

Universidad de las Ciencias Informáticas

Facultad 6



Trabajo de Diploma para optar por el título de
Ingeniero Informático

Título:

Sistema de reservación de locales docentes en la Facultad 1.

Autor: Pedro Vega Martínez

Tutor(es): Ing. Dasiel Alberto Pérez Suárez.

Ing. Danny Hernández Hernández.

La Habana, Junio 2015

Frase



Hay una fuerza motriz más poderosa que el vapor, la electricidad y la energía atómica: la voluntad.

Declaración de autoría

Declaramos ser autores de la presente tesis que tiene por título: Sistema de reservación de locales docentes en la Facultad 1 y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo. Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

Pedro Vega Martínez

Firma del autor

Ing. Dasiel Alberto Pérez Suárez

Firma del tutor

Ing. Danny Hernández Hernández.

Firma del tutor

Datos de Contacto

Tutor: Ing. Dasiel Alberto Pérez Suárez.

Graduado en la Universidad de las Ciencias Informáticas en el año 2008. Actualmente se desempeña como Jefe del Departamento de Tecnología de la Facultad 1.

Correo: daperez@uci.cu

Tutor: Ing. Danny Hernández Hernández.

Graduado en la Universidad de las Ciencias Informáticas en junio del 2013. Actualmente se desempeña como especialista en el Centro de Soporte.

Correo: danny@uci.cu

Agradecimientos

A mis padres por todo el amor y apoyo incondicional que me han brindado, por estar siempre ahí, dispuestos a ayudarme en todo momento, sin dudas, la razón por la cual hoy estoy aquí.

A mi abuela Romelia por el amor que siempre me ha dado, por su ternura, por preocuparse por mí y por estar a mi lado en todo momento.

A Junet por darme fuerzas y amor para continuar, por haber estado a mi lado y por confiar en mí hasta el último momento.

A mis compañeros de aula con los cuales he compartido momentos inolvidables.

A mis tutores Danny y Dasiel por guiarme a lo largo del desarrollo de este trabajo.

A todos los maestros y profesores que desde mi infancia han contribuido en mi formación y a todos los que me forjaron como profesional durante 6 años.

Finalmente gracias a todos aquellos que de algún modo aportaron su granito de arena en mi desarrollo profesional.

A todos Muchas Gracias.

Pedro Vega Martínez.

Dedicatoria

Dedico este trabajo, mi carrera y lo que soy:

Primero que todo a mí querida hija Aitana por prestarme el tiempo que le pertenecía.

A mis padres, que han sido la voluntad, la perseverancia y el amor durante toda mi vida.

Gracias por su apoyo y confianza.

A mi hermana Yeilyn por ser la mejor hermana mayor del mundo.

A mi abuelita Romelia por ser mi segunda madre.

A mi sobrinita que es como mi otra niña.

Esta tesis va dedicada a ustedes por ser las personas más importantes de mi vida.

Pedro Vega Martínez.

Resumen

La reservación de locales docentes en la Universidad de las Ciencias Informáticas es una de las actividades fundamentales en el proceso de enseñanza y aprendizaje. Los locales docentes son utilizados para disímiles actividades de tipo docentes y extracurriculares. Por ello se desarrolla en la Facultad 1, un sistema que sea capaz de gestionar las reservaciones de estos, que tenga en cuenta sus características y las actividades a desarrollar en los mismos. En el desarrollo del sistema de reservación de locales docentes se utiliza la metodología XP para guiar el proceso de desarrollo de software durante sus diferentes fases. Como resultado se obtuvo un sistema informático capaz de gestionar las reservaciones de los locales docentes y que además permite al usuario consultar las características tecnológicas de cada uno. De esta forma se evita que coincidan reservaciones para un mismo local de diferentes actividades a la misma hora. El sistema fue validado aplicándole pruebas de caja negra. El resultado de las pruebas realizadas mostró el buen funcionamiento del sistema desarrollado y su capacidad para solucionar el problema planteado.

Palabras claves: sistema de reservación, reservación, locales docentes.

Abstract

Reservation of school rooms at the University of Informatics Sciences is one of the key activities in the process of teaching and learning. These places are used for different teaching and extracurricular activities. Therefore in School 1 it is being developed, a system capable of managing these reservations, which take into account the features of the room and the activities that can be carried out in them. To develop the system for reserving the school rooms it is used the XP methodology to guide the software development process in its different phases. As a result, a computer system capable of handling the reservations of school rooms and also allows the user to consult each technological characteristics. This will prevent a single match reserves for different activities at the same time. The system was validated by applying black box testing. The result of the tests showed the proper functioning of the developed system and its ability to solve problems.

Keywords: reservation system, reservation, school rooms.

Índice

Introducción	- 1 -
Capítulo 1: Aspectos teóricos acerca de sistemas de reservación de locales, tecnologías para su desarrollo	- 6 -
Introducción	- 6 -
1.1. Marco teórico.....	- 6 -
1.1.1. Servicio.....	- 6 -
1.1.2. Reservación.....	- 6 -
1.1.3. Sistemas de reservaciones	- 6 -
1.1.4. Sistemas de reservaciones en línea.....	- 7 -
1.2. Soluciones existentes	- 7 -
1.2.1. Valoración de los sistemas estudiados.....	- 9 -
1.3. Tecnologías, herramientas, metodología y lenguajes definidos para la solución.....	- 10 -
1.3.1. Lenguaje de modelado.....	- 10 -
1.3.2. Herramienta CASE.....	- 11 -
1.3.3. Servidor web.....	- 11 -
1.3.4. Lenguajes de programación.....	- 12 -
1.3.5. Framework de desarrollo.....	- 14 -
1.3.6. Entorno de Desarrollo Integrado	- 16 -
1.3.7. Sistema Gestor de Base de Datos	- 17 -
1.3.8. Metodología de desarrollo de software.....	- 18 -
Conclusiones parciales	- 19 -
Capítulo 2: Descripción del sistema de reservación de locales docentes en la Facultad 1	- 20 -
Introducción	- 20 -
2.1. Modelo conceptual	- 20 -
2.2. Flujo actual del proceso.....	- 22 -
2.3. Propuesta de solución	- 22 -
2.3.1. Módulo de Gestión.....	- 23 -
2.3.2. Módulo de Reservación	- 23 -
2.3.3. Módulo de Seguridad.....	- 23 -

2.3.4. Módulo de Reporte.....	- 23 -
2.4 Lista de reserva del producto.....	- 23 -
2.5. Planificación	- 26 -
2.5.1. Historias de Usuario.....	- 27 -
2.5.2. Estimación de esfuerzo por HU.....	- 30 -
2.5.3. Iteraciones	- 32 -
2.5.4. Plan de duración de las iteraciones.....	- 32 -
2.5.5. Plan de entrega.....	- 33 -
2.6. Diseño del sistema	- 34 -
2.6.1. Tarjetas CRC	- 34 -
2.6.2. Diseño de base de datos.....	- 37 -
2.6.3. Patrones de arquitectura de software.....	- 40 -
2.6.4. Patrones de diseño	- 41 -
2.6.5 Diagrama de despliegue	- 44 -
Conclusiones parciales	- 45 -
Capítulo 3: Implementación y pruebas al sistema de reservación de locales docentes en la Facultad 1	- 47 -
Introducción	- 47 -
3.1. Implementación	- 47 -
3.1.1 Tareas de ingeniería	- 47 -
3.1.2. Estándares de codificación	- 50 -
3.2. Pruebas de software	- 52 -
3.2.1. Pruebas de aceptación	- 52 -
3.2.2 Pruebas de carga y estrés	- 58 -
Conclusiones parciales	- 60 -
Conclusiones	- 61 -
Recomendaciones	- 62 -
Referencias bibliográficas	- 63 -
Anexos.....	- 66 -

Índice de tablas

Tabla 1. Lista de reserva del producto.....	- 24 -
Tabla 2. HU Autenticar usuario.....	- 29 -
Tabla 3. HU Insertar local.....	- 29 -
Tabla 4. HU Insertar reservación.....	- 30 -
Tabla 5. Estimación de esfuerzo por HU.....	- 30 -
Tabla 6. Plan de duración de las iteraciones.....	- 32 -
Tabla 7. Plan de entrega.....	- 33 -
Tabla 8. Tarjeta CRC Tipo de local.....	- 35 -
Tabla 9. Tarjeta CRC Local.....	- 35 -
Tabla 10. Tarjeta CRC Tipo de medio.....	- 35 -
Tabla 11. Tarjeta CRC Medio.....	- 35 -
Tabla 12. Tarjeta CRC Reservación.....	- 36 -
Tabla 13. Tarjeta CRC Usuario.....	- 36 -
Tabla 14. Tarjeta CRC Rol.....	- 36 -
Tabla 15. Tarjeta CRC Menús del sistema.....	- 36 -
Tabla 16. TI Autenticar usuario.....	- 48 -
Tabla 17. TI Insertar reservación.....	- 48 -
Tabla 18. TI Listar reservaciones.....	- 49 -
Tabla 19. TI Modificar reservación.....	- 49 -
Tabla 20. TI Eliminar reservación.....	- 50 -
Tabla 21. Prueba 1: Autenticar con datos no válidos.....	- 53 -
Tabla 22. Prueba 20: Insertar una nueva reservación.....	- 53 -
Tabla 23. Resumen de no conformidades.....	- 57 -
Tabla 24. HU Insertar tipo de local.....	- 66 -
Tabla 25. HU Listar tipos de locales.....	- 66 -
Tabla 26. HU Modificar tipo de local.....	- 67 -
Tabla 27. HU Eliminar tipo de local.....	- 67 -
Tabla 28. HU Listar locales.....	- 68 -
Tabla 29. HU Modificar local.....	- 68 -
Tabla 30. HU Eliminar local.....	- 69 -
Tabla 31. HU Insertar tipo de medio.....	- 69 -
Tabla 32. HU Listar tipos de medios.....	- 69 -
Tabla 33. HU Modificar tipo de medio.....	- 70 -
Tabla 34. HU Eliminar tipo de medio.....	- 70 -
Tabla 35. HU Insertar medios a los locales.....	- 71 -
Tabla 36. HU Listar medios de locales.....	- 71 -
Tabla 37. HU Modificar medios de locales.....	- 71 -

Tabla 38. HU Eliminar medios de locales.....	- 72 -
Tabla 39. HU Mostrar horario por día, semana o mes.	- 72 -
Tabla 40. HU Listar reservaciones.....	- 73 -
Tabla 41. HU Modificar reservación.	- 73 -
Tabla 42. HU Eliminar reservación.	- 74 -
Tabla 43. HU Listar usuarios.	- 74 -
Tabla 44. HU Modificar usuario.....	- 75 -
Tabla 45. HU Eliminar usuario.....	- 75 -
Tabla 46. HU Insertar rol.	- 76 -
Tabla 47. HU Listar rol.	- 76 -
Tabla 48. HU Modificar rol.....	- 77 -
Tabla 49. HU Eliminar rol.	- 77 -
Tabla 50. HU Mostrar reportes.	- 78 -
Tabla 51. HU Insertar menú del sistema.	- 78 -
Tabla 52. HU Listar menús del sistema.	- 79 -
Tabla 53. HU Modificar menú del sistema.....	- 79 -
Tabla 54. HU Eliminar menú del sistema.....	- 80 -
Tabla 55. Descripción de la tabla reserva.	- 80 -
Tabla 56. Descripción de la tabla tipo_actividad.	- 81 -
Tabla 57. Descripción de la tabla horario.	- 81 -
Tabla 58. Descripción de la tabla local.....	- 81 -
Tabla 59. Descripción de la tabla tipo_local.....	- 82 -
Tabla 60. Descripción de la tabla tipo_local_tipo_medio.	- 82 -
Tabla 61. Descripción de la tabla tipo_medio.	- 82 -
Tabla 62. Descripción de la tabla tipo_medio_características.	- 82 -
Tabla 63. Descripción de la tabla características.	- 83 -
Tabla 64. Descripción de la tabla medio_características.	- 83 -
Tabla 65. Descripción de la tabla medio.....	- 83 -
Tabla 66. Descripción de la tabla local_medio.....	- 84 -
Tabla 67. Descripción de la tabla usuario.....	- 84 -
Tabla 68. Descripción de la tabla autenticación_activa.....	- 85 -
Tabla 69. Descripción de la tabla rol.	- 85 -
Tabla 70. Descripción de la tabla rol_menú.	- 85 -
Tabla 71. Descripción de la tabla menú.....	- 86 -
Tabla 72. Descripción de la tabla menú_acción.	- 86 -
Tabla 73. Descripción de la tabla acción.	- 87 -
Tabla 74. Descripción de la tabla acción_url.....	- 87 -
Tabla 75. TI Insertar tipo de local.....	- 88 -
Tabla 76. TI Listar tipos de locales.	- 88 -

Tabla 77. TI Modificar tipos de locales.....	- 89 -
Tabla 78. TI Eliminar tipo de local.....	- 89 -
Tabla 79. TI Insertar local.....	- 90 -
Tabla 80. TI Listar locales.....	- 90 -
Tabla 81. TI Modificar local.....	- 91 -
Tabla 82. TI Eliminar local.....	- 91 -
Tabla 83. TI Insertar tipo de medio.....	- 92 -
Tabla 84. TI Listar tipos de medios.....	- 92 -
Tabla 85. TI Modificar tipo de medio.....	- 93 -
Tabla 86. TI Eliminar tipo de medio.....	- 93 -
Tabla 87. TI Insertar medios a locales.....	- 94 -
Tabla 88. TI Listar medios de locales.....	- 94 -
Tabla 89. TI Modificar medios de locales.....	- 95 -
Tabla 90. TI Eliminar medios de locales.....	- 95 -
Tabla 91. TI Mostrar horario por día, semana o mes.....	- 96 -
Tabla 92. TI Listar usuarios.....	- 96 -
Tabla 93. TI modificar usuario.....	- 96 -
Tabla 94. TI Eliminar usuario.....	- 97 -
Tabla 95. TI Insertar rol.....	- 97 -
Tabla 96. TI Listar rol.....	- 98 -
Tabla 97. TI Modificar rol.....	- 98 -
Tabla 98. TI Eliminar rol.....	- 99 -
Tabla 99. TI Mostrar reportes.....	- 99 -
Tabla 100. TI Insertar menú del sistema.....	- 100 -
Tabla 101. TI Listar menús del sistema.....	- 100 -
Tabla 102. TI Modificar menú del sistema.....	- 101 -
Tabla 103. TI Eliminar menú del sistema.....	- 101 -
Tabla 104. Prueba 2: Autenticar con datos válidos.....	- 102 -
Tabla 105. Prueba 3: Insertar un nuevo tipo de local.....	- 102 -
Tabla 106. Prueba 4: Listar todos los tipos de locales.....	- 103 -
Tabla 107. Prueba 5: Modificar un tipo de local registrado en el sistema.....	- 103 -
Tabla 108. Prueba 6: Eliminar un tipo de local registrado.....	- 104 -
Tabla 109. Prueba 7: Insertar un nuevo local.....	- 104 -
Tabla 110. Prueba 8: Listar todos los locales insertados en el sistema.....	- 105 -
Tabla 111. Prueba 9: Modificar local.....	- 105 -
Tabla 112. Prueba 10: Eliminar un local registrado.....	- 106 -
Tabla 113. Prueba 11: Insertar un nuevo tipo de medio.....	- 106 -
Tabla 114. Prueba 12: Listar todos los tipos de medios insertados en el sistema.....	- 107 -
Tabla 115. Prueba 13: Modificar un tipo de medio registrado en el sistema.....	- 107 -

Tabla 116. Prueba 14: Eliminar un tipo de medio registrado.....	- 108 -
Tabla 117. Prueba 15: Insertar un nuevo medio a un local.	- 108 -
Tabla 118. Prueba 16: Listar los medios del local seleccionado.	- 109 -
Tabla 119. Prueba 17: Modificar un medio del local seleccionado.....	- 109 -
Tabla 120. Prueba 18: Eliminar un medio de un local.	- 110 -
Tabla 121. Prueba 19: Mostrar en el horario todas las reservaciones.....	- 110 -
Tabla 122. Prueba 21: Listar mis reservaciones.	- 111 -
Tabla 123. Prueba 22: Listar todas las reservaciones.....	- 111 -
Tabla 124. Prueba 23: Modificar mis reservaciones.	- 112 -
Tabla 125. Prueba 24: Modificar cualquier reservación.	- 113 -
Tabla 126. Prueba 25: Eliminar mi reservación.	- 113 -
Tabla 127. Prueba 26: Eliminar cualquier reservación.	- 114 -
Tabla 128. Prueba 27: Listar los usuarios del sistema.	- 114 -
Tabla 129. Prueba 28: Modificar un usuario del sistema.	- 115 -
Tabla 130. Prueba 29: Eliminar un usuario del sistema.	- 115 -
Tabla 131. Prueba 30: Insertar un nuevo rol.....	- 116 -
Tabla 132. Prueba 31: Listar los roles del sistema.	- 116 -
Tabla 133. Prueba 32: Modificar un rol en el sistema.	- 117 -
Tabla 134. Prueba 33: Eliminar un rol del sistema.....	- 117 -
Tabla 135. Prueba 34: Mostrar reportes de los laboratorios.	- 118 -
Tabla 136. Prueba 35: Insertar un nuevo menú.	- 118 -
Tabla 137. Prueba 36: Listar los menús del sistema.	- 119 -
Tabla 138. Prueba 37: Modificar un menú del sistema.	- 119 -
Tabla 139. Prueba 38: Eliminar un menú del sistema.	- 120 -

Índice de figuras

Fig 1. Modelo conceptual	- 21 -
Fig 2. Diseño de base de datos del esquema de reservación.	- 38 -
Fig 3. Diseño de base de datos del esquema de seguridad.	- 39 -
Fig 4. Modelo Vista Controlador.....	- 40 -
Fig 5. Diagrama de despliegue.	- 45 -
Fig 7. Total de funcionalidades probadas y no conformidades detectadas por iteración.	- 55 -
Fig 8. Resultados de las pruebas de carga y estrés.....	- 59 -

Introducción

El uso de las Tecnologías de la Información y las Comunicaciones (TIC) en las empresas, ha permitido la obtención de beneficios como el mejoramiento de las operaciones y los procesos asociados a estas. Las TIC permiten conocer las necesidades de los clientes, llegar a un mayor número de clientes y mercados, además de optimizar sus recursos aumentando su eficiencia. Uno de los ejemplos que se pueden mencionar sobre el uso de las TIC en las instituciones son los sistemas de reservaciones.

Cuba no queda al margen en el uso de estas tecnologías y cuenta con numerosos sistemas de reservaciones en línea. Muchos de estos sistemas son utilizados para la reservación de hoteles, renta de autos, reservación de paseos con guía turística, entre otros.

La Universidad de las Ciencias Informáticas (UCI) surge al calor de la batalla de ideas y es un centro docente en el que se desarrollan soluciones informáticas para clientes nacionales e internacionales. Tiene la misión de formar profesionales comprometidos con su Patria y altamente calificados en la rama de la Informática, producir aplicaciones y servicios informáticos, a partir de la vinculación estudio-trabajo como modelo de formación y servir de soporte a la industria cubana de la informática (1). Por su estructura posee un conjunto de sistemas que se encargan de la reservación de los diferentes servicios brindados en el campus. Estos sistemas permiten agilizar los servicios brindados en la institución.

La UCI contempla dentro de su estructura organizativa a varias facultades y estas a su vez tienen asociados locales como: aulas, salones de conferencias y laboratorios informáticos donde se realizan actividades docentes y científicas, cuyo uso se planifica semestralmente. En la Facultad 1 la mayoría de las actividades docentes se planifican en el Vicedecanato de Formación por la planificadora, mediante el sistema de escritorio denominado Horario UCI. Sin embargo, existen actividades como cursos de postgrados, consultas, segundas convocatorias de exámenes, talleres y seminarios científicos, entre otras, que no se pueden planificar utilizando el sistema mencionado. Para utilizar los locales docentes en las actividades antes mencionadas, es necesario reservar los mismos a través de la propia planificadora pero de forma manual, convirtiéndolo en un proceso lento y engorroso debido a la gran cantidad de solicitudes de locales que se realizan y las diferentes prioridades que tienen las mismas.

Antes de reservar un local se recomienda conocer su estado tecnológico. En el caso de los laboratorios, la cantidad de puestos de trabajo disponibles, las características de las computadoras y el estado de los aires

acondicionados. De las aulas y salones de conferencia es importante saber las características de la computadora y si los televisores funcionan correctamente. Conocer estas particularidades a la hora de reservar un local influye en gran medida en que las actividades que se planifiquen se realicen con la calidad requerida.

En dependencia de la magnitud de la actividad se puede reservar un local que cumpla con las condiciones de trabajo necesarias, información que es manejada por el Departamento de Tecnología de la propia facultad. Si en el momento en que se solicita la reservación no se puede contactar al personal de dicho departamento se corre el riesgo de reservar un local que no cumpla con las necesidades del usuario que lo reserva, lo que trae consigo que se afecte la calidad de la actividad y el prestigio de la facultad en la atención tanto a los usuarios internos como externos. De igual forma, el hecho de que el proceso se realice de forma manual también propicia que se cometan errores en la planificación debido al extravío o mal manejo de los documentos utilizados para ese fin. Es posible entonces que se reserven dos o más actividades en el mismo local y a la misma hora o que no se informe al personal responsable de los locales para que estos se encuentren listos para las actividades, lo que causa molestias en los usuarios y puede provocar retrasos en las actividades.

La Facultad 1 hace un uso elevado de los locales disponibles fuera del horario docente. Por lo que es necesario que las reservaciones se realicen de forma rápida y eficiente para garantizar una buena calidad en las actividades a desarrollar.

A partir de la situación problemática planteada se define el siguiente **problema de la investigación**: ¿Cómo contribuir a la gestión de reservaciones de locales docentes en la Facultad 1 logrando mayor organización y control en el proceso?

Dado el problema anterior se define como **objeto de estudio** el proceso de reservación de locales, enmarcado en el **campo de acción** sistemas informáticos de reservación de locales docentes en línea. El **objetivo general** de la investigación es desarrollar un sistema para la reservación de locales docentes en la Facultad 1, teniendo en cuenta el estado tecnológico y la disponibilidad del local a reservar, que permita mayor organización y control del proceso.

Se definen los siguientes **objetivos específicos**:

1. Elaborar el marco teórico de la investigación mediante el estudio y análisis de sistemas para la

reservación de locales docentes y las herramientas empleadas para su desarrollo.

2. Diseñar el sistema para la reservación de locales docentes en la Facultad 1.
3. Implementar el sistema para la reservación de locales docentes en la Facultad 1.
4. Validar técnicamente el sistema para la reservación de locales docentes en la Facultad 1 a través de pruebas funcionales de aceptación.

Idea a defender: la implementación de un sistema que gestione el proceso de reservación de locales docentes en la Facultad 1, contribuye a lograr una mayor organización y control del proceso.

Para dar cumplimiento a los objetivos se propone la realización de las siguientes **tareas**:

1. Análisis de sistemas de reservación de locales.
2. Definición de las características y funcionalidades del sistema para la reservación de locales docentes en la Facultad 1.
3. Análisis de las principales metodologías, tecnologías y herramientas a utilizar para el desarrollo de la solución propuesta.
4. Construcción de los artefactos requeridos en la Ingeniería de Software del sistema, comprendidos en las fases de la metodología seleccionada.
5. Definición de la arquitectura del sistema como base para el desarrollo de la solución.
6. Implementación de las clases modelos, vistas y controladoras para el desarrollo del sistema.
7. Elaboración de los casos de prueba al sistema para la reservación de locales docentes en la Facultad 1.
8. Aplicación de pruebas al sistema para la reservación de locales docentes en la Facultad 1.

La realización de la investigación está dirigida mediante **métodos científicos** que fueron aplicados durante el desarrollo de la misma:

Teóricos:

- **Histórico-lógico:** Este método es utilizado al investigar la evolución y desarrollo referente al proceso que se realiza para la reservación de locales, a la trayectoria, comportamiento y estado actual del mismo, además de estructurar la documentación investigada de manera organizada, para así tener un mejor entendimiento de la misma.
- **Analítico-Sintético:** En el análisis de diversas metodologías, herramientas y documentos relacionados con el objeto de estudio de los cuales se extraen los rasgos distintivos más importantes relacionados con el desarrollo del proceso de Gestión de la Reservación.
- **Modelación:** Este método se utilizó con el objetivo de materializar la representación abstracta de determinadas características del sistema mediante la construcción de diagramas y modelos a lo largo del desarrollo de la investigación.

Empíricos:

- **Observación:** Permite analizar cada fase del proceso de reservación de los locales docentes, observando cada tarea que se realiza para tomar experiencia de esta y poder aplicarla en todas las demás. También admite llevar a cabo la observación continua de los requisitos y objetivos que son necesarios para realizar satisfactoriamente el sistema informático.

Técnicas de recolección de la información:

- **Entrevista:** Se utilizó para obtener información referente al proceso de reservación de los locales docentes, a través de un conjunto de preguntas planificadas y orientadas a obtener conocimientos específicos sobre el tema que se investiga. En este caso se realizaron entrevistas a la planificadora, al Jefe del Departamento de Tecnología y al responsable de hardware de los laboratorios, todos estos de la Facultad 1.

El presente trabajo se encuentra **estructurado** en 3 capítulos, los cuales resumen la siguiente información:

CAPÍTULO 1: “ASPECTOS TEÓRICOS ACERCA DE SISTEMAS DE RESERVACIÓN DE LOCALES, TECNOLOGÍAS PARA SU DESARROLLO”

En este capítulo se exponen los principales conceptos asociados al dominio del problema, se realiza un estudio de homólogos, se caracteriza el estado actual de las tecnologías escogidas para la realización de la investigación. Se tratan los elementos teóricos que sustentan y fundamentan los objetivos del trabajo.

