
Proyecto EXHIBITIUM

Sistema de gestión de datos ExpoFinder
Plan de Desarrollo Software y Documentación Técnica

Versión 1.2

Hoja de control del documento

Título	Plan de desarrollo de software y documentación técnica. Sistema de gestión de datos ExpoFinder		
Descripción	ExpoFinder es una aplicación en red desarrollada para facilitar el la creación y el almacenamiento de registros de exposiciones artísticas temporales de forma que sea posible a posteriori su explotación.		
Autor	Antonio Cruces Rodríguez		
Responsable	Nuria Rodríguez Ortega		
Organización	Universidad de Málaga. Departamento de Historia del Arte. iArtHis_LAB Research Group		
Categoría	Documentación técnica		
Palabras clave	exposiciones temporales, software, FBBVA		
Versión	1.2	Fecha	16 de enero de 2016
Estadísticas	8766729 bytes, 137782 caracteres, 92 páginas, 2761 párrafos, 24445 palabras		
Edición	26 revisiones		

Historial de revisiones

Fecha	Versión	Descripción	Autor
2015-08-15	1.0	Versión inicial	Antonio Cruces Rodríguez
2016-01-12	1.1	Versión ajustada. Transición, 3ª iteración	Antonio Cruces Rodríguez
2017-01-15	1.2	Versión revisada. Modificaciones mayores	Antonio Cruces Rodríguez

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.....	15
2.3.3	Programación	15
2.3.4	Prueba	15
2.3.5	Mantenimiento	15
2.4	Ciclo de vida	15
2.5	Normativa y protocolos aplicables al ciclo de vida	17
2.6	Metodología.....	18
2.7	Entregables del proyecto.....	23
3	Organización del proyecto	25
3.1	Recursos humanos	25
3.2	Jerarquía RRHH.....	26
4	Gestión de procesos en el proyecto	29
4.1	Estimación de esfuerzo	29
4.2	Plan del proyecto.....	30
4.2.1	Plan de fases y temporalización	30
4.2.2	Actividades y disciplinas.....	31
4.2.3	Matriz de roles y responsabilidades	32
4.3	Mecanismos de seguimiento y control	32
4.3.1	Gestión de requisitos.....	32
4.3.2	Control de plazos.....	33
4.3.3	Control de calidad.....	33
4.3.4	Gestión de riesgos.....	33
4.3.5	Gestión de configuración	33
5	Infraestructura.....	34
6	Estructura física de la información	36
7	Estructura lógica de la información	40
7.1	Elementos	40
7.2	Relaciones entre elementos.....	40
7.3	Taxonomías.....	41
7.4	Metadatos.....	42

8	Diseño de la arquitectura del sistema.....	46
8.1	Descripción general.....	46
8.2	Ingeniería de requisitos	50
8.2.1	Requisitos funcionales.....	51
8.2.2	Requisitos organizacionales.....	52
8.2.3	Requisitos no funcionales.....	52
8.3	Implementación de la solución	52
9	Resumen de procesos del sistema	55
9.1	Actores y casos de uso. Modelos básicos	55
9.1.1	Actores	55
9.1.2	Casos de uso.....	56
9.2	Procesos	59
9.2.1	Procesos principales	59
9.2.2	Procesos auxiliares	60
9.2.3	Procesos del <i>framework</i>	60
10	Interfaces del sistema	61
10.1	Back-end	61
10.2	Front-end.....	62
11	Componentes SW	64
11.1	Componentes básicos.....	64
11.2	Estructura SW	64
11.3	Algoritmos y librerías básicas.....	66
11.3.1	Algoritmos.....	66
11.3.2	Librerías.....	67
12	Gestión de riesgos	69
13	Verificación de la calidad	71
13.1	Evaluación del nivel de calidad	72
13.1.1	Herramientas de medición de calidad	72
13.1.2	Matriz de control de calidad.....	73
13.1.3	Informe de errores	74
13.1.4	Evaluación	74
14	Anexo gráfico.....	75
15	Anexo documental.....	89
15.1	Informe sobre cobertura geográfica	89
15.2	Informe sobre LOC y complejidad SW	89
15.3	Normativa aplicada.....	90
15.4	Bibliografía	90

Tabla de ilustraciones

Ilustración 1. Jerarquía metodológica en EF.....	19
Ilustración 2. Ciclo SCRUM.....	21
Ilustración 3. Desarrollo secuencial y solapado.....	22
Ilustración 4. Disciplinas y fases RUP en EF	23
Ilustración 5. Interacciones entre actores en EF	28
Ilustración 6. Modelo simplificado SI EF.....	47
Ilustración 7. Comparativa de test de velocidad MariaDB y MySQL	50
Ilustración 8. Procesos GC según PMI	71
Ilustración 9. EF como subproyecto de EX. Mapa conceptual	75
Ilustración 10. EF: Esquema de flujo simplificado. Mapa conceptual.....	76
Ilustración 11. EF. Esquema procesual. Mapa conceptual	77
Ilustración 12. Stack de WordPress	78
Ilustración 13. EF. Modelo de datos	79
Ilustración 14. Relaciones entre los 3 sistemas principales de EF	80
Ilustración 15. Registro ENTIDADES. Relaciones	81
Ilustración 16. Registro PERSON. Relaciones	82
Ilustración 17. Registro PUBLICACIONES. Relaciones	82
Ilustración 18. Registro EMPRESA. Relaciones	83
Ilustración 19. Registro EXPOSICIÓN. Relaciones.....	84
Ilustración 20. Interfaz back-end	85
Ilustración 21. Interfaz front-end.....	86
Ilustración 22. Diagrama UML genérico. Modelo de casos de uso	87
Ilustración 23. Servidor admin.expofinder.es. Tráfico últimos 12 meses	87
Ilustración 24. Servidor admin.expofinder.es. Uso CPU últimos 12 meses	88

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 ExpoFinder (en adelante EF) 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 EF, 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 EF, 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 EF 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 EF.
- 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 EF 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 EF*: como herramienta de seguimiento y para control del proceso de desarrollo.
- *Desarrolladores*: como guía de actividades, calendario de fases y ejecución.

- **Audidores:** para verificar el ajuste de lo proyectado a la realidad de la implementación.

1.3 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 proyecto EX, matriz y razón de ser de EF. 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 EF (dependiente del desarrollo de EX), la captura de información para ingeniería de requisitos no se ha realizado, como suele ser habitual, a través de un *stakeholder* representante del comitente (en adelante CL) 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 EX. 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 han 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.

1.4 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 EX, 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 EX.
- **PDS. Gestión de procesos del proyecto.** Explica los costos temporales y la planificación estimada, define las fases e hitos de EX 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 EF, 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.

1.5 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.

La aplicación de exportación de datos NO FIGURA INCLUIDA en PDSDT por tratarse de una aplicación externa. En realidad, EF simplemente responde a una solicitud REST válida durante el proceso de exportación, siendo responsabilidad de de actores externos, pertenecientes a EX, la creación de un aplicativo para elaborar las solicitudes. Por deferencia, el equipo de desarrollo ha generado uno para ponerlo a disposición de los miembros de EX, pero no se recoge en el presente artefacto por no tratarse de una parte integrante del proyecto EF.

1.6 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 SW, error descubierto en un programa
CBI	Clasificador Bayesiano Ingenuo
CDU	Caso de uso
CL	Actor que realiza el papel de comitente 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
DAH	<i>Digital Art History</i>
DH	<i>Digital Humanities</i>
DHA	Departamento de Historia del Arte
DRAE	Diccionario de la Real Academia Española
DT	Documentación Técnica
EF	Proyecto ExpoFinder
ETA	Exposiciones temporales de contenido y/o carácter artístico
EX	Proyecto Exhibitium
FBBVA	Fundación Banco Bilbao Vizcaya Argentaria
FW	<i>Framework</i> . Terminología de arquitectura de desarrollo
GC	Gestión de calidad

HW	<i>Hardware</i>
IEC	<i>International Electrotechnical Commission</i>
IEEE	Institute of Electrical and Electronics Engineers
INTECO	Instituto Nacional de Tecnologías de la Comunicación
ISO	<i>International Organization for Standardization</i>
KISS	<i>Keep It Simple, Stupid</i> . Lema de trabajo que aboga por la implantación de la “navaja de Ockham” 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
OTRI	Oficina de Transferencia de Resultados de Investigación
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 “artefacto”, 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
Skeumórfico	Interfaz de usuario basado en la simulación de objetos realistas (sobreados, 3D,...)
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>
UMA	Universidad de Málaga
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 sincronizan, cómo fluye la información que soporta las tareas y cómo se le hace seguimiento. Terminología propia de RUP
XP	<i>eXtreme Programming</i> . Acrónimo de un tipo de metodología ágil

2 Vista general del proyecto

2.1 Propósito, alcance y objetivos

El propósito del proyecto de desarrollo de *software* ExpoFinder, una iniciativa del Grupo de Investigación iArHis_LAB, del Departamento de Historia del Arte de la Universidad de Málaga (España), es posibilitar la realización de Exhibitium, un proyecto de investigación que obtuvo una ayuda económica en el concurso convocado por la Fundación BBVA en 2014 (entre más de doscientas cincuenta propuestas en la categoría de Humanidades Digitales) con el objetivo definido de **generar conocimiento sobre exposiciones artísticas temporales para su reutilización y aprovechamiento multivalente**.

El proyecto EX forma parte de las líneas de trabajo emprendidas por iArHis_LAB, cuyo objetivo específico es la promoción, desarrollo y consolidación en España de los estudios sobre Historia del Arte Digital (*Digital Art History*), los cuales se inscriben, como vertiente específica y diferenciada, en el amplio campo que actualmente constituyen las DH (Baca, Helmreich and Rodríguez Ortega, 2013). Asimismo, este proyecto forma parte de las iniciativas promovidas por la Red Internacional de Estudios Digitales sobre la Cultura Artística, ReArte.Dix, la primera organización internacional dedicada a este fin en el contexto hispanohablante, financiada por la Universidad de Málaga como parte de su plan propio de investigación y como parte de los proyectos asociados al Campus de Excelencia Internacional Andalucía Tech.

El proyecto EX exige que se realicen las acciones oportunas para que un sistema lo más automatizado posible capture datos libremente disponibles en la Red sobre el objeto a analizar: exposiciones temporales de contenido y/o carácter artístico (en adelante, ETA). Los datos deben poder, a posteriori, ser susceptibles de recibir un análisis cuantitativo y/o cualitativo que permita extraer conclusiones científicamente válidas y, por tanto, generar nuevo conocimiento, en línea por lo especificado por Piscitelli en Telos 2015¹.

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

- *Captura automatizada de información.*
 - Localización de fuentes de información primarias a partir de las cuales se obtendrán los datos sobre ETA.
 - 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.
 - 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 EX.

¹ A. Piscitelli. "Humanidades digitales y nuevo normal educativo". *TELOS (Cuadernos de Comunicación e Innovación)*, pp. 10 y ss., junio-septiembre 2015.

- Completado de la información considerada esencial en cada ETA 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 EF y pertenecientes a EX, 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 EF y paralela a cualquier otra implementación que al respecto puedan llevar a cabo los oportunos actores de EX.

2.2 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 de CLI y su disponibilidad económica y temporal, la propia existencia de unos plazos de término para la entrega de la documentación y la presentación de resultados del proyecto EX condiciona el desarrollo de EF, así como el presupuesto derivado de la dotación económica obtenida con la ayuda citada en 2.1.

Este apartado 2.2 podrá ser modificado en sucesivas versiones de PDSDT de forma que se adapte a las condiciones evolutivas de EX y del propio aplicativo derivado de EF.

2.2.1 Restricciones

Derivan de las condiciones de adjudicación de la ayuda y de los recursos humanos, temporales y materiales disponibles por iArtHis_LAB Research Group.

- El proyecto EX debe estar finalizado y disponible para su auditoría por FBBVA el 31 de diciembre de 2016.
- El proyecto EX especifica que, previamente a la finalización, un grupo de actores deberá realizar el proceso de análisis de la información custodiada por EF.
- Por las condiciones de iArtHis_LAB Research Group como entidad académica existe acceso a un grupo humano con la adecuada formación para acometer la tarea de actores necesarios en EF con perfiles muy específicos, pero no existe un excesivo margen de maniobra para retribuir la colaboración de dichos actores, de manera que EF 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 organización académica dependiente de una institución como el DHA-UMA y ser EX un proyecto ajeno a la matriz institucional, EF debe utilizar recursos HW y/o SW cuyo uso no afecte al DHA.
- Por necesidad de cumplir adecuadamente los plazos estipulados en EX, EF debe estar a pleno rendimiento y completamente funcional para permitir la explotación de los datos coleccionados con antelación a enero de 2016.
- Para que la validez de los estudios cuantitativos y/o cualitativos derivados del uso de los datos coleccionados transferidos desde EF a los actores oportunos de EX sea efectiva, la información recopilada deberá abarcar de la forma más exhaustiva posible un conjunto geográfico 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

- EF 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 EF (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.
- Ítem más sobre lo expresado en b), EF 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.
- Teniendo también en cuenta las características del objeto de EX, tal y como se ha mencionado en 2.1, 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 DH y DAH como en el manejo de herramientas ofimáticas simples, no siendo posible para el plan de EF contar con una fase de formación genérica de sus actores, si bien sí la hay, en formato cíclico, para la específica.

2.3 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, FORTRAN, COBOL, C... Los programas también pueden ser generados usando generadores de programas.
- **Datos.** Elementos significativos portadores de la información que se pretende tratar me-

dian­te 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 EF 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 para 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

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.

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

2.3.5 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 EF* (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

2.4 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 planificables. 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 (re o 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 esta-

blecidos.

El modelo de ciclo de vida adoptado en el desarrollo de EF es el de **prototipos**, propio de la metodología ágil utilizada para todo el proceso completo (ver 2.6). Según los principios de este modelo, CL 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 CL 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 CL. El diseño rápido lleva a la construcción de un prototipo. El prototipo lo evalúa CL 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 CL, 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 cliente 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 cliente.
- El prototipo es un documento vivo de buen funcionamiento del producto final. El cliente 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 EF, el equipo de desarrollo ha concluido de 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 EF, puesto que tanto la inversión está predeterminada desde el primer momento (consúltese al respecto la web de EX, <http://www.exhibitium.com>) como la relación con CL es absolutamente directa, sin intermediarios.
- Con este modelo pueden surgir problemas con CL, 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 CL. No es el caso de EF, por tratarse de un comitente que, por así decirlo, es un eslabón

más del equipo de diseño.

2.5 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 EF, 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 carece de sentido en EF por no tratarse de una "venta" de software, sino de una implementación de una solución para otro proyecto, en este caso EX.

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 EF.

- **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, 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 váli-

- do (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, taxonomías,...).
- *Auditoría.* Realizada por actores externos a EF, preferiblemente por aquellos con dicha tarea determinada desde EX.
- *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 la institución matriz y con FBBVA a efectos administrativos. Se subdivide en tareas delegables.
 - *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 EF.
 - *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.

2.6 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 EF, el equipo de desarrollo optó por las denominadas “ágiles”.

Entre las tradicionales, desechadas para usarlas de base del trabajo en EF 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 promociona *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 "contínuos") 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

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

Esta metodología es muy conveniente para EF por cuanto, como se verá en 3, existe un grupo de usuarios finales, que pueden considerarse parte del CL, 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 CL. Es una evolución del Método Toyota de Producción aplicado al desarrollo

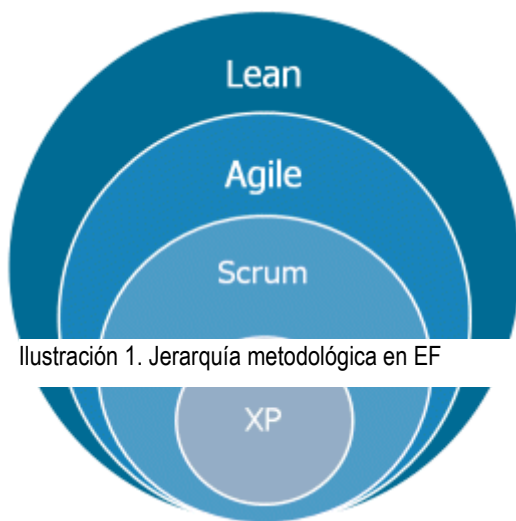


Ilustración 1. Jerarquía metodológica en EF

llo y que está muy de moda entre 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 EF 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 EF.
- Elevar la integridad conceptual a la categoría de rumbo principal del proyecto. Este término se emplea en EF 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 dependen exclusivamente de su escrupuloso cumplimiento de las normas de revisión y grabación
- Ver siempre EF 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, EF 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.

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 se basa en la adopción de **roles** por parte de quienes intervienen en el proceso de desarrollo (tanto de la parte CL 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.

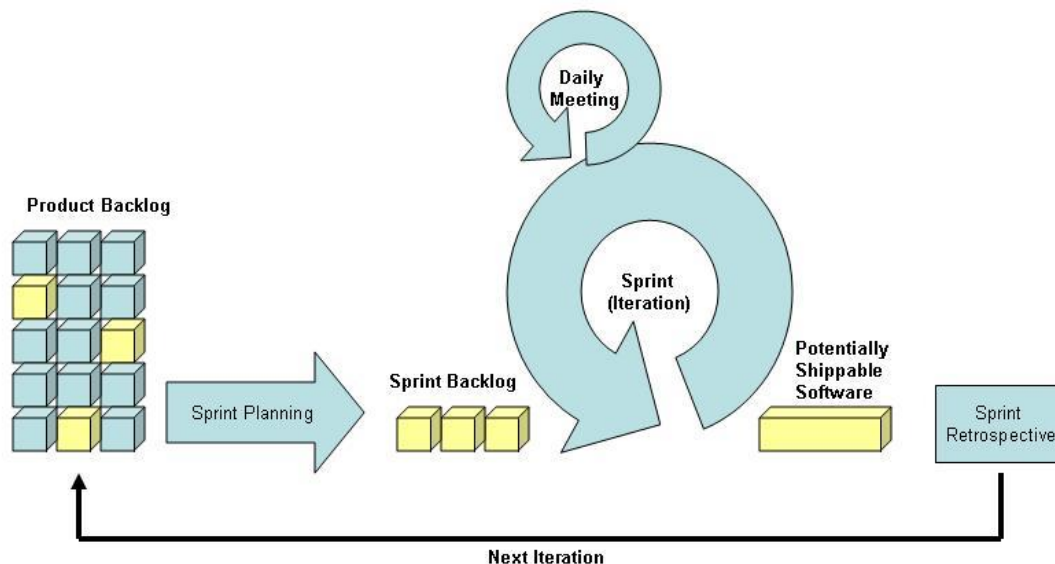


Ilustración 2. Ciclo SCRUM

SCRUM implica, por parte del equipo de desarrollo, la asunción de unos principios, a los que EF, por las circunstancias concretas de su generación (es un proyecto de apoyo a otro proyecto y carece de interés comercial en su finalidad actual, ostentando, sin embargo, una marcada vocación académica y científica), 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 de EX es “territorio inexplorado”, y es importante no dar por sentados los planteamientos iniciales. EF 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 CL 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, EF 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**.

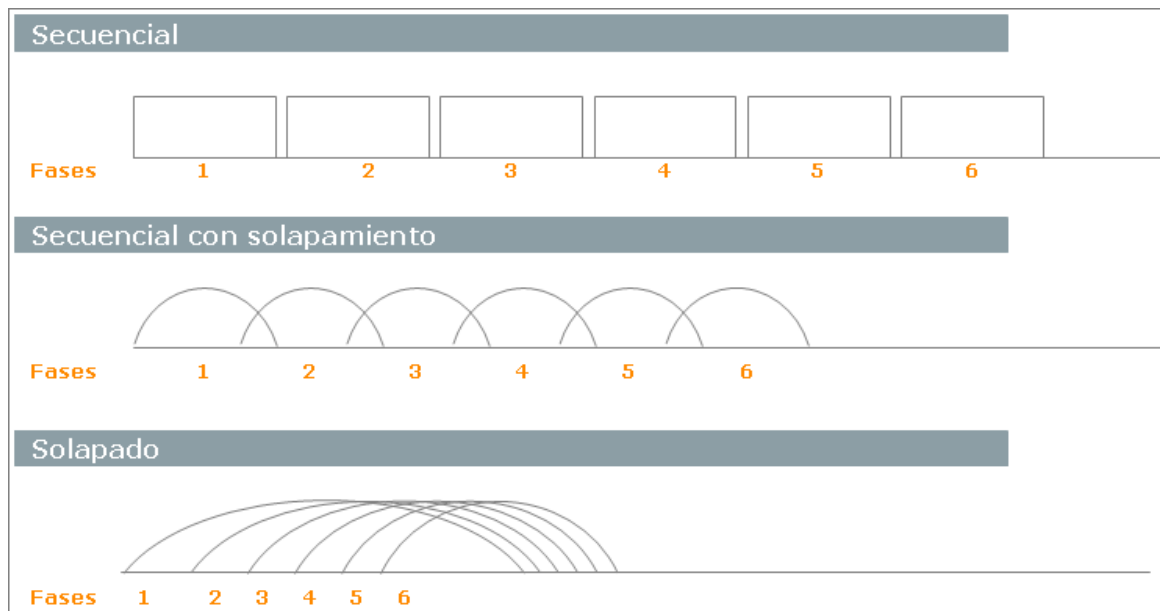


Ilustración 3. Desarrollo secuencial y solapado

EF 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 EF 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). En el caso de EF, coincide la figura de supervisión con la homóloga de EX, lo que resulta muy adecuado
Team	Desarrolladores	Realizan los artefactos determinados en las reuniones

Las reuniones en SCRUM siguen una pauta poco flexible, y por ello EF 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

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

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 EF 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 CL, a las especificidades del trabajo de los diferentes actores de EX -que también son cambiantes según se lleve a cabo el desempeño de los roles concretos- y a los límites temporales y presupuestarios.

