

**Universidad de las Ciencias Informáticas**

**FACULTAD 6**



**Título: “Primicia móvil V.1.0”**

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

**Autor:** Patricia Herrera Palacios

**Tutor:** Ing. Rafael Lorente Miranda

**La Habana, Mayo de 2013“**

**Año 55 de la Revolución”**

## DECLARACIÓN DE AUTORÍA

Declaro ser la única autora de este trabajo titulado: PRIMICIA Móvil y autorizo al proyecto PRIMICIA, de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio. Para que así conste firmo la presente a los 10 días del mes de Junio del año 2013.

Para que así conste firmo la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

Patricia Herrera Palacio

Ing. Rafael Lorente Miranda

\_\_\_\_\_

\_\_\_\_\_

Firma del Autor

Firma del Tutor

## DATOS DE CONTACTO

Tutor: Ing. Rafael Miranda Lorente.

Categoría docente: Adiestrado.

Centro de trabajo: Universidad de las Ciencias Informáticas.

Título de la especialidad de graduado: Ingeniero en Ciencias Informáticas.

Año de graduación: 2009

Institución en la que se graduó: Universidad de las Ciencias Informáticas.

Correo electrónico: rlmiranda@uci.cu

## AGRADECIMIENTOS

*A mis padres Olguita y Reynaldo, por darme la dicha de existir y ser la persona más feliz de este mundo al poder contar con ellos. A mi hermano Rey, que lo adoro. A mis abuelas, por quererme tanto, especialmente a mi abuela Olga. A mis abuelos, aunque no estén físicamente, siempre los tengo presentes. A mi padrastro Andrés, es como mi segundo papá, siempre ha estado cuando lo he necesitado. A mis tíos y tías que me han servido de ejemplo. A todos mis primos. A mis amigas de 5to año (Isis, Arianna, Dolores, Yanet, Mahelis, Lisandra, Thaimi, etc....) y a mi amigos (Asiel, Karel, Eddy, Maxwell, etc....) que me han soportado y ayudado tanto tiempo. A todas la amistades que logré en la universidad. A mi tutor y a Dayami por su colaboración. A Erick, mi churry, el amor de mi vida, por darme su apoyo, por estar en todo momento, por hacerme tan feliz y confiar en mí.*

## DEDICATORIA

*Dedico este trabajo especialmente a las personas que más quiero; a mis padres, a mi hermano, a mi abuela, a mi novio y toda mi familia, los cuáles significan todo en mi vida.*

## RESUMEN

En Cuba en los últimos años, ha existido un auge en el uso de la tecnología móvil como medio de comunicación. La cual además de posibilitar la comunicación, adiciona gran variedad de aplicaciones similares a las de un ordenador, lo cual posibilita una mayor portabilidad de los sistemas informáticos en general.

PRIMICIA es una plataforma televisiva creada en Cuba, específicamente en la Universidad de las Ciencias Informáticas, la cual permite la creación y publicación de noticias que son transmitidas por un canal de televisión. Debido a los constantes cambios inmediatos en la programación del canal, los usuarios del sistema sintieron la necesidad de ejecutar algunas de las funcionalidades en dispositivos móviles. Para darle solución a esta problemática se realizó un estudio en las funcionalidades actuales de PRIMICIA, definiendo incluir los procesos de publicación de noticias y gestión de señales en una primera versión móvil de la plataforma.

Con el desarrollo de este trabajo de diploma titulado “PRIMICIA Móvil 1.0” se logró implementar una solución informática que realiza el proceso de publicación de noticias y gestión de señales de la plataforma desde dispositivos móviles. La presente investigación recoge las etapas de desarrollo del sistema, las cuales van desde la fundamentación teórica, la propuesta y modelado de la solución hasta llegar a la implementación y validación de la aplicación. El resultado final del trabajo es una aplicación web móvil, que sirve para realizar las funcionalidades referentes al proceso de publicación de noticias y gestión de señales de PRIMICIA.

**Palabras claves:** aplicación web móvil, PRIMICIA, tecnología móvil.

## ÍNDICE

AGRADECIMIENTOS .....	I
DEDICATORIA .....	II
RESUMEN .....	III
ÍNDICE IV	
ÍNDICE DE FIGURAS .....	V
ÍNDICE DE TABLAS .....	IX
INTRODUCCIÓN.....	1
capítulo 1: Fundamentación Teórica.....	5
Introducción .....	5
1.1. Conceptos asociados al dominio del problema.....	5
1.2. Análisis del objeto de estudio .....	7
1.2.1 Tecnologías Móviles .....	7
1.2.2 Protocolos de comunicación móvil .....	8
1.2.3 Navegadores móviles .....	9
1.2.4 Aplicaciones móviles.....	10
1.3. Análisis de soluciones existentes .....	10
1.3.1 Ámbito nacional .....	11
1.3.2 Ámbito internacional.....	11
1.4. Selección de la metodología de desarrollo.....	12
1.5 Lenguaje de modelado.....	13
1.5.1 Lenguaje Unificado de Modelado (UML) .....	13
1.6. Herramienta CASE .....	14
1.6.1 Visual Paradigm 6.4 .....	14
1.7. Lenguaje de programación .....	15

1.8. Gestor de base de datos .....	16
1.9 Framework de desarrollo .....	17
1.10. Entorno de desarrollo.....	18
Conclusiones parciales .....	19
Capítulo 2: Características del Sistema: PRIMICIA MÓVIL .....	20
Introducción .....	20
2.1 Modelo de Dominio .....	20
2.2.1 Conceptos o eventos fundamentales del dominio.....	20
2.2.2 Diagrama del modelo de dominio .....	21
2.3 Especificación de requisitos de software.....	21
2.3.1 Requisitos funcionales .....	22
2.3.2 Requisitos no funcionales .....	23
2.4 Modelos de casos de usos del sistema.....	24
2.4.1 Definición de actores.....	24
2.4.2 Diagrama de casos de uso .....	25
2.4.3 Especificación de casos de usos.....	25
Conclusiones parciales .....	30
Capítulo 3: Análisis y Diseño .....	31
Introducción .....	31
3.1 Modelo de Análisis .....	31
3.2 Descripción de la Arquitectura .....	32
3.2.1 Patrón Arquitectónico.....	32
3.4 Modelo de Diseño .....	35
3.4.1 Diagrama de clases del diseño.....	36
3.5 Modelo de Datos .....	36
3.5.1 Diagrama de Clases Persistentes.....	37



3.5.2 Diagrama Entidad-Relación .....	38
Conclusiones parciales.....	38
Capítulo 4: Implementación y Prueba .....	39
Introducción.....	39
4.1 Modelo de Despliegue .....	39
4.1.1 Diagrama de Despliegue .....	39
4.2 Modelo de Implementación .....	40
4.2.1 Diagrama de Componentes.....	41
4.3 Pruebas del Sistema.....	43
4.4 Pruebas unitarias .....	43
4.5 Prueba funcionales .....	47
4.6 Resultado de las pruebas .....	49
Conclusiones parciales.....	50
CONCLUSIONES GENERALES .....	51
RECOMENDACIONES .....	52
TRABAJOS CITADOS.....	53
BIBLIOGRAFÍA .....	55
Anexos .....	57
GLOSARIO .....	58

## ÍNDICE DE FIGURAS

Figura 1: Funcionamiento de WAP.....	8
Figura 2: Diagrama del dominio. ....	21
Figura 3 : Diagrama de CU del sistema .....	25
Figura 4: Diagrama de clases del análisis CU Pulicar Noticia.....	32
Figura 5: Diagrama de clases del diseño del CUNoticia.....	36
Figura 6: Diagrama de clases persistentes.....	37
Figura 7: Diagrama Entidad-Relación .....	38
Figura 8 : Diagrama de despliegue .....	40
Figura 9 : Diagrama de componentes del CU Publicar noticia .....	42

## ÍNDICE DE TABLAS

Tabla1: Requerimiento de hardware .....	24
Tabla 2: Actor del sistema .....	24
Tabla 3: CU 1. Publicar noticia .....	25
Tabla 4 : Caminos básicos del flujo.....	46
Tabla 5 : Caso de prueba del CU Publicar Noticia. ....	48

# INTRODUCCIÓN

Hoy en día los medios de difusión juegan un papel fundamental en la vida del hombre, mediante estos se rompen barreras que permiten a las personas informarse de manera inmediata de los sucesos ocurridos tanto en su campo de acción, así como los ocurridos a miles de kilómetros. En la actualidad los medios más difundidos son la televisión, el Internet, la radio, la prensa escrita, la prensa digital y la telefonía móvil.

Es una realidad que la mayoría de los medios mencionados con anterioridad tengan dos características comunes; la primera se refiere a la presentación de las noticias en distintos formatos, siendo los más importantes el audio, el video, el texto, la imagen, la gráfica y la tabla. La segunda característica se refiere al uso de sistemas informáticos que automatizan los procesos de búsqueda, confección, y transmisión de las noticias difundidas.

Particularmente en el medio televisivo, existen diversos sistemas que posibilitan la producción de noticias, dentro de ellos se destacan ENewsRoom en España, que es un sistema de producción de noticias y contenidos audiovisuales; Sonaps en EE UU, que es un entorno en red que integra todos los aspectos del proceso de producción de informativos, desde la planificación, captación y edición, hasta la emisión, distribución y archivo; y la Plataforma de Televisión Informativa PRIMICIA. Esta última, es un producto informático cubano que ha venido desarrollándose desde el año 2005 en la Universidad de Ciencias Informáticas (UCI).

PRIMICIA constituye una solución integral capaz de proveer un canal de televisión para la transmisión automática y constante de informaciones en distintos formatos (textos, imágenes, sonidos y vídeos). Se encuentra estructurado en dos subsistemas; Administración y Transmisión, los cuales se relacionan entre sí para brindar un resultado eficiente y acorde a las necesidades de los usuarios. El Subsistema de Administración, realiza la administración del canal y toda la gestión de las informaciones y recursos multimedia que se transmiten. Mientras que el Subsistema de Transmisión es el encargado de visualizar las noticias y materiales que son publicados en la plataforma, está constituido por un único módulo para la transmisión del canal las 24 horas al día.

Es una realidad el hecho de que en Cuba en los últimos años, ha existido un gran auge del uso de la tecnología móvil como medio de comunicación. La cual ha alcanzado tal nivel que además de posibilitar la comunicación, adiciona gran variedad de aplicaciones similares a las de un ordenador. Entre ellas cabe destacar la consulta y manejo de aplicaciones desde clientes móviles con una interacción más simple y práctica con los sistemas, logrando de esta forma aumentar la portabilidad de los mismos, permitiendo además que desde diferentes geografías se pueda interactuar con una gran

variedad de funcionalidades.

Los usuarios de PRIMICIA no han estado ajenos al desarrollo y auge de la telefonía celular en Cuba, han sentido la necesidad de realizar desde sus dispositivos móviles varias operaciones que realizan a diario con el objetivo de administrar la publicación de las noticias en el canal. Estas funcionalidades se encuentran mayormente enmarcadas en el módulo de Publicación de noticias de la plataforma.

El módulo de gestión de noticias del subsistema de administración, realiza la gestión y administración de señales, además de permitir la publicación de las noticias ya editadas. En este módulo existen funcionalidades que por constante actualización se hace necesario que se integren al sistema, como son: gestionar las señales; hacer los cambios inmediatos de señal, gestionar la programación del canal, gestionar los bloques y asignarle a estos las noticias listas a publicar.

Luego de analizadas las dificultades existentes se define el **siguiente problema a resolver**: La publicación de las noticias elaboradas en la Plataforma de Televisión Informativa PRIMICIA, no se puede realizar desde dispositivos móviles. Por lo que se establece como **objeto de estudio**: las tecnologías para el desarrollo de aplicaciones móviles, enmarcado en el **campo de acción**: la publicación de noticias de la Plataforma de Televisión Informativa PRIMICIA.

Con el propósito de dar solución al problema anteriormente planteado, se ha trazado como **objetivo general** de la investigación, implementar un sistema que permita llevar a cabo el proceso de publicación de noticias haciendo uso de tecnologías móviles para la Plataforma de Televisión Informativa PRIMICIA. Se propone como **Idea a defender**: que si se logra implementar una aplicación móvil para la publicación de noticias, se garantizará la portabilidad del manejo y acceso a las principales funcionalidades de la Plataforma PRIMICIA.

Para dar cumplimiento al objetivo planteado, se han definido las siguientes **tareas de investigación**:

- Caracterización del uso de Tecnologías Móviles.
- Caracterización de las plataformas de desarrollo para móviles y seleccionar la metodología de desarrollo.
- Caracterización del proceso de publicación de noticias en el subsistema Administración de Noticias de la Plataforma de Televisión Informativa PRIMICIA.
- Diseño del sistema mediante el uso de una herramienta de modelado.
- Implementación del sistema diseñado.
- Validación de la solución mediante el método de prueba que más se adecue a la solución.

En el transcurso y desarrollo de la investigación se tendrán en cuenta una serie de métodos científicos para la obtención, procesamiento y llegada a conclusiones, los cuales se exponen a continuación:

### **Métodos Teóricos:**

- **Analítico-Sintético:** permitió analizar documentos y toda la bibliografía existente sobre tecnologías móviles, permitiendo además sintetizar ésta información.
- **Histórico-Lógico:** se utilizó para investigar lo relacionado a la información con que se cuenta hasta el momento del estado del arte, los conceptos fundamentales asociados a la factibilidad económica, así como otros sistemas informáticos encargados del estudio de la misma.

### **Métodos empíricos**

- **Entrevista:** Se utiliza para obtener información sobre el funcionamiento actual de los procesos transmisión y administración en PRIMICIA. Tomando como personal a entrevistar profesores y estudiantes que trabajan en el proyecto, específicamente con este subsistema. Para ello se toma una población de 6 individuos, de los cuales se toma como muestra 3 para ser entrevistados. Se realiza una técnica de muestreo no probabilístico, específicamente muestreo intencional. (Ver [Anexo 1](#))

El presente documento está constituido por los siguientes capítulos:

#### **Capítulo 1: Fundamentación Teórica.**

Constituye la base teórica de la investigación, en la cual son expuestos los principales conceptos que contribuyen al mejor entendimiento del problema en cuestión. Se describen los principales aspectos de las herramientas, tecnologías y metodologías a emplear durante el desarrollo de la misma.

#### **Capítulo 2: Características del Sistema**

En este capítulo se aborda acerca del negocio en que está enmarcada la investigación. Se identifican los casos de usos, actores y trabajadores del negocio. Se muestran los diagramas de dominio y se realiza un levantamiento de los requisitos funcionales y no funcionales del sistema con el fin de lograr los objetivos planteados. Además se realizar un análisis de las características específicas del sistema entre las que podemos encontrar los casos de usos, actores y trabajadores del sistema.

#### **Capítulo 3: Análisis y diseño del sistema.**

En este capítulo se hace una descripción de la arquitectura del sistema, los patrones de diseño utilizados. Se realizarán los diagramas de clases de diseño, los diagramas de interacción y el modelo de datos.

#### **Capítulo 4: Implementación y prueba.**

En este capítulo se realiza el modelo de implementación, el cual está compuesto por su respectivo diagrama de componentes y su diagrama de despliegue. Además se realiza las pruebas al sistema

para comprobar que cumple con los requerimientos funcionales y que presenta una óptima calidad

# **CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA**

## **Introducción**

En este capítulo se incluyen todos los aspectos teóricos que soportan la investigación, se brindan para ello un grupo de conceptos básicos necesarios para entender la propuesta del sistema. Se caracteriza el sistema de publicación de noticias de PRIMICIA y se informa el estado actual del proceso. Se detallan las tecnologías y herramientas seleccionadas, plataformas, modelos de desarrollo y lenguaje a utilizar.

