



Universidad Central de Venezuela
Facultad de Ciencias
Escuela de Computación

**Aplicación Web para la gestión de equipos, espacios
físicos y servicios del Centro de Experimentación de Recursos
Instruccionales de la UCV**

Trabajo Especial de Grado presentado ante la ilustre
Universidad Central de Venezuela por:

Br. Andrés Hevia

Br. Néstor Naranjo

Para optar por el título de
Licenciado en Computación

Tutores:

Profa. Yosly Hernández

Prof. Ángel Alvarado

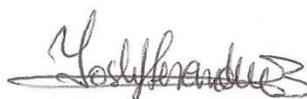
Octubre, 2016.

ACTA

Quienes suscriben, miembros del jurado designado por el Consejo de Escuela de Computación, para examinar el Trabajo Especial de Grado con el título "Aplicación web para la gestión de equipos, espacios físicos y servicios del Centro de Experimentación de Recursos Instruccionales de la UCV", el cual fue presentado por el Br. Andrés Javier Hevia Galea de Cédula de Identidad 18.493.323 y el Br. Néstor Manuel Naranjo Contrera, de Cédula de Identidad 18.713.388, a los fines de optar al título de Licenciado en Computación, dejamos en constancia lo siguiente:

Leído como fue, dicho trabajo por cada uno de los miembros del jurado, se fijó el día 19 de Octubre de 2016, a las 11:30 am en el aula PA-III de la Escuela de Computación, para que los autores lo defendieran en forma pública, mediante una presentación oral de su contenido, luego de lo cual se respondieron las preguntas formuladas. Finalizada la defensa pública del Trabajo Especial de Grado, el jurado decidió aprobarlo con una nota de 19 puntos.

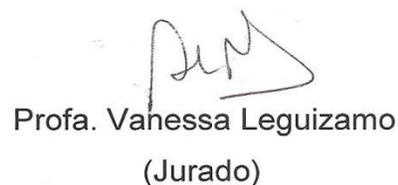
En fe de lo cual se levanta la presente Acta, en Caracas a los diecinueve (19) días del mes de Octubre de 2016.



Prof. Yosly Hernández
(Tutora)



Prof. Jaime Parada
(Jurado)



Prof. Vanessa Leguizamo
(Jurado)

Agradecimientos y Dedicatoria

A mi familia, que siempre me animó, aunque no entendiera ni un poco de lo que les contaba o hacía.

A todos mis profesores, desde preescolar hasta ahora. Cada uno, de una forma u otra, me enseñó algo importante.

A mi tutor de seminario, Franklin. A mis tutores de tesis, Yosly y Ángel. A todo el personal del CERI. Gracias por la amabilidad y atención.

A la Profa. Vanessa Leguizamo, que nos salvó en el último minuto.

A David, Adri y Beto, por su invaluable ayuda y amistad.

A Mafersita, la única despierta en los infinitos traspasos. Por favor, no duermas nunca.

A Steph, no podría haber encontrado una mejor persona para estudiar, programar, hacer equipo, o simplemente hablar, aunque sea un poco.

A todos los que seguramente olvidé mientras escribía esto.

A ti.

Néstor Manuel

Agradecimientos y Dedicatoria

A mis padres, porque sin ellos nunca hubiese llegado hasta donde he llegado, por su apoyo, paciencia, su fe y certeza en que podría lograr esto.

A mi cuñada y hermano, por su apoyo y ánimos.

A mi tutor de seminario Franklin Sandoval, por su paciencia, amistad y por encaminarnos.

A mi tutora Yosly Hernández por su paciencia, por creer en nosotros y por ser estricta, gracias a ella sacamos lo mejor de nosotros en este trabajo.

A mi tutor Ángel Alvarado, por su apoyo y buenos ánimos a pesar de las circunstancias.

A la Profa. Vanessa Leguizamo, por ser ese ángel que Dios pone en nuestro camino para hacer posible nuestra presentación.

A mi compañero de tesis, por no darse por vencido, por perseverar y por ser la calma y cordura a mi entusiasmo y brío.

A todos aquellos que de corazón estuvieron a mi lado para darme una mano, a quienes no he podido recordar, pero que me apoyaron de una u otra forma.

A ustedes.

Andrés Hevia

Resumen

El Centro de Experimentación de Recursos Instruccionales (CERI), de la Facultad de Humanidades y Educación de la Universidad Central de Venezuela pone a disposición de los miembros de la comunidad universitaria distintos equipos, espacios físicos y servicios de asesoría como apoyo a la labor educativa. Una de las múltiples labores que realiza el CERI es la gestión de estos recursos en forma de préstamos administrados por su personal. Como herramienta principal para llevar los registros de préstamos y usuarios se utilizaban hojas de cálculo en Excel, generando cuellos de botella en las solicitudes, imposibilidad de gestión remota y dificultad para resguardar los datos en ellos grabados. Por estas razones se hace imperiosa la necesidad de crear una herramienta de software que solvete estos problemas y facilite el proceso de reservas tanto para el personal del CERI como para la comunidad a la que sirve. Tomando en cuenta esta necesidad y con la intención de darle solución, se planteó como objetivo del presente Trabajo Especial de Grado el diseño, desarrollo e implementación de una aplicación web para el manejo de todos los procesos y actividades relacionadas con la reserva de equipos, espacios o servicios de asesoría que provee el CERI. En este trabajo se detallan los procesos (análisis, diseño, desarrollo, implementación y pruebas), lenguajes de programación (HTML, CSS, JavaScript, PHP), marcos de trabajo (CodeIgniter), librerías de desarrollo de software (jQuery y Bootstrap), tecnologías, herramientas y metodologías (AgilUs) utilizadas para alcanzar el objetivo, una aplicación web para la gestión de espacios físicos, equipos y servicios del CERI.

Palabras clave:

Aplicación web, gestión de recursos, CERI, préstamos, AgilUs, procesos de reserva.

Índice general

Resumen	IV
Índice general.....	VI
Índice de figuras	IX
Índice de tablas	XII
Introducción.....	10
Capítulo 1: Problema de Investigación	12
1.1. Contexto de investigación	12
1.2 Planteamiento del Problema	13
1.2. Justificación.....	17
1.3. Objetivo general	18
1.4. Objetivos específicos	18
1.5. Alcance	18
1.6. Método de desarrollo AgilUs	19
1.6.1. Ciclo de vida	20
1.6. Antecedentes	26
Capítulo 2: Marco teórico	30
2.1. Aplicaciones web	30
2.1.1. Características de las aplicaciones web	31
2.1.2. Ventajas de las aplicaciones web	33
2.1.3. Desventajas de las aplicaciones web	34
2.2. Arquitectura cliente-servidor	35
2.3. Patrón Modelo-Vista-Controlador	37
2.4. Herramientas para el desarrollo de aplicaciones web	39

2.4.1. Lenguaje de marcado de Hipertexto.....	39
2.4.2. Hojas de estilo en cascada.....	40
2.4.3. JavaScript.....	42
2.4.4. jQuery.....	43
2.4.5. JavaScript Object Notation.....	44
2.4.6. AJAX.....	46
2.4.7. PHP Hypertext Pre-processor.....	47
2.4.8 Marco de trabajo CodeIgniter.....	48
2.4.8.1. Características.....	48
2.4.8.2. Flujo de aplicación.....	50
2.4.9. Lenguaje de consulta estructurado.....	51
2.4.10. Sistema manejador de bases de datos MySQL.....	52
2.4.11. Licencia Creative Commons.....	54
Capítulo 3: Marco Aplicativo.....	56
3.1 Fase I: Requisitos.....	56
3.1.1 Tormenta de ideas.....	57
3.1.2 Entrevistas.....	58
3.1.3 Perfiles de usuario.....	59
3.1.4 Requerimientos funcionales y no funcionales.....	61
3.2 Fase II: Análisis.....	63
3.2.1 Prototipos en papel.....	64
3.2.2 Guía de estilos.....	68
3.2.3 Casos de uso.....	71
3.2.4 Modelo de objetos del sitio web de préstamos.....	86
3.3 Fase III: Prototipaje.....	88

3.3.1 Prototipo ejecutable.....	88
3.2.3 Evaluación Heurística.....	92
3.2.4 Lista de Comprobación.....	94
3.4 Fase IV: Fase de entrega.....	95
3.4.1 Protocolo de preguntas.....	95
3.4.2 Pruebas de aceptación.....	97
Capítulo 4: Resultados.....	98
4.1. Interfaces y funcionalidades de usuarios anónimos.....	98
4.2. Interfaces y funcionalidades de usuarios registrados.....	100
4.2.1. Consultar equipos y espacios.....	101
4.2.2. Ver perfil.....	101
4.2.3. Modificar perfil.....	103
4.2.4. Recuperar contraseña.....	104
4.3 Interfaces y funcionalidades de administradores.....	105
4.3.1. Panel de control.....	105
4.3.2. Menú principal.....	106
4.3.3. Interfaces de solicitud.....	107
4.3.4. Interfaces de usuarios.....	111
4.3.5. Interfaces de categorías.....	111
4.3.6. Interfaces de recursos.....	112
4.3.7. Interfaces de estadísticas.....	114
Conclusiones.....	117
Recomendaciones.....	118
Trabajos Futuros.....	118
Referencias bibliográficas.....	119

Índice de figuras

Figura 1. Diagrama de flujo del proceso de reserva.....	14
Figura 2. Hoja de cálculo para el proceso de reserva	15
Figura 3. Ciclo de vida de AgilUs	20
Figura 4. Aplicación de reserva de la Universidad de Zaragoza	26
Figura 5. Formulario de solicitud, aplicación Universidad de Zaragoza	27
Figura 6. Aplicación web de Reserva de espacios, Universidad de Murcia.	28
Figura 7. Aplicación web de reserva, Universidad Politécnica de Valencia.....	28
Figura 8. Arquitectura cliente-servidor, proceso de petición-respuesta.....	36
Figura 9. Patrón Modelo-Vista-Controlador.....	38
Figura 10. Ejemplo de código HTML.....	40
Figura 11. Ejemplo de código CSS.	42
Figura 12. Ejemplo de código JavaScript.....	43
Figura 14. Construcción de un arreglo JSON.....	45
Figura 15. Tipos de valores válidos para un arreglo u objeto JSON.	45
Figura 16. Flujo de una aplicación en CodeIgniter.	50
Figura 17. Mapa mental de tormenta de ideas	57
Figura 18. Prototipo en papel con un formulario.....	64
Figura 19. Prototipo en papel con la vista de un listado.	65
Figura 20. Prototipo con la vista de un formulario de registro.	66
Figura 21. Prototipo con la vista general de un usuario anónimo.....	66
Figura 22. Prototipo con la vista general de un usuario registrado.....	67
Figura 23. Prototipo con la vista general de un administrador.....	67
Figura 24. Casos de uso de nivel 0.	71

Figura 25. Casos de uso de nivel 1.	72
Figura 26. Casos de uso del administrador de nivel 2.....	73
Figura 27. Casos de uso del usuario registrado de nivel 2.....	74
Figura 28. Casos de uso del usuario anónimo de nivel 2.....	74
Figura 29. Modelo de objetos del dominio.....	87
Figura 30. Interfaz de ingreso al sistema.	88
Figura 31. Interfaz de recuperación de contraseña.....	89
Figura 32. Panel de control.	89
Figura 33. Menú principal.....	90
Figura 34. Opción de solicitudes del menú.	90
Figura 35. Interfaz de creación de una solicitud.....	91
Figura 36. Interfaz de listado de solicitudes.	91
Figura 37. Interfaz de ingreso al sistema.	98
Figura 31. Interfaz de registro	100
Figura 32. Interfaz de consulta de equipos y espacios disponibles.....	101
Figura 33. Interfaz de detalles del usuario	102
Figura 34. Interfaz de actualización de datos de usuario	103
Figura 35. Interfaz de recuperación de contraseña.....	104
Figura 36. Interfaz de cambio de contraseña	105
Figura 37. Panel de control.	106
Figura 38. Menú principal.....	107
Figura 39. Interfaz de creación de una solicitud.....	108
Figura 40. Interfaz de listado de solicitudes.	109
Figura 41. Interfaz de cierre de solicitudes.....	109
Figura 42. Interfaz de confirmación de cierre de solicitudes.	110

Figura 43. Listado de usuarios.	111
Figura 44. Interfaz de listado de categoría de equipos.....	112
Figura 45. Interfaz de agregar nueva categoría de equipo.....	112
Figura 46. Interfaz de agregar nuevo espacio.	113
Figura 47. Listado de espacios.....	113
Figura 48. Interfaz de modificación de espacios.	114
Figura 49. Interfaz inicial estadísticas generales.....	115
Figura 50. Interfaz simple de estadísticas generales.	116
Figura 51. Interfaz extendida de estadísticas generales.	116

Índice de tablas

Tabla 1. Guía de estilos de la aplicación web	68
Tabla 2. Continuación de guía de estilos de la aplicación web	69
Tabla 3. Continuación de guía de estilos de la aplicación web	70
Tabla 4. Caso de uso 1. Gestionar reservas	74
Tabla 5. Caso de uso 1.1. Ingresar como administrador	75
Tabla 6. Caso de uso 1.2. Administrar solicitudes.....	75
Tabla 7. Caso de uso 1.3. Administrar usuarios.....	76
Tabla 8. Caso de uso 1.4. Administrar categorías de usuario.....	77
Tabla 9. Caso de uso 1.5. Administrar categorías de equipo.....	78
Tabla 10. Caso de uso 1.6. Administrar categorías de servicios.....	79
Tabla 11. Caso de uso 1.7. Administrar equipos.....	79
Tabla 12. Caso de uso 1.8. Administrar servicios.	80
Tabla 13. Caso de uso 1.9. Administrar espacios.	81
Tabla 14. Caso de uso 1.10. Administrar usos.....	82
Tabla 15. Caso de uso 1.11. Visualizar estadísticas.	82
Tabla 16. Caso de uso 2. Consultar reservas	83
Tabla 17. Caso de uso 2.1. Ingresar como usuario	84
Tabla 18. Caso de uso 2.2. Consultar disponibilidad de espacios y equipos.	84
Tabla 19. Caso de uso 2.3. Editar perfil de usuario.....	85
Tabla 20. Caso de uso 3. Visualizar el sitio web.	85
Tabla 21. Caso de uso 3.1. Crear cuenta de usuario.....	86

Introducción

Las tecnologías de la información y comunicación son utilizadas cada día en mayor medida para optimizar todo tipo de procesos, que serían propensos a errores y bastante repetitivos, de tener que hacerlos manualmente.

El Centro de Experimentación de Recursos Instruccionales (CERI), parte de la Facultad de Humanidades y Educación de la Universidad Central de Venezuela, se encarga de apoyar directamente a la docencia mediante la producción y desarrollo de recursos instruccionales con base en los medios y tecnologías de cada época.

Una de las funciones del CERI es la gestión de los procesos de reserva de los equipos, espacios físicos y servicios de asesoría brindados por su personal. Dicha gestión ha sido llevada a cabo mediante el uso de una hoja de cálculo en Excel, limitando en gran medida las opciones del personal, careciendo de opciones de seguridad para el documento, y dificultando la generación de estadísticas, por consiguiente, haciendo que el proceso sea innecesariamente engorroso.

Para solucionar los inconvenientes planteados, se desarrolló una aplicación web para la gestión de los equipos, espacios físicos y servicios del CERI. Dicha aplicación fue desarrollada utilizando diversos lenguajes de programación, librerías, marcos de trabajo y sistemas manejadores de bases de datos (HTML, CSS, JavaScript, jQuery, PHP, CodeIgniter y MySQL), así como la metodología de desarrollo de software AgilUs.

Al implementar la aplicación desarrollada en el CERI, se puede concluir que los problemas planteados anteriormente han sido solventados de manera exitosa.

El presente documento de investigación plantea la implementación de la aplicación de reserva, y está conformado por los siguientes capítulos:

- Capítulo 1: Problema de investigación. En este capítulo se explica el proceso a seguir para realizar una reserva. Se pone en contexto el problema, los objetivos generales y específicos que se pretenden alcanzar, se justifica la investigación, se delimita su alcance, y se muestran otras aplicaciones que preceden la presente investigación.

- Capítulo 2: Marco Conceptual. En este capítulo se describen los conceptos referentes a los diferentes lenguajes, marcos de trabajo, librerías, herramientas y tecnologías, mostrando también las fases de la metodología de desarrollo de software que será utilizada para el desarrollo de la aplicación que resolverá el problema planteado en el anterior capítulo.

- Capítulo 3: Marco Aplicativo. En este capítulo se exponen y describen las iteraciones llevadas a cabo durante el desarrollo del Trabajo Especial de Grado (TEG), especificando las etapas contempladas en el método de desarrollo *AgilUS*, seleccionado para el desarrollo del proyecto, detallando las actividades realizadas, los artefactos generados y las entregas efectuadas.

- Capítulo 4: Resultados. Se exponen los resultados del presente TEG, mostrando cada una de las funcionalidades e interfaces de la aplicación desarrollada, demostrando así que los objetivos propuestos fueron alcanzados.

Capítulo 1: Problema de Investigación

En este capítulo se expone la problemática, la importancia y justificación del problema, el objetivo general y los objetivos específicos, el alcance definido y algunas aplicaciones que preceden la presente investigación.

1.1. Contexto de investigación

La investigación estuvo centrada en el Centro de Experimentación de Recursos Instruccionales (CERI) de la Facultad de Humanidades y Educación en la Universidad Central de Venezuela.

De acuerdo a Alvarado (2016), el CERI nace en marzo de 1981, debido a la necesidad de algunas de las menciones de la Escuela de Educación de contar con una unidad que pudiera asumir las funciones de diseño, producción y evaluación de medios instruccionales para apoyar directamente la docencia.

“El CERI es un centro de innovación que desarrolla y experimenta con los medios y las tecnologías de la información y comunicación al servicio de la educación. En las décadas siguientes, después de convertirse en una referencia nacional e internacional, el centro entra en un período crítico, durante el cual mermó la capacidad de producción de recursos instruccionales. Es en el año 2012, cuando el personal del CERI, un grupo de docentes y estudiantes de la FHE se dedicaron a repensar el centro y retornar al sendero que señalan los nuevos escenarios de la Educación y las Tecnologías de la Información y Comunicación (TIC)” (Alvarado, 2016).

Según Alvarado (2016), entre los múltiples objetivos del CERI, los siguientes son los más importantes:

- Producir medios instruccionales que cuenten con criterios de calidad didáctica y técnica que contribuyan con el avance y mejora de la calidad educativa.
- Prestar apoyo en el uso didáctico y técnico de equipos y espacios que faciliten el desarrollo de actividades didácticas.
- Promover el uso de las tecnologías de la información y comunicación con el fin de contribuir al avance y mejora de la calidad educativa.
- Asesorar en lo didáctico y técnico el desarrollo de estrategias y uso de espacios para la innovación tecnológica.

1.2 Planteamiento del Problema

De las muchas funciones que tiene el CERI, el foco de este trabajo de investigación fue controlar la reserva de equipos, espacios y servicios prestados por el centro. Entre los espacios que son administrados por el CERI, se encuentran:

- Sala Anfiteátrica
- Sala de Proyección
- Sala de Informática “Piso 3”
- Sala de Informática “Lautaro Videla”
- Estudio de Grabación
- Sala de Edición Digital
- Laboratorio de Fotografía

De acuerdo al diagrama de flujo de la figura 1, para realizar una reserva, el usuario debía comunicarse por correo electrónico, teléfono o presentarse en las instalaciones del CERI, tanto para solicitar información de disponibilidad como para realizar la solicitud formal de los espacios, equipos o servicios que necesitara.

Posteriormente el personal del CERI, en caso de haber disponibilidad para la fecha y hora solicitada, registraba en una hoja de cálculo los datos personales y los equipos, espacios y servicios a reservar.



Figura 1. Diagrama de flujo del proceso de reserva

En la figura 2 se puede observar la hoja de cálculo en donde se registraban los datos de la solicitud.

Universidad Central de Venezuela
Facultad de Humanidades y Educación
Escuela de Educación
Centro de Experimentación de Recursos Instruccionales
CERI

Ordenar Solicitudes

CERI UCV
Centro de Experimentación de Recursos Instruccionales

Reservación de Salas y Equipos

Nº	(A) Fecha		(B) Tiempo de Uso		(C) Datos Personales				(D) Espacios	(E) Uso	(F) Prestamos de Equipos			(G) Personal del CERI		
	(A1) Solicitud	(A2) Uso	(B1) Hora de Entrega	(B2) Hora de Recibo	(C1) C.I.	(C2) Nombre y Apellido	(C3) Teléfono	(C4) Tipo de Usuario			Equipo (F1)	Equipo (F2)	Equipo (F3)	(G1) Reservado por	(G2) Recibido por	Obse
13		26/01/16	10:30 AM	12:15 PM	10.992.516	Maria Seijas	4242814892	Docente			(1) CESIO					
14	27/01/16	27/01/16	5:00 PM	6:30 PM	11.553.600	Maria Ferreir	4141340990	Docente			(1) CESIO	(1) vit		Cecilia Obis	Ana Cristina	
15	27/01/16	03/02/16	5:00 PM	6:30 PM	11.553.600	Maria Ferreir	4141340990	Docente						Ana Cristina		
16	28/01/16	29/01/16	2:00 PM	5:00 PM	16.369.438	Laura Rodng	4142688143	Docente			(2) CESIO	(3) compaq		Efrain Zapata		
17	28/01/16	28/01/16	10:45 AM	12:00 p.m	16.369.438	Laura Rodng	4142688143	Docente			(2) CESIO	(3) compaq		Efrain Zapata	Anniel Ruiz	
18	28/01/16	28/01/16	3:26 AM		7.323.618	Ruben		Docente	Sala de Proyección	Experiencia Formativa				Dixon Guap		Conv
19	01/02/16	01/02/16	11:25 AM	12:30 p.m	6.098.219	Hiroshima P.	4123238203	Docente	Sala de Proyección	Yesis Apoyo con TIC				Ana Cristina	Ana Cristina	
20		05/02/16	1:00 AM		6.907.510	Roxana Coln	4263203335	Estudiante	Sala de Proyección		(2) CESIO	(3) compaq		Ana Cristina		
21																
22																
23																

PROGRAMACIÓN Estadísticos Enero Febrero Marzo Abril Mayo Junio Julio Agosto Septiembre Octubre Noviembre Diciembre Hoja1

Figura 2. Hoja de cálculo para el proceso de reserva

Al observar el proceso de reserva, se aprecian las siguientes situaciones problema:

- Cuello de botella: debido a que el archivo de Excel solo puede ser manipulado por uno los miembros del CERI a la vez, no pueden hacerse varias solicitudes en paralelo por lo que los usuarios que hayan establecido comunicación con el personal del CERI al mismo tiempo que otros deberán esperar su turno para ser atendidos.
- Modificación de un espacio o equipo: Si uno o más espacios o equipos dejan de estar disponible, el administrador deberá no solo borrarlos de la hoja de cálculo respectiva de forma manual, sino de cada hoja de cálculo por cada mes y de cada registro donde haya una solicitud con esos espacios o equipos eliminados.
- Seguridad de los datos: Cualquier persona con acceso al equipo donde se encuentre guardado el archivo de Excel puede fácilmente leerlo, alterarlo y en el peor de los casos eliminarlo. Esto compromete la seguridad de la información dentro de ese archivo entre los cuales se encuentran datos personales de los solicitantes.