2.7 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 EF.

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.

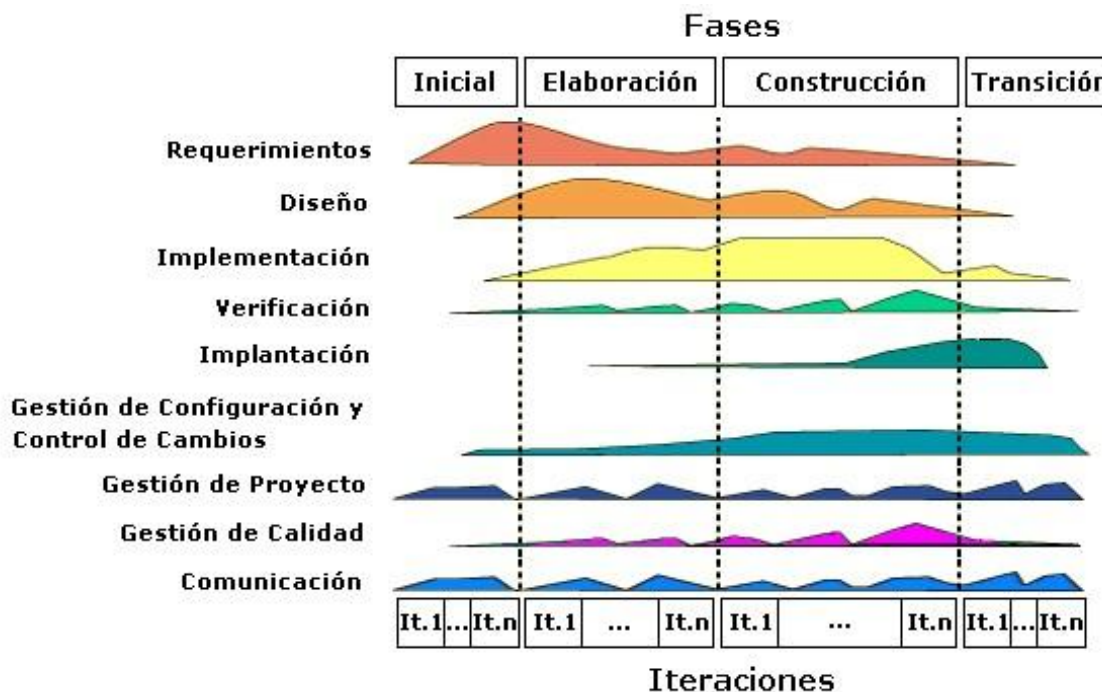


Ilustración 4. Disciplinas y fases RUP en EF

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 EF, 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 CL 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 EF. Los entregables se agrupan tal como figura en la siguiente tabla.

DISCIPLINA	ENTREGABLE	PREVISIÓN	OBSERVACIONES
Gestión del proyecto	PDS	Enero 2016	Incluido en PDSDT. En 3
Diseño (Modelado de negocio)	Casos de uso agrupados	Febrero 2015	Incluido en PDSDT. En 4
Diseño (Modelado de negocio)	Modelo de datos	Marzo 2015	Parcialmente incluido en PDSDT (modelos físico y lógico). En 6 y 7
Requerimientos (Requisitos)	Restricciones y suposiciones	Enero 2015	Incluido en PDSDT. En 2
Requerimientos (Requisitos)	Matriz de requisitos	Enero 2015	Incluido en PDSDT
Implementación	Modelo de implementación	Abril 2015	Incluido en PDSDT. En 8
Verificación	Modelo de control de errores	Mayo 2015	Incluido en PDSDT. En 10
Implantación	Sistema	Mayo 2015	Instalado en servidor
Implantación	Modelos de interfaces	Mayo 2015	Incluido en PDSDT. En 10
Implantación	Manuales de usuario	Septiembre 2015	No incluidos en PDSDT
Gestión de calidad	Matriz de PMBOK	Septiembre 2015	Incluido en PDSDT. En 13
Comunicación	Publicación de artículos y organización de <i>workshop</i>	Octubre 2016	No es competencia de EF

Nótese que no todas las disciplinas implican la existencia de entregables y que, por la propia naturaleza del proyecto EF, algunos dependen de la evolución de EX.

3 Organización del proyecto

3.1 Recursos humanos

La estructura organizativa de EF depende por entero de los recursos asignados desde EX. No en vano EF es un subproyecto del mismo. Por tanto, cabe consultar la pertinente documentación al respecto de los detalles sobre actores “externos” que intervienen en el procesado analítico de los datos consolidados en EF. El perfil del personal asignado se agrupa en **equipos** de la siguiente manera.

Supervisión

Nuria Rodríguez Ortega. NRO. Cargo: Investigadora principal. Tareas: Coordinación y dirección. Especialidades: Historia del Arte Digital, Humanidades Digitales, Análisis relacional.

María Luisa Bellido Gant. MLBG. Cargo: Investigadora principal. Tareas: Coordinación y dirección. Especialidades: Historia del Arte Digital, Humanidades Digitales, Datos, Coordinación.

Desarrollo y administración del sistema

Antonio Cruces Rodríguez. ACR. Cargo: Investigador. Tareas: Desarrollo de aplicaciones. Especialidades: Historia del Arte Digital, Humanidades Digitales, Informática, Telecomunicaciones, Desarrollo.

Revisión

Ana Carmen Benítez Hidalgo. ACBH. Cargo: Investigadora colaboradora. Tareas: Datos. Especialidades: Historia del Arte Digital, Humanidades Digitales, Datos.

Carmen Tenor Polo. CTP. Cargo: Investigadora colaboradora. Becaria. Tareas: Coordinación de equipo. Datos. Especialidades: Historia del Arte Digital, Humanidades Digitales, Datos, Coordinación.

Bárbara Romero Ferrón. BRF. Cargo: Alumna en prácticas. Tareas: Datos. Especialidades: Historia del Arte.

Grabación y verificación

María Casas González. MCG. Cargo: Investigadora colaboradora. Becaria. Tareas: Datos. Especialidades: Historia del Arte Digital, Humanidades Digitales, Datos.

Miguel Ángel Sánchez Badillo. MASB. Cargo: Investigador colaborador. Becario. Tareas: Datos. Especialidades: Historia del Arte Digital, Humanidades Digitales, Datos.

Carmen Molina Sánchez. CMS. Cargo: Investigadora colaboradora. Becaria. Tareas: Datos. Especialidades: Historia del Arte Digital, Humanidades Digitales, Datos.

Beatriz Parejo Ramos. BPR. Cargo: Alumna en prácticas. Tareas: Datos. Especialidades: Historia del Arte Digital, Humanidades Digitales, Datos.

David González Medina. DGM. Cargo: Alumno en prácticas. Tareas: Datos. Especialidades: Historia del Arte Digital, Humanidades Digitales, Datos.

Dadas las específicas condiciones de EF, 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	Profesorado de universidad.	Historia del Arte. DH. DAH.
Desarrollo	Profesorado de universidad.	Historia del Arte. DH. DAH. SW y/o Telecomunicaciones.
Administración del sistema	Profesorado de universidad.	SW y/o Telecomunicaciones
Revisión	Egresados/as de universidad.	Historia del Arte. DH. DAH. Nivel adecuado de ofimática.
Grabación	Egresados de universidad. Alumnado de universidad (últimos niveles).	Historia del Arte. DH. DAH. Nivel adecuado de ofimática.
Verificación	Egresados de universidad. Alumnado de universidad (últimos niveles).	Historia del Arte. DH. DAH. Nivel adecuado de ofimática.

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, por las condiciones específicas de EX, los perfiles profesoriales universitarios carecen de remuneración dineraria por su participación en EF y pueden solamente establecer una dedicación parcial, mientras que los restantes perfiles sí acceden a retribuciones mediante contratos especiales con la OTRI de UMA o mediante procedimientos arbitrados como becas de colaboración. Ello redundará, evidentemente, en beneficio de la intensidad de la dedicación de dichos perfiles, no obstante lo cual las limitaciones presupuestarias derivadas de CL (en este caso, el propio EX) no permiten ampliar el horario de trabajo. A los efectos organizativos dentro de EF, la **matriz de atribución de roles, actores, equipos y dedicación** puede consultarse en la siguiente tabla.

EQUIPO	ACTOR	ROL	DEDICACIÓN ⁷	REMUNERACIÓN	ASIGNACIÓN
Supervisión	Supervisor	Product Owner	Parcial	No	NRO MLBG
Desarrollo	Desarrollador	Scrum Manager. Team Manager	Parcial	No	ACR
Administración del sistema	Administrador	Team Manager	Parcial	No	ACR
Revisión	Revisor	Team	20	Sí	ACBH CTP BRF.
Grabación	Grabador verificador	Team	20	Sí	ACBH CTP BRF MCG MASB CMS BPR DGM
Verificación	Grabador verificador	Team	20	Sí	ACBH CTP BRF MCG MASB CMS BPR DGM

3.2 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 in-

⁷ Expresada, en su caso, en horas por semana.

tegridad 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 EF	Grabador verificador
Desarrollador	Informa sobre novedades en funcionamiento de EF	Revisor
Desarrollador	Informa sobre novedades en funcionamiento de EF	Supervisor
Desarrollador	Informa sobre novedades en funcionamiento de EF	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 EF	Desarrollador
Supervisor	Solicita optimizaciones en prestaciones	Administrador
Supervisor	Valida calidad final de datos	Revisor

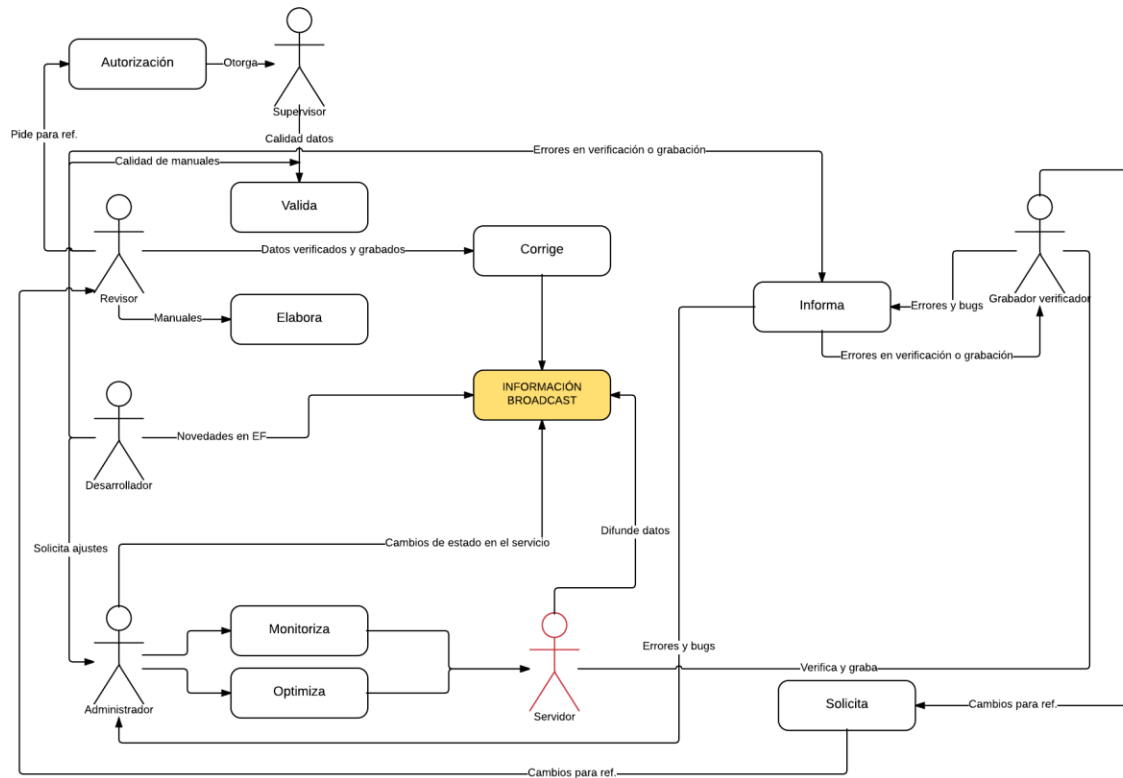


Ilustración 5. Interacciones entre actores en EF

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

4.1 Estimación de esfuerzo

Aunque habitualmente en los PDS suele incluirse un apartado con esta denominación donde se realiza una previsión económica, con un balance más o menos detallado de ingresos y gastos y un cómputo de tiempo y dinero atribuible a los diferentes miembros de los RRHH implicados en el proyecto, PDSDT, por la especial naturaleza del proyecto matriz, EX, recoge una valoración predictiva de esfuerzo temporal, puesto que el apartado dinerario es abordado desde el preceptivo documento de EX.

El cálculo de esfuerzo para EF se basa en la técnica de estimación a partir de los casos de uso. Para comprender adecuadamente tal estimación es necesario consultar los apartados 7 y 9, donde, tras establecer la estructura lógica de los datos se especifica el protocolo de verificación y grabación. De esa manera es posible evaluar el número de minutos requeridos para asentar un registro (según su tipología y casuística) y, por tanto, extrapolar los resultados.

La mecánica aquí empleada sirve también para identificar puntos de caso de uso no ajustados. La influencia de factores técnicos y ambientales en los procedimientos estándar de verificación y grabación se ha tenido en cuenta en la elaboración de estos cálculos, y se ha realizado tras culminarse con éxito la fase de elaboración (véanse, respectivamente, *Ilustración 4. Disciplinas y fases RUP en EF* y 4.2).

Para elaborar la estima, se ha empleado el patrón de metas para la evaluación de la calidad según una adaptación del modelo PMBOK⁸, en lo referido a ajuste temporal y objetivos numéricos por volumen de registros consolidados. En la siguiente tabla se recogen los valores finales a obtener, que han servido para obtener el guarismo estimativo. Se contabilizan como **dimensiones** los diversos tipos de registro de EF, algunos de los cuales se incluyen como referencia a la hora de evaluar el grado de éxito de EF, que figuran en la columna etiquetada como **PMBOK**. El **valor** se contabiliza en unidades por registro; es decir, cada registro que se ha establecido como meta final a obtener al término del proyecto. El **factor corrector** se ha introducido para considerar el peso que cada cifra tiene sobre el trabajo individual de cada actor. Para validar una exposición “capturada” (para Beagle y su operatoria automatizada, consultar 9) se requiere mayor cantidad de tiempo que para localizar y grabar una entidad. Multiplicando el valor primario por su factor corrector se obtiene el **valor ajustado**.

DIMENSIÓN	VALOR	FACTOR CORRECTOR	VALOR AJUSTADO	PMBOK
Entidades	22.000	0,8	17.600	Sí
Personas	14.500	0,7	10.150	No
Publicaciones	700	0,3	210	No
Empresas	50	0,2	10	No
Exposiciones	4.500	1	4.500	Sí
URIs RSS	6.000	0,2	1.200	Sí
URIs HTML	12.000	0,2	2.400	Sí
TOTALES	59.750		36.070	

La estimación del esfuerzo parte de una operación aritmética simple que permite averiguar el

⁸ PMI. *A Guide to the Project Management Body of Knowledge*. EEUU, 2012.

tiempo máximo que cada actor grabador verificador debe dedicar por registro. Teniendo en cuenta que no todas las entidades son productivas (es decir, que no todas son fuentes de datos), que hay que contar con periodos de baja actividad y que cada tipo de registro (ver 7) tiene, por su propia naturaleza, mayor o menor dificultad para conseguir una verificación completa, las ecuaciones necesarias se expresan a continuación. Para mayor facilidad de comprensión, se han redondeado los decimales.

Fechas de trabajo efectivo: Del 1 de abril de 2015 al 30 de septiembre de 2016 = 548 días
Días de trabajo efectivo: 70 semanas x 5 días de trabajo por semana = 350 días
Unidades de trabajo: (5 grabadores verificadores x 1) + (3 Revisores x 0,5) = 6,5
Horas diarias por unidad de trabajo: 4
Registros ajustados por día: 37.520 registros / 350 días = 107 registros/día
Registros ajustados por unidad de trabajo: 107 registros/día / 6,5 = 16 registros/día
Tiempo máximo de edición por registro: 16 registros/día / (4 horas x 60') = 15 minutos/registro

Aplicando esos parámetros, el resumen por tipo de actor puede consultarse en la siguiente tabla.

ACTOR	NÚMERO	UNIDADES	REG/DÍA	DÍAS	TOTAL/ACTOR
Grabador verificador	5	1	16	350	5.600
Revisor	3	0,5	8	350	2.800
TOTAL GRABADORES VERIF.	5			350	28.000
TOTAL REVISORES	3			350	8.400
TOTAL				350	36.400

El número total de grabaciones es **sensiblemente superior** al previsto, de 36.070 (340 registros más), con lo que el plan de despliegue de EF demuestra ser viable

4.2 Plan del proyecto

Como resultado de la estimación de esfuerzo a continuación se expone la planificación esencial del PDS de EF. Téngase en cuenta que no se trata de un marco rígido, sino que, en aplicación de los principios LP (véase 2.6) se adaptará cuando sea necesario al ambiente y las circunstancias concretas.

4.2.1 Plan de fases y temporalización

El desarrollo se llevará a cabo en fases con una o más iteraciones en cada una de ellas. La siguiente tabla muestra la previsión temporalizada, con expresión de las iteraciones. Los **hitos** son actividades, artefactos o entregables que determinan el logro de los objetivos de cada fase. Se incluye, además de las expresadas en el diagrama denominado *Ilustración 4. Disciplinas y fases RUP en EF*, la fase de *Mantenimiento*, imprescindible como parte del proyecto dada la naturaleza de EF. Dicha fase se considera integrante del ciclo de desarrollo porque la propia dinámica del proyecto, basado en capturas automatizadas (ver 9), implica ajustes finos de los sistemas de identificación de datos de manera que se garantice la resiliencia. La fase de mantenimiento no tiene un número predeterminado de iteraciones. La temporalización se expresa en semanas de trabajo.

FASE	ITERACIONES	SEMANAS	HITO
Inicio	1	5	Desarrollo de la ingeniería de requisitos a partir de las indicaciones de CL, los cuales se establecen en el artefacto PDSDT. Los principales casos de uso se identifican y se emplean para refinar PDS. La aceptación de CL del artefacto PDSDT y la puesta en marcha de PDS señalan el

FASE	ITERACIONES	SEMANAS	HITO
			final de esta fase.
Elaboración	1	10	Análisis de los requisitos, elección de un patrón de desarrollo y construcción de un prototipo de arquitectura (incluyendo las partes más relevantes o críticas del sistema). Al final de esta fase, todos los casos de uso correspondientes a requisitos que serán implementados en la primera <i>release</i> de la fase <i>Construcción</i> están analizados y diseñados (Modelo de Análisis y Diseño; véase 8). La revisión y aceptación del prototipo de la arquitectura EF señala el final de esta fase.
Construcción	3	15	Extensión de análisis y diseño a todos los casos de uso, refinando el Modelo de Análisis y Diseño. El producto se construye en base a 3 iteraciones, cada una produciendo una <i>release</i> a la cual se le aplican las pruebas y se valida con CL. Comienza la elaboración de material de apoyo al usuario. El hito que marca el fin de esta fase es la <i>release</i> 4, con la capacidad operacional parcial o totalmente implementada de los puntos críticos.
Transición	3	10	Preparación de tres <i>releases</i> para distribución, asegurando una implantación adecuada, incluyendo el entrenamiento de los usuarios. El hito que marca el fin de esta fase incluye la entrega de toda la documentación del proyecto con manuales de instalación y material de apoyo al usuario y finalización del entrenamiento del equipo de actores implicados en el mecanismo grabación - verificación - revisión.
Mantenimiento	N	30	Corrección de errores detectables en programación y lógica de captura. Ajuste del mecanismo Beagle (ver 9). Refinamiento de documentación. Entrenamiento de cualquier nuevo miembro de RRHH

4.2.2 Actividades y disciplinas

A continuación se presenta una matriz de las principales tareas y actividades programadas en EF. Como se ha establecido en 2, el proyecto sigue un patrón de proceso iterativo e incremental propio de RUP, caracterizado por la realización en paralelo de todas las disciplinas de desarrollo a lo largo del tiempo, con lo cual la mayoría de los artefactos son generados muy tempranamente pero van desarrollándose en mayor o menor grado de acuerdo a la fase e iteración, tal y como se puede apreciar en el modelo de *workflow* mostrado en *Ilustración 4. Disciplinas y fases RUP en EF*. La tabla siguiente muestra su relación con los diversos artefactos de EF.

DISCIPLINAS Y ACTIVIDADES	ARTEFACTO
Modelado de negocio <ul style="list-style-type: none"> • Describir negocio de CL • Desarrollar el modelo de dominio 	PDSDT. DT. Modelo de casos de uso
Ingeniería de requisitos <ul style="list-style-type: none"> • Analizar el problema • Clasificar y priorizar requisitos 	PDSDT. DT. Especificación de requisitos
Análisis y diseño <ul style="list-style-type: none"> • Especificar casos de uso • Realizar diagramas de procesos y secuencias • Realizar modelo de datos 	PDSDT. DT. Modelo de casos de uso PDSDT. DT. Modelo de diseño y arquitectura PDSDT. DT. Modelo de datos

DISCIPLINAS Y ACTIVIDADES	ARTEFACTO
<ul style="list-style-type: none"> Realizar prototipos 	
Implementación <ul style="list-style-type: none"> Estructurar modelo de implementación Planificar la integración Implementar componentes 	PDSDT. DT. Modelo de implementación
Pruebas <ul style="list-style-type: none"> Definir misión de pruebas Validar estabilidad de componentes 	PDSDT. DT. Plan de pruebas
Implantación <ul style="list-style-type: none"> Planificar implantación Desarrollar material de apoyo y manual de usuario 	Instalación del sistema Documentación y manual de usuario

4.2.3 Matriz de roles y responsabilidades

En la siguiente tabla se despliega la tabla cartesiana en la que se expresa el grado de participación o implicación de cada rol en los artefactos previstos y mencionados en 4.2.2. La participación del rol Supervisor en determinados artefactos viene determinada por su condición (en el caso concreto de EF) de *Product Owner* y, por tanto, “representante” de los intereses de CL.

