

# Intranet 2.0. Módulo Servicios de Postgrado

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



***Autor***

Alexander Fernando Vinent Peña

***Tutor***

Ing. Jorge Luis Tamarit Cutiño

***Asesores***

Ing. Yunaisi Renté Vázquez

Ing. Alberto Tamayo Ramos

CIUDAD DE LA HABANA, JUNIO 2009

## DECLARACIÓN DE AUTORÍA

Declaro que soy el único autor de este trabajo y autorizo a las direcciones de Formación Postgraduada; y a la de Informatización de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

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

\_\_\_\_\_  
Firma del Autor

\_\_\_\_\_  
Firma del Tutor



“Learn thoroughly what you learn;

let your conduct be worthy of what is learnt.”

Verso 391 de *Thirukural*, Collection of 1330 noble couplets

Thiruvalluvar, poeta y filósofo, 31 B.C.



## AGRADECIMIENTOS

A mi madre. Nunca viviré los años suficientes para agradecerle todo lo que ha hecho por mí. Darme la vida fue lo más grande pero enseñarme a vivir, ha sido su enseñanza eterna.

A mis amigos por haber compartido conmigo cada momento, por llorar juntos, por tenerme la confianza y contarme sus penas, por creer en mí, por hacerme aceptar mis errores, y hacer que me riese de mí en algún momento, después de explicarme las bromas que tiene la vida y que yo no lograba entender. Nosotros somos familia, porque pienso que *alguien en algún sitio* nos dio la oportunidad de escoger a una nueva y esa la que formamos con nuestros amigos.

¡GRACIAS HERMANO!

Alberto Tamayo Ramos (Bayamo), Albey Alfonso Navarro (La Habana), Miguel J. Rodríguez Lazo (Stgo.), Jorge L. Tamarit Cutiño (Colombia, LTU), Misael Fonseca Mata (Camagüey), Otto A. Padró Maurelo (Stgo.), Pedro Peña (Pinar), Alfredo (Stgo.), Yadier F. Pérez Litán (Habana Vieja), Alejandro Sánchez Ramírez (Bayamo), Yasser León Montes de Oca (Guanabo), Ernesto Tur (Baracoa) y Yunior E. Guerrero (Palma Soriano).

A la profesora de inglés, Bertha Elena.

A los que no confiaron...

*Dios sólo entiende de binario...*

Aleck



## RESUMEN

El desarrollo de nuevas tecnologías y/o técnicas de comunicación con el usuario a través de la Web, unido al incremento de los servicios que se prestan dentro de la intranet universitaria, conllevaron a la concepción e implementación del sistema *Intranet 2.0*; el cual será el punto de acceso a los servicios brindados por los diferentes sistemas de gestión dentro de la Intranet UCI. De ahí, que se considere necesaria la inclusión de componentes o servicios de estos sistemas que sirvan de enlace dentro de la nueva aplicación. Uno de dichos componentes o módulos es el relacionado con la Dirección de Formación Postgraduada de la Universidad de las Ciencias Informáticas (UCI). Como respuesta a la problemática planteada, se propone el *Módulo de Servicios de Postgrado de la Intranet 2.0*.

En el presente documento, se propone el desarrollo ágil de un componente para la *Intranet 2.0* que posibilite el acceso a los servicios públicos del sistema de Postgrado de la UCI. Dentro de las acciones a trabajar se encuentran: desarrollar el servicio web de dicho sistema; y crear un componente que posibilite la integración de los servicios brindados por el sistema de Formación Postgraduada UCI dentro del CMS Drupal.

Para guiar y controlar el proceso de desarrollo, se utilizó la metodología *MA-GMPR-UR2 o Metodología ágil para proyectos de software libre*, la cual está basada en las metodologías ágiles SCRUM y XP.

### **Palabras clave:**

Desarrollo ágil, desarrollo Web, Drupal, Intranet 2.0, módulo Drupal, SCRUM, XP.



## ÍNDICE

DECLARACIÓN DE A UTORÍA .....	2
AGRADECIMIENTOS .....	4
RESUMEN.....	5
ÍNDICE.....	6
ÍNDICE DE FIGURAS .....	8
ÍNDICE DE TABLAS .....	9
INTRODUCCIÓN .....	11
CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA .....	14
1.1. INTRODUCCIÓN.....	14
1.2. ANTECEDENTES .....	14
1.2.1. TECNOLOGÍAS EN LA WEB 2.0 .....	15
1.2.2. SOA.....	16
1.2.2.1. BENEFICIOS DE SOA .....	16
1.3. TENDENCIAS DE LAS TECNOLOGÍAS EN LAS POLÍTICAS DE INFORMATIZACIÓN .....	17
1.3.1. SERVICIO WEB .....	17
1.3.1.1. VENTAJAS DE LOS SERVICIOS WEB .....	19
1.3.1.2. ALTERNATIVAS DE IMPLEMENTACIÓN.....	19
1.3.2. XML .....	20
1.3.3. SOAP.....	21
1.3.4. WSDL.....	21
1.3.5. PHP.....	22
1.3.6. EXTENSIONES.....	24
1.3.6.1. SOAP.....	24
1.3.6.2. PDO.....	24
1.3.7. SISTEMA GESTIÓN DE CONTENIDOS .....	25
1.3.7.1. DRUPAL.....	26
1.3.7.1.1. DESARROLLO DE MÓDULOS EN DRUPAL .....	27
1.3.8. HERRAMIENTAS, LENGUAJES Y METODOLOGÍAS DE DESARROLLO .....	28
1.3.8.1. ZEND STUDIO FOR ECLIPSE.....	28
1.3.8.2. METODOLOGÍA DE DESARROLLO DE SOFTWARE.....	30
1.3.8.3. METODOLOGÍAS ÁGILES .....	31
1.3.8.3.1. SCRUM .....	32
1.3.8.3.2. EXTREME PROGRAMMING.....	33



1.3.8.4.	UML .....	35
1.3.8.5.	HERRAMIENTA CASE .....	35
1.3.8.5.1.	VISUAL PARADIGM .....	36
1.4.	CONCLUSIONES .....	38
CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA. ....		39
2.1.	INTRODUCCION .....	39
2.2.	OBJETO DE ESTUDIO .....	39
2.2.1.	INTRANET 2.0 .....	39
2.3.	VALORACIÓN DISEÑO PROPUESTO POR EL SISTEMA DE POSTGRADO UCI .....	40
2.4.	PROPUESTA DE SISTEMA .....	41
2.5.	ESPECIFICACIÓN DE LOS REQUISITOS DE SOFTWARE .....	42
2.5.1.	PILA DE PRODUCTO .....	43
2.5.1.1.	HISTORIAS TÉCNICAS .....	44
2.6.	DEFINICIÓN DE LAS HISTORIAS DE USUARIO .....	45
2.6.1.	HISTORIAS DE USUARIO.....	45
2.7.	HISTORIAS DE USUARIO POR ITERACIÓN .....	46
CAPÍTULO 3. DISEÑO DEL SISTEMA PROPUESTO. ....		47
3.1.	HISTORIAS DE USUARIO Y PROTOTIPOS UI .....	47
3.1.1.	HU SOLICITAR RESERVACIÓN DE LOCALES DE POSGRADO .....	47
3.1.2.	HU MOSTRAR RESERVACIONES DE LOCALES DE POSGRADO.....	50
3.1.3.	HU FILTRADO DE RESERVACIONES .....	53
3.1.4.	HU GESTIÓN DE SOLICITUDES .....	57
3.1.4.1.	HU GESTIÓN DE SOLICITUDES. MODIFICAR SOLICITUD .....	58
3.1.4.2.	HU GESTIÓN DE SOLICITUDES. ELIMINAR SOLICITUD.....	61
3.1.5.	HU GESTIÓN DE RESERVACIONES .....	63
3.1.5.1.	HU GESTIÓN DE RESERVACIONES. DETALLES DE RESERVACIÓN .....	64
3.1.5.2.	HU GESTIÓN DE RESERVACIONES. ADICIONAR RESERVACIÓN .....	65
3.1.5.3.	HU GESTIÓN DE RESERVACIONES. MODIFICAR RESERVACIÓN.....	68
3.1.5.4.	HU GESTIÓN DE RESERVACIONES. ELIMINAR RESERVACIÓN .....	70
3.1.6.	HU GESTIÓN DE LOCALES .....	73
3.1.6.1.	HU GESTIÓN DE RESERVACIONES. ADICIONAR LOCAL .....	74
3.1.6.2.	HU GESTIÓN DE LOCALES. MODIFICAR LOCAL.....	76
3.1.6.3.	HU GESTIÓN DE LOCALES. ELIMINAR LOCAL.....	79
3.2.	MODELO DE DESPLIEGUE.....	82
3.3.	DIAGRAMA DE COMPONENTES .....	82



3.4.	DESCRIPCIÓN DE PAQUETES .....	83
CAPÍTULO 4. VALIDACIÓN DE LA SOLUCIÓN PROPUESTA. ....		85
4.1.	INTRODUCCION.....	85
4.2.	CASOS DE PRUEBA .....	85
4.2.1.	CASO DE PRUEBA INSTALACIÓN DEL MÓDULO SERVICIOS DE POSTGRADO .....	86
4.2.2.	CASO DE PRUEBA HISTORIA DE USUARIO HU.1 .....	87
4.2.3.	CASO DE PRUEBA HISTORIA DE USUARIO HU.2 .....	88
4.2.4.	CASO DE PRUEBA HISTORIA DE USUARIO HU.3 .....	89
4.2.5.	CASO DE PRUEBA HISTORIA DE USUARIO HU.4 .....	89
4.2.6.	CASO DE PRUEBA HISTORIA DE USUARIO HU.5 .....	90
4.2.7.	CASO DE PRUEBA HISTORIA DE USUARIO HU.6 .....	91
4.2.8.	CASO DE PRUEBA HISTORIA DE USUARIO HU.7 .....	91
4.2.9.	CASO DE PRUEBA HISTORIA DE USUARIO HU.8 .....	92
4.2.10.	CASO DE PRUEBA HISTORIA DE USUARIO HU.9 .....	93
4.2.11.	CASO DE PRUEBA HISTORIA DE USUARIO HU.10 .....	93
4.2.12.	CASO DE PRUEBA HISTORIA DE USUARIO HU.11 .....	94
4.2.13.	CASO DE PRUEBA HISTORIA DE USUARIO HU.12 .....	95
4.2.14.	CASO DE PRUEBA HISTORIA DE USUARIO HU.13 .....	96
4.2.15.	CASO DE PRUEBA HISTORIA DE USUARIO HU.14 .....	96
4.2.16.	CASO DE PRUEBA HISTORIA DE USUARIO HU.15 .....	97
4.3.	RESULTADOS OBTENIDOS .....	98
4.4.	CONCLUSIONES .....	98
CONCLUSIONES GENERALES .....		99
RECOMENDACIONES .....		100
BIBLIOGRAFÍA CONSULTADA .....		101
ANEXOS .....		103
ANEXO I. PLANTILLA DESCRIPCION DE LAS HISTORIAS DE USUARIO .....		103
ANEXO II. PLANTILLA DESCRIPCION DE TAREA DE INGENIERÍA .....		103
ANEXO III PLANTILLA CASO DE PRUEBA DE ACEPTACIÓN .....		104
ANEXO IV RESULTADOS DE PRUEBAS DE ACEPTACIÓN .....		104
GLOSARIO DE TÉRMINOS .....		105

## ÍNDICE DE FIGURAS





Fig. 1 Esquema general de un servicio Web .....	18
Fig. 2 Módulos para Drupal .....	28
Fig. 3 Representación de un proyecto basado en la metodología XP .....	34
Fig. 4 Representación de las fases de la Metodología XP .....	34
Fig. 5 Evolución hacia Intranet 2.0 .....	40
Fig. 6 Propuesta de sistema .....	42
Fig. 7 Diagrama de despliegue .....	82
Fig. 8 Diagrama de componentes .....	82
Fig. 9 Diagrama de componentes. Paquete Módulo Servicios de Postgrado .....	83
Fig. 10 Diagrama de componentes. Paquete Servicios Web .....	83
Fig. 11 Resultado de la Instalación del componente en el entorno CMS DRUPAL V 6.X .....	104

## ÍNDICE DE TABLAS

Tabla 1 Diferencias entre metodologías ágiles y no ágiles .....	31
Tabla 2 Historias de Usuario .....	43
Tabla 3 Historias de Usuario .....	46
Tabla 4 Historias de Usuario por iteración .....	46
Tabla 5 HU Solicitar Reservación de Locales de Postgrado .....	48
Tabla 6 Tarea de Ingeniería: Crear funcionalidad para registrar solicitud mediante servicio web. ....	49
Tabla 7 Tarea de Ingeniería: Crear funcionalidad para registrar solicitud mediante módulo de Drupal. .	49
Tabla 8 Tarea de Ingeniería: Diseño UI de la funcionalidad registrar solicitud. ....	50
Tabla 9 Tarea de Ingeniería: Notificar usuario. ....	50
Tabla 10 HU Mostrar Reservaciones de Locales de Posgrado .....	51
Tabla 11 Tarea de Ingeniería: Crear funcionalidad para devolver los registros correspondientes a las reservaciones vía servicio web. ....	52
Tabla 12 Tarea de Ingeniería: Crear funcionalidad para devolver los registros correspondientes a las reservaciones mediante módulo de Drupal. ....	52
Tabla 13 Tarea de Ingeniería: Diseño UI de la funcionalidad Mostrar Reservaciones de Locales de Posgrado. ....	53
Tabla 14 HU Filtrado de reservaciones .....	54
Tabla 15 Tarea de Ingeniería: Filtro por locales. Servicio Web .....	54
Tabla 16 Tarea de Ingeniería: Filtro por sesión de trabajo. Servicio Web .....	54
Tabla 17 Tarea de Ingeniería: Filtro por responsable. Servicio Web .....	55
Tabla 18 Tarea de Ingeniería: Filtro por responsable. Módulo Drupal .....	55
Tabla 19 Tarea de Ingeniería: Filtro por sesión de trabajo. Módulo Drupal .....	56
Tabla 20 Tarea de Ingeniería: Filtro por responsable. Módulo Drupal .....	56
Tabla 21 Tarea de Ingeniería. Diseño UI Filtro de reservaciones .....	57
Tabla 22 HU Gestión de solicitudes .....	58
Tabla 23 HU Gestión de solicitudes. Modificar solicitud .....	59
Tabla 24 Tarea de Ingeniería: Crear funcionalidad para modificar la solicitud vía servicio web. ....	59
Tabla 25 Tarea de Ingeniería: Crear funcionalidad para Modificar solicitud mediante módulo de Drupal. ....	60
Tabla 26 Tarea de Ingeniería: Diseño UI de la funcionalidad Modificar solicitud. ....	60
Tabla 27 Tarea de Ingeniería: Notificar usuario. ....	61
Tabla 28 HU Gestión de solicitudes. Eliminar solicitud .....	61
Tabla 29 Tarea de Ingeniería: Crear funcionalidad para eliminar la solicitud vía servicio web. ....	62
Tabla 30 Tarea de Ingeniería: Crear funcionalidad para eliminar solicitud mediante módulo de Drupal. ....	62
Tabla 31 Tarea de Ingeniería: Diseño UI de la funcionalidad Eliminar solicitud. ....	63
Tabla 32 Tarea de Ingeniería: Notificar usuario. ....	63
Tabla 33 HU Gestión de solicitudes .....	64
Tabla 34 HU Gestión de reservaciones. Detalles de la reservación. ....	65
Tabla 35 HU Gestión de solicitudes. Adicionar reservación. ....	66
Tabla 36 Tarea de Ingeniería: Crear funcionalidad para registrar solicitud mediante servicio web. ....	66



Tabla 37 Tarea de Ingeniería: Crear funcionalidad para registrar solicitud mediante módulo de Drupal.	67
Tabla 38 Tarea de Ingeniería: Diseño UI de la funcionalidad registrar solicitud.	67
Tabla 39 Tarea de Ingeniería: Notificar usuario.	68
Tabla 40 HU Gestión de reservaciones. Modificar reservación	68
Tabla 41 Tarea de Ingeniería: Crear funcionalidad para modificar la reservación vía servicio web.	69
Tabla 42 Tarea de Ingeniería: Crear funcionalidad para Modificar reservación mediante módulo de Drupal.	69
Tabla 43 Tarea de Ingeniería: Diseño UI de la funcionalidad Modificar reservación.	70
Tabla 44 Tarea de Ingeniería: Notificar usuario.	70
Tabla 45 HU Gestión de reservaciones. Eliminar reservación	71
Tabla 46 Tarea de Ingeniería: Crear funcionalidad para eliminar la reservación vía servicio web.	72
Tabla 47 Tarea de Ingeniería: Crear funcionalidad para eliminar reservación mediante módulo de Drupal.	72
Tabla 48 Tarea de Ingeniería: Diseño UI de la funcionalidad Eliminar reservación.	72
Tabla 49 Tarea de Ingeniería: Notificar usuario.	73
Tabla 50 HU Gestión de locales	74
Tabla 51 HU Gestión de locales. Adicionar local.	75
Tabla 52 Tarea de Ingeniería: Crear funcionalidad para registrar local mediante servicio web.	75
Tabla 53 Tarea de Ingeniería: Crear funcionalidad para adicionar un local mediante módulo de Drupal.	76
Tabla 54 Tarea de Ingeniería: Diseño UI de la funcionalidad adicionar local.	76
Tabla 55 HU Gestión de locales. Modificar local	77
Tabla 56 Tarea de Ingeniería: Crear funcionalidad para modificar local vía servicio web.	77
Tabla 57 Tarea de Ingeniería: Crear funcionalidad para Modificar local mediante módulo de Drupal.	78
Tabla 58 Tarea de Ingeniería: Diseño UI de la funcionalidad Modificar local.	78
Tabla 59 Tarea de Ingeniería: Notificar usuario.	79
Tabla 60 HU Gestión de reservaciones. Eliminar reservación	80
Tabla 61 Tarea de Ingeniería: Crear funcionalidad para eliminar local vía servicio web.	80
Tabla 62 Tarea de Ingeniería: Crear funcionalidad para eliminar local mediante módulo de Drupal.	81
Tabla 63 Tarea de Ingeniería: Diseño UI de la funcionalidad Eliminar local.	81
Tabla 64 Tarea de Ingeniería: Notificar usuario.	82
Tabla 65 Caso de Prueba a la instalación no.1	86
Tabla 66 Caso de Prueba a la instalación no.2	87
Tabla 67 Caso de Prueba a HU1	87
Tabla 68 Caso de Prueba a HU1	88
Tabla 69 Caso de Prueba a HU1	88
Tabla 70 Caso de Prueba a HU3	89
Tabla 71 Caso de Prueba a HU4	90
Tabla 72 Caso de Prueba a HU5	90
Tabla 73 Caso de Prueba a HU5	91
Tabla 74 Caso de Prueba a HU6	91
Tabla 75 Caso de Prueba a HU7	92
Tabla 76 Caso de Prueba a HU8	93
Tabla 77 Caso de Prueba a HU9	93
Tabla 78 Caso de Prueba a HU10	94
Tabla 79 Caso de Prueba a HU10	94
Tabla 80 Caso de Prueba a HU11	95
Tabla 81 Caso de Prueba a HU12	96
Tabla 82 Caso de Prueba a HU13	96
Tabla 83 Caso de Prueba a HU14	97
Tabla 84 Caso de Prueba a HU14	97
Tabla 85 Caso de Prueba a HU15	98



## INTRODUCCIÓN

Cuba ha identificado la necesidad y conveniencia de dominar e introducir los servicios en la red como parte de la práctica social. Esta determinación va encaminada a lograr una cultura digital imprescindible, logrando acercar a la sociedad a un desarrollo cada vez más sostenible.

A la par del auge de la informática a nivel mundial, en Cuba, la Universidad de las Ciencias Informáticas (UCI), juega un papel cada vez más importante, y desde sus inicios ha desempeñado la intensa tarea de informatizar varios sectores de la sociedad. Aparejado al progreso de la institución, se ha llevado un proceso de digitalización de innumerables servicios internos con diversos fines, como la Intranet Universitaria, Sistema de Opiniones, *Akaderos*, Directorio de Personas, entre otros.

El desarrollo de nuevas tecnologías y/o técnicas de comunicación con el usuario a través de la Web, unido al incremento de los servicios que se prestan dentro de la intranet universitaria, conllevaron a la concepción e implementación del sistema denominado *Intranet 2.0*; el cual será el punto de acceso a los servicios brindados por los diferentes sistemas de gestión dentro de la Intranet UCI. De ahí, que se considere necesario la inclusión de componentes o servicios de estos sistemas, que sirvan de enlace dentro de la nueva aplicación. Uno de dichos componentes o módulos es el relacionado con la Dirección de Formación Postgraduada de la UCI. Como respuesta a la problemática planteada, se propone el *Módulo de Servicios de Postgrado de la Intranet 2.0*.

Analizando y teniendo en cuenta lo planteado anteriormente, queda expuesto el siguiente **Problema a Resolver**, ¿Cómo el sistema *Intranet 2.0* se comunicará con el sistema de Formación Postgraduada UCI? Es por ello que el **Objeto de Estudio** es el proceso de comunicación de la Intranet con sistemas externos; y dentro de éste, como **campo de acción**, el proceso de comunicación de la Intranet 2.0 con el sistema de Formación Postgraduada UCI. El **objetivo general** que se persigue es desarrollar un componente de integración para la Intranet 2.0 que posibilite el acceso a los servicios públicos del sistema anteriormente nombrado.

Se persiguen los siguientes objetivos específicos:

1. Desarrollar el servicio web del sistema de Formación Postgraduada UCI.
2. Desarrollar un módulo que posibilite la integración de los servicios brindados por el sistema de Formación Postgraduada UCI.



Para cumplir con los objetivos descritos, se trazaron las siguientes **Tareas** a tener en cuenta:

- Realizar un estudio de las tecnologías para desarrollar servicios web.
- Evaluar las alternativas de implementación de servicios web que se adecúen a las condiciones de la institución.
- Identificar las necesidades de la Dirección de Formación Postgraduada UCI en cuanto al proceso de gestión de la información en la Intranet 2.0.
- Definir los requisitos funcionales y no funcionales del componente.
- Describir el estado del arte sobre componentes de este tipo ya creados tanto en el campo nacional como internacional.
- Diseñar e implementar un servicio web que gestione información del sistema de Formación Postgraduada UCI.
- Diseñar e implementar un módulo que permita la integración de los servicios del sistema de Formación Postgraduada UCI dentro de la Intranet 2.0.
- Aplicar codificación de caracteres estándar UTF-8 a la base de datos del sistema de Postgrado UCI.

Los Aportes Prácticos que se obtienen con la solución propuesta son los siguientes:

- Módulo o componente que posibilitará la integración de los servicios del sistema de Formación Postgraduada UCI en cualquier otro basado en Drupal versión 6.x o superior<sup>1</sup>.

El presente documento se estructura en cuatro capítulos y las correspondientes conclusiones, recomendaciones, referencia bibliográfica, bibliografía, anexos y glosario de términos.

- *Capítulo 1- Fundamentación teórica.* Se describen algunos conceptos y características de los módulos de Drupal que se deben tener en cuenta para comprender los objetivos del trabajo y se realiza un análisis de las tecnologías de software empleadas para desarrollar el sistema.

---

<sup>1</sup> Tener en cuenta que el sistema Intranet 2.0 está desarrollado sobre Drupal 6.6.



- *Capítulo 2- Características del Sistema.* Se abordan las descripciones iniciales del sistema (servicio web y módulo), se hace un estudio del sistema de Formación Postgraduada UCI y un análisis crítico de los procesos involucrados.
- *Capítulo 3- Diseño del sistema propuesto.* Se describe el diseño del sistema que se propone.
- *Capítulo 4- Validación de la solución.* En este capítulo se plasman los casos de prueba o test de aceptación a las que fue sometida la aplicación en cada una de las iteraciones.



## CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

### 1.1. INTRODUCCIÓN

En este capítulo se exponen los fundamentos generales que sirven de base teórica a la solución y concepción del problema. Se tratan las tendencias actuales, los antecedentes y sistemas existentes.

### 1.2. ANTECEDENTES

Internet ha dejado de ser una web pasiva y unidireccional para convertirse en una Web social, dinámica, abierta, más humana y que promueve el dialogo, a la que se ha convenido en llamar *Web 2.0*. La *Web 2.0* se refiere a una nueva generación de Webs basadas en la creación de páginas Web donde los contenidos son compartidos y producidos por los propios usuarios del portal. El término *Web 2.0* se utilizó por primera vez en el año 2004 cuando *Dale Dougherty* de *O'Reilly Media* utilizó este término en una conferencia en la que hablaba del renacimiento y evolución de la Web.

Si hay una *Web 2.0* necesariamente debe existir una *Web 1.0* de donde evoluciona la primera. La *Web 1.0* es la Web tradicional y que se caracteriza porque el contenido e información de un sitio web es producido por un editor o *Webmaster* para luego ser consumido por los usuarios. En el modelo de la *Web 2.0*, la información y contenidos se producen directa o indirectamente por los usuarios del sitio Web y adicionalmente es compartida por varios portales Web de estas características.

Según Tim O'Reilly, presidente y máximo ejecutivo de *O'Reilly Media, Inc.*, quién escribió un artículo fundacional para explicar sus características, ésta se puede agrupar en estos seis grandes conceptos:

- La web es una plataforma sobre la cual se entregan los servicios que las personas utilizan; además mientras más los usan, estos mejoran.
- Los nuevos servicios impulsan la inteligencia colectiva, gracias a la conexión ofrecida por los hiperenlaces y a la capacidad de las personas de ir mejorando los productos que se ofrecen.
- Se le da la mayor importancia posible al contenido, más que a la tecnología. Están soportados por una base de datos compleja y completa.



- Se acaba con el ciclo del software, porque deja de ser un producto y se transforma en servicio que cuenta con modelos de programación simples y estables, haciendo que incluso los usuarios ayuden al desarrollo.
- El software ya no sólo trabaja en una computadora sino en cualquier equipo, debido a que las aplicaciones están construidas para funcionar sobre el web.
- La experiencia del usuario es cada vez más rica porque se da énfasis en la capacidad de interacción sobre cualquier otro elemento. (C. Camus, 2006) (Piedra & Chicaiza)

### 1.2.1. TECNOLOGÍAS EN LA WEB 2.0

La infraestructura de la Web 2.0 está relacionada con nuevas tecnologías que han hecho que sea más fácil publicar información y compartirla con otros sitios Web. Por un lado, se han actualizado los sistemas gestores de contenido haciendo que cualquier persona que no sepa nada sobre programación Web pueda, por ejemplo, gestionar su propio blog. Por otro lado, la tecnología de la Web 2.0 ha evolucionado hasta crear microformatos<sup>i</sup> estandarizados para compartir automáticamente la información de otros sitios Web. Un ejemplo conocido es la sindicación de contenidos bajo el formato RSS<sup>ii</sup> que nos permite acceder a fuentes de información (*feeds*) publicadas en otros portales de forma rápida y sencilla.

Una web está construida usando tecnología de la Web 2.0 si cumple con las siguientes técnicas: (*Wikipedia*)

- CSS, marcado XHTML válido semánticamente y microformatos
- Técnicas de aplicaciones ricas no intrusivas (como AJAX)
- XUL<sup>iii</sup>
- Redifusión/Agregación de datos en RSS/ATOM
- URLs sencillas con significado semántico
- Soporte para postear en un blog
- JCC y APIs REST o XML
- JSON
- Algunos aspectos de redes sociales

Motivados por este desarrollo, se decide implementar una nueva versión de la Intranet UCI, que integre todos los servicios brindados por otros sistemas externos a través de una única



interfaz. En este proceso se propone un nuevo modelo de Arquitectura de la Información así como lograr enfocarla a los servicios (SOA<sup>iv</sup>).

Ambas, tecnologías Web 2.0 y SOA, conciben el software como servicio. Quizás e incluso más importante, estas ven los servicios como plataformas. En vez de observarlos como ofertas independientes diseñados para ser consumidos exactamente como fueron escritos, ambos sets de tecnologías empiezan con la visión de que el rol de cualquier servicio es transformarse en la piedra de base para más servicios que se podrán construir sobre el original.

