



UNIVERSIDAD NACIONAL
AUTÓNOMA DE
MÉXICO

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

POSGRADO EN CIENCIA E INGENIERÍA DE LA COMPUTACIÓN

SISTEMA MANEJADOR DE PAQUETES DE PUESTA EN OPERACIÓN

TESIS

QUE PARA OPTAR POR EL GRADO DE:

MAESTRO EN INGENIERÍA (COMPUTACIÓN)

PRESENTA:

PEÑA GRANADOS FRANCISCO JAVIER

M. EN C. MA. GUADALUPE E. IBARGÜENGOITIA G.

FACULTAD DE CIENCIAS

PROGRAMA DE POSGRADO EN CIENCIA E INGENIERÍA EN LA COMPUTACIÓN

MÉXICO, D. F. AGOSTO 2013



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

Agradecimientos

A la UNAM por ser mi segunda casa y permitirme crecer profesionalmente.

*A mi tutora M. en C. Ma. Guadalupe E. Ibarquengoitia G.
por aconsejarme y alentarme durante la realización de este
trabajo y durante el transcurso de la maestría.*

A mis compañeros de generación.

A mis sinodales por leer esta tesis y mejorarla con sus comentarios.

Contenido

Agradecimientos	2
Resumen.....	7
Introducción	8
Contexto.....	8
Problema	9
Objetivo.....	9
Contribuciones	9
Organización de la tesis.....	10
1. Conceptos generales	11
1.1 MoProSoft	11
1.1.1 Descripción General	11
1.2 Norma ISO/IEC 29110.....	14
Pequeñas organizaciones - VSE (Very Small Entities)	14
Historia de la norma ISO/IEC 29110.....	15
Grupo de Perfil Genérico.....	16
Norma ISO/IEC 29110.....	17
1.3 Arquitectura de Software.....	18
1.3.1 Estilo arquitectónico	20
1.3.2 Patrón Arquitectónico.....	20
1.3.3 Patrones de Diseño	20
1.4 Tipos de datos de acuerdo a su estructura y almacenamiento	27
1.4.1 Datos estructurados	27
1.4.2 Datos no estructurados.....	28
1.4.3 Datos semiestructurados	29
2. Paquete de Puesta en Operación - PPO	31
2.1 Qué es un Paquete de Puesta en Operación.....	31

2.2 Estructura de un PPO	31
2.3 Paquetes existentes actualmente	32
2.4 Importancia de los Paquetes de Puesta en Operación	34
Construcción y Pruebas unitarias	34
Verificación y Validación	34
Integración y Pruebas.....	35
Administración del proyecto	35
Arquitectura y diseño detallado.....	35
Entrega de producto	36
Análisis de Requerimientos	36
Control de Versiones	36
Auto-evaluación	37
3. Sistema Manejador de Paquetes de Puesta en Operación - SMPPO	38
3.1 Definición de Sistema Manejador	38
3.2 Tipos de Sistemas Manejadores.....	38
Sistema Manejador de Bases de Datos - SMBD	38
Sistema Manejador de Aprendizaje - LMS (Learning Management System).....	39
Sistema Manejador de Contenido - CMS (Content Management System).....	39
3.3 Funciones de un Sistema Manejador	40
3.4 Sistema Manejador de Paquetes de Puesta en Operación - SMPPO.....	41
3.4.1 Definición	41
3.4.2 Beneficios de contar con un SMPPO	42
4 Definición de Requerimientos.....	43
4.1 Definición	43
4.2 Objetivo	43
4.3 Requerimientos Funcionales y No Funcionales.....	43
Funcionales.....	43
No Funcionales	44
4.4 Casos de uso	45

Contenido

4.4.1 Diagrama General.....	45
4.4.2 Detalle de casos de uso	46
5. Diseño.....	61
5.1 Definición de la arquitectura.....	61
5.2 Diagramas de paquetes.....	63
5.3 Persistencia	66
5.3.1 Diagrama Entidad-Relación	67
5.3.2 Diagrama Relacional.....	68
6. Construcción del Sistema Manejador de Paquetes de Puesta en Operación	69
6.1 Tecnología a usar.....	69
6.1.1 JavaServer Faces.....	69
6.1.2 PrimeFaces	70
6.2 Aplicación de las tecnologías al proyecto	71
7. Principales funcionalidades de la aplicación	73
7.1 Búsqueda de Implementaciones de Paquetes	73
7.2 Crear Implementaciones de Paquetes - Editor, Administrador	74
7.3 Consultar Implementaciones de Paquetes - Visitante Editor, Administrador	76
7.4 Actualizar Implementaciones de Paquetes - Editor, Administrador	76
7.5 Eliminar Implementaciones de Paquetes - Editor, Administrador.....	77
7.5 Retroalimentación de buenas prácticasde Implementaciones de Paquetes - Editor, Administrador	78
7.6 Otras funcionalidades	79
7.6.1 Añadir implementación de paquete a configuración - Editor y Administrador	79
7.6.2 Idiomas de la aplicación - Todos los usuarios	79
Conclusiones	81
Trabajo Futuro.....	82
Apéndices.....	83
Apéndice A. Detalle de Casos de uso	83
Apéndice B. Diccionario de datos	92

Contenido

Glosario	97
Referencias Bibliográficas	99

Resumen

Hoy en día las pequeñas y medianas empresas constituyen el tipo dominante de organizaciones de negocios en el mundo (el porcentaje varía dependiendo del país), no obstante los estándares ISO/IEC no se ajustan a las necesidades de este tipo de organizaciones.

Para las pequeñas organizaciones (VSE - Very Small Entities, de 1 a 25 empleados) los miembros del ISO/IEC decidieron crear la norma 29110, la cual se adapta a sus necesidades específicas. Con el fin de ayudar a la implementación de esta norma se crearon Paquetes de Puesta en Operación - PPO, los cuales ayudan a definir directrices y explican con más detalle los procesos que componen la norma.

Los Paquetes de Puesta en Operación pueden encontrarse en formatos PDF o DOC, el cual resulta poco apropiado para su difusión, realizar cambios, retroalimentación por parte de los usuarios, entre otros.

Con el fin de solventar los inconvenientes anteriores el objetivo del trabajo es crear un Sistema Manejador de Paquetes de Puesta en Operación - SMPPPO, la cual es una aplicación que permite a las organizaciones desarrolladoras de software buscar, crear, consultar, actualizar y eliminar implementaciones de Paquetes de Puesta en Operación, además de retroalimentar las buenas prácticas para organizar sus propios procesos.

Introducción

Contexto

De acuerdo con la Organización para la Cooperación y el Desarrollo Económico (OCDE) en su reporte Pequeñas y Medianas Empresas y Perspectiva Empresarial (2005), las Pequeñas y Medianas Empresas constituyen el tipo dominante de organizaciones de negocios en el mundo, contabilizando por encima de 95% y hasta 99% del total de los negocios dependiendo del país. El reto para los gobiernos que integran la OCDE, es proveer un ambiente de negocios que apoye la competitividad de esta gran cantidad de negocios heterogéneos y que promueva una fuerte cultura empresarial. (1)

No obstante el gran número de pequeñas y medianas empresas, la mayoría de los estándares ISO/IEC no se ajustan a sus necesidades, ya que están estructurados para ser aplicables a empresas grandes, por lo que a las VSE se les complica su implementación ya que no pueden ampliar el número de empleados, presupuesto o tiempo, y no ven un beneficio en establecer procesos de ciclo de vida de software, lo que ocasiona que no sean reconocidas como productoras de software de calidad y se encuentren aisladas de algunas actividades económicas.

Con el fin de solventar las necesidades anteriores, miembros de la ISO/IEC decidieron crear un grupo de trabajo dando como resultado la norma ISO/IEC 29110. Para ayudar a la implementación de la norma y proporcionar orientación sobre la aplicación efectiva, se crearon Paquetes de Puesta en Operación - PPO, los cuales ayudan a definir directrices y explicar con más detalle los procesos que componen los perfiles de la norma.

Se define Paquete de Puesta en Operación como:

"Un Paquete de Puesta en Operación es un conjunto de artefactos desarrollados para facilitar la implementación de un conjunto de buenas prácticas, del marco de trabajo seleccionado en una pequeña organización (VSE)." (2)

Problema

Actualmente (Febrero del 2012) sólo existen Paquetes de Puesta en Operación para el perfil básicos los cuales pueden ser encontrados y descargados en formato PDF o DOC en la página del editor de la norma y en los sitios de algunas universidades, éstos paquetes son creados por un número reducido de personas los cuales no tienen retroalimentación por parte de las pequeñas organizaciones (VSE)¹ sobre el resultado de la implementación.

Aunque por el momento sólo exista una versión de cada paquete de puesta en operación, es de esperar que conforme la norma ISO/IEC 29110 sea difundida e implementada por un número mayor de pequeñas organizaciones, estas versiones de los paquetes aumenten y sean consideradas características particulares, con lo que la organización y/o selección de los paquetes en el formato actual resultará una tarea complicada, por lo que será necesario contar con una herramienta que ayude a estructurar estos paquetes para facilitar la adopción.

Objetivo

Desarrollar un Sistema Manejador de Paquetes de Puesta en Operación que permita a las organizaciones desarrolladoras de software buscar, crear, consultar, actualizar, eliminar, migrar las implementaciones de Paquetes de Puesta en Operación, retroalimentar las buenas prácticas para organizar sus propios procesos y mejora de las implementaciones de los paquetes.

El sistema a desarrollar deberá ser modificable (3), escalable (4) y funcional (5).

Contribuciones

La principal contribución de la presente tesis es proveer una aplicación Web que permita a las pequeñas organizaciones tener acceso de forma rápida a las implementaciones de paquetes, permitiendo realizar búsquedas por criterios (paquete que se quiere implementar, metodología, requisitos, palabras clave, autores) lo que facilita la selección del paquete que quiera poner en marcha.

¹ Puede consultarse la definición en el glosario.

A los autores y/o editores de los paquetes, la aplicación les ayuda a poner a disposición sus paquetes y actualizaciones de los mismos, así como obtener retroalimentación de las pequeñas organizaciones de su trabajo.

Organización de la tesis

La tesis está organizada en siete capítulos de la siguiente forma:

Capítulo 1 – **Conceptos generales**. Presenta conceptos necesarios para el presente trabajo como: la problemática existente de las pequeñas organizaciones desarrolladoras de software y el surgimiento de la norma ISO/IEC 29110 como un intento de solucionar el problema. Se introduce el concepto de arquitectura de software, los tipos de datos de acuerdo a su estructura y almacenamiento.

Capítulo 2 – **Paquetes de Puesta en Operación - PPO**. Se describe qué es un PPO y brevemente la importancia de cada uno de ellos.

Capítulo 3 – **Sistema Manejador de Paquetes de Puesta en Operación - SMPPO**. Presenta definiciones de Sistemas Manejadores a partir de los cuales se define Sistema Manejador de Paquetes de Puesta en Operación.

Capítulo 4 – **Definición de Requerimientos**. Se define el proyecto, se plantean el objetivo y se dan las características que debe cumplir la aplicación.

Capítulo 5 – **Diseño**. Se define la arquitectura, se justifica la forma de persistencia que se usará en la aplicación y se incluyen diferentes diagramas como son: entidad relación, relacional y de paquetes.

Capítulo 6 – **Construcción del Sistema Manejador de Paquetes de Puesta en Operación**. Se exponen brevemente las tecnologías utilizadas.

Capítulo 7 – **Principales funcionalidades de la aplicación**. Finalmente se muestra como realizar cada uno de los puntos del objetivo de la tesis.

1. Conceptos generales

1.1 MoProSoft

Tomando en cuenta que la industria del software se ha convertido en una actividad económica muy importante, la Secretaría de Economía en 2002 crea el Programa de Desarrollo de la Industria de Software, PROSOFT, el cual tiene como objetivo el fortalecimiento de la Industria del Software en México. Dentro de las estrategias definidas está crear una norma que contemple: un modelo de procesos (Qué procesos), un modelo de capacidades de procesos (Qué evaluar) y un método de evaluación (Cómo evaluar). (6)

Para el desarrollo de este proyecto la Asociación Mexicana para la Calidad de Ingeniería de Software (AMCIS) convocó a personas con experiencia y conocimiento de procesos y calidad de software, de este manera en 2002 surge la primera versión del Modelo de Procesos para la Industria del Software (MoProSoft). En mayo del 2003 fue generada la versión 1.1 y en agosto del 2005 la versión 1.3, finalmente el 15 agosto del 2005 es publicada en el Diario Oficial de la Federación la declaratoria de la Norma Mexicana NMX-059/01-NYCE-2005 referente a MoProSoft. (7)

1.1.1 Descripción General

El objetivo principal de MoProSoft es proporcionar a la industria mexicana y a las áreas internas dedicadas al desarrollo y/o mantenimiento de software, un conjunto integrado de las mejores prácticas basadas en los modelos y estándares reconocidos internacionalmente (6). Surge bajo las premisas de ser fácil de entender y aplicar, una adopción no costosa y que sea la base para la adopción de otros modelos.

El Modelo de Procesos para la Industria del Software (MoProSoft) tiene tres categorías de procesos: Alta Dirección, Gerencia y Operación. Estas categorías reflejan la estructura de una organización:

Categoría	Descripción de la Categoría	Procesos	Propósito del Proceso
Alta Dirección	Aborda las prácticas de Alta Dirección relacionadas con la gestión del negocio. Proporciona los lineamientos a los procesos de la Categoría de Gerencia y se retroalimenta con la información generada por ellos	Gestión de Negocio	Establecer la razón de ser de la organización, sus objetivos y las condiciones para lograrlos, para lo cuales necesario considerar las necesidades de los clientes, así como evaluar los resultados para poder proponer cambios que permitan la mejora continua
Gerencia	Aborda las prácticas de gestión de procesos, proyectos y recursos en función de los lineamientos establecidos en la Categoría de Alta Dirección. Proporciona los elementos para el funcionamiento de los procesos de la Categoría de Operación, recibe y evalúa la información generada por éstos y comunica los resultados a la Categoría de Alta Dirección	Gestión de Procesos	Establecer los procesos de la organización, en función de los procesos requeridos identificados en el plan estratégico. Así como definir, planificar, e implantar las actividades de mejora en los mismos
		Gestión de Proyectos	Asegurar que los proyectos contribuyan al cumplimiento de los objetivos y estrategias de la organización
		Gestión de Recursos	Conseguir y dotar a la organización de los recursos humanos, infraestructura, ambiente de

			trabajo y proveedores, así como crear y mantener la base de conocimiento de la organización. La finalidad es apoyar el cumplimiento de los objetivos del plan estratégico de la organización
Operación	Aborda las prácticas de los proyectos de desarrollo y mantenimiento de software. Esta categoría realiza las actividades de acuerdo a los elementos proporcionados por la Categoría de Gerencia y entrega a ésta la información y productos generados.	Administración de proyectos específicos	Establecer y llevar a cabo sistemáticamente las actividades que permitan cumplir con los objetivos de un proyecto en tiempo y costo esperados.
		Desarrollo y mantenimiento de software	Realizar sistemáticamente las actividades de análisis, diseño, construcción, integración y pruebas de productos de software nuevos o modificados cumpliendo con los requisitos especificados.

Tabla 1 Categorías MoProSoft

(8)

1.2 Norma ISO/IEC 29110

Pequeñas organizaciones - VSE (Very Small Entities)

Tener una definición de micro, pequeña y mediana empresa es complicado ya que no hay definiciones comúnmente aceptadas, la unión europea por ejemplo las define como sigue:

Tamaño	Empleados
Micro	Menos de 10
Pequeña	10 - 49
Mediana	50 - 249

Tabla 2 Empresas (9)

Y las cataloga comúnmente como la columna vertebral de la economía europea proporcionando una fuente potencial de empleo y crecimiento económico (9). En cuanto a empresas de Tecnologías de la Información - TI en Europa el 85% tienen menos de 10 empleados, en Canadá, una encuesta en el área de Montreal revela que 78% de las empresas de desarrollo de software tienen menos de 25 empleados y 50% tienen menos de 10 empleados (10).

Para quitar la ambigüedad de las posibles definiciones para la norma ISO/IEC 29110 se define Pequeña Organización (VSE - Very Small Entities) como una empresa, organización, departamento o proyecto que tiene entre 1 y 25 empleados (2).

Haciendo una comparación entre las VSE y las grandes empresas (más de 25 empleados) existe una marcada diferencia con el uso de estándares, menos de 18% de las VSE's están certificadas, mientras que las grandes empresas el 53% lo están² (10). Los motivos por los que las VSE's no usan estándares son los siguientes:

² Esto de acuerdo a la encuesta realizada por el WG24 en Febrero de 2006.

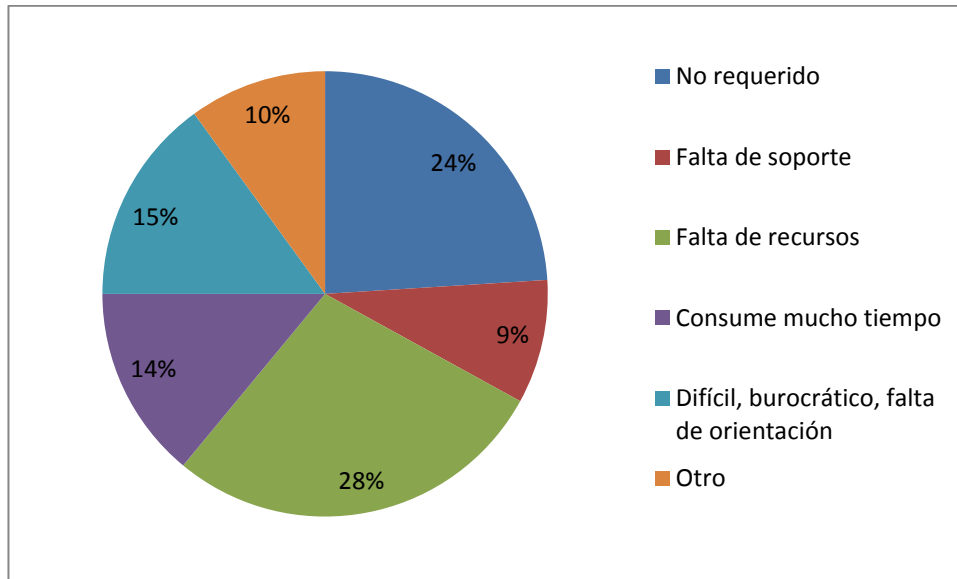


Ilustración 1. Por qué las VSE no usan estándares. Encuesta del WG24.