ARTEFACTO	ROLES				
	Administrador	Desarrollador	Supervisor	Revisor	Grabador ver.
PDSDT. DT. Modelo de casos de uso		R	A		
PDSDT. DT. Especificación de requisitos		R	A P		
PDSDT. DT. Modelo de casos de uso		R	A	P	
PDSDT. DT. Modelo de diseño y arquitectura		R			
PDSDT. DT. Modelo de datos		R	A P		
PDSDT. DT. Modelo de implementación		R			
PDSDT. DT. Plan de pruebas		R	A	P	P
Instalación del sistema	P	A			
Documentación y manual de usuario		A	A	R	P

Clave: R= Se responsabiliza; A = Aprueba; P = Participa

4.3 Mecanismos de seguimiento y control

Con el fin de garantizar el funcionamiento del proyecto, EF establece los siguientes mecanismos de control y protocolos de funcionamiento, parte de cuya lógica está expresada en 3.2 y visualizada en *Ilustración 5. Interacciones entre actores en EF.*

4.3.1 Gestión de requisitos

Los requisitos del sistema, especificados en el apartado correspondiente de DT del artefacto PDSDT son susceptibles de cambios de acuerdo con el paradigma LP expresado en 2.6. Los cambios son gestionados mediante solicitud SC, verbal, por mensajería interna o por correo electrónico, evaluada por el rol *Desarrollador* y aprobada por *Supervisor*. Caso de que existan posibles consecuencias en el funcionamiento general del sistema, se notifican las acciones modificadoras o correctoras realizadas a *Administrador*.

4.3.2 Control de plazos

La temporalización dispone de seguimiento y evaluación quincenal por parte de *Supervisor* y *Desarrollador*.

4.3.3 Control de calidad

Los defectos detectados en las revisiones y formalizados en una SC (como la expresada en 4.3.1) disponen de seguimiento para asegurar la conformidad respecto de la solución de la deficiencia. Para la revisión de cada artefacto y su correspondiente garantía de calidad se usa el plan de calidad, definido en PDSDT en 13.

4.3.4 Gestión de riesgos

A partir de la fase de *Inicio* se mantiene una lista de riesgos asociados al proyecto y de las acciones establecidas como estrategia para mitigarlos o acciones de contingencia. Esta lista es evaluada al menos una vez en cada iteración.

4.3.5 Gestión de configuración

Se establece un registro de los artefactos generados y sus versiones (*changelog*). La gestión de las SC y las modificaciones inducidas por ellas o de oficio se publica en el sistema interno de mensajería de manera que la información sobre dichos cambios sea accesible a todos los integrantes de RRHH. Al final de cada iteración se establece una *baseline* que solo es modificable mediante SC aprobada por *Supervisor*.

5 Infraestructura

La modalidad de trabajo cliente/servidor, adoptada como solución funcional para EF 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 40x ó superior, Safari, Firefox... o bien Microsoft Edge).

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

Lado servidor

Equipo blade o enracable con resiliencia eléctrica y almacenamiento RAID.

Sistema operativo Linux, con acceso *root*.

SGBD MySQL o MariaDB.

Intérprete de scripts PHP.

Conexión de alta velocidad a Internet.

Dominio Internet de tipo ".es"

De acuerdo con los requerimientos solicitados por CL, que se expresan en 8, y para el correcto funcionamiento del sistema EF, *Product Owner/Supervisor* aprueba la adquisición en modalidad **alquiler temporal** de equipamiento en la nube para que actúe como servidor por el plazo de duración establecido para EX (y, por tanto, para EF) renovable en ciclos de doce meses, del equipo que se detalla. Se opta por un alquiler de **servidor dedicado** y no de hospedaje por las superiores características de funcionamiento y por permitir el control absoluto sobre el SO.

El alquiler se ha formalizado con la empresa **OVH**⁹, en concreto de su producto **Servidor Enterprise SP-64 - 64G E5-1620v2 SoftRAID 2x 2 TB**. La contratación se ha realizado por un plazo de **12 meses**, con expiración el **1 de mayo de 2016**, con un importe total de **1.197,73 euros** por el periodo, IVA incluido, en concepto de servidor, conexión y dominio.

EQUIPAMIENTO	CARACTERÍSTICAS
Servidor dedicado	CPU: Intel(R) Xeon(R) CPU E5-1620 v2 @ 3.70GHz. Núcleos : 8 Caché : 10240KB RAM: 4x 16384MB Discos: 2 x 2000 GB RAID 5 MegaRAID 9271 Caché 1 GB + CacheVault Placa base: X9SRE/X9SRE-3F/X9SRi/X9SRi-3F
Alojamiento	Datacenter: GRA 1

⁹ OVH es un proveedor de alojamiento web francés. Ofrece servidores dedicados, alojamiento compartido, registro de dominios, VPS y servicios de Cloud Computing.

EQUIPAMIENTO	CARACTERÍSTICAS
	Rack: G115A12 ID del servidor: 544471 Versión de Real Time Monitoring: 0.9.4-3 Versión del kernel: 3.14.32-xxxx-std-ipv6-64
Software	SO: Ubuntu Server 15.04 "Vivid Vervet" Boot: 3.14.32 (Stable kernel, vanilla - 64bit) Intérprete de scripts: 5.6.11-1ubuntu3.1
Conectividad	Conexión de red: 1 Gbps Ancho de banda OVH/OVH: 1 Gbps Ancho de banda garantizado: 500 Mbps Ancho de banda Internet/OVH: 1 Gbps Anti-DDoS PRO incluido Burst de 1 Gbps Dirección IP v6: 2001:41d0:a:7bb5::/64 Dirección IP v4: 5.196.92.181 Dominio: expofinder.es

6 Estructura física de la información

Según se expresa en el análisis de requisitos y en la arquitectura consecuentemente adoptada (ver 8), existe 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 EF.

La organización responde a la correspondiente a la plataforma usada como FW: WordPress (en adelante, WP). Fue fijada desde la versión 3.8 del mismo. La lista se expresa en la siguiente tabla.

Tabla: wp_postmeta					
Campo	Tipo	Nulos	Clave	Por defecto	Extra
meta_id	bigint(20) unsigned		PRI		auto_increment
post_id	bigint(20) unsigned		IND	0	
meta_key	varchar(255)	YES	IND	NULL	
meta_value	longtext	YES		NULL	

Índices

Nombre	Tipo	Campos
PRIMARY	PRIMARY	meta_id
post_id	INDEX	post_id
meta_key	INDEX	meta_key

Tabla: wp_posts					
Campo	Tipo	Nulos	Clave	Por defecto	Extra
ID	bigint(20) unsigned		PRI & IND Pt4		auto_increment
post_author	bigint(20) unsigned		IND	0	
post_date	datetime		IND Pt3	0000-00-00 00:00:00	
post_date_gmt	datetime			0000-00-00 00:00:00	
post_content	longtext				
post_title	text				
post_excerpt	text				
post_status	varchar(20)		IND PT2	publish	
comment_status	varchar(20)			open	
ping_status	varchar(20)			open	
post_password	varchar(20)				
post_name	varchar(200)		IND		
to_ping	text				
pinged	text				
post_modified	datetime			0000-00-00 00:00:00	
post_modified_gmt	datetime			0000-00-00 00:00:00	
post_content_filtered	longtext				
post_parent	bigint(20) unsigned		IND	0	
guid	varchar(255)				
menu_order	int(11)				0
post_type	varchar(20)		IND Pt1	post	
post_mime_type	varchar(100)				
comment_count	bigint(20)				0

Índices

Nombre	Tipo	Campos
PRIMARY	PRIMARY	ID
post_name	INDEX	post_name
type_status_date	INDEX	post_type post_status post_date ID
post_parent	INDEX	post_parent
post_author	INDEX	post_author

Tabla: wp_terms					
Campo	Tipo	Nulos	Clave	Por defecto	Extra
term_id	bigint(20) unsigned		PRI		auto_increment
name	varchar(200)		IND		
slug	varchar(200)		UNI		
term_group	bigint(10)			0	

Índices

Nombre	Tipo	Campos
PRIMARY	PRIMARY	term_id
slug	UNIQUE	slug
name	INDEX	name

Tabla: wp_term_relationships					
Campo	Tipo	Nulos	Clave	Por defecto	Extra
object_id	bigint(20) unsigned		PRI Pt1	0	
term_taxonomy_id	bigint(20) unsigned		PRI Pt2 & IND	0	
term_order	int(11)			0	

Índices

Nombre	Tipo	Campos
PRIMARY	PRIMARY	object_id term_taxonomy_id
term_taxonomy_id	INDEX	term_taxonomy_id

Tabla: wp_term_taxonomy					
Campo	Tipo	Nulos	Clave	Por defecto	Extra
term_taxonomy_id	bigint(20) unsigned		PRI		auto_increment
term_id	bigint(20) unsigned		UNI Pt1	0	
taxonomy	varchar(32)		UNI Pt2 & IND		
description	longtext				
parent	bigint(20) unsigned			0	
count	bigint(20)			0	

Índices

Nombre	Tipo	Campos
PRIMARY	PRIMARY	term_taxonomy_id
term_id_taxonomy	UNIQUE	term_id taxonomy
taxonomy	INDEX	taxonomy

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

Las tablas adicionales, no previstas por el FW, que se emplean para la gestión de EF, son las siguientes.

Tabla: wpaef_xtr_activity_log			
Campo	Tipo	Nulos	Por defecto
log_id (Primaria)	bigint(20)	No	
user_id	bigint(20)	No	0
activity	varchar(30)	No	log_in
object_id	bigint(20)	No	0
object_type	varchar(20)	No	post
activity_date	datetime	No	0000-00-00 00:00:00

Índices

Nombre	Tipo	Único	Empaquetado
PRIMARY	BTREE	Sí	No
abc	BTREE	No	No

Tabla: wpaef_xtr_beaglecr_log			
Campo	Tipo	Nulos	Por defecto
log_id (Primaria)	int(11)	No	
log_date	timestamp	No	CURRENT_TIMESTAMP
checked_uris	bigint(20)	No	
invalid_uris	bigint(20)	No	
valid_uris	bigint(20)	No	
sapless_uris	bigint(20)	No	
checked_entries	bigint(20)	No	
sapless_entries	bigint(20)	No	
sapfull_entries	bigint(20)	No	
added_entries	bigint(20)	No	
discarded_entries	bigint(20)	No	
entries_by_valid_uri	float	No	
entries_by_useful_uri	float	No	
operation_time	int(11)	No	
average_time	float	No	

Índices

Nombre	Tipo	Único	Empaquetado
PRIMARY	BTREE	Sí	No
def	BTREE	No	No

Tabla: wpaef_xtr_dashboard_chat_log			
Campo	Tipo	Nulos	Por defecto
id (Primaria)	bigint(20)	No	
user_id	bigint(20)	No	
date	datetime	No	CURRENT_TIMESTAMP
content	text	No	
checked	bigint(2)	No	0

Índices

Nombre	Tipo	Único	Empaquetado
PRIMARY	BTREE	Sí	No

Tabla: wpaef_xtr_nbc_b8_wordlist			
Campo	Tipo	Nulos	Por defecto
token (Primaria)	varchar(255)	No	
count_ham	int(10)	Sí	NULL
count_spam	int(10)	Sí	NULL

Índices

Nombre	Tipo	Único	Empaquetado
PRIMARY	BTREE	Sí	No

Tabla: wpaef_xtr_projectd_log			
Campo	Tipo	Nulos	Por defecto
task_order	int(11)	No	
task_id (Primaria)	varchar(50)	No	
task_name	varchar(100)	No	
resource	varchar(50)	No	
start_date	date	No	
end_date	date	No	
dependencies	varchar(150)	Sí	NULL
duration	int(11)	No	
percent_complete	int(11)	Sí	NULL

Índices

Nombre	Tipo	Único	Empaquetado
PRIMARY	BTREE	Sí	No

Tabla: wpaef_xtr_urierror_log			
Campo	Tipo	Nulos	Por defecto
log_id (Primaria)	int(11)	No	
log_date	timestamp	No	CURRENT_TIMESTAMP
post_author	bigint(20)	No	
invalid_rss_uri	longtext	Sí	NULL
invalid_html_uri	longtext	Sí	NULL

Índices

Nombre	Tipo	Único	Empaquetado
PRIMARY	BTREE	Sí	No
def	BTREE	No	No
post_author	BTREE	No	No

7 Estructura lógica de la información

A continuación se suministra la información básica sobre la estructura lógica del sistema de información de EF. 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 EF, 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*.

Los elementos fundamentales del SI EF quedan definidos, en el contexto de WP utilizado como FW, como **custom posts**, almacenándose la información principal en la tabla *posts*, la metainformación en *postmeta* y las taxonomías en el conjunto *terms*, *term_relationships* y *term_taxonomy*.

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

- **Entidades ENT.** Se entiende como entidad (según DRAE) "cualquier corporación, compañía, institución, etc., tomada como persona jurídica". En este caso será considerado como tal cualquier organismo con entidad legal que se utilizará en el contexto de ExpoFinder bien como un origen de datos RSS/Atom o HTML, a partir de cuyos resultados el sistema explora la Red en búsqueda de información sobre exposiciones temporales, bien como cualquier otro organismo que actúe como soporte organizador o financiero de una exposición, de su catálogo o al que se vincule una persona de forma laboral, investigadora o académica.
- **Personas PER.** Se entiende como persona en el contexto DT a quienes interactúan con publicaciones o exposiciones a título de autoría o comisariado.
- **Publicaciones PUB.** Se entiende como publicación en el contexto DT cualquier artículo, libro, página web o texto en general que se emplee para ilustrar al visitante acerca de una determinada exposición. Su uso "natural" es para almacenar información sobre los "catálogos" de la misma.
- **Empresas EMP.** Se entiende como empresa en el contexto DT cualquier organismo cuya relación con las entidades, las exposiciones, las publicaciones o las personas pueda definirse EXCLUSIVAMENTE en términos mercantiles, comerciales o de negocio. La distinción, pues, no se basa en el carácter público, privado o mixto de la misma ni en el hecho de que sus fines incluyan o no ánimo de lucro, sino en que la RELACIÓN sea lucrativa.
- **Exposiciones EXP.** Se entiende como exposición en el contexto DT cualquier evento en el que se exhiba ante el público un conjunto de obras de carácter artístico o cuya intencionalidad sea la de difundir conocimientos propios de la disciplina académica de Historia del Arte.

7.2 Relaciones entre elementos

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 símbo-

lo → significa *se relaciona con*):

Exposiciones

- Exposiciones → Entidades:
- Organización [1→∞]. La entidad organiza la exposición.
- Financiación [1→∞]. La entidad financia la exposición.
- Fuente de información [1→1]. La entidad es la fuente de información sobre los datos de la exposición mediante las URIs HTM, RSS o ATOM.
- Exposiciones → Personas:
- Autoría [1→∞]. La persona es autor de alguna obra exhibida en la exposición.
- Comisariado [1→∞]. La persona es comisaria de la exposición.
- Colección [1→∞]. La persona es coleccionista de una o más obras exhibidas en la exposición.
- Exposiciones → Publicaciones:
- Catálogo [1→∞]. La persona es autor de alguna obra exhibida en la exposición.
- Exposiciones → Empresas:
- Museografía [1→∞]. La empresa ha realizado el proyecto museográfico de la exposición.
- Exposiciones → Exposiciones:
- Dependencia [1→1]. La exposición (derecha) es la principal en una relación descendente de dependencia de la exposición.

Entidades

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

Personas

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

Publicaciones

- Publicaciones → Entidades:
- Patrocinio [1→1]. La entidad patrocina o financia la publicación.
- Publicaciones → Personas:
- Autoría [1→∞]. La persona es autora de la publicación.

7.3 Taxonomías

En el vocabulario WP, por taxonomía se entiende un mecanismo de clasificación y agrupamiento de los diferentes registros portadores de información primaria (*posts* en este contexto) basado en términos y grupos de términos que pueden responder, a voluntad del desarrollador, a una estructura jerárquica. Es posible, utilizando este aplicativo, descargar la información, incluyendo los

términos, de todas las taxonomías empleadas en EF. Su organización estructural responde al siguiente esquema. Tenga en cuenta que el símbolo ® determina que los términos de la taxonomía forman un vocabulario controlado, y que el símbolo § significa que la taxonomía tiene una estructura jerárquica:

Entidades:

- Tipología ®. Caracterización básica (en calidad de qué actúa) de la entidad en el sistema de información.
- Titularidad ®. Agrupamiento elemental de la entidad en función de si el titular o propietario de la misma se acode a derecho público, privado o a una mezcla de ambos.
- Palabras clave. Términos identificativos y/o descriptivos utilizables por el usuario redactor del registro a voluntad.

Personas:

- Actividad ®. Tareas conocidas que una persona desempeña y que pueden jugar un papel significativo en el contexto de los datos de la aplicación.

Publicaciones:

- Editorial ®. Empresa u organismo que actúa como editor de la publicación.

Empresas:

- Categoría ISIC4 ®. Clasificación de la empresa por su actividad principal. Sigue la convención de *United Nations Statistics Division (UNSTAT). International Standard Industrial Classification of All Economic Activities, Rev.4.*

Exposiciones:

- Tipo de exposición ®. Caracterización básica de la exposición según jerarquía basada en los trabajos de Michael Belcher y Ángela García Blanco.
- Tipo de obras ®§. Tipologías de las obras expuestas de acuerdo con las categorías recogidas en los tesauros de la Fundación Getty.
- Tema ®§. Temáticas de las obras expuestas de acuerdo con las categorías recogidas en los tesauros de la Fundación Getty.
- Movimiento ®. Movimientos o corrientes artísticas a las que es posible adscribir las obras expuestas de acuerdo con las categorías recogidas en los tesauros de la Fundación Getty.
- Periodo ®. Categoría temporal de las obras expuestas basada en grandes periodos cronológicos (siglos y épocas históricas).

7.4 Metadatos

Cada tipo de registro asentado como *custom post* (entidades, personas, publicaciones, empresas y exposiciones) dispone de una considerable cantidad de metainformación asociada, que define sus características específicas. Al no tratarse en puridad de un mecanismo de agrupamiento o clasificatorio, dicha metainformación no se considera taxonomía. El inventario formal de los campos adicionales adscritos a cada tipo de registro se resume a continuación.

En las listas que se muestran a continuación se emplean ciertos signos convencionales:

- El símbolo \emptyset significa que el campo es autorreferenciado, es decir que busca entre los registros (*custom posts*) cuya abreviatura se expresa a continuación del símbolo. En este caso, el campo sigue la regla *ID + :*, es decir que cada valor para dicho campo va precedido por un número que coincide con el valor ID del tipo de registro mencionado, seguido por dos puntos (:) y un espacio en blanco.
- El símbolo \ast significa que el valor del campo está verificado por *Geonames*. En caso de que el campo esté destinado a almacenar el nombre de una localidad sigue la regla *; * 2*, es decir que cada valor debe contener 3 elementos separados por punto y coma (;), correspondiendo con la siguiente estructura: *municipio; región o estado; país*. En caso de que esté destinado a almacenar el nombre de un país, simplemente conservará ese valor, sin ningún separador.
- El símbolo \triangle significa que el valor del campo está verificado por Google Maps.
- El símbolo \boxtimes significa que el valor del campo no ha podido ser modificado por el usuario y se calcula automáticamente.
- El símbolo \otimes significa que el valor del campo ha sido automáticamente obtenido o calculado por el proceso Beagle.

Aquellos campos que deben contener una fecha hacen uso de las siguientes convenciones:

- FECHAS ÚNICAS (un único campo)
 - Si se conoce la fecha completa figura con el formato año de cuatro dígitos - mes de dos dígitos - día de dos dígitos.
 - Si se conoce sólo mes y año figura con el formato año de cuatro dígitos - mes de dos dígitos - 01.
 - Si se conoce sólo año figura con el formato año de cuatro dígitos - 01 - 01.
- FECHAS PAREADAS (inicio y fin)
 - Si se conocen las dos fechas completas figura en ambos casos con el formato año de cuatro dígitos - mes de dos dígitos - día de dos dígitos.
 - Si se conoce sólo mes y año de una de las dos fechas se atiene al siguiente esquema:
 - Si se conoce la fecha de inicio, la fecha de fin será año de cuatro dígitos - mes de dos dígitos - último día posible del mes.
 - Si se conoce la fecha de fin, la fecha de inicio será año de cuatro dígitos - mes de dos dígitos - 01.
 - Si se conoce sólo año de una de las dos fechas se atiene al siguiente esquema:
 - Si se conoce la fecha de inicio, la fecha de fin será año de cuatro dígitos - 12 - 31.
 - Si se conoce la fecha de fin, la fecha de inicio será año de cuatro dígitos - 01 - 01.