### 1.2.2. SOA

La arquitectura orientada a servicios es tanto un marco de trabajo para el desarrollo de software como un marco de trabajo de implementación. Para que un proyecto SOA tenga éxito los desarrolladores de software deben orientarse ellos mismos a esta mentalidad de crear servicios comunes que son orquestados por dientes o *middleware* para implementar los procesos de negocio. El desarrollo de sistemas usando SOA requiere un compromiso con este modelo en términos de planificación, herramientas e infraestructura. (SOA Agenda, 2009)

Cuando la mayoría de la gente habla de una arquitectura orientada a servicios hacen referencia a un juego de servicios residentes en Internet o en una intranet, usando servicios web. Existen diversos estándares relacionados a los servicios web. Incluyen los siguientes:

- XML (en la página 20)
- HTTP
- SOAP (en la página 21)
- WSDL (en la página 21)
- UDDI

#### 1.2.2.1. BENEFICIOS DE SOA

Los beneficios que puede obtener una organización que adopte SOA son:

- Mejora en los tiempos de realización de cambios en procesos.
- Facilidad para evolucionar a modelos de negocios basados en tercerización.
- Facilidad para abordar modelos de negocios basados en colaboración con otros entes (socios, proveedores).





- Poder para reemplazar elementos de la capa aplicativa SOA sin interrupción en el proceso de negocio.
- Facilidad para la integración de tecnologías disímiles.

Al analizar el sistema de Formación Postgraduada UCI, nos percatamos que no cuenta con un servicio que permita su utilización por entes externos.

### 1.3. TENDENCIAS DE LAS TECNOLOGÍAS EN LAS POLÍTICAS DE INFORMATIZACIÓN

Organizar el trabajo, garantiza siempre en gran medida la eficiencia de la actividad que se realiza. La UCI está trabajando arduamente para informatizar los procesos internos de todas sus áreas. La Dirección de Informatización UCI, a cargo de ejecutarlas, ha definido un conjunto de políticas de desarrollo que se deben seguir para realizar todas las aplicaciones dentro la intranet universitaria. Este trabajo forma parte de ese gran proyecto, por lo que a continuación se mencionan algunas pautas por las que se rige. (Dirección de Informatización, 2007)

#### 1.3.1. SERVICIO WEB

Los servicios web son sistemas de software que permiten el intercambio de datos y funcionalidades entre aplicaciones sobre una red. Están soportados en diferentes estándares que garantizan la interoperabilidad de los servicios. (*World Wide Web Consortium*)

Se comunica con otras aplicaciones codificando los mensajes en XML<sup>v</sup> y enviando estos mensajes a través de protocolos estándares de Internet tales como el HTTP (*Hypertext Transfer Protocol*). Intuitivamente, es similar a un sitio web pero no cuenta con una interfaz de usuario ya que da servicios a aplicaciones y no a las personas, pues en vez de obtener solicitudes desde el navegador y retomar páginas web como respuesta, lo que hace es recibir solicitudes a través de un mensaje formateado en XML desde una aplicación, realiza una tarea y devuelve un mensaje de respuesta también formateado en XML (*Saumell Reyes & Rodríguez Lazo, 2007*). Se puede definir de manera más sencilla como un conjunto de tecnologías estándares de software para el intercambio de datos entre aplicaciones tales como *SOAP*, *WDSL* y *UDDI*<sup>vi</sup>. Estos pueden ser desarrollados en una gran variedad de lenguajes para ser implementados sobre muchos tipos de redes de computadores.

Los servicios web exponen funcionalidad útil a los usuarios web mediante un protocolo web estándar. En la mayoría de los casos, el protocolo utilizado es SOAP. Distintas aplicaciones de



software desarrolladas en diferentes lenguajes de programación y ejecutadas sobre cualquier plataforma, pueden utilizar estos servicios para intercambiar datos en redes de ordenadores como Internet.



**Fig. 1 Esquema general de un servicio Web**

A la hora de diseñar servicios web que luego sirvan como base para crear orquestación de los mismos (y así generar valor añadido a las empresas con los procesos de negocio), para que sean fácilmente reorganizados según surjan las necesidades (en este punto, es donde SOA muestra su potencial); es necesario que tengan una base de datos totalmente orientados a negocio. Así, cuando se hable de datos a este nivel, se deberían tratar conceptos totalmente relacionados con el negocio de la empresa, a un nivel lógico y funcional (tipos de datos como “cliente”, “contrato” o “servicio”).

La abstracción con una capa de datos permite ocultar la complejidad de los mismos, permitiendo una estructura totalmente organizada a nivel del *middleware*. El resultado es que una aplicación o servicio puede pedir datos a nivel lógico y de una manera totalmente organizada sin preocuparse por la capa más física.

Normalmente esta capa de datos, se construye con XML, pues es un estándar muy fácil de utilizar y que permite generar datos complejos independientes de la capa física.

Para ello, se utilizan esquemas de datos XML (conocido como XSD). Existen ya estructuras de datos XML a nivel de negocio totalmente “estandarizadas” por distintas empresas (ejemplos como ACORD para seguros, HR-XML para Recursos Humanos, etc.), aunque la definición de una estructura totalmente orientada al negocio de una empresa por las personas que tienen el conocimiento de la misma suele ser la mejor opción (a partir de una estructura de datos particular, si fuera necesario, siempre existen maneras de transformar los datos a otras estructuras que sean más “estándar de facto” o no que la propia de cada uno).



Por esto generalmente la creación de una capa de datos en la creación de una Arquitectura SOA es un beneficio a la misma, ya que en caso de ser necesario reordenar los datos a nivel físico (cosas como por ejemplo, la reestructuración de una base de datos) no debería afectarle a nivel lógico y por tanto tampoco a las aplicaciones y servicios construidos por encima.

#### 1.3.1.1. VENTAJAS DE LOS SERVICIOS WEB

- Aportan interoperabilidad entre aplicaciones de software independientemente de sus propiedades o de las plataformas sobre las que se instalen.
- Los servicios Web fomentan los estándares y protocolos basados en texto, que hacen más fácil acceder a su contenido y entender su funcionamiento.
- Al apoyarse en HTTP, pueden aprovecharse de los sistemas de seguridad firewall sin necesidad de cambiar las reglas de filtrado.
- Permiten que servicios y software de diferentes compañías ubicadas en diferentes lugares geográficos puedan ser combinados fácilmente para proveer servicios integrados.
- Permiten la interoperabilidad entre plataformas de distintos fabricantes por medio de protocolos estándar y abiertos. Las especificaciones son gestionadas por una organización abierta, la W3C, por tanto no hay secretismos por intereses particulares de fabricantes concretos y se garantiza la plena interoperabilidad entre aplicaciones.

#### 1.3.1.2. ALTERNATIVAS DE IMPLEMENTACIÓN

Las alternativas más utilizadas en la actualidad para la implementación de servicios web, son .NET, J2EE y PHP.

##### **J2EE y Microsoft .NET.**

Tanto J2EE y .NET suponen la evolución de las tecnologías existentes hasta el momento para la construcción de aplicaciones distribuidas. Las anteriores versiones de estas tecnologías no incluían la posibilidad de construir servicios web, pero ahora ambas han hecho evolucionar sus plataformas para proporcionar esta posibilidad.

Java es uno de los lenguajes más utilizados para el desarrollo de Servicios Web junto con C# debido a que la plataforma Java facilita la interacción de aplicaciones sobre distintas plataformas (tanto software como hardware) gracias a su código portable.



.NET puede considerarse en ciertos aspectos como la respuesta de Microsoft a Java, aunque tiene bastantes diferencias. Reúne en una misma plataforma un conjunto interesante de características, como independencia de plataforma, independencia de lenguaje, soporte de bases de datos, soporte para XML, servicios Web y aplicaciones Web, entre otras.

Al igual que en Java, el código .NET no se compila a código máquina, sino a un código en un formato intermedio, independiente de la plataforma. Esto permite llevar los binarios producidos de una plataforma a otra, tal como sucede en Java.

En el documento *Lineamientos de Arquitectura UCI*, se orienta que los servicios web a utilizar dentro de la red universitaria sean basados en el lenguaje PHP, y esta es una de las razones por la cual se escoge dicho lenguaje. Más adelante, se le hará referencia.

### 1.3.2. XML

XML es uno de los lenguajes más utilizados para el intercambio de datos sobre la web. Está concebido para describir objetos de datos llamados *Documentos XML* y describir de cierta forma los programas que los procesan. Está restringido bajo la norma ISO 8879, el estándar *Generalized Markup Language*. Un documento XML es un objeto de datos que está bien formado, y se dice que lo está cuando tomado en su conjunto coincide con la producción del documento etiquetado, reúne todas las especificaciones de formato definidas y cada una de las entidades que se llaman directa o indirectamente están también bien definidas.

El XML es un lenguaje etiquetado, característica que le permite definir objetos de datos estructurados en partes bien definidas llamadas elementos. Una etiqueta es una señal realizada dentro del documento XML que delimita un segmento definido y con sentido de este documento.

Es un lenguaje muy similar a HTML pero su función principal es describir datos y no mostrarlos. XML es un formato que permite la lectura de datos a través de diferentes aplicaciones.

Las tecnologías XML son un conjunto de módulos que ofrecen servicios útiles a las demandas más frecuentes por parte de los usuarios. XML sirve para estructurar, almacenar e intercambiar información.

Sirve para que muchos programas interpreten bien cualquier tipo de datos y se comuniquen entre ellos sin intervención humana. Constituye el lenguaje base de los Servicios Web. (W3C Consortium, 2009)



### 1.3.3. SOAP

SOAP es un protocolo de la capa de aplicación para el intercambio de mensajes basados en XML sobre redes de computadores. Básicamente, es una vía de transmisión entre un *SOAP Sender* y un *SOAP Receiver*, pero los mensajes *SOAP* deben interactuar con un conjunto de aplicaciones para que se pueda generar un “diálogo” a través de mensajes *SOAP*. Un mensaje *SOAP* es la unidad fundamental de una comunicación entre nodos *SOAP* (*W3C Consortium*).

*SOAP* es básicamente un paradigma de una sola vía pero con la ayuda de las aplicaciones se puede llegar a crear patrones más complejos. Además, es independiente de la plataforma, y del lenguaje. Está basado en *XML* y es la base principal de los servicios web.

*SOAP* básicamente está constituido por:

- Un marco que describe el contenido del mensaje e instrucciones de proceso.
- Un conjunto de reglas para representar los tipos de datos definidos.
- Convenciones para representar llamadas a procedimientos remotos y respuestas.
- Y algunos lineamientos entre *SOAP* y *HTTP*.

El lenguaje PHP incorpora desde su versión 5.0.0 una implementación del protocolo dentro de su núcleo, a través de la extensión *soap*.

### 1.3.4. WSDL

*WEB SERVICES DESCRIPTION LANGUAGE* (*WSDL*) es un tipo de documento XML que describe lo que hace un servicio web, donde se encuentra y la forma de ser invocado (*WSDL, 2007*). Este provee información muy importante para los desarrolladores; además, este lenguaje describe el formato de los mensajes que utiliza y a cuáles puede responder.

Un documento *WSDL* proporciona la información necesaria al cliente para interactuar con el servicio Web. *WSDL* es extensible y se puede utilizar para describir, prácticamente, cualquier servicio de red, incluyendo *SOAP* sobre *HTTP* e incluso protocolos que no se basan en XML como *DCOM* sobre *UDP*.

Los documentos *WSDL* definen los servicios como colecciones de puntos finales de red o puertos. En *WSDL*, la definición abstracta de puntos finales y de mensajes se separa de la instalación concreta de red o de los enlaces del formato de datos. Esto permite la reutilización de definiciones abstractas: mensajes, que son descripciones abstractas de los datos que se están intercambiando y tipos de puertos, que son colecciones abstractas de operaciones. Las



especificaciones concretas del protocolo y del formato de datos para un tipo de puerto determinado constituyen un enlace reutilizable. Un puerto se define por la asociación de una dirección de red y un enlace reutilizable; una colección de puertos define un servicio. Por esta razón, un documento WSDL utiliza los siguientes elementos en la definición de servicios de red:

- *Types*: contenedor de definiciones del tipo de datos que utiliza algún sistema de tipos (por ejemplo *XSD*).
- *Message*: definición abstracta y escrita de los datos que se están comunicando.
- *Operation*: descripción abstracta de una acción admitida por el servicio.
- *Port Type*: conjunto abstracto de operaciones admitidas por uno o más puntos finales.
- *Binding*: especificación del protocolo y del formato de datos para un tipo de puerto determinado.
- *Port*: punto final único que se define como la combinación de un enlace y una dirección de red.
- *Service*: colección de puntos finales relacionados.

### 1.3.5. PHP

Se trata de un lenguaje interpretado de alto nivel, embebido en páginas HTML y ejecutado en el servidor. Está muy orientado al desarrollo de aplicaciones web y permite insertar contenidos dinámicos en las páginas.

A continuación, se describen algunas de sus características:

- Rapidez de ejecución.
- Es un lenguaje específicamente diseñado para realizar aplicaciones web, mientras que otros lenguajes son adaptaciones de lenguajes preexistentes, no pensados para la web.
- El software necesario para ejecutar aplicaciones es software libre. (GNU/Linux)
- Mantiene un bajo consumo de recursos de máquina.
- Gran seguridad, muy poca probabilidad de corromper los datos.
- Trabaja con una diversidad de bases de datos.
- Rico en funciones predefinidas.
- Multiplataforma.



- Fácil aprendizaje
- Es un lenguaje libre.
- Dispone de recursos *Open Source*.
- Trabaja en combinación con otras tecnologías como *Perl, JavaScript, Python, DHMTL*, etc., donde puedes usarlo, por ejemplo, para la creación de gráficos interactivos.
- Creación o uso de binarios ejecutables por parte del propio usuario.
- Crear páginas que obtienen información, como si del servidor se tratase.
- Permite embeber fragmentos de código dentro de la página HTML
- Presenta una filosofía totalmente diferente y, con un espíritu más generoso, es progresivamente construido por colaboradores desinteresados que implementan nuevas funciones en nuevas versiones del lenguaje.
- Combina excelentemente con otras inmejorables herramientas, como son el servidor Apache y base de datos como *MySQL* o *PostgreSQL*, todas ellas gratuitas.
- Dispone de librerías para la programación de entorno de escritorio, como *GTK* y *QT*.
- Buena documentación.

#### Principales Ventajas:

- Muy sencillo de aprender.
- Similar en sintaxis a *C* y a *PERL*.
- A partir de PHP5, utiliza el paradigma de la programación orientada a objetos. Clases y herencia.
- El análisis léxico para recoger las variables que se pasan en la dirección, lo hace de forma automática; librándose el usuario de tener que separar las variables y sus valores.
- Se puede incrustar código *PHP* con etiquetas *HTML*.
- Excelente soporte de acceso a base de datos.
- La comprobación de que los parámetros son válidos puede hacerse en el servidor y no en el cliente (como es el caso de *JavaScript*), de forma que se puede evitar el tener que chequear que no se reciban solicitudes adulteradas. Además, PHP viene equipado con un conjunto de funciones de seguridad que previenen la inserción de órdenes dentro de una solicitud de datos.
- Se puede hacer de todo lo que se pueda transmitir por vía *HTTP*. (*Comunidad PHP, 2007*)



La versión de PHP utilizada fue la 5.2.5.

### 1.3.6. EXTENSIONES

#### 1.3.6.1. SOAP

Extensión SOAP incorporada a partir de PHP 5.0.0 para ofrecer interoperabilidad con los servicios web.

#### 1.3.6.2. PDO

PDO (PHP Data Objects) es una librería estándar disponible a partir de PHP5.0 como PECL<sup>vii</sup> y en PHP5.1 como librería de serie que se puede usar para acceder a distintas bases de datos. Ofrece una interfaz orientada a objetos que se puede extender con facilidad.

PDO no es simplemente otra capa de abstracción como PearDB, aunque ésta lo utiliza como “backend” al igual que otros frameworks PHP modernos dentro de sus respectivas capas de abstracción. La necesidad de una librería como PDO nace de la gran cantidad de bases de datos soportadas por PHP. Cada una de estas bases de datos necesita de su propia API para hacer tareas generalmente comunes al resto. Esto provoca que nuestro código no sea fácil portar de una base de datos a otra y ayuda (con la colaboración de muchas otras cosas) a complicarnos la vida.

Para solucionar este problema en PHP existen múltiples librerías ofrecidas por terceros. Entre ellas, *ADODB*, *PEAR DB* y *MDB2* son algunas de las más populares.

Aunque la mayoría de estas librerías son mucho más maduras que PDO y en algunos aspectos ofrecen muchas más funcionalidades, PDO tiene ciertas ventajas que no podemos obviar.

Para empezar: la velocidad. PDO está compilado en C/C++, por lo que ofrece mayor rapidez de ejecución que sus competidoras (de código interpretado). Por otra parte: viene incorporado en PHP5 (es la solución “oficial” y en continua evolución) y desde PHP 5.1 viene habilitada por defecto. De esta manera no tienes que importar ni cargar código de terceros cuando quieres usarlo. (PHP.net, 2009)

Actualmente PDO soporta las bases de datos más populares, como:

- *DBLIB: FreeTDS*
- *Firebird*





- *IBM DB2*
- *IBM INFORMIX*
- *MySQL*
- *OCI (Oracle Call Interface)*
- *ODBC: ODBC v3*
- *PGSQL (PosgreSQL)*
- *SQLITE*
- *MSSQL (Microsoft SQL)*

### 1.3.7. SISTEMA GESTIÓN DE CONTENIDOS

Los *Sistemas de Gestión de Contenidos (CMS)* logran resolver infinidad de cuestiones relativas a la administración y publicación de información online. Excepto en casos muy particulares donde se requiere un programa web a medida (ya sea para gestión específica como para cubrir funcionalidades y servicios que surgen "*en demanda*"), los *CMS* son la solución ideal para cualquier tipo de sitios web.

Esto es así porque:

- Resuelven muy bien la categorización y jerarquía de contenidos.
- Facilita tremendamente la administración.
- Separan contenidos de presentación.
- Se logra separar eficientemente la información "estática" de las novedades y publicaciones de vigencia efímera.
- Ya están preparados para *bloggear*<sup>2</sup>, y para agregar funcionalidades sociales.
- Ya vienen listos para gestionar registros de usuarios, y separación entre contenidos y servicios públicos, y accesos a recursos privados.
- Entre otros.

Un sistema de administración de contenidos a menudo funciona en el servidor del sitio web. Muchos proporcionan diferentes niveles de acceso dependiendo del usuario, variando si es el administrador, editor, o creador de contenido. El acceso al CMS se realiza generalmente a través del navegador web, y a veces se requiere el uso de FTP para subir contenido, normalmente fotografías o audio.

---

<sup>2</sup> Disponer o publicar materiales en un blog.



Los creadores de contenido publican sus documentos en el sistema. Los editores comentan, aceptan o rechazan los documentos. El editor en jefe es responsable por publicar el trabajo en el sitio. El CMS controla y ayuda a manejar cada paso de este proceso, incluyendo las labores técnicas de publicar los documentos a uno o más sitios. En muchos sitios con CMS una sola persona hace el papel de creador y editor, como por ejemplo los blogs<sup>viii</sup>.

### 1.3.7.1. DRUPAL

Drupal es una solución de portal dinámico profesional concebida para el desarrollo de sitios Web corporativos e intranets. Permite lograr una sencilla administración de contenidos online.



#### *Características*

Es un paquete de software que permite publicar fácilmente, manejar y organizar una gran variedad de contenido. Drupal es utilizado en todo el mundo para distintos tipos de sitios a saber:

- portales Web de comunidades
- sitios de discusión
- sitios Web corporativos
- intranets
- blogs y sitios personales
- sitios de e-commerce
- directorios online
- otros

Drupal posee una gran cantidad de expansiones que le permiten cubrir funcionalidades como por ejemplo:

- sistema de manejo de contenidos.
- blogs
- ambientes colaborativos de generación de contenidos.
- foros.
- Peer-to-peer networking.
- Boletines web.
- podcasting.



- galerías de fotos.
- Carga/descarga de ficheros.
- y mucho más...

Es una solución Open-Source distribuida bajo la licencia GPL ("General Public License") y es mantenido y desarrollado por una gran cantidad de usuarios y desarrolladores en todo el mundo.

Por su diseño modular, puede ser fácilmente ampliado y adaptado a las necesidades de diversas empresas y proyectos de cualquier envergadura, ya que este producto ha sido concebido desde el comienzo para proveer una escalabilidad inmejorable comparado con otras soluciones similares disponibles en el mercado.

Su terminología difiere de la de otros gestores de contenido. La siguiente sección explora las partes más importantes de un sistema Drupal y explica los terminos más relevantes.

Un nodo es el bloque básico con el que construimos una página Web con Drupal: este puede ser un artículo, una imagen o una entrada de un foro. Los siguientes tipos de nodo se habilitan por defecto:

- *page*: para páginas estáticas,
- *story*: para artículos

Se pueden añadir más tipos de nodos instalando y habilitando módulos. Por ejemplo, el módulo *forum* define un nuevo tipo de contenido llamado *forum topic*. El módulo *blog* ofrece el nodo *personal blog entry*. El menú de administración da una lista de los tipos de nodos en *create content*.

Un bloque es un cuadro con contenido que aparece a la izquierda o derecha en una página Web. Drupal nos ofrece varios bloques preconfigurados, como *User Login*, *Who's online* o *Recent Comments*.

Adicionalmente, cualquier nuevo módulo que se instale puede proporcionar bloques, nuevos tipos de contenido, entre otros elementos, que habilitará en su caso el administrador.

#### 1.3.7.1.1. DESARROLLO DE MÓDULOS EN DRUPAL

Uno de los puntos fuertes de Drupal es la facilidad con la que puede extenderse o modificarse el funcionamiento del mismo a través de módulos. No es frecuente encontrar software que lo ponga tan fácil.



Un módulo es una colección de funciones que se enlazan dentro de Drupal, ofreciendo funcionalidades adicionales a la instalación Drupal. Normalmente se compone de un archivo PHP llamado *nombremodulo.module* y otros ficheros opcionales como imágenes u hojas de estilo.

El área de descarga del portal de Drupal en su portal en Internet<sup>3</sup> tiene una enorme selección de módulos que cubren una amplia gama de nuevas características, todos ellos creados por la comunidad de desarrolladores y guardados en repositorios CSV (Drupal.org), donde continúan desarrollándose. Por supuesto, dichos módulos están bajo licencia GPL.

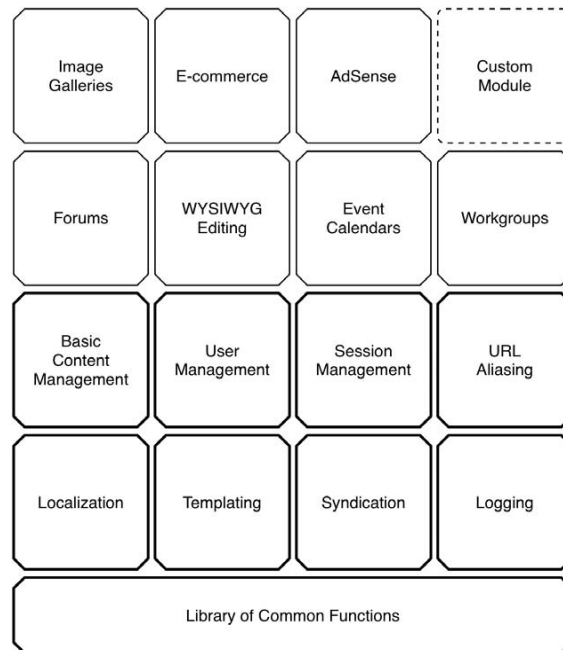


Fig. 2 Módulos para Drupal.

Tiene una sólida infraestructura de desarrollo y una API<sup>ix</sup> (Documentación API de Drupal 6.x) bien documentada. (Hermann, 2008)

### 1.3.8. HERRAMIENTAS, LENGUAJES Y METODOLOGÍAS DE DESARROLLO

A continuación, se presenta una descripción de las herramientas y metodologías empleadas en el desarrollo de la aplicación.

#### 1.3.8.1. ZEND STUDIO FOR ECLIPSE

<sup>3</sup> <http://www.drupal.org>



Zend Studio for Eclipse es un plugin desarrollado por la empresa para ayudar en el desarrollo PHP por medio de la IDE Eclipse.



El software está destinado a todo tipo de desarrolladores en PHP, pero principalmente será de gran utilidad para aquellos profesionales en la materia, debido a las posibilidades que este brinda.

Zend brinda la posibilidad de utilizar el Zend Studio for Eclipse para mejorar la calidad de los proyectos en PHP, agilizar los ciclos de desarrollos y simplificar la complejidad de los proyectos. El plugin incluye herramientas para edición, debugging, análisis, optimización y bases de datos, e incluso soportando los procesos del desarrollo por programación ágil. Es una herramienta multiplataforma.

Basado en Zend Studio y el proyecto Eclipse PHP Developers Tools (PDT), Zend Studio for Eclipse se convierte así en el IDE para PHP más potente del mercado, ofreciendo así al desarrollador profesional de PHP la potencia de Zend Studio y el soporte multilenguaje de Eclipse y su enorme conjunto de extensiones (plugins).

En concreto, Zend Studio para Eclipse aporta:

*Las ventajas de Eclipse/PDT:*

- Una comunidad de millones de usuarios y miles de desarrolladores
- Cientos de plugins
- Soporte multi-lenguaje en una única herramienta (Eclipse)
- Coloreado de sintaxis PHP, autocompletado de código e inspección de métodos y atributos.
- Soporte básico de depuración de scripts PHP

*Valor añadido de Zend Studio:*

- Soporte de depuración más avanzada (permitiendo debugging local como remoto)
- Soporte a la refactorización de código
- Análisis de rendimiento y Code Coverage
- Soporte de baterías de test PHPUnit
- Integración mejorada con Zend Framework
- Posibilidad de edición HTML WYSIWYG
- Integración con Zend Platform
- Herramientas de migración para los usuarios de Zend Studio
- Integración BIRT



- Instalación simplificada

### 1.3.8.2. METODOLOGÍA DE DESARROLLO DE SOFTWARE

En la actualidad existe un marcado progreso en el desarrollo de software, pero crearlo con la calidad requerida y que los clientes queden satisfechos con el trabajo logrado, no es tarea fácil si no se sigue una metodología que indique qué pasos seguir en cada etapa de desarrollo.

Ahora, en cuanto a la dependencia del tiempo de vida y la complejidad del proyecto que se vaya a desarrollar se proponen diferentes metodologías, pudiendo ser *tradicionales* o *ágiles*, dentro de las ágiles se encuentran: Programación extrema o XP, formulado (*Extreme Programming*), es la más destacada de las de su tipo; enfatiza mucho más la adaptabilidad que la previsibilidad también está *Microsoft Solution Framework*, entre otras. Dentro de las tradicionales está RUP, la cual constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos.

A continuación, se mencionan algunas diferencias entre las metodologías tradicionales y las ágiles.

Metodologías ágiles	Metodologías tradicionales
Basadas en heurísticas provenientes de prácticas de producción de código.	Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo.
Especialmente preparados para cambios durante el proyecto.	Cierta resistencia a los cambios.
Impuestas internamente (por el equipo)	Impuestas externamente
Proceso menos controlado, con pocos principios.	Proceso mucho más controlado, con numerosas políticas/normas.
No existe contrato tradicional o al menos es bastante flexible.	Existe un contrato prefijado.
El cliente es parte del equipo de desarrollo.	El cliente interactúa con el equipo de desarrollo mediante reuniones.
Grupos pequeños (<10 integrantes) y trabajando en el mismo sitio.	Grupos grandes y posiblemente distribuidos.
Pocos artefactos.	Más artefactos.
Pocos roles.	Más roles.
Menos énfasis en la arquitectura del software.	La arquitectura del software es esencial y se expresa mediante modelos.



**Tabla 1 Diferencias entre metodologías ágiles y no ágiles**

Las metodologías ágiles forman parte del movimiento de desarrollo ágil de software conocidos anteriormente como *metodologías livianas*, que se basan en la adaptabilidad de cualquier cambio como medio para aumentar las posibilidades de éxito de un proyecto. Se le denomina *ágil* por la habilidad de responder de forma versátil al cambio para maximizar los beneficios. Intentan evitar los tortuosos y burocráticos caminos de las metodologías tradicionales enfocándose en la gente y los resultados.

Después de realizar un estudio sobre las metodologías de desarrollo de software existentes, se opta por la utilización de las denominadas *metodologías ágiles*.<sup>4</sup>

### 1.3.8.3. METODOLOGÍAS ÁGILES

Las metodologías de desarrollo de software ágil son una supuesta solución a los inconvenientes que muestran o presentan los tradicionales métodos de desarrollo (en otras palabras una extensión de los métodos comunes).

Surgen como una extensión a las metodologías tradicionales para mejorar el desarrollo de sistemas, según el tipo de proyecto y empresa, añadiendo y mejorando (optimizando) las prácticas de desarrollo de software.