CAPÍTULO 2: “DESCRIPCIÓN DEL SISTEMA DE RESERVACIÓN DE LOCALES DOCENTES EN LA FACULTAD 1”

Describe las características del sistema a implementar. Comprende las historias de usuario (HU) y lista de reserva del producto. Se identifican las funcionalidades que compondrán el sistema y que son la base para su implementación, así como las personas que se beneficiarán directamente de su uso. Se obtienen las tarjetas CRC (Clase-Responsabilidad-Colaboración). Se muestra el diseño de la base de datos y se definen la arquitectura del sistema y los patrones de diseño.

CAPÍTULO 3: “IMPLEMENTACIÓN Y PRUEBAS AL SISTEMA DE RESERVACIÓN DE LOCALES DOCENTES EN LA FACULTAD 1”

Se muestran los estándares de codificación empleados. Se diseñan los casos de prueba para comprobar las diferentes funcionalidades del sistema. Se aplican pruebas que corroboren el correcto funcionamiento del sistema a lo largo del proceso de desarrollo y una vez finalizado este. Las pruebas se ejecutan dentro de este proceso y con circunstancias previamente identificadas para posteriormente documentar los resultados obtenidos.

Capítulo 1: Aspectos teóricos acerca de sistemas de reservación de locales, tecnologías para su desarrollo

Introducción

En el presente capítulo se realiza un análisis de los sistemas de reservación de locales estudiados. Se realiza una descripción del estado del arte de los distintos sistemas desarrollados. Se realiza un análisis crítico para detectar las principales ventajas y desventajas de estos sistemas. Se define la metodología de desarrollo de software a utilizar y las herramientas para el análisis, diseño e implementación del sistema.

1.1. Marco teórico

1.1.1. Servicio

Se refiere a una prestación para satisfacer las necesidades de las personas que no representa la producción de bienes materiales; la cual puede ser realizada bajo cierto control y regulación por la organización o entidad por la que es desarrollado con el fin de satisfacer a la sociedad (2). A partir de lo antes planteado se puede determinar que la posibilidad de reservación de un local docente se puede categorizar como la prestación de un servicio debido a que están encaminados a la satisfacción de las personas de la Facultad 1 de la UCI.

1.1.2. Reservación

Según el Diccionario de la Real Academia de la Lengua Española, una reservación es la acción de destinar una cosa o un lugar para un uso exclusivo o para una persona determinada (3). En el caso de la presente investigación una reservación es destinar un local docente a una persona que lo necesite durante el tiempo que estime necesario para desarrollar una determinada actividad.

1.1.3. Sistemas de reservaciones

Los sistemas de reservaciones son un medio para acceder a los servicios de una empresa o institución en general durante un tiempo determinado con antelación a que estos sean brindados. Estos sistemas pueden aplicarse para reservaciones de boletos de avión, boletos del cine, teatro, restaurantes, citas médicas, hoteles y habitaciones, transportaciones masivas y personales, eventos mundiales y locales, servicios a domicilios y mucho más (4). La presente investigación está enfocada a sistemas de reservación de los locales docentes de la Facultad 1 de la Universidad de las Ciencias Informáticas.

1.1.4. Sistemas de reservaciones en línea.

Teniendo en cuenta los conceptos anteriores se puede decir que los sistemas de reservaciones en línea son aplicaciones que permiten realizar reservaciones de los recursos en tiempo real a través de redes informáticas. Estos brindan constantemente mensajes de confirmación para tener informados a los clientes de las acciones realizadas en los mismos y de los cambios que puedan existir. Generalmente son de interfaces amigables y sencillas diseñadas de tal forma que el usuario pueda interactuar con ellas fácilmente. Este concepto se puede tomar como referencia a la hora de buscar las soluciones existentes a nivel mundial, que contribuyan a la implementación de la solución propuesta. En este caso serían los sistemas de reservación en línea de locales docentes.

1.2. Soluciones existentes

Durante la investigación se consultaron varios sistemas de reservaciones de locales docentes pertenecientes a instituciones tanto nacionales como extranjeras, con el objetivo de conocer sus principales funcionalidades, los elementos comunes de cada uno de ellos y tomarlos como punto de análisis a la hora de diseñar e implementar la solución propuesta.

SSRA (Sistema de Solicitud de Reserva de Aulas): Es una aplicación web desarrollada en la Universidad de Sevilla, España, para la Facultad de Farmacia, que permite al usuario consultar las reservaciones de las aulas gestionadas por la facultad y a los usuarios autorizados solicitar nuevas reservaciones que, tras ser aprobadas, se reflejarán en el sistema.

El sistema gestiona un conjunto de aulas, enmarcadas en distintas áreas (aulas generales, laboratorios). Cada aula tiene diariamente trece horas disponibles (desde las ocho de la mañana hasta las nueve de la noche). Una reservación se realiza sobre una o varias aulas, en uno o varios días y en alguna o algunas de las horas de disponibilidad. La reserva se solicita inicialmente por parte de un usuario autorizado, y se convierte en definitiva cuando un administrador la valida (5). El usuario tiene la posibilidad de consultar las reservaciones dado un día específico así como consultar detalles de estas. El SSRA fue implementado con el lenguaje del lado del servidor PHP¹.

Sistema para la Reservación de Tiempos de Máquina de los Laboratorios en la UCI (6): De acuerdo a la documentación de este sistema, permitía a los estudiantes reservar un puesto de trabajo en el horario de

¹ Del inglés Hypertext Preprocessor

Capítulo 1: Aspectos teóricos acerca de sistemas de reservación de locales, tecnologías para su desarrollo.

la noche. Cada facultad asignaba a profesores que podían reservar un laboratorio para realizar alguna actividad. Los profesores asignados reservaban con prioridad en un horario determinado y después lo hacían los estudiantes. Al finalizar el horario de reservación se emitía un reporte con la cantidad de estudiantes y profesores que habían reservado. Este sistema no brinda información acerca del estado del equipamiento tecnológico en los laboratorios. En cuanto a la interfaz gráfica solo se muestran algunos prototipos, no el sistema implementado. Cabe destacar que el sistema se implementó utilizando tecnologías libres: Apache como servidor web, PHP como lenguaje de programación del lado del servidor y MySQL como sistema gestor de bases de datos. El sistema no se está utilizando actualmente.

Gestión y control de los laboratorios de la Facultad 7 (7): Este sistema permitía a los usuarios, en este caso estudiantes, reservar un puesto de trabajo para tiempo de máquina. El control del estado del hardware se realiza en el momento de recibo del técnico de laboratorio con el alumno ayudante, por lo que los puestos se asignaban sin tener en cuenta este importante dato. El proceso de asignación solo comprueba que exista la misma cantidad de estudiantes y puestos de trabajo, dejando fuera el control de los equipos fuera de servicio por un motivo determinado. Este se implementó utilizando tecnologías libres: Apache como servidor web, PHP como lenguaje del lado del servidor y MySQL como sistema gestor de bases de datos. En la documentación del sistema no se plasman tanto prototipos de interfaz como pantallas del sistema, aunque se hace referencia a algunas buenas prácticas a tener en cuenta para la confección de esta y que resulte agradable al usuario. El sistema no se está utilizando actualmente.

Sistema Integral de Gestión de los Laboratorios. Módulo de reservación de tiempo de máquina en la UCI (8): Este sistema brinda a los usuarios solicitar un puesto de trabajo para tiempo de máquina. El técnico responsable debe conocer el estado del equipamiento del laboratorio, por lo que se encarga de asignar los puestos disponibles y aceptar o rechazar la solicitud de reservación. El usuario es quien gestiona su reservación, no existiendo una contraparte que pueda cancelarlas en caso de presentarse un evento de mayor importancia. Este se implementó utilizando tecnologías libres: Apache como servidor web, PHP como lenguaje del lado del servidor, PostgreSQL como sistema gestor de bases de datos y ExtJs para la creación de la interfaz gráfica de usuario, que permite construir interfaces gráficas para la web con aspecto similar a aplicaciones de escritorio.

1.2.1. Valoración de los sistemas estudiados

Como se ha visto hasta el momento, existen diferentes sistemas de reservación de locales docentes. Estos sistemas varían de acuerdo a la finalidad de su uso.

El sistema SSRA está enfocado a la reservación de locales docentes. Las siguientes características de dicho sistema se tomarán en cuenta para la futura implementación del sistema:

- Se muestra un calendario para conocer los locales reservados dado una fecha específica, pudiendo filtrar los resultados por día, semana o mes.
- El calendario de reservaciones permite conocer las características de estas al seleccionar alguna en particular.
- El sistema gestiona diversos tipos de locales, por lo que se brinda la posibilidad de elegir entre estos mediante una lista desplegable.
- Se puede seleccionar el tipo de actividad a realizar en el local y el motivo de la reservación.
- Se delimitan las funciones para cada tipo de usuario.

Las funcionalidades anteriores se tendrán en cuenta para el diseño e implementación de la solución propuesta, ya que con su desarrollo se resuelve una gran parte del problema. Sin embargo, la interfaz visual hace uso de colores fuertes y variados en la presentación de las reservaciones en el calendario, por lo que estos elementos se descartan para la implementación de la solución. Además, no tiene en cuenta las características de los locales ni de los medios con que cuentan, requisitos de gran importancia para el sistema a desarrollar.

En el caso de los sistemas desarrollados en la UCI, se centran en ocupar los puestos de trabajo en los laboratorios pasando por alto el estado técnico del equipamiento en la mayoría de los casos. Este aspecto cobra gran relevancia, sin este el servicio de reservación no sería tan preciso como se desea, siendo esta una de las principales características del sistema a implementar: ofrecerle al usuario de forma rápida y precisa la condición de los medios que va a utilizar.

Vale destacar el uso de tecnologías libres para la realización de estos trabajos, valorándose las siguientes para la implementación de la solución:

- Apache como servidor web.

- PHP como lenguaje de programación del lado del servidor.
- PostgreSQL como Sistema Gestor de Base de Datos.

Otra característica a tener en cuenta que ofrecieron estos sistemas es que solamente pueden acceder a estos los usuarios autorizados para realizar las reservaciones.

Los sistemas analizados han aportado importantes ideas para el diseño e implementación de la propuesta a desarrollar, sin embargo se ha evidenciado que estos no cumplen con todas las necesidades actuales. Debido a estos elementos es necesario desarrollar un sistema que no solo brinde la posibilidad de realizar reservaciones, sino también disponga de información actualizada del estado de la tecnología, para de esta forma contribuir a que este proceso sea más efectivo.

1.3. Tecnologías, herramientas, metodología y lenguajes definidos para la solución

Partiendo de los requerimientos definidos para la implementación del sistema y de las características del entorno donde se aplicará la solución propuesta, se realizó un estudio de las tecnologías, herramientas, y lenguajes existentes en la actualidad para el desarrollo de sistemas de reservaciones de locales docentes, así como de la metodología que guiará el desarrollo.

1.3.1. Lenguaje de modelado

El lenguaje de modelado es un conjunto de signos o modelos que permiten transformar el lenguaje común a un lenguaje más técnico, con el objetivo de obtener una mejor comprensión, visualización y descripción del sistema a desarrollar (9).

Lenguaje de modelado UML² 2.0

Es un lenguaje de modelado visual que se usa para especificar, visualizar, construir y documentar artefactos de un sistema de software. Captura las decisiones y los conocimientos sobre los sistemas que se deben construir. Se usa para entender, diseñar, configurar, mantener y controlar la información sobre tales sistemas. UML incluye conceptos semánticos, notación, y principios generales (10).

² Del inglés Unified Modeling Language

1.3.2. Herramienta CASE³

El término CASE se estableció en los años 80 en los Estados Unidos. Este término se hizo popular, al asociarse con potentes herramientas para el desarrollo de sistemas. Las herramientas CASE proponen una nueva filosofía del concepto de ciclo de vida basado en la automatización, para lo cual proporcionan un conjunto de herramientas bien integradas, que enmarcadas dentro de una determinada metodología, permiten automatizar las fases del ciclo de vida de un sistema software (11).

Visual Paradigm 8.0

Visual Paradigm for UML es una herramienta CASE que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, implementación y pruebas. Ayuda a una rápida construcción de aplicaciones de calidad, mejores y a un menor coste. Permite construir diagramas de diversos tipos, código inverso, generar código desde diagramas y generar documentación. Visual Paradigm también proporciona abundantes tutoriales de UML, demostraciones interactivas de UML y proyectos UML (12).

La implementación de la propuesta de solución implica realizar un diseño previo de la solución que se quiere construir, para de esta forma, obtener una mejor visión del producto, es ahí donde juega un papel fundamental el uso de UML. Otra ventaja a destacar de UML es que puede ser utilizado indistintamente de la metodología de desarrollo elegida para implementar una solución. En esta investigación se hará uso de este lenguaje así como la herramienta de modelado Visual Paradigm para la construcción de varios artefactos como: el modelo conceptual para ayudar a comprender de forma general cómo funciona el proceso de reservación, el modelo de datos como soporte para los datos relevantes que deberán persistir para la aplicación y en el diagrama de despliegue, como guía de los componentes necesarios a la hora de desplegar la aplicación en un entorno determinado.

1.3.3. Servidor web

Un servidor web es un programa que se encarga de recibir las peticiones referidas a páginas o elementos de la web mediante el protocolo HTTP⁴ a través de una conexión TCP⁵. Es un software alojado mayormente en un ordenador servidor. Normalmente es el navegador el que pide al servidor web el recurso que desea

³ Del inglés Computer Aided Software Engineering

⁴ Del inglés Hypertext Transfer Protocol.

⁵ Del inglés Transmission Control Protocol.

el usuario, para finalmente recibir dicho recurso (de ser válida la petición) y traducirlo si es necesario a una forma legible por el usuario (es decir la traducción de HTML⁶ la hace el navegador) (13).

Apache 2.4

Apache es un servidor de código abierto para aplicaciones web, el cual fue desarrollado por la Fundación Apache. El hecho de que su código sea abierto implica que pueda ser modificado y distribuido sin restricciones de ningún tipo, para diversos fines que también pueden ser comerciales.

Utiliza el protocolo de transferencia HTTP para la gestión de las transacciones que se llevan a cabo entre el cliente y el servidor. Soporta bases de datos así como el protocolo de seguridad HTTPS para la protección de la información. Se encuentra disponible para GNU/Linux y Windows (14).

Se decidió utilizar Apache como servidor web ya que la aplicación a desarrollar necesita que los usuarios que tengan acceso a la misma puedan conectarse desde cualquier estación de trabajo de la red de la UCI, y Apache permite entre otras cosas la creación de un servidor virtual que garantiza múltiples conexiones a los sistemas web, además este es compatible con el lenguaje de programación del lado del servidor PHP, a diferencia de otras soluciones existentes como IIS⁷ de Microsoft y Apache Tomcat. Además su uso se ajusta a las políticas de migración a software libre que lleva a cabo la Universidad.

1.3.4. Lenguajes de programación

Los lenguajes de programación están diseñados para permitir la comunicación entre el usuario y las computadoras. Usándolos se puede especificar al ordenador las instrucciones que desea ejecutar, con un lenguaje relativamente próximo al lenguaje humano. Se utilizan para crear software, que faciliten el trabajo y están integrados por símbolos y reglas semánticas y sintácticas que detallan la estructura y el significado de sus elementos y expresiones, las instrucciones que forman dicho programa son conocidas como código fuente (15).

PHP 5.36

⁶ Del inglés HyperText Markup Language.

⁷ Del inglés Internet Information Services

PHP es un lenguaje *script* (no se compila para conseguir códigos máquina si no que existe un intérprete que lee el código y se encarga de ejecutar las instrucciones que contiene este código), para el desarrollo de páginas web dinámicas del lado del servidor, cuyos fragmentos de código se intercalan fácilmente en páginas HTML, debido a esto, y a que es *Open Source* (código abierto), es el más popular y extendido en la web (16).

PHP es capaz de realizar determinadas acciones de una forma fácil y eficaz sin tener que generar programas programados en un lenguaje distinto al HTML. Esto se debe a que PHP ofrece un extenso conjunto de funciones para la explotación de bases de datos sin complicaciones. Es por esto, que levanta un mayor interés con respecto a los lenguajes pensados para los CGI⁸ (16).

Para el futuro desarrollo de la solución se optó por utilizar el lenguaje de programación del lado del servidor PHP debido a las funcionalidades anteriormente mencionadas, se integra perfectamente con el servidor web Apache y con el gestor de base de datos PostgreSQL. PHP se utilizará para codificar las funcionalidades de la aplicación web a implementar.

JavaScript 1.8

JavaScript es un lenguaje de *scripting* basado en objetos, utilizado para acceder a objetos en aplicaciones web. Principalmente, se utiliza integrado en un navegador web permitiendo el desarrollo de interfaces de usuario mejoradas y páginas web dinámicas. El lenguaje fue creado por Brendan Eich en la empresa *Netscape Communications*, la que desarrolló los primeros navegadores web comerciales (17).

JavaScript ha tenido influencia de múltiples lenguajes y se diseñó con una sintaxis similar al lenguaje de programación Java. Sus características más importantes son:

1. Es un lenguaje interpretado, es decir, no requiere compilación. El navegador del usuario se encarga de interpretar las sentencias JavaScript contenidas en una página HTML y ejecutarlas adecuadamente (17).
2. Puede utilizar el modelo de programación basada en eventos. En este tipo de programación, los *scripts* se dedican a esperar a que el usuario "haga algo" (que pulse una tecla, que mueva el ratón, que cierre la ventana del navegador). A continuación, el *script* responde a la acción del usuario

⁸ Del inglés Common Gateway Interface.

normalmente procesando esa información y generando un resultado. Los eventos hacen posible que los usuarios transmitan información a los programas (18).

3. No es un lenguaje orientado a objetos propiamente dicho, sino que éste está basado en prototipos, ya que las nuevas clases se generan clonando las clases base (prototipos) y extendiendo su funcionalidad (19).

1.3.5. Framework de desarrollo

Un *framework* es una estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado. Los *framework* se han convertido en la piedra angular de la moderna ingeniería del software. Se caracteriza por ser una estructura de software compuesta de componentes personalizables e intercambiables para el desarrollo de una aplicación. En otras palabras es una aplicación genérica incompleta y configurable a la que podemos añadirle las últimas piezas para construir una aplicación concreta. Entre sus objetivos se puede encontrar (20):

- Implementar rápido.
- Proceso estructurado.
- Reutilizar el código.
- Reducir costos de producción
- Disminuir el esfuerzo en el desarrollo.
- Aprovechar las funcionalidades ya implementadas.

Un *framework* hoy en día no es una opción, es una necesidad, su uso ayuda mucho con la seguridad, reducir tiempo y tener un código optimizado, ordenado y entendible. El objetivo de utilizar un *framework* en el sistema a implementar es centrarse en el verdadero problema, sin necesidad de preocuparse por implementar funcionalidades que son de uso común en muchas aplicaciones, como el proceso de autenticación de usuarios, establecer la conexión con la base de datos, entre otros. Por tanto, cuando se hace uso de un *framework*, solamente hay que enfocarse en el verdadero centro del problema y hacer fluir todos los detalles “menores” ya que seguramente el *framework* dará una solución para ellos.

Codeigniter 2.1.3

Codeigniter es un programa o aplicación web desarrollada en PHP para la creación de cualquier tipo de aplicación web bajo PHP. Es un producto de código libre (21).

Capítulo 1: Aspectos teóricos acerca de sistemas de reservación de locales, tecnologías para su desarrollo.

Como cualquier otro *framework*, Codeigniter contiene una serie de librerías que permiten el desarrollo de aplicaciones web y además propone una manera de desarrollarlas que se debe seguir para obtener provecho de la aplicación. Marca una manera específica de codificar las páginas web y clasificar sus diferentes *scripts*, que sirve para que el código esté organizado y sea más fácil de crear y mantener (21).

Codeigniter contiene muchas ayudas para la creación de aplicaciones PHP avanzadas, que hacen que el proceso de desarrollo sea más rápido. A la vez, define una arquitectura de desarrollo que hará que se programe de una manera más ordenada y contiene diversas herramientas que ayudan a desarrollar aplicaciones más versátiles y seguras (21).

Se hará uso de este *framework* debido a las características anteriormente mencionadas y también porque hace uso de una sintaxis clara y sencilla para el acceso a datos (Active Record). La puesta en marcha de este *framework* es rápida y ofrece muchas funcionalidades entre ellas la validación de formularios y encriptación de datos, aspectos a tener en cuenta para la seguridad en la aplicación.

Bootstrap 3.0.3

Es el *framework* de *Twitter* que permite crear interfaces web con CSS⁹ y Javascript que adaptan la interfaz dependiendo del tamaño del dispositivo en el que se visualice de forma nativa, es decir, automáticamente se adapta al tamaño de un ordenador o de una Tablet sin que el usuario tenga que hacer nada, esto se denomina diseño adaptativo.

Aun ofreciendo todas las posibilidades que ofrece Bootstrap a la hora de crear interfaces web, los diseños creados con Bootstrap son simples, limpios e intuitivos, esto les da agilidad a la hora de cargar y al adaptarse a otros dispositivos.

El *framework* trae varios elementos con estilos predefinidos fáciles de configurar: Botones, menús desplegados, formularios, incluyendo todos sus elementos e integración JQuery para ofrecer ventanas y *tooltips* dinámicos (22).

JQuery 1.11

⁹ Del inglés Cascading Style Sheets.

Es una biblioteca de *Javascript*, un producto que simplificará el trabajo de programar en este lenguaje. Cuando un desarrollador tiene que utilizar Javascript, generalmente tiene que preocuparse por hacer *scripts* compatibles con varios navegadores y para ello tiene que incorporar mucho código que lo único que hace es detectar el explorador del usuario, para hacer una u otra cosa dependiendo de si es Internet Explorer, Firefox, Opera, entre otros. JQuery implementa una serie de clases que permiten programar sin preocupación del navegador con el que está navegando el usuario, ya que funcionan de exacta forma en todas las plataformas más habituales (23).

Así pues, esta librería de Javascript, ofrece una infraestructura con la que se tiene mucha mayor facilidad para la creación de aplicaciones complejas del lado del cliente. Por ejemplo, con JQuery se obtiene ayuda en la creación de interfaces de usuario, efectos dinámicos, aplicaciones que hacen uso de Ajax. Cuando se programa Javascript con JQuery se tiene a disposición una interfaz para programación que permite hacer cosas con el navegador con la seguridad de que funcione para todos los visitantes. Solamente se deben conocer las librerías del *framework* y programar utilizando las clases, sus propiedades y métodos para la consecución de los objetivos (23).

1.3.6. Entorno de Desarrollo Integrado

Un Entorno de Desarrollo Integrado (IDE) es un entorno de programación que ha sido empaquetado como un programa de aplicación, o sea, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica. Los IDEs pueden ser aplicaciones por sí solos o pueden ser parte de aplicaciones existentes (20).

NetBeans 7.3

NetBeans es un proyecto exitoso de código abierto con una gran base de usuarios y una comunidad en constante crecimiento. *Sun Microsystems* fundó el proyecto de código abierto NetBeans en junio del 2000 y continúa siendo el patrocinador principal de los proyectos.

NetBeans IDE es un entorno de desarrollo - una herramienta para que los programadores puedan escribir, compilar, depurar y ejecutar programas. Está escrito en Java - pero puede servir para cualquier otro lenguaje de programación. Se caracteriza por presentar mejoras en el editor de código, su instalación y actualización

es más simple que otros IDE, presenta características visuales para el desarrollo web, mejoras para UML y soporte para PHP (24).

1.3.7. Sistema Gestor de Base de Datos

Un Sistema Gestor de Base de Datos (SGBD) es un software o conjunto de programas que permiten crear y mantener una base de datos. Actúa como interfaz entre los programas de aplicación (usuarios) y el sistema operativo. El objetivo principal de un SGBD es proporcionar un entorno eficiente a la hora de almacenar y recuperar la información de la base de datos (25).

Por tanto debe permitir:

- Definir una base de datos: especificar tipos, estructuras y restricciones de datos.
- Construir la base de datos: guardar los datos en algún medio controlado por el mismo SGBD.
- Manipular la base de datos: realizar consultas, actualizarla, generar informes.

PostgreSQL 9.3

PostgreSQL es un SGBD relacional orientado a objetos y libre, publicado bajo la licencia BSD. Es más completo que MySQL ya que permite métodos almacenados, es decir, PostgreSQL es un sistema objeto-relacional, que incluye características de la orientación a objetos, como pueden ser la herencia, tipos de datos, funciones, restricciones, disparadores, reglas e integridad transaccional.

A pesar de esto, PostgreSQL no es un SGBD puramente orientado a objetos. Es un avanzado sistema de bases de datos relacionales basado en *Open Source*. Esto quiere decir que el código fuente del programa está disponible a cualquier persona libre de cargos directos, permitiendo a cualquiera colaborar con el desarrollo del proyecto o modificar el sistema para ajustarlo a sus necesidades (26).

Será necesario utilizar un SGBD debido a que la aplicación a desarrollar necesita contar con una base de datos que se pueda actualizar constantemente, donde se almacene toda la información que gestione el sistema, como son todos los datos asociados a las reservaciones realizadas, a los locales, los medios tecnológicos y sus características, los datos de los usuarios autenticados, las funcionalidades a las que tendrán acceso. Es perfectamente compatible con el lenguaje de programación del lado del servidor seleccionado y se ajusta a las políticas de migración a software libre.

1.3.8. Metodología de desarrollo de software

Se deberá conocer antes de estudiar las metodologías el concepto de las mismas y no es más que un conjunto de procedimientos, técnicas, herramientas y un soporte documental que brinda a los desarrolladores una guía para realizar un nuevo software.