Entidades		
Nombre interno del campo	Relación	Contenido
csl_ent_alternate_name	[1→1]	Nombre alternativo
csl_ent_town	[1→1] *	Localidad
csl_ent_url	[1→1]	URL
csl_ent_email	[1→1]	Correo electrónico
csl_ent_phone	[1→1]	Teléfono
csl_ent_fax	[1→1]	Fax
csl_ent_chief_executive	[1→1]	Máximo responsable
csl_ent_relationship_type	[1→1]	Tipo de relación
csl_ent_parent_entity	[1→1] ◊ ENT	Entidad matriz
csl_ent_coordinates	[1→1] ☒	Coordenadas
csl_ent_address	[1→1] ☒	Dirección postal
csl_ent_rss_uri	[1→∞]	URI RSS
csl_ent_html_uri	[1→∞]	URI HTML
csl_ent_sirius_m_total_rank	[1→1]	SIRIUS-M. Puntuación total
csl_ent_sirius_m_general_aspects	[1→1]	SIRIUS-M. Aspectos generales
csl_ent_sirius_m_identity_and_information	[1→1]	SIRIUS-M. Identidad e información
csl_ent_sirius_m_navigation_and_structure	[1→1]	SIRIUS-M. Navegación y estructura
csl_ent_sirius_m_labeling	[1→1]	SIRIUS-M. Rotulación
csl_ent_sirius_m_page_layout	[1→1]	SIRIUS-M. Formato de página
csl_ent_sirius_m_understandability_and_ease_of_information	[1→1]	SIRIUS-M. Comprensibilidad y facilidad de información
csl_ent_sirius_m_control_and_feedback	[1→1]	SIRIUS-M. Control y retroalimentación
csl_ent_sirius_m_multimedia_elements	[1→1]	SIRIUS-M. Elementos multimedia
csl_ent_sirius_m_search	[1→1]	SIRIUS-M. Buscar
csl_ent_sirius_m_help	[1→1]	SIRIUS-M. Ayuda
csl_ent_sirius_m_seo	[1→1]	SIRIUS-M. Posicionamiento web (SEO)
csl_ent_sirius_m_social_networks	[1→1]	SIRIUS-M. Redes sociales
NOTA: Los campos SIRIUS-M puntúan entre 0 y 10. El campo Tipo de relación tiene las siguientes alternativas: Entidad principal, Entidad secundaria, Entidad auxiliar.		
Personas		
Nombre interno del campo	Relación	Contenido
csl_peo_person_type	[1→1]	Tipo de persona
csl_peo_last_name	[1→1]	Apellidos
csl_peo_first_name	[1→1]	Nombre de pila
csl_peo_gender	[1→1]	Sexo
csl_peo_birth_date	[1→1]	Fecha de nacimiento
csl_peo_death_date	[1→1]	Fecha de defunción
csl_peo_country	[1→1] *	País
csl_peo_entity_relation	[1→∞] ◊ ENT	Relación con entidad
csl_peo_person_relation	[1→∞] ◊ PER	Relación con persona
NOTA: El campo Tipo de persona tiene las siguientes alternativas: Individuo, Grupo de personas. El campo Sexo tiene las siguientes alternativas: No declarado, Masculino, Femenino.		
Publicaciones		
Nombre interno del campo	Relación	Contenido
csl_boo_publishing_date	[1→1]	Fecha de publicación
csl_boo_publishing_place	[1→1] *	Lugar de publicación
csl_boo_paper_author	[1→∞] ◊ PER	Autor de la publicación
csl_boo_sponsorship	[1→∞] ◊ ENT	Patrocinio

Empresas		
Nombre interno del campo	Relación	Contenido
_csl_com_company_headquarter_place	[1→1] *	Localidad de la sede principal
_csl_com_company_dimension	[1→1]	Dimensión
NOTA: El campo Dimensión tiene las siguientes alternativas: Pequeña empresa (< 50 empleados), Mediana empresa (50-200 empleados), Gran empresa (> 200 empleados).		
Exposiciones		
Nombre interno del campo	Relación	Contenido
_csl_exh_crc32b_identifier	[1→1] ☒ ☉	Identificador CRC32B
_csl_exh_global_keywords_weight	[1→1] ☒ ☉	Peso global de las palabras clave
_csl_exh_relative_keywords_weight	[1→1] ☒ ☉	Peso relativo de las palabras clave
_csl_exh_found_keywords	[1→1] ☒ ☉	Palabras clave localizadas
_csl_exh_original_publishing_date	[1→1] ☒ ☉	Fecha de publicación
_csl_exh_original_reference_author	[1→1] ☒ ☉	Autor de la referencia
_csl_exh_original_source_link	[1→1] ☒ ☉	Enlace a la fuente original de la referencia
_csl_exh_source_entity	[1→1] ☒ ☉ ☐ ENT	Entidad origen
_csl_exh_nbc_valuation	[1→1] ☒ ☉	Error CBI
_csl_exh_pst_list	[1→1] ☒ ☉	Posibles marcas significativas (PMS)
_csl_exh_exhibition_start_date	[1→1]	Fecha de comienzo
_csl_exh_exhibition_end_date	[1→1]	Fecha de finalización
_csl_exh_exhibition_town	[1→1] *	Localidad
_csl_exh_exhibition_site	[1→1]	Sede
_csl_exh_exhibition_visitors	[1→1]	Rango de visitas
_csl_exh_exhibition_access	[1→1]	Tipo de acceso
_csl_exh_coordinates	[1→1] ☒	Coordenadas
_csl_exh_address	[1→1] ☒	Dirección postal
_csl_exh_relational_type	[1→1]	Tipo de relación
_csl_exh_parent_exhibition	[1→1] ☐ EXP	Exposición de la que depende
_csl_exh_info_source	[1→1] ☐ ENT	Fuente de información
_csl_exh_artwork_author	[1→∞] ☐ PER	Autoría
_csl_exh_supporter_entity	[1→∞] ☐ ENT	Entidad organizadora
_csl_exh_funding_entity	[1→∞] ☐ ENT	Entidad financiadora
_csl_exh_curator	[1→∞] ☐ PER	Comisario
_csl_exh_catalog	[1→∞] ☐ PUB	Catálogo
_csl_exh_museography	[1→∞] ☐ EMP	Empresa responsable de la museografía
_csl_exh_geotag	[1→∞] *	Geoetiqueta
_csl_exh_art_collector	[1→∞] ☐ ENT PER	Coleccionista

8 Diseño de la arquitectura del sistema

Desde su fase de diseño, y de acuerdo con el análisis de ingeniería de requisitos realizado a partir de las especificaciones planteadas por EX, EF 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 FW, 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, PHP es el lenguaje de desarrollo elegido, sobre una plataforma Unix/Linux. Durante la fase de diseño se realizaron pruebas con diversas distribuciones *framework* bien conocidas y utilizadas para desarrollos RAD, como *CakePHP*, *CodeIgniter* o *Laravel*. Sin embargo, pronto surgió (y, evidentemente se afianzó) la posibilidad de utilizar un conocido CMS como **WordPress** para esa finalidad.

La prensa *online* especializada anda ciertamente dividida en cuanto a la posibilidad de usar WP como FW, con posturas a menudo encontradas y, hasta cierto punto, escasamente razonadas en numerosas ocasiones. Sin embargo es evidente que WP ofrece todo un complejo API, 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.
- Posibilidad de localización e internacionalización completa mediante el uso de *gettext* según principios i18n.
- Completo API para la gestión de formularios y la validación de datos.
- Facilidades para la administración de envío y recepción de archivos, con inserción de metadatos que permiten su localización e identificación.
- Consistente grupo de rutinas y clases para la manipulación de imágenes.
- Gestión fácil de URLs amigables para los motores de búsqueda y SEO.
- Implementación de un completo *subset* de procedimientos XML-RPC.
- Uso de *hooks*, clases abstractas, prototipos y mecanismos de tipo *plugin* que permiten ampliar o modificar la funcionalidad básica.

¹⁰ Multiple (wiki). "Web application framework". Docforge. Consultado 2015-01-19.

- 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 FW 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. Según el buscador Technorati, el 48% de sus *top 100 blogs* usan dicho software para su funcionamiento, y 74.652.825 sedes en la Red lo emplean como su sistema principal. Se han realizado 96 versiones hasta la fecha, y su núcleo cuenta con más de 40 traducciones.

Por todo lo expresado, la decisión sobre el FW a emplear resultó claramente favorable a WP. 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 DE WP, 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 de WP.

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

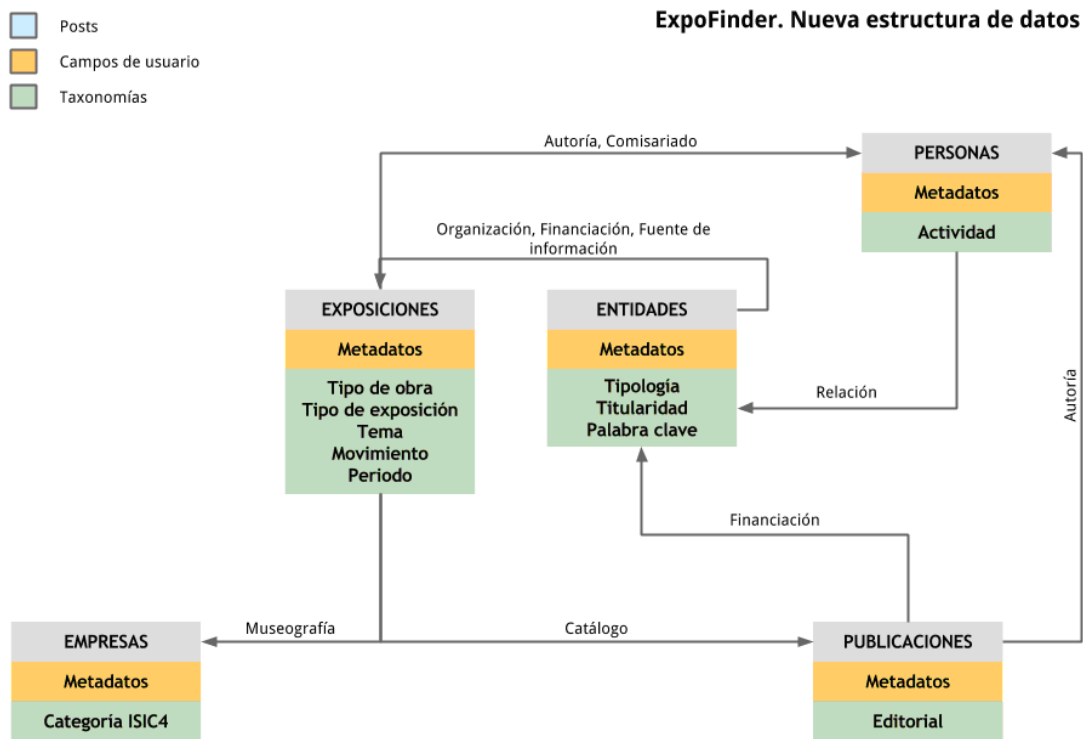


Ilustración 6. Modelo simplificado SI EF

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
Exposiciones	Fuente de información	Entidades
Exposiciones	Organización	Entidades
Exposiciones	Financiación	Entidades
Exposiciones	Autoría	Personas
Exposiciones	Comisariado	Personas
Exposiciones	Museografía	Empresas
Exposiciones	Catálogo	Publicaciones
Personas	Relación	Entidades
Publicaciones	Autoría	Personas
Publicaciones	Financiación	Entidades

WP está excelentemente diseñado, y jugando con su estructura ya se han realizado al menos dos proyectos de envergadura para transformarlo en un modelo lo más universal posible que permita realizar desarrollos sobre el. Aunque, en realidad, y de acuerdo con lo expresado por Tom McFarlin¹¹ en su conocida página “tuts+”, es más una *foundation* (cimiento) que un *framework*.

Y tal vez no ande escaso de razón: 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*. Sin embargo, una *foundation* permite extender una aplicación ya existente. WP tiene sus propios mecanismos internos bien definidos, y la *foundation* simplemente amplía su funcionamiento o lo aprovecha en su propio beneficio. Y eso es lo que EF realiza con WP: aprovecha la base de datos predefinida, las APIs disponibles, y el sistema de plantillas para la visualización de datos para construir soluciones usando un aplicativo que ya está plenamente funcional.

Existen en la Red proyectos que basan su funcionamiento en implementar un mecanismo MVC (*Model, View, Controller*) sobre WP, 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 EF porque hacían excesivamente complejo el desarrollo y obligaban a “forzar la máquina”, teniendo que realizar adaptaciones y modificaciones en el núcleo principal (*core*) de WP, lo que suponía renunciar a algunas de sus ventajas:

- Una base de datos con un esquema organizativo flexible y muy sólido.

¹¹ T. McFarlin. *Using WordPress for Web Application Development: An Introduction*. 2013. <http://code.tutsplus.com/tutorials/using-wordpress-for-web-application-development-an-introduction--wp-33828>. Consultado 20-1-2015.

- Una capa de aplicación principal con numerosos *hooks* que permiten aprovechar al máximo su funcionalidad.
- 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 WP como FW que a continuación se relacionan se incorporan a EF como núcleo fundamental sobre el que construir la arquitectura SW.

- Comunidad de desarrollo, soporte y documentación. WP 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. WP tiene un módulo de gestión de usuarios excelente, que se encarga de toda la funcionalidad, tal como el registro de usuarios y de inicio de sesión, gestión de roles, asignación de capacidades a diferentes roles y la creación de nuevas soluciones.
- Pantallas de administración. WP ofrece todo un interfaz, que permite crear páginas propias de configuración personalizada 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 o conjunto de registros almacenados.
- Completísimo juego de funciones internas perfectamente aprovechables. Incluyen desde la gestión de la internacionalización (mediante mecanismos *gettext*) 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. WP se ocupa de la carga de archivos y la gestión de los medios audiovisuales, incluyendo su securización.
- 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. Su mecanismo API interno, basado en *hooks*, filtros, preprocesadores y *callbacks*, permiten intervenir en prácticamente cualquier operación que WP tenga predefinida.
- Enrutamiento de URIs y URLs “SEO friendly”. Son características de gran importancia a la hora de optar por una modelización u otra. La trascendencia de enrutar adecuadamente las llamadas a las distintas estructuras de la aplicación es innegable. Por otra parte, la estructura de la URL mediante la cual se identifica un contenido juega un papel muy importante en el mecanismo SEO, decisivo a la hora de que un proyecto cumpla con las condiciones de la web semántica, y WordPress implementa magníficamente ambas facilidades.
- *Caching*. Si se está buscando una aplicación de alto rendimiento que pueda servir muchas

peticiones simultáneas y manejar decenas de miles de usuarios, es necesario disponer de un mecanismo de almacenamiento en caché eficiente pero que no sacrifique la precisión en aras de las prestaciones. WP y su API de *transients* proporcionan funcionalidad de almacenamiento en caché a nivel de la base de datos que se utilizará en la aplicación.

- Plantillas. La presentación o *front-end* no es un mero adorno. Mediante una peculiar arquitectura basada en funciones y con extenso uso de *hooks* y retrollamadas, WP dispone de una potentísima mecánica de formateo final de páginas, los temas, de gran flexibilidad tanto en su cara administrativa como en la de CL. Una misma página, por ejemplo puede disponer de varias plantillas que modifiquen o amplíen la experiencia del usuario.

No es, ni mucho menos, despreciable la capacidad de WP en cuanto se refiere a escalabilidad. La posibilidad de redimensionar la operatividad de EF es un punto importante a su favor. WP puede trabajar con dos SGBD: MySQL y MariaDB, heredero éste último del anterior, que obtiene unas cifras de velocidad y capacidad de respuesta en ejecución más que notables.

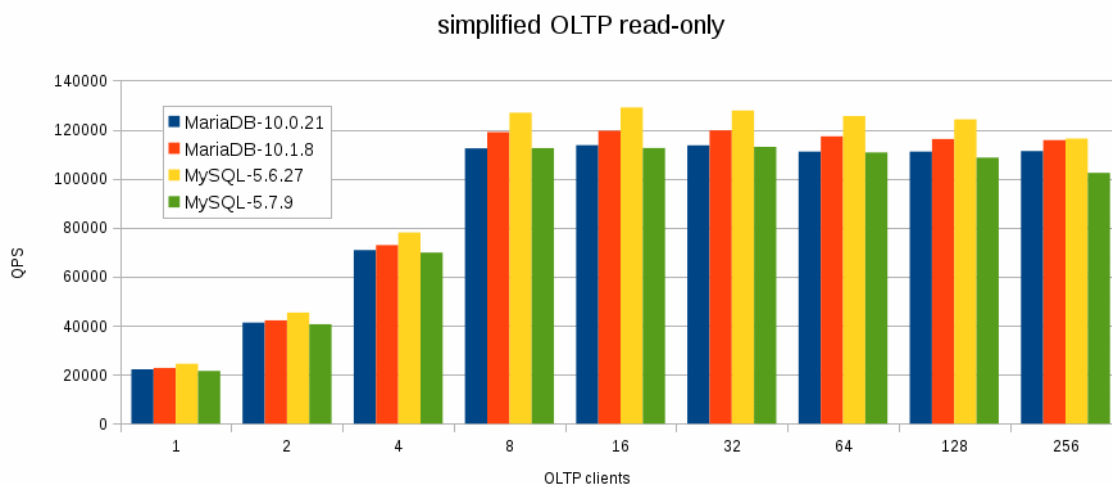


Ilustración 7. Comparativa de test de velocidad MariaDB y MySQL

Los inconvenientes más destacados pasan por una cierta rigidez estructural si se desea aprovechar la magnífica configuración de los *custom post*, los metadatos y las taxonomías, jerárquicas o no, que en ocasiones dificulta bastante la posibilidad, por ejemplo, de establecer vocabularios controlados o tesauros. No obstante, tales obstáculos pueden ser considerados menores en comparación con las capacidades de que WP hace gala.

8.2 Ingeniería de requisitos

Dadas las peculiaridades y el ambiente de EF (subproyecto al servicio de EX), el actor *Supervisor*¹² se encarga de la especificación de requisitos. La definición de los mismos fue expresada por NRO en el documento de solicitud adjuntado al proyecto EX. Según ese texto, se trataba de explo-

¹² Recuérdese que, en el caso de EF, coincide con el *Product Owner* de la filosofía SCRUM, lo que le faculta para decidir los objetivos finales en función de su pertinencia para la obtención de resultados en EX.

tar la función dinamizadora de la cultura artística desarrollada por los museos y las galerías mediante las exposiciones temporales, entendidas éstas como fenómenos complejos de actores en relación dinámica. Es un proyecto que también se enfoca a mejorar la sociedad facilitando el acceso y la reutilización de contenidos culturales digitales, y en este sentido plantea como uno de sus resultados desarrollar una plataforma web que proporcionará información de interés al usuario, sea aficionado al arte, agente de viajes, hotelero, galerista, conservador de museos, comisario de exposiciones o investigador. 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 EF, 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 NRO y MLBG en calidad de *Supervisoras*, y ha sido concretado en un documento, a partir del cual se ha elaborado la estructura lógica y física del proyecto EF, y que no se adjunta en el presente PDSDT por formar parte de la documentación de EX.

Mediante la consulta de dicho documento se ha realizado la **matriz de requisitos y procesos** que a continuación se recoge, 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 EX, 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 entidades susceptibles de producir información acerca de ETA	EX	Búsqueda ¹³ Grabación	Sí
Agregar a la información de entidades URIs que puedan ser usadas para automatizar el proceso de recogida de información	EX	Grabación	Sí
Automatizar la recogida de información sobre ETA mediante proceso sin intervención humana de carácter cíclico	EX	Captura	Sí
Seleccionar de forma automática la información sobre ETA recogida de manera que solo se almacene aquella que pueda ser relevante	EX	Filtrado	Sí
Intervenir con actores pertenecientes a RRHH dotados de la adecuada formación en DH y DAH para eliminar cualquier información sobre ETA que el proceso automatizado haya podido incluir de forma inadecuada	EX	Verificación	Sí
Completar la metainformación pertinente para la información sobre ETA automáticamente capturada, filtrada y seleccionada	EX	Grabación	Sí
Facilitar la transferencia de la información recopilada y verifica-	EX	Exportación	Sí

¹³ El proceso de búsqueda debe entenderse como actividad exclusivamente realizada por RRHH, sin intervención del sistema informático. Se trata de utilizar los conocimientos en DH y DAH de los actores *Grabador verificador* y *Revisor* para localizar posibles fuentes de información a partir de las cuales obtener una lista de URIs desde las que capturar datos sobre ETA.

REQUISITO	ATRIBUCIÓN	PROCESO	CRÍTICO
da a actores externos de EX para su posterior uso en tareas de análisis			
Facilitar el acceso para consulta, monitorización y análisis (a menor escala que en el caso del realizado por los actores de EX) a la información recopilada	EF	Interfaz público	No

8.2.2 Requisitos organizacionales

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

8.2.3 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 tesauros y vocabularios controlados	EF Desarrollador	Grabación	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

8.3 Implementación de la solución

De acuerdo con lo expresado en 8.1, EF adopta como arquitectura la construcción de un artefacto sustentado en WP utilizado como FW. 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 WP y adaptándolo a las necesidades de EF. Descartado desde un principio para poder conseguir el mantenimiento de seguridad y funcionalidad derivado de las actualizaciones periódicas del FW, 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 la API. También se lo conoce como *plugin*, *add-on*, *conector* o *extensión*. WP utiliza extensamente estos mecanismos, con el beneficio agregado de que su funcionamiento es absolutamente independiente de la presentación, acabado

¹⁴ 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.

o formato externo de la aplicación sobre ellos construida. Su 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 WP, como publicar un *post*, cambiar de plantilla (“tema”), o mostrar una página en el panel de administración. Un *plugin* puede responder al evento al ejecutar una función PHP, que a su vez puede llevar a cabo una o varias de las siguientes acciones:
 - Modificar datos en la base de datos.
 - Enviar un mensaje de correo
 - Modificar lo que es mostrado en la pantalla del navegador (ya sea al administrador o a cualquier otro usuario)
- **Filtros.** Son funciones a través de las cuales WP 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). Los filtros actúan como interfaz o punto de paso intermedio entre la base de datos y el navegador cuando WP está generando páginas, y entre el navegador y la base de datos cuando WP está agregando registros. La mayoría de los datos entrantes y salientes en WP pasan a través de por lo menos un filtro, establecido por diseño.

