

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

Facultad #2



**Sistema Integral de Gestión de los Laboratorios.
Módulo de reservación de tiempo de máquina en la
UCI**

Autores: Manuel Matos Delgado

Orley Fumero Paz

Tutor: Ing. Salvador González Gómez

Ciudad de la Habana

Julio 2008

DECLARACIÓN DE AUTORÍA

Declaramos ser autores de la presente tesis 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 2008.

Firma del Autor

Orley Fumero Paz

Firma del Autor

Manuel Matos Delgado

Firma del Tutor.

Ing. Salvador González Gómez

OPINIÓN DEL USUARIO DEL TRABAJO DE DIPLOMA

El Trabajo de Diploma, titulado **Módulo de reservación de tiempo de máquina en la UCI**, fue realizado en la **Universidad de Ciencias Informáticas**. Esta entidad considera que, en correspondencia con los objetivos trazados, el trabajo realizado le satisface

Totalmente

Parcialmente en un _____ %

Los resultados de este Trabajo de Diploma le reportan a esta entidad los beneficios siguientes (cuantificar):

Y para que así conste, se firma la presente a los _____ días del mes de _____ del año 2008.

Representante de la entidad

Cargo

Firma

Cuño

OPINIÓN DEL TUTOR DEL TRABAJO DE DIPLOMA

Título: **Módulo de reservación de tiempo de máquina en la UCI.**

Autor: **Orley Fumero Paz y Manuel Matos Delgado.**

El tutor del presente Trabajo de Diploma considera que durante su ejecución los estudiantes mostraron las cualidades que a continuación se detallan.

- **Independencia**
- **Originalidad**
- **Creatividad**
- **Laboriosidad**
- **Responsabilidad**

Por todo lo anteriormente expresado considero que el estudiante está apto para ejercer como Ingeniero Informático; y propongo que se le otorgue al Trabajo de Diploma la calificación de

_____.

Firma

Fecha

"El futuro tiene muchos nombres: para los débiles es lo inalcanzable,
para los temerosos lo desconocido,
para los valientes es la oportunidad"

Víctor Hugo

AGRADECIMIENTOS

Agradecemos a nuestros padres, desde el sentimiento más profundo que brota de nuestras almas por brindarnos su ayuda incondicional y por servirnos de apoyo y guía para llegar a alcanzar el sueño de graduarnos.

A nuestro Comandante en Jefe, por concebir, entre sus grandes proyectos e ideas, este maravilloso sueño; gracias también por cedernos la oportunidad de formar parte de los profesionales que engrandecen y honran a esta Revolución Socialista Cubana.

Gratificamos a todo educador que de una forma u otra influyó en nuestra vida de estudiante, gracias por su esmero; por su paciencia. Especialmente a Yudeisy Pérez y a Yadira Ruiz.

A nuestros compañeros de estudio, al equipo de trabajo.

Agradecemos a nuestro tutor que nos guió y exigió como un gran educador.

En fin a toda nuestra familia, por sus ánimos y esperanzas y a todos aquellos que contribuyeron a la realización de este anhelo.

Muchas Gracias.

DEDICATORIA

Manuel:

A mis padres Isaida Delgado y José Rolando García, por su guía incondicional, ejemplo y por construir un hogar lleno de amor, confianza y respeto. Gracias a ustedes por ser lo que más quiero en la vida.

A mi primo Yoen Matos, gracias por apoyarme y tenerme presente.

A mis amigos Yoander Cabanes y Dionny Cardoso, por sus consejos y ayuda. Gracias por todo lo que hemos compartido.

A Maylién Horta por soportarme en tantos años, por su apoyo, por su ejemplo, por sus consejos.

Orley:

A mis padres, que con su esfuerzo y dedicación han sabido guiarme por el camino correcto. A mi hermano que su ejemplo ha sido la guía más sólida y certera de mis años de estudiante. Gracias a ustedes que son los que más quiero en la vida.

A mi novia Yaimara que con su amor incondicional supo esperarme en estos largos cinco años de estudio. Gracias mi amor por confiar en mí.

A mi grupo 2103, mis eternos compañeros de aula y a la profesora Yamilka que supo brindarnos su apoyo incondicional en los momentos más difíciles.

RESUMEN

El proceso de reservación de tiempo de máquinas en los laboratorios de la Universidad de las Ciencias Informáticas (UCI) es un proceso de gran importancia para la comunidad universitaria. En la actualidad con la falta de medios y la gran demanda de utilización de computadoras que existe en la universidad se han generado una serie de problemas a la hora de acceder a estas los estudiantes y profesores, por ejemplo: Los usuarios necesitan interrumpir el horario docente en muchas ocasiones con el fin de reservar manualmente un tiempo de máquina; la lista empleada por el personal autorizado para archivar las reservaciones se extravían; de este listado no se realiza un control estricto, por lo que los usuarios pueden realizar doble reservación; una vez realizada la reservación cuando el usuario se dirige a su uso, dicha máquina se encuentra fuera de servicio o no posee el software necesario instalado.

Es por esto que surge la necesidad de diseñar un módulo de gestión de tiempo de máquina y laboratorios que forme parte del Sistema Integral de Gestión de los Laboratorios que le brinde tanto al estudiante como al profesor la posibilidad de reservar, en el caso de los estudiantes una PC y a los profesores un laboratorio para realizar las actividades que requieran de este servicio.

En este documento se encuentra plasmado el desarrollo del software hasta la fase de implementación, se muestran los diagramas correspondientes a todo el flujo de trabajo, así como la justificación de las herramientas utilizadas. Para desarrollar esta propuesta se tuvo en cuenta las tendencias modernas de desarrollo de interfaces con la utilización de la Ext JS como técnica para desarrollar sistemas más interactivos, utilizando como gestor de base de datos PostgreSQL y PHP como lenguaje para interactuar con el servidor. Para la descripción del negocio de este trabajo se tuvo en cuenta las métricas de IDEF (Definición de la integración para la modelización de las funciones) y para el resto del flujo de trabajo se utilizó la metodología de desarrollo RUP con la utilización del Lenguaje Unificado de Modelación (UML).

Como principal resultado se obtuvo la primera versión del módulo de reservación de tiempo de máquina en la UCI.

TABLA DE CONTENIDOS

AGRADECIMIENTOS.....	I
DEDICATORIA	II
RESUMEN.....	III
INTRODUCCIÓN.....	1
1 FUNDAMENTACIÓN TEÓRICA.....	4
Introducción	4
1.1. Sistemas de reservaciones.....	4
1.2. Sistemas de reservaciones en el mundo.....	5
1.2.1. Sistema de reservaciones en Cuba	5
1.2.2. Sistema de reservaciones en la UCI	6
1.3. Tecnologías, lenguajes de programación y librería	6
1.3.1. Lenguaje de programación.....	6
1.3.1.1. PHP	6
1.3.2. Librería Ext JS 2.0.....	7
1.3.2.1. AJAX.....	8
1.3.3. Servidor Web	12
1.3.4. Sistema gestor de base de datos.....	12
1.4. Modelación de funciones mediante IDEF0	13
1.5. Arquitectura	14
1.5.1. Patrón Modelo - Vista – Controlador	14
1.5.2. Arquitectura Cliente/Servidor.	15
1.6. Lenguaje de modelado y metodología de desarrollo	16

1.6.1. Lenguaje de modelado UML	16
1.6.2. Metodología de desarrollo RUP	17
1.7. Herramientas utilizadas	18
1.7.1. Visual Paradigm	18
1.7.2. Aptana.....	18
1.7.3. GIMP.....	19
1.7.4. PGAdmin III.....	19
Conclusiones	20
2 CARACTERÍSTICAS DEL SISTEMA.	21
Introducción	21
2.1. Objeto de estudio	21
2.1.1. Problema y situación problemática.	21
2.1.2. Objeto de automatización.....	22
2.1.3. Información que se maneja	22
2.2. Propuesta del sistema	22
2.3. Especificación de los requisitos del software.....	25
2.3.1. Requerimientos funcionales	25
2.3.2. Requerimientos no funcionales	26
2.4. Modelo de Casos de Uso del sistema	29
2.4.1. Definición de los actores del sistema a automatizar	29
2.4.2. Diagrama de Casos de Uso del sistema a automatizar	29
2.4.3. Descripción de los Casos de Uso	30
Conclusiones	30

3	ANÁLISIS Y DISEÑO DEL SISTEMA.....	31
	Introducción	31
3.1.	Análisis del sistema	31
3.1.1.	Diagramas de clases del análisis	31
3.2.	Diseño del Sistema.....	36
3.2.1.	Diagramas de interacción.....	36
	Diagramas de clases del diseño	37
3.3.	Diseño de la base de datos	47
3.3.1.	Modelo lógico de datos (diagrama de clases persistente).....	47
3.3.2.	Modelo físico de datos (modelo de datos).....	48
3.3.2.1.	Descripción de las Clases del modelo de datos.....	48
	Conclusiones	49
4	IMPLEMENTACIÓN	50
	Introducción	50
4.1.	Diagrama de Despliegue	50
4.2.	Diagrama de Componentes.....	51
	Conclusiones	51
5	ESTUDIO DE FACTIBILIDAD.....	52
	Introducción	52
5.1.	Método de Estimación por Puntos de Casos de Uso.	52
5.2.	Beneficios tangibles e intangibles.....	60
5.3.	Análisis de costos y beneficios.	60
	Conclusiones.	61



CONCLUSIONES	62
RECOMENDACIONES	63
BIBLIOGRAFÍA	64
ANEXOS	68
GLOSARIO DE TÉRMINOS.	95

ÍNDICE DE FIGURAS

FIGURA 1 ARQUITECTURA CLIENTE/SERVIDOR	15
FIGURA 2 DESCRIPCIÓN DEL NEGOCIO, SECUENCIA DE RESERVACIÓN DE TIEMPO DE MÁQUINA	23
FIGURA 3 DESCRIPCIÓN DEL NEGOCIO, SECUENCIA DE REPORTE	25
FIGURA 4 DIAGRAMA DE CASO DE USO DEL SISTEMA	30
FIGURA 5 ANÁLISIS DEL CASO DE USO REALIZAR RESERVACIÓN (ESTUDIANTE).....	31
FIGURA 6 ANÁLISIS DEL CASO DE USO MODIFICAR RESERVACIÓN (ESTUDIANTE).....	32
FIGURA 7 ANÁLISIS DEL CASO DE USO CANCELAR RESERVACIÓN (ESTUDIANTE)	32
FIGURA 8 ANÁLISIS DEL CASO DE USO REALIZAR RESERVACIÓN (PROFESOR).....	33
FIGURA 9 ANÁLISIS DEL CASO DE USO MODIFICAR RESERVACIÓN (PROFESOR).....	33
FIGURA 10 ANÁLISIS DEL CASO DE USO CANCELAR RESERVACIÓN (PROFESOR)	34
FIGURA 11 ANÁLISIS DEL CASO DE USO OBTENER REPORTE RESERVACIÓN (ESTUDIANTE)	34
FIGURA 12 ANÁLISIS DEL CASO DE USO OBTENER REPORTE RESERVACIÓN (PROFESOR)	35
FIGURA 13 ANÁLISIS DEL CASO DE USO OBTENER REPORTE RESERVACIÓN (TÉCNICO).....	35
FIGURA 14 ANÁLISIS DEL CASO DE USO CONTROLAR RESERVACIÓN (TÉCNICO)	36
FIGURA 15 DISEÑO DEL CASO DE USO REALIZAR RESERVACIÓN (ESTUDIANTE).....	37
FIGURA 16 DISEÑO DEL CASO DE USO MODIFICAR RESERVACIÓN (ESTUDIANTE).....	38
FIGURA 17 DISEÑO DEL CASO DE USO CANCELAR RESERVACIÓN (ESTUDIANTE)	39
FIGURA 18 DISEÑO DEL CASO DE USO REALIZAR RESERVACIÓN (PROFESOR).....	40
FIGURA 19 DISEÑO DEL CASO DE USO MODIFICAR RESERVACIÓN (PROFESOR).....	41
FIGURA 20 DISEÑO DEL CASO DE USO CANCELAR RESERVACIÓN (PROFESOR)	42
FIGURA 21 DISEÑO DEL CASO DE USO OBTENER REPORTE DE RESERVACIÓN (ESTUDIANTE)	43
FIGURA 22 DISEÑO DEL CASO DE USO OBTENER REPORTE DE RESERVACIÓN (PROFESOR)	44
FIGURA 23 DISEÑO DEL CASO DE USO OBTENER REPORTE DE RESERVACIÓN (TÉCNICO).....	45
FIGURA 24 DISEÑO DEL CASO DE USO CONTROLAR RESERVACIÓN (TÉCNICO).....	46
FIGURA 25 MODELO LÓGICO DE DATOS	47
FIGURA 26 MODELO FÍSICO DE DATOS.....	48
FIGURA 27 DIAGRAMA DE DESPLIEGUE.....	50
FIGURA 28 DIAGRAMA DE COMPONENTES.....	51
FIGURA 29 DIAGRAMA DE SECUENCIA DEL CASO DE USO REALIZAR RESERVACIÓN (ESTUDIANTE)	85
FIGURA 30 DIAGRAMA DE SECUENCIA DEL CASO DE USO MODIFICAR RESERVACIÓN (ESTUDIANTE)	86
FIGURA 31 DIAGRAMA DE SECUENCIA DEL CASO DE USO CANCELAR RESERVACIÓN (ESTUDIANTE)	87



FIGURA 32 DIAGRAMA DE SECUENCIA DEL CASO DE USO OBTENER REPORTE DE RESERVACIÓN (PROFESOR)88

FIGURA 33 DIAGRAMA DE SECUENCIA DEL CASO DE USO CONTROLAR RESERVACIÓN (TÉCNICO) 89

ÍNDICE DE TABLAS

TABLA 1 DEFINICIÓN DE LOS ACTORES DEL SISTEMA AUTOMATIZAR.....	29
TABLA 2 FACTOR DE PESO DE LOS ACTORES SIN AJUSTAR	53
TABLA 3 FACTOR DE PESO DE LOS CASOS DE USOS SIN AJUSTAR.....	54
TABLA 4 FACTOR DE COMPLEJIDAD TÉCNICA	56
TABLA 5 FACTOR AMBIENTE.....	57
TABLA 6 ESFUERZO TOTAL DEL PROYECTO.....	59
TABLA 7 DESCRIPCIÓN TEXTUAL DEL CASO DE USO DEL SISTEMA GESTIONAR RESERVACIÓN	76
TABLA 8 DESCRIPCIÓN TEXTUAL DEL CASO DE USO DEL SISTEMA OBTENER REPORTE DE RESERVACIÓN.....	80
TABLA 9 DESCRIPCIÓN TEXTUAL DEL CASO DE USO DEL SISTEMA CONTROLAR RESERVACIÓN	84
TABLA 10 DESCRIPCIÓN DE LA TABLA RESERVACIÓN DE LA BASE DE DATOS.....	90
TABLA 11 DESCRIPCIÓN DE LA TABLA USUARIO DE LA BASE DE DATOS	91
TABLA 12 DESCRIPCIÓN DE LA TABLA ROL DE LA BASE DE DATOS	91
TABLA 13 DESCRIPCIÓN DE LA TABLA TURNO DE LA BASE DE DATOS	92
TABLA 14 DESCRIPCIÓN DE LA TABLA FACULTAD DE LA BASE DE DATOS	92
TABLA 15 DESCRIPCIÓN DE LA TABLA TURNO-FACULTAD DE LA BASE DE DATOS	93
TABLA 16 DESCRIPCIÓN DE LA TABLA AREA DE LA BASE DE DATOS.....	93
TABLA 17 DESCRIPCIÓN DE LA TABLA PUESTO DE LA BASE DE DATOS.....	94
TABLA 18 DESCRIPCIÓN DE LA TABLA LABORATORIO DE LA BASE DE DATOS.....	94

INTRODUCCIÓN

En la Universidad de las Ciencias Informáticas diariamente centenares de estudiantes y profesores (usuarios) se dirigen a los laboratorios en horario nocturno para tener acceso a una PC, con el propósito de utilizarla en diferentes tareas que la vida en la universidad les impone. Con la falta de medios y la gran demanda de utilización de computadoras que existe en la universidad se han generado una serie de problemas a la hora de acceder a estas los estudiantes y profesores, por ejemplo: Los usuarios necesitan interrumpir el horario docente en muchas ocasiones con el fin de reservar manualmente un tiempo de máquina; la lista empleada por el personal autorizado para archivar las reservaciones se extravían; de este listado no se realiza un control estricto, por lo que los usuarios pueden realizar doble reservación; una vez realizada la reservación cuando el usuario se dirige a su uso, dicha máquina se encuentra fuera de servicio o no posee el software necesario instalado.

Con el objetivo de darle solución a este problema en la universidad se desarrollaron tres aplicaciones en años anteriores, dos de ellas con características similares, estaban orientadas al cliente, no se pudieron poner a prueba por problemas organizativos, y la otra no estaba orientada a los usuarios, o sea que los que intentaban reservar tenían que dirigirse al encargado de las reservaciones en el docente de su facultad y era casi igual de complicado que en el caso de realizarse manualmente aunque con algunas mejoras en cuanto a la recopilación y almacenamiento de la información. En el curso escolar anterior 2006-2007, en la facultad se inició la elaboración de un sistema que fuera capaz de resolver dicha situación, pero solo se desarrolló hasta lograr un prototipo no funcional.

Por estas razones surge la interrogante ¿Cómo disminuir las dificultades que existen en la reservación de tiempo de máquina y laboratorios en la UCI? Con el fin de resolver esta situación, se hace necesario desarrollar un módulo que permita la mejora de este proceso, garantizando la seguridad de la información y eliminando el engorroso trabajo manual necesario hasta el momento, pues no existe otra forma de hacerlo y es imprescindible llevar el control de este importante proceso. Así, surge el Módulo de Reservación de Tiempo de Máquina.

El Módulo de Reservación de Tiempo de Máquina formará parte de toda una infraestructura organizativa para la gestión interna de la universidad, uniéndose a otros sistemas existentes vitales

para la institución como el del Comedor, Reservación de Pase (para los estudiantes). Además de ser un módulo importante del Sistema Integral de Gestión de los Laboratorios.

Por tanto el objeto de estudio de este trabajo está relacionado con los procesos que se llevan a cabo en los laboratorios de la UCI, en lo referido a la organización y control en los mismos.

El campo de acción queda enmarcado específicamente en los procesos de gestión de tiempo de máquina y laboratorios de la UCI, referente a las entradas y salidas de los estudiantes y profesores a los laboratorios.

La idea que se persigue como objetivo general es desarrollar un módulo que forma parte del Sistema Integral de Gestión de los Laboratorios que garantice la reservación de tiempo de máquina en la UCI y brinde distintos servicios de reportes e información a usuarios, permitiendo la automatización de dicho proceso.

Específicamente en este curso el objetivo es darle continuidad al desarrollo del prototipo no funcional obtenido en el curso 2006-2007 con el propósito de informatizar los procesos que hoy se hacen de forma manual. Proporcionarles facilidades a los usuarios del sistema a través de una interfaz sencilla que se pueda acceder desde cualquier computadora de la universidad. La obtención del producto final tendrá como beneficio principal un incremento del control del personal que accede a los laboratorios, así como de los medios disponibles, contribuyendo a un mejor estado técnico de las computadoras y una reducción considerable de algunos incumplimientos por parte de los usuarios en los laboratorios.

Para cumplir el objetivo trazado, se desarrollarán las siguientes tareas:

- Realizar un estudio del estado del arte de los sistemas de reservaciones en Cuba, la UCI y el mundo.
- Realizar un estudio sobre las tendencias del desarrollo de interfaces de usuarios interactivas propuestas por el cliente.
- Revisar los diagramas hasta el flujo de diseño y realizar los correspondientes a la implementación.
- Realizar un estudio de las herramientas de desarrollo propuestas por el cliente con el propósito de emplearlas en la implementación del módulo.

Con el propósito de organizar y darle una estructura al trabajo se ha decidido dividir el trabajo en 5 capítulos.

Capítulo 1: Describe cómo se realiza el proceso de reservación actualmente en la universidad. Aparece reflejado el estado del arte del tema, tendencias, técnicas, metodologías, así como una descripción detallada de las herramientas a utilizar.

Capítulo 2: Recoge un estudio más detallado del problema de la investigación. Se desarrolla la descripción y análisis del negocio, así como el levantamiento de requisitos para la elaboración del sistema.

Capítulo 3: Esclarece los requerimientos funcionales del sistema y define la arquitectura del mismo, para ello recoge los diagramas de análisis y diseño del flujo de trabajo así como el modelo lógico y físico de datos.

Capítulo 4: Describe cómo los elementos del modelo del diseño se implementan en términos de componentes y cómo estos se organizan de acuerdo a los nodos específicos en el modelo de despliegue.

Capítulo 5: Se realiza un estudio del costo del producto a desarrollar, así como los beneficios que este reportará. Se determina la factibilidad de la elaboración del producto.

1 FUNDAMENTACIÓN TEÓRICA

Introducción

El creciente desarrollo de la informática ha potenciado el almacenamiento de grandes volúmenes de datos y ha permitido el desarrollo de sistemas para la gestión y procesamiento de los mismos. Es así como surgen los sistemas de información, los cuales realizan cuatro actividades básicas: entrada, almacenamiento, procesamiento y salida de información, con el fin de facilitar la información para su consumo en cualquier medio, momento o lugar según el tema de interés.

En el presente capítulo se describen fundamentos teóricos de los procesos a automatizar de acuerdo a las tendencias modernas. Además, una breve descripción de la tecnología usada para el desarrollo del sistema: lenguaje de programación, sistema gestor de base de datos, así como herramientas de desarrollo.

1.1. Sistemas de reservaciones

Gracias al desarrollo de las nuevas Tecnologías de la Información y Comunicación (TICs), el tiempo y la distancia dejan de ser obstáculos en el desarrollo de ciertas actividades de la vida. Las redes mundiales de información, como Internet, no conocen fronteras y ello supone la expansión y diversificación de los mercados mundiales. En la actualidad no se puede pensar en casi ninguna actividad en la cual no intervengan de alguna manera los procesos de cómputo. Las computadoras han invadido la mayoría de las labores del ser humano

Por las razones antes mencionadas, por las exigencias del mercado y la creciente demanda y utilización de de estas tecnologías por clientes más exigentes, es que las empresas han tenido que comenzar a usar Internet como un nuevo canal para efectuar sus negocios y brindar sus servicios. Muestra de esto es que muchas empresas utilizan sistemas de reservación para brindar sus servicios, por ejemplo: Sistema de reservación de cruceros, hoteles, restaurantes, pasajes de avión, entre otros.

Todos estos sistemas tienen un factor común: son aplicaciones de interfaces amigables y sencillas diseñadas de tal forma que el usuario pueda interactuar con ellas fácilmente, pues permiten realizar las reservaciones de los recursos en tiempo real y le dan al cliente o usuario la confiabilidad, seguridad y conformidad que necesitan.

1.2. Sistemas de reservaciones en el mundo.

GetThere

El líder en sistemas de reservaciones basados en Internet para empresas y proveedores de viajes, incluye la compra de boletos de avión, hotel, autos y otros servicios. Entre sus clientes cuenta más de 800 empresas líderes en el mundo. Muchas de las compañías que utilizan GetThere ahorran más de 20 por ciento en costos de viajes, en muchos de los casos esto representa millones de dólares. Se encuentra disponible en Norteamérica, Europa, Asia Pacífico y Latinoamérica y se ofrece en ocho idiomas. También es el motor de reservaciones de viajes por Internet para aerolíneas líderes.

Amadeus.

Líder mundial en soluciones tecnológicas y de distribución para el sector de los viajes y el turismo. El número de establecimientos hoteleros en los que se puede reservar a través del sistema Amadeus crece cerca de un 15% en seis meses y alcanza los 70,000 establecimientos. Amadeus distribuye los productos de 70.000 hoteles a 75.000 puntos de venta de agencia de viajes (tradicionales y por Internet) en 217 países de todo el mundo.

Las reservaciones realizadas en latinoamérica a través del sistema de Amadeus llegaron a los 32 millones en el 2007, un 6.3 por ciento más que el año anterior, además América Latina alcanzó una cifra récord de firmas de nuevas agencias que decidieron elegir Amadeus como socio tecnológico.¹

1.2.1. Sistema de reservaciones en Cuba

En Cuba los sistemas de reservaciones son utilizados principalmente en la rama del turismo, estos aparecen en el país en la década de los 70s, cuando una CID 201-B y varios teletipos, asimilaban heroicamente las tareas de reservación de capacidades hoteleras en una oficina del hotel Habana Libre.

¹ **Michelle, Javier.** Agencia de Noticias ORBITA desde el Perú. *Agencia de Noticias ORBITA desde el Perú.* [Online] [Cited: Junio 2, 2008.] http://agenciaorbita.com/index2.php?option=com_content&do_pdf=1&id=7643.

Una de las primeras aplicaciones desarrolladas en Cuba con el objetivo de permitir reservaciones online fue RODAS, esta fue elaborada por GET-SOFTUR, este sistema automatiza las principales funciones de una central hotelera de reservas, fue elaborada sobre plataformas WINDOWS y UNIX. Posteriormente aparecen en Cuba los sistemas CADOTEL y HOTSTAFF elaborados a finales de los 80s y principios de los 90s.²

La mayoría de los hoteles y establecimientos cubanos del sector turístico cuentan con sitios Web publicados en Internet y cuentan con el servicio de reservaciones online.

1.2.2. Sistema de reservaciones en la UCI

En la universidad se cuenta con el sistema de reservación de pase, el cual es de gran uso e importancia en el manejo de todo un flujo de información, contribuyendo a una mejor organización en el desarrollo y control de este proceso, garantizando una gran comodidad a los usuarios finales. Este proceso le proporciona al país un gran ahorro de combustible, ya que solo se mueven los ómnibus necesarios para transportar a una cantidad determinada de estudiantes.

1.3. Tecnologías, lenguajes de programación y librería

En la actualidad, gracias al avance de Internet, han surgido diversidad de tecnologías y lenguajes para la creación de aplicaciones web, las cuales ofrecen grandes facilidades para establecer comunicación con el usuario mediante las páginas web, por su capacidad para ser visualizadas desde cualquier parte del mundo haciendo uso de un navegador.

1.3.1. Lenguaje de programación

1.3.1.1. PHP

PHP es un lenguaje sencillo, de sintaxis cómoda y dispone de muchas librerías que facilitan en gran medida el desarrollo de las aplicaciones; convirtiéndolo en el favorito de millones de programadores en todo el mundo.

² Encinosa, Lázaro Blanco. Apuntes para una historia de la informática en Cuba. [Online] [Cited: Enero 10, 2008.] www.sld.cu/galerias/doc/sitios/infodir/apuntes_para_una_historia_de_la_informatica_en_cuba.doc

Características de PHP:

- Dispone de una conexión propia a varios sistemas de base de datos como: MySQL, PostgreSQL y Oracle.
- Incorpora bibliotecas que contienen funciones integradas para realizar útiles tareas relacionadas con la Web. Puede generar imágenes GIF al instante, establecer conexiones a otros servicios de red, enviar correos electrónicos, trabajar con cookies y generar documentos PDF, todo con unas pocas líneas de código.
- Es un producto de código abierto, soportado por una gran comunidad de desarrolladores que se encargan de encontrar y reparar los fallos de funcionamiento.
- Es un lenguaje multiplataforma.
- Permite las técnicas de Programación Orientada a Objetos.
- No requiere definición de tipos de variables.
- Posee tratamiento de errores.

PHP 5.2 es una versión de PHP que además incluye:

- Soporte sólido para Programación Orientada a Objetos (OOP) con PHP Data Objects.
- Mejoras de rendimiento.
- Mejor soporte a XML.

1.3.2. Librería Ext JS 2.0

Ext JS es una librería de componentes que facilita las herramientas necesarias para la creación de aplicaciones Web con excelentes gráficos; ya que posee una considerable colección de elementos para el diseño de interfaces, ventanas, pestañas, menús, tablas, etc.

Brinda soporte para:

- Construir interfaces gráficas complejas y dinámicas.
- Comunicar datos de forma asíncrona con el servidor.
- Diversos navegadores como: Internet Explorer, Firefox, Safari y Opera.

Actualmente Ext JS es considerado un Framework independiente; ya que a principios del 2007 se creó una compañía para comercializar y dar soporte al mismo, dicha compañía proporciona los servicios de consultoría necesarios para ayudar a los clientes en el aprovechamiento máximo de las ventajas de Ext JS. Es importante señalar que la Ext JS 2.0 tiene dos tipos de licencias, LGPL (Open Source) y la comercial, esta última es obligatoria si se desea obtener soporte.

1.3.2.1. AJAX

AJAX, acrónimo de Asynchronous JavaScript And XML (JavaScript asíncrono y XML) facilita la creación de aplicaciones interactivas en la Web que se ejecutan en el navegador de los usuarios y mantienen comunicación asíncrona con el servidor. De esta forma, es posible realizar cambios sobre una página sin necesidad de recargarla, aumentando la interactividad, velocidad y usabilidad de la misma.

“AJAX no es una tecnología en sí mismo. En realidad, se trata de la unión de varias tecnologías que se desarrollan de forma autónoma y que se unen de formas nuevas y sorprendentes.”³

AJAX está conformado por:

- XHTML y CSS, para crear una presentación basada en estándares.
- DOM, para la interacción y manipulación dinámica de la presentación.
- XML, XSLT y JSON, para el intercambio y la manipulación de información.
- XMLHttpRequest, para el intercambio asíncrono de información.
- JavaScript, para unir todas las demás tecnologías.

Además:

- Provee un mecanismo para mezclar y hacer coincidir XML con XHTML.
- Las aplicaciones son más rápidas e interactivas, al estilo de aplicaciones de escritorio.
- Reduce de manera significativa tener que cargar información continuamente del servidor, actualizando solamente porciones de la página.

³ Eguíluz Pérez, Javier. 2007. Libros Web. *Introducción a AJAX*. [Online] 2007. [Cited: Enero 8, 2008.] <http://www.librosweb.es/ajax>.

- Cuando se utiliza AJAX adecuadamente en el desarrollo de una aplicación, se reduce de manera significativa los tiempos de carga inicial.

A continuación se explican las características más importantes de algunas de las tecnologías que componen AJAX:

JSON

Notación de Objetos de JavaScript (JavaScript Object Notation) constituye un formato ligero para el intercambio de datos abierto y basado en texto. JSON está pensado principalmente para usarse en aplicaciones Web para transmitir información estructurada de forma asíncrona entre el servidor y los clientes.

JSON es un subconjunto de la notación literal de objetos de Javascript y es usado generalmente con ese lenguaje. Sin embargo, los tipos básicos y las estructuras de datos de la mayoría de los lenguajes de programación también pueden ser representados en JSON y el formato puede por tanto ser usado para intercambiar datos estructurados entre programas escritos en diferentes lenguajes. Existe código para analizar y generar JSON en una gran cantidad de lenguajes como: ActionScript, C, C#, Java, JavaScript, Perl, PHP, Python, Ruby y Smalltalk.

Características de JSON:

- La sintaxis es muy concisa por lo que requiere menos codificación y procesamiento.
- No se necesita código de aplicación adicional para analizar texto.
- Es legible e independiente de la plataforma.
- Soporta diversos lenguajes de programación.