Historia de la norma ISO/IEC 29110

De acuerdo con la Organización para la Cooperación y el Desarrollo Económico (OCDE) en su reporte Pequeñas y Medianas Empresas y Perspectiva Empresarial (2005), las Pequeñas y Medianas Empresas constituyen el tipo dominante de organizaciones de negocios en el mundo, contabilizando por encima de 95% y hasta 99% del total de los negocios dependiendo del país. El reto para los gobiernos que integran la OCDE, es proveer un ambiente de negocios que apoye la competitividad de esta gran cantidad de negocios heterogéneos y que promueva una fuerte cultura empresarial. (1)

No obstante el gran número de pequeñas y medianas empresas, la mayoría de los estándares ISO/IEC no se ajustan a sus necesidades, ya que están estructurados para ser aplicables a empresas grandes, por lo que a las VSE se les complica su implementación, ya que no pueden ampliar el número de empleados, presupuesto o tiempo, y no ven un beneficio en establecer procesos de ciclo de vida de software, lo que ocasiona que no sean reconocidas como productoras de software de calidad y se encuentren aisladas de algunas actividades económicas.

En una reunión de llevada a cabo por miembros del ISO/IEC JTC1/SC7 en Austria en 2004 se planteó que las pequeñas organizaciones requieren normas adaptadas a su tamaño y nivel de madurez, ya que los estándares actuales están hechos para grandes organizaciones. Teniendo en mente lo anterior se logró un consenso sobre los objetivos generales de un futuro grupo de trabajo:

- Hacer los estándares actuales de Ingeniería de Software más accesibles para las VSE.
- Proveer documentación que requiera el mínimo ajuste y esfuerzo de adaptación.
- Definir estándares de procesos.
- Productos de trabajo y entregables.
- Alinear los perfiles a las nociones de niveles de madurez de acuerdo al estándar ISO/IEC15504.

En el 2005 en una reunión del SC7 en Tailandia se propuso la creación de un nuevo grupo de trabajo que atendiera dichos objetivos, surgiendo de ésta manera el grupo de trabajo 24(WG24). Ese mismo año el Grupo de Trabajo decidió llevar a cabo una encuesta para recopilar la información pertinente de las VSE de todo el mundo.

El grupo de trabajo 24 volvió a reunirse en mayo del 2006 con dos nuevos integrantes, México e India. Los principales resultados de esta reunión fueron los siguientes:

- Analizar los resultados de la encuesta.
- Evaluar los documentos presentados por los delegados.
- Seleccionar la norma Mexicana MoProSoft como documento de entrada para el desarrollo de los estándares y guías.
- Desarrollar estándares y guías.

Grupo de Perfil Genérico

La característica principal de las organizaciones para las que fue creada la norma es el tamaño, sin embargo, hay otros aspectos y características que puede afectar a la preparación o selección de un perfil, como puede ser: modelo de negocio (comercial, desarrollos propios, etc.); factores situacionales (ambientes inciertos, etc.) y niveles de riesgo. La creación de un perfil para cada

posible combinación de las características anteriores se traduciría en un conjunto inmanejable de perfiles. Por lo tanto los perfiles se agrupan de tal forma que sean aplicables a más de una categoría. Los *Grupos de Perfiles* es un conjunto de perfiles que se relacionan por composición de procesos (actividades, tareas, etc.), nivel de capacidad o ambos.

El Grupo de Perfiles Genérico ha sido definido como aplicable a la gran mayoría de las VSE que no desarrollan software crítico y tienen factores típicos situacionales. Este Grupo de Perfiles es una colección de cuatro Perfiles (Ver Tabla 3 Grupo de perfiles genérico

) que proporciona un enfoque progresivo para la gran mayoría de las VSE.

Grupo de Perfiles	Perfil
Genérico	Entrada
Genérico	Básico
Genérico	Intermedio
Genérico	Avanzado

Tabla 3 Grupo de perfiles genérico

Norma ISO/IEC 29110

La norma se compone de 5 partes. (Ver Tabla 4 Audiencia objetivo de ISO/IEC 29110

ISO/IEC 29110	Título	Audiencia
Parte 1	Visión General	VSE, asesores, productor de estándares, vendedores de herramientas y metodologías.
Parte 2	Marco de trabajo y Taxonomía	Productor de estándares, vendedores de herramientas y metodologías. No enfocado a VSE.
Parte 3	Guía de evaluación	Asesores y VSE.
Parte 4	Especificaciones de Perfil	Productor de estándares, vendedores de herramientas y metodologías. No enfocado a VSE.

Parte 5	Guía de administración e ingeniería	VSE.
----------------	-------------------------------------	------

Tabla 4 Audiencia objetivo de ISO/IEC 29110

Para cada sección de la ISO/IEC 29110 se generó lo siguiente:

- ISO/IEC 29110-1: Define términos comunes de negocio, también introduce las características y requerimientos de las VSE, clarifica las razones de los perfiles, los documentos, estándares y guías.
- ISO/IEC 29110-2: Establece la lógica detrás de la definición y aplicación de los perfiles estandarizados, así como especifica los elementos comunes para los perfiles (estructura, cumplimiento, evaluación) e introduce una taxonomía (catálogo) de los perfiles.
- ISO/IEC 29110-3: Define las guías de evaluación y el cumplimiento de los requerimientos necesarios para lograr los objetivos definidos por los perfiles.
- ISO/IEC 29110-4-1: Provee la especificación de todos los perfiles de Grupo de Perfiles Genérico. Este grupo es aplicable a las VSE que no desarrollan productos de software crítico. Los perfiles están basados en un subconjunto de elementos de estándares apropiados.
- ISO/IEC 29110-5-1-2: Provee una guía de implementación sobre administración e ingeniería para el *Perfil Básico*. El perfil Básico describe el proceso de desarrollo de software de una sola aplicación por un solo equipo de proyecto, sin riesgos o factores situacionales especiales.

1.3 Arquitectura de Software

La arquitectura de software es el conjunto de decisiones significativas sobre: la organización de un sistema de software, selección de elementos estructurales y sus interfaces, su comportamiento, composición de subsistemas y el estilo arquitectónico que guía esta organización. También tiene relación con el uso, funcionalidad, rendimiento, capacidad de adaptación, reutilización y restricciones (15).

Al hablar de arquitectura de software se hace alusión a la especificación de la estructura del sistema, entendida como la organización de componentes y relaciones entre ellos; los requerimientos que debe satisfacer el sistema y las restricciones a las que está sujeto, así como las propiedades no funcionales del sistema y su impacto sobre la calidad del mismo; las reglas y decisiones de diseño que gobiernan esta estructura y los argumentos que justifican las decisiones tomadas. (11)

La calidad de software se define como el grado en el cual el software posee una combinación deseada de atributos. Tales atributos son requerimientos adicionales del sistema, que hacen referencia a características que éste debe satisfacer, diferentes a los requerimientos funcionales, estas características o atributos se conocen con el nombre de atributos de calidad, los cuales se definen como las propiedades de un servicio que presta el sistema a sus usuarios. (11)

A grandes rasgos, puede establecerse una clasificación de los atributos de calidad en dos categorías:

- Observables vía ejecución: Se determinan del comportamiento del sistema en tiempo de ejecución. Estos atributos son: Disponibilidad, Confidencialidad, Funcionalidad, Desempeño, Confiabilidad, Seguridad externa y Seguridad interna.
- No observables vía ejecución: Se establecen durante el desarrollo del sistema. Estos atributos son: Configurabilidad, Integrabilidad, Integridad, Interoperabilidad, Modificabilidad, Mantenibilidad, Portabilidad, Reusabilidad, Escalabilidad y Capacidad de prueba.

Cada decisión incorporada en una arquitectura de software puede afectar potencialmente los atributos de calidad, de aquí que la imposición de ciertos estilos arquitectónicos mejora o disminuye las posibilidades de satisfacción de ciertos atributos de calidad del sistema. Cada estilo propicia atributos de calidad y la decisión de implementar alguno depende de los requerimientos de calidad del sistema (11).

1.3.1 Estilo arquitectónico

Un estilo arquitectónico es una familia de sistemas de software en términos de un patrón de organización estructural. Expresa componentes y las relaciones entre estos, con las restricciones de su aplicación y la composición asociada, así como también las reglas para su construcción.

1.3.2 Patrón Arquitectónico

Los patrones arquitectónicos son una descripción de un problema particular y recurrente de diseño, que aparece en contextos de diseño específico y presenta un esquema genérico demostrado con éxito para la solución, expresan el esquema de organización estructural fundamental para sistemas de software. Provee un conjunto de subsistemas predefinidos, especifica sus responsabilidades e incluye reglas y pautas para la organización de las relaciones entre ellos.

1.3.3 Patrones de Diseño

Patrón de diseño

Christopher Alexander dice: "Cada patrón describe un problema que se produce una y otra vez en nuestro medio, y luego describe el núcleo de la solución a ese problema, de tal manera que se puede utilizar esta solución un millón de veces, sin tener que hacerlo de la misma manera dos veces ". Aunque Alexander estaba hablando acerca de los patrones en los edificios y ciudades, lo que dice es cierto hablando de patrones de diseño orientado a objetos. Nuestras soluciones se expresan en términos de objetos y las interfaces en lugar de muros y puertas, pero el núcleo de ambos tipo de patrones es una solución a un problema en un contexto. (12)

Puesto que un patrón es un concepto multifacético, varias respuestas se pueden dar a la pregunta "¿Qué es un patrón de diseño de software?". Un patrón de diseño es una solución común a un problema que se repite en el diseño, nombra y especifica la estructura de diseño de forma explícita. Un patrón de diseño intenta grabar una solución a un problema recurrente encontradas durante el proceso de diseño. (13)

Patrón de diseño DAO (Data Access Object)

Contexto

El acceso a los datos varía dependiendo de la fuente de los datos.

El acceso al almacenamiento, como una base de datos, varía en gran medida dependiendo del tipo (bases de datos relacionales, bases de datos orientadas a objetos, ficheros planos, etc.) y de la implementación.

Problema

Las aplicaciones necesitan tener acceso a una fuente de datos, las cuales pueden ser muy diversas (Bases de datos, archivos, servicios externos, etc.) y algún tipo de almacenamiento persistente, que pueden ser implementados usando diferentes mecanismos.

Solución

El patrón DAO suministra un único medio de acceso a los datos independientemente de la fuente, lo que oculta detalles de la implementación haciendo que la interfaz no sea modificada usando diferentes tipos de fuentes de datos. Este patrón abstrae y encapsula el acceso a la fuente de datos haciéndolo transparente para el usuario.

Estructura

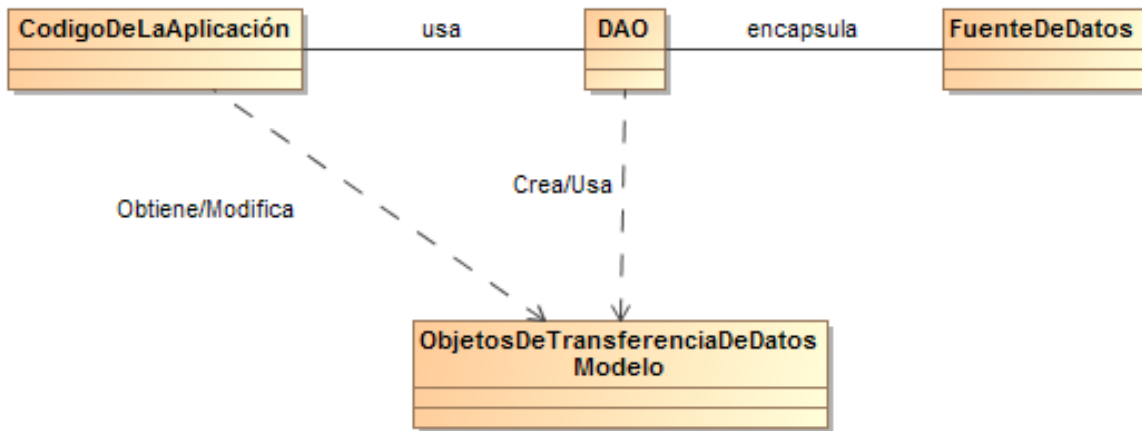


Ilustración 2 Diagrama clases patrón DAO

Diagrama de Secuencia:

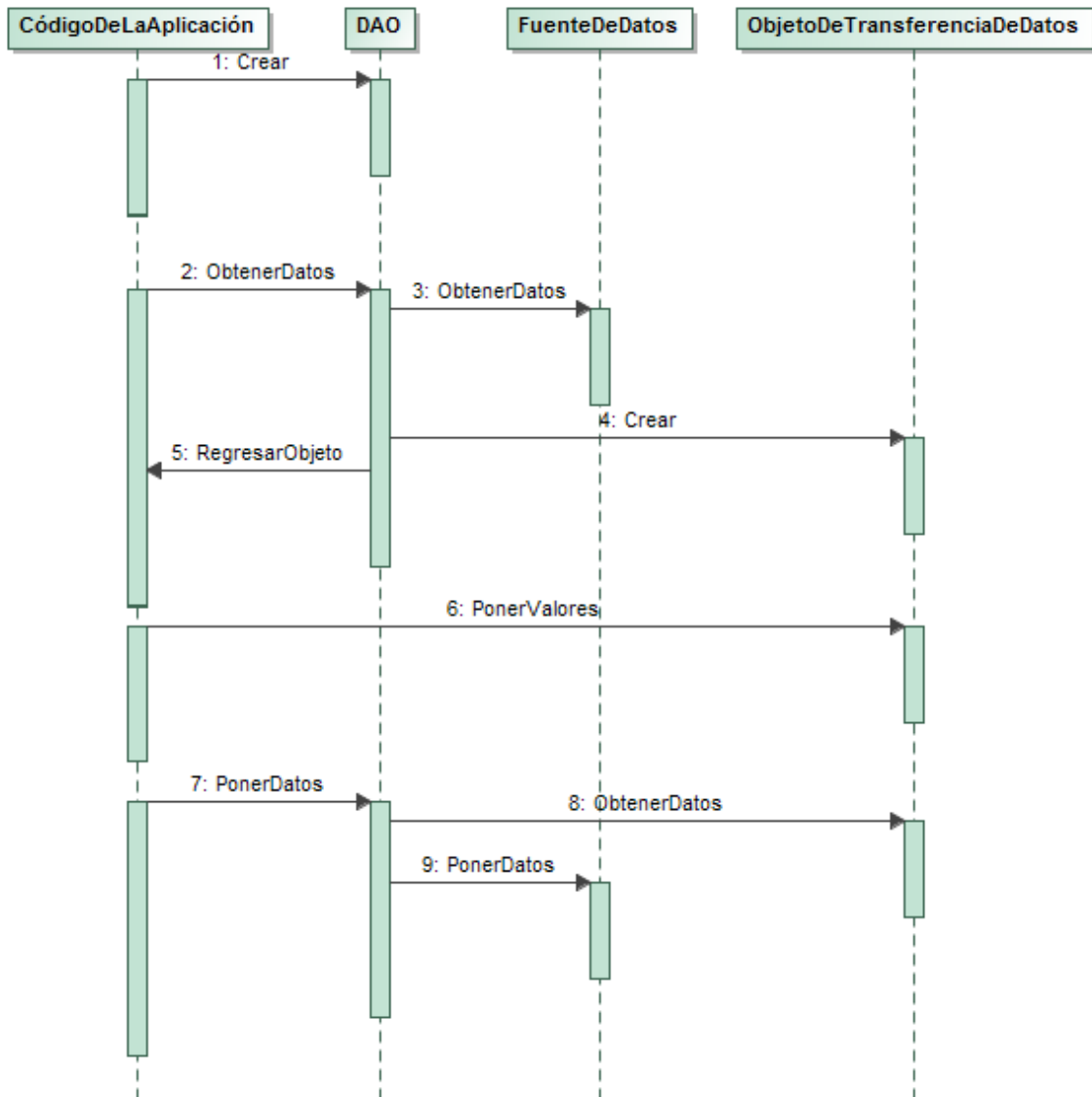


Ilustración 3 Diagrama secuencia patrón DAO

Participantes y responsabilidades:

- CódigoDeLaAplicación: Código de la aplicación que requiere el acceso a los datos para obtener y/o almacenar información.
- DAO: Interfaz que expone el patrón. Abstrae la implementación del acceso a los datos, con lo que el CódigoDeLaAplicación tiene un acceso transparente. Esta interfaz no cambia con las implementaciones para diferentes fuentes de datos.

- FuenteDeDatos: Representa una fuente de datos concreta por ejemplo: bases de datos (relacional, XML, orientada a objetos, etc.), archivos, servicios Web, servidores externos, etc.

Explicación

El patrón DAO proporciona una interfaz que debe utilizarse para acceder a los datos, de esta forma se oculta completamente la implementación del origen de los mismos. Debido a que la interfaz expuesta por este patrón no cambia con las diferentes implementaciones, permite que se adapte a diferentes formas de almacenamiento sin afectar lo programado previamente. Esencialmente, el DAO actúa como un adaptador entre el componente y la fuente de datos.

Ventajas y Desventajas

Ventajas	Desventajas
Independencia de la lógica de negocio a la de datos.	Complejidad en el diseño e implementación.
Facilidad de adaptación.	La complejidad aumenta a medida que se añaden capas de persistencia.
Reduce la complejidad de la implementación de la lógica de negocio al encapsular el acceso a los datos.	

Tabla 5 Ventajas y desventajas patrón DAO

Fábrica Abstracta

Contexto

Debemos crear diferentes objetos, todos pertenecientes a la misma familia. Por ejemplo: las librerías para crear interfaces gráficas suelen utilizar este patrón y cada familia sería un sistema operativo distinto.

Problema

El problema que intenta solucionar este patrón es crear diferentes familias de objetos cuyos productos no deben entremezclarse.