### **1.1. Conceptos asociados al dominio del problema**

#### **Tecnología**

Tecnología según la RAE es el conjunto de conocimientos técnicos, ordenados científicamente, que permiten diseñar, crear bienes y servicios que facilitan la adaptación al medio ambiente y satisfacen tanto las necesidades esenciales como los deseos de las personas (1). Las tecnologías de la información y la comunicación (TIC) agrupan los elementos y las técnicas utilizadas en el tratamiento y la transmisión de las informaciones, principalmente de informática, Internet y telecomunicaciones. Por extensión, designan el sector de actividad económica (2). Se puede definir para la investigación, que tecnología es la aplicación de un conjunto de conocimientos y habilidades, con el objetivo de lograr una solución que permita al ser humano resolver un problema determinado hasta lograr satisfacer una necesidad en un ámbito concreto.

#### **Protocolo de Comunicación**

En el campo de las telecomunicaciones, un protocolo de comunicaciones es el conjunto de reglas normalizadas para la representación, señalización, autenticación y detección de errores necesario para enviar información a través de un canal de comunicación. Un ejemplo de un protocolo de comunicaciones simple adaptado a la comunicación por voz es el caso de un locutor de radio hablando a sus radioyentes (3). Los protocolos de comunicación permiten el flujo de información entre equipos que manejan lenguajes distintos, por ejemplo, dos computadores conectados en la misma red pero con protocolos diferentes no podrían comunicarse jamás, para ello, es necesario que ambas "hablen" el mismo idioma (4). Tomando como idea, que los protocolos de comunicación para la comunicación digital por redes de computadoras tienen características destinadas a asegurar un intercambio de datos fiable a través de un canal de comunicación imperfecto.

#### **Plataforma**

En informática, una plataforma es un sistema que sirve como base para hacer funcionar determinados módulos de hardware o de software con los que es compatible. Dicho sistema está definido por un estándar alrededor del cual se determina una arquitectura de hardware y una plataforma de software (incluyendo entornos de aplicaciones). El término **plataforma cruzada** se refiere a la capacidad de un



software o hardware para correr de modo idéntico sobre plataformas diferentes (5). Utilizaremos el término plataformas, que establecen los tipos de arquitectura, sistema operativo, lenguaje de programación o interfaz de usuario compatibles.

## **Aplicaciones**

En informática, una aplicación es un programa informático que permite a un usuario utilizar una computadora con un fin específico. Las aplicaciones son parte del software de una computadora, y suelen ejecutarse sobre el sistema operativo. Una aplicación es un programa de computadora que se utiliza como herramienta para una operación o tarea específica (6).

En la ingeniería de software se denomina **aplicación web** a aquellas herramientas que los usuarios pueden utilizar accediendo a un servidor web a través de Internet o de una intranet mediante un navegador. Es una aplicación que se codifica en un lenguaje soportado por los navegadores web en la que se confía la ejecución al navegador. En otras palabras es una aplicación que se ejecuta en el navegador del SO (sistema operativo), por lo que depende del navegador web (7).

Una **aplicación de escritorio** es un programa que está instalado en el ordenador del usuario, se ejecutada directamente por el sistema operativo. Son dependientes del SO, del ordenador y sus capacidades (8).

## **Dispositivos móviles**

Los dispositivos móviles (también conocidos como computadora de mano) son aparatos de pequeño tamaño, con algunas capacidades de procesamiento con conexión permanente o intermitente a una red. Tienen memoria limitada, diseñados específicamente para una función, pero que pueden llevar a cabo otras funciones más generales (9).

## **Tecnología Inalámbrica**

El término tecnología inalámbrica hace referencia a la posibilidad de conectar varios dispositivos entre

sí o conectar un dispositivo a una red sin necesidad de cables, incorporando esquemas de conectividad como Wi-Fi, GPRS, Bluetooth, etc. Las conexiones inalámbricas que se establecen entre los empleados remotos y una red confieren a las empresas flexibilidad y un conveniente panorama de beneficios asociados a la captura procesamiento y análisis de información en tiempo real **(10)**.

## **1.2. Análisis del objeto de estudio**

Este epígrafe se centrará en la tecnología móvil; sus elementos y el desarrollo actual de los mismos. Se comentará la evolución de la telefonía celular en el ámbito nacional e internacional. Se detallará una breve explicación de los navegadores. Se mencionarán las características de los protocolos de comunicación móvil, los navegadores y los tipos de aplicaciones móviles, así como la seleccionada a desarrollar.

### **1.2.1 Tecnologías Móviles**

La telefonía móvil, también llamada telefonía celular, básicamente está formada por dos grandes partes: una red de comunicaciones (o red de telefonía móvil) y los terminales (o teléfonos móviles) que permiten el acceso a dicha red. El término “Tecnología Móvil” hace referencia a la posibilidad de trasladar una actividad determinada, que normalmente se inscribe en un espacio físico, de un sitio a otro. Es decir, se puede llevar a cabo una tarea determinada sin estar físicamente en la oficina, y efectuar un sin número de actividades potenciadas por el auge de nuevos dispositivos suficientemente pequeños **(11)**.

#### **Ventajas de la tecnología móvil (11):**

- **Aumento de la productividad:** La posibilidad de acceder a la información y realizar cualquier trabajo con la misma desde cualquier lugar, significa un ahorro de tiempo y presupuesto. Es posible minimizar el tiempo de inactividad ocasionado por viajes o interrupciones que llevan siempre las principales herramientas de productividad.
- **Flexibilidad y rapidez:** Las soluciones móviles permiten llevar las aplicaciones informáticas y de gestión de la información a cualquier lugar, lo que significa que se podrá dar respuesta a las distintas expectativas con total flexibilidad.
- **Portabilidad:** Es la capacidad de trasladar nuestro número de teléfono móvil de un operando a otro, manteniendo el mismo número. Mientras que en sus servicios influye la adaptabilidad, escalabilidad, facilidad al instalar y conformidad.
- **Usabilidad:** La interacción entre el usuario y el dispositivo debe ser sencilla y fácil de usar.

### 1.2.2 Protocolos de comunicación móvil

Wireless Application Protocol o WAP (protocolo de aplicaciones inalámbricas) es un estándar abierto internacional para aplicaciones que utilizan las comunicaciones inalámbricas. Se trata de la especificación de un entorno de aplicación y de un conjunto de protocolos de comunicaciones para normalizar el modo en que los dispositivos inalámbricos, se pueden utilizar para acceder a correo electrónico, grupo de noticias y otros **(12)**.

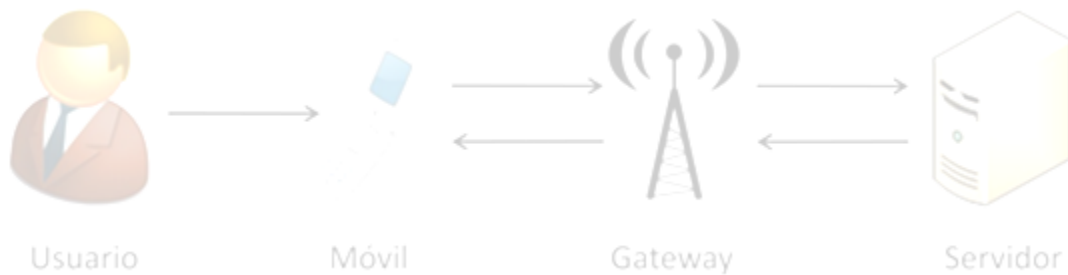
WAP define un entorno de aplicación y un grupo de protocolos para aplicaciones y servicios accesibles a través de terminales móviles. Consiste en un conjunto de especificaciones, definidas por la Open Mobile Alliance / WAP Fórum, que permiten que los desarrolladores diseñen aplicaciones de interconexión para terminales móviles, típicamente teléfonos. La tecnología WAP permite que los usuarios de estos dispositivos puedan acceder a servicios disponibles en Internet **(12)**.

Dentro de las tecnologías utilizadas en WAP se encuentran **(12)**:

- WML (Wireless Markup Language) como lenguaje de presentación de contenidos.
- WDP a la capa de transporte, con funcionalidad equivalente al protocolo UDP de Internet, y se apoya en los servicios de la portadora WAP, que depende de la red móvil que se utilice en la terminal.
- Se define como interfaz de acceso de las aplicaciones a las funciones de telefonía del terminal con WTAI (Wireless Telephony Application Interface), y también un sencillo lenguaje de "scripting", WMLScript, basado en JavaScript.
- Capa de aplicaciones (WAE): este es un ambiente de aplicación de propósito general basado en una combinación de las tecnologías WWW y de telefonía móvil.
- Capa de sección (WSP) protocolo de sesión inalámbrica (WSP) provee la capa de aplicación del WAP con una interfaz consistente para dos servicios de sesión. El primero es un servicio orientado a la conexión que opera sobre el protocolo de capa de transición (WDP).
- Capa de transición (WTP) el protocolo de transición inalámbrica (WTP), corre en el tope de un servicio de datagrama. WTP opera eficientemente sobre redes de datagramas inalámbricos seguros y no seguros.
- Capa de seguridad (WTLS) es un protocolo de seguridad basado en el protocolo de seguridad de capas de transporte estándar de la industria (TLS), antiguamente conocido como SSL. WTLS se desarrolla para usarse con los protocolos de transporte WAP y ha sido optimizado para usar sobre canales de comunicación de banda angosta.

La arquitectura WAP consta de varias partes, en la **Figura 1**, se muestra su funcionamiento cuando se

ejecutan peticiones realizadas por usuarios de terminales móviles (12):



**Figura 1:** Funcionamiento de WAP.

- 1) El usuario solicita la página WAP que quiere visualizar.
- 2) El navegador del móvil envía la petición con la dirección de la página solicitada y la información sobre el abonado al WAP Gateway (software capaz de conectarse a la red de telefonía móvil y a Internet).
- 3) El Gateway examina la petición y la envía al servidor donde se encuentra la información solicitada.
- 4) El servidor añade la información HTTP (Protocolo de Transferencia de Hipertexto) o HTTPS (Protocolo de Transferencia de Hipertexto Seguro) pertinente y envía la información devuelta al Gateway.
- 5) En el Gateway se examina la respuesta del servidor, se valida el código en busca de errores y se genera la respuesta que se envía al móvil.
- 6) El micro navegador reconoce la información recibida y si el código es correcto lo muestra en pantalla.

El Gateway es la parte más importante del esquema anterior. Su función es convertir los protocolos de comunicación WAP (ejemplo: WSP, WTP, WTLS) a protocolos HTTP, TCP/IP y viceversa (12).

El auge en el desarrollo de los portales WAP se debe en gran medida a que los sitios web no fueron construidos teniendo en cuenta las limitaciones y características que poseen los dispositivos móviles para la visualización del contenido (12).

### 1.2.3 Navegadores móviles

Los navegadores móviles son esencialmente versiones reducidas de los navegadores estándar web HTML vistos en una computadora. Cada sistema operativo móvil tiene su navegador específico. Además, a medida que avanza la tecnología de teléfonos inteligentes, también lo hace la tecnología del navegador móvil. Hay varios factores que entran en juego con los navegadores móviles como el tamaño de resolución de pantalla, el tiempo de carga de páginas web, los lenguajes web compatibles,

la visibilidad de contenido dinámico y el sistema operativo del teléfono (13).

#### **1.2.4 Aplicaciones móviles**

Las aplicaciones móviles Smartphone y PDA son extensiones informáticas para dispositivos portátiles, como los teléfonos inteligentes (14).

Las aplicaciones móviles se clasifican de tres formas (14):

- Aplicaciones web para móviles.
- Aplicaciones móviles nativas.
- Aplicaciones híbridas.

#### **Desarrollo de aplicación web móvil**

Es el desarrollo de aplicaciones web regulares, pero optimizadas para ser visualizadas desde un dispositivo móvil o tablet. Por definición, estas aplicaciones serán accedidas utilizando el navegador que viene por defecto dentro de los dispositivos: ya sea Safari (o algún otro) en iPhone o iPad y digamos Dolphin (entre otros) para Android.

Características de aplicaciones web móviles (14):

- Serán accedidas a través del navegador móvil.
- Para todos los dispositivos se desplegará de la misma forma.
- Es prácticamente lo mismo que desarrollar cualquier aplicación o sitio web: utilizas HTML, CSS y Java Script.

Analizando las características y el objetivo de la propuesta se selecciona la opción de implementar una aplicación web móvil. Con dicha elección se puede hacer uso de las facilidades de la web; permite el uso de tecnologías multiplataforma como HTML, JavaScript y CSS para que tenga alcance en varios dispositivos. No es factible realizar una aplicación nativa puesto que se debería realizar versiones para todas las plataformas móviles actuales. Por otra parte, las aplicaciones web híbridas solo serían convenientes si utilizáramos algunos de los componentes específicos de cada teléfono (cámara, cronómetro, notas, reproductor, etc.).

#### **1.3. Análisis de soluciones existentes**

Hoy en día la web móvil provee una amplia gama de servicios al cliente y todo al alcance de tu teléfono móvil. Esto ha provocado que muchas compañías motoricen el desarrollo hacia esta industria, debido al desarrollo de nuevas tecnologías y dispositivos. Implicando un auge en el desarrollo de la web móvil, para la consulta y gestión de contenidos (15).

Actualmente existe un gran número de sistemas de gestión para distintos sectores, desarrollados para

celulares. Los cuáles sirvieron como apoyo para la selección de la tecnología, lenguaje de programación, arquitectura y requisitos no funcionales para la propuesta de solución.

### 1.3.1 **Ámbito nacional**

➤ Interfaz de usuario para la administración de los servicios para la plataforma de telefonía móvil, COMCEL.

El sistema es una interfaz para la gestión de los servicios de la plataforma COMCEL<sup>1</sup>. El cual es capaz de gestionar información de los proveedores de contenidos, de los servicios que ofrece la plataforma, así como de los contenidos y su información complementaria asociada. Permite el acceso a las personas autorizadas desde cualquier sitio físico, garantizando que los proveedores puedan gestionar sus contenidos desde cualquier área o local donde se encuentren. Es capaz de integrarse a la plataforma telefónica y acceder a sus funcionalidades. Además debe garantizar la seguridad de la información que maneja así como tiempos de respuestas aceptables para las peticiones de los usuarios (16).

➤ Portal WAP de SIIPOL (Sistema de Investigación e Información Policial) Móvil.

Portal WAP desarrollado en la Universidad de Ciencias Informática (UCI) para ser utilizado en Venezuela con fines policiales, en los cuales la seguridad de los datos que se manejan es un aspecto importante a tener en cuenta, por lo que el uso de esta tecnología al servicio de los órganos policiales se pone en práctica por primera vez en este país. Este producto informatiza un conjunto de procesos para dar las respuestas operativas. El cual permite a los funcionarios conectarse mediante dispositivos móviles a la base de datos del SIIPOL de dicha institución, con el objetivo de realizar consultas sobre entidades operativas (armas, vehículos, objetos y personas) (17).

### 1.3.2 **Ámbito internacional**

➤ Movilizer

Aplicación Movilizer ofrece herramientas claves directamente a los trabajadores y fortalece el nexo de comunicación entre el campo y la oficina en todos los aspectos de la gestión de activo. Los usuarios pueden descargarse órdenes de trabajo directamente de la base de datos y asignarlas o reasignarlas a los responsables adecuados. Los usuarios móviles pueden acceder al historial de órdenes de trabajo e información de activo para facilitar el diagnóstico y la reparación, los partes de problemas y devolución, las horas de trabajo e incorporar imágenes a la orden para futuras referencia.

Permite el intercambio de datos precisos y en tiempo real entre dispositivos (18).

➤ Aplicación Móvil para la Gestión de Calidad.

---

<sup>1</sup>Plataforma de software que tiene como objetivo, integrar los servicios de telefonía móvil existentes en la actualidad en Cuba.

Sistema móvil que permite acelerar el proceso de inspección de calidad de forma personal, ya que la persona que realiza la inspección sólo tiene que introducir el ID de material / equipo o el lote de inspección, rellenar los correspondientes valores medidos, registrar los defectos y, finalmente introducir un texto corto (descripción relacionado con el lote de inspección). El nombre de los inspectores, la fecha de la inspección, el tiempo, etc. puede ser introducido de forma automática por el teléfono móvil **(19)**.