- Generación de reportes y estadísticas: se tienen fórmulas para registrar la cantidad total de usuarios de una categoría de usuario que hacen solicitudes en un mes, el número total de horas que una categoría de usuario tiene sumando todas las solicitudes hechas por esa categoría en un mes, cantidad total de equipos pertenecientes a una categoría prestados en un mes, el número de horas que se utilizó una sala o espacio en un mes, la cantidad de usos y tiempo total de esos usos de las salas y espacios en un mes. Estas fórmulas deben ser copiadas manualmente en una hoja de cálculo nueva por cada año que el CERI desee guardar esas estadísticas. Dado que existen muchas celdas combinadas, el copiado debe hacerse campo por campo ya que al copiar y pegar el programa arroja un error indicando que la operación no puede realizarse por diferencia en los tamaños de las celdas destino.
- Disponibilidad previa: el usuario no tiene manera de saber que espacios o equipos están disponibles para una fecha u hora determinada sin comunicarse directamente con el personal del CERI, lo cual lo limita al horario laboral del personal y la posibilidad que tengan de atenderlos.

Con base en lo expuesto anteriormente, surge la pregunta: ¿Cómo mejorar el proceso de gestión de reserva de los equipos, espacios físicos y servicios de asesoría del CERI? Siendo la respuesta el objetivo general de este trabajo de investigación cuya justificación se detalla a continuación.

1.2. Justificación

Desde que se liberó el uso de internet, se ha masificado de manera muy rápida, hasta convertirse en parte primordial de nuestra vida cotidiana. Son numerosos los lugares que aprovechan las ventajas que brinda para simplificar múltiples tareas, como automatizar procesos administrativos, que serían mucho más engorrosos sin la tecnología a nuestra disposición.

El CERI es un equipo de la Escuela de Educación creado por la necesidad que tenían –en aquel entonces- las menciones de tecnología educativa, planificación escolar, administración escolar y orientación escolar de contar con una unidad que asumiera las funciones y tareas en diseño, producción y evaluación de medios instruccionales de apoyo directo a la docencia.

Entre las funciones del CERI, se encuentra gestionar la reserva de equipos, espacios físicos y servicios de asesoría prestados a la comunidad. Muchos lugares que disponen de espacios propicios para diversas actividades, se apoyan en el uso de la tecnología, mediante aplicaciones web, para llevar un control de dicha reserva. Usualmente consisten en varios formularios con la intención de recabar la información necesaria para realizar la solicitud de manera satisfactoria.

Debido al aumento en el volumen de las solicitudes realizadas al CERI, el equipo se ha visto en la necesidad de obtener herramientas tecnológicas que les permitan manejar dichas solicitudes y atenderlas de forma efectiva y eficiente, para el beneficio de sus empleados y la comunidad que hace parte del centro.

Por las necesidades anteriormente planteadas, se propone implementar una aplicación web que permita gestionar todo lo relacionado con el préstamo de equipos, espacios físicos y servicios de asesoría a la par que almacene información de manera persistente para generar estadísticas que puedan ser analizadas,

facilitando la toma de decisiones en beneficio de todos los miembros de la comunidad del CERI.

1.3. Objetivo general

Desarrollar una aplicación web para la gestión y reserva de los equipos, espacios físicos y servicios de asesoría del CERI-FHU.

1.4. Objetivos específicos

- Determinar las funcionalidades que poseerá la aplicación.
- Seleccionar la arquitectura de la aplicación.
- Describir el modelo lógico y físico de la base de datos de la aplicación.
- Diseñar la interfaz gráfica de la aplicación.
- Aplicar pruebas de funcionalidad, usabilidad y aceptación, para verificar el funcionamiento apropiado de la aplicación.
- Implementar la aplicación web en un servidor del CERI.

1.5. Alcance

En esta sección se explica el alcance o delimitación del trabajo de investigación:

- **Objetivo:** Desarrollar una aplicación web para la gestión y reserva de los equipos, espacios físicos y servicios de asesoría del CERI-FHU.
- **Propósito:** Determinar los procesos que comprenden la gestión y reserva de espacios, equipos y servicios de asesorías del CERI.
- **Ubicación:** el trabajo se limita a la instalación y puesta en funcionamiento de la aplicación en una máquina servidor en las oficinas del CERI-FHU.

1.6. Método de desarrollo AgilUs

Para el desarrollo de la aplicación web se decidió usar el método de desarrollo AgilUS, el cual centra sus procesos en el usuario, brinda flexibilidad en la definición de los requerimientos del sistema y reduce los tiempos de desarrollo. Es por esta razón que AgilUS es el método que mejor se adecua al contexto en el cual se enmarca el presente trabajo de investigación.

Los cambios y adaptaciones del sistema se determinan en función de los requerimientos. Las funcionalidades del sistema se van añadiendo progresivamente, es por esta razón que el usuario no se tiene que adaptar al sistema, sino el sistema se va adaptando acorde a las necesidades del usuario.

Los aspectos de la interfaz en una aplicación resultan un punto importante que debe ser tomado en consideración con la finalidad de lograr una experiencia agradable entre el usuario final y la aplicación.

Con el método de desarrollo AgilUS se persigue abarcar aspectos de usabilidad incorporando una visión que complementa el área de Interacción Humano-Computador con la Ingeniería de Software, dando lugar a un conjunto de buenas prácticas y evaluaciones aplicadas al software. "Se plantea el desarrollo de un sistema partiendo de un prototipo de la interfaz de usuario, y que incorpora la aplicación de diversas técnicas de evaluación de usabilidad desde el inicio del ciclo de vida de la aplicación" (Acosta, 2010).

Para alcanzar la usabilidad, en cada etapa de desarrollo se considera como eje principal las características y actividades comúnmente realizadas por el usuario, aplicando técnicas de diseño centradas en el mismo; de esta manera se determinan las metas que contemplan cada una de las etapas, así como los objetivos que debe satisfacer el producto final.

AgilUS se trata de un método de desarrollo basado en iteraciones, en donde cada iteración abarca diferentes actividades que se ejecutan de manera incremental,

otorgando prioridad a los aspectos de diseño y desarrollo en términos de usabilidad. El aspecto estético de la interfaz no se deja como tarea final, sino se toma en cuenta a lo largo del desarrollo de las etapas.

1.6.1. Ciclo de vida

Como se ha mencionado anteriormente, en AgilUS, la usabilidad es un aspecto que debe incluirse en cada etapa de desarrollo; para ello se toma en consideración la evaluación que realiza el usuario luego de interactuar con el sistema. En cada etapa, se desarrollan diferentes prototipos que pueden mejorarse en base a las observaciones realizadas.

Como se observa en la figura 3, cada etapa del ciclo de desarrollo se lleva a cabo de manera incremental e iterativa mediante la construcción de prototipos hasta obtener un producto final para la entrega



Figura 3. Ciclo de vida de AgilUs

Cada una de las etapas incluye la construcción de diversos artefactos, que pueden ser sujetos a modificaciones hasta obtener el producto final. A continuación se realiza una descripción de cada etapa:

- Requisitos: se lleva a cabo el levantamiento de requerimientos, generando un perfil de usuario. Cada requerimiento definido se debe desarrollar e implementar para cumplir con las exigencias del usuario.
- Análisis: una vez estipulados los requisitos que persigue el sistema, se debe realizar un análisis sobre los mismos, para determinar de qué manera serán implementados, para lo cual se toma como apoyo los diferentes diagramas realizados bajo la notación estándar del Lenguaje de Modelado Unificado ó UML (por sus siglas en inglés: Unified Modeling Language).
- Prototipaje: son elaborados diversos prototipos, incluyendo un prototipo de Interfaz Gráfica de Usuario. En esta etapa, el usuario lleva a cabo la evaluación de los prototipos a fin de conocer si los mismos cumplen con los principios de usabilidad.
- Entrega: consiste en la aplicación de pruebas finales para confirmar que la aplicación cumple con los principios de usabilidad. Posteriormente, la aplicación es pasada a producción.

Mediante el diseño de prototipos que evolucionan en función de las necesidades del usuario, el sistema va estructurándose hasta llegar a convertirse en el producto final.

1.6.2. Características de AgilUS

El método de desarrollo AgilUS está orientado al cumplimiento de los requerimientos básicos estipulados por el usuario, avanzando incrementalmente en cada una de las etapas y tomando en cuenta los aspectos de usabilidad. Entre las

características, bondades y buenas prácticas estipuladas en el método, se encuentran:

- Diseño centrado en el usuario (DCU), basado en las preferencias e inclinaciones que presenta el usuario frente al sistema que desea. “... en el DCU se procura construir el sistema para adaptarse, a través de su interfaz, a cómo el usuario desea trabajar, en lugar de forzar al usuario a cambiar su modo de trabajar para adaptarse a lo que los desarrolladores consideraron apropiado” (Acosta, 2010).
- Diseño basado en prototipos, donde cada uno es revisado y probado por usuarios finales y especialistas en el área de usabilidad. Este proceso se lleva a cabo de manera cíclica hasta culminar la elaboración del producto final.
- Simplicidad en el desarrollo del sistema, previendo posibles cambios en el futuro.
- Desarrollo incremental, de manera de establecer y cumplir requisitos a medida que se desenvuelve el proyecto.
- Desarrollo iterativo, permitiendo el rediseño e implementación de actividades llevadas a cabo en cada una de las etapas. El objetivo al diseñar e implementar en cada iteración consiste en realizarlo de manera simple para no dejar de un lado la posibilidad de rediseñar.
- Usabilidad como requisito esencial para considerar un software de calidad.
- Interacción entre el equipo desarrollador y el usuario, un aspecto vital si se considera la usabilidad en cada etapa, ya que es el usuario quien determina la misma.

1.6.3 Entregables de AgilUS

A continuación se describen los artefactos utilizados durante el desarrollo del trabajo especial de grado. Los mismos involucran las actividades en la utilización del método de desarrollo AgilUS para el desarrollo de las funcionalidades de gestión de los procesos de reserva.

- Tormenta de ideas

La tormenta de ideas fue el primer método creativo propuesto para los grupos. Fue formulada por Alex Osborn y se trata de la práctica de una técnica de conferencia en la que un grupo de personas busca la solución a un problema específico, juntando todas las ideas aportadas en forma espontánea por sus integrantes. Es útil para atacar problemas específicos, recolectar buenas ideas y donde no hace falta analizar profundamente las decisiones que se toman.

- Entrevista

Se trata de una técnica para recopilar información, la cual permite establecer un diálogo a través de un esquema dado, persiguiendo un propósito profesional.

Ocurre entre dos o más personas, donde uno es el entrevistador y otro u otros son los entrevistados. Es uno de los procedimientos más usados en la investigación social, aunque como técnica profesional se usa en otras tareas. Cada uno lo usa de acuerdo a los fines que persiga con su actividad (Egg, 1979).

- Perfil de usuario

El perfil de usuario define un conjunto de rasgos característicos que sirven como fundamento para la planificación y/o realización de cualquier servicio, sistema de información o módulo.

Los perfiles de usuarios constituyen un elemento básico para el desarrollo de sistemas de información y pueden ser obtenidos siguiendo un proceso denominado Estudios de Usuario, el cual de acuerdo a Hernandez (2011) consiste en realizar análisis tanto en el comportamiento de los usuarios como en las tareas que los mismos desempeñan y sus responsabilidades.

- Requerimientos funcionales y no funcionales

Los requerimientos son una descripción de las necesidades o deseos de un producto. La meta principal al aplicar esta actividad es identificar y documentar lo que en realidad se necesita, en una forma en que pueda fácilmente ser transmitido al usuario y al equipo de desarrollo.

Los requerimientos funcionales corresponden a las peticiones que tienen los usuarios para cumplir con sus necesidades. Expresan la naturaleza del funcionamiento del sistema (cómo interacciona el sistema con su entorno y cuáles van a ser su estado y funcionamiento).

Los requerimientos no funcionales corresponden a los atributos de calidad del software, los cuales involucran aspectos como la seguridad, eficiencia, usabilidad, portabilidad, interoperabilidad, escalabilidad, entre otros. No se refieren directamente a las funciones específicas que entrega el sistema, sino a las propiedades emergentes de éste.

- Casos de uso

Los Casos de Uso son la base para el trabajo del equipo de desarrollo y permiten definir la arquitectura, diseño, pruebas funcionales y diseño de interfaces de usuario.

Los casos de uso vienen acompañados de un diagrama empleado para especificar la comunicación y el comportamiento de un sistema mediante su interacción con los usuarios y/u otros sistemas. Hace uso del Lenguaje de Modelado Unificado (Unified Modeling Language, en inglés) utilizado para especificar, visualizar, construir y documentar los artefactos de sistemas de software, así como modelar lógica de negocios. Permite comunicar claramente requerimientos, arquitecturas y diseños de manera estándar de la industria.

- Modelo de objetos del dominio

Un modelo de objetos del dominio describe los objetos que se identifican en el dominio de la aplicación y las relaciones con su respectiva cardinalidad. Este modelo se expresa mediante un diagrama de clases del Lenguaje de Modelado Unificado. Un diagrama de clases muestran las diferentes clases que componen un sistema y cómo se relacionan unas con otras como indica Ingeniero de software (2008).

- Guía de estilo

Se define como guía de estilo al documento que recoge normativas y patrones básicos relacionados con el aspecto de una interfaz para su aplicación en el desarrollo de nuevas pantallas dentro de un entorno concreto.

- Evaluación heurística

La evaluación heurística es un método de inspección cuyo objetivo es encontrar problemas de usabilidad en el diseño de la interfaz de usuario, tal que puedan ser atendidos como parte de un proceso de diseño iterativo. Puede ser realizada por personal no especializado e incluso por usuarios tipo.

Se pueden tomar como criterios de evaluación las siguientes heurísticas según Nielsen (1994):

- H1: Diálogo natural y simple
- H2: Hablar el lenguaje del usuario
- H3: Minimizar la carga cognitiva
- H4: Consistencia
- H5: Feedback (retroalimentación)
- H6: Proveer claramente las salidas
- H7: Proveer Shortcuts (atajos de teclado)
- H8: Mensajes de error descriptivos
- H9: Prevención de errores
- H10: Asistencia al usuario

Es necesario determinar una escala para valorar los problemas detectados al realizar la evaluación heurística, tomando en cuenta la frecuencia con la que ocurre el problema, el impacto que conlleva y la persistencia.

El presente trabajo tiene como objetivo la creación de una aplicación web cuyas funcionalidades están enteramente dictadas por las necesidades del personal del CERI en materia de reserva de espacios, equipos y servicios. Debido a que esta metodología está enfocada enteramente hacia el usuario y su principal motivación es

lograr una interfaz que no solo sea cosméticamente agradable sino también fácil de usar, fue elegida para servir de guía en la construcción del prototipo final.

1.6. Antecedentes

Debido a que el objetivo del trabajo es el desarrollo de una aplicación de índole administrativa, que cumplen con el propósito de realizar reserva de equipos o espacios. Se procederá a mencionar algunas de ellas, que sirven como antecedentes de la presente investigación.

En la figura 4 se puede apreciar la vista principal de la aplicación de reserva de la Universidad de Zaragoza. La aplicación permite elegir entre los distintos campus, edificios, y áreas de cada edificio. Permite seleccionar un día a futuro, y está seleccionado por defecto el día actual.

The screenshot displays the 'Meeting Room Booking System' interface. At the top, there is a navigation bar with the university logo and name, and a title 'Área de Ordenadores Personales Meeting Room Booking System'. Below this, there are links for 'Help', 'Admin Ver Solicitudes', 'Report', and a search field. A calendar shows the current date, 'miércoles 05 octubre 2016', and the selected location, 'San Francisco - Fac. Ciencias B-Matemáticas - SICUZ'. A table at the bottom shows the booking schedule for the selected room, with time slots from 08:00 to 11:00.

Time:	SICUZ - Aula 1(18)	Time:
08:00	*	08:00
08:30	*	08:30
09:00	*	09:00
09:30	*	09:30
10:00	*	10:00
10:30	*	10:30
11:00	*	11:00

Figura 4. Aplicación de reserva de la Universidad de Zaragoza

Fuente: reservadeaulas.unizar.es (2016)

En la figura 5 se muestra la vista después de seleccionar una fecha y hora. Como indica la imagen, se le avisa al usuario que una solicitud no es una reserva, y que recibirá dos correos de confirmación: uno que indica que se ha recibido la solicitud, y otro referente a la aprobación o rechazo de la misma.

Nueva solicitud de reserva

Aula solicitada: SICUZ- Aula 1

Tipo sala: SICUZ
 Edificio: Fac. Ciencias B-Matemáticas
 Campus: San Francisco

Recuerda:

- Una solicitud **no es una reserva**.
- La solicitudes deben ser aceptadas para que sean visibles en estas páginas.
- Recibirá **dos correos electrónicos**:
 el primero indicará que se ha registrado la solicitud
 el segundo que la solicitud ha sido aceptada/denegada y le mostrará el enlace a donde puede comprobar la reserva.

Brief Description:

Correo Electrónico:

**Full Description:
 (Number of people,
 Internal/External etc)**

Date:

Time:

Duration: All day

Rooms:

Type:

*Figura 5. Formulario de solicitud, aplicación Universidad de Zaragoza
 Fuente: reservadeaulas.unizar.es (2016)*

Entre los elementos que debe suministrar el usuario para una reserva exitosa, se encuentra una breve descripción de la actividad a realizar en el espacio, el correo electrónico al cual se enviará la información, una descripción completa de la actividad, la fecha y hora de inicio, así como duración y confirmación del espacio que desea ser reservado.

De no haber ingresado al sistema al momento de confirmar la reserva, se solicita que ingrese el nombre de usuario y la contraseña, para confirmar que es un miembro de la universidad.

Otro ejemplo se puede visualizar en la figura 6, la vista de ingreso a la aplicación web de reservas de la Universidad de Murcia.



Figura 6. Aplicación web de Reserva de espacios, Universidad de Murcia.

Fuente: carnesweb.um.esb (2016)

Como ejemplo final, se observa la aplicación web de reservas de la Universidad Politécnica de Valencia. Esta aplicación posee vistas de ocupación diaria y semanal. La figura 7 muestra la ocupación diaria, en el horario que labora la Universidad. Muestra las distintas aulas que pueden ser reservadas, y, en caso de que ya se encuentren ocupadas, identifica con las iniciales del profesor o estudiante que la solicitó. Al moverse sobre alguna de las iniciales, despliega un cuadro con detalles del solicitante.



Figura 7. Aplicación web de reserva, Universidad Politécnica de Valencia.

Fuente: www.inf.upv.es (2016)

Los anteriores son ejemplos de aplicaciones web para realizar la reserva de espacios físicos, funcionalidad que se asemeja a la aplicación a desarrollar por lo

cual sirven como antecedentes de la presente investigación. Finalmente, puede afirmarse que existen otras aplicaciones que solucionaron problemas similares al planteado anteriormente de forma exitosa, por lo tanto, este Trabajo Especial de Grado tiene la intención de desarrollar e implementar una aplicación web realizar el proceso de reservación de equipos, espacios físicos y servicios del CERI, solventando los problemas ocasionados por el uso de hojas de cálculo en Excel mencionados anteriormente.

Para concluir, en este capítulo se habló del CERI y la FHE como contexto de la investigación, se describió el problema representado por la utilización de archivos de hojas de cálculo en el proceso de reservas, se examinó la justificación del problema, se indicó la implementación de una aplicación web para la gestión de espacios físicos, equipos y servicios prestados por el CERI como objetivo general, fueron señalados los objetivos específicos, se delimitó el alcance del trabajo, fue explicado el método de desarrollo AgilUs, la razón tras su elección, y se mostraron los antecedentes asociados al trabajo de investigación.

Se procede a establecer los conceptos que sustentan el presente trabajo de investigación.

Capítulo 2: Marco teórico

En el siguiente capítulo se establecen las bases teóricas y las herramientas tecnológicas en las que se sustenta la solución propuesta. De acuerdo con Lazo (2003), el marco conceptual de un trabajo de investigación ayuda a explicar por qué se lleva a cabo un proyecto de una manera determinada. También ayuda a comprender y utilizar ideas de otras investigaciones con trabajos similares.

Los conceptos presentados a lo largo del capítulo, son organizados de tal manera que el lector pueda observar una visión en conjunto de las ideas y prácticas utilizadas en el trabajo de investigación, expresados de una forma que facilita su comprensión.

El capítulo está dividido en cuatro secciones principales. La primera son las aplicaciones web, sus características, ventajas y desventajas. La segunda parte describe la arquitectura cliente-servidor. La tercera parte, detalla el patrón modelo-vista-controlador. Finalmente, se detallan todos los lenguajes de programación, herramientas y tecnologías utilizados en el desarrollo de la aplicación web.