Con el propósito de comprender mejor el problema se dará un ejemplo: la apariencia de la interfaz gráfica (GUI) varía de acuerdo al sistema operativo (Windows, Linux, etc.). Así para poder programar una aplicación se deben crear los objetos con la apariencia apropiada para cada sistema operativo. Lo anterior ocasiona que se deba tener código para cada sistema operativo, acarreando entre otros inconvenientes: tener en la aplicación código para cada sistema operativo, lo que complica los mantenimientos y mejoras posteriores; se tienen sentencias condicionales para adaptar la apariencia.

Solución

Proporcionar una interfaz para crear familias de objetos relacionados o que dependen entre sí, sin especificar sus clases concretas

Estructura

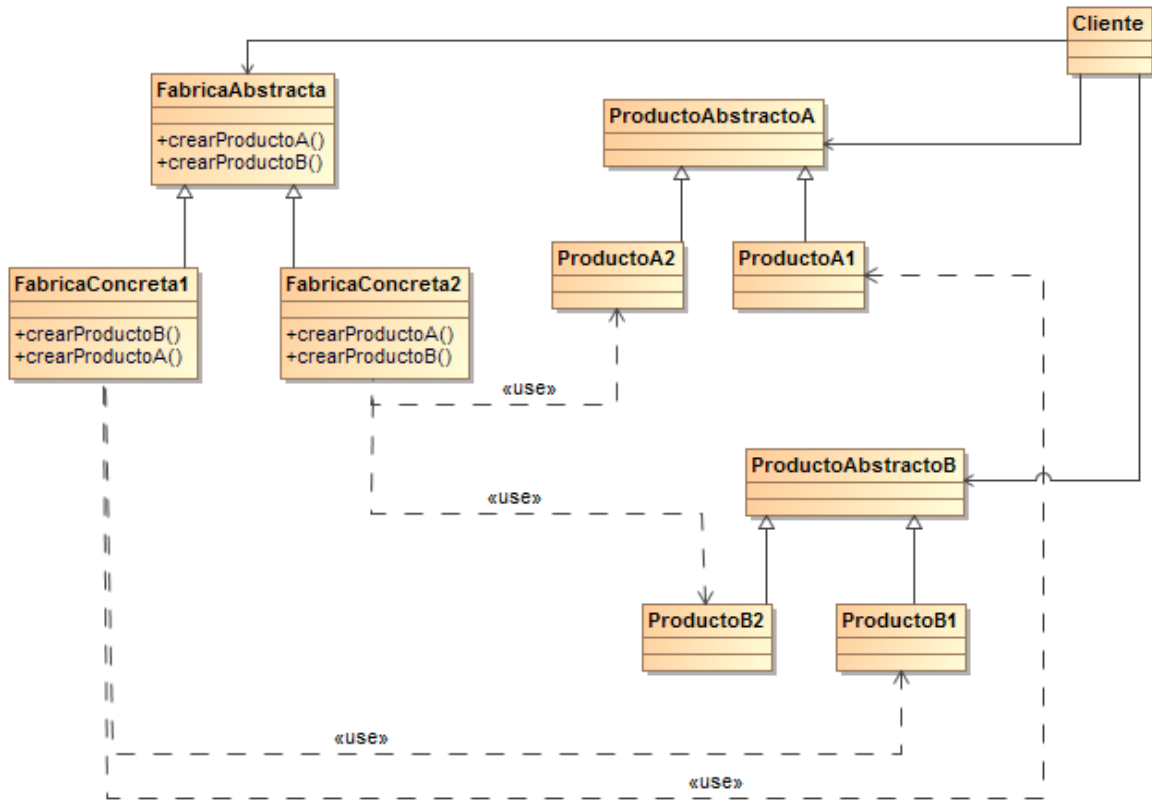


Ilustración 4 Patrón Fábrica Abstracta

Diagrama de secuencia:

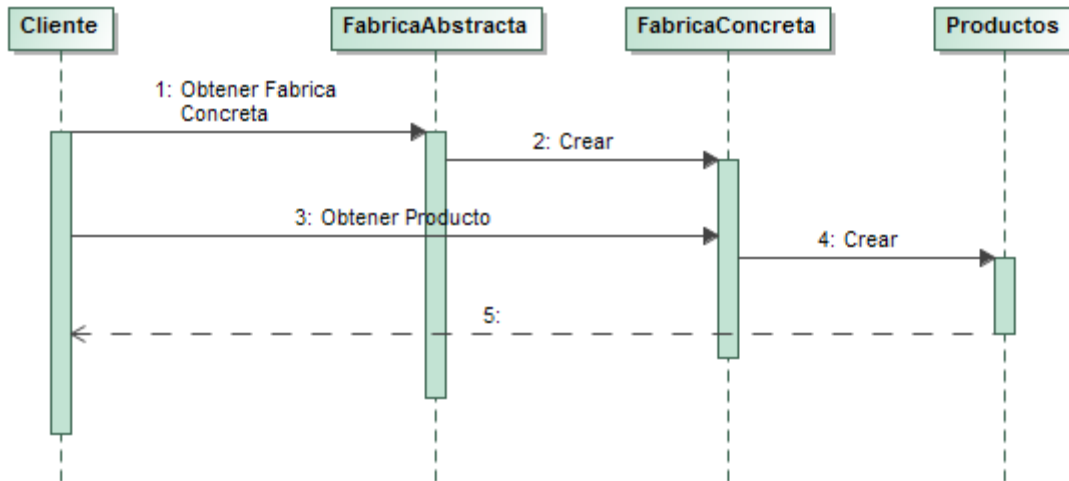


Ilustración 5 Diagrama secuencia Fábrica abstracta

Participantes y responsabilidades:

FabricaAbstracta: Interfaz que cualquier fábrica debe cumplir, tiene un método para retornar la instancia de la fábrica concreta y métodos para la creación de productos abstractos.

FabricaConcreta: Estas son las diferentes familias de productos. Implementa la interfaz “FabricaAbstracta”, por lo que define la creación de productos concretos.

ProductoAbstracto: Definición de las interfaces para la familia de productos genéricos, estos son independientes de las familias de productos. El cliente trabajará directamente sobre esta interfaz, que será implementada por los diferentes productos concretos.

Producto: Implementación de los diferentes productos abstractos.

Cliente: Utiliza la interfaz FabricaAbstracta para crear una familia de objetos (FabricaConcreta) y poder instancias productos.

Explicación

La fábrica abstracta tiene la responsabilidad de proveer una forma para crear una familia completa, por lo que el cliente nunca instanciará objetos directamente sino a través de la fábrica, por lo tanto resulta sencillo cambiar las familias según los productos que se utilizan.

Ventajas y Desventajas

Ventajas	Desventajas
Aísla los clientes de las implementaciones, ya que sólo usan la interfaz <code>FabricaAbstracta</code> .	Es difícil dar cabida a nuevos tipos de productos porque afecta a la interfaz de la fábrica abstracta y por consiguiente de todas sus subclases.
Facilita el cambio de familias de productos, ya que la clase de familia concreta sólo aparece una vez y es fácil cambiarla.	

Tabla 6 Ventajas y desventajas patrón Fábrica Abstracta

1.4 Tipos de datos de acuerdo a su estructura y almacenamiento

Una clasificación de acuerdo a la estructura es la siguiente: (14)

1.4.1 Datos estructurados

Este tipo de datos está caracterizado por:

- Tienen una estructura fija.
- Son atributos o variables fuertemente tipificados (int, float, string).
- Cada atributo en una relación está definido para todos los registros.

Hablar de datos estructurados es hablar de los primeros tipos de datos que existieron o mejor dicho, que fueron de interés para la comunidad computacional debido a su relación directa con los lenguajes de programación donde existe el concepto de estructura, registro, variable y campo, y las necesidades de las diversas compañías e instituciones de investigación. Nóminas, inventarios,

estadísticas, cuentas y catálogos, son datos estructurados y era evidente la necesidad de crear algoritmos capaces de manipular este tipo de datos, o sea algoritmos de indexamiento.

Este tipo de dato es almacenado en Sistemas Manejadores de Bases de Datos (SMBD) que son sistemas complejos que garantizan la “acidez” de los datos (ACID Atomicidad Consistencia Aislamiento Durabilidad).

También pueden ser almacenados en herramientas más sencillas que cumplen con su función, pero no se preocupan por manejo de errores, concurrencia y otros factores.

1.4.2 Datos no estructurados

Este tipo de datos está caracterizado por:

- No están organizados de acuerdo a algún patrón.
- No poseen definiciones de tipo.
- No existe el concepto de variables o atributos.

Cuando un dato no tiene estructura por lo general nos referimos a algún tipo de documento. Si se desea hacer algún tipo de búsqueda habría que revisar si los términos que conforman la consulta se encuentran en el/los documento(s). Esta tarea puede ser tan costosa que al buscar un término se convierte en una búsqueda secuencial.

Sin embargo, se puede pensar en extraer todos los términos de cada documento e indexarlos (a manera de datos estructurados) de manera que tendremos una lista invertida que nos permite ir desde los términos hacia los documentos que los contienen. A esta actividad se le conoce como “indexamiento a texto completo” (fulltextindexing).

Para no tener que almacenar e indexar todos los términos de un documento, se pueden usar los siguientes métodos para eliminar aquellos que son innecesarios:

- Lematización: eliminación de palabras comunes en el lenguaje como artículos, conjunciones, etc.

- Truncamiento: deja las palabras en sus raíces sintácticas, de manera que palabras como: computación, computacionalmente, computadora son reducidas a “comp” y se pueden hacer búsquedas por cualquiera de los términos indexados.

Desafortunadamente lo anterior no es suficiente; cuando un usuario realice una búsqueda se encontrará con miles de documentos que satisfacen su consulta.

Las posibles soluciones a esta problema no serán tocados en el presente trabajo ya que sale del alcance.

1.4.3 Datos semiestructurados

Este tipo de datos está caracterizado por:

- Son variables pobremente tipificadas (x=1 es válido y x=“hola” también es válido).
- Un registro no necesariamente tiene que tener todos sus atributos definidos. Mientras por ejemplo en una base de datos relacional un campo debe establecerse como NULL cuando no se tiene, en un ambiente de datos semiestructurados basta con omitir dicho atributo.
- Un atributo de un registro puede ser otro registro.
- No existe necesariamente una diferencia entre un identificador de un campo y el valor mismo de éste.

Este tipo de dato puede ser representado como:

- Árbol
- Texto indentado
- Lenguaje de Marcado

Es importante recordar que el máximo representante de los datos semiestructurados es XML, de manera que asociaremos estos datos con documentos XML.

XML - Lenguaje de Marcas Extensible (eXtensible Markup Language) es un metalenguaje desarrollado por la W3C (World Wide Web Consortium), está constituido por un conjunto de reglas para definir etiquetas semánticas que dividen un documento en partes e identifica las

diferentes partes del documento. XML está derivado de un lenguaje para estructurar documentos, conocido como lenguaje estándar generalizado de marcas, del inglés Standard Generalized Markup Language, SGML publicado en 1986. (8)

Entre las ventajas que ofrece XML tenemos las siguientes: (9)

- Bien formado.
- Extensible: Permite ampliar el lenguaje mediante nuevas etiquetas y la definición de lenguajes nuevos.
- Autodocumentado: La estructura de la información de alguna manera está definida dentro del mismo documento.
- Intercambiable: Portable entre distintas arquitecturas. Para su lectura e interpretación es necesario un analizador sintáctico (parser).

Para documentos XML existen dos tipos de almacenamiento:

- XML EnabledDatabases (Bases de datos habilitadas para XML): son aquellas que desglosan la información de un documento XML en su correspondiente esquema relacional o de objetos. Este tipo de base de datos mapea un XML a una base de datos relacional, aceptando un XML como entrada y haciendo una transformación para hacer el XML como salida.
- XML NativeDatabases (Bases de datos nativas de XML): son aquellas que respetan la estructura del documento, se pueden hacer consultas sobre dicha estructura y es posible recuperar el documento tal como fue insertado originalmente.

2. Paquete de Puesta en Operación - PPO

Con el fin de ayudar a la implementación de la norma ISO/IEC 29110 y proporcionar orientación sobre la aplicación efectiva, se crearon Paquetes de Puesta en Operación - PPO, los cuales ayudan a definir directrices y explicar con más detalle los procesos que componen los perfiles de la norma.

2.1 Qué es un Paquete de Puesta en Operación

"Un Paquete de Puesta en Operación es un conjunto de artefactos desarrollados para facilitar la implementación de un conjunto de buenas prácticas, del marco de trabajo seleccionado en una pequeña organización (VSE)." (2)

Actualmente (Febrero de 2012) los Paquetes de Puesta en Operación que se han desarrollado ayudan a implementar el Grupo de Perfiles Genérico Básico.

2.2 Estructura de un PPO

La estructura de los PPO se encuentra ya definida y es mostrada a continuación:

1. Descripción técnica
 - Propósito de este documento
 - Por qué el tema es importante
2. Definiciones
3. Relación con la ISO/IEC 29110
4. Resumen de procesos, actividades, tareas, roles y productos
5. Descripción de procesos, actividades, tareas, pasos, roles y productos
 - Descripción de roles
 - Descripción de productos
 - Descripción de artefactos
6. Plantilla
7. Ejemplo

- | |
|--|
| <ol style="list-style-type: none">8. Lista de verificación9. Herramientas10. Referencias a otros estándares y modelos11. Referencias12. Formulario de evaluación |
|--|

Tabla 7. Estructura de los Paquete de Puesta en Operación (2)

NOTA: Un *Paquete de Puesta en Operación* se proporciona tal cual. El autor(es) no ofrece ningún tipo de garantía, por los resultados obtenidos por el uso de éste material. (2)

2.3 Paquetes existentes actualmente

Actualmente (Febrero del 2012) sólo existen Paquetes de Puesta en Operación para el perfil básico, este perfil describe el desarrollo de una sola aplicación por un solo equipo sin riesgos o factores especiales. Los Paquetes que componen el Perfil Básico son los siguientes:

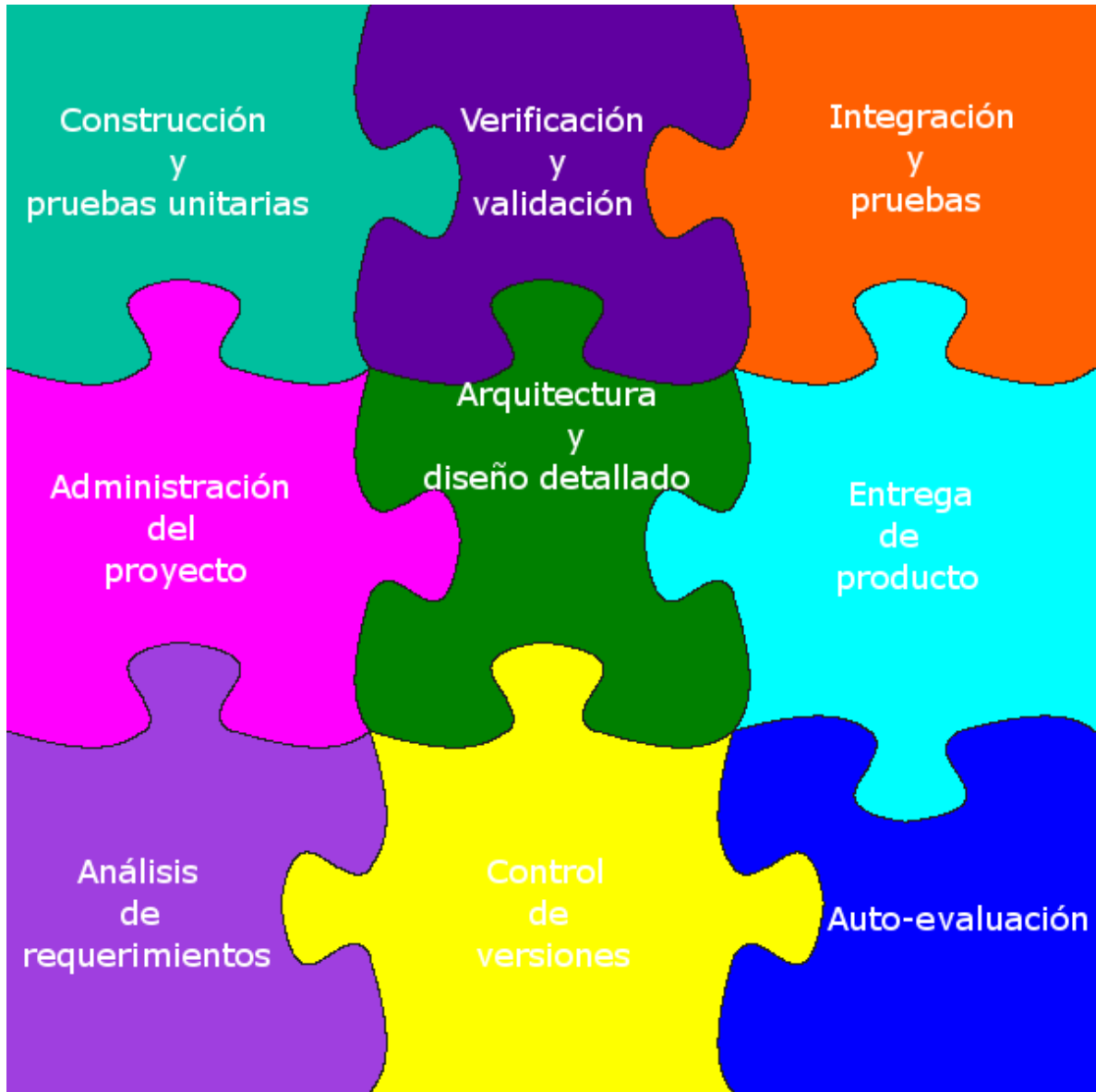


Ilustración 6. Paquetes de Puesta en Operación que componen el Perfil Básico

A continuación se muestra el país de los desarrolladores de los paquetes con los que se cuentan:

Título del Paquete	Desarrollador por:
Análisis de requerimientos	Bélgica, Canadá

Arquitectura y diseño detallado	Canadá
Construcción y pruebas unitarias	México
Integración y pruebas	Colombia
Verificación y validación	Canadá
Control de versiones	Tailandia
Administración del proyecto	Irlanda
Entrega de producto	Canadá, Tailandia
Auto-evaluación	Finlandia

Tabla 8. Paquetes del perfil básico. (15)

2.4 Importancia de los Paquetes de Puesta en Operación

Los Paquetes de Puesta en Operación son desarrollados para facilitar la implementación de alguno de los perfiles que componen la norma ISO, la importancia de cada uno de los paquetes para el perfil básico se describe a continuación:

Construcción y Pruebas unitarias

Todas las fases son importantes en el desarrollo de software, sin embargo la actividad central es la construcción, ya que en algunos proyectos es posible omitir fases (de ninguna forma es aconsejable esto, no obstante en proyectos reales puede ocurrir) como obtención de diseño y pruebas, sin embargo, no es posible prescindir la fase de construcción. Por lo tanto el mejorar la calidad de esta fase significa mejorar el desarrollo (16). Cerca del 12% de los defectos son inyectados durante esta fase, y pueden ser removidos con las pruebas unitarias, las cuales aseguran un correcto funcionamiento de un módulo de código (17).

