

Universidad de las Ciencias Informáticas
Facultad 1



Título:

Portal web para la Dirección de Informatización

Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas

Autora:

Lizandra Pérez Mejías

Tutores:

Ing. Yanet Velazquez Jimenez

Ing. Cesar González Hernández

La Habana, Junio de 2015.

“El futuro tiene muchos nombres. Para los débiles es lo inalcanzable. Para los temerosos, lo desconocido. Para los valientes es la oportunidad.”

Víctor Hugo.

DECLARACIÓN DE AUTORÍA

DECLARACIÓN DE AUTORÍA

Declaro ser autor de la presente tesis y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

Autora:

Lizandra Pérez Mejías.

Tutores:

Ing. Yanet Velazquez Jimenez.

Ing. Cesar González Hernández.

AGRADECIMIENTOS

A mi mami, por haber estado apoyándome y queriéndome todo el tiempo desde allá desde mi bello Holguín, gracias por todo tu amor y por creer siempre en tu flaca.

A mi abuelita, porque sin ella mis cinco años de universidad hubiesen sido imposibles, te agradezco tu amor infinito y que cada día me tengas presente.

A mi novio, por todo el apoyo, la alegría y el amor que compartiste conmigo; porque me has ayudado a superar momentos difíciles en los que creía que no podría seguir, por ser siempre, mi nuty.

A mi tía querida Daraysis, por tener siempre presente a su sobri Unice y por sus correitos.

A mi papá y mi abuelo Rolando, gracias por siempre estar presente.

A mi hermano Leo y a mis primos Roly y Joseito, gracias por quererme.

A mi familia en general, por siempre estar pendientes de mí.

A todos mis amigos, por hacer de mi estancia en esta escuela una experiencia increíble.

A todos los que de alguna manera u otra contribuyeron con mi formación y a hacer posible este momento.

DEDICATORIA

A las dos estrellitas de mi vida, que querían que fuera periodista o médico, pero que me apoyaron cuando decidí ser informática; para Uds. abuela y mami, yo les prometí que este momento llegaría.

A José Daniel, la luz de mi ser.

A mi tía querida Dara.

RESUMEN

La Dirección de Informatización (DIN) es la encargada de conducir el proceso de informatización en la Universidad de las Ciencias Informáticas (UCI). Actualmente la gestión y difusión de información sobre sus procesos es deficiente, haciendo más compleja la gestión de necesidades de informatización de las áreas, de las solicitudes a los problemas planteados y de las incidencias relacionadas con las aplicaciones que brindan servicios a la comunidad universitaria. El presente trabajo de diploma tiene como objetivo desarrollar un portal web para la Dirección de Informatización que permita contribuir a gestionar la información en el proceso de informatización de la Universidad de las Ciencias Informáticas.

El desarrollo de la aplicación estuvo guiado por las etapas que propone la metodología de desarrollo SXP. Se emplea el *framework* PHP Symfony 2.3 para el desarrollo de la solución, PostgreSQL 9.2, Visual Paradigm 8.0, NetBeans 8.0 y los lenguajes: UML 2.0, JavaScript 1.8, CSS 3, HTML 5 y PHP 5.4. Para la evaluación de la propuesta de solución se utiliza Apache JMeter 2.3.1 y Nikto 2.1.4.

Palabras clave: *framework*, gestión, informatización, portal web, symfony.

ÍNDICE

INTRODUCCIÓN.....	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.....	6
1.1 Introducción.....	6
1.2 Fundamentos teóricos asociados a la investigación	6
1.3 Tipos de portales web	10
1.4 Estudio de portales web corporativos	12
1.5 Arquitectura de la información	17
1.6 Usabilidad.....	18
1.7 Web 2.0.....	18
1.8 Tecnologías, lenguajes y herramientas	19
1.8.1 Metodología de desarrollo de software	19
1.8.2 Lenguajes utilizados para el desarrollo de la aplicación.....	22
1.8.3 Tecnologías.....	23
1.8.4 Herramientas a utilizar	27
1.9 Conclusiones parciales	30
CAPÍTULO 2: PROPUESTA DE SOLUCIÓN.	31
2.1 Introducción del capítulo	31
2.2 Proceso de organización de la información del portal web.....	31
2.3 Descripción, estructuración y análisis de la propuesta de solución.....	32
2.4 Modelo del dominio	34
2.5 Especificación de requisitos.....	36
Requisitos funcionales:.....	37
Requisitos no funcionales:.....	39

TABLA DE CONTENIDO

2.6	Descripción de estilos arquitectónicos y patrones de diseño	41
2.6.1	Estilo arquitectónico en Symfony 2.3	41
2.6.2	Patrones de diseño en Symfony 2.3.....	44
2.7	Modelo de despliegue de la aplicación.....	50
2.8	Conclusiones parciales.....	52
CAPÍTULO 3: IMPLEMENTACIÓN Y EVALUACIÓN DE LA SOLUCIÓN PROPUESTA.		53
3.1	Introducción.....	53
3.2	Diagrama de componentes.....	53
3.3	Estándares de codificación.....	54
3.4	Pruebas	56
3.4.1	Pruebas funcionales	56
3.4.2	Pruebas de carga.....	57
3.4.3	Pruebas de estrés.....	57
3.4.4	Pruebas de seguridad.....	57
3.4.5	Resultados de las pruebas	57
3.5	Conclusiones parciales.....	60
CONCLUSIONES GENERALES		62
RECOMENDACIONES.....		63
BIBLIOGRAFÍA		64

ÍNDICE DE FIGURAS

Figura 1. Fases de la metodología SXP (Rodríguez Villar, 2007).	22
Figura 2. Modelo de dominio.	35
Figura 3. Patrón MVC en Symfony2 (SensioLabs, 2007).	42
Figura 4. Arquitectura del portal web para la Dirección de Informatización.	43

TABLA DE CONTENIDO

Figura 5. Uso del patrón Experto en la clase <i>HitoController.php</i>	45
Figura 6. Uso del patrón Creador en la clase <i>IncidenciaController.php</i>	46
Figura 7. Uso del patrón Controlador en la clase <i>DefaultController.php</i>	48
Figura 8. Empleo del patrón Decorador en la plantilla <i>layout.html.twig</i>	49
Figura 9. Patrón Inyección de dependencias.	50
Figura 10. Modelo de despliegue.	51
Figura 11. Diagrama de componentes.	53
Figura 12. Resultado de las pruebas funcionales.	58
Figura 13. Resultados de las pruebas de carga y estrés.	59
Figura 14. Vulnerabilidades detectadas.	60
Figura 15. Resultados de las pruebas de seguridad.	60

ÍNDICE DE TABLAS

Tabla 1. Complejidad de los requisitos funcionales.	39
--	----

INTRODUCCIÓN

Actualmente la sociedad se encuentra inmersa en la era de la información, las comunicaciones y la tecnología; la unión de estos tres elementos posee un papel significativo en la vida cotidiana de muchas instituciones que prestan especial atención a la informatización de sus procesos con el propósito de aportarles mayor calidad, optimización y rapidez. La informatización es un proceso que involucra a clientes y desarrolladores o proveedores de soluciones que se encuentran en constante comunicación y en la que fluye una gran cantidad de información que necesita ser gestionada.

“La gestión de la información no es más que el proceso de organizar, evaluar, presentar, comparar los datos en un determinado contexto, controlando su calidad, de manera que esta sea veraz, oportuna, significativa, exacta y útil, así como su disponibilidad en el momento que se le necesite. La gestión de la información se encamina al manejo de la información, documentos, metodologías, informes, publicaciones, soportes y flujos en función de los objetivos estratégicos de una organización” (Marqués Graells, 2002).

La gestión ineficiente de la información que se maneja y la dificultad al identificar cuáles áreas informatizar constituyen aspectos que pueden afectar el buen funcionamiento de muchas instituciones cubanas según (Rodríguez Salas, 2002). La Universidad de las Ciencias Informáticas no se encuentra ausente a estos problemas. El constante cambio de la información, así como su gestión y recopilación de forma manual o mediante el correo electrónico, provocan que no exista integración entre la información que proporcionan los sistemas informáticos y en algunos casos esta se encuentre duplicada; por esta razón muchas veces un usuario tiene que insertar datos idénticos en varios sistemas que lo requieran, pudiendo insertarlos en solo uno y que se actualicen en las aplicaciones que los necesiten por medio de la base de datos.

Como área encargada de conducir el proceso de informatización en la UCI, la DIN realiza la gestión de la información relacionada con los procesos que tributan a la informatización de la Universidad. Este proceso ha ido mejorando haciendo uso del Plan de Informatización Anual como documento rector. En él se definen las aplicaciones que se desarrollarán en el año en curso, sus cronogramas de desarrollo y de despliegue; por lo que el proceso resulta más organizado y se puede llevar a cabo un mejor seguimiento y control. A pesar de esto, aún se encuentran deficiencias en la comunicación que debe existir entre la DIN con clientes y desarrolladores, pues muchas personas no poseen conocimiento de la existencia de este Plan de Informatización y de los proyectos que se encuentran en ejecución, así como de los procesos y servicios

que brinda la dirección. Además, la poca información de este proceso, provoca que los sistemas que se desarrollan para la informatización de la Universidad no cumplan con los lineamientos tecnológicos establecidos para el desarrollo y el mantenimiento de los sistemas.

Con el objetivo de lograr una mayor organización y control, la DIN ejecuta varios procesos como: la gestión de necesidades de informatización de las áreas, solicitudes de proyectos así como de incidencias relacionadas con los sistemas que brindan servicios a la comunidad universitaria, además de la conformación del banco de problemas. A continuación se describen estos procesos y algunas de sus deficiencias.

“Banco de Problemas de Informatización: registro de solicitudes de informatización realizado por personal de la UCI, necesidades de las unidades organizativas, problemas presentados o metas propuestas a partir del avance del proceso de informatización con vistas a darle solución inmediata o según planificación en función de la capacidad y oportunidad de las entidades ejecutoras” (Soto López, 2013).

El banco de problemas constituye uno de los principales mecanismos que utiliza la dirección en función de lograr una mayor organización de las necesidades de informatización de cada área; sin embargo, al no encontrarse publicado la mayor parte de la comunidad universitaria desconoce de su existencia, provocando que en él se encuentren problemas que podrían haber sido solucionados por los desarrolladores de la Universidad y que todavía se encuentran sin solución.

La gestión de necesidades de informatización de las áreas actualmente se realiza de forma manual, a partir de la visita de los especialistas de la dirección a las áreas, o haciendo uso de los medios de comunicación disponibles como: la telefonía y el correo electrónico, causas que propician una gestión menos eficiente de las necesidades, mayor demora al emitir una respuesta a las áreas y que se genere un gran cúmulo de correos electrónicos por parte de los responsables de registrar las necesidades de las áreas solicitando saber si estas fueron aceptadas o no.

Por otra parte, las incidencias relacionadas con los sistemas que brindan servicios a la comunidad, son recogidas en los departamentos de la dirección y recibidas por los especialistas designados, por lo que el personal que necesite reportar alguna debe dirigirse a los locales de la DIN. En caso de que la persona encargada de recepcionar las incidencias no se encuentre, el usuario deberá regresar en otro momento para registrarla. Este modo de trabajo provoca que la información en algunos casos esté propensa a la

duplicación y pérdida, además de no brindar al usuario reporte o notificación del estado en el que se encuentra el proceso de solución de su incidencia.

El éxito de los procesos anteriormente planteados depende en gran medida de la debida comunicación con las áreas de la Universidad que requieren de los servicios de informatización, así como con los proveedores que aportan al desarrollo de soluciones, tales como: centros de desarrollo, comunidades, grupos investigativos, grupos científicos estudiantiles y personal de las áreas.

Teniendo en cuenta la problemática antes planteada se determina como **problema a resolver**: ¿Cómo gestionar la información relacionada con el proceso de informatización de la Universidad de las Ciencias Informáticas?

Por tanto, el **objeto de estudio** está centrado en la gestión de la información relacionada con el proceso de informatización, enmarcado en el **campo de acción** la gestión de la información relacionada con el proceso de informatización en la Universidad de las Ciencias Informáticas.

Objetivo general:

Desarrollar un portal web haciendo uso de tecnologías libres que permita gestionar la información relacionada con el proceso de informatización de la Universidad de las Ciencias Informáticas.

Objetivos específicos:

- Caracterizar el marco teórico de la investigación y el estado del arte sobre el desarrollo de portales web corporativos.
- Realizar un prototipo de la organización de la información del portal web a partir del levantamiento de requisitos.
- Implementar el portal web corporativo para la informatización de la Universidad de las Ciencias Informáticas.
- Evaluar el correcto funcionamiento de la solución propuesta y el cumplimiento de la misma con el objetivo general de la investigación.

Para cumplir con el objetivo general se plantean las siguientes **tareas de la investigación**:

1. Revisión de bibliografía asociada a las nuevas tendencias en el desarrollo de los portales web.

2. Definición de las tecnologías a utilizar para el desarrollo del portal web.
3. Definición de los requisitos funcionales y no funcionales del portal web.
4. Organización de los contenidos a publicar en el portal web.
5. Elaboración de los artefactos requeridos por la metodología seleccionada.
6. Implementación de las funcionalidades definidas.
7. Documentación de las pruebas realizadas para evaluar el portal web.
8. Aplicación de las pruebas de software al portal web implementado.

La **idea a defender** en la investigación se define como: El desarrollo de un portal web contribuirá a gestionar la información relacionada con el proceso de informatización de la Universidad de las Ciencias Informáticas.

Métodos Teóricos

1. **Analítico-Sintético:** Es empleado para analizar y determinar los factores y rasgos distintivos que hacen a un portal web corporativo la mejor opción de solución al problema. Además permite encontrar las relaciones que se establecen entre estos rasgos y factores ya no desde la individualidad de cada uno, si no desde la unidad; para de esta manera poder establecer la mejor solución para el portal web para la Dirección de Informatización.
2. **Inductivo-Deductivo:** Este método es empleado para realizar un análisis de una muestra pequeña de los tipos de portales web, así como de algunos que posean objetivos y características comunes con la aplicación a desarrollar, para establecer patrones y semejanzas entre ellos, de esta manera se determinarán requerimientos comunes a este tipo de portales web y se podrá hacer un estudio de homólogos que tribute a una mejor solución.
3. **Modelado:** “La modelación es el método mediante el cual se crean abstracciones con el objetivo de explicar la realidad. El modelo como sustituto del objeto de investigación es semejante a él, existiendo una correspondencia objetiva entre el modelo y el objeto, siendo el investigador quien elabora dicho modelo. El modelo es el eslabón entre el sujeto y el objeto intermedio” (Hernández León, y otros, 2011). Permite modelar el proceso a seguir para desarrollar una nueva solución del portal web para la Dirección de Informatización, creando una representación abstracta con el objetivo de explicar la realidad.

Métodos Empíricos:

1. **Entrevista:** “es una técnica que se utiliza para recabar información en forma verbal. Sirve para obtener información acerca de las necesidades de los usuarios y la manera de satisfacerlas” (academia.edu, 2015) (UNAD, 2015).

“La entrevista es una conversación planificada entre el investigador y el entrevistado para obtener información. Su uso constituye un medio para el conocimiento cualitativo de los fenómenos o sobre características personales del entrevistado y puede influir en determinados aspectos de la conducta humana por lo que es importante una buena comunicación” (Hernández León, y otros, 2011).

La estructuración del contenido queda dividida en 3 capítulos:

- **Capítulo 1. Fundamentación Teórica:** En este capítulo se muestran un conjunto de conceptos útiles para lograr un mayor entendimiento de la investigación. Se realiza un análisis de las principales tendencias actuales de portales web con el objetivo de determinar el enfoque adecuado para la solución y se realiza un estudio de portales homólogos para determinar la estructura de la información adecuada para el portal web. Se muestran herramientas y tecnologías de desarrollo a utilizar, así como características del *framework* PHP Symfony 2.3 y se define cuál metodología usar en la solución.
- **Capítulo 2. Propuesta de solución:** En este capítulo se realiza una descripción de las funcionalidades del sistema, se definen los requisitos funcionales y no funcionales que debe cumplir y se describen a grandes rasgos las características de la solución propuesta mediante los diversos artefactos que especifica la metodología de desarrollo de software utilizada.
- **Capítulo 3. Implementación y evaluación de la solución propuesta:** En este capítulo se detallan los estándares de codificación usados y las pruebas realizadas a la propuesta de solución del problema planteado. Además se muestran los resultados de cada una de las pruebas.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1.1 Introducción

En el presente capítulo se definen un conjunto de conceptos y elementos teóricos que propician un mejor entendimiento de la investigación. Se caracterizan elementos fundamentales sobre los tipos de portales web para determinar el enfoque más adecuado para la solución y a través de la realización de un estudio de sistemas homólogos se analizan las características de algunos portales nacionales e internacionales con el objetivo de identificar elementos que puedan servir de apoyo en el desarrollo del portal web y la forma de organización de la información a publicar. Además se analizan las herramientas, tecnologías, lenguajes y metodología de desarrollo de software que serán utilizadas.

1.2 Fundamentos teóricos asociados a la investigación

Con el objetivo de lograr una mayor comprensión de los contenidos tratados se tuvieron en cuenta algunas definiciones importantes para el desarrollo de la investigación.

Gestión de la información

A continuación se relacionan varios conceptos de gestión de la información:

“La gestión de la información se define como el conjunto de actividades realizadas con el fin de controlar, almacenar y, posteriormente, recuperar adecuadamente la información reproducida, recibida o retenida por cualquier organización en el desarrollo de sus actividades” (Bustelo, y otros, 2001).

“El proceso mediante el cual se obtienen, despliegan o utilizan recursos básicos (económicos, físicos, humanos, materiales) para manejar información dentro y para la sociedad a la que sirve. Tiene como elemento básico la gestión del ciclo de vida de este recurso y ocurre en cualquier organización” (Ponjuán, 2004).

Teniendo en cuenta las nuevas tecnologías se define gestión de la información como un “proceso que incluye operaciones como extracción, manipulación, tratamiento, depuración, conservación, acceso y/o colaboración de la información adquirida por una organización a través de diferentes fuentes y que gestiona el acceso y los derechos de los usuarios sobre la misma” (Curto, 2006).

Luego de analizadas estas definiciones la autora de la investigación asume que la gestión de información consiste en la recolección, almacenamiento y manipulación de datos procesados que incluye operaciones como extracción, organización, tratamiento, depuración y conservación.

Sistemas de información

“Sistema integrado y automatizado para proveer la información que sostenga las funciones de operatividad, gestión y toma de decisiones en una organización” (Davis, y otros, 1985).