Las AMS (*agile methods*) metodologías ágiles de desarrollo de software son una agrupación de las prácticas tradicionales pero llevadas al extremo, tomando la esencia y aplicándolas buscando la calidad en el desarrollo desde el inicio, entregas oportunas y la entrega final del sistema, teniendo en cuenta el soporte, mantenimiento, auditoría y capacitaciones al usuario final. Esto conlleva a que se reduzcan considerablemente el coste de desarrollo y mantenimiento de los proyectos en cuestión del tiempo. (Canós, Letelier, & Penadé)

Según el *Manifiesto Ágil*<sup>x</sup> se valora:

- Al individuo y las interacciones del equipo de desarrollo sobre el proceso y las herramientas.
- Desarrollar software que funciona más que conseguir una buena documentación.
- La colaboración con el cliente más que la negociación de un contrato.
- Responder a los cambios más que seguir estrictamente un plan.

<sup>4</sup> En parte de la decisión sobre el uso de la metodología para el desarrollo de esta solución, influyó el hecho de formar parte de la estrategia de trabajo de los proyectos pertenecientes o atendidos por la Dirección de Informatización.



Dentro del amplio espectro de AMS se encuentran *Scrum*, MSF (*Microsoft Solutions framework*), FDD (*Feature Driven Development*), *Crystal Methods*, XP, entre otras.

Dada la dinámica regente en los proyectos que están a cargo de la Dirección de Informatización de la Universidad, se contempla el uso de metodologías ágiles; específicamente, *Scrum* en la Gestión de Proyectos, y la denominada *Extreme Programming* o XP para el desarrollo; razón por la cual se adopta la metodología *MA-GMPUR2* o *Metodología ágil para proyectos de software libre* (Peñalver Romero, 2008), la cual contiene la organización de los procedimientos a seguir paso a paso, con la generación de cada uno de los artefactos necesarios para lograr una documentación con el éxito y la eficiencia necesaria que requiere un proceso de software.

#### **1.3.8.3.1. SCRUM**

*Scrum* es un proceso ágil y liviano que sirve para administrar y controlar el desarrollo de software. El desarrollo se realiza en forma iterativa e incremental (una iteración es un ciclo corto de construcción repetitivo). Cada ciclo o iteración termina con una pieza de software ejecutable que incorpora nueva funcionalidad. Las iteraciones en general tienen una duración entre 2 y 4 semanas. *Scrum* se utiliza como marco para otras prácticas de ingeniería de software como *RUP* o *Extreme Programming*.

Se trata de priorizar el trabajo en función del valor que tenga para el negocio, maximizando la utilidad de lo que se construye y el retorno de inversión. Está diseñado especialmente para adaptarse a los cambios en los requerimientos, por ejemplo en un mercado de alta competitividad. Los requerimientos y las prioridades se revisan y ajustan durante el proyecto en intervalos muy cortos y regulares. De esta forma se puede adaptar en tiempo real el producto que se está construyendo a las necesidades del cliente. Se busca entregar software que realmente resuelva las necesidades, aumentando la satisfacción del cliente.

En *Scrum*, el equipo se concentra en una única cosa: construir software de calidad. Por el otro lado, la gestión de un proyecto *Scrum* se enfoca en definir cuales son las características que debe tener el producto a construir (qué construir, qué no y en qué orden) y en remover cualquier obstáculo que pudiera entorpecer la tarea del equipo de desarrollo. Se busca que los equipos sean lo más efectivos y productivos que sea posible.

*Scrum* tiene un conjunto de reglas muy pequeño y muy simple y está basado en los principios de inspección continua, adaptación, auto-gestión e innovación. El cliente se entusiasma y se compromete con el proyecto dado que ve crecer el producto iteración a iteración y encuentra





las herramientas para alinear el desarrollo con los objetivos de negocio de su empresa. Por otro lado, los desarrolladores encuentran un ámbito propicio para desarrollar sus capacidades profesionales y esto resulta en un incremento en la motivación de los integrantes del equipo. (Alliance)

Según diversos estudios estadísticos de la industria del software, se muestra que casi la mitad de la funcionalidad que se implementa en los proyectos no es utilizada. En *Scrum* se construye primero la funcionalidad que resulte en el mayor beneficio para el negocio y se evita en todo momento desarrollar cosas que no serán de utilidad para el cliente. De esta manera, se puede desarrollar un sistema en menos tiempo evitando el trabajo innecesario.

Define un marco para la gestión de proyectos, que se ha utilizado con éxito durante los últimos 10 años.

Está especialmente indicada para proyectos con un rápido cambio de requisitos. Sus principales características se pueden resumir en dos. El desarrollo de software se realiza mediante iteraciones, denominadas sprints, con una duración de 30 días. El resultado de cada sprint es un incremento ejecutable que se muestra al cliente. La segunda característica importante son las reuniones a lo largo del proyecto, entre ellas destaca la reunión diaria de 15 minutos del equipo de desarrollo para coordinación e integración.

#### **1.3.8.3.2. EXTREME PROGRAMMING**

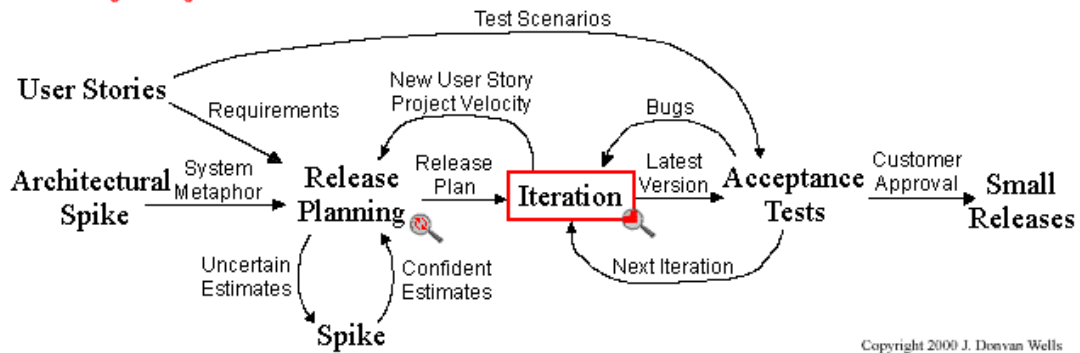
*EXTREME PROGRAMMING* o XP es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. XP se basa en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

En 1996, Kent Beck y Ward Cunningham pusieron en práctica una nueva metodología primando la simplicidad y evitando los hábitos que convertían las cosas fáciles en difíciles durante el desarrollo de un proyecto en DaimlerChrysler. El resultado fue la metodología eXtreme Programming o XP (que por supuesto nada tiene que ver con software de la compañía Microsoft).





# Extreme Programming Project



Copyright 2000 J. Donovan Wells

Fig. 3 Representación de un proyecto basado en la metodología XP

Los procesos y prácticas de esta metodología están basados en la experiencia de equipos de desarrollo, y en los errores cometidos o encontrados una y otra vez al utilizar metodologías más tradicionales. Sin embargo, algunas de sus propuestas pudieran resultar incompatibles con grandes organizaciones o grandes proyectos.

La proximidad de esta metodología al entorno del software libre viene dada por sus principios básicos, más que por su aplicación concreta. XP puede dividirse en cuatro principios sobre los que se va iterando hasta que el proyecto ha finalizado (el cliente aprueba el proyecto). Estas fases o principios son planificación, diseño, desarrollo y pruebas. (SALAZAR GUTIERREZ)

## Trabajando con Extreme Programming

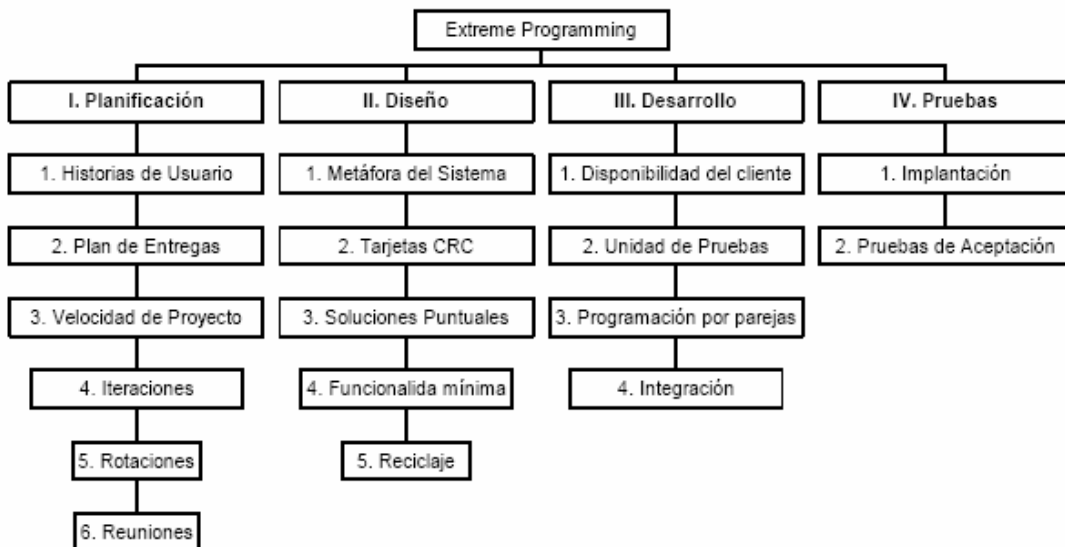


Fig. 4 Representación de las fases de la Metodología XP



#### 1.3.8.4. UML

UML son las siglas de *Unified Modeling Language* (Lenguaje Unificado de Modelado), notación (esquemática en su mayor parte) con que se construyen sistemas por medio de conceptos orientados a objetos. (Fowler & Scott, 1997)

Como todo lenguaje, UML proporciona un vocabulario y unas reglas para permitir una comunicación. Éste particularmente, está compuesto por elementos que no son más que abstracciones que constituyen los bloques básicos de construcción, los cuales pueden unirse mediante relaciones para conformar los diagramas. UML facilita la representación gráfica de un sistema, teniendo como objetivo sustancial, brindar un material de apoyo que le permita al lector poder definir diagramas propios, como también entender diagramas ya existentes. Sus principales funciones son visualizar, especificar, construir y documentar cada una de las partes que comprende el desarrollo de software.

UML tiene tres características que lo hacen ideal:

- Sirve para el modelado completo de sistemas complejos, tanto en el diseño de los sistemas software como para la arquitectura hardware donde se ejecuten.
- Se pueden automatizar determinados procesos y permite generar código a partir de los modelos y a la inversa (a partir del código fuente generar los modelos) (*Orallo Hernández*). Esto permite que el modelo y el código estén actualizados, por lo que siempre se puede mantener la visión en el diseño, de la estructura del proyecto.
- Es completamente independiente del lenguaje de implementación, de tal forma que los diseños realizados usando UML, se pueda implementar en cualquier lenguaje que soporte las posibilidades de UML (principalmente lenguajes orientados a objetos). (*Orallo Hernández*)



Producto de todas estas ventajas, UML no solo es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad; sino que además, se ha convertido en el estándar para describir el "plano" de los sistemas informáticos.

#### 1.3.8.5. HERRAMIENTA CASE



Las Herramientas CASE (*Computer Aided Software Engineering*, Ingeniería de Software Asistida por Ordenador) son aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el coste de las mismas.

Estas herramientas son utilizadas en todos los aspectos del ciclo de vida de desarrollo del software en tareas como el proceso de realizar un diseño del proyecto, cálculo de costes, implementación de parte del código automáticamente con el diseño dado, compilación automática, documentación o detección de errores entre otras.

Dentro de sus objetivos se encuentran:

- Mejorar la productividad en el desarrollo y mantenimiento del software.
- Aumentar la calidad del software.
- Reducir el tiempo y coste de desarrollo y mantenimiento de los sistemas informáticos.
- Mejorar la planificación de un proyecto
- Aumentar la biblioteca de conocimiento informático de una empresa ayudando a la búsqueda de soluciones para los requisitos.
- Automatizar el desarrollo del software, la documentación, la generación de código, las pruebas de errores y la gestión del proyecto.
- Ayuda a la reutilización del software, portabilidad y estandarización de la documentación
- Gestión global en todas las fases de desarrollo de software con una misma herramienta.
- Facilitar el uso de las distintas metodologías propias de la ingeniería del software.



#### 1.3.8.5.1. VISUAL PARADIGM

*Visual Paradigm* es una poderosa herramienta CASE que al igual que el Rational Rose utiliza UML para el modelado, es la herramienta por excelencia para ser utilizada en un ambiente de software libre. Permite crear tipos diferentes de diagramas en un ambiente totalmente visual. Es muy sencillo de usar, fácil de instalar y actualizar. Genera código para varios lenguajes.

El Visual Paradigm ofrece además:

- Soporte de UML versión 2.1.



- Diagramas de Procesos de Negocio - Proceso, Decisión, Actor de negocio, Documento.
- Modelado colaborativo con CVS y Subversion.
- Interoperabilidad con modelos UML2 (metamodelos UML 2.x para plataforma Eclipse) a través de XMI.
- Ingeniería inversa - Código a modelo, código a diagrama.
- Ingeniería inversa Java, C++, Esquemas XML, XML,.NET exe/dll, CORBA IDL.
- Generación de código - Modelo a código, diagrama a código.
- Editor de Detalles de Casos de Uso - Entorno todo-en-uno para la especificación de los detalles de los casos de uso, incluyendo la especificación del modelo general y de las descripciones de los casos de uso.
- Generación de código y despliegue de EJB's - Generación de beans para el desarrollo y despliegue de aplicaciones.
- Diagramas de flujo de datos.
- Generación de bases de datos - Transformación de diagramas de Entidad-Relación en tablas de base de datos.
- Ingeniería inversa de bases de datos - Desde Sistemas Gestores de Bases de Datos (DBMS) existentes a diagramas de Entidad-Relación.
- Generador de informes para generación de documentación.
- Distribución automática de diagramas - Reorganización de las figuras y conectores de los diagramas UML.
- Importación y exportación de ficheros XMI.
- Integración con Visio - Dibujo de diagramas UML con plantillas (stencils) de MS Visio.
- Editor de figuras.
- Más... Otras herramientas y plugins de modelado UML:
- Plataforma Java (Windows/Linux/Mac OS X). (Visual Paradigm)

Posibilita la representación gráfica de los diagramas permitiendo ver el sistema desde diferentes perspectivas, como el de componentes, despliegue, secuencia, casos de uso, clase, actividad, estado, entre otros. Además, identifica requisitos y comunica información, se centra en cómo los componentes del sistema interactúan entre ellos, sin entrar en detalles excesivos, además, permite ver las relaciones entre los componentes del diseño y mejora la comunicación entre los miembros del equipo usando un lenguaje gráfico. Facilita licencias especiales para fines académicos.



## 1.4. CONCLUSIONES

En este capítulo se hizo un estudio y valoración de las herramientas, lenguaje y herramientas propuestas para el desarrollo del sistema; así como también de la metodología de desarrollo de software a utilizar, abordando las características principales de las mismas.



## CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA.

### 2.1. INTRODUCCION

En este capítulo se realizará el análisis propuesto por el analista para la solución del problema. Se analizarán posibles componentes ya existentes los cuales pueden ser utilizados nuevamente así como la estrategia de integración al sistema.

### 2.2. OBJETO DE ESTUDIO

#### 2.2.1. INTRANET 2.0

En los inicios de la UCI, se concibió un proyecto denominado *Ciudad Digital* cuyo resultado fundamental fue la *Intranet 1.0*. Posteriormente se realiza una actualización (versión 1.5, actual), aunque la misma no cumple con las necesidades que la comunidad universitaria exige dado el incremento del personal y de los múltiples servicios que se ofertan.

Según estudios realizados por la Dirección de Informatización UCI, la intranet que se encuentra publicada en la actualidad introdujo un concepto erróneo a la comunidad. Una *intranet* es una “infraestructura de aplicaciones y servicios montada sobre una red de telecomunicaciones que interrelacione todas las computadoras existentes”. Las intranets interconectan los recursos de cómputo con que cuenta una empresa o institución; con un punto de acceso único a todos los usuarios facilitando la gestión y presentación de la información. El portal de la intranet actual rompe, además, con la gama cromática que rige nuestra institución. No refleja el quehacer universitario y está orientada a estructuras administrativas, y no a procesos enfocados en el usuario final, por lo que podemos afirmar que la arquitectura de información no es la más adecuada.

Debido a esta situación, se decide crear un sistema de gestión por cada una de las vicerrectorías o direcciones de la universidad para que controlen o gestionen las actividades internas propias del área administrativa, y que a su vez sean el soporte del portal *Intranet 2.0*. Básicamente este portal dispondrá de los servicios que los anteriores deseen brindar al público. Esta nueva intranet logrará tener un respaldo institucional de forma que se garantice que toda la información mostrada sea oficial. (Dirección de Informatización UCI)

¿Cuáles son las principales características del nuevo portal universitario?

- Identidad visual.
- Personalización al usuario.



- Integración de servicios.
- Usabilidad y comunicación.
- Acceso a un clic.
- Retroalimentación implícita.
- Incorporación de elementos de la prensa digital.
- Modernidad y nuevas tecnologías.
- Respaldo de la institucionalidad.
- Vitalidad al proyecto Ciudad Digital.
- Relaciones de trabajo con las áreas.
- Prioridad en el desarrollo y respaldo de servicios.
- Dinámica de desarrollo a la par de la Informatización de la UCI.

En resumen, el producto *Intranet 2.0* propone:

- Integración de los servicios de gestión de sistemas externos desde un solo sistema.
- Incorporación de mejoras en cuanto a experiencia de usuario, arquitectura de la información, etc.



Fig. 5 Evolución hacia Intranet 2.0

Una de las áreas a incorporar en este proceso es la referente a la Dirección de Formación Postgraduada, la cual cuenta con su propio sistema de gestión. Como parte del proceso de implementación y despliegue del sistema *Intranet 2.0*, se decide integrarle un módulo que sirviese de enlace a los servicios de postgrado. La solución propuesta incluye la incorporación de un servicio web al sistema de Formación Postgraduada UCI.

### 2.3. VALORACIÓN DISEÑO PROPUESTO POR EL SISTEMA DE POSTGRADO UCI

El diseño propuesto por el sistema de Postgrado UCI fue valorado como bueno, fácil de entender y muy específico pues permitió adquirir una comprensión de los aspectos relacionados con los requisitos funcionales y no funcionales del sistema, que sirvieron de base





para iniciar el desarrollo del servicio web y del componente modular que permitiese su integración en cualquier entorno basado en Drupal.

Se obtuvieron además, diagramas como los de clases de diseño que mostraron la parte estática del sistema, la representación de las clases y sus relaciones, aspecto que facilita en gran medida la implementación de las mismas dada sus definiciones.

Unido a esto se encuentran los diagramas de interacción, que explican la colaboración que existe entre las clases y cómo son llamados los métodos y sentencias dentro de cada una, de los cuales se obtiene la información necesaria para conocer el orden de las acciones a implementar. Mientras más legible sea el diseño propuesto más fácil es la adaptación de los programadores al mismo y por ende se agiliza el proceso de traducción de clases del diseño a clases de implementación.

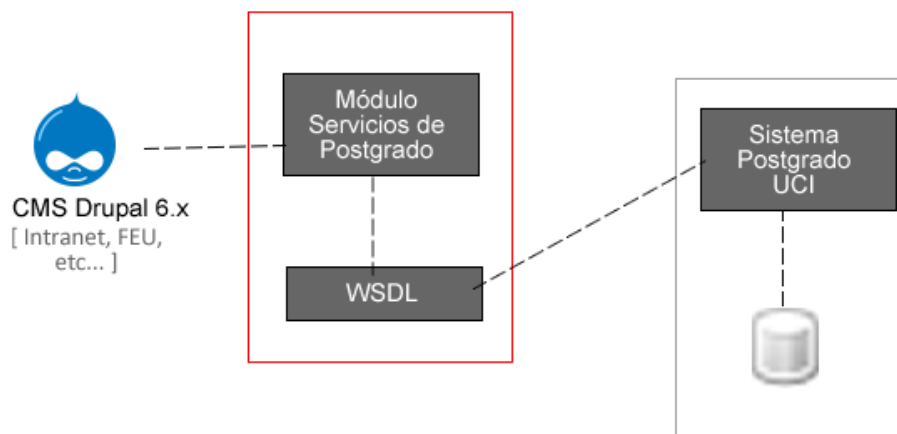
No obstante, tras un análisis efectuado a dicho sistema se proponen efectuar las siguientes modificaciones que realizan un trabajo más completo y óptimo que las propuestas:

- Ajuste del modelo de datos.
- La consulta de los datos y reservación se realizarán mediante previa autenticación usando el dominio *UCI.CU*.
- Agregar nuevos filtros de búsqueda.
- Establecer codificación de caracteres estándar UTF-8 a la base de datos.

## 2.4. PROPUESTA DE SISTEMA

En la presente investigación, con el propósito de darle cumplimiento al problema planteado y de acuerdo a los estudios realizados, se propone el desarrollo de un servicio web al sistema de Formación Postgraduada; y un módulo denominado *Servicios de Postgrado* para el CMS Drupal que utilice dichos servicios. El nuevo componente será implementado sobre la base del entorno en que se encuentra desplegado el sistema *Intranet 2.0*, o sea, Drupal.





**Fig. 6 Propuesta de sistema**

Se plantea esta solución partiendo ya que existen constantes cambios en cuanto a las tecnologías de desarrollo de la intranet, a la cual se pretenden integrar cada uno de los módulos desarrollados bajo el lenguaje de programación PHP, en este caso se encuentra el de servicios de postgrado.

El servicio web expondrá funciones que permitirán el acceso a los procesos definidos dentro del negocio definido al sistema de Formación Postgraduada, entre ellas, el registro de las solicitudes de locales para impartir cursos o conferencias; consulta y procesamiento de las reservaciones; mientras que el módulo permitirá la integración de los servicios de Postgrado dentro de la *Intranet 2.0*.

Para llevar a cabo el desarrollo de esta propuesta, se partió del análisis realizado de los procesos que se desarrollan dentro del negocio del sistema de postgrado y las necesidades de *Intranet 2.0*; se tuvo en cuenta el estudio de tecnologías, las cuales fueron reflejadas en el capítulo 1, con el objetivo de tener conocimientos de los mismos en cuanto a funcionalidades y características, destacando que dentro de las investigaciones efectuadas no se detectaron servicios web similares al que se desea implementar ni módulo alguno que sirviera con tal fin.

La propuesta de utilizar un servicio web y módulo está basada fundamentalmente en la necesidad de tener estos componentes en un entorno desacoplado, que permita una independencia de lenguaje y sistema operativo, de acuerdo con los requerimientos del sistema.

## 2.5. ESPECIFICACIÓN DE LOS REQUISITOS DE SOFTWARE



Los requerimientos de un software son las condiciones o capacidades que tiene que tener un sistema para satisfacer un contrato o documento formal.

### 2.5.1. PILA DE PRODUCTO

La *pila de producto* o *product backlog* es el inventario de funcionalidades, mejoras, tecnología y corrección de errores que deben incorporarse al producto a través de las sucesivas iteraciones de desarrollo. Representa todo aquello que esperan los clientes, usuarios, y en general los interesados en el producto. Todo lo que suponga un trabajo que debe realizar el equipo tiene que estar reflejado en el *backlog*.

A diferencia de un documento de requisitos del sistema, el *product backlog* nunca se da por completo; está en continuo crecimiento y evolución.

La aplicación a desarrollar debe brindar funciones capaces de:

Historias de Usuario	Descripción
<i>Solicitudes de locales</i>	
1.	Registrar solicitud
2.	Modificar solicitud
3.	Eliminar solicitud
<i>Gestionar reservaciones</i>	
4.	Crear reservación
5.	Modificar reservación
6.	Eliminar reservación
7.	Mostrar reservaciones
<i>Filtrar búsqueda</i>	
8.	Filtrar resultados por local
9.	Filtrar resultados por sesión de trabajo
10.	Filtrar resultados por responsable
11.	Filtrar resultados por área de trabajo <sup>5</sup>
12.	Gestión de locales <sup>6</sup>

Tabla 2 Historias de Usuario

<sup>5</sup> Hasta el momento del desarrollo del componente, no existe un servicio web que permita tener acceso a las áreas de trabajo UCI. Se encuentra en fase de construcción.

<sup>6</sup> No fue definido por el cliente. Constituye un nuevo servicio.



### 2.5.1.1. HISTORIAS TÉCNICAS

Las historias técnicas o requerimientos no funcionales son propiedades que el sistema debe poseer. A continuación, se presentan separados por categorías.

#### *Apariencia o interfaz externa*

- Uso de técnicas de diseño adecuadas al entorno UCI. Esto incluye el uso de colores y otros factores de accesibilidad.

#### *Usabilidad*

- El módulo debe proporcionar materiales de ayuda para hacer uso de los servicios que ofrece.
- Correcta visualización en cualquier navegador y/o plataforma.

#### *Rendimiento*

- Debe ser eficiente, preciso, cortos tiempos de respuesta y gran velocidad de procesamiento de la información.

#### *Soporte*

- Funcionará sobre una solución desarrollada utilizando software libre.
- Fácil instalación, mantenimiento y adaptable a cualquier plataforma.

#### *Portabilidad*

- Es accesible desde cualquier plataforma.
- Integrable con cualquier sistema web basado en el CMS <sup>xi</sup>Drupal.

#### *Seguridad*

- Se realizan validaciones de la información.

#### *Legales*

- La aplicación y toda la documentación generada pertenecen a la Universidad de las Ciencias Informáticas.



□ *Confiabilidad y Seguridad*

- El nuevo módulo permitirá ejecutar las acciones de Postgrado desde la Intranet o desde cualquier otro basado en Drupal, lo que garantiza el trabajo de los usuarios y las acciones de mantenimiento.

□ *Software*

- El módulo estará disponible para el CMS Drupal.

## 2.6. DEFINICIÓN DE LAS HISTORIAS DE USUARIO

### 2.6.1. HISTORIAS DE USUARIO

ID	ORDEN	ESTIMADO	DESCRIPCIÓN	CRITERIO VALIDACIÓN	OBSERVACIONES
1	1	20	SOLICITAR RESERVACIÓN DE LOCALES DE POSGRADO	PANTALLA DE INTERFAZ PARA LA FUNCIONALIDAD.	DEBE ESTAR LA INTERFAZ PARA LAS FUNCIONALIDADES DE LA PILA A FECHA.
2	2	20	MOSTRAR RESERVACIONES DE LOCALES DE POSGRADO	PANTALLA DE INTERFAZ PARA LA FUNCIONALIDAD.	DEBE ESTAR LA INTERFAZ PARA LAS FUNCIONALIDADES DE LA PILA A FECHA.
3	3	10	FILTRADO DE RESERVACIONES	EL SISTEMA MUESTRA UN LISTADO DE LAS RESERVACIONES Y PERMITE FILTRARLAS SEGÚN LOS CRITERIOS: LOCAL, SESIÓN DE TRABAJO, RESPONSABLE.	DEBE ESTAR LA INTERFAZ PARA LAS FUNCIONALIDADES DE LA PILA A FECHA.
4	4	30	GESTIÓN DE SOLICITUDES DE LOCALES	PANTALLA DE INTERFAZ PARA LA FUNCIONALIDAD.	PERTENECE A ADMINISTRACIÓN.
5	5	10	MODIFICAR SOLICITUD		
6	6	10	ELIMINAR SOLICITUD		
7	7	30	GESTIÓN DE RESERVACIONES	PANTALLA DE INTERFAZ PARA LA	PERTENECE A ADMINISTRACIÓN.



8		10	AÑADIR RESERVAIÓN	FUNCIONALIDAD.	
9	8	10	MODIFICAR RESERVAIÓN		
10	9	10	ELIMINAR RESERVAIÓN		
11	10	30	GESTIÓN DE LOCALES	PANTALLA DE INTERFAZ PARA LA FUNCIONALIDAD.	PERTENECE A ADMINISTRACIÓN.
12	11	10	AÑADIR LOCAL		
13	12	10	MODIFICAR LOCAL		
14	13	10	ELIMINAR LOCAL		

Tabla 3 Historias de Usuario

## 2.7. HISTORIAS DE USUARIO POR ITERACIÓN

	NOMBRE DE HISTORIA DE USUARIO
I ITERACIÓN	REGISTRAR SOLICITUD
	MOSTRAR RESERVACIONES
	FILTRAR BÚSQUEDA
II ITERACIÓN	GESTIONAR SOLICITUDES DE LOCALES
	GESTIONAR RESERVACIONES
	GESTIONAR LOCALES

Tabla 4 Historias de Usuario por iteración



## CAPÍTULO 3. DISEÑO DEL SISTEMA PROPUESTO.

### INTRODUCCION

El presente capítulo se centra en el desarrollo ágil del módulo Servicios de Postgrado, utilizando la metodología seleccionada. La fase de Desarrollo, es la segunda que define la metodología XP - Scrum, utilizada por los proyectos productivos de la Dirección de Informatización UCI.

