



UNIVERSIDAD TÉCNICA PARTICULAR DE LOJA
La Universidad Católica de Loja

ÁREA TECNICA

TÍTULO DE INGENIERO EN INFORMÁTICA

Desarrollo de una aplicación web para gestionar el proceso de préstamo de instalaciones del área de deportes, recreación y cultura de la Universidad Técnica Particular de Loja.

TRABAJO DE TITULACIÓN.

AUTOR: Castillo Ojeda, Pablo Fernando

DIRECTORA: Condolo Herrera, Lorena del Cisne, Ing.

CENTRO UNIVERSITARIO QUITO

2017



Esta versión digital, ha sido acreditada bajo la licencia Creative Commons 4.0, CC BY-NY-SA: Reconocimiento-No comercial-Compartir igual; la cual permite copiar, distribuir y comunicar públicamente la obra, mientras se reconozca la autoría original, no se utilice con fines comerciales y se permiten obras derivadas, siempre que mantenga la misma licencia al ser divulgada. <http://creativecommons.org/licenses/by-nc-sa/4.0/deed.es>

2017

APROBACIÓN DEL DIRECTOR DEL TRABAJO DE TITULACIÓN

Ingeniera.

Lorena del Cisne Condolo Herrera.

DOCENTE DE LA TITULACIÓN

De mi consideración:

El presente trabajo de titulación: **Desarrollo de una aplicación web para gestionar el proceso de préstamo de instalaciones del área de deportes, recreación y cultura de la Universidad Técnica Particular de Loja**, realizado por: **Castillo Ojeda Pablo Fernando** ha sido orientado y revisado durante su ejecución, por cuanto se aprueba la presentación del mismo.

Loja, Septiembre de 2017

f)

DECLARACIÓN DE AUTORÍA Y CESIÓN DE DERECHOS

"Yo **Castillo Ojeda Pablo Fernando**, declaro ser autor (a) del presente trabajo de titulación: **Desarrollo de una aplicación web para gestionar el proceso de préstamo de instalaciones del área de deportes, recreación y cultura de la Universidad Técnica Particular de Loja**", de la Titulación Sistemas Informáticos y Computación, siendo **Lorena Condolo** director (a) del presente trabajo; y eximo expresamente a la Universidad Técnica Particular de Loja y a sus representantes legales de posibles reclamos o acciones legales. Además certifico que las ideas, conceptos, procedimientos y resultados vertidos en el presente trabajo investigativo, son de mi exclusiva responsabilidad.

Adicionalmente declaro conocer y aceptar la disposición del Art. 88 del Estatuto Orgánico de la Universidad Técnica Particular de Loja que en su parte pertinente textualmente dice: "Forman parte del patrimonio de la Universidad la propiedad intelectual de investigaciones, trabajos científicos o técnicos y tesis de grado que se realicen a través, o con el apoyo financiero, académico o institucional (operativo) de la Universidad"

f.....

Autor: Castillo Ojeda Pablo Fernando.

Cédula: 171527658-8

DEDICATORIA

A **mi esposa Jessica** por su incomparable ayuda por estar en las buenas y en las malas, a **mis hijos Ezequiel y Saraí** que son mi inspiración, mi motor y mi empuje, a **mi madre** que siempre animándome. A mis hermanos y sobrinos y su motivación.

Pablo Castillo.

AGRADECIMIENTO

Agradezco a **Dios** por su fidelidad por darme las fuerzas para continuar adelante, a mi esposa que siempre en todo momento estuvo conmigo desde el inicio de este caminar, agradezco a **mis hijos** por comprender mis ausencias por mis estudios. A las personas que estuvieron para darme una mano cuando más lo necesitaba, sin importar el día o la hora.

Agradezco a la Ing. Lorena Condolo por su gran ayuda y su asesoría para poder culminar el presente trabajo.

Pablo Castillo.

ÍNDICE DE CONTENIDOS

CARATULA.....	i
APROBACIÓN DEL DIRECTOR DE TRABAJO DE TITULACIÓN.....	ii
DECLARACIÓN DE AUTORÍA Y CESIÓN DE DERECHOS.....	iii
DEDICATORIA.....	iv
AGRADECIMIENTO.....	v
INDICE DE CONTENIDOS.....	vi
INDICE DE FIGURAS.....	viii
INDICE DE TABLAS.....	ix
RESUMEN EJECUTIVO.....	1
ABSTRACT.....	2
INTRODUCCIÓN.....	3
OBJETIVOS.....	5
CAPITULO I: ESTADO DEL ARTE.....	6
1.1. Problemática.....	7
1.2. Justificación.....	7
1.3. Aplicaciones web.....	7
1.3.1. Aplicaciones web estáticas y dinámicas.....	8
1.3.2. Seguridad de la aplicación web.....	8
1.4. Lenguajes de desarrollo web.....	9
1.4.1. Java JSP.....	9
1.5. Base de datos.....	9
1.5.1. MySQL.....	11
1.6. Herramientas de soporte.....	12
1.6.1. Framework web.....	12
1.7. Servidor web: Apache.....	14
1.7.1. Arquitectura de apache.....	14
1.8. Entorno de desarrollo: Netbeans.....	16
1.9. Patrón Arquitectónico MVC.....	16
1.10. Metodología de desarrollo de la aplicación.....	17
1.10.1. Fases del ciclo de vida de RUP.....	18
1.10.1.1. Fase de inicio.....	18
1.10.1.2. Fase de elaboración.....	18
1.10.1.3. Fase de desarrollo.....	19
1.10.1.4. Fase de cierre.....	19
1.10.2. Elementos de RUP.....	19
CAPITULO II: DESARROLLO DE LA SOLUCIÓN.....	20
2.1. Módulos de la solución.....	21
2.2. Fase de inicio.....	21
2.2.1. Análisis del proceso actual.....	21
2.3. Fase de elaboración.....	23
2.3.1. Requerimientos funcionales.....	23
2.3.2. Requerimientos no funcionales.....	29
2.3.3. Casos de uso.....	30
2.4. Fase de desarrollo.....	42
2.4.1. Arquitectura de desarrollo MVC.....	42
2.4.1.1. Explicación del patrón MVC.....	44

2.4.2. Limitaciones.....	49
2.4.3. Base de datos.....	49
2.4.4. Herramientas utilizadas.....	50
2.4.5. Módulos.....	51
2.4.6. Interacción de procesos.....	52
2.5. Transición.....	55
2.5.1. Pruebas.....	55
CAPITULO III: PRUEBAS DE VALIDACIÓN.....	58
3.1. Tipos de pruebas.....	59
3.2. Ambiente de pruebas.....	59
3.3. Ejecución y resultado de las pruebas.....	60
3.2.1. Pruebas unitarias.....	60
CONCLUSIONES.....	68
RECOMENDACIONES.....	69
BIBLIOGRAFÍA.....	70
ANEXOS.....	73
ANEXO 1: ENTREVISTA PARA DETERMINAR REQUERIMIENTOS DE SISTEMA.....	74
ANEXO 2: DOCUMENTO VISIÓN.....	78
ANEXO 3: DOCUMENTO ARQUITECTURA.....	87
ANEXO 4: MANUAL DE PROGRAMADOR.....	101
ANEXO 5: MANUAL DE INSTALACIÓN.....	119
ANEXO 6: MANUAL DE USUARIO.....	137

ÍNDICE DE FIGURAS

Figura 1. Aplicaciones de gestión para MySql.....	12
Figura 2. Arquitectura servidor web apache.....	15
Figura 3. Patrón arquitectónico MVC.....	17
Figura 4. Esfuerzo en actividades según fase del proyecto.....	18
Figura 5. Modelo de negocios.....	22
Figura 6. Diagrama de casos de uso general.....	30
Figura 7: Caso de uso registrar usuario.....	31
Figura 8: Caso de uso registrar usuario por un administrador.....	31
Figura 9: Caso de uso resetear clave de usuario.....	22
Figura 10: Caso de uso ver lista de usuarios registrados.....	33
Figura 11: Caso de uso eliminar un usuario.....	33
Figura 12: Caso de uso revisar lista de canchas.....	34
Figura 13: Caso de uso revisar normativa de préstamo.....	35
Figura 14: Caso de uso comprobar disponibilidad de canchas.....	35
Figura 15: Caso de uso reservar una cancha.....	37
Figura 16: Caso de uso revisar reportes.....	38
Figura 17: Caso de uso guardar reportes.....	38
Figura 18: Caso de uso crear nueva cancha.....	39
Figura 19: Caso de uso renombrar una cancha.....	40
Figura 20: Caso de uso modificar capacidad de usuarios por cancha.....	41
Figura 21: Arquitectura de la aplicación (flujo de la información).....	43
Figura 22: Arquitectura de la aplicación.....	44
Figura 23: Archivos capa modelo.....	45
Figura 24: Archivos capa vista.....	46
Figura 25: Archivos capa controlador.....	46
Figura 26: Diagrama de flujo.....	47
Figura 27: Diagrama de clases.....	48
Figura 28: Variables y atributos de las tablas de la base de datos.....	50
Figura 29: Módulos del sistema.....	51
Figura 30: Modulo de verificación.....	53
Figura 31: Módulo de gestión y reservas.....	53
Figura 32: Módulo de reportes.....	54
Figura 33: Módulo de mantenimiento.....	55
Figura 34: Pruebas unitarias REC001.....	61
Figura 35: Resultado pruebas unitarias REC001.....	61
Figura 36: Pruebas unitarias REC002.....	62
Figura 37: Resultado pruebas unitarias REC002.....	62
Figura 38: Pruebas unitarias REC003.....	63
Figura 39: Resultado pruebas unitarias REC003.....	63
Figura 40: Pruebas unitarias REC008 y REC009.....	64
Figura 43: Resultado pruebas unitarias REC008 y REC009.....	65
Figura 44: Pruebas unitarias REC010.....	66
Figura 46: Resultado pruebas unitarias REC010.....	66

ÍNDICE DE TABLAS

Tabla 1. Módulos de la estructura del servidor Apache.....	15
Tabla 2. REQ001 Registro de usuario.....	24
Tabla 3. REQ002 Registro de usuarios por un administrador.....	24
Tabla 4. REQ003 Reseteo de clave de un usuario.....	25
Tabla 5. REQ004 Revisar usuarios registrados en la aplicación.....	25
Tabla 6. REQ005 Eliminar usuario.....	25
Tabla 7. REQ006 Revisar lista de canchas.....	26
Tabla 8: REQ007 Revisar normativas de préstamo de las canchas.....	26
Tabla 9: REQ008 Verificar disponibilidad de las canchas.....	26
Tabla 10: REQ009 Reservar una cancha.....	27
Tabla 11: REQ010 Revisar reportes.....	27
Tabla 12: REQ011 Guardar reportes.....	27
Tabla 13: REQ012 Imprimir reportes.....	28
Tabla 14: REQ013 Crear cancha.....	28
Tabla 15: REQ014 Modificar nombre de una cancha.....	28
Tabla 16: REQ015 Modificar capacidad de personas de una cancha.....	29
Tabla 17. CU001 Registro de usuario.....	31
Tabla 18. CU001 Registro de usuario por un administrador.....	32
Tabla 19. CU003 Reseteo de clave de un usuario.....	32
Tabla 20. CU004 Ver lista de usuarios registrados en la aplicación.....	33
Tabla 21. CU005 Eliminar usuario.....	34
Tabla 22. CU006 Revisar lista de canchas.....	34
Tabla 23: CU007 Revisar normativas de préstamo de las canchas.....	35
Tabla 24: CU008 Verificar disponibilidad de las canchas.....	36
Tabla 25: CU009 Reservar una cancha.....	37
Tabla 26: CU010 Revisar reportes.....	38
Tabla 27: CU011 Guardar reportes.....	39
Tabla 28: CU012 Crear nueva cancha.....	39
Tabla 29: CU013 Renombrar cancha.....	40
Tabla 30: CU014 Modificar capacidad de personas de una cancha.....	41
Tabla 31: Requerimientos hardware.....	49
Tabla 32: Tipos de pruebas.....	59
Tabla 33: Configuración del ambiente de pruebas y configuración de equipos.....	59
Tabla 34: Errores presentados.....	67

RESUMEN EJECUTIVO

El presente Trabajo de Titulación busca proveer una herramienta web que permita gestionar el proceso de préstamo de las instalaciones del Área de Deportes, Recreación y Cultura de la Universidad Técnica Particular de Loja, el mismo que permitirá y se constituirá en un instrumento que mejore el proceso de automatización de la disponibilidad, registro, asignación y control de las actividades dentro de las instalaciones a docentes, estudiantes, personal administrativo, etc. que hagan uso de las mismas. Para el desarrollo de esta aplicación se emplea la metodología de desarrollo de software RUP, respetando en cada fase con los entregables, para satisfacer las necesidades del cliente; se utiliza herramientas aplicadas y basadas principalmente en JAVA, por su gran versatilidad; con lo cual se podrá dar mayor cobertura a las necesidades requeridas por los usuarios que la manejen.

PALABRAS CLAVES: Aplicación web, Lenguaje JAVA, Base de datos, UPTL.

ABSTRACT

The present Work of Titulación looks for the tester a tool that allows to manage the process of loan of the facilities of the Area of Sports, Recreation and Culture of the Technical University Particular of Loja, the same that allows and it constitutes an instrument that improves the process of automation of the availability, registration, allocation and control of activities within facilities to teachers, students, personal administrative, etc. For the development of this application is used the methodology of RUP software development, respecting in each phase with the deliverables, to satisfy the needs of the client; we use tools applied and based mainly on JAVA, for its great versatility; with which it can give greater coverage to the necessities required by the users that handle it

KEY WORDS: Web application, JAVA language, Database, UPTL.

INTRODUCCIÓN

El presente trabajo tiene como finalidad implementar una aplicación web, que apoye el proceso de préstamo de las instalaciones deportivas y recreativas con las que cuenta actualmente la Universidad Técnica Particular de Loja, logrando eficiencia y control al momento de asignar o prestar las instalaciones a docentes, estudiantes y personal administrativo.

Para el desarrollo de la aplicación web, se utilizará la metodología RUP con herramientas principalmente programado en JAVA, motor de base de datos MySQL, servidor web (Apache 8.0) y el entorno de desarrollo de la aplicación es Netbeans. El desarrollo de la aplicación a través de la web permitirá a la misma integrarse al sistema de la universidad, con sus respectivas seguridades y de acuerdo al nivel de usuario correspondiente, sin embargo, la aplicación será utilizada en y para el campus principal en la ciudad de Loja.

Esta herramienta permitirá a cualquier usuario observar la disponibilidad de las instalaciones por áreas, fecha y hora en las que han sido reservadas, con lo cual se podrá asignar las áreas de forma ágil e imprimir la documentación de responsabilidad, presentar la información por fecha, por área de forma calendarizada, entre otros usos que tendrá la aplicación.

La información teórica, fortalecerá y argumentará el desarrollo de este trabajo de titulación, para organizar el proceso de préstamos de instalaciones del área de deportes, recreación y cultura de la Universidad de Técnica Particular de Loja. Y así poder desarrollar una aplicación utilizando herramientas web de condición Open Source como Java JSP como lenguaje de programación, MySQL como base de datos de la aplicación. Estas herramientas son económicas, son de fácil acceso, y poseen una amplia documentación que se apoya en una comunidad que evoluciona constantemente, estando de acuerdo a las necesidades del departamento de Misiones Universitarias.

La aplicación tendrá características como funcionalidad, confidencialidad, portabilidad entre otras que le permitirá cumplir con los objetivos propuestos. Con los atributos de calidad que posee la aplicación se llegara a cumplir los objetivos; el trabajo de la persona encargada en el área de asignación de áreas será planificada y los resultados de esta nueva implementación aportarán a la imagen de la universidad, a los estudiantes quienes deberán solicitar sus requerimientos para que las instalaciones se ocupen de acuerdo a la disponibilidad que existirá en cada una de las áreas.

En capítulo I tenemos el estado del arte en donde se encuentra la problemática y la justificación, así como las características y herramientas que se van a utilizar para el desarrollo de la aplicación. El capítulo II está el desarrollo de la aplicación centrada en la metodología RUP.

El capítulo III corresponde a las pruebas que se realizó al aplicativo, por ultimo están las conclusiones y recomendaciones del presente trabajo.

La solución desarrollada está hecha en Java JSP, con acceso a servicios web, utilizando el entorno de desarrollo Netbeans, con un servidor de Web apache 8.0, y MySQL como base de datos.

OBJETIVOS

Objetivo General

- Desarrollar una aplicación web para gestionar el préstamo de las instalaciones deportivas y recreativas para la Universidad Técnica Particular de Loja.

Objetivos Específicos

- Identificar los elementos teóricos y metodológicos relacionados con una aplicación web, para gestionar el proceso de préstamo de las instalaciones deportivas de la universidad.
- Desarrollar una herramienta web que sea de apoyo al momento de realizar la asignación o préstamo de las instalaciones, esta aplicación web ayudaría a los usuarios a conocer la disponibilidad en tiempo real.

**CAPÍTULO I.
ESTADO DEL ARTE**

1.1. Problemática.

En la actualidad no existe una aplicación que ayude al área de Misiones Universitarias de la Universidad Técnica Particular de Loja en el proceso de gestión de préstamo de instalaciones deportivas, este proceso se lo hace de manera semiautomática, esto genera problemas ya que no hay un registro donde se muestre el estado (si está reservada o no) de una instalación, se puede evidenciar lo siguiente:

- Los procesos de préstamo de instalaciones deportivas se podría realizar de manera más rápida, esto ayudará al departamento de Misiones Universitarias a mejorar la gestión y administración en el préstamo de las instalaciones.
- Al ser un proceso semiautomático, hace que sea un proceso lento.
- Se torna complejo el realizar un reporte calendarizado y un reporte de actividades en lo que se refiere a la disponibilidad de las instalaciones deportivas ya que es un proceso manual.
- Realizar cálculos estadísticos también es un proceso que se lo realiza manualmente y por lo tanto es demoroso ya que no se cuenta con la información actualizada.

En base a todo lo mencionado anteriormente se ve la necesidad de desarrollar una aplicación que ayude en la realización de todo estos procesos al área de misiones universitarias, docentes, estudiantes para hacer más eficiente el proceso de reserva.

1.2. Justificación.

Frente a la evidente necesidad de tener una herramienta que ayude al proceso de reserva y préstamo de las instalaciones deportivas con que cuenta la universidad, se realizó un análisis de la situación del departamento de misiones universitarias para analizar la creación de una aplicación que ayude en todo este proceso. Este análisis se centra en la entrevista realizada a personal del departamento de misiones universitarias, donde se pudo conocer las necesidades de contar con una herramienta para gestionar este proceso, posteriormente los datos obtenidos se transformaron en los requerimientos de la aplicación. Esta herramienta vendría a ser uno de los engranajes para el proceso de gestión y préstamo de las instalaciones deportivas.

Para el desarrollo de esta aplicación web se utilizara la metodología RUP, herramientas basadas principalmente en Java, esta aplicación será empleada en el campus de la ciudad de Loja.

1.3. Aplicaciones web.

Llamadas también webapps, es una categoría de software basado es redes y agrupa una amplia gama de aplicaciones, con la aparición de la web 2.0 estas webapps han evolucionado y han pasado de ofrecer características aisladas, y contenido para el usuario a integración

con bases de datos corporativas y aplicaciones de negocio. De acuerdo a (López Peralta, 2015) indica:

Una aplicación web es un conjunto de páginas que interactúan unas con otras y con diversos recursos en un servidor web, incluidas bases de datos. Esta interacción permite implementar características en su sitio como catálogos de productos virtuales y administradores de noticias y contenidos. Adicionalmente podrá realizar consultas a bases de datos, registrar e ingresar información, solicitudes, pedidos y múltiples tipos de información en línea en tiempo real.

De esta manera los datos en los que se trabaja son procesados y almacenados dentro de la web, es por esto que toda la información se guardará dentro de servidores de internet y se los podrá enviar a cualquier dispositivo o equipos al momento que se requiera; de esta forma se puede crear una copia temporal dentro del equipo.

1.3.1. Aplicaciones web estáticas y dinámicas.

Las aplicaciones web estáticas constan de un conjunto de páginas y archivos HTML, relacionados alojados en un equipo que ejecuta un servidor Web, el mismo que es un software que suministra páginas Web en respuesta a las peticiones de los navegadores; el cual solicita una página cuando hace se da clic en el vínculo que desea o introduce un URL. Con la aparición de la Web 2.0, las aplicaciones web fueron mejorando, y así fueron introduciéndose más las web con funcionamiento dinámico las cuales interactúan con el usuario.

1.3.2. Seguridad de la aplicación web.

La seguridad web es primordial en la aplicación al momento de mantener la institución protegida, ya que los ataques siempre están cambiando lo cual es peligroso y perjudicial ya que son difíciles de detectar. Sin embargo es uno de los problemas que más se presentan.

Existen organizaciones que definen políticas y buenas prácticas al momento de implementar la seguridad en las aplicaciones y sistemas informáticos. Por ejemplo debe haber la capacidad de proteger los datos sensibles y evitar que estos salgan de la institución.

Es importante identificar rápidamente los posibles ataques y a su vez corregirlos, ya que existen amenazas en la web sofisticadas proyectadas para ocultarse a simple vista en páginas auténticas que tiene mucho flujo de datos.

La aplicación desarrollada, cuenta con autenticación de usuarios (login) al momento de ingresar a la misma donde se solicita el nombre de usuario y contraseña, además tiene un proceso de autorización que restringe el acceso a los diferentes módulos dependiendo del rol asignado a un usuario.

1.4. Lenguaje para desarrollo web.

Han ido surgiendo según las necesidades de las plataformas, intentando facilitar el trabajo de los desarrolladores de aplicaciones, se clasifican en lenguajes del lado cliente y lenguajes del lado servidor. (Campo, 2015)

1.4.1. Java JSP

Castilla & León (2006) manifiesta que el lenguaje JAVA:

Es un lenguaje muy poderoso y especialmente diseñado para la creación de scripts, que se alojan dentro de un documento HTML. Dicho lenguaje es propiedad de Netscape. Es un API programable que permite crear escritos de evento objetos y acciones bajo cualquier plataforma, gracias a que JavaScript es parte de la conexión en vivo, se puede usar para crear interacciones entre documentos HTML, plug-ins (aplicaciones que corren dentro del programa) y java.

El lenguaje de programación de JAVA esta complementado por API's mediante subrutinas matemáticas o rutinas asociadas las mismas que se incluyen a través de interfaces graficas de usuario, para la comunicación de red, para lectura y escritura de archivo, por ende JAVA es una plataforma independiente que puede que se adapta a cualquier necesidad para garantizar que la ejecución sea exitosa.

Los sistemas hechos en Java son compilados a un lenguaje intermedio formado por bytecodes (código máquina de un microprocesador virtual almacenado en un fichero de extensión .class) y después es interpretado por una máquina virtual (JVM – *Java Virtul Machine*). De este modo se consigue la independencia de la máquina, el código compilado se ejecuta en máquinas virtuales que si son dependientes de la plataforma. (Gálvez & García, 2007)

Para (Gálvez & García, 2007) Java es una plataforma de software desarrollada por Sun Microsystems la misma que consta de:

- Lenguaje de programación en sí, totalmente orientado a objetos.
- Máquina virtual de JAVA junto con el entorno de ejecución, también llamado JRE (Java Runtime Environment), que debe existir en cada plataforma en la que se quiere ejecutar programas Java (pág. 23).
- Las API de JAVA, un conjunto de bibliotecas estándares para el lenguaje que lo dotan de múltiples capacidades: acceso a base de datos, entrada/ salida de ficheros, conexiones a redes, construcción de ventana y sus componentes (botones, cuadros de texto, listas, etc.) gestión de eventos, etc.

1.5. Base de datos

(León, 2008) Menciona que “Un sistema de manejo se base de datos (en Inglés DBMS Database Management System) es un conjunto de elementos interrelacionados y una serie

de programas que permiten a varios usuarios tener acceso a estos archivos ya sea consultarlos o actualizarlos”.

En esta misma línea el autor (Kendall K. , 2005) afirma que:

Las bases de datos no son tan solo una colección de archivos, más bien, una base de datos es una fuente central de datos destinados a compartirse entre muchos usuarios para una diversidad de aplicación. El corazón de una base de datos lo constituye el sistema de administración de la base de datos (DBMS, database management system) el cual permite la creación y modificación y actualización de la base de datos, la recuperación de datos y la generación de informes y pantallas. (Pág. 444)

Una base de datos debe ser integral y eficiente para que se cumpla con los objetivos propuestos.

Seguidamente se menciona algunos de los objetivos que se persigue al utilizar una base de datos, según (Kendall K. , 2005) estos son:

1. Asegurar que los datos se puedan compartir entre los usuarios para una diversidad de aplicaciones.
2. Mantener datos que sean exactos y consistente.
3. Asegurar que todos los datos requeridos por las aplicaciones actuales y futuras se podrán acceder con facilidad.
4. Permitir a la base de datos evolucionar conforme aumente las necesidades de los usuarios. (pág. 444)

De tal manera que la base de datos se puede decir que es un sistema que almacena información en grandes volúmenes, además esta base de datos debe contar con varios parámetros como la seguridad (sólo personas autorizadas podrán acceder a la información), integridad (la información se mantendrá sin pérdidas de datos), e independencia (esta característica es fundamental ya que una buena base de datos debería ser independiente del sistema operativo o programas que interactúen con ella), que haga de la base un sistema seguro para quien lo requiere (APR, 2014).

Pero existen otras bases de datos de pago, entre las cuales se tiene:

- MySQL (ediciones de pago).
- dBase.
- IBM Informix.
- Microsoft SQL SERVER.
- Orade.
- Sybase

La base de datos tiene dos tipos, entre los cuales se puede encontrar: OLTP (On Line Transaction Processing) o también llamadas base de datos dinámicas, lo cual quiere decir que la información puede ser modificada en tiempo real, es decir, se insertan, se eliminan, se modifican y se consultan datos en línea durante la operación del sistema” (IBM, 2015).

Otro tipo es el OLAP (On Line Analytical Processing), también son llamadas bases de datos estáticas, con lo cual la información en tiempo real no se ven afectadas, es decir, no se insertan, no se eliminan y tampoco se modifican datos, de tal manera que solo se permite realizar consultas sobre información ya existente para su posterior análisis y toma de decisiones. (IBM, 2015)

1.5.1. MySQL.

Tomando en cuenta lo mencionado por Cobo (2005) El cual indica que:

Es un sistema de administración de base de datos, relacional, rápida, sólida y flexible. Es ideal para crear base de datos con acceso desde páginas web dinámicas, para la creación de sistemas de transmisión on –line o para cualquier otra solución profesional que implique almacenar datos, teniendo la posibilidad de realizar múltiples y rápidas consultas.

Según (MySQL, 2015) MySQL “es la base de datos de código abierto más popular del mundo, lo que permite la entrega rentable de aplicaciones de base de datos basados en la web e integrados y escalables, incluidos los cinco mejores sitios web, de alto rendimiento y fiable”.

Minerva (2014) concluye que MySQL se puede ir adaptar a los siguientes sistemas:

- Windows (MWS 2003, 2008, 2012; Windows 7, 8, 10).
- Linux.
- Solaris.
- Mac OS.
- Novell Net Ware (6.0 o superior).
- BSD (FreeBSD, NetBSD, OPenBSD, BSD/OS).
- OS/2.
- BeOS
- RISC OS.
- AS/400.
- SGI IRIX 6.5.x (pág. 70).

Un aspecto importante, es que parte del código de MySQL se encuentra escrito en lenguaje C/C++ y la sintaxis de su uso es bastante simple, de tal manera que permite a los programadores crear bases de datos simples o complejas con mucha facilidad. La siguiente figura muestra como son las aplicaciones de gestión de MySQL.