“Un sistema de información se puede definir técnicamente como un conjunto de componentes relacionados que recolectan (o recuperan), procesan, almacenan y distribuyen información para apoyar la toma de decisiones y el control en una organización” (Instituto Tecnológico de Sonora, 2015).

Según (Peralta, 2008), sistema de información es: un “conjunto de elementos que interactúan entre sí con el fin de apoyar las actividades de una empresa o negocio”.

Herramientas para la gestión de información en la UCI

El proceso de informatización en la Universidad de las Ciencias Informáticas se descompone en varios procesos como la gestión de necesidades de informatización de las áreas, de incidencias relacionadas con los sistemas que brindan servicios a la comunidad universitaria, la gestión de proyectos y la publicación de información. Actualmente se pueden encontrar herramientas que se utilizan para la gestión de la información tales como:

Gespro: permite la gestión de alcance y tiempo así como la construcción semiautomática de cronogramas. Facilita además, la identificación de la línea base de los proyectos y la asignación de recursos materiales y humanos. Garantiza una vista completa al estado del proyecto respecto a la calidad, costos, tiempo, recursos humanos y contratos, a través del cálculo automático de indicadores y visualización de reportes de estado. Basa su funcionamiento en la implementación de buenas prácticas sugeridas por los estándares de mejora de procesos PMBOK y CMMI (Piñero Pérez, y otros, 2013). Este sistema se utiliza en la gestión de proyectos de desarrollo en la producción, para realizar un seguimiento de sus hitos y el cronograma que deben seguir.

ServiceDesk Plus: es una plataforma Service Desk modular y escalable, permite implantar procesos de gestión de soporte técnico, integrado con inventario automático y la gestión de recursos TI. Es compatible con procesos ITIL (la gestión de incidencias, problemas, cambios, etc.) además de inventario automático,

CMDB, catálogo de servicios y otras prestaciones estratégicas (ireo.com, 2015). Este sistema se utiliza en el Centro de Soporte de la Universidad para gestionar las incidencias de software de las empresas. Esta herramienta es privativa.

GRHS: el Gestor de Recursos de Hardware y Software es un Sistema de Gestión de Inventarios de Red (SGIR) que implementa la arquitectura cliente-servidor. Realiza inventarios de hardware y software, monitorización de instalación de programas y conexión de dispositivos USB. GRHS cuenta con un conjunto de colectores capaces de detectar cambios en la configuración de los equipos, así como de ejecutar acciones y alarmas una vez identificados estos cambios (Hernández Pérez, y otros, 2013). Dentro de los módulos de esta herramienta se encuentra el Módulo de alarmas y acciones ante incidencias (MAAI) dedicado a la gestión de incidencias, realización de acciones y la emisión de alarmas ante anomalías en la UCI.

OTRS: *Open-source Ticket Request System* u OTRS (en español "sistema de solicitud de tickets de código abierto") es un sistema libre que cualquier institución puede utilizar para asignar identificadores únicos llamados tickets a solicitudes de servicio o de información, de forma que facilite el seguimiento y manejo de dichas solicitudes así como cualquier otra interacción con sus clientes o usuarios. Se distribuye bajo la licencia GNU *Affero General Public License* (otrs.com, 2015). Mediante esta herramienta se realiza la gestión de incidencias de seguridad informática en la Universidad de las Ciencias Informáticas.

Los sistemas mencionados anteriormente se utilizan en la Universidad para contribuir a la gestión de la información de determinados procesos. Ninguno de ellos realiza la gestión de información referente al proceso de informatización completo, cada una de ellas se dedica a determinadas áreas como la gestión simple de incidencias enfocada a las aplicaciones que prestan servicios a la comunidad universitaria.

El uso de aplicaciones web para gestionar la información es una solución viable en la Universidad. Actualmente se necesita una herramienta que abarque la gestión de todo el proceso de informatización y no solo de algunas de sus áreas.

Aplicaciones web

Una aplicación web es aquella aplicación que los usuarios pueden utilizar accediendo a un servidor web a través de Internet o una Intranet. Es una aplicación de software que se codifica en un lenguaje soportado por los navegadores web. Poseen compatibilidad multiplataforma (Mora, 2002).

“Una aplicación web es básicamente una manera de facilitar el logro de una tarea específica (...) es una herramienta para la comunicación. La aplicación web permite al usuario interactuar directamente con los datos en forma personalizada para llevar a cabo esa tarea específica” (Menéndez-Barzanallana, 2012).

“En la ingeniería de software se denomina aplicación web a aquellas aplicaciones que los usuarios pueden utilizar accediendo a un servidor web a través de Internet o de una Intranet mediante un navegador. En otras palabras, es una aplicación de software que se codifica en un lenguaje soportado por los navegadores web” (Peñafiel, 2013).

Luego del análisis de estos conceptos la autora puede concluir que las aplicaciones web permiten una comunicación más fluida y dinámica entre el cliente y el servidor en tiempo real. Además de facilitar su mantenimiento que se realiza vía Internet, consumen pocos recursos dado que la aplicación se encuentra instalada en otro ordenador (servidor). Estas aplicaciones ofrecen una gran disponibilidad de la información.

Portal web

“Portal es un término utilizado para referirse a un sitio web el cual pretende servir como sitio de partida para las personas que se conectan a la *World Wide Web*. Los usuarios lo utilizan como sitios anclas ya que tienen gran reconocimiento en Internet por el poder de influencia que tienen sobre comunidades” (W3C, 2015).

Son sitios que los usuarios tienden a visitar como páginas de inicio. La idea es emplear estos portales para localizar la información deseada. Un portal es más bien una plataforma de despegue para la navegación en la Web (Monzón, y otros, 2010).

“De modo simple podríamos definir portal como un punto de entrada a Internet donde se organizan sus contenidos, ayudando al usuario y concentrando servicios y productos, de forma que le permitan a éste hacer cuanto necesite, o al menos que pueda encontrar allí todo cuanto utiliza a diario sin tener que salir de dicho *website*. El objetivo pretende ser fidelizar a los usuarios y que se establezca algún tipo de vínculo casi personal entre el usuario y el portal” (García Gómez, 2012).

Luego de analizados los conceptos anteriores la autora de este trabajo considera que los portales web están dirigidos a resolver las necesidades específicas de un grupo de personas o de acceso a la información y servicios de una institución. Un portal web se puede ver como una plataforma de despegue para la navegación en la Web donde se reúne toda la información que los usuarios necesitan en un único lugar para acceder a ella de forma lógica.

1.3 Tipos de portales web

“Los portales web pueden ser clasificados en dependencia del público a los que son dirigidos o la línea de contenido que desean cubrir, ya sea conocimientos científicos o información de manera general. En este caso se enfocará la clasificación hacia aquellos que se encuentran orientados a toda la población, denominados Generales (megaportales o portales horizontales); los que son dedicados a usuarios interesados en un tema específico, Especializados, que a su vez se divide en los destinados a personas relacionadas con una empresa o institución Corporativos y los Verticales que proveen información y servicios a un sector o industria en particular” (García Gómez, 2012).

Portales generales (Megaportales)

“Un portal de carácter general está orientado a todo tipo de público, ofrece contenidos de carácter muy amplio, cuya pretensión es cubrir las temáticas más demandadas. También ofrece habitualmente servicios de valor añadido como comunidades virtuales, espacio web gratuito, información de diverso tipo, personalización de la información, chat, e-mail gratuito, mensajes a teléfonos móviles, software gratuito, grupos de discusión, comercio electrónico, buscador, etc.” (García Gómez, 2012).

Desventajas:

“Sin embargo, este modelo empieza a quedarse obsoleto, pues a ese nivel tan amplio de cobertura solo pueden sobrevivir unos pocos y empieza a haber demasiados portales generalistas (megaportales o portales horizontales), orientados fundamentalmente a usuarios principiantes y basados en servicios poco especializados, inadecuados para los usuarios más expertos” (García Gómez, 2012). Ejemplo de este tipo de portales web son: (Terra, 2015) y (AOL.com, 2015).

Portales especializados

“Este modelo de portal está orientado a aspectos más específicos, que cubran por ejemplo un área geográfica determinada, como puede ser la Región de Murcia (tubitacora.com, 2015), un tema concreto como la educación familiar (Sandbox Networks, 2015), o bien destinados a cubrir las necesidades de las personas relacionadas con una corporación concreta (empleados, directivos, clientes, proveedores, etc.)” (García Gómez, 2012).

Ventajas:

“En los portales especializados, son muchos los usuarios que satisfacen convenientemente sus necesidades al abordar temas más específicos y la simplificación de las búsquedas de informaciones específicas e interesantes para el usuario; su potencialidad es basada en el grado de profundización que pueden lograr en sus contenidos y servicios sobre determinadas temáticas” (García Gómez, 2012).

- **Portales verticales (Vortales)**

“Un portal vertical, también conocido como Vortal (*Vertical Portal*) es un sitio web que provee de información y servicios a una industria en particular. Es el equivalente industrial específico de los portales generales de la Web, pero en este caso, además de ofrecer los típicos servicios de valor añadido característicos de los portales generales, la cobertura en cuando a contenidos se centra en un tema o sector concreto” (García Gómez, 2012).

Ventajas:

“Los portales verticales han de contener más datos sobre su mercado que cualquier otro sitio, lo que en sí mismo ya es un valor añadido considerable, siempre manteniendo los datos de las instituciones que lo integran a buen recaudo tras el cortafuegos” (García Gómez, 2012). Algunos portales verticales son: (toronto.com, 2015) e (infonomia.com, 2015).

- **Portales corporativos**

“Los portales corporativos tienden a ser una prolongación natural de las intranet corporativas, en las que se ha cuidado la organización de la información y la navegación, donde se permite, y sobre todo se potencia, el acceso a información de la propia institución, la edición de material de trabajo propio, el contacto con clientes y proveedores, etc.” (García Gómez, 2012).

“Los portales corporativos proporcionan un punto de acceso único para toda la información. Los mismos cuentan con mecanismos de autenticación únicos para el usuario, así como para la realización de búsquedas, proporcionando herramientas para la colaboración de equipos a distancia e incorporando procesos colaborativos que permiten identificar prioridades al usuario. Integran la información externa de proveedores, entorno, clientes, etc., mediante mecanismos de contenido indexado” (Días, 2001). Entre los más representativos se encuentran (Microsoft, 2015) y (Oracle, 2015).

Ventajas (García Gómez, 2012):

- Optimizan la gestión de la organización.
- Incorporan funcionalidades complementarias importantes, como la integración con otros sistemas relacionados con la organización.
- Disponen de la información precisa que sirve de base para el trabajo de la organización.
- Centralizan toda la información generada por la entidad para una mejor gestión de los procesos de la empresa.

Para cumplir con los objetivos propuestos, luego de analizar las características de los portales anteriormente mencionados, se decidió darle a la solución el enfoque de portal corporativo, debido a que se requiere que el portal brinde servicios para usuarios más especializados como los usuarios de la Dirección de Informatización, que potencie la interacción entre los clientes y proveedores de las distintas áreas de la Universidad además de permitir el acceso a los contenidos e información de la DIN.

1.4 Estudio de portales web corporativos

Luego de determinar la tendencia de portal web a utilizar para la solución, en este caso el enfoque de portal corporativo, se realizó un estudio de algunos de los portales web a nivel internacional que siguen este enfoque o tendencia, así como de algunos portales nacionales y de la propia Universidad con el objetivo de identificar los servicios que prestan y la organización de la información que brindan. Además se realizó un estudio del trabajo de diploma Portal de la Dirección de Informatización de la UCI del año 2007 con el objetivo de extraer elementos comunes entre el trabajo de diploma mencionado y el actual trabajo.

Internacionales

Portal de la APETI (Asociación profesional de expertos técnicos inmobiliarios) (APETI, 2015)

El portal de la APETI posee en la página principal una cabecera publicitaria en la que se especifican los servicios que se prestan a través de un carrusel de imágenes. Posee en la parte superior un menú principal dividido en 10 secciones (Inicio, La asociación, Asóciate, Formación, Boletines, Noticias, Entrevistas, Artículos de opinión, Intranet, Contáctenos). El área de contenido se encuentra dividida en varias secciones donde se muestra un resumen de los temas más relevantes para la asociación; para ampliar la información referente a cada tema se cuenta con un botón de **“leer más”**. El portal tiene además enlaces a sus

respectivas páginas en las redes sociales e información referente a la localización de la institución así como otros servicios generales como Aviso Legal y Contacto.

Mancomunidad Sierra Occidental Huelva (Mancomunidad Sierra Occidental Huelva, 2015)

En la cabecera del portal se puede apreciar el logotipo de la organización y su nombre, junto a una imagen representativa de la empresa. Posee un menú principal con enlaces (Inicio, Presentación, Turismo, RSU, Tablón de Anuncios, Formación, Nuestros Pueblos, Empresas, Directorios). La sección de contenido se encuentra dividida en dos áreas: la izquierda muestra una imagen animada con publicidad de la organización acompañada de una sección donde se ofrece documentación de varios tipos y un apartado de noticias; en la sección derecha se encuentra un servicio de suscripción de alertas y un listado de novedades seguido por un área de contacto. En el pie de página se ofrece información referente a la empresa.

Komelsat Inversiones & Gestión de Activos (Komelsat, 2015)

El portal posee una cabecera superior con el nombre de la organización y el logotipo además posee un selector para cambiar el idioma por defecto (español, inglés, francés). Se puede observar un menú desplegable donde se muestran distintos enlaces (Inicio, Sobre nosotros, Servicios, Búsqueda de inmuebles, Colaboradores, Contacto). La parte de contenido se divide en tres secciones. La primera orienta al usuario sobre las acciones a realizar en el sitio referente a los servicios que presta. Seguidamente se muestran enlaces publicitarios y luego muestra opciones extra (localizar, alarma, calculadora inmobiliaria, ofertas). El pie de página muestra enlaces hacia el mapa del sitio, preguntas frecuentes, las políticas de privacidad, quiénes somos y buscador inmobiliario.

Tecnología de la Información y Comunicación (TIC) Pontificia Universidad Católica de Ecuador (TIC, 2015)

Posee una cabecera de página con el nombre de la institución y su logotipo correspondiente. En el menú principal posee diferentes enlaces (Inicio, Moodle, Blackboard Collaborate, Sitio Institucional), además de un buscador para el sitio, y enlaces a redes sociales. Posee una sección de publicidad mostrándola a través de un carrusel de imágenes. El área de contenido se encuentra estructurada en tres columnas, la primera muestra varios bloques de contenido referentes a plataforma virtual con enlaces hacia los documentos y normativas de la institución. La segunda columna muestra una sección donde se puede visualizar el

propósito general del sitio. En la tercera columna se brindan enlaces a la plataforma y recursos de aprendizaje disponibles. Muestra también una sección con un canal RSS para noticias de interés.

Nacionales

Cubagob (Cubagob, 2015)

Es el Sitio del Gobierno de la República de Cuba en el que se puede encontrar información sobre los símbolos nacionales, las provincias y municipios, así como enlaces a otros sitios oficiales. Ese portal tiene una cabecera en la que se muestra el nombre de la institución e imágenes representativas de los símbolos patrios de Cuba. Tiene un menú principal con las opciones (Gobierno, Noticias Oficiales, Política Exterior, Desarrollo Económico, Desarrollo Social, Otras Informaciones y Sitios Vinculados). Posee un mapa del sitio que se encuentra en el área del contenido de la página principal del sitio, en esta área también se encuentra una pequeña promoción de la información y los servicios que presta el portal web.

Se realizó además una búsqueda de los portales que corresponden a los centros productivos en la Universidad de las Ciencias Informáticas, llegando a la conclusión de que algunos no poseen un portal y si lo tienen la mayoría no incluyen servicios o no los mantienen actualizados, por lo cual es pobre la información adquirida sobre los portales institucionales en la Universidad. Entre ellos se pueden citar:

Portal del Centro de Soporte (Centro de Soporte, 2015)

Este portal posee en el área de la cabecera un carrusel de imágenes mediante el que se promocionan algunos de los servicios que proporciona la institución; luego posee un menú principal en el que se listan las opciones (Catálogo de Servicios, Centro de Descargas, Base de Conocimiento y Comunidades de Soporte). El área de contenido está dividida en secciones entre las que se encuentran noticias y avisos. Posee una sección dedicada a preguntas frecuentes y enlaces de interés, así como los términos de uso asociados a las aplicaciones web en la Universidad.

Portal de la Dirección de Investigaciones (Dirección de Investigaciones, 2015)

En este portal web se puede apreciar una pequeña cabecera interactiva que presenta los servicios que presta la dirección, así como un menú principal con las siguientes opciones (Inicio, Eventos, Descargas, Foro de Discusión, Seminarios Científicos, Buscar, ¿Cómo se hace?). Además posee una sección de noticias y otras secciones importantes y de interés para los usuarios como las descargas, la de información de la institución (¿Quiénes somos?), enlaces a sitios de la Universidad y políticas científicas.

Portal de la Dirección de Formación Postgraduada (Dirección de Formación Postgraduada, 2015)

Este portal posee una cabecera estática en la que se puede apreciar el nombre de la institución, además tiene un menú principal con las siguientes opciones (Inicio, Descargas, Enlaces, Categoría de Noticias, Buscar). En el área del contenido se muestra además una sección que permite realizar la reservación de aulas y consultar las distribuciones realizadas.

Portal de la Dirección de Redes y Servicios Telemáticos (Dirección de Redes y Servicios Telemáticos, 2015)

Este portal web posee una cabecera en la que se puede encontrar un menú superior con algunas opciones como Sobre nosotros, además se puede apreciar un *slider* que informa sobre algunos servicios brindados por la dirección. Posee una sección de noticias y muestra una lista de enlaces de interés.

Portal de la Dirección de Seguridad Informática (Dirección de Seguridad Informática, 2015)

Este portal posee una cabecera en la que se muestra el logo de la dirección y seguidamente se puede encontrar el menú principal con varias opciones. Posee una sección de noticias y luego varias secciones (entre las que se encuentra la de preguntas frecuentes), seguidas de los bloques que muestran los enlaces de interés y los sitios que se recomienda visitar.

Además de analizar portales web nacionales e internacionales con este enfoque de portal corporativo, se tuvo en cuenta para la realización de esta investigación el trabajo de diploma: Portal de la Dirección de Informatización de la UCI realizado en el año 2007.

Trabajo de diploma Portal de la Dirección de Informatización de la UCI (Porta García, y otros, 2007)

El trabajo de diploma se realizó con el objetivo de desarrollar un portal web donde se expusiera o publicara información para la realización de los procesos de la dirección. En él se identificaron características comunes con la presente investigación entre las que se encuentran una sección de noticias relevantes, una sección que presenta información referente a la propia dirección como: misión, visión, estructura y proyectos en ejecución. Se propone la utilización de un portal web corporativo para la solución. Esta propuesta se encuentra desarrollada sobre una versión antigua de Drupal y no incluye la gestión de información como incidencias, necesidades y solicitudes de proyecto de informatización; solo brinda información acerca de este proceso. Debido a que en el presente trabajo de diploma se necesita realizar la gestión de los procesos

anteriormente mencionados, la solución propuesta por (Porta García, y otros, 2007) no se considera como una alternativa para la realización del presente trabajo de diploma.