### 3.1. HISTORIAS DE USUARIO Y PROTOTIPOS UI

A continuación se exponen las historias de usuario, los prototipos UI y las tareas asociadas a ellas; se relacionan con la prioridad que tienen y los usuarios que se encargan de desarrollarlas. Esta es solo una planificación inicial, el proceso es cambiante para ir adecuándolo a las necesidades y nuevas propuestas. Todas las decisiones se toman de conjunto con el cliente que es parte del equipo de desarrollo.

#### 3.1.1. HU SOLICITAR RESERVACIÓN DE LOCALES DE POSGRADO

<b>NÚMERO:</b> <i>HU.1</i>	<b>USUARIO:</b> <i>ALEXANDER F. VINENT PEÑA</i>	
<b>NOMBRE HISTORIA DE USUARIO:</b> <i>SOLICITAR RESERVACIÓN DE LOCALES DE POSGRADO</i>		
<b>ITERACIÓN ASIGNADA:</b> <i>1RA. ITERACIÓN</i>		
<b>PRIORIDAD EN NEGOCIO:</b> <i>ALTA</i>	<b>PUNTOS ESTIMADOS:</b> <i>20</i>	
<b>RIESGO EN DESARROLLO:</b> <i>MEDIA</i>	<b>PUNTOS REALES:</b> <i>50</i>	

**DESCRIPCIÓN:**

EL SISTEMA MUESTRA EL FORMULARIO DE SOLICITUD DE RESERVACIÓN DE LOCALES DE POSGRADO, DONDE EL USUARIO INTRODUCE LOS SIGUIENTES DATOS:

- FECHA
- RESPONSABLE (NOMBRE Y APELLIDOS DE LA PERSONA UCI QUE REALIZA LA SOLICITUD)
- TIPO DE LOCAL (AULA DE 30 PERSONAS, SALÓN DE 90 PERSONAS Y LABORATORIOS DE 20 PERSONAS),
- SESIÓN (MAÑANA, TARDE, NOCHE Y TODO EL DÍA),
- DESCRIPCIÓN DE LA ACTIVIDAD.

UNA VEZ INTRODUCIDOS TODOS LOS CAMPOS DEL FORMULARIO EL USUARIO PRESIONA EL BOTÓN SOLICITAR RESERVA Y LA SOLICITUD PASA A FORMAR PARTE DEL LISTADO DE SOLICITUDES EN ESPERA DE CONFIRMACIÓN POR LA PERSONA CORRESPONDIENTE.

UNA VEZ REALIZADA LA SOLICITUD DE RESERVACIÓN DE LOCAL SE MUESTRA UNA INTERFAZ CON LA POSIBILIDAD DE DADA UNA FECHA ESCOGIDA POR EL USUARIO ESCOGER LAS SIGUIENTES OPCIONES: REALIZAR OTRA SOLICITUD, Y VER LAS RESERVACIONES QUE EXISTEN ACTUALMENTE.

**OBSERVACIONES:**

CONFIRMADO CON EL CLIENTE. DEBE ESTAR AUTENTICADO PARA REALIZAR ESTA OPERACIÓN.

**PROTOTIPO UI:**

Abril 2009						
L	M	M	J	V	S	D
30	31	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	1	2	3

## SISTEMA DE RESERVAS DE LOCALES PARA LA ACTIVIDAD DE POSTGRADO

**SOLICITUD DE LOCAL**      FECHA DE RESERVA: 2/04/2009

TIPO DE LOCAL	SESION	RESPONSABLE	AREA
Aula	mañana	Yamilis Fernández	Dirección Formación Posgraduada

DESCRIPCIÓN DE LA ACTIVIDAD

Discusión de Tesis

**SOLICITAR RESERVA** 

Tabla 5 HU Solicitar Reservación de Locales de Postgrado

TAREA DE INGENIERÍA	
<b>NÚMERO DE TAREA:</b> T.11	<b>NUMERO HISTORIA DE USUARIO:</b> HU_1
<b>NOMBRE TAREA:</b> CREAR FUNCIONALIDAD PARA REGISTRAR SOLICITUD VÍA SERVICIO WEB.	
<b>TIPO DE TAREA:</b> DESARROLLO	<b>PUNTOS ESTIMADOS:</b> 3



<b>FECHA INICIO:</b> 24/02/2009	<b>FECHA FIN:</b> 26/02/2009
<b>PROGRAMADOR RESPONSABLE:</b> ALEXANDER F. VINENT PEÑA	
<b>DESCRIPCIÓN:</b> CREAR FUNCIONALIDAD PARA REGISTRAR SOLICITUD VÍA SERVICIO WEB.	

Tabla 6 Tarea de Ingeniería: Crear funcionalidad para registrar solicitud mediante servicio web.

TAREA DE INGENIERÍA	
<b>NÚMERO DE TAREA:</b> T.12	<b>NUMERO HISTORIA DE USUARIO:</b> HU_1
<b>NOMBRE TAREA:</b> CREAR FUNCIONALIDAD PARA REGISTRAR SOLICITUD MEDIANTE MÓDULO DE DRUPAL.	
<b>TIPO DE TAREA:</b> DESARROLLO	<b>PUNTOS ESTIMADOS:</b> 3
<b>FECHA INICIO:</b> 25/03/2009	<b>FECHA FIN:</b> 10/04/2009
<b>PROGRAMADOR RESPONSABLE:</b> ALEXANDER F. VINENT PEÑA	
<b>DESCRIPCIÓN:</b> CREAR FUNCIONALIDAD PARA REGISTRAR SOLICITUD MEDIANTE MÓDULO DE DRUPAL VÍA SERVICIO WEB.	

Tabla 7 Tarea de Ingeniería: Crear funcionalidad para registrar solicitud mediante módulo de Drupal.

TAREA DE INGENIERÍA	
<b>NÚMERO DE TAREA:</b> T.13	<b>NUMERO HISTORIA DE USUARIO:</b> HU_1
<b>NOMBRE TAREA:</b> DISEÑO UI DE LA FUNCIONALIDAD REGISTRAR SOLICITUD.	
<b>TIPO DE TAREA:</b> MEJORA	<b>PUNTOS ESTIMADOS:</b> 3
<b>FECHA INICIO:</b> 24/03/2009	<b>FECHA FIN:</b> 31/03/2009
<b>PROGRAMADOR RESPONSABLE:</b> ALEXANDER F. VINENT PEÑA	
<b>DESCRIPCIÓN:</b> DISEÑO UI DE LA FUNCIONALIDAD REGISTRAR SOLICITUD.	

Tabla 8 Tarea de Ingeniería: Diseño UI de la funcionalidad registrar solicitud.

<b>TAREA DE INGENIERÍA</b>	
<b>NÚMERO DE TAREA:</b> <i>T.14</i>	<b>NUMERO HISTORIA DE USUARIO:</b> <i>HU_1</i>
<b>NOMBRE TAREA:</b> <i>NOTIFICAR USUARIO SOBRE SOLICITUD REALIZADA.</i>	
<b>TIPO DE TAREA:</b> <i>DESARROLLO</i>	<b>PUNTOS ESTIMADOS:</b> <i>3</i>
<b>FECHA INICIO:</b> <i>24/03/2009</i>	<b>FECHA FIN:</b> <i>31/03/2009</i>
<b>PROGRAMADOR RESPONSABLE:</b> <i>ALEXANDER F. VINENT PEÑA</i>	
<b>DESCRIPCIÓN:</b> <i>NOTIFICAR AL USUARIO INTERESADO QUE LA SOLICITUD DE RESERVACIÓN FUE REALIZADA.</i>	

Tabla 9 Tarea de Ingeniería: Notificar usuario.

### 3.1.2. HU MOSTRAR RESERVACIONES DE LOCALES DE POSGRADO

<b>NÚMERO:</b> <i>HU.2</i>	<b>USUARIO:</b> <i>ALEXANDER F. VINENT PEÑA</i>
<b>NOMBRE HISTORIA DE USUARIO:</b> <i>MOSTRAR RESERVACIONES DE LOCALES DE POSGRADO</i>	
<b>ITERACIÓN ASIGNADA:</b> <i>1RA. ITERACIÓN</i>	
<b>PRIORIDAD EN NEGOCIO:</b> <i>ALTA</i>	<b>PUNTOS ESTIMADOS:</b> <i>20</i>
<b>RIESGO EN DESARROLLO:</b> <i>MEDIA</i>	<b>PUNTOS REALES:</b> <i>50</i>

**DESCRIPCIÓN:**

EL ESCENARIO COMIENZA CUANDO EL ACTOR SELECCIONA LA OPCIÓN VER RESERVAS. EL SISTEMA MUESTRA UN LISTADO DE LAS RESERVACIONES PLANIFICADAS PARA LA FECHA ACTUAL Y UN CALENDARIO, BRINDANDO LA POSIBILIDAD DE ESCOGER UNA FECHA EN EL CALENDARIO Y MOSTRAR LAS RESERVACIONES REALIZADAS EN LA FECHA ESCOGIDA POR EL USUARIO.

DE LA RESERVACIÓN SE MUESTRAN LOS SIGUIENTES DATOS:

- AULA (AULA 1,2,3,4 Y 5 DE GASTEX, LAB. 1,2,3,4 DE GASTEX, AULA 1, 2 RECTORADO, SALÓN 1,2,3 RECTORADO),
- SESIÓN (MAÑANA, TARDE Y NOCHE),
- RESPONSABLE (CUALQUIER PERSONA UCI AUTENTICADA PREVIAMENTE),
- ÁREA (ÁREA A LA QUE PERTENECE LA PERSONA QUE REALIZA LA SOLICITUD DE RESERVACIÓN)
- ACTIVIDAD (DESCRIPCIÓN DE LA ACTIVIDAD QUE SE VA A REALIZAR EN EL LOCAL RESERVADO).

**OBSERVACIONES:**

CONFIRMADO CON EL CLIENTE.

**PROTOTIPO UI:**

Abril 2009						
L	M	M	J	V	S	D
30	31	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	1	2	3

## SISTEMA DE RESERVAS DE LOCALES PARA LA ACTIVIDAD DE POSTGRADO

**RESERVAS DE LOCALES**

FECHA DE RESERVA : 2/04/2009

AULA	SESION	ACTIVIDAD	RESPONSABLE	AREA
Aula 1 GASTEX	mañana	Discusión de Tesis	Yamilis Fernández	Dirección Formación Posgraduada
	tarde	Discusión de Tesis	Yamilis Fernández	Dirección Formación Posgraduada
	noche	Discusión de Tesis	Yamilis Fernández	Dirección Formación Posgraduada
Aula 2 GASTEX	mañana	Discusión de Tesis	Yamilis Fernández	Dirección Formación Posgraduada
	tarde	Discusión de Tesis	Yamilis Fernández	Dirección Formación Posgraduada
	noche	Discusión de Tesis	Yamilis Fernández	Dirección Formación Posgraduada

Tabla 10 HU Mostrar Reservas de Locales de Posgrado

TAREA DE INGENIERÍA	
<b>NÚMERO DE TAREA:</b> T.21	<b>NUMERO HISTORIA DE USUARIO:</b> HU_2
<b>NOMBRE TAREA:</b> CREAR FUNCIONALIDAD PARA DEVOLVER LOS REGISTROS CORRESPONDIENTES A LAS RESERVACIONES VÍA SERVICIO WEB.	
<b>TIPO DE TAREA:</b> DESARROLLO	<b>PUNTOS ESTIMADOS:</b> 3
<b>FECHA INICIO:</b> 26/02/2009	<b>FECHA FIN:</b> 28/02/2009

<b>PROGRAMADOR RESPONSABLE:</b> <i>ALEXANDER F. VINENT PEÑA</i>
<b>DESCRIPCIÓN:</b> <i>CREAR FUNCIONALIDAD PARA DEVOLVER LOS REGISTROS CORRESPONDIENTES A LAS RESERVACIONES VÍA SERVICIO WEB.</i>

Tabla 11 Tarea de Ingeniería: Crear funcionalidad para devolver los registros correspondientes a las reservaciones vía servicio web.

<b>TAREA DE INGENIERÍA</b>	
<b>NÚMERO DE TAREA:</b> <i>T.22</i>	<b>NUMERO HISTORIA DE USUARIO:</b> <i>HU_2</i>
<b>NOMBRE TAREA:</b> <i>CREAR FUNCIONALIDAD PARA DEVOLVER LOS REGISTROS CORRESPONDIENTES A LAS RESERVACIONES MEDIANTE MÓDULO DE DRUPAL.</i>	
<b>TIPO DE TAREA:</b> <i>DESARROLLO</i>	<b>PUNTOS ESTIMADOS:</b> <i>3</i>
<b>FECHA INICIO:</b> <i>26/03/2009</i>	<b>FECHA FIN:</b> <i>10/04/2009</i>
<b>PROGRAMADOR RESPONSABLE:</b> <i>ALEXANDER F. VINENT PEÑA</i>	
<b>DESCRIPCIÓN:</b> <i>CREAR FUNCIONALIDAD PARA DEVOLVER LOS REGISTROS CORRESPONDIENTES A LAS RESERVACIONES MEDIANTE MÓDULO DE DRUPAL.</i>	

Tabla 12 Tarea de Ingeniería: Crear funcionalidad para devolver los registros correspondientes a las reservaciones mediante módulo de Drupal.


<b>TAREA DE INGENIERÍA</b>	
<b>NÚMERO DE TAREA:</b> <i>T.23</i>	<b>NUMERO HISTORIA DE USUARIO:</b> <i>HU_2</i>
<b>NOMBRE TAREA:</b> <i>DISEÑO UI DE LA FUNCIONALIDAD MOSTRAR RESERVACIONES DE LOCALES DE POSGRADO.</i>	
<b>TIPO DE TAREA:</b> <i>MEJORA</i>	<b>PUNTOS ESTIMADOS:</b> <i>3</i>
<b>FECHA INICIO:</b> <i>24/03/2009</i>	<b>FECHA FIN:</b> <i>31/03/2009</i>
<b>PROGRAMADOR RESPONSABLE:</b> <i>ALEXANDER F. VINENT PEÑA</i>	

**DESCRIPCIÓN:**

*DISEÑO UI DE LA FUNCIONALIDAD MOSTRAR RESERVACIONES DE LOCALES DE POSGRADO.*

Tabla 13 Tarea de Ingeniería: Diseño UI de la funcionalidad Mostrar Reservas de Locales de Posgrado.

**3.1.3. HU FILTRADO DE RESERVACIONES**

<b>NÚMERO:</b> <i>HU.3</i>	<b>USUARIO:</b> <i>ALEXANDER F. VINENT PEÑA</i>	
<b>NOMBRE HISTORIA DE USUARIO:</b> <i>FILTRADO DE RESERVACIONES</i>		
<b>ITERACIÓN ASIGNADA:</b> <i>1RA. ITERACIÓN</i>		
<b>PRIORIDAD EN NEGOCIO:</b> <i>ALTA</i>	<b>PUNTOS ESTIMADOS:</b> <i>10</i>	
<b>RIESGO EN DESARROLLO:</b> <i>MEDIA</i>	<b>PUNTOS REALES:</b> <i>80</i>	
<b>DESCRIPCIÓN:</b> <i>EL SISTEMA MUESTRA UN LISTADO DE LAS RESERVACIONES.          APARECEN LAS SIGUIENTES OPCIONES:</i> <ul style="list-style-type: none"> <li>● <i>FILTRAR RESULTADOS POR LOCAL.</i></li> <li>● <i>FILTRAR RESULTADOS POR SESIÓN DE TRABAJO</i></li> <li>● <i>FILTRAR RESULTADOS POR RESPONSABLE</i></li> <li>● <i>FILTRAR RESULTADOS POR ÁREA DE TRABAJO<sup>7</sup></i></li> </ul>		
<b>OBSERVACIONES:</b> <i>CONFIRMADO CON EL CLIENTE.</i>		
<b>PROTOTIPO UI:</b> 		

<sup>7</sup> No disponible por no existir un servicio web que permita acceder a las áreas definidas dentro de la Universidad.

Tabla 14 HU Filtrado de reservaciones

TAREA DE INGENIERÍA	
<b>NÚMERO DE TAREA:</b> T.31	<b>NUMERO HISTORIA DE USUARIO:</b> HU_3
<b>NOMBRE TAREA:</b> CREAR FUNCIONALIDAD PARA DEVOLVER LOS REGISTROS CORRESPONDIENTES A LAS RESERVACIONES VÍA SERVICIO WEB. FILTRO POR LOCAL.	
<b>TIPO DE TAREA:</b> DESARROLLO	<b>PUNTOS ESTIMADOS:</b> 3
<b>FECHA INICIO:</b> 03/03/2009	<b>FECHA FIN:</b> 06/03/2009
<b>PROGRAMADOR RESPONSABLE:</b> ALEXANDER F. VINENT PEÑA	
<b>DESCRIPCIÓN:</b> CREAR FUNCIONALIDAD PARA DEVOLVER LOS REGISTROS CORRESPONDIENTES A LAS RESERVACIONES VÍA SERVICIO WEB.	

Tabla 15 Tarea de Ingeniería: Filtro por locales . Servicio Web

TAREA DE INGENIERÍA	
<b>NÚMERO DE TAREA:</b> T.32	<b>NUMERO HISTORIA DE USUARIO:</b> HU_3
<b>NOMBRE TAREA:</b> CREAR FUNCIONALIDAD PARA DEVOLVER LOS REGISTROS CORRESPONDIENTES A LAS RESERVACIONES VÍA SERVICIO WEB. FILTRO POR SESIÓN DE TRABAJO.	
<b>TIPO DE TAREA:</b> DESARROLLO	<b>PUNTOS ESTIMADOS:</b> 3
<b>FECHA INICIO:</b> 03/03/2009	<b>FECHA FIN:</b> 06/03/2009
<b>PROGRAMADOR RESPONSABLE:</b> ALEXANDER F. VINENT PEÑA	
<b>DESCRIPCIÓN:</b> CREAR FUNCIONALIDAD PARA DEVOLVER LOS REGISTROS CORRESPONDIENTES A LAS RESERVACIONES VÍA SERVICIO WEB.	

Tabla 16 Tarea de Ingeniería: Filtro por sesión de trabajo . Servicio Web

TAREA DE INGENIERÍA
---------------------

<b>NÚMERO DE TAREA:</b> T.33	<b>NUMERO HISTORIA DE USUARIO:</b> HU_3
<b>NOMBRE TAREA:</b> CREAR FUNCIONALIDAD PARA DEVOLVER LOS REGISTROS CORRESPONDIENTES A LAS RESERVACIONES VÍA SERVICIO WEB. FILTRO POR RESPONSABLE.	
<b>TIPO DE TAREA:</b> DESARROLLO	<b>PUNTOS ESTIMADOS:</b> 3
<b>FECHA INICIO:</b> 10/03/2009	<b>FECHA FIN:</b> 12/03/2009
<b>PROGRAMADOR RESPONSABLE:</b> ALEXANDER F. VINENT PEÑA	
<b>DESCRIPCIÓN:</b> CREAR FUNCIONALIDAD PARA DEVOLVER LOS REGISTROS CORRESPONDIENTES A LAS RESERVACIONES VÍA SERVICIO WEB.	

Tabla 17 Tarea de Ingeniería: Filtro por responsable. Servicio Web

TAREA DE INGENIERÍA	
<b>NÚMERO DE TAREA:</b> T.34	<b>NUMERO HISTORIA DE USUARIO:</b> HU_3
<b>NOMBRE TAREA:</b> CREAR FUNCIONALIDAD PARA DEVOLVER LOS REGISTROS CORRESPONDIENTES A LAS RESERVACIONES VÍA MÓDULO DRUPAL. FILTRO POR RESPONSABLE.	
<b>TIPO DE TAREA:</b> DESARROLLO	<b>PUNTOS ESTIMADOS:</b> 3
<b>FECHA INICIO:</b> 27/03/2009	<b>FECHA FIN:</b> 10/04/2009
<b>PROGRAMADOR RESPONSABLE:</b> ALEXANDER F. VINENT PEÑA	
<b>DESCRIPCIÓN:</b> CREAR FUNCIONALIDAD PARA DEVOLVER LOS REGISTROS CORRESPONDIENTES A LAS RESERVACIONES VÍA MÓDULO DRUPAL.	

Tabla 18 Tarea de Ingeniería: Filtro por responsable. Módulo Drupal

TAREA DE INGENIERÍA	
<b>NÚMERO DE TAREA:</b> T.35	<b>NUMERO HISTORIA DE USUARIO:</b> HU_3

<b>NOMBRE TAREA:</b> <i>CREAR FUNCIONALIDAD PARA DEVOLVER LOS REGISTROS CORRESPONDIENTES A LAS RESERVACIONES VÍA MÓDULO DRUPAL. FILTRO POR SESIÓN DE TRABAJO.</i>	
<b>TIPO DE TAREA:</b> <i>DESARROLLO</i>	<b>PUNTOS ESTIMADOS:</b> 3
<b>FECHA INICIO:</b> <i>27/03/2009</i>	<b>FECHA FIN:</b> <i>10/04/2009</i>
<b>PROGRAMADOR RESPONSABLE:</b> <i>ALEXANDER F. VINENT PEÑA</i>	
<b>DESCRIPCIÓN:</b> <i>CREAR FUNCIONALIDAD PARA DEVOLVER LOS REGISTROS CORRESPONDIENTES A LAS RESERVACIONES VÍA MÓDULO DRUPAL.</i>	

Tabla 19 Tarea de Ingeniería: Filtro por sesión de trabajo. Módulo Drupal

TAREA DE INGENIERÍA	
<b>NÚMERO DE TAREA:</b> <i>T.36</i>	<b>NUMERO HISTORIA DE USUARIO:</b> <i>HU_3</i>
<b>NOMBRE TAREA:</b> <i>CREAR FUNCIONALIDAD PARA DEVOLVER LOS REGISTROS CORRESPONDIENTES A LAS RESERVACIONES VÍA MÓDULO DRUPAL. FILTRO POR RESPONSABLE.</i>	
<b>TIPO DE TAREA:</b> <i>DESARROLLO</i>	<b>PUNTOS ESTIMADOS:</b> 3
<b>FECHA INICIO:</b> <i>27/03/2009</i>	<b>FECHA FIN:</b> <i>10/04/2009</i>
<b>PROGRAMADOR RESPONSABLE:</b> <i>ALEXANDER F. VINENT PEÑA</i>	
<b>DESCRIPCIÓN:</b> <i>CREAR FUNCIONALIDAD PARA DEVOLVER LOS REGISTROS CORRESPONDIENTES A LAS RESERVACIONES VÍA MÓDULO DRUPAL.</i>	

Tabla 20 Tarea de Ingeniería: Filtro por responsable. Módulo Drupal

TAREA DE INGENIERÍA	
<b>NÚMERO DE TAREA:</b> <i>T.37</i>	<b>NUMERO HISTORIA DE USUARIO:</b> <i>HU_3</i>
<b>NOMBRE TAREA:</b> <i>DISEÑO UI DE LA FUNCIONALIDAD PARA DEVOLVER LOS REGISTROS CORRESPONDIENTES A LAS RESERVACIONES.</i>	
<b>TIPO DE TAREA:</b> <i>DESARROLLO</i>	<b>PUNTOS ESTIMADOS:</b> 3



<b>FECHA INICIO:</b> <i>24/03/2009</i>	<b>FECHA FIN:</b> <i>31/03/2009</i>
<b>PROGRAMADOR RESPONSABLE:</b> <i>ALEXANDER F. VINENT PEÑA</i>	
<b>DESCRIPCIÓN:</b> <i>CREAR FUNCIONALIDAD PARA DEVOLVER LOS REGISTROS CORRESPONDIENTES A LAS RESERVACIONES VÍA MÓDULO DRUPAL.</i>	

Tabla 21 Tarea de Ingeniería. Diseño UI Filtro de reservaciones

### 3.1.4. HU GESTIÓN DE SOLICITUDES

<b>NÚMERO:</b> <i>HU.4</i>	<b>USUARIO:</b> <i>ALEXANDER F. VINENT PEÑA</i>		
<b>NOMBRE HISTORIA DE USUARIO:</b> <i>GESTIÓN DE SOLICITUDES</i>			
<b>ITERACIÓN ASIGNADA:</b> <i>1RA. ITERACIÓN</i>			
<b>PRIORIDAD EN NEGOCIO:</b> <i>ALTA</i>		<b>PUNTOS ESTIMADOS:</b> <i>10</i>	
<b>RIESGO EN DESARROLLO:</b> <i>MEDIA</i>		<b>PUNTOS REALES:</b> <i>80</i>	
<b>DESCRIPCIÓN:</b> <i>SE LE MUESTRA AL ADMINISTRADOR UN LISTADO CON TODAS LAS SOLICITUDES QUE SE HAYAN REALIZADO HASTA LA FECHA ACTUAL, BRINDÁNDOLE LA OPCIÓN DE MODIFICAR O ELIMINAR CUALQUIERA DE LAS SOLICITUDES QUE EXISTAN.</i>			
<b>OBSERVACIONES:</b> <i>CONFIRMADO CON EL CLIENTE.</i>			
<b>PROTOTIPO UI:</b>			

Abril 2009

L	M	M	J	V	S	D
30	31	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	1	2	3

## SISTEMA DE RESERVAS DE LOCALES PARA LA ACTIVIDAD DE POSTGRADO

**SOLICITUDES DE LOCALES** FECHA : 09/04/2009

	RESPONSABLE	ACTIVIDAD	PROCESAR	MODIFICAR	ELIMINAR
	1. Yamilis Fernández	Discusión de Tesis			
	2. Yamilis Fernández	Discusión de Tesis			
	3. Yamilis Fernández	Discusión de Tesis			
	4. Yamilis Fernández	Discusión de Tesis			
	5. Yamilis Fernández	Discusión de Tesis			
	6. Yamilis Fernández	Discusión de Tesis			

Tabla 22 HU Gestión de solicitudes

La historia de usuario definida en la tabla anterior, se decidió dividir en historias más pequeñas, las cuales se muestran a continuación.

### 3.1.4.1. HU GESTIÓN DE SOLICITUDES. MODIFICAR SOLICITUD

<b>NÚMERO:</b> <i>HU.5</i>	<b>USUARIO:</b> <i>ALEXANDER F. VINENT PEÑA</i>
<b>NOMBRE HISTORIA DE USUARIO:</b> <i>GESTIÓN DE SOLICITUDES. MODIFICAR SOLICITUD</i>	
<b>ITERACIÓN ASIGNADA:</b> <i>1RA. ITERACIÓN</i>	
<b>PRIORIDAD EN NEGOCIO:</b> <i>ALTA</i>	<b>PUNTOS ESTIMADOS:</b> <i>10</i>
<b>RIESGO EN DESARROLLO:</b> <i>MEDIA</i>	<b>PUNTOS REALES:</b> <i>80</i>
<b>DESCRIPCIÓN:</b> <i>ÉL SISTEMA MUESTRA LOS DATOS DE LA SOLICITUD SELECCIONADA.</i>	
<b>OBSERVACIONES:</b> <i>CONFIRMADO CON EL CLIENTE.</i>	
<b>PROTOTIPO UI:</b>	

Abril 2009

L	M	M	J	V	S	D
30	31	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	1	2	3

## SISTEMA DE RESERVAS DE LOCALES PARA LA ACTIVIDAD DE POSTGRADO

**RESERVAR LOCAL**      FECHA DE RESERVA: 2/04/2009

TIPO DE LOCAL	SESION	RESPONSABLE	AREA
Aula 1 GASTEX <input type="button" value="▼"/>	mañana <input type="button" value="▼"/>	Yamills Fernández	Dirección Formación Posgraduada
<p><b>DESCRIPCION DE LA ACTIVIDAD</b></p> <div style="border: 1px solid #ccc; padding: 5px; min-height: 40px;">           Discusión de Tesis         </div>			
<input type="button" value="GUARDAR DATOS"/> <input type="button" value="↕"/>			

Tabla 23 HU Gestión de solicitudes. Modificar solicitud

<b>TAREA DE INGENIERÍA</b>	
<b>NÚMERO DE TAREA:</b> <i>T.51</i>	<b>NÚMERO HISTORIA DE USUARIO:</b> <i>HU_5</i>
<b>NOMBRE TAREA:</b> <i>CREAR FUNCIONALIDAD PARA MODIFICAR LA SOLICITUD VÍA SERVICIO WEB.</i>	
<b>TIPO DE TAREA:</b> <i>DESARROLLO</i>	<b>PUNTOS ESTIMADOS:</b> <i>3</i>
<b>FECHA INICIO:</b> <i>24/02/2009</i>	<b>FECHA FIN:</b> <i>26/02/2009</i>
<b>PROGRAMADOR RESPONSABLE:</b> <i>ALEXANDER F. VINENT PEÑA</i>	
<b>DESCRIPCIÓN:</b> <i>CREAR FUNCIONALIDAD PARA MODIFICAR LA SOLICITUD VÍA SERVICIO WEB.</i>	

Tabla 24 Tarea de Ingeniería: Crear funcionalidad para modificar la solicitud vía servicio web.