Este formato de intercambio de datos es relativamente nuevo, por lo que no cuenta con el soporte de proveedores de XML; sin embargo es ante todo un formato de datos que no está limitado a las aplicaciones Web y prácticamente se puede usar en cualquier escenario en que los sistemas necesiten intercambiar o almacenar información estructurada como texto.

XML

Es el estándar de Extensible Markup Language (Lenguaje de Etiquetado Extensible), conformado por un conjunto de reglas para definir etiquetas semánticas orientadas a organizar un documento en diferentes partes. Permite al usuario definir sus propios lenguajes de anotación adaptados a sus necesidades y contiene tres características muy importantes que son: extensibilidad, estructura y

validación. Además, ofrece un formato para la descripción de datos estructurados, facilitando declaraciones de contenido más precisas y resultados de búsquedas más significativos en varias plataformas.

Ventajas de XML:

- Las aplicaciones se pueden generar rápidamente y su mantenimiento es más sencillo.
- Separa los datos de la presentación y del proceso, lo que permite mostrar y procesar los datos al gusto deseado con sólo aplicar distintas hojas de estilo y aplicaciones.
- La información es más accesible y reutilizable, por la flexibilidad de las etiquetas de XML que permiten su utilización sin tener que amoldarse a reglas específicas de un fabricante.

XHTML

“XHTML es el acrónimo en inglés de Extensible Hypertext Markup Language (lenguaje extensible de marcado de hipertexto). Es una versión más estricta y limpia de HTML, que nace precisamente con el objetivo de reemplazar a HTML ante su limitación de uso con las cada vez más abundantes herramientas basadas en XML. XHTML extiende HTML 4.0 combinando la sintaxis de HTML, diseñado para mostrar datos, con la de XML, diseñado para describir los datos.”⁴

XHTML reúne la capacidad de formato de HTML y se consolida con la formalidad del XML (y sus reglas) a la hora de estructurar documentos para la portación de datos. Está encaminado al uso de un etiquetado correcto, por lo que exige una serie de requisitos básicos a cumplir en cuanto al código.

Algunos de estos requisitos son:

- Elementos correctamente anidados.
- Etiquetas en minúsculas.
- Elementos cerrados correctamente.
- Atributos de valores entrecomillados.

⁴ **Eguíluz Pérez, Javier.2007.** Libros Web. *Introducción a XHTML*. [Online] 2007. [Cited: Enero 15, 2008.] <http://www.librosweb.es/xhtml>.

CSS

Es un lenguaje de hojas de estilos (Cascading Style Sheets) creado para controlar la presentación de documentos estructurados y escritos en XHTML, aspectos como: el color, el tamaño, el tipo de letra, la separación entre párrafos y la tabulación con la que se muestran los elementos de una lista. El propósito del desarrollo de CSS es separar la estructura y el contenido de la presentación estética en un documento, esto permite un control mayor del documento y sus atributos, convirtiendo al XHTML en un documento muy versátil y liviano.

“Separar la definición de los contenidos y la definición de su aspecto presenta numerosas ventajas, ya que obliga a crear documentos HTML/XHTML bien definidos y con significado completo (también llamados "documentos semánticos"). Además, mejora la accesibilidad del documento, reduce la complejidad de su mantenimiento y permite visualizar el mismo documento en infinidad de dispositivos diferentes.”⁵

Entre los beneficios concretos de CSS se encuentran⁶:

- Control de la presentación de muchos documentos desde una única hoja de estilo.
- Control más preciso de la presentación.
- Aplicación de diferentes presentaciones a diferentes tipos de medios (pantalla, impresión, etc.);

JavaScript

Es un lenguaje basado en objetos y guiado por eventos, diseñado específicamente para el desarrollo de aplicaciones cliente-servidor dentro del ámbito de Internet. Los programas escritos con este lenguaje se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios, convirtiéndolo en un lenguaje interpretado.

⁵ **Eguíluz Pérez, Javier. 2007.** Libros Web. *Introducción a CSS*. [Online] 2007. [Cited: Enero 20, 2008.] <http://www.librosweb.es/css>.

⁶ **HTML.net.** Tutoriales sobre HTML y CSS - Construye tu propio sitio web. [Online] [Cited: Marzo 15, 2008.] <http://es.html.net/tutorials/css/lesson1.asp>.

Ventajas de JavaScript:

- Los programas escritos en este lenguaje no requieren de mucha memoria ni tiempo adicional de transmisión, por ser pequeños y compactos.
- JavaScript no requiere un tiempo de compilación; ya que los scripts se pueden desarrollar en un período de tiempo relativamente corto.
- Es independiente de la plataforma hardware o sistema operativo, y funciona correctamente siempre y cuando exista un navegador con soporte JavaScript.

1.3.3. Servidor Web

Apache es un servidor Web potente, flexible y disponible para distintas plataformas y entornos. Es altamente configurable, de diseño modular, posibilitando que los administradores de sitios Web puedan elegir los módulos que serán incluidos y ejecutados en el servidor.

Características de Apache:

- Es una tecnología gratuita y de código abierto, lo que proporciona transparencia en todo el proceso de instalación.
- Es prácticamente universal, por su disponibilidad en multitud de sistemas operativos.
- Posee alta configurabilidad en la creación y gestión de logs, de este modo es posible tener un mayor control sobre lo que sucede en el servidor.

Este servidor Web tiene una fácil integración con varios lenguajes de programación como: Java, Perl y especialmente PHP. Dicha relación ha dado lugar al desarrollo de aplicaciones como el APPSERV y XAMPP, los cuales instalan el Apache y el PHP configurados para su uso.

1.3.4. Sistema gestor de base de datos

PostgreSQL es un gestor de bases de datos relacional orientado a objetos, libre y gratuito. Está liberado bajo la licencia BSD, lo que significa que se puede disponer de su código fuente, modificarlo a voluntad y redistribuirlo libremente.

Presenta las siguientes propiedades:

- Atomicidad: Asegura la realización de una operación, por lo que ante un fallo del sistema esta no queda a medias.

- Consistencia: Posibilita la ejecución de aquellas operaciones que no van a romper las reglas y directrices de integridad de la base de datos.
- Aislamiento: Mediante un sistema denominado MVCC (Acceso concurrente multiversión) asegura que una operación no pueda afectar a otras, de esta manera dos transacciones sobre la misma información no genera error.
- Durabilidad: Asegura la permanencia de una operación realizada, y aunque falle el sistema esta no podrá deshacerse.

PostgreSQL presenta las siguientes características:

- Se puede instalar un número ilimitado de veces sin temor de sobrepasar la cantidad de licencias.
- Soporta integridad referencial, la cual es utilizada para garantizar la validez de los datos.
- Posee confiabilidad, seguridad y flexibilidad.

1.4. Modelación de funciones mediante IDEF0

IDEF0 (Integration Definition for Function Modeling) consiste en una serie de normas que definen la metodología para la representación de funciones. Representa de manera estructurada y jerárquica las actividades que conforman un sistema o empresa y los objetos o datos que soportan la interacción de esas actividades.

IDEF0 posee las siguientes características:

- Es capaz de representar gráficamente los principales procesos del negocio de una empresa a cualquier nivel de detalle.
- Permite analizar, documentar y mejorar los procesos del negocio.
- Facilita la comunicación y captura de información.
- Es muy coherente y simple, pero a la vez rigurosa y exacta.
- Puede ser generado fácilmente por una gran variedad de herramientas gráficas en computadores.
- Posee flexibilidad ante cambios.

La notación IDEF puede ser representada de dos formas:

La forma **AS-IS** (tal como es) es la empleada por el negocio para representar los procesos a la manera actual, antes de la automatización.

La forma **TO-BE** (cómo debe ser) es utilizada para transformar el proceso a la manera deseada para los objetivos de la organización.

1.5. Arquitectura

1.5.1. Patrón Modelo - Vista – Controlador

El Modelo-Vista-Controlador (MVC) se creó para Smalltalk a finales de los setenta. A partir de entonces su uso se ha ido extendiendo cada día más para la construcción de sistemas software con interfaz gráfica. MVC es un patrón de arquitectura utilizado en sistemas Web para separar los datos de una aplicación, la interfaz de usuario y la lógica de control en tres componentes distintos, permitiendo flexibilidad y facilidad a la hora de hacer futuros cambios.

El **Modelo** representa las estructuras de datos. Típicamente el modelo de clases contendrá funciones para consultar, insertar y actualizar información de la base de datos.

La **Vista** es la información presentada al usuario. Una vista puede ser una página Web o una parte de una página.

El **Controlador** actúa como intermediario entre el Modelo, la Vista y cualquier otro recurso necesario para generar una página, es el objeto que proporciona significado a las órdenes del usuario, actuando sobre los datos representados por el Modelo.

Este patrón presenta las siguientes ventajas:

- Cuando se realiza un cambio en la base de datos, programación o interfaz de usuario solo se toca uno de los componentes.
- Se puede modificar uno de los componentes sin conocer cómo funcionan los otros.
- Lenguaje de modelado y metodologías de desarrollo

1.5.2. Arquitectura Cliente/Servidor.

La arquitectura cliente-servidor es una nueva tendencia en el desarrollo de redes locales, que tiene como objetivo optimizar el uso tanto del hardware como del software a través de la separación de funciones: el cliente, que maneja la porción de la aplicación y el servidor que administra los procesos de almacenamiento y recuperación de los datos.

Puede presentarse como uno a varios clientes y uno o más servidores, junto con un sistema operativo y una plataforma de comunicación para formar un sistema cooperativo que permita la computación distribuida, el análisis y la presentación de datos. Un único servidor típicamente sirve a una multitud de clientes, ahorrando a cada uno de ellos el problema de tener la información instalada y almacenada localmente.

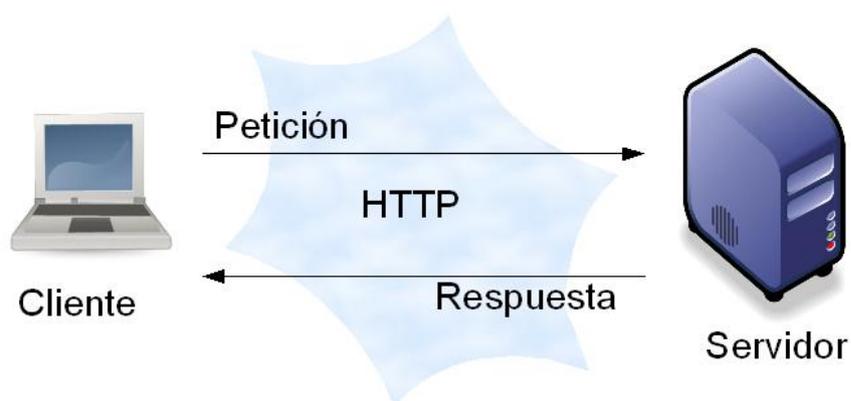


Figura 1 Arquitectura Cliente/Servidor

Características de la arquitectura Cliente/Servidor:

- El servidor presenta una interfaz única y bien definida a todos sus clientes.
- El cliente no necesita conocer la lógica del servidor, sólo su interfaz externa.
- El cliente no depende de la ubicación física del servidor, ni del tipo de equipo físico en el que se encuentra, ni de su sistema operativo.
- Los cambios en el servidor no afectan al cliente.

1.6. Lenguaje de modelado y metodología de desarrollo

1.6.1. Lenguaje de modelado UML

UML es un lenguaje utilizado para el modelado de un sistema, permitiendo en mayor o menor medida representar todas las fases de un proyecto informático: desde el análisis con los casos de uso, hasta la implementación y configuración mediante los diagramas de despliegue.

UML permite:

- Visualizar gráficamente un sistema de manera que otros puedan entenderlo.
- Especificar cuáles son las características de un sistema antes de su construcción.
- Construir sistemas diseñados a partir de modelos especificados.
- Documentar los elementos gráficos del sistema desarrollado para futuras revisiones.

Aporta las siguientes ventajas:

- Permite realizar una verificación y validación del modelo realizado.
- El modelado con UML es independiente del lenguaje de implementación, de tal forma que los diseños realizados se pueden implementar en cualquier lenguaje.
- Permite generar código a partir de los modelos y a la inversa, lo que posibilita la constante actualización.
- Aunque UML está pensado para modelar sistemas complejos con gran cantidad de software, el lenguaje es suficientemente expresivo como para modelar sistemas que no son informáticos, como: flujos de trabajo en una empresa y diseño de la estructura de una organización.

1.6.2. Metodología de desarrollo RUP

“Un proceso de desarrollo de software es el conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema software. Sin embargo, el Proceso Unificado es más que un simple proceso; es un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas software, para diferentes áreas de aplicación, diferentes tipo de organizaciones, diferentes niveles de aptitud y diferentes tamaños de proyecto.”⁷

RUP (Rational Unified Process) es un proceso de desarrollo de software que junto al Lenguaje Unificado de Modelado (UML), constituye una metodología estándar muy utilizada para el análisis, implementación y documentación de sistemas orientados a objetos.

- Se utiliza en proyectos que se desarrollan a largo plazo.
- Permite una mejor comunicación entre ingenieros de software y de negocio, manejando un lenguaje común para facilitar la comprensión de procesos del negocio y su traducción a las funcionalidades que brindará el sistema.
- Genera un volumen considerable de documentación, posibilitando que los cambios realizados en los miembros del equipo no resulte un factor negativo para el avance del proyecto.
- Propone el desarrollo en ciclos e iteraciones con los artefactos que se generan, siendo esto un elemento importante para alcanzar una categoría de certificación en el desarrollo del software.
- Asegura la producción de software de calidad dentro de plazos y presupuestos predecibles.

El ciclo de vida de RUP se caracteriza por:

- Dirigido por casos de uso: Los casos de uso reflejan lo que los usuarios futuros necesitan, lo cual se capta cuando se modela el negocio y se representa a través de los requerimientos. A partir de este momento los casos de uso guían el proceso de desarrollo, pues los modelos que se obtienen en los diferentes flujos de trabajo, representan la realización de los casos de uso.
- Centrado en la arquitectura: La arquitectura muestra la visión común del sistema en la que el equipo del proyecto y los usuarios deben estar de acuerdo. RUP se desarrolla mediante iteraciones, comenzando por los CU relevantes desde el punto de vista de la arquitectura.

⁷ **Rumbaugh, James, Jacobson, Ivar and Booch, Grady.** 2000. El Lenguaje Unificado de Modelado. [Online] 2000. [Cited: Enero 10, 2008.] <http://bibliodoc.uci.cu/pdf/reg00060.pdf>.

- Iterativo e incremental: Propone que cada fase se desarrolle en iteraciones que involucran actividades de todos los flujos de trabajo. Es práctico dividir el trabajo en partes más pequeñas o miniproyectos. Cada miniproyecto es una iteración que resulta un incremento. Las iteraciones hacen referencia a pasos en los flujos de trabajo, y los incrementos, al crecimiento del producto.

1.7. Herramientas utilizadas

1.7.1. Visual Paradigm

Visual Paradigm para UML (VP-UML) es una poderosa herramienta CASE multiplataforma (Windows/Linux/Mac OS X) que soporta el ciclo de vida completo del desarrollo de software. Está diseñado para un amplio rango de usuarios, incluyendo ingenieros de software, analistas de sistema, analistas de negocio, arquitectos de sistema y quienes estén interesados en la construcción de sistemas de software confiables mediante el uso de la Orientación a Objetos.

Este software facilita una rápida construcción de aplicaciones de calidad y a un menor coste. Visual Paradigm para UML soporta un conjunto de lenguajes (Java, C + +, PHP, Ada y Python), tanto en generación de código como ingeniería inversa.

Entre sus características principales se pueden citar:

- Soporta UML en su versión 2.1.
- Facilita la comunicación de todo el equipo de desarrollo mediante el uso de un lenguaje estándar común.
- Posibilita el desarrollo de la ingeniería directa e inversa.
- Durante todo el ciclo de desarrollo el modelo y el código permanecen sincronizados, permitiendo la generación de código a partir de diagramas y viceversa.
- Se encuentra disponible en múltiples versiones y plataformas.
- Permite la generación de bases de datos a partir de la transformación de diagramas de Entidad-Relación en tablas de base de datos y viceversa.

1.7.2. Aptana

Aptana Studio es un IDE de desarrollo para aplicaciones de la Web 2.0, gratuito y de código fuente abierto. Está basado en el conocido entorno de desarrollo Eclipse, también de código abierto; pero

mientras que Eclipse está focalizado en el desarrollo para Java, Aptana Studio es una distribución focalizada en el desarrollo Web, con soporte a HTML, CSS y Javascript.

Soporta varias librerías como: Prototype, Yahoo UI y JQuery, pudiendo combinarlas fácilmente en una aplicación. Aptana Studio está disponible como una aplicación independiente o como plug-in para Eclipse y se puede encontrar para tres plataformas fundamentales: Windows, Mac OS y GNU/Linux.

La gestión de proyectos, vista previa, autocompletado de código y gestión de documentación, son algunas de las características similares que presenta Aptana con otros entornos de desarrollo integrado (Eclipse, C++ Builder, Visual Studio. NET), pero además:

- Visualiza los errores de sintaxis a medida que se escribe.
- Presenta una previa visualización de estilos CSS con el editor CSS.

1.7.3. GIMP

GIMP es el acrónimo para GNU Image Manipulation Program. Es un programa libre distribuido bajo la licencia GPL, apropiado para tareas como: arreglo fotográfico, composición y edición de imagen. Es especialmente útil para la creación de logotipos y otros gráficos para páginas Web. Inicialmente se desarrolló para sistemas Unix; sin embargo, actualmente existen versiones totalmente funcionales para Windows y para Mac OS.

Características fundamentales de GIMP:

- Lee y escribe la mayoría de los formatos de ficheros gráficos, entre ellos jpg, gif, png, pcx y tiff.
- Es capaz de importar ficheros en pdf y también imágenes vectoriales en formato svg.
- Permite la automatización de muchos procesos mediante macros o secuencias de comandos.
- Dispone de varias herramientas o filtros para la manipulación de los colores y el aspecto de las imágenes.

1.7.4. PGAdmin III

Es una aplicación gráfica usada para la gestión de PostgreSQL, siendo la más completa y popular con licencia Open Source. PGAdmin está escrito en C++ y utiliza la librería gráfica multiplataforma wxWidgets, permitiendo que se pueda usar en sistemas operativos como: GNU/Linux, FreeBSD,

Solaris, Mac OS y Windows. Es capaz de gestionar versiones a partir de la PostgreSQL 7.3, ejecutándose en cualquier plataforma.

PGAdmin III está diseñado para responder a las necesidades de todos los usuarios, desde escribir consultas SQL simples, hasta desarrollar bases de datos complejas.

El producto también incluye:

- Editor SQL con resaltado de sintaxis.
- Editor de código de la parte del servidor.
- Amplia documentación para ayudar a los usuarios menos experimentados.

Conclusiones

En este capítulo se dio una visión general del ambiente en que diseñará el módulo. Partiendo del estudio en el mundo de los sistemas de gestión de información que hoy en día son de gran importancia para todos los organismos que manipulan gran flujo de datos. Debido a la gran variedad de tecnologías de desarrollo web y a las tendencias interactivas a las que se mueven, se ha decidido diseñar el sistema utilizando el Ext JS para el desarrollo del diseño de las interfaces, PHP como lenguaje para interactuar con el servidor y como gestor de base de datos PostgreSQL. Además este módulo se diseñará sobre el sistema operativo Debian GNU/Linux, dadas las potencialidades que brinda y la evidente migración en el país hacia este sistema operativo.

2 CARACTERÍSTICAS DEL SISTEMA.

Introducción

En este capítulo se realiza un estudio más detallado al problema de la investigación, con el objetivo de darle una solución. Solución que está dirigida a crear un sistema que automatice el proceso de reservación de tiempo de máquinas en los laboratorios de la UCI. En el capítulo se desarrollará la descripción y análisis del negocio, así como el levantamiento de requisitos para la elaboración del sistema.

2.1. Objeto de estudio

2.1.1. Problema y situación problemática.

En la UCI con la falta de medios y la gran demanda de utilización de computadoras, unido a que el proceso de reservación de tiempo de máquina se está realizando de forma manual se han generado una serie de problemas a la hora de acceder a las computadoras los estudiantes y profesores, por ejemplo: Los usuarios necesitan interrumpir el horario docente en muchas ocasiones con el fin de reservar manualmente un tiempo de máquina; la lista empleada por el personal autorizado para archivar las reservaciones se extravían; de este listado no se realiza un control estricto, por lo que los usuarios pueden realizar doble reservación; una vez realizada la reservación cuando el usuario se dirige a su uso, dicha máquina se encuentra fuera de servicio o no posee el software necesario instalado. Además los profesores a la hora de reservar un laboratorio para recuperar clases, impartir cursos optativos y dar consultas, actividades vitales para el desarrollo docente e investigativo de la universidad no tienen como hacer una reservación de un laboratorio completo formalmente. También aparecen dificultades con la recopilación y almacenamiento de las listas de reservaciones efectuadas por días en los diferentes turnos para poder identificar al posible culpable de cualquier incumplimiento en su tiempo de máquina, esta es otra de las necesidades imperiosas que existen por resolver.

Por lo tanto el problema queda planteado de la siguiente forma. **¿Cómo disminuir las dificultades que existen en la reservación de tiempo de máquina y laboratorios en la UCI?**

2.1.2. Objeto de automatización

La UCI no cuenta con un sistema de reservación de tiempo de máquina que sea capaz de brindar este servicio en línea a toda la comunidad universitaria, dificultando de esta forma la organización y control del personal que utiliza este servicio universitario.

Esta aplicación tiene como objetivo brindarles la posibilidad a estudiantes y profesores de realizar la reservación de tiempo de máquina desde cualquier puesto de trabajo donde se encuentre. Además que le será de gran ayuda a la dirección de los laboratorios como herramienta de control de acceso de personal a los laboratorios docentes.

2.1.3. Información que se maneja

La información necesaria que será manejada por el sistema esta relacionada con los datos de los usuarios. Además de la fecha en que se realiza la reservación, comprendiendo hora y turno teniendo en cuenta los puestos disponibles por cada laboratorio.

2.2. Propuesta del sistema

Se propone diseñar un módulo que gestione las reservaciones de tiempo de máquinas mediante una interfaz amigable, permitiendo a estudiantes y profesores realizar la reservación, además de darle la posibilidad de cancelarla o modificarla. Los usuarios solo podrán solicitar el servicio en los laboratorios de su facultad. Todas estas acciones estarán regidas por un horario, fuera de este no se podrá realizar la solicitud del servicio. El módulo brindará una confirmación al usuario cuando el proceso termine. Los responsables del acceso a los laboratorios tendrán la posibilidad de controlar las reservaciones hechas para un laboratorio determinado, mediante una interfaz que mostrará los datos de los usuarios que solicitaron el servicio para de esta forma controlar el acceso. Estos responsables tendrán la posibilidad de modificar una reservación hecha cuando esta halla pasado de los 15 minutos y el usuario que solicitó el servicio no se presente, permitiendo insertar los datos del nuevo usuario. El sistema guardará historiales de las reservaciones hechas para en caso de alguna indisciplina poder tomar medidas con la persona que incurrió en el problema y de esa manera mejorar la seguridad de los laboratorios.

Descripción del negocio

El Módulo de Reservación de Tiempo de Máquina en los Laboratorios del proyecto SIGLA, corresponde al desarrollo de un conjunto de procesos que se inician con la solicitud de un tiempo de trabajo por parte de estudiantes y profesores en una computadora personal que radica en los laboratorios y culmina con el acceso por los mismos a los puestos de trabajo. En el negocio se describen dos secuencias, la secuencia de reservación y la secuencia de reporte.

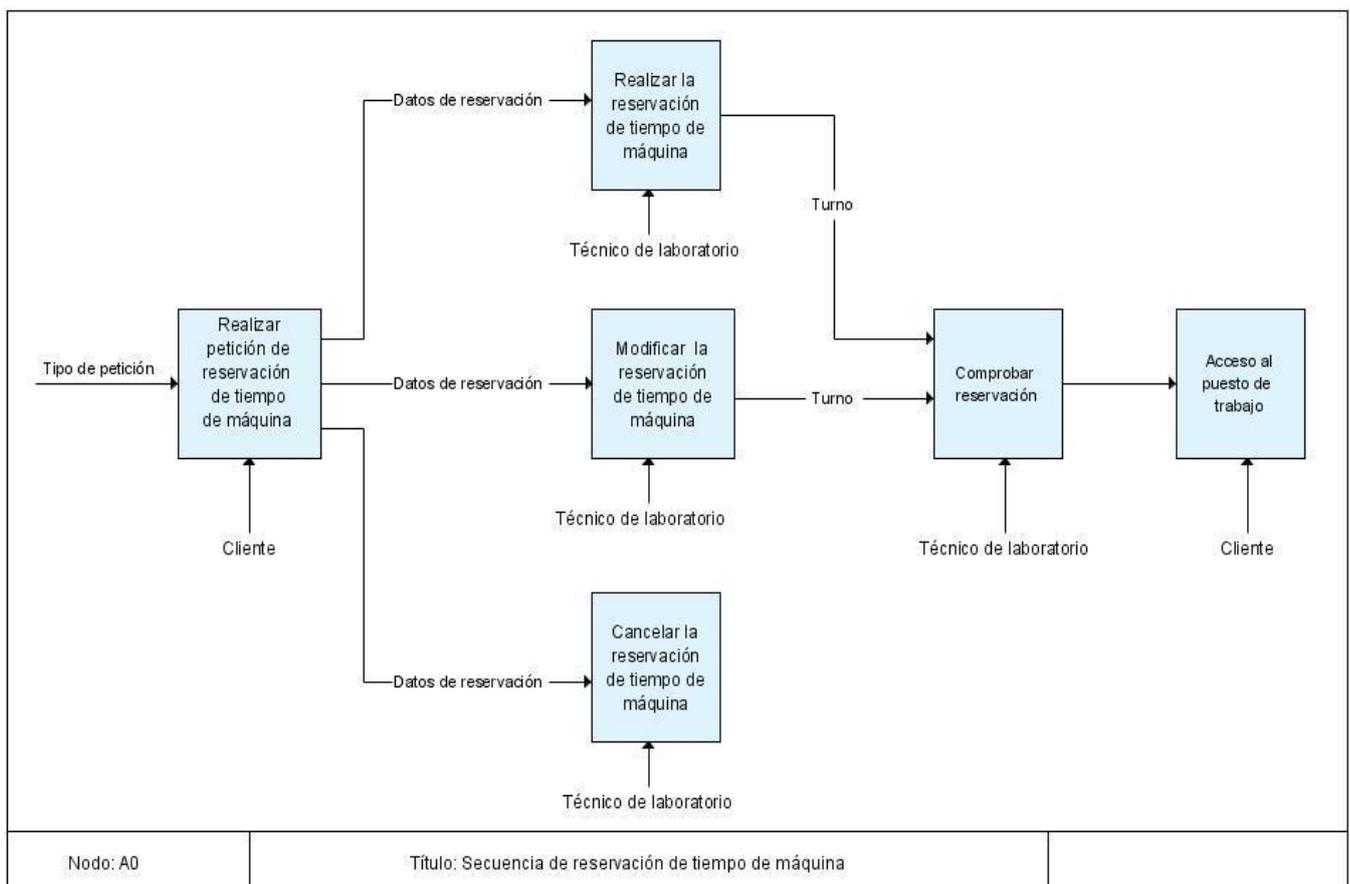


Figura 2 Descripción del negocio, secuencia de reservación de tiempo de máquina

Descripción detallada de los procesos.

Realizar petición de reservación de tiempo de máquina: Este proceso se inicia con la necesidad que tienen los clientes (estudiantes y profesores) de acceder a un puesto de trabajo (PC) para realizar las actividades que la vida universitaria les impone. Es en este proceso donde el cliente le comunica al técnico del laboratorio, responsable de hacer la reservación la opción que desea hacer, entendiéndose por

esto realizar modificar o cancelar una reservación y el técnico de laboratorio a razón de esta opción le hace una solicitud de datos.

Realizar reservación de tiempo de máquina: Este proceso se inicia cuando el cliente haya solicitado la reservación de tiempo de máquina. El técnico registra en una planilla los datos solicitados anteriormente así como el puesto de trabajo reservado en su respectivo horario. Una vez aceptada la reservación de tiempo de máquina, esta no estará disponible para realizar otra reservación en ese tiempo de trabajo a menos que el cliente que hizo la reservación la cancele o se ausente en la comprobación de la misma a la hora de acceder a la PC.

Modificar reservación de tiempo de máquina: Este proceso se inicia solo cuando haya sido realizada la reservación de tiempo de máquina y el cliente quiere hacer uso de otra reservación. El cliente tiene la posibilidad de optar por otro horario disponible de dicha PC reservada o por reservar en otro horario disponible en otra PC. Como poscondición, el tiempo de máquina de la PC que había reservado con anterioridad queda disponible para ser reservada nuevamente por otro usuario.

Cancelar reservación de tiempo de máquina: Este proceso se inicia solo cuando haya sido realizada la reservación de tiempo de máquina y el cliente no quiere hacer uso de dicha reservación. El cliente opta por no usar el tiempo de máquina reservado y este último queda disponible.

Realizar comprobación de reservación: Este proceso se inicia solo cuando haya sido realizada la reservación de tiempo de máquina y el cliente quiere acceder al puesto de trabajo. El técnico verifica los datos del solapín del cliente con los entrados en la realización de la reservación, si coinciden el cliente accede al puesto de trabajo; en caso contrario no puede acceder. Una vez realizadas todas las comprobaciones de los clientes que previamente habían reservado y pasado 15 minutos, si quedan PC disponibles el técnico de laboratorio modifica las reservaciones que no hallan sido comprobadas con los datos de los nuevos clientes.

Acceso a la PC de trabajo: Este proceso se inicia después que el técnico haya realizado las comprobaciones de reservación. Consiste en el consumo del tiempo de máquina disponible por el cliente.

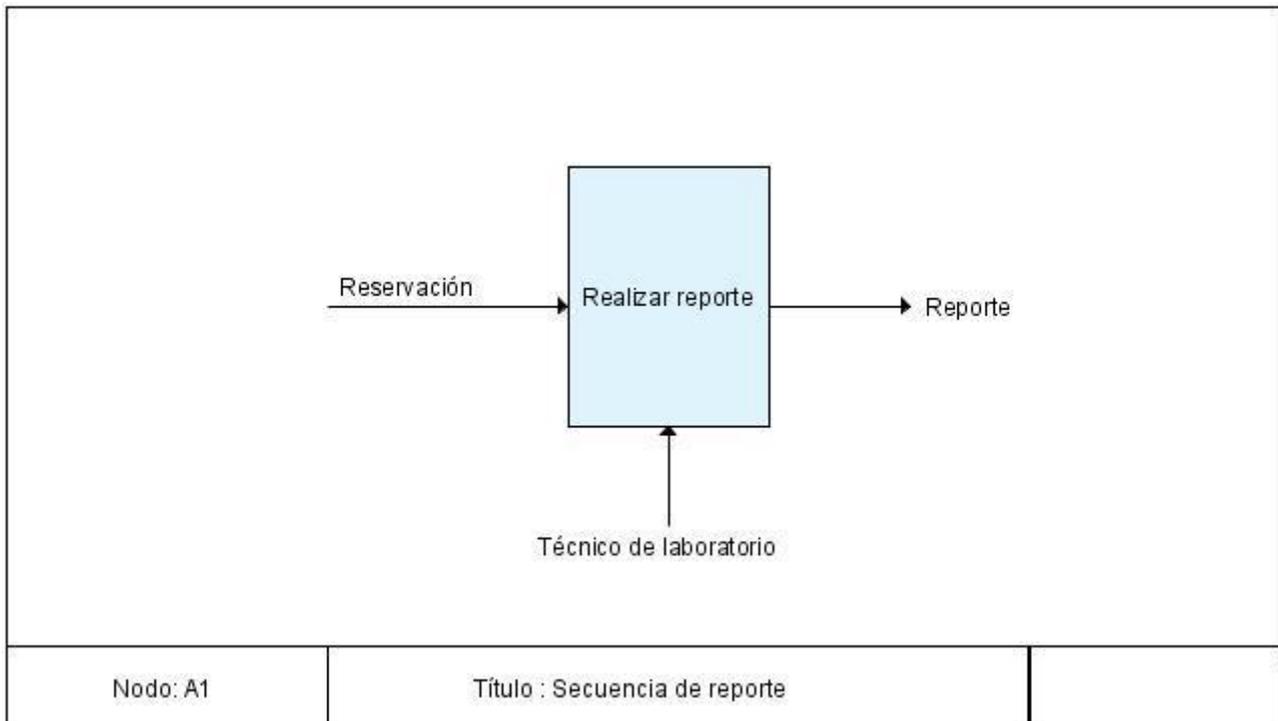


Figura 3 Descripción del negocio, secuencia de reporte

Descripción Detallada de los Procesos.