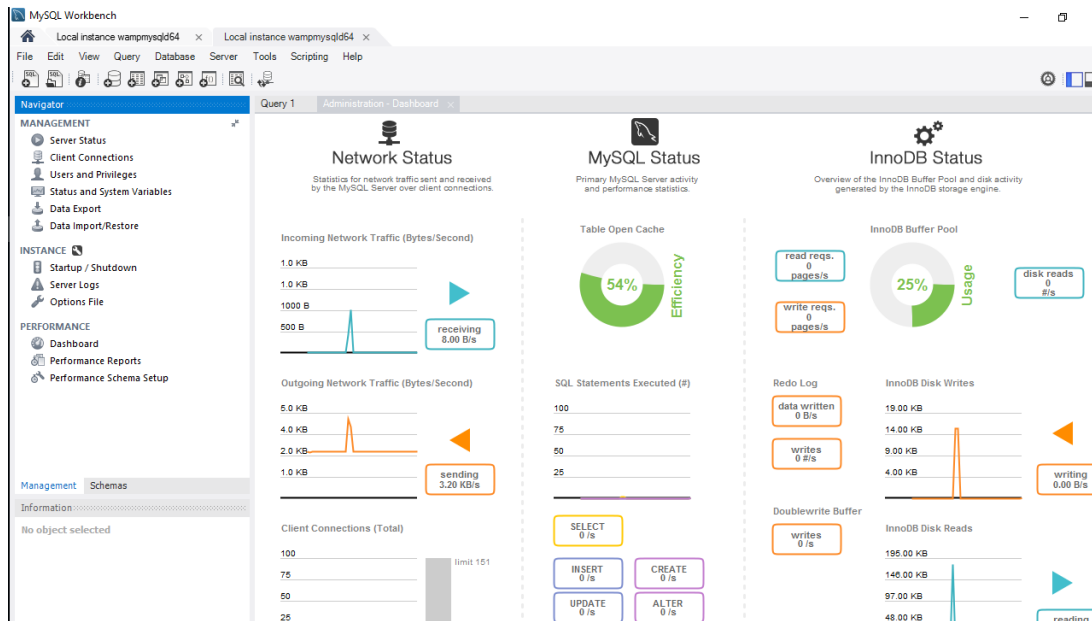


Figura 1. Aplicaciones de gestión para MySQL
Fuente: (Minera, 2011).

Este gestor de base de datos es bastante utilizado en el desarrollo web, ya que permite a los programadores, realizar cambios en los sitios de manera simple, con tan sólo cambiar un archivo, evitando tener que modificar todo el código web. Esto se debe a que MySQL, trabaja con un sistema centralizado de gestión de datos, que permite realizar cambios en un solo archivo y que se ejecuta en toda la estructura de datos que se comparte en la red.

1.6. Herramientas de soporte utilizadas

Algunas herramientas de soporte en aplicaciones web son los frameworks web:

1.6.1. Framework Web

Conjunto de componentes (por ejemplo clases en java y descriptores y archivos de configuración en XML) que componen un diseño reutilizable que facilita y agiliza el desarrollo de sistemas web. El objetivo principal de Framework, es que acelera el proceso de desarrollo, reutiliza los códigos ya existentes y promueve las buenas prácticas de desarrollo como el uso de patrones. (Gutiérrez, 2013, pág. 1).

En los últimos años MVC ha sido utilizado con mucha frecuencia en aplicaciones web debido a que han aparecido varios framework de desarrollo web que utilizan este patrón como modelo arquitectónico de las aplicaciones web

Existen varios tipos de frameworks, pero los más utilizados, según Ríos (2010) son:

- a.- Framework JSF
- b.- Framework JSP
- c.- Framework JQUERY:

a) **Framework: JSF:** Java Server Faces (JSF) es una tecnología y framework para aplicaciones Java basadas en web que simplifica el desarrollo de interfaces de usuario en aplicaciones Java EE. JSF usa Java Server Pages (JSP) como la tecnología que permite hacer un despliegue de las páginas, pero también se pueden acomodar a otras tecnologías como XUL (Acrónimo de XML –based User –interface language, lenguaje basado en XML para la interfaz del usuario)

b) **Framework: JSP:** Según (Maciá, Administración de servicios de Internet, 2008) indica que:

Java Server Pages (JSP) es la propuesta desarrollada por Sun Microsystems para generar páginas Web de forma dinámica en el servidor. Esta tecnología está basada en Java y permite a los programadores generar dinámicamente HTML, XML o algún tipo de página web. Para ello, como en el resto de casos, se inserta código, en este caso Java, dentro del contenido estático.

Las versiones recientes hacen posible que ejecutar clases java referenciadas en la página HTML a través de etiquetas que se denominan “Tablib” siendo un archivo de configuración escrito en XML que puede ser usado por todos los programadores para el almacenamiento. “Entre las principales ventajas de JSP frente a otras tecnologías es que puede integrarse con Java (.class) esto permite organizar las aplicaciones web en niveles, otra ventaja es que se podrá disponer de API en su totalidad” (Maciá, 2008, pág. 108).

Para la aplicación de este trabajo de investigación es importante establecer las ventajas de la aplicación del JPS.

- Se puede ingresar con clases Java con lo cual se puede organizar las aplicaciones web.
- Las clases Java se almacenan con la finalidad de dar mayor seguridad a la información.
- Puede ser usado en cualquier sistema sin ningún inconveniente.
- La portabilidad es factible con esta aplicación.
- Se puede disponer de API en su totalidad. (Maciá, Administración de servicios de Internet, 2008)

c) **Framework JQUERY:** este framework cuenta con numerosas especificaciones, “la más importante es la flexibilidad para acceder a todos los elementos del documento HTML, esto se da mediante la multitud de selectores. Hay que destacar que jQUERY cuenta con varias versiones, que van de la mano con la tecnología y necesidades de los usuarios” (Van Lancker, 2014).

De acuerdo a Van Lanckers (2014), "JQUERY es un framework JavaScript libre y Open Source, del lado cliente, que se centra en la interacción entre el DOM, JavaScript, AJAX y Html. El objetivo de esta librería JavaScript es simplificar los comandos comunes de JavaScript". (pág. 4)

JQUERY es sin lugar a dudas un sistema que está integrado a JavaScript, es decir es una biblioteca que cuenta con una diversidad de funciones, no obstante trabaja como HTML y AJAX haciendo que la ejecución sea más simple que un API es decir, fácil de usar y una mayor adaptabilidad.

1.7. Servidor web: Apache.

En su libro Administración de servicios de Internet De la Teoría a la Práctica (Maciá, 2008) menciona que:

El servidor Web Apache es un servidor HTTP de código abierto el cual ha sido desarrollado por el grupo de Apache Software Foundation dentro del proyecto HTTP Server (httpd). Este servidor ha sido creado sobre los principales sistemas operativos (Unix y Windows) haciendo frente a los servidores Web propietario de mayor uso en el mercado. Para ello sus creadores han desarrollado el servidor con los requerimientos de: seguridad, eficiencia, extensibilidad y estandarización. (pág. 40)

Es un servidor web que está orientado a la seguridad, eficiencia, extensibilidad y estandarización funciones que se adaptan a las necesidades del mercado y a su vez es uno de los servidores más usados por los programadores por la adaptabilidad y la evolución que tiene; también es un servidor web flexible, rápido y eficiente, continuamente actualizado y adaptado a los nuevos protocolos HTTP.

1.7.1. Arquitectura de apache.

Apache es un servidor web de software libre que tiene como objetivo proveer páginas web a los clientes o navegadores que las solicitan. A continuación se muestra un modelo grafico del funcionamiento:

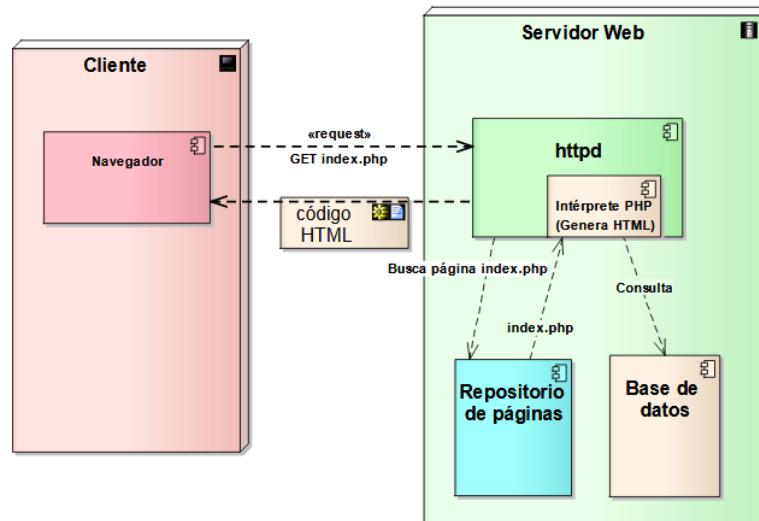


Figura 2. Arquitectura Servidor Web Apache
 Fuente: <https://www.fdi.ucm.es/profesor/jpavon/web/31-ServidoresWeb-Apache.pdf>

El servidor apache cuenta con tres módulos que están integrados entre sí, que se detallan en la siguiente tabla:

Tabla 1. Módulos de la estructura del servidor Apache

Modulo	Descripción
Módulos base	Se trata de módulo con las funcionalidades básicas del Apache.
Módulos multiproceso	Son los responsables de la unión con los puertos de la máquina, aceptando las peticiones y enviando a los hijos a atender a las peticiones.
Módulos adicionales	Módulos de terceros que añaden funcionalidad adicional al servidor.

Fuente: (Maciá, 2008, pág. 40)

Las funcionalidades más elementales se encuentran en el módulo base, siendo necesario un módulo multiproceso para manejar las peticiones. Se han diseñado varios módulos multiproceso para cada uno de los sistemas operativos sobre los que se ejecuta el Apache, optimizando el rendimiento y rapidez del código.

Los módulos permiten contar con una estructura sólida que se componen de un núcleo central de funciones en donde se pueden añadir un conjunto de funcionalidades que sirven de base para garantizar una carga dinámica este acorde al multiproceso para que el tiempo de compilación sea el adecuado.

1.8. Entorno del desarrollo: Netbeans.

El entorno de desarrollo de Netbeans. “Nació con el apoyo de Oracle quienes son desarrolladores de Java, en este proceso están respaldados por la comunidad quienes trabajan en un mismo lenguaje con la finalidad de establecer un sistema integrado que puede ser usado por cualquiera”. (Granados, 2014, pág. 107)

Según (Garrido P. , 2015) menciona que:

El entorno de desarrollo integrado (IDE) NetBeans es un entorno gratuito para el desarrollo de programas Java estándar (stand –alone) y applets. Además, NetBeans también facilita la tarea del desarrollo de aplicaciones empresariales Java EE o J2EE permitiendo a compilación, empaquetado, distribución y ejecución dentro del mismo entorno. (pág. 6)

Es un programa versátil y está disponible para plataformas como Microsoft Windows, Linux y Mac OSX permitiendo que sea usado con frecuencia para el desarrollo de aplicaciones empresariales

1.9. Patrón arquitectónico MVC

MVC es patrón arquitectónico de la familia de estilos arquitectónicos de Llamada y Retorno, este estilo arquitectónico enfatiza la modificabilidad y la escalabilidad. Considerado por David Garlan como una micro-arquitectura. El propósito que cumple MVC es tomar datos de un almacenamiento y mostrarlos al usuario, luego de que el usuario introduce modificaciones, las mismas se reflejan en el almacenamiento, esto es posible dado que el flujo de información ocurre entre el almacenamiento y la interfaz.

El patrón MVC separa el modelado del dominio, la presentación y las acciones basadas en datos ingresados por el usuario.

El Modelo administra el comportamiento y los datos del dominio de aplicación, responde a requerimientos de información sobre su estado (usualmente formulados desde la vista) y responde a interacciones de cambiar el estado (habitualmente desde el controlados); la Vista maneja la visualización de la información; el Controlador interpreta las acciones del ratón y teclado, informando al modelo y/o a la vista para que cambien según resulte apropiado. Esta separación permite construir y probar el modelo independientemente de la representación visual, la separación vista y controlador puede ser secundaria, de hecho muchos framework de interfaz implementan ambos roles en un solo objeto. (Reynoso, 2017)

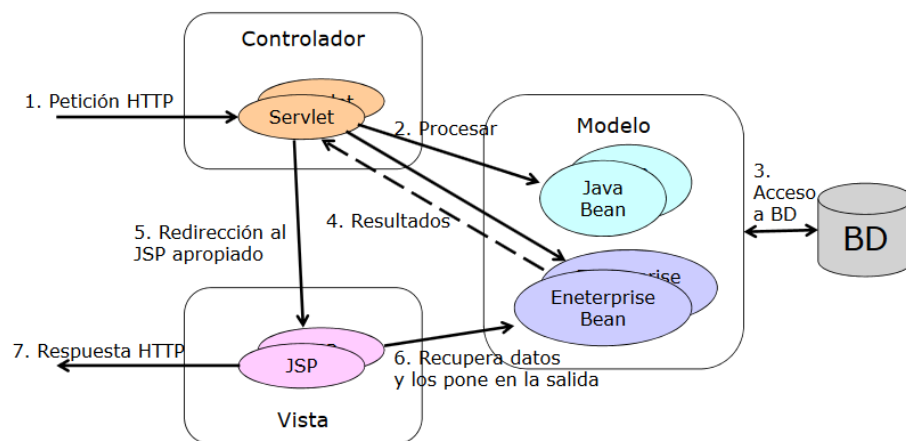


Figura 3: Patrón Arquitectónico MVC
Fuente: <https://www.fdi.ucm.es/profesor/jpavon/web/46-mvc.pdf>

1.10. Metodología de desarrollo de la aplicación RUP

Para el desarrollo de la aplicación la misma que se ajusta al tipo de software que se va a realizar y dispone de herramientas que sustentan el desarrollo.

La metodología RUP tiene como fin la entrega de un producto finalizado de software, a través de la estructuración de los procesos y con la medición de la eficiencia de la organización. En el proceso de desarrollo de la aplicación o software se utiliza el lenguaje unificado de modelado UML, el cual constituye la metodología más utilizada para análisis e implementación.

El RUP constituye un grupo de metodologías que se adaptan al contexto y necesidades de la organización, mediante pasos que permitirán la realización.

La metodología “se centra en la producción y mantenimiento de modelos del sistema”.

Las características que tiene la metodología RUP son:

- Forma disciplinada de asignar tareas y responsabilidades (quién hace qué, cuándo y cómo)
- Pretende implementar las mejores prácticas en Ingeniería de Software.
- Desarrollo iterativo
- Administración de requisitos
- Uso de arquitectura basada en componentes
- Control de cambios
- Modelado visual del software
- Verificación de la calidad del software (Procesos de Software, 2013).

RUP es un producto creado por Rational (IBM) y su característica es ser iterativo e incremental, se centra en la arquitectura y se guía a través de los casos de uso. Esta metodología incluye artefactos (que llegan a ser los tangibles del proceso por ejemplo el

modelado de casos de uso, el código fuente) y roles (papel que desempeña una persona en determinado momento, aunque las mismas pueden desempeñar distintos roles a lo largo del proceso).

El ciclo de vida del proceso de acuerdo a la metodología RUP, considerando el esfuerzo de las actividades según la fase del proyecto es como sigue:

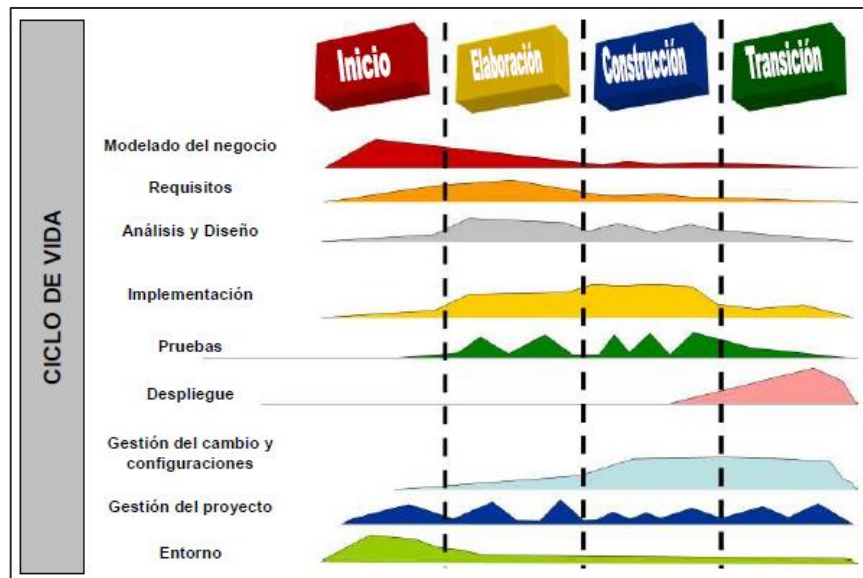


Figura 4. Esfuerzo en actividades según fase del proyecto

Fuente: <https://procesosdesoftware.wikispaces.com/METODOLOGIA+RUP>

Se considera el ciclo de vida del proceso en el RUP como una implementación en espiral.

El RUP divide el proceso en 4 fases, en las cuales se realizan varias iteraciones, de acuerdo al tipo de proyecto y de acuerdo a estas, se hará un mayor o menor hincapié en las diferentes actividades (Procesos de Software, 2013).

1.10.1. Fases del ciclo de vida del RUP.

A continuación se detalla las fases que se incluyen en el ciclo de vida del producto de acuerdo a RUP:

1.10.1.1. Fase de inicio

“Esta fase tiene como propósito definir y acordar el alcance del proyecto con los patrocinadores, identificar los riesgos asociados al proyecto, proponer una visión muy general de la arquitectura de software y producir el plan de las fases y el de iteraciones posteriores.” (Procesos de Software, 2013).

1.10.1.2. Fase de elaboración:

“En la fase de elaboración se seleccionan los casos de uso que permiten definir la arquitectura base del sistema y se desarrollaran en esta fase, se realiza la especificación de los casos de uso seleccionados y el primer análisis del dominio del problema, se diseña la solución preliminar.” (Procesos de Software, 2013).

1.10.1.3. Fase de desarrollo:

“El propósito de esta fase es completar la funcionalidad del sistema, administrar los cambios de acuerdo a las evaluaciones realizados por los usuarios y se realizan las mejoras para el proyecto”. (Procesos de Software, 2013).

1.10.1.4. Fase de cierre:

“El propósito de esta fase es asegurar que el software esté disponible para los usuarios finales, ajustar los errores y defectos encontrados en las pruebas de aceptación, capacitar a los usuarios y proveer el soporte técnico necesario. Se debe verificar que el producto cumpla con las especificaciones entregadas por las personas involucradas en el proyecto”. (Procesos de Software, 2013).

1.10.2. Elementos del RUP.

- **Actividades:** Definir los procesos que se realizarán en cada etapa del desarrollo.
- **Artefactos:** Herramientas que se utilizan en el desarrollo de la aplicación, ya sea documentos, modelos, elementos, etc. (Procesos de Software, 2013).

Artefactos

Los denominados artefactos para el desarrollo de la metodología RUP de acuerdo a las fases, son los que sirven para comprender tanto el análisis como el diseño del sistema. (Procesos de Software, 2013).

CAPÍTULO II.
DESARROLLO DE LA SOLUCIÓN

Para el desarrollo de la aplicación web para gestionar el proceso de préstamo de instalaciones del área de deportes, recreación y cultura de la Universidad Técnica Particular de Loja se ha considerado los principales artefactos de la metodología RUP y las características han sido adaptadas a los requerimientos y funcionalidades del software a desarrollar. El desarrollo del proyecto será diseñado solamente por el desarrollador y su relación con el coordinador de asignación de áreas. Las principales fases del proceso serán inicio, elaboración, construcción y transición, las cuales son las que permitirán la construcción del sistema.

2.1. Módulos de la Solución.

- Módulo de administración y seguridad.
- Módulo de verificación.
- Módulo de gestión y reserva.
- Módulo de reportes.
- Módulo de mantenimiento.

2.2. Fase de inicio.

En esta fase inicial de la metodología de desarrollo utilizada se empleó la técnica de la entrevista para la recolección de información, la entrevista realizada al personal del departamento de Misiones Universitarias permitió conocer y entender que necesidades que se pretende solucionar; la información obtenida sirve de apoyo para realizar los documentos de la visión y para la especificación de requerimientos.

Dentro de esta fase se determinaron la visión y el alcance del proyecto (Documento Visión), en el que aparecen las características, beneficios de la aplicación; además se definieron los requerimientos; las situaciones y problemas a los que se debe dar una solución.

Es en esta fase que el cliente realiza la solicitud de la aplicación, entrega los datos del proyecto así como aspectos generales y técnicos de la aplicación web que se va a desarrollar.

2.2.1. Análisis del proceso actual.

La persona encargada de revisar el proceso de préstamo de las instalaciones, facilitó la documentación existente acerca del proceso de préstamo de las instalaciones lo que permitió analizar y conocer más detalladamente acerca de las actividades que realiza el personal involucrado. Además se revisó el calendario de reservas de las canchas, y las normativas de préstamo. De la documentación proporcionada se pudo conocer aspectos como:

- Los formularios de préstamo de las canchas se realizan manualmente y en el departamento de Misiones Universitarias.
- No hay reportes en tiempo real de las reservas realizadas.

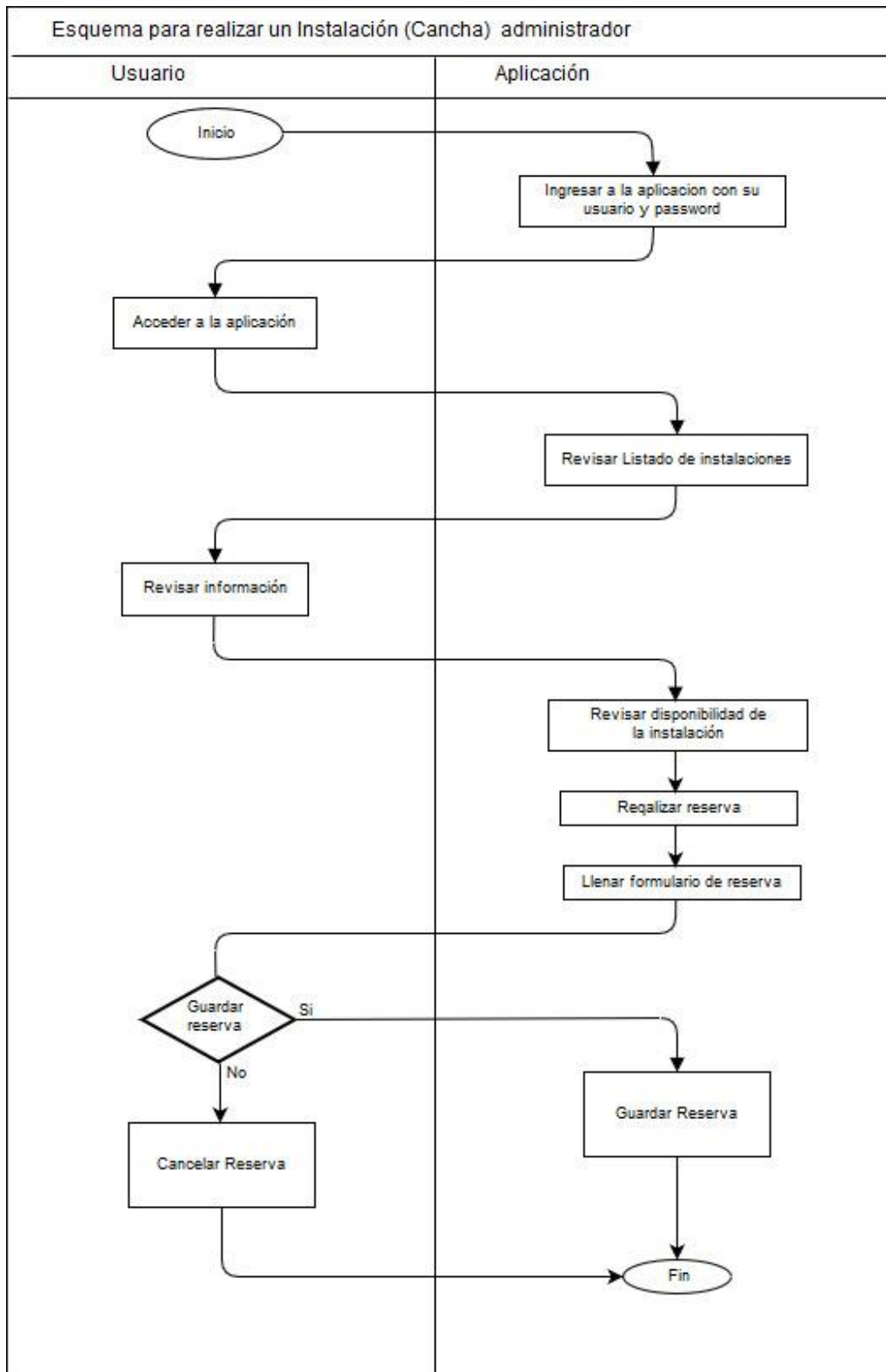


Figura 5. Modelo de Negocios
 Autor: Pablo Castillo

El presente diagrama podemos observar el modelo del negocio que describe las actividades entre el usuario y la aplicación.

Artefactos Fase 1

Entrevista para determinar requerimientos	ANEXO 1
Documento Visión	ANEXO 2

2.3. Fase de elaboración.

De la instrucción obtenida en la fase anterior se obtuvo la especificación de los requerimientos funcionales y los no funcionales, también se identifican las limitaciones del diseño, en esta fase se establecen los casos de uso empleando UML donde se detalla el comportamiento y sus respectivas interacciones de los usuarios con la aplicación, debemos asumir que los casos de uso están relacionados con los requerimientos que se tienen del sistema, la representación de los casos de uso se hacen a través de un diagrama general de casos de uso y a su vez en forma detallada en casos de uso específicos. Con la especificación y los diagramas de caso de uso se desarrollan las respectivas interfaces de usuario o prototipos que configuran la manejabilidad del sistema. Una vez desarrollados los prototipos y los casos de uso se puede modelar los correspondientes diagramas de clases donde se indican las clases con que cuenta el sistema y sus relaciones. También en base al diagrama conceptual de la base de datos, podremos generar el diagrama físico de los datos, donde la estructura de la base de datos se podrá modelar, aquí podremos almacenar la información de la aplicación, incluyendo sus atributos y características.

En esta fase se ha desarrollado los requerimientos del proyecto, de donde se desprende los casos de uso que indican la funcionalidad del sistema. Se desarrolla el diseño y la arquitectura del proyecto.

2.3.1. Requerimientos funcionales.

Para identificar los requerimientos se ha utilizado entrevistas con el personal del departamento de Misiones Universitarias que han aportado directamente con los requerimientos que al aplicativo debe tener para que cumpla su función.

REQ001 Registro de usuario

Tabla 2. REQ001 Registro de usuario

Nombre	Registro de usuarios
Descripción	El usuario debe poder registrarse para poder ingresar a la aplicación, y tener acceso a verificar que canchas están reservadas
Prioridad	Obligatorio
Datos de entrada	Los datos necesarios para este requerimiento son los datos del usuario de debe ingresar a la aplicación para quedar registrado. Los datos solicitados son: nombre, apellido, email, teléfono, password.
Proceso	Una vez que el usuario ingresa los datos solicitados por la aplicación estos se registran en la base de datos.

Elaborado por: Pablo castillo

REQ002 Registro usuarios por un administrador

Tabla 3. Registro usuarios por un administrador

Nombre	Registro usuarios por un administrador
Descripción	Un usuario administrador registra usuarios, teniendo la posibilidad de asignar un rol (usuario, coordinador, administrador) siendo el rol de usuario el que tiene menos privilegios, y administrador, el que tiene todos los privilegios del aplicativo.
Prioridad	Obligatorio
Datos de entrada	<p>La información que ingresa es la que la aplicación solicita de cada usuario para que quede registrada en la base de datos, a más del rol que tiene que definir el administrador, los cuales se detallan a continuación.</p> <p>Roles</p> <p>Usuario: Podrá visualizar la información de áreas disponibles y no disponibles. No puede realizar reservas en el sistema. Puede también ver el listado de las instalaciones, y la Normativa de Uso y préstamo de las instalaciones.</p> <p>Coordinador: Es el encargado de realizar el proceso de reserva, tiene a más de las opciones del usuario anterior, la opción de reservar una instalación de acuerdo a la disponibilidad, puede llenar el formulario de reserva.</p> <p>Administrador: El administrador puede tener acceso a la información total del sistema, modificar la base de datos, será el encargado de dar mantenimiento, así como realizar el control del programa. Tiene a disposición todos los módulos del sistema.</p>
Proceso	El usuario administrador registra a los usuarios del sistema y tiene la facultad de designar un rol a cada usuario que registra.