<b>TAREA DE INGENIERÍA</b>	
<b>NÚMERO DE TAREA:</b> <i>T.52</i>	<b>NÚMERO HISTORIA DE USUARIO:</b> <i>HU_5</i>
<b>NOMBRE TAREA:</b> <i>CREAR FUNCIONALIDAD PARA MODIFICAR LA SOLICITUD MEDIANTE MÓDULO DE DRUPAL.</i>	

<b>TIPO DE TAREA:</b> <i>DESARROLLO</i>	<b>PUNTOS ESTIMADOS:</b> 3
<b>FECHA INICIO:</b> <i>25/03/2009</i>	<b>FECHA FIN:</b> <i>10/04/2009</i>
<b>PROGRAMADOR RESPONSABLE:</b> <i>ALEXANDER F. VINENT PEÑA</i>	
<b>DESCRIPCIÓN:</b> <i>CREAR FUNCIONALIDAD PARA MODIFICAR LA SOLICITUD MEDIANTE MÓDULO DE DRUPAL VÍA SERVICIO WEB.</i>	

Tabla 25 Tarea de Ingeniería: Crear funcionalidad para Modificar solicitud mediante módulo de Drupal.

<b>TAREA DE INGENIERÍA</b>	
<b>NÚMERO DE TAREA:</b> <i>T.53</i>	<b>NÚMERO HISTORIA DE USUARIO:</b> <i>HU_5</i>
<b>NOMBRE TAREA:</b> <i>DISEÑO UI DE LA FUNCIONALIDAD MODIFICAR SOLICITUD.</i>	
<b>TIPO DE TAREA:</b> <i>MEJORA</i>	<b>PUNTOS ESTIMADOS:</b> 3
<b>FECHA INICIO:</b> <i>24/03/2009</i>	<b>FECHA FIN:</b> <i>31/03/2009</i>
<b>PROGRAMADOR RESPONSABLE:</b> <i>ALEXANDER F. VINENT PEÑA</i>	
<b>DESCRIPCIÓN:</b> <i>DISEÑO UI DE LA FUNCIONALIDAD MODIFICAR SOLICITUD.</i>	

Tabla 26 Tarea de Ingeniería: Diseño UI de la funcionalidad Modificar solicitud.

<b>TAREA DE INGENIERÍA</b>	
<b>NÚMERO DE TAREA:</b> <i>T.53</i>	<b>NUMERO HISTORIA DE USUARIO:</b> <i>HU_5</i>
<b>NOMBRE TAREA:</b> <i>NOTIFICAR USUARIO SOBRE RESERVACIÓN ACEPTADA.</i>	
<b>TIPO DE TAREA:</b> <i>DESARROLLO</i>	<b>PUNTOS ESTIMADOS:</b> 3
<b>FECHA INICIO:</b> <i>24/03/2009</i>	<b>FECHA FIN:</b> <i>31/03/2009</i>

**PROGRAMADOR RESPONSABLE:**

ALEXANDER F. VINENT PEÑA

**DESCRIPCIÓN:**

NOTIFICAR AL USUARIO INTERESADO QUE LA SOLICITUD DE RESERVACIÓN FUE MODIFICADA.

Tabla 27 Tarea de Ingeniería: Notificar usuario.

**3.1.4.2. HU GESTIÓN DE SOLICITUDES. ELIMINAR SOLICITUD**

<b>NÚMERO:</b> HU.6	<b>USUARIO:</b> ALEXANDER F. VINENT PEÑA
<b>NOMBRE HISTORIA DE USUARIO:</b> GESTIÓN DE SOLICITUDES. ELIMINAR SOLICITUD	
<b>ITERACIÓN ASIGNADA:</b> 1RA. ITERACIÓN	
<b>PRIORIDAD EN NEGOCIO:</b> ALTA	<b>PUNTOS ESTIMADOS:</b> 10
<b>RIESGO EN DESARROLLO:</b> MEDIA	<b>PUNTOS REALES:</b> 80
<b>DESCRIPCIÓN:</b> CUANDO EL ADMINISTRADOR SELECCIONA LA OPCIÓN ELIMINAR SOLICITUD, EL SISTEMA MUESTRA UN MENSAJE DE CONFIRMACIÓN.	
<b>OBSERVACIONES:</b> CONFIRMADO CON EL CLIENTE.	
<b>PROTOTIPO UI:</b>	

**SISTEMA DE RESERVAS DE LOCALES PARA LA ACTIVIDAD DE POSTGRADO**

ABRIL 2009

L	M	M	J	V	S	D
30	31	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	1	2	3

SOLICITUDES DE LOCALES      FECHA : 09/04/2009

RESPONSABLE	ACTIVIDAD	APROBAR	MODIFICAR	ELIMINAR
1. Yamilis Fernández	Discusión de Tesis			
2. Yar...				
3. Yar...				
4. Yar...				
5. Yar...				
6. Yamilis Fernández	Discusión de Tesis			

**¿Está seguro de eliminar la solicitud? Los datos se perderán.**

RESPONSABLE	ACTIVIDAD	FECHA
Yamilis Fernández	Discusión de Tesis	11 Abril 2009

CONFIRMAR ELIMINAR      CANCELAR

Tabla 28 HU Gestión de solicitudes. Eliminar solicitud

TAREA DE INGENIERÍA	
<b>NÚMERO DE TAREA:</b> T.61	<b>NÚMERO HISTORIA DE USUARIO:</b> HU_6
<b>NOMBRE TAREA:</b> CREAR FUNCIONALIDAD PARA ELIMINAR LA SOLICITUD VÍA SERVICIO WEB.	
<b>TIPO DE TAREA:</b> DESARROLLO	<b>PUNTOS ESTIMADOS:</b> 3
<b>FECHA INICIO:</b> 24/02/2009	<b>FECHA FIN:</b> 26/02/2009
<b>PROGRAMADOR RESPONSABLE:</b> ALEXANDER F. VINENT PEÑA	
<b>DESCRIPCIÓN:</b> CREAR FUNCIONALIDAD PARA ELIMINAR LA SOLICITUD VÍA SERVICIO WEB.	

Tabla 29 Tarea de Ingeniería: Crear funcionalidad para eliminar la solicitud vía servicio web.

TAREA DE INGENIERÍA	
<b>NÚMERO DE TAREA:</b> T.62	<b>NÚMERO HISTORIA DE USUARIO:</b> HU_6
<b>NOMBRE TAREA:</b> CREAR FUNCIONALIDAD PARA ELIMINAR LA SOLICITUD MEDIANTE MÓDULO DE DRUPAL.	
<b>TIPO DE TAREA:</b> DESARROLLO	<b>PUNTOS ESTIMADOS:</b> 3
<b>FECHA INICIO:</b> 25/03/2009	<b>FECHA FIN:</b> 10/04/2009
<b>PROGRAMADOR RESPONSABLE:</b> ALEXANDER F. VINENT PEÑA	
<b>DESCRIPCIÓN:</b> CREAR FUNCIONALIDAD PARA ELIMINAR LA SOLICITUD MEDIANTE MÓDULO DE DRUPAL VÍA SERVICIO WEB.	

Tabla 30 Tarea de Ingeniería: Crear funcionalidad para eliminar solicitud mediante módulo de Drupal.

TAREA DE INGENIERÍA	
<b>NÚMERO DE TAREA:</b> T.63	<b>NÚMERO HISTORIA DE USUARIO:</b> HU_6

<b>NOMBRE TAREA:</b> <i>DISEÑO UI DE LA FUNCIONALIDAD ELIMINAR SOLICITUD.</i>	
<b>TIPO DE TAREA:</b> <i>MEJORA</i>	<b>PUNTOS ESTIMADOS:</b> <i>3</i>
<b>FECHA INICIO:</b> <i>24/03/2009</i>	<b>FECHA FIN:</b> <i>31/03/2009</i>
<b>PROGRAMADOR RESPONSABLE:</b> <i>ALEXANDER F. VINENT PEÑA</i>	
<b>DESCRIPCIÓN:</b> <i>DISEÑO UI DE LA FUNCIONALIDAD ELIMINAR SOLICITUD.</i>	

Tabla 31 Tarea de Ingeniería: Diseño UI de la funcionalidad Eliminar solicitud.

<b>TAREA DE INGENIERÍA</b>	
<b>NÚMERO DE TAREA:</b> <i>T.63</i>	<b>NUMERO HISTORIA DE USUARIO:</b> <i>HU_6</i>
<b>NOMBRE TAREA:</b> <i>NOTIFICAR USUARIO SOBRE SOLICITUD ELIMINADA.</i>	
<b>TIPO DE TAREA:</b> <i>DESARROLLO</i>	<b>PUNTOS ESTIMADOS:</b> <i>3</i>
<b>FECHA INICIO:</b> <i>24/03/2009</i>	<b>FECHA FIN:</b> <i>31/03/2009</i>
<b>PROGRAMADOR RESPONSABLE:</b> <i>ALEXANDER F. VINENT PEÑA</i>	
<b>DESCRIPCIÓN:</b> <i>NOTIFICAR AL USUARIO INTERESADO QUE LA SOLICITUD DE RESERVACIÓN FUE ELIMINADA.</i>	

Tabla 32 Tarea de Ingeniería: Notificar usuario.

### 3.1.5. HU GESTIÓN DE RESERVACIONES

<b>NÚMERO:</b> <i>HU.7</i>	<b>USUARIO:</b> <i>ALEXANDER F. VINENT PEÑA</i>
<b>NOMBRE HISTORIA DE USUARIO:</b> <i>GESTIÓN DE RESERVACIONES</i>	
<b>ITERACIÓN ASIGNADA:</b> <i>1RA. ITERACIÓN</i>	
<b>PRIORIDAD EN NEGOCIO:</b> <i>ALTA</i>	<b>PUNTOS ESTIMADOS:</b> <i>10</i>

<b>RIESGO EN DESARROLLO:</b> <i>MEDIA</i>	<b>PUNTOS REALES:</b> <i>80</i>																																			
<b>DESCRIPCIÓN:</b> SE LE MUESTRA AL ADMINISTRADOR UN LISTADO DE LAS RESERVACIONES QUE SE HAN REALIZADO BRINDÁNDOLE LA OPCIÓN DE AGREGAR, MODIFICAR O ELIMINAR CUALQUIERA DE LAS RESERVACIONES QUE EXISTAN.																																				
<b>OBSERVACIONES:</b> <i>CONFIRMADO CON EL CLIENTE.</i>																																				
<b>PROTOTIPO UI:</b>																																				
<p><b>SISTEMA DE RESERVAS DE LOCALES PARA LA ACTIVIDAD DE POSTGRADO</b></p> <p><b>RESERVACIONES DE LOCALES</b>      <b>FECHA :</b> 09/04/2009</p> <table border="1"> <thead> <tr> <th>RESPONSABLE</th> <th>ACTIVIDAD</th> <th>DETALLES</th> <th>MODIFICAR</th> <th>ELIMINAR</th> </tr> </thead> <tbody> <tr> <td>1. Yamilis Fernández</td> <td>Discusión de Tesis</td> <td></td> <td></td> <td></td> </tr> <tr> <td>2. Yamilis Fernández</td> <td>Discusión de Tesis</td> <td></td> <td></td> <td></td> </tr> <tr> <td>3. Yamilis Fernández</td> <td>Discusión de Tesis</td> <td></td> <td></td> <td></td> </tr> <tr> <td>4. Yamilis Fernández</td> <td>Discusión de Tesis</td> <td></td> <td></td> <td></td> </tr> <tr> <td>5. Yamilis Fernández</td> <td>Discusión de Tesis</td> <td></td> <td></td> <td></td> </tr> <tr> <td>6. Yamilis Fernández</td> <td>Discusión de Tesis</td> <td></td> <td></td> <td></td> </tr> </tbody> </table>		RESPONSABLE	ACTIVIDAD	DETALLES	MODIFICAR	ELIMINAR	1. Yamilis Fernández	Discusión de Tesis				2. Yamilis Fernández	Discusión de Tesis				3. Yamilis Fernández	Discusión de Tesis				4. Yamilis Fernández	Discusión de Tesis				5. Yamilis Fernández	Discusión de Tesis				6. Yamilis Fernández	Discusión de Tesis			
RESPONSABLE	ACTIVIDAD	DETALLES	MODIFICAR	ELIMINAR																																
1. Yamilis Fernández	Discusión de Tesis																																			
2. Yamilis Fernández	Discusión de Tesis																																			
3. Yamilis Fernández	Discusión de Tesis																																			
4. Yamilis Fernández	Discusión de Tesis																																			
5. Yamilis Fernández	Discusión de Tesis																																			
6. Yamilis Fernández	Discusión de Tesis																																			

Tabla 33 HU Gestión de solicitudes

La historia de usuario definida en la tabla anterior, se decidió dividir en historias más pequeñas, las cuales se muestran a continuación.

### 3.1.5.1. HU GESTIÓN DE RESERVACIONES. DETALLES DE RESERVACIÓN

<b>NÚMERO:</b> <i>HU.8</i>	<b>USUARIO:</b> <i>ALEXANDER F. VINENT PEÑA</i>
<b>NOMBRE HISTORIA DE USUARIO:</b> <i>GESTIÓN DE RESERVACIONES. DETALLES DE RESERVACIÓN</i>	
<b>ITERACIÓN ASIGNADA:</b> <i>2DA. ITERACIÓN</i>	
<b>PRIORIDAD EN NEGOCIO:</b> <i>ALTA</i>	<b>PUNTOS ESTIMADOS:</b> <i>10</i>
<b>RIESGO EN DESARROLLO:</b> <i>MEDIA</i>	<b>PUNTOS REALES:</b> <i>80</i>



**DESCRIPCIÓN:**

SE LE MUESTRA AL ADMINISTRADOR LA INFORMACIÓN COMPLETA PERTENECIENTE A LA RESERVACIÓN SELECCIONADA:  
SE MUESTRAN LOS CAMPOS:

- FECHA DE RESERVACIÓN
- RESPONSABLE
- DESCRIPCIÓN
- LOCAL ASIGNADO
- SESIÓN

**OBSERVACIONES:**

CONFIRMADO CON EL CLIENTE.

**PROTOTIPO UI:**

Tabla 34 HU Gestión de reservaciones. Detalles de la reservación.

### 3.1.5.2. HU GESTIÓN DE RESERVACIONES. ADICIONAR RESERVACIÓN

<b>NÚMERO:</b> HU.9	<b>USUARIO:</b> ALEXANDER F. VINENT PEÑA	
<b>NOMBRE HISTORIA DE USUARIO:</b> GESTIÓN DE RESERVACIONES. ADICIONAR RESERVACIÓN		
<b>ITERACIÓN ASIGNADA:</b> 2DA. ITERACIÓN		
<b>PRIORIDAD EN NEGOCIO:</b> ALTA	<b>PUNTOS ESTIMADOS:</b> 10	
<b>RIESGO EN DESARROLLO:</b> MEDIA	<b>PUNTOS REALES:</b> 80	
<b>DESCRIPCIÓN:</b> PROCESAR SOLICITUD DE RESERVACIÓN DE LOCALES DE POSTGRADO. SE LE MUESTRA AL ADMINISTRADOR UN LISTADO CON TODAS LAS SOLICITUDES QUE SE HAYAN REALIZADO HASTA LA FECHA ACTUAL: DE LAS SOLICITUDES SE MUESTRAN LOS CAMPOS: <ul style="list-style-type: none"> <li>• FECHA DE SOLICITUD</li> <li>• RESPONSABLE</li> <li>• DESCRIPCIÓN</li> <li>• TIPO DE LOCAL</li> <li>• SESIÓN</li> <li>• LOCAL (LISTADO DE LOCALES EXISTENTES EN POSTGRADO)</li> </ul> PERMITIENDO ASIGNARLE UN LOCAL SEGÚN EL TIPO DE LOCAL SOLICITADO POR EL USUARIO. EL ADMINISTRADOR ASIGNA UN LOCAL A LA SOLICITUD Y PASA A CONVERTIRSE DICHA SOLICITUD EN UNA RESERVACIÓN. NOTIFICANDO ESTA AL USUARIO INTERESADO.		
<b>OBSERVACIONES:</b> CONFIRMADO CON EL CLIENTE.		
<b>PROTOTIPO UI:</b>		

Abril 2009

L	M	M	J	V	S	D
30	31	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	1	2	3

## SISTEMA DE RESERVAS DE LOCALES PARA LA ACTIVIDAD DE POSTGRADO

**PROCESAR SOLICITUD DE RESERVACION DE LOCAL**    FECHA DE RESERVA : 2/04/2009

LOCAL QUE SOLICITA	ASIGNAR LOCAL	RESPONSABLE	AREA
<input type="text" value="Aula"/>	<input type="text" value="Aula 1 GASTEX"/> ↓	Yamilis Fernández	Dirección Formación Posgraduada
<b>SESION</b>			
<input type="text" value="mañana"/>			
<b>DESCRIPCION DE LA ACTIVIDAD</b>			
<input type="text" value="Discusión de Tesis"/>			



Tabla 35 HU Gestión de solicitudes. Adicionar reservación.

<b>TAREA DE INGENIERÍA</b>	
<b>NÚMERO DE TAREA:</b> <i>T.91</i>	<b>NUMERO HISTORIA DE USUARIO:</b> <i>HU_9</i>
<b>NOMBRE TAREA:</b> <i>CREAR FUNCIONALIDAD PARA ADICIONAR RESERVACIÓN VÍA SERVICIO WEB.</i>	
<b>TIPO DE TAREA:</b> <i>DESARROLLO</i>	<b>PUNTOS ESTIMADOS:</b> <i>3</i>
<b>FECHA INICIO:</b> <i>24/02/2009</i>	<b>FECHA FIN:</b> <i>26/02/2009</i>
<b>PROGRAMADOR RESPONSABLE:</b> <i>ALEXANDER F. VINENT PEÑA</i>	
<b>DESCRIPCIÓN:</b> <i>CREAR FUNCIONALIDAD PARA ADICIONAR RESERVACIÓN VÍA SERVICIO WEB.</i>	

Tabla 36 Tarea de Ingeniería: Crear funcionalidad para registrar solicitud mediante servicio web.

<b>TAREA DE INGENIERÍA</b>	
<b>NÚMERO DE TAREA:</b> <i>T.92</i>	<b>NUMERO HISTORIA DE USUARIO:</b> <i>HU_9</i>
<b>NOMBRE TAREA:</b> <i>CREAR FUNCIONALIDAD PARA ADICIONAR RESERVACIÓN MEDIANTE MÓDULO DE DRUPAL.</i>	

<b>TIPO DE TAREA:</b> <i>DESARROLLO</i>	<b>PUNTOS ESTIMADOS:</b> 3
<b>FECHA INICIO:</b> <i>25/03/2009</i>	<b>FECHA FIN:</b> <i>10/04/2009</i>
<b>PROGRAMADOR RESPONSABLE:</b> <i>ALEXANDER F. VINENT PEÑA</i>	
<b>DESCRIPCIÓN:</b> <i>CREAR FUNCIONALIDAD PARA ADICIONAR RESERVACIÓN MEDIANTE MÓDULO DE DRUPAL VÍA SERVICIO WEB.</i>	

Tabla 37 Tarea de Ingeniería: Crear funcionalidad para registrar solicitud mediante módulo de Drupal.

<b>TAREA DE INGENIERÍA</b>	
<b>NÚMERO DE TAREA:</b> <i>T.93</i>	<b>NUMERO HISTORIA DE USUARIO:</b> <i>HU_9</i>
<b>NOMBRE TAREA:</b> <i>DISEÑO UI DE LA FUNCIONALIDAD ADICIONAR RESERVACIÓN.</i>	
<b>TIPO DE TAREA:</b> <i>MEJORA</i>	<b>PUNTOS ESTIMADOS:</b> 3
<b>FECHA INICIO:</b> <i>24/03/2009</i>	<b>FECHA FIN:</b> <i>31/03/2009</i>
<b>PROGRAMADOR RESPONSABLE:</b> <i>ALEXANDER F. VINENT PEÑA</i>	
<b>DESCRIPCIÓN:</b> <i>DISEÑO UI DE LA FUNCIONALIDAD ADICIONAR RESERVACIÓN.</i>	

Tabla 38 Tarea de Ingeniería: Diseño UI de la funcionalidad registrar solicitud.

<b>TAREA DE INGENIERÍA</b>	
<b>NÚMERO DE TAREA:</b> <i>T.93</i>	<b>NUMERO HISTORIA DE USUARIO:</b> <i>HU_9</i>
<b>NOMBRE TAREA:</b> <i>NOTIFICAR USUARIO SOBRE SOLICITUD REALIZADA.</i>	
<b>TIPO DE TAREA:</b> <i>DESARROLLO</i>	<b>PUNTOS ESTIMADOS:</b> 3
<b>FECHA INICIO:</b> <i>20/05/2009</i>	<b>FECHA FIN:</b> <i>20/05/2009</i>
<b>PROGRAMADOR RESPONSABLE:</b> <i>ALEXANDER F. VINENT PEÑA</i>	

**DESCRIPCIÓN:**

NOTIFICAR AL USUARIO INTERESADO QUE LA SOLICITUD DE RESERVACIÓN FUE ACEPTADA.

Tabla 39 Tarea de Ingeniería: Notificar usuario.

### 3.1.5.3. HU GESTIÓN DE RESERVACIONES. MODIFICAR RESERVACIÓN

<b>NÚMERO:</b> HU.10	<b>USUARIO:</b> ALEXANDER F. VINENT PEÑA																																																								
<b>NOMBRE HISTORIA DE USUARIO:</b> GESTIÓN DE RESERVACIONES. MODIFICAR RESERVACIÓN																																																									
<b>ITERACIÓN ASIGNADA:</b> 1RA. ITERACIÓN																																																									
<b>PRIORIDAD EN NEGOCIO:</b> ALTA		<b>PUNTOS ESTIMADOS:</b> 10																																																							
<b>RIESGO EN DESARROLLO:</b> MEDIA		<b>PUNTOS REALES:</b> 80																																																							
<b>DESCRIPCIÓN:</b> EL SISTEMA MUESTRA LOS DATOS DE LA RESERVACIÓN SELECCIONADA: <ul style="list-style-type: none"> <li>• TIPO DE LOCAL QUE SOLICITA</li> <li>• LOCAL ASIGNADO</li> <li>• SESIÓN</li> <li>• DATOS DEL RESPONSABLE (NOMBRES Y APELLIDOS, IMAGEN DEL USUARIO)</li> <li>• ÁREA A LA QUE PERTENECE</li> <li>• DESCRIPCIÓN DE LA ACTIVIDAD A REALIZAR</li> </ul>																																																									
<b>OBSERVACIONES:</b> CONFIRMADO CON EL CLIENTE.																																																									
<b>PROTOTIPO UI:</b>																																																									
<div style="display: flex; justify-content: space-between; align-items: flex-start;"> <div style="text-align: center;"> <p>Abril 2009</p> <table border="1"> <thead> <tr> <th>L</th> <th>M</th> <th>M</th> <th>J</th> <th>V</th> <th>S</th> <th>D</th> </tr> </thead> <tbody> <tr> <td>30</td> <td>31</td> <td>1</td> <td style="background-color: #e0f2f1;">2</td> <td>3</td> <td>4</td> <td>5</td> </tr> <tr> <td>6</td> <td>7</td> <td>8</td> <td>9</td> <td>10</td> <td>11</td> <td>12</td> </tr> <tr> <td>13</td> <td>14</td> <td>15</td> <td>16</td> <td>17</td> <td>18</td> <td>19</td> </tr> <tr> <td>20</td> <td>21</td> <td>22</td> <td>23</td> <td>24</td> <td>25</td> <td>26</td> </tr> <tr> <td>27</td> <td>28</td> <td>29</td> <td>30</td> <td>1</td> <td>2</td> <td>3</td> </tr> </tbody> </table> </div> <div style="text-align: center;"> <h2>SISTEMA DE RESERVAS DE LOCALES PARA LA ACTIVIDAD DE POSTGRADO</h2> </div> </div> <div style="margin-top: 10px;"> <p><b>MODIFICAR DATOS DE RESERVACION DE LOCAL</b>    FECHA DE RESERVA : 2/04/2009</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 25%;"><b>LOCAL QUE SOLICITA</b> Aula</td> <td style="width: 25%;"><b>ASIGNAR LOCAL</b> Aula 1 GASTEX</td> <td style="width: 25%;"><b>RESPONSABLE</b> Yamilis Fernández</td> <td style="width: 25%;"><b>AREA</b> Dirección Formación Posgraduada</td> </tr> <tr> <td colspan="4"><b>SESION</b> mañana</td> </tr> <tr> <td colspan="4"><b>DESCRIPCION DE LA ACTIVIDAD</b> Discusión de Tesis</td> </tr> </table> <div style="text-align: center; margin-top: 10px;">  </div> <div style="text-align: center; margin-top: 10px;"> <input type="button" value="GUARDAR DATOS"/> </div> </div>				L	M	M	J	V	S	D	30	31	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	1	2	3	<b>LOCAL QUE SOLICITA</b> Aula	<b>ASIGNAR LOCAL</b> Aula 1 GASTEX	<b>RESPONSABLE</b> Yamilis Fernández	<b>AREA</b> Dirección Formación Posgraduada	<b>SESION</b> mañana				<b>DESCRIPCION DE LA ACTIVIDAD</b> Discusión de Tesis			
L	M	M	J	V	S	D																																																			
30	31	1	2	3	4	5																																																			
6	7	8	9	10	11	12																																																			
13	14	15	16	17	18	19																																																			
20	21	22	23	24	25	26																																																			
27	28	29	30	1	2	3																																																			
<b>LOCAL QUE SOLICITA</b> Aula	<b>ASIGNAR LOCAL</b> Aula 1 GASTEX	<b>RESPONSABLE</b> Yamilis Fernández	<b>AREA</b> Dirección Formación Posgraduada																																																						
<b>SESION</b> mañana																																																									
<b>DESCRIPCION DE LA ACTIVIDAD</b> Discusión de Tesis																																																									

Tabla 40 HU Gestión de reservas. Modificar reservación

TAREA DE INGENIERÍA	
<b>NÚMERO DE TAREA:</b> <i>T.101</i>	<b>NÚMERO HISTORIA DE USUARIO:</b> <i>HU_10</i>
<b>NOMBRE TAREA:</b> <i>CREAR FUNCIONALIDAD PARA MODIFICAR LA RESERVACIÓN VÍA SERVICIO WEB.</i>	
<b>TIPO DE TAREA:</b> <i>DESARROLLO</i>	<b>PUNTOS ESTIMADOS:</b> <i>3</i>
<b>FECHA INICIO:</b> <i>24/02/2009</i>	<b>FECHA FIN:</b> <i>26/02/2009</i>
<b>PROGRAMADOR RESPONSABLE:</b> <i>ALEXANDER F. VINENT PEÑA</i>	
<b>DESCRIPCIÓN:</b> <i>CREAR FUNCIONALIDAD PARA MODIFICAR LA SOLICITUD VÍA SERVICIO WEB.</i>	

Tabla 41 Tarea de Ingeniería: Crear funcionalidad para modificar la reservación vía servicio web.

TAREA DE INGENIERÍA	
<b>NÚMERO DE TAREA:</b> <i>T.102</i>	<b>NÚMERO HISTORIA DE USUARIO:</b> <i>HU_10</i>
<b>NOMBRE TAREA:</b> <i>CREAR FUNCIONALIDAD PARA MODIFICAR LA RESERVACIÓN MEDIANTE MÓDULO DE DRUPAL.</i>	
<b>TIPO DE TAREA:</b> <i>DESARROLLO</i>	<b>PUNTOS ESTIMADOS:</b> <i>3</i>
<b>FECHA INICIO:</b> <i>25/03/2009</i>	<b>FECHA FIN:</b> <i>10/04/2009</i>
<b>PROGRAMADOR RESPONSABLE:</b> <i>ALEXANDER F. VINENT PEÑA</i>	
<b>DESCRIPCIÓN:</b> <i>CREAR FUNCIONALIDAD PARA MODIFICAR LA RESERVACIÓN MEDIANTE MÓDULO DE DRUPAL VÍA SERVICIO WEB.</i>	

Tabla 42 Tarea de Ingeniería: Crear funcionalidad para Modificar reservación mediante módulo de Drupal.