Realizar reporte: Este proceso se inicia cuando previamente se hallan hecho reservaciones de tiempo de máquina. El técnico de laboratorio puede realizar una consulta sobre la cantidad de reservaciones que se hayan realizado en un período de tiempo determinado correspondiente a un laboratorio.

2.3. Especificación de los requisitos del software

2.3.1. Requerimientos funcionales

R1. Realizar reservación

1.1. Escoger turno

1.1.1. PC

1.1.2. Turno

1.2. Generar mensaje de confirmación.

- 1.3. Proveer datos de reservación.
 - 1.3.1. Turno de reservación
 - 1.3.2. PC reservada
 - 1.3.3 Numero laboratorio reservado

R2. Modificar reservación

- 2.1. Cambiar de turno o de puesto de trabajo
 - 2.1.1. PC reservada
 - 2.1.2. Turno de reservación
- 2.2. Generar mensaje de confirmación

R3. Cancelar reservación

- 3.1. PC reservada
- 3.2. Generar mensaje de confirmación

R4. Obtener reportes de reservación.

- 4.1. Introducir período de tiempo
- 4.2. Mostrar resultado.

R5. Validar reservación

R6. Renovar reservación

- 6.1. Insertar nuevo usuario
- 6.3. Generar mensaje de información.

2.3.2. Requerimientos no funcionales

Requerimientos de Seguridad:

- La información será almacenada en una base de datos, dejando registradas todas las operaciones mediante copias de seguridad.
- El uso y manejo del sistema estará controlado; ya que la información podrá ser consultada y modificada solamente por el personal autorizado; estableciendo para ello una serie de roles con funcionalidades específicas.

- La aplicación estará funcionando a tiempo completo; de esta forma es posible que los usuarios tengan acceso (según sus permisos) en todo momento a la información solicitada.
- Es muy importante que los servidores estén sincronizados correctamente con el día y hora, ya que si ocurre un fallo el sistema no funcionaría correctamente.

Requerimientos de Confiabilidad:

- Todas las salidas del sistema tendrán 100% de veracidad y precisión.
- El sistema tendrá la capacidad de recuperarse rápidamente ante cualquier fallo mediante las copias de seguridad que serán realizadas.
- Se verificará constantemente la conexión con el servidor y en caso de fallo se le notificará al usuario.
- En caso de alguna dificultad con el funcionamiento del sistema, el tiempo medio de reparación deberá ser menor de 1 día.

Requerimientos de Rendimiento:

- El sistema debe tener un tiempo de respuesta rápido y eficiente, inferior a 10 segundos.
- La aplicación debe ser capaz de soportar gran cantidad de usuarios conectados simultáneamente.

Requerimientos de apariencia o interfaz externa:

- La interfaz será agradable, sencilla y sugerente; ya que su diseño simulará una aplicación de escritorio, presentará por defecto un color azul claro y textos que muestran la acción a realizar con un tipo de letra Verdana de tamaño 10pt.
- Se le brindará al usuario la posibilidad de personalizar la apariencia de interfaz externa.
- Debe contener pocas imágenes y gráficos para acelerar la velocidad de respuesta del sistema; por lo que se emplearán íconos sencillos que representarán la operación a realizar.
- El sistema será diseñado para una resolución de pantalla de 1024 X 768 píxeles.

Requerimientos de Hardware:

El servidor Web y de base de datos deben tener como mínimo las siguientes características:

- Computador PENTIUM IV.
- 1 GB de memoria RAM.

- 10 GB de espacio libre en disco duro.

Las computadoras situadas en los puestos de trabajo de los usuarios requieren como mínimo:

- Computador PENTIUM II.
- 128 MB de memoria RAM.

Además es necesario contar con una impresora para poder imprimir los reportes y tarjetas de red de 10/100 MB/s para la conexión.

Requerimientos de software:

- Las estaciones de trabajo clientes utilizarán como sistema operativo GNU/Linux en sus diversas distribuciones o la familia de Windows superior a Windows 98.
- Las máquinas clientes contarán con un navegador, ya sea el Internet Explorer en versiones superiores a la 5.0 o Mozilla Firefox en versiones superiores a la 1.5, ambos con las opciones de JavaScript y cookies habilitadas.
- El servidor Web funcionará sobre el sistema operativo GNU/Linux y tendrá instalado el Apache en su versión 2.0 y PHP 5.2 con las librerías PDO, pgsq y json habilitadas.
- El servidor de base de datos funcionará sobre el sistema operativo GNU/Linux y tendrá instalado el gestor de bases de datos PostgreSQL 8.2.

Requerimientos de Soporte:

- El sistema contará con la documentación apropiada para agilizar su mantenimiento y configuración.
- Se ofrecerán servicios de adiestramiento al personal que trabajará con el software.

Requerimientos de Usabilidad:

- La interfaz será fácil de usar para los diversos usuarios que interactúen con ella.
- El sistema estará bien documentado, con el fin de lograr un mejor uso de los servicios que este ofrecerá, para ello se realizará una ayuda que explique paso a paso cada una de las funcionalidades del software.

Requerimientos de Ayuda y documentación en línea:

- El sistema proporcionará en todo momento la documentación necesaria para que los usuarios puedan acceder a la misma en caso de que necesiten realizar alguna consulta.

Requerimientos legales:

- Los módulos de desarrollo del proyecto SIGLA así como la documentación del mismo, son propios de la UCI y específicamente del área de los laboratorios.

2.4. Modelo de Casos de Uso del sistema

2.4.1. Definición de los actores del sistema a automatizar

Actores	Justificación
Usuario(Profesor, estudiante)	Es la persona que solicita el servicio de reservación
Técnico	Se encarga de controlar las reservaciones en los laboratorios

Tabla 1 Definición de los actores del sistema automatizar

2.4.2. Diagrama de Casos de Uso del sistema a automatizar

Los diagramas de casos de uso se emplean para modelar la vista de casos de uso de un sistema, representa gráficamente a los procesos y su interacción con los actores.

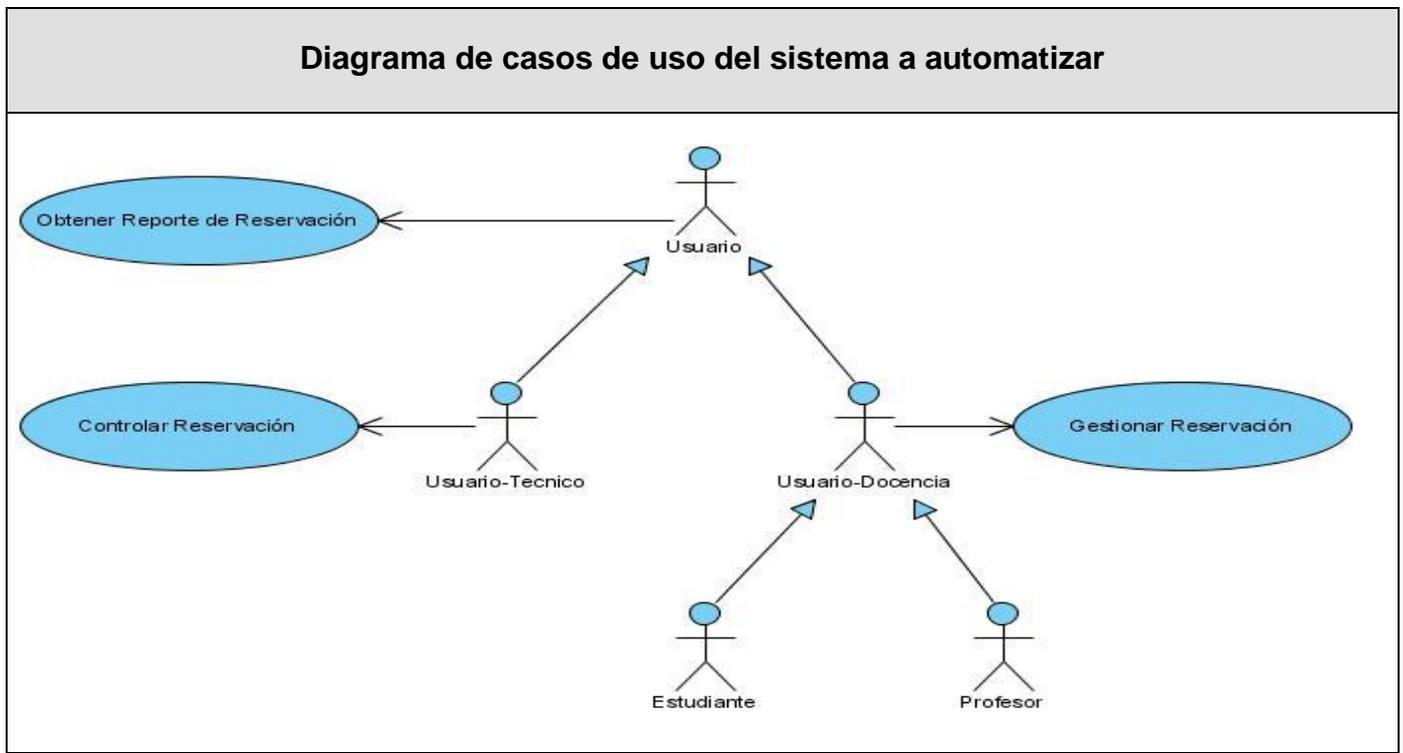


Figura 4 Diagrama de caso de uso del sistema

2.4.3. Descripción de los Casos de Uso

En el **Anexo_1** se muestran las descripciones textuales de los casos de usos descritos en el diagrama con las interfaces correspondientes.

Conclusiones

En este capítulo se trataron las ideas esenciales sobre el sistema a automatizar, se realizó un estudio detallado del negocio y se completó su análisis y modelado. Se realizó una propuesta de sistema que satisfaga las especificaciones de automatización de los procesos del negocio actuales y se completó el levantamiento de los requerimientos esperados.

3 ANÁLISIS Y DISEÑO DEL SISTEMA

Introducción

En este capítulo se abordará sobre el análisis y diseño del sistema, en correspondencia con los requerimientos funcionales. Se confeccionará para mayor comprensión los diagramas de clases de análisis, los diagramas de clases del diseño y los de secuencia del diseño por cada caso de uso del sistema. Se realiza además el modelo lógico y físico de la base de datos.

3.1. Análisis del sistema

El modelo de análisis proporciona una comprensión detallada de los requisitos. Y lo que es más importante, impone una estructura del sistema que debe ser conservada lo más fielmente posible cuando se le de forma al sistema.

3.1.1. Diagramas de clases del análisis

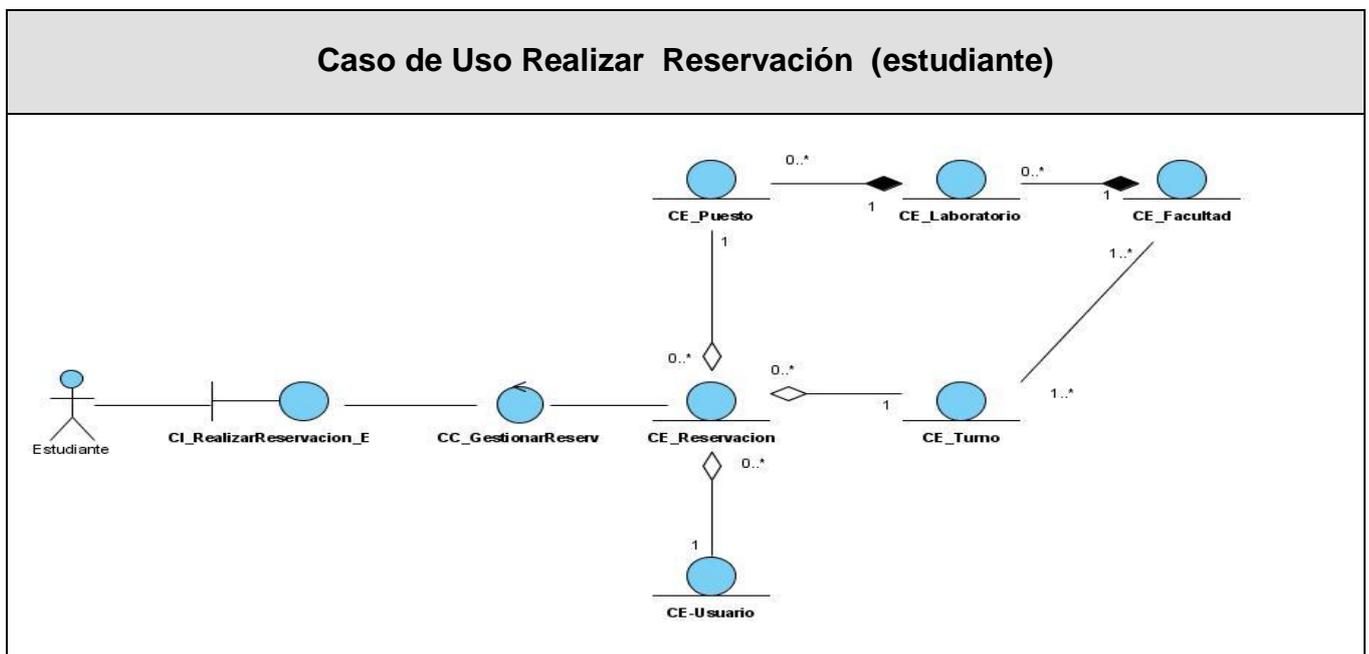


Figura 5 Análisis del caso de uso realizar reservación (estudiante)

Caso de Uso Modificar Reservación (estudiante)

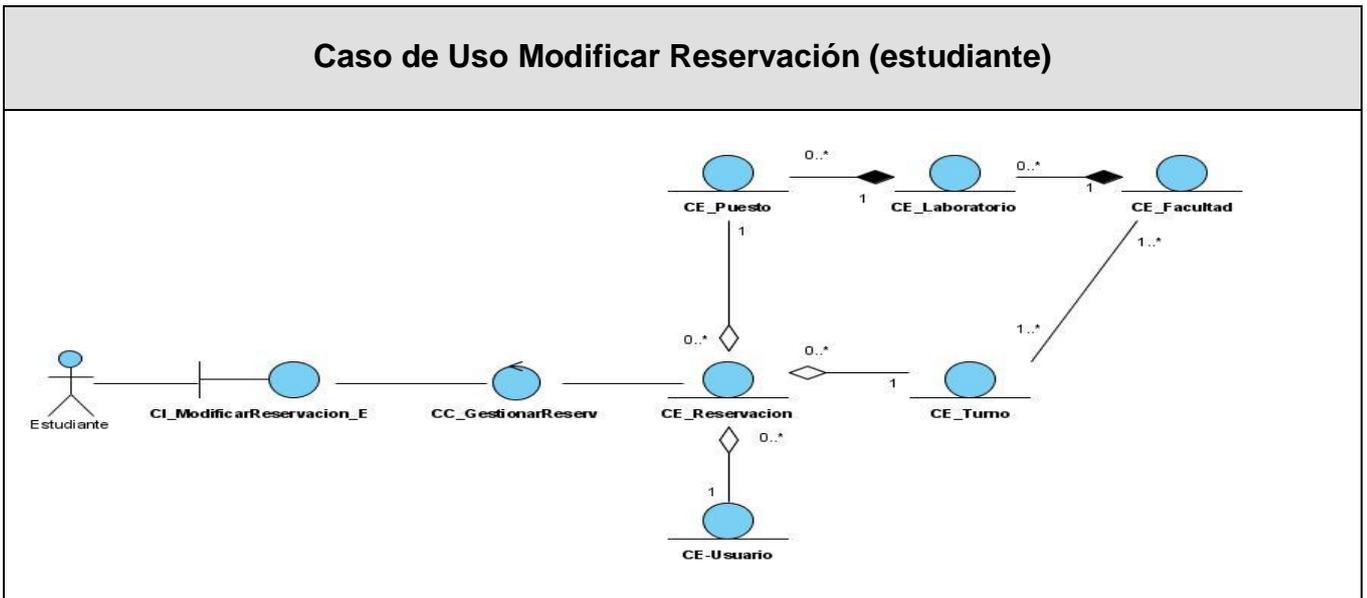


Figura 6 Análisis del caso de uso modificar reservación (estudiante)

Caso de Uso Cancelar Reservación (estudiante)

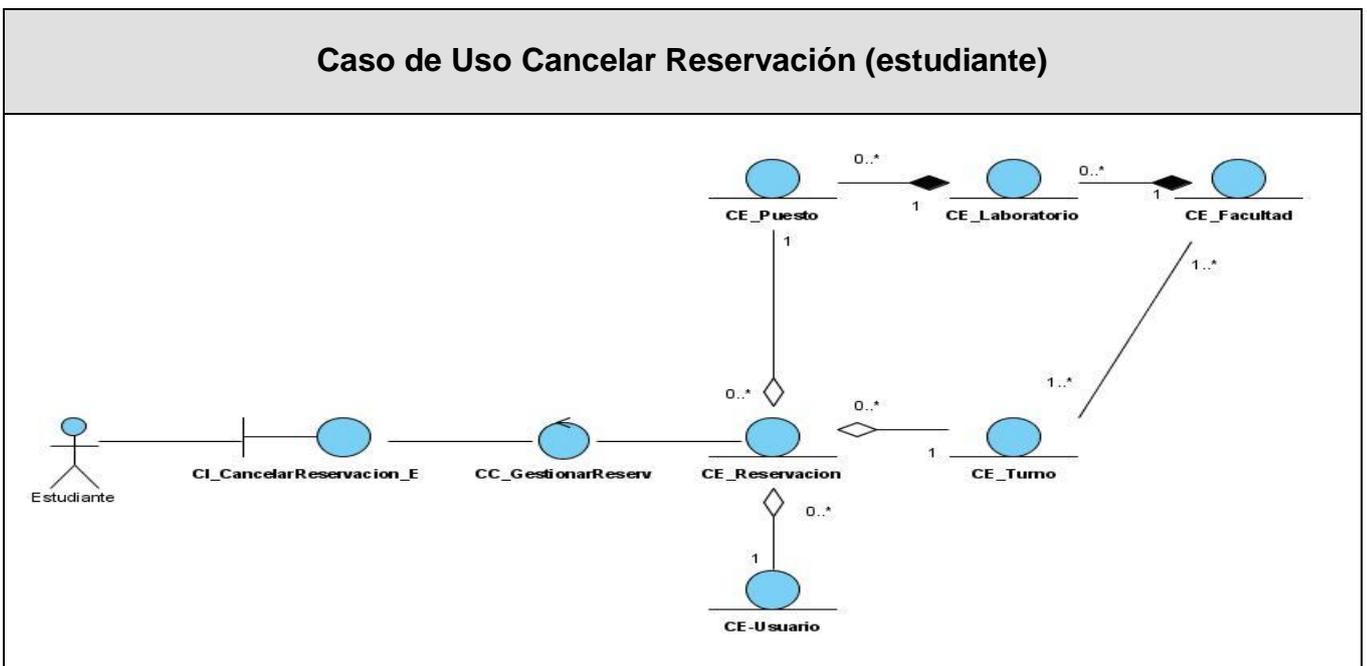


Figura 7 Análisis del caso de uso cancelar reservación (estudiante)

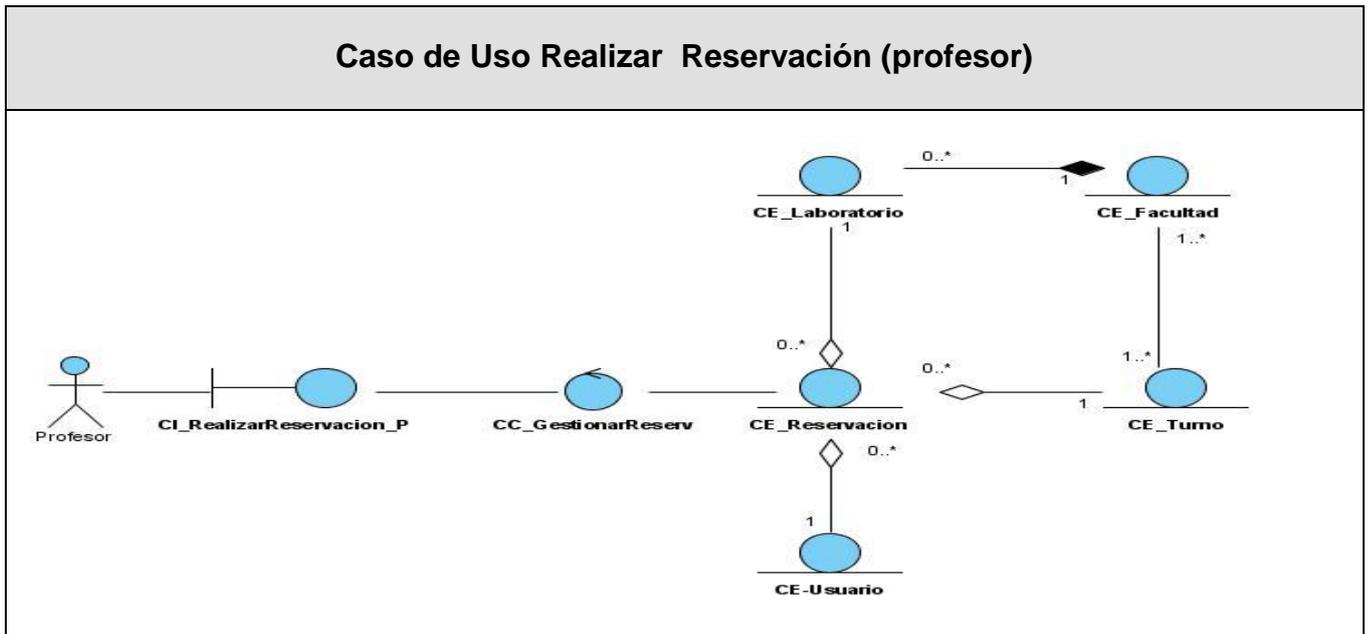


Figura 8 Análisis del caso de uso realizar reservación (profesor)

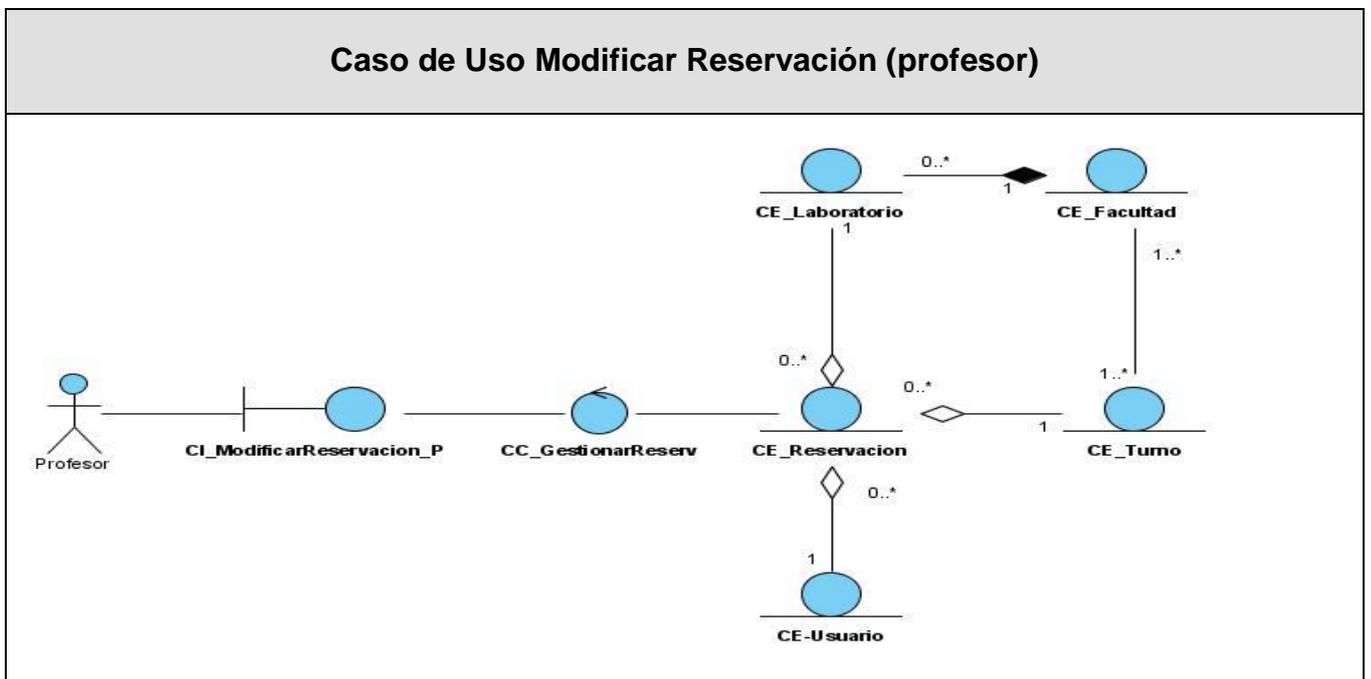


Figura 9 Análisis del caso de uso modificar reservación (profesor)

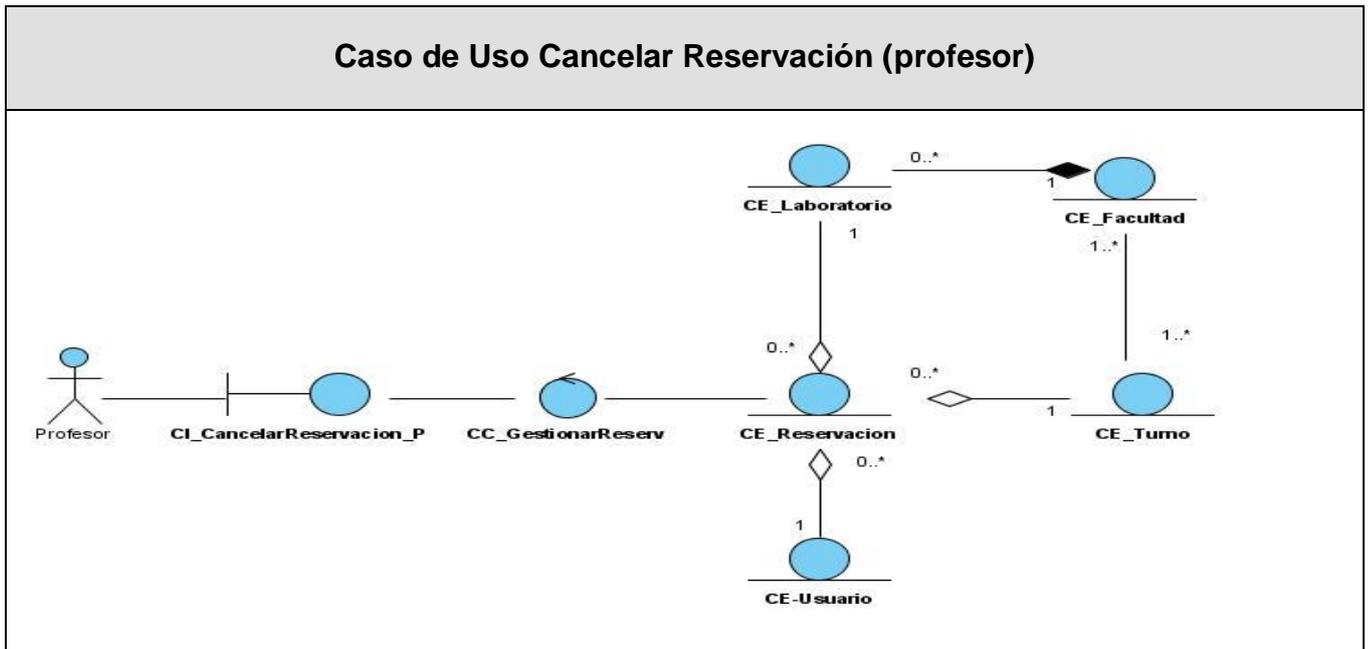


Figura 10 Análisis del caso de uso cancelar reservación (profesor)

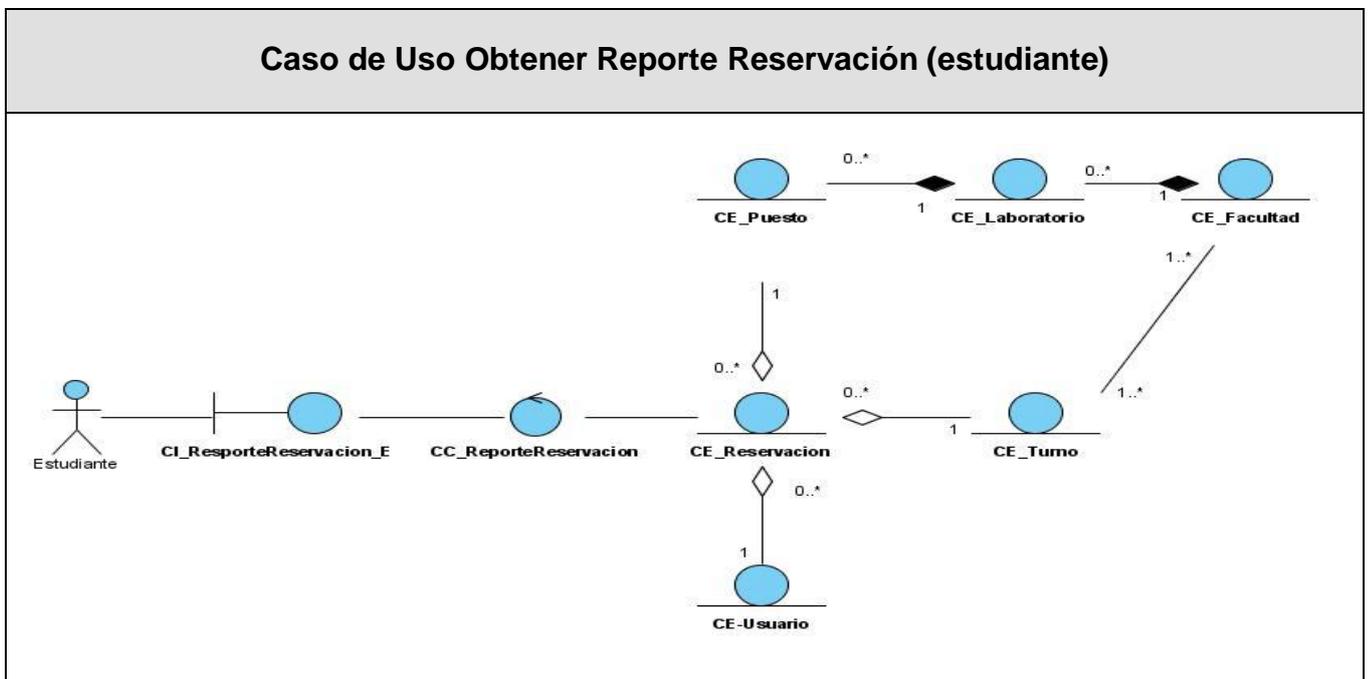


Figura 11 Análisis del caso de uso obtener reporte reservación (estudiante)