Utilizando un tema. Es la solución adoptada. Se define un “tema” como un conjunto de archivos que trabajan coordinados para crear el diseño y funcionalidad de una sede WP. Cada tema puede ser diferente, ofreciendo muchas opciones para cambiar al instante la apariencia y/o la funcionalidad de la aplicación. Ofrece numerosas ventajas:

- Permite crear una apariencia única y personalizada de una aplicación.
- Permite disponer de la ventaja de una mecánica basada en plantillas (*templates*), etiquetas de plantillas (*template tags*) para generar diferentes resultados y apariencias del sitio web, lo que aproxima el funcionamiento de WP al de una estructura MVC.
- Para presentar los resultados en un tema, WP utiliza un bucle por diseño (*the WordPress Loop*). La intervención sobre él también posibilita actuar sobre los resultados obtenidos en cada consulta.
- Es posible utilizar plantillas alternativas para características específicas de la aplicación, tales como páginas por taxonomía o páginas de archivo.
- Para rápidamente cambiar entre dos disposiciones de sitio, o aprovecharse de un selector de tema o estilo Theme or style switcher para permitir a los propietarios de sitio el cambiar la apariencia de tu sitio.
- Separa los estilos de presentación y archivos de plantilla de los del sistema, de forma tal que es posible realizar intervenciones y ajustes sin cambios drásticos para la presentación visual del sitio.

Los temas tienen disponibles todos los *hooks*, las acciones y los filtros del API WP. La descripción de la mecánica de integración con el *core* de WP aparece recogida en 11.

En 14 figura una representación gráfica del *stack* o pila de capas de funcionamiento de WP. 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 WP utilizados desde EF para implementar su funcionalidad.

SUBSISTEMA WP	FUNCIONALIDAD EF
Ajax	Control de funciones ejecutadas en cliente
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)
RSS/Atom Reader	Captura automatizada de datos
Sanitization	Seguridad frente a intrusiones
Usuarios	Acceso granular a funcionalidades EF. Seguridad de acceso
WP Cookies	Control de acceso. Control de sesiones
WP Cron Jobs	Captura automatizada de datos. Mantenimiento

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 (1.2), 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 EX 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 ETA.
- 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, tales como el *cowording*, trazado de redes, etc.

Esas condiciones determinan la creación de **tres sistemas** dentro del conjunto de EF:

ABRV	SISTEMA	NOMBRE	TAREAS
SC	Captura Filtrado	Beagle	∞ Leer URIs almacenadas en entidades y construir lista
			∞ Buscar en lista información sobre ETA
			∞ Filtrar información sobre ETA y “aprender” sobre calidad de información
			∞ Guardar en borrador información encontrada
SG	Grabación	ExpoFinder Admin	μ Verificación de validez de información capturada
			μ Completado de metainformación sobre información capturada
			μ Grabación de metainformación e información capturada
SE	Exportación	ExpoFinder Export	α Construcción de tablas de datos exportables
			α Generación de formatos exportables

Clave: ∞ = automática cíclica; α = automática; μ = manual

La denominación *Beagle* se asigna en honor al buque con el que Charles Darwin recorrió medio mundo y por su significado en inglés: “sabueso”.

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.

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 WP
Beagle	Proceso automatizado cíclico que captura la información procedente de las URIs RSS y HTML, filtra la que alcanza un nivel mínimo para ser considerada válida y la graba como borrador para ponerla a disposición de <i>Grabador verificador</i>
Grabador verificador	Valida la información capturada por Beagle. Descarta aquella que pueda haber sido filtrada como adecuada por error de Beagle. Tiene derechos de modificación de la información de la cual es responsable. Carece de derechos de modificación de la información verificada y grabada por otros usuarios de su tipo. Tiene derechos de modificación de la información

NOMBRE	DESCRIPCIÓN
	capturada por Beagle
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>
Sistema	Sistema de software EF
Supervisor	Usuario que valida la información, autoriza sus modificaciones y evalúa el éxito de EF. 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
WP	API de WordPress a disposición de EF

La equivalencia entre los actores pertenecientes a los grupos de RRHH y los usuarios estándar ofrecidos por el API WP, que se utilizan en EF como mecanismo de seguridad y acceso, se expresa en la siguiente tabla.

ACTOR	TIPO DE USUARIO WP	DERECHOS ¹⁵
Usuario externo	[Sin acceso a EF]	[Sin acceso a EF]
Usuario no perteneciente a RRHH	Suscriptor	R T. Front-end
Grabador verificador	Autor	CRUD P. Front-end y back-end
Revisor	Editor	CRUD T. Front-end y back-end
Administrador	Administrador	CRUD T. Front-end y back-end
Supervisor	Editor	CRUD T. Front-end y 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	Lectura de URIs
Descripción	SC selecciona registros de tipo entidad que contengan metainformación sobre URIs RSS y/o HTML. Con la selección elabora una lista de URIs.
Actores	Beagle
Precondiciones	Deben existir registros de entidades. Debe estar grabada metainformación sobre entidades. La metainformación debe incluir URIs.
Flujo normal	SC consulta SGBD usando API WP para realizar la selección.
Flujo alternativo	SC dispara una condición de error usando API WP.
Postcondiciones	Captura de información debe disponer de una lista (array) de URIs RSS y HTML válidas.
Referencias	Captura de información

Nombre	Captura de información
Descripción	SC lee la lista de URIs. Secuencialmente, y mediante el acceso a la lista de URIs, consulta, usando el API WP, la información almacenada y extrae la información relevante.
Actores	Beagle, WP.

¹⁵ Clave de derechos de acceso a información en SGBD. C=Creación; R=Lectura, U=Actualización; D=Borrado. T=acceso a los registros generados por cualquier actor; P=acceso solo a los registros creados por el propio actor.

Precondiciones	Debe existir la lista de URIs.
Flujo normal	SC lee una URI. Mediante API WP extrae el resultado (protocolo http, estado y contenido).
Flujo alternativo	SC dispara una condición de error usando API WP.
Postcondiciones	Filtrado deba disponer de un contenido HTML o XML analizable.
Referencias	Lectura de URIs, Filtrado

Nombre	Filtrado
Descripción	SC usa la información relevante obtenida por Captura de información para transformarla en texto y aplicar el algoritmo de selección. Si la información obtiene o sobrepasa el nivel mínimo, se transfiere a Aprendizaje como válida; en caso contrario, se descarta y se transfiere a Aprendizaje como no válida.
Actores	Beagle.
Precondiciones	Debe existir la información relevante. Debe poder ser transformada desde su formato original (XML o HTML) en texto.
Flujo normal	SC lee la información relevante. Evalúa el contenido mediante aplicación de algoritmo de filtro. Transfiere resultados a Aprendizaje.
Flujo alternativo	SC dispara una condición de error usando API WP.
Postcondiciones	Aprendizaje debe disponer de un contenido en formato texto.
Referencias	Captura de información, Aprendizaje

Nombre	Aprendizaje
Descripción	SC usa la información filtrada. Si procede de Filtrado como válida, aplica algoritmo clasificador bayesiano ingenuo (CBI) y la clasifica como válida. En caso contrario, aplica el mismo algoritmo y la clasifica como no válida.
Actores	Beagle, Sistema
Precondiciones	Debe existir la información filtrada y evaluada.
Flujo normal	SC lee la información evaluada. Aplica algoritmo CBI y la clasifica.
Flujo alternativo	SC dispara una condición de error usando API WP.
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	SC recoge la información transferida por Aprendizaje. Añade la metainformación (entidad origen de URI usada para capturar la información, fecha y hora de captura,...). Graba un registro de tipo exposición en SGBD marcado como "borrador". Atribuye dicho registro al actor Grabador verificador que grabó la información de la entidad origen.
Actores	Beagle, Sistema
Precondiciones	Debe existir la información capturada, filtrada y aprendida. Debe existir la metainformación asociada.
Flujo normal	SC lee la información capturada, filtrada y aprendida. Graba una exposición en formato "borrador". Asigna la exposición a usuario correspondiente a Grabador verificador responsable de la entidad origen.
Flujo alternativo	SC dispara una condición de error usando API WP.
Postcondiciones	Grabador verificador debe disponer de un registro con la información capturada y la metainformación en formato borrador.
Referencias	Aprendizaje, Verificación

Nombre	Verificación
Descripción	Grabador verificador lee registros grabados por Grabación asignados a su usuario WP. Verifica que la información capturada es válida. Si no lo es, utiliza el flujo alternativo. Completa la metainforma-

	ción obligatoria y la que sea posible no obligatoria. Cambia el formato a "publicado". Graba el resultado.
Actores	Grabador verificador, Sistema
Precondiciones	Debe existir un registro guardado en formato "borrador" asignado al usuario WP que identifica a actor.
Flujo normal	Grabador verificador lee registro guardado. Verifica validez de datos. Caso de no ser correctos, utiliza el flujo alternativo. Completa la metainformación obligatoria y, en su caso, no obligatoria recurriendo a búsquedas en la Red y cualesquier otros procedimientos externos que sean necesarios. Cambia el formato del registro a "publicado". Graba el registro.
Flujo alternativo	Grabador verificador constata que el registro guardado no cumple las condiciones para ser válido. Lo marca como "papelera".
Postcondiciones	Grabador verificador debe disponer de un registro con la información capturada y la metainformación en formato borrador.
Referencias	Grabación, Revisión

Nombre	Grabación auxiliar
Descripción	Inducida por Verificación. Si es preciso grabar cualquier registro de entidad, persona, publicación o empresa, Grabador verificador procede a cumplimentar la información y la metainformación propia de cada caso específico. El registro, una vez guardado, es susceptible de Revisión.
Actores	Grabador verificador, Revisor
Precondiciones	Debe existir un registro guardado en formato "borrador" asignado a Grabador verificador. De las circunstancias concretas de dicho registro, debe desprenderse la necesidad de grabar información y/o metainformación adicional de tipo entidad, persona, publicación o empresa.
Flujo normal	Grabador verificador completa la información y la metainformación obligatoria y, en su caso, no obligatoria recurriendo a búsquedas en la Red y cualesquier otros procedimientos externos que sean necesarios. Cambia el formato del registro a "publicado". Graba el registro.
Flujo alternativo	No contemplado.
Postcondiciones	Revisor debe disponer de un registro grabado de tipo entidad, persona, publicación o empresa y en formato "publicado".
Referencias	Revisión

Nombre	Revisión
Descripción	Revisor lee registro grabado y en formato "publicado". 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 en formato "publicado" asignado a Grabador verificador.
Flujo normal	Revisor lee registro grabado por Grabador verificador como "publicado". Verifica validez de datos.
Flujo alternativo	Revisor constata que la metainformación grabada en el registro guardado no se adecua a la normativa sobre taxonomía y metainformación (vocabularios controlados). Emite una SC a Supervisor.
Postcondiciones	Revisor debe disponer de un registro grabado y en formato "publicado".
Referencias	Verificación, Borrado, Supervisión

Nombre	Supervisión
Descripción	Supervisor recibe una SC de Revisor. Evalúa si se trata de una actuación errónea o hay que modificar el vocabulario controlado o las normas de verificación. En el primer caso, utiliza el flujo alternativo. En el segundo, gestiona el alta de la metainformación en el vocabulario controlado o altera la normativa en consecuencia.
Actores	Grabador verificador, Revisor, Supervisor
Precondiciones	Debe existir una SC de Revisor.
Flujo normal	Supervisor recibe una SC de Revisor. Evalúa positivamente la solicitud. Realiza los cambios en el vocabulario controlado o modifica las normas

Flujo alternativo	Supervisor recibe una SC 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 una SC. Debe existir un registro grabado y en formato "publicado".
Referencias	Revisión, Borrado

Nombre	Borrado
Descripción	Por iniciativa propia durante Verificación o a instancias de Supervisor tras Supervisión, Grabador verificador borra un registro no válido.
Actores	Grabador verificador, Revisor, Supervisor, Sistema
Precondiciones	Debe existir un mensaje por mensajería interna de Supervisor.
Flujo normal	Grabador verificador recibe un mensaje por mensajería interna para borrado de un registro. Grabador verificador cambia su estado a "papelera". Grabador verificador "vacía" la Papelera de su usuario WP. Sistema borra el registro del CBI como positivo. Sistema da de alta el registro en CBI como negativo.
Flujo alternativo	SC dispara una condición de error usando API WP.
Postcondiciones	El registro erróneo debe ser eliminado con efectividad de la SGBD.
Referencias	Supervisión

Nombre	Exportación
Descripción	A través de API WP, Sistema recibe una solicitud REST de preparar una exportación de datos. La solicitud incluye parámetros sobre tipo de exportación. Sistema verifica autenticidad de la solicitud. En caso de verificación positiva, sigue flujo normal. En caso de verificación negativa, sigue flujo alternativo.
Actores	Sistema, Usuario externo
Precondiciones	Debe existir una solicitud REST. La solicitud debe superar el control de seguridad e identificación.
Flujo normal	La solicitud REST es evaluada como válida (identificación, formato de salida y tipo de datos). Sistema selecciona los datos solicitados. Sistema da formato a dichos datos de acuerdo a la solicitud. Sistema devuelve un archivo con la información solicitada.
Flujo alternativo	Sistema devuelve un archivo en formato XML-RPC con la condición de error.
Postcondiciones	El archivo devuelto debe ser legible.
Referencias	Ninguna

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

9.2 Procesos

Los procesos de EF 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 EF se expresa en la siguiente tabla.

CDU	PROCESO	SISTEMA
Lectura de URIs	Captura	Beagle
Captura de información	Captura	Beagle
Filtrado	Filtrado	Beagle
Aprendizaje	Filtrado	Beagle
Grabación	Grabación	Beagle
Grabación auxiliar	Grabación	ExpoFinder Admin
Verificación	Grabación	ExpoFinder Admin
Supervisión	Grabación	ExpoFinder Admin

CDU	PROCESO	SISTEMA
Borrado	Grabación	ExpoFinder Admin
Exportación	Exportación	ExpoFinder Export

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 WP
Mensajería interna	Disponible para todos los usuarios WP
Avisos administrativos	Emitidos por Supervisor o Administrador. Legibles para todos los usuarios WP
Control de calidad y Actividad	Evaluación automática de errores formales, estadísticas de trabajo y sesiones por usuario WP. Disponible para todos los usuarios WP
Evolución	Evaluación automática del nivel de éxito de EF. Disponible para todos los usuarios WP
Mantenimiento	Realización de tareas de optimización, inicialización y mejora de los datos guardados en SGBD. Monitorización de estado de memoria y microprocesador en servidor. Disponible solo para Administrador
Ajuste	Modificación y control de los objetivos numéricos finales de EF. Control de otros parámetros básicos de la aplicación. Disponible solo para Administrador.
Consultas	Generación de <i>queries</i> SQL para uso en front-end. Disponible solo para Administrador.

9.2.3 Procesos del *framework*

No se incluye en PDSDT una descripción detallada de los procesos inherentes al FW, que abarcan ámbitos como la seguridad, el acceso, la gestión CRUD, la gestión de plantillas, etc. Recomendamos consultar la información al respecto en la página <https://codex.wordpress.org/>.

10 Interfaces del sistema

Como ha quedado establecido en 9, existen dos grandes bloques de interfaces que permiten acceder a la aplicación EF. No todos los interfaces se corresponden con tareas. Por ejemplo, los procesos *Captura* y *Filtrado*, ambos integrantes de Beagle, al ser ejecutados por el proceso del sistema WP Cron Jobs no precisan disponer de una “cara” en pantalla o en consola. En realidad, sí que existe, a efectos de depuración, una salida STD para ambos empleando el interfaz CLI de PHP, pero no forma parte de una especificación formal, sino de un simple mecanismo de control.

En esta descripción de interfaces NO SE INCLUYE INFORMACIÓN sobre elementos propios del FW WP (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 FW WP, pero añade pantallas y elementos de menú propios, alterando algunas de las informaciones ofrecidas por defecto por WP por razones de seguridad¹⁷. 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*: *entidades*, *personas*, *publicaciones*, *empresas*, *exposiciones* 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: *Mensajería*, *Control de calidad*, *Actividad*, *Evolución* y (de uso exclusivo de Administrador) *Mantenimiento*.

Las pantallas CRUD incluyen, básicamente, un complejo formulario con los diversos tipos de campos que almacenan la información y la metainformación 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 (Geonames, Fundación Getty,...).
- Campos georreferenciados. Incluyen un mapa (en esta versión de PDSDT procedente de

¹⁶ Esa expresión (público) no es especialmente exacta. Por razones inherentes a la estructura del proyecto EX, desde un principio se ha adoptado la decisión de que solamente puedan acceder al front-end los usuarios WP autorizados con el tipo “Suscriptor”. Por tanto, para consultar datos (no ya para intervenir sobre ellos) es preciso estar identificado. El usuario puramente externo no tiene acceso a ninguna información. Esta situación previsiblemente evolucionará con el tiempo.

¹⁷ Por ejemplo, se sustituye la pantalla WP conocida como “Desktop” por una de elaboración propia, que incluye información de los procesos *Mensajería interna*, *Avisos administrativos*, *Control de calidad*, *Actividad*, *Evolución* y *Mantenimiento*.

Google Maps), un campo de sólo lectura con las coordenadas y un campo editable con la dirección, sincronizados ambos con un marcador de posición que se muestra en el mapa.

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

Cada formulario CRUD incluye un mecanismo “nonce” de verificación de origen por razones de seguridad e integridad de la información. Algunos formularios (como el de entidades) 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.

- Campos cuyo contenido sea una URI¹⁸. Verificación de su adecuación a norma. En el caso de las URIs RSS y HTML del registro de tipo entidad, Sistema realiza una consulta remota usando el API WP para verificar que la dirección conduce a un extremo cuya respuesta http es válida y el contenido de sus datos es coherente con la solicitud (XML en caso de URIs RSS y HTML en caso de URIs de ese tipo).
- Campos cuyo contenido sea un número de teléfono. Verificación de que solo contiene caracteres numéricos.
- Campos que siguen la norma **2 * ;**. Aplicable a nomenclaturas geográficas basadas en Geonames. Verifica que contengan 3 bloques de texto separados por punto y coma (;).
- Campos que siguen la norma **1 * :**. Aplicable a campos autorreferenciados. Verifica que contenga un bloque numérico separado de un bloque de texto por un signo de dos puntos seguido de un espacio en blanco (:).

En 14 se incluyen volcados de modelos de pantalla del interfaz back-end.

10.2 Front-end

En el momento de elaborar la versión actual de PDSDT, se encuentra aún en fase de desarrollo y no ha sido completado. Para la elaboración se ha seguido una plantilla basada en el FW CSS3 + HTML5 + JavaScript Bootstrap 3.0, 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 presentación. Simple *splash* de conexión. Con información resumida sobre las estadísticas de grabación de los distintos tipos de CPs y sus etiquetas más significativas, así como sobre el número de URIs HTML y RSS/Atom activas. Además, información gráfica sobre:
 - Grado de finalización del proyecto.
 - Porcentaje de hombres y mujeres registrados en los CP de tipo *Personas*.
 - Las 5 relaciones más fuertes entre distintos tipos de registro (diagrama Sankey).

¹⁸ Según el esquema oficial de URIs de IANA. Véase <http://www.iana.org/assignments/uri-schemes/uri-schemes.xhtml>.

- Mapa con las 30 ciudades más activas (con más exposiciones o entidades).
- Menú Proyecto. Incluye información resumida sobre EX, EF, equipo de trabajo y estado del proyecto EF resumido en estadísticas más relevantes.
- Menú Documentos. Incluye información sobre EX, EF y otro material de carácter técnico pero de tono divulgativo.
- Menú Consultas. Incluye dos submenús.
 - Banco de datos. Acceso detallado a los campos *no restringidos* de los distintos CPs. Incluye listado y ficha de detalle, con mapa de ubicación en caso de que sea posible ofrecerlo, y enlaces a otros elementos y CPs. Permite búsqueda de texto libre sobre cualquier campo, filtrable por tipo de CP.
 - Análisis con tres pestañas, que muestran los datos obtenidos de las consultas estadísticas y de agrupación mencionadas en 9.2.2 en tres formatos:
 - Tabla simple exportable como XLS, CSV, PDF o al portapapeles del SO, ordenable y buscable.
 - Tabla manipulable por el usuario para crear un conjunto de referencias cruzadas (pivotable) y mostrarlo en formato gráfico o de tabla simple. Imprimible y exportable a formato CSV.
 - Gráfico de barras, líneas o áreas para representar la tabla simple. Dispone de subgráfico para seleccionar zona de interés.
- Menú Usuario. Permite acceder al interfaz Back-end con los derechos inherentes al perfil del usuario que haya iniciado la sesión y consultar el estado de la evaluación (datos individuales).
- Barra de estado, con información de tiempos de acceso y recursos empleados para cada página, así como enlaces utilitarios y obligatorios (información legal y créditos y licencias) más acceso al informe de errores y la evaluación (véase 13.1.3 y 13.1.4).
- Visita guiada. El usuario puede elegir realizar un “paseo virtual” automatizado por los elementos principales del interfaz. La visita puede ser abandonada o retomada a voluntad en cualquier momento.

11 Componentes SW

La estructura de componentes SW de EF viene determinada por el análisis de requisitos, la arquitectura y los procesos inherentes al proyecto. Véase al respecto 2.2, 8 y 9.

Como principio rector, dada la naturaleza científica y académica de los proyectos EF y EX, se ha adoptado el de usar única y exclusivamente componentes no propietarios de software libre.

11.1 Componentes básicos

La aplicación se construye empleando **PHP**, acrónimo de *Hypertext Preprocessor*. Se trata de un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML en forma de scripts ejecutables desde el extremo servidor.