TAREA DE INGENIERÍA	
<b>NÚMERO DE TAREA:</b> <i>T.93</i>	<b>NÚMERO HISTORIA DE USUARIO:</b> <i>HU_9</i>

<b>NOMBRE TAREA:</b> <i>DISEÑO UI DE LA FUNCIONALIDAD MODIFICAR RESERVACIÓN.</i>	
<b>TIPO DE TAREA:</b> <i>MEJORA</i>	<b>PUNTOS ESTIMADOS:</b> <i>3</i>
<b>FECHA INICIO:</b> <i>24/03/2009</i>	<b>FECHA FIN:</b> <i>31/03/2009</i>
<b>PROGRAMADOR RESPONSABLE:</b> <i>ALEXANDER F. VINENT PEÑA</i>	
<b>DESCRIPCIÓN:</b> <i>DISEÑO UI DE LA FUNCIONALIDAD MODIFICAR RESERVACIÓN.</i>	

Tabla 43 Tarea de Ingeniería: Diseño UI de la funcionalidad Modificar reservación.

<b>TAREA DE INGENIERÍA</b>	
<b>NÚMERO DE TAREA:</b> <i>T.104</i>	<b>NUMERO HISTORIA DE USUARIO:</b> <i>HU_10</i>
<b>NOMBRE TAREA:</b> <i>NOTIFICAR USUARIO SOBRE RESERVACIÓN MODIFICADA.</i>	
<b>TIPO DE TAREA:</b> <i>DESARROLLO</i>	<b>PUNTOS ESTIMADOS:</b> <i>3</i>
<b>FECHA INICIO:</b> <i>20/05/2009</i>	<b>FECHA FIN:</b> <i>20/05/2009</i>
<b>PROGRAMADOR RESPONSABLE:</b> <i>ALEXANDER F. VINENT PEÑA</i>	
<b>DESCRIPCIÓN:</b> <i>NOTIFICAR AL USUARIO INTERESADO QUE LA SOLICITUD DE RESERVACIÓN FUE MODIFICADA.</i>	

Tabla 44 Tarea de Ingeniería: Notificar usuario.

### 3.1.5.4. HU GESTIÓN DE RESERVACIONES. ELIMINAR RESERVACIÓN

<b>NÚMERO:</b> <i>HU.11</i>	<b>USUARIO:</b> <i>ALEXANDER F. VINENT PEÑA</i>
<b>NOMBRE HISTORIA DE USUARIO:</b> <i>GESTIÓN DE RESERVACIONES. ELIMINAR RESERVACIÓN</i>	
<b>ITERACIÓN ASIGNADA:</b> <i>2DA. ITERACIÓN</i>	

<b>PRIORIDAD EN NEGOCIO:</b> ALTA	<b>PUNTOS ESTIMADOS:</b> 10						
<b>RIESGO EN DESARROLLO:</b> MEDIA	<b>PUNTOS REALES:</b> 80						
<b>DESCRIPCIÓN:</b> CUANDO EL ADMINISTRADOR SELECCIONA LA OPCIÓN ELIMINAR SOLICITUD, EL SISTEMA MUESTRA UN MENSAJE DE CONFIRMACIÓN.							
<b>OBSERVACIONES:</b> CONFIRMADO CON EL CLIENTE.							
<b>PROTOTIPO UI:</b>							
<p>RESERVACIONES DE LOCALES      FECHA : 09/04/2009</p> <p>RESPONSABLE    ACTIVIDAD    DETALLES    MODIFICAR    ELIMINAR</p> <p>1. Yamilis Fernández    Discusión de Tesis    [?]    [✎]    [✖]</p> <p>2. Y.    [✖]</p> <p>3. Y.    [✖]</p> <p>4. Y.    [✖]</p> <p>5. Y.    [✖]</p> <p>6. Y.    [✖]</p> <p>¿Está seguro de eliminar la reservación? Los datos se perderán.</p> <table border="1"> <thead> <tr> <th>RESPONSABLE</th> <th>ACTIVIDAD</th> <th>FECHA</th> </tr> </thead> <tbody> <tr> <td>Yamilis Fernández</td> <td>Discusión de Tesis</td> <td>11 Abril 2009</td> </tr> </tbody> </table> <p>CONFIRMAR ELIMINAR [✓]    CANCELAR [✖]</p>		RESPONSABLE	ACTIVIDAD	FECHA	Yamilis Fernández	Discusión de Tesis	11 Abril 2009
RESPONSABLE	ACTIVIDAD	FECHA					
Yamilis Fernández	Discusión de Tesis	11 Abril 2009					

Tabla 45 HU Gestión de reservaciones. Eliminar reservación

TAREA DE INGENIERÍA	
<b>NÚMERO DE TAREA:</b> T.111	<b>NÚMERO HISTORIA DE USUARIO:</b> HU_11
<b>NOMBRE TAREA:</b> CREAR FUNCIONALIDAD PARA ELIMINAR LA RESERVACIÓN VÍA SERVICIO WEB.	
<b>TIPO DE TAREA:</b> DESARROLLO	<b>PUNTOS ESTIMADOS:</b> 3
<b>FECHA INICIO:</b> 24/02/2009	<b>FECHA FIN:</b> 26/02/2009
<b>PROGRAMADOR RESPONSABLE:</b> ALEXANDER F. VINENT PEÑA	
<b>DESCRIPCIÓN:</b> CREAR FUNCIONALIDAD PARA ELIMINAR LA RESERVACIÓN VÍA SERVICIO WEB.	

Tabla 46 Tarea de Ingeniería: Crear funcionalidad para eliminar la reservación vía servicio web.

TAREA DE INGENIERÍA	
<b>NÚMERO DE TAREA:</b> <i>T.112</i>	<b>NÚMERO HISTORIA DE USUARIO:</b> <i>HU_11</i>
<b>NOMBRE TAREA:</b> <i>CREAR FUNCIONALIDAD PARA ELIMINAR LA RESERVACIÓN MEDIANTE MÓDULO DE DRUPAL.</i>	
<b>TIPO DE TAREA:</b> <i>DESARROLLO</i>	<b>PUNTOS ESTIMADOS:</b> <i>3</i>
<b>FECHA INICIO:</b> <i>25/03/2009</i>	<b>FECHA FIN:</b> <i>10/04/2009</i>
<b>PROGRAMADOR RESPONSABLE:</b> <i>ALEXANDER F. VINENT PEÑA</i>	
<b>DESCRIPCIÓN:</b> <i>CREAR FUNCIONALIDAD PARA ELIMINAR LA RESERVACIÓN MEDIANTE MÓDULO DE DRUPAL VÍA SERVICIO WEB.</i>	

Tabla 47 Tarea de Ingeniería: Crear funcionalidad para eliminar reservación mediante módulo de Drupal.

TAREA DE INGENIERÍA	
<b>NÚMERO DE TAREA:</b> <i>T.113</i>	<b>NÚMERO HISTORIA DE USUARIO:</b> <i>HU_11</i>
<b>NOMBRE TAREA:</b> <i>DISEÑO UI DE LA FUNCIONALIDAD ELIMINAR RESERVACIÓN.</i>	
<b>TIPO DE TAREA:</b> <i>MEJORA</i>	<b>PUNTOS ESTIMADOS:</b> <i>3</i>
<b>FECHA INICIO:</b> <i>24/03/2009</i>	<b>FECHA FIN:</b> <i>31/03/2009</i>
<b>PROGRAMADOR RESPONSABLE:</b> <i>ALEXANDER F. VINENT PEÑA</i>	
<b>DESCRIPCIÓN:</b> <i>DISEÑO UI DE LA FUNCIONALIDAD ELIMINAR RESERVACIÓN.</i>	

Tabla 48 Tarea de Ingeniería: Diseño UI de la funcionalidad Eliminar reservación.

TAREA DE INGENIERÍA	
<b>NÚMERO DE TAREA:</b> <i>T.114</i>	<b>NUMERO HISTORIA DE USUARIO:</b> <i>HU_11</i>



<b>NOMBRE TAREA:</b> <i>NOTIFICAR USUARIO SOBRE RESERVACIÓN ELIMINADA.</i>	
<b>TIPO DE TAREA:</b> <i>DESARROLLO</i>	<b>PUNTOS ESTIMADOS:</b> <i>3</i>
<b>FECHA INICIO:</b> <i>20/05/2009</i>	<b>FECHA FIN:</b> <i>20/05/2009</i>
<b>PROGRAMADOR RESPONSABLE:</b> <i>ALEXANDER F. VINENT PEÑA</i>	
<b>DESCRIPCIÓN:</b> <i>NOTIFICAR AL USUARIO INTERESADO QUE LA SOLICITUD DE RESERVACIÓN FUE ELIMINADA.</i>	

Tabla 49 Tarea de Ingeniería: Notificar usuario.

### 3.1.6. HU GESTIÓN DE LOCALES

<b>NÚMERO:</b> <i>HU.12</i>	<b>USUARIO:</b> <i>ALEXANDER F. VINENT PEÑA</i>
<b>NOMBRE HISTORIA DE USUARIO:</b> <i>GESTIÓN DE SOLICITUDES</i>	
<b>ITERACIÓN ASIGNADA:</b> <i>2DA. ITERACIÓN</i>	
<b>PRIORIDAD EN NEGOCIO:</b> <i>ALTA</i>	<b>PUNTOS ESTIMADOS:</b> <i>10</i>
<b>RIESGO EN DESARROLLO:</b> <i>MEDIA</i>	<b>PUNTOS REALES:</b> <i>80</i>
<b>DESCRIPCIÓN:</b> <i>SE LE MUESTRA AL ADMINISTRADOR UN LISTADO CON TODAS LOS LOCALES QUE SE ENCUENTRAN DISPONIBLES HASTA LA FECHA ACTUAL, BRINDÁNDOLE LA OPCIÓN DE MODIFICAR O ELIMINAR. ADEMÁS, SE LE MUESTRA UN ENLACE PARA ADICIONAR NUEVOS LOCALES.</i>	
<b>OBSERVACIONES:</b> <i>NO FORMA PARTE DE LOS REQUERIMIENTOS INICIALES.</i>	
<b>PROTOTIPO UI:</b>	

## SISTEMA DE RESERVAS DE LOCALES PARA LA ACTIVIDAD DE POSTGRADO

### GESTION DE LOCALES



NOMBRE DE LOCAL	TIPO DE LOCAL	MODIFICAR	ELIMINAR
1.Aula Gastex 1	Aula		
2.Aula Gastex 1	Aula		
3.Aula Gastex 1	Aula		
4.Aula Gastex 1	Aula		
5.Aula Gastex 1	Aula		

**ADICIONAR NUEVO LOCAL**

Tabla 50 HU Gestión de locales

La historia de usuario definida en la tabla anterior, se decidió dividir en historias más pequeñas, las cuales se muestran a continuación.

### 3.1.6.1. HU GESTIÓN DE RESERVACIONES. ADICIONAR LOCAL

<b>NÚMERO:</b> <i>HU.13</i>	<b>USUARIO:</b> <i>ALEXANDER F. VINENT PEÑA</i>		
<b>NOMBRE HISTORIA DE USUARIO:</b> <i>GESTIÓN DE LOCALES. ADICIONAR LOCAL</i>			
<b>ITERACIÓN ASIGNADA:</b> <i>2DA. ITERACIÓN</i>			
<b>PRIORIDAD EN NEGOCIO:</b> <i>ALTA</i>		<b>PUNTOS ESTIMADOS:</b> <i>10</i>	
<b>RIESGO EN DESARROLLO:</b> <i>MEDIA</i>		<b>PUNTOS REALES:</b> <i>80</i>	
<b>DESCRIPCIÓN:</b> <i>EL ADMINISTRADOR CREA UN LOCAL Y ÉSTE PASA AL LISTADO DE LOS LOCALES DISPONIBLES.</i>			
<b>OBSERVACIONES:</b> <i>NO FORMA PARTE DE LOS REQUERIMIENTOS INICIALES.</i>			
<b>PROTOTIPO UI:</b>			

**ADICIONAR LOCAL**

**TIPO DE LOCAL**

Aula

**NOMBRE DE LOCAL**

Aula 1 GASTEX

---

Tabla 51 HU Gestión de locales. Adicionar local.

<b>TAREA DE INGENIERÍA</b>	
<b>NÚMERO DE TAREA:</b> <i>T.131</i>	<b>NUMERO HISTORIA DE USUARIO:</b> <i>HU_13</i>
<b>NOMBRE TAREA:</b> <i>CREAR FUNCIONALIDAD PARA ADICIONAR LOCAL VÍA SERVICIO WEB.</i>	
<b>TIPO DE TAREA:</b> <i>DESARROLLO</i>	<b>PUNTOS ESTIMADOS:</b> <i>3</i>
<b>FECHA INICIO:</b> <i>08/05/2009</i>	<b>FECHA FIN:</b> <i>10/05/2009</i>
<b>PROGRAMADOR RESPONSABLE:</b> <i>ALEXANDER F. VINENT PEÑA</i>	
<b>DESCRIPCIÓN:</b> <i>CREAR FUNCIONALIDAD PARA ADICIONAR LOCAL VÍA SERVICIO WEB.</i>	

Tabla 52 Tarea de Ingeniería: Crear funcionalidad para registrar local mediante servicio web.

<b>TAREA DE INGENIERÍA</b>	
<b>NÚMERO DE TAREA:</b> <i>T.132</i>	<b>NUMERO HISTORIA DE USUARIO:</b> <i>HU_13</i>
<b>NOMBRE TAREA:</b> <i>CREAR FUNCIONALIDAD PARA ADICIONAR LOCAL MEDIANTE MÓDULO DE DRUPAL.</i>	

<b>TIPO DE TAREA:</b> <i>DESARROLLO</i>	<b>PUNTOS ESTIMADOS:</b> 3
<b>FECHA INICIO:</b> <i>08/05/2009</i>	<b>FECHA FIN:</b> <i>15/05/2009</i>
<b>PROGRAMADOR RESPONSABLE:</b> <i>ALEXANDER F. VINENT PEÑA</i>	
<b>DESCRIPCIÓN:</b> <i>CREAR FUNCIONALIDAD PARA ADICIONAR LOCAL MEDIANTE MÓDULO DE DRUPAL VÍA SERVICIO WEB.</i>	

Tabla 53 Tarea de Ingeniería: Crear funcionalidad para adicionar un local mediante módulo de Drupal.

TAREA DE INGENIERÍA	
<b>NÚMERO DE TAREA:</b> <i>T.133</i>	<b>NUMERO HISTORIA DE USUARIO:</b> <i>HU_13</i>
<b>NOMBRE TAREA:</b> <i>DISEÑO UI DE LA FUNCIONALIDAD ADICIONAR LOCAL.</i>	
<b>TIPO DE TAREA:</b> <i>MEJORA</i>	<b>PUNTOS ESTIMADOS:</b> 3
<b>FECHA INICIO:</b> <i>14/05/2009</i>	<b>FECHA FIN:</b> <i>14/05/2009</i>
<b>PROGRAMADOR RESPONSABLE:</b> <i>ALEXANDER F. VINENT PEÑA</i>	
<b>DESCRIPCIÓN:</b> <i>DISEÑO UI DE LA FUNCIONALIDAD ADICIONAR LOCAL.</i>	

Tabla 54 Tarea de Ingeniería: Diseño UI de la funcionalidad adicionar local.

### 3.1.6.2. HU GESTIÓN DE LOCALES. MODIFICAR LOCAL

<b>NÚMERO:</b> <i>HU.14</i>	<b>USUARIO:</b> <i>ALEXANDER F. VINENT PEÑA</i>
<b>NOMBRE HISTORIA DE USUARIO:</b> <i>GESTIÓN DE LOCALES. MODIFICAR LOCAL</i>	
<b>ITERACIÓN ASIGNADA:</b> <i>2DA. ITERACIÓN</i>	
<b>PRIORIDAD EN NEGOCIO:</b> <i>ALTA</i>	<b>PUNTOS ESTIMADOS:</b> <i>10</i>
<b>RIESGO EN DESARROLLO:</b> <i>MEDIA</i>	<b>PUNTOS REALES:</b> <i>80</i>

<b>DESCRIPCIÓN:</b> <i>EL SISTEMA MUESTRA LOS DATOS DEL LOCAL SELECCIONADO.</i>
<b>OBSERVACIONES:</b> <i>NO FORMA PARTE DE LOS REQUERIMIENTOS INICIALES.</i>
<b>PROTOTIPO UI:</b>

**MODIFICAR LOCAL**

<b>TIPO DE LOCAL</b>	<b>NOMBRE DE LOCAL</b>
<input type="text" value="Aula"/>	<input type="text" value="Aula 1 GASTEX"/>
<input type="button" value="GUARDAR DATOS"/>	

Tabla 55 HU Gestión de locales. Modificar local

<b>TAREA DE INGENIERÍA</b>	
<b>NÚMERO DE TAREA:</b> <i>T.141</i>	<b>NÚMERO HISTORIA DE USUARIO:</b> <i>HU_14</i>
<b>NOMBRE TAREA:</b> <i>CREAR FUNCIONALIDAD PARA MODIFICAR EL LOCAL VÍA SERVICIO WEB.</i>	
<b>TIPO DE TAREA:</b> <i>DESARROLLO</i>	<b>PUNTOS ESTIMADOS:</b> <i>3</i>
<b>FECHA INICIO:</b> <i>08/05/2009</i>	<b>FECHA FIN:</b> <i>10/05/2009</i>
<b>PROGRAMADOR RESPONSABLE:</b> <i>ALEXANDER F. VINENT PEÑA</i>	
<b>DESCRIPCIÓN:</b> <i>CREAR FUNCIONALIDAD PARA MODIFICAR LOCAL VÍA SERVICIO WEB.</i>	

Tabla 56 Tarea de Ingeniería: Crear funcionalidad para modificar local vía servicio web.

<b>TAREA DE INGENIERÍA</b>
----------------------------

<b>NÚMERO DE TAREA:</b> <i>T.142</i>	<b>NÚMERO HISTORIA DE USUARIO:</b> <i>HU_14</i>
<b>NOMBRE TAREA:</b> <i>CREAR FUNCIONALIDAD PARA MODIFICAR LOCAL MEDIANTE MÓDULO DE DRUPAL.</i>	
<b>TIPO DE TAREA:</b> <i>DESARROLLO</i>	<b>PUNTOS ESTIMADOS:</b> <i>3</i>
<b>FECHA INICIO:</b> <i>11/05/2009</i>	<b>FECHA FIN:</b> <i>13/05/2009</i>
<b>PROGRAMADOR RESPONSABLE:</b> <i>ALEXANDER F. VINENT PEÑA</i>	
<b>DESCRIPCIÓN:</b> <i>CREAR FUNCIONALIDAD PARA MODIFICAR EL LOCAL MEDIANTE MÓDULO DE DRUPAL VÍA SERVICIO WEB.</i>	

Tabla 57 Tarea de Ingeniería: Crear funcionalidad para Modificar local mediante módulo de Drupal.

<b>TAREA DE INGENIERÍA</b>	
<b>NÚMERO DE TAREA:</b> <i>T.143</i>	<b>NÚMERO HISTORIA DE USUARIO:</b> <i>HU_14</i>
<b>NOMBRE TAREA:</b> <i>DISEÑO UI DE LA FUNCIONALIDAD MODIFICAR LOCAL.</i>	
<b>TIPO DE TAREA:</b> <i>MEJORA</i>	<b>PUNTOS ESTIMADOS:</b> <i>3</i>
<b>FECHA INICIO:</b> <i>14/05/2009</i>	<b>FECHA FIN:</b> <i>14/05/2009</i>
<b>PROGRAMADOR RESPONSABLE:</b> <i>ALEXANDER F. VINENT PEÑA</i>	
<b>DESCRIPCIÓN:</b> <i>DISEÑO UI DE LA FUNCIONALIDAD MODIFICAR LOCAL.</i>	

Tabla 58 Tarea de Ingeniería: Diseño UI de la funcionalidad Modificar local.

<b>TAREA DE INGENIERÍA</b>	
<b>NÚMERO DE TAREA:</b> <i>T.144</i>	<b>NUMERO HISTORIA DE USUARIO:</b> <i>HU_14</i>
<b>NOMBRE TAREA:</b> <i>NOTIFICAR USUARIO SOBRE DATOS DE LOCAL MODIFICADOS.</i>	

<b>TIPO DE TAREA:</b> <i>DESARROLLO</i>	<b>PUNTOS ESTIMADOS:</b> 3
<b>FECHA INICIO:</b> <i>20/05/2009</i>	<b>FECHA FIN:</b> <i>20/05/2009</i>
<b>PROGRAMADOR RESPONSABLE:</b> <i>ALEXANDER F. VINENT PEÑA</i>	
<b>DESCRIPCIÓN:</b> <i>SI EL LOCAL HA MODIFICAR TIENE RESERVACIONES ASOCIADAS, NOTIFICAR AL USUARIO INTERESADO DE LOS CAMBIOS.</i>	

Tabla 59 Tarea de Ingeniería: Notificar usuario.

### 3.1.6.3. HU GESTIÓN DE LOCALES. ELIMINAR LOCAL

<b>NÚMERO:</b> <i>HU.15</i>	<b>USUARIO:</b> <i>ALEXANDER F. VINENT PEÑA</i>	
<b>NOMBRE HISTORIA DE USUARIO:</b> <i>GESTIÓN DE LOCALES. ELIMINAR LOCAL</i>		
<b>ITERACIÓN ASIGNADA:</b> <i>2DA. ITERACIÓN</i>		
<b>PRIORIDAD EN NEGOCIO:</b> <i>ALTA</i>	<b>PUNTOS ESTIMADOS:</b> <i>10</i>	
<b>RIESGO EN DESARROLLO:</b> <i>MEDIA</i>	<b>PUNTOS REALES:</b> <i>80</i>	
<b>DESCRIPCIÓN:</b> <i>CUANDO EL ADMINISTRADOR SELECCIONA LA OPCIÓN ELIMINAR LOCAL, EL SISTEMA MUESTRA UN MENSAJE DE CONFIRMACIÓN. SI SE DECIDE ELIMINAR EL LOCAL, ENTONCES SE DEBEN ELIMINAR EN CASCADA TODAS LAS RESERVACIONES ASOCIADAS A DICHO LOCAL. EN CASO QUE EXISTIESE UN USUARIO AFECTADO, SE NOTIFICARÁ VÍA EMAIL.</i>		
<b>OBSERVACIONES:</b> <i>CONFIRMADO CON EL CLIENTE.</i>		
<b>PROTOTIPO UI:</b>		



Tabla 60 HU Gestión de reservaciones. Eliminar reservación

TAREA DE INGENIERÍA	
<b>NÚMERO DE TAREA:</b> <i>T.151</i>	<b>NÚMERO HISTORIA DE USUARIO:</b> <i>HU_15</i>
<b>NOMBRE TAREA:</b> <i>CREAR FUNCIONALIDAD PARA ELIMINAR LOCAL VÍA SERVICIO WEB.</i>	
<b>TIPO DE TAREA:</b> <i>DESARROLLO</i>	<b>PUNTOS ESTIMADOS:</b> <i>3</i>
<b>FECHA INICIO:</b> <i>08/05/2009</i>	<b>FECHA FIN:</b> <i>10/05/2009</i>
<b>PROGRAMADOR RESPONSABLE:</b> <i>ALEXANDER F. VINENT PEÑA</i>	
<b>DESCRIPCIÓN:</b> <i>CREAR FUNCIONALIDAD PARA ELIMINAR LOCAL VÍA SERVICIO WEB.</i>	

Tabla 61 Tarea de Ingeniería: Crear funcionalidad para eliminar local vía servicio web.

TAREA DE INGENIERÍA	
<b>NÚMERO DE TAREA:</b> <i>T.152</i>	<b>NÚMERO HISTORIA DE USUARIO:</b> <i>HU_15</i>
<b>NOMBRE TAREA:</b> <i>CREAR FUNCIONALIDAD PARA ELIMINAR LOCAL MEDIANTE MÓDULO DE DRUPAL</i>	
<b>TIPO DE TAREA:</b> <i>DESARROLLO</i>	<b>PUNTOS ESTIMADOS:</b> <i>3</i>



<b>FECHA INICIO:</b> 11/05/2009	<b>FECHA FIN:</b> 13/05/2009
<b>PROGRAMADOR RESPONSABLE:</b> ALEXANDER F. VINENT PEÑA	
<b>DESCRIPCIÓN:</b> CREAR FUNCIONALIDAD PARA ELIMINAR LOCAL MEDIANTE MÓDULO DE DRUPAL VÍA SERVICIO WEB. <i>NOTA: TENER EN CUENTA QUE LAS RELACIONES DE DEPENDENCIA CON LAS RESERVACIONES.</i>	

Tabla 62 Tarea de Ingeniería: Crear funcionalidad para eliminar local mediante módulo de Drupal.

TAREA DE INGENIERÍA	
<b>NÚMERO DE TAREA:</b> T.153	<b>NÚMERO HISTORIA DE USUARIO:</b> HU_15
<b>NOMBRE TAREA:</b> DISEÑO UI DE LA FUNCIONALIDAD ELIMINAR LOCAL.	
<b>TIPO DE TAREA:</b> MEJORA	<b>PUNTOS ESTIMADOS:</b> 3
<b>FECHA INICIO:</b> 14/05/2009	<b>FECHA FIN:</b> 14/05/2009
<b>PROGRAMADOR RESPONSABLE:</b> ALEXANDER F. VINENT PEÑA	
<b>DESCRIPCIÓN:</b> DISEÑO UI DE LA FUNCIONALIDAD ELIMINAR LOCAL.	

Tabla 63 Tarea de Ingeniería: Diseño UI de la funcionalidad Eliminar local.

TAREA DE INGENIERÍA	
<b>NÚMERO DE TAREA:</b> T.154	<b>NUMERO HISTORIA DE USUARIO:</b> HU_15
<b>NOMBRE TAREA:</b> NOTIFICAR USUARIO SOBRE LOCAL ELIMINADO.	
<b>TIPO DE TAREA:</b> DESARROLLO	<b>PUNTOS ESTIMADOS:</b> 3
<b>FECHA INICIO:</b> 20/05/2009	<b>FECHA FIN:</b> 20/05/2009
<b>PROGRAMADOR RESPONSABLE:</b> ALEXANDER F. VINENT PEÑA	
<b>DESCRIPCIÓN:</b> NOTIFICAR AL USUARIO INTERESADO QUE EL LOCAL DE RESERVACIÓN FUE ELIMINADO.	

### 3.2. MODELO DE DESPLIEGUE

El modelo de despliegue contiene los nodos que conforman la topología de hardware sobre la cual se ejecuta el sistema software implementado (módulo y servicio web).



Fig. 7 Diagrama de despliegue

### 3.3. DIAGRAMA DE COMPONENTES

Los diagramas de componentes describen los elementos físicos del sistema y sus relaciones. Muestran las opciones de realización, incluyendo código fuente, binario y ejecutable. Los componentes representan todos los tipos de elementos de software que entran en la fabricación de aplicaciones informáticas. Pueden ser simples ficheros, paquetes, bibliotecas cargadas dinámicamente, etc.

A continuación, se presenta el diagrama de componentes para el sistema que se propone.