Conclusiones del estudio

En el estudio realizado se tuvieron en cuenta las características en cuanto a la estructuración de la información y los servicios que ofrecen cada uno de los portales web sobre las instituciones a la que pertenecen. Para el mismo se analizaron 4 portales institucionales o corporativos a nivel internacional y 6 a nivel nacional, enfocando la búsqueda hacia los portales de centros productivos y direcciones de la Universidad. Finalmente se realizó un estudio del trabajo de diploma Portal de la Dirección de Informatización de la UCI. Dentro de las características comunes a la mayoría de los portales web estudiados se pueden mencionar:

- **Cabeceras publicitarias:** de los 10 portales analizados 5 poseen cabeceras con publicidad acerca de los servicios prestados por las instituciones, sin embargo otros, a pesar de no tener cabeceras publicitarias, sí tienen componentes dedicados a este apartado, como son carruseles de imágenes o imágenes animadas.
- **Logotipo y nombre de la organización:** en su totalidad los portales analizados muestran el logotipo (en caso de poseer) y el nombre de la organización generalmente en la posición superior izquierda de la cabecera.
- **Menús desplegables:** 2 de los sitios web estudiados hacen uso de menús desplegables.
- **Información de contacto:** todos los portales de las organizaciones analizadas poseen información de contacto distribuidas en diferentes partes, generalmente en el menú principal, mediante los enlaces Acerca de, Sobre nosotros, Quiénes somos, Contactos, Contáctenos, o en el pie de página.
- **Enlaces a la Intranet:** de los 10 portales analizados (3) poseen enlaces a la Intranet de sus instituciones ubicados principalmente en el menú principal y otras áreas como barras laterales.
- **Principales servicios:** todos los portales muestran información, en algunos casos abundante (como el Portal de la Dirección de Redes y Servicios Telemáticos y el portal de la APETI) y en otros un poco menos (como Cubagob y el portal de la Dirección de Formación Postgraduada), sobre los servicios que prestan de una manera intuitiva y fácil de entender.

- **Enlaces de interés:** entre los enlaces de interés encontrados se encuentran: redes sociales, canales de noticias y novedades o noticias relevantes, aunque no todos los enlaces se encuentran en la generalidad de los portales web vistos.

Teniendo en cuenta los elementos mencionados se puede determinar que la realización de un estudio de portales que siguen la tendencia de portal corporativo y que se asemejan a la propuesta de solución en cuanto a servicios que prestan, influyó positivamente en la definición de los contenidos a mostrar y el lugar específico para mostrarlos.

Con el objetivo de estructurar y organizar la información a mostrar en los portales web se tiene muy presente la arquitectura de la información. A continuación se describe brevemente qué es la AI y se habla de la Usabilidad que es una disciplina muy relacionada con esta.

1.5 Arquitectura de la información

La Arquitectura de Información (AI) según Louis Rosenfeld y Peter Morville “se refiere al diseño, organización, etiquetado, navegación y sistemas de búsqueda que ayudan a los usuarios a encontrar y gestionar la información de manera efectiva” (Morville, y otros, 1998).

Estos autores agregan que la AI:

- Determina la funcionalidad que el sitio va a tener.
- Especifica cómo los usuarios van a encontrar la información al definir su organización, navegación, etiquetado y sistemas de búsqueda.
- Representa cómo el sitio se va a acomodar al cambio y crecimiento en el tiempo.

Para crear una buena AI es importante tener en cuenta tres elementos claves: los círculos de Rosenfeld y Morville que son: Contexto, Contenido y Usuarios (Morville, y otros, 1998).

La propuesta de los contenidos que formarán parte del sitio web debe estar en correspondencia con los objetivos del mismo y sus usuarios principales. Este no es un proceso que se realiza una sola vez. Los usuarios, sus necesidades y el contexto varían. Por lo tanto, conocer las principales necesidades de información o inquietudes de los usuarios respecto al sitio, debe ser un proceso cíclico que garantice su efectividad (Morville, y otros, 1998).

La AI es una disciplina que tiene una estrecha relación con la Usabilidad del sistema. A continuación se especifica el concepto de Usabilidad.

1.6 Usabilidad

De acuerdo con la norma ISO 9241 es definida como el “rango en el que un producto puede ser usado por un grupo de usuarios específicos para alcanzar ciertas metas definidas con efectividad, eficiencia y satisfacción en un contexto de uso especificado. La usabilidad es la percepción de qué tan consistente, organizado, eficiente, productivo, fácil de usar e intuitivo es el proceso de completar una tarea en particular dentro de un sistema” (ISO 9241, 1998).

La usabilidad es definida como “el grado con el que los usuarios pueden realizar una serie de tareas requeridas” (Brinck, y otros, 2002); “calidad de un sistema con respecto a la facilidad de aprendizaje, de uso y satisfacción del usuario” (Rosson, y otros, 2002).

La usabilidad tiene cinco atributos definidos (Nielsen, 1993):

1. **Facilidad de aprendizaje:** El sistema debe ser sencillo de aprender, de tal manera que el usuario pueda comenzar rápidamente a utilizarlo.
2. **Eficiencia:** Debe ser eficiente en su uso, de forma que al aprender a usarlo el usuario tenga un nivel de productividad alto.
3. **Retención sobre el tiempo:** Su uso será fácil de recordar incluso si se deja de trabajar con el sistema por algún tiempo.
4. **Tasas de error por parte de los usuarios:** Se debe procurar que el sistema tenga una baja tasa de errores y en caso de que se produzcan, existirán las facilidades necesarias para proveer una solución.
5. **Satisfacción subjetiva:** Qué tan placentera es la utilización del sistema para los usuarios.

A continuación se especifica en que consiste la Web 2.0 y su presencia en el presente trabajo.

1.7 Web 2.0

En la web las tecnologías presentan un avance acelerado con relación a otros tipos de tecnología en la rama de la informática, son renovadas o mejoradas en breves espacios de tiempo. “La Web 2.0 es la

representación de la evolución de las aplicaciones tradicionales, enfocada hacia el usuario final, quien es el encargado de brindar una adecuada personalización a cada aplicación. Cuando se habla de la Web 2.0, se refiere más a una actitud que a la tecnología” (O’Reilly, 2004).

“La Web 2.0 nace de la necesidad de interactuar más con los usuarios a través del ciberespacio y no quedarse con simples páginas HTML estáticas, las cuales no mostraban contenido actualizado, ni brindaban opciones en la que los usuarios pudieran interactuar en la red. (...) esta nueva era del Internet permite que las personas puedan comunicarse a través de blogs, redes sociales y foros, ocasionando que el uso masivo del Internet deje a un lado la comunicación tradicional” (Canales, 2007).

“La Web 2.0 es la transición que se ha dado de aplicaciones tradicionales hacia aplicaciones que funcionan a través del web enfocadas al usuario final. Se trata de aplicaciones que generen colaboración y de servicios que reemplacen las aplicaciones de escritorio” (Van Der Henst S, 2005).

Teniendo en cuenta los conceptos anteriormente planteados se decide dar cumplimiento al objetivo general siguiendo la tendencia de la Web 2.0 pues el portal para la DIN cumple con lo determinado en (INTEF, 2015). Al ser un sitio dinámico, propicia el intercambio entre usuarios mediante comentarios a noticias y gestión de contenidos como incidencias y necesidades, enfocando la información hacia los usuarios que accederán a él. Además permite la interacción con la información dinámica que se genera según la petición de cada usuario.

1.8 Tecnologías, lenguajes y herramientas

1.8.1 Metodología de desarrollo de software

En ingeniería de software una metodología es un marco de trabajo usado para estructurar, planificar y controlar el proceso de desarrollo en sistemas de información. Según lo planteado por (Piattini Velthuis, 1996) las metodologías de desarrollo de software se pueden agrupar en dos grandes grupos:

- **Ágiles**

Orientadas a la generación de código con ciclos muy cortos de desarrollo manteniendo un proceso incremental, son capaces de permitir cambios en los requisitos de último momento, además el equipo de desarrollo mantiene una comunicación constante con el cliente (Piattini Velthuis, 1996).

Metodologías orientadas a la interacción con el cliente y el desarrollo incremental del software, mostrando versiones parcialmente funcionales del software al cliente en intervalos cortos de tiempo, para que pueda evaluar y sugerir cambios en el producto según se va desarrollando (Pérez González, y otros, 2008).

- **Tradicionales o pesadas:**

Realizan mayor énfasis en la planificación y control del proyecto, en la especificación precisa de requisitos y modelado, estableciendo estrictamente las actividades involucradas, los roles definidos, los artefactos que se deben producir, las herramientas y la documentación usada (Piattini Velthuis, 1996).

Metodologías orientadas al control de los procesos, estableciendo rigurosamente las actividades a desarrollar, herramientas a utilizar y notaciones que se usarán (Pérez González, y otros, 2008).

Debido a que se necesita la introducción constante de cambios en el sistema, la mayor colaboración y acercamiento entre el cliente y el desarrollador, además se cuenta con un equipo de desarrollo pequeño y se necesita mejorar la satisfacción del cliente dado que se involucrará y comprometerá a lo largo del proyecto, sumando su experiencia para optimizar las características del producto final del que poseerá una completa visión; se determinó solo centrarse en el estudio de las metodologías ágiles para darle solución al problema planteado.

Metodología XP

“XP es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. XP se basa en la retroalimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico” (Beck, 1999).

Metodología Scrum

Desarrollada por Ken Schwaber, Jeff Sutherland y Mike Beedle. Está especialmente indicada para proyectos con un rápido cambio de requisitos. El desarrollo de software se realiza mediante iteraciones, denominadas *sprints*, con una duración de 30 días. El resultado de cada *sprint* es un incremento ejecutable que se muestra

al cliente. Otra característica importante son las reuniones a lo largo del proyecto, entre ellas se destaca la reunión diaria de 15 minutos del equipo de desarrollo para coordinación e integración (Canós, y otros, 2003).

Metodología Open Up

“Open Up es una metodología de desarrollo de software, de código abierto diseñada para pequeños equipos organizados, tomando una aproximación ágil del desarrollo. Open UP abraza una filosofía pragmática y ágil que se centra en la naturaleza colaborativa de desarrollo de software. Es un proceso iterativo que es Mínimo, Completo y Extensible que puede utilizarse tal cual o ampliarse para tratar una amplia variedad de tipos de proyecto. Se caracteriza por ser iterativo e incremental, estar centrado en la arquitectura y guiado por los casos de uso. Está organizada dentro de cuatro áreas principales de contenido: Comunicación Y Colaboración, Intención, Solución y Administración” (Sánchez, 2014).

Metodología DAC (Desarrollo Ágil con Calidad)

“La metodología DAC es un proceso de desarrollo de software que combina las metas y prácticas de las áreas de procesos del nivel 2 de CMMI (*Capability Maturity Model Integration*) con las buenas prácticas de la dirección y desarrollo ágil de proyectos de software. Es un proceso colaborativo, recursivo-iterativo, incremental y guiado por procesos y requisitos. Su modelo del proceso es una adaptación del modelo en Cascada a los modelos Programación Extrema y Desarrollo Concurrente. Está enfocado a proyectos pequeños o proyectos grandes divididos en sub-proyectos que desarrollan software de gestión basado en componentes” (Sánchez Méndez, 2013).

Metodología SXP

“La metodología ágil SXP, desarrollada en el 2007 en la Universidad de las Ciencias Informáticas está especialmente indicada para proyectos de pequeños equipos de trabajo, rápido cambio de requisitos o requisitos imprecisos, muy cambiantes, donde existe un alto riesgo técnico y se orienta a una entrega rápida de resultados y una alta flexibilidad. Ayuda a que trabajen todos juntos, en la misma dirección, con un objetivo claro, permitiendo además seguir de forma clara el avance de las tareas a realizar, de forma que los jefes pueden ver día a día cómo progresa el trabajo” (Peñalver Romero, 2008).

“SXP usa SCRUM para la planificación de los proyectos que utilicen metodologías ágiles, dado que SCRUM en sí no es una metodología de análisis ni de diseño, es una metodología de gestión de trabajo. De esta no se toma todo, solo algunas buenas prácticas. Para llevar a cabo el proceso de desarrollo se utiliza XP, con

la idea principal de ir entregando versiones del producto que sean operativas aunque no cuenten con todas las funcionalidades requeridas por el cliente final” (Rodríguez Villar, 2007).

Luego de realizado el análisis de las metodologías ágiles escogidas se seleccionó SXP para desarrollar el portal web para la Dirección de Informatización, pues es una metodología bajo una licencia libre, diseñada en la propia universidad para equipos de desarrollo pequeños (en este caso una sola desarrolladora), que estén muy organizados y se basen en desarrollo iterativo, ágil e incremental. Está centrada en el cliente incorporando al mismo como parte del equipo y las iteraciones son cortas, se utiliza principalmente en proyectos con requisitos muy cambiantes; además hace referencia a la entrega rápida de software de calidad en un corto período de tiempo.



Figura 1. Fases de la metodología SXP (Rodríguez Villar, 2007).

1.8.2 Lenguajes utilizados para el desarrollo de la aplicación

PHP 5.4

“PHP (acrónimo recursivo de *PHP: Hypertext Preprocessor*) es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML” (The PHP Group, 2001 - 2015).

“El código fuente escrito en PHP es invisible al navegador y al cliente ya que es el servidor el que se encarga de ejecutar el código y enviar su resultado HTML al navegador. Tiene la capacidad de expandir su potencial utilizando la enorme cantidad de módulos existentes (llamados ext's o extensiones), así como conectarse con la mayoría de los motores de base de datos” (Welling, y otros, 2005).

CSS 3

Las hojas de estilo en cascada (*Cascading Style Sheets* o *CSS*) son las que ofrecen la posibilidad de definir las reglas y estilos de representación en diferentes dispositivos, ya sean pantallas de equipos de escritorio, portátiles, móviles, impresoras u otros dispositivos capaces de mostrar contenidos web (De Luca, 2010).

HTML 5

“HTML es el lenguaje básico de la Web para crear documentos y aplicaciones para que todos puedan usar en cualquier lugar” (W3C, 2015).

HTML 5 (*HyperText Markup Language*, versión 5) establece una serie de nuevos elementos y atributos que reflejan el uso típico de los sitios web modernos. Mejora el elemento <canvas>, capaz de renderizar elementos 3D en los navegadores (Firefox, Chrome, Opera, Safari e Internet Explorer).

1.8.3 Tecnologías

Razones por las cuales se escogió un *framework* PHP como tecnología de desarrollo:

CMS

Según Rodríguez Gairín: “Las siglas CMS (del término inglés *Content Management Systems*) identifican un conjunto de programas informáticos destinados a gestionar la presentación de los contenidos de una sede Web” (Rodríguez Gairín, 2004).

Los CMS son sistemas que permiten la creación de aplicaciones web de manera fácil y en poco tiempo que poseerán una buena estructura a nivel de base de datos, de archivos y de privilegios. Entre sus ventajas se puede mencionar que pueden ser manejados y configurados por personas que no posean muchos conocimientos de programación. Poseen una amplia comunidad de desarrollo en la que se pueden encontrar distintos módulos, temas y componentes. Estos sistemas son más cómodos para diseñadores que para desarrolladores y son recomendados para aplicaciones que requieran de un tiempo de desarrollo corto (Rodríguez, 2013).

Como desventajas se puede señalar que estos sistemas no permiten grandes modificaciones pues son menos flexibles a cambios. Las modificaciones en el núcleo podrían traer consigo graves errores en el sistema, además si se requiere alguna funcionalidad nueva se necesita crear un módulo, lo cual exige una curva de aprendizaje de toda la API (Rodríguez, 2013).

Framework

Los *frameworks* se pueden definir como un conjunto estructurado de librerías y funciones que facilitan el desarrollo de aplicaciones.

Como una de sus ventajas se puede mencionar que los *frameworks* se ajustan a todo tipo de proyectos. Además brindan la oportunidad de trabajar desde cero en cuanto a estructura, por lo que se pueden ir adaptando a las características del proyecto y hacen más fácil el mantenimiento (Riehle, 2000). Además son más cómodos para programadores y poseen una gran comunidad de desarrollo detrás de cada uno (Rodríguez, 2013).

Dentro de las desventajas que se pueden citar entre los *frameworks* se encuentran las siguientes: dependiendo del *framework* escogido la línea de tiempo de aprendizaje puede ser mayor o menor. Además se deben tener conocimientos de programación de antemano (Rodríguez, 2013).

Luego de analizadas las características planteadas el equipo de desarrollo decidió escoger como tecnología de desarrollo web a los *frameworks* PHP debido a que son más flexibles a cambios y permiten una mayor personalización de las funcionalidades de las aplicaciones a desarrollar.

Teniendo en cuenta que el portal web para la dirección se encuentra actualmente en su primera versión, previéndose que en versiones posteriores se integre con sistemas como el Sistema de Gestión Universitaria (con el objetivo de generar estadísticas y reportes que serían de interés para la comunidad universitaria) y con el Sistema de Gestión de Incidencias (con el objetivo de realizar una gestión avanzada de estos contenidos); la utilización de un *framework* sería la mejor opción.

Teniendo en cuenta la selección de los *frameworks* PHP como tecnología de desarrollo se decidió realizar un estudio de Symfony2, CakePHP2, CodeIgniter y Zend con el objetivo de seleccionar el que mejor se adaptara a la solución. Los aspectos que se tuvieron en cuenta en la comparación se pueden encontrar en el **Anexo 8**.

Luego del análisis de los aspectos de la tabla comparativa entre *frameworks*, el equipo de desarrollo determina la selección de Symfony2 para el desarrollo del portal web para la Dirección de Informatización, debido a los aspectos siguientes: a diferencia de algunos de los *frameworks* presentados Symfony2 realiza la generación de código a través de la consola de comandos lo cual facilita el trabajo a los desarrolladores, utiliza uno de los ORM más populares y mejores según (Eguiluz, 2009) como es Doctrine, facilitando el desarrollo de las aplicaciones al no tener que realizar complejas consultas a la base de datos, pues se trata a las tablas como objetos.

Además hace uso del componente inyección de dependencias, que permite estandarizar y centralizar la forma en que se construyen los objetos en la aplicación y según (Potencier, y otros, 2009) “la inyección de dependencias consiste en pasar a cada componente todo lo que necesita a través de sus constructores, métodos o campos”, esta característica de Symfony2 hace que los desarrolladores no tengan que crear instancias de distintas clases cada vez que las usen, sino “inyectarlas” por medio del método *get()*.