Al realizar el estudio de los sistemas web móviles de gestión de contenido se han podido precisar las características generales presentes en los sistemas de este tipo, tales como:

- Facilitan las tareas a sus usuarios, mediante una sencilla y agradable interfaz de trabajo.
- Funcionamiento independiente o integrado a un sistema.
- Trabajo con base de datos relacionales.
- Realizar búsquedas avanzadas.

Para el desarrollo del sistema, se tomaron varios elementos de las soluciones existentes. Se decidió la reutilización de algunos requerimientos y consideraciones para el diseño, arquitectura e implementación de la aplicación móvil. Se utilizarán los criterios de funcionalidad, destacando la facilidad de manejo por el usuario y la amplia gama de operaciones y utilidades. Además de las particularidades a tener en cuenta para el diseño de la interfaz del sistema móvil. Tomando en consideración, que no existe actualmente ningún sistema móvil para la gestión y manejo de elementos de la plataforma de noticias PRIMICIA. Se puede valorar la importancia de contar con una solución móvil que se ajuste a los requisitos tratados.

#### **1.4. Selección de la metodología de desarrollo**

Una metodología de desarrollo es un conjunto de procedimientos, técnicas, herramientas y un soporte documental que ayuda a los desarrolladores a realizar un software (20).

Para la elección de una metodología de desarrollo se realizó un análisis de las metodologías más usadas y conocidas, con el objetivo de determinar cual se ajusta a las características del sistema. Definiendo como metodología para guiar el proceso de desarrollo del software RUP (Proceso Unificado de Desarrollo), además de ser la establecida en el proyecto (20).

RUP es un conjunto de metodologías adaptables al contexto y necesidades de cada organización, que define claramente quién, cómo, cuándo y qué debe hacerse en el proyecto (20).

### **Características esenciales (20):**

Los autores de RUP destacan que el proceso de software propuesto por RUP tiene tres características esenciales: está dirigido por Casos de Uso, Centrado en la Arquitectura, e Iterativo e Incremental (20).

- **Dirigido por Casos de Uso:** Los casos de uso reflejan lo que los usuarios futuros necesitan y desean. A partir de aquí los casos de uso guían el proceso de desarrollo.
- **Centrado en la Arquitectura:** La arquitectura muestra la visión común del sistema completo en la que el equipo de proyecto y los usuarios deben estar de acuerdo.
- **Iterativo e Incremental:** RUP propone que cada fase se desarrolle en iteraciones. Una iteración involucra actividades de todos los flujos de trabajo y los incrementos al crecimiento del producto.

RUP le permitirá al equipo de desarrollo laborar con mayor precisión y calidad, mejorando la aplicación en cualquier instante del desarrollo, ganando de este modo un sistema con la robustez necesaria, dentro del cronograma planeado y la inversión prevista. Si se opera bien los procesos, es permisible cumplir con todos los artefactos que propone RUP independientemente del tamaño del equipo de desarrollo. Por otra parte, la documentación que se obtiene mediante RUP favorecerá a lograr un mejor entendimiento de la aplicación por parte del equipo de desarrollo, y sirve de referencia para posteriores trabajos sobre la misma.

RUP establece actividades y criterios que conducen a un sistema desde su máximo nivel de abstracción hasta su nivel más concreto. UML (Unified Modeling Language) ofrece la notación gráfica necesaria a RUP para representar los sucesivos modelos que se obtienen en este proceso.

### **1.5 Lenguaje de modelado**

El lenguaje de modelado, se emplea para especificar o para describir métodos o procesos. Se utiliza para definir un sistema, para detallar los artefactos en el sistema y para documentar y construir (21).

#### **1.5.1 Lenguaje Unificado de Modelado (UML)**

Para el modelado se usa UML. Este lenguaje de modelado aparece en 1995 como el Método Unificado, que posteriormente se reorienta para dar lugar al Unified Modeling Language (UML), la notación orientada a objetos más popular en la actualidad. UML es un lenguaje de modelado gráfico para visualizar, especificar, construir y documentar los artefactos de un sistema. Está compuesto por diversos elementos gráficos que se combinan para conformar diagramas. La finalidad de estos es representar diversas perspectivas de un sistema, a las cuales se les conoce como modelo. Un modelo UML describe lo que supuestamente hará un sistema, pero no dice cómo implementar dicho sistema. Ha sido adoptado por muchos proveedores de herramientas CASE y es el lenguaje de modelado soportado por RUP, metodología escogida para el desarrollo del sistema (22).



Se decide utilizar el Lenguaje Unificado de Modelado (UML) pues permite visualizar, especificar, construir y documentar el sistema.

Para realizar la modelación con este lenguaje se utilizan las herramientas CASE las cuales facilitan la planificación y gestión del proyecto informático.

## **1.6. Herramienta CASE**

La principal ventaja de la utilización de las herramientas CASE (Computer Aided Software Engineering: Ingeniería de Software Asistida por Ordenador) es la mejora en la calidad de los desarrollos realizados y, en segundo término, el aumento de la productividad. Estas herramientas permiten un desarrollo y un refinamiento visual de las aplicaciones mediante la utilización de controles gráficos, permiten además generar automáticamente el código base, proporcionan la reutilización de componentes y hace más portable la documentación a generar (23).

### **1.6.1 Visual Paradigm 6.4**

Es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientado a objetos, construcción, pruebas y despliegue. Utilizar UML ayuda a construir en menor tiempo aplicaciones de calidad a menor costo. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación minimizando el tiempo invertido y aumentando la eficiencia al modelar. Es básicamente una herramienta diseñada para cumplir las mismas funcionalidades que Rational Rose, con la diferencia de que es compatible con sistemas Linux (23).

Las principales ventajas de Visual Paradigm son (23):

- Presenta un diseño centrado en casos de uso y enfocado al negocio.
- Usa un lenguaje estándar común a todo el equipo de desarrollo y facilita la comunicación.
- Tiene capacidades de ingeniería directa (versión profesional) e inversa.
- Tiene modelos y códigos que permanecen sincronizados en todo el ciclo de desarrollo.
- Presenta disponibilidad de múltiples versiones, para cada necesidad y múltiples plataformas.

El mismo puede ejecutarse de diferentes formas, puede ser instalado o portable y posee varios ambientes, es capaz de generar código para Java<sup>2</sup> y .NET<sup>3</sup>, de manera que proyectos escritos en estos lenguajes puedan desarrollarse fácilmente o hacer la ingeniería inversa, a partir del código fuente generar diagramas (23).

---

<sup>2</sup> Lenguaje de programación orientado a objetos, permite crear programas que funcionan en cualquier tipo de ordenador y sistema operativo.

<sup>3</sup> Plataforma de desarrollo que ha lanzado al mercado Microsoft.

Por lo antes analizado en el trabajo de diploma se hace uso de Visual Paradigm en su versión 6.4 ya que es una herramienta CASE que permite construir diagramas UML, como son los flujos de eventos del sistema, las clases, todo lo que es documentación tanto de desarrollo como procesos de negocio. Además genera el mapeo de objetos-relacionales para el lenguaje de programación PHP.

### **1.7. Lenguaje de programación**

Se define como lenguaje de programación al elemento dentro de la informática que permite crear programas mediante un conjunto de instrucciones, operadores y reglas de sintaxis; que se pone a disposición del programador para que este pueda comunicarse con los dispositivos hardware y software existentes (24).

#### **Preprocesador de hipertexto (PHP):**

Es un lenguaje de programación del lado del servidor, gratuito e independiente de plataforma, rápido, con una librería de funciones y abundante documentación. Se encuentra embebido dentro del HTML y utilizado mayormente para desarrollar servicios Web (24).

Generalmente es utilizado como módulo de Apache, lo que lo hace extremadamente veloz. Se ejecuta rápidamente utilizando poca memoria. PHP es Open Source, lo cual significa que el usuario no depende de una compañía específica para arreglar cosas que no funcionan, además no se está forzado a pagar actualizaciones anuales para tener una versión que funcione (24).

Se decidió utilizar PHP en su versión 5.3 como lenguaje de programación del lado del servidor ya que es un lenguaje de código libre y tiene abundante documentación. Es muy fácil integrar el acceso a bases de datos en páginas HTML, no requiere demasiados recursos del sistema ya que utiliza su propio sistema de administración de recursos y dispone de un sofisticado método de manejo de variables, conformando un sistema robusto y estable. Como está embebido en HTML es muy fácil ejecutarlo en el servidor (24).

#### **Lenguaje de Marcado de Hipertexto (HTML):**

HTML es un lenguaje de marcas orientado a la publicación de documentos en Internet. Se ha convertido en un estándar debido a la gran popularidad que ha alcanzado. Puede ser interpretado por cualquier navegador independiente del sistema operativo. La última versión de este lenguaje es HTML5, el cual en un término de marketing, es mencionado para agrupar las nuevas tecnologías de desarrollo de aplicaciones web: HTML5, CSS3 y nuevas capacidades de JavaScript. Es el lenguaje que más está creciendo para el desarrollo de aplicaciones móviles. Permite acceso a audio, video, geolocalización, soporte de aplicaciones web offline, almacenamiento web, selectores css3 y animación 2D (25).

Se decide utilizar HTML5 en la solución ya que permite a la aplicación funcionar igual de bien en cualquiera de los SO móviles actuales, sin necesidad de cambios. Además la última iteración de HTML5 admite la visualización de video sin el uso del complemento Flash de los navegadores móviles y estándar.

### **JavaScript**

Es un lenguaje interpretado, no requiere compilación ya que funciona del lado del cliente, los navegadores son los encargados de interpretar estos códigos. Es similar a Java, aunque no es un lenguaje orientado a objetos, el mismo no dispone de herencias. La mayoría de los navegadores en sus últimas versiones interpretan código JavaScript, el mismo puede ser integrado dentro de las páginas Web (26).

Se decide hacer uso de este lenguaje ya que aporta beneficios para la creación de aplicaciones web móviles. Además de que es dinámico y responde a eventos en tiempo real. Con JavaScript se consigue la validación de la información, controla el movimiento de imágenes, menús interactivos y animaciones de elementos. Es soportado por todos los navegadores, es sencillo de aprender y está contemplado por los distintos organismos de normalización.

### **Hojas de estilo en cascada (CSS)**

Es un lenguaje usado para definir la presentación de un documento estructurado escrito en HTML o XML, y por extensión en XHTML. El W3C (World Wide Web Consortium) es el encargado de formular la especificación de las hojas de estilo que servirán de estándar para los agentes de usuario o navegadores. La intención que persigue el desarrollo de CSS es separar la estructura de un documento de su presentación (27).

Entre los beneficios que aporta para la creación del sistema, es que por un lado, se evita hacer los archivos demasiado pesados, y se define el estilo visual de un sitio entero sin necesidad de hacerlo etiqueta por etiqueta para cada una de las páginas. Por otra parte, se trabaja con estándares, logrando una manera más nítida de trabajar.

### **1.8. Gestor de base de datos**

Un sistema gestor de bases de datos (SGBD) es la aplicación que interacciona con los usuarios de los programas de aplicación y la base de datos. Gestiona la estructura física de los datos y su almacenamiento (28).

Consiste en una colección de datos interrelacionados y una colección de programas para acceder a los datos. Proporciona un entorno conveniente y eficiente para los usuarios que lo usan para la recuperación y almacenamiento de la información.

## **PostgreSQL 9**

PostgreSQL es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD y con su código fuente disponible libremente. Es el sistema de gestión de bases de datos de código abierto que utiliza un modelo cliente/servidor. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando. Actualmente se considera un sistema libre bastante avanzado, ya que soporta la gran mayoría de las transacciones SQL y el control concurrente. Se ejecuta en la mayoría de los Sistemas Operativos más utilizados en el mundo incluyendo Linux, varias versiones de Unix y Windows **(28)**.

Por todo lo anteriormente expuesto se consideró utilizar PostgreSQL en su versión 9 para el desarrollo de la solución.

### **1.9 Framework de desarrollo**

Un framework no es generalmente una aplicación específica. Una aplicación, en cambio puede ser construida con uno o más framework. Una posible definición de framework es: conjunto de clases cooperantes que hace reusable un diseño para una clase específica de software. Un framework proporciona una guía arquitectónica para dividir el diseño en clases y definir sus responsabilidades y colaboraciones, además de dictar la arquitectura de la aplicación (29).

#### **Symfony 2.1**

Symfony es un framework para construir aplicaciones Web con PHP. En otras palabras, Symfony es un enorme conjunto de herramientas y utilidades que simplifican el desarrollo de las aplicaciones Web. Emplea el tradicional patrón arquitectónico MVC (modelo-vista-controlador) para separar las distintas partes que forman una aplicación Web, al emplear MVC favorece un mantenimiento más sencillo de las aplicaciones (29). Es multiplataforma, fácil de instalar y configurar y sigue las mejores prácticas y patrones de diseño para la web.

Se decide usar Symfony en su versión 2.1 como marco de desarrollo teniendo en cuenta que la Plataforma de Televisión Informativa PRIMICIA en sus versiones anteriores se ha desarrollado sobre este marco de trabajo. Permite un mantenimiento muy sencillo, el código es fácil de leer y está publicado bajo una licencia de software libre.

#### **JQuery Mobile 1.2:**

Es un framework de JavaScript, construido encima del popular jQuery y de la interfaz del usuario jQuery. JQuery Mobile es un marco de trabajo efectivo y unificado para escribir una aplicación web móvil, que garanticen una apariencia y un comportamiento consistente a través de las diferentes

plataformas móviles. Aunque no solo es simplemente una capa para realizar código JavaScript que funcione en todos los navegadores, sino un conjunto de herramientas que simplificará el proceso de crear páginas para móviles, desde la escritura del propio código HTML, la maquetación con CSS y la creación de efectos dinámicos con JavaScript. Aporta un diseño adaptable a móviles, proporciona velocidad y funcionalidad. Ayuda a reducir el ancho de banda usado en el móvil (30).

### **Las características básicas de jQuery Mobile incluyen (30):**

- Utilizar JavaScript avanzado y eventos.
- Utilizar un solo documento HTML con varias páginas incluidas.
- Separar su aplicación en varias páginas.
- Está desarrollado para trabajar con HTML5, para aprovechar todas las características del framework.
- Para dispositivos táctiles: Los dispositivos táctiles tienen cambios en la gestión de eventos y jQuery Mobile nos facilita la labor de adaptarnos a ellos.

Se decide utilizar en la solución jQuery Mobile 1.2, ya que es compatible tanto con dispositivos de alta gama como con dispositivos con menor capacidad, como aquellos que no son compatibles con JavaScript. Además de haber sido creado específicamente para el desarrollo de aplicaciones usadas en dispositivos móviles.

### **1.10. Entorno de desarrollo**

Un entorno de desarrollo integrado o IDE, es un programa informático compuesto por un conjunto de herramientas de programación. Puede dedicarse en exclusiva a un sólo lenguaje de programación, o poder utilizarse para varios. Un IDE es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica (31).

#### **Netbeans 7.2**

Permite que las aplicaciones sean desarrolladas a partir de un conjunto de componentes de software llamados módulos. Un módulo es un archivo Java que contiene clases de java escritas para interactuar con las APIs de Netbeans y un archivo especial que lo identifica como módulo. Las aplicaciones construidas a partir de módulos pueden ser extendidas agregándole nuevos módulos. Debido a que los módulos pueden ser desarrollados independientemente, las aplicaciones basadas en la plataforma Netbeans pueden ser extendidas fácilmente por otros desarrolladores de software (31)

El IDE en el cual se decide desarrollar es NetBeans con su versión 7.2, teniendo en cuenta que el

framework utilizado es Symfony 2.1 y presenta módulos de soporte para dicho framework, ofrece completamiento de código para las variables de Symfony como por ejemplo:

**generate:bundle** → Crea un nuevo Bundle

**doctrine:generate:entity** → Genera una nueva entidad dentro del Bundle

**doctrine:schema:drop [--force]** → Elimina todas las tablas de la base de datos del sistema.

**doctrine: database: create** → Crea la base de datos del sistema según los parámetros de configuración especificados.

**doctrine:generate:form MibundleMyBundle:Clase** → Crea un formulario con los campos de la clase referenciada (Clase) perteneciente al Bundle: Mibundle, la nueva clase será ClaseType.php ubicada en la carpeta Form del Bundle.