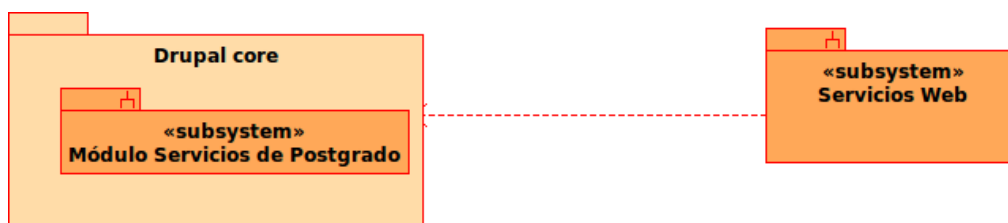


Fig. 8 Diagrama de componentes

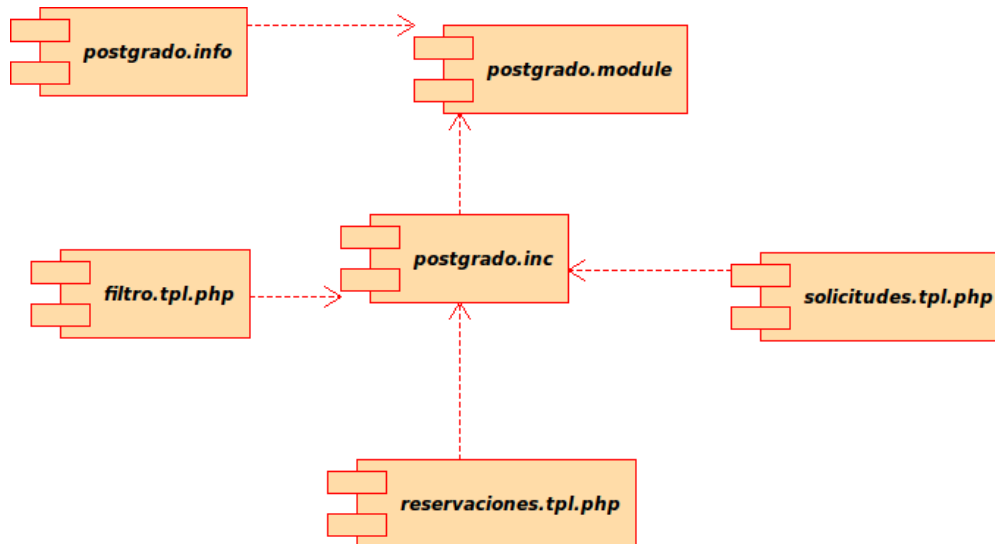


Fig. 9 Diagrama de componentes. Paquete Módulo Servicios de Postgrado

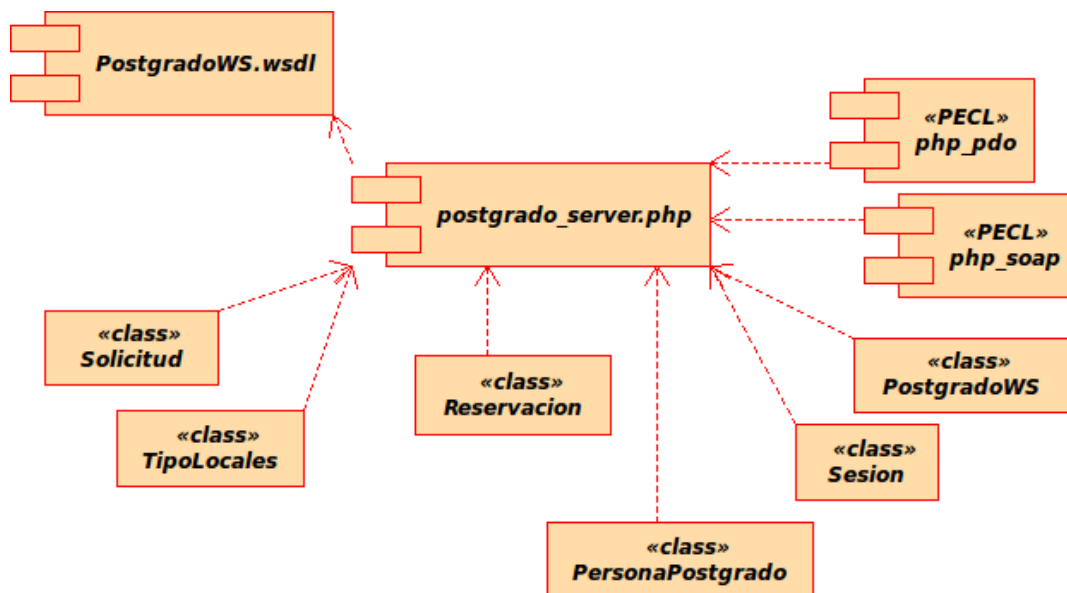


Fig. 10 Diagrama de componentes. Paquete Servicios Web

### 3.4. DESCRIPCIÓN DE PAQUETES

- **Paquete servicios web**

Se encuentran los componentes que contienen las implementaciones de las dases interfaz del servicio web, donde se exponen todas las funcionalidades que brinda el mismo, a través de la extensión *soap*.

- **Paquete Módulo Servicios de Postgrado**

En este paquete se encuentran los componentes que contienen la implementación del módulo.

## CAPÍTULO 4. VALIDACIÓN DE LA SOLUCIÓN PROPUESTA.

### 4.1. INTRODUCCION

Para lograr un producto con calidad es necesario trazarse un plan de pruebas desde el principio. Darle seguimiento a los cambios y desarrollar iterativamente. En este capítulo se plasman los casos de pruebas o test de aceptación a las que fue sometida la aplicación en cada una de las iteraciones. El cumplimiento de los anteriores fue el criterio fundamental para avanzar hacia la próxima iteración. Se expone además una relación de las funcionalidades con las que cuenta el sistema hasta la fecha.

### 4.2. CASOS DE PRUEBA

Durante el desarrollo del servicio web y el *Módulo de Servicios de Postgrado*, se diseñaron un conjunto de casos de prueba para comprobar el funcionamiento de acuerdo a las Historias de Usuarios definidas en capítulos anteriores.

Uno de los pilares de la metodología XP es el proceso de pruebas. XP anima aprobar constantemente tanto como sea posible. Esto permite aumentar la calidad de los sistemas, reduciendo el número de errores no detectados y disminuyendo el tiempo transcurrido entre la aparición de un error y su detección. Divide las pruebas del sistema en dos grupos: pruebas unitarias<sup>xii</sup>, y pruebas de aceptación o pruebas funcionales, destinadas a evaluar si al final de una iteración se consiguió la funcionalidad requerida diseñadas por el cliente final.

La producción de código está dirigida por las pruebas unitarias; las cuales son preestablecidas y son ejecutadas constantemente ante cada modificación del sistema. Como se menciona en la sección TENDENCIAS DE LAS TECNOLOGÍAS EN LAS POLÍTICAS DE INFORMATIZACIÓN, se toman en cuenta los estándares de codificación definidos para el lenguaje de programación PHP<sup>8</sup> y para la producción basada en Drupal<sup>9</sup>.

Según la propuesta establecida en la tesis *Metodología Ágil para Proyectos de Software Libre* (Peñalver Romero, 2008), el desarrollador escribe las pruebas realizadas según la historia de usuario seleccionada para realizar la comprobación y validar las funcionalidades del sistema, y de esta forma saber si está apto para ser liberado.

---

<sup>8</sup> <http://pear.php.net/manual/en/standards.php>

<sup>9</sup> <http://drupal.org/node/318>

De acuerdo a lo anterior, se genera el artefacto plantilla de Caso de Prueba de aceptación<sup>10</sup>.

#### 4.2.1. CASO DE PRUEBA INSTALACIÓN DEL MÓDULO SERVICIOS DE POSTGRADO

Esta sección cubre el conjunto de pruebas funcionales relacionadas con la instalación del *módulo Servicios de Postgrado*.

En esta historia se intenta probar que el módulo es capaz de establecer conexión con el servicio web del sistema de Formación Postgraduada.

CASO DE PRUEBA DE ACEPTACIÓN	
<b>CÓDIGO DE CASO DE PRUEBA:</b> <i>CPINSTALL1</i>	<b>NÚMERO HISTORIA DE USUARIO:</b> -
<b>DESCRIPCIÓN DE LA PRUEBA:</b> <i>INSTALACIÓN DEL COMPONENTE EN EL ENTORNO DEL CMS DRUPAL V 6.X</i> <i>SE COPIAN LOS FICHEROS DEL MÓDULO DENTRO DE LA CARPETA CORRESPONDIENTE, ESTAS PUEDEN SER:</i> <ul style="list-style-type: none"> <li>• <i>SITES/MODULES/POSTGRADO, O</i></li> <li>• <i>MODULES/POSTGRADO</i></li> </ul> <i>NOTA: DRUPAL RECOMIENDA UBICAR LOS MÓDULOS QUE NO PERTENECEN A LA INSTALACIÓN ORIGINAL EN LA PRIMERA.</i>	
<b>CONDICIONES DE EJECUCIÓN:</b> <ul style="list-style-type: none"> <li>• <i>SE NECESITA TENER INSTALADO DRUPAL V 6.X.</i></li> <li>• <i>AUTENTICADO COMO USUARIO CON PERMISOS DE ADMINISTRACIÓN DEL DRUPAL</i></li> </ul>	
<b>ENTRADA / PASOS DE EJECUCIÓN:</b> (K. VanDyk & Westgate, 2007) <ul style="list-style-type: none"> <li>• <i>PARA INSTALAR EL MÓDULO, ACCEDER A LA SECCIÓN DE MÓDULOS<sup>11</sup>.</i></li> <li>• <i>SELECCIONARLO EN LA LISTA, Y</i></li> <li>• <i>CLIC EN EL BOTÓN 'GUARDAR CONFIGURACIÓN'<sup>12</sup>.</i></li> </ul>	
<b>RESULTADO ESPERADO:</b> <i>AL EFECTUAR LA ACTIVACIÓN DEL MÓDULO, EL SISTEMA UN MENSAJE, ALERTANDO QUE LOS CAMBIOS SE REALIZARON Y SE GUARDARON. (VER EN ANEXO <u>RESULTADO DE LA INSTALACIÓN DEL COMPONENTE EN EL ENTORNO CMS DRUPAL V 6.X</u>)</i>	
<b>EVALUACIÓN DE LA PRUEBA:</b> <i>SATISFACTORIA</i>	

Tabla 65 Caso de Prueba a la instalación no.1

CASO DE PRUEBA DE ACEPTACIÓN	
<b>CÓDIGO DE CASO DE PRUEBA:</b> <i>CPINSTALL2</i>	<b>NÚMERO HISTORIA DE USUARIO:</b> -

<sup>10</sup> Se anexa la plantilla suministrada por la metodología.

<sup>11</sup> [http://sitio.midominio/mi\\_drupal/admin/build/modules](http://sitio.midominio/mi_drupal/admin/build/modules)

<sup>12</sup> El nombre de los componentes de UI pudiera variar según el idioma que se encuentre activo dentro de su instalación Drupal. Trae por defecto idioma inglés.

<b>DESCRIPCIÓN DE LA PRUEBA:</b> <i>LUEGO DE INSTALADO EL COMPONENTE SE PROCEDE ACCEDER A LA SECCIÓN DE POSTGRADO.</i>
<b>CONDICIONES DE EJECUCIÓN:</b> <i>AUTENTICADO COMO USUARIO CON PERMISOS DE ADMINISTRACIÓN DEL DRUPAL</i>
<b>ENTRADA / PASOS DE EJECUCIÓN:</b> <i>ACCEDER A LA SECCIÓN DE POSTGRADO<sup>13</sup> YA DISPONIBLE DENTRO DEL SISTEMA. EN CASO DE NO TENER ACCESO, VERIFICAR PERMISOS EN LA SECCIÓN DE GESTIÓN DE USUARIOS.</i>
<b>RESULTADO ESPERADO:</b> <i>SE MUESTRA LA INTERFAZ CORRESPONDIENTE A POSTGRADO EN LA DIRECCIÓN DEFINIDA.</i>
<b>EVALUACIÓN DE LA PRUEBA:</b> <i>SATISFACTORIA</i>

Tabla 66 Caso de Prueba a la instalación no.2

#### 4.2.2. CASO DE PRUEBA HISTORIA DE USUARIO HU.1

Esta sección cubre el conjunto de pruebas funcionales relacionadas con la historia de usuario *Solicitar reservación de locales de Postgrado.*

En esta historia se intenta probar que el módulo es capaz de establecer realizar la operación solicitada.

CASO DE PRUEBA DE ACEPTACIÓN	
<b>CÓDIGO DE CASO DE PRUEBA:</b> <i>CPHU1.11</i>	<b>NÚMERO HISTORIA DE USUARIO:</b> <i>HU1</i>
<b>DESCRIPCIÓN DE LA PRUEBA:</b> <i>SE PRUEBA LA CONEXIÓN DEL SISTEMA CON EL SERVICIO WEB DEL SISTEMA DE POSTGRADO.</i>	
<b>CONDICIONES DE EJECUCIÓN:</b> <i>SE NECESITA CONEXIÓN CON EL SERVICIO WEB.</i>	
<b>ENTRADA / PASOS DE EJECUCIÓN:</b> <i>ACCEDER A LA OPCIÓN DE SOLICITAR LOCAL, DE LA SECCIÓN DE POSTGRADO.</i>	
<b>RESULTADO ESPERADO:</b> <i>SE ESTABLECE LA CONEXIÓN Y SE COMIENZA A RECIBIR DATOS DEL MISMO.</i>	
<b>EVALUACIÓN DE LA PRUEBA:</b> <i>SATISFACTORIA</i>	

Tabla 67 Caso de Prueba a HU1

<sup>13</sup> [http://sitio.midominio/mi\\_drupal/postgrado](http://sitio.midominio/mi_drupal/postgrado)

CASO DE PRUEBA DE ACEPTACIÓN	
<b>CÓDIGO DE CASO DE PRUEBA:</b> <i>CPHU1.12</i>	<b>NÚMERO HISTORIA DE USUARIO:</b> <i>HU1</i>
<b>DESCRIPCIÓN DE LA PRUEBA:</b> <i>SE LLENAN LOS DATOS QUE PIDE EL FORMULARIO PARA SOLICITAR LOCALES. (VER TABLA 5 HU SOLICITAR RESERVACIÓN DE LOCALES DE POSTGRADO)</i>	
<b>CONDICIONES DE EJECUCIÓN:</b> <i>SE NECESITA CONEXIÓN CON EL SERVICIO WEB.</i>	
<b>ENTRADA / PASOS DE EJECUCIÓN:</b> <i>SE LLENAN LOS DATOS QUE PIDE EL FORMULARIO PARA SOLICITAR LOCALES. EN ESTA VENTANA SE MOSTRARÁ UN BOTÓN QUE PERMITA ENVIAR LA INFORMACIÓN.</i>	
<b>RESULTADO ESPERADO:</b> <i>SE MUESTRA UN MENSAJE QUE SE EFECTUÓ CORRECTAMENTE.</i>	
<b>EVALUACIÓN DE LA PRUEBA:</b> <i>SATISFACTORIA</i>	

Tabla 68 Caso de Prueba a HU1

#### 4.2.3. CASO DE PRUEBA HISTORIA DE USUARIO HU.2

Esta sección cubre el conjunto de pruebas funcionales relacionadas con la historia de usuario *Mostrar reservaciones de locales de Postgrado.*

En esta historia se intenta probar que el módulo es capaz de establecer realizar la operación solicitada.

CASO DE PRUEBA DE ACEPTACIÓN	
<b>CÓDIGO DE CASO DE PRUEBA:</b> <i>CPHU2.21</i>	<b>NÚMERO HISTORIA DE USUARIO:</b> <i>HU2</i>
<b>DESCRIPCIÓN DE LA PRUEBA:</b> <i>MOSTRAR LAS RESERVACIONES REALIZADAS.</i>	
<b>CONDICIONES DE EJECUCIÓN:</b> <i>SE NECESITA CONEXIÓN CON EL SERVICIO WEB.</i>	
<b>ENTRADA / PASOS DE EJECUCIÓN:</b> <i>ACCEDER A LA OPCIÓN DE RESERVACIONES, DE LA SECCIÓN DE POSTGRADO.</i>	
<b>RESULTADO ESPERADO:</b> <i>SE ESTABLECE LA CONEXIÓN Y SE COMIENZA A RECIBIR DATOS DEL MISMO.</i>	
<b>EVALUACIÓN DE LA PRUEBA:</b> <i>SATISFACTORIA</i>	

Tabla 69 Caso de Prueba a HU1



#### 4.2.4. CASO DE PRUEBA HISTORIA DE USUARIO HU.3

Esta sección cubre el conjunto de pruebas funcionales relacionadas con la historia de usuario *Filtrado de reservaciones de locales de Postgrado*.

En esta historia se intenta probar que el módulo es capaz de establecer realizar la operación solicitada.

CASO DE PRUEBA DE ACEPTACIÓN	
<b>CÓDIGO DE CASO DE PRUEBA:</b> <i>CPHU3.31</i>	<b>NÚMERO HISTORIA DE USUARIO:</b> <i>HU3</i>
<b>DESCRIPCIÓN DE LA PRUEBA:</b> <i>MOSTRAR LAS RESERVACIONES REALIZADAS A TRAVÉS DE FILTROS (VER Tabla 14 HU Filtrado de reservaciones)</i>	
<b>CONDICIONES DE EJECUCIÓN:</b> <i>SE NECESITA CONEXIÓN CON EL SERVICIO WEB.</i>	
<b>ENTRADA / PASOS DE EJECUCIÓN:</b> <i>SE LLENAN LOS DATOS QUE PIDE EL FORMULARIO PARA EL FILTRO SELECCIONADO. EN ESTA VENTANA SE MOSTRARÁ UN BOTÓN QUE PERMITA ENVIAR LA INFORMACIÓN.</i>	
<b>RESULTADO ESPERADO:</b> <i>SE MUESTRA UNA LISTA DE LAS RESERVACIONES REALIZADAS. EN CASO DE NO EXISTIR, SE MUESTRA UN MENSAJE.</i>	
<b>EVALUACIÓN DE LA PRUEBA:</b> <i>SATISFACTORIA</i>	

Tabla 70 Caso de Prueba a HU3

#### 4.2.5. CASO DE PRUEBA HISTORIA DE USUARIO HU.4

Esta sección cubre el conjunto de pruebas funcionales relacionadas con la historia de usuario *Gestión de Solicitudes de locales de Postgrado*.

En esta historia se intenta probar que el módulo es capaz de establecer realizar la operación solicitada.

CASO DE PRUEBA DE ACEPTACIÓN	
<b>CÓDIGO DE CASO DE PRUEBA:</b> <i>CPHU4.41</i>	<b>NÚMERO HISTORIA DE USUARIO:</b> <i>HU4</i>
<b>DESCRIPCIÓN DE LA PRUEBA:</b> <i>MOSTRAR LA UI DE GESTIÓN DE SOLICITUDES DE LOCALES. (VER Tabla 22 HU Gestión de solicitudes)</i>	
<b>CONDICIONES DE EJECUCIÓN:</b> <ul style="list-style-type: none"><li>• <i>SE NECESITA CONEXIÓN CON EL SERVICIO WEB.</i></li><li>• <i>SE NECESITA SER USUARIO DEL DOMINIO UCI CON PRIVILEGIOS DE ADMINISTRACIÓN DENTRO DEL SISTEMA DE</i></li></ul>	

<i>POSTGRADO UCI.</i>
<b>ENTRADA / PASOS DE EJECUCIÓN:</b> <i>ACCEDER A LA OPCIÓN DE ADMINISTRACIÓN/SOLICITUDES, DE LA SECCIÓN DE POSTGRADO.</i>
<b>RESULTADO ESPERADO:</b> <ul style="list-style-type: none"> <li>• <i>EN CASO DE TENER PERMISO DE NO SER USUARIO AUTORIZADO, SE MUESTRA UN MENSAJE DE ERROR.</i></li> <li>• <i>SE MUESTRA UNA LISTA DE LAS SOLICITUDES REALIZADAS. EN CASO DE NO EXISTIR, SE MUESTRA UN MENSAJE.</i></li> <li>• <i>POR SOLICITUD, SE MUESTRA LA OPCIÓN DE ‘PROCESAR’, ‘MODIFICAR’ Y ‘ELIMINAR’.</i></li> </ul>
<b>EVALUACIÓN DE LA PRUEBA:</b> <i>SATISFACTORIA</i>

Tabla 71 Caso de Prueba a HU4

#### 4.2.6. CASO DE PRUEBA HISTORIA DE USUARIO HU.5

Esta sección cubre el conjunto de pruebas funcionales relacionadas con la historia de usuario *GESTIÓN DE SOLICITUDES. MODIFICAR SOLICITUD.*

En esta historia se intenta probar que el módulo es capaz de establecer realizar la operación solicitada.

<b>CASO DE PRUEBA DE ACEPTACIÓN</b>	
<b>CÓDIGO DE CASO DE PRUEBA:</b> <i>CPHU5.51</i>	<b>NÚMERO HISTORIA DE USUARIO:</b> <i>HU5</i>
<b>DESCRIPCIÓN DE LA PRUEBA:</b> <i>SE MOSTRARÁN LOS DATOS DE LA SOLICITUD SELECCIONADA. (VER Tabla 23 HU Gestión de solicitudes. Modificar solicitud)</i>	
<b>CONDICIONES DE EJECUCIÓN:</b> <i>SE NECESITA CONEXIÓN CON EL SERVICIO WEB.</i>	
<b>ENTRADA / PASOS DE EJECUCIÓN:</b> <i>SELECCIONAR LA OPCIÓN CORRESPONDIENTE A MODIFICAR LA SOLICITUD.</i>	
<b>RESULTADO ESPERADO:</b> <i>SE DEBEN MOSTRAR LOS DATOS ACTUALES DE LA SOLICITUD, Y CAMPOS PARA MODIFICAR SU CONTENIDO.</i>	
<b>EVALUACIÓN DE LA PRUEBA:</b> <i>SATISFACTORIA</i>	

Tabla 72 Caso de Prueba a HU5

<b>CASO DE PRUEBA DE ACEPTACIÓN</b>
-------------------------------------

<b>CÓDIGO DE CASO DE PRUEBA:</b> <i>CPHU5.52</i>	<b>NÚMERO HISTORIA DE USUARIO:</b> <i>HU5</i>
<b>DESCRIPCIÓN DE LA PRUEBA:</b> <i>VALIDACIÓN Y MODIFICACIÓN DE LOS DATOS DE LA SOLICITUD. (VER Tabla 23 HU Gestión de solicitudes. Modificar solicitud)</i>	
<b>CONDICIONES DE EJECUCIÓN:</b> <i>SE NECESITA CONEXIÓN CON EL SERVICIO WEB.</i>	
<b>ENTRADA / PASOS DE EJECUCIÓN:</b> <i>SELECCIONAR LA OPCIÓN CORRESPONDIENTE A MODIFICAR LA SOLICITUD.</i>	
<b>RESULTADO ESPERADO:</b> <i>EN CASO DE OMISION DE ALGUNO DE LOS DATOS QUE SE SOLICITAN EN EL FORMULARIO, SE MUESTRA UN MENSAJE DE ERROR.</i>	
<b>EVALUACIÓN DE LA PRUEBA:</b> <i>SATISFACTORIA</i>	

Tabla 73 Caso de Prueba a HU5

#### 4.2.7. CASO DE PRUEBA HISTORIA DE USUARIO HU.6

Esta sección cubre el conjunto de pruebas funcionales relacionadas con la historia de usuario *GESTIÓN DE SOLICITUDES. ELIMINAR SOLICITUD.*

En esta historia se intenta probar que el módulo es capaz de establecer realizar la operación solicitada.

CASO DE PRUEBA DE ACEPTACIÓN	
<b>CÓDIGO DE CASO DE PRUEBA:</b> <i>CPHU6.61</i>	<b>NÚMERO HISTORIA DE USUARIO:</b> <i>HU6</i>
<b>DESCRIPCIÓN DE LA PRUEBA:</b> <i>SE ELIMINARÁ LA SOLICITUD SELECCIONADA.</i>	
<b>CONDICIONES DE EJECUCIÓN:</b> <i>SE NECESITA CONEXIÓN CON EL SERVICIO WEB.</i>	
<b>ENTRADA / PASOS DE EJECUCIÓN:</b> <i>SELECCIONAR LA OPCIÓN CORRESPONDIENTE A ELIMINAR LA SOLICITUD.</i>	
<b>RESULTADO ESPERADO:</b> <i>AL ELIMINAR LA SOLICITUD, SE DEBE ACTUALIZAR EL LISTADO DE SOLICITUDES REALIZADAS.</i>	
<b>EVALUACIÓN DE LA PRUEBA:</b> <i>SATISFACTORIA</i>	

Tabla 74 Caso de Prueba a HU6

#### 4.2.8. CASO DE PRUEBA HISTORIA DE USUARIO HU.7

Esta sección cubre el conjunto de pruebas funcionales relacionadas con la historia de usuario *Gestión de Reservaciones de locales de Postgrado*.

En esta historia se intenta probar que el módulo es capaz de establecer realizar la operación solicitada.

<b>CASO DE PRUEBA DE ACEPTACIÓN</b>	
<b>CÓDIGO DE CASO DE PRUEBA:</b> <i>CPHU7.71</i>	<b>NÚMERO HISTORIA DE USUARIO:</b> <i>HU7</i>
<b>DESCRIPCIÓN DE LA PRUEBA:</b> <i>MOSTRAR LA UI DE GESTIÓN DE RESERVACIONES DE LOCALES.</i>	
<b>CONDICIONES DE EJECUCIÓN:</b> <ul style="list-style-type: none"> <li>• <i>SE NECESITA CONEXIÓN CON EL SERVICIO WEB.</i></li> <li>• <i>SE NECESITA SER USUARIO DEL DOMINIO UCI CON PRIVILEGIOS DE ADMINISTRACIÓN DENTRO DEL SISTEMA DE POSTGRADO UCI.</i></li> </ul>	
<b>ENTRADA / PASOS DE EJECUCIÓN:</b> <i>ACCEDER A LA OPCIÓN DE ADMINISTRACIÓN/RESERVACIONES, DE LA SECCIÓN DE POSTGRADO.</i>	
<b>RESULTADO ESPERADO:</b> <ul style="list-style-type: none"> <li>• <i>EN CASO DE TENER PERMISO DE NO SER USUARIO AUTORIZADO, SE MUESTRA UN MENSAJE DE ERROR.</i></li> <li>• <i>SE MUESTRA UNA LISTA DE LAS RESERVACIONES REALIZADAS. EN CASO DE NO EXISTIR, SE MUESTRA UN MENSAJE.</i></li> <li>• <i>POR RESERVACIÓN, SE MUESTRA LA OPCIÓN DE ‘VER DETALLES’, ‘MODIFICAR’ Y ‘ELIMINAR’.</i></li> </ul>	
<b>EVALUACIÓN DE LA PRUEBA:</b> <i>SATISFACTORIA</i>	

Tabla 75 Caso de Prueba a HU7

#### 4.2.9. CASO DE PRUEBA HISTORIA DE USUARIO HU.8

Esta sección cubre el conjunto de pruebas funcionales relacionadas con la historia de usuario *GESTIÓN DE RESERVACIONES. DETALLES DE RESERVACIÓN*.

En esta historia se intenta probar que el módulo es capaz de establecer realizar la operación solicitada.

<b>CASO DE PRUEBA DE ACEPTACIÓN</b>	
<b>CÓDIGO DE CASO DE PRUEBA:</b> <i>CPHU8.81</i>	<b>NÚMERO HISTORIA DE USUARIO:</b> <i>HU8</i>
<b>DESCRIPCIÓN DE LA PRUEBA:</b> <i>SE MOSTRARÁN LOS DATOS DE LA RESERVACIÓN SELECCIONADA.</i>	

<b>CONDICIONES DE EJECUCIÓN:</b> <i>SE NECESITA CONEXIÓN CON EL SERVICIO WEB.</i>
<b>ENTRADA / PASOS DE EJECUCIÓN:</b> <i>SELECCIONAR LA OPCIÓN CORRESPONDIENTE A MODIFICAR LA SOLICITUD.</i>
<b>RESULTADO ESPERADO:</b> <i>SE DEBEN MOSTRAR LOS DATOS DE LA RESERVACIÓN.</i>
<b>EVALUACIÓN DE LA PRUEBA:</b> <i>SATISFACTORIA</i>

Tabla 76 Caso de Prueba a HU8

#### 4.2.10. CASO DE PRUEBA HISTORIA DE USUARIO HU.9

Esta sección cubre el conjunto de pruebas funcionales relacionadas con la historia de usuario *GESTIÓN DE RESERVACIONES. ADICIONAR RESERVACIÓN.*

En esta historia se intenta probar que el módulo es capaz de establecer realizar la operación solicitada.

<b>CASO DE PRUEBA DE ACEPTACIÓN</b>	
<b>CÓDIGO DE CASO DE PRUEBA:</b> <i>CPHU9.91</i>	<b>NÚMERO HISTORIA DE USUARIO:</b> <i>HU9</i>
<b>DESCRIPCIÓN DE LA PRUEBA:</b> <i>SE SELECCIONA UNA SOLICITUD, SE MUESTRAN SUS DATOS Y SE ASIGNA UN LOCAL.</i>	
<b>CONDICIONES DE EJECUCIÓN:</b> <i>SE NECESITA CONEXIÓN CON EL SERVICIO WEB.</i>	
<b>ENTRADA / PASOS DE EJECUCIÓN:</b> <i>SELECCIONAR LA OPCIÓN CORRESPONDIENTE A PROCESAR LA SOLICITUD.</i>	
<b>RESULTADO ESPERADO:</b> <i>SE DEBE ACTUALIZAR LAS SOLICITUDES Y LAS RESERVACIONES.</i>	
<b>EVALUACIÓN DE LA PRUEBA:</b> <i>SATISFACTORIA</i>	

Tabla 77 Caso de Prueba a HU9

#### 4.2.11. CASO DE PRUEBA HISTORIA DE USUARIO HU.10

Esta sección cubre el conjunto de pruebas funcionales relacionadas con la historia de usuario *GESTIÓN DE RESERVACIONES. MODIFICAR RESERVACIÓN.*

En esta historia se intenta probar que el módulo es capaz de establecer realizar la operación solicitada.