De los *frameworks* estudiados es el único que hace uso del generador de la capa del modelo y del generador de CRUD, esta última funcionalidad es muy usada por los programadores pues permite generar los métodos necesarios para gestión de cualquier contenido solo mediante comandos de consola. También permite la gestión de los *logs* del sistema, registrando en ellos toda la actividad que se genera en la aplicación, integra la validación contra los ataques de XSS y utiliza por omisión un *token* en los formularios para validar los ataques XSRF. Symfony es uno de los *frameworks* mejor documentados, pues se han publicado cinco libros y constantemente se liberan actualizaciones debido a la gran comunidad de desarrolladores que contribuyen en su desarrollo. Además su código fuente incluye por defecto gran cantidad de pruebas unitarias y funcionales.

Se utilizará Symfony 2.3 debido a que según el tiempo de inicio de la investigación esta versión del *framework* constituía la más reciente de largo soporte.

Symfony 2.3

“Symfony es un *framework* que simplifica el desarrollo de una aplicación web mediante la automatización de algunos de los patrones utilizados para resolver las tareas comunes permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación. Está basado en el clásico patrón de diseño web conocido como arquitectura MVC (Modelo-Vista-Controlador). Symfony 2.3 está desarrollado completamente con PHP 5, es compatible con la mayoría de gestores de bases de datos como MySQL, PostgreSQL, Oracle y SQL Server de Microsoft. Se puede ejecutar tanto en plataformas (Unix, Linux, etc.) como en plataformas Windows” (Potencier, 2009).

Esta decisión está dada además por los siguientes factores:

- Flexibilidad: es muy importante el tener varias vías de solución de un problema y Symfony2 brinda más de una vía tanto para la construcción de plantillas como para integración con las bases de datos

y para los formatos de archivos de configuración. Para las plantillas permite su elaboración en PHP pero también ofrece la opción de utilizar el motor de plantillas *Twig*.

- Con respecto a las bases de datos este *framework* permite la integración con gestores de bases de datos tanto relacionales como no relacionales y para los formatos de los archivos de configuración es opcional el uso de YAML, XML o PHP.
- Amplia comunidad de desarrollo: además se toma en cuenta la gran utilización que ha tenido este *framework* PHP en los últimos años, pues muchas empresas lo han definido como la mejor solución a sus necesidades y otras lo han integrado a sus productos como Drupal8.
- Gran cantidad de sistemas que hacen uso de componentes que emplea Symfony2: incluso algunos de los *frameworks* PHP utilizan componentes de los que componen Symfony2, así como algunos CMS entre los que se encuentra Drupal8. Según (SensioLabs, 2015) algunos de los sistemas que hacen uso de componentes de Symfony2 son:
 1. **Drupal 8:** se trata de uno de los CMS más importantes del mundo según (CMSMatrix, 2015). Drupal 8 integró componentes de Symfony2 entre los que se encuentran: *HttpFoundation*, *HttpKernel*, *Routing*, *EventDispatcher*, *DependencyInjection* y *ClassLoader*, lo cual permite integrar aplicaciones de Symfony2 y Drupal.
 2. **phpBB:** es un sistema de foros gratuito que se basa en un conjunto de paquetes de código programados en PHP. Utiliza componentes de Symfony2 como *BrowserKit*, *Config*, *CssSelector*, *Debug*, *DependencyInjection*, *DomCrawler*, *EventDispatcher*, *Filesystem*, *Finder*, *HttpKernel*, *Process*, *Routing* y *Yaml*.
 3. **Laravel:** es un *framework* de aplicaciones web que facilita las tareas comunes que se utilizan en la mayoría de proyectos web, como la autenticación, enrutamiento, sesiones, y el almacenamiento en caché. Utiliza *BrowserKit*, *Console*, *CssSelector*, *Debug*, *DomCrawler*, *EventDispatcher*, *Filesystem*, *Finder*, *HttpFoundation*, *HttpKernel*, *Process*, *PropertyAccess*, *Routing*, *Translation*.
 4. **Silex:** es un *microframework* para PHP 5.3. Está construido sobre los hombros de Symfony2. Hace uso de los componentes de Symfony2 *BrowserKit*, *Config*, *CssSelector*, *Debug*, *DomCrawler*, *EventDispatcher*, *Finder*, *Form*, *HttpFoundation*, *HttpKernel*, *Locale*, *OptionsResolver*, *Process*, *Routing*, *Security*, *Serializer*, *Validator*.

La totalidad de sistemas que hacen uso de componentes de Symfony2 pueden verse en (SensioLabs, 2015).

- Ambientes de trabajo: si se trabaja en un ambiente de desarrollo Symfony2 muestra una barra de "debug" y muestra dónde están los cuellos de botella en la aplicación, cuánta memoria consume y cuánto tiempo se tardó en cargar la página.
- El sistema de rutas: otros *frameworks* como CodeIgniter o Cake PHP obligan al desarrollador a modificar el *htaccess* para poder tener URLs amigables. Además otros *frameworks* fuerzan a que se utilicen sus convenciones y sus nomenclaturas para las rutas. Al contrario, Symfony2 da la libertad de especificar las rutas ya sea a través de un archivo XML, un archivo YML o anotaciones en los controladores.

jQuery 2.1.3

Es una biblioteca de JavaScript que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX a páginas web. jQuery es software libre y de código abierto (The jQuery Foundation, 2015).

Bootstrap 3.0

Es un marco de trabajo de software libre para diseño de sitios y aplicaciones web. Contiene plantillas de diseño con tipografía, formularios, botones, cuadros, menús de navegación y otros elementos de diseño basado en HTML y CSS, así como, extensiones de JavaScript opcionales. Los desarrolladores eligen en un formulario los componentes y ajustes deseados, y de ser necesario, los valores de varias opciones a sus necesidades. El paquete consecuentemente generado ya incluye la hoja de estilo CSS pre-compilada (Cochran, 2012), (Sears, y otros, 2015).

1.8.4 Herramientas a utilizar

PostgreSQL 9.2

“Es un sistema gestor de base de datos (SGBD) relacional orientado a objetos y libre, publicado bajo la licencia BSD y con su código fuente disponible libremente. Utiliza un modelo cliente/servidor y usa multiprocesos en vez de multihilos para garantizar la estabilidad del sistema. La aplicación pgAdmin III es el entorno de escritorio visual que permite conectarse a las bases de datos del PostgreSQL que estén

ejecutándose en cualquier plataforma. Este facilita la gestión y administración de bases de datos ya sea mediante instrucciones SQL o con ayuda de un entorno gráfico” (Martínez, 2010).

Nginx 1.8

“Es un servidor web de alto rendimiento y proxy inverso. Nginx es el cuarto servidor web más popular, y actualmente está sirviendo a más de dos millones de sitios web. Ya que sólo va a la zaga de Apache, IIS y GFE, es efectivamente el segundo servidor web más popular disponible para Linux” (Reese, 2008).

“Nginx es un servidor gratis, de HTTP de código abierto, servidor HTTP de alto rendimiento y proxy inverso, así como también un servidor proxy de IMAP/POP3. Nginx es conocido por su alto rendimiento, estabilidad, set sustancioso de características, configuración simple, y consumo bajo de recursos. A diferencia de servidores tradicionales, Nginx no confía en flujos de ejecución para manejar peticiones” (Nginx Community, 2015).

Debido a que en la Universidad se encuentran en un proceso de migración todos los sistemas sobre Apache hacia Nginx y por pedido del cliente, se utiliza esta herramienta como servidor web.

NetBeans IDE 8.0

Permite programar en distintos lenguajes en los cuales incluye PHP, HTML, CSS y JavaScript, además presenta una serie de características que facilitan en gran parte el desarrollo:

- Multiplataforma.
- Historial de cambios.
- Facilidades para la programación:
 - completamiento de código.
 - comprobación y corrección de errores en tiempo real.
 - resaltado de variables o etiquetas seleccionadas.

Visual Paradigm 8.0

Herramienta para desarrollo de aplicaciones utilizando modelado UML, ideal para quienes están interesados en construcción de sistemas a gran escala y necesitan confiabilidad y estabilidad en el desarrollo orientado a objetos. Constituye una herramienta privada pero en la Universidad de las Ciencias Informáticas, se posee

una licencia que es usada con fines educativos. Dentro de sus características se pueden mencionar (Visual Paradigm International, 2015):

- Modelo y código permanecen sincronizados en todo el ciclo de desarrollo.
- Interoperabilidad con modelos UML.
- Generación de bases de datos.
- Ingeniería inversa de bases de datos.

Nikto 2.1.4

Nikto es una herramienta de escaneo de servidores web que se encarga de efectuar diferentes tipos de actividades tales como: detección de malas configuraciones y vulnerabilidades en el servidor objetivo, detección de ficheros en instalaciones por defecto, listado de la estructura del servidor, versiones y fechas de actualizaciones de servidores, pruebas de vulnerabilidades XSS, ataques de fuerza bruta por diccionario, reportes en formatos txt, csv, html, etc (CIRT.net, 2015).

Algunas de sus características son:

- Soporte SSL (Unix con *OpenSSL* o tal vez de Windows con *ActiveState* de *Perl / NetSSL*).
- Soporte de proxy HTTP completo.
- Escanear múltiples puertos en uno o varios servidores a través de archivo de entrada (incluyendo salida de nmap).
- Técnicas de codificación de IDS *LibWhisker*.
- Identifica el software instalado mediante encabezados, *favicons* y archivos.
- Anfitrión de autenticación con Basic y NTLM.
- Técnicas de mutación a "pescado" para el contenido en servidores web.

Apache JMeter 2.3.1

JMeter es una aplicación de escritorio de código abierto para realizar pruebas funcionales de software y medir el rendimiento. Inicialmente se diseñó para pruebas de aplicaciones web, aunque en versiones posteriores, ha aumentado su funcionalidad a otro tipo de pruebas (The Apache Software Foundation, 2015).

Funcionalidades

- Probar el rendimiento de recursos.
- Pruebas de rendimiento de servidores.
- Construcción y ejecución de un plan de pruebas.
- Simulación.

1.9 Conclusiones parciales

A partir de la fundamentación teórica realizada en esta investigación se puede concluir que:

La caracterización de varios tipos de portales permitió establecer sus principales funcionalidades y características para la determinación de un enfoque de portal corporativo en la propuesta de solución.

La realización del estudio de portales que siguen la tendencia de portal corporativo garantizó un mayor conocimiento de sus funcionalidades y permitió la definición de la estructura de la información del portal web.

Definir el tipo de portal adecuado y las características del mismo permitió identificar herramientas, lenguajes y metodología que regirán el desarrollo del portal web para la Dirección de Informatización.

CAPÍTULO 2: PROPUESTA DE SOLUCIÓN.

2.1 Introducción del capítulo

En el presente capítulo se realiza una descripción de la solución y se explica el proceso de captura de los requisitos funcionales y no funcionales que posee la propuesta de solución, así como las técnicas aplicadas para obtenerlos y sus ventajas. Mediante el modelo de dominio se muestran los principales objetos que se manejan en el contexto del sistema a desarrollar y se realiza una caracterización detallada de la solución a partir de los estilos arquitectónicos y los patrones de diseño aplicables al *framework* PHP Symfony 2.3.

2.2 Proceso de organización de la información del portal web

La AI es un proceso que implica 4 etapas (EVA, 2015), requiere tiempo y diversos conocimientos interdisciplinarios por parte del que la realiza, investigación y dominio de diferentes técnicas. En el presente trabajo no se llevará a cabo el proceso completo de Arquitectura de Información pero sí se tienen en cuenta algunos de sus principios y se aplican técnicas con el objetivo de organizar y diseñar estructuralmente los contenidos de información, como por ejemplo:

1. Se realizaron reuniones con el cliente, aplicándose el diseño participativo para definir las necesidades y contenidos a publicar en el portal.
2. Se realizó un estudio de homólogos que se describe en el acápite 1.4 para identificar elementos que pudieran servir de apoyo en el desarrollo del portal web y la forma de organización de la información a publicar.
3. Se realizó un prototipo del portal que se utilizó para definir elementos de contenido con el cliente.
4. Se etiquetó el contenido de manera simple, tratando de proporcionar información instantánea y evitando incoherencias entre el enlace y la página a la que conduce.
5. Se evitaron cambios bruscos en la navegación manteniendo la misma estructura general en las diferentes secciones.
6. Se muestra al usuario todo el tiempo que se encuentra en el portal web de la DIN por la identificación.
7. Se realizaron críticas del diseño por parte del tribunal que evalúa el presente trabajo, que contribuyeron a mejorar el portal, siendo esta una técnica utilizada en la etapa de Prueba del proceso de AI.

2.3 Descripción, estructuración y análisis de la propuesta de solución

A partir de la realización de un estudio detallado sobre los portales web dedicados a corporaciones y empresas, se evidenció que estos poseen características comunes en cuanto a secciones y servicios. Esto hizo posible obtener una visión que contribuyera a la organización y estructuración de la información en el portal web.

El diseño participativo con el cliente (entre uno de los especialistas de la dirección, el cliente y la desarrolladora) fue la técnica empleada para la elección de los principales contenidos que se deseaban mostrar en el portal web. Esta técnica se aplicó con el objetivo de obtener la mayor cantidad de ideas posibles acerca de la información relevante a ubicar. En este caso el cliente expresó la información que requería ser mostrada y el especialista de la dirección, en conjunto a la desarrolladora, propusieron las diferentes formas de mostrarla, con el objetivo de llegar a un consenso que satisficiera al cliente.

De acuerdo con los resultados arrojados por esta técnica y por el estudio de homólogos, el portal web contará con varias secciones, en las que se mostrarán los contenidos de forma organizada mediante las distintas opciones que ofrece el menú principal.

En la portada se muestra una cabecera publicitaria en la que se exponen los principales servicios que ofrece la Dirección de Informatización, seguida por una vista de las 2 últimas noticias relevantes e interesantes publicadas en las que se permite que los usuarios comenten. Se podrá visualizar el resumen, la fecha, el autor y la cantidad de comentarios que ha recibido esta noticia (**Ver Anexo 1**); así como opciones que permiten ampliar la información sobre la noticia y ver las que se encuentran publicadas anteriormente.

El portal web debe proporcionar información sobre las funciones que realiza la DIN, su estructura, misión, visión y las funciones de cada uno de sus departamentos en la sección **¿Quiénes somos?**; además debe permitir la publicación de los **objetivos estratégicos** que se traza la DIN para el año actual y los **términos de uso** de las aplicaciones web en la UCI. El portal debe posibilitar la revisión de los lineamientos tecnológicos propuestos por la dirección y se enumerarán un conjunto de documentos que constituyen los requerimientos mínimos de documentación en la sección de **Regulaciones de proyecto**.

Se presentarán además, secciones como el **Banco de problemas** que será publicado en el sitio con el objetivo de que la comunidad pueda conocer las deficiencias y necesidades que existen en la universidad y

en la que los usuarios del portal podrán realizar solicitudes de solución a algún problema de los que se encuentran en él mediante la opción “**Realizar solicitud de solución**”.

Los usuarios que accedan al portal pueden autenticarse con las credenciales del dominio *uci.cu*. Además se brinda el servicio de registro de incidencias relacionadas con los sistemas que prestan servicios a la comunidad (la inserción de incidencias es accesible solo para los usuarios autenticados), quedando registradas para su posterior solución en la sección **Incidencias (Ver Anexo 2)**, de esta manera se garantiza la comunicación entre el personal de la Dirección de Informatización y la comunidad universitaria. Los usuarios podrán ver algunas cuestiones en las que tengan dudas mediante la sección de **Preguntas frecuentes (Ver Anexo 3)** y como una forma de ofrecer un mejor acceso a la información del sitio se mostrará el **mapa del sitio (Ver Anexo 5)**.

Con el objetivo de ofrecer mayor información sobre el **Plan de Informatización** que se lleva a cabo en el año actual, se presentará un bloque que mostrará el porcentaje de cumplimiento de los 5 hitos más importantes que se encuentran en ejecución y su cumplimiento.

El portal web para la DIN debe poseer una sección para la gestión de necesidades de informatización (**Necesidades**), que serán insertadas por los encargados de las áreas y revisadas por un especialista de la dirección para aceptarlas o no, esta respuesta será comunicada por correo electrónico.

Se decidió poner en el menú principal las opciones más importantes que pudieran ser requeridas por los usuarios que entren al portal, con el objetivo de que al acceder al sistema encuentren fácilmente la información que buscan.

Luego de analizadas las características del portal web se establece la siguiente jerarquía de roles, en la que cada uno posee acceso específico sobre los contenidos del portal:

- **Usuario anónimo:** Usuario que navega por el portal sin estar autenticado, solo tiene permiso de lectura en el portal.
- **Usuario autenticado:** Tendrá permiso para la publicación de comentarios y para responder preguntas frecuentes; así como para la lectura de contenidos y la creación de nuevas incidencias y solicitudes de informatización.
- **Gestor de área:** Es el encargado de insertar en el portal las necesidades de informatización de su área.

- **Gestor de información:** Es el usuario de la Dirección de Informatización encargado de revisar las necesidades de informatización insertadas y de decidir si proceden o no. Además revisa las solicitudes, determinando si son aceptadas o no y ofrece respuestas a las incidencias de los usuarios acerca del proceso de informatización.
- **Administrador:** Es el responsable de definir los permisos de acceso a los usuarios del sistema. Su principal tarea es mantener el portal actualizado, es decir, gestionar los servicios que va a brindar.

2.4 Modelo del dominio

“El modelo de dominio es un artefacto de la disciplina de análisis, construido con las reglas de UML. Permite mostrar de manera visual los principales tipos de objetos que se manejan en el contexto del sistema a desarrollar. El objetivo del modelado del dominio es comprender y describir las clases del dominio más importantes, y representar como se relacionan mediante asociaciones” (Hall, 2003).

Con el objetivo de poseer una mejor visión y organización de las clases y sus asociaciones se realizó un modelo de dominio que permitió identificar personas, eventos, transacciones y objetos que se manejan. Además permitió que se mostraran los distintos tipos de usuarios que existen en el sistema y su interacción en el mismo, pues en él se representan las relaciones de los usuarios que interactúan con la aplicación y la manera en la que lo hacen; definiendo de esta forma los permisos y restricciones para cada uno. Además se considera un punto de partida para el diseño del portal web, permitiendo al analista comprender el sector al cual el sistema va a servir y a expresar estos conceptos y relaciones en un lenguaje común tanto para desarrolladores como para clientes.