Verificación y Validación

Son un conjunto de procesos de comprobación y análisis que aseguran que el software que se desarrolla está acorde a su especificación y cumple con las necesidades de los clientes.

Cada una de éstas responde a preguntas diferentes:

- Verificación: Responde a la pregunta, ¿Se está construyendo el producto correctamente? Se comprueba que el software cumple los requisitos funcionales y no funcionales de su especificación.
- Validación: Responde a la pregunta, ¿Se está construyendo el producto correcto? Comprueba que el software cumple las expectativas que el cliente espera.

Integración y Pruebas

El objetivo de las pruebas de integración es verificar que componentes o subsistemas interactúan correctamente a través de sus interfaces, tanto internas como externas, tienen la funcionalidad establecida y se ajustan a los requisitos especificados en las verificaciones correspondientes.

Administración del proyecto

La administración de proyectos es la aplicación de conocimientos, habilidades, herramientas y técnicas a las actividades del proyecto para cumplir con los requisitos del mismo. La conforman 5 grupos de procesos: Iniciación, Planificación, Ejecución, Seguimiento y Control y Cierre.

Dirigir un proyecto implica por lo general: identificar requisitos; abordar necesidades e inquietudes de los interesados; equilibrar restricciones de alcance, calidad, cronograma, presupuesto, recursos y riesgos.

Aproximadamente medio millón de administradores de proyectos en el mundo son responsables de un millón de proyectos cada año. Muchos de estos proyectos no cumplirán con las expectativas del cliente o se entregarán fuera de presupuesto y tiempo.

Los proyectos que cumplen con sus objetivos se destacan en estas actividades fundamentales: la planificación, estimación, control de cambios y de calidad. Por el contrario, los proyectos que se ejecutan tarde o fracasan normalmente tenían planes defectuosos u optimistas, tenían estimaciones que no se anticipan cambios y fallan en controlar la calidad.

Arquitectura y diseño detallado

La arquitectura de software es el conjunto de decisiones significativas sobre: la organización de un sistema de software, selección de elementos estructurales y sus interfaces, su comportamiento, composición de subsistemas y el estilo arquitectónico que guía esta

organización. También tiene relación con el uso, funcionalidad, rendimiento, capacidad de adaptación, reutilización y restricciones (18).

El diseño se refiere al establecimiento de las estructuras de datos, la arquitectura general de software, representaciones de interfaz y algoritmos. El proceso de diseño traduce requisitos en una representación de software.

Siguiendo este proceso se sabrá como crecerá el software y se tendrá la seguridad que representa los requerimientos. Aunado a lo anterior la documentación de la arquitectura y diseño permite tener un canal común de comunicación entre los interesados, el cual no solo es útil durante el desarrollo, también sirve para el mantenimiento futuro y las mejoras.

Entrega de producto

Se debe formalizar que el proyecto ha cumplido con los requisitos acordados, verificar que se han suministrado y aceptado todos los productos entregables y validar que se ha cumplido con los criterios de aceptación y salida.

Análisis de Requerimientos

Un requerimiento es una condición o capacidad que un sistema, producto, servicio, resultado o componente debe satisfacer o poseer para cumplir con un contrato, norma, especificación u otros documentos formalmente impuestos. Los requisitos incluyen las necesidades, deseos y expectativas cuantificadas y documentadas del cliente. (19)

Estudios destacan que cerca del 50% de los defectos de software son introducidos en la fase de requerimientos, por lo que es fundamental definir lo menos ambiguo posible las necesidades de los clientes para asegurar una comprensión común. (2)

Control de Versiones

Un sistema de control de versiones proporciona un depósito para el registro de los cambios en el código fuente y artefactos asociados. Es un lugar para realizar un seguimiento de cómo el software cambia, cuando cambió, y quién lo cambió.

El control de versiones ayuda a determinar por qué se tomaron las decisiones y que opciones se han intentado.

Auto-evaluación

El proceso de evaluación tiene dos funciones principales:

1. Determinar la capacidad de algún proceso: Típicamente es realizado por un tercero ajeno a la organización y su resultado puede ser utilizado para la selección de proveedores o el reconocimiento de la organización. Antes de someterse a la evaluación por el tercero, la pequeña organización debe autoevaluarse.
2. Comprensión para una mejora continua: proporciona una visión concreta de fortalezas y debilidades de los procesos.

3. Sistema Manejador de Paquetes de Puesta en Operación - SMPPPO

3.1 Definición de Sistema Manejador

Existen varias definiciones para un "Sistema Manejador" las cuales varían de acuerdo al contexto, por lo que expondrán algunas y se llegará a una definición que se usará en el resto del presente trabajo:

"Sistema Manejador se refiere a lo que hace la organización para gestionar sus procesos o actividades, de manera que sus productos o servicios cumplen con los objetivos que se ha propuesto." (20)

"Un Sistema Manejador es un conjunto de actividades interrelacionadas o que interactúan, elementos que utilizan las organizaciones para implementar políticas y alcanzar los objetivos." (21)

"Un Sistema Manejador es el marco de los procesos y procedimientos utilizados para garantizar que una organización pueda cumplir todas las tareas necesarias para lograr sus objetivos." (22)

3.2 Tipos de Sistemas Manejadores

Existen varios tipos de Sistemas Manejadores cada uno de ellos en diferentes contextos, por lo que para obtener un consenso se revisarán varios.

Sistema Manejador de Bases de Datos - SMBD

"Un Sistema Manejador de Bases de Datos consiste en una colección de datos interrelacionados y un conjunto de programas para acceder a dichos datos." (23)

"El objetivo principal de un SMBD es proporcionar una forma de almacenar y recuperar la información de una base de datos de manera que sea tanto práctica como eficiente." (23)

Los sistemas manejadores de bases de datos proporcionan un lenguaje definición de datos (LDD) para especificar el esquema y un lenguaje de manipulación de datos (LMD) para expresar las consultas y las modificaciones de la base de datos.

Sistema Manejador de Aprendizaje - LMS (Learning Management System)

"Los LMS son aplicaciones utilizadas para crear y gestionar plataformas educativas, es decir, entornos virtuales donde una institución, empresa o centro educativo, gestiona sus recursos educativos proporcionados por docentes y desde donde se organiza el acceso a esos recursos por los estudiantes, y se permite la comunicación entre alumnado y profesorado."

"Las principales funciones del LMS son: gestionar usuarios, recursos, así como materiales y actividades, administrar el acceso, controlar y hacer seguimiento del proceso de aprendizaje, realizar evaluaciones, generar informes, gestionar servicios de comunicación como foros de discusión, videoconferencias, entre otros."

"Un LMS generalmente no incluye la posibilidad de crear sus propios contenidos, pero se encarga de gestionar contenidos creados por fuentes diferentes. La labor de crear los contenidos para los cursos se desarrolla mediante un LCMS (Learning Content Management System)." (24)

Sistema Manejador de Contenido - CMS (Content Management System)

"Un CMS es un software que ofrece herramientas para la creación, publicación y gestión de contenido en un sitio web. "Contenido" es cualquier cosa, una noticia, un blog, un vídeo, una fotografía, un podcast, un artículo, o una descripción de un producto que se está vendiendo. En términos generales, el contenido es cualquier combinación de texto, gráficos, fotografías, audio y video que representa algo que los visitantes de su sitio web pueden leer, ver y oír." (25)

"Un CMS ofrece una serie de características que simplifican el proceso de construcción, implementación y administración de sitios Web, incluye también lo siguiente:"

- *Interfaz de administración.*
- *Depósito de base de datos para el contenido.*
- *Mecanismo para asociar la información que se almacena en la base de datos con una página física en el sitio.*

- *Conjunto de herramientas para la creación, publicación y gestión de contenidos.*
- *Gestión de usuarios.*
- *Marco de seguridad.*
- *Taxonomía y etiquetado. (25)*

3.3 Funciones de un Sistema Manejador

Recopilando las funciones de los Sistemas Manejadores mencionados anteriormente se obtiene que debe realizar lo siguiente:

- Proporcionar algún medio de manipular (normalmente una interfaz gráfica, sin embargo no está limitada a esta opción) la información de una manera práctica y eficiente, esta debe cubrir la posibilidad de consultar, crear, modificar y eliminar información. El tipo de información varía dependiendo del Sistema Manejador.
- Mantener la seguridad de la información almacenada.
- Administrar usuarios.

Los sistemas manejadores son herramientas que ayudan a definir y explotar algún tipo de contenido, definiendo una estructura y proporcionando un acceso sencillo y eficiente al manejo de ese contenido.

En el caso de SMDB la estructura de las bases es el modelo de datos, que consiste en una colección de herramientas para describir datos, relaciones, semántica y restricciones de consistencia.

En los LMS ese contenido son recursos educativos, sin embargo, en este caso en particular nos ofrece una posibilidad de crear estos recursos, pero si se ponen a disposición de los usuarios, los cuales también se pueden administrar.

Finalmente en un CMS aunque el contenido que maneja puede ser muy extenso, su objetivo es dar un punto de acceso a los usuarios y la posibilidad de gestionar la información.

3.4 Sistema Manejador de Paquetes de Puesta en Operación - SMPPPO

3.4.1 Definición

Un Sistema Manejador de Paquetes de Puesta en Operación - SMPPPO es una aplicación que aloja, permite el acceso y administra de Paquetes de Puesta en Operación - PPO.

El objetivo principal de un SMPPPO es la difusión y construcción de Paquetes de Puesta en Operación en forma colaborativa.

Sus principales características son:

- Interfaz de administración de Implementación de Paquetes de Puesta en Operación: El usuario puede crear, consultar, actualizar y eliminar implementaciones de paquetes.
- Los PPO's deben ser almacenados: se debe contar con alguna forma de persistencia.
- Administración de usuarios: Un usuario con suficientes privilegios puede administrar los datos de los usuarios.
- Mantener la seguridad de las implementaciones de paquetes: La creación, consulta, actualización y eliminación debe permitirse solamente a usuarios con los privilegios suficientes.
- Permitir retroalimentar las buenas prácticas de los PPO's: Los usuarios pueden retroalimentar la utilidad o buen trabajo de una implementación de paquete.

Note que se omitieron muchas funciones de los Sistemas Manejadores mencionados como son:

- No está permitido definir o añadir otro tipo de contenido: Los PPO's ya tienen una estructura bien definida, por lo que no es recomendable cambiarla.
- No cuenta con un proceso de seguimiento: Dado que el fin es tener PPO's terminados, el seguimiento puede ser omitido.
- No cuenta con un lenguaje que proporcione acceso a los datos: En lugar de un lenguaje se proporcionará una interfaz gráfica, con el fin de facilitar la creación de implementaciones de paquetes.

3.4.2 Beneficios de contar con un SMPP0

- Las pequeñas organizaciones tienen a su disposición diferentes implementaciones de paquetes: Resulta de suma importancia para las pequeñas organizaciones que no cuentan con los recursos para pagar a un tercero por la asesoría para implementar esta norma.
- Facilidad para crear y actualizar implementaciones de paquetes: Realizar estas acciones no debe representar dificultad para las pequeñas organizaciones. El formato en el cual se pueden encontrar los paquetes, documentos de tipo PDF o DOC, resulta poco apropiado para realizar modificaciones, mejoras o comentarios respecto a las implementaciones de paquetes existentes, alguna organización que haya decidido poner en funcionamiento una parte de la norma con alguna implementación de paquete.
- Permitir la mejora constante de las implementaciones de paquetes: Dado que se tienen métodos para retroalimentar las prácticas, los autores pueden mantener las implementaciones en mejora constante.
- Distribución de las implementaciones de paquetes: La filosofía de los paquetes de puesta en operación es que sean de libre uso (No obstante esto, el contenido de los paquetes son propiedad intelectual para la organización del autor).
- Creación de las implementaciones de paquetes de forma colaborativa: Este tipo de aplicación permite que un grupo de personas trabajen en conjunto para alcanzar implementaciones de paquetes que faciliten la aplicación de la norma.

4 Definición de Requerimientos

4.1 Definición

Se requiere una aplicación que cumpla con la definición de Sistema Manejador de Paquetes de Puesta en Operación. Esta aplicación deberá permitir a las pequeñas organizaciones desarrolladoras de software buscar, crear, consultar, actualizar, eliminar, migrar las implementaciones de Paquetes de Puesta en Operación, retroalimentar las buenas prácticas para organizar sus propios procesos y mejora de las implementaciones de los paquetes.

4.2 Objetivo

Desarrollar un Sistema Manejador de Paquetes de Puesta en Operación que permita a las organizaciones desarrolladoras de software buscar, crear, consultar, actualizar, eliminar, migrar las implementaciones de Paquetes de Puesta en Operación, retroalimentar las buenas prácticas para organizar sus propios procesos y mejorar de las implementaciones de los paquetes.

El sistema a desarrollar deberá ser modificable (3), escalable (4) y funcional (5).

4.3 Requerimientos Funcionales y No Funcionales

Funcionales

Se espera que la aplicación cumpla con las siguientes características:

- La retroalimentación se hará de dos formas:
 - Asignando una puntuación entre 1 (en caso de ser insuficiente) y 5 (en caso de ser sobresaliente) de acuerdo a la percepción del usuario.
 - Realizando un comentario a la implementación de paquete.
- Un *visitante* podrá realizar las siguientes tareas:
 - Buscar y consultar las Implementaciones de Paquetes de Puesta en Operación contenidas en la aplicación.
 - Registrarse (proporcionando nombre, apellidos, clave, contraseña, correo, descripción, institución y país) para obtener una cuenta de editor.

- Ver la retroalimentación que ha tenido la implementación de paquete (comentarios y valoración).
- El *editor* podrá:
 - Reportar contenido no apropiado, proporcionando un pequeño texto justificando el motivo de su reporte.
 - Retroalimentar al(los) autor(es) mediante un comentario a una implementación de paquete.
 - Valorar las implementaciones de paquete. Lo que ayudara a visitantes y/o editores a seleccionar una implementación de paquete de las disponibles en la aplicación.
 - Actualizar sus datos personales.
- El *administrador* podrá:
 - Liberar de un reporte las implementaciones de paquete.
 - Administrar las cuentas de los usuarios.
 - Administrar los paquetes de los usuarios.
- La aplicación permitirá crear, consultar, actualizar y borrar al *editor* y *administrador*:
 - Implementaciones de Paquetes de Puesta en Operación.
 - Implementaciones de sección de algún Paquete de Puesta en Operación.
 - Configuraciones de paquetes.³

No Funcionales

Se espera que el desarrollo cumpla con los siguientes requerimientos No Funcionales:

- La aplicación deberá ser Web ejecutándose sobre un contenedor Java.
- La persistencia de los datos será en un sistema manejador relacional. Se debe considerar la posibilidad de cambiar la persistencia a una base de datos con XML, sin significar grandes cambios a lo programado.
- La interfaz debe permitir el cambio de idioma entre Español e Inglés.

³Para más información acerca de **configuración de paquete** consultar el término en el Glosario.

4.4 Casos de uso

4.4.1 Diagrama General

Con el fin visualizar los escenarios y actores del sistema, se presenta el diagrama general de Casos de Uso.

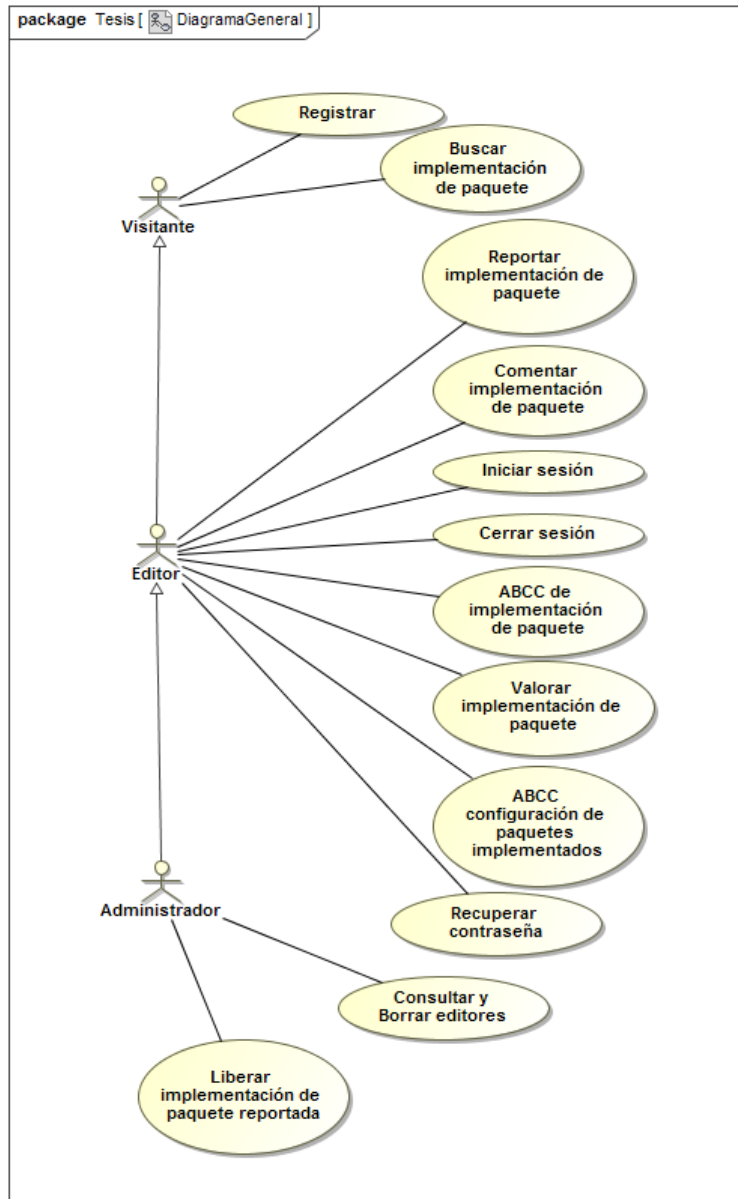


Ilustración 7 Diagrama general de Casos de Uso

4.4.2 Detalle de casos de uso

A continuación se dará el detalle de los casos de uso para nada uno de los puntos del objetivo.