CASO DE PRUEBA DE ACEPTACIÓN	
<b>CÓDIGO DE CASO DE PRUEBA:</b> <i>CPHU10.101</i>	<b>NÚMERO HISTORIA DE USUARIO:</b> <i>HU10</i>
<b>DESCRIPCIÓN DE LA PRUEBA:</b> <i>SE MOSTRARÁN LOS DATOS DE LA RESERVACIÓN SELECCIONADA.</i>	
<b>CONDICIONES DE EJECUCIÓN:</b> <i>SE NECESITA CONEXIÓN CON EL SERVICIO WEB.</i>	
<b>ENTRADA / PASOS DE EJECUCIÓN:</b> <i>SELECCIONAR LA OPCIÓN CORRESPONDIENTE A MODIFICAR LA RESERVACIÓN.</i>	
<b>RESULTADO ESPERADO:</b> <i>SE DEBEN MOSTRAR LOS DATOS ACTUALES DE LA RESERVACIÓN, Y CAMPOS PARA MODIFICAR SU CONTENIDO.</i>	
<b>EVALUACIÓN DE LA PRUEBA:</b> <i>SATISFACTORIA</i>	

Tabla 78 Caso de Prueba a HU10

CASO DE PRUEBA DE ACEPTACIÓN	
<b>CÓDIGO DE CASO DE PRUEBA:</b> <i>CPHU10.102</i>	<b>NÚMERO HISTORIA DE USUARIO:</b> <i>HU10</i>
<b>DESCRIPCIÓN DE LA PRUEBA:</b> <i>VALIDACIÓN Y MODIFICACIÓN DE LOS DATOS DE LA RESERVACIÓN.</i>	
<b>CONDICIONES DE EJECUCIÓN:</b> <i>SE NECESITA CONEXIÓN CON EL SERVICIO WEB.</i>	
<b>ENTRADA / PASOS DE EJECUCIÓN:</b> <i>SELECCIONAR LA OPCIÓN CORRESPONDIENTE A MODIFICAR LA RESERVACIÓN.</i>	
<b>RESULTADO ESPERADO:</b> <i>EN CASO DE OMISION DE ALGUNO DE LOS DATOS QUE SE SOLICITAN EN EL FORMULARIO, SE MUESTRA UN MENSAJE DE ERROR.</i>	
<b>EVALUACIÓN DE LA PRUEBA:</b> <i>SATISFACTORIA</i>	

Tabla 79 Caso de Prueba a HU10

#### 4.2.12. CASO DE PRUEBA HISTORIA DE USUARIO HU.11

Esta sección cubre el conjunto de pruebas funcionales relacionadas con la historia de usuario *GESTIÓN DE RESERVACIONES. ELIMINAR RESERVACIÓN.*

En esta historia se intenta probar que el módulo es capaz de establecer realizar la operación solicitada.

CASO DE PRUEBA DE ACEPTACIÓN	
<b>CÓDIGO DE CASO DE PRUEBA:</b> <i>CPHU11.111</i>	<b>NÚMERO HISTORIA DE USUARIO:</b> <i>HU11</i>
<b>DESCRIPCIÓN DE LA PRUEBA:</b> <i>SE ELIMINARÁ LA RESERVACIÓN SELECCIONADA.</i>	
<b>CONDICIONES DE EJECUCIÓN:</b> <i>SE NECESITA CONEXIÓN CON EL SERVICIO WEB.</i>	
<b>ENTRADA / PASOS DE EJECUCIÓN:</b> <i>SELECCIONAR LA OPCIÓN CORRESPONDIENTE A ELIMINAR LA RESERVACIÓN.</i>	
<b>RESULTADO ESPERADO:</b> <i>AL ELIMINAR LA RESERVACIÓN, SE DEBE ACTUALIZAR EL LISTADO DE RESERVACIONES REALIZADAS.</i>	
<b>EVALUACIÓN DE LA PRUEBA:</b> <i>SATISFACTORIA</i>	

Tabla 80 Caso de Prueba a HU11

#### 4.2.13. CASO DE PRUEBA HISTORIA DE USUARIO HU.12

Esta sección cubre el conjunto de pruebas funcionales relacionadas con la historia de usuario *Gestión de locales de Postgrado*.

En esta historia se intenta probar que el módulo es capaz de establecer realizar la operación solicitada.

CASO DE PRUEBA DE ACEPTACIÓN	
<b>CÓDIGO DE CASO DE PRUEBA:</b> <i>CPHU12.121</i>	<b>NÚMERO HISTORIA DE USUARIO:</b> <i>HU12</i>
<b>DESCRIPCIÓN DE LA PRUEBA:</b> <i>MOSTRAR LA UI DE GESTIÓN DE LOCALES.</i>	
<b>CONDICIONES DE EJECUCIÓN:</b> <ul style="list-style-type: none"> <li>● <i>SE NECESITA CONEXIÓN CON EL SERVICIO WEB.</i></li> <li>● <i>SE NECESITA SER USUARIO DEL DOMINIO UCI CON PRIVILEGIOS DE ADMINISTRACIÓN DENTRO DEL SISTEMA DE POSTGRADO UCI.</i></li> </ul>	
<b>ENTRADA / PASOS DE EJECUCIÓN:</b> <i>ACCEDER A LA OPCIÓN DE ADMINISTRACIÓN/NOMENCLADORES, DE LA SECCIÓN DE POSTGRADO.</i>	
<b>RESULTADO ESPERADO:</b> <ul style="list-style-type: none"> <li>● <i>EN CASO DE TENER PERMISO DE NO SER USUARIO AUTORIZADO, SE MUESTRA UN MENSAJE DE ERROR.</i></li> <li>● <i>SE MUESTRA UNA LISTA DE LOS LOCALES CON QUE CUENTA LA DIRECCIÓN DE FORMACIÓN POSTGRADUADA. EN CASO DE NO EXISTIR, SE MUESTRA UN MENSAJE.</i></li> <li>● <i>POR LOCAL, SE MUESTRA LA OPCIÓN DE 'MODIFICAR' Y 'ELIMINAR'.</i></li> </ul>	

**EVALUACIÓN DE LA PRUEBA:**  
*SATISFACTORIA*

Tabla 81 Caso de Prueba a HU12

#### 4.2.14. CASO DE PRUEBA HISTORIA DE USUARIO HU.13

Esta sección cubre el conjunto de pruebas funcionales relacionadas con la historia de usuario *GESTIÓN DE LOCALES. ADICIONAR LOCAL.*

En esta historia se intenta probar que el módulo es capaz de establecer realizar la operación solicitada.

CASO DE PRUEBA DE ACEPTACIÓN	
<b>CÓDIGO DE CASO DE PRUEBA:</b> <i>CPHU13.131</i>	<b>NÚMERO HISTORIA DE USUARIO:</b> <i>HU13</i>
<b>DESCRIPCIÓN DE LA PRUEBA:</b> <i>SE ADICIONARÁ UN NUEVO LOCAL.</i>	
<b>CONDICIONES DE EJECUCIÓN:</b> <i>SE NECESITA CONEXIÓN CON EL SERVICIO WEB.</i>	
<b>ENTRADA / PASOS DE EJECUCIÓN:</b> <i>SE PROPORCIONA EL NOMBRE DEL LOCAL Y EL TIPO AL QUE PERTENECE (AULA/LABORATORIO/SALÓN).</i>	
<b>RESULTADO ESPERADO:</b> <i>SE DEBE ACTUALIZAR LA LISTA DE LOCALES.</i>	
<b>EVALUACIÓN DE LA PRUEBA:</b> <i>SATISFACTORIA</i>	

Tabla 82 Caso de Prueba a HU13

#### 4.2.15. CASO DE PRUEBA HISTORIA DE USUARIO HU.14

Esta sección cubre el conjunto de pruebas funcionales relacionadas con la historia de usuario *GESTIÓN DE LOCALES. MODIFICAR LOCAL.*

En esta historia se intenta probar que el módulo es capaz de establecer realizar la operación solicitada.

CASO DE PRUEBA DE ACEPTACIÓN	
<b>CÓDIGO DE CASO DE PRUEBA:</b> <i>CPHU14.141</i>	<b>NÚMERO HISTORIA DE USUARIO:</b> <i>HU14</i>



<b>DESCRIPCIÓN DE LA PRUEBA:</b> <i>SE MOSTRARÁN LOS DATOS DEL LOCAL SELECCIONADO.</i>
<b>CONDICIONES DE EJECUCIÓN:</b> <i>SE NECESITA CONEXIÓN CON EL SERVICIO WEB.</i>
<b>ENTRADA / PASOS DE EJECUCIÓN:</b> <i>SELECCIONAR LA OPCIÓN CORRESPONDIENTE A MODIFICAR LOCAL.</i>
<b>RESULTADO ESPERADO:</b> <i>SE DEBEN MOSTRAR LOS DATOS ACTUALES DEL LOCAL, Y CAMPOS PARA MODIFICAR SU CONTENIDO.</i>
<b>EVALUACIÓN DE LA PRUEBA:</b> <i>SATISFACTORIA</i>

Tabla 83 Caso de Prueba a HU14

<b>CASO DE PRUEBA DE ACEPTACIÓN</b>	
<b>CÓDIGO DE CASO DE PRUEBA:</b> <i>CPHU14.142</i>	<b>NÚMERO HISTORIA DE USUARIO:</b> <i>HU14</i>
<b>DESCRIPCIÓN DE LA PRUEBA:</b> <i>VALIDACIÓN Y MODIFICACIÓN DE LOS DATOS DEL LOCAL.</i>	
<b>CONDICIONES DE EJECUCIÓN:</b> <i>SE NECESITA CONEXIÓN CON EL SERVICIO WEB.</i>	
<b>ENTRADA / PASOS DE EJECUCIÓN:</b> <i>SELECCIONAR LA OPCIÓN CORRESPONDIENTE A MODIFICAR LOCAL.</i>	
<b>RESULTADO ESPERADO:</b> <i>EN CASO DE OMISION DE ALGUNO DE LOS DATOS QUE SE SOLICITAN EN EL FORMULARIO, SE MUESTRA UN MENSAJE DE ERROR.</i>	
<b>EVALUACIÓN DE LA PRUEBA:</b> <i>SATISFACTORIA</i>	

Tabla 84 Caso de Prueba a HU14

#### 4.2.16. CASO DE PRUEBA HISTORIA DE USUARIO HU.15

Esta sección cubre el conjunto de pruebas funcionales relacionadas con la historia de usuario *GESTIÓN DE LOCALES. ELIMINAR LOCAL.*

En esta historia se intenta probar que el módulo es capaz de establecer realizar la operación solicitada.

<b>CASO DE PRUEBA DE ACEPTACIÓN</b>
-------------------------------------

<b>CÓDIGO DE CASO DE PRUEBA:</b> <i>CPHU15.151</i>	<b>NÚMERO HISTORIA DE USUARIO:</b> <i>HU15</i>
<b>DESCRIPCIÓN DE LA PRUEBA:</b> <i>SE ELIMINARÁ EL LOCAL SELECCIONADO.</i>	
<b>CONDICIONES DE EJECUCIÓN:</b> <i>SE NECESITA CONEXIÓN CON EL SERVICIO WEB.</i>	
<b>ENTRADA / PASOS DE EJECUCIÓN:</b> <i>SELECCIONAR LA OPCIÓN CORRESPONDIENTE A ELIMINAR EL LOCAL.</i>	
<b>RESULTADO ESPERADO:</b> <i>AL ELIMINAR EL LOCAL, SE DEBEN ELIMINAR LAS RESERVACIONES ASOCIADAS A DICHO LOCAL (SI LAS HUBIESE), Y ACTUALIZAR EL LISTADO DE LOCALES.</i>	
<b>EVALUACIÓN DE LA PRUEBA:</b> <i>SATISFACTORIA</i>	

Tabla 85 Caso de Prueba a HU15

#### 4.3. RESULTADOS OBTENIDOS

El desarrollo de una nueva intranet universitaria acorde a la situación actual suponía un trabajo colosal pues se requería que en un período relativamente corto se lograra la migración de los sistemas existentes a una nueva plataforma donde todos los servicios estuviesen integrados. En base a lo anterior, se logró crear un módulo que permite integrar tanto la parte pública como la gestión del sistema de la Dirección de Postgrado, dentro del nuevo sistema Intranet 2.0.

#### 4.4. CONCLUSIONES

Se presentaron algunos casos de pruebas que guiaron la calidad del sistema, y determinaron en cada momento si se estaba o no en condiciones de continuar avanzando.

## **CONCLUSIONES GENERALES**

La implantación de nuevos conceptos para el desarrollo de las aplicaciones a brindar dentro de la Intranet UCI supone cambios importantes en el modo de interactuar con los sistemas informáticos. Durante la investigación se demostró la necesidad de desarrollar el componente para integrar el área de Postgrado dentro de la Intranet 2.0. Se ha realizado un estudio de las técnicas de desarrollo de componentes basados en el CMS Drupal, se ha analizado su funcionamiento y características fundamentales. Se implementó y probó un nuevo módulo que permite acceder a las funcionalidades definidas por el servicio web del sistema de la Dirección de Postgrado UCI, lo cual satisface las necesidades de los clientes. Por todo lo anteriormente expuesto se concluye que los objetivos trazados para el presente trabajo se han cumplido satisfactoriamente.

## RECOMENDACIONES

- Que el presente trabajo se siga desarrollando para implementar nuevas funcionalidades y mejorar su calidad y robustez.
- Que se tracen nuevas metas en el desarrollo del producto con el objetivo de mejorar su rendimiento.
- Que el producto sea probado en el sistema *Intranet 2.0*, a fin de validar su funcionamiento.

*Se tengan en cuentas nuevas funcionalidades como:*

- *Solicitudes*

Que el usuario antes de realizar solicitud pueda ver el listado de las solicitudes que ha realizado y que aún no tienen respuesta de reservación.

- *Reservaciones*

Realizar Notificación de Reservación por correo, para ello la Dirección de Postgrado debe tener una cuenta de correo activa.

## BIBLIOGRAFÍA CONSULTADA

- Alliance, S. (s.f.). *Scrum Alliance*. Obtenido de <http://www.scrumalliance.org>
- Booch, G., Rumbaugh, J., & Jacobson. (1999). *El Lenguaje Unificado de Modelado. Manual de usuario*. Addison-Wesley ed. B.J. Rumbaugh.
- C. Camus, J. (2006, Diciembre 16). Web 2.0: el ruido, las nueces y el futuro del web. Disponible en usando.info: [http://www.usando.info/main\\_file.php/us\\_gc/8513/](http://www.usando.info/main_file.php/us_gc/8513/)
- Canós, J. H., Letelier, P., & Penadé, M. C. *Métodologías Ágiles en el Desarrollo de Software*. Valencia. España: DSIC -Universidad Politécnica de Valencia.
- Dirección de Informatización UCI. Fundamentación de la necesidad de crear la Intranet 2.0.
- Documentación API de Drupal 6.x. (n.d.). Retrieved from <http://api.drupal.org/>
- Drupal.org. (n.d.). Contribuciones a Drupal: repositorio CSV. Retrieved from <http://cvs.drupal.org/viewcvs/contributions>
- Fowler, M., & Scott, K. (1997). *UML gota a gota*. Prentice Hall.
- Hermann, U. (2008). Gotas en la red. Retrieved from <http://www.linux-magazine.es>
- K. VanDyk, J., & Westgate, M. (2007). Capítulo 3 Module-specific settings. In J. K. VanDyk, & M. Westgate, *Pro Drupal Development* (p. 456). Apress.
- Orallo Hernández, E. (s.f.). *El Lenguaje Unificado de Modelado (UML)*. Obtenido de <http://www.disca.upv.es/enheror/pdf/ActaUML.PDF>
- Peñalver Romero, G. M. (2008). *Metodología ágil para Proyectos de Software Libre*. Ciudad de La Habana: UCI.
- PHP.net. (2009). PDO manual. Obtenido de <http://www.php.net/pdo>
- Piedra, N., & Chicaiza, J. Generación de Recursos y Prácticas Educativas abiertas usando Web 2.0 una vía para Desarrollar competencias en la Sociedad del Conocimiento. In O. S. University/FLACSO/UTPL (Ed.). Disponible en <http://www.scribd.com/doc/3218556/Generacion-de-Recursos-y-Practicas-Educativas-Abiertas-usando-Web-20> .
- Rodríguez Corbea, M., & Ordóñez Pérez, M. (2007). *La metodología XP aplicable al desarrollo del software educativo en Cuba*. Ciudad de La Habana: Universidad de las Ciencias Informáticas. Facultad 9.
- Rodríguez Villar, M. (2007). *Introducción de procedimientos ágiles en la producción de software en la facultad 7 de la UCI*. Ciudad de La Habana: UCI.

- SALAZAR GUTIERREZ, Y. (s.f.). *UCIPEDIA - XP*. Obtenido de <http://ucipedia.uci.cu/index.php/XP>
- Saumell Reyes, M., & Rodríguez Lazo, M. J. (2007). *Módulo Control de Acceso del proyecto Intranet del Centro Rector de Universidad para Todos*. Ciudad de La Habana.
- SOA Agenda. (2009, Agosto). Arquitectura SOA. Retrieved Abril 2009, from <http://soaagenda.com>: <http://soaagenda.com/journal/articulos/arquitectura-de-referencia-soa/>
- UCIPEDIA. (2007). Visual Paradigm. Obtenido de [http://ucipedia.uci.cu/index.php/Visual\\_Paradigm](http://ucipedia.uci.cu/index.php/Visual_Paradigm)
- Van de Putte, G., & otros, y. (2004). *Using Web Services for Bussiness integration*. IBM.
- Visual Paradigm. (s.f.). Visual Paradigm for UML. Obtenido de <http://www.visual-paradigm.com/product/vpuml/vpumldocuments.jsp>
- W3C Consortium. (n.d.). *Extensible Markup Language 1.0 (Fourth Edition)*. Retrieved febrero 25, 2009, from <http://www.w3c.org/TR/2006/REC-xml-20060816> [online]
- Wikipedia. (n.d.). Web 2.0. Retrieved Abril 10, 2009, from [http://es.wikipedia.org/wiki/Web\\_2.0](http://es.wikipedia.org/wiki/Web_2.0)
- Wikipedia. (s.f.). *Aplicación\_web*. Obtenido de [http://es.wikipedia.org/wiki/Aplicación\\_web](http://es.wikipedia.org/wiki/Aplicación_web)
- World Wide Web Consortium. (n.d.). *Web Services Architecture [online]*. Retrieved febrero 25, 2009, from <http://www.w3.org/TR/2004/NOTE-ws-arch-20040211/#whatis>
- WSDL. (2007). *WSDL para la documentación de Servicios Web*. Retrieved from <http://www.desarrolloweb.com/articulos/1581.php> [online]

## ANEXOS

### ANEXO I. PLANTILLA DESCRIPCION DE LAS HISTORIAS DE USUARIO

(Rodríguez Villar, 2007)

<b>NÚMERO:</b> <i>[NÚMERO DE LA HISTORIA]</i>	<b>USUARIO:</b> <i>[ROL DE USUARIO QUE REALIZA LA ACCIÓN]</i>
<b>NOMBRE HISTORIA DE USUARIO:</b> <i>[NOMBRE QUE IDENTIFICA LA HISTORIA]</i>	
<b>USUARIO:</b> <i>[PROGRAMADOR RESPONSABLE DE SU IMPLEMENTACIÓN]</i>	<b>ITERACIÓN ASIGNADA:</b> <i>[QUE ITERACIÓN SE DESARROLLARÁ (SEGÚN SU IMPORTANCIA)]</i>
<b>PRIORIDAD EN NEGOCIO:</b> <i>[PRIORIDAD PUEDE SER ALTA, MEDIA O BAJA](SEGÚN CLIENTE)</i>	<b>PUNTOS ESTIMADOS:</b> <i>[TIEMPO EN SEMANAS (ESTIMADO)]</i>
<b>RIESGO EN DESARROLLO:</b> <i>[RIESGO PUEDE SER ALTO, MEDIO O BAJO (SEGÚN PROGRAMADORES)]</i>	<b>PUNTOS REALES:</b> <i>[TIEMPO REAL DEDICADO A LA REALIZACIÓN DE LA HU EN SEMANAS]</i>
<b>DESCRIPCIÓN:</b> <i>[BREVE DESCRIPCIÓN DEL PROCESO QUE DEFINE LA HISTORIA]</i>	
<b>OBSERVACIONES:</b> <i>[ALGUNA ACOTACIÓN IMPORTANTE A SEÑALAR ACERCA DE LA HU]</i>	
<b>PROTOTIPO UI:</b> <i>[IMAGEN DE CADA UNA DE LAS INTERFACES RELACIONADAS CON LA HU]</i>	

### ANEXO II. PLANTILLA DESCRIPCION DE TAREA DE INGENIERÍA

(Rodríguez Villar, 2007)

TAREA DE INGENIERÍA	
<b>NÚMERO DE TAREA:</b> <i>[NÚMERO QUE IDENTIFICA LA TAREA]</i>	<b>NUMERO HISTORIA DE USUARIO:</b> <i>[HU A LA QUE PERTENECE]</i>
<b>NOMBRE TAREA:</b> <i>[NOMBRE DE LA TAREA]</i>	
<b>TIPO DE TAREA:</b> <i>[DESARROLLO / CORRECCIÓN / MEJORA / OTRA (ESPECIFICAR)]</i>	<b>PUNTOS ESTIMADOS:</b> <i>[TIEMPO EN SEMANAS (ESTIMADO)]</i>
<b>FECHA INICIO:</b> <i>[DD/MM/AAAA]</i>	<b>FECHA FIN:</b> <i>[DD/MM/AAAA]</i>
<b>PROGRAMADOR RESPONSABLE:</b> <i>[RESPONSABLE DE IMPLEMENTACIÓN]</i>	

**DESCRIPCIÓN:**

[DESCRIPCIÓN BREVE DE LA TAREA. P.E: CREA EL SOCKET PARA PODER ESTABLECER LA CONEXIÓN POR TCP/IP CON EL SERVIDOR Y INICIALIZA TODAS LAS VARIABLES DEL ENTORNO.]

**ANEXO III PLANTILLA CASO DE PRUEBA DE ACEPTACIÓN**

(Rodríguez Villar, 2007)

CASO DE PRUEBA DE ACEPTACIÓN	
<b>CÓDIGO DE CASO DE PRUEBA:</b> [CÓDIGO QUE IDENTIFICA LA PRUEBA]	<b>NÚMERO HISTORIA DE USUARIO:</b> [HU A LA QUE PERTENECE]
<b>DESCRIPCIÓN DE LA PRUEBA:</b> [DESCRIPCIÓN BREVE DE LA PRUEBA.]	
<b>CONDICIONES DE EJECUCIÓN:</b> [CONDICIONES IMPRESCINDIBLES PARA EFECTUAR LA PRUEBA.]	
<b>ENTRADA / PASOS DE EJECUCIÓN:</b> [BREVE DESCRIPCIÓN DE LOS PASOS PARA EFECTUAR LA VERIFICACIÓN DEL CASO DE PRUEBA.]	
<b>RESULTADO ESPERADO:</b> [PONER EL RESULTADO DE LA PRUEBA. SE PUEDEN INCLUIR OBSERVACIONES.]	
<b>EVALUACIÓN DE LA PRUEBA:</b> [SATISFACTORIA /NO SATISFACTORIA]	

**ANEXO IV RESULTADOS DE PRUEBAS DE ACEPTACIÓN**

The screenshot shows the Drupal 6.x administration interface for 'localhost'. The breadcrumb trail is 'Home > Administer > Site building'. The 'Modules' page is active, with a 'List' button and an 'Uninstall' link. A green message box at the top states: 'The configuration options have been saved.' Below this, there is explanatory text about modules and a warning about the update.php script. A table lists installed modules:

Enabled	Name	Version	Description
<input checked="" type="checkbox"/>	componente-postgrado	6.x-0.1.0	Componente de postgrado a integrar en la Intranet Depends on: componente-ws (enabled), Perfil (enabled)
<input checked="" type="checkbox"/>	componente-ws		Componente de ws comunes Required by: Perfil (enabled), componente-postgrado (enabled)

**FIG. 11** Resultado de la Instalación del componente en el entorno CMS DRUPAL V 6.X



## GLOSARIO DE TÉRMINOS

---

### **i MICROFOMATO**

Es una forma simple de agregar significado semántico a un contenido legible por el humano y que para la máquina es sólo texto plano. Están ideados para ser usadas en páginas web que usen HTML o XHTML, de manera tal de que la información pueda ser indexada, guardada, referenciada, reusada o combinada.

### **ii RSS**

Es el acrónimo para “*Really Simple Syndication*”, RSS es una tecnología que crea el contenido sindicado de un sitio web y permite a los usuarios web suscribirse a sitios web o blogs. Los usuarios de RSS reciben el contenido de varios sitios web o blogs de donde se han suscrito en una forma agregada. El contenido que es leído en un solo lugar a través de un Lector de *Feed*.

### **LECTOR DE FEED:**

Es la herramienta utilizada para leer RSS Feeds. Un usuario puede suscribirse a varios Feeds y leerlos en un solo lector que le sea más conveniente. Los lectores de Feeds están disponibles gratuitamente en muchos sitios. Alternativamente, existen lectores que son pagados los cuales el usuario los puede descargar e instalar en el disco duro de su computadora.

### **iii XUL**

XUL (acrónimo de XML-based User-interface Language, lenguaje basado en XML para la interfaz de usuario) es la aplicación de XML a la descripción de la interfaz de usuario en el navegador Mozilla.

### **iv SOA**

(en inglés, *Service Oriented Architecture*), es un concepto de arquitectura de software que define la utilización de servicios para dar soporte a los requisitos del negocio.

Permite la creación de sistemas altamente escalables que reflejan el negocio de la organización, a su vez brinda una forma estándar de exposición e invocación de servicios (comúnmente pero no exclusivamente servicios web), lo cual facilita la interacción entre diferentes sistemas propios o de terceros.

### **v XML:**

Es un lenguaje desarrollado por el *W3 Consortium* para permitir la descripción de información contenida en la red de redes a través de estándares y formatos comunes, de manera que tanto los usuarios de Internet como programas específicos (agentes) puedan buscar, comparar y compartir información en la red. El formato de XML es muy parecido al del HTML aunque no es una extensión ni un componente de éste. Sirve para que muchos programas interpreten bien cualquier tipo de dato y se comuniquen entre ellos sin intervención humana. Constituye el lenguaje base de los Servicios Web.

### **vi UDDI (UNIVERSAL DESCRIPTION DISCOVERY AND INTEGRATION):**

UDDI es un marco independiente de la plataforma para describir servicios, negocios e integrar servicios de negocios. La estructura de UDDI está basada sobre los servicios estándares de la web, lo que quiere decir que UDDI es accesible como otros servicios web. Los propietarios de los servicios web Servicios Web los publican en el registro UDDI. Una vez publicados se mantienen allí apuntadores a la descripción del Servicio Web y al servicio. UDDI permite a los clientes buscar tal registro, encontrar el servicio deseado y extraer sus detalles. Estos detalles incluyen el punto de invocación así como otras características del servicio y su funcionalidad.

---

### **vii EXTENSIÓN PECL**

(PHP Extension Community Library) es un conjunto de módulos para ampliar las capacidades de PHP (creación de pdf, manipulación de archivos zip, etc.). Varios de estos módulos estaban incluidos en PHP 4, pero ya no lo están en PHP 5.

### **viii BLOG**

Contracción de *Weblog*. Sitio web que contiene textos, enlaces o fotos publicados cronológicamente, en general por una sola persona, desde una perspectiva personal. [En el idioma español también se utiliza el término bitácora]

### **ix API**

Una interfaz de programación de aplicaciones o API (del inglés *Application Programming Interface*) es el conjunto de funciones y procedimientos (o métodos, si se refiere a programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.

Representa una interfaz de comunicación entre componentes software. Se trata del conjunto de llamadas a ciertas bibliotecas que ofrecen acceso a ciertos servicios desde los procesos y representa un método para conseguir abstracción en la programación, generalmente (aunque no necesariamente) entre los niveles o capas inferiores y los superiores del software. Uno de los principales propósitos de una API consiste en proporcionar un conjunto de funciones de uso general, por ejemplo, para dibujar ventanas o iconos en la pantalla. De esta forma, los programadores se benefician de las ventajas de la API haciendo uso de su funcionalidad, evitándose el trabajo de programar todo desde el principio. Las APIs asimismo son abstractas: el software que proporciona una cierta API generalmente es llamado la implementación de esa API.

### **x MANIFIESTO ÁGIL**

*The Agile Alliance*, una organización, sin ánimo de lucro, dedicada a promover los conceptos relacionados con el desarrollo ágil de software y ayudar a las organizaciones para que adopten dichos conceptos. El punto de partida es fue el *Manifiesto Ágil*, un documento que resume la filosofía ágil.

### **xi SISTEMAS GESTORES DE CONTENIDOS**

Logran resolver infinidad de cuestiones relativas a la administración y publicación de información online. Excepto en casos muy particulares donde se requiere un programa web a medida (ya sea para gestión específica como para cubrir funcionalidades y servicios que surgen "on demand"), los CMS son la solución ideal para cualquier tipo de sitios web.

### **xii PRUEBAS UNITARIAS**

Encargadas de verificar el código y diseñada por los programadores.