Elaborado por: Pablo castillo

REQ003 Resetear la clave de un usuario

Tabla 4. REQ003 Resetear la clave de un usuario

Nombre	Resetear la clave de un usuario
Descripción	Si algún usuario olvida su clave, el usuario administrador puede realizar el proceso de resetear la clave de ese usuario.
Prioridad	Media
Datos de entrada	Los datos de entrada que se requieren es este requerimiento son los datos del usuario al que se le va a resetear la clave. Hay que tomar en cuenta que este requerimiento solo puede ser realizado por un usuario administrador
Proceso	El usuario administrador ingresa al módulo de mantenimiento, submódulo de usuarios, selecciona el usuario al que le va a cambiar la clave y procesa a resetearla. La clave quedará reseteada y el usuario al que se le resetea la clave podrá volver a ingresar al aplicativo.

Elaborado por: Pablo castillo

REQ004 Revisar usuarios registrados en la aplicación.

Tabla 5. REQ004 Revisar usuarios registrados en la aplicación.

Nombre	Revisar usuarios registrados.
Descripción	La aplicación puede mostrar la lista de usuarios que se encuentran registrados en la aplicación.
Prioridad	Alta
Datos de entrada	Los datos de entrada de este requerimiento son los datos del usuario que se desea realizar la consulta
Proceso	Una vez que el usuario ingresa al aplicativo, podrá ir al módulo para poder visualizar la lista de usuarios.

Elaborado por: Pablo castillo

REQ005 Eliminar usuario.

Tabla 6. REQ005 Eliminar usuario

Nombre	Eliminar usuario
Descripción	Se puede eliminar un usuario de la aplicación cuando ya no sea necesario que este siga registrado en la aplicación.
Prioridad	Baja
Datos de entrada	En este caso es necesario ubicar en el aplicativo al usuario que se va a eliminar.
Proceso	El usuario administrador ingresa al módulo de mantenimiento y procede a eliminar al usuario.

Elaborado por: Pablo castillo

REQ006 Revisar lista de canchas.

Tabla 7. REQ006 Revisar lista de canchas.

Nombre	Revisar lista de canchas.
Descripción	Cualquier usuario registrado en el aplicativo (usuario, coordinador, administrador) puede realizar esta revisión ya que está presente para los tres tipos de roles.
Prioridad	Media
Datos de entrada	Aquí se debe ingresar al aplicativo, y listar las canchas.
Proceso	Una vez que se ingresa al aplicativo se puede ver la lista de canchas.

Elaborado por: Pablo Castillo

REQ007 Revisar normativas de préstamo.

Tabla 8. REQ007 Revisar normativas de préstamo

Nombre	Revisar normativas de préstamo de las canchas
Descripción	Los usuarios registrados en el aplicativo (usuario, coordinador, administrador) pueden revisar las normativas de préstamo de las canchas.
Prioridad	Media
Datos de entrada	Se debe ingresar al aplicativo y revisar las normativas.
Proceso	Una vez que se ha ingresado al aplicativo se puede revisar las normativas de préstamo.

Elaborado por: Pablo castillo

REQ008 Verificar disponibilidad de las canchas.

Tabla 9. REQ008 Verificar disponibilidad de las canchas

Nombre	Verificar disponibilidad de las canchas
Descripción	Revisar la disponibilidad de las canchas para poder reservarlas, este requerimiento lo puede realizar cualquier usuario (usuario, coordinador, administrador) que se encuentre registrado en el aplicativo.
Prioridad	Alta
Datos de entrada	Se debe ingresar al aplicativo y revisar la disponibilidad de las canchas.
Proceso	Una vez que se ha ingresado al aplicativo se puede revisar la disponibilidad de las canchas.

Elaborado por: Pablo castillo

REQ009 Reservar una cancha.

Tabla 10. REQ009 Reservar una cancha

Nombre	Reservar una cancha
Descripción	Este requerimiento lo pueden realizar los usuarios coordinadores y administradores, y es donde se realiza la reserva de una cancha en la aplicación.
Prioridad	Alta
Datos de entrada	En este caso es necesario ubicar en el aplicativo al usuario que se va a eliminar.
Proceso	La reserva de una cancha lo puede realizar un usuario coordinador o un administrador, después de verificar la disponibilidad de una cancha en el módulo de verificación se procede a reservar la cancha en el calendario de la aplicación, y después de este paso, se llena el formulario de reserva de la cancha donde se llena los datos con la información de la reserva. Una vez que se ha ingresado toda la información correctamente se guarda esta información en la base de datos de la aplicación.

Elaborado por: Pablo castillo

REQ010 Revisar reportes.

Tabla 11. REQ010 Revisar reportes

Nombre	Revisar reportes
Descripción	Permite revisar los reportes que la aplicación disponible.
Prioridad	Alta
Datos de entrada	Los datos que aquí se ingresa son los datos del usuario que va a revisar los reportes. Dependiendo del tipo de reporte debe ingresar los rangos de las fechas que desea generar los reportes.
Proceso	El usuario administrador ingresa al módulo de reportes y puede revisar todos los reportes que tiene a su disposición, de acuerdo al tipo de reporte que desee revisar, tiene que ingresar los rangos de las fechas que desea revisar los reportes.

Elaborado por: Pablo castillo

REQ011 Guardar reportes.

Tabla 12. REQ011 Guardar reportes

Nombre	Guardar reportes
Descripción	Permite guardar los reportes que se ha sacado.
Prioridad	Media
Datos de entrada	Los datos que aquí se ingresa son los datos del usuario que va a revisar los reportes. Dependiendo del tipo de reporte debe ingresar los rangos de las fechas que desea generar los reportes.
Proceso	El usuario administrador ingresa al módulo de reportes y puede revisar todos los reportes que tiene a su disposición, de acuerdo al tipo de reporte que desee revisar, tiene que ingresar los

	rangos de las fechas que desea revisar los reportes. Una vez que a sacado los reportes que necesita, los puede guardar.
--	---

Elaborado por: Pablo castillo

REQ012 Imprimir reportes.

Tabla 13. REQ012 Imprimir reportes

Nombre	Imprimir reportes
Descripción	Permite imprimir los reportes que se ha sacado.
Prioridad	Baja
Datos de entrada	Los datos que aquí se ingresa son los datos del usuario que va a revisar los reportes. Dependiendo del tipo de reporte debe ingresar los rangos de las fechas que desea generar los reportes.
Proceso	El usuario administrador ingresa al módulo de mantenimiento y procede a imprimir los reportes.

Elaborado por: Pablo castillo

REQ013 Crear cancha.

Tabla 14. REQ005 Crear cancha

Nombre	Crear cancha
Descripción	Crear una nueva cancha en el aplicativo y que esta se guarde en la base de datos.
Prioridad	Media
Datos de entrada	Los datos de entrada de este requerimiento son los datos de la cancha que se va a crear: nombre de la cancha, capacidad de personas que pueden ingresar a esta cancha.
Proceso	El usuario administrador ingresa al módulo de mantenimiento y procede a ingresar los datos que solicita la aplicación, una vez que ha ingresado esta información esta se guarda en la base de datos de la aplicación.

Elaborado por: Pablo castillo

REQ014 Modificar nombre de una cancha.

Tabla 15. REQ014 Modificar nombre de una cancha

Nombre	Modificar nombre de una cancha
Descripción	Se puede modificar el nombre de una cancha.
Prioridad	Baja
Datos de entrada	Los datos de entrada de este requerimiento son el nuevo nombre de la cancha que se requiere renombrar.
Proceso	El usuario administrador ingresa al módulo de mantenimiento, ubica la cancha que desea modificar el nombre y procede a modificarlo.

Elaborado por: Pablo Castillo

REQ015 Modificar capacidad de personas de una cancha.

Tabla 16. REQ015 Modificar capacidad de personas de una cancha

Nombre	Modificar capacidad de una cancha
Descripción	Modificar la capacidad de personas que de una cancha.
Prioridad	Media
Datos de entrada	Los datos de entrada de este requerimiento son el nuevo número de personas que permite esa cancha.
Proceso	El usuario administrador ingresa al módulo de mantenimiento, ubica la cancha que desea modificar y procede a ingresar el nuevo número de personas que esa cancha va a permitir.

Elaborado por: Pablo castillo

2.3.2. Requerimientos no funcionales

Facilidad de uso: Interfaz amigable que encontramos en la presentación de la aplicación, permita acceder fácilmente a los diferentes módulos de la aplicación como son: reservas, visualización de la disponibilidad de las instalaciones, generación y visualización de reportes; a la vez que nos permite realizar el proceso de mantenimiento como es el registro de usuarios, registro o modificación de alguna instalación.

Como ya se mencionó anteriormente la interfaz es muy amigable a la vez que intuitiva, ya que una vez que se despliega el index.html nos aparece el botón de iniciar sesión.

Seguido a esto nos aparece la parte donde ponemos los datos de usuario para ingreso a la aplicación, estos se verifican en la base de datos si son los datos correctos para entrar a la aplicación. Como se puede observar se ha minimizado en lo posible los pasos de ingreso sin dejar de lado la seguridad, se ha verificado que los tiempo de respuesta son instantáneos al ingresar a la aplicación. También podemos encontrar legibilidad, comprensibilidad, claridad.

Seguridad de los datos: Los datos que se ingresan a la base deben estar disponibles permanentemente para los usuarios y deben guardarse de forma confiable. Adicional a esto se solicita a todos los usuarios datos personales para saber que usuario está registrado en el aplicativo. Esta información como ya se mencionó esta guardada en una base de datos que solo tienen acceso los usuarios administradores, no cualquier usuario puede ver esta información confidencial. La capa de persistencia de la aplicación hace que esté disponible en todo momento.

Eficiencia y escalabilidad: La aplicación deberá permitir realizar cambios o mejoras y debe ser muy rápida en su acceso. El patrón arquitectónico utilizado para el desarrollo de la aplicación permite que esta se pueda ampliar en sus funciones y características, ya que al ser una aplicación que tiene su código distribuido en capas, le permite agregar (o en su efecto quitar) funcionalidades. Se ha verificado que el aplicativo es rápido en su procesamiento, teniendo buenos tiempos de respuesta.

2.3.3. Casos de uso.

Los casos de uso detallan las acciones del sistema desde el punto de vista del usuario, en base a lo mencionado se han desarrollado los siguientes casos de uso que se han empleado para el desarrollo de la aplicación.

Usuario:

Podrá ver el listado y comprobar la disponibilidad de las canchas que se muestran en el calendario de la aplicación, así como ver las Normativas de Préstamo de las Canchas.

Coordinador:

Podrá ver el listado y comprobar la disponibilidad de las canchas que se muestran en el calendario de la aplicación, así como ver las Normativas de Préstamo de las Canchas. Además de poder realizar las reservas en la aplicación.

Administrador:

Además de tener los privilegios de los usuarios mencionados anteriormente, este puede registrar usuarios en el aplicativo, también tiene los permisos para dar mantenimiento a los usuarios y a las canchas que se encuentran registradas. Tiene acceso a los reportes que facilita la aplicación así como también puede guardar e imprimir estos reportes.

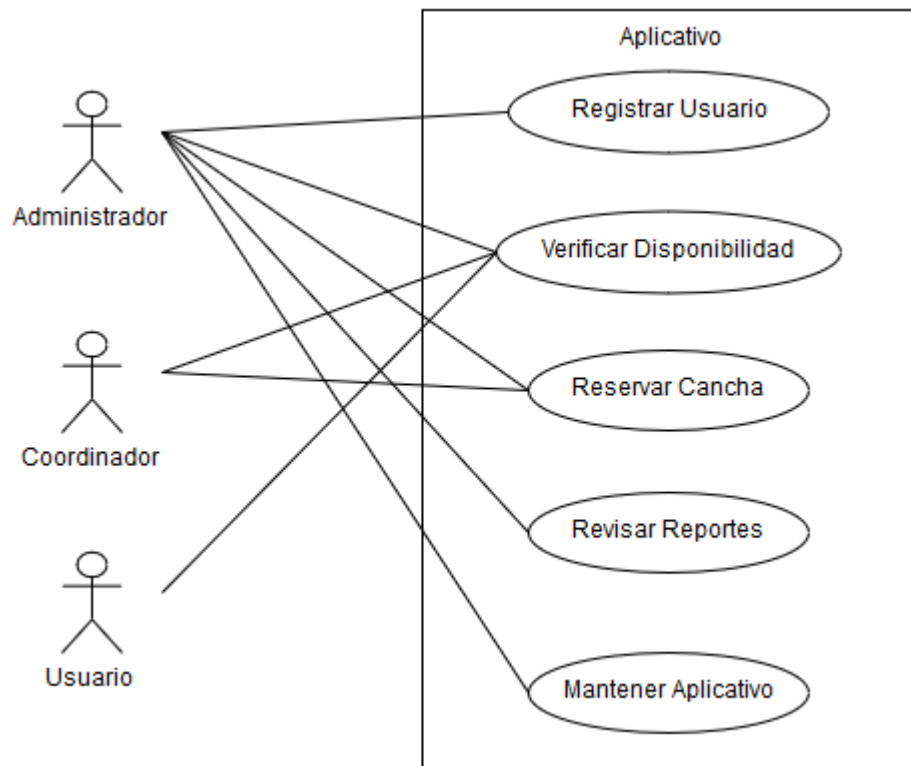


Figura 6: Diagrama de Casos de Uso General
Elaborado por: Pablo Castillo

En la figura 13 se muestra el diagrama completo de los casos de uso que utilicé para estructurar los respectivos requerimientos de la aplicación

CU001 Registrar usuario

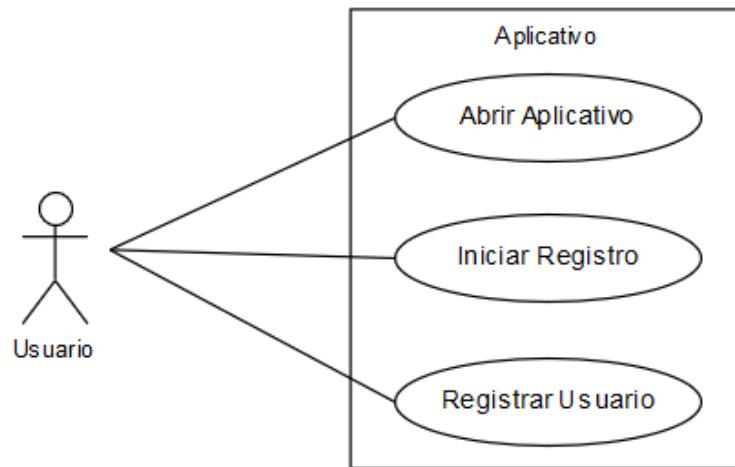


Figura 7: Caso de uso registrar usuario
Elaborado por: Pablo Castillo

Tabla 17. CU001 Registrar usuario

Caso de Uso	Registrar usuario
Actores	Usuarios normales.
Objetivo	Registrar un usuario en el aplicativo
Precondición	Ninguna
Descripción	El Usuario abre la aplicación. El usuario se dirige a la parte de la aplicación donde se puede registrar. El usuario ingresa los datos solicitados por la aplicación en un formulario. El usuario queda registrado

Fuente: Pablo Castillo

CU002 Registrar usuario por un administrador

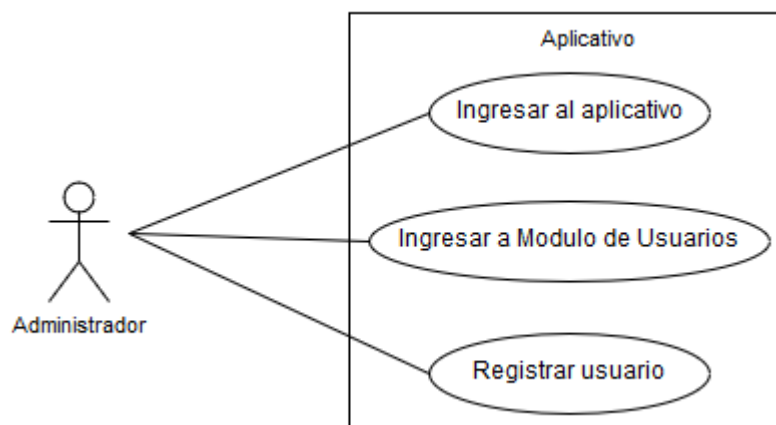


Figura 8: Caso de uso registrar usuario por un administrador
Elaborado por: Pablo Castillo

Tabla 18. CU002 Registrar usuario por un administrador

Caso de Uso	Registrar usuario por un administrador
Actores	Administrador, usuario normal.
Objetivo	Registrar un usuario en el aplicativo
Precondición	Ninguna
Descripción	El usuario administrador ingresa al aplicativo. Ingresa al módulo de mantenimiento. Ingresa los datos del usuario en el formulario de registro. Registra al usuario en el aplicativo (los datos se guardan en la base de datos).

Fuente: Pablo Castillo

CU003 Resetear clave de un usuario

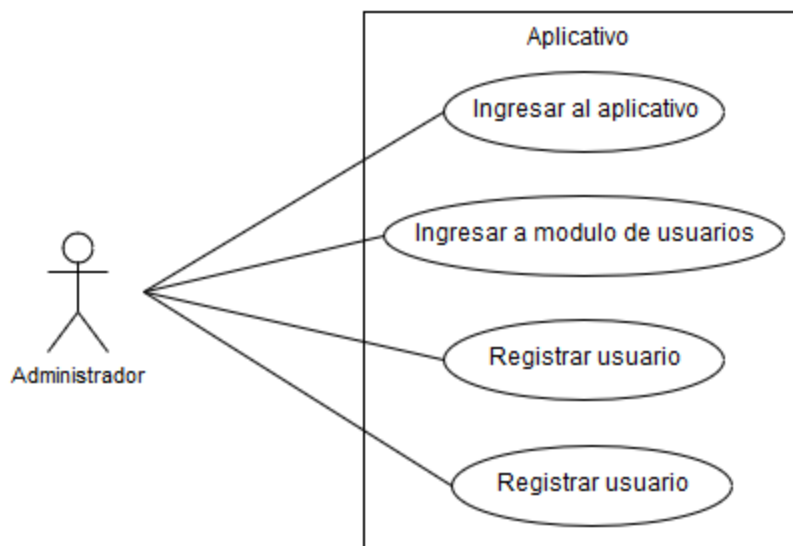


Figura 9: Caso de uso resetear clave de un usuario
Elaborado por: Pablo Castillo

Tabla 19. CU003 Resetear clave de un usuario

Caso de Uso	Resetear clave de usuario.
Actores	Administrador, coordinador, usuario.
Objetivo	Resetear la clave de un usuario que la olvido
Precondición	Que el usuario este registrado en el aplicativo
Descripción	El usuario administrador ingresa al aplicativo. Ingresa al módulo de mantenimiento. Selecciona al usuario que va a resetear la clave. Resetea la clave del usuario

Fuente: Pablo Castillo

CU004 Ver lista de usuarios registrados

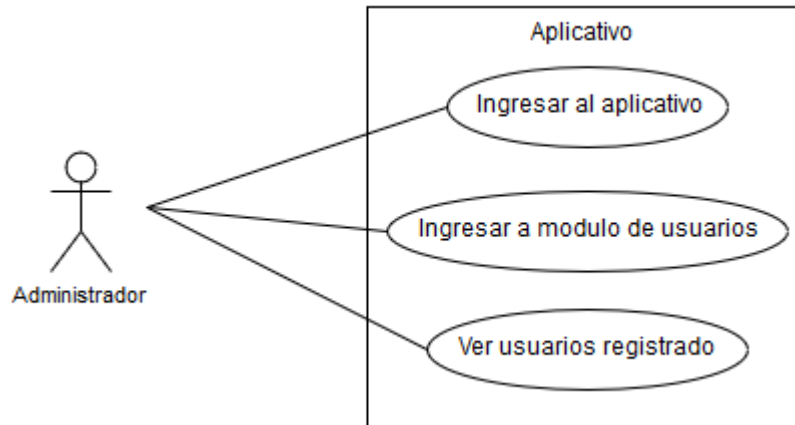


Figura 10: Caso de uso ver lista de usuarios registrados
Elaborado por: Pablo Castillo

Tabla 20. CU004 ver lista de usuarios registrados

Caso de Uso	Ver lista de usuarios registrados
Actores	Administrador.
Objetivo	Ver en una lista los usuarios registrados
Precondición	Que haya usuarios registrados en el aplicativo
Descripción	El administrador ingresa al aplicativo. Ingresa al módulo de mantenimiento, submodulo de usuarios. Observa la lista de usuarios registrados

Fuente: Pablo Castillo

CU005 Eliminar usuario

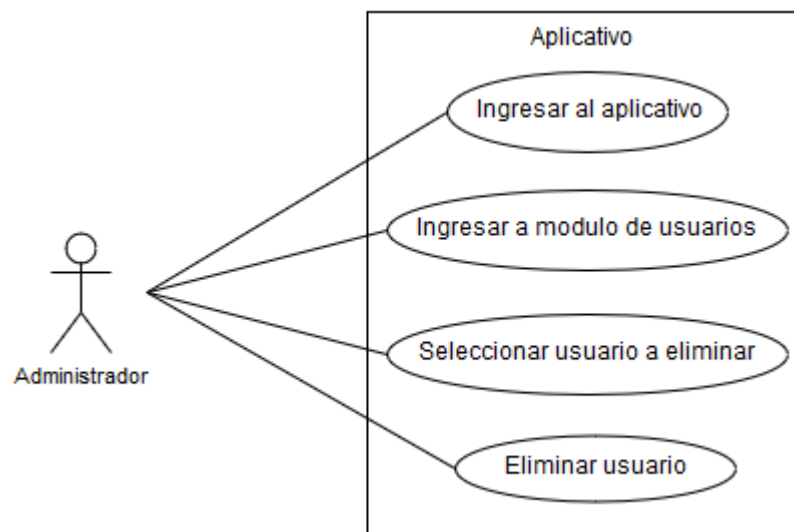


Figura 11: Caso de uso eliminar usuario
Elaborado por: Pablo Castillo

Tabla 21. CU005 eliminar usuario

Caso de Uso	Eliminar usuario
Actores	Administrador.
Objetivo	Eliminar un usuario del aplicativo
Precondición	Ninguna
Descripción	El Usuario administrador ingresa a la aplicación. Entra al módulo de mantenimiento submódulo de usuarios. Selecciona el usuario a eliminar. Elimina el usuario seleccionado.

Fuente: Pablo Castillo

CU006 Revisar lista de canchas

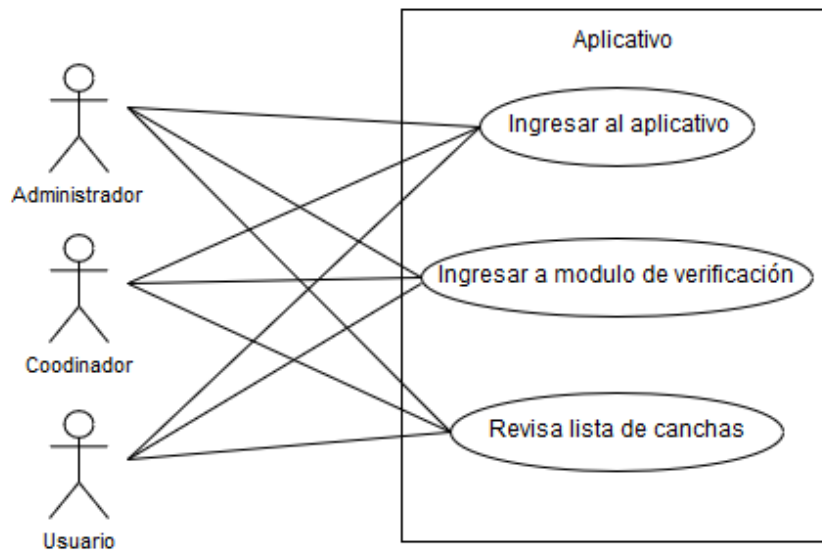


Figura 12: Caso de uso revisar lista de canchas
Elaborado por: Pablo Castillo

Tabla 22. CU006 revisar lista de canchas

Caso de Uso	Revisar lista de canchas
Actores	Usuarios normales.
Objetivo	Registrar un usuario en el aplicativo
Precondición	Ninguna
Descripción	El Usuario abre la aplicación. El usuario se dirige a la parte de la aplicación donde se puede registrar. El usuario ingresa los datos solicitados por la aplicación en un formulario. El usuario queda registrado

Fuente: Pablo Castillo

CU007 Revisar normativa de préstamo

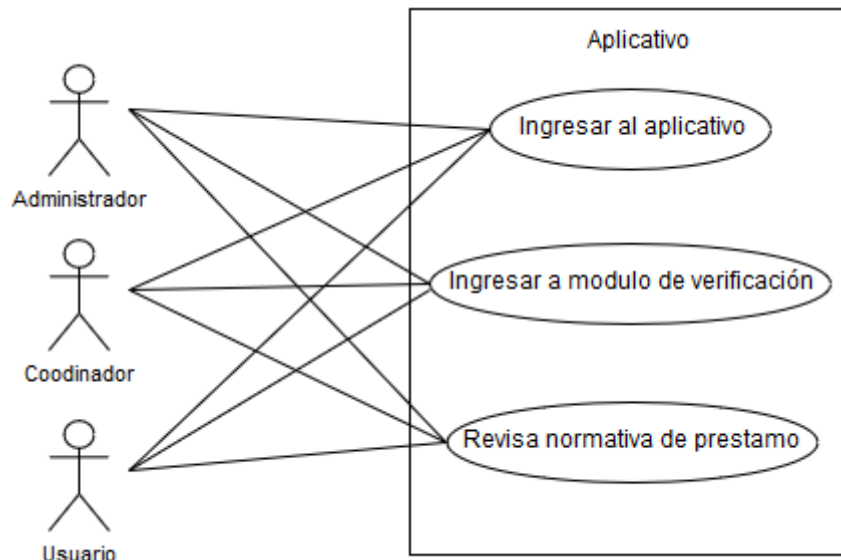


Figura 13: Caso de uso revisar normativa de préstamo
Elaborado por: Pablo Castillo

Tabla 23. CU007 revisar normativa de préstamo

Caso de Uso	Revisar normativas de préstamo
Actores	Administrador, coordinador, usuarios normales.
Objetivo	Revisar normativa de préstamo de las canchas
Precondición	Ninguna
Descripción	EL usuario ingresa a la aplicación, va al módulo de verificación, revisa las normativas de préstamos de las instalaciones

Fuente: Pablo Castillo

CU008 Comprobar disponibilidad de canchas

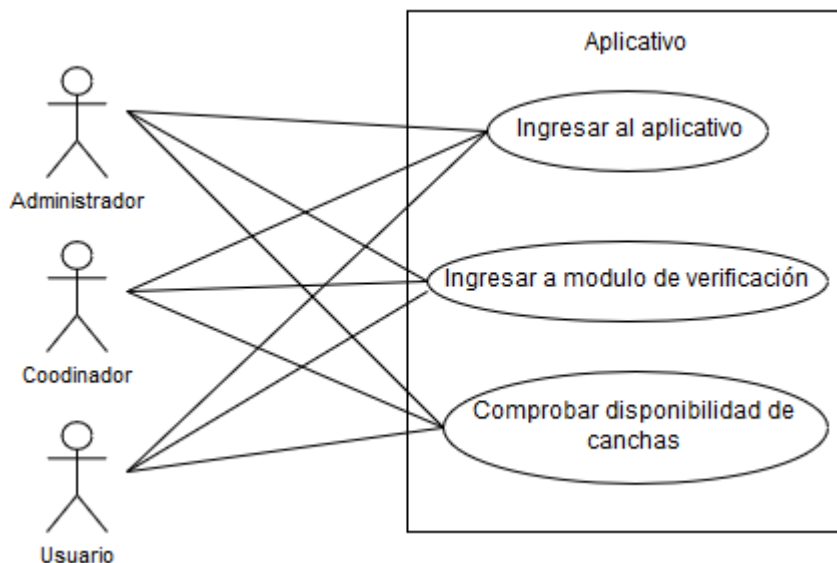


Figura 14: Caso de uso comprobar disponibilidad de canchas
Elaborado por: Pablo Castillo

Tabla 24. CU008 comprobar disponibilidad de canchas

Caso de Uso	Comprobar disponibilidad de las canchas
Actores	Administrador, coordinador, usuarios normales.
Objetivo	Comprobar la disponibilidad de las canchas
Precondición	Ninguna
Descripción	El usuario abre la aplicación. Ingresa al módulo de verificación. Ingresa al calendario para verificar la disponibilidad de las canchas para poder reservar la que necesite.