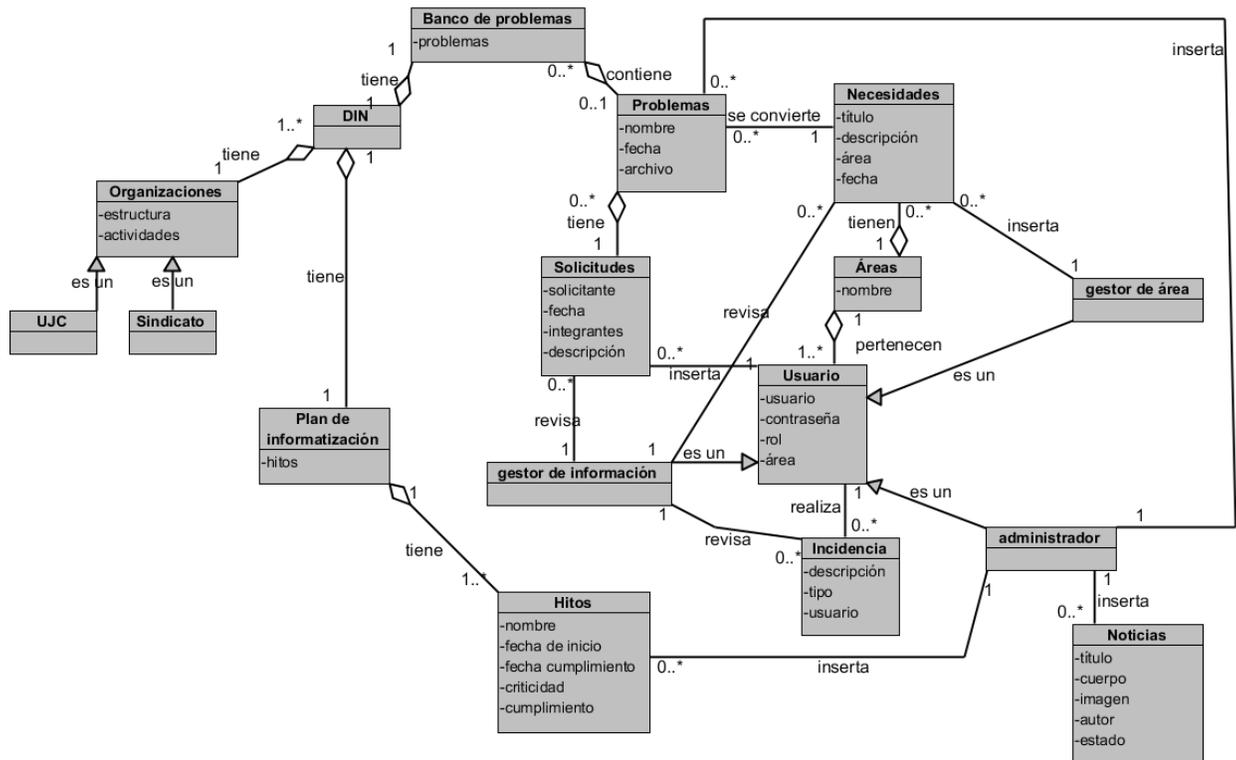


Figura 2. Modelo de dominio.

Para un mayor entendimiento del modelo de dominio y con el objetivo de lograr un lenguaje que sea común para la comprensión del contexto del sistema se muestra a continuación un glosario de términos que identifiquen los conceptos más importantes utilizados en el modelo:

- **Banco de problemas:** contiene la recopilación de los problemas de informatización que existen en las áreas de la Universidad.
- **Usuario:** representa a todos estudiantes y trabajadores de la Universidad que acceden al portal con el objetivo de buscar información sobre la Dirección de Informatización y que insertan incidencias, incluyendo entre ellos a los usuarios con permisos para insertar necesidades y problemas de informatización.
- **Gestor de área:** usuario con los permisos necesarios para insertar necesidades de informatización de su área.
- **Áreas:** distintas áreas de la Universidad.

- **Plan de informatización:** documento rector del proceso de informatización que recoge los proyectos que ejecuta la dirección en un año.
- **Gestor de información:** usuario de la Dirección de Informatización con los permisos necesarios para revisar, aprobar, eliminar y contestar incidencias, necesidades y solicitudes; además de agregar problemas al banco de problemas.
- **Administrador:** usuario que gestiona todos los servicios y contenidos del portal web.
- **Hitos:** proyectos que ejecuta la dirección.
- **Problema:** necesidad de informatización de un área que ha sido aceptada y que se redacta de manera técnica.
- **Necesidad:** contiene la solicitud de informatización de un área.
- **Solicitud:** pedido de contribuir con la solución de un problema.
- **Incidencia:** quejas de los usuarios acerca de los servicios prestados por los sistemas de la dirección.

2.5 Especificación de requisitos

Con el objetivo de lograr el éxito de la aplicación a desarrollar se determinaron las funcionalidades que debe realizar el sistema y sus características y cualidades. La extracción tanto de los requerimientos funcionales como no funcionales se basó en la solicitud de funcionalidades que realizó el cliente, en el estudio de documentación y en una entrevista aplicada a los directivos de la DIN.

El cliente informó acerca de los procesos que se realizan en la Dirección de Informatización y la documentación que se genera; además expresó las dificultades que poseía el proceso de informatización y las causas de las deficiencias, con lo cual se pudo determinar un conjunto de funcionalidades de vital importancia para la dirección, además de las que el mismo cliente propuso como parte del flujo de información que debía seguirse.

Se realizó además un estudio de la documentación de la dirección, principalmente de los manuales de procedimientos y funcionamiento, los cuales proporcionaron información valiosa con respecto a la dirección y a sus operaciones y fueron de gran importancia para introducir a la autora al dominio de operación y el vocabulario que se utiliza.

Finalmente a partir de la aplicación de una entrevista con la finalidad de que se expresara qué contenidos o información se podría mostrar en el portal y porqué; el equipo de desarrollo logró la extracción de los requisitos funcionales, los cuales son las condiciones que un producto debe cumplir y los requisitos no funcionales, que constituyen las cualidades que el producto debe tener. Para una mayor información acerca de las preguntas planteadas en la entrevista (**Ver Anexo 4**).

Requisitos funcionales:

“Describen las interacciones entre el sistema y su ambiente, en forma independiente a su implementación. El ambiente incluye al usuario y cualquier otro sistema externo con el cual interactúe el sistema” (Sommerville, 2002).

“Los requisitos funcionales describen las funciones que el software va a ejecutar; por ejemplo, ajustarse a un formato de texto o modular una señal. Se conocen también como capacidades” (Kotonya, y otros, 2000).

Seguidamente se muestra una lista de los requisitos funcionales del sistema. De forma general el sistema está compuesto por las funcionalidades ordenadas según la prioridad del cliente que se muestran en el documento Lista de Reserva del Producto.

Requisitos funcionales	Complejidad
RF1. Autenticar usuarios.	Alta
RF2. Mostrar estructura del sindicato.	Baja
RF3. Mostrar la estructura de la UJC.	Baja
RF4. Insertar necesidades de informatización.	Alta
RF5. Revisar las necesidades de informatización.	Media
RF6. Eliminar necesidades de informatización.	Media
RF7. Mostrar necesidades de informatización.	Baja
RF8. Insertar solicitud.	Alta
RF9. Modificar solicitud.	Media
RF10. Eliminar solicitud propia.	Media
RF11. Mostrar solicitud.	Alta
RF12. Revisar solicitud.	Alta
RF13. Eliminar solicitudes.	Media

RF14. Publicar el banco de problemas.	Media
RF15. Insertar hito.	Alta
RF16. Modificar hito.	Media
RF17. Eliminar hito.	Media
RF18. Mostrar los 5 hitos más importantes.	Baja
RF19. Mostrar la misión y visión de la Dirección de Informatización.	Baja
RF20. Mostrar la infografía de la Dirección de Informatización.	Baja
RF21. Mostrar los objetivos estratégicos de la Dirección de Informatización.	Baja
RF22. Mostrar lineamientos tecnológicos.	Baja
RF23. Mostrar requerimientos mínimos de informatización.	Alta
RF24. Insertar requerimientos mínimos de informatización.	Alta
RF25. Eliminar requerimientos mínimos de informatización.	Media
RF26. Modificar requerimientos mínimos de informatización.	Media
RF27. Agregar noticia.	Alta
RF28. Modificar noticias.	Alta
RF29. Eliminar noticias.	Media
RF30. Mostrar noticias.	Media
RF31. Realizar búsquedas simples en el sitio.	Baja
RF32. Mostrar el mapa del sitio.	Baja
RF33. Publicar los Términos de uso del sitio.	Baja
RF34. Insertar preguntas frecuentes.	Alta
RF35. Mostrar preguntas frecuentes.	Alta
RF36. Eliminar preguntas frecuentes.	Media
RF37. Modificar preguntas frecuentes.	Media
RF38. Insertar incidencias.	Alta
RF39. Modificar incidencias.	Media
RF40. Eliminar incidencias.	Media
RF41. Mostrar incidencias.	Alta
RF42. Revisar incidencias.	Media

RF43. Mostrar enlaces de interés.	Baja
RF44. Insertar tipo de incidencia.	Alta
RF45. Modificar tipo de incidencia.	Alta
RF46. Eliminar tipo de incidencia.	Media
RF47. Mostrar tipo de incidencia.	Media
RF48. Insertar comentario.	Alta
RF49. Eliminar comentario.	Media
RF50. Mostrar comentario.	Media
RF51. Mostrar todos los hitos del Plan de Informatización.	Baja
RF52. Mostrar cantidad de comentarios de una noticia.	Media
RF53. Insertar usuario.	Alta
RF54. Modificar usuario.	Alta
RF55. Eliminar usuario.	Media
RF56. Mostrar usuario.	Media

Tabla 1. Complejidad de los requisitos funcionales.

Requisitos no funcionales:

“Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener, además son aspectos importantes que el producto debe cumplir para lograr que el mismo sea atractivo, usable, rápido o confiable” (Pressman, 2002).

“Los requisitos no funcionales se conocen a veces como apremios o requisitos de calidad. Pueden ser clasificados más a fondo según si son requisitos de funcionamiento, requisitos de capacidad de mantenimiento, requisitos de seguridad, requisitos de confiabilidad, o uno de muchos otros tipos de requisitos del software” (Kotonya, y otros, 2000).

“Los requisitos no funcionales detallan las propiedades o cualidades que el producto debe tener, aumentan funcionalidad al sistema, pues hacen al producto fácil de usar, rápido y confiable. Se encuentran separados por categorías” (Oliveros, 2012).

A continuación se listan los requisitos no funcionales del sistema agrupados según un conjunto de categorías:

Requerimientos de Software

Para el correcto despliegue y funcionamiento del portal web se utilizarán las siguientes herramientas:

- RNF1. Servidor web Nginx 1.8.
- RNF2. Gestor de base de datos PostgreSQL en su versión 9.2.

Requerimientos de Hardware

- RNF3. El servidor de aplicaciones web y de base de datos deben poseer como mínimo un CPU Dual Core a 2.30 GHz.
- RNF4. El servidor de aplicaciones web debe poseer una capacidad mínima de 10 GB pues la aplicación no debe crecer en el tiempo.
- RNF5. El servidor de base de datos debe poseer una capacidad mínima de 80 GB.
- RNF6. Los servidores web y de base de datos deben poseer como mínimo 1 GB de memoria RAM.

Requerimientos de Seguridad

1. Confidencialidad

- RNF7. El sistema debe registrar cualquier fallo de su funcionamiento con el objetivo de permitir su solución por los administradores.
- RNF8. En caso de que ocurran fallas, los errores deben mostrarse sin detalles de información que puedan comprometer la integridad y seguridad del mismo. Sólo se mostrarán detalles ampliados del error a usuarios con privilegios de administración.
- RNF9. En el sistema se deben establecer mecanismos que garanticen la confiabilidad e integridad de la información ante posibles accesos no autorizados tales como: la protección CSRF, la autenticación mediante usuario y contraseña y garantizar los niveles de acceso a través de roles.
- RNF10. El acceso al portal será controlado a través de la autenticación por nombres de usuario y contraseñas que provee el servicio de la pasarela de datos de la Universidad.
- RNF11. El sistema podrá estar inactivo durante 5 minutos, al cabo de este tiempo se cerrará la sesión que esté abierta.

2. Integridad

- RNF12. Se garantizará la integridad de la información mediante mecanismos de control de acceso, utilizando, usuario, contraseña y niveles de accesos mediante el uso de roles para cada usuario, de

manera que cada uno pueda tener disponible solamente las opciones que se encuentran en correspondencia con su rol y las funciones que el mismo realiza.

Requerimientos de Usabilidad

RNF13. El sistema debe presentar una estructura de navegación semejante en todo el sitio.

RNF14. El sistema debe proveer una iconografía descriptiva y representativa de las funcionalidades del sistema.

RNF15. El sistema debe presentar una letra más grande en los encabezados de sección que posibilite la orientación en el portal web, tanto para los usuarios con conocimientos avanzados en informática como los usuarios más inexpertos.

RNF16. El sistema debe tener buena visibilidad en los navegadores web a partir de las siguientes versiones: Internet Explorer 9, Firefox 4, Chrome 16 y Opera 9 (Gitonga, 2014).

RNF17. El sistema debe mostrar un mapa del sitio que permita una fácil navegación y localización de los contenidos.

RNF18. El Sistema podrá ser visualizado en dispositivos móviles para las resoluciones 320x480, 800x600 y 1024x768.

Requerimientos Legales

RNF19. Utilizar sistemas de código abierto para el desarrollo de la aplicación.

2.6 Descripción de estilos arquitectónicos y patrones de diseño

La presencia de estilos arquitectónicos y patrones de diseño constituyen buenas prácticas que poseen como objetivos codificar y hacer reutilizable un conjunto de principios a fin de diseñar aplicaciones de alta calidad.

Existen actualmente una gran cantidad de patrones y estilos arquitectónicos y la decisión de cuál utilizar es fundamental en el desarrollo de un sistema. El estilo arquitectónico y los patrones de diseño utilizados en el presente trabajo son los heredados por el *framework* Symfony 2.3 debido a que será el empleado en la implementación del portal web para la Dirección de Informatización.

2.6.1 Estilo arquitectónico en Symfony 2.3

Apoyados en los principios y arquitectura que propone el marco de trabajo Symfony 2.3, para el desarrollo del portal web para la Dirección de Informatización se define una arquitectura en capas basada en el patrón

MVC (Modelo-Vista-Controlador), debido a las facilidades que ofrece tanto en términos de escalabilidad (pues divide la lógica del negocio de la lógica del diseño) como en la sencillez de mantenimiento, además de mantener un bajo acoplamiento.

Dicho patrón divide la aplicación en tres capas o niveles de abstracción las cuales son: Modelo, Vista y Controlador. Este patrón se muestra de forma implícita en el uso del marco de trabajo Symfony 2.3, definiendo este último dónde se ubican las clases del modelo, las de la vista y las del controlador.

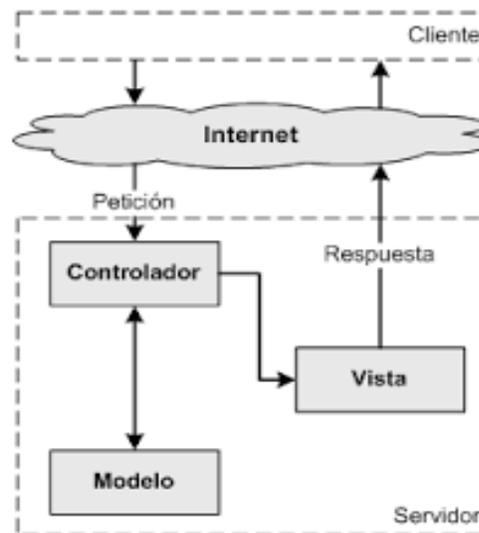


Figura 3. Patrón MVC en Symfony2 (SensioLabs, 2007).

A continuación se muestra la arquitectura propuesta para el portal web para la DIN basada en MVC.

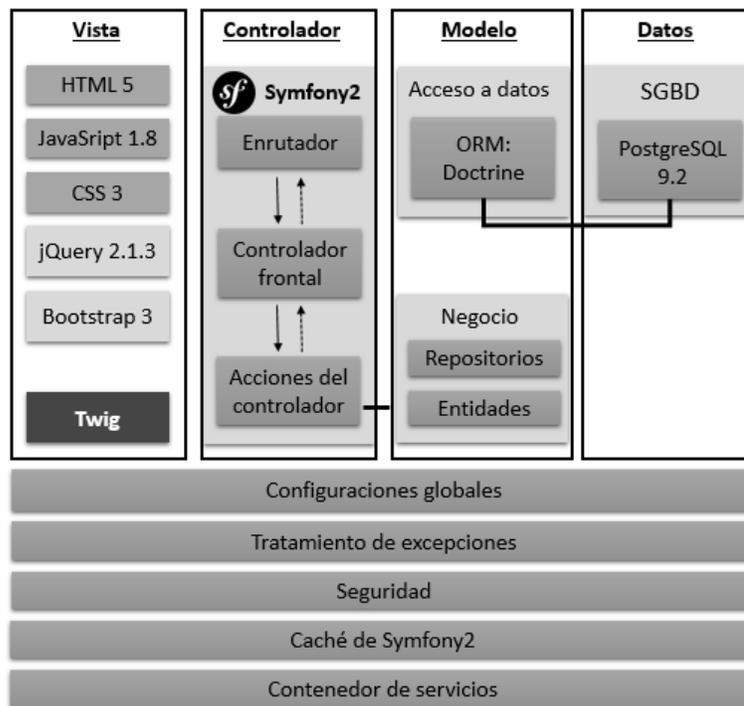


Figura 4. Arquitectura del portal web para la Dirección de Informatización.

Vista: La vista es la capa con la que interactúan directamente los usuarios finales, dentro de ella se ubica todo lo referente a la visualización de la información en las páginas de la aplicación.

La lógica de las vistas será manejada por el motor de plantillas Twig, la biblioteca jQuery, en conjunto con los archivos CSS, JavaScript y HTML, que se encargarán de estructurar y aplicar estilos a las interfaces creadas. Dichos archivos se encuentran ubicados en el directorio `/Resources/views` dentro de cada *bundle* de la aplicación y específicamente en `"DIN/AppBundle/Resources/public"` en el portal web para la Dirección de Informatización.

Controlador: El controlador es la capa intermedia entre la vista y el modelo. Es la encargada de responder a las acciones del usuario e invocar cambios en el modelo o generar la vista apropiada, dependiendo de las peticiones del usuario.