2.1. Aplicaciones web

Según Mora (2002), una aplicación web es aquella a la que un usuario, desde un navegador, realiza peticiones remotamente, accediendo a ella a través de Internet (o de una intranet), cuya respuesta se muestra en el mismo navegador.

De acuerdo con Niño (2010), las aplicaciones web son software que se codifica en un lenguaje soportado por los navegadores web en la que se confía la ejecución al navegador. Una página web puede contener elementos que permiten una comunicación activa entre el usuario y la información. Esto permite que el usuario acceda a los datos de modo interactivo, gracias a que la página responderá a cada una de sus acciones.

En la última década, en todo el mundo, se ha incrementado el desarrollo de aplicaciones para el entorno web codificadas en diversos lenguajes de programación, utilizando las distintas alternativas propuestas y tecnologías presentes. Los cambios y las mejoras realizadas en una aplicación web son incorporados de manera casi instantánea (a diferencia de las aplicaciones de escritorio, las cuales deben instalarse y actualizarse en cada computadora de manera individual).

Las aplicaciones web pueden ser desplegadas en dispositivos móviles como teléfonos inteligentes, tabletas, laptops, entre otros. La intención de que sea posible visualizar la aplicación desde un dispositivo móvil es llegar a más personas. Si una aplicación web no es responsiva, puede resultar complejo visualizarla, provocando así que la persona termine por abandonarla.

2.1.1. Características de las aplicaciones web

De acuerdo a Good (2015), algunas de las principales características de las Aplicaciones Web son:

- **Acceso inmediato:** la mayoría de las aplicaciones web no necesitan ser descargadas, instaladas y configuradas. Para su acceso, simplemente se requiere hacer uso de un navegador web y conexión a Internet.
- **Compatibilidad multiplataforma:** las Aplicaciones Web tienen un camino mucho más sencillo para la portabilidad multiplataforma que las aplicaciones de escritorio. Las distintas tecnologías permiten un desarrollo efectivo de programas soportando todos los sistemas operativos principales (Windows, Linux, MacOS).
- **Actualización:** las aplicaciones basadas en web están siempre actualizadas con la última versión sin requerir que el usuario tome acciones proactivas, y

sin necesitar llamar la atención del usuario o interferir con sus hábitos de trabajo con la esperanza de que va a iniciar nuevas descargas y procedimientos de instalación.

- **Facilidad de prueba:** no existen obstáculos para permitir pruebas sencillas y efectivas de herramientas y aplicaciones antes de realizar una operación que afecta la realidad. Por ejemplo, cargar la tarjeta de crédito es una operación que se puede simular para realizar pruebas.
- **Menor requerimiento de memoria:** las aplicaciones web tienen demandas notablemente inferiores de memoria RAM que los programas instalados localmente. Al residir y correr en los servidores del proveedor, esas aplicaciones basadas en web usan en muchos casos la memoria del servidor, dejando más espacio para correr múltiples aplicaciones al mismo tiempo sin afectar el rendimiento.
- **Datos online:** con el desplazamiento de las aplicaciones locales a aquellas basadas en web también los datos que se crean y acceden necesitan experimentar cambios porque van a estar en constante movimiento dentro de la red al alcance de todo el mundo.
- **Múltiples usuarios concurrentes:** las aplicaciones web pueden ser accedidas por miles de usuarios al mismo tiempo. No hay necesidad de compartir recursos cuando múltiples usuarios pueden ver e incluso editar el mismo documento de manera conjunta. En el caso de que llegase a actualizarse la aplicación, todos los usuarios estarán al tanto de esta actualización de manera inmediata.
- **Datos seguros:** A medida que las compañías se hagan cargo del almacenamiento de los datos del usuario, el almacenamiento de datos redundantes, altamente fiables, serán la norma más que la excepción, y los

usuarios van a tener mucho menos riesgo de perder sus datos debido a una falla del equipo.

- Fácil desarrollo: Debido a que las aplicaciones han sido separadas de computadoras locales y sistemas operativos específicos, estas pueden también ser escritas en prácticamente cualquier lenguaje de programación.

Aunque es necesario poseer una conexión a internet, esto es compensado por las características de mencionadas anteriormente, lo que les brinda numerosas ventajas frente a las aplicaciones de escritorio. A continuación se listan las ventajas más importantes de las aplicaciones web.

2.1.2. Ventajas de las aplicaciones web

Algunas de las ventajas de utilizar aplicaciones web, de acuerdo con Rivas (2013) son las siguientes:

- Permite la interactividad al hacer posible la comunicación total, bidireccional y multidireccional; la comunicación es inmediata y se posibilita la interacción, tanto de manera síncrona como asíncrona.
- Proporciona el aprendizaje colaborativo debido a que se puede aprender con otros y de otros a través del intercambio de contenidos, trabajos y artículos.
- Provee libertad de edición y difusión porque cada persona puede expresar sus ideas y compartirlas con el resto de los usuarios a los que les interese la información.
- Un usuario puede ser tanto productor como consumidor de información en todo momento.
- Abre nuevos espacios de comunicación entre docentes, estudiantes, familias, entre otros.

Es por las ventajas mencionadas, así como por la interoperabilidad y el gran número de herramientas tecnológicas para ambientes web, que fueron seleccionadas éste tipo de aplicaciones para el desarrollo del presente trabajo de investigación. Sin embargo, las aplicaciones web también tienen algunas desventajas, que se mencionan a continuación.

2.1.3. Desventajas de las aplicaciones web

De acuerdo con Mendez (2012), las cuatro desventajas principales de las aplicaciones web son: la necesidad de una conexión a internet permanente, la seguridad de los datos, la utilización de recursos y el uso de plugins del navegador.

Sobre la necesidad de una conexión a internet permanente, es una desventaja debido a que la aplicación no está instalada de manera local, se establece una comunicación con el servidor para conectarse a ella.

Como la seguridad no depende del usuario, sino de las configuraciones y precauciones tomadas, tanto en la aplicación como en el servidor, muchos usuarios evitan acceder a las aplicaciones web por la posibilidad de comprometer sus datos.

Al conectarse los usuarios al mismo servidor, los recursos como la Base de Datos y todos los tipos de memoria son compartidos. Al agotarse alguno de los recursos por una gran cantidad de tráfico, se interrumpe el servicio para todos los usuarios.

Finalmente, aunque se usan menos cada día, los diversos plugins necesarios para utilizar ciertas aplicaciones (siendo los más usados Java y Flash), causan que el usuario deba instalarlos y conceder permisos, exponiéndose a posibles vulnerabilidades.

Después de mencionar las características de las aplicaciones web, así como sus ventajas y desventajas, es posible concluir que son una buena alternativa para el desarrollo de la aplicación propuesta. Luján (2002), indica que las aplicaciones web son un tipo especial de aplicaciones cliente/servidor. A continuación se describe dicha arquitectura.

2.2. Arquitectura cliente-servidor

De acuerdo con Barros & Dumas (2006), la arquitectura cliente-servidor es un modelo de aplicación distribuida en el que los clientes solicitan servicios y las tareas se reparten entre los servidores, que proveen dichos servicios. Aunque es mucho más provechoso usar la arquitectura en un sistema distribuido, no es necesario. Los ejemplos más populares de la arquitectura son el correo electrónico y la WWW.

Según Comer & Stevens (2013), el cliente es el dispositivo maestro, pues inicia las solicitudes. Puede conectarse a varios servidores e interactúa con el usuario a través de una GUI (interfaz gráfica de usuario, por sus siglas en inglés). Se mantiene en espera para recibir y procesar la respuesta del servidor. El servidor, por su parte, es el dispositivo esclavo, se encuentra siempre esperando las solicitudes, en un papel pasivo. Al recibir dichas peticiones, las procesa y envía una respuesta. El servidor tiene la capacidad de aceptar conexiones de una gran cantidad de clientes.

En la figura 8, se puede observar la interacción entre el cliente y el servidor, a través del proceso petición-respuesta. La aplicación web, visualizada en el navegador, realiza una petición a un servidor de internet, que envía dicha petición a un servidor web, el cual responde al navegador con la página solicitada por el usuario.

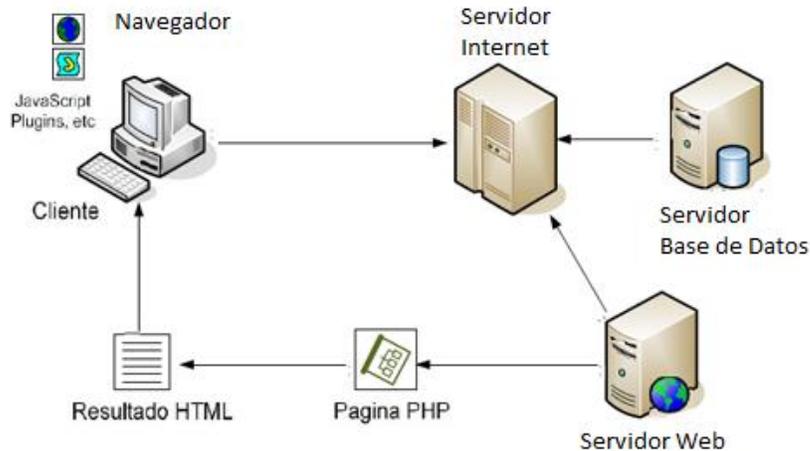


Figura 8. Arquitectura cliente-servidor, proceso de petición-respuesta.

Entre las ventajas de esta arquitectura se pueden contar la centralización del control, es decir, los recursos y la integridad de los datos son controlados por el servidor, para que un programa no autorizado o dañino pueda afectar el sistema. También podemos destacar el fácil mantenimiento, debido a que las funciones y responsabilidades se encuentran distribuidas entre varios ordenadores. Finalmente, de acuerdo con Dustdar & Schreiner (2005), es notable la forma en que se puede escalar por separado el cliente o el servidor, dependiendo de las necesidades particulares.

Según Leal (2012), una de las desventajas más comunes de la arquitectura es la congestión del tráfico, es decir, cuando se envía una gran cantidad de peticiones simultáneas al servidor, es posible que no tenga la capacidad para responderlas todas. También es otro inconveniente la robustez, puesto que si un servidor se cae, las peticiones no pueden ser satisfechas. Finalmente, el coste es más elevado, porque el servidor necesita software y hardware específico para satisfacer a un gran número de clientes.

Habiendo determinado que la aplicación del presente trabajo de investigación utilizará la arquitectura cliente-servidor, es necesario especificar el patrón de software a utilizar. A continuación se describe el patrón Modelo-Vista-Controlador.

2.3. Patrón Modelo-Vista-Controlador

De acuerdo con su creador, Reenskaug (2009), se denomina Modelo-Vista-Controlador a un patrón de arquitectura de software que se basa en la separación de conceptos y reutilización de código. Para lograr la separación de la lógica de negocio, los datos y su representación en la interfaz de usuario, se dispone de los tres componentes antes mencionados. Fue una de las primeras ideas en el campo de interfaces gráficas de usuario (GUI, por sus siglas en inglés), y también uno de los primeros en describir las aplicaciones de software según sus funcionalidades.

En su concepción original, el controlador y la vista eran análogos a la entrada y la salida. Sin embargo, dicha analogía no es válida en la actualidad. Según lo establecido por Rayfield & Leff (2001), se pueden definir los componentes, de manera genérica, como sigue:

Modelo: Se encarga de todo lo referente a los datos e información, el acceso a la misma, consultas, actualizaciones, eliminaciones, implementar privilegios de acceso según las especificaciones de la lógica del negocio. Después de obtener la información solicitada, la envía a la vista para que sea mostrada. Dicha solicitud es recibida por el modelo a través del controlador.

Vista: Toma la información enviada por el modelo, y la representa en un formato legible por el usuario, se encarga de toda la parte visual y de interacción (usabilidad, accesibilidad y demás).

Controlador: Su función principal es manejar los eventos, y enviar peticiones al modelo cuando se solicita alguna información. Hace de intermediario entre la vista y el modelo, manejando también detalles como cambios en la representación (ordenamiento, filtrado, entre otros).

Existen distintas implementaciones del patrón, sin embargo, de acuerdo con Moore, Budd & Benson (2007), en la mayoría el usuario realiza las mismas acciones

generales y se desencadenan las mismas respuestas por parte de la aplicación. Esto es: el usuario interactúa con la interfaz de alguna manera, el controlador recibe el aviso de la acción solicitada por el usuario y gestiona el evento. Posteriormente el controlador accede al modelo, haciendo modificaciones apropiadas a la acción realizada. Al obtener los objetos del modelo, el controlador los envía a la vista, que tiene la tarea de desplegar la interfaz de usuario de manera apropiada. Finalmente, la interfaz espera nuevas interacciones, con lo que se repite el ciclo.

En la figura 9 podemos observar una generalización del patrón de arquitectura de software MVC.

Arquitectura y diseño: Patrón MVC

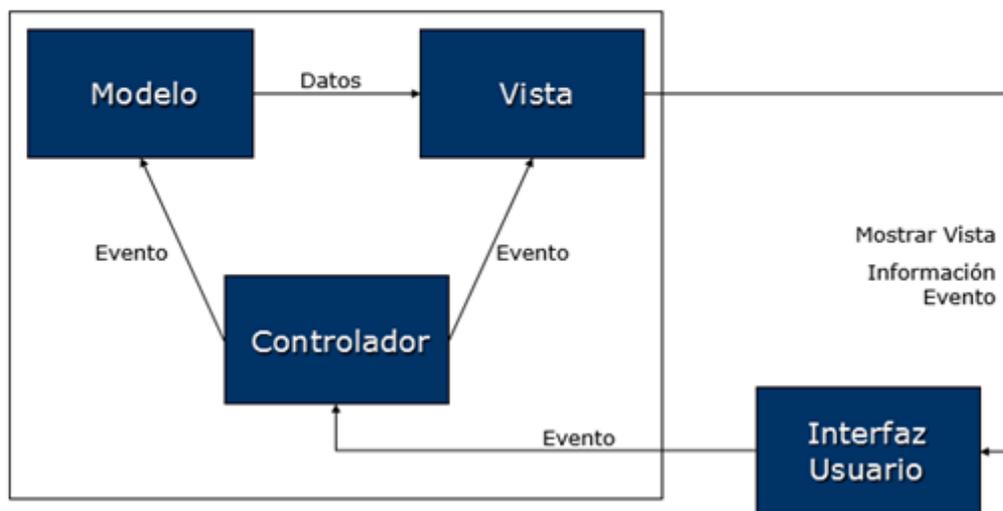


Figura 9. Patrón Modelo-Vista-Controlador.

En los apartados anteriores, se describieron tres de los elementos fundamentales en las cuales se basa el desarrollo del presente trabajo de investigación; se trata de una aplicación web, construida bajo una arquitectura cliente-servidor, haciendo uso del patrón de arquitectura de software Modelo-Vista-

Controlador. A continuación, se describen el conjunto de herramientas tecnológicas utilizadas.

2.4. Herramientas para el desarrollo de aplicaciones web

De acuerdo con el Luján (2002), son muchas las organizaciones que se apoyan en la tecnología para la ejecución de sus procesos. El ámbito tecnológico evoluciona en cortos períodos de tiempo, surgen nuevas herramientas y técnicas, ampliando la cantidad de alternativas a la hora de desarrollar una aplicación web.

Para el desarrollo de aplicaciones web es necesario usar un conjunto de tecnologías y lenguajes de programación tanto del lado del servidor como del cliente. Al tratarse de un desarrollo para la Universidad Central de Venezuela, (que es una organización pública) se investiga a fin de usar herramientas de Código Abierto, como es sugerido en el Decreto N° 3.390, publicado en la Gaceta Oficial N° 38.095 de fecha 28/12/2004 sobre el uso obligatorio del software libre en el país para todas las dependencias públicas de carácter oficial.

La ventaja principal de lo anteriormente expresado se evidencia desde el punto de vista monetario, ya que se eliminan los costos de licencia para el producto. Otra ventaja importante es que se puede disponer del código fuente, y la posibilidad de modificarlo a futuro según las necesidades.

2.4.1. Lenguaje de marcado de Hipertexto

El Lenguaje de Marcado de Hipertexto (HTML por sus siglas en inglés), es el lenguaje utilizado para crear páginas web. Es uno de los componentes principales de la web abierta, de acuerdo con el W3C (2011).

Es un lenguaje muy sencillo que permite describir hipertexto, es decir, texto presentado de forma estructurada, con enlaces que conducen a otros documentos o

fuentes de información relacionadas. Dicho texto se estructura mediante el uso de etiquetas, que le dan un significado especial a una sección determinada.

Según Stallman (2013), HTML5 es la última versión de HTML. Con esta versión se introducen nuevas características en una variedad de áreas. Se trata de una tecnología creada para modernizar la web y el desarrollo, pensando en proporcionar una plataforma que permita desarrollar aplicaciones web con mayor semejanza a las aplicaciones de escritorio, donde su ejecución dentro de un navegador no implique falta de recursos o facilidades para resolver las necesidades reales de los desarrolladores.

En la figura 10 se puede apreciar un ejemplo de código HTML, mostrando la estructura típica de un documento.

```
1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4   <meta charset="UTF-8">
5   <title>¡Hola, mundo!</title>
6 </head>
7 <body>
8   ¡Hola, mundo!
9 </body>
10 </html>
```

Figura 10. Ejemplo de código HTML.

En el presente trabajo de investigación, el lenguaje es utilizado porque es el enlace entre el navegador y el usuario. El navegador toma el código HTML, y, de acuerdo a las etiquetas utilizadas le muestra al usuario una página web con una estructura específica. Sin embargo, no es suficiente mostrar la información de una manera estructurada, también debe ser atractiva visualmente. A continuación, se describen las hojas de estilo en cascada.

2.4.2. Hojas de estilo en cascada

Las hojas de estilo en cascada (CSS, por sus siglas en inglés), surgen, según Meyer (2006), con la finalidad de definir cómo se han de mostrar los elementos de HTML y con qué estilo se han de presentar, separando así la estructura del aspecto visual.

Se pueden definir múltiples hojas de estilo para una misma página web. Con una hoja de estilo se puede cambiar la apariencia de una página web modificando algunas características como la fuente, márgenes y cualquier elemento visual como colores y bordes.

Es el World Wide Web Consortium (W3C) quien se encarga de formular la especificación de las hojas de estilo que servirán de estándar para los navegadores. El Responsive Web Design se hace posible gracias a la introducción de las media queries en las propiedades de los estilos CSS en su versión número 3. Las media queries son una serie de órdenes que se incluyen en la hoja de estilos que indica al documento HTML cómo debe comportarse en diferentes resoluciones de pantalla.

Los estilos CSS se asignan mediante reglas. Las reglas poseen un selector y una declaración. Los selectores cumplen de enlace entre el estilo y el documento. El selector hace referencia a qué elemento se aplicará el estilo. Y las declaraciones son conjuntos pares estilo-valor que definen cuál será el estilo a aplicarse.

La figura 11 muestra un ejemplo de código CSS, con dos reglas y cinco declaraciones, definiendo la fuente, el ancho y la posición del contenedor principal.

```
1  body {  
2      font-family: 'Raleway', sans-serif;  
3  }  
4  
5  .main {  
6      width: 1015px;  
7      position: absolute;  
8      top: 10%;  
9      left: 20%;  
10 }
```

Figura 11. Ejemplo de código CSS.

Durante el desarrollo de la aplicación, las hojas de estilo en cascada fueron utilizadas para determinar el aspecto visual de cada vista. Al tener el contenido y su aspecto, lo siguiente es determinar los modos de interactuar con ese contenido. Para conseguirlo, se utiliza JavaScript, descrito a continuación.

2.4.3. JavaScript

De acuerdo con Flanagan (1998), JavaScript es un lenguaje de programación interpretado que puede ser embebido directamente en código HTML. Permite crear aplicaciones web dinámicas e interactivas ejecutadas por completo dentro de un navegador web.

Según el W3C (2011), los navegadores modernos interpretan el código JavaScript integrado en las páginas web. Para interactuar con una página web se provee al lenguaje de una implementación del Document Object Model (DOM, Modelo de Objetos del Documento en español), que es una Interfaz de Programación de Aplicaciones (API, por sus siglas en inglés) de lenguaje neutro, que permite a los programas y scripts acceder y actualizar dinámicamente el contenido, estructura y estilo de los documentos web. El documento puede ser procesado y los resultados del procesamiento se pueden incorporar de nuevo en la página presentada.

En la figura 12 se muestra un ejemplo de código JavaScript, una función que filtra los elementos de un arreglo si el índice es negativo, y la inicialización de tres variables.

```
48     Array.prototype.diff = function(a){
49         return this.filter(function(i){
50             return a.indexOf(i) < 0;
51         });
52     };
53
54     var equipos = [];
55     var equipos_disponibles = [];
56     var valor_equipo_anterior = -1;
```

Figura 12. Ejemplo de código JavaScript.

En la aplicación, JavaScript es utilizado para interactuar con los elementos de la página y hacerlos dinámicos.

2.4.4. jQuery

De acuerdo con su creador, Resig (2008), jQuery es una biblioteca de JavaScript, cuya función principal es reducir la dificultad inherente a escribir código directamente en el lenguaje de programación.

La biblioteca, al igual que otras, ofrece funcionalidades que requerirían mucho más código y tiempo de desarrollo para llevar a cabo. De acuerdo con Google Analytics (2016), es la librería de JavaScript más utilizada. Por sus múltiples características y funciones, además de su compatibilidad, ha sido utilizado por grandes empresas, y prácticamente cualquier desarrollo que involucre JavaScript, incluye también jQuery.

Según Resig & jQuery Foundation (2009), aunque la librería brinda una gran cantidad de funciones mediante su API, consiste en un sólo archivo JavaScript. El atractivo principal de la biblioteca es que logra manipular el contenido de una página sin recargarla, haciendo peticiones AJAX y manipulando el árbol DOM. Sin embargo, esas son sólo dos de sus características. Con la biblioteca tenemos soporte para CSS, del 1 al 3, manejo de eventos, creación de efectos y animaciones, animaciones personalizadas, utilidades variadas como información del navegador, operar con

vectores (arreglos dinámicos), funciones de recorrido, búsqueda y cualquier otra rutina considerada común.