Las metodologías de desarrollo van indicando paso a paso las actividades que se van a realizar, qué personas deben participar en el desenvolvimiento de las mismas y qué papel deben desempeñar con el objetivo de lograr el producto final y además brindan la información necesaria para el comienzo de las mismas (27).

Metodología a utilizar

Teniendo en cuenta las características del software a desarrollar se ha definido utilizar la metodología de desarrollo bajo XP.

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

XP se basa en los siguientes principios (28).

- Planificación incremental.
- Entregas pequeñas.
- Diseño sencillo.
- Desarrollo previamente probado.
- Refactorización.
- Programación en parejas.
- Propiedad colectiva.
- Integración continua.
- Ritmo sostenible.

- Cliente presente.

Para la elección de XP como metodología de desarrollo se tuvieron en cuenta varios aspectos: principalmente que los clientes estarán presente a lo largo del desarrollo del sistema a implementar, elemento que constituye uno de los requisitos fundamentales para llegar al éxito del proyecto. En el caso de esta investigación los clientes son la planificadora docente, el Jefe del Departamento de Tecnología y el responsable de hardware, todos de la Facultad 1. El software a desarrollar es relativamente pequeño y no posee una gran complejidad por lo que no es necesaria una amplia documentación de cómo se realizó. Dadas las características del sistema, algunas de sus funcionalidades podrían cambiar una vez iniciada la implementación las mismas y XP está especialmente indicada para proyectos con rápido cambio de requisitos o requisitos imprecisos. Además una vez planteada la propuesta de solución se determinó que XP cuenta con la cantidad necesaria de artefactos para que el análisis y diseño del sistema a implementar se realice con la calidad requerida.

Conclusiones parciales

A pesar de que se pueden encontrar en Internet numerosos sistemas que automatizan los procesos de reservación en varios ámbitos de la actividad humana, son pocos los que se utilizan para reservar locales docentes. Su estudio demostró que no satisfacen todos los requerimientos que demanda hoy la Facultad 1, evidenciándose la necesidad de desarrollar un sistema a la medida.

Para la selección de las herramientas y tecnologías a utilizar se tuvieron en cuenta las políticas de soberanía tecnológica. Esto permitió dirigir la investigación hacia el estudio de herramientas robustas, con un alto nivel de popularidad, aceptación y utilización por parte de las comunidades de desarrolladores, con soporte técnico y documentación abundantes.

Las condiciones existentes para el desarrollo: personal dedicado, tiempo para desarrollar el sistema, participación del cliente, variabilidad de los requerimientos, se tuvieron en cuenta a la hora de seleccionar a XP como metodología de desarrollo, comprobando que esta se ajusta perfectamente para el diseño e implementación de la solución propuesta.

Capítulo 2: Descripción del sistema de reservación de locales docentes en la Facultad 1

Introducción

En el presente capítulo se exponen las características del sistema a implementar. Se describe el flujo actual de los procesos que se llevan a cabo en la Facultad 1 en el momento de reservar un local. Este análisis permite identificar las actividades que pueden ser objeto de automatización y realizar una propuesta del sistema. Se identifican las funcionalidades del sistema como base para la confección de la Lista de reserva del producto y las Historias de usuario (HU). Se explican los módulos que componen el sistema y los usuarios que están involucrados en este. Como artefactos fundamentales se destacan: las Historias de usuario, Lista de reserva del producto, Plan de entrega, Plan de iteraciones y Tarjetas CRC¹⁰.

2.1. Modelo conceptual

Para una mejor comprensión del problema se hace uso de un modelo conceptual. XP no propone este modelo como uno de sus artefactos, pero al ser una metodología bastante flexible se puede utilizar cualquier elemento que ayude a un mejor entendimiento del problema.

Un modelo conceptual tiene como objetivo identificar y explicar los conceptos significativos en un dominio de problema, identificando los atributos y las asociaciones existentes entre ellos. Puede ser visto, también como una representación de las cosas, entidades, ideas, conceptos u objetos del “mundo real” o dominio de interés.

Los principales conceptos identificados en este dominio fueron:

- **Local:** Es el lugar que se va a reservar, en este caso puede ser un aula, laboratorio o salón de conferencias.
- **Medio:** Son los medios con los que cuenta cada local.
- **Características:** Hace referencia a las características de cada medio de los locales.
- **Reservaciones:** Son las reservaciones de los locales realizadas por la planificadora.

¹⁰ Clase - Responsabilidad - Colaboración

Capítulo 2: Descripción del sistema de reservación de locales docentes en la Facultad 1.

- **Planificadora:** Realiza las reservaciones, para eso debe consultar la disponibilidad de los locales en el horario.
- **Responsable de hardware:** Actualiza diariamente las características de los locales docentes.
- **Solicitante:** Es el que realiza la reservación de los locales.
- **Horario:** Es donde se guarda el horario de las reservaciones realizadas por la planificadora.

La figura 1 muestra el modelo conceptual del sistema a implementar.

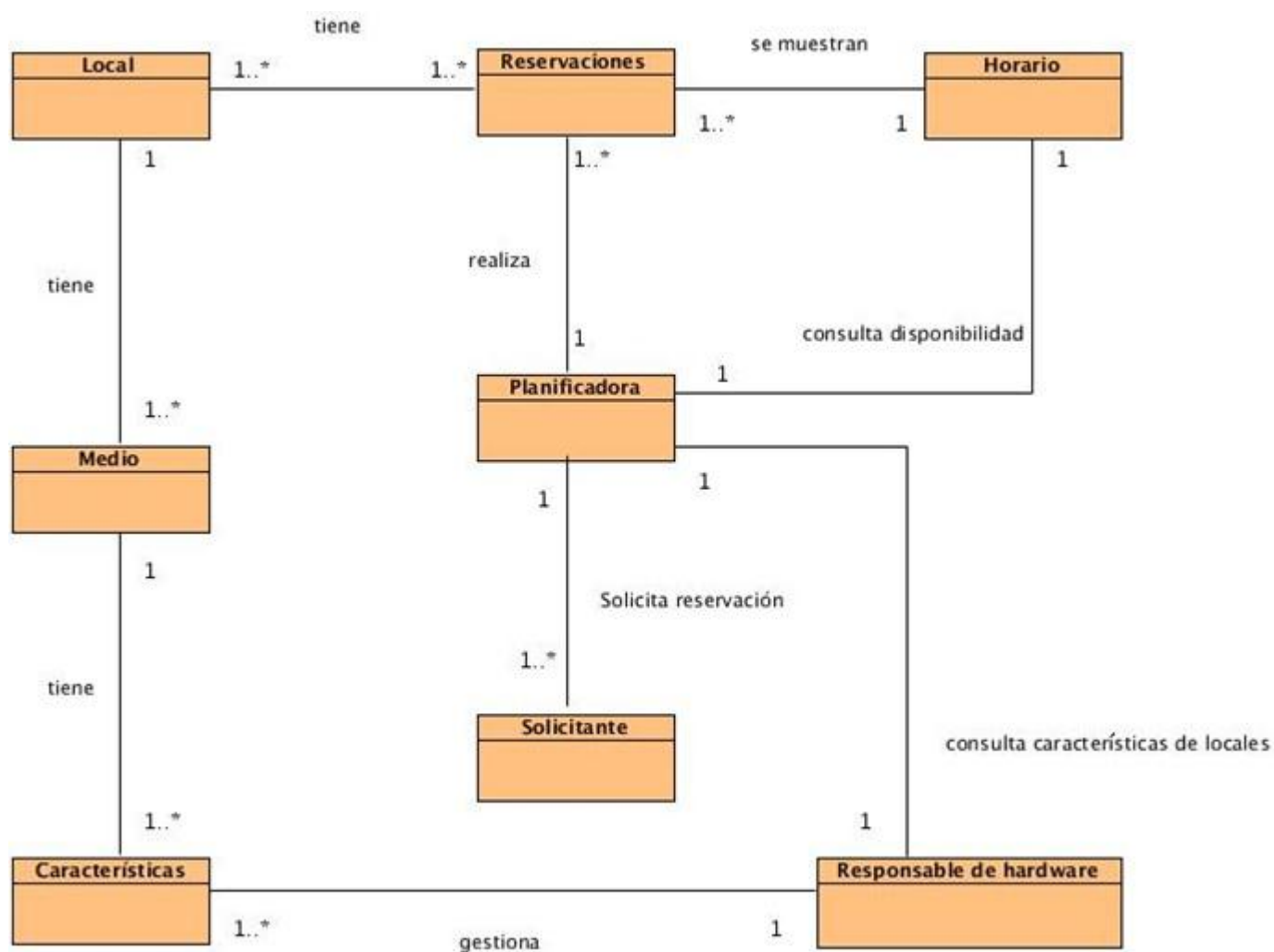


Fig 1. Modelo conceptual

2.2. Flujo actual del proceso

El proceso de reservación de un local docente para las actividades no planificadas en el horario en la Facultad 1, comienza cuando el interesado se dirige a la planificadora para solicitar la reservación, esta busca los locales que están disponibles en el horario solicitado. Si es un aula o un salón de conferencia se reserva en dependencia de si es necesaria o no la tecnología. En caso de ser un laboratorio debe ponerse en contacto con el Departamento de Tecnología e informar de su necesidad al responsable de hardware. En dependencia de la actividad a realizar se reserva el que cumpla con las condiciones de trabajo solicitadas por los interesados. En caso de que no se encuentre nadie en el Departamento de Tecnología se reserva cualquier local de los disponibles. Esto hace que la reservación se vuelva un proceso largo y engorroso que conlleva a la pérdida de tiempo y a posibles errores en las reservaciones, pues los interesados en reservar locales docentes deben esperar a que la planificadora revise las reservaciones previas en cada uno de los documentos dispuestos para este fin.

Debido a lo planteado anteriormente es necesaria la implementación de un sistema automatizado que permita reservar los locales docentes en la Facultad 1, ya que la forma en que se ejecuta actualmente no satisface las necesidades de dicha facultad. El objetivo de este trabajo es facilitar dicho proceso que hasta ahora se realiza de forma manual. Así se garantizará una mejor organización y control de todas las reservaciones.

2.3. Propuesta de solución

A partir del análisis anterior se propone como solución la implementación de un sistema para la reservación de los locales docentes en la Facultad 1, que contemple las características técnicas del equipamiento disponible en cada local, con el objetivo de agilizar y hacer más eficiente el proceso de reservación.

Para ello se contará con un rol “Planificador” el cual se encargará de gestionar las reservaciones en el sistema, pudiendo listar, eliminar o modificar cualquier reservación realizada en el sistema. Uno de los aspectos relevantes del sistema radica en que brindará la posibilidad de realizar reservaciones de acuerdo a las características técnicas de los medios ubicados en los diferentes locales. Esto conlleva a la necesidad de contar con datos actualizados diariamente del equipamiento, contando con un rol “Administrador” para ello. Este trabajo será llevado a cabo por el responsable de hardware de la Facultad 1, de acuerdo a sus

permisos podrá gestionar: locales, equipamiento y sus características técnicas. Por otra parte el “Solicitante” podrá consultar los horarios de las reservaciones realizadas en el sistema, las características de cada local, gestionar sus reservaciones y ver los reportes de los locales docentes, siendo estos sus permisos básicos. Una vez consultados estos datos el solicitante podrá realizar la reservación, para ello introducirá los datos que se le indiquen. El sistema, de forma general estará dividido en cuatro módulos: Gestión, Reservación, Seguridad y Reporte.

2.3.1. Módulo de Gestión

En este módulo se podrán gestionar los locales, los medios de estos y sus características, a este módulo tendrán acceso el rol de “Administrador”.

2.3.2. Módulo de Reservación

En este módulo se gestionarán las reservaciones, y se mostrará un calendario con todas las reservaciones realizadas en el sistema. A este módulo podrán acceder todos los roles del sistema.

2.3.3. Módulo de Seguridad

En este módulo se podrán gestionar los roles y los menús del sistema, añadirle a cada menú las funcionalidades que podrá realizar, modificar el rol de los usuarios y asignarle a cada rol los menús a los cuales tendrá acceso. A este módulo solo tendrá acceso el rol de “Administrador”.

2.3.4. Módulo de Reporte

En este módulo se implementará un reporte que muestre la cantidad de medios y de reservaciones que tiene cada local.

2.4 Lista de reserva del producto

Artefacto que usa XP para identificar las funcionalidades del sistema y la prioridad de cada una. XP plantea que las funcionalidades de alta prioridad son las que se deben desarrollar primero. Por otra parte en este artefacto se definen también los requisitos no funcionales que son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido y confiable. En muchos casos los requerimientos no funcionales son fundamentales en el éxito del producto. La tabla 1 muestra la lista de reserva del producto.

Capítulo 2: Descripción del sistema de reservación de locales docentes en la Facultad 1.

Tabla 1. Lista de reserva del producto.

Código	Descripción	Prioridad
Funcionalidades del Sistema		
F 1	Autenticar usuario	Alta
F 2	Insertar tipo de local	Alta
F 3	Listar tipos de locales	Alta
F 4	Modificar tipo de local	Alta
F 5	Eliminar tipo de local	Alta
F 6	Insertar local	Alta
F 7	Listar locales	Alta
F 8	Modificar local	Alta
F 9	Eliminar local	Alta
F 10	Insertar tipo de medio	Alta
F 11	Listar tipos de medios	Alta
F 12	Modificar tipo de medio	Alta
F 13	Eliminar tipo de medio	Alta
F 14	Insertar medios a los locales	Alta
F 15	Listar medios de locales	Alta
F 16	Modificar medios de locales	Alta
F 17	Eliminar medios de locales	Alta
F 18	Mostrar horario por día, semana o mes	Alta
F 19	Insertar reservación	Alta
F 20	Listar reservaciones	Alta
F 21	Modificar reservación	Alta
F 22	Eliminar reservación	Alta
F 23	Listar usuarios	Media
F 24	Modificar usuario	Media
F 25	Eliminar usuario	Media

Capítulo 2: Descripción del sistema de reservación de locales docentes en la Facultad 1.

F 26	Insertar rol	Media
F 27	Listar rol	Media
F 28	Modificar rol	Media
F 29	Eliminar rol	Media
F 30	Mostrar reportes	Media
F 31	Insertar menú del sistema	Baja
F 32	Listar menús del sistema	Baja
F 33	Modificar menú del sistema	Baja
F 34	Eliminar menú del sistema	Baja
Requisitos no Funcionales (RNF)		
RNF 1	Confiabilidad	
	El sistema deberá brindar información periódicamente actualizada acerca del estado del equipamiento tecnológico disponible en cada local, para evitar errores que puedan incidir negativamente en el proceso de reservación así como en las actividades a planificar. Para garantizar que se cumpla este requisito el administrador deberá actualizar cualquier cambio que se realice en los locales y en el equipamiento con que cuentan	
RNF 2	Seguridad	
RNF 2.1	La accesibilidad de los datos estará restringida de acuerdo al rol asignado al usuario, para mantener la integridad de los datos en función del nivel de acceso asignado.	
RNF 2.2	El sistema brindará mecanismos de seguridad que garanticen los niveles de acceso de acuerdo al rol asignado por el usuario administrador. El rol Administrador tendrá acceso a todos los módulos del sistema. El rol Planificador tendrá acceso a la gestión de todas las reservaciones del sistema, mientras que el rol Solicitante solo tendrá acceso a consultar el estado técnico de los locales y a gestionar sus reservaciones.	
RNF3	Restricciones de diseño e implementación	
RNF 3.1	Para la implementación del componente se utilizará el lenguaje de programación PHP 5.36	
RNF 3.2	Como framework de PHP se utilizará Codeigniter 2.1.3.	
RNF 3.3	Se hará uso de la biblioteca JQuery 1.11 de JavaScript.	

RNF 3.4	Se utilizará Bootstrap 3.0.3 como framework de CSS para la creación de los elementos visuales del sistema.
RNF 3.5	Para el modelado del sistema se hará uso de la herramienta CASE Visual Paradigm 8.0
RNF 4	Apariencia o interfaz externa
	El sistema deberá tener una interfaz sobria, sencilla e intuitiva para los usuarios.
RNF 5	Software(Cliente)
RNF 5.1	Debe tener un Navegador Web, puede ser IE, Opera, Google Chrome, Firefox o cualquier otro que cumpla con los estándares de la W3C ¹¹ .
RNF 6	Software(Servidor)
RNF 6.1	Gestor de base de datos PostgreSQL versión 9.3 o superior.
RNF 6.2	Servidor web Apache versión 2.4 o superior.
RNF 7	Hardware(Cliente)
RNF 7.1	Requerimientos Mínimos: PC con microprocesador con al menos 1.0 GHz de velocidad y 256 MB de memoria RAM o superior.
RNF 7.2	Conexión de red.
RNF 8	Hardware(Servidor)
RNF 8.1	Requerimientos Mínimos: PC con microprocesador de 2 núcleos o superior con al menos 2.8 GHz de velocidad y 2 GB de memoria RAM o superior.
RNF 8.2	Contar con al menos 20 GB de espacio libre en el HDD.
RNF 8.3	Conexión de red.

2.5. Planificación

La metodología XP plantea la planificación como un diálogo continuo entre las partes involucradas en el proyecto, incluyendo al cliente, a los programadores y a los coordinadores o gerentes. El proyecto comienza transformando las funcionalidades en HU, las que sustituyen a los tradicionales “casos de uso”. Obtenidas las HU, se evalúa el tiempo de desarrollo de cada una. Una vez realizadas estas estimaciones, se organiza una reunión de planificación con el cliente, para establecer un plan o cronograma de entregas (“*Release*”).

¹¹ Del inglés World Wide Web Consortium.

Plan”). Acordado este cronograma, comienza una fase de iteraciones, en donde en cada una de ellas se implementan algunas HU y se le aplican las pruebas para validar su correcto funcionamiento.

2.5.1. Historias de Usuario

Las HU reflejan los requisitos del cliente. Tienen el mismo propósito que los casos de uso, con la excepción de que no se limitan a la descripción de la interfaz de usuario. Se utilizan en la creación de las pruebas, para verificar si el programa cumple con lo que especifica cada HU. La principal diferencia entre estas y la tradicional especificación de requisitos, es el nivel de detalle (29).

Campos de las historias de usuario

- **Número:** Contiene el identificador de la HU.
- **Nombre:** Contiene el nombre que identifica a la HU.
- **Actor:** Roles que pueden acceder a la HU.
- **Modificación de la historia de usuario:** Número de veces que se ha modificado la HU.
- **Iteración Asignada:** Número de la iteración en que se va a implementar la HU.
- **Prioridad del Negocio:** Prioridad que se le asigna a la HU en el negocio. Si es importante debe ser implementada lo antes posible, esta se clasifica en:
 - ✓ **Alta:** Se le otorga a las HU que resultan funcionalidades fundamentales en el desarrollo del sistema, las que el cliente define como principales para el control integral del sistema.
 - ✓ **Media:** Se le otorga a las HU que resultan para el cliente como funcionalidades a tener en cuenta, sin que estas tengan una afectación sobre el sistema que se esté desarrollando.
 - ✓ **Baja:** Se le otorga a las HU que constituyen funcionalidades que sirven de ayuda al control de elementos asociados al equipo de desarrollo, a la estructura y no tienen nada que ver con el sistema en desarrollo.

Capítulo 2: Descripción del sistema de reservación de locales docentes en la Facultad 1.

- **Puntos Estimados:** Estimación hecha por el equipo de desarrollo del tiempo de duración, en este caso la estimación se realizó en días.
- **Riesgo en Desarrollo:** Riesgo que representa para el desarrollo la HU. En dependencia del riesgo que represente se le asigna un valor que puede ser:
 - ✓ **Alta:** Cuando en la implementación de las HU se considera la posible existencia de errores que lleven a la inoperatividad del código.
 - ✓ **Media:** Cuando pueden aparecer errores en la implementación de las HU que puedan retrasar la entrega de la versión.
 - ✓ **Baja:** Cuando pueden aparecer errores que serán tratados con relativa facilidad, sin que traigan perjuicios para el desarrollo del proyecto.
- **Descripción:** Breve descripción del proceso que define la HU.
- **Observaciones:** Comentarios aclaratorios relacionados con la HU.
- **Prototipo de interfaz:** Prototipo de interfaz que tendrá la HU.

A continuación se muestran algunas HU del sistema. El resto de las HU se incluyen en el [Anexo 1](#).

Capítulo 2: Descripción del sistema de reservación de locales docentes en la Facultad 1.

Tabla 2. HU Autenticar usuario.

Historia de Usuario	
Número: 1	Nombre Historia de Usuario: Autenticar usuario
Modificación de Historia de Usuario Número: Ninguna	
Actor: Solicitante, Planificadora, Administrador	Iteración Asignada: 1
Prioridad en Negocio: Alta	Puntos Estimados: 5
Riesgo en Desarrollo: Alta	
Descripción: Para acceder al sistema se deberá introducir el usuario y la contraseña.	
Observaciones: solamente podrán acceder los usuarios pertenecientes a la Facultad 1	

Tabla 3. HU Insertar local.

Historia de Usuario	
Número: 6	Nombre Historia de Usuario: Insertar local
Modificación de Historia de Usuario Número: Ninguna	
Actor: Administrador	Iteración Asignada: 1
Prioridad en Negocio: Alta	Puntos Estimados: 3
Riesgo en Desarrollo: Media	
Descripción: El administrador deberá poder adicionar locales docentes en el sistema.	
Observaciones:	

Tabla 4. HU Insertar reservación.

Historia de Usuario	
Número: 19	Nombre Historia de Usuario: Insertar reservación
Modificación de Historia de Usuario Número: Ninguna	
Actor: Solicitante, planificadora y administrador	Iteración Asignada: 2
Prioridad en Negocio: Alta	Puntos Estimados: 3
Riesgo en Desarrollo: Alta	
Descripción: Los usuarios podrán adicionar una reservación de uno o varios locales docentes a la hora que soliciten.	
Observaciones: El sistema debe validar que no coincidan dos reservaciones en el mismo local a la misma hora. En caso de que se quiera reservar en el horario de 12:01 AM a 7:59 AM el sistema deberá mostrar un mensaje indicando que a esa hora no se pueden realizar reservaciones.	

2.5.2. Estimación de esfuerzo por HU

Después de definir las HU, se realiza una planificación del tiempo que se necesita para implementarlas y así determinar el tiempo total para el desarrollo de la aplicación. En este caso la estimación será en días por la poca complejidad de las HU.

Tabla 5. Estimación de esfuerzo por HU.

Historia de usuario	Estimación de esfuerzo
Autenticar usuario	5
Insertar tipo de local	3
Listar tipos de locales	2
Modificar tipo de local	3
Eliminar tipo de local	2
Insertar local	3
Listar local	2
Modificar local	3

Capítulo 2: Descripción del sistema de reservación de locales docentes en la Facultad 1.

Eliminar local	2
Insertar tipo de medio	3
Listar tipos de medios	2
Modificar tipo de medio	3
Eliminar tipo de medio	2
Insertar medios a los locales	3
Listar medios de locales	2
Modificar medios de locales	3
Eliminar medios de locales	2
Mostrar horario por día, semana o mes	3
Insertar reservación	3
Listar reservaciones	2
Modificar reservación	3
Eliminar reservación	2
Listar usuarios	2
Modificar usuario	3
Eliminar usuario	2
Insertar rol	3
Listar rol	2
Modificar rol	3
Eliminar rol	2
Mostrar reportes	5
Insertar menú del sistema	3
Listar menús del sistema	2
Modificar menú del sistema	3
Eliminar menú del sistema	2
Total de historias de usuario: 34	Total de esfuerzo: 90

A partir de la suma de los puntos de estimación de esfuerzo por cada historia de usuario se calcula que el desarrollo de la aplicación tendrá una duración de 90 días, un total de 15 semanas.

2.5.3. Iteraciones

En la metodología XP, el desarrollo del sistema se divide en etapas para facilitar su realización. Por lo general, los proyectos constan de más de tres etapas, las cuales toman el nombre de iteraciones, de allí se obtiene el concepto de metodología iterativa.

Para cada iteración se define un módulo o conjunto de HU que se van a implementar. Al final de la iteración se obtiene como resultado la entrega del módulo correspondiente, el cual debe haber superado las pruebas de aceptación que establece el cliente para verificar el cumplimiento de los requisitos (30).

2.5.4. Plan de duración de las iteraciones

Tabla 6. Plan de duración de las iteraciones.

Iteración	# HU	HU	Duración estimada
1	1	Autenticar usuario	35 días
	2	Insertar tipo de local	
	3	Listar tipos de locales	
	4	Modificar tipo de local	
	5	Eliminar tipo de local	
	6	Insertar local	
	7	Listar locales	
	8	Modificar local	
	9	Eliminar local	
	10	Insertar tipo de medio	
	11	Listar tipos de medios	
	12	Modificar tipo de medio	
	13	Eliminar tipo de medio	
2	14	Insertar medios a los locales	23 días
	15	Listar medios de locales	
	16	Modificar medios de locales	
	17	Eliminar medios de locales	

Capítulo 2: Descripción del sistema de reservación de locales docentes en la Facultad 1.

	18	Mostrar horario por día, semana o mes	32 días
	19	Insertar reservación	
	20	Listar reservaciones	
	21	Modificar reservación	
	22	Eliminar reservación	
3	23	Listar usuarios	
	24	Modificar usuario	
	25	Eliminar usuario	
	26	Insertar rol	
	27	Listar rol	
	28	Modificar rol	
	29	Eliminar rol	
	30	Mostrar reportes	
	31	Insertar menú del sistema	
	32	Listar menús del sistema	
	33	Modificar menú del sistema	
	34	Eliminar menú del sistema	

2.5.5. Plan de entrega

En el plan de entregas se establecen el orden y las historias de usuario que serán agrupadas para conformar una entrega. Este plan tiene como finalidad reflejar la duración de cada iteración, lo que ayuda a obtener una idea aproximada del tiempo que durará la confección del sistema en su totalidad. Este cronograma es resultado de una reunión entre los clientes y el desarrollador.