**cache:clear** → Limpia la cache almacenada por Symfony

### **Conclusiones parciales**

En el capítulo se analizaron los conceptos vinculados con el tema de la investigación, como bases para el entendimiento del mismo. Se realizó el análisis de las soluciones existentes, lo que ayudó a definir el prototipo de interfaz y la arquitectura de información de las aplicaciones móviles. Se seleccionaron las herramientas, tecnologías y metodologías para el desarrollo de la aplicación. Con el uso de RUP como metodología de desarrollo se logró generar los artefactos necesarios, siendo este un proceso de vital importancia para el entendimiento de la aplicación a desarrollar. Con el lenguaje de modelado UML se crea una abstracción del sistema a implementar, para describir las funcionalidades básicas. Permitiendo además una posible continuidad de desarrollo. Se defiende la idea de usar como herramienta de modelado Visual Paradigm 6.4 ya que presenta mecanismos de ingeniería inversa y generación de código. El uso del framework Symfony 2.1 permite una construcción bien estructurada de la aplicación, optimizando el uso de lenguajes como JavaScript y PHP5. Con el uso del framework de JavaScript, jQuery Mobile 1.2, se dirige la aplicación a la web móvil.

## **CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA: PRIMICIA MÓVIL**

### **Introducción**

En este capítulo se abordarán argumentos importantes para la elaboración del sistema móvil para la publicación de noticias de PRIMICIA, además de efectuar la representación de los artefactos logrados a través del modelo de dominio y la especificación de los requisitos funcionales y no funcionales que debe presentar la aplicación una vez terminado. Se determinan los casos de uso del sistema y los actores que interactuarán con este, elaborando una descripción de cada uno de ellos.

Se plantea alcanzar la administración y gestión de los recursos necesarios para la publicación de noticias. Una vez integradas estas funcionalidades a la propuesta se elevarán las prestaciones de la misma.

### **2.1 Modelo de Dominio**

Un sistema, por pequeño que sea, generalmente es complicado. Por eso se necesita dividirlo en piezas si se pretende comprenderlo y gestionar su complejidad. Esas piezas se pueden representar a través de modelos que permitan abstraer sus características esenciales (32).

Se propone un modelo del dominio, ya que en el sistema los procesos del negocio no están bien definidos, no son visibles y las fronteras no están bien establecidas. Además permite de manera visual mostrar al usuario los principales conceptos que se manejan en el dominio del sistema en desarrollo. En este caso no se selecciona el modelo de negocio, ya que es para sistemas que den soporte a un conjunto de procesos de un negocio de la vida real.

#### **2.2.1 Conceptos o eventos fundamentales del dominio**

Para brindar una mejor comprensión del diagrama del modelo de dominio a continuación se realiza una breve descripción de los conceptos o eventos encontrados en el problema.

**Primicia:** Es una plataforma que permite la administración y transmisión de noticias en diferentes formatos mediante un canal de televisión.

**Transmisión:** Transmite un canal que recepciona las señales que pueden ser interna o las externas. Es el envío y recepción de la información en forma de noticias a través de un equipo informático por medio de señales encaminadas en un canal.

**Canal:** Es el emisor por el cual se van a enviar las diferentes señales. Por él se transmite audio y video a receptores de televisión en un área concreta.

**Señal:** Es la señal que es transmitida por el Subsistema de Transmisión.

**Externa:** Son aquellas señales con las que enlazará la plataforma para una transmisión en vivo. Pertenece a una entidad en particular y no fue generado por las noticias de PRIMICIA.

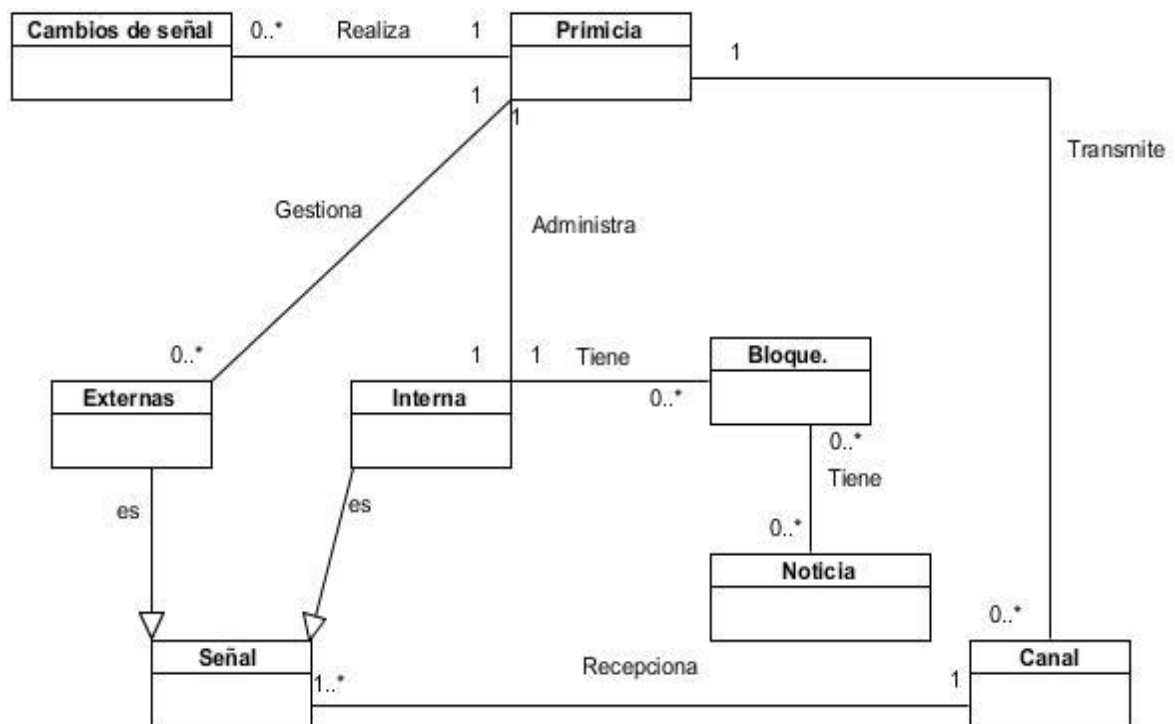
**Interna:** Es la señal del propio canal, es decir aquella generada por las noticias y bloques.

**Cambios de señal:** Son cambios programados a transmitirse en una fecha y hora indicada y que tienen una señal asociada.

**Noticia:** Son aquellas noticias que no han sido publicadas, estas pueden que se hayan obtenido por fuentes web o por otra vía.

**Bloque:** Es un elemento utilizado para agrupar y planificar las noticias que serán mostradas.

### 2.2.2 Diagrama del modelo de dominio



**Figura 2:** Diagrama del dominio.

### 2.3 Especificación de requisitos de software

Los requisitos se pueden clasificar en: funcionales y no funcionales. Los requerimientos deben ser descritos como una característica del sistema a entregar (33).

Los requisitos se han convertido en un punto clave en el desarrollo de las aplicaciones informáticas. Un gran número de proyectos de software naufragan debido a una mala definición, especificación o



administración de requisitos. La calidad con que se realice la captura de los requisitos influye en todo el proceso de desarrollo del software, repercutiendo en el resto de las fases de desarrollo del mismo (33).

### **2.3.1 Requisitos funcionales**

Los requisitos funcionales definen las acciones que debe realizar el sistema. Son capacidades o condiciones que el sistema debe cumplir, estos se mantienen invariables sin importar con que propiedades o cualidades se relacionen (33).

**RF 1:** Insertar señal: El sistema debe permitir que se inserten señales.

**RF 2:** Modificar señal: El sistema debe permitir que se modifique una señal seleccionada.

**RF 3:** Eliminar señal: El sistema debe permitir que se elimine una señal o varias señales.

**RF 4:** Listar señales: El sistema debe permitir que se muestre un listado de las señales.

**RF 5:** Filtrar señal: El sistema debe permitir la búsqueda de señales según el nombre y si se encuentra activa o no.

**RF 7:** Cambiar señal: El sistema debe permitir que se active la señal a la que el usuario desea cambiar y se le envíe un mensaje a la transmisión informando el cambio.

**RF 8:** Insertar cambio programado: El sistema debe permitir que se inserten cambios de señal. La fecha de inicio y fecha de fin del cambio no puede estar en el rango de la de otros cambios.

**RF 9:** Eliminar cambio programado: El sistema debe permitir que se elimine un cambio programado de señal

**RF 10:** Modificar cambio programado: El sistema debe permitir que se modifique un cambio programado de señal.

**RF 11:** Listar cambio programado: El sistema debe permitir que se muestre un listado de los cambios programados.

**RF 12:** Filtrar cambio programado: El sistema debe permitir que el usuario pueda buscar un cambio según la señal asociada al mismo.

**RF 13:** Insertar bloque: El sistema debe permitir que se inserten bloques. La fecha de inicio y fecha de fin del bloque no puede estar en el rango de la de otros bloques.

**RF 14:** Modificar bloque: El sistema debe permitir que se modifique un bloque seleccionado.

**RF 15:** Eliminar bloque: El sistema debe permitir que se elimine un bloque o varios bloques.

**RF 16:** Listar bloques: El sistema debe permitir que se muestre un listado de los bloques.

**RF 17:** Filtrar bloque: El sistema debe permitir la búsqueda de bloques según el nombre y si se encuentra activo o no.

**RF 18:** Asignar noticias a bloque: El sistema debe permitir que se asignen noticias a los bloques.

**RF 19:** Eliminar noticia de los bloques: El sistema debe permitir eliminar noticias que han sido asignada a los bloques.

**RF 20:** Eliminar noticias: El sistema debe permitir que se eliminen noticias que se encuentran publicadas en un bloque.

### **2.3.2 Requisitos no funcionales**

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener; son las características que lo hacen atractivo, usable, rápido y confiable. Para lograr la satisfacción del cliente y una buena calidad en el sistema se listaron los siguientes requerimientos no funcionales (33).

#### **Apariencia o interfaz externa:**

**RNF 1:** Usar la menor cantidad de imágenes y el formato más idóneo de acuerdo a las características del móvil, además de tener buena calidad visual y ser comprensibles.

**RNF 2:** La aplicación se debe visualizar de forma correcta en todos los dispositivos móviles.

#### **Usabilidad:**

**RNF3:** La información debe poder ser accedida por todos los usuarios de móviles que tengan habilitado servicio de navegación, sin poder modificar el contenido de las páginas.

**RNF4:** La aplicación debe ser sencilla de utilizar para cualquier cliente que necesite acceder, que exista la menor cantidad de interacciones del usuario con la aplicación y que se ajuste a las dimensiones y características del dispositivo móvil.

**RNF5:** Mensajes de error: Intentar que los mensajes de error sean informativos y facilitar la forma de regresar al contenido anterior.

**RNF6:** Evitar la introducción de texto: Siempre que sea posible, evita la introducción de texto por parte de los usuarios, utilizando:

➤ Selección por defecto: Siempre que sea posible, establecer valores preseleccionados por defecto.

➤ Minimizar el uso del teclado: Reducir el uso del teclado al mínimo.

**RNF7:** Limitar el scrolling a una sola dirección a no ser que sea imposible evitar un desplazamiento secundario

**RNF8:** Redimensión de las imágenes: Redimensionar las imágenes en el servidor si tienen un tamaño grande.

**RNF9:** No mostrar las tablas en pantallas con todos sus campos, resumir los mismos en la opción de detalles, mostrándolo en un cuadro de diálogo, de forma organizada y entendible.

#### **Software:**

**RNF10:** Cliente: El dispositivo móvil debe tener conectividad WIFI.

#### **Hardware:**

**RNF11:** A continuación se muestran los requerimientos recomendados que debe cumplir el equipamiento tecnológico del sistema:

**Tabla1:** Requerimiento de hardware

	Procesador	Memoria RAM	Disco Duro	Tarjeta de Red
Servidor de administración	Pentium IV 2.8 GHz	1 GB	120 GB	Ethernet 10/100 Mbps

## **2.4 Modelos de casos de usos del sistema**

El modelo de caso de uso debe ser un medio de comunicación entre el cliente, los usuarios y los desarrolladores del sistema. Este modelo consta de casos de uso y actores. Cada caso de uso del modelo se describe detalladamente, mostrando paso a paso el modo en que el sistema interactúa con los actores y lo que el sistema hace en el caso de uso. Los casos de uso funcionan como hebra unificadora en todo el ciclo vital de la aplicación; el mismo modelo de caso de uso se utiliza en el análisis, diseño, implementación y prueba del sistema (34).

### **2.4.1 Definición de actores**

**Tabla 2:** Actor del sistema

Actor	Descripción
Usuario	Es la persona que utiliza las funcionalidades implementadas en el sistema.

### 2.4.2 Diagrama de casos de uso

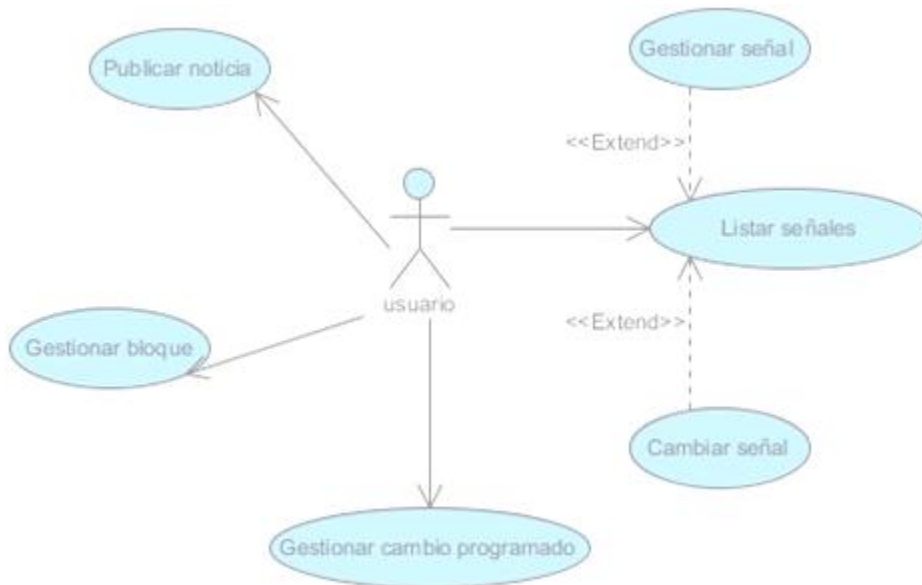


Figura 3: Diagrama de CU del sistema

### 2.4.3 Especificación de casos de usos

Tabla 3: CU 1. Publicar noticia

<b>Objetivo</b>	El objetivo es permitirle al actor poder publicar la o las noticias	
<b>Actores</b>	Usuario	
<b>Resumen</b>	El caso de uso inicia cuando el actor accede a la sección de noticia, mostrando una lista de noticias a publicar, dándole la posibilidad de asignarla a un bloque o eliminarlas.	
<b>Complejidad</b>	Media.	
<b>Prioridad</b>	Crítico	
<b>Precondiciones</b>		
<b>Pos condiciones</b>	Se publicó la noticia	
<b>Flujo de eventos</b>		
<b>Sección “Publicar Noticia”</b>		
	<b>Actor</b>	<b>Sistema</b>
1		Muestra un listado de noticias
2	Selecciona una de las siguientes opciones: <ul style="list-style-type: none"> <li>➤ Asignar bloque. Ver sección Asignar bloque.</li> <li>➤ Eliminar. Ver sección Eliminar noticia</li> </ul>	
3		Muestra la interfaz de cada opción.
<b>Sección “Asignar Bloque”</b>		
<b>Flujo básico Asignar Bloque</b>		

	<b>Actor</b>	<b>Sistema</b>
1	Selecciona la noticia publicar	
2		El sistema muestra un listado con los bloques que existen. Si la noticia ha sido asignada anteriormente algunos de estos bloques, aparecerán los bloques marcados. Las asignaciones anteriores pueden ser modificadas.
3	Selecciona el bloque y acepta	
4		El sistema verifica que se halla seleccionado un bloque, y publica la noticia.
<b>Flujos alternos</b>		
<b>3.1a .Ninguna selección.</b>		
	<b>Actor</b>	<b>Sistema</b>
3	No selecciona ningún bloque y acepta	
4		Muestra un mensaje que debe seleccionar al menos un elemento
<b>Sección: "Eliminar Noticia"</b>		
<b>Flujo básico Eliminar Noticia</b>		
	<b>Actor</b>	<b>Sistema</b>
1	Selecciona la noticia a eliminar y acepta	
2		El sistema verifica que se halla seleccionado al menos un elemento y elimina la noticia.