Caso de uso: Buscar implementación de paquete

Actor: Visitante

Diagrama de caso de uso:



Descripción: El visitante proporcionara alguna(s) palabra(s) clave para realizar una búsqueda por algunos de los siguientes campos: nombre del paquete, metodología, requisitos, palabras clave, autores.

Precondición: El visitante debe encontrarse en el sitio.

Flujo:

Actor		Sistema		
Paso	Acción	Paso	Acción	Flujo Alternativo
1	Presionar liga Buscar .	2	Mostrar el formulario de búsqueda.	
3	Proporcionar algún criterio de búsqueda.			
4	Presionar botón Buscar .	5	Mostrar lista con la implementación de paquetes que cumplan con los criterios de búsqueda.	E1

Flujo Alternativo:

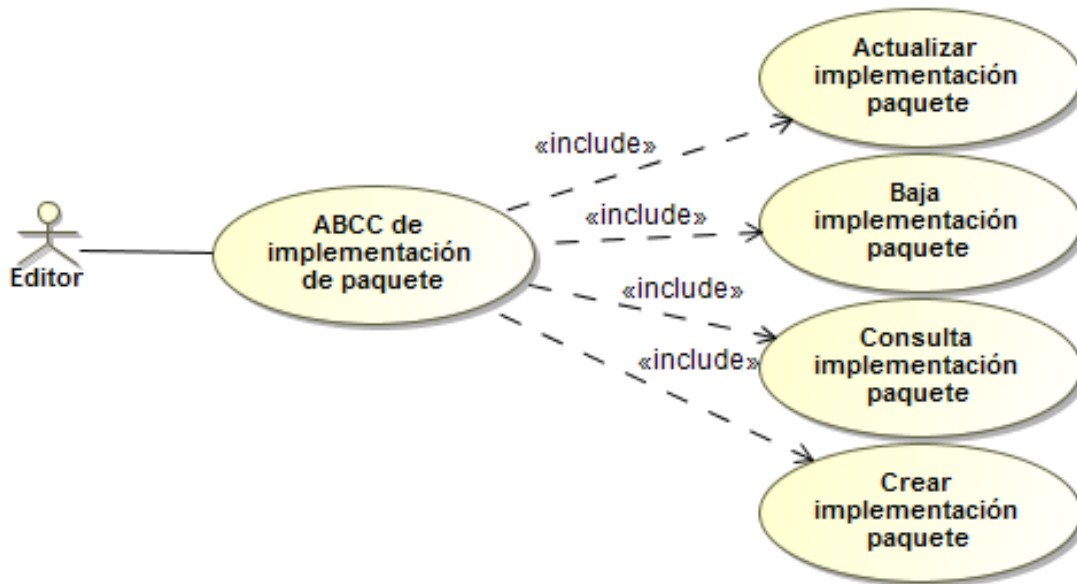
ID	Nombre	Acción
E1	Error de base de datos.	Se mostrará un mensaje informando que ocurrió algún problema.

Poscondición:

Caso de uso: ABCC de implementación de paquete

Actor: Editor

Diagrama de caso de uso:



Descripción: El editor podrá actualizar, borrar, consultar y crear las implementaciones de paquete que él desee.

Precondición: El visitante debe iniciar sesión y desear realizar alguna operación relacionada con implementaciones de paquetes.

Flujo:

Actor		Sistema		
Paso	Acción	Paso	Acción	Excepción
1	Para llevar a cabo alguna de las operaciones el editor debe seleccionar algunas de las opciones: implementar paquete, Ver mis paquetes, Todos los paquetes, editar o eliminar. (al tener en pantalla alguna implementación de paquete en la que el editor	2	Mostrar la interfaz correspondiente a la opción seleccionada.	

Definición de Requerimientos

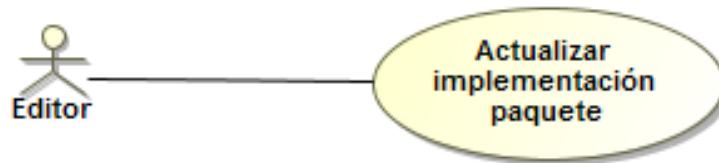
	pueda realizar esta acción)			
Los siguientes pasos serán descritos en los casos de uso incluidos en éste.				

Poscondición: Las implementaciones de paquete del editor publicadas en el sitio.

Caso de uso: Actualizar implementación paquete (ABCC de implementación de paquete)

Actor: Editor

Diagrama de caso de uso:



Descripción: El editor modificará los datos de alguna implementación de paquete. Autores y editores del paquete podrán ser agregados o quitados de la implementación.

Precondición: El visitante debe iniciar sesión y presionar **Ver mis Paquetes** en el menú **Paquetes**.

Flujo:

Actor		Sistema		
Paso	Acción	Paso	Acción	Excepción
1	Presionar la opción Editar .	2	Mostrar formulario de modificación de implementación paquete.	E1
3	Modificar los datos que se requiera. NOTA: en este punto se puede modificar los autores y editores de la implementación de paquete.			
4	Presionar la opción Actualizar .	5	Validar los datos.	E2
		6	Almacenar las modificaciones de la implementación de paquete.	E1
		7	Mostrar mensaje de éxito en la actualización.	

Excepciones:

ID	Nombre	Acción
E1	Error de base de datos.	Se mostrará un mensaje informando que ocurrió algún problema.

Definición de Requerimientos

E2	Datos inválidos.	Se informará si alguno(s) de los datos son inválidos. Se verifica longitud mínima, si el datos es obligatorio y formato.
----	------------------	--

Poscondición: Los datos de la implementación del paquete habrán sido modificados. Además se tendrán los autores y editores correctos.

Caso de uso: Baja implementación paquete (ABCC de implementación de paquete)

Actor: Editor

Diagrama de caso de uso:



Descripción: El editor eliminará una implementación de paquete.

Precondición: El visitante debe iniciar sesión y desear dejar de ofrecer una implementación de paquete de la cual el editor sea autor.

Flujo:

Actor		Sistema		
Paso	Acción	Paso	Acción	Flujo Alternativo
1	Presionar la liga Ver mis Paquetes en el menú Paquetes .	2	Mostrar tabla de las implementaciones de paquete de las cuales el editor sea autor.	
3	Seleccionar la implementación de paquete que se desee eliminar.	4	Indicar la selección de la implementación de paquete con un cambio de color.	
5	Elegir la opción de "Eliminar".	6	Mostrar mensaje de confirmación.	
6	Confirmar o abortar la eliminación de la implementación de paquete.		Eliminar la implementación del paquete.	E1 E2

Flujo Alternativo:

ID	Nombre	Acción
E1	Error de base de datos.	Se mostrará un mensaje informando que ocurrió algún problema.
E2	El usuario no confirma la	El sistema no realiza ninguna acción.

Definición de Requerimientos

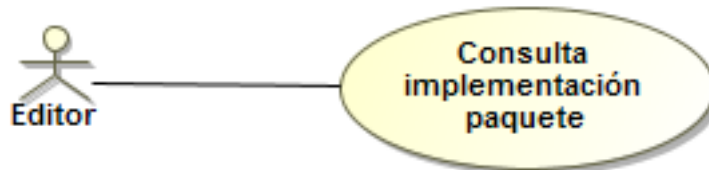
	eliminación de implementación de paquete.	
--	---	--

Poscondición: La implementación de paquete ya no se verá en el sitio y se eliminará de la base de datos.

Caso de uso: Consulta implementación paquete (ABCC de implementación de paquete)

Actor: Editor

Diagrama de caso de uso:



Descripción: El editor consultará los datos de las implementaciones de paquete que haya ingresado.

Precondición: El visitante debe iniciar sesión y desear ver los datos que proporcionó para alguna implementación de paquete específica.

Flujo 1 - Resumen de los datos:

Actor		Sistema		
Paso	Acción	Paso	Acción	Flujo Alternativo
1	Presionar la liga Ver mis Paquetes en el menú Paquetes .	2	Mostrar tabla de implementaciones de paquete que le pertenezcan al autor.	E1

Flujo 2 - Datos completos de la implementación de paquete:

Actor		Sistema		
Paso	Acción	Paso	Acción	Flujo Alternativo
1	Presionar la liga Ver mis Paquetes en el menú Paquetes .	2	Mostrar tabla de implementaciones de paquete que le pertenezcan al autor.	E1
3	Presionar la liga Ver de la tabla de implementaciones de paquete.	4	Mostrar datos completos de la implementación de paquete.	E1

Flujo Alternativo:

Definición de Requerimientos

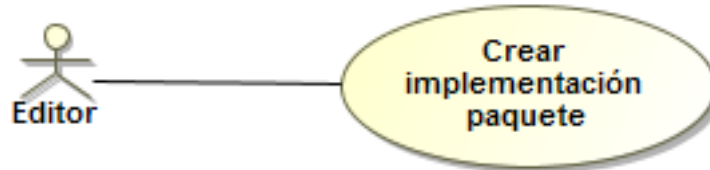
ID	Nombre	Acción
E1	Error de base de datos.	Se mostrará un mensaje informando que ocurrió algún problema.

Poscondición: El editor observará las implementaciones de paquete que haya proporcionado.

Caso de uso: Crear implementación paquete (ABCC de implementación de paquete)

Actor: Editor

Diagrama de caso de uso:



Descripción: El editor agregará una nueva implementación de paquete.

Precondición: El visitante debe iniciar sesión y tener la intención de agregar una nueva implementación de paquete.

Flujo:

Actor		Sistema		
Paso	Acción	Paso	Acción	Flujo Alternativo
1	Presionar la liga Implementar paquete en el menú Paquetes .	2	Mostrar formulario de implementar paquete.	
3	Ingresar los datos requeridos.			
4	Presionar la opción Crear .	5	Validar los datos.	E1 E2 E3
		6	Almacenar una nueva implementación de paquete.	
		7	Mostrar mensaje de confirmación.	

Flujo Alternativo:

ID	Nombre	Acción
E1	Datos inválidos.	Se informará si alguno(s) de los datos son inválidos. Se verifica longitud mínima, si el datos es obligatorio y formato.
E2	Clave o Correo electrónico ya utilizado.	Se informará un mensaje informando que algún dato ya ha sido utilizado por otro editor.

Definición de Requerimientos

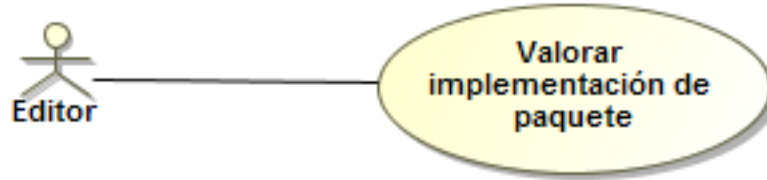
E3	Error de base de datos.	Se mostrará un mensaje informando que ocurrió algún problema.
----	-------------------------	---

Poscondición: El editor será autor de una nueva implementación de paquete.

Caso de uso: Valorar implementación de paquete

Actor: Editor

Diagrama de caso de uso:



Descripción: El editor asignará una valoración entre 0 a 5 según considere el trabajo de los autores en esta implementación de paquete.

Precondición: El editor verá en pantalla alguna implementación de paquete que quiera valorar.

Flujo:

Actor		Sistema		
Paso	Acción	Paso	Acción	Flujo Alternativo
1	Asignar el número de estrellas que desee.	2	Mostrar mensaje de confirmación de valoración mostrando el valor que seleccionó.	E1
		3	Mostrará mensaje de confirmación de la valoración.	
		4	Actualizar el número de participantes y la valoración promedio de la implementación de paquete.	

Flujo Alternativo:

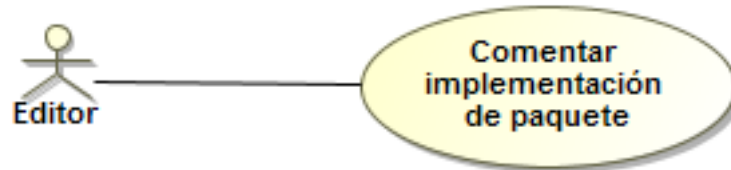
ID	Nombre	Acción
E1	Error de base de datos.	Se mostrará un mensaje informando que ocurrió algún problema.

Poscondición: El número de participantes y el valor promedio de la implementación de paquete reflejará la valoración del editor.

Caso de uso: Comentar implementación de paquete

Actor: Editor

Diagrama de caso de uso:



Descripción: El editor puede publicar uno o varios comentarios de las implementaciones de paquete.

Precondición: El visitante debe iniciar sesión y tener una implementación de paquete concreta para realizar el comentario.

Flujo:

Actor		Sistema		
Paso	Acción	Paso	Acción	Flujo Alternativo
1	Presionar la liga Comentar .	2	Mostrar una ventana con el formulario para realizar un comentario y los comentarios anteriores.	
3	Proporcionar un comentario.			
4	Presionar Enviar comentario .	5	Guardar comentario.	E2
		6	Publicar el comentario.	
		7	Mostrar mensaje de publicación exitosa.	E1

Excepciones:

ID	Nombre	Acción
E1	Datos inválidos.	Se informará si alguno(s) de los datos son inválidos. El comentario es obligatorio.
E2	El editor no proporciona texto para el comentario.	El sistema envía un mensaje informando que es necesario introducir algún texto.

Poscondición: Comentario publicado en el sitio.

5. Diseño

5.1 Definición de la arquitectura

Se eligió el estilo arquitectónico de "Llamada y Retorno" a partir de las especificaciones ya que se quiere tomar especial atención en los atributos de calidad que éste estilo tiene asociados. La escalabilidad y modificabilidad son los dos atributos que se desea alcanzar ya que se espera que la aplicación crezca.

Dentro de los patrones de arquitectónicos relacionados con el estilo, se eligió el Modelo-Vista-Controlador (Model-View-Controller). Este patrón divide la aplicación en tres componentes: Modelo (Model), contiene la información central y los datos; Vistas (View), despliegan información al usuario; Controladores (Controller), captura la entrada del usuario. Pone hincapié en atributos de calidad de funcionalidad y mantenibilidad.

Finalmente se usarán los siguientes patrones de diseño:

- Fábrica Abstracta: Con este patrón se intenta aislar el código de negocio de las implementaciones de los datos y más importante aún se pretende tener cierta independencia del modelo de almacenamiento de datos.
- Data Access Object (DAO): Con la implementación de este patrón se quiere tener encapsulado el acceso de los datos, además de tener independencia entre éstos y la lógica de negocio.

Un diagrama del patrón MVC es el siguiente:

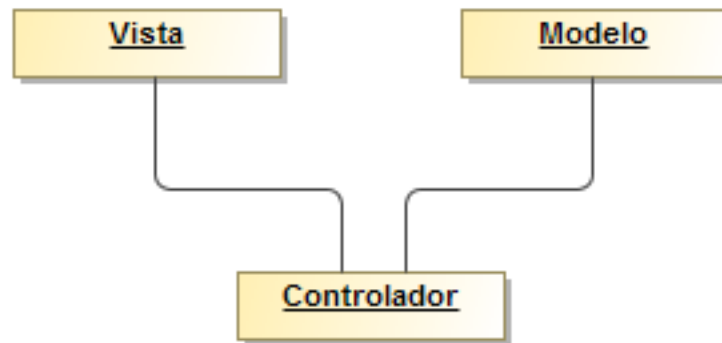


Ilustración 8 MVC

Los patrones de diseño fueron usados para el componente del modelo, para la persistencia y consulta de datos. La siguiente figura describe el uso de éstos:

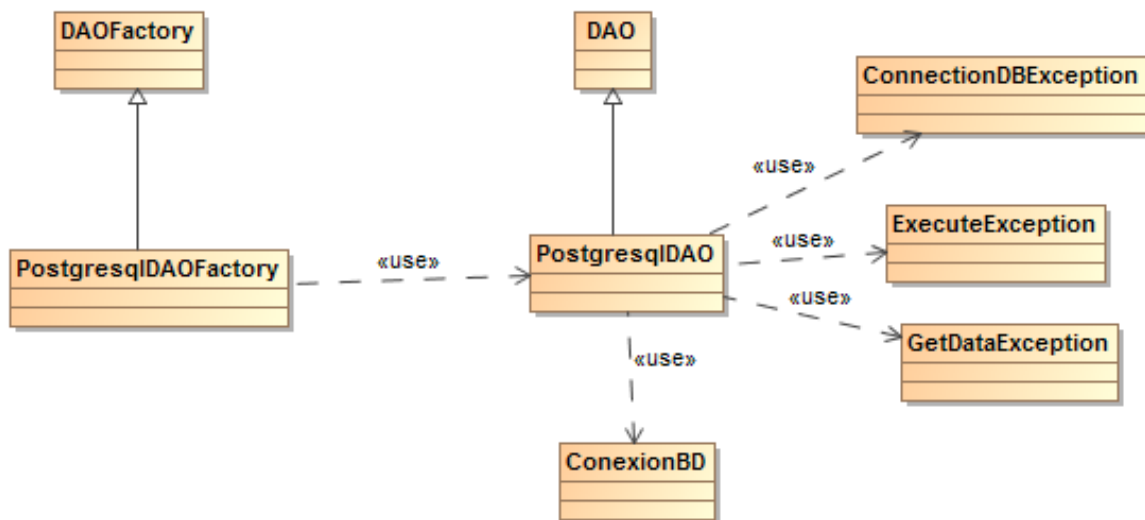


Ilustración 9 Diagrama de la Arquitectura

Tenemos una fábrica abstracta, DAOFactory es una clase abstracta en la cual definimos 2 cosas:

- La instancia concreta de la fábrica. Para este caso específico PostgresqlDAOFactory.
- Los métodos que deberá implementar la fábrica concreta. Métodos "get" que retornan interfaces DAO.