Tabla 7. Plan de entrega.

Iteración	1ra Iteración	2da Iteración	3ra Iteración
HU	1-13	14-22	23-34
Fecha de entrega	5 de enero de 2015	7 de febrero de 2015	23 de marzo de 2015

2.6. Diseño del sistema

Uno de los elementos más importantes de la filosofía XP es la simplicidad en todos los aspectos. Se considera que un diseño sencillo se logra más rápido y se implementa en menos tiempo. La idea es que se haga el diseño más sencillo que cumpla con los requerimientos de las historias de usuario.

Sobre los diagramas, se es muy claro que se pueden usar siempre que no tome mucho tiempo en realizarlos, que sean de verdadera utilidad y que se esté dispuesto a “tirarlos a la basura”. En XP se prefiere tener una descripción del sistema o parte de él, en lugar de una serie de complejos diagramas que probablemente tomen más tiempo y sean menos instructivos (30).

2.6.1. Tarjetas CRC

Las tarjetas CRC permitirán desprenderse del método de trabajo basado en procedimientos y trabajar con una metodología basada en objetos, permiten también que el equipo completo contribuya en la tarea del diseño.

Estas tarjetas se dividen en tres secciones que contienen la información del nombre de la clase, sus responsabilidades y sus colaboradores.

Una clase es cualquier persona, cosa, evento, concepto, pantalla o reporte. Las responsabilidades de una clase son las cosas que conoce y las que realizan, sus atributos y métodos. Los colaboradores de una clase son las demás clases con las que trabaja en conjunto para llevar a cabo sus responsabilidades (31).

Los pasos a seguir para realizar las tarjetas son los siguientes:

- Encontrar clases.
- Encontrar responsabilidades.
- Definir colaboradores.
- Disponer las tarjetas.

Tabla 8. Tarjeta CRC Tipo de local.

Tipo de local	
Responsabilidad	Colaboración
Insertar tipo de local Listar tipos de locales Modificar tipos de locales Eliminar tipos de locales	

Tabla 9. Tarjeta CRC Local.

Local	
Responsabilidad	Colaboración
Insertar local Listar locales Modificar locales Eliminar locales	Tipo de local

Tabla 10. Tarjeta CRC Tipo de medio.

Tipo de medio	
Responsabilidad	Colaboración
Insertar tipo de medio Listar tipo de medio Modificar tipo de medio Eliminar tipo de medio	

Tabla 11. Tarjeta CRC Medio.

Medio	
Responsabilidad	Colaboración
Insertar medio a local Listar medios de locales Modificar medio de local Eliminar medio de local	Tipo de medio Local

Capítulo 2: Descripción del sistema de reservación de locales docentes en la Facultad 1.

Tabla 12. Tarjeta CRC Reservación.

Reservación	
Responsabilidad	Colaboración
Insertar reservación Listar reservación Modificar reservación Eliminar reservación	Local

Tabla 13. Tarjeta CRC Usuario.

Usuario	
Responsabilidad	Colaboración
Autenticar usuario Listar usuario Modificar usuario Eliminar usuario	Rol

Tabla 14. Tarjeta CRC Rol.

Rol	
Responsabilidad	Colaboración
Insertar rol Listar rol Modificar rol Eliminar rol	

Tabla 15. Tarjeta CRC Menús del sistema.

Menús del sistema	
Responsabilidad	Colaboración
Insertar menús del sistema Listar menús del sistema Modificar menús del sistema Eliminar menús del sistema	

2.6.2. Diseño de base de datos

El objetivo de diseñar una base de datos es generar un conjunto de esquemas de relaciones que permitan almacenar información con un mínimo de redundancia, pero que permita facilitar la recuperación de la información. La metodología XP no propone un modelo de datos, pero es importante el almacenamiento de información debido que es una actividad fundamental para el funcionamiento del sistema a implementar. Esto permite almacenar y posteriormente acceder a los datos de forma rápida y estructurada.

La base de datos se dividió en dos esquemas, las figuras 2 y 3 describen el diseño de la base de datos.

Para consultar la descripción de las tablas, remitirse al [Anexo 3](#).

Capítulo 2: Descripción del sistema de reservación de locales docentes en la Facultad 1.

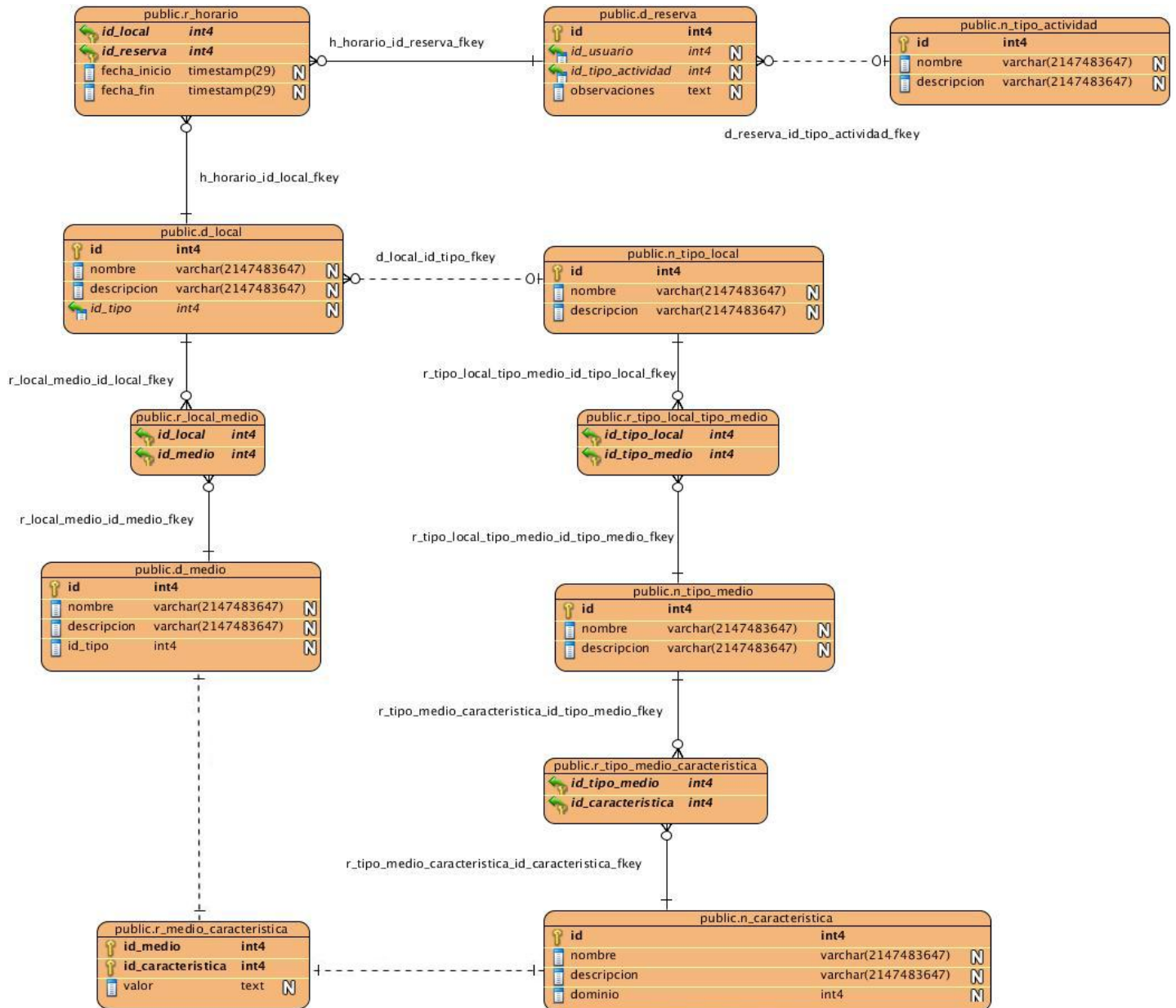


Fig 2. Diseño de base de datos del esquema de reservación.

Capítulo 2: Descripción del sistema de reservación de locales docentes en la Facultad 1.

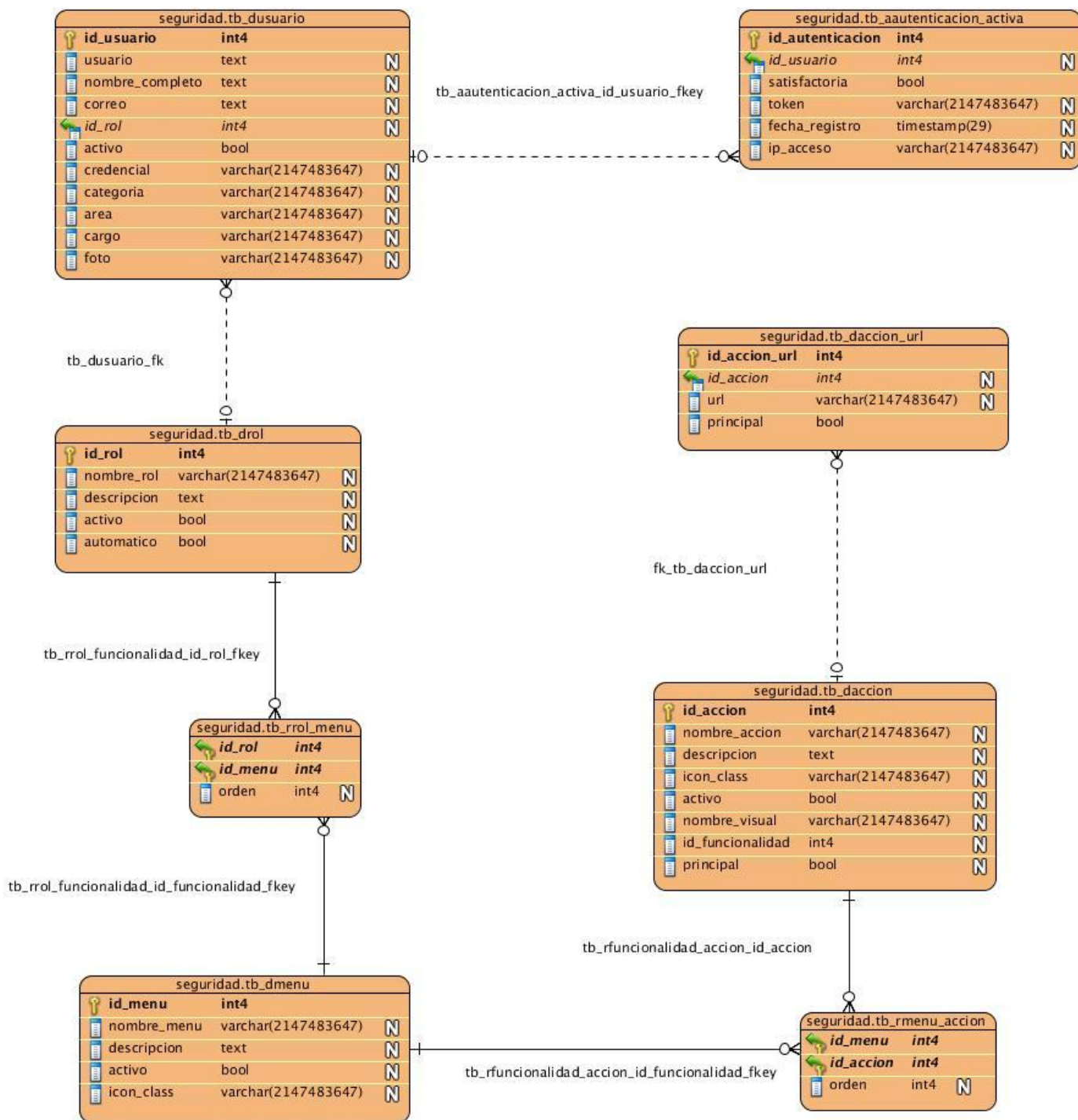


Fig 3. Diseño de base de datos del esquema de seguridad.

2.6.3. Patrones de arquitectura de software

Los patrones de arquitectura de software se utilizan para expresar una estructura de organización base o esquema para un software. Proporcionando un conjunto de subsistemas predefinidos, especificando sus responsabilidades, reglas, directrices que determinan la organización, comunicación, interacción y relaciones entre ellos. Los patrones arquitectónicos heredan mucha de la terminología y conceptos de patrones de diseño, pero se centran en proporcionar modelos y métodos reutilizables específicamente para la arquitectura general de los sistemas de información (32).

Modelo Vista Controlador (MVC)

Modelo Vista Controlador (MVC) es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. El patrón MVC se ve frecuentemente en aplicaciones web, donde la vista es la página HTML y el código que provee de datos dinámicos a la página (33).

La figura 4 es un ejemplo de funcionamiento del MVC en una funcionalidad del sistema.

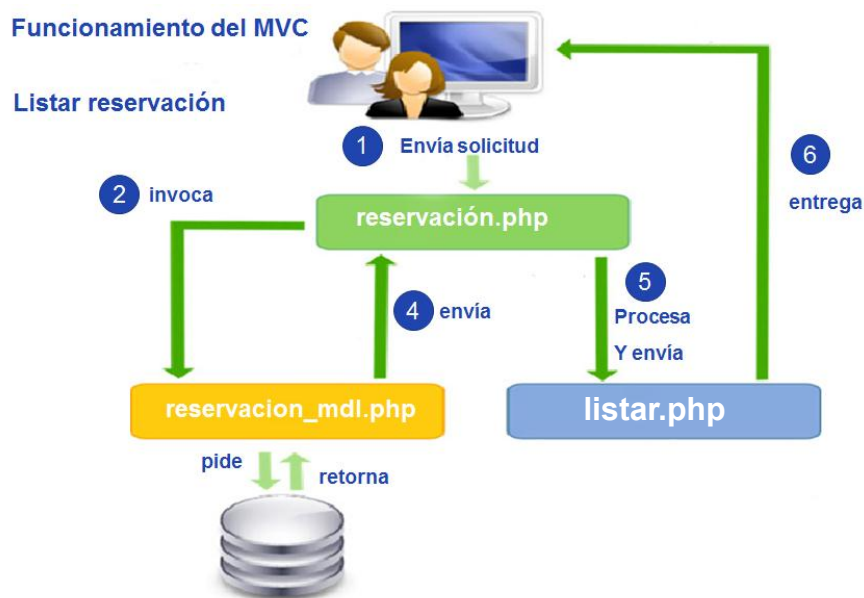


Fig 4. Modelo Vista Controlador.

- **Modelo:** Es la capa encargada de encapsular toda la lógica de negocio de la aplicación. Esta capa se puede subdividir en varias:
 - **Lógica de negocio:** Contiene clases para constituir lo referente a la capa de aplicación, se encarga de atender a las peticiones de los controladores y así dar una respuesta acorde con lo recibido.
 - **Capa de datos:** Se encarga de gestionar toda la interconexión con el SGBD, así mismo, puede contener un gestor Mapeo Objeto Relacional (ORM¹²) para su aprovechamiento máximo y mejor mantenimiento. Solo se comunica con la lógica de negocio.
 - **Helpers:** Llamados “ayudantes” apoyan tanto al controlador como a la vista para hacer más livianas algunas tareas.
- **Controlador:** Es el eje central de la arquitectura, encargada de gestionar todas las peticiones, validar los inputs recibidos y dirigir cualquier petición de cualquier tipo. Solo se comunica con el modelo y responde a través de vistas.
- **Vista:** Es la respuesta de cada controlador y lo que se le presenta al usuario final, se puede comunicar con el controlador, los “helpers” y el modelo (en algunas ocasiones) (34).

2.6.4. Patrones de diseño

Los patrones de diseño son la base para la búsqueda de soluciones a problemas comunes en el desarrollo de software y otros ámbitos referentes al diseño de interacción o interfaces. Para que una solución sea considerada un patrón debe poseer ciertas características. Una de ellas es su efectividad resolviendo problemas similares en ocasiones anteriores. Otra es que debe ser reusable, lo que significa que es aplicable a diferentes problemas de diseño en distintas circunstancias (35).

¹² Del inglés Object-relational mapping.

Patrones de diseño GRASP¹³

Los patrones GRASP describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en formas de patrones. El nombre se eligió para indicar la importancia de captar estos principios si se quiere diseñar eficazmente el software orientado a objetos.

Los patrones GRASP no establecen nuevas ideas, son una codificación de principios básicos ampliamente utilizados. La decisión acerca de la asignación de responsabilidades, a menudo, tiene lugar durante la creación de los diagramas de interacción y con seguridad durante la programación (36).

Los patrones de diseño GRASP utilizados fueron:

- **Patrón Experto:** Es el principal responsable de la asignación de responsabilidades. Se refiere a la clase que cuenta con la información o atributos necesarios para cumplir con la responsabilidad o responsabilidades para la que fue implementada. De esta forma se conserva la encapsulación y se logra una mayor cohesión (36).

La aplicación de este patrón se evidencia en la clase **reservación_lib** la cual tiene la información necesaria para hacer la llamada al método **obtener**, el cual llama al método **obtenerParametros**.

- **Patrón Creador:** Guía la asignación de responsabilidades relacionadas con la creación de objetos, tarea muy frecuente en los sistemas orientados a objetos, de forma tal que una instancia de un objeto solo pueda ser creada por el objeto que contiene la información necesaria. Tiene como ventaja el bajo acoplamiento que supone facilidad de mantenimiento y reutilización (36).

La aplicación de este patrón se evidencia en la clase **local** en el objeto load del constructor que es el encargado de cargar las librerías y los modelos. Ej: `$this->load->library('tipo_local_lib', null, 'tl_lib');`

- **Patrón Controlador:** Puede ser definido como el elemento intermedio entre una interfaz determinada y el algoritmo que la implementa. Es quien ayuda a decidir quién debería encargarse

¹³ Del inglés General Responsibility Assignment Software Patterns.

de un evento del sistema (Evento de alto nivel generado por un actor externo al sistema). Un controlador es un objeto de interfaz no destinada al usuario que se encarga de manejar un evento del sistema. Al llegar el evento, una petición del usuario al sistema el controlador decidirá quién se va a encargar de manejar dicho evento o petición. Define además el método de su operación (36).

La aplicación de este patrón se pone de manifiesto en la clase **reservación** en el método **registrar** donde se obtienen y se envían los datos a la clase **reservacion_lib**.

- **Patrón Alta Cohesión:** Una clase con alta cohesión tiene un número relativamente pequeño de métodos, con funcionalidades altamente relacionadas, y no realiza mucho trabajo. Colabora con otros objetos para compartir el esfuerzo si la tarea es extensa (36).
- **Patrón Bajo Acoplamiento:** Una clase con bajo acoplamiento asigna las responsabilidades a cada clase para mantener un bajo acoplamiento entre las mismas. El bajo acoplamiento apoya al diseño de clases más independientes, que reduce el impacto de los cambios, así como clases más reutilizables. El acoplamiento no es importante sino se busca la reutilización (36).

Estos dos últimos patrones se evidencian en la implementación del propio marco de trabajo Codeigniter, el cual permite el uso de forma individual de los componentes, poniendo de manifiesto el bajo acoplamiento, así como la dependencia existente entre ellos o alta cohesión.

Patrones de diseño GOF¹⁴

Son un conjunto de patrones recopilados y documentados por cuatro expertos diseñadores de software orientado a objetos, estos son Erich Gamma, Richard Helm, Ralph Johnson y John Vlissides. Los patrones de GOF se clasifican en tres categorías basadas en su propósito: creacionales, estructurales y de comportamiento.

- **Creacionales:** Tratan con las formas de crear instancias de objetos. El objetivo de estos patrones es abstraer el proceso de instanciación y ocultar los detalles de cómo los objetos son creados o inicializados.

¹⁴ Del inglés Gang of Four

- **Estructurales:** Describen cómo las clases y objetos pueden ser combinados para formar grandes estructuras y proporcionar nuevas funcionalidades. Estos objetos adicionales pueden ser incluso objetos simples u objetos compuestos.
- **Comportamiento:** Ayudan a definir la comunicación e iteración entre los objetos de un sistema. El propósito de este patrón es reducir el acoplamiento entre los objetos (37).

Los patrones de diseño GOF utilizados fueron:

- **Instancia única (*Singleton*):** Garantiza que solamente se crea una instancia de la clase y provee un punto de acceso global a él. Todos los objetos que utilizan una instancia de esa clase usan la misma instancia (38).

La aplicación de este patrón se puede ver en las clases controladoras pues todas ellas son instancias únicas.

- **Observador (*Observer*):** Permite a los objetos captar dinámicamente las dependencias entre objetos, de tal forma que un objeto notificará a los objetos dependientes de él cuando cambia su estado, siendo actualizados automáticamente (38).

La puesta en práctica de este patrón se evidencia en la clase *loader* que es el objeto load de las clases controladoras, la cual se encarga de cargar las librerías y los modelos, además de actualizar la clase controladora instanciada.

2.6.5 Diagrama de despliegue

Los diagramas de despliegue muestran las relaciones físicas de los distintos nodos que componen un sistema y el reparto de los componentes sobre dichos nodos. La vista de despliegue representa la disposición de las instancias de componentes de ejecución en instancias de nodos conectados por enlaces de comunicación (39).

Capítulo 2: Descripción del sistema de reservación de locales docentes en la Facultad 1.

La configuración propuesta estará conformada por un servidor web. Este a su vez se conectará con el servidor LDAP¹⁵ UCI para verificar la autenticidad del usuario, posteriormente se conectará con la base de datos para almacenar toda la información del sistema. Las máquinas clientes a su vez estarán conectadas con el servidor web.

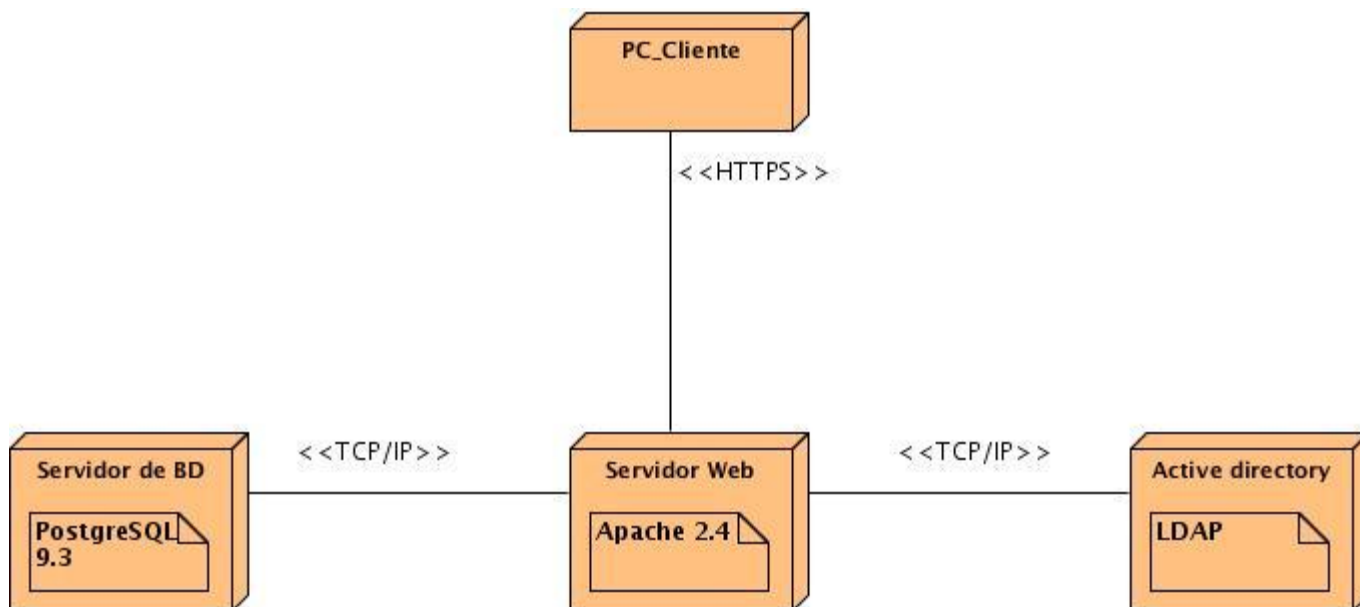


Fig 5. Diagrama de despliegue.

Conclusiones parciales

La participación del cliente junto al equipo de desarrollo fue un elemento clave en la comprensión del proceso de reservación de locales docentes en la Facultad 1, lo que permitió definir las funcionalidades del sistema y asignarles prioridad en el negocio.

La descripción de las principales clases a través de tarjetas CRC y la definición de planes de iteración y de entrega valorando las principales limitaciones del equipo de desarrollo permitieron fijar un cronograma acorde a las expectativas del cliente que será punto de partida para la implementación de las HU asociadas al sistema, demostrándose la importancia de estas actividades dentro de la metodología XP.

¹⁵ Del inglés Lightweight Directory Access Protocol.

Capítulo 2: Descripción del sistema de reservación de locales docentes en la Facultad 1.

Teniendo en cuenta las características del entorno donde será desplegado el sistema y las del propio sistema, su puesta en funcionamiento no consume gran cantidad de recursos desde el punto de vista de hardware para su uso en la Facultad 1.

Capítulo 3: Implementación y pruebas al sistema de reservación de locales docentes en la Facultad 1

Introducción

En el presente capítulo se describe el proceso de implementación del sistema de reservación de locales docentes. Se muestran las diferentes tareas de ingeniería que se desarrollan y los estándares de codificación utilizados en el sistema. Se valida la solución propuesta mediante la aplicación de pruebas de aceptación que son un tipo de pruebas de caja negra elaboradas por los clientes, para comprobar la correcta implementación del software.