**Flujos alternos****1.1a: Ninguna selección**

	<b>Actor</b>	<b>Sistema</b>
1		Muestra un mensaje que debe seleccionar al menos un elemento.

## Prototipo de Interfaz

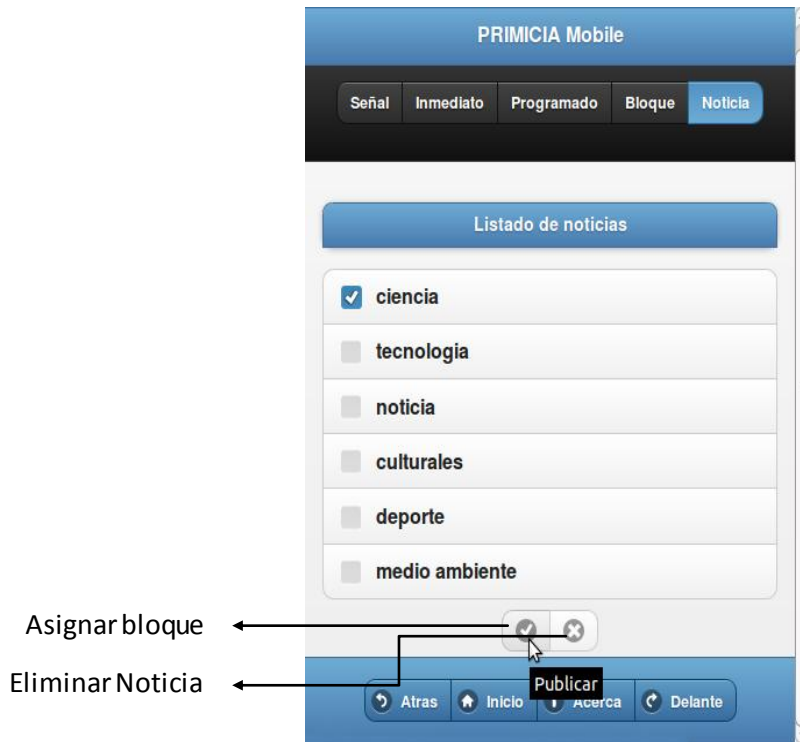


Figura 2: Prototipo de interfaz de listas noticias



Figura 3: Prototipo de publicar noticia



<b>Relaciones</b>	CU Incluidos	No existen.
	CU Extendidos	No existen
<b>Requisitos no funcionales</b>	<p><b>RNF5:</b> La aplicación debe ser sencilla de utilizar para cualquier cliente que necesite acceder, que exista la menor cantidad de interacciones usuario con la aplicación y que se ajuste a las dimensiones y características del dispositivo móvil.</p> <p><b>RNF6:</b> Mensajes de error: Intentar que los mensajes de error sean informativos y facilitar la forma de regresar al contenido anterior.</p> <p><b>RNF7:</b> Evitar la introducción de texto: Siempre que sea posible, evita la introducción de texto por parte de los usuarios.</p>	
<b>Asuntos pendientes</b>	No existen.	

Ver [anexo 2](#), para las restantes descripciones de los CU

### Conclusiones parciales

La realización del modelo de dominio permitió comprender mejor el funcionamiento del sistema, pues se identificaron y detallaron los principales eventos, términos y conceptos presentes en el entorno donde este trabajará. El proceso de captura de requisitos concluyó con la obtención de los nuevos requerimientos que la publicación de noticias, desde dispositivos móviles, debe brindar y las cualidades que el sistema debe cumplir para poder funcionar correctamente. El diagrama de casos de uso del sistema tradujo los requisitos funcionales en interacciones del actor con el sistema, permitiendo una mejor comprensión de todos los procesos a automatizar. Mediante la descripción de los casos de uso se detalla el flujo de sucesos y la manera en que se realiza la interacción entre el actor y el componente, mostrando información de cómo labora el sistema, que dará solución al problema científico planteado.

## **CAPÍTULO 3: ANÁLISIS Y DISEÑO**

### **Introducción**

En el presente capítulo se desarrolla el análisis y diseño del sistema, donde se construyen los diagramas de clases del análisis y del diseño, los cuales servirán de punto de partida para la implementación del sistema. Además se describe la arquitectura del sistema, haciendo énfasis en los patrones arquitectónicos y de diseño empleados en el framework Symfony, para de esta forma alcanzar un mejor entendimiento del diseño a realizar.

### **3.1 Modelo de Análisis**

El modelo de análisis contiene las clases del análisis y cualquier producto de trabajo asociado. El modelo del análisis puede ser un producto de trabajo temporal, como cuando evoluciona hacia un modelo de diseño, o puede permanecer durante algunos o todos los proyectos, y quizás más, sirviendo como visión general conceptual del sistema (22).

Las clases que se encuentran en estos diagramas se clasifican en Interfaz, Controladora y Entidad.

- Las clases interfaces (CI) se encargan de modelar la interacción del actor con el sistema.
- Las clases controladoras (CC) modelan los aspectos dinámicos del sistema, de forma tal, que puedan coordinar las acciones y los flujos de control.
- Las clases entidades (CE) modelan información que posee una larga vida y que a menudo es persistente. Suelen mostrar una estructura de datos lógicos y contribuyen a comprender de qué información depende el sistema.

A continuación se mostrará el diagrama de clases del caso de uso Publicar Noticia, para los restantes diagramas ver [Anexo 3](#)

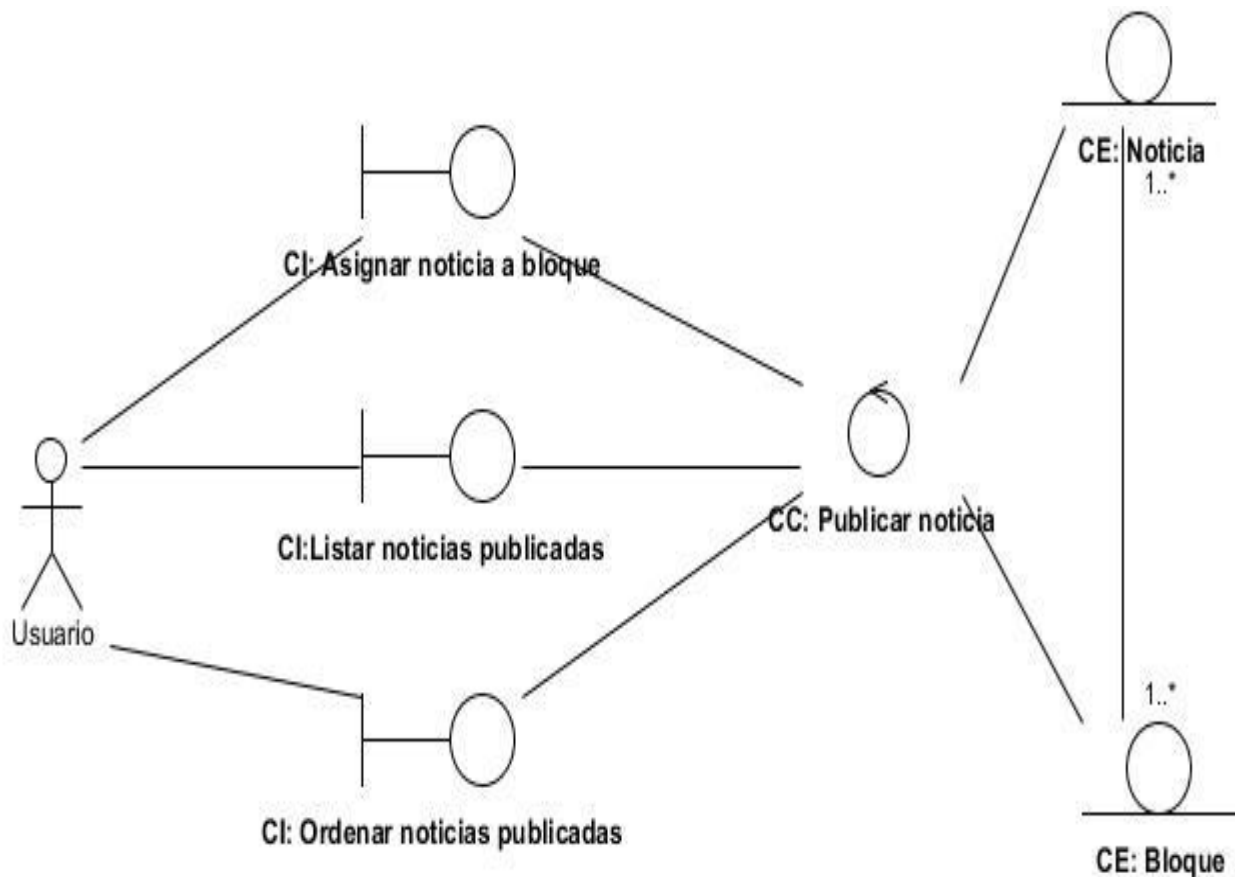


Figura 4: Diagrama de clases del análisis CU Pulicar Noticia

### 3.2 Descripción de la Arquitectura

La arquitectura del software desempeña un papel fundamental durante el desarrollo de un sistema ya que permite representar su estructura, además permite organizar sus componentes, las relaciones entre ellos, el ambiente y los principios que orientan su diseño y evolución (22).

Las técnicas metodológicas desarrolladas con el fin de facilitar la programación se engloban dentro de la llamada Arquitectura de Software o Arquitectura Lógica. Se refiere a un grupo de abstracciones y patrones que brindan un esquema de referencia útil para guiarse en el desarrollo de software dentro de un sistema informático. Dichos patrones serán capaces de aportar elementos idóneos para tomar decisiones (22).

#### 3.2.1 Patrón Arquitectónico

Los patrones arquitectónicos expresan un esquema organizativo estructural fundamental para sistemas de software y definen las reglas generales de organización, las restricciones en la forma y la estructura de un grupo numeroso de sistemas de software. La selección de un patrón arquitectónico es una decisión fundamental de diseño en el desarrollo de un sistema de software (35).

Symfony emplea el tradicional patrón arquitectónico Modelo Vista Controlador para separar las distintas partes que forman una aplicación web. Este patrón es muy usado en aplicaciones web (29).

Con el Modelo Vista Controlador la aplicación se puede desarrollar rápidamente. Separar las funciones de la aplicación en modelos, vistas y controladores hace que la aplicación sea muy ligera. El estilo de llamada y retorno Modelo Vista Controlador, se ve frecuentemente en aplicaciones web, donde la vista es la página HTML y el código que provee de datos dinámicos a la página. Esta separación también permite hacer cambios en una parte de la aplicación sin que las demás se vean afectadas (35).

- La vista transforma la información obtenida por el modelo en las páginas web a las que acceden los usuarios. Es con lo que el usuario interactúa (un motor de plantillas es parte de esta capa). En Symfony, la vista es principalmente la capa de plantillas PHP. Estas son guardadas en varios directorios views/.
- El controlador es el encargado de coordinar todos los demás elementos y transformar las peticiones del usuario en operaciones sobre el modelo y la vista. Es la pieza de código que llama al Modelo para obtener algunos datos que le pasa a la Vista para la presentación al cliente. Cuando se instala Symfony todas las solicitudes son gestionadas por un controlador frontal (app.php y app\_dev.php). Estos controladores frontales delegan la verdadera labor a las acciones. Estas acciones son, lógicamente, agrupadas en paquetes (bundles).
- El modelo representa la información con la que trabaja la aplicación y se encarga de acceder a los datos. Define la lógica de negocio, Symfony guarda todas las clases y archivos relacionados con el modelo en el directorio SenalBundle/Entity/.

### **3.2.2 Patrones de Diseño**

Los patrones de diseño son la base para la búsqueda de soluciones a problemas comunes en el desarrollo de software y otros ámbitos referentes al diseño de interacción o interfaces (35).

Dentro de los patrones de producto de software se encuentran los de análisis, arquitectura, diseño y lenguaje de programación. Para el desarrollo de la solución se aplicaron diferentes patrones de diseño, fundamentalmente: General Responsibility Assignment Software Patterns (GRASP) y Gang of Four (GoF), con el objetivo de facilitar el mantenimiento del software. A continuación se abordarán los

patrones de diseños utilizados en el sistema desarrollado para asegurar una solución mucho más confiable y contribuir a la realización de un producto reutilizable y escalable (35).

### **Patrones GRASP**

Permiten asignar correctamente las responsabilidades a cada una de las clases que intervienen en el modelo; de este grupo de patrones fueron tomados en cuenta para una correcta asignación de las relaciones entre las clases y un correcto diseño de las mismas, los siguientes (35).

**Experto:** Es un patrón que se usa más que cualquier otro al asignar responsabilidades; es un principio básico que suele utilizarse en el diseño orientado a objetos. Da origen a diseños donde el objeto de software realiza las operaciones que normalmente se aplican a la realidad que representa, por lo que ofrece una analogía con el mundo real. Con la utilización de este patrón se conserva el encapsulamiento, ya que los objetos se valen de su propia información para hacer lo que se les pide. 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 mantener (35).

Este patrón se pone de manifiesto con la utilización de Doctrine2, que es la librería externa que se utiliza para realizar la capa de abstracción en el modelo, encapsula toda la lógica de los datos y son generadas las clases con todas las funcionalidades comunes de las entidades (35).

**Creador:** Se refiere a asignar responsabilidades a las clases de crear instancias de otras conociendo que las primeras son las que contienen la información para ello. El patrón Creador guía la asignación de responsabilidades relacionadas con la creación de objetos. El propósito fundamental de este patrón es encontrar un creador que se debe conectar con el objeto producido en cualquier evento (35).

Este patrón se pone de manifiesto en la clase DefaultController donde se definen acciones y se ejecutan cada una de ellas. En esta clase se crean los objetos de las clases que representan las entidades, evidenciando de este modo que la clase DefaultController es "creador" de dichas entidades

**Alta cohesión:** El patrón propone cuán relacionadas y orientadas están las responsabilidades de una clase. Planteando la contribución entre clases para realizar tareas de elevada complejidad. Soporta un aumento de la capacidad de reutilización, siendo una gran funcionalidad, ya que una clase muy cohesiva puede destinarse a un propósito muy específico (35).

Symfony permite asignar responsabilidades con una alta cohesión, por ejemplo la clase Default Controller tiene la responsabilidad de definir las acciones para las plantillas y colabora con otras para realizar diferentes operaciones, instanciar objetos y acceder a las propiedades. Está formada por

diferentes funcionalidades que se encuentran estrechamente relacionadas proporcionando que el software sea flexible frente a grandes cambios.

**Patrón Controlador:** La primera categoría de controlador es un controlador de fachada que representa al "sistema" global. Es una clase que, para el diseñador representa de alguna manera al sistema entero. La mayor parte de los sistemas reciben eventos de entrada externa, los cuales generalmente incluyen una interfaz gráfica para el usuario operado por una persona. La misma clase controlador debería utilizarse con todos los eventos sistémicos de un caso de uso, de modo que se pueda conservar la información referente al estado del caso (35).