PostgresqlDAOFactory: Hereda de DAOFactory, retorna instancias concretas (implementaciones) de las interfaces DAO, en esta caso PostgresqlDAO.

DAO: Interfaces que definen las operaciones que se realizarán sobre los datos. Por ejemplo: crear, consultar, actualizar y borrar.

PostgresqlDAO: Implementaciones concretas para la manipulación de los datos con el manejador de bases de datos Postgresql. Sólo estas clases tienen SQL (StructuredQueryLanguage).

ConexionDB: Clase encargada de realizar la conexión a la base de dato. Sólo esta clase puede realizar esta función.

ConnectionDBException, ExecuteExcepcion, GetDataException: Excepciones que pueden ser lanzadas al realizar alguna operación con la base de datos.

5.2 Diagramas de paquetes

Para entender el diseño de este Sistema Manejador, se presentan los diagramas de paquetes. Estos pueden dividirse en 4 grandes grupos:

1. Persistencia - Modelo: En este paquete se encuentran las clases referentes a la persistencia y consulta de datos.

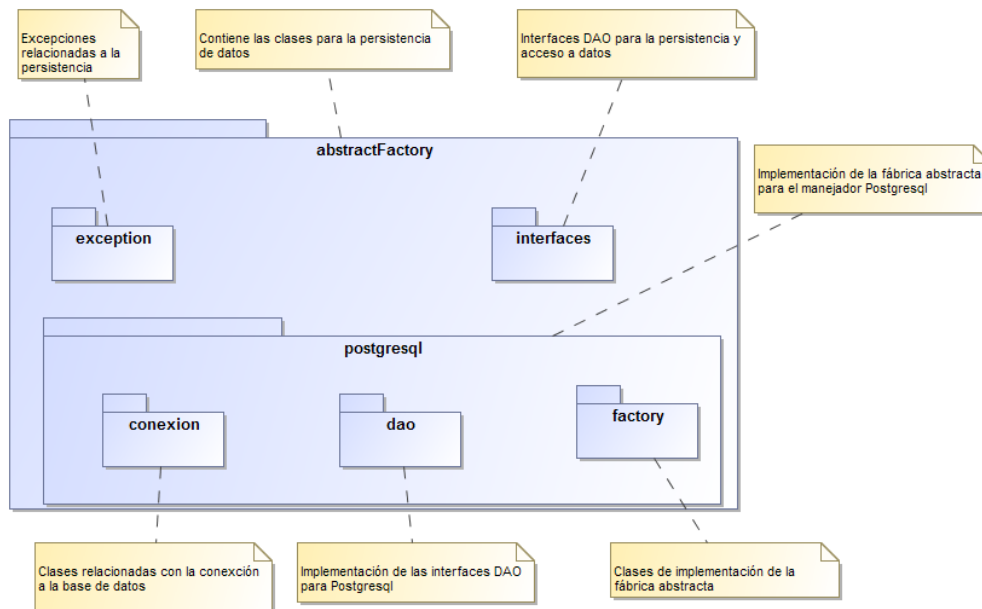


Ilustración 10 Diagrama de persistencia

- Acciones - Controlador: Se tienen las clases que realizan todas las funciones del sitio. Cada sub-package tiene el nombre de la tabla que afecta.

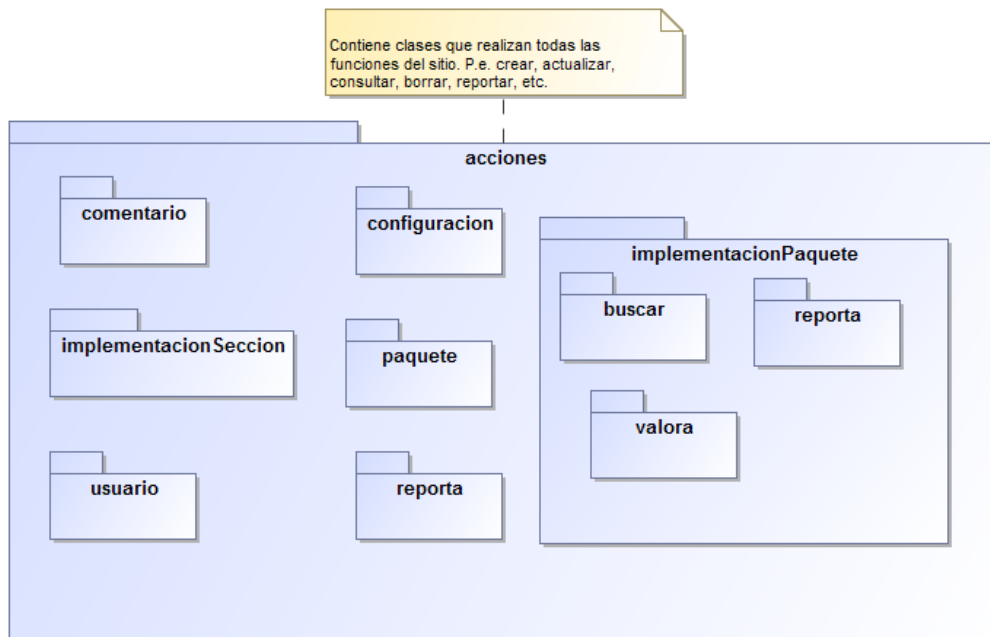


Ilustración 11 Diagrama de acciones

- Presentación - Vista: Contiene los recursos con la lógica para que la información sea visualizada por los diferentes actores. P.e. imágenes, páginas jsf, archivos js, etc.

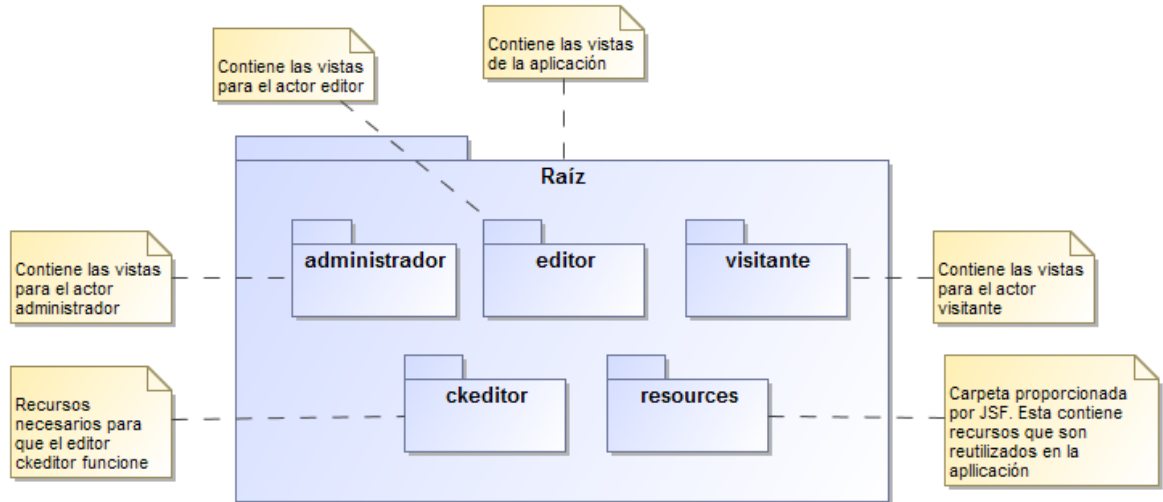


Ilustración 12 Diagrama de presentación

- Estos paquetes tienen un menor número de clases y ofrecen funcionalidad a los dos anteriores.

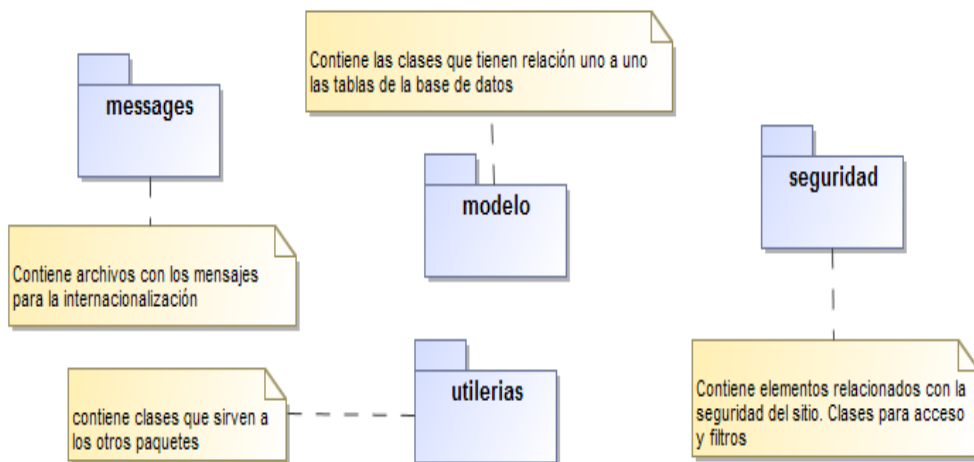


Ilustración 13 Diagrama de otros paquetes

5.3 Persistencia

Para seleccionar la forma de persistencia se evaluaron 2 opciones: Bases de datos XML y Bases Relacionales. Dada la naturaleza de los datos que se quieren almacenar:

- Tienen una estructura bien definida: la cual consta de 12 secciones. Aunque no son obligatorias no se agregarán más (Por el momento).
- Los tipos están bien definidos:
- Existe una relación bien definida entre los datos:

Tomando en cuenta las características anteriores, se optó por almacenar los datos en una base de datos relacional, ya que cumple muy bien con las características de datos que guardan este tipo de bases (Ver Capítulo 1- 1.4 Tipos de datos de acuerdo a su estructura y almacenamiento).

No obstante a la decisión tomada respecto al almacenamiento de información en una base de datos relacional, dada la arquitectura seleccionada y específicamente la implementación de los dos patrones de diseño, DAO y Fábrica Abstracta, es posible cambiar el modelo de almacenamiento de información con relativa facilidad, reusando el código de las vistas y los controladores, solo es necesario volver a implementar la fábrica tomando en cuenta las particularidades de la persistencia no estructurado o semiestructurado.

El diseño de la base de datos está modelado con los diagramas de Entidad-relación y relacional.

6. Construcción del Sistema Manejador de Paquetes de Puesta en Operación

6.1 Tecnología a usar

Para desarrollar el Sistema Manejador de Puesta en Operación se usaron tecnologías novedosas que a continuación se describen.

6.1.1 JavaServer Faces

JavaServer Faces es un marco de trabajo de interfaces de usuario del lado del servidor para aplicaciones Web basadas en Java (26).

Los principales componentes de JavaServer Faces (JSF) son:

- Un API para representar componentes UI⁴(Interfaz de usuario) y manejar su estado; manejo de eventos, validación del lado del servidor y conversión de datos; definición de la navegación entre páginas; soporte de internacionalización y accesibilidad; además de proporcionar extensibilidad para todas estas características.
- Dos bibliotecas de etiquetas JavaServerPages (JSP) personalizadas para dibujar componentes UI dentro de una página JSP.

Este modelo de programación bien definido y las bibliotecas de etiquetas para componentes UI facilitan de forma significativa la tarea de la construcción y mantenimiento de aplicaciones Web con UI's del lado del servidor. Con un mínimo esfuerzo, podemos:

- Agregar componentes en la página añadiendo etiquetas.
- Mapear eventos generados por los componentes a código del lado del servidor.
- Construir componentes reutilizables y extensibles.
- Guardar y restaurar el estado de la interfaz de usuario más allá de las peticiones al servidor.

⁴Medio con que el usuario puede comunicarse con una máquina, equipo o computadora y comprende todos los puntos de contacto entre el usuario y el equipo.

Beneficios de JSF

Un beneficio de JavaServer Faces es su clara separación entre el comportamiento y la presentación, permitiendo que el desarrollo pueda llevarse paralelamente, además de proporcionar un sencillo modelo de programación para enlazar todas las piezas.

Otro objetivo importante de la tecnología JavaServer Faces es mejorar los conceptos familiares de componente-UI y capa-Web sin limitarnos a una tecnología de script particular o un lenguaje de marcas. Aunque la tecnología JavaServer Faces incluye una librería de etiquetas JSP personalizadas para representar componentes en una página, JSF ha sido creado directamente como Servlets

Pero lo más importante, la tecnología JavaServer Faces proporciona una rica arquitectura para manejar el estado de los componentes, procesar los datos, validar la entrada del usuario, y manejar eventos.

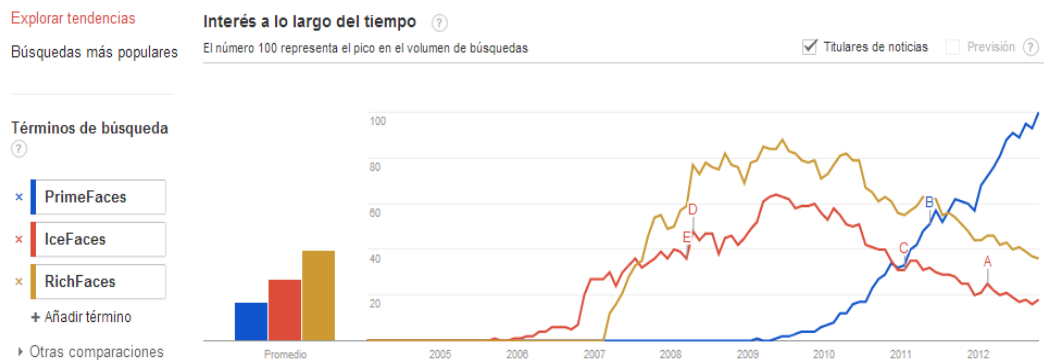
6.1.2 PrimeFaces

PrimeFaces es un conjunto de componentes JSF de código abierto, que cuenta con varias extensiones.

Las principales características son:

- Gran número de componentes: Al trabajar con este marco de trabajo se encontrará con un gran número de componentes (aproximadamente 117), como pueden ser editor HTML, calendarios, tablas, autocompletado, diálogos, etc.
- Soporte nativo para Ajax.
- Fácil de instalar y sin necesidad de configuración: Esta biblioteca está desarrollada con el principio de "Un buen componente de interfaz gráfica debe ocultar la complejidad, pero manteniendo la flexibilidad".
- Kit para crear aplicaciones Web para móviles.
- Temas prediseñados.
- Extensa documentación: Documentación en libro para descarga libre y ejemplos disponibles para el público en general.

- Rendimiento: PrimeFaces es una biblioteca donde todas las decisiones están pensadas en mantenerla lo más ligera posible (tratando de usar los menos recursos posibles). En internet se encuentran varias pruebas de desempeño de la herramienta colocándola por encima de sus competidoras (27).
- Tendencia de uso: La siguiente grafica obtenida de Google Trends (obtenida el 19 de Septiembre de 2012) muestra las frecuencias con las que se ha buscado PrimeFaces contra sus competidores (IceFaces y RichFaces).



Puede apreciarse un alza en la tendencia de la búsqueda de este marco de trabajo.

6.2 Aplicación de las tecnologías al proyecto

JavaServer Faces proporciona un entorno de trabajo (Framework) vía web que gestiona las acciones producidas por el usuario en su página HTML y las traduce a eventos que son enviados al servidor con el objetivo de regenerar la página original y reflejar los cambios pertinentes provocados por dichas acciones. Por su parte PrimeFaces permitió integrar componentes muy completos además de un tema prediseñado.

Específicamente en el desarrollo del proyecto usamos las siguientes características:

- JSF proporcionó el patrón de diseño MVC (Model-View-Controller, Modelo-Vista-Controlador).
- Manejo de eventos:
- Validación de datos: longitud, tipo, si es obligatorio un campo con validaciones proporcionadas.

- Internacionalización (i18n): los mensajes de toda la aplicación se encuentran en estos archivos, por lo que añadir un nuevo idioma no resulta complicado.
- Introducción de Javascript en la página: usado para AJAX (Asynchronous JavaScript And XML), validaciones, etc.
- Tener un archivo de disposición (Layout): Con esto tenemos un reuso de código y definimos un esquema de distribución de los elementos que conforman las páginas.
- Componentes: En el sitio se cuentan con varios componentes como son: tablas con paginación para presentar los elementos, autocompletado de palabras, ventanas de diálogos (para la retroalimentación), editor HTML, componente para subir archivos, barra de progreso, marcas de agua, Menús, elementos de navegación (BreadCrumb) y elementos de formulario (aunque JSF los proporciona, se usaron los de PrimeFaces).

7. Principales funcionalidades de la aplicación

Se mostrarán paso a paso como realizar las funciones que se plantearon en el objetivo para la aplicación construida, así como el actor que las puede llevar a cabo.

7.1 Búsqueda de Implementaciones de Paquetes

Visitante

Para realizar una búsqueda debe seleccionarse la opción **Buscar**.



En el formulario que aparece debe proporcionar algún criterio de búsqueda y presionar **Buscar**.

🏠 ▶ Buscar

Encuentra el paquete que mejor se adapte a tus necesidades

Proporciona los datos para encontrar el paquete.

Paquete:

Metodología:

Requisitos:

Palabras clave:

Autores:

Se mostrará una tabla con los resultados de la búsqueda.

Principales funcionalidades de la aplicación

Nombre	Metodología	Fecha creación	Palabras clave	Autores	Ver secciones
Verificación y validación	metodología	2012-07-19	palabras	Francisco Javier	Ver

Búsqueda con filtros de implementaciones de paquetes - Editor y Administrador

Los filtros están disponibles en la parte superior de las tablas de presentación de las implementaciones de paquetes.

Nombre	Metodología	Fecha creación	Palabras clave	Idioma	Autores	Editores	Acciones
Selecciona							

Para que realizar el filtro de información, basta seleccionar alguna opción es los campos de selección o escribir alguna palabra en el caso de los campos de texto.