Fuente: Pablo Castillo

CO009 Reservar una cancha

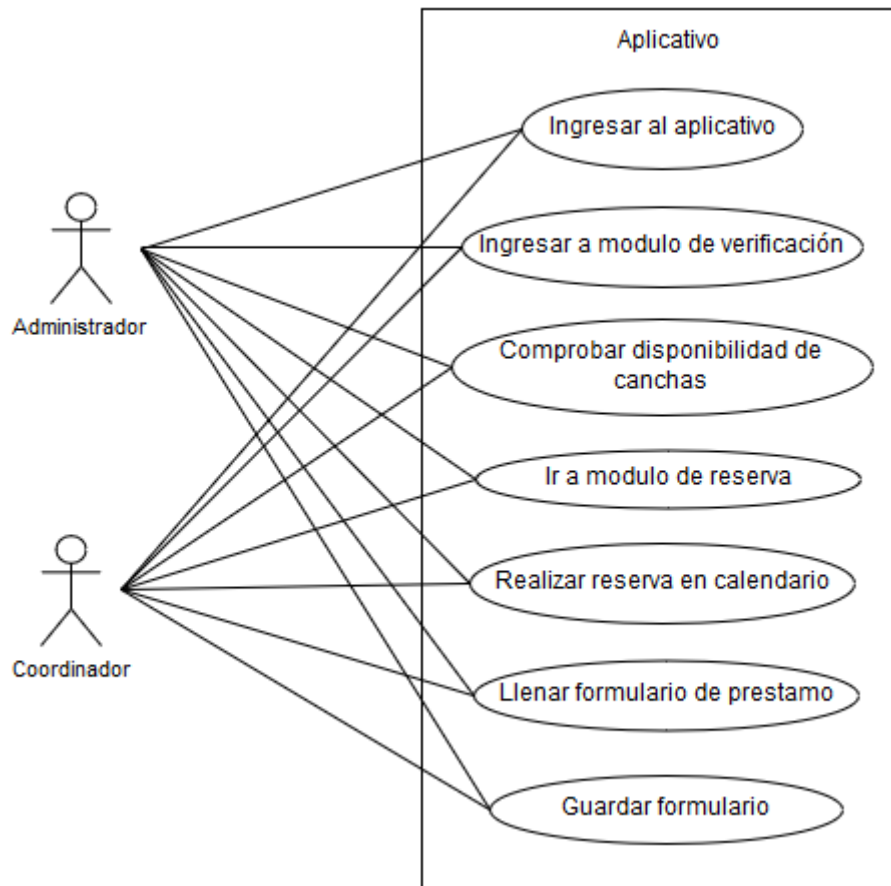


Figura 15: Caso de uso reservar una cancha
Elaborado por: Pablo Castillo

Tabla 25. CU009 reservar una cancha

Caso de Uso	Reservar cancha
Actores	Administrador, coordinador.
Objetivo	Reservar una cancha
Precondición	Que la cancha esté disponible
Descripción	El usuario abre la aplicación. Verifica la disponibilidad de la cancha en el módulo de verificación. Va al módulo de reserva. Realiza la reserva en el calendario. Llena el formulario de préstamo de la cancha. Guarda el formulario

Fuente: Pablo Castillo

CU010 Revisar reportes

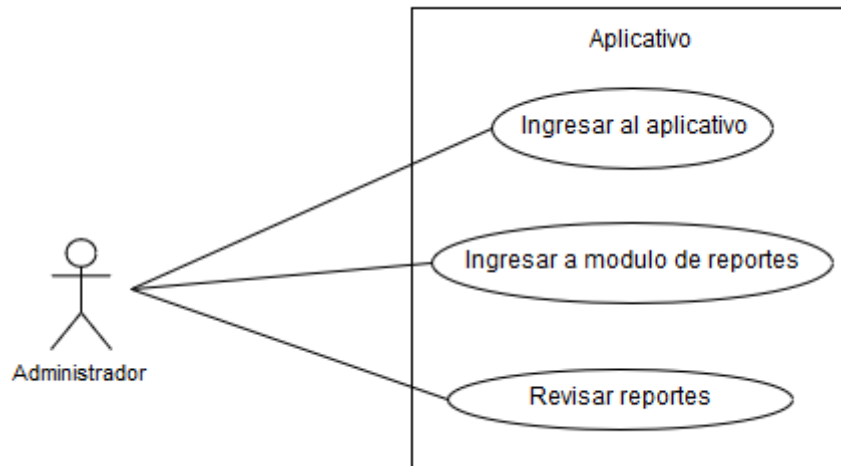


Figura 16: Caso de uso revisar reportes
Elaborado por: Pablo Castillo

Tabla 26. CU010 revisar reportes

Caso de Uso	Revisar reportes
Actores	Administrador.
Objetivo	Revisar los reportes que da la aplicación
Precondición	Ninguna
Descripción	El usuario abre la aplicación. Va al módulo de reportes. Revisa los reportes que la aplicación le ofrece.

Fuente: Pablo Castillo

CU011 Guardar Reportes

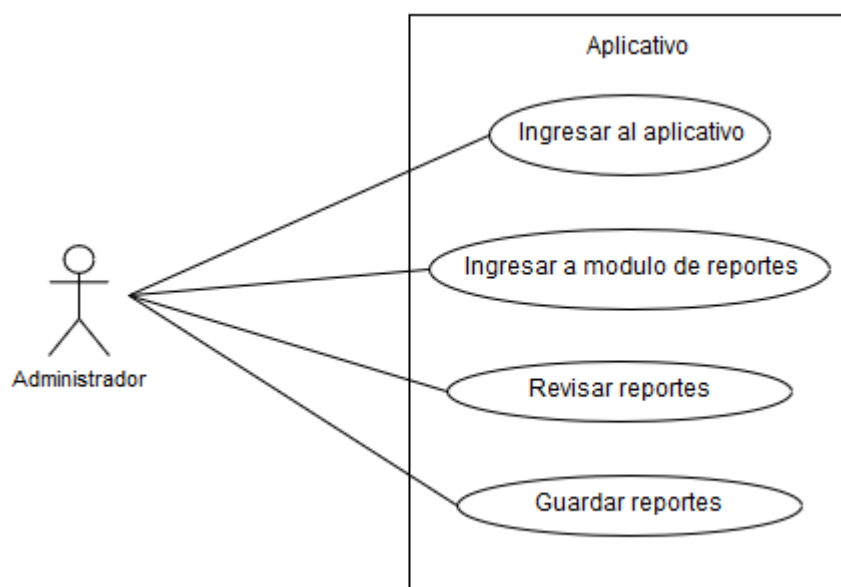


Figura 17: Caso de uso guardar reportes
Elaborado por: Pablo Castillo

Tabla 27. CU011 guardar reportes

Caso de Uso	Revisar reportes
Actores	Administrador.
Objetivo	Revisar los reportes que da la aplicación
Precondición	Ninguna
Descripción	El usuario abre la aplicación. Va al módulo de reportes. Revisa los reportes que la aplicación le ofrece. El usuario guarda los reportes que necesite guardar después de revisarlos.

Fuente: Pablo Castillo

CU012 Crear nueva cancha

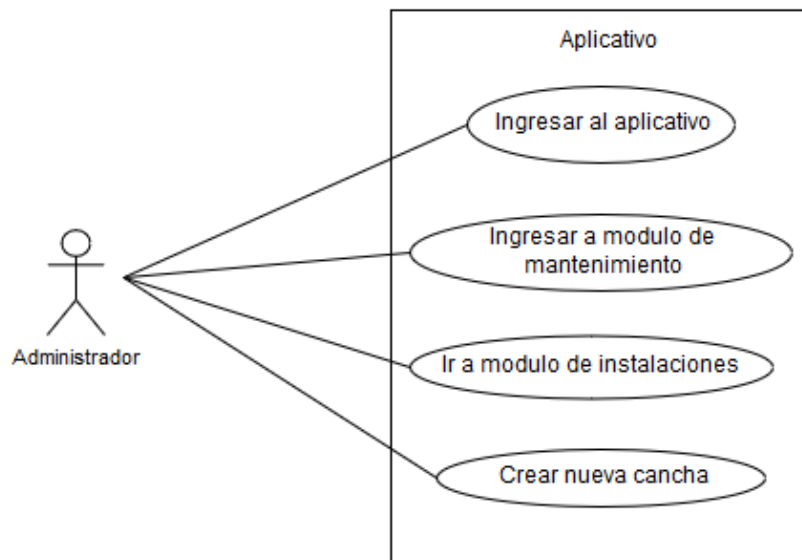


Figura 18: Caso de uso crear nueva cancha
Elaborado por: Pablo Castillo

Tabla 28 CU012 crear nueva cancha

Caso de Uso	Crear nueva cancha
Actores	Administrador.
Objetivo	Crear nueva cancha en la aplicación
Precondición	Ninguna
Descripción	El usuario ingresa a la aplicación. Ingresa al módulo de mantenimiento. Ir al módulo de instalaciones. Crear una nueva instalación.

Fuente: Pablo Castillo

CU013 Renombrar cancha

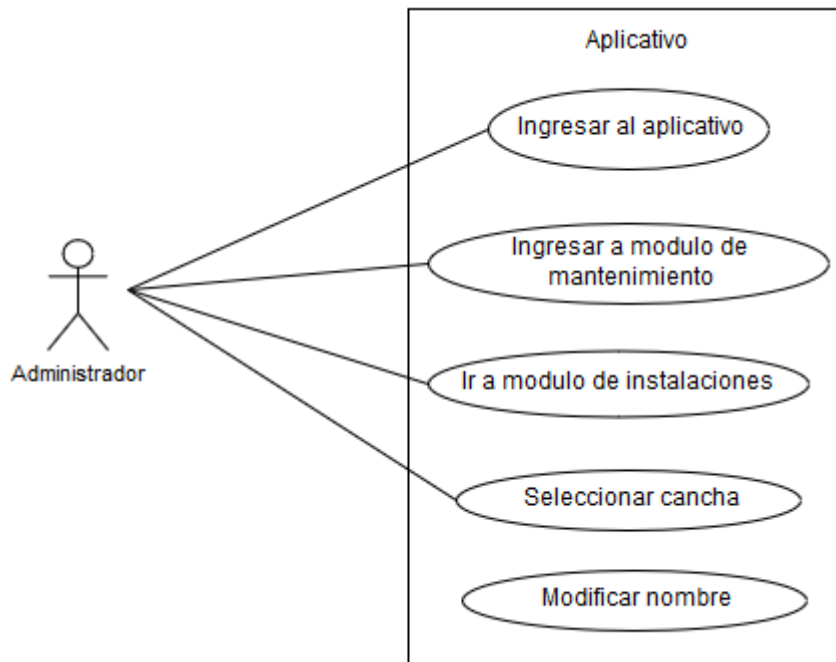


Figura 19: Caso de uso renombrar cancha
Elaborado por: Pablo Castillo

Tabla 29: CU013 renombrar cancha

Caso de Uso	Renombrar cancha
Actores	Administrador.
Objetivo	Renombrar el nombre de una cancha
Precondición	Ninguna
Descripción	El usuario ingresa a la aplicación. Ingresar al módulo de mantenimiento. Ingresar al módulo de las instalaciones. Selecciona la cancha a la que se desea renombrar. Renombrar la cancha

Fuente: Pablo Castillo

CU014 Modificar capacidad de usuarios por cancha

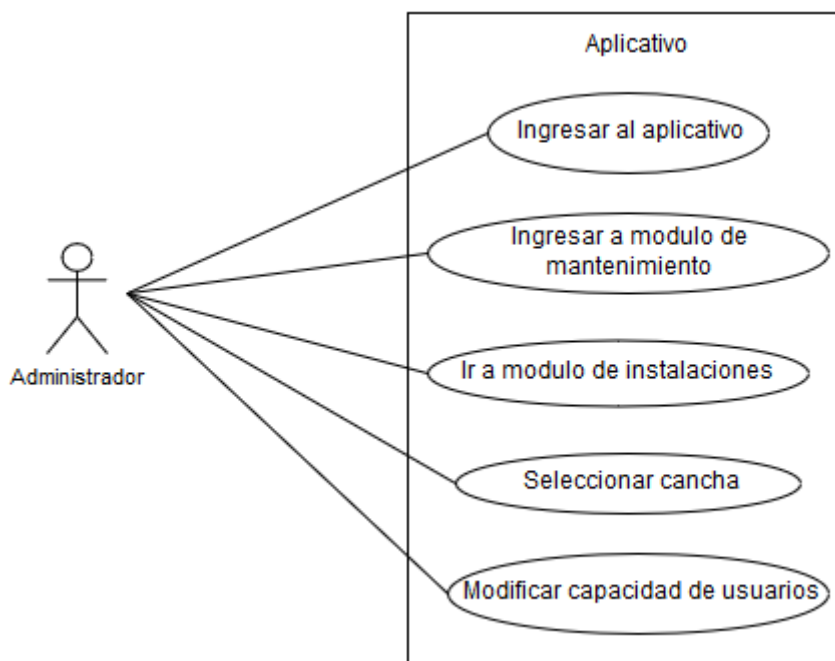


Figura 20: Caso de uso modificar capacidad de usuarios por cancha
Elaborado por: Pablo Castillo

Tabla 30: CU014 modificar capacidad de usuarios por cancha

Caso de Uso	Modificar capacidad de usuarios por cancha
Actores	Administrador.
Objetivo	Modificar la capacidad de usuarios de una cancha
Precondición	Ninguna
Descripción	El usuario ingresa al aplicativo. Ingresa al módulo de mantenimiento. Ingresa al módulo de instalaciones. Selecciona la cancha a la que desea modificar la capacidad de usuarios que admite esa cancha. Modifica la capacidad de usuarios.

Fuente: Pablo Castillo

Artefactos Fase 2

Especificación de requerimientos

PAGINA 23

Especificación de Casos de Uso

PAGINA 30

2.4. Fase de desarrollo.

En la presente fase se desarrolla la aplicación de acuerdo a la arquitectura planteada, siguiendo los lineamientos hasta que se termine la aplicación y así ser entregada al usuario. Es posible que se den cambios en los diagramas en el proceso de desarrollo. En base a las herramientas utilizadas se implementará la arquitectura. Se realiza también un manual de programador. Este manual tiene información específica referente al aplicativo. Los archivos de funcionamiento de la aplicación se hallan en la carpeta UTPL específicamente en la carpeta build. Algunos de los archivos más importantes son los siguientes:

- ψ Conexion.class que contiene la cadena de conexión a la base de datos, dentro del código de este archivo se encuentra el driver de conexión nuestra base de datos MySql.
- ψ Index.jsp que es el archivo con que se empieza a utilizar la aplicación, es aquí donde se ingresa a la aplicación.
- ψ Principal.jsp es aquí donde se muestran los módulos de la aplicación.
- ψ Reservas.jsp donde podemos realizar nuestras reservas en el calendario de la aplicación, este calendario se presta en forma de grilla.
- ψ Pedido.jsp es el archivo del formulario principal por el cual se ingresa los datos de los usuarios involucrados en la solicitud de una instalación, así como el tiempo que se utilizara las instalaciones, y si fuera el caso los materiales que serán prestados.....
- ψ Reportes.jsp, este es un archivo que muestra todos los reportes disponibles, los cuales brindan información para poder realizar estadísticas de los préstamos así como el número de personas que utilizan las instalaciones, ya sean estudiantes, docentes o personal administrativo
- ψ registrarse.jsp, en este pequeño formulario es donde se ingresa la información de los usuarios que se registra en el sistema, de igual manera toda esta información se guarda en la base de datos de la aplicación.

2.4.1. Arquitectura de desarrollo - MVC.

Es importante mencionar que este patrón se acopla a la metodología de desarrollo utilizada en este trabajo, con el fin de obtener una base teórica para poder plantear la arquitectura y asegurar el cumplimiento de las funciones esperadas por el usuario.

Patrón arquitectónico de la familia de estilos arquitectónicos de Llamada y Retorno, separa el modelado del dominio, la presentación y las acciones basadas en datos ingresados por el usuario.

El modelo, administra el comportamiento y los datos del dominio de la aplicación, responde a requerimientos de información sobre su estado (usualmente formulados desde la vista) y responde a instrucciones de cambiar de estado (habitualmente desde el controlador).

La Vista maneja la visualización de la información.

El Controlador, interpreta las acciones de los dispositivos (ejemplo: mouse y teclado), informando al modelo y/o a la vista que cambien de estado según resulte apropiado.

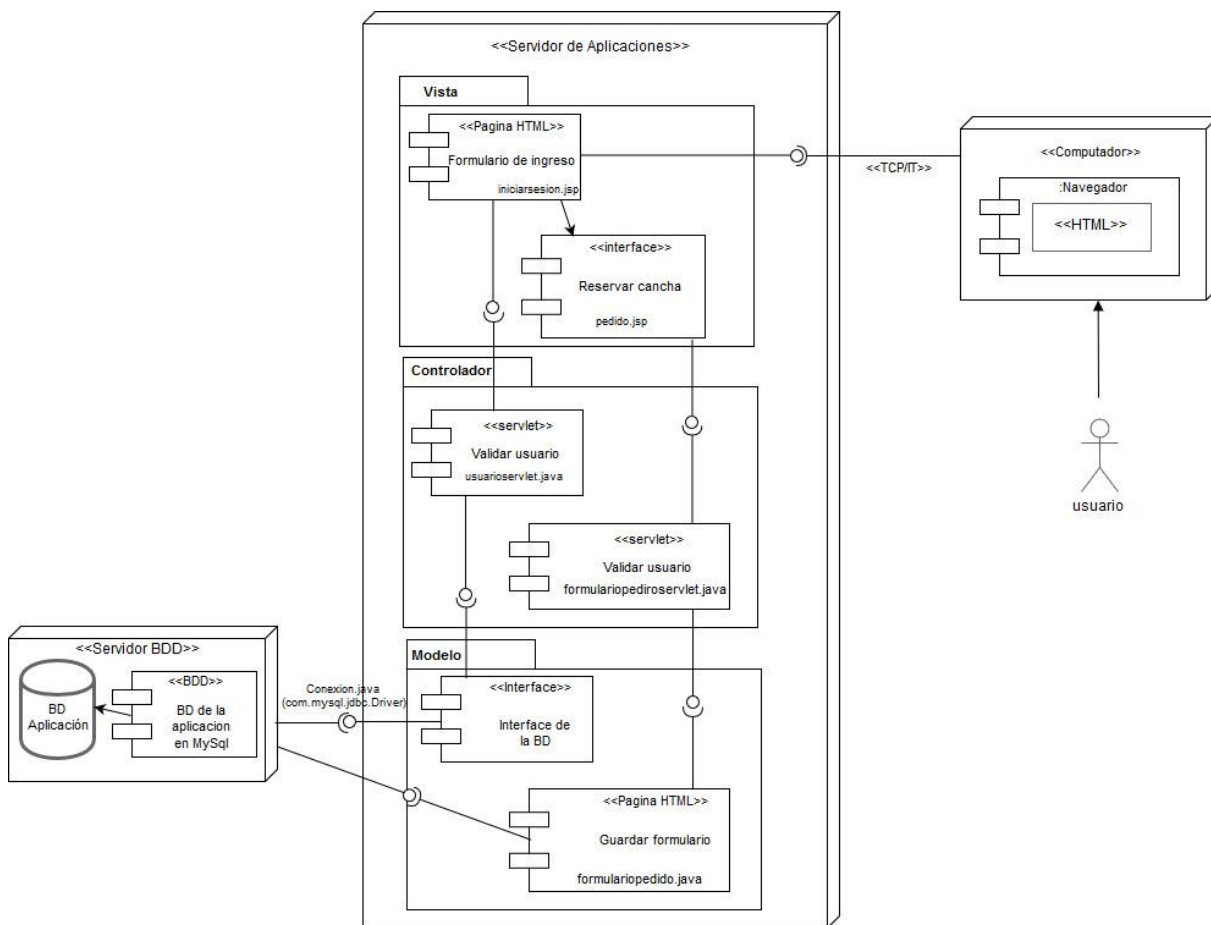


Figura 21: Arquitectura de la aplicación (Flujo de información)
Elaborado por: Pablo Castillo

Con este patrón arquitectónico la vista y el controlador dependen del modelo, el cual no depende de las anteriores.

Arquitectura MVC (Modelo-Vista-Controlador) para el Sistema de Reservas de Canchas UTPL

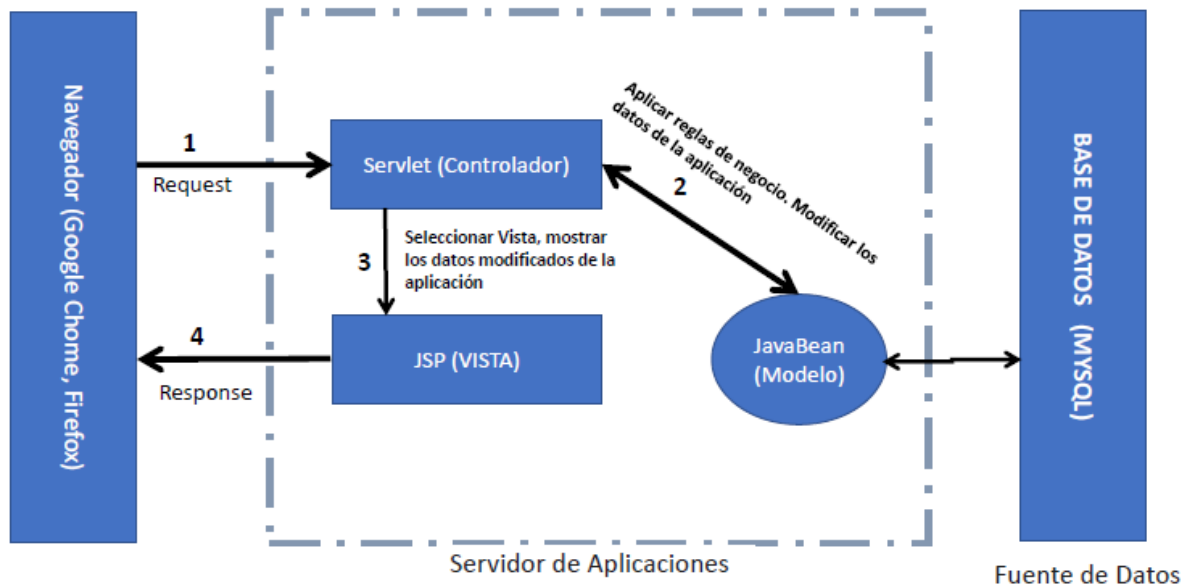


Figura 22: Arquitectura de la aplicación
Elaborado por: Pablo Castillo

2.4.1.1. Explicación del patrón arquitectónico MVC

2.4.1.1.1. Modelo.

En este componente se encapsula la lógica de negocio, se encarga de la persistencia e interacción con la base de datos trabajando mayormente con los select, insert, update; guardando el estado de la aplicación.

En la siguiente figura encontramos los archivos que forman parte de este componente, en estos archivos beans encontramos los métodos get y set que devuelven y actualizan valores. Por ejemplo el bean ReservasCalendar.java que tiene los métodos get para actualizar la información que ingresa por los campos del formulario de reserva.

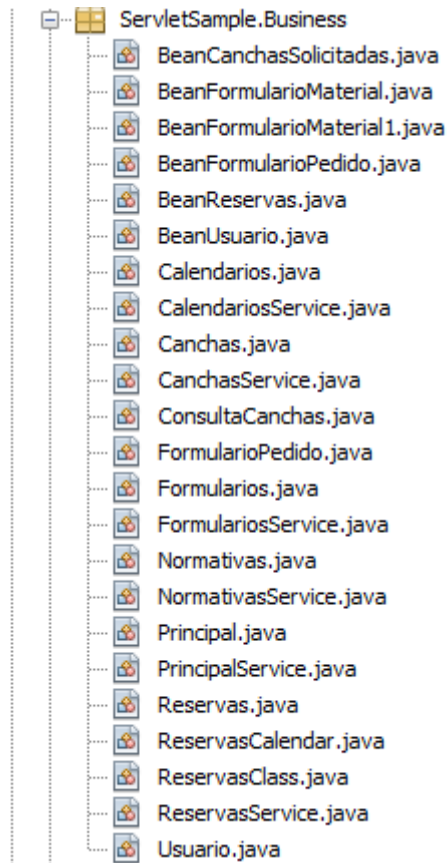


Figura 23: Archivos de la capa de modelo.
Elaborado por: Pablo Castillo

2.4.1.1.2. Vista.

Esta capa se encarga del código de la aplicación que va a crear la visualización de la interfaz del usuario, en esta capa tenemos los códigos HTML que muestran la salida de datos, aquí no se modifican los datos, solo se los consulta para visualizar los resultados.

En esta capa encontramos los siguientes archivos.

Por ejemplo tenemos el archivo `iniciarsesion.jsp` que muestra el formulario de ingreso a la aplicación.

La siguiente figura muestra los archivos de la capa vista.

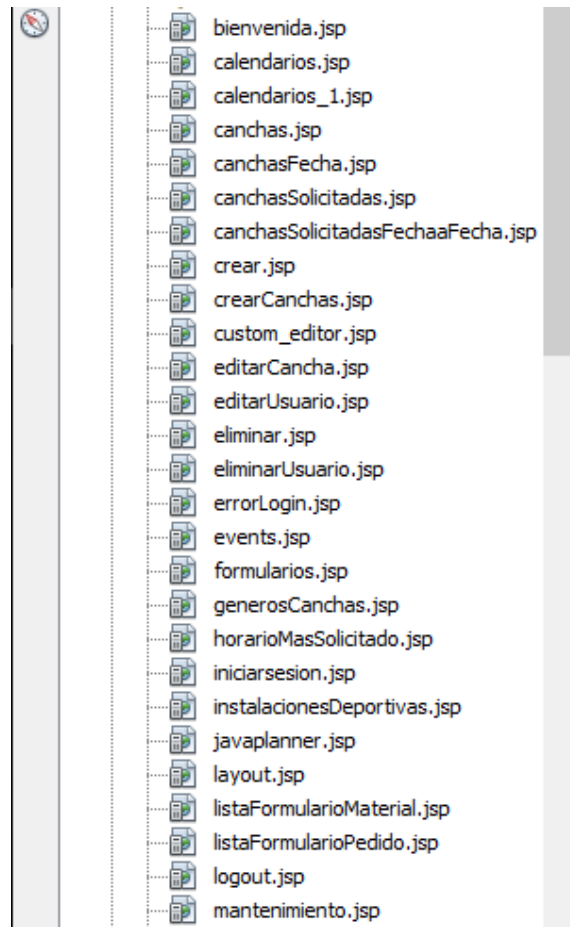


Figura 24: Archivos de la capa vista.
Elaborado por: Pablo Castillo

2.4.1.1.3. Controlador.

Esta capa incluye el código necesario para contestar a las acciones que se solicitan a la aplicación, esta capa se utiliza de enlace entre la capa de vista y modelo.

En esta capa encontramos los servlets que solicitan recursos – en este caso - de la aplicación, por ejemplo el servlet FormularioPedidoServlet.java tiene el método RequestDispatcher invoca a los archivos de la capa vista cuando se termina de llenar el formulario de reserva, para presentarlos al usuario. En la siguiente figura observamos los archivos de la capa controlador.