Todas las peticiones Web son manejadas por un solo controlador frontal (app php), que es el punto de entrada único de toda la aplicación en un entorno determinado. Cuando el controlador frontal recibe una petición, utiliza el sistema de enrutamiento para asociar el nombre de una acción y el nombre de un módulo con la URL entrada por el usuario. Este patrón se encuentra ejemplificado en los archivos DefaultController.php, InmediatoController.php, BloqueController.php, NoticiaController.php y CambiosProgramadosController.php.

#### **Patrones GoF:**

Decorator (Envoltorio): Añade dinámicamente nuevas responsabilidades a un objeto, proporcionando una alternativa flexible a la herencia (35).

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.

### **3.4 Modelo de Diseño**

El modelo de diseño es una abstracción de la implementación del sistema. Se utiliza para concebir y para documentar el diseño del sistema de software. Es un producto de trabajo integral y compuesto que abarca todas las clases de diseño, subsistemas, paquetes, colaboraciones y las relaciones entre ellos (34).

### 3.4.1 Diagrama de clases del diseño

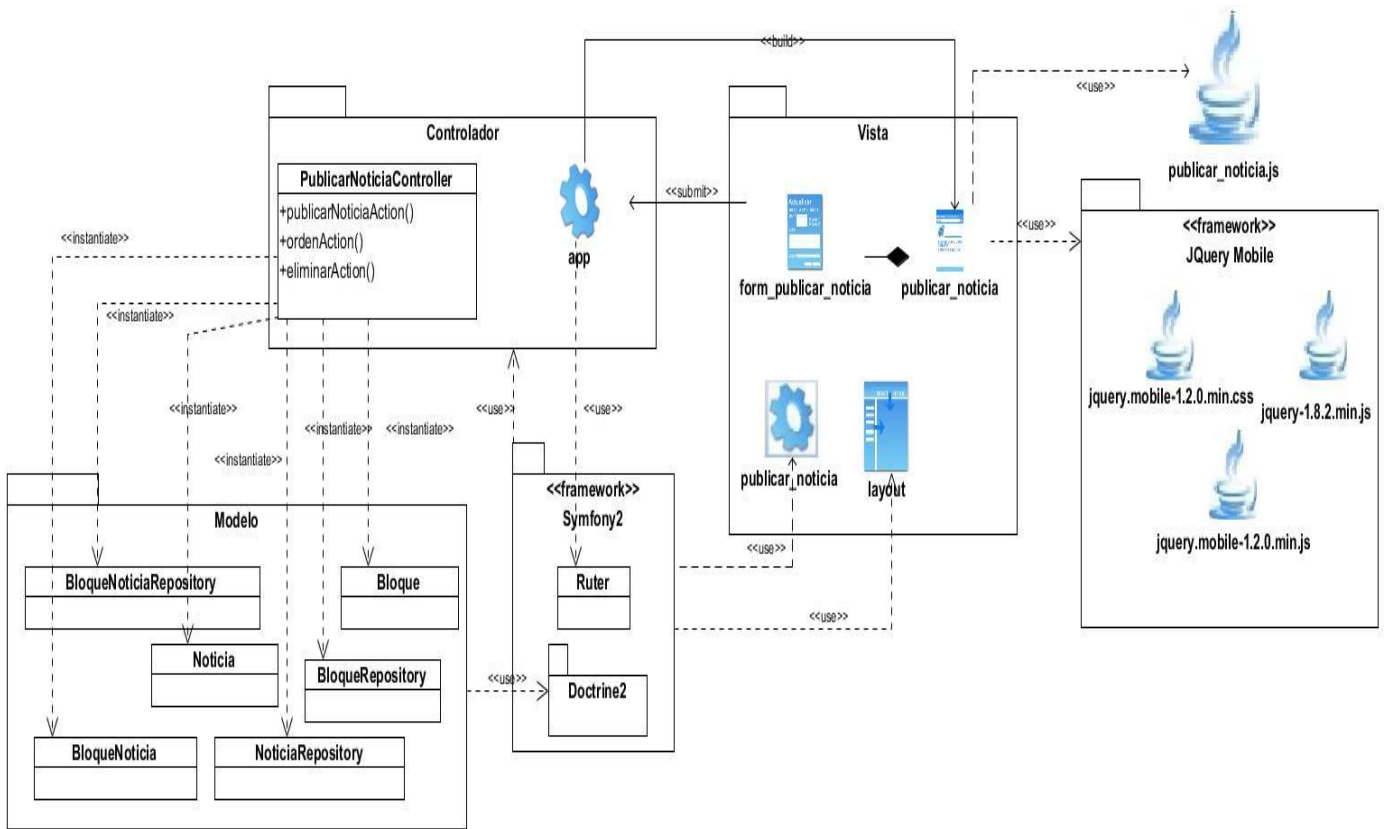


Figura 5: Diagrama de clases del diseño del CUNoticia

Para los restantes diagrama ver [anexo 4](#)

### 3.5 Modelo de Datos

Este modelo representa la realidad a través de un esquema gráfico empleando las terminologías de entidades, que son objetos que existen y son los elementos principales que se identifican en el problema a resolver con el diagramado y se distinguen de otros por sus características particulares denominadas atributos, el enlace que rige la unión de las entidades está representada por la relación del modelo. Describe las representaciones lógicas y físicas de datos persistentes utilizados por la aplicación. Para diseñar la base de datos se utilizan el diagrama de clases persistentes, el diagrama entidad-relación.

### 3.5.1 Diagrama de Clases Persistentes

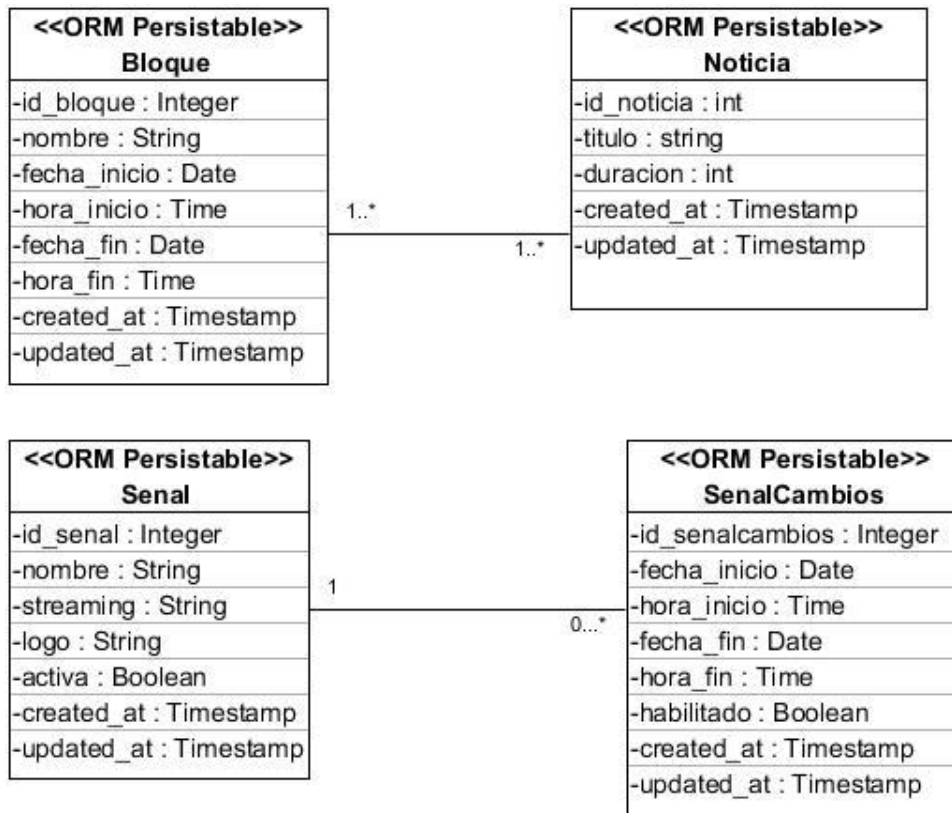


Figura 7: Diagrama de Clases Persistentes



### 3.5.2 Diagrama Entidad-Relación

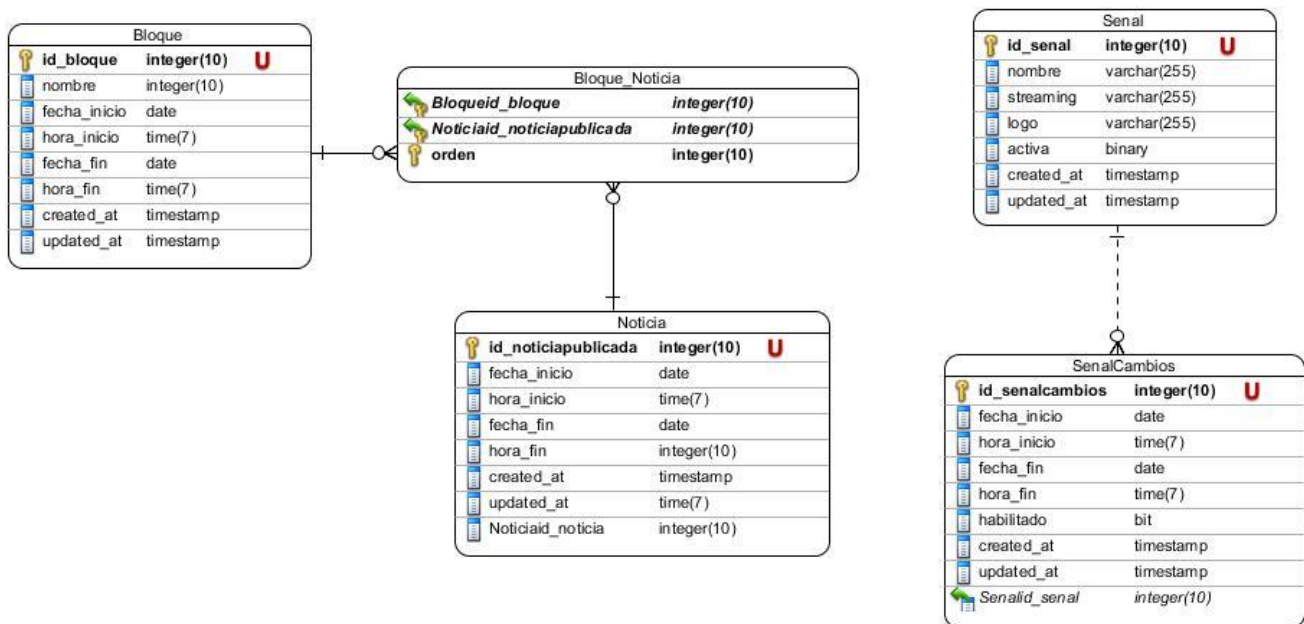


Figura 7: Diagrama Entidad-Relación

### Conclusiones parciales

Los diagramas de clases de análisis constituyeron el primer paso para la construcción de los diagramas de clases del diseño y sirvieron para reflejar una visión general de las clases a tener en cuenta para implementar las nuevas funcionalidades. El análisis de la arquitectura y diseño de Symfony permitió obtener una mejor representación de los diagramas de clases del diseño apoyado en los patrones de diseño GRASP y GOF. Las clases persistentes dieron paso a la construcción del modelo de datos para describir la estructura lógica y física de la información persistente gestionada por el sistema. Con la interpretación del diagrama entidad relación surgió la necesidad de realizar cambios en la base de datos de PRIMICIA para desarrollar satisfactoriamente las funcionalidades identificadas.

## **CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBA**

### **Introducción**

En la implementación se parte de los resultados obtenidos en el diseño. Primeramente se presenta donde va a ser ubicada o desplegada la herramienta, para lo cual se elabora el diagrama de despliegue. Se desarrolla el diagrama de componentes, donde se describen los elementos físicos del sistema y sus relaciones. Se efectúa además la realización de pruebas encaminadas a detectar fallas en el funcionamiento del sistema.

### **4.1 Modelo de Despliegue**

Este modelo tiene como objetivo capturar la configuración de los elementos de proceso y las conexiones entre elementos de proceso en el sistema (21).

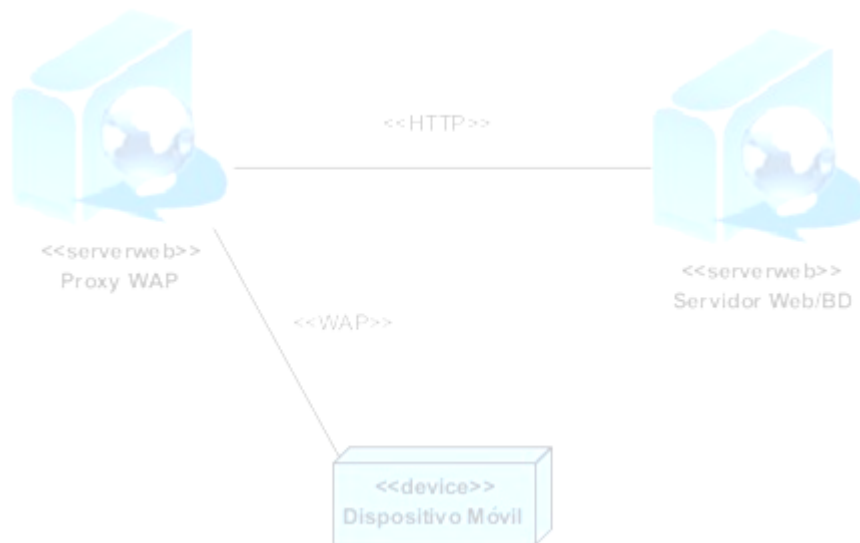
#### ***4.1.1 Diagrama de Despliegue***

Los diagramas de despliegue muestran las relaciones físicas de los distintos nodos que componen un sistema y el reparto de los componentes sobre dichos nodos. Un nodo es un recurso de ejecución tal como un computador, un dispositivo o memoria. Los estereotipos permiten precisar la naturaleza del equipo (21).

**Nodos:** Elementos de procesamiento con al menos un procesador, memoria, y posiblemente otros dispositivos.

**Dispositivos:** Nodos estereotipados sin capacidad de procesamiento en el nivel de abstracción que se modela.

**Conectores:** Expresan el tipo de conector o protocolo utilizado entre el resto de los elementos del modelo.



**Figura 8:** Diagrama de Despliegue Descripción de los nodos:

**Servidor Web/BD:** Es el nodo que albergará el servidor web y la base de datos. Es el encargado de responder a las peticiones de los clientes. El servidor tendrá instalado como sistema operativo una distribución de Linux específicamente Ubuntu 12.4, el cual sólo constará con las aplicaciones básicas para que la Plataforma brinde sus servicios, estos son el gestor de base de datos PostgreSQL, Apache y un servidor FTP.

**Dispositivo móvil:** Estos dispositivos serán los que interactuarán con la plataforma, los cuales pueden tener cualquier sistema operativo.

Descripción de los conectores:

**HTTP:** Se utilizará HTTP ya que el sistema está implementado en tecnología Web. Es un protocolo orientado a transacciones y sigue el esquema petición-respuesta entre un cliente y un servidor.

**WAP:** Protocolo de comunicaciones inalámbricas. Permite que los usuarios de los dispositivos móviles puedan acceder a servicios disponibles en la intranet o en la Internet.