Nombre	Metodología	Fecha creación	Palabras clave	Idioma	Autores	Editores	Acciones
Verificación y validación	metodología	2012-07-19	palabras	español	Francisco Javier		Ver Editar

[x Eliminar](#)

Nombre	Metodología	Fecha creación	Palabras clave	Idioma	Autores	Editores	Acciones
Verificación y validación	ágil	2012-07-19	palabras	español	Francisco Javier		Ver Editar

[x Eliminar](#)

7.2 Crear Implementaciones de Paquetes - Editor, Administrador

Esta operación se realiza en 2 pasos:

1. Proporcionar los metadatos del paquete.
Debe seleccionarse la opción "Implementar paquete".



Lenar los campos requeridos y seleccionar la opción "Crear".

Paquete: * Administración del proyecto

Metodología: * metodología

Requisitos: * requisito1, requisito2

Palabras clave: * palabra1, palabra2

Visible: *

* Campo obligatorios

Crear

2. Proporcionar el contenido de las secciones.

Se debe buscar entre sus implementaciones de paquete y seleccionar "Ver"

Nombre	Metodología	Fecha creación	Palabras clave	Idioma	Autores	Editores	Acciones
Administración del proyecto	metodología	2012-10-14	palabra1, palabra2	español	Francisco Javier		Ver Editar

Eliminar

Seleccionar la sección que se quiera modificar y seleccionar "Editar".

Descripción técnica Definiciones Relación con la ISO/IEC 29110 Resumen de procesos, actividades, tareas, roles y productos

Descripción de procesos, actividades, tareas, pasos, roles y productos Plantilla Ejemplo Lista de verificación Herramientas

Referencias a otros estándares y modelos Referencias Formulario de evaluación

Editar

Con ayuda del editor introducir el texto deseado y seleccionar "Actualizar".

Estilo Normal Fuente Ta... A+ A- [?]

Purpose of this document

This Deployment Package (DP) supports the Basic Profile as defined in ISO/IEC 29110 Part 5-1: Management and Engineering Guide. A DP is a set of artefacts developed to facilitate the implementation of a set of practices in a Very Small Entity (VSE). A DP is not a process reference model (i.e. it is not prescriptive). The elements of a typical DP are: description of processes, activities, tasks, roles and products, template, checklist, reference to standards, etc.

Information contained in this DP is applicable to VSEs that do not develop critical software, commercial off the shelf software products. For a project with intensive verification and validation (V&V) activities, the use of the appropriate standards and guides are recommended (ex. IEEE 1012, DO-178B, etc.)

body p span

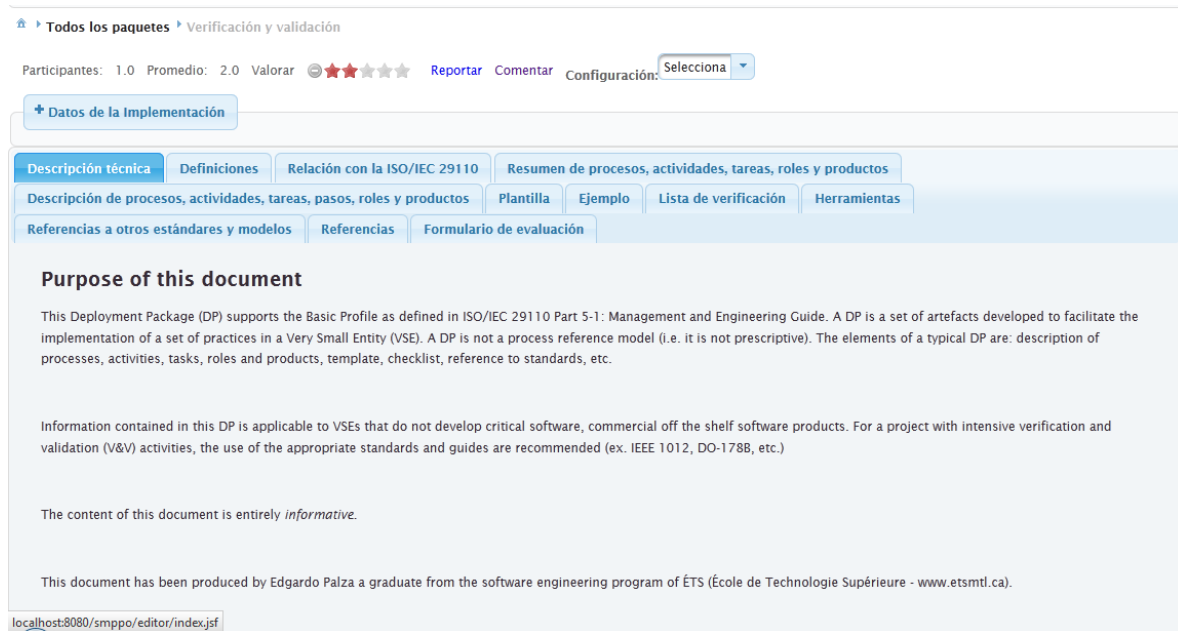
Actualizar

El proceso anterior se repite con el texto de las secciones que desea modificar.

7.3 Consultar Implementaciones de Paquetes - Visitante Editor, Administrador

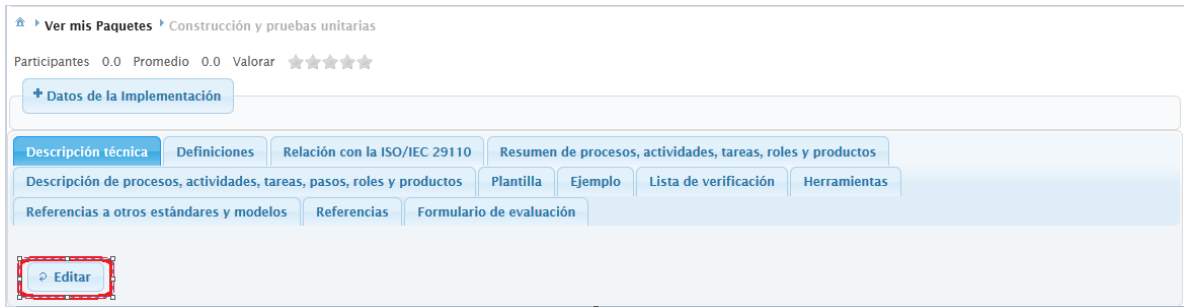
Esta acción puede realizarla todos los usuarios, la diferencia radicar  en los permisos que tengamos sobre las implementaciones que se visualizar n. Si consultamos alguna implementaci n de paquete nuestra, podremos editar el contenido, de lo contrario s lo se podr  consultarlo.

Primeramente debe buscarse la implementaci n que se desea consultar. Ya localizado debe seleccionarse la opci n de **ver**.



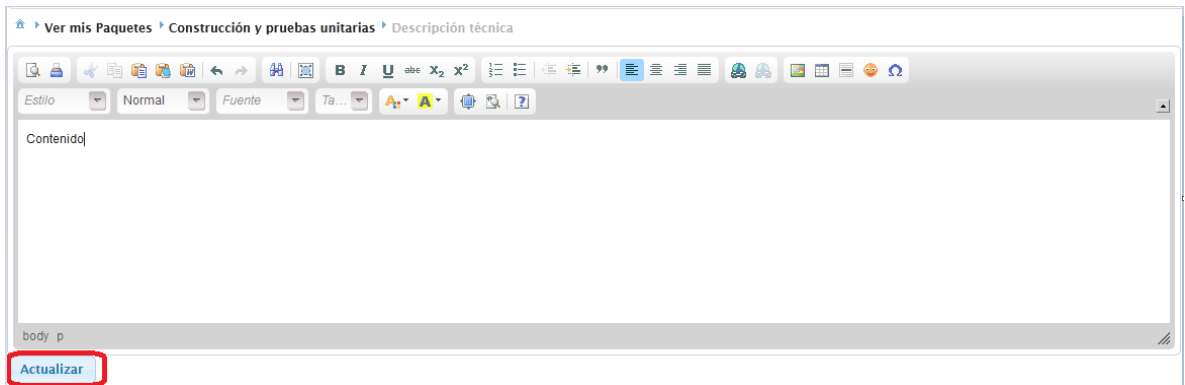
7.4 Actualizar Implementaciones de Paquetes - Editor, Administrador

Esta acci n solo puede realizarse con los permisos adecuados. Primeramente debe buscarse la implementaci n que se desea consultar. Ya localizado debe seleccionarse la opci n de **ver**.



Se debe seleccionar la implementación de sección que se desee actualizar y seleccionar **Editar**.

Finalmente con ayuda de editor se modifica el contenido y se seleccionar **Actualizar**.



7.5 Eliminar Implementaciones de Paquetes - Editor, Administrador

Esta acción solo puede realizarse con los permisos adecuados. Primeramente debe buscarse la implementación que se desea eliminar. Ya localizado debe seleccionarse la implementación y seleccionar la opción **Eliminar**.



Note como la implementación seleccionada cambia de color.

7.5 Retroalimentación de buenas prácticas de Implementaciones de Paquetes - Editor, Administrador

Para realizar esta acción se debe estar visualizando una implementación de paquete.

El menú superior contiene las dos formas en que se puede retroalimentar al autor y/o otros usuarios acerca de la utilidad de la implementación de paquete.

Participantes: 1.0 Promedio: 2.0 Valorar ☆☆☆☆☆ Reportar Comentar Configuración: Selecciona

+ Datos de la Implementación

Descripción técnica Definiciones Relación con la ISO/IEC 29110 Resumen de procesos, actividades, tareas, roles y productos

Descripción de procesos, actividades, tareas, pasos, roles y productos Plantilla Ejemplo Lista de verificación Herramientas

Referencias a otros estándares y modelos Referencias Formulario de evaluación

Purpose of this document

This Deployment Package (DP) supports the Basic Profile as defined in ISO/IEC 29110 Part 5-1: Management and Engineering Guide. A DP is a set of artefacts developed to facilitate the implementation of a set of practices in a Very Small Entity (VSE). A DP is not a process reference model (i.e. it is not prescriptive). The elements of a typical DP are: description of processes, activities, tasks, roles and products, template, checklist, reference to standards, etc.

Information contained in this DP is applicable to VSEs that do not develop critical software, commercial off the shelf software products. For a project with intensive verification and validation (V&V) activities, the use of the appropriate standards and guides are recommended (ex. IEEE 1012, DO-178B, etc.)

The content of this document is entirely *informative*.

This document has been produced by Edgardo Palza a graduate from the software engineering program of ÉTS (École de Technologie Supérieure - www.etsmtl.ca).

De izquierda a derecha las opciones o datos mostrados son:

- Participantes: es el número de participantes que han valorado la implementación de paquete.
- Promedio: es la valoración promedio de la implementación de paquete.
- Valorar: si se desliza el ratón por las estrellas y valoramos la implementación.
- Reportar: en caso de que la implementación no sea apropiada (sea ofensivo, este violando algún derecho, etc.) puede reportarse para que no sea mostrado en la aplicación.
- Comentar: El actor podrá proporcionar un comentario acerca de la implementación de paquete.

7.6 Otras funcionalidades

7.6.1 Añadir implementación de paquete a configuración - Editor y Administrador

Para realizar esta actividad es necesario haber creado una configuración⁵.

Se debe tener identificada la implementación de paquete que se desea agregar a la configuración y seleccionar la opción "Ver".



Metodología	Fecha creación	Palabras clave	Requisitos	Autores	Editores	Ver secciones
ágil	2012-07-19	palabras	requisitos	Francisco Javier		^ Ver

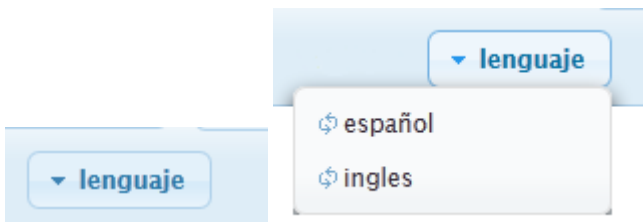
En la pantalla siguiente debe seleccionarse del combo la configuración a la que se desea añadir la implementación del paquete.



7.6.2 Idiomas de la aplicación - Todos los usuarios

Los usuarios pueden elegir el idioma en que se mostrará la información propia de la aplicación, esto no incluye el contenido proporcionados por lo usuarios.

Para cambiar el idioma de la aplicación basta con seleccionar el idioma que se prefiera en el menú.



⁵ Ver configuración en el glosario

Login Password Enter
Recover password register language

Clave Contraseña Entrar
Recuperar contraseña registro lenguaje

Conclusiones

Es clara la necesidad del uso de estándares como la norma ISO/IEC 29110 para las pequeñas organizaciones. Sin embargo estas empresas no siempre cuentan con los recursos (financieros o humanos) para poner en marcha algún modelo o norma. Para ello se crearon los Paquetes de Puesta en Operación (Deployment Package), para ayudar a la implementación y proporcionar orientación sobre la aplicación efectiva de esta norma ISO para pequeñas organizaciones.

Se planteó como objetivo de esta tesis desarrollar un Sistema Manejador de Paquetes de Puesta en Operación que permita a las organizaciones desarrolladoras de software buscar, crear, consultar, actualizar, eliminar, migrar las implementaciones de Paquetes de Puesta en Operación, retroalimentar las buenas prácticas para organizar sus propios procesos y mejorar de las implementaciones de los paquetes, además se pretendió desarrollar un sistema modificable, escalable y funcional.

Para cumplir con este objetivo primeramente se definió y puntualizó las características de Sistema Manejador de Puesta en Operación basándose en otras definiciones como Sistema Manejador de Bases de Datos.

Habiendo definido Sistema Manejador de Puesta en Operación - SMMPO como: *"Un Sistema Manejador de Paquetes de Puesta en Operación - SMPPO es una aplicación que aloja, permite el acceso y administra de Paquetes de Puesta en Operación - PPO"*.

Se procedió al desarrollo de una aplicación que satisfaga esta definición y cumpliera los objetivos propuestos del presente trabajo.

Para cada uno de los puntos del objetivo se obtuvo lo siguiente:

- Buscar o consultar implementaciones de paquetes: Se tienen centralizadas las implementaciones de paquetes, por lo que una pequeña organización tiene a su disposición un gran número de paquetes para realizar búsquedas. La aplicación cuenta

con una sección búsqueda estableciendo criterios, además de diferentes filtros para paquetes.

- Crear y actualizar paquetes: La aplicación proporciona la estructura para los PPO, por lo que permanecerá estándar, además para facilitar estas acciones se proporciona un editor.
- Valorar las buenas prácticas: Las buenas prácticas son valoradas por los usuarios, lo que permite a las organizaciones tener una referencia de las mejores prácticas para implementar la norma y a los autores les permite mejorar su implementación de paquete.

Algunos beneficios adicionales que proporciona la aplicación son los siguientes:

- Los paquetes no sólo son desarrollados por un pequeño grupo de personas, sino por los usuarios que ingresen al sitio.
- La migración de los paquetes actualmente creados (encontrados como archivos DOC) no resulta complicada para el usuario.
- La aplicación puede traducirse a varios idiomas (los mensajes propios de la aplicación, no el contenido de la implementación de los paquetes).

Trabajo Futuro

Se propone como trabajo futuro para la continuación de esta tesis lo siguiente:

- Validar con más Paquetes de Puesta en Operación la funcionalidad correcta de la aplicación.
- Identificar y agregar si es necesario nuevos metadatos a los paquetes.
- Hacer una prueba con usuarios y en base a los resultados, realizar las correcciones y/o añadir la funcionalidad que los usuarios esperaban o mejorar la existente.

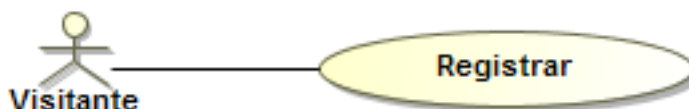
Apéndices

Apéndice A. Detalle de Casos de uso

Caso de uso: Registrar

Actor: Visitante

Diagrama de caso de uso:



Descripción: El visitante se registrará en la aplicación proporcionando algunos datos personales, un identificador único o clave de usuario, correo electrónico entre otros.

Precondición: El visitante debe encontrarse en el sitio.

Flujo:

Actor		Sistema		
Paso	Acción	Paso	Acción	Flujo Alternativo
1	Presionar el botón registro localizado en la cabecera.	2	Mostrar el formulario de Registro de usuario .	
3	Proporcionar los datos solicitados.			
4	Presionar el botón Registrar .	5	Validar los datos.	E1 E2 E3
		6	Almacenar el nuevo editor.	
		7	Mostrar mensaje de confirmación de registro.	

Flujo alternativo:

ID	Nombre	Acción
E1	Datos inválidos.	Se informará si alguno(s) de los datos son inválidos. Se

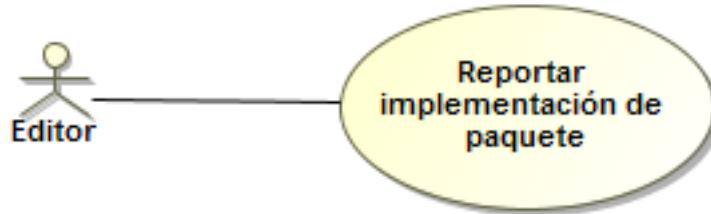
		verifica longitud mínima, si el datos es obligatorio y formato.
E2	Clave o Correo electrónico ya utilizado.	Se informará un mensaje informando que algún dato ya ha sido utilizado por otro editor.
E3	Error de base de datos.	Se mostrará un mensaje informando que ocurrió algún problema.

Poscondición: El visitante se habrá registrado, por lo que podrá acceder a funciones del tipo de usuario editor.

Caso de uso: Reportar implementación de paquete

Actor: Editor

Diagrama de caso de uso:



Descripción: El editor reportará una implementación de paquete proporcionando un comentario el cual debe describir de forma breve el porqué lo reporta. La implementación de paquete reportada no se mostrará en el sitio.

Precondición: El visitante debe iniciar sesión y ubicar un paquete el cual contenga información que considere inapropiada.