En Symfony 2.3 como marco de trabajo para el desarrollo de la solución, queda evidenciada esta capa a través del Controlador Frontal, mediante los archivos `app.php` para entornos de producción y `app_dev.php` para entornos de desarrollo. Por último las clases *Controller*, que controlarán el flujo de información que se

recibe y se envía hacia la vista, y estarán ubicadas dentro del paquete “DIN/AppBundle/Controller” junto a las demás clases que se localizan en el directorio /Controller de cada *bundle* en la estructura del portal web a desarrollar.

Modelo: En la estructura del portal web para la Dirección de Informatización esta capa está dividida en 2, la Capa de Negocio y la Capa de Acceso a datos.

- **Capa de Negocio:** En la Capa de Negocio se localizan las entidades y los repositorios. Las entidades son la representación de las tablas de la base de datos previamente mapeadas por el ORM Doctrine que se encuentran en “DIN/AppBundle/Entity”. Para abstraer la lógica de negocio de los controladores se crean repositorios de entidades personalizados que permiten llevar el manejo de la base de datos a una clase aislada del controlador.
- **Capa de acceso a datos:** La Capa de Acceso a datos es la encargada de establecer la conexión con la base de datos. Dentro de esta capa se encuentra ubicado el ORM Doctrine, que posibilita la separación en la aplicación del gestor de base de datos mediante su lenguaje propio de consultas DQL.

Elementos transversales: De manera transversal se pueden encontrar las configuraciones globales, el tratamiento de excepciones, los componentes de seguridad, la caché de Symfony2 y el contenedor de servicios.

2.6.2 Patrones de diseño en Symfony 2.3

El patrón es una pareja de problema/solución con un nombre y que es aplicable a otros contextos, con una sugerencia sobre la manera de usarlo en situaciones nuevas (Larman, 1999).

Para la implementación de Symfony se utilizan varios patrones, situándolos en las capas de Modelo y Control que plantea el patrón arquitectónico MVC (Potencier, y otros, 2009).

Patrones GRASP

- **Experto:** Es uno de los patrones que más se utiliza cuando se trabaja con Symfony, con la inclusión del ORM *Doctrine* para mapear la Base de Datos. Symfony utiliza este ORM para realizar su capa de abstracción en el modelo, encapsular toda la lógica de los datos y generar las clases con todas las funcionalidades comunes de las entidades, las clases de abstracción de datos poseen un grupo

de funcionalidades que están relacionadas directamente con la entidad que representan y contienen la información necesaria de la tabla que representan. El patrón Experto se encarga de asignar una responsabilidad al experto en información: la clase que cuenta con la información necesaria para cumplir la responsabilidad.

Algunas ventajas del patrón Experto son según (Larman, 1999):

- Se conserva el encapsulamiento, pues los objetos se valen de su propia información para hacer lo que se les pide. Esto soporta un bajo acoplamiento, lo que favorece al hecho de tener sistemas más robustos y de fácil mantenimiento.
- El comportamiento se distribuye entre las clases que cuentan con la información requerida, alentando con ello definiciones de clases “sencillas” y más cohesivas que son más fáciles de comprender y de mantener.

Este patrón se evidencia en el desarrollo del portal web para la Dirección de Informatización siendo las entidades las expertas en la información. En la siguiente figura se muestra un ejemplo donde se especifica que la entidad *Hito* es la responsable de conocer su peso.

```
/**
 * @var integer
 * @Assert\Range(
 *     min = "0",
 *     max = "10",
 *     minMessage = "El valor mínimo permitido para el campo peso es 0",
 *     maxMessage = "El valor máximo permitido para el campo peso es 10")
 * @Assert\Type(type="integer", message="El valor entrado para 'Peso' debe ser un entero.")
 * @ORM\Column(name="peso", type="integer", nullable=true)
 */
private $peso;

/**
 * @return int
 */
public function getPeso()
{

/**
 * @param int $peso
 */
public function setPeso($peso)
{
```

Figura 5. Uso del patrón Experto en la clase *HitoController.php*.

- **Creador:** “El patrón Creador guía la asignación de responsabilidades relacionadas con la creación de objetos, tarea muy frecuente en los sistemas orientados a objetos. El propósito fundamental de este patrón es encontrar un creador que se debe conectar con el objeto producido en cualquier evento. Al escogerlo como creador, se da soporte al bajo acoplamiento, constituyendo su principal beneficio, lo cual supone menos dependencias respecto al mantenimiento y mejores oportunidades de reutilización” (Larman, 1999).

Este patrón se evidencia en el desarrollo del portal web para la Dirección de Informatización en las clases controladoras, que son las responsables de la construcción de los formularios que se visualizarán en las vistas de la aplicación.

```
private function createCreateForm(Incidencia $entity)
{
    $form = $this->createForm(new IncidenciaType(), $entity, array(
        'action' => $this->generateUrl('incidencia_create'),
        'method' => 'POST',
    ));

    $form->add('submit', 'submit', array('label' => 'Create'));

    return $form;
}
```

Figura 6. Uso del patrón Creador en la clase *IncidenciaController.php*.

- **Alta cohesión:** “En la perspectiva del diseño orientado a objetos, la cohesión (o más exactamente, la cohesión funcional) es una medida de cuán relacionadas y enfocadas están las responsabilidades de una clase. Una alta cohesión caracteriza a las clases con responsabilidades estrechamente relacionadas que no realicen un trabajo enorme” (Larman, 1999).

Algunas ventajas del patrón Alta cohesión son:

- Mejora la claridad y la facilidad con que se entiende el diseño.
- Se simplifican el mantenimiento y las mejoras en funcionalidad.
- A menudo se genera un bajo acoplamiento.
- La ventaja de una gran funcionalidad soporta una mayor capacidad de reutilización, porque una clase muy cohesiva puede destinarse a un propósito muy específico.

Symfony permite la organización del trabajo en cuanto a la estructura del proyecto y la asignación de responsabilidades con una alta cohesión. En el portal web para la Dirección de Informatización se evidencia mediante la clase *Actions*, que está formada por varias funcionalidades que están estrechamente relacionadas, siendo la misma la responsable de definir las acciones para las plantillas y colaborar con otras para realizar diferentes operaciones, instanciar objetos y acceder a las *properties*.

- **Bajo acoplamiento:** Las clases que implementan la lógica del negocio y de acceso a datos se encuentran en el modelo, las cuales no tienen asociaciones con las de la vista o el controlador, lo que proporciona que la dependencia en este caso sea baja.

El acoplamiento según (Larman, 1999) “es una medida de la fuerza con que una clase está conectada a otras clases, con que las conoce y con que recurre a ellas. Una clase con bajo (o débil) acoplamiento no depende de muchas otras; “muchas otras” depende del contexto. El bajo acoplamiento soporta el diseño de clases más independientes, que reducen el impacto de los cambios, y también más reutilizables, que acrecientan la oportunidad de una mayor productividad. No puede considerarse en forma independiente de otros patrones como Experto, sino que más bien ha de incluirse como uno de los principios del diseño que influyen en la decisión de asignar responsabilidades.”

En la implementación del portal web para la Dirección de Informatización este patrón se evidencia en las entidades que son las clases más reutilizadas y son totalmente independientes, pues no están asociadas ni a la vista ni al controlador.

- **Controlador:** El patrón Controlador se encarga de asignar la responsabilidad del manejo de un mensaje de los eventos de un sistema a una clase que represente una de las siguientes opciones (Larman, 1999):
 - El “sistema” global (controlador de fachada).
 - La empresa u organización global (controlador de fachada).
 - Algo en el mundo real que es activo (por ejemplo, el papel de una persona) y que pueda participar en la tarea (controlador de tareas).

Todas las peticiones web son manipuladas por un solo controlador frontal que es el punto de entrada único de toda la aplicación en un entorno determinado.

Un defecto frecuente al diseñar controladores consiste en asignarles demasiada responsabilidad. Normalmente un controlador debería delegar a otros objetos el trabajo que ha de realizarse mientras coordina la actividad.

Este patrón se evidencia a través de las clases controladoras que se localizan en el directorio *Controller* perteneciente a cada *Bundle* del portal web para la Dirección de Informatización y en el uso del controlador frontal ubicado en directorio web de cada aplicación concebida en Symfony 2.3.

```

/**
 * @Route("/", name="pagina_inicio")
 */
public function indexAction(Request $request)
{
    $em = $this->getDoctrine()->getManager();

    // ...

    return $this->render('AppBundle:Default:index.html.twig', array(
        'pagination' => $pagination,
    ));
}

```

Figura 7. Uso del patrón Controlador en la clase *DefaultController.php*.

Patrones GoF

- **Patrón Decorador:**

“Extiende la funcionalidad de un objeto dinámicamente de tal modo que es transparente a sus clientes, utilizando una instancia de una subclase de la clase original que delega las operaciones al objeto original. Provee una alternativa muy flexible para agregar funcionalidad a una clase” (Gamma, y otros, 1995).

En el archivo nombrado *layout.html.twig* es en uno de los que se pone de manifiesto este patrón. Este archivo, conocido también como plantilla global, guarda el código HTML que es usual en todas las páginas del sistema, para no tener que repetirlo en cada página. El contenido de la plantilla se integra en el *layout*, o si se mira desde el otro punto de vista, el *layout* decora la plantilla.

En la implementación del portal web para la Dirección de Informatización el patrón Decorador se evidencia en esta potencialidad que posee *Twig* relacionada con la herencia entre plantillas, es decir,

en el uso de una plantilla global que contendrá los elementos comunes del sitio para las vistas que decorarán las demás páginas de la aplicación.

```
{% extends '::layout.html.twig' %}

{% block content %}
  <div class="well">
    <h4>Lista de problemas</h4>
    <hr/>
  </div>
{% endblock %}
```

Figura 8. Empleo del patrón Decorador en la plantilla *layout.html.twig*.

- **Patrón *Front controller* (Controlador frontal):**

Posee una estructura bien organizada de controladores, que comienza desde el “*index.php*” del ambiente y termina en los “*Actions*”. Cada clase de esta capa tiene su responsabilidad y es única, hay controladores que se encargan de la seguridad del sistema trabajando con ficheros YML, y otros que se encargan de identificar mediante algunos datos las clases que deben realizar determinadas tareas.

Otros patrones

Inyección de dependencias

“La inyección de dependencias consiste en pasar (inyectar) a las clases todos los objetos que necesitan (dependencias) ya creados y configurados. Se utiliza cuando se requiere que la aplicación pueda generar mensajes de *debug* o depuración con información sobre las operaciones que está ejecutando. Consiste en crear y configurar primero el objeto Archivo y pasárselo después a la clase *Logger*, Archivo es una clase genérica llamada con métodos que permitan manipular los contenidos de un archivo y *Logger* es una clase que utiliza la clase Archivo para guardar los mensajes en el archivo de *log*” (Eguiluz, 2007).

Dicho patrón garantiza que se suministren objetos a una clase en lugar de ser la propia clase quien cree el objeto, según las relaciones que fueron establecidas entre las clases.

Este patrón es la clave para la comprensión del funcionamiento del marco de trabajo, permitiendo que este sea rápido y flexible, se evidencia en los servicios, que no son más que un objeto PHP que se crea cuando es necesario utilizar la función a la que hace referencia dicho servicio y en los contenedores de

servicios los cuales son objetos PHP que gestionan la creación de instancias de servicios, es decir, objetos. Sin el contenedor de servicios sería necesario crear cada servicio manualmente. En el portal web para la Dirección de Informatización se evidencia en las llamadas que se hacen desde los controladores a objetos que no se han creado previamente en estos controladores, sino que se “inyectan” mediante el método `get()`.

```

public function responderIncidenciaAction($id)
{
    if ($this->get('security.context')->isGranted('ROLE_ADMIN')) {
        $em = $this->getDoctrine()->getManager();

        $incidencia = $em->getRepository('AppBundle:Incidencia')->find($id);

        if ($incidencia) {
            return $this->render('@App/Incidencia/responder.html.twig', array('incidencia' => $incidencia));
        } else {
            return new Response("404 No encontrado");
        }
    }
}

```

Figura 9. Patrón Inyección de dependencias.

2.7 Modelo de despliegue de la aplicación

El Diagrama de Despliegue es utilizado para capturar los elementos de configuración del procesamiento y las conexiones entre dichos elementos. También se utiliza para visualizar la distribución de los componentes de software en los nodos físicos. Entre los nodos existen relaciones que representan los protocolos de comunicación que se utilizan para acceder a cada uno.

En la figura se muestra el diagrama de despliegue que corresponde a la solución propuesta. PC_Cliente representa las computadoras de los usuarios que se conectan al sistema, las cuales realizan peticiones al servidor web mediante el protocolo HTTPS. Este servidor mantendrá una conexión mediante la arquitectura TCP/IP al servidor de bases de datos y mediante el protocolo SMTP con el servidor de correo.

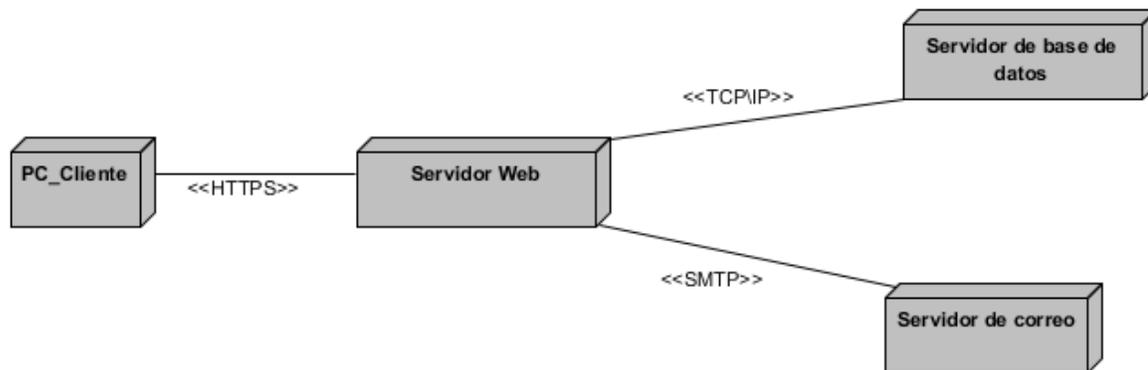


Figura 10. Modelo de despliegue.

Descripción de elementos e interfaces de comunicación:

<<SMTP>>: “El protocolo SMTP (Protocolo simple de transferencia de correo) es el protocolo estándar que permite la transferencia de correo de un servidor a otro mediante una conexión punto a punto” (Kioskea.net, 2015).

<<HTTPS>>: “Protocolo seguro de transferencia de hipertexto (HTTPS) es un protocolo de comunicación diseñado para transferir información encriptada entre computadoras sobre la *World Wide Web*. HTTPS es HTTP haciendo uso de la capa de conexión segura (SSL)” (Microsoft, 2015). Protocolo para establecer a través del puerto 443 la conexión segura entre la PC_Cliente y el servidor de aplicaciones. La conexión es por cable vía modem, LAN o red inalámbrica con una velocidad de más de 64 Kbps.

<<TCP/IP>>: TCP/IP es un conjunto de protocolos. La sigla TCP/IP significa "Protocolo de control de transmisión/Protocolo de Internet" (Kioskea.net, 2015). Este conjunto de protocolos establece la conexión entre el servidor de aplicaciones y el servidor de base de datos. Para el servidor de base de datos de PostgreSQL se define el puerto 5432.

PC_Cliente: Computadora desde donde se realizan las peticiones de información y hacia la que regresan las respuestas de esas peticiones.

Servidor web Nginx: “Nginx es un servidor HTTP de alto rendimiento y proxy inverso, así como también un servidor proxy de IMAP/POP3” (Nginx Community, 2015).

Servidor de correo Zimbra: Proveedor de servicios de correo electrónico en la Universidad.

SGBD PostgreSQL: “Es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD y con su código fuente disponible libremente” (PostgreSQL-es, 2010).

2.8 Conclusiones parciales

En el presente capítulo se describieron las características del portal. Se detallaron los elementos significativos de la arquitectura de software, así como los requisitos funcionales y no funcionales. Luego del análisis realizado se arriba a las siguientes conclusiones:

La aplicación de las técnicas de recopilación de información tales como diseño participativo, garantizó la obtención de los requerimientos funcionales y no funcionales, que sirvieron de guía para desarrollar las distintas funcionalidades de la propuesta de solución.

La descripción de los estilos y patrones arquitectónicos de Symfony2 permitió que se definieran con facilidad la composición y el comportamiento del sistema, para alcanzar los requerimientos de calidad.

El estudio de la estructura y funcionamiento de Symfony2 ofreció una visión gráfica de las especificaciones de las clases y las interfaces del sistema gestionadas por este *framework*.

Definir el diagrama de despliegue de la aplicación propició la comprensión de la relación entre los elementos de hardware del sistema.

La representación y descripción de los artefactos generados garantizaron un mejor entendimiento de los flujos de trabajos presentes en el proceso de informatización.

CAPÍTULO 3: IMPLEMENTACIÓN Y EVALUACIÓN DE LA SOLUCIÓN PROPUESTA.

3.1 Introducción

En este capítulo se presenta el diagrama de componentes y se realizan pruebas a la solución desarrollada con el objetivo de evaluar los resultados obtenidos en el diseño e implementación del sistema. Se analizan las diferentes pruebas de software que fueron aplicadas para medir la calidad de la aplicación, así como los estándares de codificación utilizados por el equipo de desarrollo en todo el proceso de implementación para mejorar la comprensión y organización del código.

3.2 Diagrama de componentes

En un diagrama de componentes se representa un sistema de software dividido en componentes, estos a la vez son elementos del sistema que ofrecen un conjunto de funcionalidades o servicios; además, muestra su organización y dependencias representadas en una vista estática del sistema. En él se sitúan librerías, tablas, archivos, ejecutables y documentos que formen parte de la aplicación. En la siguiente figura se muestra el diagrama de componentes del sistema.