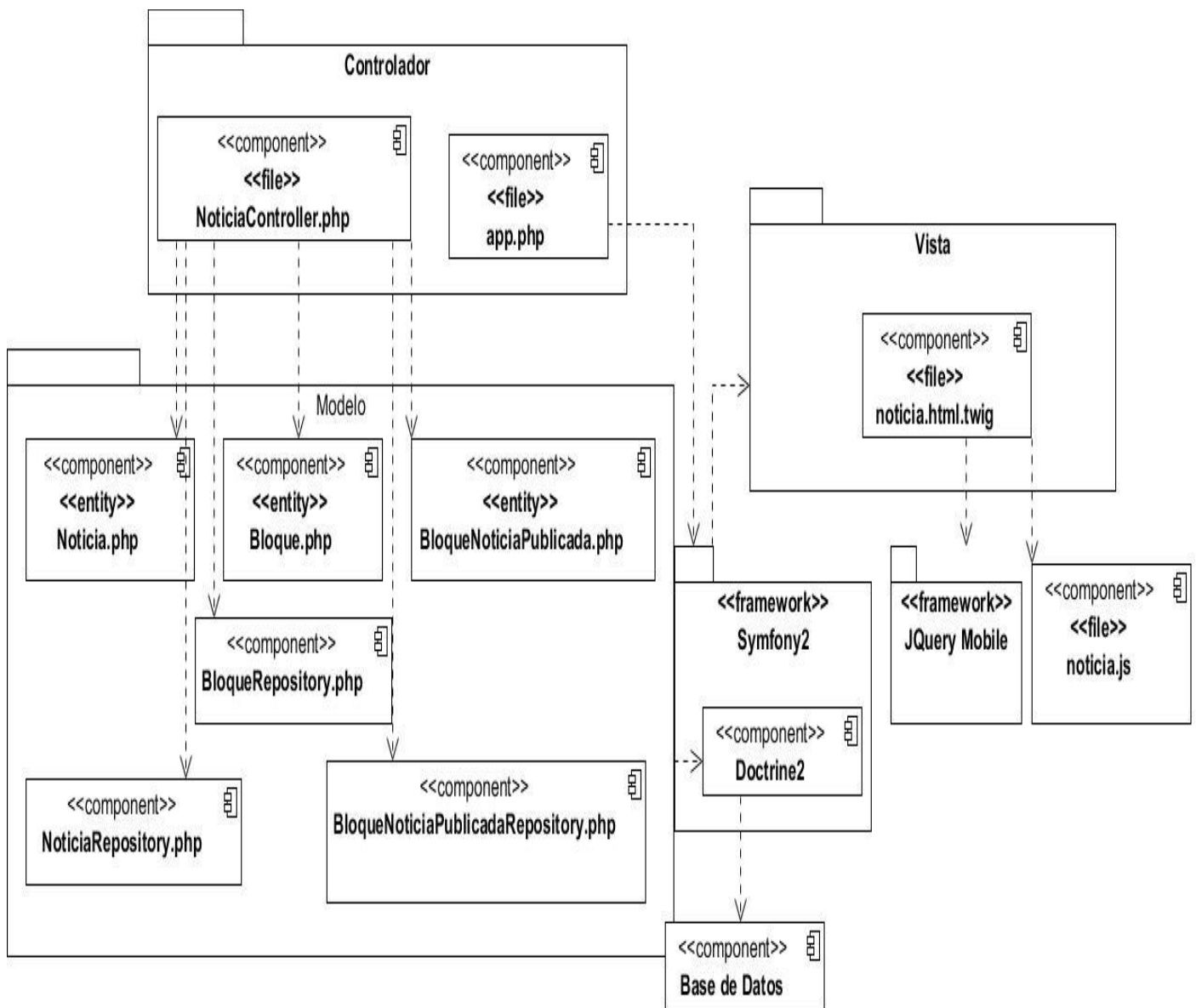
## 4.2 Modelo de Implementación

El Modelo de Implementación describe cómo las clases se implementan en términos de componentes, como ficheros de código fuente, ejecutables, entre otros. El Modelo de Implementación describe también cómo se organizan los componentes de acuerdo con los mecanismos de estructuración y

modelación disponible en el entorno de implementación, con el lenguaje o lenguajes de programación utilizados y cómo dependen los componentes uno de otros (21).

#### ***4.2.1 Diagrama de Componentes***

Los diagramas de componentes describen los elementos físicos del sistema y sus relaciones. Representa como un sistema de software es dividido en componentes y muestra dependencias entre estos componentes. Prevalcen en el campo de la arquitectura de software pero pueden ser usados para modelar y documentar cualquier arquitectura de sistema. Se puede decir que un componente es la materialización de una o más clases (21).



**Figura 9:** Diagrama de Componentes del CU Publicar Noticia. Ver [Anexo 5](#), para los restantes diagramas de componentes

**app.php:** Corresponde con el controlador frontal del sistema que es el único punto de entrada a la aplicación.

**NoticiaControler.php:** Corresponde a la clase que contiene todas las acciones que manipulan los datos de la entidad noticia.

**Bloque.php:** Es la entidad que contiene los datos de la tabla bloque

**Noticia.php:** Es la clase entidad que contiene la información de la tabla noticia

**BloqueNoticiaPublicada.php:** Es la clase entidad que contiene los id de las noticias y los bloques.

**noticia.html.twig:** Se encarga de visualizar la interfaz de publicar noticia.

**Noticia.js:** Archivo de JavaScript para el estilo de la pagina interfaz de Publicar Noticia.

**Doctrine2:** Es el ORM que utiliza Symfony2 para el mapeo de objetos.

### 4.3 Pruebas del Sistema

El único instrumento adecuado para determinar el status de la calidad de un producto de software es el proceso de pruebas. En este proceso se ejecutan pruebas dirigidas a componentes del software o al sistema de software en su totalidad, con el objetivo de medir el grado en que el software cumple con los requerimientos (21).

### 4.4 Pruebas unitarias

Cuando se desarrolla un software o solución informática, la primera etapa de pruebas a considerar aplicar por los analistas son las pruebas unitarias o también llamada pruebas de caja blanca (White Box). Son pruebas estructurales se basan en evaluar minuciosamente los detalles del código, siendo necesario conocer la lógica del programa. Es considerada como uno de los tipos de pruebas más importantes que se le aplican al software, logrando como resultado la disminución del número de errores existentes, aportando mayor calidad y confiabilidad al producto (21).

La técnica utilizada en este método es la prueba del camino básico, para ello es necesario conocer el número de caminos independientes de un determinado algoritmo mediante el cálculo de la complejidad ciclomática. Esta consiste en derivar casos de prueba a partir de un conjunto de caminos independientes por los cuales puede circular el flujo de control. Para ello es necesario construir el grafo de flujo asociado y calcular su complejidad ciclomática. Una vez identificados los caminos independientes se procede a preparar los casos de prueba que obliguen a la ejecución de cada camino del conjunto básico (21).

La complejidad ciclomática se calcula mediante tres formas:

$$V(G) = (A - N) + 2$$

$$V(G) = P + 1.$$

$$V(G) = R$$

Donde  $V(G)$  es la complejidad ciclomática de un grafo  $G$ , "A" la cantidad total de aristas, "N" la cantidad total de nodos, "P" la cantidad total de nodos predicados (son los nodos de los cuales parten dos o más aristas) y "R" la cantidad total de regiones (21).

A continuación se describe un ejemplo que muestra cómo se realiza este tipo de pruebas.

Se analizó la clase `NoticiaController.php`, específicamente el procedimiento `noticiaBloqueAction()`, luego se identificaron y se enumeró los bloques de ejecución como se muestra a continuación.

```

//asignar noticia al bloque//
public function noticiaBloqueAction() {
    $em = $this->getDoctrine()->getEntityManager();
    $peticion = $this->getRequest();
    $idNoticia = $peticion->get('idNoticia');
    $idBloques = $peticion->get('idBloques');

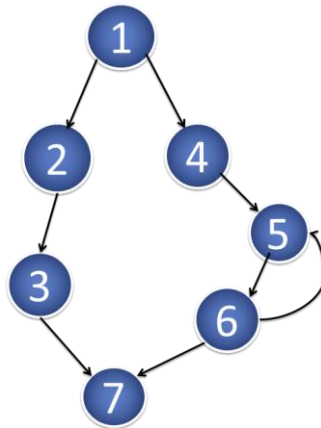
    if($idBloques)
        foreach ($idBloques as $bloque) {
            $nuevoBNP = new BloqueNoticiaPublicada();
            $nuevoBNP->setBloqueId($bloque);
            $nuevoBNP->setNoticiaId($idNoticia);
            $nuevoBNP->setOrden(1);
            $em->persist($nuevoBNP);
        }
    else {
        $noticiaspub = $em->getRepository('TransmisionBundle:BloqueNoticiaPublicada')->findAll();
        foreach ($noticiaspub as $noticiapub) {
            if($noticiapub->getNoticiaId() == $idNoticia)
                $em->remove($noticiapub);
        }
    }

    $em->flush();
    $response = new Response('{Success:true}');
    return $response;
}

```

**Figura 5** : Representación del código del método noticiaBloqueAction ().





**Figura 6:** Grafo del flujo asociado al método noticiaBloqueAction ().

Se logró obtener 20 nodos, se dibujó el grafo de flujo asociado y se determinó el camino básico. Como se muestra en el grafo de flujo las aristas indican los posibles caminos a seguir a partir del nodo correspondiente. Partiendo del camino básico determinado, fue aplicado uno de los tres métodos para calcular la complejidad ciclomática, específicamente el método  $V(G) = A - N + 2$ , aunque todos ellos arrojan al mismo resultado. Se obtuvieron 8 aristas y 7 nodos, quedando la fórmula de la siguiente forma:  $V(G) = 8 - 7 + 2$ . Por lo tanto la complejidad ciclomática tiene un valor de 3 lo cual significa que existen 6 posibles casos de ejecución para la función.

Número	Camino básico
1	1-2-3-7
2	1-2-4-5-6-7

**Tabla 4 :** Caminos básicos del flujo

#### **Caso de prueba para el primer camino básico:**

La confección del caso de prueba sobre el método noticiaBloqueAction (), se realizó con la utilización del grafo modelado anteriormente utilizando la técnica de camino básico. En este grafo se representaron 2 caminos de los que se obtiene diferentes resultados en su ejecución. Para el caso de

la ejecución del primer camino se pudo identificar que no se obtenían los datos esperados para la creación de una nueva instancia de noticiaBloque.php. Detectándose para el caso del nodo 3 un error al no estarse modificando el id de noticia en la entidad noticiaBloque.php. Para el camino restante, el método se efectuó de forma satisfactoria al no detectarse ningún error en el código.

#### **4.5 Prueba funcionales**

El método de pruebas funcionales o de caja negra, se centran en los requisitos funcionales del software y se llevan a cabo sobre la interfaz del software, obviando el comportamiento interno y la estructura del programa (21).

Dentro del método de caja negra la técnica de la partición equivalente es una de las más efectivas pues permite examinar los valores válidos e inválidos de las entradas existentes en el software (21).

La técnica de partición equivalente divide el campo de entrada de un programa en clases de datos de los que se pueden derivar casos de prueba. Un caso de prueba ideal descubre de forma inmediata una clase de errores que, de otro modo, requerirían la ejecución de muchos casos antes de detectar el error genérico. La partición equivalente se dirige a la definición de casos de prueba que descubran clases de errores, reduciendo así el número total de casos de prueba a desarrollar (21).

#### **Casos de Prueba**

Un caso de prueba es un conjunto de entradas, condiciones de ejecución y resultados esperados desarrollados para un objetivo particular. Los casos de pruebas se pueden derivar de los casos de usos del sistema o de la realización de estos en el modelo de diseño, permitiendo así validar los requerimientos funcionales del sistema (21).

En la prueba de caja negra, los casos de prueba pretenden demostrar que las funciones del software son operativas, que la entrada se acepta de forma adecuada y que se produce una salida correcta (21). Para el desarrollo de las pruebas al componente se desarrollaron cinco casos de pruebas, uno por cada caso de uso del sistema. Los casos de pruebas desarrollados se muestran a continuación (21):

Descripción del CU Publicar Noticia: El caso de uso inicia cuando el usuario accede a la sección de Noticia, mostrándose el listado de las noticias editadas sin publicar almacenadas en la base de datos. Termina al publicar la(s) noticia(s) o con un mensaje de proceso fallido.

#### **Caso de prueba: Publicar Noticia**

**Tabla 5 :** Caso de prueba del CU Publicar Noticia.

Escenario	Descripción	Nombre	Descripción	Permiso	Respuesta del sistema	Flujo central
EC 1.1 Asignar Bloque.	Se selecciona la opción Asignar Noticia. El sistema muestra un listado de todos los bloques que existen, a los que la noticia está asignada y los que no. Aparece un botón para publicar. El sistema valida la selección, de ser correctos publica la noticia.	V Usuario	V Este es un cliente de la plataforma	NA	Publica la noticia y muestra un mensaje de confirmación "La noticia ha sido publicada correctamente". El sistema actualiza en la base de datos la noticia publicada.	URL de acceso/ pulsar el en la sección deseada /botón Noticia/botón Asignar Bloque/ botón Publicar
EC 1.2 Eliminar Noticia	El sistema valida al menos una selección.	I Ninguna selección/ V Usuario	V Este es un cliente de la plataforma. I Campo Vacío	NA NA	El sistema elimina en la base de datos las noticias no publicadas que han sido seleccionadas.	URL de acceso/ pulsar en la sección deseada /botón Noticia/selecciona las noticias / botón Eliminar/ botón Aceptar. 48

EC 1.3 Asignar Bloque.	El sistema cancela la operación de asignar Bloque	NA	NA	NA	Cancela la operación y muestra la interfaz Publicar Noticia.	URL de acceso/ pulsar el en la sección deseada /botón Noticia/botón Asignar Bloque/ botón Cancelar
------------------------------	---	----	----	----	---	--

Para los restantes casos de pruebas [Anexo 6](#)

#### 4.6 Resultado de las pruebas

A continuación se detallan las no conformidades halladas durante las pruebas funcionales.

Los resultados de las pruebas de caja blanca aplicada a los diferentes métodos detectaron solo un error al asignar la noticia al bloque, puesto que no se recorrían todos los nodos del camino básico. Esta deficiencia fue resuelta en la primera iteración del proceso de desarrollo. Los restantes resultados de las pruebas al código fuente, fueron satisfactorios, ya que todos los nodos son visitados al menos una vez en alguno de los caminos básicos y todas las operaciones son válidas.

Luego de realizada las pruebas de caja negra en su primera iteración se obtuvo como resultado elementos que afectaban el buen funcionamiento de la aplicación, el error fue detectado ya que el sistema no validaba que estuvieran todos los campos llenos al adicionar una señal , un cambio programado y un bloque. Se resuelve el problema detectado, luego se realiza una segunda iteración donde se detecta que no se podían eliminar las señales que estaban siendo utilizadas en la sección cambios programados. Se resuelve el problema y para la tercera iteración se concluye que el sistema cumple con todos los requisitos planteados al inicio de la investigación logrando el correcto funcionamiento de todos los eventos. Ello se logra principalmente por las revisiones realizadas en cada una de las iteraciones en el proceso de desarrollo de software por parte de los revisores técnicos.

Una vez que se realizó un análisis detallado de cada uno de los resultados anteriormente expuestos se llega a la conclusión que se detectaron 5 no conformidades en las diferentes iteraciones al sistema las cuales fueron solucionadas. Concluyendo que la aplicación PRIMICIA Móvil no presenta ningún problema en cuanto a las funcionalidades del código fuente. Las pruebas de software devuelven un producto con la calidad requerida y que cumple con las especificidades del cliente.

### **Conclusiones parciales**

Se realizó el diagrama de despliegue donde se representaron los nodos y conexiones que intervienen en el funcionamiento del sistema, mostrando la configuración física sobre la que será desplegado el software. El diagrama de componentes realizado sirvió para visualizar la distribución física de estos, y sus relaciones, permitiendo tener una visión de la estructura del código fuente del sistema. La realización de las pruebas de caja negra y caja blanca permitió detectar, documentar y solucionar los errores existentes en el sistema implementado, dando como resultado que la aplicación cumple con los requisitos solicitados.

## **CONCLUSIONES GENERALES**

Durante el desarrollo del trabajo de diploma se obtuvieron conocimientos necesarios para el desarrollo del sistema. Una vez realizadas todas las tareas de investigación científicas, es posible afirmar que se han cumplido satisfactoriamente, todos los objetivos propuestos a lo largo del presente trabajo de diploma arribando a las siguientes conclusiones:

- Se desarrolló una aplicación que permite realizar la publicación de noticias desde dispositivos móviles.
- Las tecnologías y herramientas seleccionadas para el desarrollo del componente ofrecieron el soporte necesario para lograr un producto que cumpla con los requisitos.
- La implementación de la aplicación permitió realizar la administración del módulo de publicar noticias de la plataforma PRIMICIA a través de dispositivos móviles.