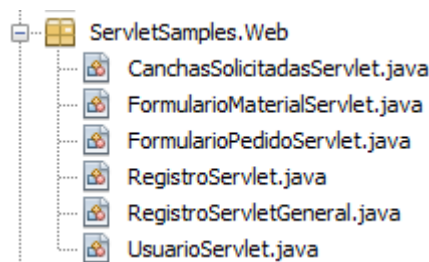


Figura 25: Archivos de la capa controlador.
Elaborado por: Pablo Castillo

Para poder representar la arquitectura de la aplicación se mostrara las vistas de casos de uso, la lógica que tiene la aplicación que se presentaran en lenguaje UML.

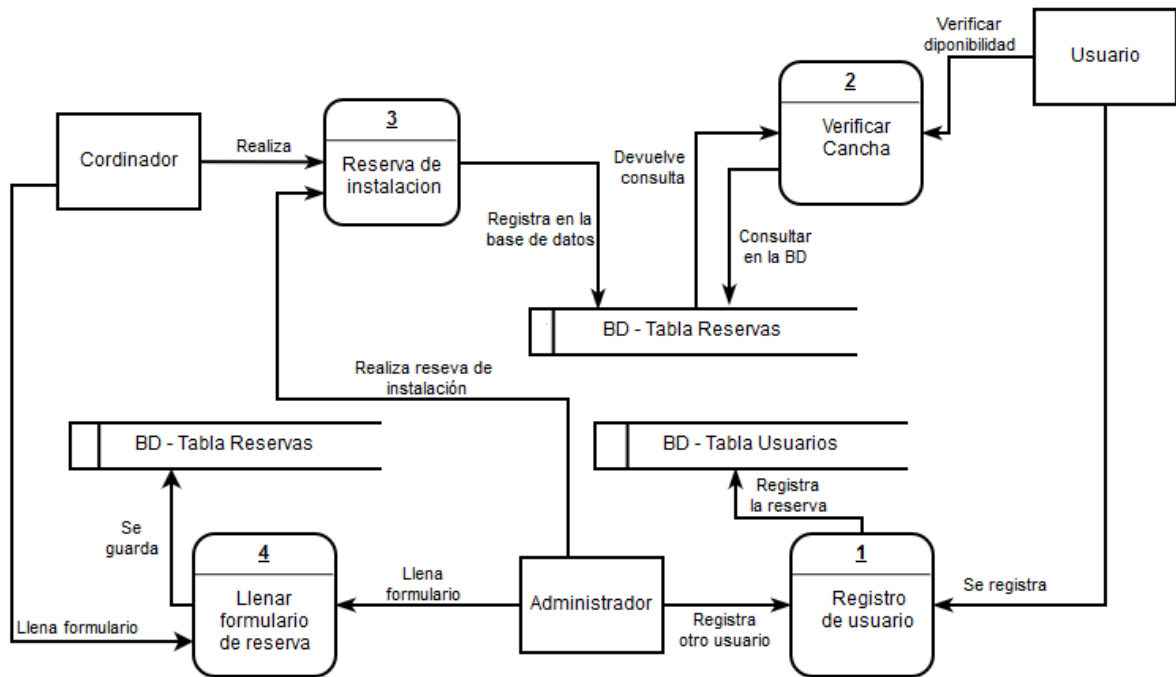


Figura 26: Diagrama de flujo
Autor: Pablo Castillo

Por medio del presente diagrama de flujo podemos observar el respectivo flujo de la aplicación, y nos ayuda a identificas que tipo de arquitectura es la que necesitamos utilizar, para que la interacción con el usuario sea la mejor.

Para el desarrollo de esta aplicación se utiliza el patrón arquitectónico MVC es patrón arquitectónico de la familia de estilos arquitectónicos de Llamada y Retorno, este estilo arquitectónico enfatiza la modificabilidad y la escalabilidad

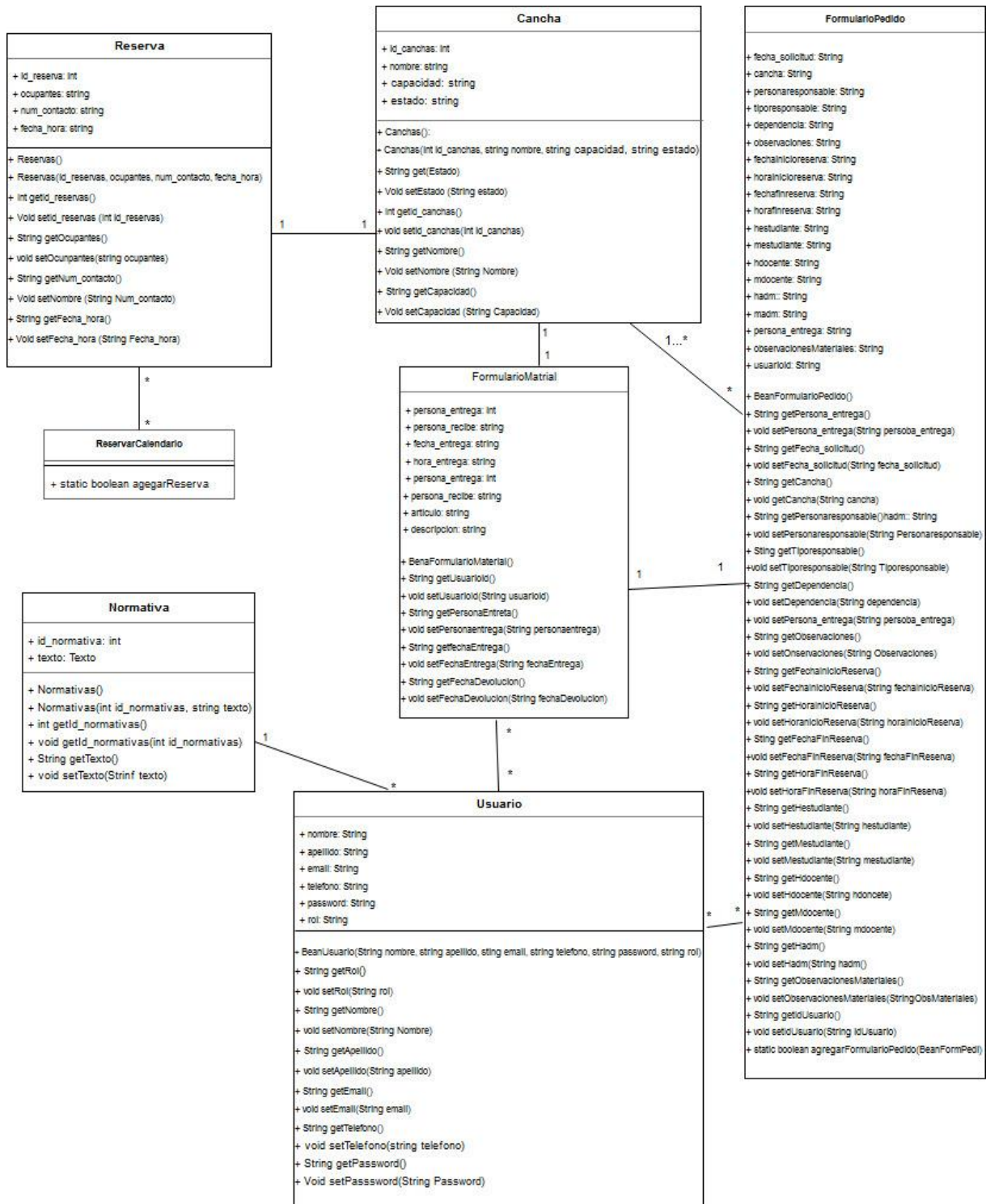


Figura 27: Diagrama de Clases
 Autor: Pablo Castillo

Podemos observar en el diagrama anterior, las clases que interactúan y la estructura de las relaciones, esto nos ayuda a especificar las respectivas clases y posteriormente tablas que vamos a necesitar en nuestro desarrollo de la aplicación. Además cabe mencionar que este es uno de los artefactos que requiere la metodología que se solicita en la fase de construcción.

Podemos observar en la figura 21 cómo funciona la aplicación, podemos observar que los usuarios (administradores y usuarios normales) envían peticiones en formato HTML al servidor a través de la red, estas peticiones llegan al servidor Web, este a su vez procesa dicha petición e interactúa con la base de datos, ya sea para guardar la información que se envía a través de los formularios, o para sacar información en este caso para la presentación de los reportes, y esta solicitud una vez procesada es devuelta al usuario.

2.4.2. Limitaciones.

Tabla 31: Requerimientos de hardware

Características del Hardware	
Procesador:	Mínimo Corei 3 o Corei 5
Memoria RAM:	Mínimo 4 GB
Tarjeta de Red:	100/1000 GB Ethernet
Disco Duro:	Mínimo 500 GB

Es importante mencionar en esta fase a la arquitectura física donde se aloja y se ejecuta la aplicación, este alojamiento se hace en un servidor físico ya que depende de las características que tenga en este caso el equipo donde ira alojada la aplicación para tener en cuenta las conexiones y las peticiones que acepte la aplicación. En la tabla anteriormente mencionada se muestra los requerimientos mínimos para la puesta en marcha del aplicativo: podemos decir que como parte de la arquitectura física en la parte de la red se utilizar la red misma de la universidad.

2.4.3. Base de datos.

Después del análisis de los módulos necesarios, revisando y definiendo los casos de uso, se obtiene la información necesaria para el diseño de la base de datos con sus respectivas tablas y sus atributos relaciones.

Es en el diseño de la base de datos donde se realiza el proceso de definición y creación de la base de datos, se estable los tipos de datos para cada tabla; de este diseño se implementara el desarrollo de la aplicación, y también servirá como repositorio de la información que se ingrese por los respectivos formularios.

La base de datos del sistema para asignación de áreas de la UTPL considera las diferentes variables como sigue:

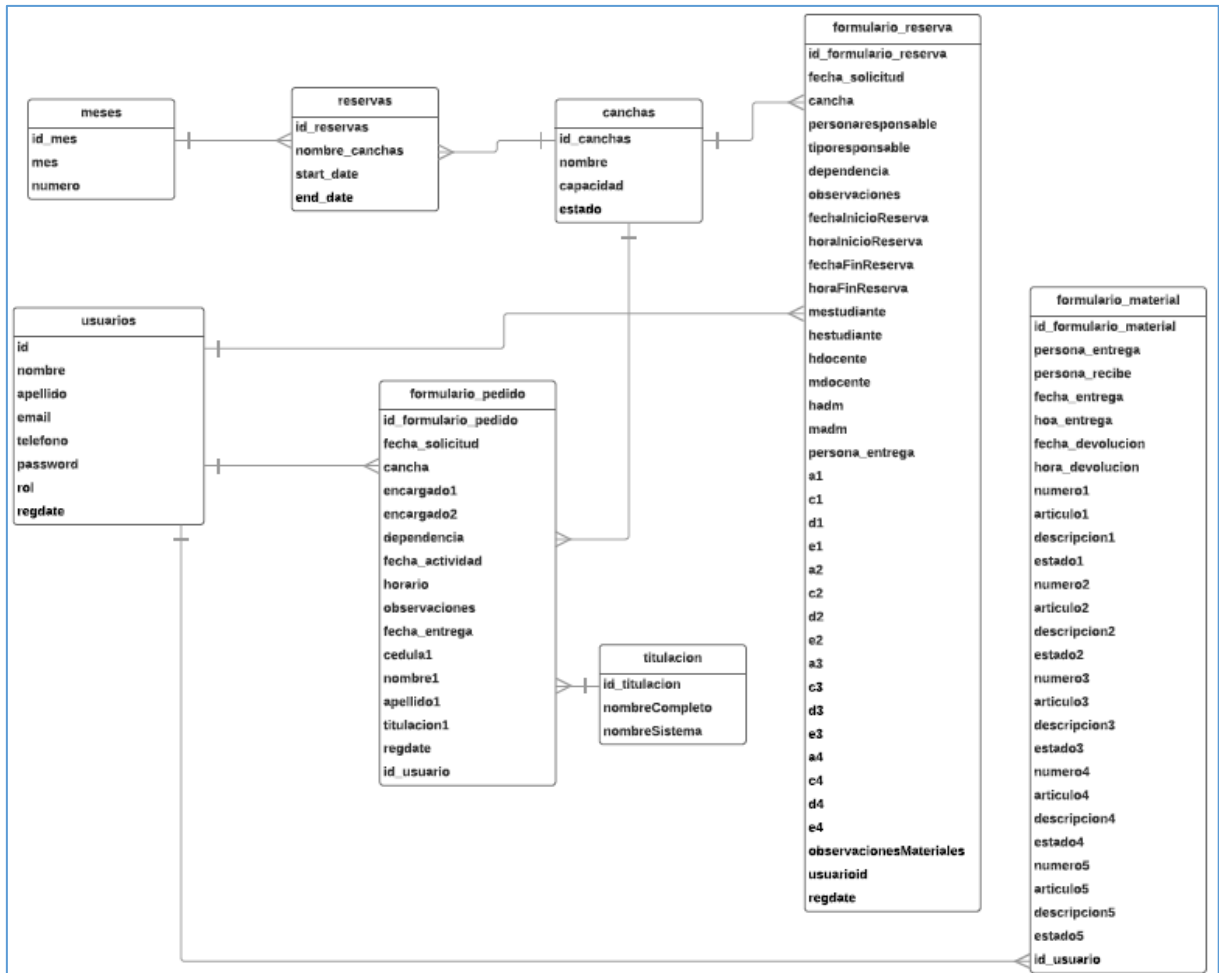


Figura 28: Variables y atributos de las tablas de la base de datos (Diagrama UML) E-R

Elaborado por: Pablo Castillo

En la presente figura se puede observar los atributos de las tablas de la base de datos, la cual nos ayuda a ver la estructura de la base de datos en la que se almacena la información que ingresa por medio de las reservas realizadas por los usuarios de la aplicación.

2.4.4. Herramientas utilizadas.

Para la implementación del sistema se ha utilizado:

Lenguajes de programación:

- JSP para el backend
- JQuery (biblioteca multiplataforma de Java script de JAVA)
- Bootstrap framework de desarrollo para el diseño web para el front end
- HTML5 se utiliza para la maquetación de las páginas

Gestor de base de datos

- Mysql como motor de base de datos
- WampServer administrador de la base de datos

Servidor web

- Apache Tomcat contenedor de los servlets y sirve para la implementación de la JSP Entorno de desarrollo

- Netbeans IDE

El sistema desarrollado es multinavegador y multiplataforma, los requerimientos del sistema para la aplicación son:

- Java instalado en servidor web
- Mysql instalado en el servidor web
- Aplicación web en servidor
- Herramienta para los diagramas UML
 - Se utilizó la aplicación para realizar diagramas StartUML

2.4.5. Módulos.

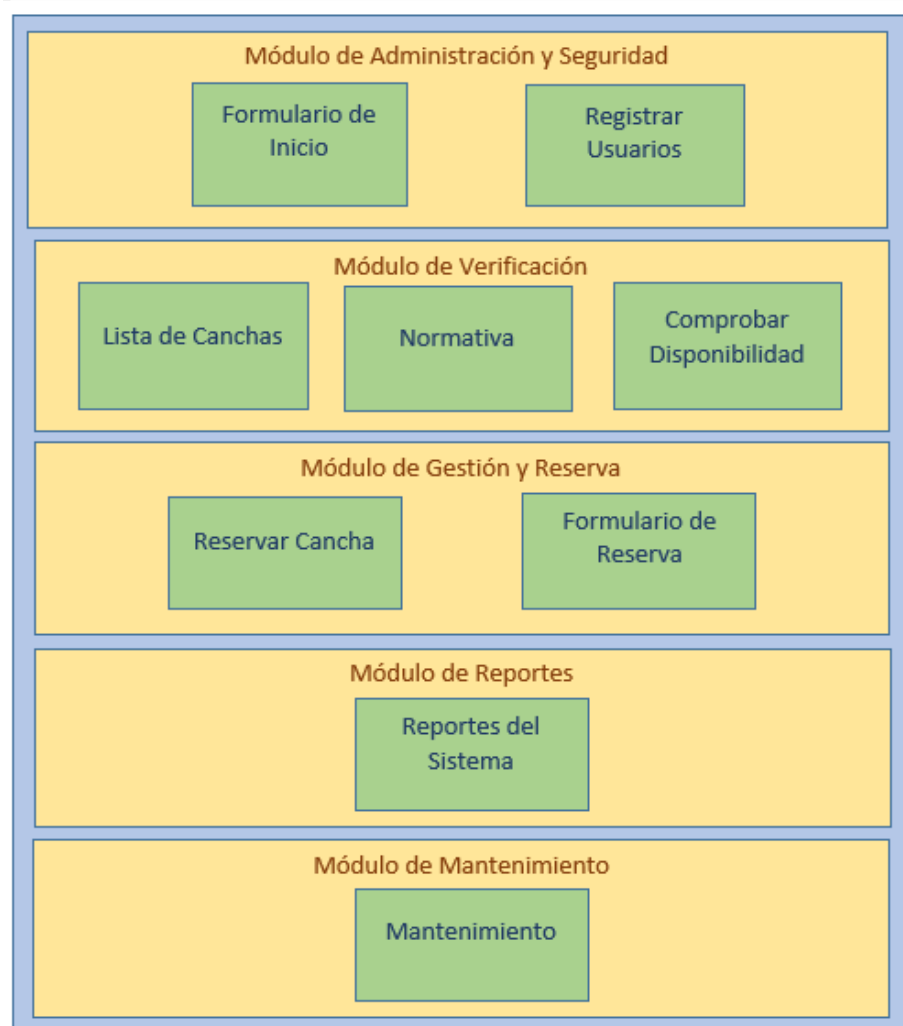


Figura 29: Módulos del sistema
Elaborado por: Pablo Castillo

De acuerdo a las necesidades del área y las funciones del sistema el diagrama de la figura 7, donde se detallan los principales módulos y sub-módulos:

2.4.6. Interacción de procesos.

Podemos ver en cada módulo las respectivas interacciones que se tienen entre ellos, en cada uno de los procesos y funcionalidades que tiene la aplicación.

Al hablar de interacción observamos cómo estas afectan la experiencia del usuario, ya que se puede evidenciar como los requerimientos van cumpliéndose.

Módulo de Administración y Seguridad

Este módulo es el que permite ingresar a la aplicación y que el usuario pueda quedar registrado, además permite asignar los respectivos roles a los usuarios para que la aplicación solamente pueda ser modificado en su aspecto interno por el administrador.

En este caso específicamente un usuario con rol de administrador es el encargado de llevar a cabo la administración de la aplicación, ya que es el único que delega los roles a los usuarios registrados la base de datos. Además el mantenimiento de la aplicación es realizada por el usuario administrador, los roles de usuario y coordinador no pueden realizar esta tarea. En cuanto a la seguridad podemos mencionar que la información almacenada en la base de datos (MySQL) no está disponible para todos los usuarios, los usuarios administradores son los únicos que pueden tener acceso a esta información. Cada usuario registrado en la aplicación tiene un rol asignado – administrador, coordinador, usuario.

Módulo de verificación

Este módulo se permite a todos los usuarios verificar cuáles son las instalaciones que están reservadas y sus horarios de disponibilidad, puede ser a través de un listado, con el calendarios donde ese muestran las reservas realizadas, organizadas por día, semana o mes, una vez verificado el estado de las canchas en este módulo, se pasa al siguiente modulo que es el de reservas para poder realizar la reserva. En este módulo hay el sub-módulo correspondiente a la presentación de las Normativas de uso de las instalaciones, donde se encuentran cuáles son los requisitos de uso de instalaciones y de las canchas, los cuidados que se debe tener al usarlas, como se debe utilizar los materiales que la universidad presta a las personas que van a utilizar las canchas, la responsabilidad de los solicitantes así como acciones en caso de incumplimiento de estos requisitos. Estas normativas se encuentran en formato pdf.

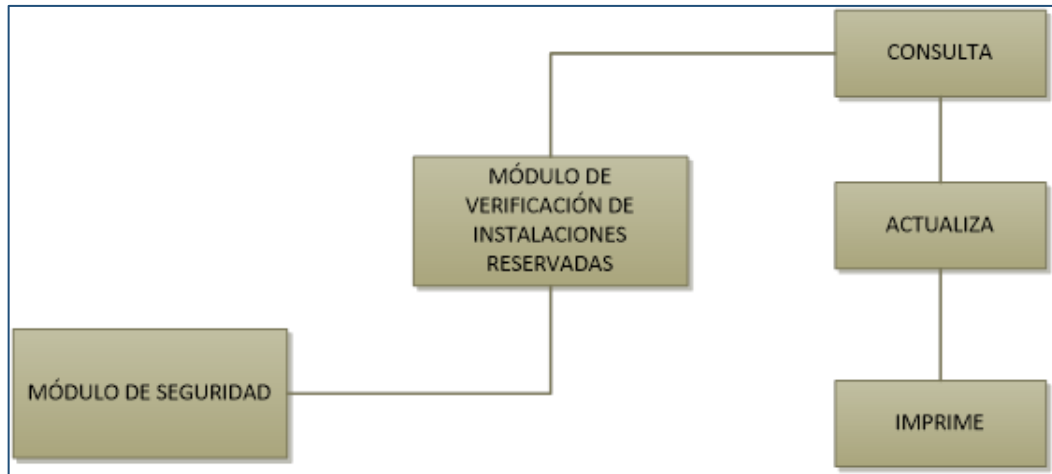


Figura 30: Módulo de Verificación
 Elaborado por: Pablo Castillo

Módulo de gestión y reserva

Éste módulo podría considerarse uno de los más importantes y corresponde al módulo central, que se encarga de reservar una instalación específica, quedando registrada en los reportes para que no sea utilizada nuevamente o existan confusiones de espacios tiempo otorgados a otros usuarios. Aquí se encuentran los sub-módulos que se necesita para completar una reserva y son; llenar en el calendario de la aplicación que es donde marcamos y especificamos la reserva de una instalación, la fecha y el tiempo que se va a utilizarla; y también se llena el formulario de préstamo de la instalación, que es donde marcamos todos los datos importantes para la verificación de la información que a la vez sirve al departamento de misiones para futuros controles, es aquí donde se recoge la información de las reservas realizadas en la aplicación.

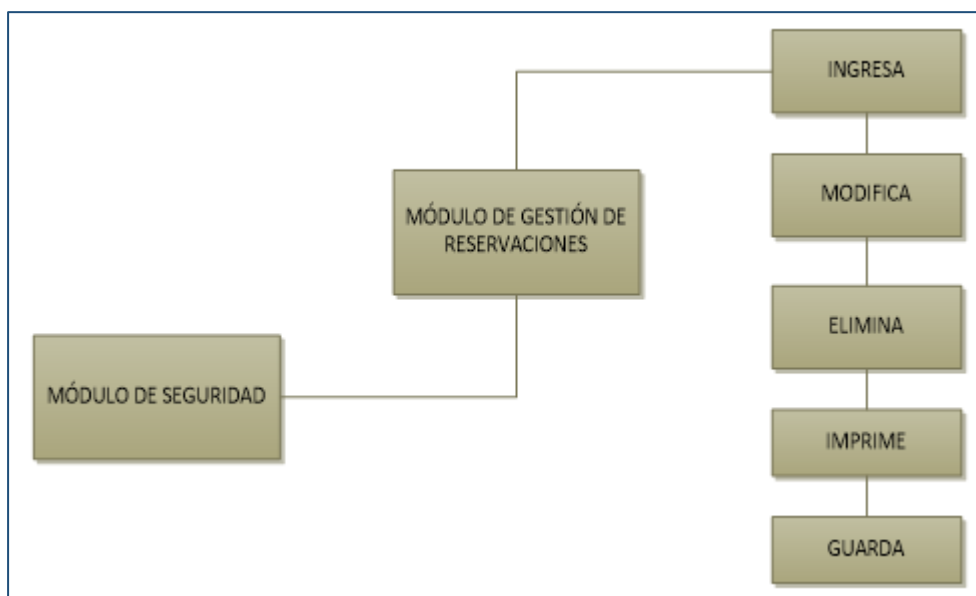


Figura 31: Módulo de gestión y reserva
 Elaborado por: Pablo Castillo

Módulo de Reportes

Una vez que se ha ingresado la información en el módulo de reserva mencionado anteriormente y a su vez almacenada en la base de datos de la aplicación, podemos utilizar el módulo de reportes. La información que brinda este módulo a los administradores es variada y además de proveer datos significativos que sirven para la toma de decisiones, evaluación de datos estadísticos. Esta información es posible extraerla y guardarla directamente en el computador que utiliza el usuario.

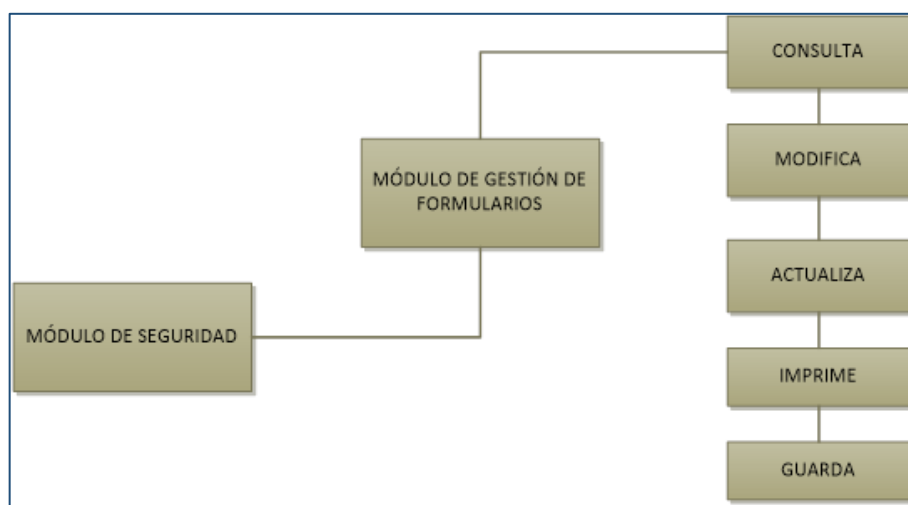


Figura 32: Módulo de reportes
Elaborado por: Pablo Castillo

Módulo de Mantenimiento

Este módulo ayuda al administrador de la aplicación a gestionar algunos de los módulos anteriores; por ejemplo en el módulo de administración y seguridad se puede dar soporte a algún usuario que perdió su clave de acceso, en este caso el usuario administrador puede resetear la clave de ese usuario, también es posible renombrar las canchas o instalaciones que se encuentran registradas en la base de datos de la aplicación, es posible también agregar más nombres de canchas si se lo requiere.

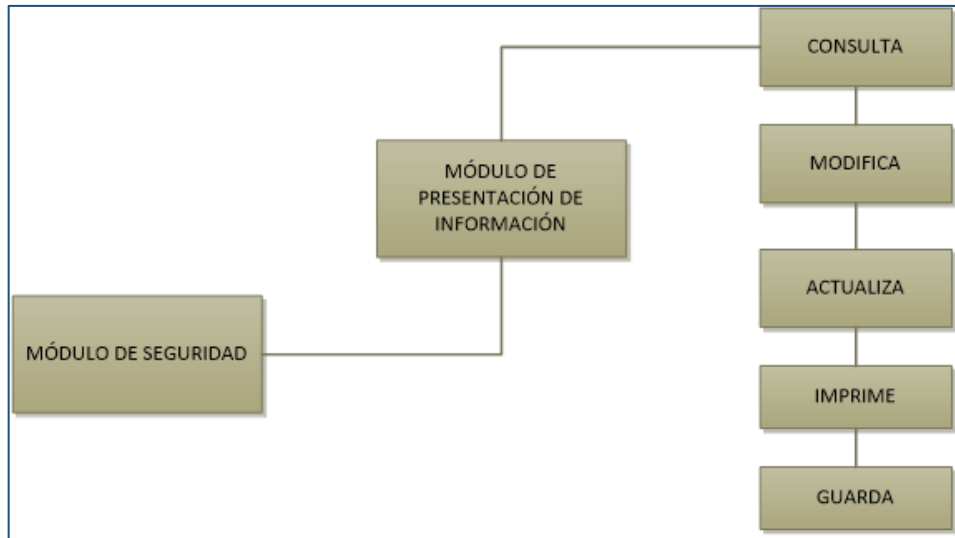


Figura 33: Módulo de mantenimiento
Elaborado por: Pablo Castillo

Artefactos Fase 3

Documento Arquitectura	ANEXO 3
Manual de Programador	ANEXO 4
Manual de Instalación	ANEXO 5

2.5. Transición.

2.5.1. Pruebas.

Para realizar las pruebas se ha buscado técnicas que permitan encontrar los defectos o fallas en la funcionalidad del sistema. Para las pruebas se ha considerado los distintos requerimientos funcionales y no funcionales. A continuación el listado de pruebas que se ejecutarán:

Pruebas de navegación

- Verificar el REQ001 Registro de usuario
- Verificar el REQ002 Registro de usuario por un administrador
- Verificar el REQ003 Resetear clave de un usuario
- Verificar el REQ004 Revisar usuarios registrados en la aplicación
- Verificar el REQ005 Eliminar usuario.
- Verificar el REQ006 Revisar lista de canchas.
- Verificar el REQ007 Revisar normativas de préstamo.
- Verificar el REQ008 Verificar disponibilidad de las canchas.
- Verificar el REQ009 Reservar una cancha.
- Verificar el REQ010 Revisar reportes.