3.1. Implementación

En esta fase, las historias de usuario se descomponen en tareas de ingeniería y son asignadas a un grupo de desarrollo o a una persona como responsable de su implementación. Dichas tareas, al ser asignadas a programadores, se escriben en lenguaje técnico y no en un lenguaje entendible por el cliente. Las historias de usuario seleccionadas para ser implementadas en cada iteración se van realizando durante el transcurso de la iteración a la cual pertenecen. También se describen los estándares de codificación utilizados en la elaboración del código.

3.1.1 Tareas de ingeniería

Las tareas de ingeniería (TI) son otro artefacto de la metodología XP, estas se establecieron, indicando por cada una de las iteraciones la forma en que se desarrollaría la aplicación. A continuación, se observa la TI correspondiente a las historias de usuario número 1, 19, 20, 21 y 22.

Capítulo 3: Implementación y pruebas al sistema de reservación de locales docentes en la Facultad 1.

Tabla 16. TI Autenticar usuario.

Tarea de ingeniería	
Número de tarea: 1	Número de la HU : 1
Nombre de la tarea: Autenticar usuario	
Tipo de tarea: Desarrollo	Puntos estimados (días): 5
Fecha inicio: 10/11/14	Fecha fin: 14/11/14
Programador responsable: Pedro Vega Martínez	
Descripción: El usuario entra sus datos (usuario y contraseña), el sistema verifica que los datos entrados estén correctos y que el usuario pertenezca a la Facultad 1, en caso correcto se le permite el acceso, en caso contrario se le informa un mensaje de aviso indicándole la negativa del acceso.	

Tabla 17. TI Insertar reservación.

Tarea de ingeniería	
Número de tarea: 19	Número de la HU : 19
Nombre de la tarea: Insertar reservación	
Tipo de tarea: Desarrollo	Puntos estimados (días): 3
Fecha inicio: 21/1/15	Fecha fin: 23/1/1
Programador responsable: Pedro Vega Martínez	
Descripción: El usuario accede a la interfaz de la aplicación que le permite insertar una reservación, llena los datos que se le solicitan, por último selecciona la opción "Insertar reservación". El sistema verifica que los campos obligatorios estén llenos, de no ser así mostrará un mensaje indicándolo y no mostrará los locales que están reservados en el horario seleccionado. En caso de que se quiera reservar en el horario de 12:01 AM a 7:59 AM el sistema deberá mostrar un mensaje indicando que a esa hora no se pueden realizar reservaciones.	

Capítulo 3: Implementación y pruebas al sistema de reservación de locales docentes en la Facultad 1.

Tabla 18. TI Listar reservaciones.

Tarea de ingeniería	
Número de tarea: 20	Número de la HU : 20
Nombre de la tarea: Listar reservaciones	
Tipo de tarea: Desarrollo	Puntos estimados (días): 2
Fecha inicio: 24/1/15	Fecha fin: 26/1/15
Programador responsable: Pedro Vega Martínez	
Descripción: El usuario selecciona la opción “Listar reservaciones”, el sistema muestra una vista con las reservaciones insertadas previamente. El usuario con el rol de “Solicitante” solamente podrá listar sus reservaciones, mientras que el rol “Planificador” además de poder listar sus reservaciones, tendrá la opción de listar todas las reservaciones insertadas en el sistema.	

Tabla 19. TI Modificar reservación.

Tarea de ingeniería	
Número de tarea: 21	Número de la HU : 21
Nombre de la tarea: Modificar reservación	
Tipo de tarea: Desarrollo	Puntos estimados (días): 3
Fecha inicio: 27/1/15	Fecha fin: 29/1/15
Programador responsable: Pedro Vega Martínez	
Descripción: El usuario accede a la interfaz de la aplicación que le permite modificar una reservación, modifica los datos que necesite, por último selecciona la opción “Modificar reservación”. El sistema verifica que los campos obligatorios estén llenos, de no ser así mostrará un mensaje indicándolo y no mostrará los locales que están reservados en el horario seleccionado. El usuario con el rol de “Solicitante” solamente podrá modificar sus reservaciones, mientras que el rol “Planificador” además de poder modificar sus reservaciones, tendrá la opción de modificar cualquier reservación insertada en el sistema.	

Tabla 20. TI Eliminar reservación.

Tarea de ingeniería	
Número de tarea: 22	Número de la HU : 22
Nombre de la tarea: Eliminar reservación	
Tipo de tarea: Desarrollo	Puntos estimados (días): 2
Fecha inicio: 30/1/15	Fecha fin: 31/1/15
Programador responsable: Pedro Vega Martínez	
Descripción: El usuario podrá eliminar cualquier reservación hecha por él previamente en el sistema. El usuario con el rol de “Solicitante” solamente podrá eliminar sus reservaciones, mientras que el rol “Planificador” además de poder eliminar sus reservaciones, tendrá la opción de eliminar cualquier reservación insertada en el sistema. El sistema mostrará un mensaje para confirmar que se desea eliminar la reservación seleccionada.	

3.1.2. Estándares de codificación

Los estándares de codificación son pautas de programación que no están enfocadas a la lógica del programa, sino a su estructura y apariencia física para facilitar la lectura, comprensión y mantenimiento del código (33).

Adoptar un estándar permite enfocarse más en el código que en el formato del mismo. Asimismo, aporta consistencia pues las diferentes porciones de código guardan un patrón que se puede reconocer, mejorando así la legibilidad del mismo. En consecuencia, el mantenimiento es menos pesado como resultado de los aportes anteriormente mencionados. A continuación se muestran los estándares definidos para la implementación del sistema de reservación de locales docentes en la Facultad 1.

Nomenclatura de las clases: Los nombres de las clases comienzan con la primera letra en mayúscula y el resto en minúscula, en caso de que sea un nombre compuesto se separan con el signo “_” y se sigue con minúscula.

Ejemplo:

- Nombre simple: *class Local*.
- Nombre compuesto: *class Tipo_local*.

Nomenclatura de los métodos y las variables: El nombre a emplear para los métodos y las variables se escribe con la primera palabra en minúscula, en caso de que sea un nombre compuesto se empleará notación *CamelCase*, que consiste en que la primera letra de la segunda palabra comience con mayúscula.

Ejemplo:

- Nombre simple: *public function eliminar()*.
- Nombre compuesto: *public function listarMedios(\$id)*.

Nomenclatura de las llamadas de funciones: Las funciones deben ser llamadas sin espacios entre el nombre de la función, el signo de paréntesis y el primer parámetro, espacios entre cada coma por parámetro y sin espacios entre el último paréntesis, el signo de paréntesis cerrado y el signo de punto y coma al final.

Ejemplo: *obtenerReservacionesPorTiempo(\$desde/1000,\$hasta/1000);*

Nomenclatura de los comentarios: Para comentar una línea se utiliza *//*, si es un bloque de texto se comienza con */** y termina con **/*.

Ejemplo:

```
// Código de función
```

```
/* Descripción del método */
```

Nomenclatura de las instrucciones: Las llaves que se utilizan para abrir y cerrar un bloque de instrucciones deben estar al mismo nivel.

3.2. Pruebas de software

La calidad de los sistemas informáticos se ha convertido en uno de los principales objetivos estratégicos de las organizaciones, puesto que su éxito depende en gran medida de esta, siendo las pruebas un elemento decisivo en la obtención de la calidad del producto de software. Esto permite a los desarrolladores detectar posibles fallos para solucionarlos a tiempo y con el menor costo posible. Teniendo en cuenta esta práctica se reduce el número de errores no detectados así como la posibilidad de ocurrencia de un error. Esto contribuye a elevar la calidad de los productos desarrollados.

XP enfatiza mucho los aspectos relacionados con las pruebas, clasificándolas en diferentes tipos y funcionalidades específicas, indicando quién, cuándo y cómo deben ser implementadas y ejecutadas. Dicha metodología divide las pruebas en dos grupos: pruebas unitarias, desarrolladas por los programadores, encargadas de verificar el código de forma automática y las pruebas de aceptación, destinadas a evaluar si al final de una iteración se obtuvo la funcionalidad requerida, además de comprobar que dicha funcionalidad sea la esperada por el cliente. Un equipo de desarrollo que siga el proceso de XP debe primero probar su código de forma unitaria, integrar este a la aplicación y después realizar las pruebas de aceptación (40).

3.2.1. Pruebas de aceptación

Las pruebas de aceptación están destinados a evaluar si al final de una iteración se obtuvo la funcionalidad requerida, además comprobar que dicha funcionalidad sea la esperada por el cliente. Las pruebas de aceptación son creadas a partir de las HU, estas son consideradas como pruebas de caja negra; durante una iteración la HU seleccionada en la planificación de iteraciones se convertirá en una prueba de aceptación. Una HU puede tener una prueba de aceptación o tantas como sean necesarias para garantizar su correcto funcionamiento. El cliente o usuario en conjunto con el encargado de pruebas, especifican los aspectos a probar, por lo que es responsabilidad de estos verificar la corrección de las pruebas de aceptación (41). En este caso el aspecto que se tomó en cuenta en conjunto con el cliente fue que las funcionalidades estuvieran implementadas correctamente. El objetivo final de estas pruebas es garantizar que los requerimientos sean cumplidos y que el sistema sea aceptable.

Los casos de prueba son agregados a los artefactos que se le entregan al cliente al terminar cada fase o iteración del proyecto. A continuación en las tablas 19 y 20 se muestran algunos de los casos de pruebas de la primera iteración, el resto se puede encontrar en los [Anexos](#).

Tabla 21. Prueba 1: Autenticar con datos no válidos.

Caso de prueba de aceptación	
Código: 1	Historia de Usuario #1: Autenticar usuario
Nombre: Autenticar con datos no válidos	
Responsable: Pedro Vega Martínez	
Descripción: Probar que se pueda autenticar un usuario en el sistema con al menos un dato no válido.	
Condiciones de ejecución: Usuario y/o contraseña no puede(n) ser válido(s).	
Entradas/ Pasos de ejecución: Intentar acceder al sistema con al menos un dato (Usuario y/o contraseña) no válido (Entrar solo uno de ellos y dejar el otro vacío, que usuario y/o contraseña no estén registrados en el LDAP o que no pertenezca a la Facultad 1).	
Resultado esperado: El sistema informa al usuario que su usuario y/o contraseña no son válidos, y se le da la posibilidad de intentarlo nuevamente.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 22. Prueba 20: Insertar una nueva reservación.

Caso de prueba de aceptación	
Código: 20	Historia de Usuario #19: Insertar reservación
Nombre: Insertar una nueva reservación	
Responsable: Pedro Vega Martínez	
Descripción: Prueba de funcionalidad para la inserción de una reservación en el sistema.	
Condiciones de ejecución: El usuario debe estar registrado previamente en el sistema con su permiso pertinente.	
Entradas/ Pasos de ejecución: El usuario procede a la inserción de los datos necesarios para realizar la reservación y se ejecuta la acción.	
Resultado esperado: La reservación es insertada sin generar error. En caso que el usuario no inserte algún dato obligatorio, el sistema informará que existen campos obligatorios vacíos. Los locales que estén reservados a la hora y fecha que se solicita la reservación no se deben mostrar en la lista de locales a reservar.	
Evaluación de la prueba: Prueba satisfactoria.	

Validación de los resultados

En la metodología XP, la aplicación de pruebas se realiza de forma iterativa para comprobar la correcta implementación de las historias de usuario. Con la realización de los casos de pruebas de aceptación pertenecientes a cada historia de usuario se obtuvo una lista de no conformidades.

Las pruebas se realizaron de la siguiente forma: La primera iteración de pruebas se realizó una vez implementadas las HU de la primera iteración de desarrollo, para validar que estas fueron implementadas correctamente. Posteriormente se corrigieron las HU que arrojaron no conformidades, se implementaron las HU de la segunda iteración de desarrollo y se realizó la segunda iteración de pruebas. En esta iteración se probaron nuevamente las HU de la primera iteración de desarrollo, más las HU de la segunda iteración de desarrollo. Se corrigieron las HU que arrojaron no conformidades, se implementaron las HU de la tercera iteración de desarrollo y se realizó la tercera iteración de pruebas donde se probaron todas las HU. Seguidamente se corrigieron las HU que arrojaron no conformidades, se realizó una cuarta iteración de pruebas nuevamente a todas las HU y se comprobó que todas las no conformidades fueron erradicadas.

A continuación se refleja en la Figura 7, la cantidad de funcionalidades probadas y las no conformidades obtenidas por cada iteración.

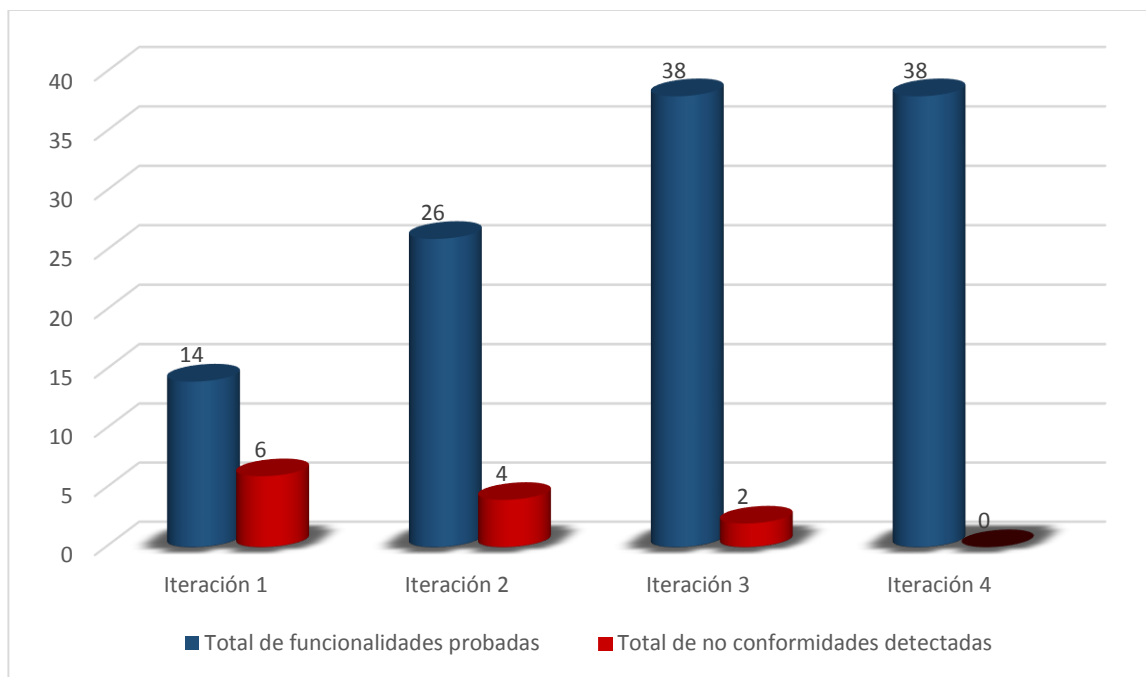


Fig 6. Total de funcionalidades probadas y no conformidades detectadas por iteración.

A continuación se desglosa para cada iteración el error detectado:

Iteración 1.

Se realizaron un total de 14 casos de pruebas detectándose 6 no conformidades. A continuación se desglosan los casos de prueba de aceptación que arrojaron alguna no conformidad durante la ejecución de la primera iteración:

1. Insertar tipo de local
2. Modificar tipo de local
3. Insertar local
4. Modificar local
5. Insertar tipo de medio
6. Modificar tipo de medio

Iteración 2.

Se realizaron un total de 26 casos de pruebas detectándose 4 no conformidades. A continuación se desglosan los casos de prueba de aceptación que arrojaron alguna no conformidad durante la ejecución de la segunda iteración:

1. Eliminar medios de locales
2. Eliminar reservación
3. Insertar reservación
4. Mostrar horario por día, semana o mes

Iteración 3.

Se realizaron un total de 38 casos de pruebas detectándose 2 no conformidades. A continuación se desglosan los casos de prueba de aceptación que arrojaron alguna no conformidad durante la ejecución de la tercera iteración de pruebas:

1. Insertar rol
2. Modificar rol

Iteración 4.

Se realizaron un total de 38 casos de pruebas y no se detectaron no conformidades.

La plantilla de no conformidades constituye un registro de los defectos y fallos encontrados en el transcurso de las pruebas, cuyo principal objetivo es verificar en un futuro que estos errores fueron erradicados en posteriores iteraciones. La tabla 18 muestra un resumen del registro de defectos y dificultades encontradas durante la ejecución de las iteraciones de pruebas:

Capítulo 3: Implementación y pruebas al sistema de reservación de locales docentes en la Facultad 1.

Tabla 23. Resumen de no conformidades.

Iteración	No.	No conformidad	Respuesta del desarrollador
1	1	La inserción permitió crear un tipo de local con el mismo nombre que otro que ya se había insertado.	Resuelto una vez terminada la primera iteración de pruebas.
	2	La modificación permitió modificar un tipo de local con el mismo nombre de otro que ya se había insertado.	Resuelto una vez terminada la primera iteración de pruebas.
	3	La inserción permitió crear un local con el mismo nombre que otro que ya se había insertado.	Resuelto una vez terminada la primera iteración de pruebas.
	4	La modificación permitió modificar un local con el mismo nombre de otro que ya existía.	Resuelto una vez terminada la primera iteración de pruebas.
	5	La inserción permitió crear un tipo de medio con el mismo nombre que otro que ya se había insertado.	Resuelto una vez terminada la primera iteración de pruebas.
	6	La modificación permitió modificar un tipo de medio con el mismo nombre de otro que ya existía.	Resuelto una vez terminada la primera iteración de pruebas.
2	1	La opción eliminar medios de locales no eliminó el medio seleccionado de la base de datos.	Resuelto una vez terminada la segunda iteración de pruebas.

Capítulo 3: Implementación y pruebas al sistema de reservación de locales docentes en la Facultad 1.

	2	La opción eliminar reservación no eliminó la reservación seleccionada de la base de datos.	Resuelto una vez terminada la segunda iteración de pruebas.
	3	La opción insertar reservación permitía insertar una reservación con fecha de inicio mayor a fecha de fin.	Resuelto una vez terminada la segunda iteración de pruebas
	4	La opción de mostrar reservaciones por día no mostraba las reservaciones realizadas el día seleccionado.	Resuelto una vez terminada la segunda iteración de pruebas.
3	1	La inserción permitió dejar el campo vacío de nombre del rol.	Resuelto una vez terminada la tercera iteración de pruebas.
	2	La inserción permitió dejar el campo vacío de modificar del rol.	Resuelto una vez terminada la tercera iteración de pruebas.

Con la validación de los resultados se mostró la evaluación del producto que se está probando de acuerdo a todos los defectos y fallos del sistema encontrados a lo largo del proceso. Fueron probados los 38 casos de pruebas de aceptación diseñados en los cuales se detectaron 12 no conformidades. Las pruebas se realizaron en cuatro iteraciones, al final de cada iteración se corrigieron las no conformidades encontradas. Al concluir la cuarta iteración, los resultados de los 38 casos de pruebas fueron satisfactorios.

3.2.2 Pruebas de carga y estrés

Las pruebas de carga y estrés se realizan al sistema antes de que sea puesto en funcionamiento, sin este tipo de prueba es imposible saber que pasaría cuando la aplicación se encuentre con cargas elevadas, es decir, una gran cantidad de usuarios realizando peticiones al sistema, a las cuales se deben dar respuestas rápidas. Estas pruebas tienen grandes ventajas ya que permiten identificar riesgos antes de que el sistema

Capítulo 3: Implementación y pruebas al sistema de reservación de locales docentes en la Facultad 1.

sea utilizado y evitar problemas de rendimientos. Lo que permite garantizar un desempeño óptimo en la aplicación.

Las pruebas fueron realizadas utilizando la herramienta Apache JMeter. El sistema fue instalado en un entorno de pruebas con las siguientes características en el servidor:

- Microprocesador: Intel(R) dual core E6300 2.8 ghz 1066 2m
- Memoria RAM: 2.00 GB
- Disco duro: 500 GB
- Sistema Operativo: kubuntu 14.10.

El resultado de la prueba se interpreta de la siguiente tabla, donde se ejecutó la prueba para un total de 90 muestras (90 usuarios conectados a través de la URL) concurrentes, donde cada usuario realiza una nueva petición cada 1 segundo.

Informe Agregado

Nombre: Informe Agregado

Comentarios

Escribir todos los datos a Archivo

Nombre de archivo Log/Display Only: Escribir en Log Sólo Errores Successes

Label	# Muestras	Media	Mediana	Linea de 90%	Min	Máx	% Error	Rendimiento	Kb/sec
/assets/js/s...	90	63	7	47	1	2757	0,00%	40,8/min	.1
/assets/js/s...	90	37	7	37	1	2202	0,00%	40,8/min	20,7
/assets/js/j...	90	35	8	68	1	1294	0,00%	40,8/min	.2
/assets/js/j...	90	59	7	94	1	2100	0,00%	40,4/min	.6
/assets/js/j...	90	39	8	68	0	1748	0,00%	40,4/min	.6
/favicon.ico	90	37	12	46	1	1238	0,00%	40,4/min	1,8
/index.php/s...	90	22	8	63	0	213	0,00%	40,4/min	.6
/index.php	90	53	9	72	0	2838	0,00%	39,9/min	2,1
/index.php/s...	90	21	12	45	0	177	0,00%	39,9/min	2,4
/assets/js/...	90	16	10	40	0	159	0,00%	39,9/min	1,9
/assets/js/j...	90	16	8	42	1	152	0,00%	39,9/min	2,0
/assets/js/j...	90	20	11	43	0	299	0,00%	39,9/min	1,9
/assets/js/j...	90	19	10	47	0	144	0,00%	39,9/min	.4
/assets/js/j...	90	36	9	46	1	1804	0,00%	39,9/min	2,2
/assets/js/j...	90	31	9	44	0	1297	0,00%	39,9/min	2,1
/assets/js/j...	90	15	9	38	1	103	0,00%	39,9/min	2,1
/assets/js/j...	90	27	8	32	0	1289	0,00%	39,9/min	2,0
/assets/js/j...	90	26	11	49	0	439	0,00%	39,9/min	1,9
/assets/js/j...	90	19	9	61	0	151	0,00%	39,9/min	2,3
/assets/js/j...	90	19	9	50	0	212	0,00%	39,9/min	2,3
/index.php/r...	90	62	9	44	0	1928	0,00%	39,8/min	.1
/assets/js/...	90	17	9	39	1	222	0,00%	39,8/min	2,1
/assets/js/j...	90	57	12	154	1	1930	0,00%	39,8/min	253,7
/assets/js/j...	90	26	10	77	0	194	0,00%	39,8/min	2,0
/assets/js/j...	90	19	11	57	1	161	0,00%	39,8/min	2,2
/index.php/r...	90	46	7	36	0	2814	0,00%	39,8/min	2,0
/reservacio...	90	22	9	57	1	197	0,00%	39,8/min	.3
/index.php/r...	180	31180	32178	38679	14927	67664	0,00%	56,4/min	18,2
TOTAL	30330	746	7	67	0	115036	0,00%	102,4/sec	1061,0

Include group name in label?

Fig 7. Resultados de las pruebas de carga y estrés.

Las pruebas de carga y estrés arrojaron buenos resultados, por lo que se puede afirmar que la aplicación cumple con el rendimiento esperado. El sistema respondió para 90 usuarios conectados concurrentemente y un total de 30330 peticiones, con un rendimiento de 102,4/seg (102,4 peticiones / segundo), con un 0.00% de error.

Conclusiones parciales

En el transcurso de este capítulo se ha mostrado que una historia de usuario no es solo un resumen de una funcionalidad que debe tener el sistema, los desarrolladores van ampliando sus conocimientos a medida que se va implementando, siendo el cliente quien tiene la idea de que es lo que quiere y que va a comprobar mediante las pruebas de aceptación. Por ello, es más práctico que sea el cliente quien elabore las pruebas.

La utilización de los estándares de codificación, permitió la obtención del código del sistema de manera limpia y organizada, siendo esta entendible para cualquier programador que utilice el código desarrollado.

Se validó el sistema propuesto mediante la realización de pruebas de aceptación a las HU, donde las pruebas fueron evaluadas por parte del cliente. A través de estas pruebas y de las de carga y estrés realizadas al sistema se pudo asegurar el correcto funcionamiento de la aplicación desarrollada.

Conclusiones

La realización de este trabajo de diploma permitió entre otras actividades realizar un análisis del estado del arte sobre diferentes sistemas de gestión de reservaciones existentes. A partir de este análisis se propusieron objetivos que fueron cumplidos y se arriba a las siguientes conclusiones:

- El estudio de las diferentes herramientas para la reservación de locales permitió seleccionar las principales características del sistema de reservación de locales docentes en la Facultad 1. Este análisis permitió además seleccionar elementos que aportaron ideas para la futura implementación del Sistema.
- Se estudiaron y seleccionaron las tecnologías y herramientas más idóneas para el desarrollo del sistema, teniendo en cuenta tecnologías libres, por las que no hay que realizar pagos de licencias.
- Mediante la aplicación de la metodología XP se logró la realización del análisis y diseño, permitiendo desarrollar un sistema informático acorde con las necesidades planteadas para realizar el proceso de reservación de locales docentes en la Facultad 1.
- Se diseñó un sistema que permite gestionar la reservación de locales en la Facultad 1. Generando los diferentes artefactos requeridos por la metodología seleccionada.
- El sistema desarrollado permite la reservación de locales en la Facultad 1 teniendo en cuenta la disposición tecnológica de los mismos y las necesidades de los usuarios. El sistema de reservación automatiza el proceso y mejora la gestión de las reservaciones.
- Se aplicaron pruebas de aceptación al sistema para validar la propuesta desarrollada. Las pruebas, permitieron constatar que las funcionalidades implementadas cumplen con los requisitos establecidos.