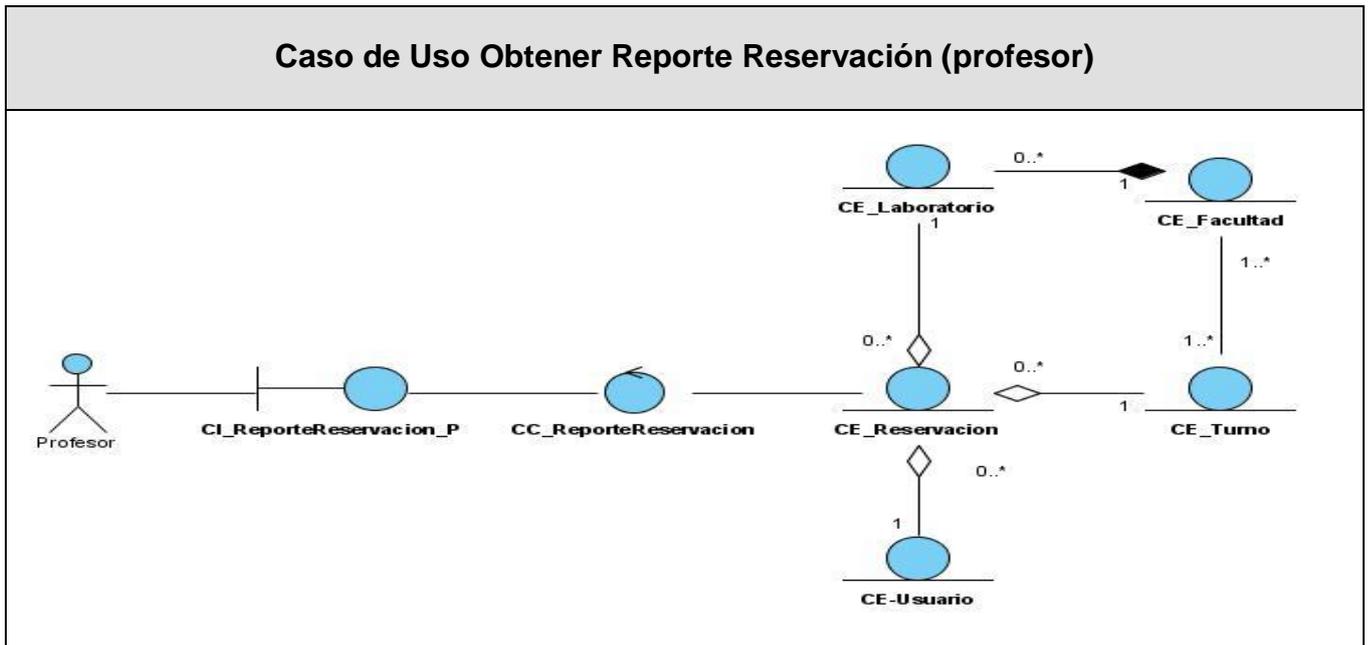


Figura 12 Análisis del caso de uso obtener reporte reservación (profesor)

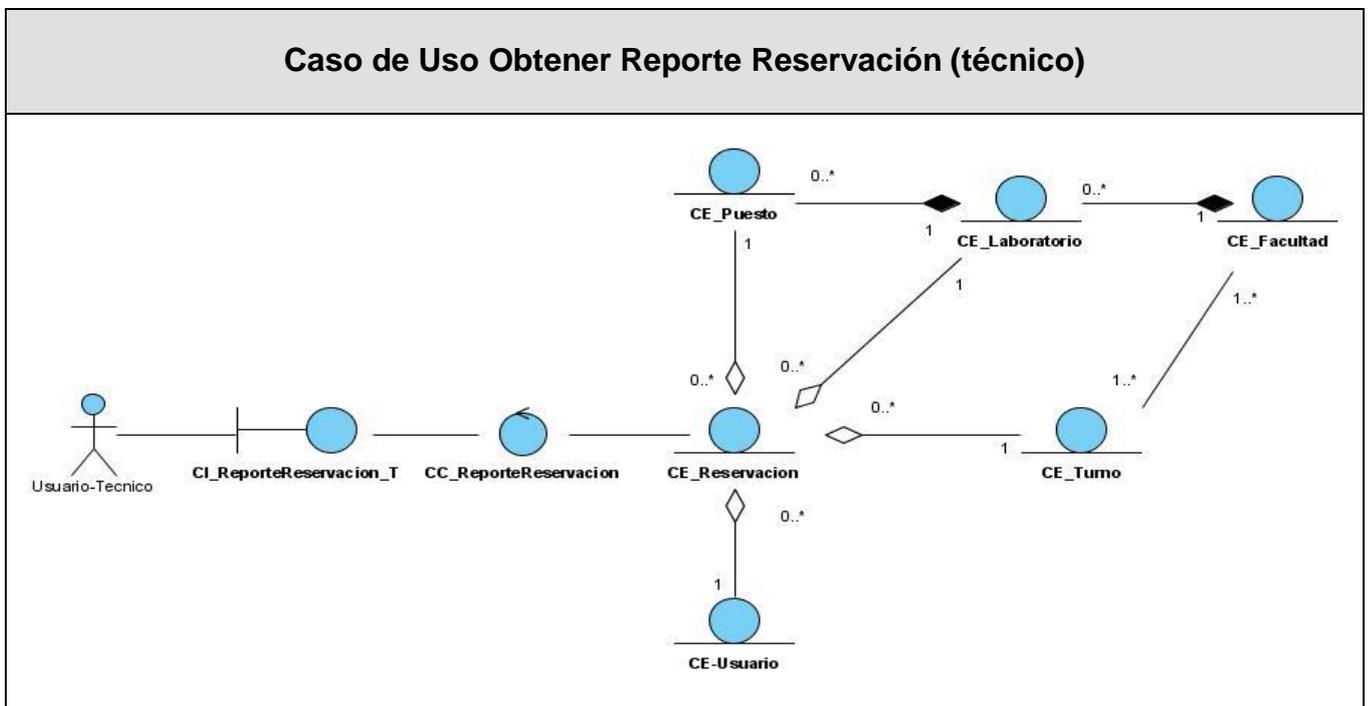


Figura 13 Análisis del caso de uso obtener reporte reservación (técnico)

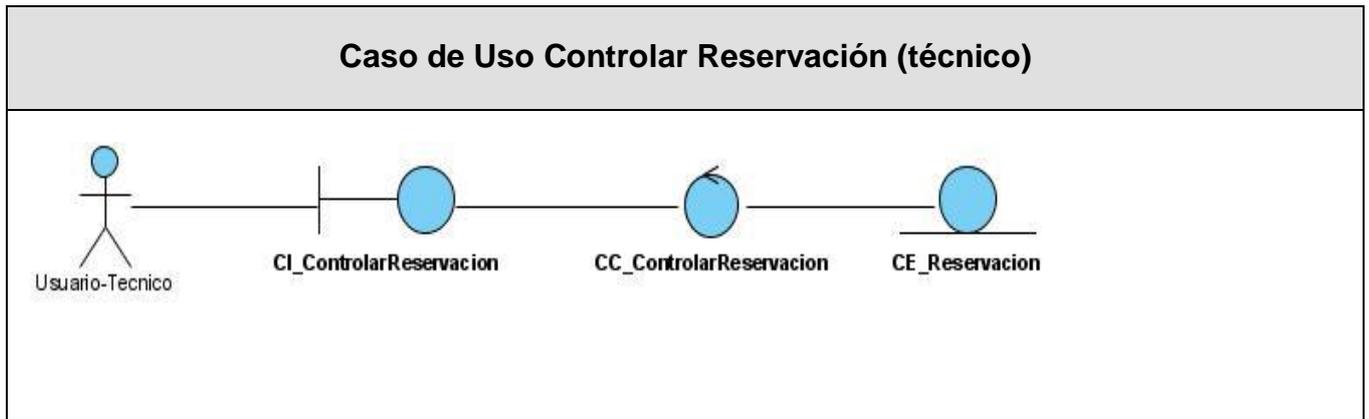


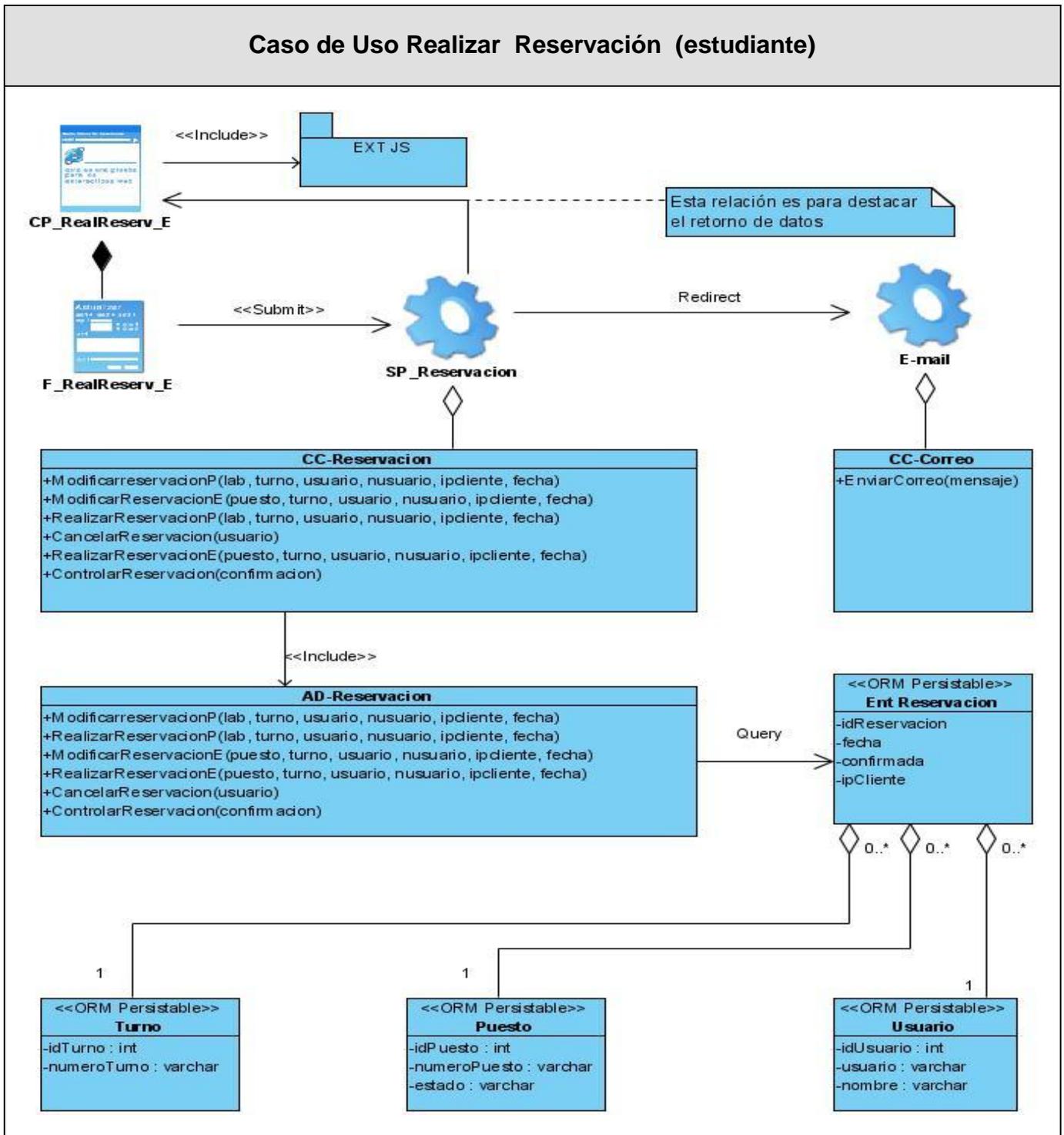
Figura 14 Análisis del caso de uso controlar reservación (técnico)

3.2. Diseño del Sistema

El diseño es un refinamiento del análisis que tiene en cuenta los requisitos no funcionales, define cómo cumple el sistema sus objetivos. El diseño debe garantizar que el sistema pueda ser implementado sin ambigüedades.

3.2.1. Diagramas de interacción

En el **Anexo_2** se muestran los diagramas de secuencia de los casos de uso para uno de los roles en específico.

Diagramas de clases del diseño

Figura 15 Diseño del caso de uso realizar reservación (estudiante)

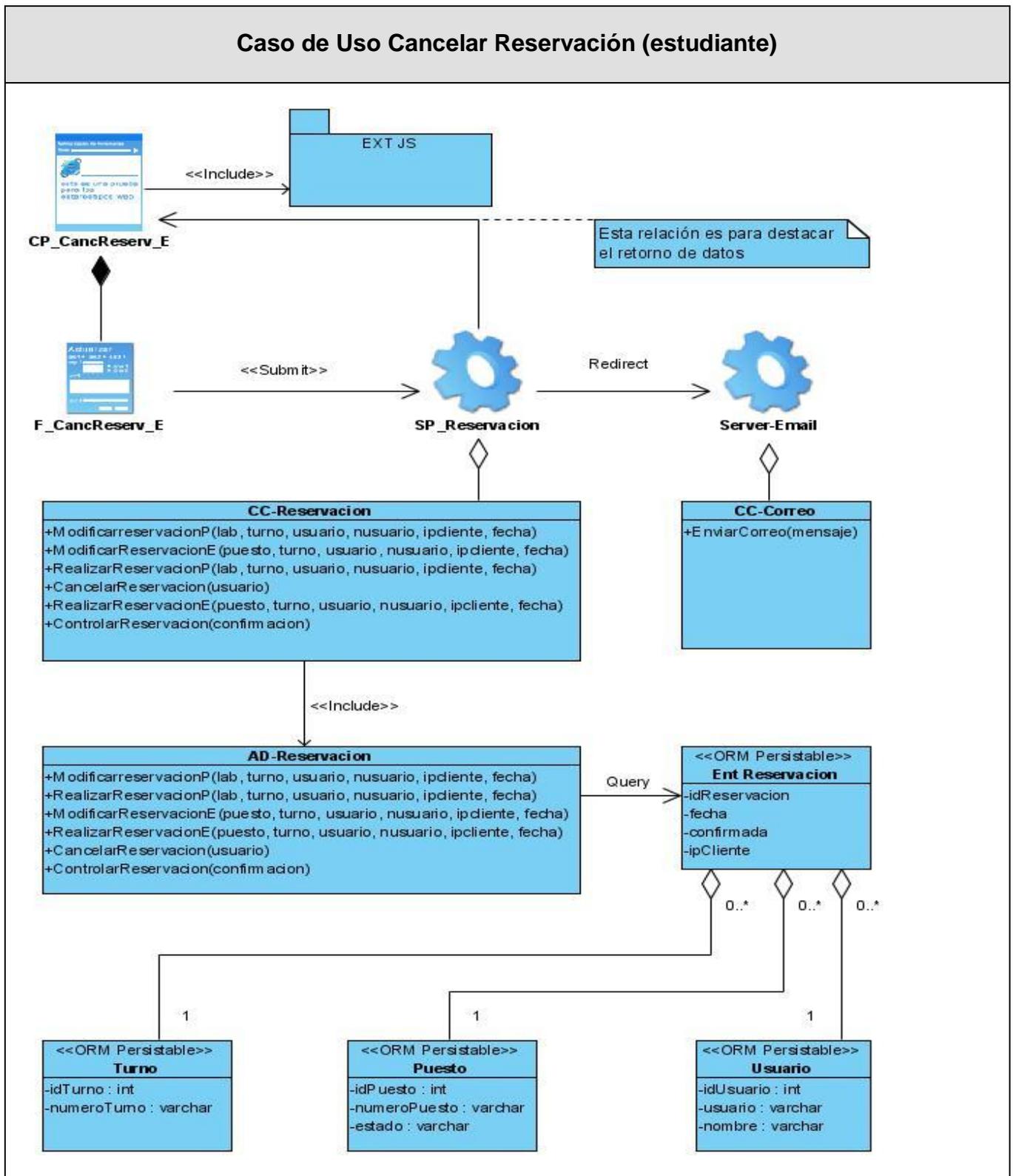


Figura 17 Diseño del caso de uso cancelar reservación (estudiante)

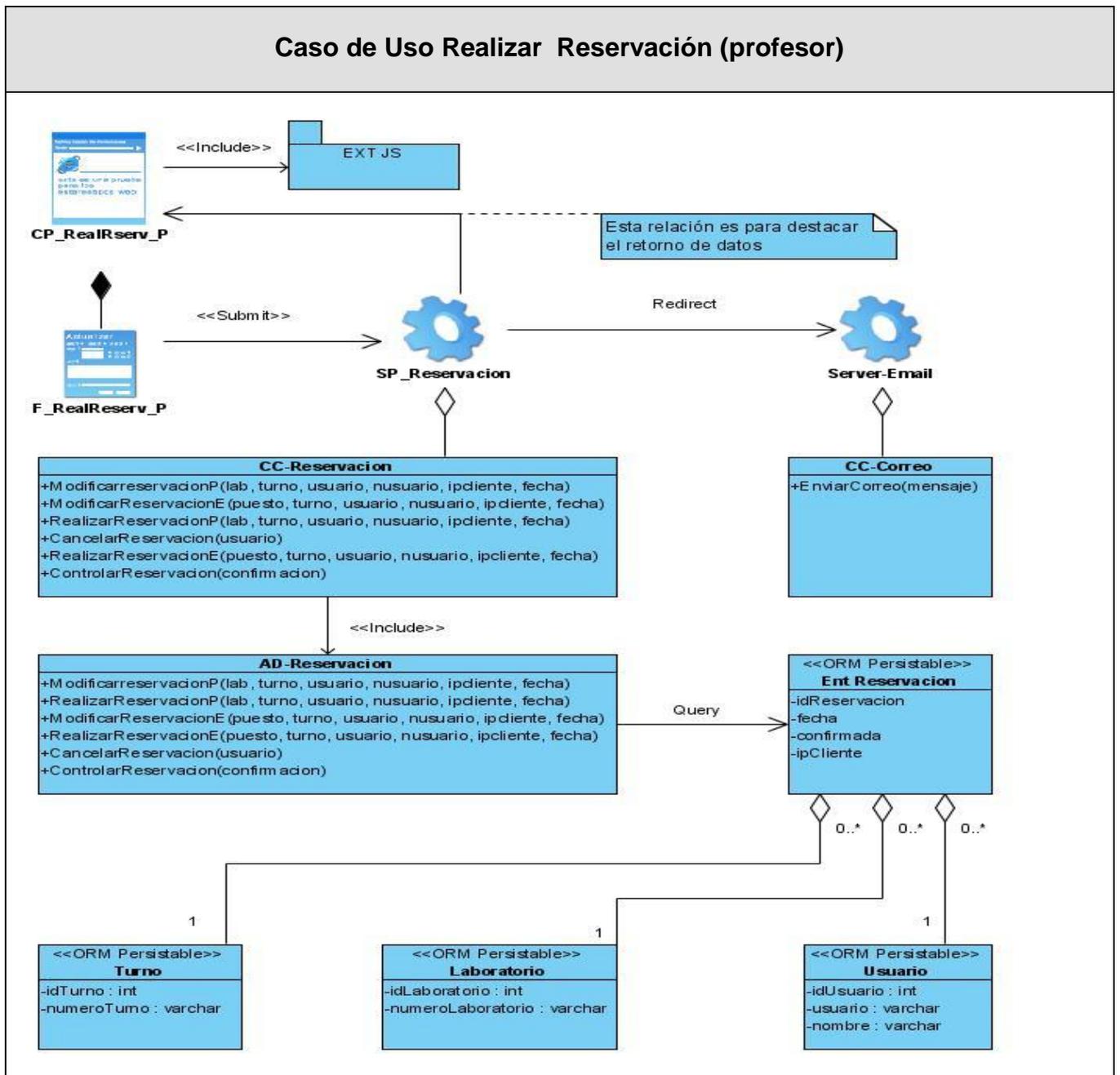


Figura 18 Diseño del caso de uso realizar reservación (profesor)

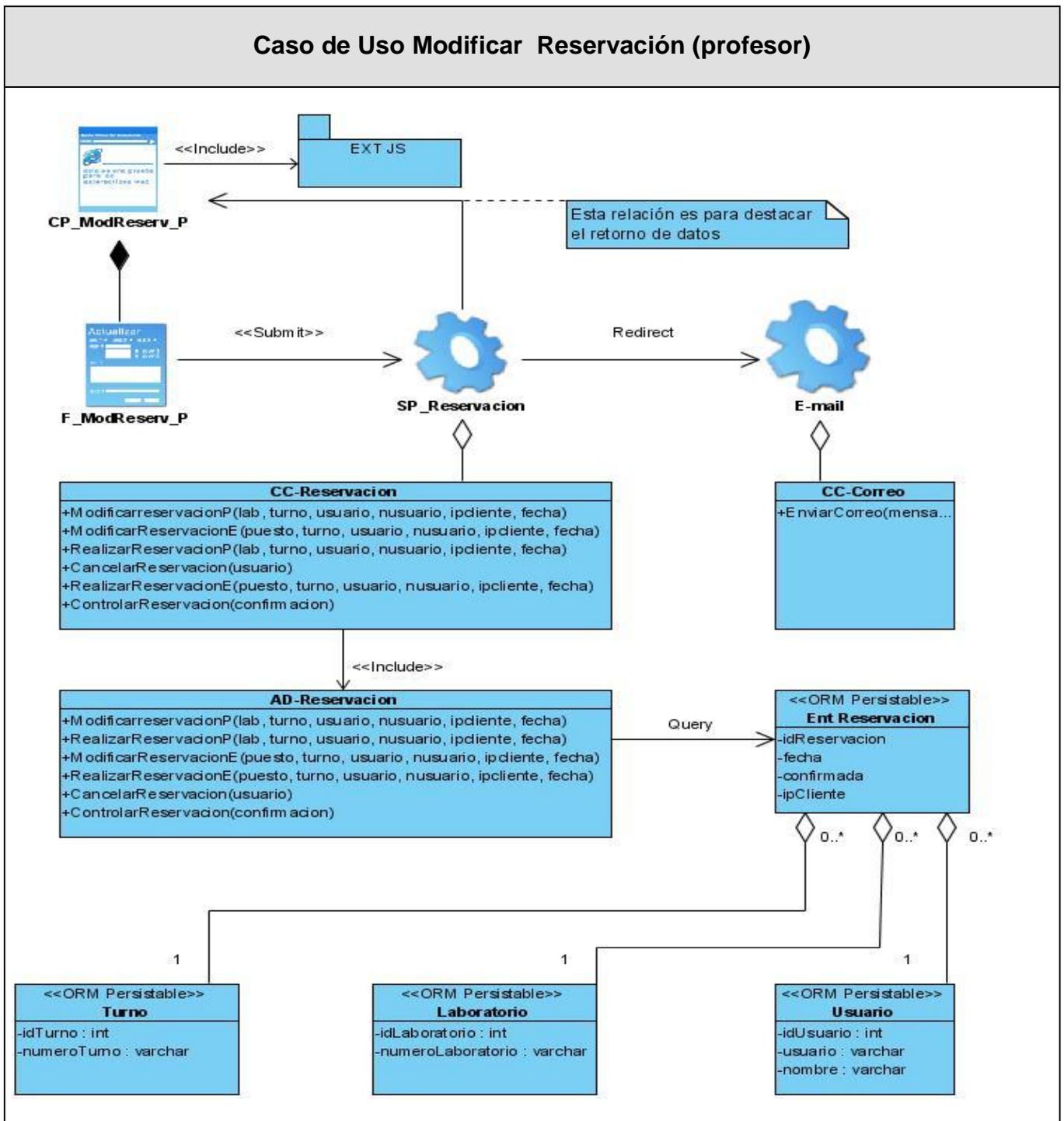
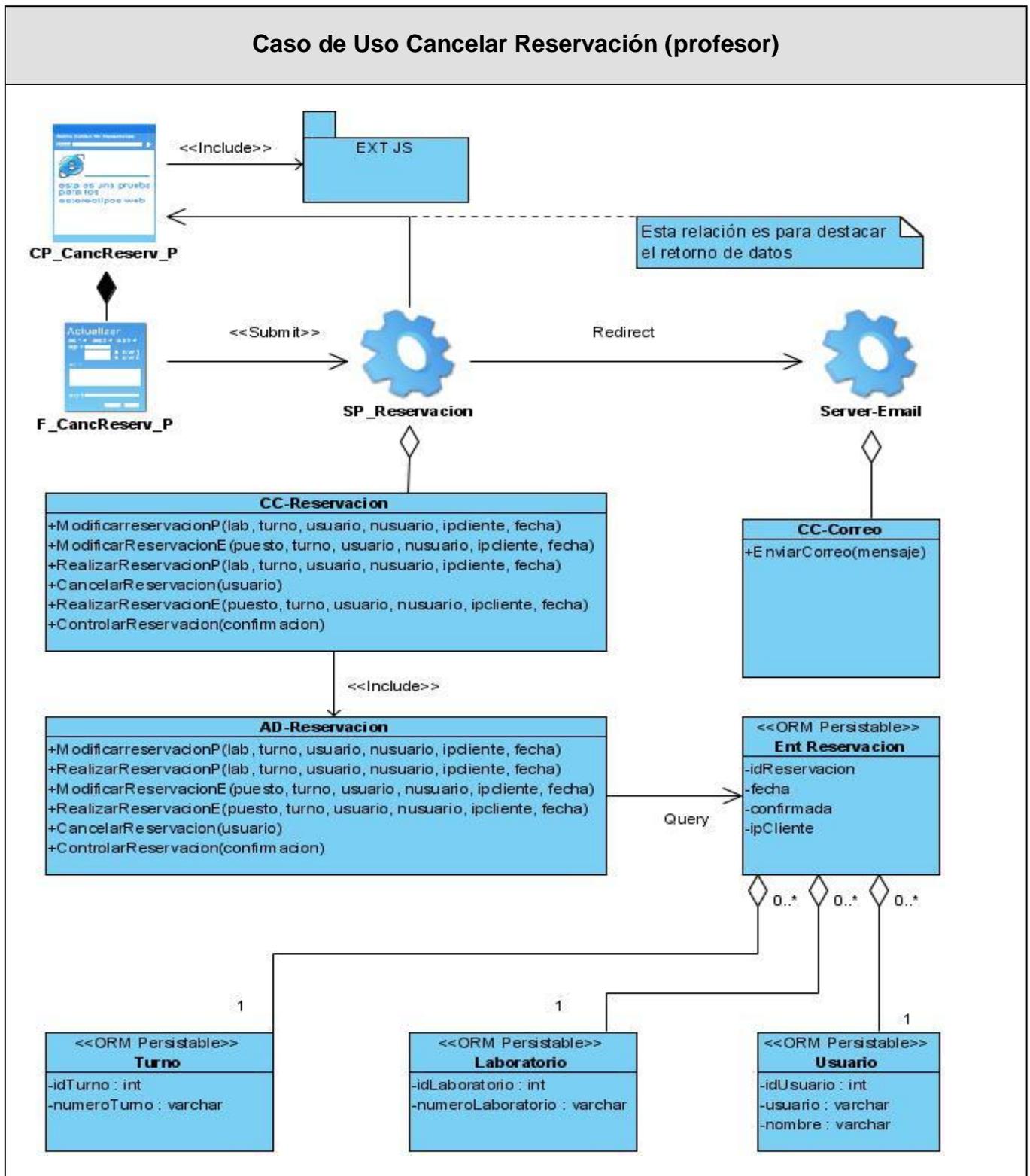


Figura 19 Diseño del caso de uso modificar reservación (profesor)


Figura 20 Diseño del caso de uso cancelar reservación (profesor)

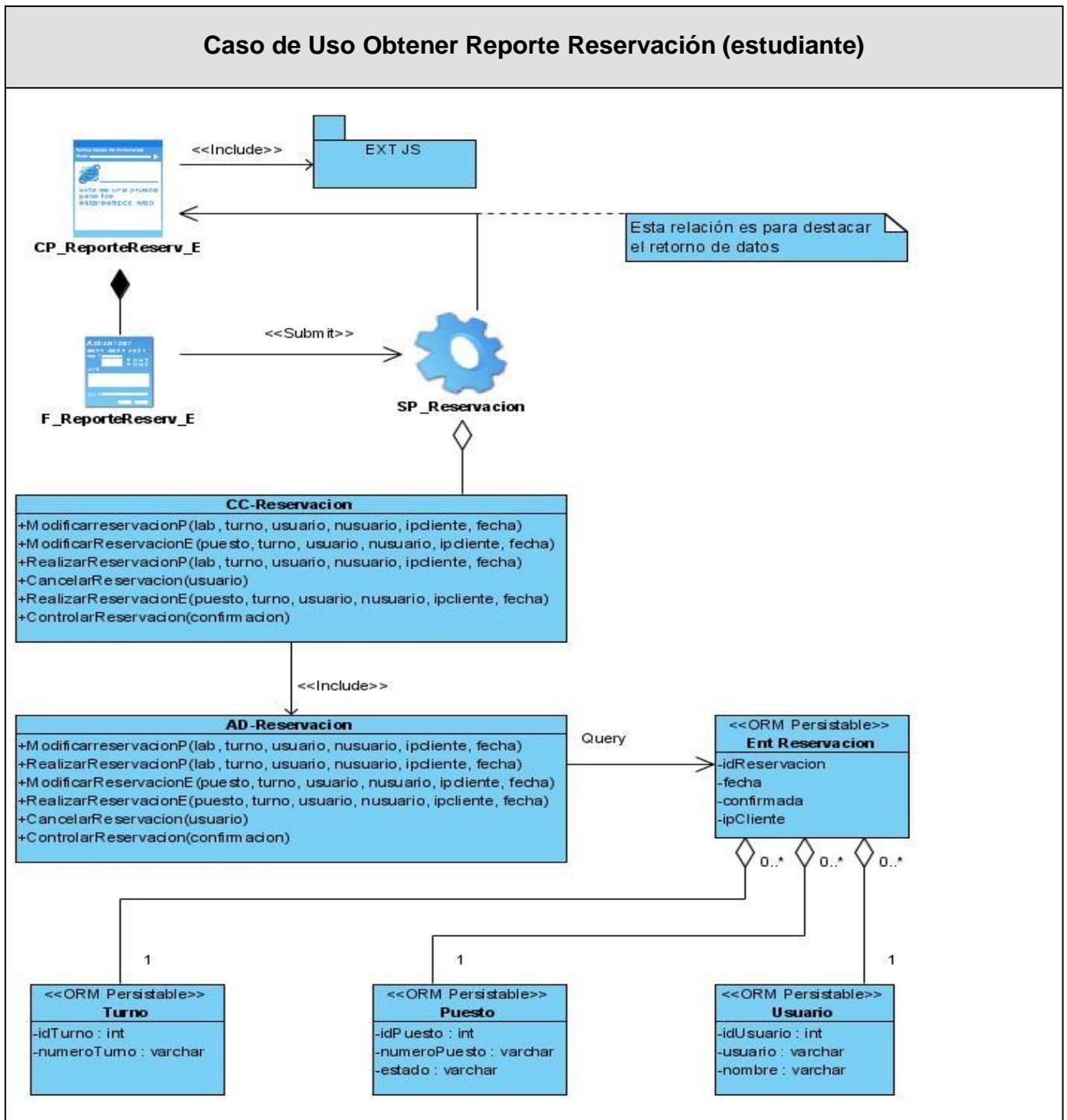


Figura 21 Diseño del caso de uso obtener reporte de reservación (estudiante)