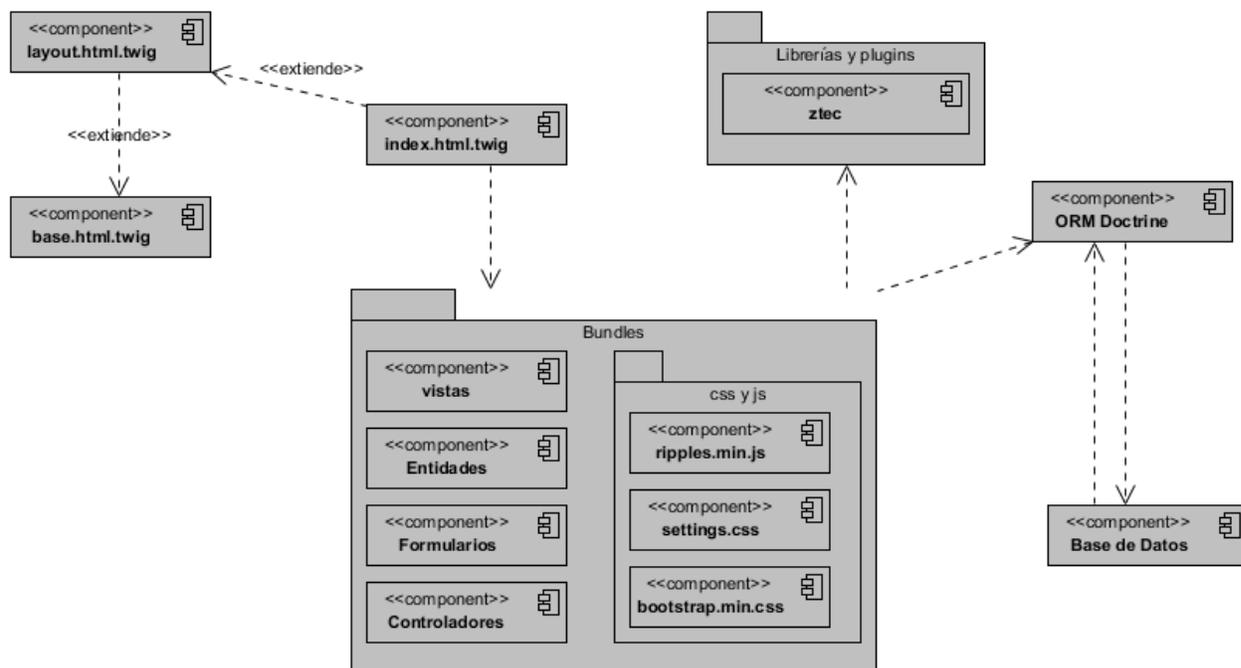


Figura 11. Diagrama de componentes.

Para una mayor especificidad del diagrama mostrado a continuación se explicarán los componentes:

- **base.html.twig:** plantilla en la que se encuentra el diseño principal de la aplicación.
- **layout.html.twig:** plantilla que define la estructura para cada sección del sitio. La misma hereda de base.html.twig.
- **index.html.twig:** página cliente mediante la que se realizan las peticiones al servidor, la misma hereda de layout.html.twig.
- **Bundle:** paquete de funcionalidades en el que se agrupa todo el código de la aplicación estructurado siguiendo el patrón arquitectónico MVC.
- **Vistas:** páginas clientes de la aplicación, son las encargadas de mostrar la información al usuario de forma gráfica.
- **Controladores:** páginas servidoras de la aplicación mediante las cuales se realizan las funcionalidades. Son los intermediarios entre la vista y el modelo, quienes controlan las interacciones del usuario solicitando los datos al modelo y entregándolos a la vista para que ésta lo presente al usuario.
- **Entidades:** clases PHP que constituyen una réplica de las entidades de la base de datos en el proyecto.
- **Bibliotecas y plugins:** conjuntos de archivos que permiten agregar nuevas funcionalidades en el proyecto.
- **ORM Doctrine:** conjunto de objetos que permiten acceder a los datos en la base de datos y que contienen en sí mismos el código necesario para hacerlo.

3.3 Estándares de codificación

Cuando se desarrollan aplicaciones web es importante que el código sea fácil de leer y modificar, para esto se deben seguir una serie de normas comunes para todos los desarrolladores. En Symfony2 se siguen estándares para facilitar el desarrollo de estas aplicaciones y la contribución de sus comunidades, algunos de estos estándares son:

Indentación: La indentación consiste en insertar dos espacios en determinadas líneas de código y no se debe dejar espacios en blanco al final de cada línea para facilitar su comprensión.

Etiquetas de apertura y cierre de PHP: Siempre se debe utilizar la versión de las etiquetas (`<?php`) y (`?>`) para abrir y cerrar el código PHP, por lo general se omite la etiqueta de cierre (`?>`) al final de algunos archivos, evitando que se queden espacios olvidados no deseados al final luego de la etiqueta de cierre (`?>`), que serían identificados como salida HTML y podrían provocar errores, por lo que la etiqueta de cierre final del archivo (`?>`) es opcional.

Operadores: Los operadores binarios, utilizados entre dos valores, deben estar separados a ambos lados del operador por un espacio. Esto se aplica a operadores como `+`, `-`, `*`, `/`, `=`, `==`, `!=`, `>`, `<`, `.` (Concatenación de cadenas), `.=`, `+=`, `-=`, etc. Los operadores unarios como `++`, `--` no deben tener separación.

Uso de comillas: pueden ser usadas tanto las comillas simples ('cadena') como las comillas dobles ("cadena") para delimitar las cadenas de caracteres, si es necesario el uso de variables dentro de la cadena, se deben usar las comillas dobles.

Uso de punto y coma (;) en código PHP: el uso del punto y coma (;) al final del código en las líneas individuales es siempre obligatorio.

Funciones: Los nombres de las funciones se escribirán utilizando el estilo camello (la primera palabra comenzará con minúscula y la letra inicial de las próximas palabras estará en mayúscula), con el objetivo de evitar duplicidad de funciones. Symfony2 establece que el nombre de cada función que se defina debe terminar con la palabra **Action**. Después del nombre de la función el paréntesis de los argumentos no debe llevar espacios, pero sí los argumentos entre sí, luego de cada coma (,).

Comentar el código: El uso de los comentarios pueden hacerse utilizando las etiquetas `/* */` para comentarios en varias líneas y `//` para comentarios de una única línea.

Estructuras de control: Las estructuras de control deben cumplir un conjunto de normas para su correcto funcionamiento; las estructuras (*if*, *while*, *for*, etc.) y el primer paréntesis deben tener un espacio intermedio, para no confundirlas con la nomenclatura de las funciones; la llave de apertura (`{`) estará en la primera línea separada por un espacio y es recomendado usar las dos llaves (`{ }`) aunque el código lo permita; y las estructuras *else* y *elseif* serán escritas en la línea siguiente de la llave de cierre anterior (`}`).

Arrays: Los valores dentro de un *array* deben ser separados por un espacio después de la coma. El operador => debe separarse por un espacio a ambos lados. Cuando la línea de declaración del *array* supera los 80 caracteres, cada elemento se debe escribir en una única línea, indentándolo una vez (usando dos espacios).

3.4 Pruebas

En la evaluación del portal web para la Dirección de Informatización es necesario tener en cuenta varias características dependiendo del tipo de aplicación, además del objetivo que persiguen las pruebas a realizar. Después de analizar las propiedades del portal web se determinó que es necesario medir su reacción integral frente a diversas acciones que podrán efectuar los usuarios, para ello es conveniente la selección de las siguientes pruebas: pruebas de funcionalidad, pruebas de seguridad, pruebas de carga y pruebas de estrés. A continuación se describirán las pruebas seleccionadas y los resultados arrojados con su ejecución.

3.4.1 Pruebas funcionales

“Las pruebas funcionales están centradas en comprobar que las funcionalidades descritas en el documento de requisitos del sistema se cumplen con la implementación realizada. A este tipo de pruebas también se les denomina pruebas de comportamiento o de caja negra, debido a que los analistas enfocan su atención a las respuestas del sistema de acuerdo a los datos de entrada y sus resultados en los datos de salida, los cuales se definen generalmente en los casos de prueba que se crean antes del inicio de las pruebas” (Oré B., 2009).

El proceso para ejecutar este tipo de pruebas es el siguiente (Serna M., 2011):

1. Analizar los requisitos y sus especificaciones.
2. Seleccionar entradas válidas y no válidas de acuerdo con las especificaciones.
3. Determinar las salidas esperadas para cada entrada.
4. Diseñar los casos de pruebas con las entradas seleccionadas.
5. Ejecutar los casos de prueba.
6. Comparar las salidas encontradas con las salidas esperadas.
7. Determinar si el funcionamiento del software en prueba es apropiado.

En el **Anexo 6** se muestra el diseño de casos de pruebas correspondiente a un escenario de la funcionalidad Gestionar necesidad. Los casos de pruebas restantes se encuentran en los documentos Diseño de Casos de Pruebas del portal web para la Dirección de Informatización.

3.4.2 Pruebas de carga

“Las pruebas de carga permiten la simulación del acceso de muchos usuarios a un servidor al mismo tiempo, posibilitando observar el comportamiento de una aplicación bajo una cantidad de peticiones esperadas. La carga puede ser el número de usuarios concurrentes que se espera que utilicen la aplicación, y que realizan durante el tiempo en que dura la carga un número específico de transacciones. Este tipo de prueba facilita la monitorización del servidor y la base de datos para obtener el cuello de botella en la aplicación, y puede mostrar los tiempos de respuesta de todas las transacciones importantes” (msdn.microsoft.com, 2013).

3.4.3 Pruebas de estrés

“Las pruebas de estrés posibilitan la obtención de datos sobre la carga del sistema. Tiene como objetivo generar cargas en el sistema hasta hacerlo inutilizable, para centrarse en verificar la calidad de los mensajes de error o establecer alertas para anticipar los fallos. Las pruebas de estrés son uno de los últimos tipos de pruebas que se deben efectuar, debido a que tienen un carácter poco realista ya que podría darse el caso que nunca se diera en la vida real la situación de carga simulada” (Fraile, 2011).

3.4.4 Pruebas de seguridad

“Las pruebas de seguridad permiten realizar una evaluación de los sistemas desde el punto de vista externo y sin conocimiento previo del mismo. Tienen como objetivo hacer un análisis con el fin de encontrar fallos de seguridad tanto en el diseño como en la implementación de la aplicación. Además buscan medir la confidencialidad, integridad y disponibilidad de los datos, partiendo de la identificación de amenazas y riesgos en el uso de interfaces de usuarios final. Una vez terminadas las pruebas es posible medir y cuantificar los riesgos a los cuales se ven expuestos aplicativos de la infraestructura interna y externa” (vyvquality.com, 2012).

3.4.5 Resultados de las pruebas

Resultados de las pruebas funcionales

Las pruebas funcionales que fueron aplicadas a la propuesta de solución se realizaron utilizando el documento Diseño de casos de pruebas, recogiendo los distintos escenarios que se correspondían a cada uno de los requisitos funcionales. Para la realización de las pruebas se utilizaron en la ejecución de las historias de usuario datos válidos e inválidos, eligiendo de forma correcta los valores de entrada, con el objetivo de abarcar la mayor cantidad posible de combinaciones, sin hacer que la cantidad de casos de prueba fuera muy elevada.

Para ejecutar las pruebas se realizaron tres iteraciones, en la primera se encontraron 15 no conformidades que fueron solucionadas en su totalidad; en la segunda iteración se detectaron 4 nuevas no conformidades a las que se les dio solución, para un tercera iteración en la que todas las no conformidades estaban resueltas.

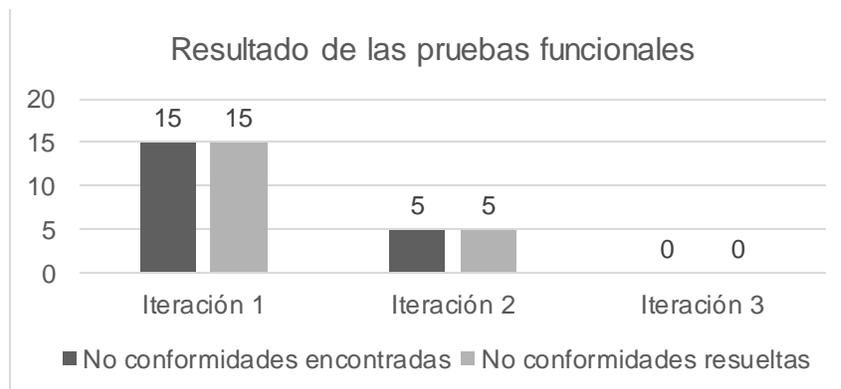


Figura 12. Resultado de las pruebas funcionales.

Resultados de las pruebas de carga y estrés

Para realizar las pruebas de carga y estrés se efectuó un análisis del tráfico de usuarios en los sistemas homólogos anteriormente estudiados, arrojando que las visitas a los portales corporativos se encontraban en un rango de 800 a 1000 visitas diarias. Debido a los datos obtenidos y a que en la Universidad se cuenta aproximadamente con 7000 usuarios, se realizaron las pruebas de carga y estrés en el caso crítico en que se conecten 100 usuarios concurrentes.

Las pruebas de carga y estrés se desarrollaron con la ayuda de la herramienta Apache JMeter, en un ambiente utilizando un ordenador con las siguientes características:

- Sistema operativo Linux Mint 17.

- Microprocesador Intel Dual Core a 2.30 GHz.
- Memoria RAM 4GB.



Figura 13. Resultados de las pruebas de carga y estrés.

Los resultados obtenidos en las pruebas de carga se consideran satisfactorios. La propuesta de solución generó una buena transferencia de datos para 100 usuarios concurrentes esperados, lo que incurrió en un rendimiento de 15,0 seg. Se demuestra que la propuesta de solución es estable, ya que se mantuvo prestando servicios todo el tiempo sin incurrir en fallos. Además se probó para un total de 50 usuarios concurrentes, obteniendo los resultados mostrados en la Figura 13.

Resultados de las pruebas de seguridad

Las pruebas de seguridad fueron aplicadas haciendo uso de la herramienta Nikto 2.1.4.

La primera iteración realizada con la herramienta Nikto arrojó un total de 3 elementos afectados para 3 no conformidades medias quedando evaluada de insatisfactoria la prueba. Las no conformidades encontradas fueron solucionadas en su totalidad por el equipo de desarrollo. En la segunda iteración la prueba fue evaluada de satisfactoria, debido a que no se encontraron no conformidades.

URI	/config.php
HTTP Method	GET
Description	/config.php: PHP Config file may contain database IDs and passwords.
Test Links	http://localhost:10000/config.php http://127.0.0.1:10000/config.php
OSVDB Entries	OSVDB-0
URI	/server-status
HTTP Method	GET
Description	/server-status: This reveals Apache information. Comment out appropriate line in httpd.conf or restrict access to allowed hosts.
Test Links	http://localhost:10000/server-status http://127.0.0.1:10000/server-status
OSVDB Entries	OSVDB-561
URI	/phpmyadmin/
HTTP Method	GET
Description	/phpmyadmin/: phpMyAdmin is for managing MySQL databases, and should be protected or limited to authorized hosts.
Test Links	http://localhost:10000/phpmyadmin/ http://127.0.0.1:10000/phpmyadmin/
OSVDB Entries	OSVDB-3092

Figura 14. Vulnerabilidades detectadas.

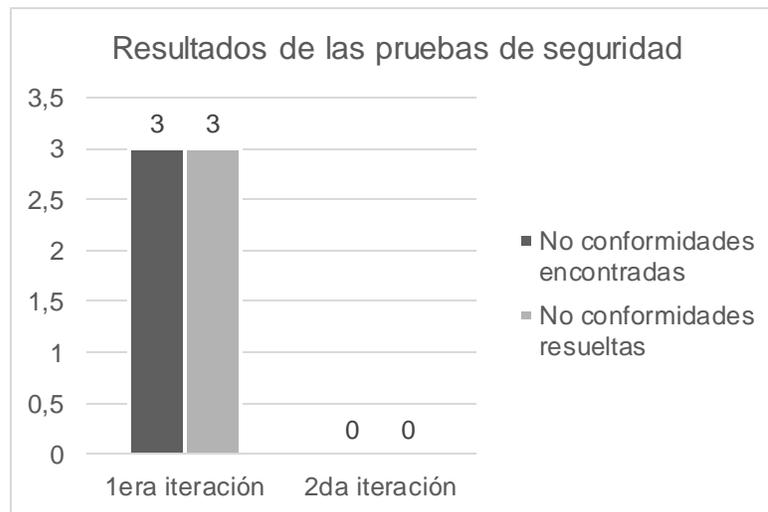


Figura 15. Resultados de las pruebas de seguridad.

3.5 Conclusiones parciales

En este capítulo se realizó una descripción de los estándares de codificación utilizados y se aplicaron las pruebas al sistema, concluyendo que:

La confección del diagrama de componentes ofreció una vista arquitectónica de alto nivel para ayudar al equipo de desarrollo en la implementación.

La definición de los estándares de codificación a tener en cuenta para la implementación del portal web, permitió garantizar que el código posea alta calidad, menos errores y que pueda ser mantenido fácilmente, así como reutilizado por desarrolladores de otros proyectos que lo necesiten.

La realización de las pruebas de software al portal web facilitó la identificación de errores en la solución para su rápida solución, posibilitando determinar y asegurar la calidad del portal web para la Dirección de Informatización.

CONCLUSIONES GENERALES

Con la culminación del presente trabajo se dio solución a la problemática planteada, contribuyendo a una mejor gestión de la información para el proceso de informatización, arribando a las siguientes conclusiones:

El estudio del estado del arte, permitió definir que la tendencia de portal corporativo era la que se ajustaba al pedido del cliente y permitió identificar las tecnologías y herramientas a utilizar.

La correcta selección de lenguajes, tecnologías y herramientas garantizó el desarrollo del portal web para la DIN, satisfaciendo las necesidades del cliente.

La realización de un estudio sobre un conjunto de portales web que siguen la tendencia de portal corporativo permitió determinar cómo quedaría distribuida y estructurada la información en el portal para un mejor diseño y navegabilidad del mismo.

Las técnicas de obtención de información utilizadas permitieron definir los requisitos funcionales y no funcionales del sistema, dando paso a una mejor comprensión, por parte de la autora, de los resultados a obtener de una manera precisa y sirvieron de guía para la implementación del sistema.

Las pruebas de software funcionales, de carga, de estrés y de seguridad aplicadas, permitieron determinar satisfactoriamente la calidad de la solución propuesta.

RECOMENDACIONES

Para futuras versiones del portal web para la Dirección de Informatización se recomienda:

- Lograr la integración con sistemas de la Universidad como el Sistema de Gestión de Incidencias y el Sistema de Gestión de Proyectos, para la generación de reportes y estadísticas de importancia para la comunidad universitaria y para lograr una gestión más eficiente de las incidencias.
- Utilizar el Servicio de Autenticación Central para la autenticación de los usuarios en el sistema.
- Incrementar el número de funcionalidades para lograr una mayor atención de los usuarios y que utilicen mucho más el portal web.
- Aplicar otras técnicas que tributen a la usabilidad y accesibilidad del portal web, para lograr un mayor perfeccionamiento del diseño del mismo.