Recomendaciones

Luego de terminada la investigación, el autor recomienda:

- Garantizar que el sistema implementado tenga debida atención y actualización para posibles cambios e incorporación de nuevas funcionalidades.
- Extender el uso del sistema a las demás facultades de la universidad, para que sirva como herramienta en la gestión de reservaciones de locales docentes.

Referencias bibliográficas

1. Universidad de las Ciencias Informáticas. [En línea] 2012. [Citado el: 4 de Enero de 2015.] <http://www.uci.cu/?q=mision>.
2. RAE. *Diccionario de la Lengua Española*. [En línea] 2012. [Citado el: 2015 de mayo de 12.] <http://lema.rae.es/drae/?val=servicio>.
3. RAE. *Diccionario de la Lengua Española*. [En línea] 2012. [Citado el: 12 de mayo de 2015.] <http://lema.rae.es/drae/?val=reservacion>.
4. *APLICACIONES WEB INTEGRALES. BREVES APUNTES*. Nieves Naranjo, David Leonardo. 3, Holguín : SOCICT, septiembre de 2008, ENTRE LINEAS, Vol. 2. 1818-3018.
5. Sistema de Solicitud de Reserva de Aulas. [En línea] [Citado el: 13 de Octubre de 2014.] <http://farmaplica.us.es/ssra/week.php?year=2015&month=06&day=12>.
6. Bello del Pino, Daily y Fernández Fernández., Susana Alicia. *Desarrollo del Sistema para la Reservación de Tiempos de Máquina de los Laboratorios en la Universidad de las Ciencias Informáticas*. Habana : s.n., 2009.
7. González, Yanelys Hernández y Ribero Carballo, Milerkys. *Gestión y Control de los Laboratorios de la Facultad 7*. Habana : s.n., 2007. pág. 88.
8. Delgado, Manuel Matos y Fumero Paz, Orley. *Sistema Integral de Gestión de los Laboratorios. Módulo de reservación de tiempo de máquina en la UCI*. Habana : s.n., 2008. pág. 96.
9. Frías, Roberley Cuadra. *Desarrollo de una librería de componentes de interfaz gráfica de usuario basada en Swing Application Framework y AWT Framework*. UCI. Habana : s.n., 2012.
10. Dominguez, Yoevis Diaz. *Personalización del Open Harvester Systems*. Habana : s.n., 2013.
11. López González, Pascual, González López, Ana Amelia y Gallud Lázaro, José Antonio. Herramientas Case. ¿Cómo incorporarlas con éxito en nuestra organización? [En línea] [Citado el: 12 de febrero de 2015.] dialnet.unirioja.es/descarga/articulo/2282561.pdf.
12. *Extensión de Visual Paradigm for UML para el desarrollo dirigido por modelos de aplicaciones de gestión de información*. Cabrera González, Lianet y Pompa Torres, Enrique Roberto. 10, La Habana : s.n., 15 de octubre de 2012, Serie Científica de la Universidad de las Ciencias Informáticas, Vol. 5.
13. Morales, Humberto Martín y Abreu Lugo, Marlon . *Herramienta para la gestión de contenido y configuración de ejercicios y clases reutilizables*. UCI. Habana : s.n., 2009.

14. Ciberaula, Una introducción a Apache. [En línea] 2010. [Citado el: 2015 de marzo de 20.] http://linux.ciberaula.com/articulo/linux_apache_intro.
15. Rodríguez, Luis Olfrides Pérez y González Martínez, Daliana . *Sistema de Control de Laboratorios (CLAB)*. UCI. Habana : s.n., 2014.
16. ciberaula. [En línea] 2014. [Citado el: 19 de Septiembre de 2014.] http://php.ciberaula.com/articulo/introduccion_php.
17. Almaraz Hernández, Jesús Matías , Campos Cantero, Pablo y Castelo Delgado, Tamara . Desarrollo de una aplicación Web para la gestión de Entornos Virtuales. [En línea] 2011. [Citado el: 4 de marzo de 2015.] http://eprints.ucm.es/13083/1/Memoria_SI_Final.pdf.
18. Pérez, Javier Eguíluz. Introducción a JavaScript. [En línea] 2008. [Citado el: 2 de mayo de 2015.] http://sunshine.prod.uci.cu/gridfs/sunshine/books/introduccion_javascript_2caras.pdf.
19. Flanagan, David. *JavaScript. La Guía Definitiva*. España : s.n., 2007. ISBN: 978-84-415-2202-2 84-415-2202-2.
20. Domínguez, Leandro Núñez y Abella Paumier, Ariannis . *Sistema para gestionar las pruebas no funcionales según las sub-características de Calidad*. UCI. Habana : s.n., 2013.
21. CodeIgniter. [En línea] 1 de mayo de 2015. [Citado el: 6 de junio de 2015.] http://www.codeigniter.com/user_guide/overview/at_a_glance.html.
22. getbootstrap. [En línea] 2 de diciembre de 2012. [Citado el: 12 de mayo de 2015.] <http://getbootstrap.com/2.3.2/scaffolding.html#responsive>.
23. Jquery. [En línea] 2015. [Citado el: 25 de mayo de 2015.] <http://jquery.com/>.
24. Bienvenido a NetBeans y www.netbeans.org. [En línea] 2015. [Citado el: 19 de mayo de 2015.] https://netbeans.org/index_es.html.
25. Cobo, Angel. *Diseño y programación de bases de datos*. Madrid : Vision Libros. ISBN: 978-84-9821-459-8.
26. Martínez, Rafael. PostgreSQL. [En línea] 2 de 10 de 2010. [Citado el: 12 de febrero de 2015.] http://www.postgresql.org.es/sobre_postgresql.
27. Arjona Miranda, Roberto. Estudio y propuesta de metodologías de desarrollo para los proyectos de la Facultad 1 según su alcance. [En línea] 4 de julio de 2008. [Citado el: 5 de enero de 2015.] http://repositorio_institucional.uci.cu/jspui/bitstream/ident/TD_1689_08/1/TD_1689_08.pdf.
28. Sommerville, Iam. *Ingeniería de software*. [ed.] Addison-Wesley. [trad.] María Isabel Alfonso Galipienso y Antonio Botía Martínez. séptima. Madrid : PEARSON EDUCACIÓN, 2005. pág. 188. 84-7829-074-5.

29. Fernández Escribano, Gerardo. Universidad de Mendoza. [En línea] 2002. [Citado el: 15 de mayo de 2015.] <http://www.um.edu.ar/catedras/claroline/backends/download.php?url=L01ldG9kb3NfQWdpbGVzL1Byb2dyYW1hY2lvb19FeHRyZW1hLVhQLnBkZg%3D%3D&cidReset=true&cidReq=II0162004>.
30. ECHEVERRY TOBÓN, LUIS MIGUEL y DELGADO CARMONA, LUZ ELENA. Repositorio.utp. [En línea] 2007. [Citado el: 5 de mayo de 2015.] <http://repositorio.utp.edu.co/dspace/bitstream/11059/794/1/0053E18cp.pdf>.
31. ESPIN, WILSON ALEJANDRO HIDALGO. *ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE AUTOMATIZACIÓN Y CONTROL ORIENTADO A LA GENERACIÓN DE RETENCIONES DEL ROL DE PAGOS EN AMBIENTE*. UNIVERSIDAD POLITÉCNICA SALESIANA. Quito : s.n., 2013.
32. Ingenio DS. [En línea] 2013. <https://ingeniods.wordpress.com/2013/09/16/patrones-arquitectonicos/>.
33. Quevedo Campins, Virgen Damaris, y otros. *CEDRUX. Solución para sistemas de control*. La Habana : s.n., 2010.
34. Márquez Gómez, José Jorge . *Arquitectura MVC Visión General*. 2011.
35. Merlino, H., Vranić, A., Rodríguez, D., Pytel, P., García-Martínez, R. SEDICI. [En línea] [Citado el: 13 de marzo de 2015.] http://sedici.unlp.edu.ar/bitstream/handle/10915/19554/Documento_completo.pdf?sequence=1.
36. Ing. Sánchez Rios, Sergio. *Metodologías de Análisis y Diseño*. 2015.
37. Angelfire. Geet the planet. [En línea] 11 de Mayo de 2011. [Citado el: 2 de Mayo de 2015.] <http://geektheplanet.net/5462/patrones-gof.xhtml#..>
38. MARTÍNEZ JUAN, FRANCISCO JAVIER. *GUÍA DE CONSTRUCCIÓN DE SOFTWARE EN JAVA CON*. Oviedo : s.n.
39. Marca Huallpara , Hugo Michael y Quisbert Limachi , Nancy Susana. [En línea] virtual.usalesiana.edu.bo/web/practica/archiv/despliegue.doc.
40. Rodríguez Corbea, Maite y Ordóñez Pérez , Meylin . *LA METODOLOGÍA XP APLICABLE AL DESARROLLO DEL*. La Habana : s.n., 2007.
41. J. J. Gutiérrez, y otros. *PRUEBAS DEL SISTEMA EN PROGRAMACIÓN*. Sevilla : s.n. pág. 12.

Anexos

Anexo 1: Historias de usuario

Tabla 24. HU Insertar tipo de local.

Historia de Usuario	
Número: 2	Nombre Historia de Usuario: Insertar tipo de local
Modificación de Historia de Usuario Número: Ninguna	
Actor: Administrador	Iteración Asignada: 2
Prioridad en Negocio: Alta	Puntos Estimados: 3
Riesgo en Desarrollo: Baja	
Descripción: El administrador podrá adicionar tipos de local en el sistema.	
Observaciones:	

Tabla 25. HU Listar tipos de locales.

Historia de Usuario	
Número: 3	Nombre Historia de Usuario: Listar tipos de locales
Modificación de Historia de Usuario Número: Ninguna	
Actor: Administrador	Iteración Asignada: 2
Prioridad en Negocio: Alta	Puntos Estimados: 2
Riesgo en Desarrollo: Baja	
Descripción: El administrador podrá mostrar una lista con todos los tipos de locales que se adicionaron previamente en el sistema.	
Observaciones:	

Tabla 26. HU Modificar tipo de local.

Historia de Usuario	
Número: 4	Nombre Historia de Usuario: Modificar tipo de local
Modificación de Historia de Usuario Número: Ninguna	
Actor: Administrador	Iteración Asignada: 2
Prioridad en Negocio: Alta	Puntos Estimados: 3
Riesgo en Desarrollo: Baja	
Descripción: El administrador podrá modificar los datos de cualquier tipo de local creado previamente en el sistema.	
Observaciones:	

Tabla 27. HU Eliminar tipo de local.

Historia de Usuario	
Número: 5	Nombre Historia de Usuario: Eliminar tipo de local
Modificación de Historia de Usuario Número: Ninguna	
Actor: Administrador	Iteración Asignada: 2
Prioridad en Negocio: Alta	Puntos Estimados: 2
Riesgo en Desarrollo: Baja	
Descripción: El administrador podrá eliminar cualquier tipo de local creado previamente en el sistema.	
Observaciones:	

Tabla 28. HU Listar locales.

Historia de Usuario	
Número: 7	Nombre Historia de Usuario: Listar locales
Modificación de Historia de Usuario Número: Ninguna	
Actor: Administrador	Iteración Asignada: 1
Prioridad en Negocio: Alta	Puntos Estimados: 2
Riesgo en Desarrollo: Baja	
Descripción: El administrador podrá mostrar una lista con todos los locales que se adicionaron previamente en el sistema.	
Observaciones:	

Tabla 29. HU Modificar local.

Historia de Usuario	
Número: 8	Nombre Historia de Usuario: Modificar local
Modificación de Historia de Usuario Número: Ninguna	
Actor: Administrador	Iteración Asignada: 1
Prioridad en Negocio: Alta	Puntos Estimados: 3
Riesgo en Desarrollo: Baja	
Descripción: El administrador podrá modificar los datos de cualquier local creado previamente en el sistema.	
Observaciones:	

Tabla 30. HU Eliminar local.

Historia de Usuario	
Número: 9	Nombre Historia de Usuario: Eliminar local
Modificación de Historia de Usuario Número: Ninguna	
Actor: Administrador	Iteración Asignada: 1
Prioridad en Negocio: Alta	Puntos Estimados: 2
Riesgo en Desarrollo: Baja	
Descripción El administrador podrá eliminar cualquier local creado previamente en el sistema.	
Observaciones:	

Tabla 31. HU Insertar tipo de medio.

Historia de Usuario	
Número: 10	Nombre Historia de Usuario: Insertar tipo de medio
Modificación de Historia de Usuario Número: Ninguna	
Actor: Administrador	Iteración Asignada: 2
Prioridad en Negocio: Alta	Puntos Estimados: 3
Riesgo en Desarrollo: Baja	
Descripción: El administrador podrá adicionar tipos de medio en el sistema.	
Observaciones:	

Tabla 32. HU Listar tipos de medios.

Historia de Usuario	
Número: 11	Nombre Historia de Usuario: Listar tipos de medios
Modificación de Historia de Usuario Número: Ninguna	
Actor: Administrador	Iteración Asignada: 2
Prioridad en Negocio: Alta	Puntos Estimados: 2
Riesgo en Desarrollo: Baja	
Descripción: El administrador podrá mostrar una lista con todos los tipos de medios que se adicionaron previamente en el sistema.	
Observaciones:	

Tabla 33. HU Modificar tipo de medio.

Historia de Usuario	
Número: 12	Nombre Historia de Usuario: Modificar tipo de medio
Modificación de Historia de Usuario Número: Ninguna	
Actor: Administrador	Iteración Asignada: 2
Prioridad en Negocio: Alta	Puntos Estimados: 3
Riesgo en Desarrollo: Baja	
Descripción: El administrador podrá modificar los datos de cualquier tipo de medio creado previamente en el sistema.	
Observaciones:	

Tabla 34. HU Eliminar tipo de medio.

Historia de Usuario	
Número: 13	Nombre Historia de Usuario: Eliminar tipo de medio
Modificación de Historia de Usuario Número: Ninguna	
Actor: Administrador	Iteración Asignada: 2
Prioridad en Negocio: Alta	Puntos Estimados: 2
Riesgo en Desarrollo: Baja	
Descripción: El administrador podrá eliminar cualquier tipo de medio creado previamente en el sistema.	
Observaciones:	

Tabla 35. HU Insertar medios a los locales.

Historia de Usuario	
Número: 14	Nombre Historia de Usuario: Insertar medios a los locales
Modificación de Historia de Usuario Número: Ninguna	
Actor: Administrador	Iteración Asignada: 2
Prioridad en Negocio: Alta	Puntos Estimados: 3
Riesgo en Desarrollo: Baja	
Descripción: El administrador podrá adicionarle medios a los locales.	
Observaciones:	

Tabla 36. HU Listar medios de locales.

Historia de Usuario	
Número: 15	Nombre Historia de Usuario: Listar medios de locales
Modificación de Historia de Usuario Número: Ninguna	
Actor: Administrador	Iteración Asignada: 2
Prioridad en Negocio: Alta	Puntos Estimados: 2
Riesgo en Desarrollo: Baja	
Descripción: El administrador podrá mostrar una lista con todos los medios que pertenezcan al local seleccionado.	
Observaciones:	

Tabla 37. HU Modificar medios de locales.

Historia de Usuario	
Número: 16	Nombre Historia de Usuario: Modificar medios de locales
Modificación de Historia de Usuario Número: Ninguna	
Actor: Administrador	Iteración Asignada: 2
Prioridad en Negocio: Alta	Puntos Estimados: 3
Riesgo en Desarrollo: Baja	
Descripción: El administrador podrá modificar los datos de los medios asignados a los locales.	
Observaciones:	

Tabla 38. HU Eliminar medios de locales.

Historia de Usuario	
Número: 17	Nombre Historia de Usuario: Eliminar medios de locales
Modificación de Historia de Usuario Número: Ninguna	
Actor: Administrador	Iteración Asignada: 2
Prioridad en Negocio: Alta	Puntos Estimados: 2
Riesgo en Desarrollo: Baja	
Descripción: El administrador podrá eliminar cualquier medio del local seleccionado.	
Observaciones:	

Tabla 39. HU Mostrar horario por día, semana o mes.

Historia de Usuario	
Número: 18	Nombre Historia de Usuario: Mostrar horario por día, semana o mes
Modificación de Historia de Usuario Número: Ninguna	
Actor: Solicitante, planificadora y administrador	Iteración Asignada: 1
Prioridad en Negocio: Alta	Puntos Estimados: 3
Riesgo en Desarrollo: Media	
Descripción: Los usuarios podrán mostrar un calendario donde aparezcan todas las reservaciones creadas previamente en el sistema, podrá mostrar las reservaciones por día, semana o mes.	
Observaciones:	

Tabla 40. HU Listar reservaciones.

Historia de Usuario	
Número: 20	Nombre Historia de Usuario: Listar reservaciones
Modificación de Historia de Usuario Número: Ninguna	
Actor: Solicitante, planificadora y administrador	Iteración Asignada: 1
Prioridad en Negocio: Alta	Puntos Estimados: 2
Riesgo en Desarrollo: Media	
Descripción: El usuario podrá mostrar una lista con todas las reservaciones realizadas por él previamente en el sistema.	
Observaciones: El usuario con el rol de “Planificador” además de sus reservaciones podrá mostrar también una lista con todas las reservaciones realizadas previamente en el sistema.	

Tabla 41. HU Modificar reservación.

Historia de Usuario	
Número: 21	Nombre Historia de Usuario: Modificar reservación
Modificación de Historia de Usuario Número: Ninguna	
Actor: Solicitante, planificadora y administrador	Iteración Asignada: 1
Prioridad en Negocio: Alta	Puntos Estimados: 3
Riesgo en Desarrollo: Alta	
Descripción: El usuario podrá modificar los datos de cualquier reservación creada por él previamente en el sistema.	
Observaciones: El usuario con el rol de “Planificador” además de poder modificar sus reservaciones, tendrá la opción de modificar cualquier reservación creada previamente en el sistema.	

Tabla 42. HU Eliminar reservación.

Historia de Usuario	
Número: 22	Nombre Historia de Usuario: Eliminar reservación
Modificación de Historia de Usuario Número: Ninguna	
Actor: Solicitante, planificadora y administrador	Iteración Asignada: 1
Prioridad en Negocio: Alta	Puntos Estimados: 2
Riesgo en Desarrollo: Alta	
Descripción: El usuario podrá eliminar cualquier reservación hecha por él previamente en el sistema.	
Observaciones: El usuario con el rol de “Planificador” además de poder eliminar sus reservaciones, tendrá la opción de eliminar cualquier reservación insertada previamente en el sistema.	

Tabla 43. HU Listar usuarios.

Historia de Usuario	
Número: 23	Nombre Historia de Usuario: Listar usuarios
Modificación de Historia de Usuario Número: Ninguna	
Actor: Administrador	Iteración Asignada: 3
Prioridad en Negocio: Media	Puntos Estimados: 2
Riesgo en Desarrollo: Baja	
Descripción: El administrador podrá mostrar una lista con todos los usuarios autenticados previamente en el sistema.	
Observaciones:	

Tabla 44. HU Modificar usuario.

Historia de Usuario	
Número: 24	Nombre Historia de Usuario: Modificar usuario
Modificación de Historia de Usuario Número: Ninguna	
Actor: Administrador	Iteración Asignada: 3
Prioridad en Negocio: Media	Puntos Estimados: 3
Riesgo en Desarrollo: Baja	
Descripción: El administrador podrá modificar los datos de cualquier usuario autenticado previamente en el sistema.	
Observaciones: Solamente se podrá modificar el rol del usuario.	

Tabla 45. HU Eliminar usuario.

Historia de Usuario	
Número: 25	Nombre Historia de Usuario: Eliminar usuario
Modificación de Historia de Usuario Número: Ninguna	
Actor: Administrador	Iteración Asignada: 3
Prioridad en Negocio: Media	Puntos Estimados: 2
Riesgo en Desarrollo: Baja	
Descripción: El administrador podrá eliminar cualquier usuario autenticado previamente en el sistema.	
Observaciones:	

Tabla 46. HU Insertar rol.

Historia de Usuario	
Número: 26	Nombre Historia de Usuario: Insertar rol
Modificación de Historia de Usuario Número: Ninguna	
Actor: Administrador	Iteración Asignada: 3
Prioridad en Negocio: Media	Puntos Estimados: 3
Riesgo en Desarrollo: Baja	
Descripción: El administrador podrá insertar un rol en el sistema.	
Observaciones:	

Tabla 47. HU Listar rol.

Historia de Usuario	
Número: 27	Nombre Historia de Usuario: Listar rol
Modificación de Historia de Usuario Número: Ninguna	
Actor: Administrador	Iteración Asignada: 3
Prioridad en Negocio: Media	Puntos Estimados: 2
Riesgo en Desarrollo: Baja	
Descripción: El administrador podrá mostrar una lista con todos los roles creados previamente en el sistema.	
Observaciones:	

Tabla 48. HU Modificar rol.

Historia de Usuario	
Número: 28	Nombre Historia de Usuario: Modificar rol
Modificación de Historia de Usuario Número: Ninguna	
Actor: Administrador	Iteración Asignada: 3
Prioridad en Negocio: Media	Puntos Estimados: 3
Riesgo en Desarrollo: Baja	
Descripción: El administrador podrá modificar los datos de cualquier rol creado previamente en el sistema.	
Observaciones:	

Tabla 49. HU Eliminar rol.

Historia de Usuario	
Número: 29	Nombre Historia de Usuario: Eliminar rol
Modificación de Historia de Usuario Número: Ninguna	
Actor: Administrador	Iteración Asignada: 3
Prioridad en Negocio: Media	Puntos Estimados: 2
Riesgo en Desarrollo: Baja	
Descripción: El administrador podrá eliminar cualquier rol creado previamente en el sistema.	
Observaciones:	

Tabla 50. HU Mostrar reportes.

Historia de Usuario	
Número: 30	Nombre Historia de Usuario: Mostrar reportes
Modificación de Historia de Usuario Número: Ninguna	
Actor: Solicitante, planificadora y administrador	Iteración Asignada: 2
Prioridad en Negocio: Media	Puntos Estimados: 5
Riesgo en Desarrollo: Alta	
Descripción: El usuario podrá mostrar un reporte donde se muestre la cantidad de medios de cada local y la cantidad de reservaciones.	
Observaciones:	

Tabla 51. HU Insertar menú del sistema.

Historia de Usuario	
Número: 31	Nombre Historia de Usuario: Insertar menú del sistema
Modificación de Historia de Usuario Número: Ninguna	
Actor: Administrador	Iteración Asignada: 3
Prioridad en Negocio: Baja	Puntos Estimados: 3
Riesgo en Desarrollo: Media	
Descripción: El administrador podrá insertar un menú en el sistema.	
Observaciones:	

Tabla 52. HU Listar menús del sistema.

Historia de Usuario	
Número: 32	Nombre Historia de Usuario: Listar menús del sistema
Modificación de Historia de Usuario Número: Ninguna	
Actor: Administrador	Iteración Asignada: 3
Prioridad en Negocio: Baja	Puntos Estimados: 2
Riesgo en Desarrollo: Media	
Descripción: El administrador podrá mostrar una lista con todos los menús creados previamente en el sistema.	
Observaciones:	

Tabla 53. HU Modificar menú del sistema.

Historia de Usuario	
Número: 33	Nombre Historia de Usuario: Modificar menú del sistema
Modificación de Historia de Usuario Número: Ninguna	
Actor: Administrador	Iteración Asignada: 3
Prioridad en Negocio: Baja	Puntos Estimados: 3
Riesgo en Desarrollo: Media	
Descripción: El administrador podrá modificar los datos de cualquier menú creado previamente en el sistema.	
Observaciones:	

Tabla 54. HU Eliminar menú del sistema.

Historia de Usuario	
Número: 34	Nombre Historia de Usuario: Eliminar menú del sistema
Modificación de Historia de Usuario Número: Ninguna	
Actor: Administrador	Iteración Asignada: 3
Prioridad en Negocio: Baja	Puntos Estimados: 2
Riesgo en Desarrollo: Media	
Descripción: El administrador podrá eliminar cualquier menú creado previamente en el sistema.	
Observaciones:	

Anexo 2: Descripción de las tablas de la base de datos.

Tabla 55. Descripción de la tabla reserva.

Nombre: reserva		
Descripción: En esta tabla se recoge las reservaciones realizadas en el sistema.		
Atributo	Tipo	Descripción
id	integer	identificador de la tabla
id_usuario	integer	identificador del usuario que realizó la reservación
id_tipo_actividad	integer	identificador del tipo de actividad a realizar
observaciones	text	alguna observación relacionada con la reservación

Tabla 56. Descripción de la tabla tipo_actividad.

Nombre: tipo_actividad		
Descripción: En esta tabla se recoge el tipo de actividad que se puede realizar.		
Atributo	Tipo	Descripción
id	integer	identificador de la tabla
nombre	varchar	nombre del tipo de actividad
descripción	varchar	una descripción de la actividad

Tabla 57. Descripción de la tabla horario.

Nombre: horario		
Descripción: En esta tabla se recoge el horario en el que se va a realizar la actividad.		
Atributo	Tipo	Descripción
id_local	integer	identificador de la tabla
id_reserva	integer	identificador de la reservación
fecha_inicio	timestamp	fecha y hora en la que comienza la actividad
fecha_fin	timestamp	fecha y hora en la que termina la actividad

Tabla 58. Descripción de la tabla local.

Nombre: local		
Descripción: En esta tabla se recoge todos los locales existentes en el sistema.		
Atributo	Tipo	Descripción
id	integer	identificador de la tabla
nombre	varchar	nombre del local
descripción	varchar	descripción del local
id_tipo_local	integer	identificador del tipo de local

Tabla 59. Descripción de la tabla tipo_local

Nombre: tipo_local		
Descripción: En esta tabla se recogen todos los tipos de locales existentes en el sistema.		
Atributo	Tipo	Descripción
id	integer	identificador de la tabla
nombre	varchar	nombre del tipo de local
descripción	varchar	descripción del tipo de local

Tabla 60. Descripción de la tabla tipo_local_tipo_medio.