Para el trabajo de investigación, jQuery se utilizó para modificar el árbol DOM y comunicar el cliente con el servidor. Mediante AJAX, se envía una petición al servidor, que responde con un arreglo JSON. A continuación se describen ambas tecnologías.

2.4.5. JavaScript Object Notation

De acuerdo con ECMA International (2013), la Notación de Objetos de JavaScript, o por sus siglas en inglés, JSON, es un formato de intercambio de datos ligero. Para los humanos es de fácil lectura y escritura y para las máquinas es de sencillo análisis sintáctico.

JSON es un formato de texto completamente independiente del lenguaje, pero usa convenciones que son comunes a la mayoría de los lenguajes imperativos. Está basado en dos estructuras: una colección de pares nombre/valor que dependiendo del lenguaje puede ser un *objeto*, *registro*, *diccionario*, *arreglo asociativo*, *tabla hash* o cualquier nombre que se desee darle y una estructura en forma de lista ordenada de valores llamada *arreglo*, *vector*, *lista* o *secuencia*, dependiendo también del lenguaje. Las estructuras antes mencionadas son universales, y están presentes en casi todos los lenguajes de programación modernos.

En la figura 13 se ilustra la construcción de un objeto JSON, un conjunto de cero o más pares string-valor:

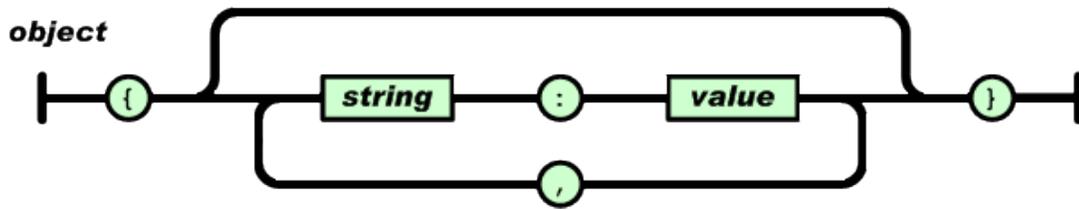


Figura 13. Construcción de un objeto JSON.

Fuente: Estándar ECMA-404. ECMA International (2013).

Un arreglo es una colección ordenada de valores, cuya construcción se puede apreciar en la figura 14:

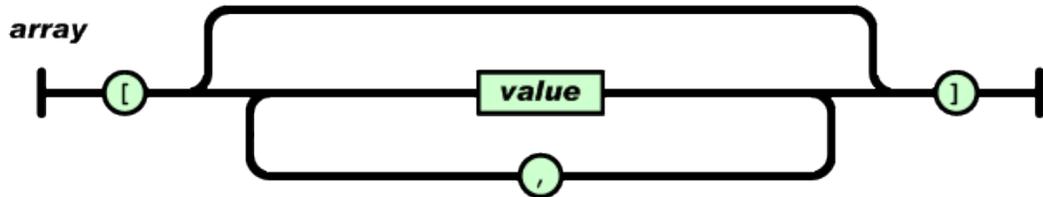


Figura 14. Construcción de un arreglo JSON.

Fuente: Estándar ECMA-404. ECMA International (2013).

Un valor puede ser una cadena de caracteres (string) en comillas dobles, un número (number), un booleano (true o false), un objeto (object), un arreglo (array) o un valor nulo (null). La figura 15 muestra los valores válidos en JSON.

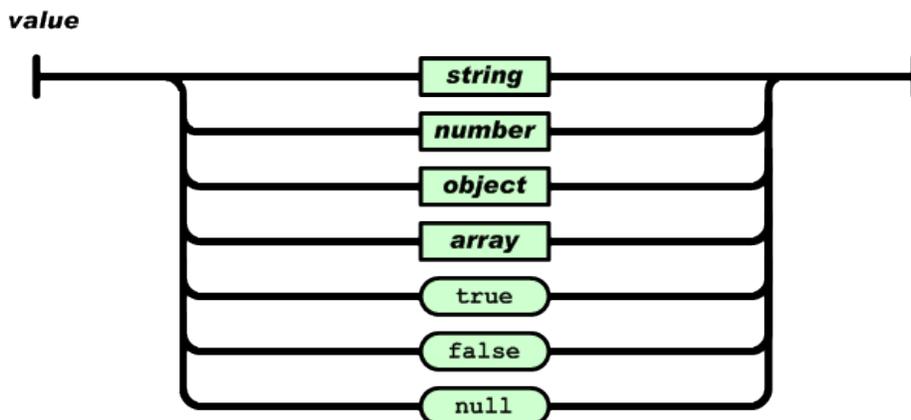


Figura 15. Tipos de valores válidos para un arreglo u objeto JSON.

Fuente: Estándar ECMA-404. ECMA International (2013).

En la aplicación web del presente trabajo de investigación, JSON es utilizado para enviar datos de forma estándar. Tanto el cliente como el servidor pueden crear y leer los arreglos y objetos fácilmente, aunque no comparten el mismo lenguaje de programación. Para la comunicación entre ambas partes, se utiliza AJAX, que se describe a continuación.

2.4.6. AJAX

De acuerdo con Garret (2005), el JavaScript asíncrono y XML (AJAX, por sus siglas en inglés), es una técnica de desarrollo para crear aplicaciones de internet enriquecidas (RIA, por sus siglas en inglés). Dichas aplicaciones se ejecutan en el cliente, es decir, el navegador web. Mientras que se mantiene una comunicación asíncrona con el servidor, en segundo plano.

Debido a que AJAX se encuentra basado en estándares abiertos como JavaScript y el DOM, es una técnica válida para múltiples plataformas, navegadores y sistemas operativos. Como la tecnología es asíncrona, mejora la velocidad, interactividad, y usabilidad de las aplicaciones que la usan. Aunque se encuentran en el nombre, no es necesario realizar la petición de contenido en JavaScript (es lo más común), así como tampoco es necesario dar formato a ese contenido en XML (de hecho, JSON ha tenido un gran auge en la actualidad, por su facilidad de serialización). “Llamamos AJAX a cualquier grupo de tecnologías que hacen una petición asíncrona y devuelven contenido en algún formato, sin la necesidad de recargar la página para mostrar ese contenido” (Ullman, 2007).

Según Hopmann (2010), aunque las ventajas del uso de ésta tecnología son evidentes, posee algunas desventajas. Entre ellas se pueden contar que las páginas creadas dinámicamente no son registradas automáticamente en el historial del navegador, puede aumentar significativamente la carga en el servidor, al ser saturado con peticiones Finalmente, también es posible que se reduzca la

accesibilidad, debido a que la tecnología no es compatible con todos los navegadores que poseen ayudas especiales para personas invidentes.

Usar AJAX en la aplicación web permite la comunicación entre las tecnologías del lado del cliente, con las tecnologías del lado del servidor. Entre ellas, PHP, que se describe a continuación.

2.4.7. PHP Hypertext Pre-processor

PHP (por sus siglas en inglés), es un lenguaje de programación de propósito general desarrollado originalmente por Rasmus Lerdorf en 1995.

De acuerdo Lerdorf (2007), PHP es un lenguaje de programación del lado del servidor, diseñado en su origen para el desarrollo web, específicamente para manejar contenido dinámico. Fue uno de los primeros lenguajes de programación del lado del servidor que se incrustan directamente en el documento HTML y el código es interpretado por un servidor web con un módulo que procesa el PHP y genera la página web resultante. El lenguaje puede ser ejecutado en la mayoría de los sistemas operativos, ya sean basados en Unix o de Microsoft Windows.

Según Trachtenberg (2004), entre las características principales de PHP se puede contar que es un lenguaje con una curva de aprendizaje muy poco pronunciada, facilitando así el desarrollo de aplicaciones web y el acceso de éstas a bases de datos. Es también un lenguaje interpretado, en consecuencia es de tipado dinámico y débil (no es necesario declarar el tipo de las variables, el tipo de las mismas puede cambiar, y las funciones pueden utilizar tipos compatibles). El código fuente PHP no es visible por el navegador, debido a que el código es procesado y ejecutado por el servidor, retornando únicamente HTML. El núcleo del lenguaje puede ser expandido utilizando módulos. Su documentación es muy amplia, cada función está explicada y ejemplificada. Es un lenguaje de programación libre, y permite paradigmas como la programación orientada a objetos.

A pesar de sus numerosas ventajas, como todo lenguaje, posee algunos inconvenientes. Entre ellos “el más importante es que puede ser considerablemente más lento que otras alternativas, debido a que debe interpretarse en cada ejecución, no es compilado (aunque podemos mitigar estos inconvenientes con técnicas de caché, tanto de memoria como de archivos)” (Scriven, 2010).

PHP fue utilizado en el presente trabajo de investigación como el lenguaje del lado del servidor. Sin embargo, los marcos de trabajo facilitan el desarrollo de aplicaciones web, proporcionando herramientas que tendrían que ser escritas desde cero de no utilizar alguno. A continuación se describen las características principales del marco de trabajo CodeIgniter.

2.4.8 Marco de trabajo CodeIgniter

De acuerdo con sus creadores, Ellislab (2006), CodeIgniter es un marco de desarrollo de software basado en PHP creado para desarrolladores que requieren de un kit de herramientas completo, pero a la vez de uso simple y elegante para la creación de aplicaciones web con amplias funcionalidades. Para el desarrollo de la aplicación web del presente trabajo de investigación, fue utilizado en su versión 3.

Según los actuales encargados de mantener el marco de trabajo, British Columbia Institute of Technology (2015), CodeIgniter es ideal como marco de desarrollo ligero ya que su estructura es simple, provee librerías para funciones básicas y tareas comunes como manejo de formularios y sesiones, también permite tener un rendimiento excepcional al mismo tiempo que provee amplia compatibilidad con varias versiones de PHP y configuraciones de hospedaje para sitios web.

2.4.8.1. Características de CodeIgniter

De acuerdo con Ellislab (2006), algunos de los puntos más interesantes sobre este marco de desarrollo, sobre todo en comparación con otros productos similares, son los siguientes:

- Versatilidad: quizás la característica principal de CodeIgniter, en comparación con otros frameworks PHP. CodeIgniter es capaz de trabajar la mayoría de los entornos o servidores, incluso en sistemas de alojamiento compartido, donde sólo tenemos un acceso por FTP para enviar los archivos al servidor y donde no tenemos acceso a su configuración.
- Compatibilidad: el marco de trabajo es compatible con la versión PHP 4, lo que hace que se pueda utilizar en cualquier servidor, incluso en algunos antiguos. Funciona correctamente también en PHP 5 y 7.
- Actualizado: desde la versión 2 de CodeIgniter ya solo es compatible con la versión 5 de PHP. Para los que todavía usen PHP 4 pueden descargar una versión antigua del framework, como CodeIgniter V 1.7, que todavía era compatible.
- Facilidad de instalación: no es necesario más que una cuenta de FTP para subir CodeIgniter al servidor, y su configuración se realiza con apenas la edición de un archivo, donde debemos escribir cosas como el acceso a la base de datos. Durante la configuración no necesitamos la línea de comandos, que no suelen estar disponibles en todos los alojamientos.
- Flexibilidad: es bastante menos rígido que otros frameworks. Define una manera de trabajar específica, pero en muchos de los casos podemos seguirla, o no. Algunos módulos como el uso de plantillas son totalmente opcionales. Esto ayuda muchas veces también a que la curva de aprendizaje sea más sencilla al principio.

- **Ligereza:** el núcleo de CodeIgniter es bastante ligero, lo que permite que el servidor no se sobrecargue interpretando o ejecutando grandes porciones de código. La mayoría de los módulos o clases que ofrece se pueden cargar de manera opcional, sólo cuando se van a utilizar realmente.
- **Documentación tutorializada:** la documentación es fácil de seguir y de asimilar, porque está escrita en modo de tutorial. Esto facilita mucho la referencia rápida, cuando ya se tiene conocimiento del marco de desarrollo se puede consultar sobre una función o un método concreto.

2.4.8.2. Flujo de aplicación

Según el British Columbia Institute of Technology (2015), en CodeIgniter existe un procedimiento para atender una solicitud de página del cliente. Este proceso se realiza internamente por el propio CodeIgniter y de manera transparente para el usuario. Durante el proceso participan varios módulos como el enrutamiento de la solicitud, la caché interna, etc.

En la figura 16, pueden apreciarse los módulos que participan en la ejecución de una aplicación construida con este marco de desarrollo:



Figura 16. Flujo de una aplicación en CodeIgniter.

Fuente: http://www.codeigniter.com/user_guide/appflow.html. Ellislab (2016).

2.4.9. Lenguaje de consulta estructurado

Según Beaulieu (2009), el lenguaje de consulta estructurado (SQL, por sus siglas en inglés), es un lenguaje utilizado para realizar distintas operaciones sobre bases de datos relacionales. Está basado en la teoría de conjuntos y mediante el manejo del álgebra y el cálculo relacional, permite manipular la información de forma sencilla mediante consultas.

De acuerdo a la propuesta de Codd (1970), surge el modelo relacional, y con éste, un lenguaje de acceso a los datos. Es basado en éstas ideas que IBM define su lenguaje SEQUEL, implementado en System R, un Sistema Manejador de Bases de datos, también de IBM. Dicho lenguaje pasaría a ser el predecesor de lo que hoy se conoce como SQL.

Debido a su amplia difusión e implementación en los Sistemas Manejadores de Bases de Datos (SMBD), el Instituto Nacional de Estándares Internacionales (ANSI, por sus siglas en inglés) decide estandarizar el lenguaje en 1986. Al año siguiente dicho estándar sería adoptado también por la Organización Internacional por la Estandarización (ISO, por sus siglas en inglés) y ha sufrido numerosas modificaciones hasta el actual SQL:2008.

“Entre las características más notables del lenguaje encontramos que es declarativo y de alto nivel, no procedimental. Está orientado al manejo de conjuntos de registros, en lugar de registros individuales. Posee un LDD (Lenguaje de Definición de Datos) y un LMD (Lenguaje de Manipulación de Datos). Posee comandos para garantizar la integridad, así como autorización, definición de vistas y control de transacciones” (Kauffman, 1993).

Para consultar los datos almacenados por la aplicación web en el presente trabajo de investigación, se utilizó SQL, y para gestionar dichos datos, el SMBD MySQL, que se describe a continuación.

2.4.10. Sistema manejador de bases de datos MySQL

De acuerdo con Oracle Corporation (2016), MySQL es un sistema de administración de bases de datos relacional que es usado para agregar, eliminar, modificar y consultar los datos del repositorio.

Se encarga de almacenar y distribuir gran cantidad de datos para satisfacer las necesidades de cualquier organización. “MySQL incluye todos los elementos necesarios para instalar el programa, preparar diferentes niveles de acceso de un usuario, administrar el sistema y proteger y hacer volcados de datos” (Gilfillan, 2003).

Según Oracle Corporation (2016), una de las características de MySQL es que su arquitectura se encuentra basada en un esquema cliente-servidor, es decir, se puede utilizar del lado del cliente o del lado del servidor. Existe una base de datos del servidor (MySQL) y varios clientes (aplicaciones de programas), los cuales se comunican con el servidor. Los clientes pueden ejecutarse en la misma computadora.

2.4.10.1. Ventajas y desventajas de MySQL

De acuerdo con Sun Microsystems (2008), entre las principales ventajas de MySQL se encuentran:

- Compatibilidad con el lenguaje SQL.
- Con el manejador MySQL se pueden crear vistas, procedimientos, disparadores, definir restricciones de integridad.
- Posee la alternativa de replicación de los datos para incrementar la protección de los mismos al ocurrir una falla del sistema y para proveer mayor eficiencia y rapidez al momento de realizar consultas sobre la base de datos.
- Manejo de transacciones. “En el contexto de un sistema de bases de datos, una transacción significa la ejecución de varias operaciones de bases de datos como un bloque. El sistema de base de datos asegura que, o bien todas

las operaciones se ejecutan correctamente o ninguna de ellas se ejecuta. Las transacciones permiten al programador interrumpir comandos ejecutados previamente, lo cual simplifica muchas veces el proceso de programación” (Kofler, 2005).

- Se pueden utilizar diversos lenguajes de programación para desarrollar aplicaciones de MySQL, tales como C, C++, Java, Ruby, Perl, PHP, Python, Tcl, etc.
- Es multiplataforma, es decir, puede ejecutarse en diversos sistemas operativos, tales como Linux, Microsoft Windows, Apple Macintosh OS X y algunas variantes de Unix.

Y entre las desventajas más importantes tenemos:

- Cuando se utilizan Tablas con formato tipo MyISAM, al momento de consultar o modificar la información de la base de datos, toda la Tabla donde se encuentra dicha información es bloqueada.
- Al utilizar Tablas MyISAM, MySQL no puede llevar a cabo copias de seguridad “en caliente” (las cuales son copias de la información durante la operación sin bloquear las Tablas).
- Muchos manejadores de bases de datos permiten definir tipos de datos por el usuario. MySQL no provee tal funcionalidad.

Es por las ventajas mencionadas anteriormente, además de la sencilla integración de MySQL con el marco de trabajo CodeIgniter, que se eligió este SMD para el desarrollo de la aplicación web en el presente trabajo de investigación.

Como se mencionó anteriormente, al tratarse de un desarrollo para la Universidad Central de Venezuela, es obligatorio el uso de herramientas de software libre, así como una licencia no comercial, la cual permita que la aplicación pueda ser reutilizada y modificada para fines no comerciales dentro de la universidad. Una licencia que cumple con lo expresado es la Creative Commons, que se describe a continuación.

2.4.11. Licencia Creative Commons

Según Creative Commons (2011), son licencias inspiradas en la licencia GPL (General Public License) que permiten copiar y distribuir las obras intelectuales siempre y cuando se respeten las condiciones escogidas por los autores.

De acuerdo con el Repositorio Nacional de Aplicaciones de Venezuela (2016), desde el 29 de noviembre de 2013 se cuenta con las licencias Creative Commons, adaptadas a las leyes venezolanas en su versión 3.0. Se describen las seis licencias Creative Commons para Venezuela:

- **Atribución CC BY:** esta licencia permite a otros distribuir, mezclar, ajustar y construir sobre su trabajo, incluso comercialmente siempre y cuando se de crédito a la creación original.
- **Atribución – Compartir Igual CC BY-SA:** esta licencia permite a otros mezclar, modificar, y construir sobre su trabajo, incluso para fines comerciales, siempre y cuando se de crédito y la licencia de sus nuevas creaciones en los términos idénticos.
- **Atribución – Sin Obras Derivadas CC BY-ND:** esta licencia permite la redistribución, comercial y no comercial, siempre y cuando se pase sin cambios y en su totalidad, con crédito al creador.
- **Atribución – No Comercial CC BY-NC:** esta licencia permite a otros, modificar y ampliar su trabajo para fines no comerciales. No es necesario que la licencia de obras derivadas se genere en las mismas condiciones.
- **Atribución – No Comercial – Compartir Igual CC BY-NC-SA:** esta licencia permite a otros, modificar y ampliar su trabajo para fines no comerciales, siempre y cuando se dé crédito y la licencia de sus nuevas creaciones sea en los términos idénticos.
- **Atribución – No Comercial – Sin Obras Derivadas CC BY-NC-N:** esta licencia es la más restrictiva de las seis licencias principales, que sólo permite que

otros puedan descargar sus obras y compartirlas con los demás, siempre y cuando se de crédito, pero no se pueden cambiar en cualquier forma o utilizarlos con fines comerciales.

Para la aplicación web se decidió utilizar la licencia Creative Commons Venezuela. Atribución – No Comercial CC BY-NC, permitiendo a los miembros del CERI realizar modificaciones para mejoras futuras.

Capítulo 3: Marco Aplicativo

En este capítulo se exponen y describen las iteraciones llevadas a cabo durante el desarrollo del Trabajo Especial de Grado (TEG), especificando las etapas contempladas en la metodología de desarrollo seleccionado AgilUS, para el desarrollo del sistema y detallando las actividades realizadas, los artefactos generados y las entregas efectuadas.

En el caso del sistema producto de este trabajo, AgilUS fue escogida debido a su enfoque en la creación de software de calidad de acuerdo a las necesidades de los usuarios y la alta interactividad entre usuarios y aplicación sino también a sus capacidades y visión de un software fácil de usar a fin de facilitar la transición a un sistema que, con todas las ventajas de una aplicación web, incluyera las funcionalidades requeridas y gozara de un nivel de usabilidad aún mayor.

Cada fase y su aplicación en el desarrollo del sistema serán explicadas con detalle a continuación.

3.1 Fase I: Requisitos

En esta etapa se realizó el análisis global del problema a solucionar, se generaron perfiles de usuario, y se definieron la lista de requerimientos a desarrollar. Las técnicas utilizadas en esta etapa fueron: tormenta de ideas, entrevistas, perfiles de usuario y requerimientos funcionales y no funcionales los cuales serán descritos a continuación.

3.1.1 Tormenta de ideas

Se realizó una reunión con los miembros del personal del CERI para que aportaran sus ideas a partir de una explicación sobre el sistema que se desea construir para elaborar otras nuevas. Se registraron las ideas que se observan en el mapa mental de la figura 17.