Flujo:

Actor		Sistema		
Paso	Acción	Paso	Acción	Flujo Alternativo
1	Ubicar una implementación de paquete que contenga información inapropiada.			
2	Presionar la liga Reportar .			
3	Escribir un comentario argumentando el porqué realiza el reporte.			
4	Presionar Enviar .	5	Mostrar mensaje de que la implementación de paquete ha sido reportada.	E1
		6	Mostrar las implementaciones de paquetes.	

Flujo Alternativo:

ID	Nombre	Acción

E1	Error de base de datos.	Se mostrará un mensaje informando que ocurrió algún problema.
----	-------------------------	---

Poscondición: La implementación de paquete no será mostrara a los visitantes o editores.

Caso de uso: Liberar implementación de paquete reportada

Actor: Administrador

Diagrama de caso de uso:



Descripción: El administrador eliminará el reporte de alguna implementación de paquete.

Precondición: El visitante debe iniciar sesión con una cuenta con privilegios de administrador. Debe existir por lo menos una implementación de paquete reportada.

Flujo:

Actor		Sistema		
Paso	Acción	Paso	Acción	Flujo Alternativo
1	Presionar la liga Paquetes reportados en el menú Paquetes .	2	Muestra una lista con los paquetes reportados.	E1
3	Presionar la liga Liberar de la implementación que se desee.	4	El reporte es eliminado.	
		5	El estado de la implementación de paquete es actualizado.	
		6	Un mensaje confirmando la liberación es mostrado.	E2

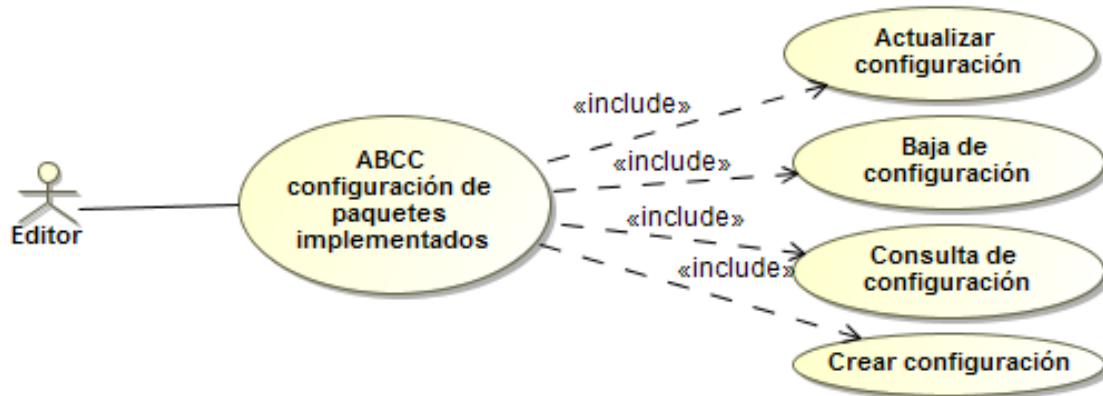
Flujo Alternativo:

ID	Nombre	Acción
E1	No existen paquetes reportados.	Se muestra un mensaje informando que no hay paquetes reportados.
E2	La implementación de paquete no puede ser liberada.	Se muestran un mensaje informando que no fue posible liberar la implementación de paquete.

Caso de uso: ABCC configuración de paquetes implementados

Actor:

Diagrama de caso de uso:



Descripción: El editor podrá actualizar, borrar, consultar y crear configuraciones de paquetes implementados.

Precondición: El visitante debe iniciar sesión y desear realizar alguna operación relacionada con configuraciones de paquetes.

Flujo:

Actor		Sistema		
Paso	Acción	Paso	Acción	Excepción
1	Para llevar a cabo alguna de las operaciones el editor debe seleccionar algunas de las opciones: Nueva Configuración, Ver mis Configuraciones,	2	Mostrar la interfaz correspondiente a la opción seleccionada.	
Los detalles serán omitidos.				

Poscondición: Configuraciones que el editor desee tener.

Caso de uso: Consultar y Borrar Editores.

Actor: Administrador

Diagrama de caso de uso:



Descripción: El administrador podrá revisar los datos de los editores y en caso de ser necesario eliminar alguna cuenta.

Precondición: El visitante debe iniciar sesión con una cuenta con privilegios de administrador.

Flujo:

Actor		Sistema		
Paso	Acción	Paso	Acción	Flujo Alternativo
1	Presionar la liga Todos los usuarios en el menú Cuentas .	2	Mostrar tabla de todos los usuarios del sitio.	
3	Seleccionar al usuario que se desee eliminar.			
4	Presionar la liga Eliminar en la parte inferior de la tabla.	5	Muestra mensaje pidiendo confirmación de la eliminación del usuario.	E1
		6	Elimina la cuenta y todo lo relacionado con ésta.	
		7	Actualizan los usuarios.	

Flujo Alternativo:

ID	Nombre	Acción
E1	Si el administrador no confirma.	El sistema no realiza ninguna acción.

Poscondición: Todos los datos (datos personales, implementación de paquetes de secciones, comentarios, valoraciones, configuraciones y reportes) del usuario eliminado han sido removidos de la base de datos. Ese usuario ya no podrá iniciar sesión.

Apéndice B. Diccionario de datos

Tabla: rol			
Descripción: Contiene los roles permitidos en la aplicación. Solo se tienen editores y administrador.			
Campo	Tipo	Restricciones	Descripción
Id_rol	serial	Llave primaria	
Nombre	charactervarying(30)		Nombre del rol.

Tabla: usuario			
Descripción: Contiene los usuarios registrados en el sitio.			
Campo	Tipo	Restricciones	Descripción
id_usuario	serial	llave primaria	
Nombre	charactervarying(75)		
Apellidos	charactervarying(75)		
Clave	charactervarying(10)	Unique	
Contraseña	charactervarying(12)		
Correo	charactervarying(75)	Unique	
id_rol	integer	llaveforánea, rol(id_rol)	Referencia del rol que tiene el usuario en la aplicación.
Descripcion	charactervarying(200)		
Institucion	charactervarying(200)		Institución a la que pertenece el usuario.
Pais	charactervarying(75)		
Editor	boolean		Permite el usuario crear o actualiza paquetes.

Tabla: valor			
Descripción: Contiene el valor numérico que el usuario asigna a una Implementación de Paquete según su apreciación de funcionalidad. El valor varía de [0 a 5].			
Campo	Tipo	Restricciones	Descripción
id_usuario	integer	llave primaria	
id_imple_paquete	integer	llave primaria	
Valor	integer		Valor numérico asignado por el usuario.
fecha_valoracion	date	default now()	

Tabla: reporta			
<p>Descripción: Si algún usuario considera que algún paquete tiene contenido que no debería estar en el sitio, puede reportarlo, ese reporte se guarda en esta tabla. El usuario podrá guardar las razones por la que reporta la implementación de paquete.</p> <p>Ese contenido no deseado puede ser: contenido con propiedad intelectual, ofensivo de alguna forma para la comunidad, o cualquier contenido no deseado por los usuarios.</p>			
Campo	Tipo	Restricciones	Descripción
id_usuario	integer	llave primaria	
id_imple_paquete	integer	llave primaria	
fecha_creacion	date	default now()	
Comentario	charactervarying(200)		Guarda el comentario por el cual el usuario reporta la implementación de paquete.

Tabla: usuario_imple_paquete			
<p>Descripción: Relaciona el usuario y la implementación de paquete. También contiene la función que desempeña en el paquete. Esta función puede ser autor o editor.</p>			
Campo	Tipo	Restricciones	Descripción
Id_usuario	integer	Llave primaria	
Id_imple_paquete	Integer	Llave primaria	
Id_funcion	integer	Llave foránea, funcion(id_funcion)	Función que tiene el usuario en esta implementación de paquete.

Tabla: funcion			
<p>Descripción: Contiene las funciones que puede tener un usuario dentro de un paquete. Solo se tienen 2 funciones autor y editor.</p>			
Campo	Tipo	Restricciones	Descripción
Id_funcion	Serial	Llave primaria	
Nombre	charactervarying(50)		

Tabla: comentario			
<p>Descripción: Contiene los comentarios hechos por los usuarios a las implementaciones de paquetes.</p>			
Campo	Tipo	Restricciones	Descripción

id_comentario	serial	Llave primaria	
id_usuario	Integer	Llave foránea, usuario(id_usuario)	
id_imple_paquete	Integer	Implementacion_paquete (id_imple_paquete)	
Comentario	charactervarying(500)		
fecha_creacion	Date	Default now()	

Tabla: configuracion			
Descripción: Para implementar la norma es necesario aplicar cada uno de los 9 paquetes actualmente existente eso es llamado configuración y es guardado en esta tabla. La también contiene un comentario que idealmente guarda la razón o comentarios que el usuario desea proporcionar por la cual creó esta configuración.			
Campo	Tipo	Restricciones	Descripción
id_configuracion	Serial	Llave primaria	
id_usuario	Integer	Llave foránea, usuario(id_usuario)	
Comentario	charactervarying(500)		
fecha_creacion	date	Default now()	
Nombre	charactervarying(50)		Nombre de la configuración. Sirve para que el usuario pueda identificar la configuración.

Tabla: idioma			
Descripción: Contiene los idiomas en los que puede estar la implementación. De inicio solo se consideran dos: español e inglés.			
Campo	Tipo	Restricciones	Descripción
id_idioma	Serial	Llave primaria	
Abreviatura	charactervarying(5)		Contiene la abreviatura del idioma. Español - es, inglés - en.
Nombre	charactervarying(50)		Nombre del idioma.

Tabla: implementacion_paquete			
Descripción: Contiene las implementaciones hechas por los usuarios de los paquetes			

actualmente definidos.			
Campo	Tipo	Restricciones	Descripción
id_imple_paquete	Serial	Llave primaria	
id_paquete	Integer	Llave foránea, paquete(id_paquete)	
metodologia	charactervarying(200)		
requisitos	charactervarying(200)		
palabras_clave	charactervarying(200)		
fecha_creacion	Date	Default now()	
Reporte	Text		
Visible	Boolean	Default true	
id_idioma	Integer	Llave foránea, idioma(id_idioma)	

Tabla: configuracion_imple_paquete			
Descripción: Contiene las implementaciones de paquetes que conforman cada configuración.			
Campo	Tipo	Restricciones	Descripción
id_configuracion	integer	Llave primaria, unique (id_configuracion, id_paquete)	
id_imple_paquete	Integer	Llave primaria	
id_paquete	integer	unique (id_configuracion, id_paquete)	

Tabla: paquete			
Descripción: Catalogo con los paquetes actualmente existentes.			
Campo	Tipo	Restricciones	Descripción
id_paquete	Serial	Llave primaria	
Nombre	charactervarying(100)		
Orden	integer	Unique	Indica el orden de los paquetes.

Tabla: implementacion_seccion			
Descripción: Contiene las implementaciones de sección hechas por los editores. Estas implementaciones de secciones deben ser instancias de las secciones que ya están definidas. Además cada implementación de sección debe pertenecer a una implementación de paquete.			
Campo	Tipo	Restricciones	Descripción
id_seccion	Integer	Llave primaria	

id_imple_paquete	Integer	Llave primaria	
Contenido	text		Almacena el contenido de cada una de las secciones. Este campo tendrá código HTML.

Tabla: seccion			
Descripción: Catálogo con las secciones de los paquetes. Estas secciones ya esta predefinidas.			
Campo	Tipo	Restricciones	Descripción
id_seccion	Serial	Llave primaria	
Nombre	charactervarying(100)		
Orden	integer	Unique	Indica el orden de las secciones.

Glosario

ABCC: Operaciones básicas (Actualiza, Borrar, Crear y Consultar) sobre entidades del software.

Administrador: Persona con más privilegios en la aplicación.

Configuración de paquete o configuración: Una de cada una de las implementaciones de paquete que conforman la norma.

Editor: Persona que se ha registrado, es decir, ha proporcionado los datos requeridos por el software.

Escalabilidad (Scalability): Es el grado con el que se puede ampliar el diseño arquitectónico, de datos o procedimental (4).

Funcionalidad (Functionality): Habilidad del sistema para realizar el trabajo para el cual fue concebido (5).

Implementación de paquete: Los Paquetes de Puesta en Operación para el perfil básico ya están definidos, pueden verse en el capítulo 3 de la presente tesis. Implementar un paquete es crear el contenido de las secciones predefinidas de alguno de esos paquetes.

Implementación de sección: Contenido de las secciones ya predefinidas. Las secciones de los paquetes de puesta en operación ya están definidas y se encuentran en el capítulo 2.

Modificabilidad (Modifiability): Es la habilidad de realizar cambios futuros en el sistema (3).

Observaciones a secciones de un paquete: Con el fin de estimular la constante mejora de los paquetes contenidos en el software, los usuarios pueden realizar observaciones a las secciones de los paquetes, con el fin de que su(s) autor(es) o editor(es) hagan las modificaciones que consideren convenientes.

Sistema manejador de Paquetes de Puesta en Operación: El objetivo del Sistema Manejador de Paquetes de Puesta en Operación y objetivo central de ésta tesis, es desarrollar un software que ayude a la difusión y construcción de Paquetes de Puesta en Operación en forma colaborativa.

SMPPP: Sistema Manejador de Paquetes de Puesta en Operación.

Paquete, Paquete de Puesta en Operación: Hace referencia a unos de los 9 paquetes que componen el perfil básico.

PPO: Paquete de Puesta en Operación.

Visitante: Cualquier persona que acceda a la aplicación, sin estar registrado o en caso de estar registrado no ha iniciado sesión.

VSE - Very Small Entities: Empresa, organización, departamento o proyecto que tiene entre 1 y 25 empleados.

Referencias Bibliográficas

1. *Software engineering — Lifecycle profiles for Very Small Entities (VSEs) — Part 5-1-2: Management and engineering guide: Generic profile group: Basic profile*. 2011.
2. **Y. Laporte, Claude**. [Online] <http://profs.etsmtl.ca/claporte/English/VSE/index.html>.
3. **Bosch, J.** *Design & Use of software Architectures*. s.l. : Addison-Wesley, 2000.
4. **Pressman, R.** *Ingeniería de software. Un enfoque práctico*. s.l. : Mc Graw Hill, 2002.
5. **Kazman, R., Clements, P. and Klein, M.** *Evaluating software architectures. Methods and case studies*. s.l. : Addison Wesley.
6. **Lozada Rodríguez, Aurora**. *Sistema Multi-agente de apoyo para el procesamiento de administración de proyectos específicos de Moprosoft*. 2007.
7. Diario Oficial de la Federación. [Online] 08 15, 2005.
http://dof.gob.mx/nota_detalle.php?codigo=2088919&fecha=15/08/2005.
8. **Oktaba, Hanna, et al.** *Modelo de Procesos para la Industria de Software, MoProSoft, Versión 1.3*. 2005.
9. European Commission. [Online] 10 2010. [Cited: 08 12, 2012.]
http://epp.eurostat.ec.europa.eu/portal/page/portal/european_business/special_sbs_topics/small_medium_sized_enterprises_SMEs.
10. **Claude, Laporte, Alexandre, Simon and Renault, Alain**. *The Application of International Software Engineering Standards in Very Small Enterprises*. 2008.
11. **Camacho, Erika, Cardeso, Fabio and Nuñez, Gabriel**. *ARQUITECTURAS DE SOFTWARE, GUÍA DE ESTUDIO*. Caracas, Venezuela : s.n., 2004.
12. **Gamm, Erich, et al.** *Design Patterns. Elements of Reusable Object-Oriented Software*. s.l. : Addison-Wesley, 2009.
13. *AGILE PROGRAMMING AND DESIGN PATTERNS IN WEB DEVELOPMENT -A CASE STUDY*.
Dattatreya, V., Chalapati Rao1, K.V. and Rayudu2, V.M. s.l. : International Journal of Software Engineering & Applications (IJSEA), 2012.
14. **Blanco, Jaime**. *Notas de almacenamiento y recuperación de información*. Caracas : s.n., 207.

15. *The Development and Diffusion of International Standard ISO/IEC 29110 for Very Small Enterprises Involved in Software Development.* **Laporte, Claude.** 2011.
16. **Flores, Javier and Vazquez, Ana.** *Paquete de Puesta en Operación de Construcción y Pruebas Unitarias Perfil Básico.* 2011.
17. **Vazquez, Ana.** *Package, Deployment Construction and Unit Testing Basic Profile.* 2009.
18. **Booch, Grady, Rumbaugh, James and Jacobson, Ivar.** *El lenguaje unificado de modelado.* s.l. : Pearson, 2006.
19. **Project Management Institute, Inc.** *Guía de los Fundamentos de la Dirección de Proyectos (Guía del PMBOK) - Cuarta edición.* 2008.
20. **www.iso.org.** [Online]
http://www.iso.org/iso/iso_catalogue/management_and_leadership_standards/management_system_basics/understand_the_basics.htm.
21. **Praxiom Research Group.** praxiom. [Online] 10 29, 2001. <http://praxiom.com/iso-definition.htm#Management%20system>.
22. **www.answers.com.** [Online] <http://www.answers.com/topic/management-system>.
23. **Silberschatz, Abraham, Korth, Henry and Sudarshan, S.** *Fundamentos de bases de datos.* s.l. : Mc Graw Hill, 2006.
24. **Sánchez Rojo, Ignacio Javier.** *Plataforma educativa Moodle, Administración y Gestión.* s.l. : Alfaomega Ra-Ma, 2010.
25. **Tomlinson, Todd.** *Beginning Drupal 7.* s.l. : Apress, 2010.
26. **ORACLE.** The Java EE 5 Tutorial. [Online] 09 2010. [Cited: 01 11, 2013.] <http://docs.oracle.com/javase/5/tutorial/doc/bnaph.html>.
27. **Mastertheboss.com.** mastertheboss. [Online] [Cited: 10 07, 2012.] <http://www.mastertheboss.com/richfaces/primefaces-vs-richfaces-vs-icefaces>.
28. **Pérez, Manuel Sotillo.** *Bases de datos XML nativas.* España : s.n.
29. **Espino Barrios, Luis Fernando.** *Desarrollo de una Base de Datos Nativa XML.* Costa Rica : s.n., 2009.