## RECOMENDACIONES

Teniendo en cuenta que es una primera versión del sistema, se recomienda:

- Incluir los módulos restantes de Primicia a la aplicación móvil.
- Realizar mejoras a la apariencia visual del sistema.
- Realizar un manual de usuario para un mejor entendimiento del sistema.

## TRABAJOS CITADOS

1. **Solivérez, C. E.** *Tecnológica para comprender el fenómeno tecnológico*. Buenos Aires Argentina : Instituto Nacional de Educación Técnica, 2003. 1.
2. **Silva, Ricardo Nieto.** *Internet*. Madrid, España : 1ª ed, 1ª imp, 2002. 3.
3. *Protocolos de comunicaciones*. **Suyama, Maria**. Sao Paulo, Brasil : ed 4ta, 2011. 4.
4. **Guido, Luciana Mónica.** *Campus Vituales*. Buenos Aires. Argentina : s.n., 2010. 1.
5. [www.reddeaprendizaje.com](http://www.reddeaprendizaje.com). [En línea] Joomla, 6 de 12 de 2010. [Citado el: 8 de 3 de 2013.] <http://www.reddeaprendizaje.com/inicio/item/47-plataforma-informatica>. . 5.
6. **Stallings, William.** *Sistemas Operativos*. España : P R E N T I C E H A L L, 2000. 6.
7. *Asignatura de Tecnologías de Sistemas Informáticos*. **Ford, Andrew**. Madrid. : Universidas Politécnica , 2012. 7.
8. **Michael Angel López.** *ArticuloZ.com*. [En línea] 9 de 3 de 2005-2013. [Citado el: 2 de 11 de 2010.] <http://www.articuloz.com/informatica-y-tecnologia-articulos/aplicaciones-informaticas-3753302.html>. . 8.
9. *Dispositivos Moviles*. **Villanueva, Carlos**. 2011. 11.
10. **Smith, Jonh.** [www.informatica-hoy.com.ar](http://www.informatica-hoy.com.ar). [En línea] [www.informatica-hoy.com.ar/soluciones-moviles/Los-mejores-navegadores-para-celulares.php](http://www.informatica-hoy.com.ar/soluciones-moviles/Los-mejores-navegadores-para-celulares.php). 19.
11. **Velásquez Pastrana, Lourdes.** *Generaciones en la Telefonía Celular*. [En línea] 2012. [Citado el: 20 de 9 de 2013.] <http://www.enterate.unam.mx/Articulos/2013/agosto/telefcel.htm...> 14.
12. **Pérez, Pedro Navarro.** *W A P: Wireless application protocol*. 2010. 17.
13. **Warner, Janine.** [www.redusers.com](http://www.redusers.com). [En línea] 2012. [www.redusers.com/so-moviles/](http://www.redusers.com/so-moviles/). 18.
14. *Aplicaciones Web vs Aplicaciones de Escritorio*. **Angelica, Bexton**. Argentina. : s.n., 2012. 10.
15. [www.techbloog.com](http://www.techbloog.com). [En línea] [http://www.techbloog.com/talks/netbeans65es\\_cl.pdf](http://www.techbloog.com/talks/netbeans65es_cl.pdf). 30.. 38.
16. *Cubacel*. [Cubacel.com](http://www.cubacel.com). [En línea] [Citado el: 2 de 11 de 2012.] [www.cubacel.com/telefoniacelular..](http://www.cubacel.com/telefoniacelular..) 15.
17. **Venzant, Maikel Bradshaw.** *Análisis, diseño e implementación de Portal WAP , SIIPOL Móvil*. Habana : UCI, 2010.



18. **Alierta, César.** Movistar. Telefónica S.A,. [En línea] 2007. [Citado el: 10 de 1 de 2010.] [www.movistar.com.pa/informe2007.pdf](http://www.movistar.com.pa/informe2007.pdf).. 13.
19. *www.Infor.com.* **Calidad de Producción.** 2012. 25.
20. **Booch Grady, Jacobson Ivar, Rumbaugh James.** *El Proceso Unificado de Desarrollo de Software.* 2010. 26.
21. **Pressman, Roger S.** *Ingeniería de Software, Un enfoque práctico.*2010. 40.
22. **Larman, Craig.** *UML y Patrones Introducción al análisis y diseño orientado a objetos.* Mexico : s.n., 2010. 46.
23. *Introducción a la herramienta CASE.* **Asignatura Gestión de Software.** Habana : s.n., 2012. 29.
24. **Alvarez, Rubén.** *Programación en PHP.* . 2009. 32.
25. **Peter Lubbers, Brian Albers.** *Programando en HTML5.* . 2010. . 32.
26. **Pérez, Javier Eguíluz.** *Introducción a JavaScript.* 2008. 33.
27. **Eguíluz, Javier.** *Introducción a CSS.* 2010. 34.
28. *Aplicaciones Infomáticas. Desarrollo.* **Hamilton, Erick.** Canada. : s.n., 2011. 9.
29. **Eguiluz, Javier.** *El framework Symfony, una introducción práctica.* 2010. 29. 36.
30. **Pérez, Javier Eguíluz.** *Introducción a jQueryMobile.* . 2012. 38.
31. **Garcerant, Iván.** synergix.com. [En línea] <http://synergix.wordpress.com/2008/07/10/modelo-de-dominio/>. 39.
32. **Larman, Craig.** **UML.** *Una introducción al análisis y diseño del proceso unificado.* s.l. : 2da edic, 2009. 47.
33. Buenas Tareas.com. [En línea] <http://www.buenastareas.com/ensayos/Patrones..> 44.
34. **Corporación IBM.** *RUP., Ayuda.* 2008. 41.
35. *UML y Patrones.* **Larman, Craig.** La Habana : Editorial Felix Varela, 2004. . 27.

## BIBLIOGRAFÍA

1. **Solivérez, C. E.** *Tecnológica para comprender el fenómeno tecnológico*. Buenos Aires Argentina : Instituto Nacional de Educación Técnica, 2003. 1.
2. **Silva, Ricardo Nieto.** *Internet*. Madrid, España : 1ª ed, 1ª imp, 2002. 3.
3. *Protocolos de comunicaciones*. **Suyama, Maria**. Sao Paulo, Brasil : ed 4ta, 2011. 4.
4. **Guido, Luciana Mónica.** *Campus Vituales*. Buenos Aires. Argentina : s.n., 2010. 1.
5. [www.reddeaprendizaje.com](http://www.reddeaprendizaje.com). [En línea] Joomla, 6 de 12 de 2010. [Citado el: 8 de 3 de 2013.] <http://www.reddeaprendizaje.com/inicio/item/47-plataforma-informatica>. . 5.
6. **Stallings, William.** *Sistemas Operativos*. España : P R E N T I C E H A L L, 2000. 6.
7. *Asignatura de Tecnologías de Sistemas Informáticos*. **Ford, Andrew**. Madrid. : Universidas Politécnica , 2012. 7.
8. **Michael Angel López.** *ArticuloZ.com*. [En línea] 9 de 3 de 2005-2013. [Citado el: 2 de 11 de 2010.] <http://www.articuloz.com/informatica-y-tecnologia-articulos/aplicaciones-informaticas-3753302.html>. . 8.
9. *Dispositivos Moviles*. **Villanueva, Carlos**. 2011. 11.
10. **Smith, Jonh.** [www.informatica-hoy.com.ar](http://www.informatica-hoy.com.ar). [En línea] [www.informatica-hoy.com.ar/soluciones-moviles/Los-mejores-navegadores-para-celulares.php](http://www.informatica-hoy.com.ar/soluciones-moviles/Los-mejores-navegadores-para-celulares.php). 19.
11. **Velásquez Pastrana, Lourdes.** *Generaciones en la Telefonía Celular*. [En línea] 2012. [Citado el: 20 de 9 de 2013.] <http://www.enterate.unam.mx/Articulos/2013/agosto/telefcel.htm...> 14.
12. **Pérez, Pedro Navarro.** *W A P: Wireless application protocol*. 2010. 17.
13. **Warner, Janine.** [www.redusers.com](http://www.redusers.com). [En línea] 2012. [www.redusers.com/so-moviles/](http://www.redusers.com/so-moviles/). 18.
14. *Aplicaciones Web vs Aplicaciones de Escritorio*. **Angelica, Bexton**. Argentina. : s.n., 2012. 10.
15. [www.techblog.com](http://www.techblog.com). [En línea] [http://www.techblog.com/talks/netbeans65es\\_cl.pdf](http://www.techblog.com/talks/netbeans65es_cl.pdf). 30.. 38.
16. *Cubacel*. [Cubacel.com](http://Cubacel.com). [En línea] [Citado el: 2 de 11 de 2012.] [www.cubacel.com/telefoniacelular..](http://www.cubacel.com/telefoniacelular..) 15.
17. **Venzant, Maikel Bradshaw.** *Análisis, diseño e implementación de Portal WAP , SIIPOL Móvil*. Habana : UCI, 2010.

18. Alierta, César. Movistar. Telefónica S.A,. [En línea] 2007. [Citado el: 10 de 1 de 2010.] [www.movistar.com.pa/informe2007.pdf](http://www.movistar.com.pa/informe2007.pdf). 13.
19. [www.Infor.com](http://www.Infor.com). Calidad de Producción. 2012. 25.
20. Booch Grady, Jacobson Ivar, Rumbaugh James. El Proceso Unificado de Desarrollo de Software. 2010. 26.
21. Pressman, Roger S. Ingeniería de Software, Un enfoque práctico.2010. 40.
22. Larman, Craig. UML y Patrones Introducción al análisis y diseño orientado a objetos. Mexico : s.n., 2010. 46.
23. Introducción a la herramienta CASE. Asignatura Gestión de Software. Habana : s.n., 2012.
24. Alvarez, Rubén. Programación en PHP. . 2009. 32.
25. Peter Lubbers, Brian Albers. .Progrmando en HTML5. . 2010. . 32.
26. Pérez, Javier Eguíluz. Introducción a JavaScript. 2008. 33.
27. Eguíluz, Javier. Introduccion a CSS. 2010. 34.
28. Aplicaciones Infórmaticas. Desarrollo. Hamilton, Erick. Canada. : s.n., 2011. 9.
29. Eguiluz, Javier. El framework Symfony, una introducción práctica. 2010. 29. 36.
30. Pérez, Javier Eguíluz. Introducción a jQueryMobile. . 2012. 38.
31. Garcerant, Iván. synergix.com. [En línea] <http://synergix.wordpress.com/2008/07/10/modelo-de-dominio/>. 39.
32. Larman, Craig. UML. Una introducción al análisis y diseño del proceso unificado. s.l. : 2da edic, 2009. 47.
33. Buenas Tareas.com. [En línea] <http://www.buenastareas.com/ensayos/Patrones..> 44.
34. Corporación IBM. RUP., Ayuda . 2008. 41.
35. UML y Patrones. Larman, Craig. La Habana : Editorial Felix Varela, 2004. . 27.0

## ANEXOS

### Anexo 1

Fecha: 12/11/2012

Entrevista

Realizado por:

Nombre y Apellidos	Cargo
Patricia Herrera Palacios	Estudiante

### Entrevistados

Nombre y Apellidos	Cargo
Carlos de Jesús Andrés González	Líder de proyecto
Félix Iván Romero Rodríguez	Desarrollador
Rafael Lorente Miranda	Desarrollador

### Objetivo:

Obtener información sobre el funcionamiento actual de los procesos transmisión y administración en PRIMICIA.

**Pregunta 1:** ¿Cómo funciona actualmente el subsistema de transmisión de PRIMICIA?

**Pregunta 2:** ¿Cómo están constituidas las noticias?

**Pregunta 3:** ¿Qué elementos se tienen en cuenta para la publicación de noticias?

**Pregunta 4:** ¿Qué información se registra en la BD de las noticias?

## **GLOSARIO**

**Wi-Fi:** Tecnología de comunicación inalámbrica mediante ondas, también llamada WLAN (Wirelesslan, red inalámbrica). Necesita un punto de acceso, que se conectaría al módem, y un dispositivo WIFI que se conectaría en nuestro aparato.

**GPRS:** Servicio general de paquetes vía radio, que utiliza el concepto de comunicación por paquetes. Solo se necesita un dispositivo móvil que soporte esta tecnología inalámbrica.

**Bluetooth:** Redes Inalámbricas de Área Personal (WPAN) que posibilita la transmisión de voz y datos entre diferentes dispositivos que requieren un corto alcance de emisión.

**iPhone:** es un teléfono inteligente multimedia con conexión a Internet, pantalla táctil y una interfaz de hardware minimalista.

**Tablet PC:** Es una computadora a medio camino entre una computadora portátil y un PDA, en el que se puede escribir a través de una pantalla táctil. Un usuario puede utilizar un estilete para trabajar con el ordenador sin necesidad de teclado o ratón. Este aparato fue propugnado por Microsoft y otros fabricantes.

**iPad:** Es un dispositivo electrónico de tipo Tablet PC desarrollado por la Apple ,su clasificación está entre las categorías de “Teléfono inteligente” y una computadora portátil, su implementación está más ligada al acceso a la información, la navegación y la lectura digital que a la creación de contenido.

**API:** Interfaz de Programación de Aplicaciones: es una interfaz de comunicación entre componentes de software. Uno de los principales consiste en proporcionar un conjunto de funciones de uso general además de representar un método para conseguir abstracción en la programación.

**HTTP:** Es un protocolo orientado a transacciones y sigue el esquema petición-respuesta entre un cliente y un servidor. El cliente efectúa la petición en un navegador web. A la información transmitida se la llama recurso y se la identifica mediante un localizador uniforme de recursos (URL). Los recursos pueden ser archivos, el resultado de la ejecución de un programa, una consulta a una base de datos, la traducción automática de un documento, enviados desde el servidor web.

**Interfaz de usuario:** es el medio con que el usuario puede comunicarse con una máquina, un equipo o una computadora, y comprende todos los puntos de contacto entre el usuario y el equipo. Normalmente suelen ser fáciles de entender y fáciles de accionar.

**Hipertexto:** Es el nombre que recibe el texto que en la pantalla de un dispositivo electrónico conduce a otro texto relacionado. La forma más habitual de hipertexto en informática es la de hipervínculos o referencias cruzadas automáticas que van a otros documentos.

**World Wide Web Consortium (W3C):** Es una comunidad internacional donde la organización miembro, un equipo de trabajo a tiempo completo, y el público, trabajan de conjunto para desarrollar estándares web.

**HTML:** Lenguaje de marcado diseñado para estructurar textos y presentarlos en forma de hipertexto, que es el formato estándar de las páginas web.

**XHTML** (Lenguaje xtensible de marcado de hipertexto en ingles eXtensible Hypertext Markup Language): Es el lenguaje de marcado pensado para sustituir a HTML como estándar para las Páginas Web.

**XML** (eXtensible Markup Language): Es un lenguaje de marcado sencillo similar al HTML. Su objetivo es facilitar la representación, almacenamiento y transmisión de información varia por parte de aplicaciones informáticas, computadoras y medios de comunicación digital en general.

**jQuery:** consiste en un único fichero JavaScript que contiene las funcionalidades comunes de DOM, eventos, efectos y AJAX. Es una biblioteca o Framework de JavaScript.

**Módulo:** Es una pieza o un conjunto unitario de piezas que se repiten en una construcción de cualquier tipo, para hacerla más fácil, regular y económica. El módulo, forma parte de un sistema y mantiene algún tipo de relación o vínculo con el resto de los componentes.

**Subsistema:** Los subsistemas nos permiten dividir el sistema entero en partes más manejables y fáciles de entender. Cada subsistema abarca aspectos del sistema que comparten alguna propiedad común, y es por eso que decimos que es un conjunto de elementos ordenados o funciones relacionados para cumplir con un propósito o fin determinado y cuyas partes deben reunir ciertas condiciones de tal manera que se complementen formando el sistema.

**Hardware:** corresponde a todas las partes físicas y tangibles de una computadora.