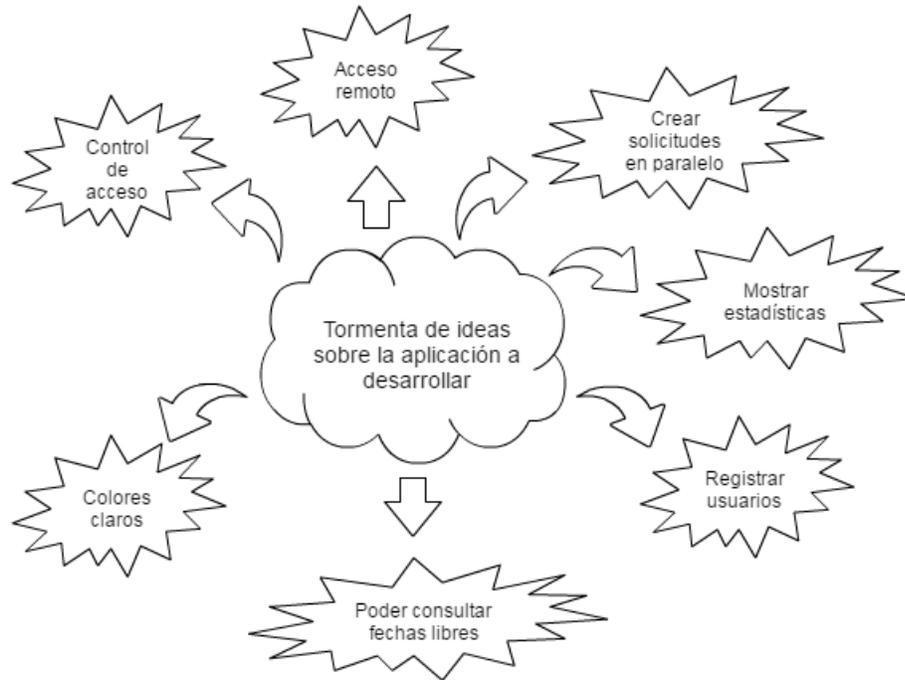


Figura 17. Mapa mental de tormenta de ideas

Varias de las ideas que pudieron recogerse de la tormenta fueron relacionadas con las funcionalidades de la aplicación. Para desarrollarlas se llevaron a cabo entrevistas utilizando las palabras claves en las ideas para obtener más información.

3.1.2 Entrevistas

Se llevó a cabo una (1) entrevista con tres (3) empleados que conforman el personal del CERI encargados de las reservas y préstamos para orientar el curso que debía tomar el desarrollo del sistema.

Se realizaron preguntas relacionadas con los estados de tiempo pasado y presente y preguntas sobre los deseos y expectativas a futuro del manejo de los préstamos de equipos, espacios y servicios en el CERI. Los tipos de preguntas eran cerradas.

- ¿De qué forma se ha llevado el registro de los préstamos que han realizado de espacios, equipos y servicios en el CERI?

Ángel - “En un principio, cuando el CERI empezó a ocuparse de las reservas, se tomaba nota a mano (con papel y lápiz) de los espacios y equipos que se prestaban y luego se anotaba cuando estos eran devueltos. Los servicios como las asesorías y ayudas técnicas no eran registrados y quien estuviera disponible era quien atendía el requerimiento de los solicitantes. Hasta hace unos pocos años fue que se pudo llevar un mejor registro utilizando una hoja de cálculo en Excel”.

- ¿Qué sistema manejan actualmente para llevar el registro de las reservas de espacios, equipos y servicios del CERI?

Anniel - “Actualmente utilizamos un archivo en Microsoft Excel con varias hojas de cálculo para llevar por separado los préstamos de los equipos, espacios y servicios; tomar nota de los usos más frecuentes que le dan a los espacios, los servicios que solicitan los usuarios y llevar las estadísticas de que usuarios piden cuales recursos y cuáles son los recursos más solicitados y utilizados frecuentemente”.

- ¿Qué funcionalidades que no tienen en este momento les gustaría tener en un nuevo sistema para gestionar el préstamo de estos recursos y servicios?

Efrain - “Nos gustaría el poder realizar las reservas de forma remota, tener una base de datos segura que no fuese un solo archivo, poder guardar los datos personales de los usuarios con seguridad para poder contactarlos luego. También nos gustaría que las estadísticas se generen de forma automática y solo debamos consultar el mes y el año que se quiera ver”.

- ¿Cuál de las funcionalidades antes mencionadas considera la más importante que debe tener el sistema?

Efrain - “Las estadísticas en primer lugar. El personal debe elaborar reportes sobre el uso de los recursos y servicios, y en esos reportes deben aparecer los datos de las estadísticas antes mencionadas”.

La tormenta de ideas en conjunto con las respuestas obtenidas de la entrevista proporcionaron las bases para definir los requerimientos funcionales y no funcionales de la fase I.

3.1.3 Perfiles de usuario

Tanto de la tormenta de ideas como de la entrevista con el personal del CERI se determinó que únicamente existen tres perfiles de usuarios y estos se clasifican de acuerdo a sus privilegios. Los involucrados en los procesos de gestión de las reservas de espacios, equipos, servicios y usuarios son los miembros del personal del CERI quienes entran en la categoría de Administrador. Los usuarios que posean cuenta en el sistema se clasifican como Usuarios Registrados, y finalmente aquellos que accedan al sistema sin poseer cuenta en éste se les denomina Usuarios Anónimos.

A continuación, se describen cada uno de los roles:

- Administrador: Este tipo de usuario posee destrezas desarrolladas en el uso del computador, con un nivel intermedio en el uso de la aplicación web ya existente. Es el encargado de manejar los procesos de las reservas y estadísticas, y es un miembro del personal del CERI.

Responsabilidades:

- Gestionar la creación, modificación, eliminación y finalización de las solicitudes de reserva de los usuarios.
 - Gestionar la creación y eliminación de los usuarios, así como la modificación parcial o total de sus datos personales.
 - Otorgar o remover privilegios de administrador a un usuario.
 - Agregar o desincorporar espacios y equipos al sistema.
 - Agregar, modificar o eliminar usos para espacios y equipos.
 - Agregar, modificar o eliminar servicios.
 - Consultar las estadísticas de uso mensual de los equipos, espacios y servicios.
 - Solicitar la recuperación de su contraseña de usuario en caso de no recordarla.
- Usuario Registrado: Este tipo de usuario posee destrezas en el uso del computador, con un conocimiento general de la aplicación web y su uso. Se encuentra registrado en el sistema por lo que puede consultar disponibilidad de espacios y equipos para una fecha y hora determinada.

Responsabilidades:

- Consultar la disponibilidad de espacios y/o equipos en una fecha y hora determinada.
- Consultar la disponibilidad de un servicio en una fecha y hora determinada.

- Modificar los datos personales de su cuenta de usuario en el sistema.
 - Solicitar la recuperación de su contraseña de usuario en caso de no recordarla.
- Usuario Anónimo: Es cualquier usuario no registrado en el sistema, posee las destrezas y conocimientos mínimos necesarios para el manejo de un computador y un navegador de internet.

Responsabilidades:

- Consultar la disponibilidad de espacios y/o equipos en una fecha y hora determinada.
- Crear una cuenta de usuario en el sistema.

3.1.4 Requerimientos funcionales y no funcionales

El resultado final de la realización de la tormenta de ideas, las entrevistas y los perfiles de usuario fue la determinación de los requerimientos puntuales considerados al momento del desarrollo de las funcionalidades de la aplicación web. A continuación, se detallan cada uno de ellos:

- Requerimientos funcionales:
 - Manejo de operaciones de creación, consulta, actualización y eliminación sobre todos los recursos del sistema: se debe proveer una interfaz de usuario que permite hacer uso de las operaciones CRUD (Create, Read, Update, Delete, por sus siglas en inglés). La operación de consulta (Read) será la única disponible para todo usuario que no sea administrador.
 - Creación, edición, eliminación y finalización de reservas: se requiere una funcionalidad que permita la creación, edición, eliminación y finalización de las reservas. A través de esta funcionalidad, los administradores pueden crear y modificar una reserva con los espacios, equipos y servicios para una fecha y

hora determinada, así como eliminarla en caso de que el usuario cancele su solicitud. Finalmente, esta funcionalidad permitirá dar por finalizada la reserva una vez que los espacios, equipos y servicios solicitados hayan sido utilizados y estos se liberen para que otros usuarios puedan solicitar una reserva con ellos.

- Creación, edición y eliminación de usuarios: esta funcionalidad es esencial ya que es la manera en que las reservas de los usuarios pueden ser gestionadas individualmente. Los administradores podrán crear un usuario con sus datos personales, modificarlos posteriormente o eliminarlos si se requiere.
- Consulta de disponibilidad online: esta funcionalidad permite a los usuarios consultar la disponibilidad de equipos y espacios en una fecha y hora determinada.
- Edición de los datos personales: esta funcionalidad permite a los usuarios modificar sus datos personales en sus cuentas de usuario.
- Consulta de las estadísticas de uso: los administradores podrán consultar por mes y año las estadísticas de uso de los equipos y espacios del CERI de acuerdo a las solicitudes que se hayan hecho en el sistema.

- Requerimientos no funcionales:

- Usabilidad y navegabilidad. el sistema debe ser de fácil uso para cualquier usuario, ofreciendo funcionalidades que requieran un bajo conocimiento cognitivo. La solución propuesta debe tener interfaces graficas de administración y de operación en idioma español y en ambiente web, para permitir su utilización a través de cualquiera de los exploradores o navegadores de internet.
- Portabilidad. el desarrollo bajo un enfoque web garantiza que las funcionalidades puedan ser accedidas desde diferentes plataformas, ya que solo se requiere de la disposición de un navegador web o browser independientemente del sistema operativo.

- Disponibilidad. al tratarse de una aplicación web, la disponibilidad se encuentra implícita; las funcionalidades deben estar disponibles y ser accesibles desde cualquier ubicación geográfica con conexión a internet sin restricciones de horarios.
- Eficiencia. Es tomada en cuenta como la capacidad de que el sistema cumpla con sus funcionalidades en un tiempo aceptable para el usuario. El sistema debe ser capaz de responder a las peticiones de los usuarios en un corto tiempo ya que de lo contrario no será satisfactorio y no será utilizado por ellos.
- Tolerancia a fallas. En el momento que ocurra una falla, ya sea intencional o no, el sistema debe ser capaz de responder al usuario y volver al estado que se encontraba al momento anterior de ocurrir la falla.
- Escalabilidad. las funcionalidades son implementadas previendo futuras mejoras sobre las mismas, de manera tal que pueden ser integradas con un nuevo código sin complicaciones. El desarrollo bajo el framework de aplicaciones CodeIgniter respalda el cumplimiento de este requerimiento, ya que sigue el paradigma de la arquitectura MVC explicada anteriormente.

3.2 Fase II: Análisis

En esta etapa se realiza un estudio detallado de los elementos estéticos e interactivos del sistema a desarrollar, empleando prototipos en papel, guías de estilo, modelos de casos de uso y modelos de objetos del dominio. A continuación, se presentarán cada uno de estos artefactos y actividades.

3.2.1 Prototipos en papel

En los primeros encuentros con el personal del CERI, se planteó un primer bosquejo donde una barra lateral fuese el contenedor de las funciones principales de la aplicación, una barra superior donde los logos de la UCV y del CERI estuviesen presentes junto con el nombre de la aplicación, y un área central donde se mostrasen los formularios para la recolección de datos y las listas con los recursos disponibles como puede apreciarse en la figura 18.

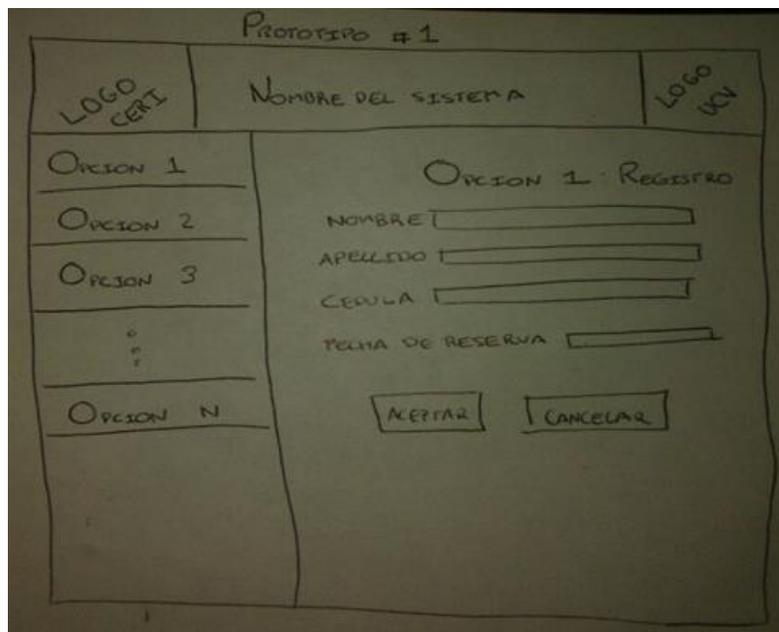


Figura 18. Prototipo en papel con un formulario.

Posteriormente, en la figura 19 puede observarse el prototipo en papel de una interfaz donde se muestra un listado general de datos.

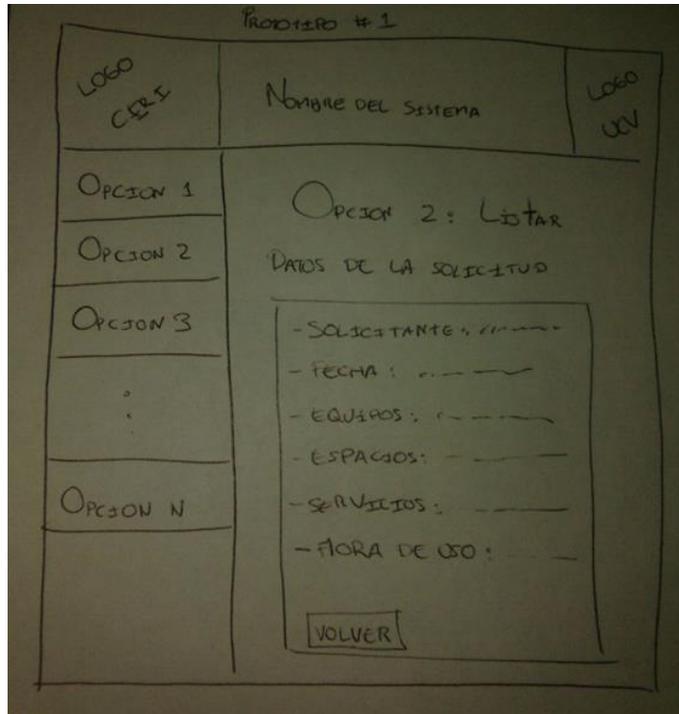


Figura 19. Prototipo en papel con la vista de un listado.

Luego de que fuese establecido como marco de referencia un primer prototipo en papel, se acordó el uso de software en línea Cacao para la creación de los diagramas de los posteriores prototipos, de forma que estos pudiesen enviarse por correo electrónico al personal del CERI para su consideración y evaluación. Los casos de uso, guía de estilos, modelos de objetos del dominio y patrones de interacción fueron también elaborados con el uso de esta herramienta.

Los siguientes prototipos fueron elaborados con la herramienta antes mencionada. En el segundo prototipo se modificó la posición del logo de la UCV y se decidió añadir un pie de página con la dirección del CERI y teléfonos de contacto, como se puede apreciar en la figura 20.

Logo CERl

Nombre del sistema

Menú

- Opción 1
- Opción 2
- Opción 3
- Opción 4
- Opción 5
- Opción 6

OPCION 1

Nombre

Apellido

Cedula

Fecha de la reserva

Logo UCV

Dirección y telefonos del CERl

Figura 20. Prototipo con la vista de un formulario de registro.

En el tercer prototipo, se diseñaron y definieron vistas generales basadas en los tipos de usuarios del sistema. En la figura 21 se ilustra la interfaz general de la aplicación para un usuario anónimo.

Logo CERl

Nombre del sistema

Inicio

Nombre de usuario

Contraseña

¿No tiene cuenta? [Regístrate aquí](#)

Logo UCV

Dirección y telefonos del CERl

Figura 21. Prototipo con la vista general de un usuario anónimo.

En la figura 22 se puede visualizar la interfaz general que ve un usuario registrado.

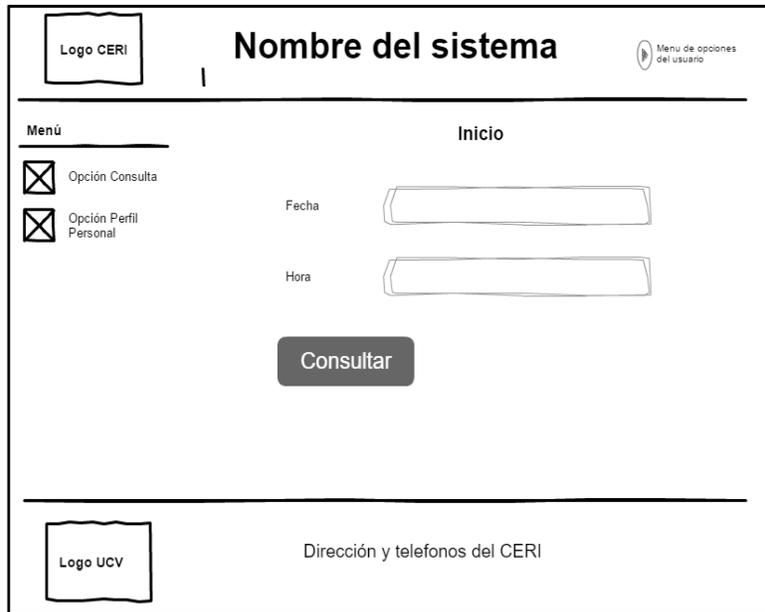


Figura 22. Prototipo con la vista general de un usuario registrado.

En la figura 23, se puede observar la interfaz general de un administrador.

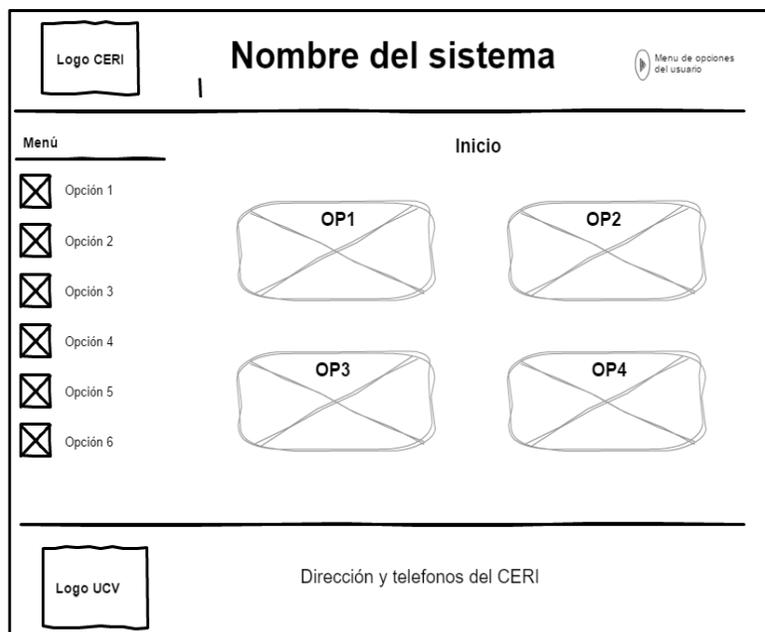


Figura 23. Prototipo con la vista general de un administrador.

Como puede verse, se muestran dos primeros prototipos en papel con los cuales se plantea establecer las bases del diseño de la aplicación. Luego de una revisión de estos es que logra obtenerse un tercer y último prototipo que se utilizará de base para la distribución de los elementos y funcionalidades de la aplicación. Obtenidos los resultados finales de esta actividad, se procede a presentar las guías de estilos por las cuales se regirá la aplicación, para mantener la consistencia a lo largo de todas las interfaces de la aplicación.

3.2.2 Guía de estilos

La guía de estilo es un documento en el cual se toman las características relacionadas con el aspecto de la interfaz de usuario. A continuación, se presenta en la tabla 1 la guía de estilos referente al portal inicial con el panel de control.

Tabla 1. Guía de estilos de la aplicación web

LOGOS		
 <p>Centro de Recursos Instruccionales</p>	 <p>Universidad Central de Venezuela</p>	 <p>Facultad de Humanidades y Educación</p>
COLORES		
 <p>#185983</p>	 <p>#D9D9D9</p>	 <p>#1AA8E1</p>
TIPOGRAFIA		<p>Titulos: Helvetica 14 px Contenido: Arial 12 px</p>

Tabla 2. Continuación de guía de estilos de la aplicación web

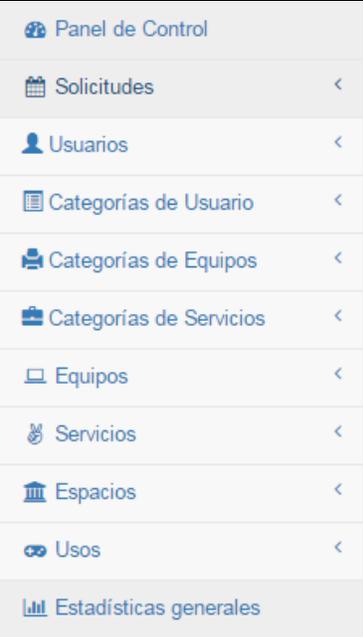
NAVEGACIÓN	
	
BOTONES	
 AGREGAR MODIFICAR	 VER DETALLES
 ELIMINAR	 DESHABILITAR
 HABILITAR	 MENSAJES AL USUARIO

Tabla 3. Continuación de guía de estilos de la aplicación web



Las imágenes que serán utilizadas en todas las interfaces son las del logo de la Universidad Central de Venezuela y el de la Facultad de Ciencias de cada lado de la cabecera. Como colores de fondo se encuentra el color gris, el cual identifica a la Facultad de Humanidades y Educación. El siguiente color a utilizar es el azul rey que complementa el tono claro del color gris y por último como color destacado un azul con una tonalidad más clara para resaltar opciones seleccionadas. La tipografía

seleccionada para el sistema fue Arial y para los títulos Helvetica pero en un mayor tamaño. Estos tipos de letras fueron utilizados ya que son letras básicas sin muchos detalles para no cansar la vista en ningún momento y son tipos de letra *sans serif* recomendadas para aplicaciones web.

Al navegar a través de las diferentes opciones del portal, la opción seleccionada se sombrea de azul para facilitar la ubicación de las opciones como se muestra en el Gráfico anterior.

La maqueta general de la aplicación, se encuentra separada en cabecera, panel central y menú lateral. La cabecera y el menú lateral no cambian a lo largo de la navegación del sistema, a diferencia del panel central que dependiendo del usuario muestra diferentes opciones en los paneles izquierdo, central y derecho.

Esta guía permite tener una idea más clara sobre los elementos a utilizar, los colores y la disposición de estos en toda la navegación del sistema. A continuación, se describirán los casos de uso considerados para el desarrollo de las funcionalidades del sistema.