Nombre: tipo_local_tipo_medio		
Descripción: En esta tabla se recoge una relación entre tipo de local y tipo de medio		
Atributo	Tipo	Descripción
id_tipo_local	integer	identificador de tipo de local
id_tipo_medio	integer	identificador de tipo de medio

Tabla 61. Descripción de la tabla tipo_medio.

Nombre: tipo_medio		
Descripción: En esta tabla se recogen todos datos de los tipos de medios.		
Atributo	Tipo	Descripción
id	integer	identificador de la tabla
nombre	varchar	nombre del tipo de medio
descripción	varchar	descripción del tipo de medio

Tabla 62. Descripción de la tabla tipo_medio_características.

Nombre: tipo_medio_características		
Descripción: En esta tabla se recoge una relación entre tipo de medio y las características.		
Atributo	Tipo	Descripción
id_tipo_medio	integer	identificador de tipo de medio
id_características	integer	identificador de la característica

Tabla 63. Descripción de la tabla características.

Nombre: características		
Descripción: En esta tabla se recogen las características que pueden tener los distintos tipos de medios.		
Atributo	Tipo	Descripción
id	integer	identificador de la tabla
nombre	varchar	nombre de la característica
descripción	varchar	descripción de la característica
dominio	integer	posibles datos que puede tener la característica

Tabla 64. Descripción de la tabla medio_características.

Nombre: medio_características		
Descripción: En esta tabla se recoge el valor de las características del medio		
Atributo	Tipo	Descripción
id_medio	integer	identificador del medio
id_características	integer	identificador de la característica
valor	text	el valor de la característica que va a tener el medio

Tabla 65. Descripción de la tabla medio.

Nombre: medio		
Descripción: En esta tabla se recogen todos los medios creados en el sistema.		
Atributo	Tipo	Descripción
id	integer	identificador de la tabla
nombre	varchar	nombre del medio
descripción	varchar	descripción del medio
id_tipo_medio	integer	identificador de tipo de medio

Tabla 66. Descripción de la tabla local_medio.

Nombre: local_medio		
Descripción: En esta tabla se recoge una relación entre los locales y sus medios.		
Atributo	Tipo	Descripción
id_local	integer	identificador del local
id_medio	integer	identificador del medio

Tabla 67. Descripción de la tabla usuario.

Nombre: usuario		
Descripción: En esta tabla se recogen los datos de los usuarios que se registran en el sistema.		
Atributo	Tipo	Descripción
id	integer	identificador de la tabla
usuario	text	nombre del usuario
nombre_completo	text	nombre completo del usuario
correo	text	correo del usuario
id_rol	integer	identificador del rol
activo	bool	si el usuario está activo o no
credencial	varchar	la credencial del usuario
categoría	varchar	categoría del usuario
área	varchar	área a la que pertenece el usuario
cargo	varchar	cargo que ocupa el usuario
foto	varchar	foto del usuario

Tabla 68. Descripción de la tabla autenticación_activa.

Nombre: autenticación_activa		
Descripción: En esta tabla se recogen los datos de los usuarios activos en el sistema.		
Atributo	Tipo	Descripción
id	integer	identificador de la tabla
id_usuario	integer	identificador del usuario
satisfactoria	bool	si se autenticó correctamente
token	varchar	comprobación de seguridad de las actividades del usuario.
fecha_registro	timestamp	fecha en la que se autenticó en el sistema
ip_acceso	varchar	ip por el que accedió al sistema

Tabla 69. Descripción de la tabla rol.

Nombre: rol		
Descripción: En esta tabla se recogen todos datos de los roles del sistema.		
Atributo	Tipo	Descripción
id	integer	identificador de la tabla
nombre_rol	varchar	nombre del rol
descripción	text	descripción del rol
activo	bool	si está activo o no
automático	bool	si tiene acceso a todas las funcionalidades del sistema

Tabla 70. Descripción de la tabla rol_menú.

Nombre: rol_menú		
Descripción: En esta tabla se guarda la relación entre la tabla rol y la tabla menú.		
Atributo	Tipo	Descripción
id_menú	integer	identificador de la tabla menú
id_acción	integer	identificador de la tabla acción
orden	varchar	orden en el que se muestran los menús del sistema según el rol que este autenticado

Tabla 71. Descripción de la tabla menú.

Nombre: menú		
Descripción: En esta tabla se recogen todos menús que va a tener el sistema.		
Atributo	Tipo	Descripción
id	integer	identificador de la tabla
nombre_menú	varchar	nombre del menú
descripción	text	descripción del menú
activo	bool	si está activo o no
icon_class	varchar	el ícono que saldrá

Tabla 72. Descripción de la tabla menú_acción.

Nombre: menú_acción		
Descripción: En esta tabla se guarda la relación entre la tabla menú y la tabla acción		
Atributo	Tipo	Descripción
id_menú	integer	identificador de la tabla menú
id_acción	integer	identificador de la tabla acción
orden	varchar	Orden en que van a salir los submenús en el sistema según el usuario autenticado.

Tabla 73. Descripción de la tabla acción.

Nombre: acción		
Descripción: En esta tabla se recogen las acciones que se van a realizar cuando seleccionamos un menú.		
Atributo	Tipo	Descripción
id	integer	identificador de la tabla
nombre_acción	varchar	nombre de la acción que va a realizar
descripción	text	descripción de la acción
icon_class	varchar	ícono de la acción
activo	bool	si está activa o no
nombre_visual	varchar	nombre que se ve en el menú la acción
id_funcionalidad	integer	id de la funcionalidad que va a ejecutar la acción
principal	bool	si el menú de la acción se va a mostrar o no

Tabla 74. Descripción de la tabla acción_url.

Nombre: acción_url		
Descripción: En esta tabla se recoge el url de todas las funcionalidades que va a tener el sistema.		
Atributo	Tipo	Descripción
id	integer	identificador de la tabla
id_acción	integer	identificador de la acción que va a realizar
url	varchar	muestra la url de la funcionalidad
principal	bool	si se muestra en el menú principal

Anexo 3: Tareas de ingeniería

Tabla 75. TI Insertar tipo de local.

Tarea de ingeniería	
Número de tarea: 2	Número de la HU : 2
Nombre de la tarea: Insertar tipo de local	
Tipo de tarea: Desarrollo	Puntos estimados (días): 3
Fecha inicio: 15/11/14	Fecha fin: 18/11/14
Programador responsable: Pedro Vega Martínez	
Descripción: El administrador accede a la interfaz de la aplicación que le permite insertar un tipo de local, llena los datos que se le solicitan, por último selecciona la opción "Insertar tipo de local". El sistema verifica que los datos obligatorios estén llenos y que no exista ningún otro tipo de local con el mismo nombre, en caso de que exista un local con el mismo nombre o quede algún campo obligatorio vacío el sistema mostrará un mensaje indicando el error.	

Tabla 76. TI Listar tipos de locales.

Tarea de ingeniería	
Número de tarea: 3	Número de la HU : 3
Nombre de la tarea: Listar tipos de locales	
Tipo de tarea: Desarrollo	Puntos estimados (días): 2
Fecha inicio: 19/11/14	Fecha fin: 20/11/14
Programador responsable: Pedro Vega Martínez	
Descripción: El administrador selecciona la opción "Listar tipos de locales", el sistema muestra una vista con los tipos de locales que se han creado hasta el momento.	

Tabla 77. TI Modificar tipos de locales.

Tarea de ingeniería	
Número de tarea: 4	Número de la HU : 4
Nombre de la tarea: Modificar tipo de local	
Tipo de tarea: Desarrollo	Puntos estimados (días): 3
Fecha inicio: 21/11/14	Fecha fin: 24/11/14
Programador responsable: Pedro Vega Martínez	
<p>Descripción: El administrador accede a la interfaz de la aplicación que le permite modificar un tipo de local, modifica los datos que necesite, por último selecciona la opción “Modificar tipo de local”. El sistema verifica que los datos obligatorios estén llenos y que no exista ningún otro tipo de local con el mismo nombre, en caso de que exista un local con el mismo nombre o quede algún campo obligatorio vacío el sistema mostrará un mensaje indicando el error.</p>	

Tabla 78. TI Eliminar tipo de local.

Tarea de ingeniería	
Número de tarea: 5	Número de la HU : 5
Nombre de la tarea: Eliminar tipo de local	
Tipo de tarea: Desarrollo	Puntos estimados (días): 2
Fecha inicio: 25/11/14	Fecha fin: 26/1/14
Programador responsable: Pedro Vega Martínez	
<p>Descripción: El administrador podrá eliminar cualquier tipo de local creado previamente en el sistema. El sistema mostrará un mensaje para confirmar que se desea eliminar el tipo de local seleccionado.</p>	

Tabla 79. TI Insertar local.

Tarea de ingeniería	
Número de tarea: 6	Número de la HU : 6
Nombre de la tarea: Insertar local	
Tipo de tarea: Desarrollo	Puntos estimados (días): 3
Fecha inicio: 27/11/14	Fecha fin: 29/11/14
Programador responsable: Pedro Vega Martínez	
Descripción: El administrador accede a la interfaz de la aplicación que le permite insertar un local, llena los datos que se le solicitan, por último selecciona la opción "Insertar". El sistema verifica que los datos obligatorios estén llenos y que no exista ningún local con el mismo nombre, en caso de que exista un local con el mismo nombre o quede algún campo obligatorio vacío el sistema mostrará un mensaje indicando el error.	

Tabla 80. TI Listar locales.

Tarea de ingeniería	
Número de tarea: 7	Número de la HU : 7
Nombre de la tarea: Listar locales	
Tipo de tarea: Desarrollo	Puntos estimados (días): 2
Fecha inicio: 1/12/14	Fecha fin: 2/12/14
Programador responsable: Pedro Vega Martínez	
Descripción: El administrador selecciona la opción "Listar locales", el sistema muestra una vista con los locales que se han creado hasta el momento.	

Tabla 81. TI Modificar local.

Tarea de ingeniería	
Número de tarea: 8	Número de la HU : 8
Nombre de la tarea: Modificar local	
Tipo de tarea: Desarrollo	Puntos estimados (días): 3
Fecha inicio: 3/12/14	Fecha fin: 5/12/14
Programador responsable: Pedro Vega Martínez	
<p>Descripción: El administrador accede a la interfaz de la aplicación que le permite modificar un local, modifica los datos que necesite, por último selecciona la opción “Modificar local”. El sistema verifica que los datos obligatorios estén llenos y que no exista ningún local con el mismo nombre, en caso de que exista un local con el mismo nombre o quede algún campo obligatorio vacío el sistema mostrará un mensaje indicando el error.</p>	

Tabla 82. TI Eliminar local.

Tarea de ingeniería	
Número de tarea: 9	Número de la HU : 9
Nombre de la tarea: Eliminar local	
Tipo de tarea: Desarrollo	Puntos estimados (días): 2
Fecha inicio: 6/12/14	Fecha fin: 8/12/14
Programador responsable: Pedro Vega Martínez	
<p>Descripción: El administrador podrá eliminar cualquier local creado previamente en el sistema. El sistema mostrará un mensaje para confirmar que se desea eliminar el local seleccionado.</p>	

Tabla 83. TI Insertar tipo de medio.

Tarea de ingeniería	
Número de tarea: 10	Número de la HU : 10
Nombre de la tarea: Insertar tipo de medio	
Tipo de tarea: Desarrollo	Puntos estimados (días): 3
Fecha inicio: 9/12/14	Fecha fin: 11/12/14
Programador responsable: Pedro Vega Martínez	
Descripción: El administrador accede a la interfaz de la aplicación que le permite insertar un tipo de local, llena los datos que se le solicitan, por último selecciona la opción "Insertar tipo de medio". El sistema verifica que los datos obligatorios estén llenos y que no exista ningún otro tipo de medio con el mismo nombre, en caso de que exista un tipo de medio con el mismo nombre o quede algún campo obligatorio vacío el sistema mostrará un mensaje indicando el error.	

Tabla 84. TI Listar tipos de medios.

Tarea de ingeniería	
Número de tarea: 11	Número de la HU : 11
Nombre de la tarea: Listar tipos de medios	
Tipo de tarea: Desarrollo	Puntos estimados (días): 2
Fecha inicio: 12/12/14	Fecha fin: 13/12/14
Programador responsable: Pedro Vega Martínez	
Descripción: El administrador selecciona la opción "Listar tipos de medio", el sistema muestra una vista con los tipos de medio que se han creado hasta el momento.	

Tabla 85. TI Modificar tipo de medio.

Tarea de ingeniería	
Número de tarea: 12	Número de la HU : 12
Nombre de la tarea: Modificar tipo de medio	
Tipo de tarea: Desarrollo	Puntos estimados (días): 3
Fecha inicio: 15/12/14	Fecha fin: 17/12/14
Programador responsable: Pedro Vega Martínez	
<p>Descripción: El administrador accede a la interfaz de la aplicación que le permite modificar un tipo de medio, modifica los datos que necesite, por último selecciona la opción “Modificar tipo de medio”. El sistema verifica que los campos obligatorios estén llenos y que no exista otro tipo de medio con el mismo nombre, en caso de que exista un tipo de medio con el mismo nombre o quede algún campo obligatorio vacío el sistema mostrará un mensaje indicando el error.</p>	

Tabla 86. TI Eliminar tipo de medio.

Tarea de ingeniería	
Número de tarea: 13	Número de la HU : 13
Nombre de la tarea: Eliminar tipo de medio	
Tipo de tarea: Desarrollo	Puntos estimados (días): 2
Fecha inicio: 18/12/14	Fecha fin: 19/12/14
Programador responsable: Pedro Vega Martínez	
<p>Descripción: El administrador podrá eliminar cualquier tipo de medio creado previamente en el sistema. El sistema mostrará un mensaje para confirmar que se desea eliminar el tipo de medio seleccionado.</p>	

Tabla 87. TI Insertar medios a locales.

Tarea de ingeniería	
Número de tarea: 14	Número de la HU : 14
Nombre de la tarea: Insertar medios a los locales	
Tipo de tarea: Desarrollo	Puntos estimados (días): 3
Fecha inicio: 6/1/15	Fecha fin: 8/1/15
Programador responsable: Pedro Vega Martínez	
Descripción: El administrador accede a la interfaz de la aplicación que le permite insertar un medio al local seleccionado, llena los datos que se le solicitan, por último selecciona la opción "Insertar medio". El sistema verifica que los datos obligatorios estén llenos y que no exista ningún otro medio en el local con el mismo nombre, en caso de que exista un medio con el mismo nombre en el local o quede algún campo obligatorio vacío el sistema mostrará un mensaje indicando el error.	

Tabla 88. TI Listar medios de locales.

Tarea de ingeniería	
Número de tarea: 15	Número de la HU : 15
Nombre de la tarea: Listar medios de locales	
Tipo de tarea: Desarrollo	Puntos estimados (días): 2
Fecha inicio: 9/1/15	Fecha fin: 10/1/15
Programador responsable: Pedro Vega Martínez	
Descripción: El administrador selecciona la opción "Listar medio" del local seleccionado, el sistema muestra una vista con los medios del local.	

Tabla 89. TI Modificar medios de locales.

Tarea de ingeniería	
Número de tarea: 16	Número de la HU : 16
Nombre de la tarea: Modificar medios de locales	
Tipo de tarea: Desarrollo	Puntos estimados (días): 3
Fecha inicio: 12/1/15	Fecha fin: 14/11/15
Programador responsable: Pedro Vega Martínez	
<p>Descripción: El administrador accede a la interfaz de la aplicación que le permite modificar un medio, modifica los datos que necesite, por último selecciona la opción “Modificar medio”. El sistema verifica que los campos obligatorios estén llenos y que no exista otro medio con el mismo nombre en el local, en caso de que exista un medio con el mismo nombre en el local o quede algún campo obligatorio vacío el sistema mostrará un mensaje indicando el error.</p>	

Tabla 90. TI Eliminar medios de locales.

Tarea de ingeniería	
Número de tarea: 17	Número de la HU : 17
Nombre de la tarea: Eliminar medios de locales	
Tipo de tarea: Desarrollo	Puntos estimados (días): 2
Fecha inicio: 15/1/15	Fecha fin: 16/1/15
Programador responsable: Pedro Vega Martínez	
<p>Descripción: El administrador podrá eliminar cualquier medio del local seleccionado que se halla asignado previamente en el sistema. El sistema mostrará un mensaje para confirmar que se desea eliminar al local el medio seleccionado.</p>	

Tabla 91. TI Mostrar horario por día, semana o mes.

Tarea de ingeniería	
Número de tarea: 18	Número de la HU : 18
Nombre de la tarea: Mostrar horario por día, semana o mes	
Tipo de tarea: Desarrollo	Puntos estimados (días): 3
Fecha inicio: 17/1/15	Fecha fin: 20/1/15
Programador responsable: Pedro Vega Martínez	
Descripción: El usuario selecciona la opción “Mostrar horario”, el sistema muestra todas las reservaciones creadas previamente en un calendario, deberá permitirle al usuario ver las reservaciones filtrando por día, semana o mes.	

Tabla 92. TI Listar usuarios.

Tarea de ingeniería	
Número de tarea: 23	Número de la HU : 23
Nombre de la tarea: Listar usuarios	
Tipo de tarea: Desarrollo	Puntos estimados (días): 2
Fecha inicio: 9/2/15	Fecha fin: 10/2/15
Programador responsable: Pedro Vega Martínez	
Descripción: El administrador selecciona la opción “Listar usuarios”, el sistema muestra una vista con los usuarios que se han adicionado hasta el momento.	

Tabla 93. TI modificar usuario.

Tarea de ingeniería	
Número de tarea: 24	Número de la HU : 24
Nombre de la tarea: Modificar usuario	
Tipo de tarea: Desarrollo	Puntos estimados (días): 3
Fecha inicio: 10/2/15	Fecha fin: 12/2/15
Programador responsable: Pedro Vega Martínez	
Descripción: El administrador accede a la interfaz de la aplicación que le permite modificar un usuario, modifica los datos que necesite, por último selecciona la opción “Modificar usuario”. Solamente se podrá modificar el rol del usuario.	

Tabla 94. TI Eliminar usuario.

Tarea de ingeniería	
Número de tarea: 25	Número de la HU : 25
Nombre de la tarea: Eliminar usuario	
Tipo de tarea: Desarrollo	Puntos estimados (días): 2
Fecha inicio: 13/2/15	Fecha fin: 14/2/15
Programador responsable: Pedro Vega Martínez	
Descripción: El administrador podrá eliminar cualquier usuario insertado previamente en el sistema. El sistema mostrará un mensaje para confirmar que se desea eliminar el usuario seleccionado.	

Tabla 95. TI Insertar rol.

Tarea de ingeniería	
Número de tarea: 26	Número de la HU : 26
Nombre de la tarea: Insertar rol	
Tipo de tarea: Desarrollo	Puntos estimados (días): 3
Fecha inicio: 16/2/15	Fecha fin: 18/2/15
Programador responsable: Pedro Vega Martínez	
Descripción: El administrador accede a la interfaz de la aplicación que le permite insertar un rol, llena los datos que se le solicitan, por último selecciona la opción "Insertar rol". El sistema verifica que los datos obligatorios estén llenos y que no exista ningún otro rol con el mismo nombre, en caso de que exista un rol con el mismo nombre o quede algún campo obligatorio vacío el sistema mostrará un mensaje indicando el error.	

Tabla 96. TI Listar rol.

Tarea de ingeniería	
Número de tarea: 27	Número de la HU : 27
Nombre de la tarea: Listar rol	
Tipo de tarea: Desarrollo	Puntos estimados (días): 2
Fecha inicio: 19/2/15	Fecha fin: 20/3/15
Programador responsable: Pedro Vega Martínez	
Descripción: El administrador selecciona la opción "Listar rol", el sistema muestra una vista con los roles que se han creado hasta el momento.	

Tabla 97. TI Modificar rol.

Tarea de ingeniería	
Número de tarea: 28	Número de la HU : 28
Nombre de la tarea: Modificar rol	
Tipo de tarea: Desarrollo	Puntos estimados (días): 3
Fecha inicio: 21/2/15	Fecha fin: 24/2/15
Programador responsable: Pedro Vega Martínez	
Descripción: El administrador accede a la interfaz de la aplicación que le permite modificar un rol, modifica los datos que necesite, por último selecciona la opción "Modificar rol". El sistema verifica que los campos obligatorios estén llenos y que no exista otro rol con el mismo nombre, en caso de que exista un rol con el mismo nombre o quede algún campo obligatorio vacío el sistema mostrará un mensaje indicando el error.	

Tabla 98. TI Eliminar rol.

Tarea de ingeniería	
Número de tarea: 29	Número de la HU : 29
Nombre de la tarea: Eliminar rol	
Tipo de tarea: Desarrollo	Puntos estimados (días): 2
Fecha inicio: 25/2/15	Fecha fin: 26/2/15
Programador responsable: Pedro Vega Martínez	
Descripción: El administrador podrá eliminar cualquier rol creado previamente en el sistema. El sistema mostrará un mensaje para confirmar que se desea eliminar el rol seleccionado.	

Tabla 99. TI Mostrar reportes.

Tarea de ingeniería	
Número de tarea: 30	Número de la HU : 30
Nombre de la tarea: Mostrar reportes	
Tipo de tarea: Desarrollo	Puntos estimados (días): 5
Fecha inicio: 27/2/15	Fecha fin: 4/3/15
Programador responsable: Pedro Vega Martínez	
Descripción: El administrador selecciona la opción “Mostrar reportes” el sistema muestra una lista con todos los locales donde muestre la cantidad de medios que tiene cada local y la cantidad de reservaciones.	

Tabla 100. TI Insertar menú del sistema.

Tarea de ingeniería	
Número de tarea: 31	Número de la HU : 31
Nombre de la tarea: Insertar menú del sistema	
Tipo de tarea: Desarrollo	Puntos estimados (días): 3
Fecha inicio: 5/3/15	Fecha fin: 7/3/15
Programador responsable: Pedro Vega Martínez	
Descripción: El administrador accede a la interfaz de la aplicación que le permite insertar un menú, llena los datos que se le solicitan, por último selecciona la opción "Insertar menú" .El sistema verifica que los datos obligatorios estén llenos, de no estar llenos envía un mensaje indicando que existen campos obligatorios vacíos.	

Tabla 101. TI Listar menús del sistema.

Tarea de ingeniería	
Número de tarea: 32	Número de la HU : 32
Nombre de la tarea: Listar menús del sistema	
Tipo de tarea: Desarrollo	Puntos estimados (días): 2
Fecha inicio: 9/3/15	Fecha fin: 10/3/15
Programador responsable: Pedro Vega Martínez	
Descripción: El administrador selecciona la opción "Listar menús", el sistema muestra una vista con los menús que se han insertado hasta el momento.	

Tabla 102. TI Modificar menú del sistema.

Tarea de ingeniería	
Número de tarea: 33	Número de la HU : 33
Nombre de la tarea: Modificar menú del sistema	
Tipo de tarea: Desarrollo	Puntos estimados (días): 3
Fecha inicio: 11/3/15	Fecha fin: 13/3/15
Programador responsable: Pedro Vega Martínez	
Descripción: El administrador accede a la interfaz de la aplicación que le permite modificar un menú, modifica los datos que necesite, por último selecciona la opción “Modificar menú”. El sistema verifica que los datos obligatorios estén llenos, de no estar llenos envía un mensaje indicando que existen campos obligatorios vacíos.	

Tabla 103. TI Eliminar menú del sistema.

Tarea de ingeniería	
Número de tarea: 34	Número de la HU : 34
Nombre de la tarea: Eliminar menú del sistema	
Tipo de tarea: Desarrollo	Puntos estimados (días): 2
Fecha inicio: 14/3/15	Fecha fin: 16/3/15
Programador responsable: Pedro Vega Martínez	
Descripción: El administrador podrá eliminar cualquier menú creado previamente en el sistema. El sistema mostrará un mensaje para confirmar que se desea eliminar el menú seleccionado.	

Anexo 4: Casos de pruebas de aceptación

Tabla 104. Prueba 2: Autenticar con datos válidos.

Caso de prueba de aceptación	
Código: 2	Historia de Usuario #1: Autenticar usuario
Nombre: Autenticar con datos válidos	
Responsable: Pedro Vega Martínez	
Descripción: Probar que se pueda autenticar un usuario en el sistema con datos válidos.	
Condiciones de ejecución: Usuario y contraseña deben ser válidos.	
Entradas/ Pasos de ejecución: Intentar acceder al sistema con usuario y contraseña válidos.	
Resultado esperado: Se le muestra una página al usuario según sus privilegios, en la cual se le brindan las funcionalidades correspondientes a su rol.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 105. Prueba 3: Insertar un nuevo tipo de local.

Caso de prueba de aceptación	
Código: 3	Historia de Usuario #2: Insertar tipo de local
Nombre: Insertar un nuevo tipo de local	
Responsable: Pedro Vega Martínez	
Descripción: Prueba de funcionalidad para insertar un tipo de local en el sistema.	
Condiciones de ejecución: El usuario debe estar registrado previamente en el sistema con su permiso pertinente.	
Entradas/ Pasos de ejecución: El usuario procede a la inserción de los datos necesarios para insertar un tipo de local y se ejecuta la acción.	
Resultado esperado: El tipo de local es insertado sin generar error. En caso que el usuario no llene algún dato obligatorio, el sistema informará que existen campos obligatorios vacíos. En caso de que ya exista un tipo de local con el mismo nombre, el sistema informará al usuario que ya existe un tipo de local con ese nombre.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 106. Prueba 4: Listar todos los tipos de locales.