BIBLIOGRAFÍA

- academia.edu. 2015.** academia.edu. [En línea] 2015. [Citado el: 3 de Mayo de 2015.] http://www.academia.edu/8497289/Tecnicas_para_recopilar_informacion.
- AOL.com. 2015.** AOL.com. [En línea] 2015. [Citado el: 11 de Noviembre de 2015.] <http://www.aol.com/>.
- APETI. 2015.** Portal de la APETI. [En línea] 2015. <http://www.apeti.com/>.
- Auger, Robert. 2010.** Cgisecurity.com. *Cgisecurity.com*. [En línea] 4 de Abril de 2010. [Citado el: 5 de Diciembre de 2015.] <http://www.cgisecurity.com/csrf-faq.html>.
- Beck, K. 1999.** *Extreme Programming Explained. Embrace Change*. [trad.] Addison Wesley. s.l. : Pearson Education, 1999.
- Brinck, T., Gergle, D. y Wood, S. 2002.** *Designing web sites that work, usability for the web*. 2002.
- Bustelo, Carlota y Amarilla, Raquel. 2001.** *Gestión del conocimiento y gestión de la información*. 2001.
- Canales, M. C. 2007.** *Maestros de la web. Web 2.0 ¿Reconfiguración social o tecnológica?* 2007. Vol. 2012.
- Canós, José H, Letelier, Patricio y Penadés, María Carmen. 2003.** *Métodologías Ágiles en el Desarrollo de Software*. DSIC -Universidad Politécnica de Valencia : s.n., 2003.
- Centro de Soporte. 2015.** Portal del Centro de Soporte. [En línea] 2015. <http://soporte.uci.cu>.
- CIRT.net. 2015.** CIRT.net. [En línea] 2015. [Citado el: 20 de Mayo de 2015.] <https://cirt.net/Nikto2>.
- CMSMatrix. 2015.** CMSMatrix. [En línea] 2015. [Citado el: 20 de Mayo de 2015.] <http://www.cmsmatrix.org/matrix/cms-matrix/>.
- Cochran, David. 2012.** *Twitter Bootstrap Web Development*. 1st edición. s.l. : Packt Publishing, 2012. pág. 100. ISBN 978-1849518826.
- Cubagob. 2015.** Cubagob. [En línea] 2015. <http://www.cubagob.cu/>.
- Curto, Josep. 2006.** *Gestión de la información. Information Management*. 2006.
- Davis, G. y Olsón. 1985.** *Management Information Systems: Conceptual foundations, Structure and Development*. 2da. Nueva York : Universidad de Antioquia, 1985.

- De Luca, Damián. 2010.** CSS3 HTML5. [En línea] 23 de Noviembre de 2010. [Citado el: 22 de Mayo de 2015.] <http://html5.dwebapps.com/que-es-css3/>.
- Días, Cláudia Augusto. 2001.** Portal corporativo: conceitos e características. s.l. : Ciência da Informação, 2001, Vol. 30, págs. 50-60.
- Dirección de Formación Postgraduada. 2015.** Portal de la Dirección de Formación Postgraduada. [En línea] 2015. <http://postgrado.uci.cu>.
- Dirección de Investigaciones. 2015.** Portal de la Dirección de Investigaciones. [En línea] 2015. <http://investigaciones.uci.cu>.
- Dirección de Redes y Servicios Telemáticos. 2015.** Portal de la Dirección de Redes y Servicios Telemáticos. [En línea] 2015. <http://drst.uci.cu>.
- Dirección de Seguridad Informática. 2015.** Portal de la Dirección de Seguridad Informática. [En línea] 2015. <http://seguridad.uci.cu>.
- Eguiluz, Javier. 2007.** *Introducción a JavaScript*. 2007.
- . **2009.** symfony.es. [En línea] 2009. [Citado el: 1 de Mayo de 2015.] <http://symfony.es/noticias/2009/10/29/zend-framework-se-rinde-ante-doctrine/>.
- Endler, David. 2002.** Cgsecurity.com. *Cgsecurity.com*. [En línea] 2002. [Citado el: 3 de Mayo de 2015.] <http://www.cgsecurity.com/xss-faq.html>.
- EVA. 2015.** EVA. [En línea] 2015. [Citado el: 12 de Febrero de 2015.] <http://eva.uci.cu/course/view.php?id=193>.
- Fogie, Seth. 2007.** Cross Site Scripting Attacks: Xss Exploits and Defense. [En línea] 2007.
- Fraile, Luis. 2011.** [En línea] 2011. [Citado el: 8 de Abril de 2015.] <http://www.globetesting.com/pruebas-de-rendimiento>.
- Gamma, Erich, y otros. 1995.** *Design Patterns: Elements of Reusable Object-Oriented Software*. 1995.
- García Gómez, Juan Carlos. 2012.** [En línea] 2012. [Citado el: 6 de Febrero de 2015.] <http://aprendeonline.udea.edu.co>.

- Gitonga, David. 2014.** eHow en Español. [En línea] 2014. [Citado el: 1 de Marzo de 2015.] http://www.ehowenespanol.com/navegadores-soportan-css3-info_539163/.
- Hall, Prentice y Larman, Craig. 2003.** [En línea] 2003. [Citado el: 1 de Marzo de 2015.] <http://is.ls.fi.upm.es/docencia/is2/documentacion/ModeloDominio.pdf>.
- Hernández León, Rolando Alfredo y Coello Gonz, Sayda. 2011.** *El proceso de investigación científica*. Ciudad de La Habana : Editorial Universitaria del Ministerio de Educación Superior, 2011. SBN 978-959-16-1307-3.
- Hernández Pérez, Julio Antonio, Ordoñez Leyva, Yoanni y Aviles Vazquez, Ernesto. 2013.** *Gestión de incidencias de inventarios de red*. Universidad de las Ciencias Informáticas : s.n., 2013.
- infonomia.com. 2015.** infonomia.com | Building innovation teams. [En línea] 2015. [Citado el: 12 de Diciembre de 2015.] <http://www.infonomia.com/>.
- Instituto Tecnológico de Sonora. 2015.** biblioteca.itson. [En línea] 2015. [Citado el: 2 de Mayo de 2015.] http://biblioteca.itson.mx/oa/dip_ago/introduccion_sistemas/p3.htm.
- INTEF. 2015.** [En línea] 2015. [Citado el: 28 de Mayo de 2015.] http://www.ite.educacion.es/formacion/materiales/155/cd/modulo_1_Iniciacionblog/principios_generales_de_la_web_20.html.
- ireo.com. 2015.** ireo.com. [En línea] 2 de Febrero de 2015. <http://www.ireo.com/fabricantes-y-productos/manageengine/servicedesk-plus/resumen/>.
- ISO 9241. 1998.** *Ergonomic requirements for visual display terminals. Guidance for usability*. 1998.
- Kioskea.net. 2015.** Kioskea.net. [En línea] Mayo de 2015. <http://es.kioskea.net/contents/279-protocolos-de-mensajeria-smtp-pop3-e-imap4>.
- . 2015.** Kioskea.net. [En línea] Mayo de 2015. [Citado el: 12 de Mayo de 2015.] <http://es.kioskea.net/contents/282-tcp-ip>.
- Komelsat. 2015.** Komelsat Inversiones & Gestión de Activos. [En línea] 2015. <http://www.komelsat.com/>.
- Kotonya, G. y Sommerville, I. 2000.** *Requirements Engineering: Processes and Techniques*. s.l. : John Wiley & Sons, 2000. págs. 2-3.

Larman, Craig. 1999. *UML y Patrones*. México : s.n., 1999.

Mancomunidad Sierra Occidental Huelva. 2015. Mancomunidad Sierra Occidental Huelva. [En línea] 2015. <http://www.sierrahuelvaocc.com/>.

Marqués Graells, P. 2002. *La Información y el conocimiento*. Departamento de Pedagogía Aplicada, Facultad de Educación, UAB : s.n., 2002.

Martínez, R. 2010. [En línea] 2010. [Citado el: 14 de diciembre de 2015.] http://www.postgresql.org.es/sobre_postgresql.

Menéndez-Barzanallana, Rafael. 2012. *Historia del desarrollo de aplicaciones Web*. Universidad de Murcia : s.n., 2012.

Microsoft. 2015. Microsoft. [En línea] 2015. [Citado el: 2 de Mayo de 2015.] <https://msdn.microsoft.com/en-us/library/aa767735%28VS.85%29.aspx>.

—. **2015.** Microsoft – Official Home Page. *Microsoft – Official Home Page*. [En línea] 2015. [Citado el: 1 de Diciembre de 2015.] <http://www.microsoft.com/en-us/default.aspx>.

Monzón, A y Moynard, M. 2010. *Trabajo de diploma para optar por el título de ingeniero en Ciencias Informáticas Desarrollo de Portal Web de la Federación Cubana de Voleibol*. Universidad de las Ciencias Informáticas, La Habana. : s.n., 2010.

Mora, Sergio. 2002. *Programación de aplicaciones web: Historia, principios básicos y clientes web*. s.l. : Editorial Club Universitario, 2002.

Morville, Peter y Rosenfeld, Louis . 1998. *Information Architecture for the World Wide Web*. s.l. : O'Reilly Media, 1998.

msdn.microsoft.com. 2013. msdn.microsoft.com. [En línea] 2013. [Citado el: 8 de Abril de 2015.] <http://msdn.microsoft.com/es-es/library/ms182562%28v=vs.80%29.aspx>.

Nginx Community. 2015. Nginx. [En línea] 16 de Abril de 2015. [Citado el: 27 de Abril de 2015.] <http://wiki.nginx.org/Main>.

Nielsen, J. 1993. *Usability engineering*. 1993.

O'Reilly, T. 2004. *¿Qué es Web 2.0?* San Francisco : O'Reilly Media, 2004.

- Oliveros, Alejandro y del Valle Rojo, Silvana. 2012.** Requerimientos No funcionales para aplicaciones Web. [En línea] 2012. [Citado el: 5 de marzo de 2015.] http://www.41jaiio.org.ar/sites/default/files/432_ASSE_2012.pdf. ISSN: 1850-2792..
- Oracle. 2015.** Oracle España | Hardware and Software, Engineered to Work Together. *Oracle España | Hardware and Software, Engineered to Work Together*. [En línea] 2015. [Citado el: 14 de Diciembre de 2015.] <http://www.oracle.com/es/index.html>.
- Oré B., Ing. Alexander. 2009.** [En línea] 2009. [Citado el: 1 de Marzo de 2015.] http://www.calidadyssoftware.com/testing/pruebas_funcionales.php.
- otrs.com. 2015.** otrs.com. [En línea] 2015. <https://www.otrs.com/>.
- Peñafiel, M. 2013.** *Aplicaciones Web. Tesis para optar por el título de Ingeniero en Sistemas*. Facultad de Informática y Electrónica, Chimborazo : s.n., 2013.
- Peñalver Romero, G.M. 2008.** “*Trabajo de diploma: Metodología ágil para proyectos de software libre*”. Ciudad de La Habana, Universidad de las Ciencias Informáticas. : s.n., 2008.
- Peralta, Manuel. 2008.** *Sistema de Información*. 2008.
- Pérez González, Rodrigo y David Rodríguez, Aureliano . 2008.** [En línea] 2008. [Citado el: 10 de febrero de 2015.]
- Piattini Velthuis, Mario G. 1996.** *Análisis y Diseño Detallado de Aplicaciones Informáticas de Gestión Rama*. Madrid : RA-MA EDITORIAL, 1996.
- Piñero Pérez, P. Y, y otros. 2013.** *GESPRO. Paquete para la gestión de proyectos*. s.l. : Revista Nueva Empresa, 2013. págs. p. 45-53.
- Ponjuán, G. 2004.** *Gestión de la Información: dimensiones e implementación para el éxito organizacional*. Rosario : Ediciones Nuevo Paradigma, 2004.
- Porta García, Saily y Berrio Aguilar, Yudermis Caridad. 2007.** *Portal de la Dirección de Informatización de la UCI*. La Habana, Universidad de las Ciencias Informáticas : s.n., 2007.
- PostgreSQL-es. 2010.** PostgreSQL-es. [En línea] 2 de Octubre de 2010 . [Citado el: 11 de Abril de 2015.] http://www.postgresql.org.es/sobre_postgresql.

- Potencier, Fabien y Zaninotto, François. 2009.** *The Definitive Guide to symfony*. s.l. : Apress, 2009. ISBN 1-59059-786-9.
- Potencier, Fabien. 2009.** *Practical symfony*. s.l. : Sensio Labs Books. Doctrine edition, ISBN 978-2-918390-06-0, 2009.
- Pressman, Roger. 2002.** *Ingeniería del Software. Un enfoque práctico*. 5ta Edición. McGraw-Hill : s.n., 2002. ISBN 84-481-3214-9.
- Reese, Will. 2008.** Linux Journal. [En línea] 1 de septiembre de 2008. [Citado el: 7 de mayo de 2015.] <http://www.linuxjournal.com/article/10108>.
- Riehle, Dirk . 2000.** *Framework Design. A Role Modeling Approach*. Universität Hamburg : s.n., 2000.
- Rodríguez Gairín, JM. 2004.** *PHP-Nuke: software de código abierto en la gestión de contenidos Web. El profesional de la Información*. 2004. págs. 1(6):458-66.
- Rodríguez Salas, Karla. 2002.** *GESTIÓN DE LA INFORMACIÓN EN LAS ORGANIZACIONES*. 2002.
- Rodríguez Villar, Malay. 2007.** *Trabajo de Diploma. Introducción de procedimientos ágiles en la producción de software en la Facultad 7 de la Universidad de las Ciencias Informáticas*. La Habana, Universidad de las Ciencias Informáticas : s.n., 2007.
- Rodríguez, Luis Miguel. 2013.** ifimedia. [En línea] 11 de Septiembre de 2013. [Citado el: 30 de Mayo de 2015.] <http://www.ifimedia.com/cms-joomla-wordpress-vs-framework-laravel-django-ruby-on-rails>.
- Rosson, M. B. y Carroll, J. 2002.** *Usability engineering*. 2002.
- Sánchez Méndez, Ing. Alelí. 2013.** *Proceso de desarrollo de software*. Departamento de Desarrollo, Dirección de Informatización de la Universidad de las Ciencias Informáticas. La Habana : s.n., 2013.
- Sánchez, I.B. 2014.** *Aplicación de soporte a la metodología de desarrollo de distribuciones GNU/ LINUX, Nova-OpenUp. Trabajo de diploma para optar por el título de ingeniero en Ciencias Informáticas*. Universidad de las Ciencias Informáticas, La Habana : s.n., 2014.
- Sandbox Networks. 2015.** Parenting Advice, Activities for Children, Family Games. *Parenting Advice, Activities for Children, Family Games*. [En línea] 2015. [Citado el: 12 de Diciembre de 2015.] <http://www.familyeducation.com/>.

- Sears, Connors y Rebert, Chris. 2015.** Bootstrap The world's most popular mobile-first and responsive front-end framework. *Bootstrap The world's most popular mobile-first and responsive front-end framework*. [En línea] 2015. [Citado el: 24 de Marzo de 2015.] <http://getbootstrap.com/>.
- SensioLabs. 2015.** symfony.com. [En línea] 2015. [Citado el: 2 de Mayo de 2015.] <http://symfony.com/projects>.
- . **2007.** symfony.com. [En línea] 2007. [Citado el: 31 de Mayo de 2015.] http://symfony.com/legacy/doc/book/1_0/en/02-Exploring-Symfony-s-Code.
- Serna M., Edgar. 2011.** [En línea] 2011. [Citado el: 7 de Abril de 2015.] http://www.bdigital.unal.edu.co/6114/1/71617999._2012.pdf.
- SocialCompare. 2015.** SocialCompare. [En línea] 20 de Mayo de 2015. [Citado el: 24 de Mayo de 2015.] <http://socialcompare.com/es/comparison/php-frameworks-comparison>.
- Sommerville, Ian. 2002.** *Ingeniería de Software*. 6ta Edición. s.l. : Prentice-Hall, 2002. ISBN 970-26-0206-8.
- Soto López, Nilet. 2013.** *Ejecución de un servicio del Plan de Informatización*. Universidad de las Ciencias Informáticas, La Habana : s.n., 2013. Manual de procedimientos.
- Terra. 2015.** Terra España - Noticias, deportes, música, moda y más. [En línea] 2015. [Citado el: 11 de Noviembre de 2015.] <http://www.terra.es>.
- The Apache Software Foundation. 2015.** apache.org. [En línea] 2015. [Citado el: 3 de Mayo de 2015.] <http://jmeter.apache.org/>.
- The jQuery Foundation. 2015.** jQuery. *jQuery*. [En línea] The jQuery Foundation, 2015. [Citado el: 23 de Marzo de 2015.] <https://jquery.com/>.
- The PHP Group. 2001 - 2015.** Manual de PHP. [En línea] 2001 - 2015. [Citado el: 22 de Mayo de 2015.] <https://php.net/manual/es/intro-what-is.php>.
- TIC. 2015.** Tecnología de la Información Comunicación (TIC) Pontificia Universidad Católica de Ecuador. [En línea] 2015. <http://www.puce.edu.ec>.

- toronto.com. 2015.** Toronto Entertainment News City of Toronto Downtown Events Toronto Star. *Toronto Entertainment News City of Toronto Downtown Events Toronto Star*. [En línea] 2015. [Citado el: 13 de Diciembre de 2015.] <http://www.toronto.com/>.
- tubitacora.com. 2015.** tubitacora.com. [En línea] 2015. [Citado el: 11 de Diciembre de 2015.] <http://www.tubitacora.com/>.
- UNAD. 2015.** UNAD(Universidad Nacional Abierta y a Distancia). [En línea] 2015. [Citado el: 2 de Diciembre de 2015.] http://datateca.unad.edu.co/contenidos/107010/107010/tcnicas_para_recopilar_informacin.html.
- Van Der Henst S, Christian. 2005.** Maestros del web. [En línea] 27 de Octubre de 2005. [Citado el: 5 de Mayo de 2015.] <http://www.maestrosdelweb.com/web2/>.
- Visual Paradigm International. 2015.** Herramientas de diseño de software para equipos ágiles con UML, BPMN y más. *Software Design Tools for Agile Teams, with UML, BPMN and More*. [En línea] 2015. [Citado el: 20 de Marzo de 2015.] http://www.visual-paradigm.com/support/documents/vpuserguide/12/13/5963_visualparadi.html.
- vyvquality.com. 2012.** vyvquality.com. [En línea] 2012. [Citado el: 7 de Abril de 2015.] <http://www.vyvquality.com/w1/index.php/servicios/pruebas-de-seguridad.html>. ISO 9001: 2008.
- W3C. 2015.** Sitio oficial de la W3C. [En línea] 2015. [Citado el: 17 de Febrero de 2015.] www.w3c.org.
- . 2015.** W3C. [En línea] 2015. [Citado el: 12 de Mayo de 2015.] <http://www.w3.org/html/>.
- Welling, Luke y Thomson, Laura. 2005.** *Desarrollo web con php y mysql php 5 y mysql 4.1 y 5*. Madrid : s.n., 2005.