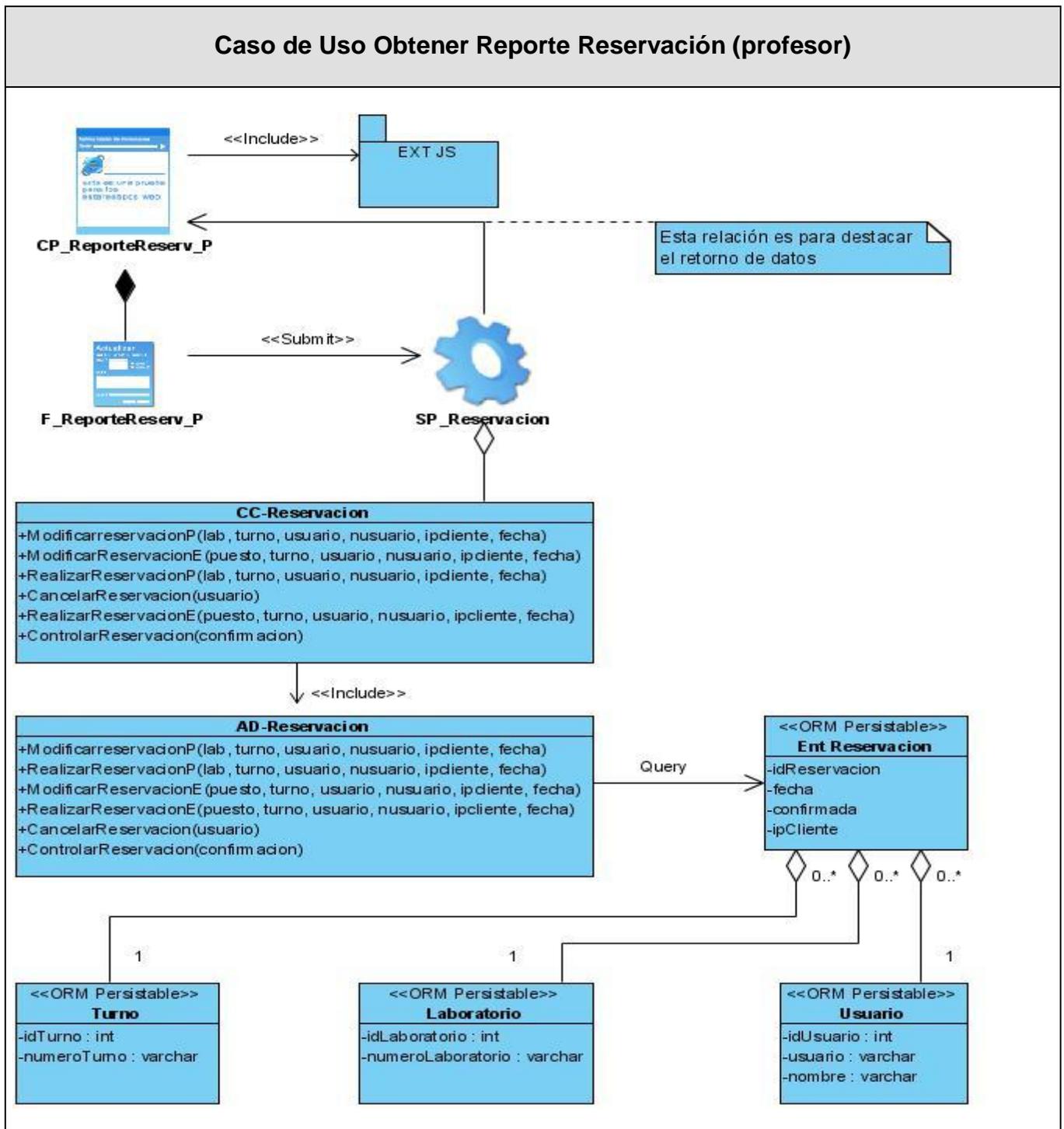


Figura 22 Diseño del caso de uso obtener reporte de reservación (profesor)

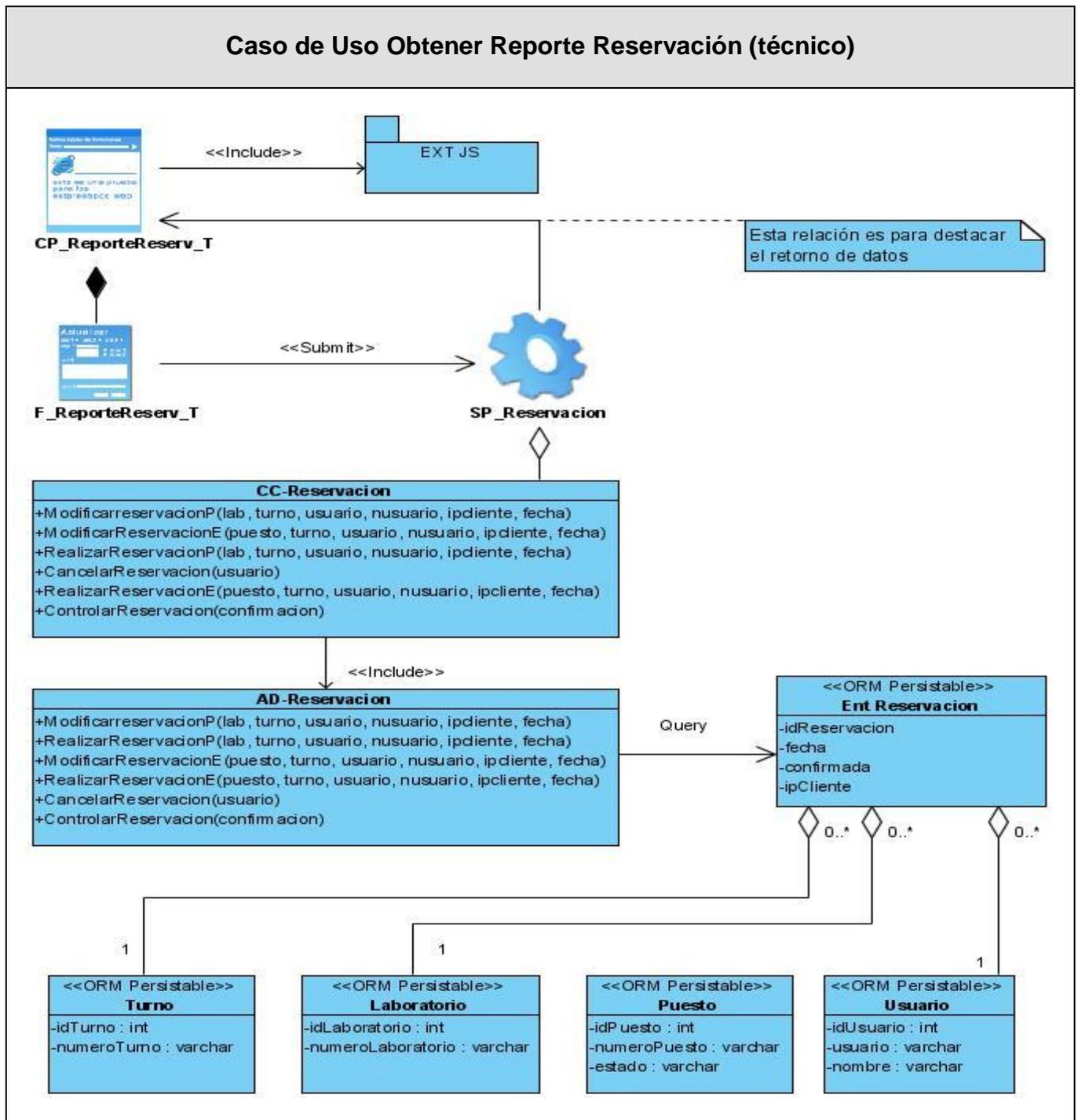
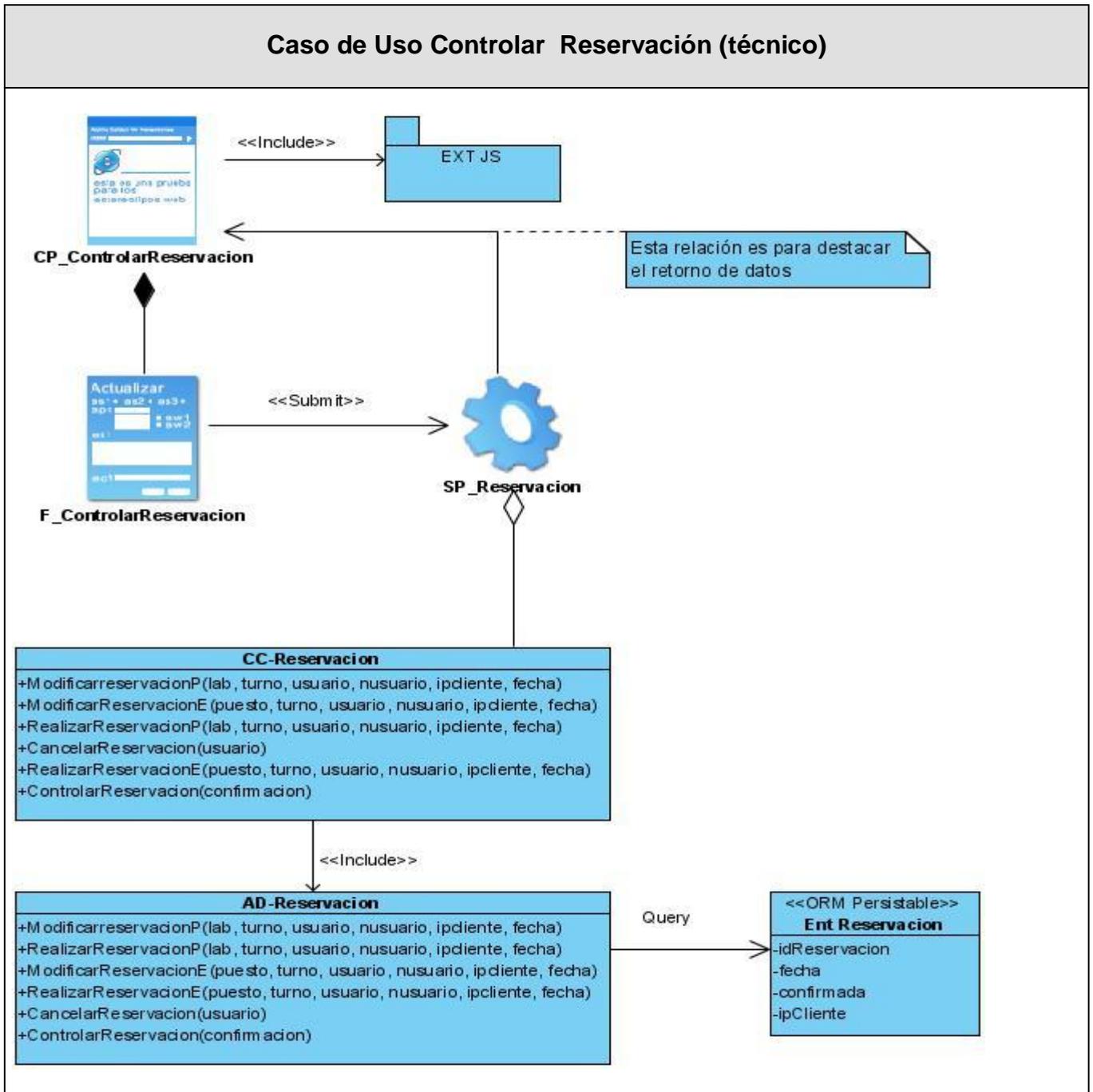


Figura 23 Diseño del caso de uso obtener reporte de reservación (técnico)


Figura 24 Diseño del caso de uso controlar reservación (técnico)

3.3. Diseño de la base de datos

Hoy en día las empresas manejan una gran cantidad de datos. Cualquier empresa que se precie debe tener almacenados todos estos datos en una base de datos para poder realizarlos mediante una aplicación profesional; sin esta funcionalidad resultaría imposible tratar y manejar en su totalidad los datos que lleva a cabo la empresa y se perdería un tiempo y un dinero muy valiosos. Uno de los pasos cruciales en la construcción de una aplicación que maneje una base de datos, es sin duda, el diseño de la base de datos.

3.3.1. Modelo lógico de datos (diagrama de clases persistente)

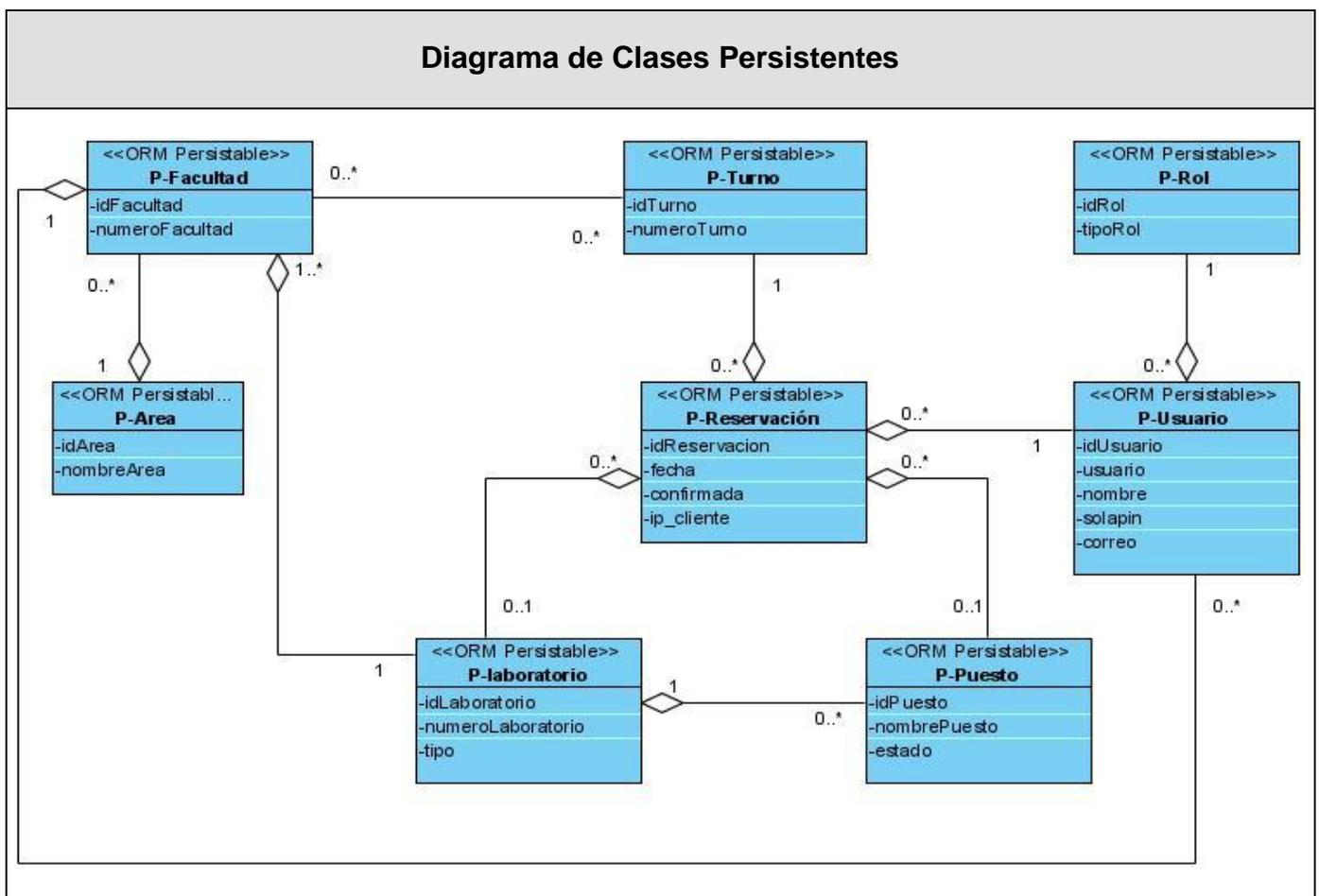
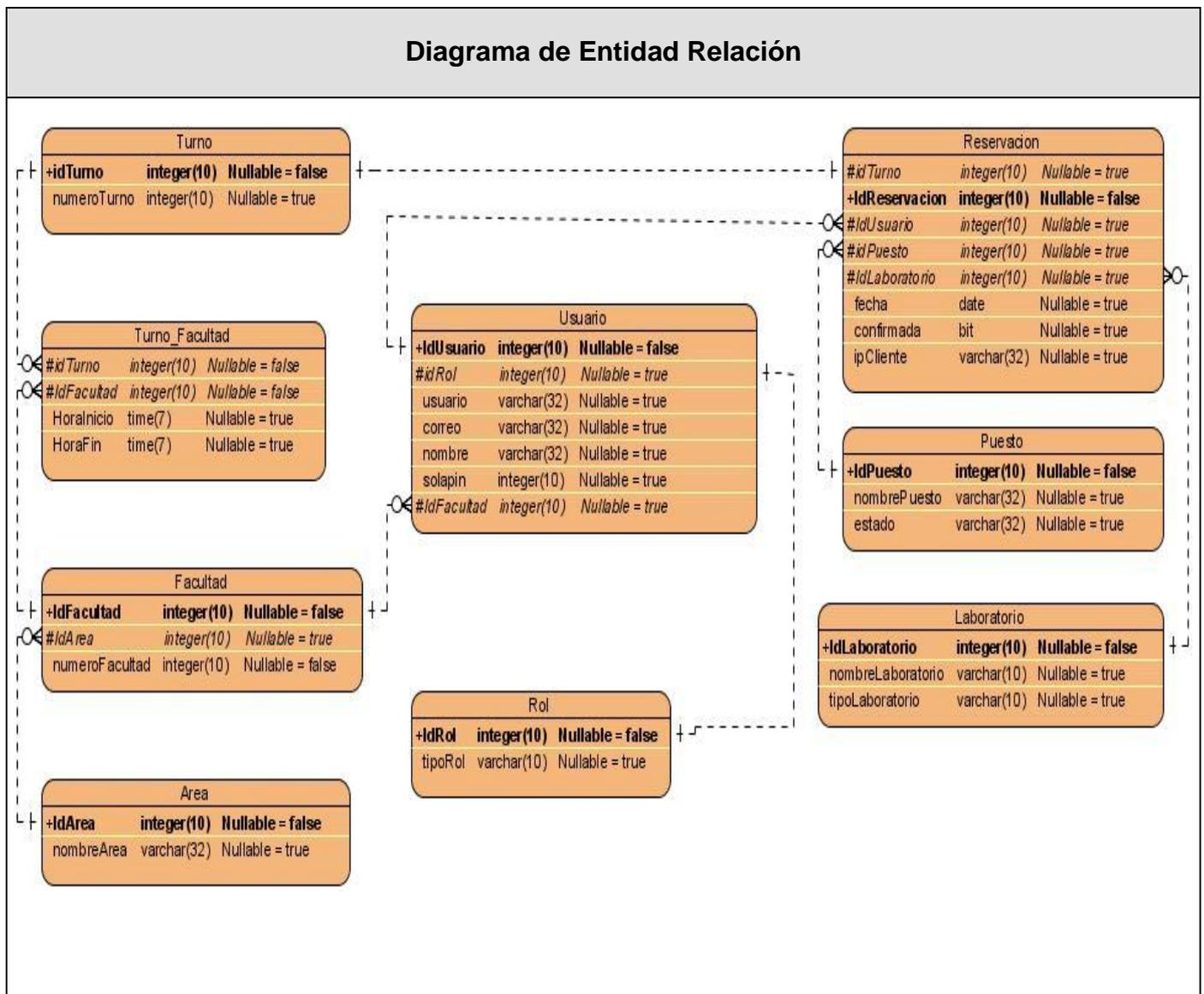


Figura 25 Modelo lógico de datos

3.3.2. Modelo físico de datos (modelo de datos)

Figura 26 Modelo físico de datos
3.3.2.1. Descripción de las Clases del modelo de datos

En el Anexo_3 se muestran las descripciones de las clases correspondientes al modelo de datos

Conclusiones

En este capítulo se abordó el análisis y el diseño del módulo propuesto, se desarrollaron los diagramas de clases del análisis con el objetivo de estructurar los requisitos para una mejor comprensión y el modelo del diseño, basándonos en un arquitectura de tres capas, además se detallan las secuencias de acciones del proceso a automatizar en los diagramas de interacción de los casos de usos críticos. Se obtuvo el diagrama de clases persistentes y el diagrama entidad relación donde se describen los atributos y las relaciones entre las tablas, quedando definida la arquitectura base del sistema a implementar.

4 IMPLEMENTACIÓN

Introducción

En este capítulo se comienza con el resultado del diseño y se implementa el sistema en términos de componentes, es decir, ficheros de código fuente, scripts, ficheros de código binario, ejecutables y similares. El principal objetivo es describir cómo los elementos del modelo del diseño se implementan en términos de componentes y cómo estos se organizan de acuerdo a los nodos específicos en el modelo de despliegue.

4.1. Diagrama de Despliegue

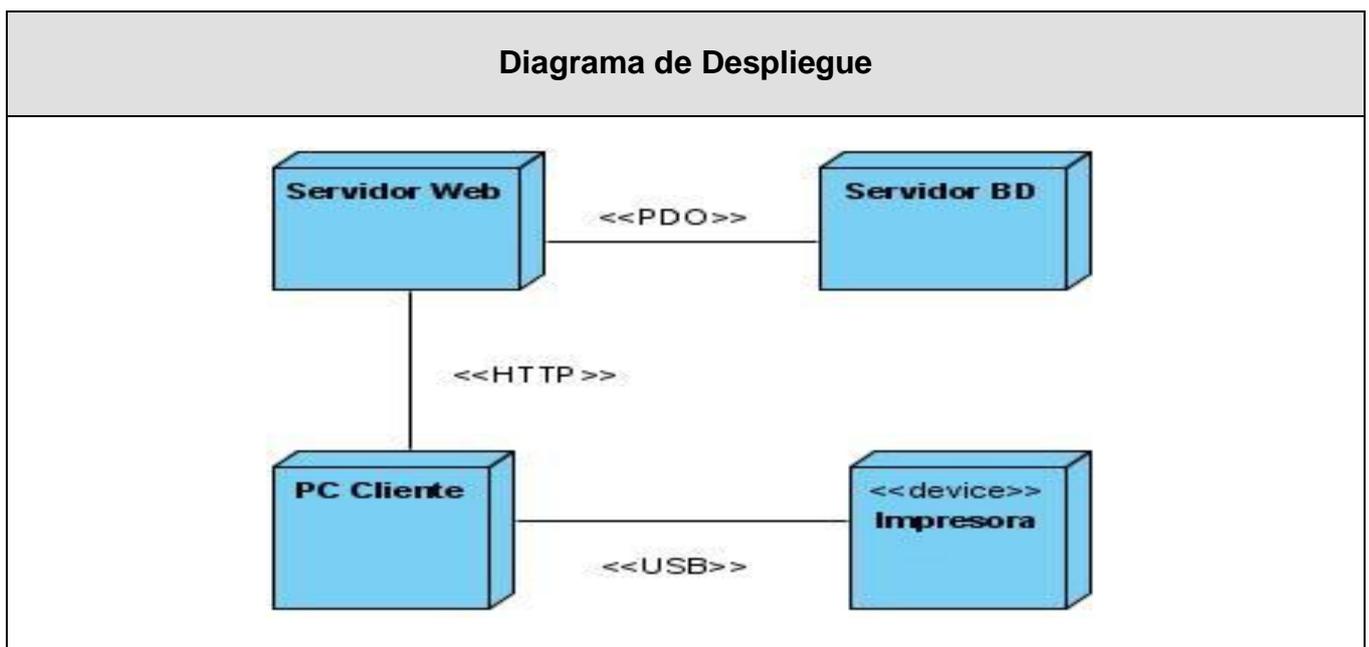


Figura 27 Diagrama de despliegue

4.2. Diagrama de Componentes

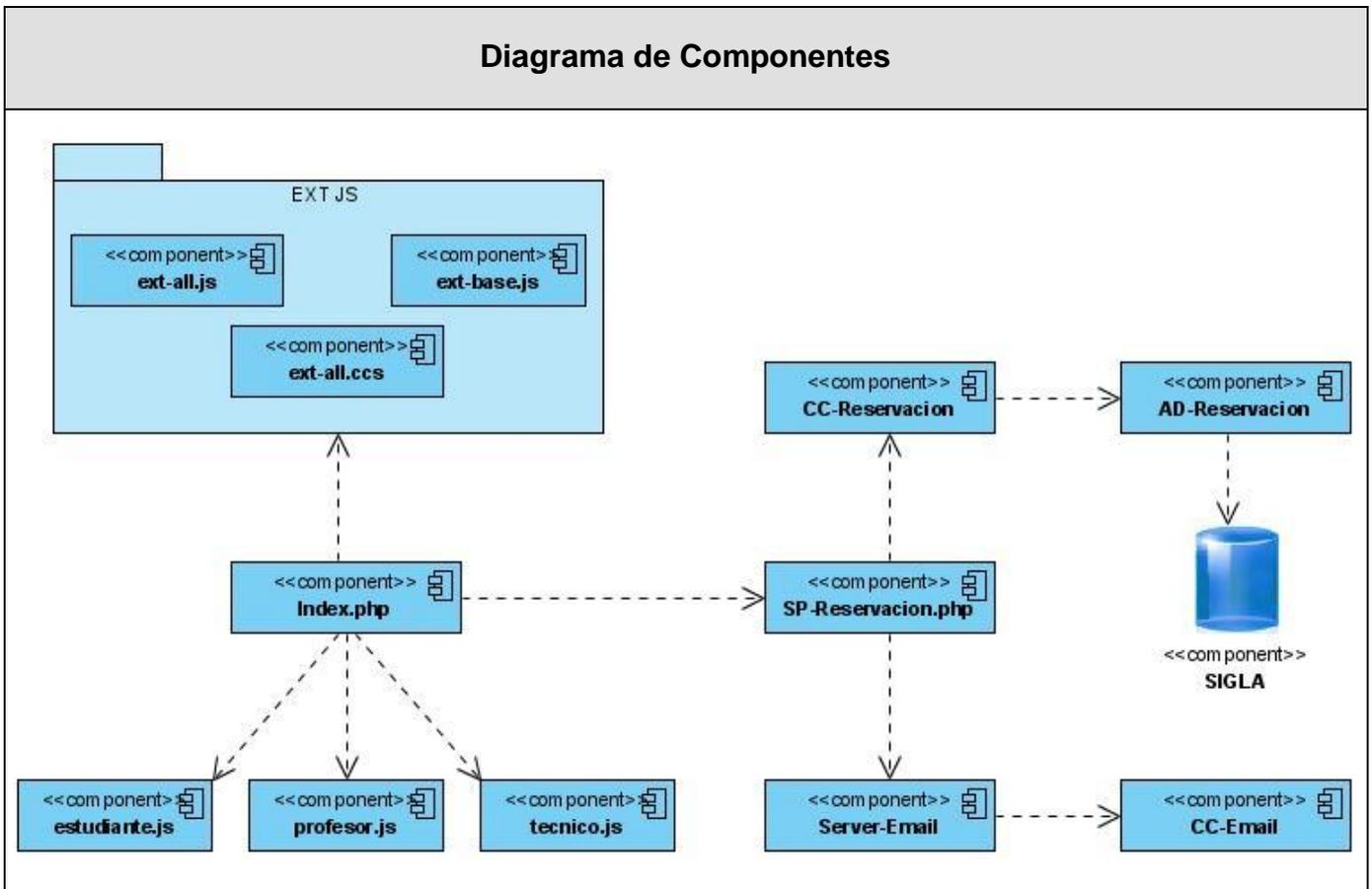


Figura 28 Diagrama de componentes

Conclusiones

En este capítulo se tiene como principal resultado el refinamiento de la vista de arquitectura del modelo de despliegue. Además, a través del modelo de componentes se muestran las relaciones y dependencia entre los componentes físicos existentes en la implementación del caso de estudio. Este capítulo constituye la entrada principal a las etapas de pruebas que siguen a la implementación.

5 ESTUDIO DE FACTIBILIDAD

Introducción

Antes de desarrollar un producto de software es de vital importancia realizar un estudio del costo del mismo, así como los beneficios que reportará. Como resultado de este análisis se obtiene el tiempo de desarrollo en meses, costo y la cantidad de personas que se necesitan para desarrollar el proyecto. En este capítulo se describe la estimación de costos del sistema propuesto y sus beneficios.

5.1. Método de Estimación por Puntos de Casos de Uso.

La estimación mediante el análisis de Puntos de Casos de Uso es un método propuesto originalmente por Gustav Karner, y posteriormente refinado por muchos otros autores.

Se trata de un método de estimación del tiempo de desarrollo de un proyecto mediante la asignación de "pesos" a un cierto número de factores que lo afectan, para finalmente, contabilizar el tiempo total estimado para el proyecto a partir de esos factores. A continuación se despliega detalladamente dicho método.

1. Cálculo de Puntos de Casos de Uso sin ajustar.

Se calcula a partir de la siguiente ecuación:

$$\text{UUCP} = \text{UAW} + \text{UUCW}$$

Donde:

UUCP: Puntos de Casos de Uso sin ajustar

UAW: Factor de Peso de los Actores sin ajustar

UUCW: Factor de Peso de los Casos de Uso sin ajustar

Factor de Peso de los Actores sin ajustar (UAW)

Este valor se calcula mediante un análisis de la cantidad de Actores presentes en el sistema y la complejidad de cada uno de ellos. La complejidad de los Actores se establece según la descripción de la siguiente tabla:

Para calcular UAW

Tipo	Descripción	Peso	Cant* peso
Simple	Otro sistema que interactúa con el sistema a desarrollar mediante una interfaz de programación (API, Application Programming Interface)	1	0*1
Medio	Otro sistema que interactúa con el sistema a desarrollar mediante un protocolo o una interfaz basada en texto	2	0*2
Complejo	Una persona que interactúa con el sistema mediante una interfaz gráfica	3	3*3
Total			9

Tabla 2 factor de peso de los actores sin ajustar

UAW=9.

Factor de Peso de los Casos de Uso sin ajustar (UUCW)

Este valor se calcula mediante un análisis de la cantidad de Casos de Uso presentes en el sistema y la complejidad de cada uno de ellos. La complejidad de los Casos de Uso se establece teniendo en cuenta la cantidad de transacciones efectuadas en el mismo, donde una transacción se entiende como una secuencia de actividades atómica, es decir, se efectúa la secuencia de actividades completa, o no se efectúa ninguna de las actividades de la secuencia. Los criterios se muestran en la siguiente tabla:

Para calcular UUCW

Tipo	Descripción	Peso	Cant * peso
Simple	El Caso de Uso contiene de 1 a 3 transacciones	5	1*5
Medio	El Caso de Uso contiene de 4 a 7 transacciones	10	0*10
Complejo	El Caso de Uso contiene más de 8 transacciones	15	2*15
Total			35

Tabla 3 Factor de peso de los casos de usos sin ajustar

UUCW = 35.