Caso de prueba de aceptación	
Código: 4	Historia de Usuario #3: Listar tipos de locales
Nombre: Listar todos los tipos de locales	
Responsable: Pedro Vega Martínez	
Descripción: Prueba de funcionalidad para listar los tipos de locales insertados previamente en el sistema.	
Condiciones de ejecución: El usuario debe estar registrado previamente en el sistema con su permiso pertinente.	
Entradas/ Pasos de ejecución: El usuario verifica que se muestren en una lista todos los tipos de locales insertados previamente en el sistema.	
Resultado esperado: los tipos de locales son mostrados correctamente.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 107. Prueba 5: Modificar un tipo de local registrado en el sistema.

Caso de prueba de aceptación	
Código: 5	Historia de Usuario #4: Modificar tipo de local
Nombre: Modificar un tipo de local registrado en el sistema	
Responsable: Pedro Vega Martínez	
Descripción: Prueba de funcionalidad para modificar un tipo de local registrado en el sistema.	
Condiciones de ejecución: El usuario debe estar registrado previamente en el sistema con su permiso pertinente.	
Entradas/ Pasos de ejecución: El usuario selecciona un tipo de local y procede a ejecutar la acción modificar tipo de local.	
Resultado esperado: El tipo de local es modificado sin generar error. En caso que el usuario no llene algún dato obligatorio, el sistema informará que existen campos obligatorios vacíos. En caso de que ya exista un tipo de local con el mismo nombre, el sistema informará al usuario que ya existe un tipo de local con ese nombre.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 108. Prueba 6: Eliminar un tipo de local registrado.

Caso de prueba de aceptación	
Código: 6	Historia de Usuario #5: Eliminar tipo de local
Nombre: Eliminar un tipo de local registrado	
Responsable: Pedro Vega Martínez	
Descripción: Prueba de funcionalidad para eliminar un tipo de local registrado en el sistema.	
Condiciones de ejecución: El usuario debe estar registrado previamente en el sistema con su permiso pertinente.	
Entradas/ Pasos de ejecución: El usuario selecciona un tipo de local y procede a ejecutar la acción eliminar tipo de local.	
Resultado esperado: El tipo de local es eliminado correctamente.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 109. Prueba 7: Insertar un nuevo local.

Caso de prueba de aceptación	
Código: 7	Historia de Usuario# 6: Insertar local
Nombre: Insertar un nuevo local	
Responsable: Pedro Vega Martínez	
Descripción: Prueba de funcionalidad para insertar un local en el sistema.	
Condiciones de ejecución: El usuario debe estar registrado previamente en el sistema con su permiso pertinente.	
Entradas/ Pasos de ejecución: El usuario procede a la inserción de los datos necesarios para insertar un local y se ejecuta la acción.	
Resultado esperado: El local es insertado sin generar error. En caso que el usuario no llene algún dato obligatorio, el sistema informará que existen campos obligatorios vacíos. En caso de que ya exista un local con el mismo nombre, el sistema informará al usuario que ya existe un local con ese nombre.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 110. Prueba 8: Listar todos los locales insertados en el sistema.

Caso de prueba de aceptación	
Código: 8	Historia de Usuario #7: Listar locales
Nombre: Listar todos los locales insertados en el sistema	
Responsable: Pedro Vega Martínez	
Descripción: Prueba de funcionalidad para listar los locales registrados en el sistema.	
Condiciones de ejecución: El usuario debe estar registrado previamente en el sistema con su permiso pertinente.	
Entradas/ Pasos de ejecución: El usuario verifica que se muestren en una lista todos los locales insertados previamente.	
Resultado esperado: Los locales son mostrados correctamente.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 111. Prueba 9: Modificar local.

Caso de prueba de aceptación	
Código: 9	Historia de Usuario #8: Modificar local
Nombre: Modificar un local registrado en el sistema	
Responsable: Pedro Vega Martínez	
Descripción: Prueba de funcionalidad para modificar un local registrado en el sistema.	
Condiciones de ejecución: El usuario debe estar registrado previamente en el sistema con su permiso pertinente.	
Entradas/ Pasos de ejecución: El usuario selecciona un local y procede a ejecutar la acción modificar local.	
Resultado esperado: El local es modificado sin generar error. En caso que el usuario no llene algún dato obligatorio, el sistema informará que existen campos obligatorios vacíos. En caso de que ya exista un local con el mismo nombre, el sistema informará al usuario que ya existe un local con ese nombre.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 112. Prueba 10: Eliminar un local registrado.

Caso de prueba de aceptación	
Código: 10	Historia de Usuario #9: Eliminar local
Nombre: Eliminar un local registrado	
Responsable: Pedro Vega Martínez	
Descripción: Prueba de funcionalidad para eliminar un local registrado en el sistema.	
Condiciones de ejecución: El usuario debe estar registrado previamente en el sistema con su permiso pertinente.	
Entradas/ Pasos de ejecución: El usuario selecciona un local y procede a ejecutar la acción eliminar local.	
Resultado esperado: El local es eliminado correctamente.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 113. Prueba 11: Insertar un nuevo tipo de medio.

Caso de prueba de aceptación	
Código: 11	Historia de Usuario #10: Insertar tipo de medio
Nombre: Insertar un nuevo tipo de medio	
Responsable: Pedro Vega Martínez	
Descripción: Prueba de funcionalidad para insertar un tipo de medio en el sistema.	
Condiciones de ejecución: El usuario debe estar registrado previamente en el sistema con su permiso pertinente.	
Entradas/ Pasos de ejecución: El usuario procede a la inserción de los datos necesarios para insertar un tipo de medio y se ejecuta la acción.	
Resultado esperado: El tipo de medio es insertado sin generar error. En caso que el usuario no llene algún dato obligatorio, el sistema informará que existen campos obligatorios vacíos. En caso de que ya exista un tipo de medio con el mismo nombre, el sistema informará al usuario que ya existe un tipo de medio con ese nombre.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 114. Prueba 12: Listar todos los tipos de medios insertados en el sistema.

Caso de prueba de aceptación	
Código: 12	Historia de Usuario #11: Listar tipos de medios
Nombre: Listar todos los tipos de medios insertados en el sistema	
Responsable: Pedro Vega Martínez	
Descripción: Prueba de funcionalidad para listar los tipos de medios insertados en el sistema.	
Condiciones de ejecución: El usuario debe estar registrado previamente en el sistema con su permiso pertinente.	
Entradas/ Pasos de ejecución: El usuario verifica que se muestren en una lista todos los tipos de medios insertados previamente.	
Resultado esperado: Los tipos de medios son mostrados correctamente.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 115. Prueba 13: Modificar un tipo de medio registrado en el sistema.

Caso de prueba de aceptación	
Código: 13	Historia de Usuario #12: Modificar tipo de medio
Nombre: Modificar un tipo de medio registrado en el sistema	
Responsable: Pedro Vega Martínez	
Descripción: Prueba de funcionalidad para modificar un tipo de medio registrado en el sistema.	
Condiciones de ejecución: El usuario debe estar registrado previamente en el sistema con su permiso pertinente.	
Entradas/ Pasos de ejecución: El usuario selecciona un tipo de medio y procede a ejecutar la acción modificar tipo de medio.	
Resultado esperado: El tipo de medio es modificado sin generar error. En caso que el usuario no llene algún dato obligatorio, el sistema informará que existen campos obligatorios vacíos. En caso de que ya exista un tipo de medio con el mismo nombre, el sistema informará al usuario que ya existe un tipo de medio con ese nombre.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 116. Prueba 14: Eliminar un tipo de medio registrado.

Caso de prueba de aceptación	
Código: 14	Historia de Usuario #13: Eliminar tipo de medio
Nombre: Eliminar un tipo de medio registrado	
Responsable: Pedro Vega Martínez	
Descripción: Prueba de funcionalidad para eliminar un tipo de medio registrado en el sistema.	
Condiciones de ejecución: El usuario debe estar registrado previamente en el sistema con su permiso pertinente.	
Entradas/ Pasos de ejecución: El usuario selecciona un tipo de medio y procede a ejecutar la acción eliminar tipo de medio.	
Resultado esperado: El tipo de medio es eliminado correctamente.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 117. Prueba 15: Insertar un nuevo medio a un local.

Caso de prueba de aceptación	
Código: 15	Historia de Usuario #14: Insertar medios a los locales
Nombre: Insertar un nuevo medio a un local	
Responsable: Pedro Vega Martínez	
Descripción: Prueba de funcionalidad para insertar un medio a un local.	
Condiciones de ejecución: El usuario debe estar registrado previamente en el sistema con su permiso pertinente.	
Entradas/ Pasos de ejecución: El usuario procede a la inserción de los datos necesarios para insertar un medio y se ejecuta la acción.	
Resultado esperado: El medio es insertado sin generar error. En caso que el usuario no llene algún dato obligatorio, el sistema informará que existen campos obligatorios vacíos. En caso de que ya exista un medio con el mismo nombre, el sistema informará al usuario que ya existe un medio con ese nombre.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 118. Prueba 16: Listar los medios del local seleccionado.

Caso de prueba de aceptación	
Código: 16	Historia de Usuario #15: Listar medios de locales
Nombre: Listar los medios del local seleccionado	
Responsable: Pedro Vega Martínez	
Descripción: Prueba de funcionalidad para listar los medios insertados en el local seleccionado.	
Condiciones de ejecución: El usuario debe estar registrado previamente en el sistema con su permiso pertinente.	
Entradas/ Pasos de ejecución: El usuario verifica que se muestren en una lista todos los medios insertados previamente en el local seleccionado.	
Resultado esperado: Los medios son mostrados correctamente.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 119. Prueba 17: Modificar un medio del local seleccionado.

Caso de prueba de aceptación	
Código: 17	Historia de Usuario #16: Modificar medios de locales
Nombre: Modificar un medio del local seleccionado	
Responsable: Pedro Vega Martínez	
Descripción: Prueba de funcionalidad para modificar un medio del local seleccionado.	
Condiciones de ejecución: El usuario debe estar registrado previamente en el sistema con su permiso pertinente.	
Entradas/ Pasos de ejecución: El usuario selecciona un medio y procede a ejecutar la acción modificar medio.	
Resultado esperado: El medio es modificado sin generar error. En caso que el usuario no llene algún dato obligatorio, el sistema informará que existen campos obligatorios vacíos. En caso de que ya exista un medio con el mismo nombre, el sistema informará al usuario que ya existe un medio con ese nombre.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 120. Prueba 18: Eliminar un medio de un local.

Caso de prueba de aceptación	
Código: 18	Historia de Usuario #17: Eliminar medio
Nombre: Eliminar un medio de un local	
Responsable: Pedro Vega Martínez	
Descripción: Prueba de funcionalidad para eliminar un medio a un local.	
Condiciones de ejecución: El usuario debe estar registrado previamente en el sistema con su permiso pertinente.	
Entradas/ Pasos de ejecución: El usuario selecciona un medio y procede a ejecutar la acción eliminar medio.	
Resultado esperado: El medio es eliminado correctamente.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 121. Prueba 19: Mostrar en el horario todas las reservaciones.

Caso de prueba de aceptación	
Código: 19	Historia de Usuario #18: Mostrar horario por día, semana o mes
Nombre: Mostrar en el horario todas las reservaciones	
Responsable: Pedro Vega Martínez	
Descripción: Prueba de funcionalidad para mostrar el calendario con todas las reservaciones registradas en el sistemas.	
Condiciones de ejecución: El usuario debe estar registrado previamente en el sistema con su permiso pertinente.	
Entradas/ Pasos de ejecución: El usuario ejecuta la acción mostrar horario.	
Resultado esperado: El usuario verifica que el sistema muestra todas las reservaciones que están registradas y que las pueda buscar por un día, semana o mes en específico.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 122. Prueba 21: Listar mis reservaciones.

Caso de prueba de aceptación	
Código: 21	Historia de Usuario #20: Listar reservaciones
Nombre: Listar mis reservaciones	
Responsable: Pedro Vega Martínez	
Descripción: Prueba de funcionalidad para listar las reservaciones realizadas por el usuario autenticado.	
Condiciones de ejecución: El usuario debe estar registrado previamente en el sistema con su permiso pertinente.	
Entradas/ Pasos de ejecución: El usuario verifica que se muestre en una lista todas las reservaciones que ha insertados previamente.	
Resultado esperado: Las reservaciones son mostradas correctamente.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 123. Prueba 22: Listar todas las reservaciones.

Caso de prueba de aceptación	
Código: 22	Historia de Usuario #20: Listar reservaciones
Nombre: Listar todas las reservaciones	
Responsable: Pedro Vega Martínez	
Descripción: Prueba de funcionalidad para listar las reservaciones realizadas por cualquier usuario en el sistema.	
Condiciones de ejecución: El usuario debe estar registrado previamente en el sistema con su permiso pertinente.	
Entradas/ Pasos de ejecución: El usuario verifica que se muestre en una lista todas las reservaciones insertadas en el sistema.	
Resultado esperado: Las reservaciones son mostradas correctamente.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 124. Prueba 23: Modificar mis reservaciones.

Caso de prueba de aceptación	
Código: 23	Historia de Usuario # 21: Modificar reservación
Nombre: Modificar mis reservaciones	
Responsable: Pedro Vega Martínez	
Descripción: Prueba de funcionalidad para modificar las reservaciones realizadas por el usuario autenticado.	
Condiciones de ejecución: El usuario debe estar registrado previamente en el sistema con su permiso pertinente.	
Entradas/ Pasos de ejecución: El usuario selecciona una reservación y procede a ejecutar la acción modificar reservación.	
Resultado esperado: La reservación es modificada sin generar error. En caso que el usuario no llene algún dato obligatorio, el sistema informará que existen campos obligatorios vacíos. En caso de que el tiempo de la reservación coincida con el tiempo de otra reservación del mismo local, el sistema informará al usuario que existe una reservación en ese horario para el local seleccionado.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 125. Prueba 24: Modificar cualquier reservación.

Caso de prueba de aceptación	
Código: 24	Historia de Usuario #21: Modificar reservación
Nombre: Modificar cualquier reservación	
Responsable: Pedro Vega Martínez	
Descripción: Prueba de funcionalidad para modificar cualquier reservación del sistema.	
Condiciones de ejecución: El usuario debe estar registrado previamente en el sistema con su permiso pertinente.	
Entradas/ Pasos de ejecución: El usuario selecciona una reservación y procede a ejecutar la acción modificar reservación.	
Resultado esperado: La reservación es modificada sin generar error. En caso que el usuario no llene algún dato obligatorio, el sistema informará que existen campos obligatorios vacíos. En caso de que el tiempo de la reservación coincida con el tiempo de otra reservación del mismo local, el sistema informará al usuario que existe una reservación en ese horario para el local seleccionado.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 126. Prueba 25: Eliminar mi reservación.

Caso de prueba de aceptación	
Código: 25	Historia de Usuario #22: Eliminar reservación
Nombre: Eliminar mi reservación	
Responsable: Pedro Vega Martínez	
Descripción: Prueba de funcionalidad para eliminar alguna reservación realizada por el usuario autenticado.	
Condiciones de ejecución: El usuario debe estar registrado previamente en el sistema con su permiso pertinente.	
Entradas/ Pasos de ejecución: El usuario selecciona una reservación y procede a ejecutar la acción eliminar reservación.	
Resultado esperado: La reservación es eliminada correctamente.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 127. Prueba 26: Eliminar cualquier reservación.

Caso de prueba de aceptación	
Código: 26	Historia de Usuario #22: Eliminar reservación
Nombre: Eliminar cualquier reservación	
Responsable: Pedro Vega Martínez	
Descripción: Prueba de funcionalidad para eliminar cualquier reservación insertada previamente en el sistema.	
Condiciones de ejecución: El usuario debe estar registrado previamente en el sistema con su permiso pertinente.	
Entradas/ Pasos de ejecución: El usuario selecciona una reservación y procede a ejecutar la acción eliminar reservación.	
Resultado esperado: La reservación es eliminada correctamente.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 128. Prueba 27: Listar los usuarios del sistema.

Caso de prueba de aceptación	
Código: 27	Historia de Usuario #23: Listar usuarios
Nombre: Listar los usuarios del sistema	
Responsable: Pedro Vega Martínez	
Descripción: Prueba de funcionalidad para listar los usuarios registrados en el sistema.	
Condiciones de ejecución: El usuario debe estar registrado previamente en el sistema con su permiso pertinente.	
Entradas/ Pasos de ejecución: El usuario verifica que se muestren en una lista todos los usuarios insertados previamente en el sistema.	
Resultado esperado: Los usuarios son mostrados correctamente.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 129. Prueba 28: Modificar un usuario del sistema.

Caso de prueba de aceptación	
Código: 28	Historia de Usuario #24: Modificar usuario
Nombre: Modificar un usuario del sistema	
Responsable: Pedro Vega Martínez	
Descripción: Prueba de funcionalidad para modificar un usuario en el sistema.	
Condiciones de ejecución: El usuario debe estar registrado previamente en el sistema con su permiso pertinente.	
Entradas/ Pasos de ejecución: El usuario selecciona el usuario que desee modificar y procede a ejecutar la acción modificar usuario.	
Resultado esperado: El usuario es modificado sin generar error. En caso que el usuario no llene algún dato obligatorio, el sistema informará que existen campos obligatorios vacíos. En caso de que ya exista un usuario con el mismo nombre, el sistema informará al usuario que ya existe un usuario con ese nombre.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 130. Prueba 29: Eliminar un usuario del sistema.

Caso de prueba de aceptación	
Código: 29	Historia de Usuario #25: Eliminar usuario
Nombre: Eliminar un usuario del sistema	
Responsable: Pedro Vega Martínez	
Descripción: Prueba de funcionalidad para eliminar un usuario del sistema.	
Condiciones de ejecución: El usuario debe estar registrado previamente en el sistema con su permiso pertinente.	
Entradas/ Pasos de ejecución: El usuario selecciona el usuario que desee eliminar y procede a ejecutar la acción eliminar usuario.	
Resultado esperado: El usuario es eliminado correctamente.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 131. Prueba 30: Insertar un nuevo rol.

Caso de prueba de aceptación	
Código: 30	Historia de Usuario #26: Insertar rol
Nombre: Insertar un nuevo rol	
Responsable: Pedro Vega Martínez	
Descripción: Prueba de funcionalidad para insertar un rol.	
Condiciones de ejecución: El usuario debe estar registrado previamente en el sistema con su permiso pertinente.	
Entradas/ Pasos de ejecución: El usuario procede a la inserción de los datos necesarios para insertar un rol y se ejecuta la acción.	
Resultado esperado: El medio es insertado sin generar error. En caso que el usuario no llene algún dato obligatorio, el sistema informará que existen campos obligatorios vacíos. En caso de que ya exista un rol con el mismo nombre, el sistema informará al usuario que ya existe un rol con ese nombre.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 132. Prueba 31: Listar los roles del sistema.

Caso de prueba de aceptación	
Código: 31	Historia de Usuario #27: Listar rol
Nombre: Listar los roles del sistema	
Responsable: Pedro Vega Martínez	
Descripción: Prueba de funcionalidad para listar los roles del sistema.	
Condiciones de ejecución: El usuario debe estar registrado previamente en el sistema con su permiso pertinente.	
Entradas/ Pasos de ejecución: El usuario verifica que se muestren en una lista todos los roles insertados previamente en el sistema.	
Resultado esperado: Los roles son mostrados correctamente.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 133. Prueba 32: Modificar un rol en el sistema.

Caso de prueba de aceptación	
Código: 32	Historia de Usuario #28: Modificar rol
Nombre: Modificar un rol en el sistema	
Responsable: Pedro Vega Martínez	
Descripción: Prueba de funcionalidad para modificar un rol en el sistema.	
Condiciones de ejecución: El usuario debe estar registrado previamente en el sistema con su permiso pertinente.	
Entradas/ Pasos de ejecución: El usuario selecciona el rol que desee modificar y procede a ejecutar la acción modificar rol.	
Resultado esperado: El rol es modificado sin generar error. En caso que el usuario no llene algún dato obligatorio, el sistema informará que existen campos obligatorios vacíos. En caso de que ya exista un rol con el mismo nombre, el sistema informará al usuario que ya existe un rol con ese nombre.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 134. Prueba 33: Eliminar un rol del sistema.

Caso de prueba de aceptación	
Código: 33	Historia de Usuario #29: Eliminar rol
Nombre: Eliminar un rol del sistema	
Responsable: Pedro Vega Martínez	
Descripción: Prueba de funcionalidad para eliminar un rol del sistema.	
Condiciones de ejecución: El usuario debe estar registrado previamente en el sistema con su permiso pertinente.	
Entradas/ Pasos de ejecución: El usuario selecciona el rol que desee eliminar y procede a ejecutar la acción eliminar rol.	
Resultado esperado: El rol es eliminado correctamente.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 135. Prueba 34: Mostrar reportes de los laboratorios.

Caso de prueba de aceptación	
Código: 34	Historia de Usuario #30: Mostrar reportes
Nombre: Mostrar reportes de los laboratorios	
Responsable: Pedro Vega Martínez	
Descripción: Prueba de funcionalidad para mostrar reportes de los laboratorios.	
Condiciones de ejecución: El usuario debe estar registrado previamente en el sistema con su permiso pertinente.	
Entradas/ Pasos de ejecución: El usuario ejecuta la acción mostrar reportes.	
Resultado esperado: El usuario verifica que el sistema muestra un reporte con la cantidad de computadoras de cada laboratorio que les funciona el audio, y la cantidad de computadoras que tengan 256 MB, 512 MB, 768 MB, 1 GB, 2 GB, y 4 GB de memoria RAM.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 136. Prueba 35: Insertar un nuevo menú.

Caso de prueba de aceptación	
Código: 35	Historia de Usuario #31: Insertar menú del sistema
Nombre: Insertar un nuevo menú	
Responsable: Pedro Vega Martínez	
Descripción: Prueba de funcionalidad para insertar un menú.	
Condiciones de ejecución: El usuario debe estar registrado previamente en el sistema con su permiso pertinente.	
Entradas/ Pasos de ejecución: El usuario procede a la inserción de los datos necesarios para insertar un menú y se ejecuta la acción.	
Resultado esperado: El menú es insertado sin generar error. En caso que el usuario no llene algún dato obligatorio, el sistema informará que existen campos obligatorios vacíos. En caso de que ya exista un menú con el mismo nombre, el sistema informará al usuario que ya existe un menú con ese nombre.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 137. Prueba 36: Listar los menús del sistema.

Caso de prueba de aceptación	
Código: 36	Historia de Usuario #32: Listar menús del sistema
Nombre: Listar los menús del sistema	
Responsable: Pedro Vega Martínez	
Descripción: Prueba de funcionalidad para listar los menús del sistema.	
Condiciones de ejecución: El usuario debe estar registrado previamente en el sistema con su permiso pertinente.	
Entradas/ Pasos de ejecución: El usuario verifica que se muestren en una lista todos los menús insertados previamente en el sistema.	
Resultado esperado: Los menús son mostrados correctamente.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 138. Prueba 37: Modificar un menú del sistema.

Caso de prueba de aceptación	
Código: 37	Historia de Usuario #33: Modificar menú del sistema
Nombre: Modificar un menú del sistema	
Responsable: Pedro Vega Martínez	
Descripción: Prueba de funcionalidad para modificar un menú del sistema.	
Condiciones de ejecución: El usuario debe estar registrado previamente en el sistema con su permiso pertinente.	
Entradas/ Pasos de ejecución: El usuario selecciona el menú que desee modificar y procede a ejecutar la acción modificar menú.	
Resultado esperado: El menú es modificado sin generar error. En caso que el usuario no llene algún dato obligatorio, el sistema informará que existen campos obligatorios vacíos. En caso de que ya exista un menú con el mismo nombre, el sistema informará al usuario que ya existe un menú con ese nombre.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 139. Prueba 38: Eliminar un menú del sistema.

Caso de prueba de aceptación	
Código: 38	Historia de Usuario #34: Eliminar menú del sistema
Nombre: Eliminar un menú del sistema	
Responsable: Pedro Vega Martínez	
Descripción: Prueba de funcionalidad para eliminar un menú del sistema.	
Condiciones de ejecución: El usuario debe estar registrado previamente en el sistema con su permiso pertinente.	
Entradas/ Pasos de ejecución: El usuario selecciona el menú que desee eliminar y procede a ejecutar la acción eliminar menú.	
Resultado esperado: El menú es eliminado correctamente.	
Evaluación de la prueba: Prueba satisfactoria.	

Anexo 5: Entrevistas

Entrevistas a los clientes:

Nombre: Dasiel Alberto Pérez Suárez

Cargo: Jefe de Departamento de Tecnología

Nombre: Danny Hernández Hernández

Cargo: Especialista general

Preguntas:

¿Por qué es necesaria la elaboración de un sistema informático para gestionar las reservaciones de los locales docentes en la Facultad 1?

¿Cuáles son las principales funcionalidades que debe tener el sistema?

¿Cómo se garantiza que las características de los locales y de sus medios se mantengan actualizadas?

¿En qué se beneficia el Departamento de Tecnología con la implementación del sistema?

Nombre: Osneidis de la Caridad Mendoza Tamayo

Cargo: Planificadora docente

Preguntas:

¿Cómo se realiza actualmente el proceso de reservación de locales docentes para las actividades que no se planifican en el sistema Horario UCI?

¿En qué se basa la asignación de locales a una actividad específica?

¿Qué limitaciones tiene dicho proceso?

¿Por qué no se reservan todas las actividades utilizando el sistema de escritorio Horario UCI?

¿Cuáles son las principales funcionalidades que debe tener el sistema?

¿Cuáles son los usuarios que podrán tener acceso al sistema una vez que se implemente el mismo?