3.2.3 Casos de uso

Los casos de uso son empleados para la especificación de requerimientos funcionales y en ellos se indican las acciones concretas que un usuario puede ejecutar en una interfaz específica de la aplicación. En la figura 24 se puede apreciar el diagrama de casos de uso de nivel cero (0).



Figura 24. Casos de uso de nivel 0.

Luego, en la figura 25 se puede ver el siguiente nivel de casos de uso, el nivel uno (1).

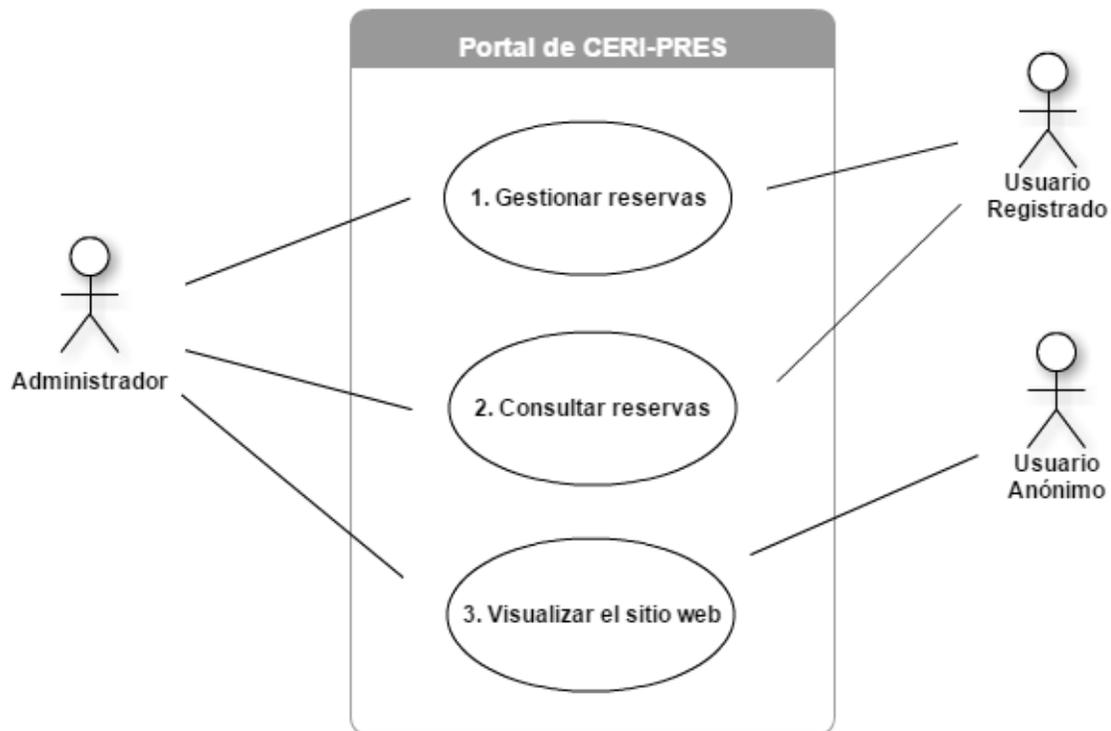


Figura 25. Casos de uso de nivel 1.

Una vez observados los casos de uso de nivel uno (1), se indican por separado su especificación en el nivel dos (2). En la figura 26 se observan los casos de uso de este nivel para el administrador.

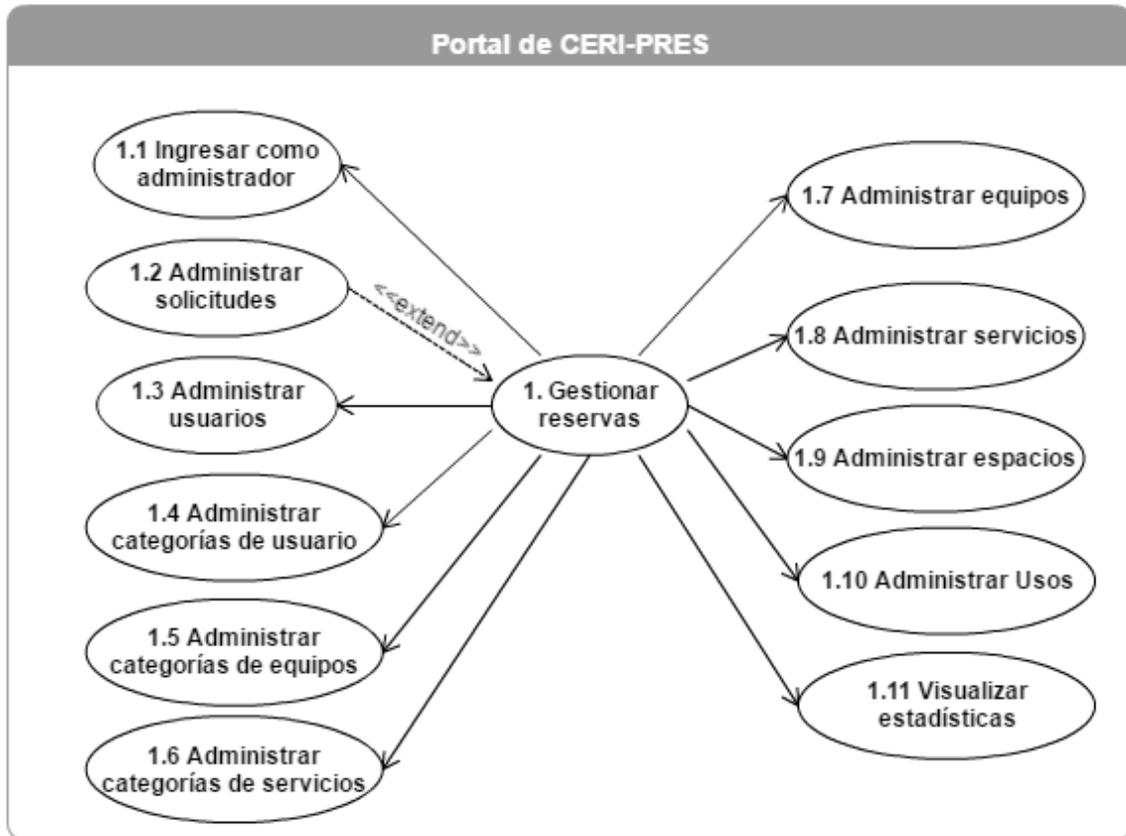


Figura 26. Casos de uso del administrador de nivel 2.

Luego, en la figura 27 se presenta el diagrama de casos de uso de nivel dos (2) del usuario registrado.

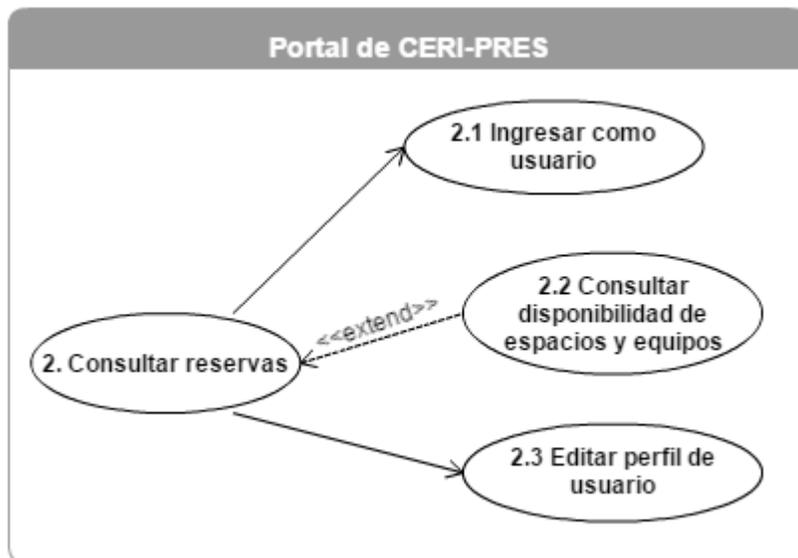


Figura 27. Casos de uso del usuario registrado de nivel 2.

Finalmente, en la figura 28 se muestra el diagrama de casos de uso de nivel dos (2) del usuario anónimo.

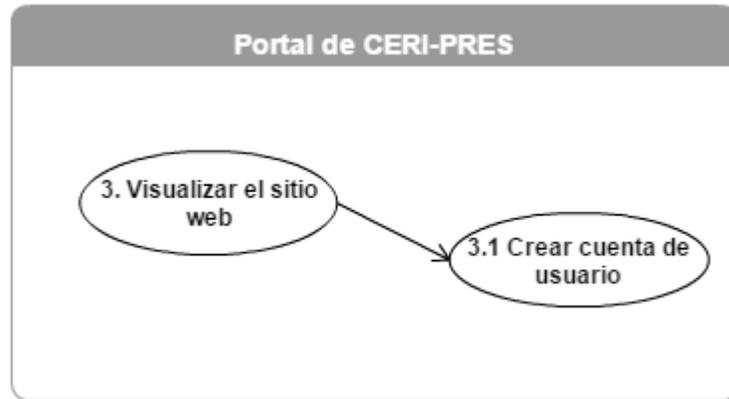


Figura 28. Casos de uso del usuario anónimo de nivel 2.

Una vez presentados los diagramas, se presenta la especificación de estos, que se refiere a la descripción de cada una de las partes definidas para lograr la descripción completa del sistema (Pérez, 2013). Las descripciones de los casos de uso representan todas las posibles interacciones de los actores con el sistema en los eventos enviados o recibidos por estos.

Tabla 4. Caso de uso 1. Gestionar reservas

CASO DE USO	CU.1. Gestionar reservas
ACTORES	Administrador
DESCRIPCION	El administrador puede gestionar el contenido a mostrar en el portal. 1. Seleccionar opción a administrar: El usuario selecciona la opción solicitudes, usuarios, categorías de usuario, categorías de equipos,

	<p>categorías de servicios, equipos, servicios, espacios, usos.</p> <p>2. Agregar elemento al sistema: El usuario agrega un elemento de la opción seleccionada ingresando sus datos en un formulario.</p> <p>3. Validar datos: El sistema valida los datos ingresados y envía un mensaje de confirmación si estos son correctos o indica cuales están incorrectos para que sean corregidos.</p>
--	---

Tabla 5. Caso de uso 1.1. Ingresar como administrador

CASO DE USO	CU.1.1 Ingresar como administrador
ACTORES	Administrador
DESCRIPCION	<p>El administrador puede iniciar sesión en el portal.</p> <p>1. Introducir datos: El administrador introduce su cédula y contraseña en la interfaz de ingreso.</p> <p>2. Validar datos: El sistema valida los datos ingresados y muestra la interfaz principal del panel de control si son correctos o indica cuales están incorrectos y permanece en la interfaz de ingreso.</p>

Tabla 6. Caso de uso 1.2. Administrar solicitudes

CASO DE USO	CU.1.2 Administrar solicitudes
ACTORES	Administrador

DESCRIPCION	<p>El administrador crea, lista, edita o elimina solicitudes en el sistema.</p> <ol style="list-style-type: none"> 1. Seleccionar opción de agregar en solicitudes: El administrador selecciona la opción para crear nuevas solicitudes. 2. Ingresar los datos: el administrador ingresa la información de la fecha, hora, equipos, espacios y servicios a reservar en la interfaz del formulario que se muestra. 3. Validar datos: El sistema valida los datos ingresados y muestra la interfaz con los detalles de la solicitud si son correctos o indica cuales están incorrectos y permanece en la interfaz de creación. 4. Listar solicitudes: el administrador selecciona la opción de listar y se muestra una tabla con todas las solicitudes.
--------------------	---

Tabla 7. Caso de uso 1.3. Administrar usuarios

CASO DE USO	CU.1.3 Administrar usuarios
ACTORES	Administrador
DESCRIPCION	<p>El administrador crea, lista, edita o elimina usuarios en el sistema.</p> <ol style="list-style-type: none"> 1. Seleccionar opción de agregar en usuarios: El administrador selecciona la opción para crear nuevos usuarios. 2. Ingresar los datos: el administrador

	<p>ingresa los datos personales del nuevo usuario en la interfaz del formulario que se muestra.</p> <p>3. Validar datos: El sistema valida los datos ingresados y muestra la interfaz con los detalles del usuario si son correctos o indica cuales están incorrectos y permanece en la interfaz de registro.</p> <p>4. Listar usuarios: el administrador selecciona la opción de listar y se muestra una tabla con todos los usuarios.</p>
--	---

Tabla 8. Caso de uso 1.4. Administrar categorías de usuario.

CASO DE USO	CU.1.4 Administrar categorías de usuario
ACTORES	Administrador
DESCRIPCION	<p>El administrador crea, lista, edita o elimina categorías de usuario en el sistema.</p> <p>1. Seleccionar opción de agregar en categoría de usuario: El administrador selecciona la opción para agregar nuevas categorías.</p> <p>2. Ingresar los datos: el administrador ingresa el nombre de la nueva categoría de usuario en la interfaz del formulario que se muestra.</p> <p>3. Validar datos: El sistema valida los datos ingresados y muestra la interfaz</p>

	<p>con la lista de las categorías si son correctos o indica cuales están incorrectos y permanece en la interfaz de agregar.</p> <p>4. Listar categorías de usuario: el administrador selecciona la opción de listar y se muestra una tabla con todas las categorías de usuario.</p>
--	---

Tabla 9. Caso de uso 1.5. Administrar categorías de equipo.

CASO DE USO	CU.1.5 Administrar categorías de equipos
ACTORES	Administrador
DESCRIPCION	<p>El administrador crea, lista, edita o elimina categorías de equipos en el sistema.</p> <p>1. Seleccionar opción de agregar en categoría de equipos: El administrador selecciona la opción para agregar nuevas categorías.</p> <p>2. Ingresar los datos: el administrador ingresa el nombre de la nueva categoría de equipo en la interfaz del formulario que se muestra.</p> <p>3. Validar datos: El sistema valida los datos ingresados y muestra la interfaz con la lista de las categorías si son correctos o indica cuales están incorrectos y permanece en la interfaz de agregar.</p> <p>4. Listar categorías de equipo: el</p>

	administrador selecciona la opción de listar y se muestra una tabla con todas las categorías de equipo.
--	---

Tabla 10. Caso de uso 1.6. Administrar categorías de servicios.

CASO DE USO	CU.1.6 Administrar categorías de servicios
ACTORES	Administrador
DESCRIPCION	<p>El administrador crea, lista, edita o elimina categorías de servicios en el sistema.</p> <ol style="list-style-type: none"> 1. Seleccionar opción de agregar en categoría de servicios: El administrador selecciona la opción para agregar nuevas categorías. 2. Ingresar los datos: el administrador ingresa el nombre de la nueva categoría de servicio en la interfaz del formulario que se muestra. 3. Validar datos: El sistema valida los datos ingresados y muestra la interfaz con la lista de las categorías si son correctos o indica cuales están incorrectos y permanece en la interfaz de agregar. 4. Listar categorías de servicios: el administrador selecciona la opción de listar y se muestra una tabla con todas las categorías de servicios.

Tabla 11. Caso de uso 1.7. Administrar equipos.

CASO DE USO	CU.1.7 Administrar equipos
ACTORES	Administrador
DESCRIPCION	<p>El administrador crea, lista, edita o elimina equipos en el sistema.</p> <ol style="list-style-type: none"> 1. Seleccionar opción de agregar en equipos: El administrador selecciona la opción para agregar nuevos equipos. 2. Ingresar los datos: el administrador ingresa el nombre del nuevo equipo en la interfaz del formulario que se muestra. 3. Validar datos: El sistema valida los datos ingresados y muestra la interfaz con la lista de los equipos si son correctos o indica cuales están incorrectos y permanece en la interfaz de agregar. 4. Listar equipos: el administrador selecciona la opción de listar y se muestra una tabla con todos los equipos.

Tabla 12. Caso de uso 1.8. Administrar servicios.

CASO DE USO	CU.1.8 Administrar servicios
ACTORES	Administrador
DESCRIPCION	<p>El administrador crea, lista, edita o elimina servicios en el sistema.</p> <ol style="list-style-type: none"> 1. Seleccionar opción de agregar en servicios: El administrador selecciona la opción para agregar nuevos equipos.

	<p>2. Ingresar los datos: el administrador ingresa el nombre del nuevo servicio en la interfaz del formulario que se muestra.</p> <p>3. Validar datos: El sistema valida los datos ingresados y muestra la interfaz con la lista de los servicios si son correctos o indica cuales están incorrectos y permanece en la interfaz de agregar.</p> <p>4. Listar servicios: el administrador selecciona la opción de listar y se muestra una tabla con todos los servicios.</p>
--	---

Tabla 13. Caso de uso 1.9. Administrar espacios.

CASO DE USO	CU.1.9 Administrar espacios
ACTORES	Administrador
DESCRIPCION	<p>El administrador crea, lista, edita o elimina espacios en el sistema.</p> <p>1. Seleccionar opción de agregar en espacios: El administrador selecciona la opción para agregar nuevos equipos.</p> <p>2. Ingresar los datos: el administrador ingresa el nombre del nuevo espacio en la interfaz del formulario que se muestra.</p> <p>3. Validar datos: El sistema valida los datos ingresados y muestra la interfaz</p>

	<p>con la lista de los espacios si son correctos o indica cuales están incorrectos y permanece en la interfaz de agregar.</p> <p>4. Listar espacios: el administrador selecciona la opción de listar y se muestra una tabla con todos los espacios.</p>
--	---

Tabla 14. Caso de uso 1.10. Administrar usos.

CASO DE USO	CU.1.10 Administrar usos
ACTORES	Administrador
DESCRIPCION	<p>El administrador crea, lista, edita o elimina usos en el sistema.</p> <p>1. Seleccionar opción de agregar en usos: El administrador selecciona la opción para agregar nuevos equipos.</p> <p>2. Ingresar los datos: el administrador ingresa el nombre del nuevo uso en la interfaz del formulario que se muestra.</p> <p>3. Validar datos: El sistema valida los datos ingresados y muestra la interfaz con la lista de los usos si son correctos o indica cuales están incorrectos y permanece en la interfaz de agregar.</p> <p>4. Listar usos: el administrador selecciona la opción de listar y se muestra una tabla con todos los usos.</p>

Tabla 15. Caso de uso 1.11. Visualizar estadísticas.

CASO DE USO	CU.1.11 Visualizar estadísticas
ACTORES	Administrador
DESCRIPCION	<p>El administrador puede visualizar las estadísticas de uso de espacios, equipos y usuarios del sistema.</p> <ol style="list-style-type: none"> 1. Seleccionar opción de estadísticas: El administrador selecciona la opción de estadísticas. 2. Ingresar los datos: el administrador ingresa el año y mes que desea ver. 3. Validar datos: El sistema valida los datos ingresados y muestra la interfaz con las estadísticas si son válidos o muestra un mensaje de error y permanece en la página si son incorrectos.

Tabla 16. Caso de uso 2. Consultar reservas

CASO DE USO	CU.2. Gestionar reservas
ACTORES	Administrador, usuario registrado
DESCRIPCION	<p>El administrador o usuario registrado puede consultar sus datos de usuario o la disponibilidad de equipos y espacios en una fecha y hora determinada.</p> <ol style="list-style-type: none"> 1. Consultar disponibilidad: El usuario o administrador consulta la disponibilidad de equipos y espacios. 2. Visualizar datos de usuario: El usuario o administrador selecciona la opción de perfil de usuario para visualizar sus datos en el sistema.

Tabla 17. Caso de uso 2.1. Ingresar como usuario

CASO DE USO	CU.2.1. Ingresar como usuario
ACTORES	Usuario registrado
DESCRIPCION	<p>El usuario registrado puede iniciar sesión en el portal.</p> <ol style="list-style-type: none"> 1. Introducir datos: El usuario registrado introduce su cédula y contraseña en la interfaz de ingreso. 2. Validar datos: El sistema valida los datos ingresados y muestra la interfaz principal de consulta si son correctos o indica cuales están incorrectos y permanece en la interfaz de ingreso.

Tabla 18. Caso de uso 2.2. Consultar disponibilidad de espacios y equipos.

CASO DE USO	CU.2.2. Consultar disponibilidad de espacios y equipos
ACTORES	Usuario registrado
DESCRIPCION	<p>El usuario registrado puede consultar que equipos y espacios están disponibles en una fecha y hora determinada.</p> <ol style="list-style-type: none"> 1. Introducir datos: El usuario registrado introduce la fecha y hora a consultar. 2. Validar datos: El sistema valida los datos ingresados y muestra la interfaz de resultados indicando si hay disponibilidad de equipos o espacios

	respectivamente.
--	------------------

Tabla 19. Caso de uso 2.3. Editar perfil de usuario.

CASO DE USO	CU.2.3. Editar perfil de usuario
ACTORES	Usuario registrado
DESCRIPCION	<p>El usuario registrado puede visualizar y editar sus datos personales en el sistema.</p> <ol style="list-style-type: none"> 1. Editar campos: El usuario registrado edita los campos con los datos que desea cambiar en su perfil de usuario. 2. Validar datos: El sistema valida los datos ingresados y muestra la interfaz de perfil de usuario si son correctos o indica cuales están incorrectos y permanece en la interfaz de edición. 3. Visualizar perfil de usuario: El usuario selecciona la opción de perfil de usuario y visualiza sus datos.

Tabla 20. Caso de uso 3. Visualizar el sitio web.

CASO DE USO	CU.3. Visualizar el sitio web
ACTORES	Usuario anónimo
DESCRIPCION	<p>El administrador, usuario registrado y usuario anónimo pueden visualizar la página de ingreso del sistema; el usuario anónimo por no poseer cuenta de usuario puede registrarse.</p> <ol style="list-style-type: none"> 1. Crear cuenta de usuario: el usuario anónimo crea una cuenta de usuario

	para convertirse en un usuario registrado.
--	--

Tabla 21. Caso de uso 3.1. Crear cuenta de usuario.