UUCP = **UAW** + **UUCW** = 9+ 35 = 44

2. Cálculo de Puntos de Casos de Uso ajustados

Una vez que se tienen los Puntos de Casos de Uso sin ajustar, se debe ajustar éste valor mediante la siguiente ecuación:

UCP = **UUCP** x **TCF** x **EF**

Donde:

UCP: Puntos de Casos de Uso ajustados

UUCP: Puntos de Casos de Uso sin ajustar

TCF: Factor de complejidad técnica

EF: Factor de ambiente

Factor de complejidad técnica (TCF)

Este coeficiente se calcula mediante la cuantificación de un conjunto de factores que determinan la complejidad técnica del sistema. Cada uno de los factores se cuantifica con un valor de 0 a 5, donde 0 significa un aporte irrelevante y 5 un aporte muy importante. En la siguiente tabla se muestra el peso y el valor asignado de cada uno de éstos factores.

Factor	Descripción	Peso	Valor asignado	Σ (Peso_i * Valor_i)
T1	Sistema distribuido	2	3	6
T2	Tiempo de respuesta	1	4	4
T3	Eficiencia del usuario final	1	4	4
T4	Procesamiento interno complejo	1	0	0
T5	El código debe ser reutilizable	1	5	5
T6	Facilidad de instalación	0.5	5	2.5
T7	Facilidad de uso	0.5	4	2.0
T8	Portabilidad	2	4	8
T9	Facilidad de cambio	1	4	4
T10	Concurrencia	1	5	5
T11	Incluye objetivos especiales de seguridad	1	3	3

T12	Provee acceso directo a terceras partes	1	2	2
T13	Se requieren facilidades especiales de entrenamiento a los usuarios.	1	2	2
Total				47.5

Tabla 4 Factor de complejidad técnica

$$\text{TCF} = 0.6 + 0.01 * \Sigma (\text{Peso}_i * \text{Valor}_i)$$

$$\text{TCF} = 0.6 + 0.01 * 47.5 = 1.075$$

Factor de ambiente (EF)

Las habilidades y el entrenamiento del grupo involucrado en el desarrollo tienen un gran impacto en las estimaciones de tiempo. Estos factores son los que se contemplan en el cálculo del Factor de ambiente. El cálculo del mismo es similar al cálculo del Factor de complejidad técnica, es decir, se trata de un conjunto de factores que se cuantifican con valores de 0 a 5. En la siguiente tabla se muestra el peso y el valor asignado de cada uno de éstos factores.

Factor	Descripción	Peso	Valor asignado	$\Sigma (\text{Peso}_i * \text{Valor}_i)$
E1	Familiaridad con el modelo de proyecto utilizado	1.5	0	0
E2	Experiencia en la aplicación	0.5	0	0
E3	Experiencia en	1	3	3



	orientación a objetos			
E4	Capacidad del analista líder	0.5	4	2.0
E5	Motivación	1	5	5
E6	Estabilidad de los requerimientos	2	4	8
E7	Personal part-time	-1	3	-3
E8	Dificultad del lenguaje de programación	-1	4	-4
Total				11

Tabla 5 Factor ambiente

$$EF = 1.4 - 0.03 * \sum (\text{Peso}_i * \text{Valor}_i)$$

$$EF = 1.4 - 0.03 * 10 = \mathbf{1.07}$$

$$UUCP = 44$$

$$TCF = 1.075$$

$$EF = 1.07$$

$$UCP = UUCP * TCF * EF = 44 * 1.075 * 1.07 = \mathbf{50.611}$$

3. De los Puntos de Casos de Uso a la estimación del esfuerzo

El esfuerzo en horas-hombre viene dado por:

$$E = UCP \times CF$$

Donde:

E: Esfuerzo estimado en horas-hombre

UCP: Puntos de Casos de Uso ajustados

CF: Factor de conversión

Este método proporciona una estimación del esfuerzo en horas-hombre contemplando sólo el desarrollo de la funcionalidad (programación) especificada en los casos de uso.

Se contabilizan cuántos factores de los que afectan al Factor de ambiente están por debajo del valor medio (3), para los factores E1 a E6.

Se contabilizan cuántos factores de los que afectan al Factor de ambiente están por encima del valor medio (3), para los factores E7 y E8.

CF = 20 horas-hombre (si $Total_{EF} \leq 2$)

CF = 28 horas-hombre (si $Total_{EF} = 3$ ó $Total_{EF} = 4$)

CF = abandonar o cambiar proyecto (si $Total_{EF} \geq 5$)

Total_{EF} = Cant EF < 3 (entre E1 –E6) + Cant EF > 3 (entre E7, E8)

Total_{EF} = 2+1 = 3

CF = 28 horas-hombre (porque $Total_{EF} = 3$)

UCP =50.611

CF = 28 horas-hombre.

E = UCP x CF= 50.611 * 28 horas-hombre=1417.108~1418.

4. Esfuerzo de todo el proyecto

Actividad	% esfuerzo	Valor esfuerzo
Análisis	10%	354.5 horas-hombre
Diseño	20%	709 horas-hombre
Programación	40%	1418 horas-hombre
Prueba	15%	531.75 horas-hombre
Sobrecarga	15%	531.75 horas-hombre
Total	100%	3545 horas-hombre

Tabla 6 Esfuerzo total del proyecto

$E_T = 3545$ horas-hombre y se estima que cada mes tiene como promedio 240 horas laborables eso daría un $E_T = 14.77$ mes-hombre.

Se estima que una persona podrá realizar el problema analizado en 15 meses aproximadamente.

Costo del Proyecto.

Se asume como salario promedio mensual \$100.00

CH: Cantidad de hombres.

T: Tiempo total del proyecto.

CHM: Costo de hombre por mes

CH = 2 hombres

CHM = 2 * Salario Promedio = 200.00 \$/mes

$$\text{Costo} = \text{CHM} * E_T / \text{CH}$$

$$\text{Costo} = 200.00 * 14.77 / 2 = \$1477$$

$$\text{Tiempo} = E_T / \text{CH}$$

$$\text{Tiempo} = 14.77 / 2 = 7.385 \text{ meses}$$

De los resultados obtenidos se estima que con dos hombres trabajando en el proyecto, el mismo se desarrolla en 7 meses y medio aproximadamente, con un costo total estimado en 1477 pesos.

5.2. Beneficios tangibles e intangibles

El producto de software desarrollado aportará como beneficio fundamental a los usuarios una vía dinámica para realizar, modificar y cancelar reservaciones. El mismo permitirá conocer de una forma más precisa y en el menor tiempo posible la información referente a las reservaciones, además de posibilitar un mayor control y seguridad en los laboratorios.

Por tanto, los beneficios inmediatos son generalmente intangibles:

- Disminución del tiempo y esfuerzo que se invierte en esta tarea que se realiza, hasta ahora, de forma manual.
- Disminución de la acumulación de materiales impresos relacionados con los procesos de reservación de los laboratorios.
- Ahorro de gran cantidad de material de oficina, al contar con una base de datos para el almacenamiento de la información.
- Mayor seguridad, control y organización en los laboratorios.
- Facilidad en la obtención de reportes de las reservaciones.
- Acceso al servicio desde cualquier puesto de trabajo.

5.3. Análisis de costos y beneficios.

Teniendo en cuenta el costo del proyecto y los beneficios que reportará el módulo que se propone para automatizar el proceso de reservación de tiempo de máquina en la UCI se puede concluir que la elaboración de este es factible.

Además, se puede agregar a esto que el desarrollo del producto será con tecnología libre, factor que anulará gastos en licencia.



Conclusiones.

En el capítulo se realizó un estudio de la factibilidad del producto de software. Se puede concluir que la elaboración del producto es factible dado los beneficios importante que arrojará en el proceso de informatización en la UCI, especialmente en el servicio y control de los laboratorios docentes; teniendo en cuenta esto, se destaca que el costo del producto es tolerable.

CONCLUSIONES

Mediante el desarrollo de este trabajo se realizó un estudio de diferentes sistemas de reservaciones, obteniendo con el mismo una visión clara del ascenso que tiene el uso de este tipo de aplicaciones en el mundo dado los aportes económicos y sociales que tributan.

Se logró obtener una interfaz interactiva y amena. Se abaló la propuesta del cliente de utilizar la Ext JS, dado que la misma se identifica con la web 2.0.

Se desarrollaron los diagramas correspondientes del desarrollo del software hasta el flujo de trabajo de implementación, garantizando con esto una mejor comprensión del funcionamiento del sistema.

Para la elaboración de la aplicación se tuvo en cuenta la petición por parte del cliente, así como las recomendaciones elaboradas por los primeros desarrolladores de este trabajo de usar PHP como lenguaje de programación para interactuar con el servidor, Ext JS como librería emergente en el mundo para el desarrollo de interfaces interactivas y como gestor de base de datos PostgreSQL.

Se cumplió con el objetivo general propuesto, ya que se obtuvo la primera versión del módulo de reservación de tiempo de máquinas en la Universidad de las Ciencias Informáticas, perteneciente al proyecto Sistema Integral de Gestión de los Laboratorios (SIGLA). Dicha versión permite realizar reservación a usuarios del dominio uci y brinda distintos tipos de servicios.

RECOMENDACIONES

Realizarle las pruebas al sistema, así como su despliegue a nivel de universidad para contribuir a una mejor gestión de la reservación de tiempo de máquina en la UCI, automatizando dicho proceso y dándole cumplimiento al objetivo general de este trabajo de diploma.

Se recomienda además integrar el sistema con un framework de PHP, principalmente Symfony por sus características y fácil integración con la librería de JavaScript Ext JS.

BIBLIOGRAFÍA

Álvarez Romero, Eduardo and Pueyo, Daniel. 2004-2005. *Integration Definition For Funcion Modeling.* [Online] 2004-2005. [Cited: Diciembre 12, 2007.] <http://dmi.uib.es/~burguera/download/IDEF0trabajo.doc>.

Alvarez, Miguel Angel. *Aptana Studio.* [Online] [Cited: Diciembre 7, 2007.] <http://www.desarrolloweb.com/articulos/aptana-studio.html>.

Amadeus. [Online] [Cited: Noviembre 28, 2007.] <http://www.amadeus.com/mx/x39150.html>.

Apache. 1995-2006. *Versión 2.2 de la documentación del Servidor de HTTP Apache.* [Online] 1995-2006. [Cited: Enero 20, 2008.] <http://www.apache2.org/>.

Artiles Visual, Leticia, et al. *PERFECCIONAMIENTO DE LA GESTION UNIVERSITARIA EN EL ISCMH: Proyecto Tocooro.* [Online] [Cited: Enero 8, 2008.] http://www.ucmh.sld.cu/rhab/vol5_num1/rhcm10106.htm.

Departamento de Control de Calidad y Auditoría Informática. 2001. *Sistemas en Arquitectura Cliente/Servidor.* [Subdirección de Sistemas] s.l. : Dirección General de Servicios de Cómputo Académico, 2001.

DesarrolloWeb. *Qué es PHP.* [Online] [Cited: Noviembre 17, 2007.] <http://www.desarrolloweb.com/articulos/392.php>.

Edinet. 2002. *Qué es XML?* [Online] 2002. [Cited: Diciembre 12, 2007.] <http://www.edinet.com/sabia2.asp>.

Eguíluz Pérez, Javier. 2007. *Libros Web. Introducción a JavaScript.* [Online] 2007. [Cited: Diciembre 5, 2007.] <http://www.librosweb.es/javascript>.

—. **2007.** *Libros Web. Introducción a CSS.* [Online] 2007. [Cited: Enero 20, 2008.] <http://www.librosweb.es/css>.

—. 2007. Libros Web. *Introducción a XHTML*. [Online] 2007. [Cited: Enero 15, 2008.] <http://www.librosweb.es/xhtml>.

—. 2007. Libros Web. *Introducción a AJAX*. [Online] 2007. [Cited: Enero 8, 2008.] <http://www.librosweb.es/ajax>.

Encinosa, Lázaro Blanco. Apuntes para una historia de la informática en Cuba. [Online] [Cited: Enero 10, 2008.] www.sld.cu/galerias/doc/sitios/infodir/apuntes_para_una_historia_de_la_informatica_en_cuba.doc.

Ext JS. 2006. *Ext JS - JavaScript Library*. [Online] 2006. [Cited: Diciembre 10, 2007.] [http://extjs.com/.](http://extjs.com/)

GNU Image Manipulation Program. 2001. *GIMP*. [Online] 2001. [Cited: Diciembre 10, 2007.]

HTMLPOINT.com. 1997-2006. *¿Qué es Apache?* [Online] 1997-2006. [Cited: Diciembre 10, 2007.] <http://www.htmlpoint.com/faq/apache/01.htm>.

Lago, Ramiro. 2007. *Patrón "Modelo-Vista-Controlador"*. [Online] 2007. [Cited: Diciembre 10, 2007.] <http://www.proactiva-calidad.com/java/patrones/mvc.html>.

López Guzmán, Clara. 2000. *Modelo para el Desarrollo de Bibliotecas Digitales Especializadas*. [Online] Marzo 2000. [Cited: Enero 20, 2008.] http://www.bibliodgsca.unam.mx/tesis/tes7c1lg/sec_26.htm.

Maestros del Web. 1997-2007. *Conociendo los principales editores de páginas web del mercado*. [Online] 1997-2007. [Cited: Enero 10, 2008.] <http://www.maestrosdelweb.com/editorial/conociendo-los-principales-editores-de-paginas-web-del-mercado/>.

Michelle, Javier. Agencia de Noticias ORBITA desde el Perú. *Agencia de Noticias ORBITA desde el Perú*. [Online] [Cited: Junio 2, 2008.] http://agenciaorbita.com/index2.php?option=com_content&do_pdf=1&id=7643.

Microsoft. 2006. *La Arquitectura Orientada a Servicios (SOA) de Microsoft.* [Online] Diciembre 2006. [Cited: Marzo 28, 2008.] http://download.microsoft.com/download/c/2/c/c2ce8a3a-b4df-4a12-ba18-7e050aef3364/070717-Real_World_SOA.pdf.

pgAdmin PostgreSQL Tools . pgAdmin. [Online] [Cited: Diciembre 10, 2008.] <http://www.pgadmin.org/>.

Pitschke, Jurguen. 2002-2008. *Visual Paradigm Suite.* [Online] 2002-2008. [Cited: Enero 20, 2008.] <http://www.visual-paradigm.eu/content/view/12/1/lang,en/>.

ProgramacionWeb.net . 2003-2008. *MVC - Modelo Vista Controlador.* [Online] 2003-2008. [Cited: Enero 9, 2008.] <http://www.programacionweb.net/articulos/articulo/?num=505>.

Rumbaugh, James, Jacobson, Ivar and Booch, Grady. 2000. *El Lenguaje Unificado de Modelado.* [Online] 2000. [Cited: Enero 10, 2008.] <http://bibliodoc.uci.cu/pdf/reg00060.pdf>.

Sabre. Sabre. [Online] [Cited: Noviembre 29, 2007.] http://209.85.215.104/search?q=cache:wp4S5OdpkEQJ:www.educahres.com.ar/_db/_vitx/files/Docs/GDS/Sabre.doc+GetThere+%2BEI+l%C3%ADder+en+sistemas+de+reservaciones+basados&hl=es&ct=clnk&cd=5&gl=cu&lr=lang_es.

San Félix, Alvaro del Castillo. 2000. *El servidor de web Apache: Introducción práctica.* [Online] 2000. [Cited: Enero 8, 2008.] <http://acsblog.es/articulos/trunk/LinuxActual/Apache/html/index.html>.

Sánchez González, Carlos. 2004. ONess: un proyecto open source para el negocio textil mayorista desarrollado con tecnologías open source innovadoras. [Online] Septiembre 28, 2004. [Cited: Enero 20, 2008.] <http://oness.sourceforge.net/proyecto/html/ch03s02.html>.

Visual Paradigm. *Visual Paradigm for UML.* [Online] [Cited: Diciembre 10, 2007.] <http://www.visual-paradigm.com/product/vpuml/>.

Vizcaíno, Aurora, García, Felix Óscar and Caballero, Ismael. *Una Herramienta CASE para ADOO: Visual Paradigm.* [Online] [Cited: Noviembre 27, 2007.] http://alarcos.inf-cr.uclm.es/per/fgarcia/isoftware/doc/LabTr1_VP.pdf.

Welling, Luke and Thomson, Laura. *Desarrollo Web con PHP y MySQL.* [Online] [Cited: Enero 8, 2008.] <http://bibliodoc.uci.cu/pdf/reg02819.pdf>.

World Wide Web Consortium. 1994-2005. *Guía Breve de Tecnologías XML .* [Online] 1994-2005. [Cited: Noviembre 25, 2007.] <http://www.w3c.es/divulgacion/guiasbreves/TecnologiasXML>.

Worsley, John and Drake, Joshua. 2001. *Características de PostgreSQL.* [Online] 2001. [Cited: Noviembre 29, 2007.] <http://www.sobl.org/traduccion/practical-postgres/node19.html>.

—. **2001.** *¿Qué es PostgreSQL?* [Online] 2001. [Cited: Noviembre 29, 2007.] <http://www.sobl.org/traduccion/practical-postgres/node12.html>.

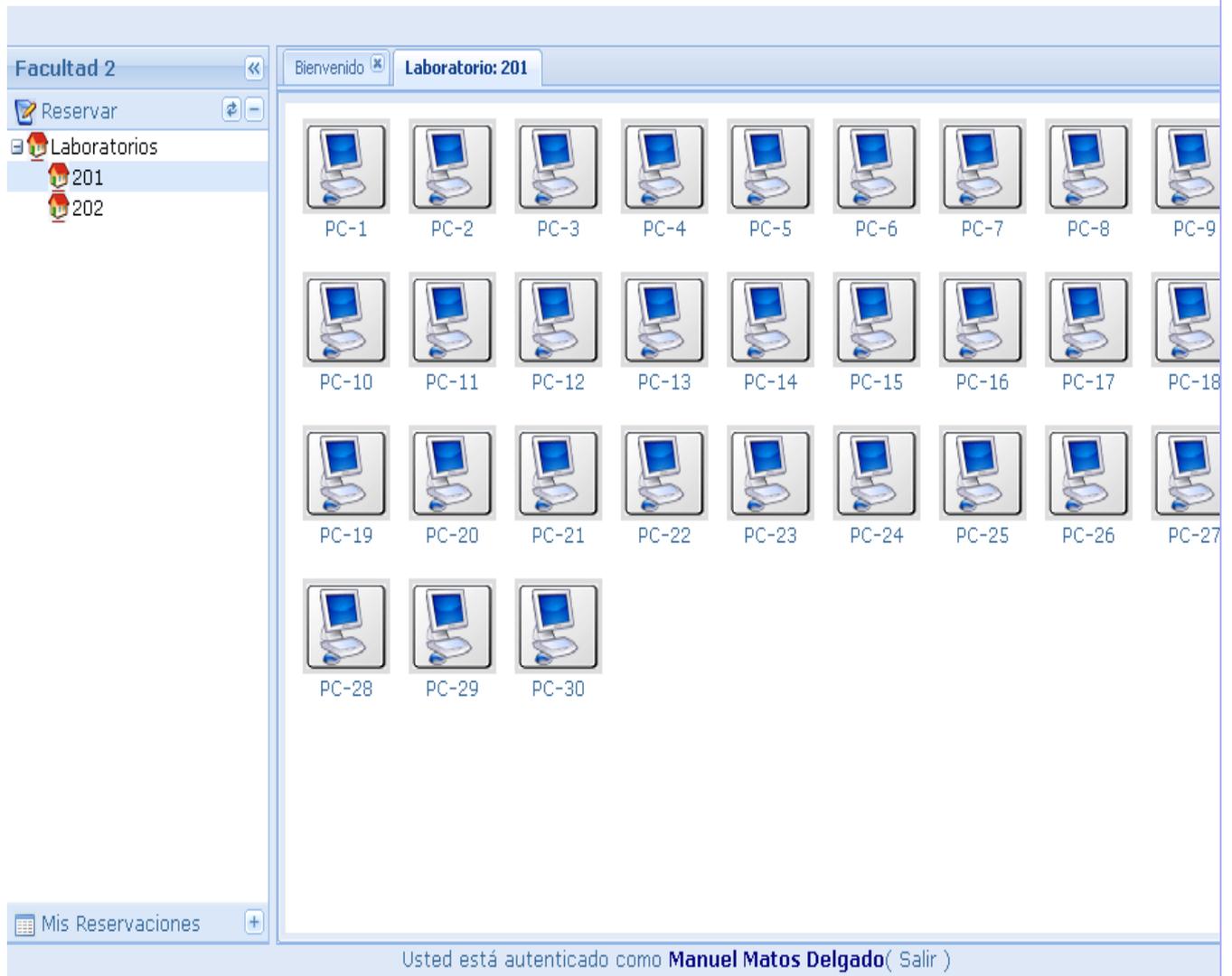
XperimentoS. *Los mejores 12 Frameworks JavaScript.* [Online] [Cited: Noviembre 25, 2007.] <http://www.xperimentos.com/2007/09/04/los-mejores-12-frameworks-javascript/>.

ANEXOS

Anexo-1 Descripción textual de los casos de uso del sistema

Caso de Uso:	Gestionar Reservación	
Actores:	Usuario-Docencia (inicia)	
Resumen:	El caso de uso se inicia cuando el usuario después de estar autenticado necesita realizar una nueva reservación; modificar o eliminar una ya existente.	
Precondiciones:	El usuario debe tener una cuenta en el dominio UCI y estar autenticado en la aplicación.	
Referencias	R1, R2, R3	
Prioridad	Primario	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1. Este caso de uso se inicia cuando el usuario solicita el servicio de reservación	2. Verifica el usuario y muestra el recurso a Gestionar (En caso de ser un estudiante los recursos que se le muestran son las computadoras correspondientes a un laboratorio)	
3. Selecciona el recurso a gestionar.	4. Muestra la opción a) Reservar, consúltese la sección Reservar (Si el usuario no ha	

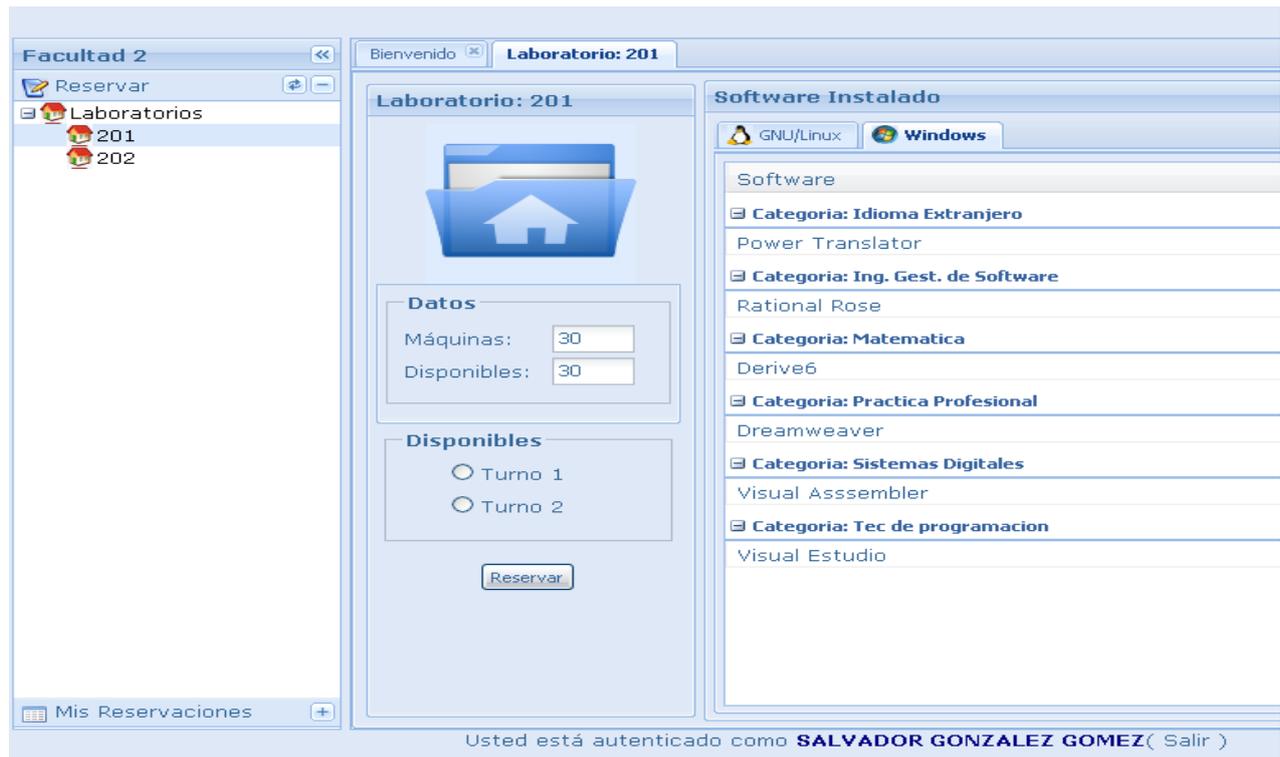
hecho reservación previamente)

Prototipo de Interfaz

Flujo Alterno
Acción del Actor
Respuesta del Sistema

2.1 Verifica el usuario y muestra el recurso

a Gestionar (En caso de ser un profesor el recurso que se le muestra es el laboratorio con características de software generales).

Prototipo de Interfaz



Flujo Alterno

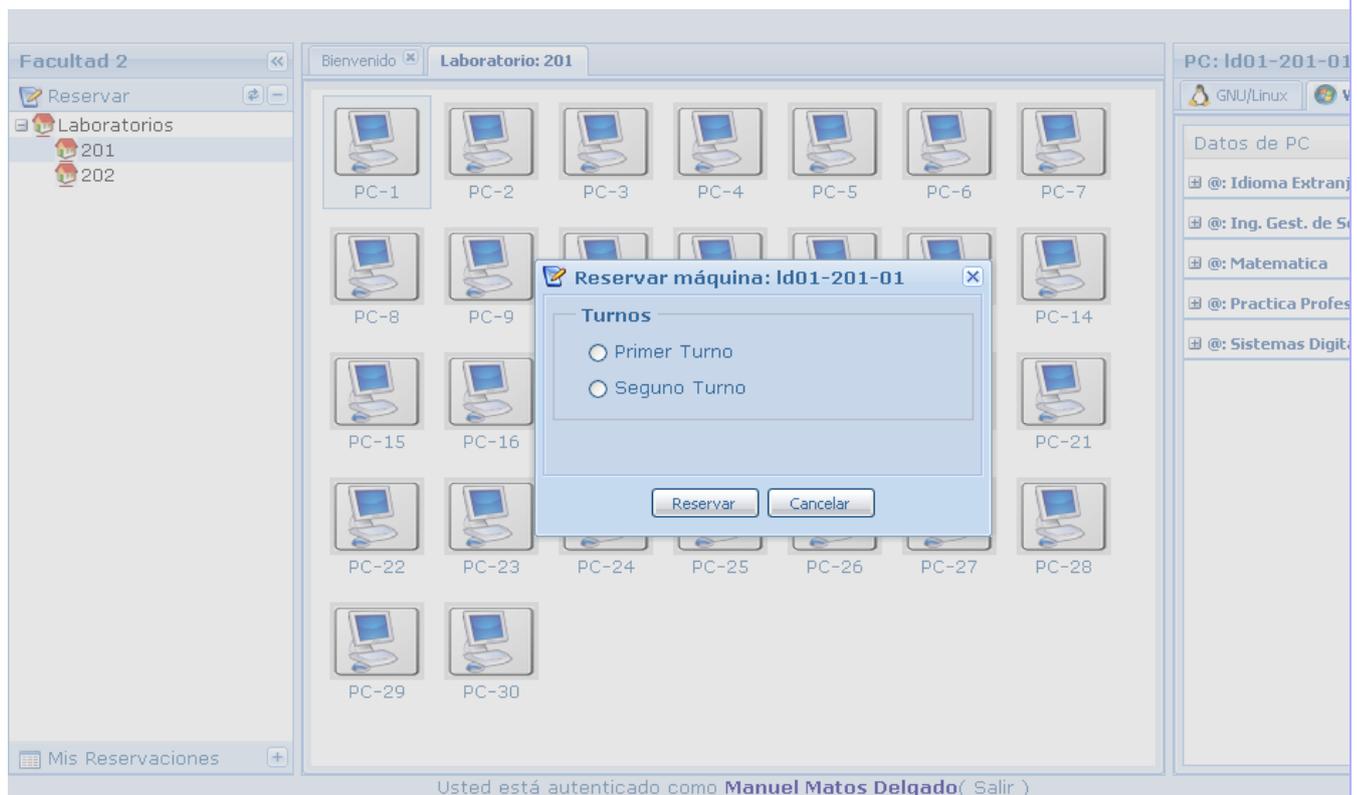
Acción del Actor	Respuesta del Sistema
	<p>4. 1 Muestra la opción(Si el usuario ya ha hecho la reservación previamente)</p> <ul style="list-style-type: none"> a) Modificar, consúltese la sección Modificar b) Cancelar, consúltese la sección

Cancelar

Sección Reservar

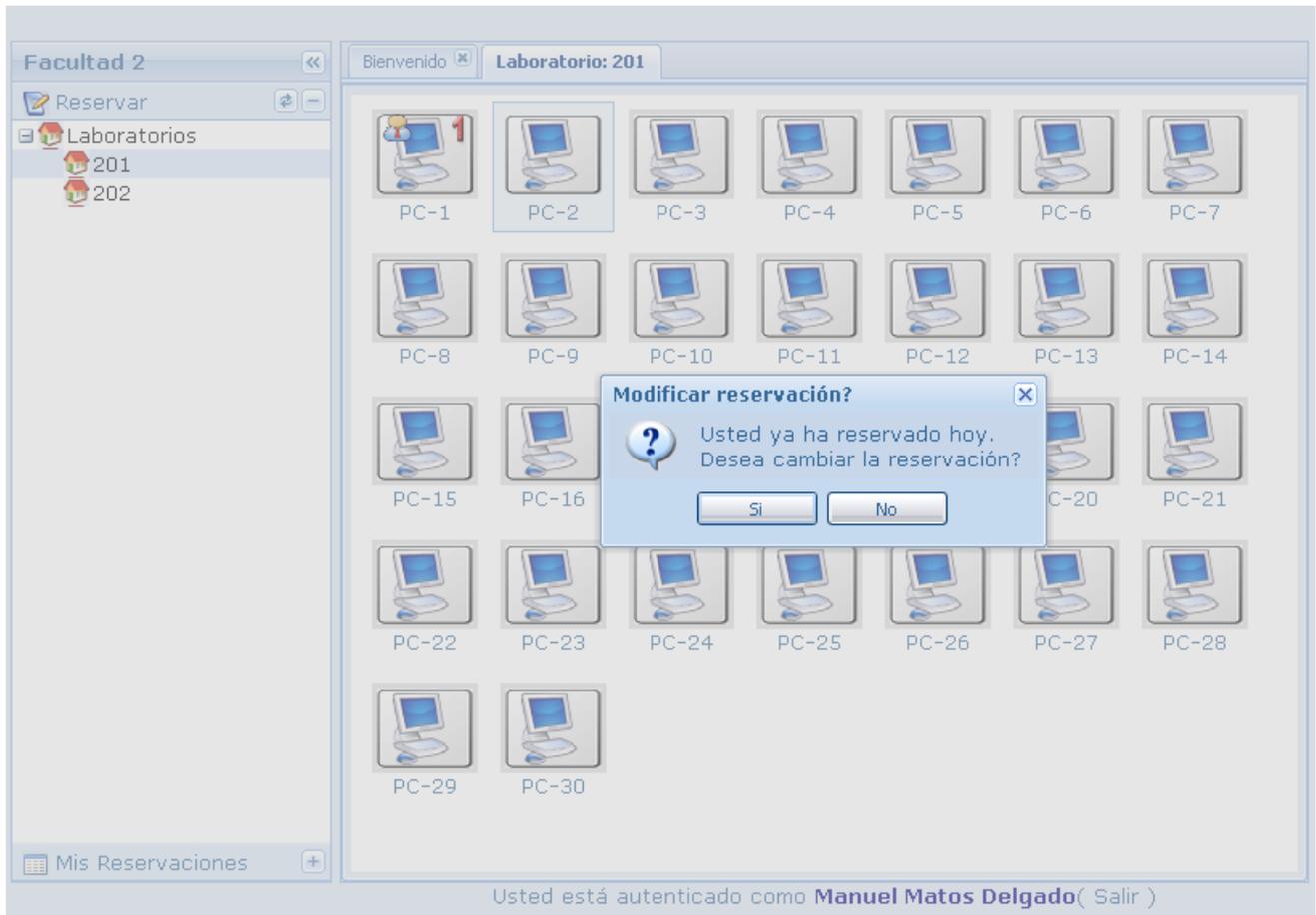
Acción del Actor	Respuesta del Sistema
5. Selecciona la opción reservar	6. Brinda los turnos disponibles para reservar.
7. Escoge turno de reservación.	8. Actualiza la base de datos 9. Genera mensaje de información. 10. Provee los datos de la reservación.