- Verificar el REQ011 Guardar reportes.
- Verificar el REQ012 Imprimir reportes.
- Verificar el REQ013 Crear una cancha.
- Verificar el REQ014 Modificar el nombre de una cancha.
- Verificar el REQ015 Modificar la capacidad de personas de una cancha.

Pruebas de interfaz de usuario

- Verificar la navegabilidad de la página, a través de la visualización de diferentes usuarios considerando aspectos de estética y funcionalidad
- Verificar URL corrompidas

Pruebas de desempeño

- Verificar velocidad de acceso a la aplicación web
- Verificación de tiempo de ingreso para visualizar la información
- Verificación tiempo de ingreso para reservar un área

Pruebas de carga

- Verificación de uso en el caso de tener 50 usuarios en la verificación de áreas

Estrategias de pruebas

A continuación se define los objetivos, alcances y criterios para cumplimiento de los requerimientos de pruebas.

Pruebas de navegación

Objetivo: Verificar que la navegación para el usuario, como para el coordinador y administrador sea funcional de acuerdo a los requerimientos del sistema desde su inicio.

Técnicas: Ejecutar cada uno de los casos de uso, ingresando información válida e inválida para verificar errores que se puedan presentar.

Criterio de cumplimiento: Se desarrollaron las pruebas de acuerdo a los casos de uso.

Se aplicó con usuarios aleatorios registrados, no registrados, además ingresando como administrador y como coordinador.

Tipo de error:

- Discrepancia: No tiene incidencia en los resultados
- Incidente: Produce variaciones en los resultados
- Defecto: No permite que se ejecute o continúe la aplicación

Prioridad: Indica el nivel de efecto que existe sobre la aplicación ya sea alta, media o baja y por tanto con que prioridad o velocidad debe corregirse.

Estado: Abierto: si el error no se ha corregido aún.

Cerrado: Si el error en la prueba ha sido corregido totalmente

Error: Describe el error presentado.

Descripción de error: Presenta con detalle el error encontrado.

Artefactos Fase 4

Pruebas de Validación CAPITULO 3

Manual de Usuario ANEXO 6

CAPITULO III
PRUEBAS DE VALIDACION

En este capítulo se detalla las pruebas tomando como referencia la metodología RUP utilizada para el desarrollo del presente trabajo, se establecen técnicas relacionadas con las pruebas para el aplicativo, lo que permitirá garantizar el cumplimiento de los requerimientos funcionales planteados en este trabajo.

3.1. Tipos de pruebas.

Tabla 32: tipos de pruebas

TIPOS DE PRUEBAS	DEFINICIONES	FASE DE RUP
Unitarias	Pruebas utilizadas en desarrollo para probar el funcionamiento de un módulo de código. Ayuda asegurar que las partes que integran la aplicación funciones adecuadamente.	Elaboración
Sistema	Este tipo de pruebas busca discriminar entre la aplicación y los requerimientos, y su finalidad es encontrar errores que puedan darse entre los requisitos y el diseño del aplicativo. Algunos tipos de estos son: Estrés: Evaluación que sirve para poner a prueba robustez y confiabilidad del sistema, sobrecargando las condiciones de uso. Interfaz de Usuario: Valida que la navegación por los elementos que se están probando evidencie las funciones del negocio y los requerimientos funcionales. Rendimiento: Ayuda a verificar si pa aplicación cumple con tiempos de respuesta óptimos. Seguridad: Valida que se cumpla con las seguridad que se estipularon en los requerimientos funcionales.	Construcción
Funcionales	Usabilidad: este tipo de pruebas proporciona evidencias de problemas de factores humanos.	Transición

Fuente: Pablo Castillo

3.2 Ambiente de pruebas

Para el ambiente de pruebas que se implementó se lo realizo en una red local con los siguientes equipos.

Tabla 33: configuración del ambiente de pruebas y configuración de equipos

Componente	Características	Herramienta o software instalado.
Servidor de aplicaciones	Procesador: Intel(R) Core(TM) i5-2400 CPU @ 3.1GHz.	Apache 8.0, Wampserver 3.0.6 de 64 bits, netbeans 8.0

	RAM: 10 GB (9.89 utilizable) Sistema: Sistema operativo de 64 bits. Red: Intel(R) 82579LM Gigabit Network Connection	
Laptop (usuario)	Procesador: Intel(R) Core(TM)2 Duo CPU T7300 @ 2.00GHz. RAM: 4 GB (3.49 utilizable) Sistema: Sistema operativo de 32 bits. Red: Controladora Gigabit Broadcom NetXtreme 57xx	Navegadores para probar la aplicación, Firefox, Chrome, Internet Explorer

Fuente: Pablo Castillo

Podemos observar en la tabla 33, la configuración a los equipos para las pruebas, con el servidor de aplicaciones donde se alojan, el servidor Web Apache 8.0 que representa la ubicación física donde se aloja el modelo del patrón MVC, al ambiente de desarrollo NetBeans, el servidor de Wampserver para el manejo y gestión de la base de datos MySQL

3.3 Ejecución y resultado de las pruebas.

Las pruebas realizadas son las que se mencionan en la tabla ## y su propósito es:

- Mostrar la mayoría de los errores que se puedan presentar.
- Reconocer posibles falencias de la aplicación.
- Evaluar el producto software.
- Satisfacer los requerimientos del cliente.

3.3.1 Pruebas Unitarias.

Colección de pruebas aplicadas a los casos de uso de cada módulo, siendo cada una independiente de las demás. Estas permitirán:

- Verificar que los módulos de la aplicación no tengan errores.
- Que los controles deben ser probados.

3.3.1.1 Pruebas unitarias de REQ001

El objetivo de esta prueba es comprobar el funcionamiento del REQ001 relaciona con el módulo de Administración y Seguridad.

Podemos observar en la figura que cuando no se ingresan correctamente los datos la aplicación no permite avanzar, es necesario corregir e ingresar los datos correctamente (la aplicación ofrece ayudas visuales para el registro)

Figura 34: Pruebas unitarias REC001
Elaborado por: Pablo Castillo

Una vez que se ingresan correctamente los datos solicitados el usuario queda registrado y la información de ese usuario se guarda en la base de datos. Podemos verificar en el código que el registro es efectivo, también podemos verificar que el requerimiento se cumplió.

```

<center>
  <section style="margin:80px;">
    <h1 style="padding-top: 100px;margin-bottom:40px;">El registro se ha realizado exitosamente.</h1>

    <a href="/UTPL/iniciarsesion.jsp"><b>Iniciar Sesion</b></a>
  </section>
</center>

```

Figura 35: Resultado pruebas unitarias REC001
Elaborado por: Pablo castillo

3.2.1.2 Pruebas unitarias del REQ002

El objetivo de esta prueba es comprobar el funcionamiento del REQ001 relacionado con el módulo de Administración y Seguridad.

Esta prueba nos permitirá evidenciar que la aplicación realiza validaciones el momento de guardar la información de los usuarios, ya que el momento de guardarlo en la base de datos esta es comparada con la demás información que ya está guardada en la base de datos por ejemplo validación se hace con la cuenta de correo electrónico, en este caso este registro no se puede guardar ya que el email es similar al que ya está en la base de datos.

Registro de usuario	
Nombre:	<input type="text" value="Norma"/>
Apellido:	<input type="text" value="Martinez"/>
Email:	<input type="text" value="martinezn@utpl.edu.ec"/>
Telefono:	<input type="text" value="96325874"/>
Password:	<input type="password" value="....."/>
Permiso:	<input type="text" value="Usuario"/>
<input type="button" value="Registrar"/>	
<input type="button" value="Pagina Principal"/>	

Figura 36: Pruebas unitarias REC002
Elaborado por: Pablo Castillo

Como se realizó la verificación y no se cumple el registro ya que el archivo registroServlet hace esta validación, la aplicación nos devuelve el mensaje “El usuario ya está registrado, Intente de nuevo”.

```

<center>
  <section style="margin:80px;">
    <h1 style="padding-top: 100px;">El Usuario ya esta registrado.</h1>
    <h1 style="margin-bottom:40px;">Intentar de Nuevo</h1>
  </section>
</center>

```

Figura 37: Resultado pruebas unitarias REC002
Elaborado por: Pablo castillo

3.2.1.3. Pruebas unitarias del REQ003

El objetivo de esta prueba es comprobar el funcionamiento del REQ003 relacionado con el módulo de Administración y Seguridad.

En esta prueba vamos a demostrar que se cumple una de las políticas de seguridad del aplicativo, que es que el password del usuario debe tener como mínimo 6 caracteres, si es menos de eso la aplicación no permite registrar el usuario. Como podemos observar en la figura siguiente si se pone un password que no cumple la política de seguridad nos aparece el mensaje “Su password debe tener al menos 6 caracteres”

Figura 38: Pruebas unitarias REC003
Elaborado por: Pablo Castillo

El código de esta prueba nos indica que no se está cumpliendo la política y nos indica que intentemos nuevamente poner un password correcto.

```
// Specify the validation error messages
messages: {
  nombre: " Ingrese sus nombres",
  apellido: " Ingrese sus apellidos",
  telefono: {
    required: " Ingrese su numero de Telefono",
    minlength: " Telefono debe de tener al menos 8 numeros",
    number: " El telefono debe ser solo numeros"
  },
  password: {
    required: " Por Favor Ingrese su Password",
    minlength: " Su Password debe de tener al menos 6 caracteres"
```

Figura 39: Resultado pruebas unitarias REC003
Elaborado por: Pablo Castillo

3.2.1.4. Pruebas unitarias del REQ008 y REQ009

El objetivo de esta prueba es comprobar el funcionamiento de los requerimientos REQ008 y REQ009 relacionados con los módulos de Verificación y de Gestión y reserva.

En esta prueba vamos a revisar el funcionamiento del formulario, ya que al ser una de las partes más importantes del aplicativo, es necesario validar su funcionamiento.

Podemos ver en las figuras siguiente que si no se llena correctamente el formulario no es posible guardar este en la base de datos, ya que no nos aparece el botones Guardar que es el que guarda la información de reserva en curso.

Formulario para pedido de instalaciones deportivas

Fecha de solicitud: Instalacion solicitada:

Nombre persona responsable: Responsable:

Dependencia y/o titulacion: Observaciones:

Fecha inicio reserva: Hora inicio reserva:

Fecha fin reserva: Hora de devolucion:

Participantes

Estudiantes

Hombres: Mujeres:

Docentes

Figura 40: Pruebas unitarias REQ008 y REQ009
Elaborado por: Pablo Castillo

Prestamo de material deportivo

Nombre de la persona que entrega:

Articulo	Cantidad	Descripcion	Estado implemento
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

Observaciones entrega de materiales

Figura 41: Pruebas unitarias REQ008 y REQ009
Elaborado por: Pablo Castillo

Una vez que se llena todos los campos requeridos nos aparece el botón guardar como podemos ver en fa figura siguiente.

Figura 42: Pruebas unitarias REQ008 y REQ009
Elaborado por: Pablo Castillo

Si toda la información es llenada de manera correcta el código genera de manera satisfactoria el botón Guardar.

```

</table>
  <input type="hidden" name="usuarioid" value="1">
  <section style="display:inline-table; margin-right: 20px;" >
    <center style="margin-top: 20px;margin-bottom: 40px;">
      <input type="submit" id="enviarFormulario" class="btn btn-success" value="Guardar" style="display:none"/>
    </center>
  </section>
  <section style="display:inline-table;">
    <center style="margin-top: 20px;margin-bottom: 40px;">
      <input type="reset" class="btn btn-success" value="Borrar" />
    </center>
  </section>
</center>

```

Figura 43: Resultado pruebas unitarias REC008 y REC009
Elaborado por: Pablo Castillo

3.2.1.4. Pruebas unitarias del REQ010

El objetivo de esta prueba es comprobar el funcionamiento de los requerimientos REQ010 y 9 relacionados con los módulos de reportes.

En esta prueba vamos a revisar el funcionamiento del módulo de reportes, que es de donde se sacara la información para uso administrativo.

Se debe especificar bien las fechas en este módulo ya que de lo contrario no arrojará información que se pueda utilizar.

Figura 44: Pruebas unitarias REQ010
Elaborado por: Pablo Castillo

Como podemos observar el momento que se ingresan fechas donde no habido reservas saldra el siguiente mensaje “No data available in table” que nos indica que no hay informacion es las tablas de las reservas correspondientes a esa fecha

Canchas Reservadas por Fecha Desde 2017-08-31 Hasta 2017-09-30

Nombre Cancha	Numero de Reservas
No data available in table	

Figura 45: Pruebas unitarias REQ010
Elaborado por: Pablo Castillo

Como podemos ver el codigo para esta prueba indica que no hay reservas en las fechas indicadas para el resporte.

```

<h1 id="titulo" style="padding-top: 100px;">Canchas Reservadas por Fecha </h1>
<h3 id="titulo" style='padding-bottom: 20px;margin-bottom: 40px;' >Desde 2017-08-31 Hasta 2017-09-30 </h3>

<center>
<table id='example' class='display nowrap' cellspacing='0' width='100%'>
<thead>
```

Figura 46: Resultado pruebas unitarias REC010
Elaborado por: Pablo Castillo

En las pruebas realizadas anteriormente, el escenario propuesto fue la aplicacion misma sin realizar modificaciones al codifo fuente, se ingreso datos y se evidencia que la aplicacion tabaja de manera correcta, ya que si la informacion es erronea, o no cumple con los parametros y

políticas de seguridad la aplicación no procesa esta información, o a su vez da información que el usuario puede interpretar.

Cuadro de errores presentados en el desarrollo de la aplicación

Tabla 34: Errores presentados

No.	Elemento de prueba	Tipo de error	Prioridad	Estado	Error	Descripción error
1	CU001 Registro de usuario	Discrepancia	Alta	Cerrado	No tiene funcionalidad el registro de usuarios	Al ingresar al registro de usuario presenta error
2	CU002 Registro de usuario por un administrador	Discrepancia	Alta	Cerrado	El criterio de asignación no considera algunos permisos	No se dan todos los permisos necesarios al coordinador para asignar áreas.
3	CU009 Reservar una cancha	Discrepancia	Alta	Cerrado	No se ingresa la información en algunos campos	Cuando se realiza el ingreso de datos, algunos campos no pueden ser llenados, sale en blanco.
4	CU008 Verificación de disponibilidad de canchas	Discrepancia	Alta	Cerrado	No se visualiza el cronograma por áreas	Al ingresar a la visualización de instalaciones disponibles por día, semana, mes y canchas, no se visualiza el momento de colocar canchas.
5	CU012 Crear nueva cancha	Discrepancia	Alta	Cerrado	No se visualiza algunos campos del área para el ingreso	Al abrir el ingreso de nueva área no se presentan todos los campos para ingresar

Elaborado por: Pablo castillo

CONCLUSIONES

- La ejecución y desarrollo de los artefactos mencionados en las fases de la metodología de desarrollo de software RUP, permitió obtener un producto escalable y mantenible en cualquiera de los componentes: Modelo, Vista, Controlador.
- Después de analizar las principales arquitecturas, se hizo posible identificar al patrón MVC como patrón arquitectónico para la construcción de esta aplicación.
- Poner en funcionamiento al servidor Web apache como parte del modelo, ayudo a procesar de mejor manera, las peticiones de información hacia la base de datos, respondiendo rápidamente las solicitudes hechas al controlador evitando la carga de datos en el equipo cliente.
- El desarrollo de la presente tesis ayudara a optimizar recursos en el proceso de reservar las canchas e instalaciones de la universidad, mejorando los tiempos de respuesta en todo el proceso.

RECOMENDACIONES

- Para elaborar nuevas versiones del aplicativo, se debería utilizar correctamente la metodología de desarrollo de software RUP, siguiendo con el control del proceso tomando en cuenta los requisitos, el análisis y diseño.
- Los frameworks utilizados permiten trabajar con otras bases de datos, mientras se mantenga un formato compatible con la aplicación que sea gestionado y encausado mediante JPS.
- En futuras actualizaciones se debe seguir utilizando adecuadamente el patrón arquitectónico MVC, ya que este permite separar la interfaz de usuario de la lógica de negocio, permitiendo que el aplicativo sea escalable y de fácil mantenimiento.

BIBLIOGRAFÍA

- APR. (18 de 01 de 2014). *Aprenderaprogramar.com*. Obtenido de La base de datos:
http://aprenderaprogramar.com/index.php?option=com_content&view=article&id=554:que-es-y-para-que-sirve-una-base-de-datos-principales-tipos-oracle-mysql-sqlserver-postgre&catid=46:lenguajes-y-entornos&Itemid=163
- Belmonte, O. (2011). *Introducción al lenguaje de programación Java. Una guía básica*. España: OBF.
- Bernal, C. A. (2010). *Metodología de la Investigación*. Bogotá D.C., Colombia: PEARSON EDUCACIÓN.
- Capuñay, O. (2014). *Desarrollo Web con PHP*. España.
- Castilla, & León. (2006). *Técnico de soporte informático*. Sevilla : MAD .
- Cobo, A. (2005). *PHP y MySQL: Tecnología para el desarrollo de aplicaciones web*. Obtenido de (<https://books.google.com.co/books?id=zMK3GOMOpQ4C&printsec=frontcover&num=11&hl=es#v=onepage&q&f=false>)
- De Laurentiis, R. (2011). *Metodología BPM: RAD*. España : Club BPM.
- ECK, D. (2010). *La programación JAVA*. California .
- Gálvez, S., & García, I. (2007). *Java a Tope Java Mail*. España: Universidad de Málaga .
- Garrido, P. (2015). *Comenzamos a programar en JAVA*. España: Universidad Miguel Hernández de ELCHE.
- Garrido, P. (2015). *Comenzamos a programar en JAVA*. España: Universidad Miguel Hernández de ELCHE.
- Granados, L. (2014). *UF1844 Desarrollo de aplicaciones Web en el entorno del servidor*. Barcelona : IC editorial .
- Gutiérrez, J. (2013). *¿Qué es un framework web?* España: US.
- Hernández, j. (2014). *Análisis y Desarrollo WEB*. México: Drupal .
- IBM. (26 de 09 de 2015). *www.ibm.com*. Obtenido de Características y tipos de bases de datos: https://www.ibm.com/developerworks/ssa/data/library/tipos_bases_de_datos/
- Ibrugor. (21 de 10 de 2014). *www.ibrugor.com*. Obtenido de ¿Qué es PHP? ¿Para qué sirve?: <http://www.ibrugor.com/blog/que-es-php-para-que-sirve/>
- Kendall, J. (2005). *Análisis y diseño de sistemas*. México : Pearson Education .
- Kendall, K. (2015). *Análisis y Diseño de Sistemas*. México: PEARSON.
- León, F. (2008). *Base de datos relacionadas*. Colombia : Instituto Tecnológico Metropolitano .
- Maciá, J., Mora, F., Gilart, V., Berná, J., Monllor, J., & Hernández. (2008). *Administración de servicios de Internet*. España : Universidad de Alicante .

- Maciá, J., Mora, F., Gilart, V., Berná, J., Monllor, J., & Hernández. (2008). *Administración de servicios de Internet De la Teoría a la Práctica*. España: Universidad de Alicante.
- Martínez M., A. (18 de 03 de 2010). *Niveles de Investigación*. Recuperado el 27 de 01 de 2015, de http://www.arnaldomartinez.net/enfermeria/niveles_de_investigacion.pdf
- Martínez, C. (2012). *Wikipedia: Inteligencia colectiva en la red*. Barcelona : PROFIT .
- Microsoft . (12 de 03 de 2013). *Developer Network* . Obtenido de Descripción general de Microsoft Solutions Framework (MSF): [https://msdn.microsoft.com/es-es/library/jj161047\(v=vs.120\).aspx](https://msdn.microsoft.com/es-es/library/jj161047(v=vs.120).aspx)
- Minera, F. (2010). *PHP 6 Sitios dinámicos con el lenguaje más robusto*. Argentina: USERS.
- Minera, F. (2011). *Desarrollo PHP+MySQL*. Obtenido de <http://blogfbd.blogspot.com/2011/03/mysql-workbench-y-antecesores.html>
- Minera, F. (2014). <http://blogfbd.blogspot.com/2011/03/mysql-workbench-y-antecesores.html>.
- Muñoz, P. (2010). *Mantenimiento de portales* . Madrid : Visión libros .
- MySQL. (01 de 01 de 2015). *The world's most popular open source database* . Obtenido de MYSQL CLUSTER: <http://www.oracle.com/us/products/mysql/overview/index.html>
- Niño, J. (2010). *Sistemas operativos monopuestos* . México : EDITEX.
- Olivia, K., & Carrillo, M. (01 de 01 de 2012). *Metodología RAD* . Obtenido de <http://metodologiarad.weebly.com/index.html>
- Penadés, M. (2010). *Metodologías ágiles para el desarrollo de software* . México : Programming (XP) .
- Procesos de Software. (2013). *Metodología RUP*. Obtenido de <https://procesosdesoftware.wikispaces.com/METODOLOGIA+RUP>
- Quero, E., García, A., & Peña, J. (2010). *Mantenimiento de los portales de Información* . España : Paraninfo .
- Ramos, A. (2011). *Aplicaciones Web*. Madrid, España: Paraninfo.
- Ramos, A. (2011). *Aplicaciones Web*. Madrid-España: Paraninfo.
- Rios, S. (2010). *JFS 2+Hibernate 4+Spring 4* . Estados Unidos : Java Revolutions .
- Sánchez, E. (2012). *Manual de JAVA*. México: McGrawHill.
- SURONLINE. (18 de 06 de 2008). *Aplicaciones WEB*. Obtenido de Qué es una aplicación web: http://www.suronline.net/nuevo_sitio/beneficios-funcionamiento-aplicaciones-web.asp
- UNIXWIZ. (15 de 10 de 2007). *SQL Injection Attacks by example* . Obtenido de <http://unixwiz.net/techtips/sqlinjection.html>.
- Van Lancker, L. (2014). *El Framework JavaScript de la web 2.0*. Barcelona : ENI.

Web appsec. (15 de 08 de 2009). *Web Application Security Consortium*. Obtenido de <http://www.webappsec.org/>.

ANEXOS

ANEXO 1:
ENTREVISTA PARA DETERMINAR LOS REQUERIMIENTOS DEL SISTEMA

Requisitos

Que procedimiento se sigue para el préstamo de las instalaciones y cómo se gestiona el préstamo.

El personal docente y administrativo con anticipación, se verifica disponibilidad, se ve en el calendario.

Este departamento (Misiones Universitarias) está regida por la Dirección de Estudiantes.

- Si es un profesor se necesita un oficio firmado por el Director del Estudiante.
- Hay un listado de estudiantes que ingresan a las instalaciones.

Si es estudiante debe estar firmado por el director de Titulación que indique el préstamo de las instalaciones, una copia del documento de préstamo lo tiene el guardia que está en la puerta de la instalación.

Formulario.

- Instalación – se utiliza el mismo formulario para todos, no está ninguna página, pero si puede estar colgado en la página.
- Material

Publican el Formulario en la web.

Actualmente no está publicado.

Como se Calendariza el préstamo de las instalaciones.

Está en Google Doc. Esta calendarizado, la idea es que se pueda ver y que se pueda elegir la zona para poder reservar.

Como se controla el tiempo de préstamo de las instalaciones.

Se controla en el formulario.

Tiempo en general son 2 horas, si es más se estipula en el formulario

Requisitos de uso.

Si hay, en la Normativa de uso general,

Hay garantías para el préstamo de las instalaciones

Debe estar en la hoja de préstamo o formulario, un Responsable, por ejemplo si se pierde un balón, tienen que devolver un balón, no se acepta el valor del balón.

¿Se prestan las instalaciones a personal externo o solo interno (Estudiantes, docentes, administrativo)?

Solo a personal interno, hay muy pocas excepciones a empresas, pero muy raro.

¿Hay suficientes instalaciones para todos los estudiantes?

Se está ampliando, por lo general solo utilizan para fútbol, pero hay otras opciones de deporte.

Como es el estado de las instalaciones.

Sintética = Bueno

Polideportiva = Bueno

Canchas externas (vóley y Baloncesto) = Regular (Sin luz ni techo)

Notas:

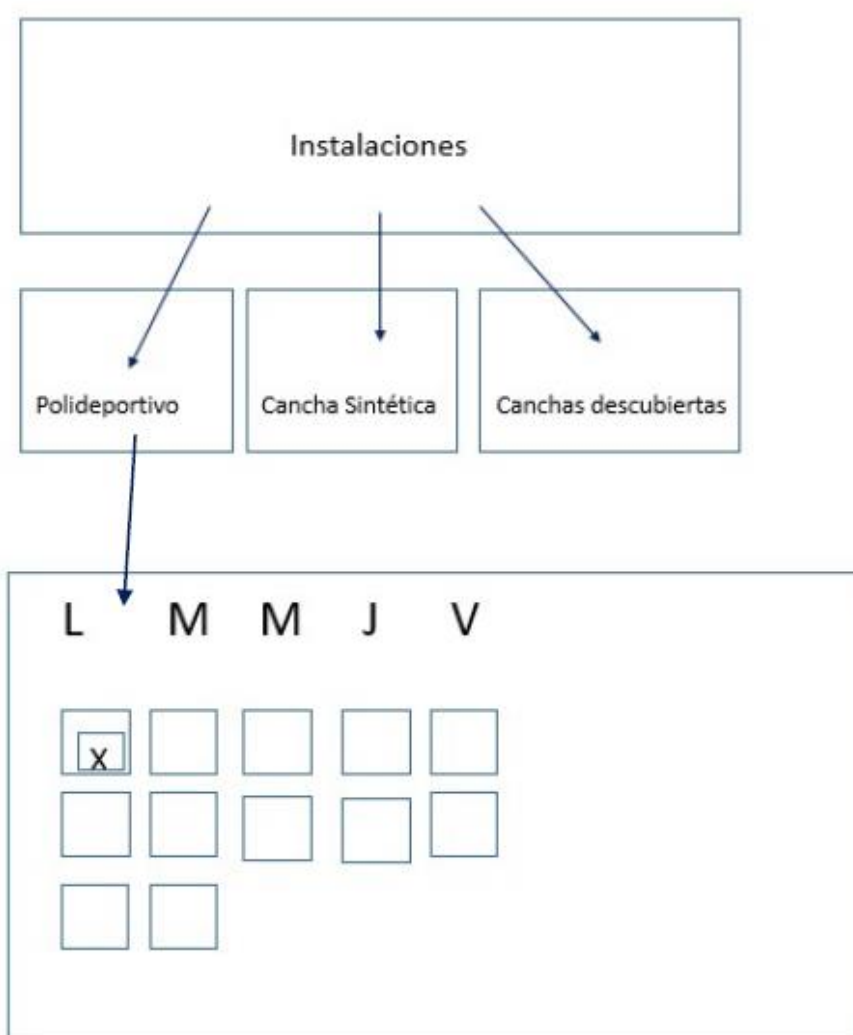
- Hay un número máximo y mínimo de estudiantes permitido.
- Horario de Préstamo: 7 am a 9 pm en el año escolar
- Verificar disponibilidad, el horario puede variar – el club deportivo tiene prioridad, el préstamo se reparte en tiempo libre de las instalaciones
- Sábado no se presta – solo de lunes a viernes, con excepción de solicitud de administración.
- No se presta en vacaciones ni en mantenimiento.
- Una hoja de material si se necesita
- Se presta las instalaciones 1 hora mínimo, 3 horas máximo
- Se reserva máximo con 14 días, mínimo con un día.
- Se puede reservar por un mes máximo.