WordPress es, en palabras de sus autores, “es una poderosa plataforma de publicación personal, y viene con una gran cantidad de características incorporadas, diseñadas para hacer tan fácil, placentera y atractiva como sea posible la experiencia de publicar en Internet”. Se distribuye bajo licencia estándar GPL.

MySQL es un SGBD relacional, multihilo y multiusuario, cuya compañía matriz desde enero de 2008 es subsidiaria de Sun Microsystems, y ésta a su vez de Oracle Corporation desde abril de 2009. Está desarrollado como software libre en un esquema de licenciamiento dual.

MariaDB es un SGBD derivado de MySQL con licencia GPL y absolutamente compatible en cuanto a instrucciones y lenguaje con el mismo. Está desarrollado por el fundador de MySQL y la comunidad de desarrolladores de software libre. En el momento de redactar la versión 1.1 de PDSDT, EF está funcionando empleando MySQL como gestor de bases de datos, pero está prevista la migración a MariaDB.

Ubuntu es un sistema operativo basado en GNU/Linux y que se distribuye como software libre. Su patrocinador, Canonical, es una compañía británica propiedad del empresario sudafricano Mark Shuttleworth. Ofrece el sistema de manera gratuita, y se financia por medio de servicios vinculados al sistema operativo.

EF utiliza ampliamente las posibilidades de la POO mediante el empleo de clases, subclasses y funciones. Toda la nomenclatura de código está normalizada mediante el prefijo “`cs1_`” en variables de alcance global, funciones, clases y constantes. La grafía del código sigue las normas de estilo de WP en cuanto a estilos de redacción (Allman preferentemente, K&R ocasionalmente).

Las versiones en uso de los componentes en el momento de redactar el presente artefacto PDSDT son las siguientes.

COMPONENTE	VERSIÓN
PHP	5.6.11-1ubuntu3.1
WordPress	4.4.1 es-ES
MySQL	5.6.27-0ubuntu1
Ubuntu	Ubuntu Server 15.04 "Vivid Vervet". Kernel 3.14.32-xxxx-std-ipv6-64 #5 SMP Tue Sep 8 17:41:55 CEST 2015 x86_64

11.2 Estructura SW

Como ya se ha mencionado en 8, EF se construye como extensión sobre el FW WP. La construcción se realiza adoptando el formato de “tema”, con la denominación específica **CSL**.

El tema consiste en un directorio, que se instala en la raíz física de la instalación de WP, en la

subcarpeta `wp-content/themes`. No se recoge en PDSDT la organización normalizada de un tema estándar WP, pero sí se detalla la composición propia de CSL.

Archivos obligatorios: `functions.php`, `index.php`, `styles.php`

`functions.php`. Contiene las inclusiones de librerías, la lógica del programa, la inicialización y las llamadas a las clases que generan los *custom post types*.

`index.php`. Contiene la lógica de arranque para front-end. Se organiza en tres bloques: footer, sidebar y WP Loop. Se ha optado por insertar TODA la estética del interfaz en este único archivo en vez de emplear múltiples plantillas.

`styles.css`. Contiene los estilos CSS3 básicos. Puede ser complementado con otras hojas en cascada.

El directorio (o carpeta) CSL se organiza de acuerdo al siguiente esquema.

```

assets
  css
  img
  js
  keywords
  logs
  tools
includes
  languages
  libraries
    Readability
    XMLRPC
    b8
  cs1_custom_admin_posts_helper.php
  cs1_custom_ajax_helper.php
  cs1_custom_dashboard_chat.php
  cs1_custom_dashboard_helper.php
  cs1_custom_data_retrieval_helper.php
  cs1_custom_definitions_helper.php
  cs1_custom_global_helper.php
  cs1_custom_idle_autologout.php
  cs1_custom_local_avatars.php
  cs1_custom_log_helper.php
  cs1_custom_mail_helper.php
  cs1_custom_post_helper.php
  cs1_custom_profiling_tools.php
  cs1_custom_scripts_helper.php
  cs1_custom_secure_xmlrpc_server.php
  cs1_custom_shortcodes_helper.php
  cs1_custom_survey_helper.php
  cs1_custom_terms_management_helper.php
_cs1_cron_beagle.php
changelog.txt
functions.php
index.php
license.txt
readme.txt
screenshot.png
style.css
    
```

Los directorios figuran en color rojo. El contenido de las carpetas se describe a continuación.

CARPETA	CONTENIDO
assets	Cualquier elemento auxiliar en el funcionamiento del programa
css	Hojas de estilo
img	Imágenes de uso en EF
keywords	Archivos textuales con palabras clave utilizadas en el proceso Filtrado
logs	Archivos textuales con los registros de las operaciones automatizadas cíclicas regidas por WP Cron Jobs, como Beagle
includes	Librerías y archivos con prototipos de clases y funciones de uso común. Los archivos, en la raíz de esta carpeta, prefijados por “cs1_” contienen dichas clases

11.3 Algoritmos y librerías básicas

11.3.1 Algoritmos

Los procesos de EF emplean determinados algoritmos. A continuación los describimos.

Búsqueda de palabras clave en texto

Se basa en la existencia de una lista de palabras clave, clasificada por idiomas, que se hace coincidir con el detectado (a través de la estructura XML o HTML y no de la cabecera HTTP) en el documento a analizar. El proceso es el siguiente.

- Mediante el API WP para CURL y para captura de RSS y/o Atom se obtiene el contenido de la URI explorada.
- Al contenido se le aplica un filtro para la extracción de contenido principal basado en Reability (ver lista de carpetas más arriba). Se obtiene en codificación UTF-8.
- Al contenido principal así extraído se le aplica un filtro basado en el API WP KSES para eliminar caracteres diacríticos y reducir los elementos no analizables. Se obtiene un texto depurado.
- Sobre el texto depurado se aplica estadística TF-IDF (*Term Frequency - Inverse Document Frequency*), basada en la Asunción de Luhn y en IDF de Spärck Jones mediante peso binario para TF y peso unario para IDF. Está previsto adaptar un sistema basado en normalización doble K para TF y máximo de frecuencia inversa para IDF.
- El resultado se compara con la lista de palabras válidas mediante intersección de *arrays*.
- Si el valor total de los *arrays* intersecados iguala o supera el umbral mínimo, el documento se considera válido.

Evaluación del grado de éxito del proyecto

Su algoritmo es la aplicación de lo expresado en 4.1. Es posible consultar más elementos empleados en Verificación de la calidad en 13.

Detección y recuento de errores formales

Se basa en una compleja consulta SQL sobre las tablas estándar suministradas por el API WP, que no se reproduce en PDSDT por su elevada extensión (650 líneas en formato normalizado). Básicamente busca ausencias de metainformación obligatoria o de cumplimiento de normas (véase 10.1).

El sistema calcula una tasa media de error por usuario para todos aquellos registros que al menos tengan uno, y una tasa individual, para cada Grabador verificador. Los cálculos finales se emplean también en la verificación de la calidad (véase 13).

Se clasifican en los siguientes tipos:

- Duplicados: Posible título duplicado
- Obligatorios: Ausencia de uno ó más campos obligatorios
- Geoname: Error en uno o más campos de la convención de Geonames para entidades locales (regla 2 * ;)
- Autorreferencia: Error en uno o más campos de la convención de autorreferencia (regla 1

* :)

- Taxonomías: Ausencia de una o más taxonomías obligatorias

Clasificación bayesiana ingenua

Se emplea la librería de Tobias Leupold B8 (véase 11.2). Un clasificador de Bayes ingenuo asume que la presencia o ausencia de una característica particular no está relacionada con la presencia o ausencia de cualquier otra característica, dada la clase variable. Los CBI se pueden entrenar de manera muy eficiente en un entorno de aprendizaje supervisado¹⁹, que es modelo que tal librería emplea, de manera que obtiene una “opinión probabilística” acerca de la validez o ausencia de ella de determinado texto capturado. Se utiliza en el contexto de los procesos Filtrado (Beagle) y Borrado.

11.3.2 Librerías

Como se ha dicho anteriormente, en ningún apartado de 11 se hace mención de los elementos que forman parte del API de WP, usado en el proyecto como FW. EF emplea dos categorías de librerías: para ejecución en servidor y para ejecución en cliente. Todas las usadas disponen de licencias que las caracterizan como SW libre, pudiendo ser dispuestas de acuerdo a los términos establecidos en cada caso.

- Para ejecución en servidor (PHP)
 - **PHP Readability**
 - Release: 0.21. 12 de febrero de 2014.
 - Licencia: MIT.
 - Autor: feelinglucky.
 - Descripción: Port de la librería homónima de Arc90 (<http://graceco.de/readability/>).
 - Función: Selección de texto relevante en página HTML.
 - **B8**
 - Release: 0.6.1. 12 de marzo de 2014.
 - Licencia: GPL 2.
 - Autor: Tobias Leupold.
 - Descripción: Clasificador Bayesiano Ingenuo basado en PHP y MySQL.
 - Función: Aplicar CBI a textos recuperados por Beagle y filtrados.
- Para ejecución en cliente (Javascript)
 - **PivotTable.js**
 - Release: 2.0.2. 12 de enero de 2016.
 - Licencia: MIT.

¹⁹ Técnica matemática que posibilita deducir o detectar la existencia de una función (o “patrón oculto”) a partir de datos suministrados a modo de entrenamiento, que suelen ser pares de objetos (vectores, habitualmente), que trabajan mediante sus componentes: una representa los datos de entrada y la otra el resultado esperado de salida.

- Autor: Nicolas Kruchten.
- Descripción: Formateador de *arrays* como tablas de referencias cruzadas configurable por el usuario final
- Función: Parte de la funcionalidad ofertada por front-end (Inteligencia empresarial).
- **DataTables**
 - Release: 1.10.10. Vía CDN.
 - Licencia: MIT.
 - Autor: SpryMedia Ltd.
 - Descripción: Formateo de tablas a partir de *arrays*.
 - Función: Parte de la funcionalidad ofertada por front-end (Inteligencia empresarial).
- **C3**
 - Release: 0.4.10. 17 de marzo de 2015.
 - Licencia: MIT.
 - Autor: Masayuki Tanaka.
 - Descripción: Trazado de gráficos a partir de *arrays*. Depende de D3.js <= 3.5.0.
 - Función: Parte de la funcionalidad ofertada por front-end (Inteligencia empresarial).

12 Gestión de riesgos

Se entiende por riesgo en el contexto PDSDT cualquier acontecimiento futuro o no predecible que pueda afectar de forma negativa a EF. 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 EF 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, EF 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.

FACTOR RIESGO	ACCIONES PREVENTIVAS
---------------	----------------------

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

²¹ BOEHM 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 MySQL. Uso del recurso proporcionado por la instalación RAID 5 del servidor.
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 instalación RAID 5 del servidor.
Acceso no deseado a información	Uso del API WP para asegurar la identificación de cualquier usuario externo.
Falta de competencia adecuada de miembros de RRHH	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 XML-RPC, facilitadas por API WP, mediante clave asimétrica única por usuario autorizado para la extracción y operación.

13 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 CL como desde el de Desarrollador.

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

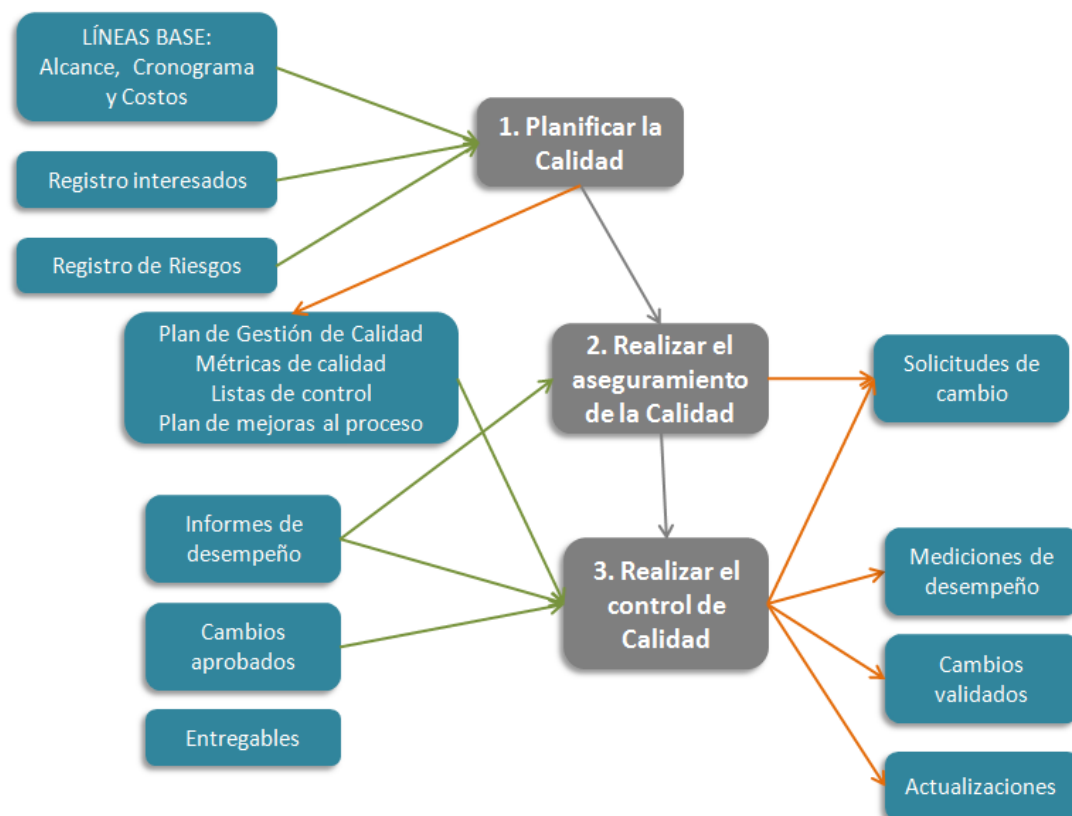


Ilustración 8. Procesos GC según PMI

EF 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 EF. 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 científica y académica tanto de EF como de su matriz EX, además de la cualidad del proyecto de ser un PDS, 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 WP	10.1
	7 Información en tiempo real. Número de errores por actor. En interfaz back-end. Accesible a todos los usuarios WP	10.1
	8 Suministro de información. Errores formales individuales por actor. En interfaz back-end. Accesible a todos los usuarios WP	10.1
Condicionantes adicionales	9 Cobertura geográfica mínima. Dado el alcance del proyecto EX (ámbito europeo), se verifica que las zonas geográficas previstas son cubiertas	N/A
	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

13.1 Evaluación del nivel de calidad

Con el fin de conocer el grado de calidad alcanzado por EF en su estado en el momento de entregar el artefacto PDSDT actual, a continuación se detallan los procesos GC considerados, las acciones verificadoras realizadas y los resultados obtenidos. En primer lugar se citan los requisitos GC mensurables [RGC] mediante técnicas cuantificadoras, las herramientas empleadas, su ubicación en EF y los resultados a la fecha de entrega de la versión actual de PDSDT. En la tabla final y en la descripción de las herramientas se especifican los condicionantes de calidad y el nivel de cumplimiento [NC] ($0 \geq \text{nivel de cumplimiento} \leq 1$).

13.1.1 Herramientas de medición de calidad

Nombre	1 Grado global de finalización
Procedencia	Acción 1
Herramienta	Cálculo global de nivel de finalización
Procedimiento	Detallado en 4.1
Nivel debido	54%
Nivel alcanzado	60%
Cumplimiento NC	1

Nombre	2 Número de errores global y por actor
Procedencia	Acción 7

Herramienta	Cálculo global de nivel de errores y cálculo individual por actor
Procedimiento	Detallado en 11.3
Nivel debido	5%
Nivel alcanzado	9%
Cumplimiento NC	0,6

Nombre	3 Cobertura geográfica mínima
Procedencia	Acción 9
Herramienta	Informe específico (ver 15)
Procedimiento	Especificado por CL. Verificación de que está cubierto el 100% de los municipios >= 20.000 habitantes en España
Nivel debido	100%
Nivel alcanzado	94%
Cumplimiento NC	0,9

Nombre	4 Disponibilidad SW
Procedencia	Acción 10
Herramienta	Informe específico (ver 15)
Procedimiento	Especificado por Desarrollador. Nivel de disponibilidad de acceso al servidor y funcionalidad de aplicación de 99,9%
Nivel debido	99,9%
Nivel alcanzado	99,8%
Cumplimiento NC	1

Nombre	5 Calidad SW
Procedencia	Acción 11
Herramienta	Informe específico (ver 15)
Procedimiento	Especificado por Desarrollador. Complejidad ciclomática por LLOC <= 0,35
Nivel debido	0,35
Nivel alcanzado	0,38
Cumplimiento NC	0,9

13.1.2 Matriz de control de calidad

PROCESO	ACCIÓN			CUMP
Planificación	1	Restricciones y suposiciones	Verificado por CL y herramienta 1	1
Planificación	2	Requisitos	Verificado por CL	1
Planificación	3	Plan de fases	Verificado por CL	1
Planificación	4	Tabla de esfuerzo	Verificado por CL y herramienta 1	1
Aseguramiento	5	Reuniones Sprint	Verificado por CL	1
Control	7	Errores por actor	Verificado por CL y herramienta 2	0,6
Condicionantes adicionales	9	Cobertura geográfica	Verificado por CL y herramienta 3	0,9
Condicionantes adicionales	10	Disponibilidad SW	Verificado por Desarrollador y herramienta 4	1
Condicionantes adicionales	11	Calidad SW	Verificado por Desarrollador y herramienta 5	0,9
GRADO TOTAL DE CALIDAD ALCANZADA A FECHA DE ENTREGA DE PDSDT				0,93

En consecuencia, y a la vista de los resultados obtenidos, el grado de cumplimiento alcanzado (93% de los objetivos previstos) se considera **satisfactorio** desde el punto de vista de PDSDT.

13.1.3 Informe de errores

La necesidad de implementar un adecuado seguimiento de los errores se solventa mediante el establecimiento de un formulario de notificación, categorizado y con campos de cumplimentación por defecto (solo lectura) o selectiva, que es enviado automáticamente por correo electrónico a la administración del sistema, y al que se puede acceder desde cualquier página del interfaz *front-end*.

También se ha establecido un formulario de contacto simple que permite, con un procedimiento similar, obtener información o enviar sugerencias a la administración del sistema.

13.1.4 Evaluación

EF establece un mecanismo de evaluación mediante un formulario, que se muestra de forma singularizada para cualquier página del interfaz *front-end*, que conserva la valoración que cada usuario de tipo suscriptor (con acceso a EF) realiza de ella, mediante un sistema de puntuación simple (mostrado en la tabla siguiente) que se basa en cuatro aspectos, con un rango posible entre un mínimo de 0 (cero) y un máximo de 36. Los resultados se conservan de manera individualizada y EF calcula los promedios y una valoración global del aplicativo. A tal información solamente tendrá acceso el administrador del sistema.