Prototipo de Interfaz



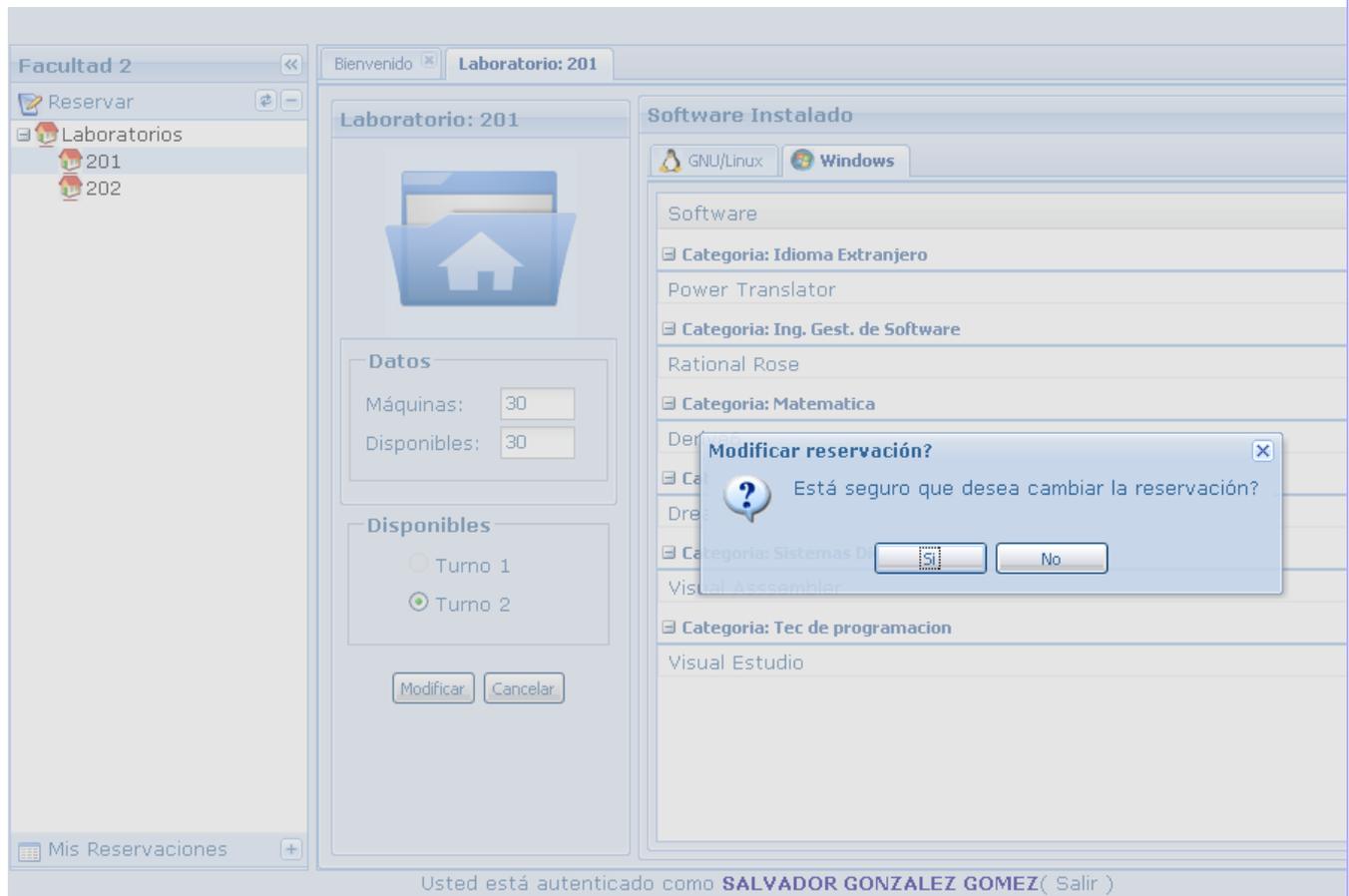
Flujo Alterno	
Acción del Actor	Respuesta del Sistema
7.1 Cancela reservación (este cancelar se refiere a no escoger turno de reservación)	
Sección Modificar	
Acción del Actor	Respuesta del Sistema
5. Selecciona la opción modificar	6. Muestra mensaje de confirmación
7. Acepta mensaje de confirmación	8. Brinda los turnos disponibles para modificar.
9. Escoge turno disponible	10. Actualiza base de datos. 11. Genera mensaje de información. 12. Provee los datos de modificación.

Prototipo de Interfaz



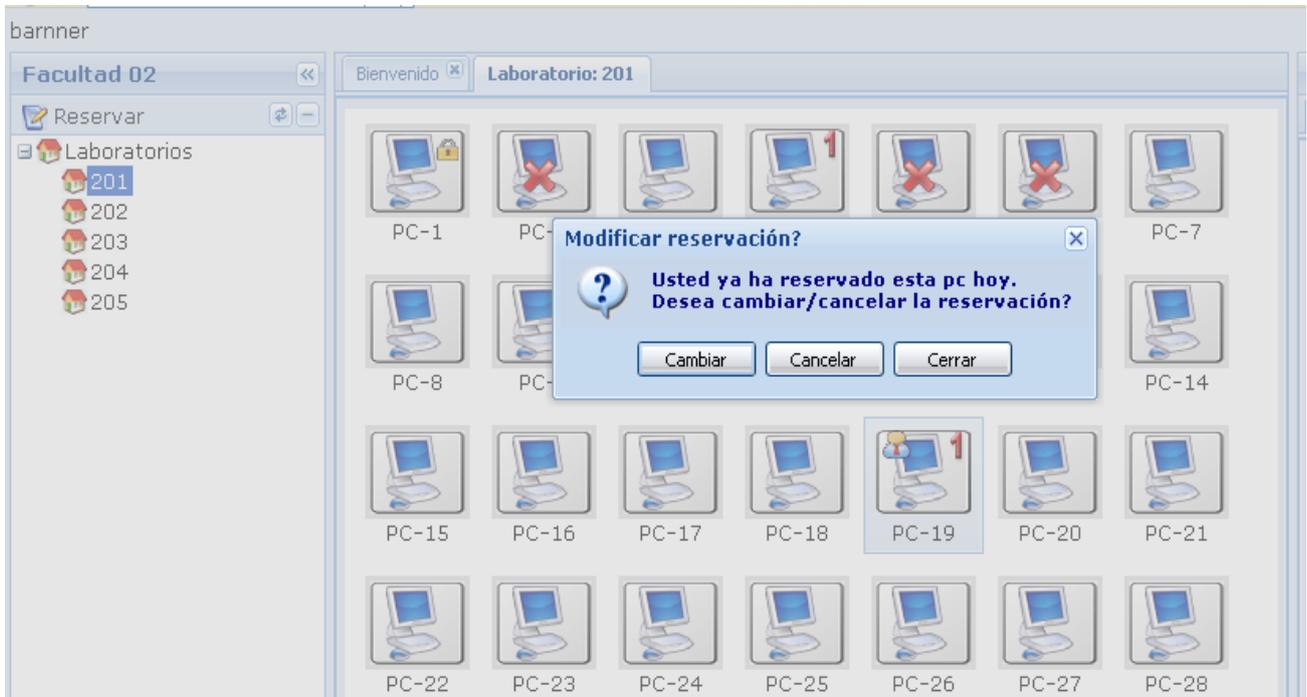
Flujo Alterno

Acción del Actor	Respuesta del Sistema
	6.1 Muestra mensaje de confirmación (en caso de ser profesor)
7.1. Niega mensaje de confirmación.	

Prototipo de Interfaz

Sección Cancelar

Acción del Actor	Respuesta del Sistema
5. Busca la reservación 6. Selecciona la opción Cancelar	7. Muestra mensaje de Confirmación
8. Acepta confirmación	9. Actualiza la base de datos. 10. Genera mensaje de información 11. Provee los datos cancelación.

Prototipo de Interfaz



Flujo Alternativo

Acción del Actor	Respuesta del Sistema
	7.1 Muestra mensaje de confirmación (en caso de ser profesor)
8.1. Niega confirmación	

Prototipo de interfaz

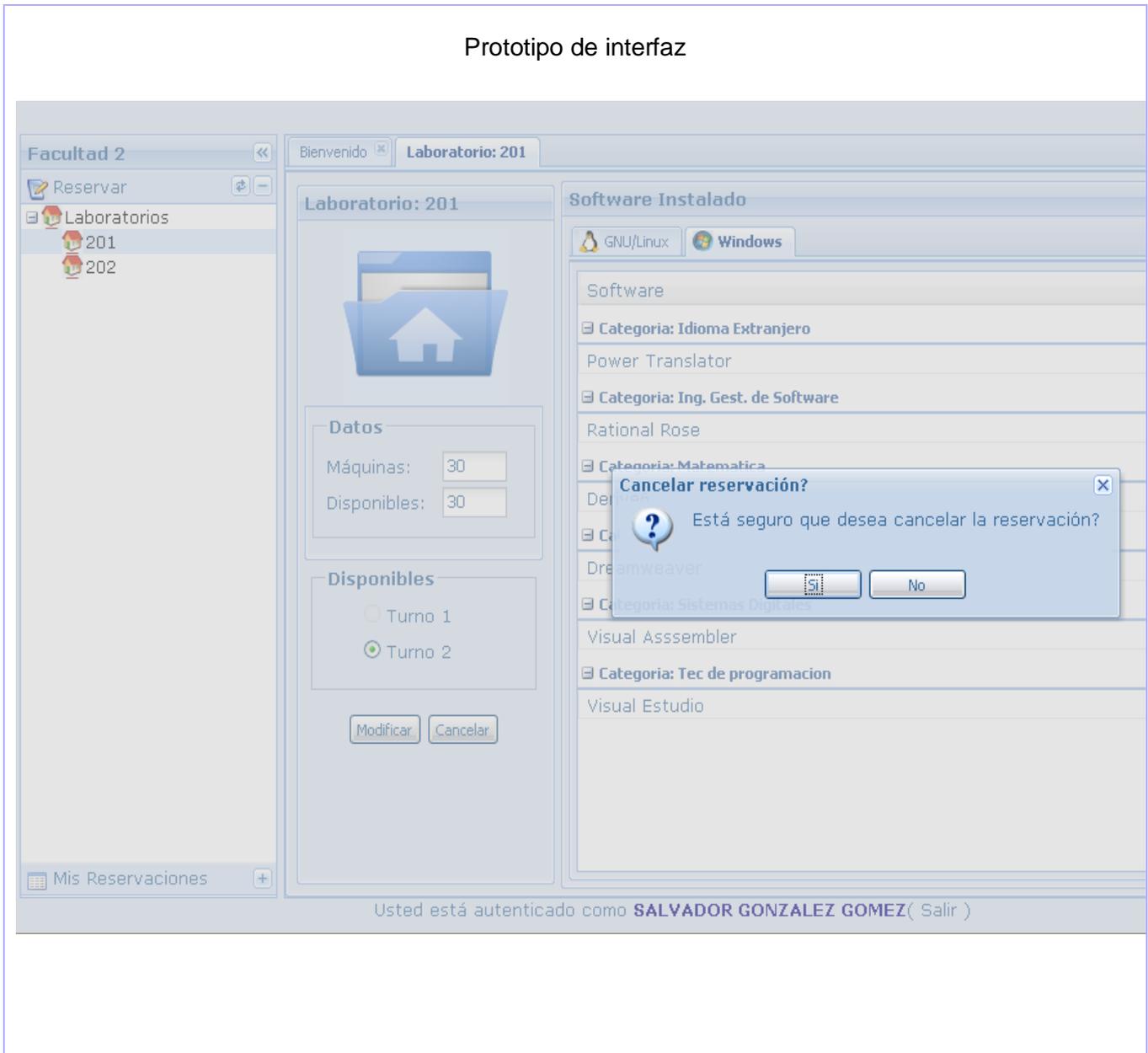
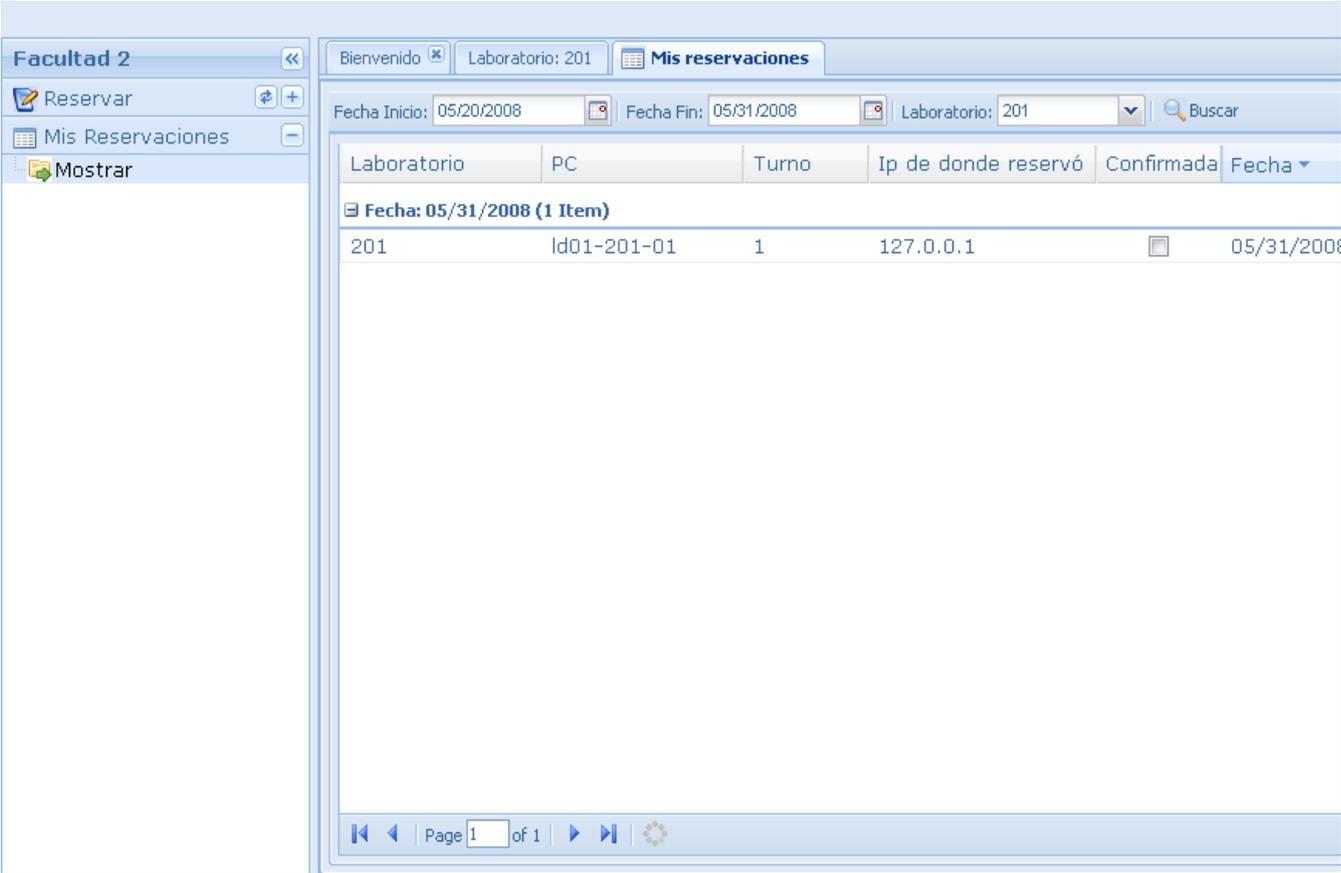


Tabla 7 Descripción textual del caso de uso del sistema gestionar reservación

Caso de Uso:	Obtener Reporte de Reservación	
Actores:	Usuario(Inicia)	
Resumen:	Este caso de uso se inicia cuando el usuario del sistema necesita realizar un estudio de las reservaciones realizadas anteriormente. El Usuario-Docencia solo accederá a sus reservaciones, en cambio el Usuario-Técnico accederá a las reservaciones correspondientes a su laboratorio.	
Precondiciones:	El usuario debe haber hecho al menos una reservación.	
Referencias	R4	
Prioridad	Primario	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1. Solicita el reporte de reservación	2. Brinda la opción de escoger período de tiempo	
3. Escoge período de tiempo 4. Acepta	5. Muestra los datos de las reservaciones correspondientes en dicho período de tiempo	

Prototipo de Interfaz



Facultad 2

Bienvenido Laboratorio: 201 Mis reservaciones

Reservar

Mis Reservaciones

Mostrar

Fecha Inicio: 05/20/2008 Fecha Fin: 05/31/2008 Laboratorio: 201 Buscar

Laboratorio	PC	Turno	Ip de donde reservó	Confirmada	Fecha
Fecha: 05/31/2008 (1 Item)					
201	ld01-201-01	1	127.0.0.1	<input type="checkbox"/>	05/31/2008

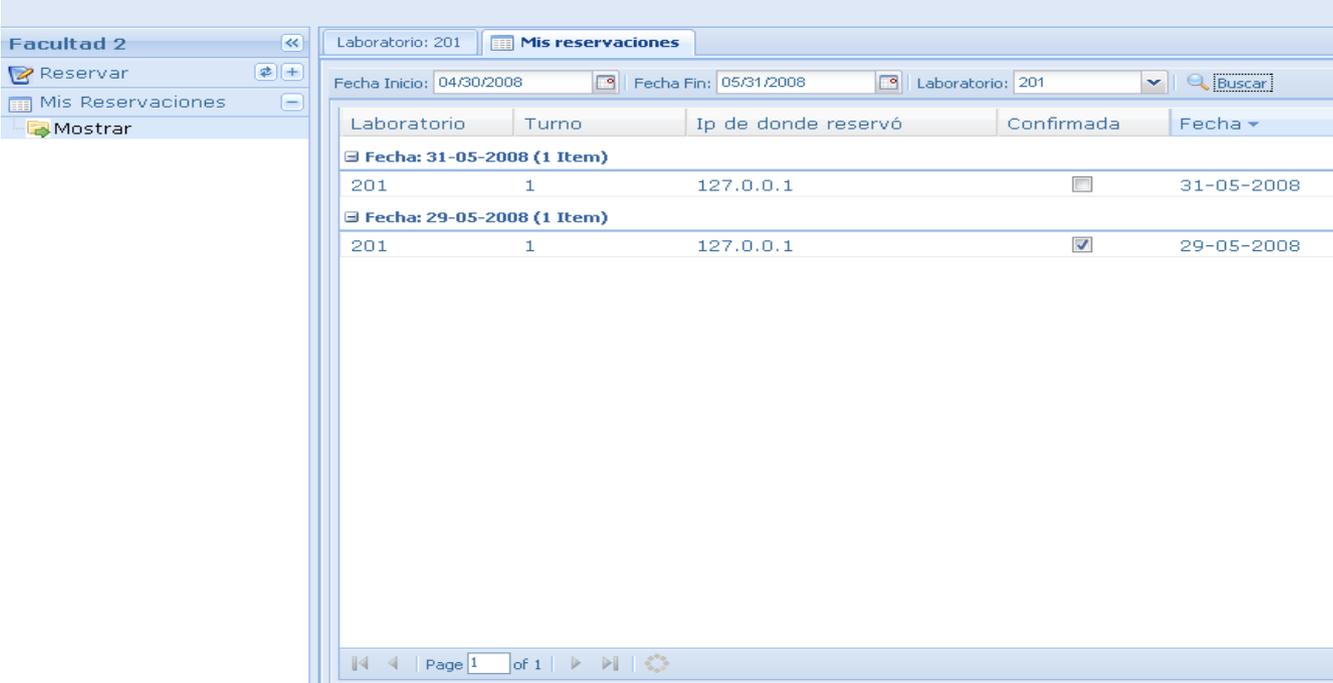
Page 1 of 1

Usted está autenticado como **Manuel Matos Delgado** (Salir)

Flujos Alternos

Acción del Actor	Respuesta del Sistema
	5.1 Muestra los datos de las reservaciones correspondiente en dicho período de tiempo(en caso de ser profesor)

Prototipo de Interfaz



Facultad 2

Reservar

Mis Reservasiones

Mostrar

Laboratorio: 201 **Mis reservasiones**

Fecha Inicio: 04/30/2008 Fecha Fin: 05/31/2008 Laboratorio: 201

Laboratorio	Turno	Ip de donde reservó	Confirmada	Fecha
Fecha: 31-05-2008 (1 Item)				
201	1	127.0.0.1	<input type="checkbox"/>	31-05-2008
Fecha: 29-05-2008 (1 Item)				
201	1	127.0.0.1	<input checked="" type="checkbox"/>	29-05-2008

Page 1 of 1

Usted está autenticado como **SALVADOR GONZALEZ GOMEZ** (Salir)

Flujos Alternos

Acción del Actor	Respuesta del Sistema
	5.2 Muestra los datos de las reservas correspondiente en dicho período de tiempo(en caso de ser técnico)

Prototipo de Interfaz

Area: Docente 1

- Reservaciones
- Controlar
- Busqueda

Bienvenido
Búsqueda

Fecha Inicio: 05/25/2008
Fecha Fin: 06/02/2008
Laboratorio: 201
Buscar

Usuario	Nombre	PC	Laboratorio	Turno	Ip Cliente	Co	Fecha
Fecha: 06/01/2008 (1 Elemento)							
ofumero	Orley Fumero P:	ld01-201-0	201	1	127.0.0.1	<input checked="" type="checkbox"/>	06/01/2008
Fecha: 05/31/2008 (2 Elementos)							
sgonzalezg	SALVADOR GON	Todas	201	1	127.0.0.1	<input type="checkbox"/>	05/31/2008
mmatos	Manuel Matos D	ld01-201-1:	201	2	127.0.0.1	<input type="checkbox"/>	05/31/2008
Fecha: 05/30/2008 (3 Elementos)							
cchaveco	Carlandis Chave	ld01-201-0:	201	1	10.8.15.106	<input type="checkbox"/>	05/30/2008
ofumero	Orley Fumero P:	ld01-201-0:	201	1	127.0.0.1	<input checked="" type="checkbox"/>	05/30/2008
yrdguez	Yaily Rodriguez	ld01-201-0	201	2	127.0.0.1	<input checked="" type="checkbox"/>	05/30/2008
Fecha: 05/29/2008 (1 Elemento)							
sgonzalezg	SALVADOR GON	Todas	201	1	127.0.0.1	<input checked="" type="checkbox"/>	05/29/2008

Page 1 of 2
Página 1

Usted está autenticado como **Carlos Rafael Gonzales Perdomo** (Salir)

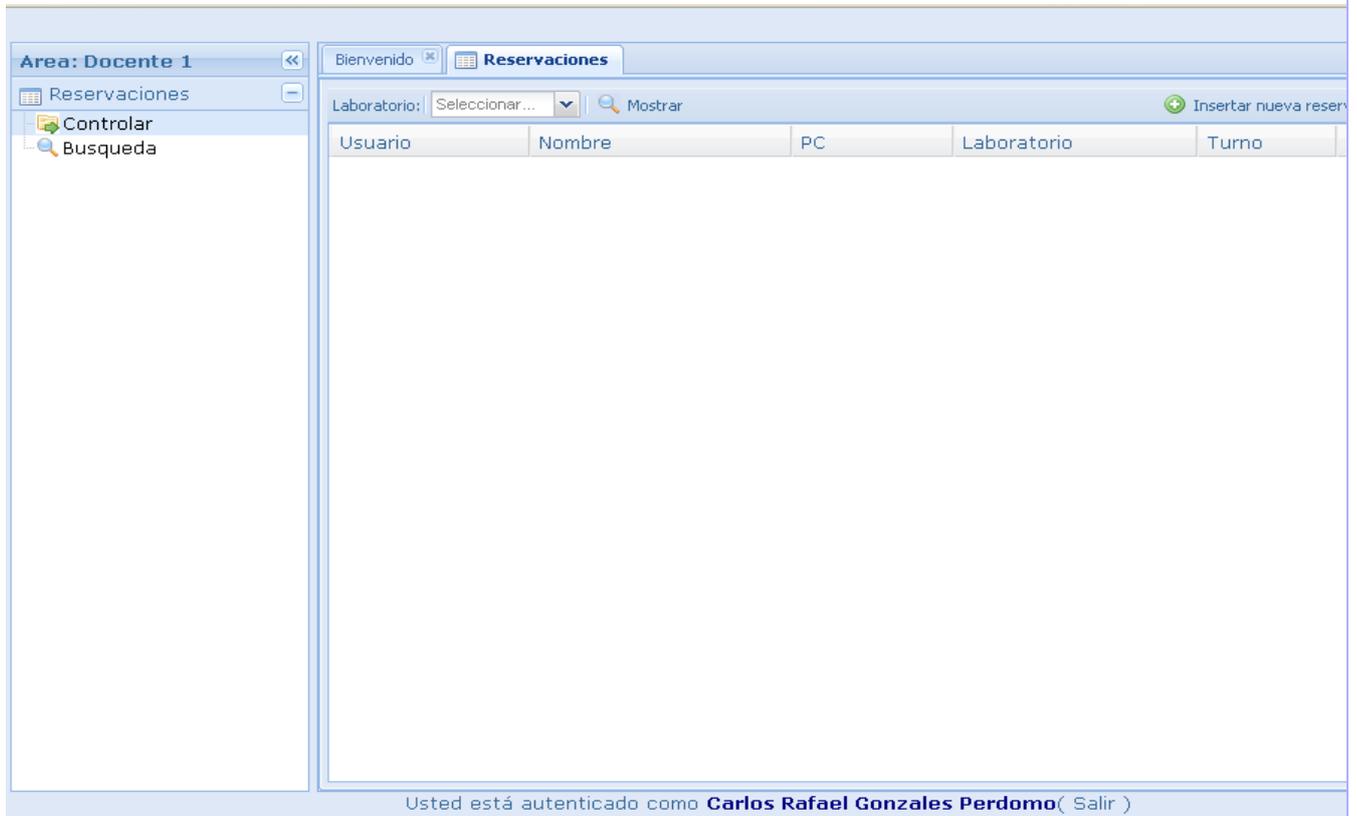
Tabla 8 Descripción textual del caso de uso del sistema obtener reporte de reservación

Caso de Uso:	Controlar Reservación
Actores:	Usuario-Técnico (Inicia)
Resumen:	El caso de uso se inicia cuando el técnico necesita controlar las reservaciones del laboratorio al que ha sido asignado. Validando la reservación cancelándola o modificándola.
Precondiciones:	El técnico debe estar autenticado
Referencias	R5, R6
Prioridad	Primario

Flujo Normal de Eventos

Acción del Actor	Respuesta del Sistema
1. Selecciona el laboratorio donde va a controlar la reservación.	2. El sistema muestra los puestos del laboratorio con el estado.
3. Selecciona la reservación	4. El sistema muestra las operaciones para controlar la reservación. a) Validar (ir a sección Validar Reservación) b) Modificar (ir a sección Modificar Reservación)

Prototipo de Interfaz



Area: Docente 1

Bienvenido | Reservaciones

Laboratorio: Seleccionar... | Mostrar | Insertar nueva reservación

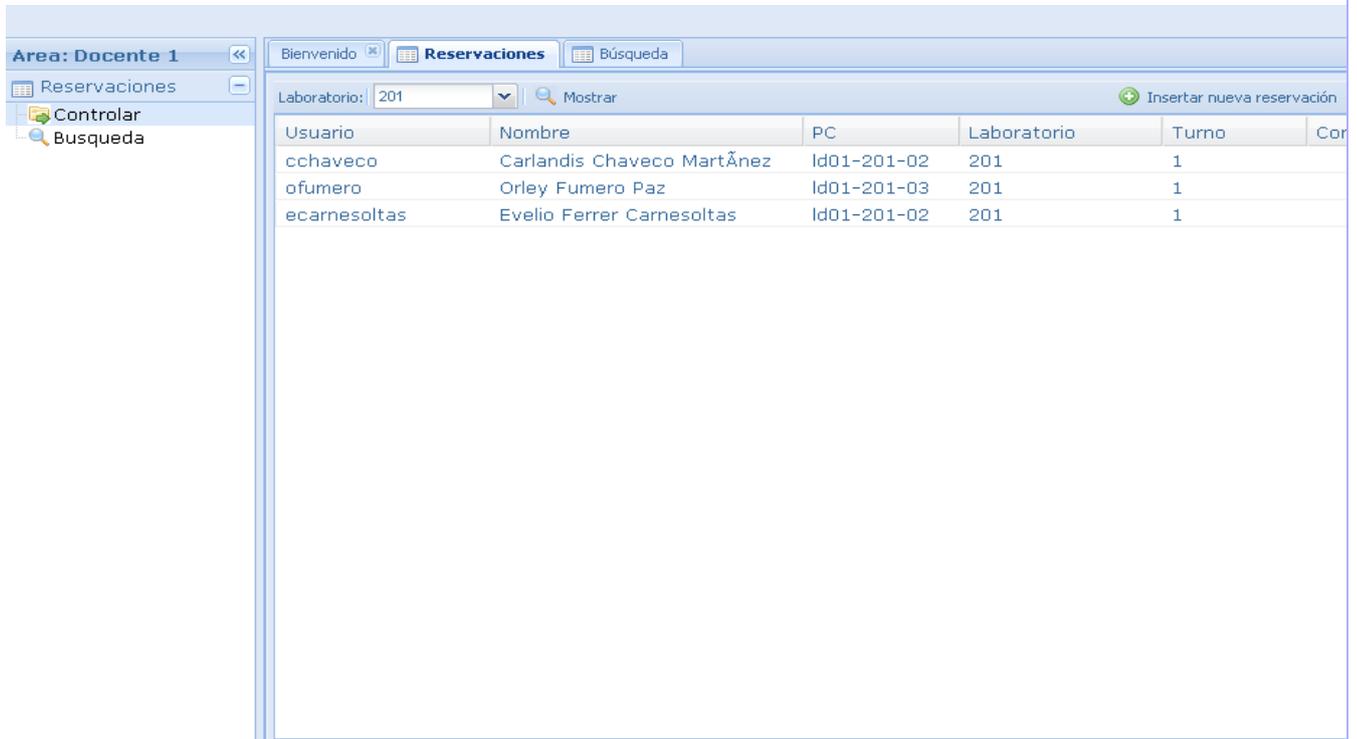
Usuario	Nombre	PC	Laboratorio	Turno

Usted está autenticado como **Carlos Rafael Gonzales Perdomo** (Salir)