Implementar:

- Pago de las instalaciones

- Estadísticas

- Por meses
- Por personas
- Por actividad.



**ANEXO 2:
DOCUMENTO VISION**

Documento Visión

Versión [1.0]

Visión del Sistema

Información del Documento

TÍTULO: DA GESTION PRESTAMO INSTALACIONES

SUBTÍTULO: [Desarrollo Aplicativo para Gestión de Préstamo de Instalaciones]

VERSIÓN: [1.0]

AUTOR: Pablo Catillo

ESTADO: Final

Contenido

1 Referencias	2
2 Posicionamiento del Producto.....	3
2.1 Definición del Problema	3
2.2 Posicionamiento del Producto.....	4
3. Resumen de Afectados/Involucrados	5
4 Resumen de Usuarios	5
5 Necesidades de los Afectados/Usuarios.....	6
5.1 Necesidades de los afectados	6
5.2 Necesidades Personal Área Misiones Universitarias.....	6
6 Resumen del Producto	7
6.1 Perspectiva del Producto	7
6.2 Resumen de Capacidades	8
6.2.1 Beneficios.....	8
6.2.2 Características	8
6.3 Supuestos y dependencias	8

1 Referencias

- [1] Definición de Requerimientos de Software ANSI/IEEE std. 830-1984
- [2] Rational Unified Process RUP

2 Posicionamiento del Producto

2.1 Definición del Problema

Tabla 11: Definición del problema

El Problema de	Al momento se utiliza un proceso semiautomático para la tarea de asignación y préstamo de instalaciones, haciendo que esta gestión sea un proceso lento, estos procesos en las diferentes área con respecto a las actividades mencionadas anteriormente pueden realizarse con mayor eficiencia y con tiempos de proceso menores esto ayudara al departamento de Misiones Universitarias a mejorar la gestión y administración en el préstamo de las instalaciones, ayudando a realizar el proceso de calendarización y reportes de actividades ya que es un proceso manual
Afecta a	<ul style="list-style-type: none">• Estudiante• Docentes• Personal Administrativo
El impacto es	<ul style="list-style-type: none">• Demora al gestionar el proceso de préstamo de las instalaciones.• Los formularios se realizan manualmente y se demora el digitarlos principalmente en la tipada de los mismos• Demora y dificultad en elaboración de estadísticas y reportes.
Una solución exitosa es	<ul style="list-style-type: none">• Permite gestionar un proceso más rápido en el préstamo de las instalaciones.• Permitirá controlar de mejor manera las admisiones a las instalaciones logrando una mayor interacción entre todos los usuarios, a la vez que permite optimizar en tiempo el proceso realizado por el recurso humano.• Permitirá generar reportes estadísticos de acuerdo al uso y disponibilidad de las instalaciones y de acuerdo al número de personas que las solicitan.• Permitirá tener informes impresos y estadísticos con mayor rapidez.• Permitirá tener una visualización de todas las instalaciones con que cuenta la universidad.• Permitirá verificar la disponibilidad de las instalaciones a las personas que van a utilizar las instalaciones.• Permitirá un rápido ingreso de datos en el formulario digitalizado.

Elaborado por: Pablo castillo

2.2 Posicionamiento del Producto

Tabla 12: Posicionamiento del producto

Para	<ul style="list-style-type: none"> • Estudiantes. • Docentes. • Personal Administrativo. • Coordinadores
Quien (es)	<ul style="list-style-type: none"> • Pretenden tener una rápida gestión en la asignación de las instalaciones. • Pretenden tener información estadística de las asignaciones y reservas de las instalaciones. • Desean hacer un seguimiento a una instalación reservada.
El (nombre del producto)	Aplicación web para gestionar el proceso de préstamo de instalaciones.
Que	<ul style="list-style-type: none"> • Permita al personal del departamento de Misiones Universitarias tener información de la aplicación. • Permitirá visualizar el estado en función de reservada o no, de una instalación específica. • Permitirá que estudiantes, docentes y personal administrativo reserven las instalaciones de forma más rápida. • Permitirá generar información estadística del préstamo de las instalaciones, a quien se asignó, que tiempo y la fecha de forma más rápida.
A diferencia	<p>Del momento actual que es semiautomático:</p> <ul style="list-style-type: none"> • Dificulta llevar un control de los reportes sobre el préstamo de las instalaciones. • Genera demora en el tiempo de generar las estadísticas. • Hace que el proceso de préstamo de las instalaciones sea lento.
Esta aplicación	<p>Tiene la capacidad de:</p> <ul style="list-style-type: none"> • Generar información por usuarios que solicitan las instalaciones. • Permite generar un formulario pormenorizado de la instalación que se desea reservar. • Permite que los usuario se registren por si solos como usuarios para poder visualizar la disponibilidad de las instalaciones. • Provee de un módulo para generar reportes actualizados de la información que ingresa a la base de datos por medio de la aplicación.

Elaborado por: Pablo castillo

3. Resumen de Afectados/Involucrados

Tabla 13: Resumen de afectados/involucrados

Nombre	Descripción	Responsabilidad
Personal del Área de Misiones Universitarias	Son los responsables del control y gestión del proceso de préstamo de las instalaciones de la universidad.	Definir los parámetros y políticas de préstamo de las instalaciones. Establecen los parámetros de supervisión y control (Normativas de Uso). Generan reportes de acuerdo a las facilidades que brinda la aplicación. Asignan encargados de custodio de las instalaciones de la universidad.
Estudiantes, docentes, personal administrativo.	Son los encargados de ingresar información a la base de datos de la aplicación uniéndolo con dos campos que se van a guardar en la base, para poder generar los reporte.	Establecer requerimientos sobre la necesidad de uso de las instalaciones y poder plasmarlas en la aplicación.

Elaborado por: Pablo castillo

4. Resumen de Usuarios

Tabla 14: Resumen de usuarios

Nombre	Descripción	Afectado al que representa
Personal del Área de Misiones Universitarias	Se encargan de revisar la información concerniente a las instalaciones que posee la universidad en el campus principal de Loja. Son los responsables de gestionar y controlar el acceso a las instalaciones. Se encargan de validar las estadísticas y los reportes.	Personal del Área de Misiones Universitarias
Usuarios	Se encargan de generar la información que ingresa a la base de datos de las instalaciones que reservan, para poder generar los reportes que tiene la aplicación.	Estudiantes, docentes, personal administrativo.

Elaborado por: Pablo castillo

5. Necesidades de los Afectados/Usuarios

5.1. Necesidades de los afectados

Tabla 15: Necesidades de los afectados

Necesidad	Prioridad	Solución Actual	Soluciones Propuestas	Preocupación
Realizar el proceso de reserva de una instalación de forma rápida.	Alta	Las reservas se realizan de manera semiautomática, los informes se no son automáticos y se demoran en generarse.	La aplicación web ayudara a generar las reservas de forma rápida y a generar los respectivos informes manera automática.	Seguridad en el acceso, rapidez en generar los informes.

Elaborado por: Pablo castillo

5.2. Necesidades Personal Área Misiones Universitarias.

Tabla 16: Necesidades personal de Misiones Universitarias

Necesidad	Prioridad	Solución Actual	Soluciones Propuestas	Preocupación
Realizar un seguimiento rápido en el proceso de préstamo	Alta	Se realiza un seguimiento lento y con un retraso de tiempo, los informes no son automatizados	Aplicación que permita realizar un seguimiento rápido al proceso.	Parámetros de control y seguimiento.
Acceder a la información de cada reserva y poder gestionar de manera rápida el proceso de préstamo	Alta	Se realizar mediante un Excel que está en un sistema Web, es un proceso semiautomático.	Aplicación que permita revisar la información necesaria y de forma rápida.	El conocimiento de la herramienta. Y la integración al sistema informático de la universidad
Preparación de estadísticas con la información ingresada por los usuarios, y trasladarle a otros formatos de documentos como Excel o pdf.	Alta	Se tiene formatos en digitales e impresos para su revisión, y las estadísticas se generan manualmente	La aplicación puede generar reportes automáticos, y exportarlos a otros tipos de formatos.	Constatar las necesidades de reportes por parte del área de Misiones Universitarias.

Elaborado por: Pablo castillo

6. Resumen del Producto
6.1. Perspectiva del Producto

Tabla 17: Perspectiva del producto

Módulo de	Descripción
Módulo de Administración y Seguridad	Permite ingresar a la aplicación y que el usuario pueda quedar registrado, además permite asignar los respectivos roles a los usuarios para que la aplicación solamente pueda ser modificada en su aspecto interno por el administrador.
Módulo de Verificación	Permite realizar una revisión de las instalaciones disponibles por fecha (día) y hora, para que los usuarios realicen reservas de manera efectiva.
Módulo de gestión y reserva	Encarga de reservar una instalación específica, quedando registrada en los reportes para que no sea utilizada nuevamente o existan confusiones de espacios tiempo otorgados a otros usuarios. Están aquí los sub-módulos que se necesitan para completar una reserva y son; llenar la grilla calendarizada que es donde marcamos y especificamos la reserva de una instalación, la fecha y el tiempo que se va a utilizarla; y también se llena el formulario de Préstamo de la instalación, que es donde marcamos todos los datos importantes para la verificación de la información que a la vez sirve al departamento para futuros controles, es aquí donde se recoge la información de las reservas realizadas en la aplicación.
Módulo de Reportes	Este módulo sirve a las personas que manejan la información, provee datos significativos para la toma de decisiones, evaluación de datos estadísticos para las personas encargadas de manejar esta información.
Módulo de Mantenimiento	Este módulo ayuda a la persona que administrará la aplicación a dar soporte a los usuarios que por algún motivo pierdan su clave de acceso y también a eliminar usuarios que ya no estén en la universidad, también es de ayuda a la modificación de los nombres de las instalaciones y si fuera el caso a agregar nuevas instalaciones en la base de datos del sistema.

Elaborado por: Pablo castillo

6.2 Resumen de Capacidades

6.2.1 Beneficios.

- Seguimiento rápido al proceso de reserva y préstamo de las instalaciones.
- Generación de reportes y estadísticas automáticamente y exportarlos a otros tipos de formatos.
- Rápido y fácil acceso a la aplicación web y a los reportes.
- Reducción del tiempo para el personal del área de Misiones Universitarias, al realizar el seguimiento de una reserva.
- Reducción de tiempo en la generación de reportes y de las estadísticas, por usuarios, por instalaciones y por tiempo.

6.2.2. Características.

- Facilidad en la recuperación de información para realizar los reportes.
- Facilidad de recuperación para hacer el seguimiento de reserva de una instalación.
- Facilidad de uso de la aplicación.
- Facilidad para hacer revisiones y verificaciones de las instalaciones reservadas
- Facilidad para el mantenimiento.

6.3. Supuestos y dependencias

- Se requiere al acceso a la aplicación para poder hacer el seguimiento de una reserva.
- Se requiere el acceso a la aplicación para poder generar y visualizar los respectivos reportes.
- Los usuarios que se registren desde la interfaz de ingreso, podrán solo visualizar el módulo de verificación.

ANEXO 3:
DOCUMENTO ARQUITECTURA

Documento Arquitectura de software
Gestión de Reservas
[Sistema de Gestión de Reservas de Instalaciones deportivas y recreativas]

[Versión 1.1]

Elaborado por: Pablo Castillo

Contenido

1. Introducción.....	4
2. Contemplación de la Arquitectura.....	4
3. Casos de uso.....	4
4. Vista Lógica.....	5
5. Vista de Procesos – Diagrama de componentes.....	6
6. Vista de Datos – Diagrama de Clases.....	6
7. Diccionario de datos.....	7

1. Introducción.

Este documento proporciona un panorama del funcionamiento de la aplicación donde se incluyen las respectivas vistas del sistema.

El propósito de este documento es proporcionar un panorama general de la arquitectura de la aplicación. Se plasma en el documento las decisiones que se tomaron para la arquitectura y sus partes importantes que se aplicaron en el sistema.

2. Contemplación de la Arquitectura.

Para representar la arquitectura se mostraron casos de uso principalmente y en los demás diagramas que se realizaron en UML.

3. Casos de uso.

En los casos de uso tenemos los siguientes.

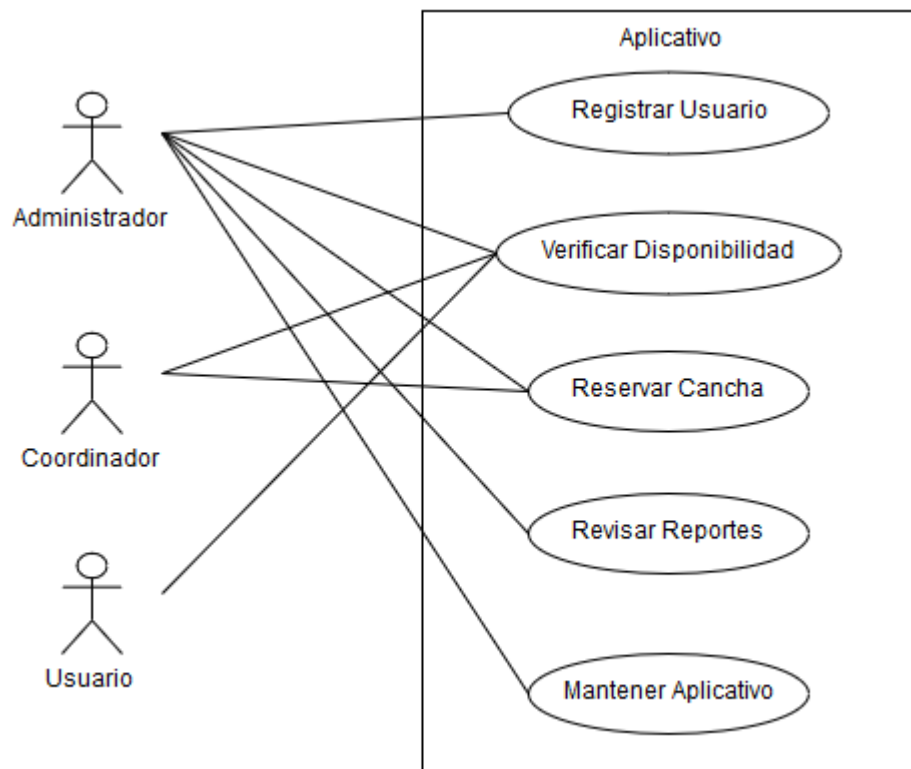


Figura 29: Diagrama de casos de uso

Elaborado por: Pablo castillo

4. Vista Lógica – Diagrama de clases conceptuales

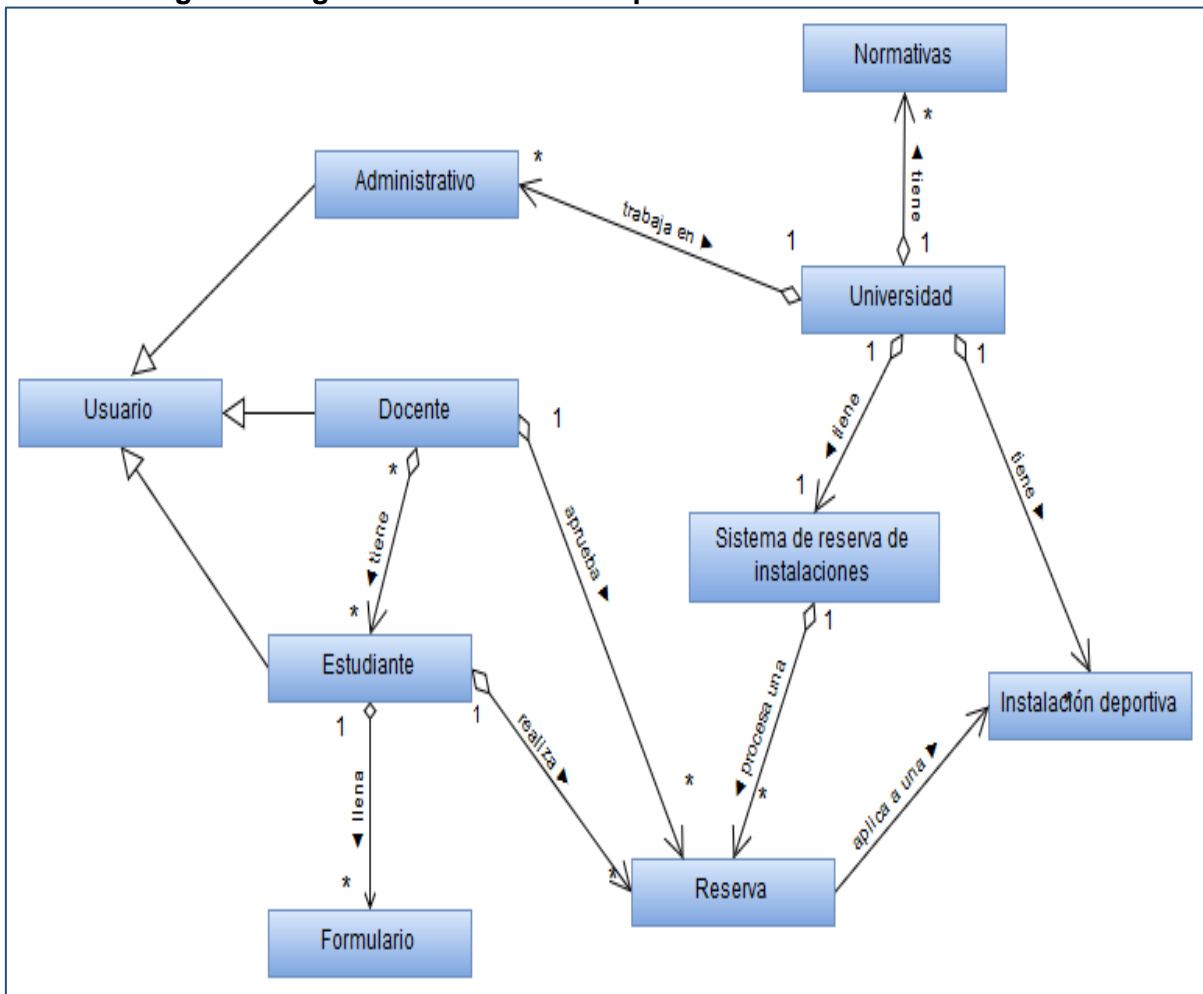


Figura 30: Diagrama de clases conceptuales

Elaborado por: Pablo castillo

5. Vista de Procesos – Diagrama de componentes.

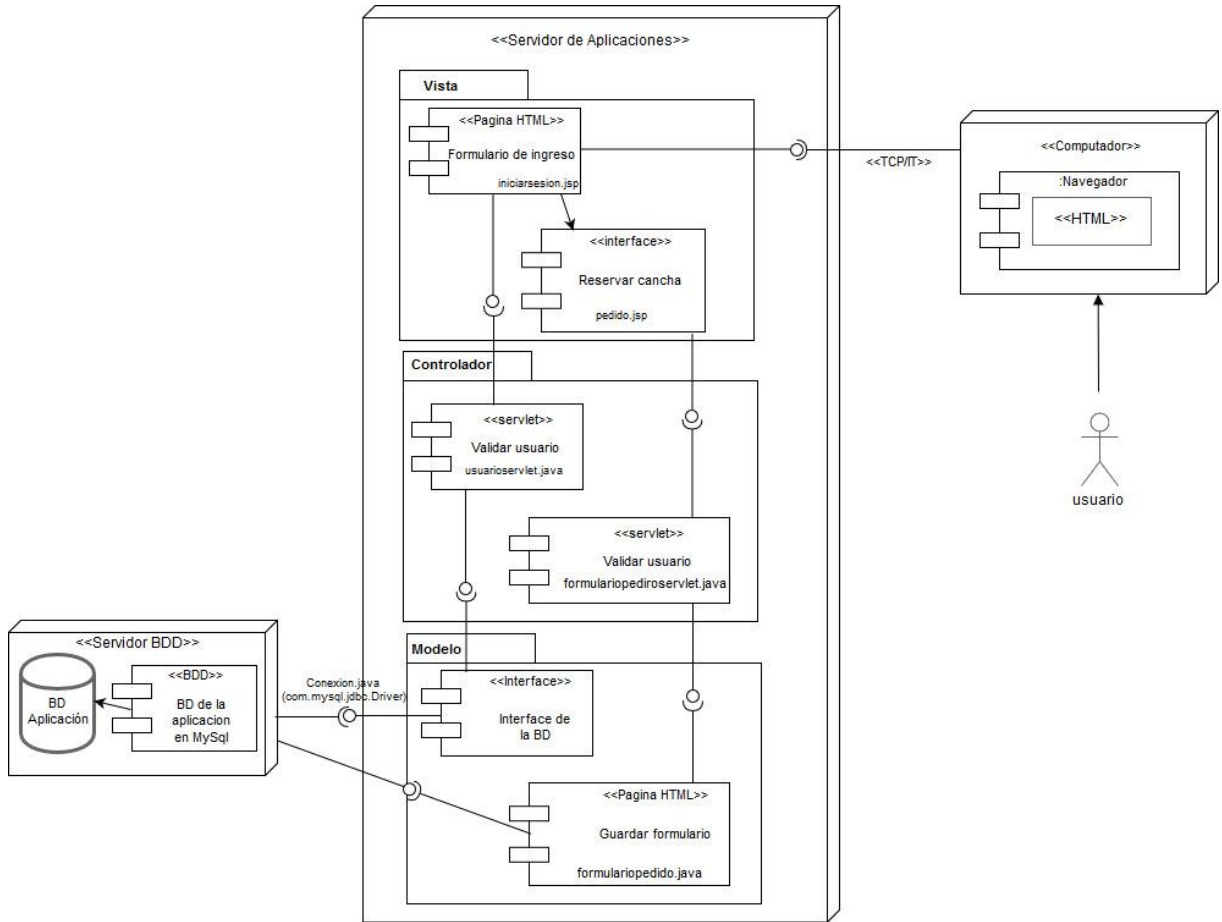


Figura 31: Diagrama de componentes

Elaborado por: Pablo castillo

6. Vista de Datos – Diagrama de Clases.

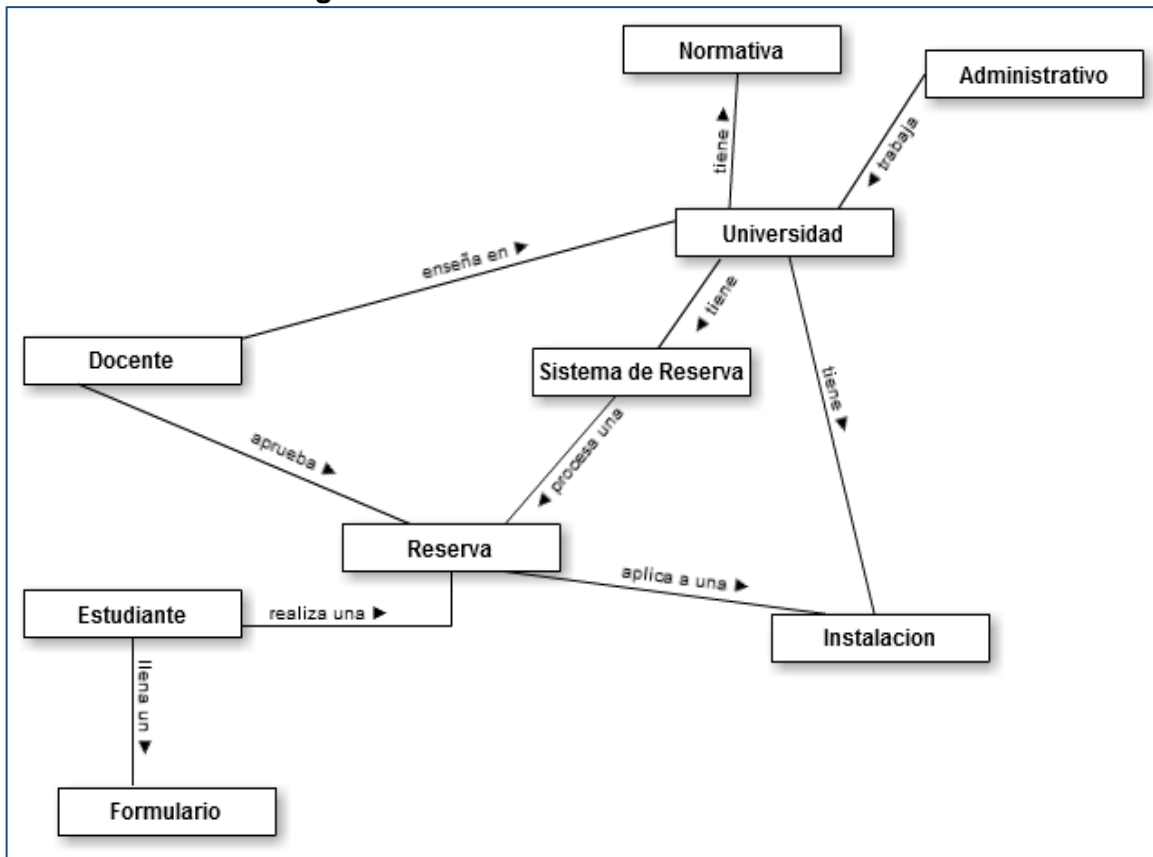


Figura 32: Diagrama de Clases – Vista de Datos

Elaborado por: Pablo castillo

7. Diccionario de datos.

7.1. usuarios

Campo	Tipo	Null	Extra	Comentario
id	int(11)	NO	auto_increment	
nombre	varchar(150)	YES		
apellido	varchar(150)	YES		
email	varchar(150)	YES		
telefono	varchar(150)	YES		
password	varchar(300)	YES		
rol	varchar(150)	NO		
regdate	Date	YES		

7.2. reservas

Campo	Tipo	Null	Extra	Comentario
id_reservas	int(11)	NO	auto_increment	
nombre_canchas	varchar(100)	NO		
start_date	datetime	NO		
end_date	datetime	NO		

7.3. canchas

Campo	Tipo	Null	Extra	Comentario
id_canchas	int(11)	NO	auto_increment	
nombre	varchar(100)	YES		
capacidad	varchar(100)	YES		
estado	varchar(100)	YES		

7.4. formulario_material

Campo	Tipo	Null	Extra	Comentario
id_formulario_material	int(11)	NO	auto_increment	
persona_entrega	varchar(150)	YES		
persona_recibe	varchar(150)	YES		
fecha_entrega	date	YES		
hora_entrega	varchar(150)	YES		
fecha_devolucion	date	YES		
hora_devolucion	varchar(150)	YES		
numero1	varchar(150)	YES		
articulo1	varchar(150)	YES		
descripcion1	varchar(150)	YES		
estado1	varchar(150)	YES		
numero2	varchar(150)	YES		
articulo2	varchar(150)	YES		
descripcion2	varchar(150)	YES		
estado2	varchar(150)	YES		
numero3	varchar(150)	YES		
articulo3	varchar(150)	YES		
descripcion3	varchar(150)	YES		
estado3	varchar(150)	YES		
numero4	varchar(150)	YES		
articulo4	varchar(150)	YES		
descripcion4	varchar(150)	YES		
estado4	varchar(150)	YES		
numero5	varchar(150)	YES		
articulo5	varchar(150)	YES		
descripcion5	varchar(150)	YES		
estado5	varchar(150)	YES		
id_usuario	int	NO		
regdate	date	YES		

7.5. formulario_pedido

Campo	Tipo	Null	Extra	Comentario
id_formulario_pedido	int(11)	NO	auto_increment	
fecha_solicitud	date	YES		
cancha	varchar(150)	YES		
encargado1	varchar(150)	YES		
encargado2	varchar(150)	YES		
dependencia	varchar(150)	YES		
fecha_actividad	date	YES		
horario	varchar(150)	YES		
observaciones	varchar(400)	YES		
fecha_entrega	date	YES		
cedula1	varchar(150)	YES		
nombre1	varchar(300)	YES		
apellido1	varchar(150)	YES		
titulacion1	varchar(150)	YES		
id_usuario	int	NO		
regdate	date	YES		