ASPECTO	DESCRIPCIÓN	PUNTUACIÓN
Diseño	El diseño de los elementos gráficos y textuales de la página es correcto y adecuado a su funcionalidad.	De 0 a 9
Funcionalidad	La funcionalidad que debe ofrecer la página sirve para cumplir su cometido en el conjunto de acciones propias de la aplicación.	De 0 a 9
Utilidad	La utilidad que ofrece la página es adecuada (sirve para cumplir un propósito necesario para el buen funcionamiento de la aplicación).	De 0 a 9
Velocidad	La velocidad con la que la página se genera u ofrece resultados es aceptable.	De 0 a 9

14 Anexo gráfico

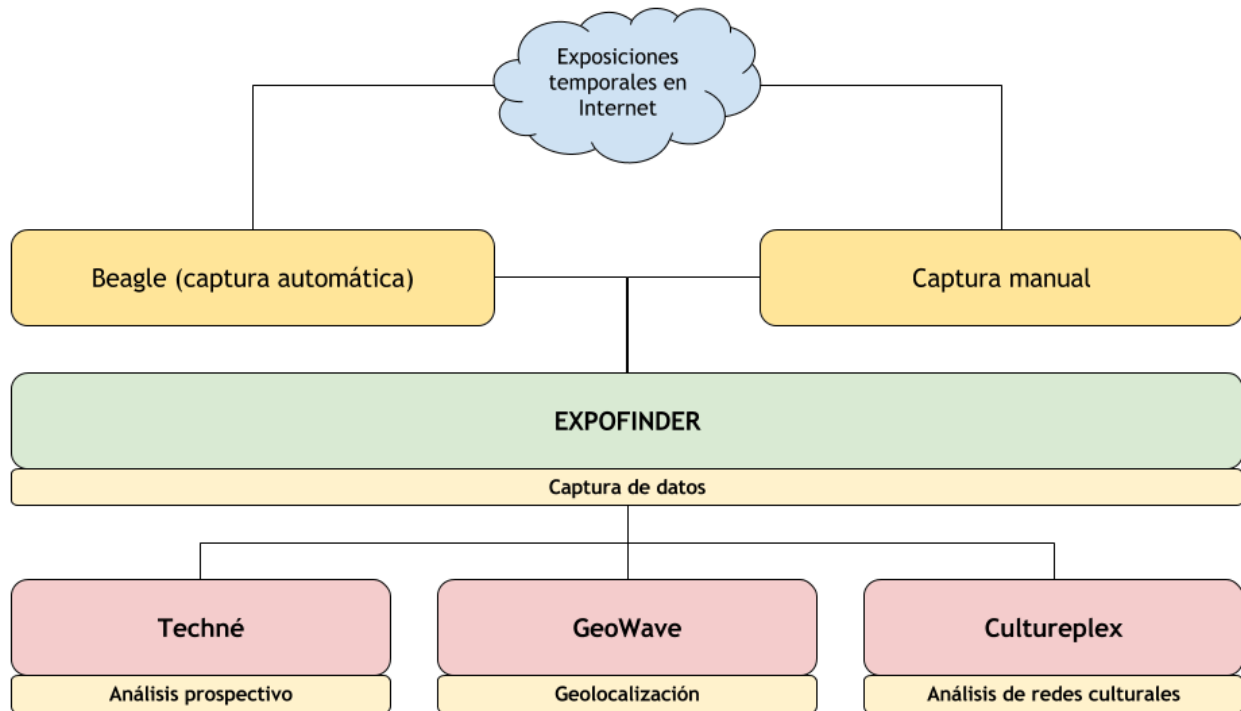


Ilustración 9. EF como subproyecto de EX. Mapa conceptual

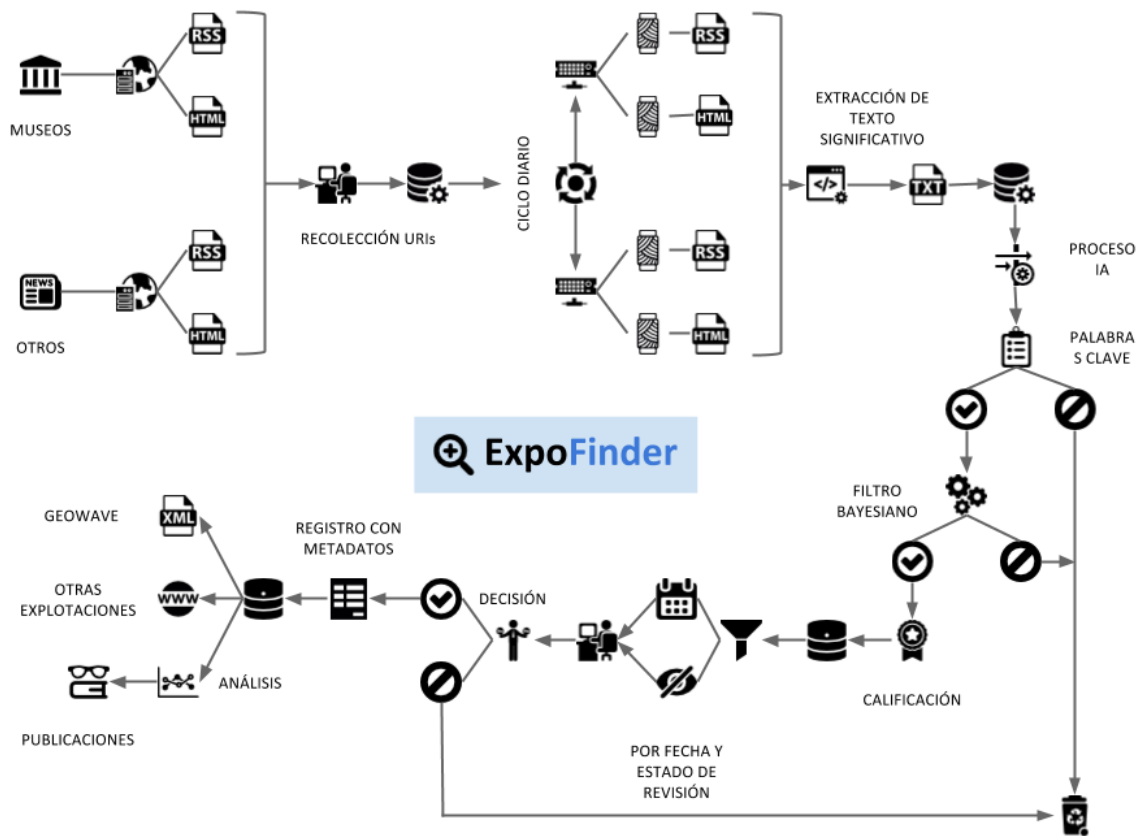


Ilustración 10. EF: Esquema de flujo simplificado. Mapa conceptual

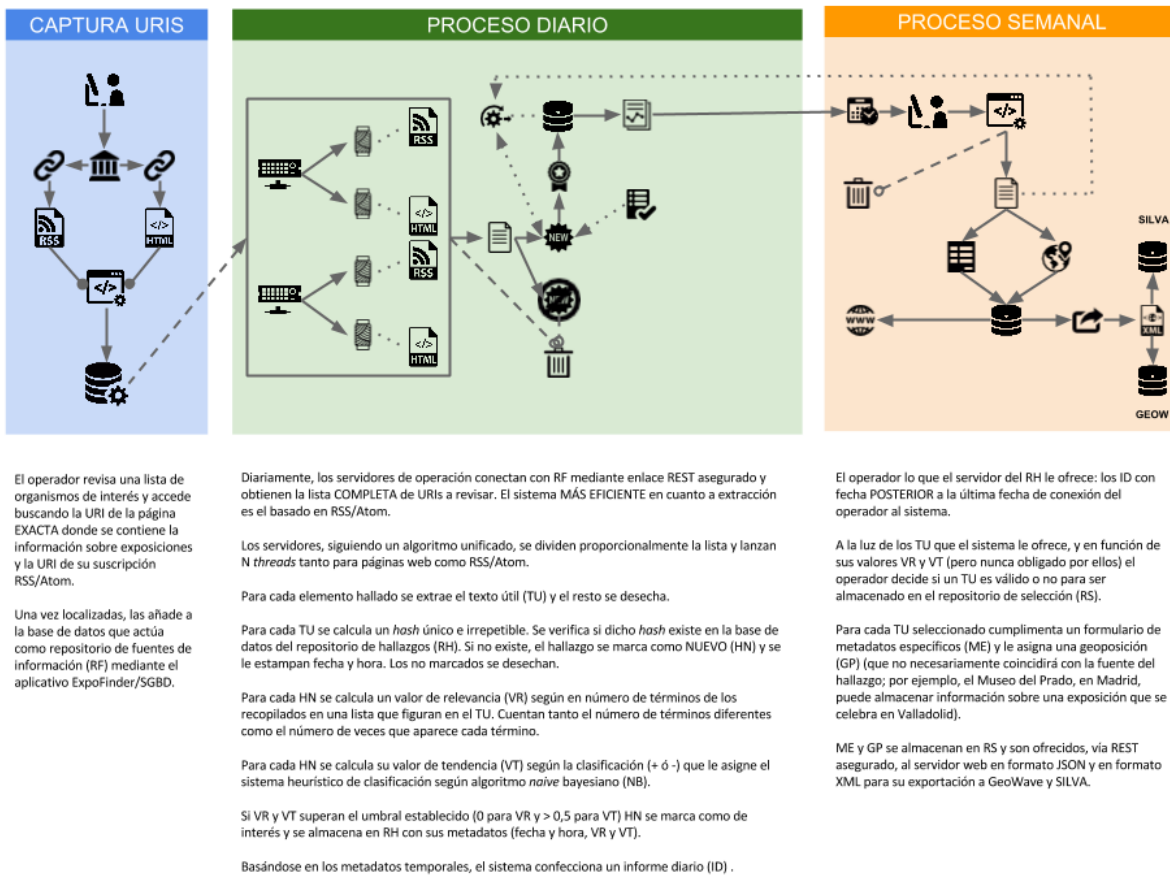


Ilustración 11. EF. Esquema procesual. Mapa conceptual

index.php WP_USE_THEMES

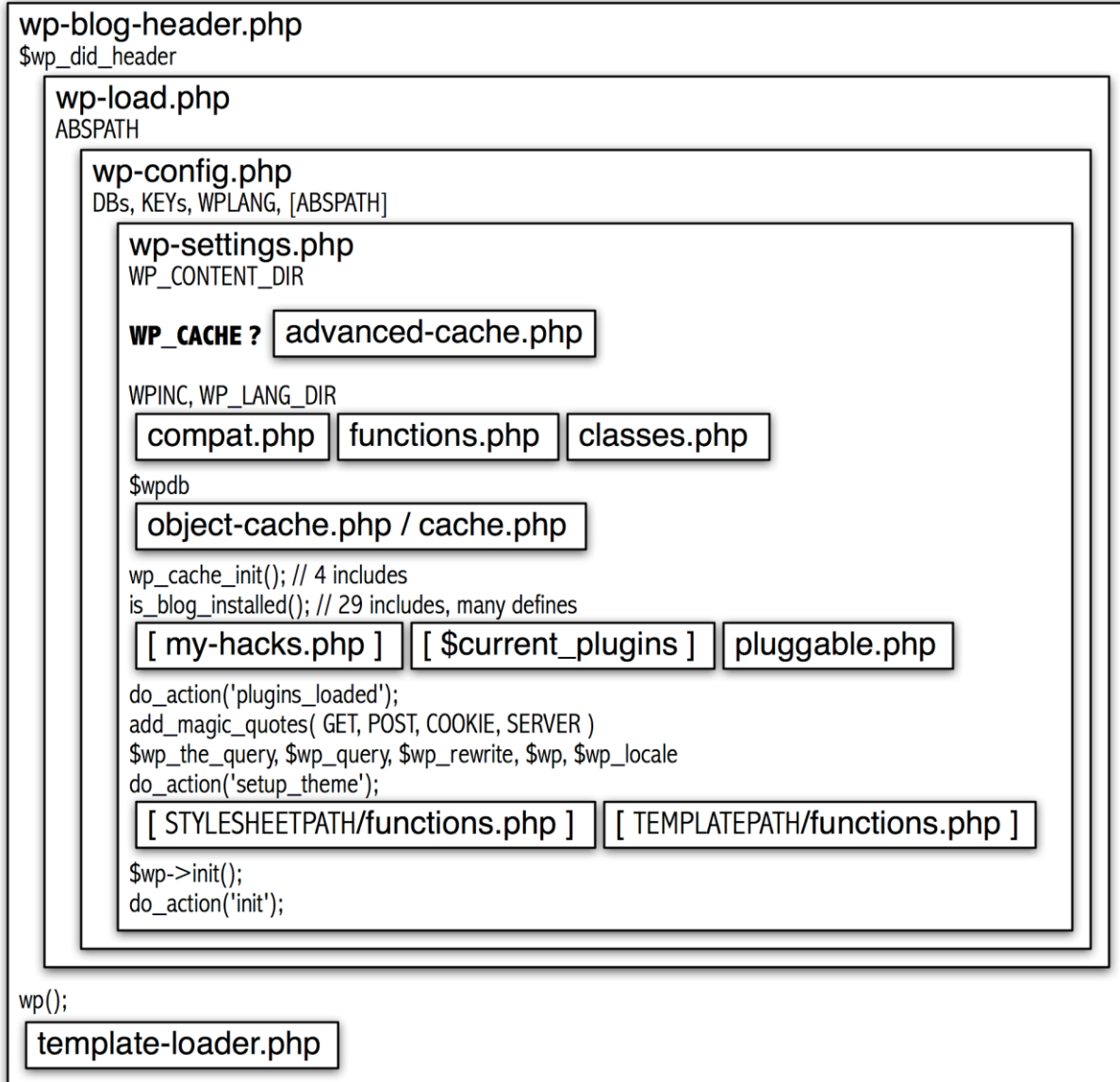


Ilustración 12. Stack de WordPress

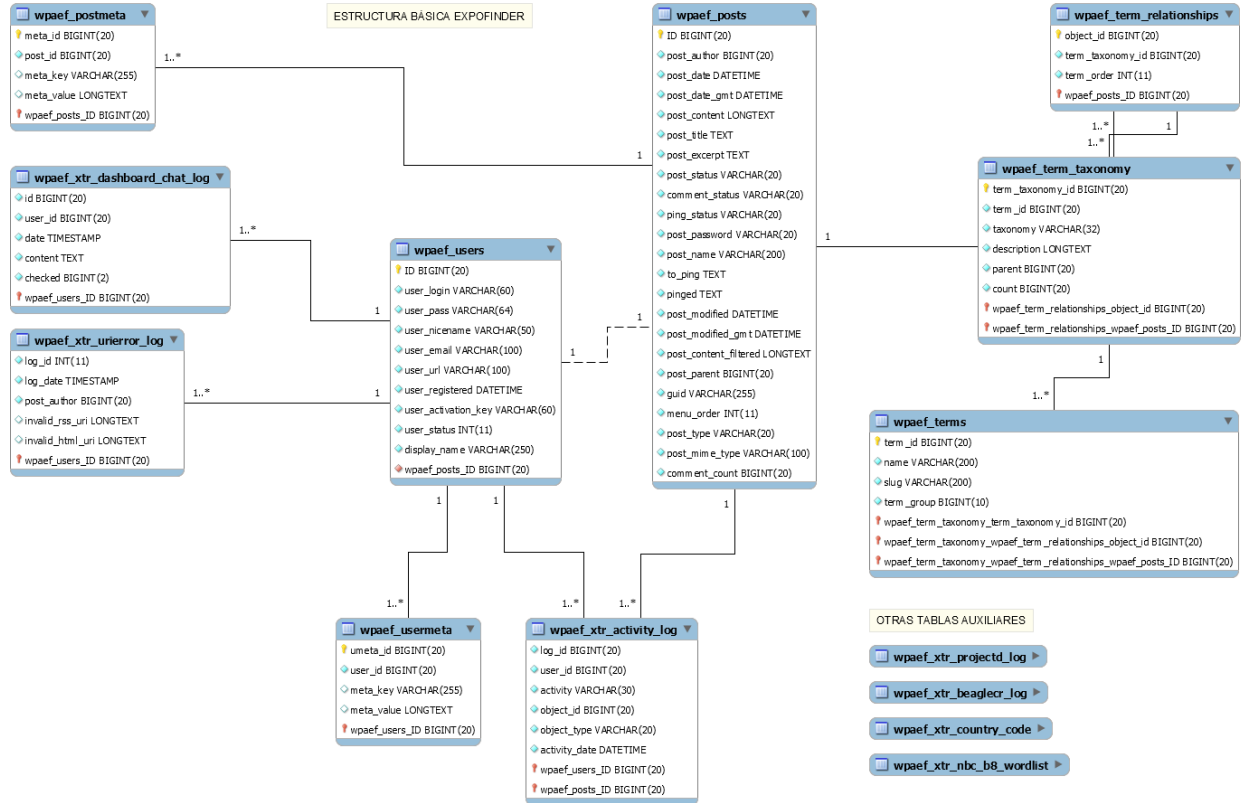


Ilustración 13. EF. Modelo de datos

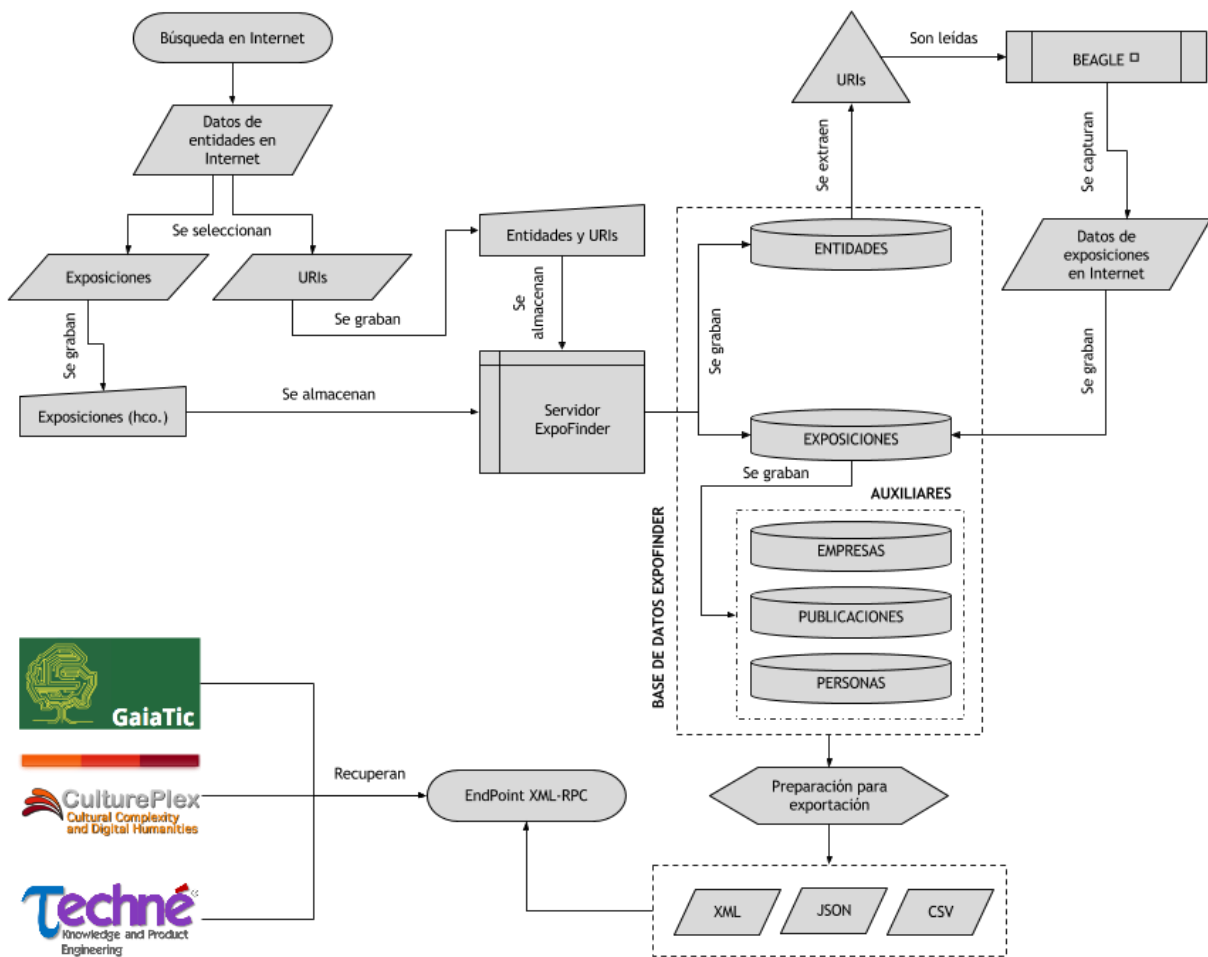


Ilustración 14. Relaciones entre los 3 sistemas principales de EF

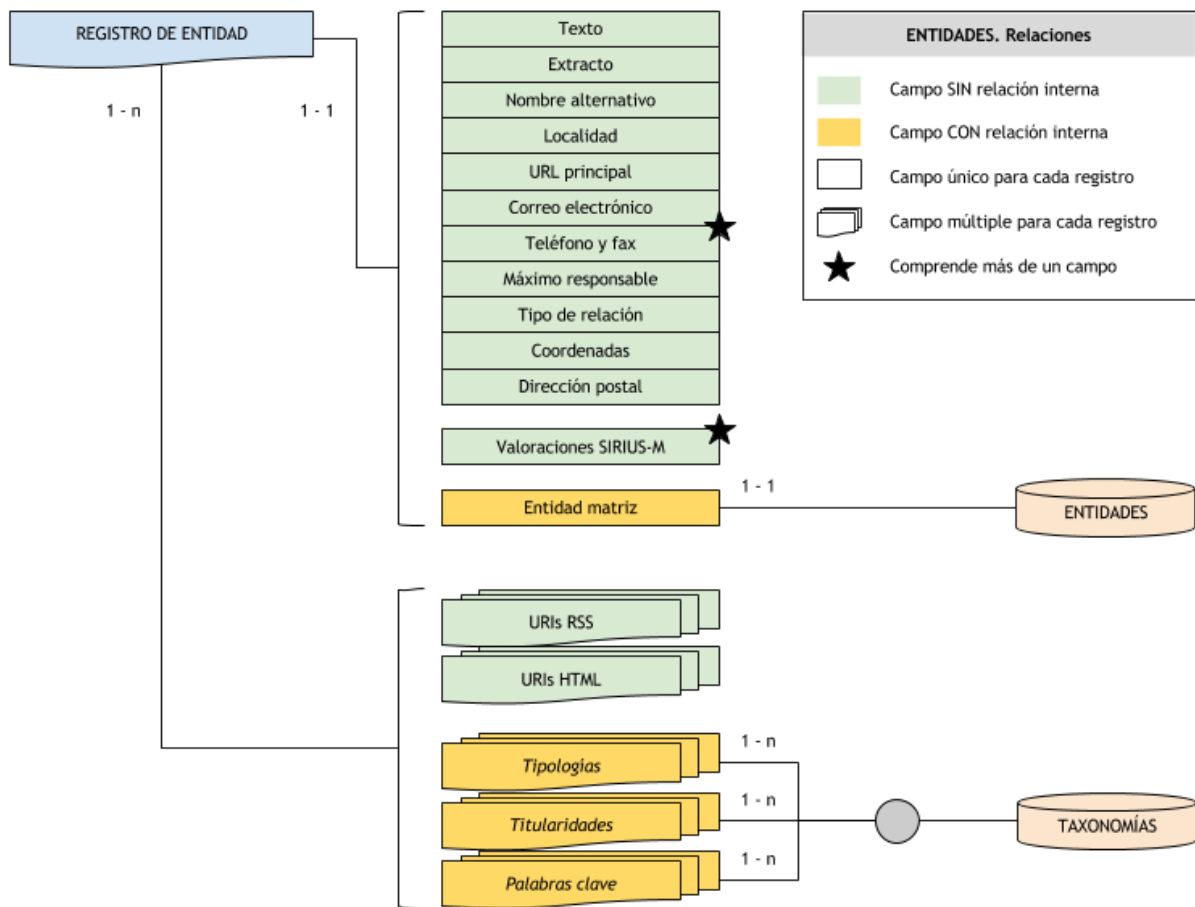


Ilustración 15. Registro ENTIDADES. Relaciones

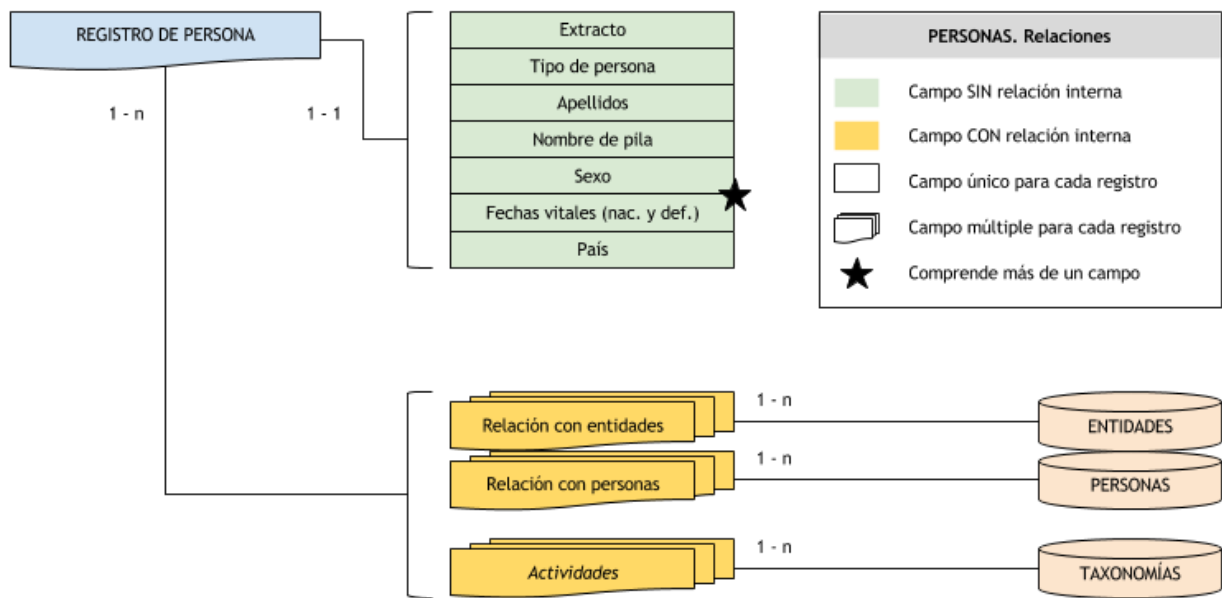


Ilustración 16. Registro PERSONA. Relaciones

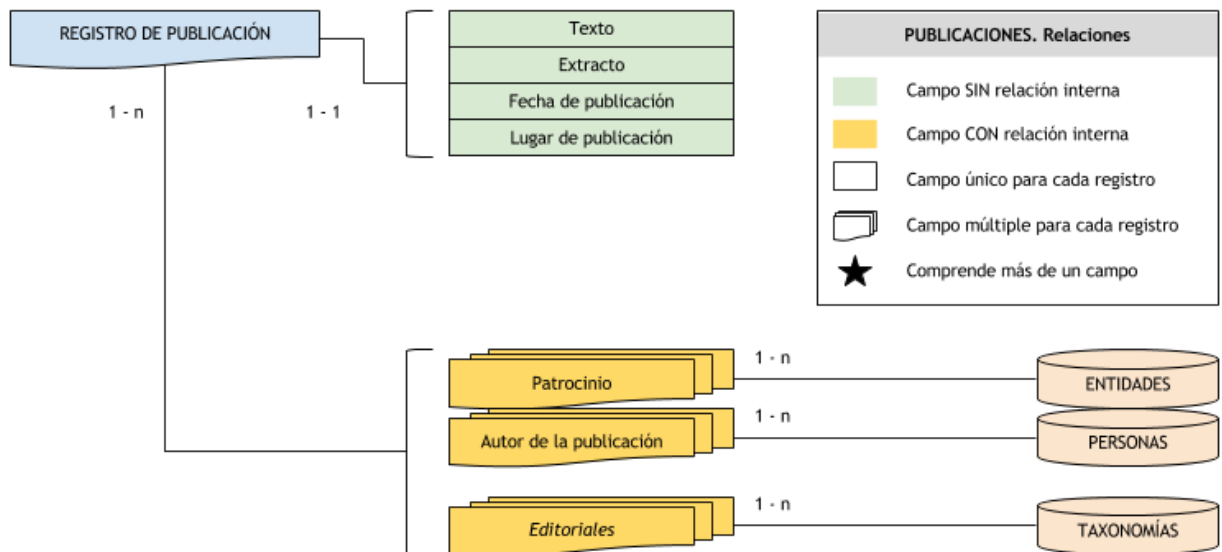


Ilustración 17. Registro PUBLICACIONES. Relaciones

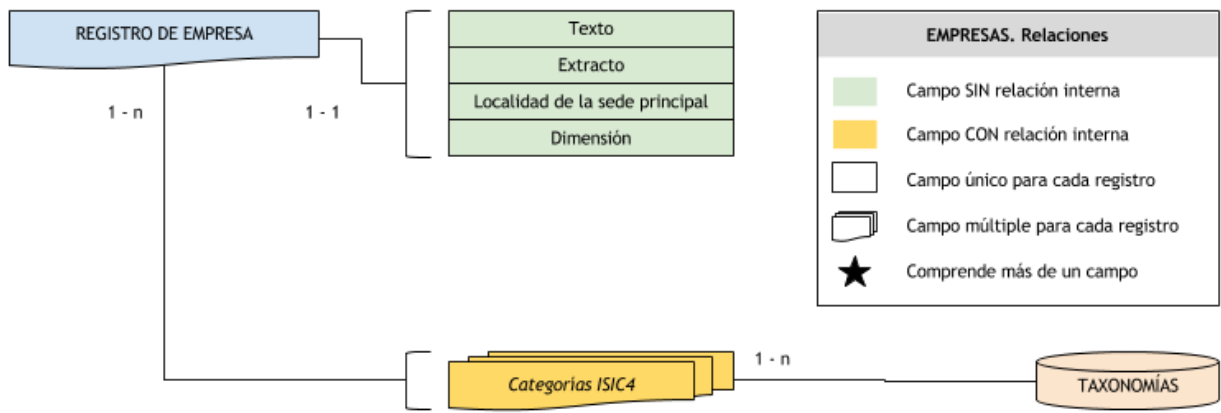


Ilustración 18. Registro EMPRESA. Relaciones

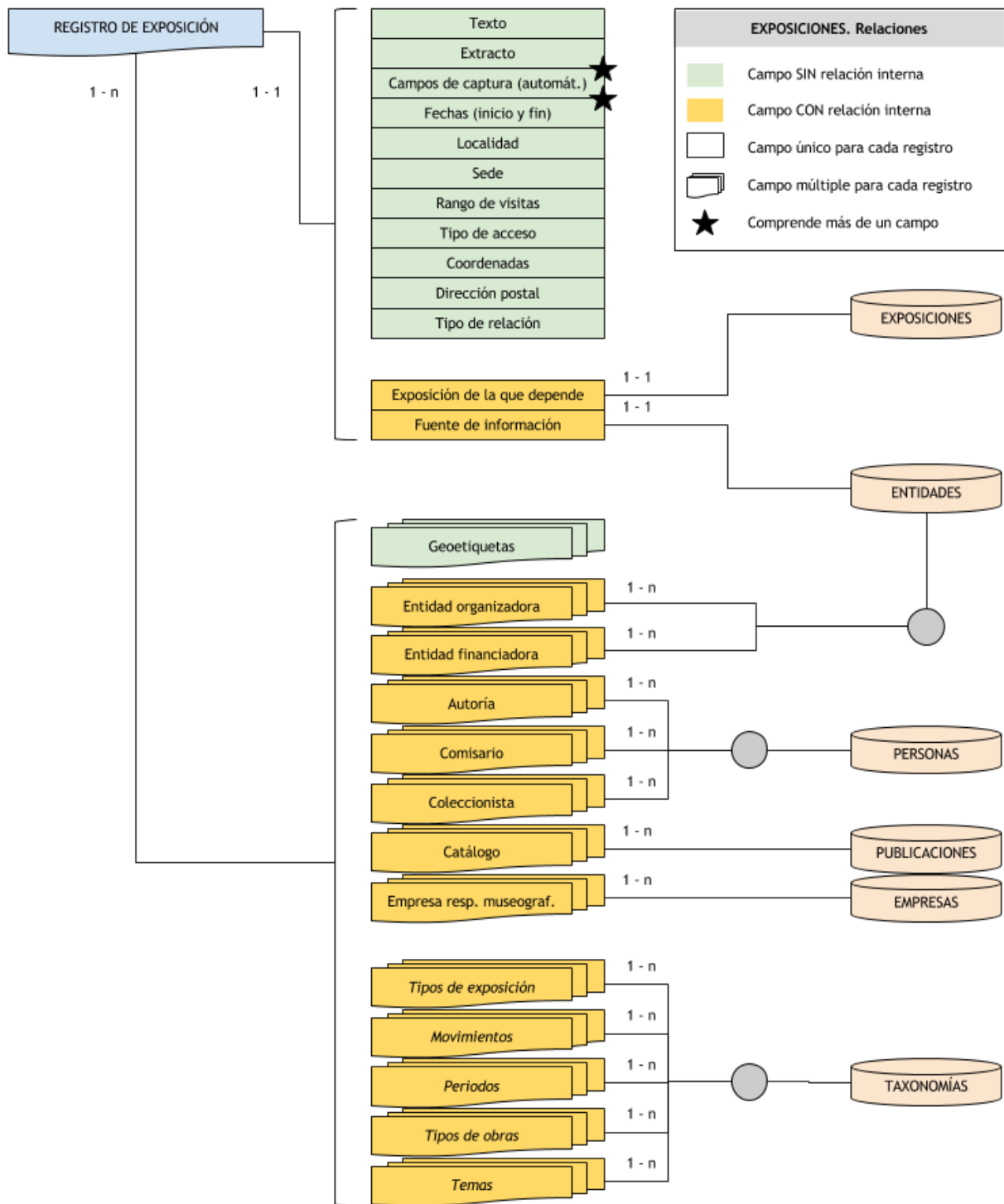
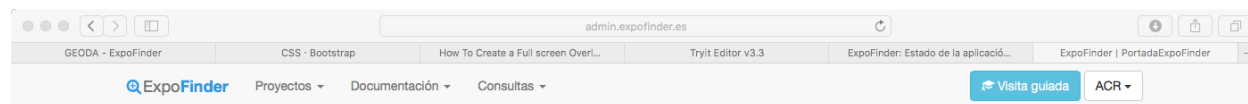


Ilustración 19. Registro EXPOSICIÓN. Relaciones

The screenshot displays the ExpoFinder back-end interface. On the left is a dark sidebar menu with options like 'Inicio', 'Actualizaciones', 'Estado de la aplicación', 'Ajustes', 'Entradas', 'Medios', 'Páginas', 'Comentarios', 'Consultas', 'Entidades', 'Personas', 'Publicaciones', 'Empresas', 'Exposiciones', 'Apariencia', 'Plugins', 'Usuarios', 'Herramientas', 'Ajustes', and 'Cerrar menú'. The 'Estado de la aplicación' menu item is highlighted with a green box, and a green arrow points from it to the main content area. The main content area is titled 'ExpoFinder: Estado de la aplicación' and shows a notification for Bárbara Romero Ferrón. Below this is a navigation bar with tabs: 'La aplicación', 'Mensajería', 'Control de calidad', 'Actividad', 'Evolución', and 'Mantenimiento'. To the right of these tabs are three colored boxes with labels: a yellow box for 'Acceso a todos los usuarios', a red box for 'Acceso a Administrador', and a green box for 'Acceso a todos los usuarios. Control de calidad y mensajería'. The main content area contains a section titled 'Qué es @ExpoFinder' with a detailed paragraph about the project's origin and goals. Below this is another section titled 'Estado actual del proyecto Exhibitium' which features a Gantt chart showing the project's progress. The Gantt chart has a vertical axis with tasks: 'Análisis de requerimientos', 'Análisis de riesgos', 'Modelo de estructura de datos', 'Pruebas funcionales', 'Ajustes', 'Puesta en marcha', 'Captura automática y manual', 'Exportación REST', 'GeoWave', 'Sylva', and 'Clusters con KDD'. Each task is represented by a horizontal bar of a specific color (blue, red, orange, yellow) indicating its duration and status.

Ilustración 20. Interfaz back-end



ExpoFinder 1.7.6

Aplicación distribuida que utiliza el paradigma cliente-servidor para la gestión de la información específica del proyecto **Exhibitium**. Grupo de investigación **iArHis_LAB**. Departamento de Historia del Arte. Universidad de Málaga.

El proyecto **ExpoFinder** es una de las más destacadas aportaciones del grupo de investigación **iArHis_LAB** al mundo de las herramientas especializadas en la gestión del conocimiento en el mundo de las Humanidades Digitales.

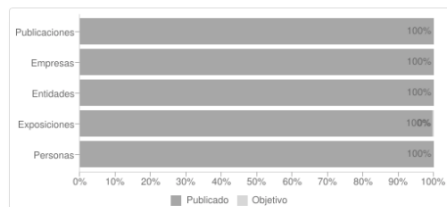
Ayuda y manuales

En el menú de **Proyectos** podrá encontrar información resumida sobre la finalidad de **Exhibitium** y **ExpoFinder**.

El menú de **Documentación** contiene los manuales esenciales que explican el manejo de las herramientas disponibles y los principios de funcionamiento del sistema.

