)*, pp. 10 y ss., junio-septiembre 2015.

IEEE. *IEEE Standard Glossary of Software Engineering Terminology*, IEEE std 610.12-1990, 1990.

INTECO. *Ingeniería de Software: metodologías y ciclos de vida. Laboratorio Nacional de Calidad del Software*. España: Instituto Nacional de Tecnologías de la Comunicación. 2009.

ISO/IEC. *ISO/IEC 12207:2008 Systems and software engineering – Software life cycle processes*. EEUU: 2008.

J. Palacio. *Gestión de proyectos Scrum Manager (Scrum Manager I y II). Versión 2.5.1*. España, 2015.

M. Cusmano y M. Selby. *Microsoft Secrets*. EEUU: HarperCollins Business, 1997.

M. Poppendieck y T. Poppendieck. *Lean Development Software: An Agile Toolkit for Software Development Managers (The Agile Software Development Series)*. EEUU: Addison-Wesley, 2003.

PMI. *A Guide to the Project Management Body of Knowledge*. EEUU, 2012.