CASO DE USO	CU.3.1. Crear cuenta de usuario
ACTORES	Usuario anónimo
DESCRIPCION	<p>El usuario anónimo puede crear una cuenta de usuario.</p> <ol style="list-style-type: none"> 1. Seleccionar opción de registrarse: el usuario anónimo selecciona la opción de registrarse. 2. Crear cuenta de usuario: el usuario anónimo ingresa sus datos personales en el formulario que se muestra. 3. Validar datos: el sistema valida los datos y muestra la interfaz de ingreso si son correctos o indica cuales son incorrectos y permanece en la interfaz de registro.

3.2.4 Modelo de objetos del sitio web de préstamos

En esta sección se describen los objetos que existen en el contexto del sistema. A continuación, se presenta el modelo de objetos de dominio en la figura 29.

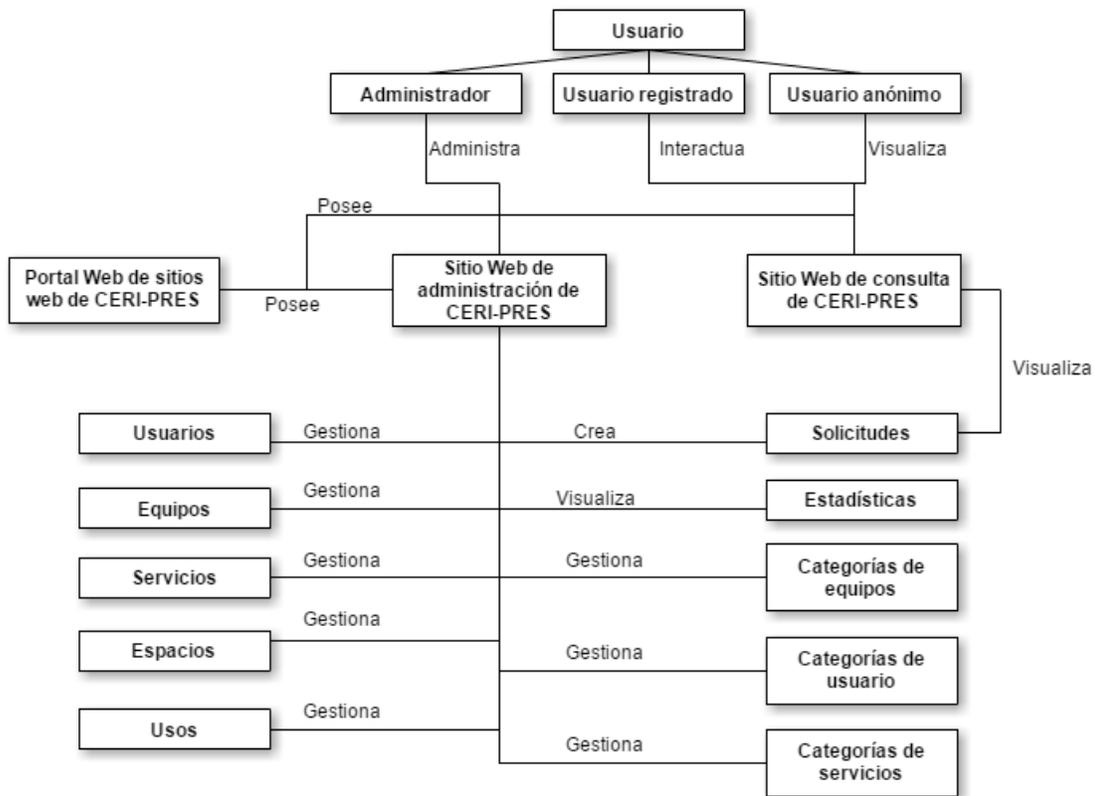


Figura 29. Modelo de objetos del dominio.

Como se puede apreciar en la figura 26, se tiene como dominio el portal de sitios web de CERI-PRES, que posee los sitios web de administración y de consulta de CERI-PRES. El primero es gestionado por el administrador y el segundo es visualizado por el usuario registrado en la sección de consulta y el usuario anónimo en la sección de ingreso.

El sitio web de administración contiene los usuarios, equipos, servicios, espacios, usos, categorías de equipo, categorías de usuario y categorías de servicios que son gestionados por el administrador. La creación de solicitudes es también manejada por el administrador, así como la visualización de las estadísticas.

Los usuarios registrados pueden ver la disponibilidad de equipos y espacios de las solicitudes por medio de la interfaz de consulta del sitio web de consulta de

CERI-PRES, y los usuarios anónimos visualizan únicamente el sitio web de consulta en su interfaz de ingreso al sistema.

Una vez finalizadas las fases de requisitos y análisis de la metodología AgilUs se procede a detallar las fases de prototipaje y de entrega.

3.3 Fase III: Prototipaje

La tercera etapa es el prototipaje en la cual se implementa un prototipo rápido de la interfaz de usuario a partir de los patrones de interacción, el cual va evolucionando hasta convertirse en el producto final, y se realizan evaluaciones de usabilidad apropiadas a esta etapa: la evaluación heurística y la lista de comprobación. De igual manera en esta etapa se realiza el pensamiento en voz alta que permite tener un *feedback* sobre lo que el usuario está pensando a lo largo de su recorrido por el sistema.

3.3.1 Prototipo ejecutable

En esta sección se muestra el prototipo ejecutable inicial con las interfaces y funcionalidades consideradas. La primera interfaz, visible para todos los perfiles de usuarios es la de acceso al sistema o *login*.



Por favor indique sus datos de acceso

Cédula
Cédula

Contraseña
Contraseña

Ingresar

¿No tienes cuenta? [Regístrate](#)

[Recuperar mi contraseña](#)

Figura 30. Interfaz de ingreso al sistema.

Por medio de esta interfaz el usuario registrado y el administrador pueden ingresar al sistema colocando su cédula y contraseña en los campos correspondientes y luego haciendo click en el botón “ingresar”.

Para estos usuarios está disponible también la opción de recuperar su contraseña como se indica en la figura 31.



Interfaz de recuperación de contraseña. El formulario tiene un encabezado que dice "Ingrese el correo con el cual está registrado". Debajo, hay un campo de texto etiquetado "Correo electrónico" con el texto "Correo electrónico" dentro. Al final del formulario hay dos botones: "Enviar" (azul) y "Cancelar" (rojo).

Figura 31. Interfaz de recuperación de contraseña.

Para recuperar su contraseña, el usuario registrado o administrador deberá indicar el correo electrónico con el cual se registró en el sistema, luego recibirá un correo electrónico con un enlace único para crear una nueva contraseña.

En el caso del usuario anónimo, dado que este no posee cuenta en el sistema, su única opción posible es registrarse para convertirse en usuario registrado.

En primer lugar, se dispone de un panel de control como página de inicio al ingresar al sistema como se muestra en la figura 32.

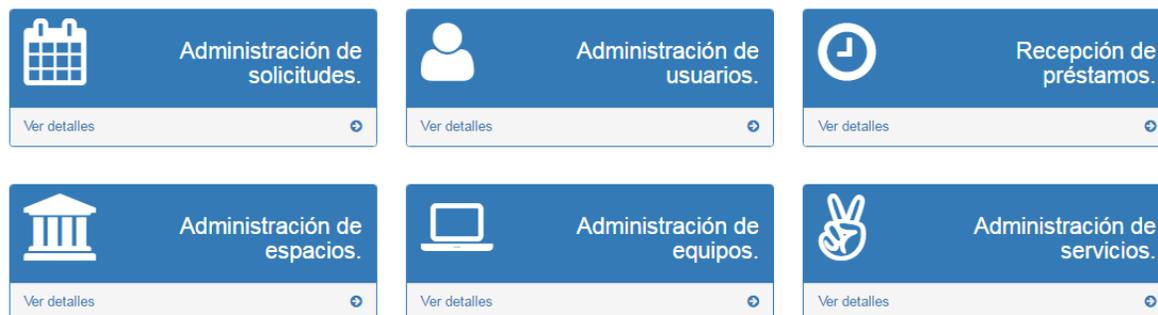


Figura 32. Panel de control.

Con este panel de control el administrador puede acceder a las secciones más importantes de la aplicación. Los listados de solicitudes, usuarios, recepción de préstamos, espacios, equipos y servicios pueden ser visualizados al hacer click en el botón “ver detalles” de cada sección.

Luego, el administrador podrá desplazarse por todas las secciones de la aplicación disponible por medio de un menú lateral que las contiene como se muestra en la figura 33.

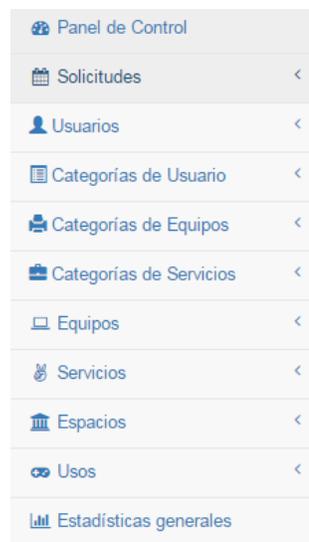


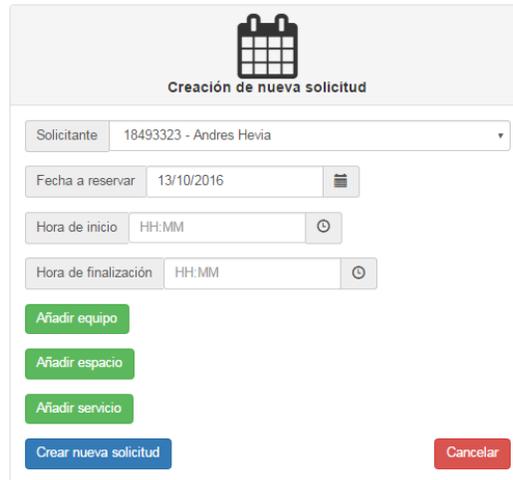
Figura 33. Menú principal.

Primeramente, de acuerdo con la figura 34, se tiene la funcionalidad para manejar las solicitudes, crearlas, listarlas y recibir los préstamos.



Figura 34. Opción de solicitudes del menú.

La opción de crear una solicitud nos muestra la siguiente vista de la figura 35.



Creación de nueva solicitud

Solicitante: 18493323 - Andres Hevia

Fecha a reservar: 13/10/2016

Hora de inicio: HH:MM

Hora de finalización: HH:MM

Añadir equipo

Añadir espacio

Añadir servicio

Crear nueva solicitud Cancelar

Figura 35. Interfaz de creación de una solicitud.

En ella se puede crear una nueva solicitud indicando el usuario que desea hacer la reserva, la fecha a reservar, la hora de inicio, la hora de finalización, y por medio de los botones en color verde, se agregan los equipos, espacios y servicios a reservar. Una vez agregados todos los datos, el administrador debe hacer click en el botón “crear nueva solicitud”.

En la opción de listar, como se ve en la figura 36, se pueden visualizar todas las solicitudes creadas en el sistema.



Solicitudes

Mostrar 10 entradas

Buscar:

#	Solicitante	Fecha de solicitud	Fecha de uso	Habilitado	Ver detalles	Modificar	Eliminar	Habilitar / Deshabilitar
32	Andres Hevia	2016-09-30	2016-10-08	Sí				
33	Andres Hevia	2016-09-30	2016-10-07	Sí				

Mostrando 1 hasta 2 de 2 entradas

Nueva solicitud

Anterior 1 Siguiente

Volver al inicio

Figura 36. Interfaz de listado de solicitudes.

3.2.3 Evaluación Heurística

La evaluación heurística fue realizada por cinco (5) personas involucradas directamente en el desarrollo del prototipo, encontrando la mayor cantidad de errores en la interfaz posible evaluando individualmente todos los aspectos de la interfaz de acuerdo a los siguientes criterios generales:

H1: Diálogo natural y simple.

H2: Hablar el lenguaje del usuario.

H3: Minimizar la carga cognitiva.

H4: Consistencia.

H5: Feedback.

H6: Proveer claramente las salidas.

H7: Proveer Shortcuts.

H8: Mensajes de Error descriptivos.

H9: Prevención de Errores.

H10: Asistencia al Usuario.

Como siguiente paso se realiza una evaluación numérica que permita calificar el tipo de error que tiene una interfaz, de acuerdo a una clasificación del cero (0) al cuatro (4) en la que el primero nos indica que no existe problema de usabilidad alguno mientras que el último nos indica un problema de usabilidad grave, escalando en los valores intermedios de la siguiente forma:

0 - No es un problema de usabilidad.

1 - Problema cosmético.

2 - Problema menor.

3 - Problema mayor de usabilidad; importante fijar solución.

4- Usabilidad catastrófica, imperativo fijar solución.

Estos valores son colocados en una tabla donde muestra el problema, la heurística a la que pertenece, la valoración y la posible solución que se le puede dar. A continuación, en la tabla TAL se presenta la evaluación heurística realizada.

Tabla 22. Evaluación heurística

Problema	Heurística	Valor	Solución
En los detalles de una solicitud, no se observa la hora de inicio y finalización del préstamo.	H6	2	Agregar a los detalles la hora de inicio y devolución de los espacios y equipos.
Al presionar el botón de eliminar, elimina el elemento sin confirmar.	H9, H10	4	Agregar una ventana emergente de confirmación
En ocasiones muestra error de conexión a la base de datos, aunque el error real es otro.	H6, H8, H10	3	Mostrar un error de acuerdo al contexto que lo provoca.
En los formularios de modificar, no se carga la información ya existente.	H3	3	Cargar la información de la base de datos y mostrarla al usuario en el

			formulario
En la mayoría de los botones se observa el color azul, y en algunos el color amarillo.	H4	2	Colocar todos los botones de éxito o agregar en verde, en azul los informativos, rojo los sensibles y grises los de volver.

3.2.4 Lista de Comprobación

Para este trabajo se utilizó la lista de comprobación siguiente:

Tabla 23. Lista de comprobación

Elemento a comprobar	Se cumple	No se cumple
Reconocer en lugar de recordar	X	
Usar indicadores visuales	X	
Utilizar metáforas adecuadas	X	
Los controles del sistema deben estar claramente visibles y sus funciones identificables	X	
Flexibilidad en la interfaz de usuario		X
Hacer sentir al usuario que puede tomar decisiones	X	
Proveer <i>feedback</i> a las acciones del usuario	X	
Minimizar las posibilidades de cometer errores	X	
Permitir al usuario recuperarse de los errores		X
Mantener la interfaz simple, sencilla y organizada	X	
Permitir al usuario personalizar la IU		X
Mostrarle al usuario lo que necesite según cada sección que consulte	X	

Evitar palabras coloquiales y abreviaturas	X	
Consistencia en apariencia y uso	X	

De acuerdo a los resultados obtenidos e indicados en la tabla, la mayoría de las acciones de la lista de comprobación se cumplen para todo el sistema, tanto para el sitio de usuarios registrados y no registrados como para el sitio de administradores, lo que permite concluir que el software cumple con la mayoría de los parámetros de usabilidad propuestos en la lista descrita anteriormente para el diseño de las interfaces de usuario.

3.4 Fase IV: Fase de entrega

La última fase de la metodología AgilUs se desglosa en la aplicación de un protocolo de preguntas y la realización de pruebas de aceptación para certificar que el producto desarrollado sea un software usable. Como cierre, se pone en producción la aplicación.

3.4.1 Protocolo de preguntas

Esta actividad se realizó formulando preguntas directas acerca del sistema y la capacidad del usuario (o ausencia) para contestar. Para realizar este protocolo se proporcionó a los participantes el producto que va a ser sometido a prueba y un escenario.

Para el protocolo de preguntas fueron cuestionadas cuatro (4) personas que utilizaron el sistema y se les realizaron diferentes preguntas, con respecto al portal y a los sitios web creados. Estas preguntas fueron evaluadas entre uno (1) y cinco (5), siendo los valores: muy malo en valor uno (1), malo en valor dos (2), regular en valor tres (3), muy bueno en valor cuatro (4) y excelente en valor cinco (5).

A continuación, se presenta la tabla con las preguntas.

Tabla 24. Protocolo de preguntas

Pregunta	1	2	3	4	5
¿Se puede identificar con rapidez la página que se quiere visitar y llegar fácil y directamente a ella?				60%	40%
¿Tiene el portal una dirección URL correcta, clara y fácil de recordar?			20%	60%	20%
¿Está claramente indicado el nombre de la página que se está navegando en cada caso?				80%	20%
¿Hay enlaces a la página inicial desde todas las páginas?					100%
¿Se puede volver fácilmente a una página ya visitada?			20%	60%	20%
¿Los enlaces son fáciles de identificar?				80%	20%
¿Los íconos se interpretan con facilidad?					100%
¿Se informa constantemente al usuario acerca de lo que está pasando? Por ejemplo: notificaciones de éxito o falla.					100%
¿Tiene buscador? ¿La caja de texto es lo suficientemente ancha?			20%	80%	
¿Se informa cómo solucionar el problema cuando se produce un error?		100%			
¿Se conoce de forma precisa y completa qué contenidos o servicios ofrece realmente el sitio web?					100%
¿Los títulos describen la información que encabezan?				20%	80%
¿Tiene subtítulos, encabezamientos o capítulos?		80%	20%		
¿La información está ordenada lógicamente?				80%	20%
¿La información se presenta con claridad?				80%	20%
¿La información está libre de errores gramaticales y ortográficos?					100%
¿Emplea un lenguaje claro y conciso?					100%
¿La imagen de fondo ofrece un buen contraste con el tipo de letra?			20%	80%	
¿Es adecuado el tamaño de la letra utilizada?				60%	40%
¿Se reconocen fácilmente los enlaces? ¿Se puede observar su estado (visitados, activos, etc.)?				80%	20%
¿Es predecible la respuesta antes de hacer clic sobre el enlace?					100%
El vocabulario, el lenguaje, los conceptos, ¿son adecuados para el destinatario previsto?					100%

Gracias a estos resultados se concluye que el sistema tuvo una buena aceptación entre los usuarios, mostrando en la mayoría de las respuestas que satisface las necesidades y cumple con los requerimientos funcionales descritos desde principios de la aplicación del método AgilUs. A continuación se prosigue con las pruebas de aceptación.

3.4.2 Pruebas de aceptación

Las pruebas de aceptación verifican que el sistema satisfaga los requerimientos funcionales y no funcionales. Son básicamente pruebas sobre el funcionamiento del sistema completo; sin embargo, se pueden utilizar para verificar la satisfacción de los clientes en relación a los requerimientos no funcionales. La ejecución y aprobación final de las pruebas de aceptación le corresponden al usuario final (Acosta, 2011).

Las pruebas de aceptación para el sistema completo de la aplicación web, fueron realizadas con los mismos clientes quienes en el caso de este trabajo son los usuarios principales. Dichas pruebas de aceptación fueron el protocolo de preguntas y el pensamiento en voz alta aplicadas de forma extendida, es decir, posterior a la obtención de los resultados iniciales, para obtener los resultados definitivos mostrados anteriormente con retroalimentación constante de los usuarios mientras utilizaban la aplicación, dando como resultado un producto final, totalmente funcional y alojado en servidores del CERI-FHE de la UCV. Con estas pruebas y junto a la entrega final del producto, se finalizó la aplicación de la metodología AgilUs para el desarrollo de la aplicación web para la gestión de equipos, espacios físicos y servicios prestados por el Centro de Experimentación de Recursos Instruccionales.

Capítulo 4: Resultados

En este capítulo se muestra el producto final obtenido como resultado de la aplicación de la metodología de desarrollo. En primer lugar se muestran cada una de las interfaces y funcionalidades de la aplicación desarrollada. La descripción de las funcionalidades se hará de forma progresiva de acuerdo al nivel de privilegios de los tres perfiles de usuario de la aplicación.

En primer lugar se mostrarán las interfaces y funcionalidades de los usuarios anónimos y usuarios registrados, y luego las interfaces y funcionalidades de los administradores.

4.1. Interfaces y funcionalidades de usuarios anónimos

En la figura 37 se puede apreciar la primera interfaz, visible para todos los perfiles de usuarios: la de acceso al sistema o *login*.

Por favor indique sus datos de acceso

Cédula

Contraseña

Ingresar

[¿No tienes cuenta? Regístrate](#) [Recuperar mi contraseña](#)

Figura 37. Interfaz de ingreso al sistema.

Para los usuarios anónimos la única opción posible es presionar el enlace de “Regístrate” para crear una cuenta de usuario, y se cargará la interfaz de registro que se muestra en la figura 38.

→ Registro de nuevo usuario ←

Primer nombre

Segundo nombre

Primer apellido

Segundo apellido

Correo electrónico

Correo institucional

Teléfono

Cédula

Contraseña

Confirmar contraseña

Figura 38. Interfaz de registro

La figura 39 muestra la continuación de la interfaz de registro.



The image shows a registration form with the following elements:

- Tipo de usuario:** A dropdown menu with "Estudiante" selected.
- Twitter:** A text input field with an "@" icon and the text "Twitter".
- Facebook:** A text input field with an "@" icon and the text "Facebook".
- Instagram:** A text input field with an "@" icon and the text "Instagram".
- Registrar:** A large blue button with the text "Registrar".
- Cancelar registro:** A small blue link with a cross icon and the text "Cancelar registro".

Figura 39. Interfaz de registro

Cuando han sido llenados todos los datos del formulario de registro correctamente, el usuario anónimo deberá presionar el botón registrar para finalizar con esta tarea.

Una vez finalizado el proceso de registro, el usuario anónimo se convierte en un usuario registrado y éste podrá ingresar al sistema usando la interfaz de *login* mostrada anteriormente. A continuación, se muestran las interfaces relacionadas con los usuarios registrados.

4.2. Interfaces y funcionalidades de usuarios registrados

El usuario registrado cuenta con los privilegios para ejecutar las acciones de un usuario anónimo y adicionalmente posee las funcionalidades siguientes.

4.2.1. Consultar equipos y espacios

A través de la interfaz mostrada en la figura 40, el usuario registrado puede consultar que equipos y espacios se encuentran disponibles en una fecha y hora determinada.



La interfaz de consulta de equipos y espacios disponibles se muestra en un recuadro con un fondo gris claro. En la parte superior, hay un icono de dos burbujas de chat y el título "Consultar disponibilidad de espacios y equipos". Debajo del título, hay tres campos de entrada de texto con botones de selección a la derecha. El primer campo es para la fecha, con el valor "13/10/2016" y un icono de calendario. El segundo campo es para la hora de inicio, con el valor "HH:MM" y un icono de reloj. El tercer campo es para la hora de fin, con el valor "HH:MM" y un icono de reloj. En la parte inferior, hay dos botones: "Consultar" (azul) y "Cancelar" (rojo).