Sección Validar Reservación

Acción del Actor	Respuesta del Sistema
5. El técnico selecciona la opción controlar	6. El sistema muestra campo de búsqueda según laboratorio
7. El técnico realiza búsqueda	8. El sistema muestra resultados de la búsqueda.
9. Confirma reservación	

Prototipo de Interfaz



Usuario	Nombre	PC	Laboratorio	Turno	Cor
cchaveco	Carlandis Chaveco Martínez	Id01-201-02	201	1	
ofumero	Orley Fumero Paz	Id01-201-03	201	1	
ecarnesoltas	Evelio Ferrer Carnesoltas	Id01-201-02	201	1	

Sección Modificar Reservación

Acción del Actor	Respuesta del Sistema
5. Selecciona la opción de insertar nueva reservación.	6. El sistema muestra un formulario
7. Llena formulario y acepta	8. Actualiza la Base de Datos 9. Genera un mensaje de información

Prototipo de Interfaz

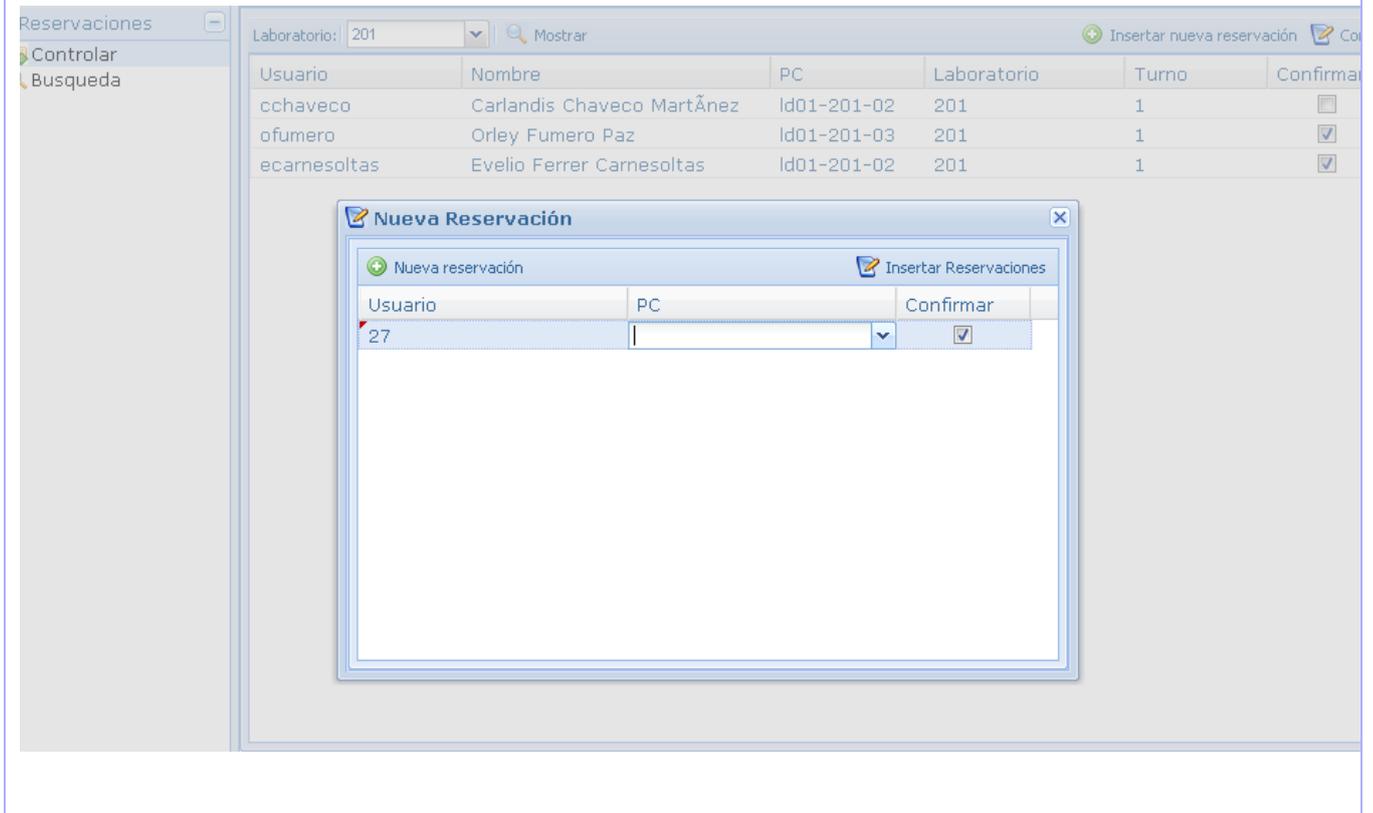
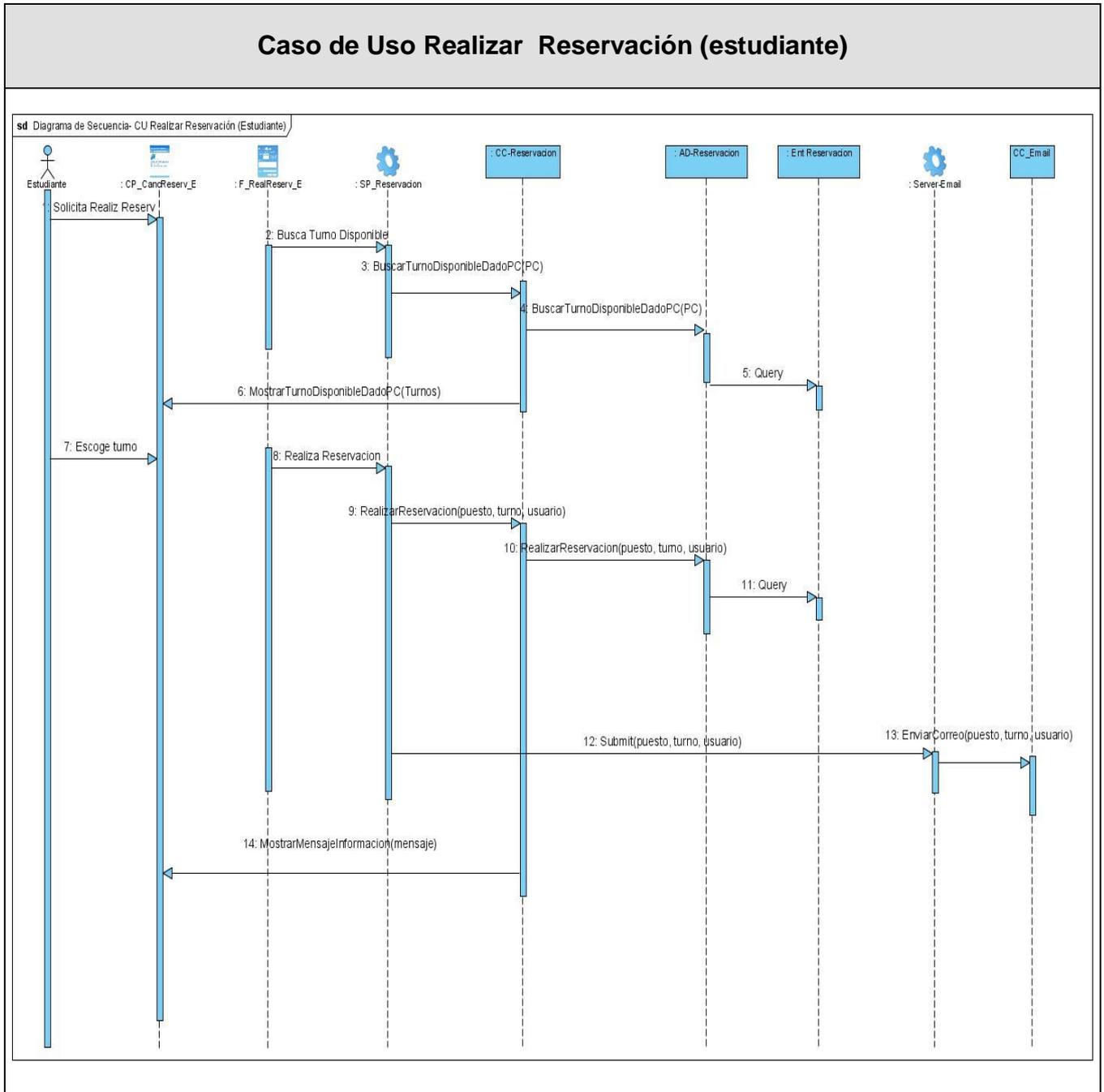


Tabla 9 Descripción textual del caso de uso del sistema controlar reservación

Anexo_2 Diagramas de secuencia del diseño.

Figura 29 Diagrama de secuencia del caso de uso realizar reservación (estudiante)

Caso de Uso Modificar Reservación (estudiante)

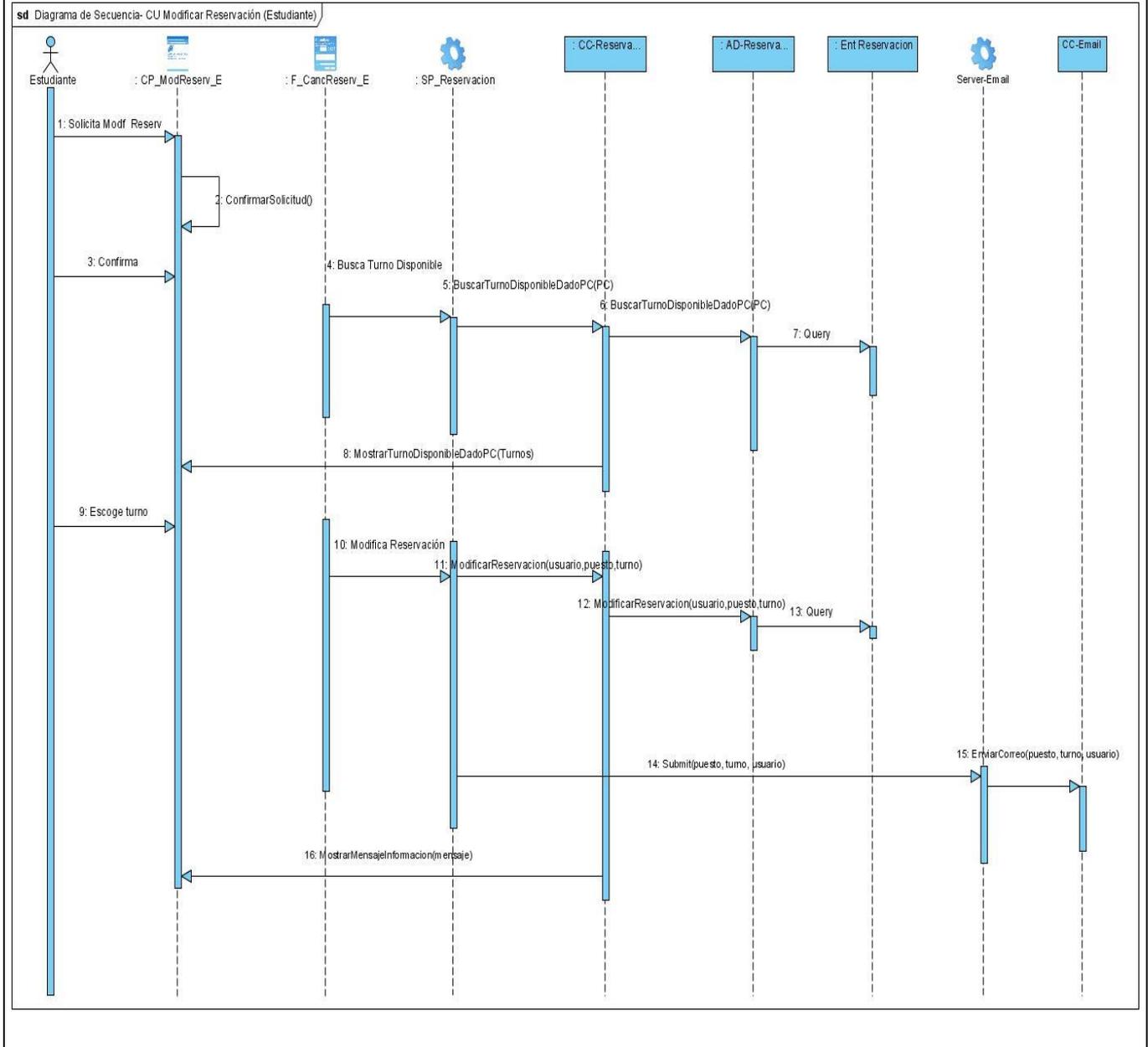


Figura 30 Diagrama de secuencia del caso de uso modificar reservación (estudiante)

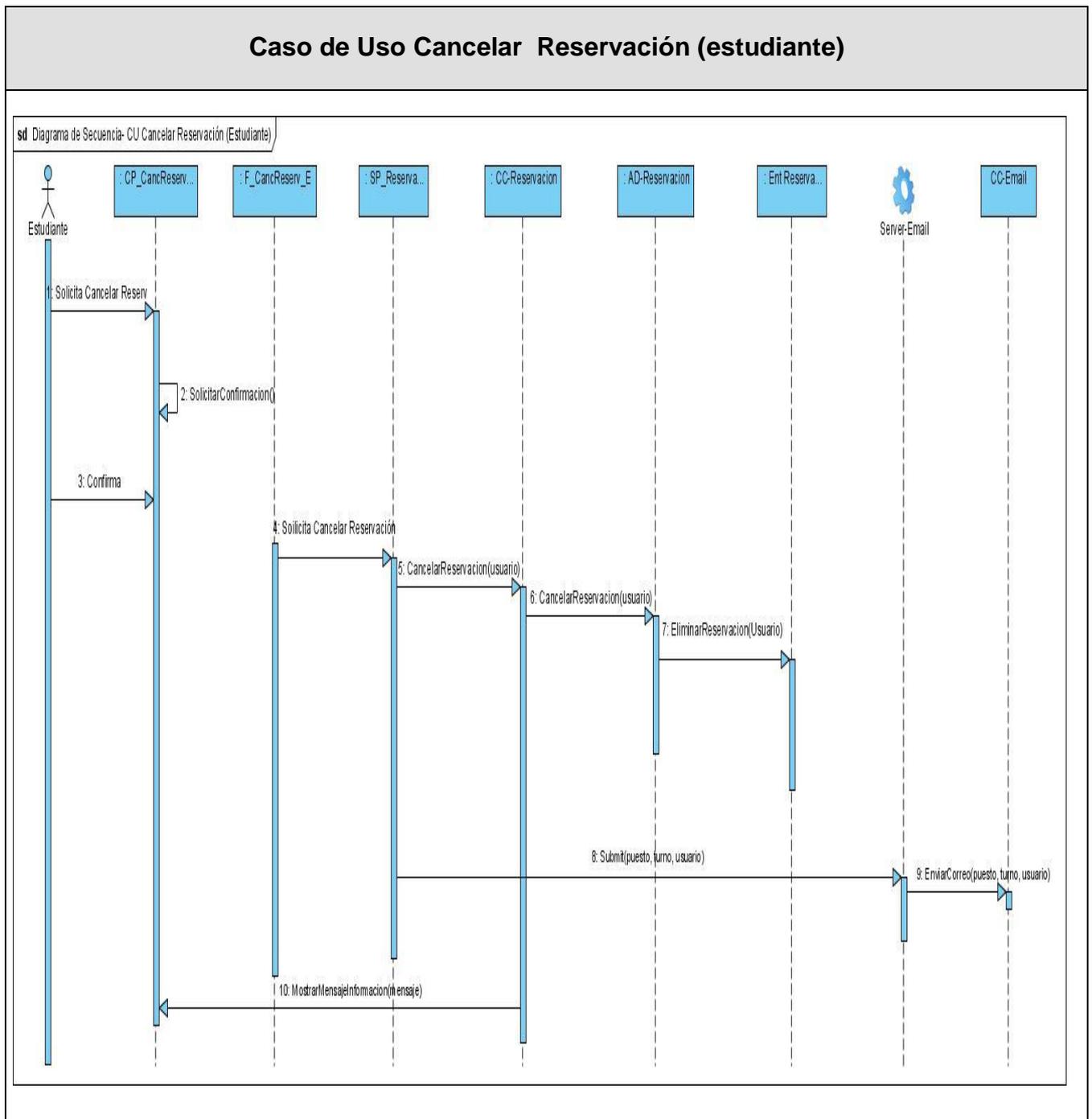


Figura 31 Diagrama de secuencia del caso de uso cancelar reservación (estudiante)

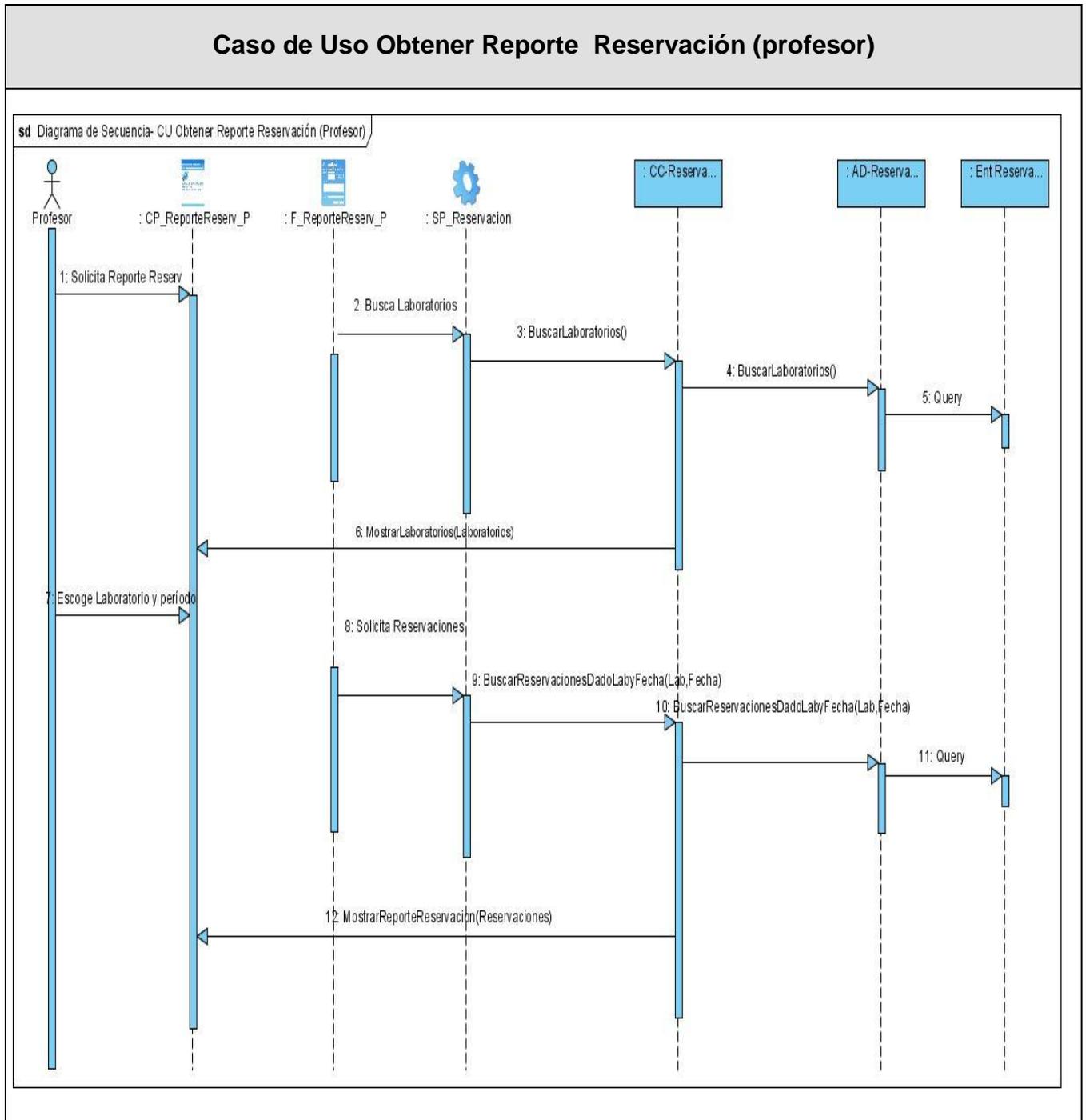


Figura 32 Diagrama de secuencia del caso de uso obtener reporte de reservación (profesor)

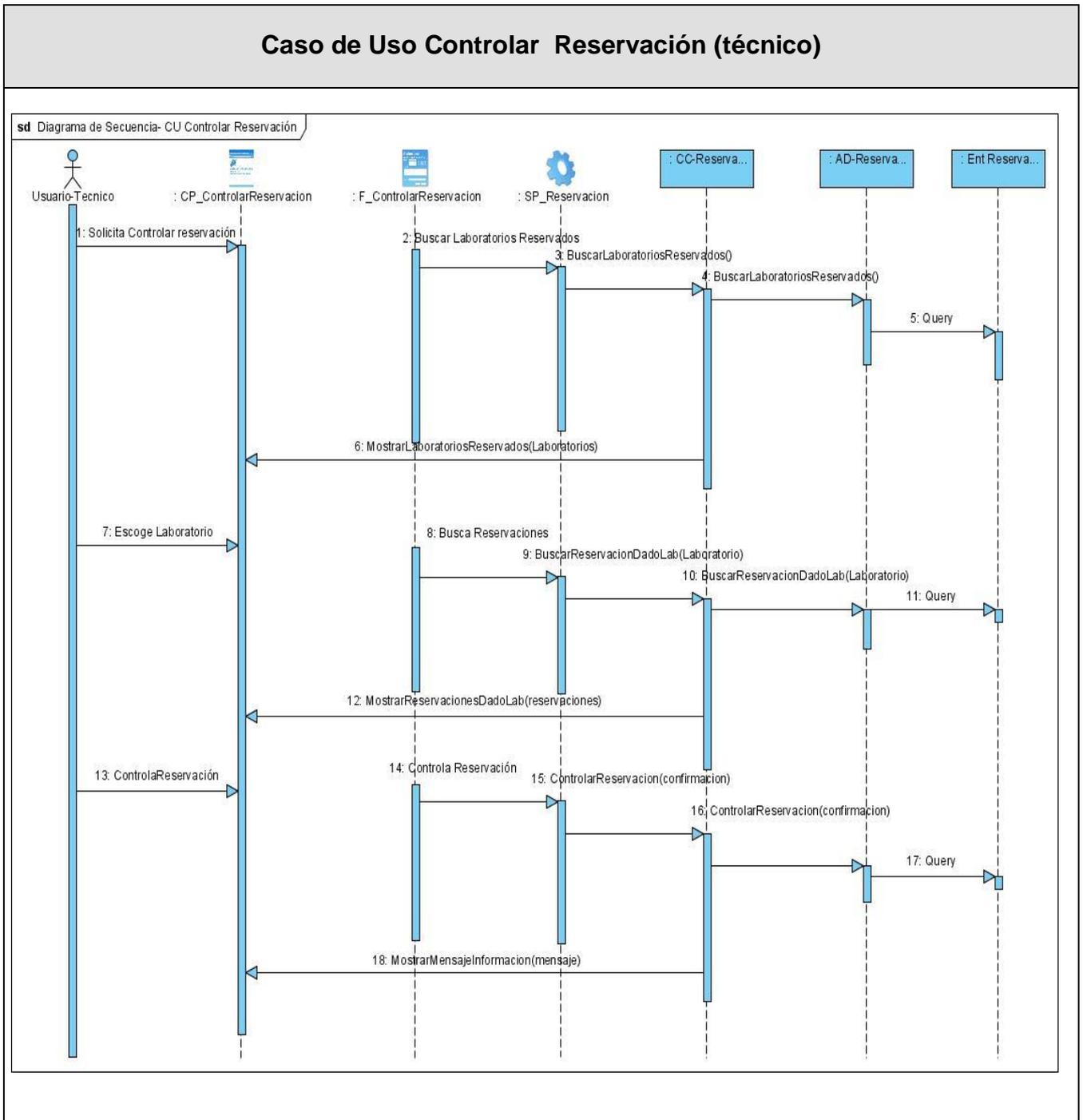


Figura 33 Diagrama de secuencia del caso de uso controlar reservación (técnico)

Anexo_3 Descripción de las tablas del modelo de datos

Nombre: Reservación		
Descripción: Recoge los datos de las reservaciones realizadas		
Atributo	Tipo	Descripción
idTurno	int(10)	Identificador de la tabla Turno
idReservacion	int(10)	Identificador de la tabla Reservacion
idUsuario	int(10)	Identificador de la tabla Usuario
idPuesto	int(10)	Identificador de la tabla Puesto
idLaboratorio	int(10)	Identificador de la tabla Laboratorio
fecha	date	Fecha en que se realiza la reservación
confirmada	bit	Guarda el control de la reservación
ipCliente	varchar(32)	Dirección ip de la PC desde la cual se hizo la reservación.

Tabla 10 Descripción de la tabla reservación de la base de datos

Nombre: Usuario		
Descripción: Recoge los datos de los usuarios que gestionan una reservación		
Atributo	Tipo	Descripción
idUsuario	int(10)	Identificador de la tabla Usuario
idRol	int(10)	Identificador de la tabla Rol
usuario	varchar(32)	Usuario del dominio que gestiona la reservación
correo	varchar(32)	Correo del usuario que gestiona la reservación
nombre	varchar(32)	Nombre del usuario que gestiona la reservación
solapín	int(10)	Solapín del usuario que gestiona la reservación
idFacultad	int(10)	Identificador del tabla Facultad

Tabla 11 Descripción de la tabla Usuario de la base de datos

Nombre: Rol		
Descripción: Recoge datos del rol que tiene el usuario dentro de la aplicación		
Atributo	Tipo	Descripción
idRol	int(10)	Identificador de la tabla Rol
tipoRol	varchar(32)	Rol que tiene un usuario dentro de la aplicación

Tabla 12 Descripción de la tabla Rol de la base de datos

Nombre: Turno		
Descripción: Recoge datos del turno que define cada facultad		
Atributo	Tipo	Descripción
idTurno	int(10)	Identificador de la tabla Turno
numeroTurno	int(10)	Numero de turno

Tabla 13 Descripción de la tabla Turno de la base de datos

Nombre: Facultad		
Descripción: Recoge datos de la Facultad		
Atributo	Tipo	Descripción
idFacultad	int(10)	Identificador de la tabla Facultad
idArea	int(10)	Identificador de la tabla Area
numeroFacultad	int(10)	Numero de facultad

Tabla 14 Descripción de la tabla Facultad de la base de datos

Nombre: Turno-Facultad		
Descripción: Esta es la tabla que se genera de la relación que existe entre Turno y Facultad		
Atributo	Tipo	Descripción
idTurno	int(10)	Identificador de la tabla Turno
idFacultad	int(10)	Identificador de la tabla Facultad
horaInicio	time(7)	Hora de inicio de un turno
horaFin	time(7)	Hora fin de un turno

Tabla 15 Descripción de la tabla Turno-Facultad de la base de datos

Nombre: Area		
Descripción: Recoge datos de la tabla Area, a la cual pertenece una facultad		
Atributo	Tipo	Descripción
idArea	int(10)	Identificador de la tabla Area
nombreArea	varchar(32)	Nombre de la facultad

Tabla 16 Descripción de la tabla Area de la base de datos

Nombre: Puesto		
Descripción: Recoge datos de la tabla Puesto(PC)		
Atributo	Tipo	Descripción
idPuesto	int(10)	Identificador de la tabla Puesto
nombrePuesto	int(10)	Nombre del puesto de trabajo
estado	varchar(32)	Estado del puesto de trabajo

Tabla 17 Descripción de la tabla Puesto de la base de datos

Nombre: Laboratorio		
Descripción: Recoge datos de la tabla Laboratorio		
Atributo	Tipo	Descripción
idLaboratorio	int(10)	Identificador de la tabla Facultad
nombreLaboratorio	varchar(32)	Nombre del laboratorio
tipoLaboratorio	varchar(32)	Tipo de laboratorio, si es de producción o de docencia.

Tabla 18 Descripción de la tabla Laboratorio de la base de datos

GLOSARIO DE TÉRMINOS.

Bugs: Es el resultado de un fallo o deficiencia durante el proceso de creación de programas de ordenador o computadora (software). Dicho fallo puede presentarse en cualquiera de las etapas del ciclo de vida del software aunque los más evidentes se dan en la etapa de desarrollo y programación.

CASE: Ingeniería de Software Asistida por Computadoras .Una herramienta CASE tiene como objetivo principal conseguir la generación automática de programa desde una especificación a nivel de diseño.

Cliente: El cliente es una aplicación informática que se utiliza para acceder a los servicios que ofrece un servidor.

DOM: El Modelo de Objetos del Documento (DOM) es una interfaz de programación de aplicaciones (API) para documentos HTML y XML. Con el Modelo de Objetos del Documento los programadores pueden construir documentos, navegar por su estructura, y añadir, modificar o eliminar elementos y contenido. Se puede acceder a cualquier cosa que se encuentre en un documento HTML o XML, y se puede modificar, eliminar o añadir usando el Modelo de Objetos del Documento, salvo algunas excepciones.

IDE: Entorno de desarrollo integrado o en inglés Integrated Development Environment ('IDE') es un programa compuesto por un conjunto de herramientas para un programador.

Licencia BSD: Es la licencia de software otorgada principalmente para los sistemas BSD (*Berkeley Software Distribution*). Pertenece al grupo de licencias de software Libre. Esta licencia tiene menos restricciones en comparación con otras como la GPL estando muy cercana al dominio público. La licencia BSD al contrario que la GPL permite el uso del código fuente en software no libre.

LOG: Registro oficial de eventos durante un período de tiempo en particular.

Navegador web: Un navegador web o explorador web es una aplicación software que permite al usuario recuperar y visualizar documentos de hipertexto, comúnmente descritos en HTML, desde servidores web de todo el mundo a través de Internet

Puig in: Es una aplicación informática que interactúa con otra aplicación para aportarle una función o utilidad específica.

Servidor en informática o computación: Es una aplicación informática o programa que realiza algunas tareas en beneficio de otras.

SIGLA: Sistema Integral de Gestión de los Laboratorios.

UCI: Universidad de las Ciencias Informáticas.

XSLT: Es un estándar que presenta una forma de transformar documentos XML en otros e incluso a formatos que no son XML.