Le sugerimos que, en caso de ser su primera visita, utilice los servicios prestados por la **Visita guiada**, que irá mostrándole las principales opciones disponibles le asesorará sobre el uso y la utilidad de cada opción. Puede abandonar o retomar la visita en cualquier momento que desee.

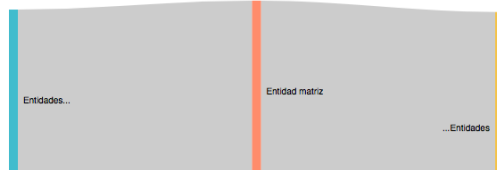
Grado de finalización del proyecto



Personas registradas por sexo



Las 5 relaciones más fuertes



Las 30 ciudades más activas



Ilustración 21. Interfaz front-end



Ilustración 22. Diagrama UML genérico. Modelo de casos de uso

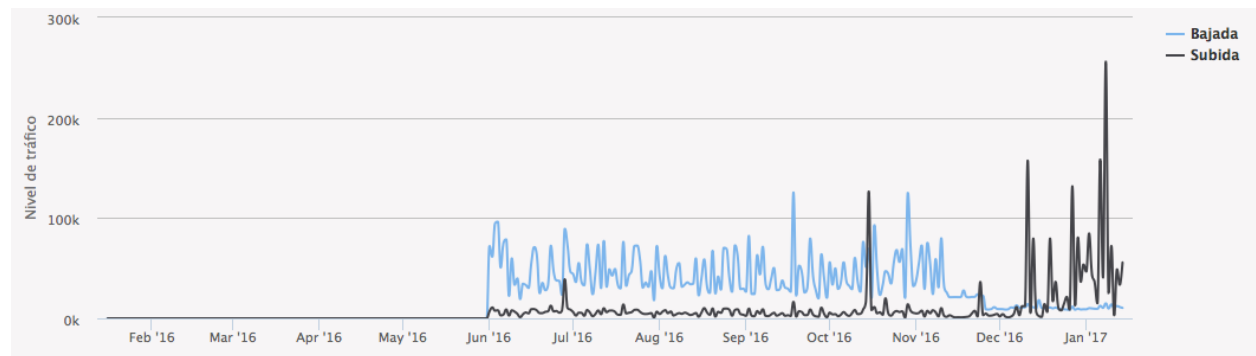


Ilustración 23. Servidor admin.expofinder.es. Tráfico últimos 12 meses

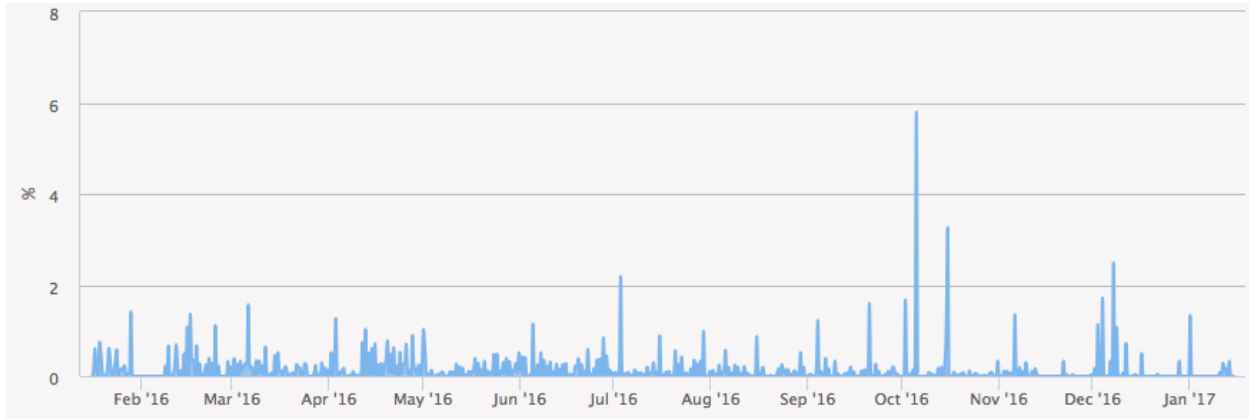


Ilustración 24. Servidor admin.expofinder.es. Uso CPU últimos 12 meses

15 Anexo documental

15.1 Informe sobre cobertura geográfica

Datos analizados directamente de SGBD a 30 de diciembre de 2015. Datos sobre municipios procedentes de Instituto Nacional de Estadística (INE). Padrón consolidado 2014.

	MUNICIPIOS TOTALES (TOT)						MUNICIPIOS CON ENTIDADES (CIENT)						MUNICIPIOS SIN ENTIDADES (SIENT)					
	< 5K	5K-10K	10K-15K	15K-20K	20K-25K	>25K	< 5K	5K-10K	10K-15K	15K-20K	20K-25K	>25K	< 5K	5K-10K	10K-15K	15K-20K	20K-25K	>25K
Andalucía	512	105	45	29	27	53	204	49	22	20	22	50	308	56	23	9	5	3
Aragón	708	10	4	5	1	3	708	10	4	5	1	3						
Asturias	47	10	12	2	1	6	47	10	12	2	1	6						
Canarias	27	19	7	9	8	18	27	19	7	9	8	18						
Cantabria	82	10	4	1	1	4	81	10	4	1	4	1						
Castilla y León	2.190	35	4	4	2	13	1.679	30	4	3	1	13	511	5		1	1	
Castilla-La Mancha	843	36	19	5	1	15	300	19	11	4		14	542	17	8	1	1	1
Cataluña	740	87	33	23	12	52	739	86	33	23	12	52	1	1				
Ceuta						1						1						
Extremadura	348	25	4	3		7	344	25	4	3		7	4					
Galicia	200	57	23	12	4	18	18	11	7	4	3	8	182	46	16	8	1	10
Islas Baleares	27	17	8	3	1	11	27	17	8	3	1	11						
La Rioja	165	5	2		1	1	165	5	2		1	1						
Madrid	98	32	8	7	5	29	17	12	2	5	3	28	81	20	6	2	2	1
Melilla						1						1						
Murcia	9	6	6	7	3	14	9	6	6	7	3	14						
Navarra	250	12	5	2	1	2	250	12	5	2	1	2						
País Vasco	181	29	12	11	3	15	179	29	12	11	2	15	2					1
Valencia	386	57	21	13	22	43	382	57	21	13	22	43	4					
	6.813	552	217	136	93	306	5.176	407	164	115	82	291	1.636	145	53	21	11	15

15.2 Informe sobre LOC y complejidad SW

Herramienta de cálculo: PHP LOC 2.1.3 (Sebastian Bergmann). Fecha de captura: 14 de enero de 2017.

Directories	10
Files	42
Size	
Lines of Code (LOC)	26651
Comment Lines of Code (CLOC)	4046 (15.18%)
Non-Comment Lines of Code (NCLOC)	22605 (84.82%)
Logical Lines of Code (LLOC)	7033 (26.39%)
Classes	2592 (36.85%)
Average Class Length	83
Minimum Class Length	4
Maximum Class Length	370
Average Method Length	8
Minimum Method Length	0
Maximum Method Length	108
Functions	3596 (51.13%)
Average Function Length	11
Not in classes or functions	845 (12.01%)
Cyclomatic Complexity	
Average Complexity per LLOC	0.38
Average Complexity per Class	39.87
Minimum Class Complexity	2.00

Maximum Class Complexity	203.00
Average Complexity per Method	5.11
Minimum Method Complexity	1.00
Maximum Method Complexity	58.00
Dependencies	
Global Accesses	2677
Global Constants	2249 (84.01%)
Global Variables	190 (7.10%)
Super-Global Variables	238 (8.89%)
Attribute Accesses	1839
Non-Static	1839 (100.00%)
Static	0 (0.00%)
Method Calls	914
Non-Static	859 (93.98%)
Static	55 (6.02%)
Structure	
Namespaces	0
Interfaces	0
Traits	0
Classes	31
Abstract Classes	2 (6.45%)
Concrete Classes	29 (93.55%)
Methods	268
Scope	
Non-Static Methods	233 (86.94%)
Static Methods	35 (13.06%)
Visibility	
Public Methods	189 (70.52%)
Non-Public Methods	79 (29.48%)
Functions	307
Named Functions	299 (97.39%)
Anonymous Functions	8 (2.61%)
Constants	119
Global Constants	85 (71.43%)
Class Constants	34 (28.57%)

15.3 Normativa aplicada

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

15.4 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.

T. McFarlin. *Using WordPress for Web Application Development: An Introduction*. 2013. <http://code.tutsplus.com/tutorials/using-wordpress-for-web-application-development-an-introduction--wp-33828>. Consultado 20-1-2015.