Figura 40. Interfaz de consulta de equipos y espacios disponibles.

La segunda funcionalidad disponible para los usuarios registrados es el poder ver su perfil de usuario con sus datos personales.

4.2.2. Ver perfil

Como se muestra en la figura 41, el usuario registrado podrá ver los detalles de su perfil.

Detalles del Usuario	
ID Base de datos:	4
Primer nombre:	John
Segundo nombre:	E
Primer apellido:	Darwin
Segundo apellido:	L
Cédula:	7654321
Teléfono:	0412-4443331
Correo electrónico:	jonhdarwin@doe.com
Correo institucional:	jonhdarwin@foo.com
Tipo de usuario:	Regular
Categoría:	Estudiante
Facebook:	jonhdarwin
Instagram:	jonhdarwin
Twitter:	jonhdarwin
Editar usuario Volver	

Figura 41. Interfaz de detalles del usuario

Y desde esta interfaz, el usuario puede acceder a la tercera y última funcionalidad para su perfil: modificar sus datos personales.

4.2.3. Modificar perfil

En la figura 42, se muestra la interfaz para modificar los datos personales del usuario.



The image shows a user profile configuration form titled "Configuración del usuario". At the top, there is a user icon and the title. Below the title, there are several input fields for personal information:

- Primer nombre: John
- Segundo nombre: E
- Primer apellido: Darwin
- Segundo apellido: L
- Correo electrónico: jonhedarwin@doe.com
- Correo institucional: jonhedarwin@foo.com
- Teléfono: 0412 (country code dropdown) and 4443331 (number)
- Cédula: 7654321
- Contraseña: Contraseña
- Confirmar contraseña: Confirmar contraseña
- Tipo de usuario: Estudiante (dropdown menu)
- Twitter: jonhdarwin
- Facebook: jonhdarwin
- Instagram: jonhdarwin

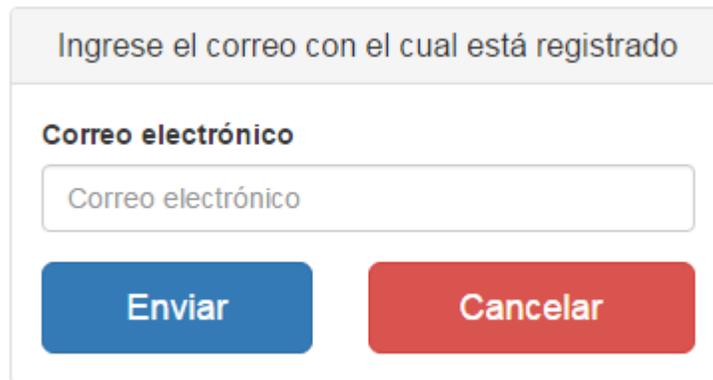
At the bottom of the form, there are two buttons: "Actualizar datos" (blue) and "Cancelar" (red).

Figura 42. Interfaz de actualización de datos de usuario

A continuación se muestra la funcionalidad para recuperar la contraseña del usuario.

4.2.4. Recuperar contraseña

La última funcionalidad disponible para los usuarios registrados es la de recuperación de contraseña, para ello deberán presionar el enlace de la interfaz de *login* para tal fin, mostrando como resultado la interfaz observada en la figura 43.



Interfaz de recuperación de contraseña. El formulario tiene un encabezado que dice "Ingrese el correo con el cual está registrado". Debajo de esto, hay un campo de entrada etiquetado "Correo electrónico" con el texto "Correo electrónico" como marcador de posición. Al final del formulario, hay dos botones: "Enviar" (azul) y "Cancelar" (rojo).

Figura 43. Interfaz de recuperación de contraseña.

Para recuperar su contraseña, el usuario deberá indicar el correo electrónico con el cual se registró en el sistema, luego recibirá mensaje con un enlace único para modificar su contraseña.

La contraseña podrá ser cambiada en la interfaz que se muestra en la figura 44.

The image shows a web form for changing a password. At the top, there is a header with the text "Ingrese y confirme su nueva contraseña". Below this, there are two input fields. The first is labeled "Contraseña" and contains the placeholder text "Contraseña". The second is labeled "Confirmar contraseña" and contains the placeholder text "Confirmar contraseña". At the bottom of the form, there are two buttons: a blue button labeled "Cambiar contraseña" and a white button labeled "Volver".

Figura 44. Interfaz de cambio de contraseña

Una vez exploradas las interfaces y funcionalidades disponibles para usuarios registrados y anónimos, se procede con la descripción de aquellas disponibles solo para administradores.

4.3 Interfaces y funcionalidades de administradores

Los administradores son los usuarios que disponen de más funcionalidades en la aplicación. Además de todas las funcionalidades de los usuarios anónimos y los registrados, pueden listar, modificar, eliminar, deshabilitar y habilitar todas las categorías y elementos presentes. A continuación se detallan las interfaces sobre dichas categorías y elementos.

4.3.1. Panel de control

En primer lugar, se dispone de un panel de control, como página de inicio al ingresar al sistema. En la figura 45 se puede observar dicho panel.



Figura 45. Panel de control.

Con este panel de control el administrador puede acceder a las secciones más importantes de la aplicación. Los listados de solicitudes, usuarios, recepción de préstamos, espacios, equipos y consultas pueden ser visualizados al presionar el botón “Ver detalles” de cada sección.

El administrador también puede volver a la sección de panel de control al hacer clic en el botón correspondiente del menú lateral.

4.3.2. Menú principal

El administrador podrá desplazarse por todas las secciones de la aplicación disponible por medio de un menú lateral que las contiene. En la figura 46 se puede observar dicho menú.

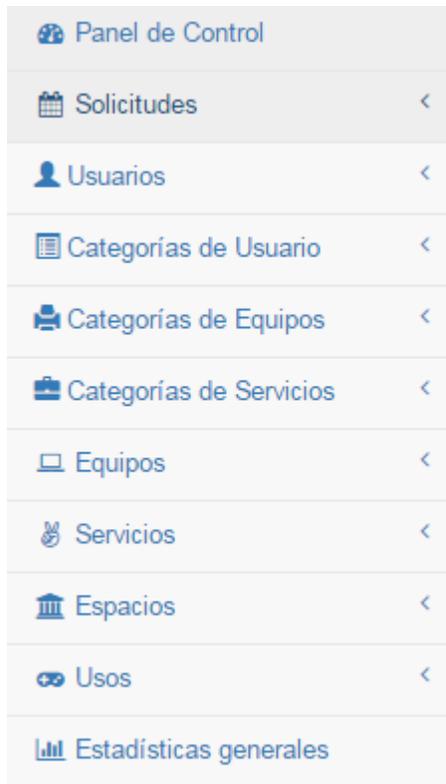


Figura 46. Menú principal.

A continuación, se procede a desglosar las distintas opciones pertenecientes al menú.

4.3.3. Interfaces de solicitud

Las funcionalidades pertenecientes a las interfaces de solicitud son crear, listar y recepción de préstamos. En la figura 47 se puede visualizar la interfaz de creación de una solicitud.

Creación de nueva solicitud

Solicitante 18493323 - Andres Hevia

Fecha a reservar 13/10/2016

Hora de inicio HH:MM

Hora de finalización HH:MM

Añadir equipo

Añadir espacio

Añadir servicio

Crear nueva solicitud

Cancelar

Figura 47. Interfaz de creación de una solicitud.

En ella se puede crear una nueva solicitud indicando el usuario que desea hacer la reserva, la fecha a reservar, la hora de inicio, la hora de finalización, y por medio de los botones en color verde, se agregan los equipos, espacios y servicios a reservar. Una vez agregados todos los datos, el administrador debe presionar el botón “Crear nueva solicitud”.

Al finalizar de manera exitosa la creación de una solicitud, el sistema dirige al usuario a la lista de solicitudes, que se puede observar en la figura 48.

Solicitudes								
Mostrar 10 entradas						Buscar: <input type="text"/>		
# ^	Solicitante	Fecha de solicitud	Fecha de uso	Habilitado	Ver detalles	Modificar	Eliminar	Habilitar / Deshabilitar
32	Andres Hevia	2016-09-30	2016-10-08	Sí				
33	Andres Hevia	2016-09-30	2016-10-07	Sí				

Mostrando 1 hasta 2 de 2 entradas

[Anterior](#) **1** [Siguiente](#)

[Nueva solicitud](#) [Volver al inicio](#)

Figura 48. Interfaz de listado de solicitudes.

La lista de solicitudes permite ver los detalles de cada solicitud, modificarla, eliminarla, deshabilitarla y filtrar por medio de un campo de búsqueda una solicitud en específico de acuerdo a cualquier dato que se tenga de ella (solicitante, fecha, entre otros).

La última opción entre las interfaces de solicitud es recepción de préstamos. En la figura 49, se puede observar la lista con las solicitudes que aún no han sido cerradas.

Cierre de solicitud				
Mostrar 10 entradas				Buscar: <input type="text"/>
# ^	Solicitante	Fecha de solicitud	Fecha de uso	Finalizar y cerrar
32	Andres J Hevia G	2016-09-30	2016-10-08	
33	Andres J Hevia G	2016-09-30	2016-10-07	

Mostrando 1 hasta 2 de 2 entradas

[Anterior](#) **1** [Siguiente](#)

[Volver al inicio](#)

Figura 49. Interfaz de cierre de solicitudes.

En esta interfaz se tiene la opción de hacer la devolución de un préstamo, dando por finalizada esa solicitud. Al presionar el botón de la lista “Finalizar y cerrar”

en alguna de las solicitudes, se muestra la interfaz de confirmación, como se puede observar en la figura 50.

Confirmar cierre de la solicitud #33

Por favor confirme los datos del préstamo que será cerrado.

Solicitud número: 33
Reservado por: Andres J Hevia G
Solicitante: Andres J Hevia G
Fecha solicitud: 2016-09-30
Fecha uso: 2016-10-07
Hora entrega: 07:04 AM
Hora devolución: 09:10 AM

Equipos reservados:

- Camara Digital Samsung

Observaciones

Observaciones

Confirmar cierreVolver

Figura 50. Interfaz de confirmación de cierre de solicitudes.

En esta interfaz el administrador podrá cerrar la solicitud y deberá colocar las observaciones correspondientes al caso antes de presionar el botón “Confirmar cierre”.

Una vez culminada la descripción de la sección de solicitudes, se procede a detallar las funcionalidades en la sección de usuarios.

4.3.4. Interfaces de usuarios

Las opciones disponibles en la sección de usuarios son agregar y listar. En la figura 51 se puede observar la interfaz de listar usuarios.

Cédula*	Nombres	Apellidos	Correo electrónico	Habilitado	Ver detalles	Modificar	Eliminar	Habilitar / Deshabilitar
7654321	John E	Darwin L	jonhedarwin@doe.com	Si				
18493323	Andres J	Hevia G	eternal.aj@gmail.com	Si				
19991992	Manuel Alejandro	Silva Blanco	manuelsilva@hotmail.com	Si				

Mostrar 10 entradas

Mostrando 1 hasta 3 de 3 entradas Anterior 1 Siguiente

[Agregar usuario](#) [Volver al inicio](#)

Figura 51. Listado de usuarios.

La vista de listado le brinda al administrador la posibilidad de ver los detalles de cada perfil, modificar sus datos, habilitar, deshabilitar o eliminar un usuario.

Al detallar las funcionalidades de los usuarios, se procede a describir las funcionalidades en categorías de usuario.

4.3.5. Interfaces de categorías

Las interfaces de categorías engloban las categorías de usuario, de equipo y de servicios. Todas brindan al usuario las mismas opciones: agregar y listar. En la figura 52, se observa la interfaz de listado de categoría de equipos.

Categorías de equipo					
Mostrar	10	entradas	Buscar: <input type="text"/>		
#	Categoría de equipo	Habilitado	Modificar	Eliminar	Habilitar / Deshabilitar
4	Laptops	Sí			
5	Proyectores	Sí			
6	Accesorios	Sí			
7	Videos	Sí			

Mostrando 1 hasta 4 de 4 entradas

Anterior **1** Siguiente

[Agregar categoría de equipo](#) [Volver al inicio](#)

Figura 52. Interfaz de listado de categoría de equipos.

Al presionar el botón “Agregar categoría de equipo”, se visualiza la interfaz de añadir categoría de equipo, que se puede observar en la figura 53.



Nombre de nueva categoría de equipo

Añadir categoría

Cancelar

Figura 53. Interfaz de agregar nueva categoría de equipo.

4.3.6. Interfaces de recursos

Los recursos del sistema son los equipos, servicios, espacios, y los usos de cada espacio. Las interfaces de recursos brindan las opciones de agregar y listar. En la figura 54 se puede visualizar la interfaz de agregar un nuevo espacio.



Figura 54. Interfaz de agregar nuevo espacio.

Al presionar el botón “Añadir espacio”, el sistema redirige al administrador al listado de espacios disponibles, que se puede observar en la figura 55.

Espacios						
#	Nombre espacio	¿Otro espacio?	Habilitado	Modificar	Eliminar	Habilitar / Deshabilitar
24	Auditorio AE5	No	Si			
25	Terraza AE1	No	Si			
27	Terraza AE2	No	Si			

Mostrando 1 hasta 3 de 3 entradas

Mostrar entradas

Buscar:

Anterior **1** Siguiente

[Nuevo espacio](#) [Volver al inicio](#)

Figura 55. Listado de espacios.

Como todas las interfaces de listar presentadas anteriormente, el administrador tiene las funcionalidades de modificar, eliminar, habilitar o deshabilitar alguno de los espacios mostrados. Al seleccionar la opción modificar, se presenta la vista mostrada en la figura 56.



The image shows a web interface for modifying a space. At the top, there is a grey header with a circular refresh icon and the text "Modificar espacio". Below the header is a white form area. Inside the form, there is a text input field with the value "Terraza AE1". Below the input field is a checkbox with the label "Es otro espacio.". At the bottom of the form, there are two buttons: a blue button labeled "Actualizar espacio" and a red button labeled "Cancelar".

Figura 56. Interfaz de modificación de espacios.

En esta interfaz, el administrador puede modificar el nombre del espacio, así como indicar si éste será considerado como “otro espacio” haciendo clic en la casilla de verificación. Una vez finalizada la edición, el administrador debe hacer clic en el botón “Actualizar espacio”.

La última opción que tienen los administradores es la sección de estadísticas, que se procede a describir.

4.3.7. Interfaces de estadísticas

Al ingresar a la sección de estadísticas por medio del menú principal, el administrador debe seleccionar el año y mes que desea visualizar, en la vista que se observa en la figura 57.



Estadísticas generales.
Seleccione el mes y el año que desea visualizar:

Año: 2015

Mes: Enero

Mostrar Cancelar

Figura 57. Interfaz inicial estadísticas generales.

Una vez seleccionado el año y el mes, el administrador deberá presionar el botón “mostrar” y se mostrará una interfaz con las opciones simplificadas en renglones de estadísticas por categorías de usuario, por préstamo de salas y espacios, por el uso de salas y espacios y por préstamos de equipos.

En la figura 58, se puede observar la vista de la interfaz simple de estadísticas generales, para seleccionar un renglón en particular.



Figura 58. Interfaz simple de estadísticas generales.

Y al hacer clic en una de las categorías, ésta se despliega con la lista de estadísticas del renglón correspondiente. En la figura 59, se puede visualizar la sección de estadísticas de préstamo de equipos.

Estadísticas del servicio de préstamos de equipos

Mostrar 10 entradas

Equipos	Cantidad de prestamos	Tiempo de uso
CESIO 1	1	01:00
Convertidor trifásico	0	00:00
Infocus	1	02:06
Laptop HP	0	00:00
TOTALES	2	03:06

Mostrando 1 hasta 5 de 5 entradas Anterior 1 Siguiente

Figura 59. Interfaz extendida de estadísticas generales.

Conclusiones

Finalizadas las etapas contempladas en el método de desarrollo de software AgilUs, especialmente el protocolo de preguntas y las pruebas de aceptación, se puede concluir que la aplicación web resultado cumple con los objetivos planteados, esto es, gestiona todos los procesos relacionados con la reserva de equipos, espacios físicos y servicios del CERI.

Para el desarrollo de la aplicación se utilizó una combinación de lenguajes de programación, librerías, marcos de trabajo y tecnologías, tanto del lado del cliente como del servidor. En el lado del cliente, se utilizaron HTML, CSS, JavaScript y jQuery. Para comunicar al cliente con el servidor, se usó AJAX, enviando objetos JSON de extremo a extremo. Y del lado del servidor, se utilizó PHP, con el marco de trabajo CodeIgniter. Todos los datos manejados por el SMDB MySQL.

Además de las funcionalidades que brinda la aplicación, como posibilidades de crear, listar, actualizar, eliminar, deshabilitar y habilitar usuarios, categorías y recursos, existen otras ventajas, inherentes a las aplicaciones web. Ahora el CERI puede realizar múltiples reservas de manera concurrente y al estar instalada la aplicación en un servidor del centro, no requieren acceso permanente a internet para usarla.

Los datos del sistema son utilizados para generar estadísticas que faciliten la toma de decisiones, y ahora se encuentran a buen resguardo, respaldados por las medidas de seguridad del SMDB, además del módulo de perfiles de acceso al sistema.

Finalmente, aunque la aplicación cumple con las expectativas proyectadas, se exhorta al personal del CERI y a los desarrolladores que forman parte de su comunidad, proponer e implementar mejoras no contempladas en el presente trabajo de investigación.

Recomendaciones

- Debido a las características del usuario que utilizará el sistema, se recomienda un previo adiestramiento sobre el uso de la herramienta.
- Se recomienda la generación de respaldos semanales de la base de datos.
- Invitar a la comunidad del CERI a una jornada de registro y presentación del sistema.

Trabajos Futuros

- Integrar la aplicación de reservas a la página web del CERI.
- Agregar al módulo de estadísticas la posibilidad de generar reportes en formato PDF.
- Desarrollar un módulo de pagos, para espacios o equipos que requieran mantenimiento o reparación
- Desarrollar una aplicación móvil nativa, que se conecte a la base de datos del sistema de reservas brindando las mismas funcionalidades que aplicación web.

Referencias bibliográficas

Barranco, J. (2001). *Metodología del análisis estructurado de sistemas*. Madrid: España. Servicio de publicaciones de la Universidad Pontificia Comillas.

Beck, K. (2004). *Extreme Programming Explained [Programación extrema explicada] (2a. ed.)*. Nueva Jersey, Estados Unidos: Addison-Wesley Professional.

Duggar, V. (2008). *Annotating Decision Analyses Using Semantic Web Technologies [Anotando análisis de decisiones usando tecnologías de semantic web]*. ProQuest.

Frain, B. (2012). *Responsive Web Design with HTML5 and CSS3 [Diseño web responsivo con HTML5 y CSS3]*. Birmingham, Reino Unido: Packt Publishing Ltd.

García, C. (2003). *Diseño web para tod@s I*. Barcelona: España: Icaria Editorial.

Hernandez, J. (2014). *Análisis y desarrollo web*.

Heurtel, O. (2011). *PHP 5.3: Desarrollar un sitio Web dinámico e interactivo*. Barcelona, España. Ediciones ENI.

Kendall, K & Kendall, J. (2005). *Análisis y diseño de sistemas*. México: Pearson Educación.

Kent, B. (2002). *Una explicación de la programación extrema. Aceptar el cambio*. Madrid, España: Iberoamericana de Méjico.

Kruchten, P. (2004). *The Rational Unified Process: An Introduction [El proceso racional unificado: Una introducción] (3a. ed.)*. Boston, Estados Unidos: Addison-Wesley Professional.

Lafosse, J. (2010). *Struts 2: El framework de desarrollo de aplicaciones Java EE*. Barcelona: España. Ediciones ENI.

Lerdorf, R., MacIntyre, P. & Tatroe K. (2013) *Programming PHP [Programando PHP] (3ª. Ed.)*. California, Estados Unidos: O'Reilly Media.

Luján, S. (2002). *Programación de aplicaciones web: historia, principios básicos y clientes web*. Alicante, España: Editorial Club Universitario.

Marchesi, M. & Succi, G. (2003). *Extreme Programming and Agile Processes in Software Engineering [Programación extrema y procesos ágiles en ingeniería del software]*. Alemania: Springer Science & Business Media.

Navas, A. (2012). *Guía completa de CSS3*. Autoedición.

Niño, J. (2010). *Aplicaciones web*. Editex.

Pressman, R. (2010). *Ingeniería del software: un enfoque práctico (7a. ed.)*. DF, México: McGrawHill.

Roebuck, K. (2012). *Agile software development: High-impact Strategies - What You Need to Know: Definitions, Adoptions, Impact, Benefits, Maturity, Vendors [Desarrollo ágil de software: Estrategias de alto impacto – Lo que necesitas saber: definiciones, aprobaciones, impacto, beneficios, madurez, vendedores]*. Emereo Publishing.

Rusty E. & Scott W. (2004). *XML in a nutshell [XML en una cascara de nuez](3a. ed.)*. California, Estados Unidos: O'Reilly Media, Inc.

Smashing magazine (2012). *Responsive design [Diseño responsivo]*. Freiburg, Alemania: Smashing magazine.

Rossmann, B. (2012). *Application Lifecycle Management - Activities, Methodologies, Disciplines, Tools, Benefits, ALM Tools and Products [Administración del ciclo de vida de la aplicación – Actividades, metodologías, disciplinas, herramientas, beneficios, herramientas ALM y productos]*. Newstead: Australia. Emereo Publishing.

Sommerville, I. (2005). *Ingeniería del software (7a. Ed.)*. Madrid, España: Pearson educación.

Woods, D. (2012). *HTML5 and CSS: Complete [HTML5 y CSS: Completo]*. Bostón, Estados Unidos: Cengage Learning.

Walsh, N. (1998). *A Technical Introduction to XML [Una introducción técnica a XML]*. ArborText.