7.6. formulario_reserva

Campo	Tipo	Null	Extra	Comentario
id_formulario_reserva	int(11)	NO	auto_increment	
fecha_solicitud	date	YES		
cancha	varchar(150)	YES		
personaresponsable	varchar(150)	YES		
tiporesponsable	varchar(150)	YES		
dependencia	varchar(150)	YES		
observaciones	varchar(300)	YES		
fechaInicioReserva	date	YES		
horaInicioReserva	varchar(150)	YES		
fechaFinReserva	date	YES		
horaFinReserva	varchar(150)	YES		
hestudiante	varchar(150)	YES		
mestudiante	varchar(150)	YES		
hdocente	varchar(150)	YES		
mdocente	varchar(150)	YES		
hadm	varchar(150)	YES		
madm	varchar(10)	YES		
persona_entrega	varchar(150)	YES		
a1	varchar(10)	YES		
c1	varchar(10)	YES		
d1	varchar(10)	YES		
e1	varchar(10)	YES		
a2	varchar(10)	YES		
c2	varchar(10)	YES		

d2	varchar(10)	YES		
e2	varchar(10)	YES		
a3	varchar(10)	YES		
c3	varchar(10)	YES		
d3	varchar(10)	YES		
e3	varchar(10)	YES		
a4	varchar(10)	YES		
c4	varchar(10)	YES		
d4	varchar(10)	YES		
e4	varchar(10)	YES		
observacionesMateriales	varchar(500)	YES		
usuarioid	int(11)	NO		
regdate	date	YES		

7.7. titulación

Campo	Tipo	Null	Extra	Comentario
id_titulacion	int(11)	NO	auto_increment	
nombreCompleto	varchar(500)	YES		
nombreSistema	varchar(500)	YES		

7.8. mes

Campo	Tipo	Null	Extra	Comentario
id_mes	int(11)	NO	auto_increment	
mes	varchar(250)	NO		
numero	varchar(10)	NO		

8. Vista de implementación.
Diagramas de secuencia.

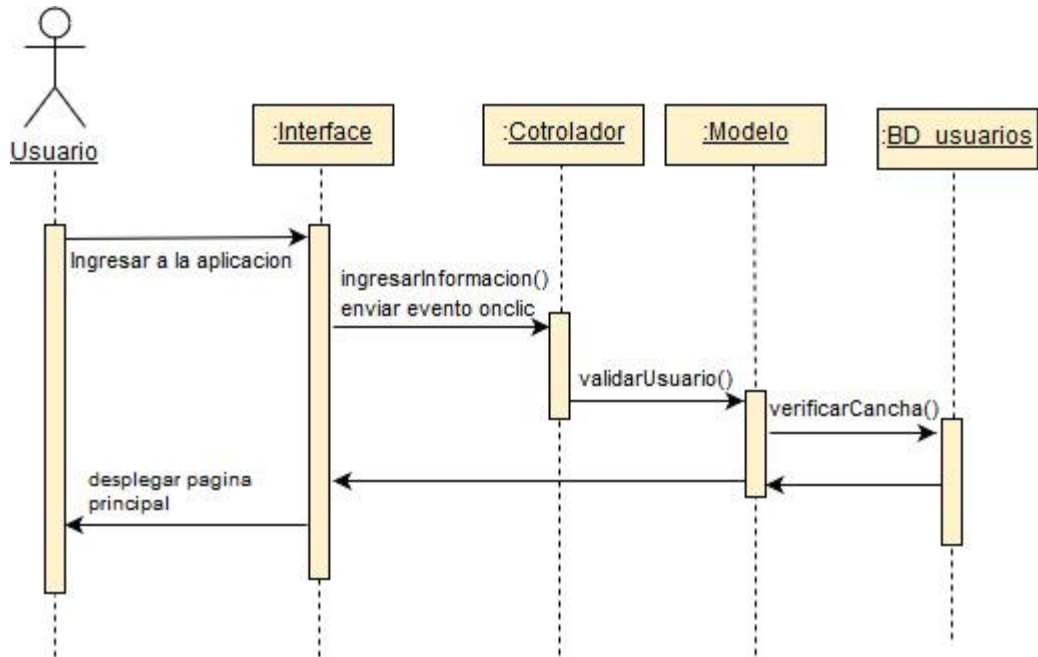


Figura 33: Diagrama de secuencia – Autenticación de usuario

Elaborado por: Pablo castillo

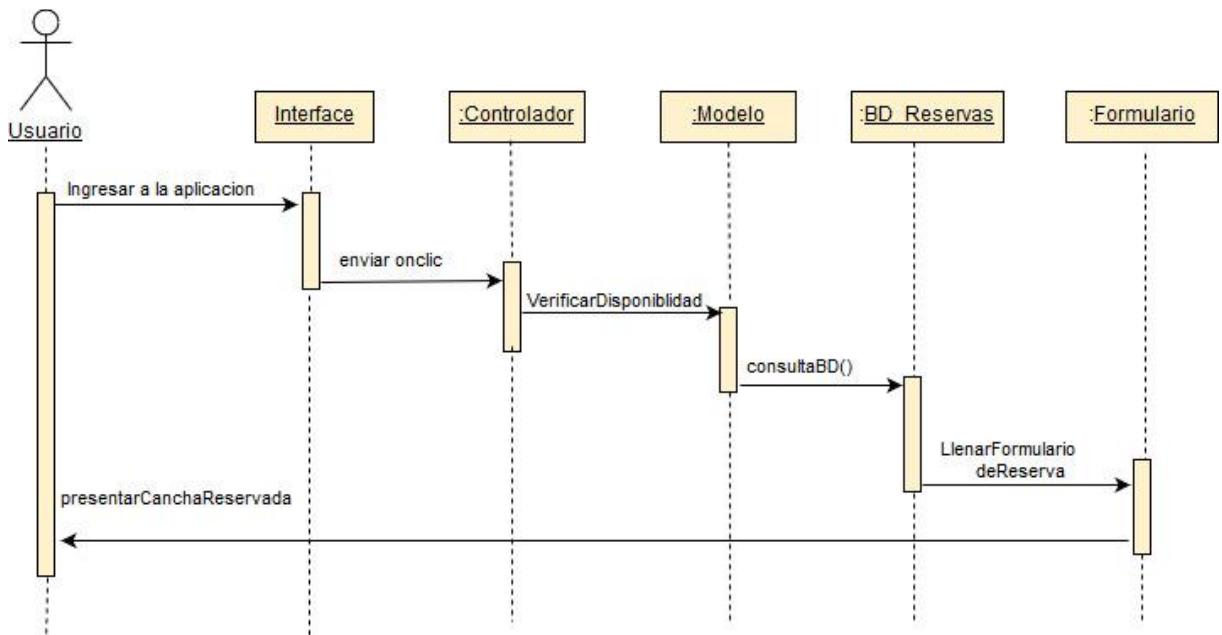


Figura 34: Diagrama de secuencia – Realizar Reserva

Elaborado por: Pablo castillo

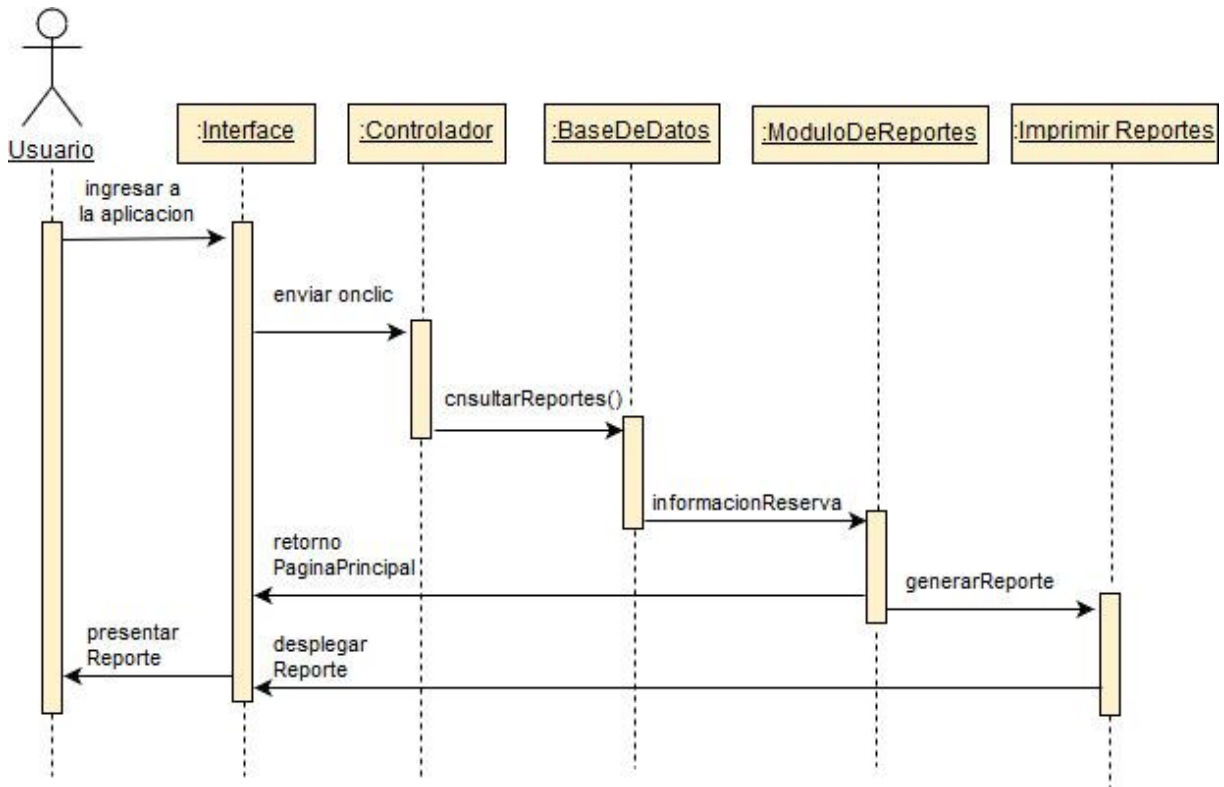


Figura 35: Diagrama de secuencia – Reporte

Elaborado por: Pablo castillo

Diagrama de estados.

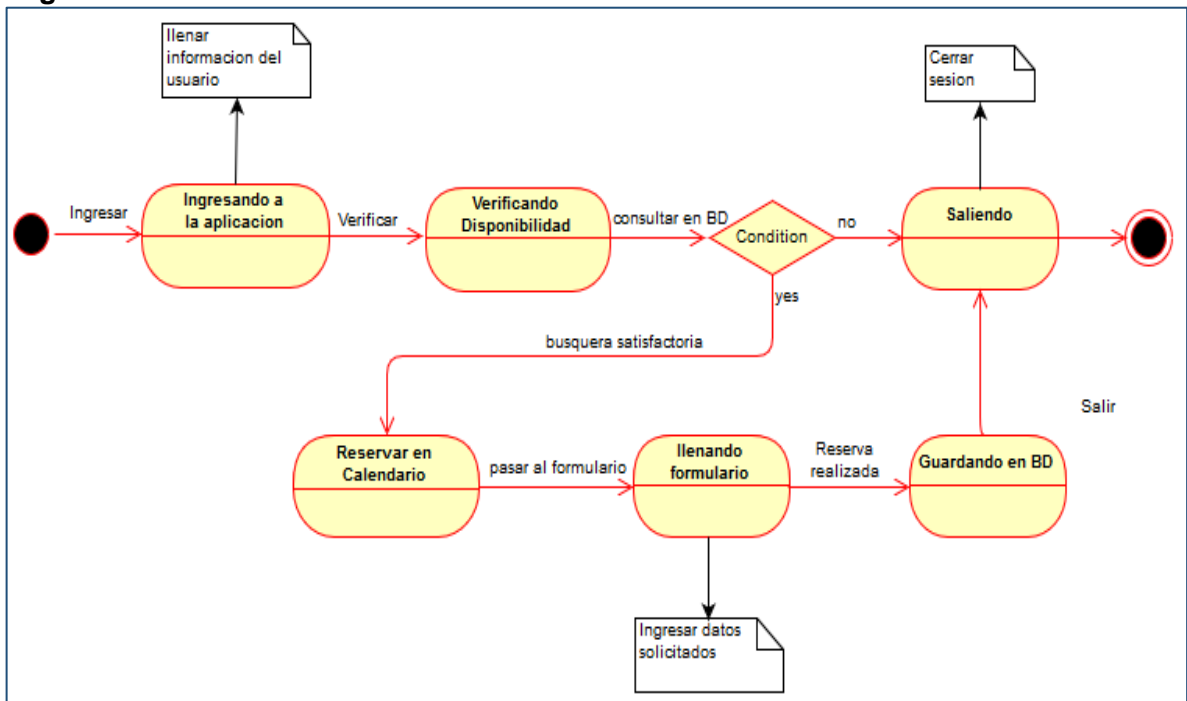


Figura 36: Diagrama de Estados

Elaborado por: Pablo castillo

Diagrama de colaboración.

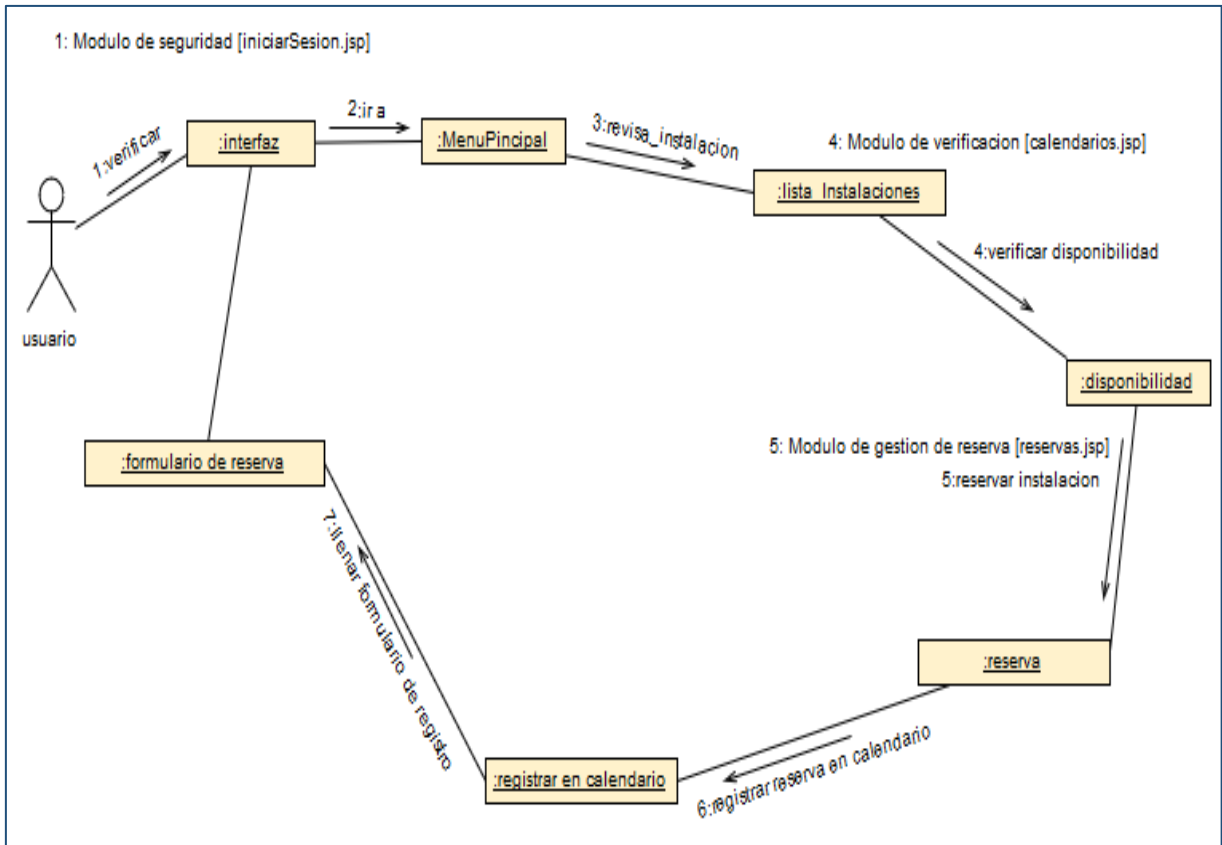


Figura 37: Diagrama de colaboración

Elaborado por: Pablo castillo

**9. Vista conceptual.
Modelo del dominio.**

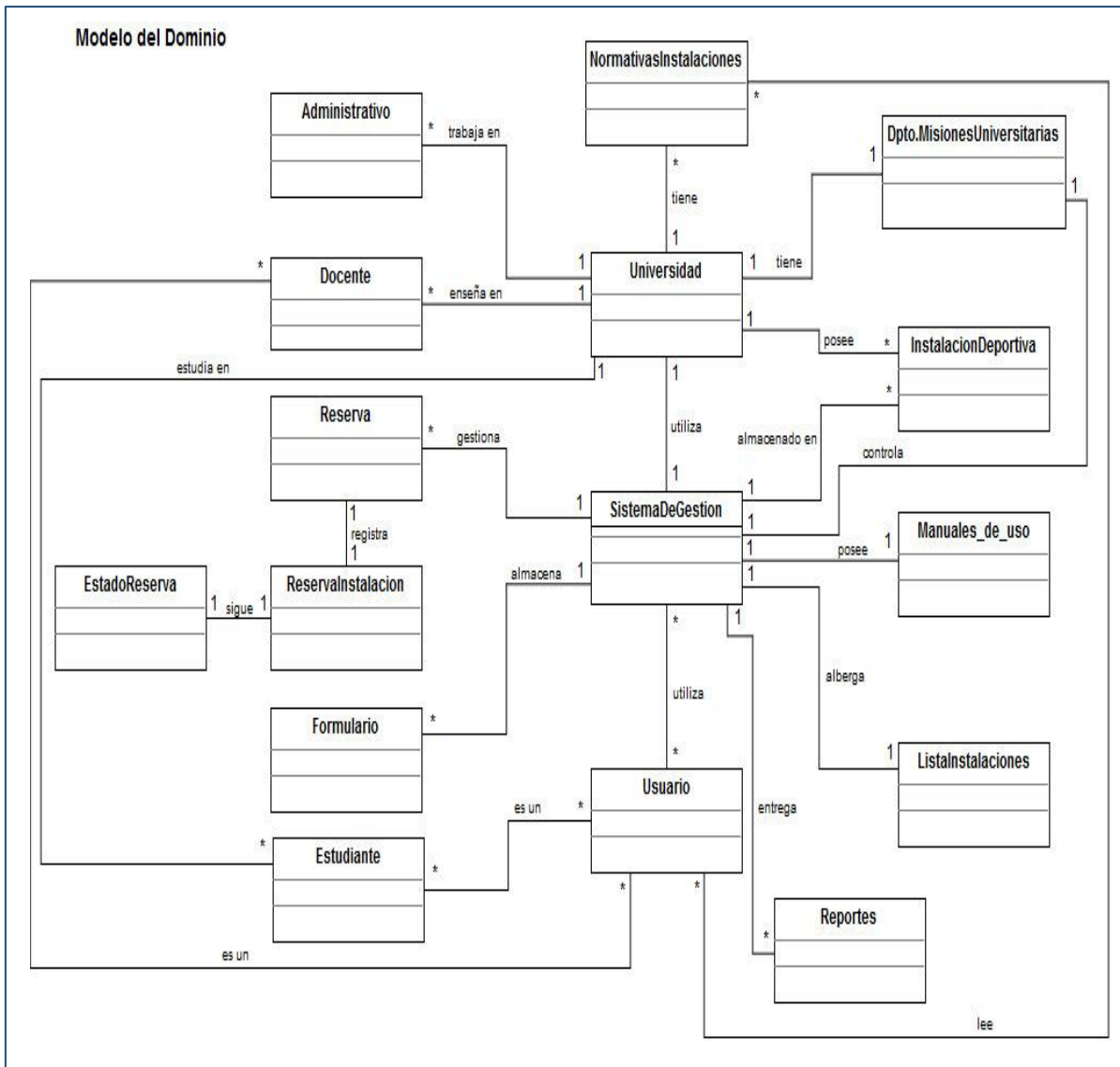


Figura 38: Modelo del dominio – Vista conceptual

Elaborado por: Pablo castillo

ANEXO 4:
MANUAL DE PROGRAMADOR

Manual de Programador

1. Herramientas empleadas en la aplicación

La aplicación de Gestión para la gestión de préstamos de instalaciones deportivas y recreativas es un aplicativo de entorno web, y para su desarrollo de emplearon tecnologías Java principalmente con algunos de sus principales frameworks como JQuery que se empleó para la generación del formulario de registro, JavaScript para la parte de los formularios de reserva. Como entorno de desarrollo se utilizó Netbeans 8, la base de datos que utiliza la aplicación es MySQL. Como servidor web tenemos a Apache Tomcat para la implementación de los servlet JSP, en la parte de reporte se utilizó el complemento datatables dentro de los recursos en el frontend de la aplicación.

1.1. Software de la base de datos

MySQL.- Es nuestro motor de base de datos para guardar la información que ingresa por medio de los formularios de la aplicación, tiene como una de sus principales ventajas ser opensource, además de ser confiable, es multiplataforma, utilizado también por su alto rendimiento y poseer características de escalabilidad. Para poder conectarnos con la base de datos se lo realiza mediante el driver utilizado en este caso por Java para podernos conectar a la base de datos en MySQL:

```
Class.forName("com.mysql.jdbc.Driver");
```

Después de este especificamos los datos de conexión a nuestra base de datos.

```
Connection conexion =
```

```
DriverManager.getConnection("jdbc:mysql://localhost:3306/proyectoutpl", "root", "xxx");
```

1.2. Lenguaje de Programación

Estas herramientas serán instaladas en un computador, por un lado tendremos la interfaz web que está hecha en HTML,

Java JSP (Java Server Page), utilizado para el desarrollo de la aplicación como para la implementación de los componentes.

HTML 5, se utiliza para la maquetación de las páginas, respectivamente la parte del frontEnd, la parte de la interfaz de usuario junto con CSS.

JavaScript, se utiliza para el formulario principal de reserva.

Jquery, es utilizado para el formulario de registro de usuario.

J-Scheduler y Javaplanner, son librerías necesarias para el funcionamiento del calendario de reservas. JavaPlanner es un control que se utiliza en varios frameworks de Java que permite crear calendarios y planificaciones, con capacidades importantes en la web.

Itex, es un componente desarrollado bajo la filosofía openSource, que se utiliza para manipular o crear archivos PDF en Java y otros lenguajes de programación.

MySQL JDBC Driver, librería que se facilita la conexión con nuestra base de datos MySQL.

JTLS, es un componente de Java que extiende las especificaciones JSP, y proporciona 4 bibliotecas de extensiones utilizadas para el desarrollo de páginas web dinámicas. Las bibliotecas englobadas en JSTL son:

- core: iteraciones, condiciones, manipulación de URL.
- xml; para manipulación de XML y para XML-Transformation.
- sql: para gestionar conexiones de base de datos. .
- fmt: para la internalización y formateo de las cadenas de caracteres como cifras.

1.3. Servido web.

Apache Tomcat, nos sirve para la implementación de los servlets y como contenedor de las mismas. Este contenedor trabaja recibiendo peticiones de páginas web y re-direcciona estas peticiones a un objeto servlet; el funcionamiento es el siguiente: Primero, el browser pide una página al servidor HTTP que es un contenedor de servlets; segundo, el contenedor de servlets delega la petición a un servlet en particular elegido de entre los servlets que contiene; tercero, el servlet que es un objeto java, se encarga de generar el texto de la página web que se entrega al contenedor; cuarto, el contenedor devuelve la página web al browser que la solicitó.

1.4. Reportes.

En el módulo de reporte se cuenta con varios reportes que ayudara al departamento a generar información relevante para la toma de decisiones.

DataTable, para la creación de los reportes que se maneja en el módulo de reportes, las tablas dinámicas,

Bootstrap, es un framework que ayuda a la creación de diseños web, este framework es fácilmente integrable con otras librerías por ejemplo JQuery, también con CSS.

2. Base de Datos

La base de datos que trabaja con nuestra aplicación de gestión tiene 8 tablas con sus respectivas relaciones. A continuación se presentan las tablas y sus características.

2.1. Estructura de las tablas

1. **Tabla usuarios:** guarda la información correspondiente a los usuarios

Columnas Tabla usuarios					
Field	Type	Null	Key	Default	Extra
Id	int(11)	NO	PRI	NULL	auto_increment
Nombre	varchar(150)	YES		NULL	
Apellido	varchar(150)	YES		NULL	
Email	varchar(150)	YES		NULL	
telefono	varchar(150)	YES		NULL	
password	varchar(300)	YES		NULL	
Rol	varchar(150)	NO		NULL	
regdate	date	YES		NULL	

2. **Tabla reservas:** Guarda la información de las reservas que ser realizan en el calendario.

Columnas Tabla reservas					
Field	Type	Null	Key	Default	Extra
id_reservas	int(11)	NO	PRI	NULL	auto_increment
nombre_canchas	varchar(100)	NO	NULL		
start_date	datetime	NO	NULL		
end_date	datetime	NO	NULL		

3. **Tabla canchas:** Guarda la información de las instalaciones y canchas registradas en el sistema

Columnas Tabla Canchas					
Field	Type	Null	Key	Default	Extra
id_canchas	int(11)	NO	PRI	NULL	auto_increment
nombre	varchar(100)	YES	NULL		
capacidad	varchar(100)	YES	NULL		

estado	varchar(100)	YES	NULL		
--------	--------------	-----	------	--	--

4. **Tabla formulario_material:** Guarda la información del material solicitado

Columnas Tabla formulario_material					
Field	Type	Null	Key	Default	Extra
id_formulario_material	int(11)	NO	PRI	NULL	auto_increment
persona_entrega	varchar(150)	YES	NULL		
persona_recibe	varchar(150)	YES	NULL		
fecha_entrega	date	YES	NULL		
hora_entrega	varchar(150)	YES	NULL		
fecha_devolucion	date	YES	NULL		
hora_devolucion	varchar(150)	YES	NULL		
numero1	varchar(150)	YES	NULL		
articulo1	varchar(150)	YES	NULL		
descripcion1	varchar(150)	YES	NULL		
estado1	varchar(150)	YES	NULL		
numero2	varchar(150)	YES	NULL		
articulo2	varchar(150)	YES	NULL		
descripcion2	varchar(150)	YES	NULL		
estado2	varchar(150)	YES	NULL		
numero3	varchar(150)	YES	NULL		
articulo3	varchar(150)	YES	NULL		
descripcion3	varchar(150)	YES	NULL		
estado3	varchar(150)	YES	NULL		
numero4	varchar(150)	YES	NULL		
articulo4	varchar(150)	YES	NULL		
descripcion4	varchar(150)	YES	NULL		
estado4	varchar(150)	YES	NULL		
numero5	varchar(150)	YES	NULL		
articulo5	varchar(150)	YES	NULL		
descripcion5	varchar(150)	YES	NULL		
estado5	varchar(150)	YES	NULL		
id_usuario	int	NO	NULL		
regdate	date	YES	NULL		

5. **Tabla formulario_pedido:** Guarda la información de los pedidos de las instalaciones.

Columnas Tabla formulario_pedido					
Field	Type	Null	Key	Default	Extra
id_formulario_pedido	int(11)	NO	PRI	NULL	auto_increment
fecha_solicitud	date	YES	NULL		
cancha	varchar(150)	YES	NULL		
encargado1	varchar(150)	YES	NULL		
encargado2	varchar(150)	YES	NULL		
dependencia	varchar(150)	YES	NULL		
fecha_actividad	date	YES	NULL		
horario	varchar(150)	YES	NULL		
observaciones	varchar(400)	YES	NULL		
fecha_entrega	date	YES	NULL		
cedula1	varchar(150)	YES	NULL		
nombre1	varchar(300)	YES	NULL		
apellido1	varchar(150)	YES	NULL		
titulacion1	varchar(150)	YES	NULL		
id_usuario	int	NO	NULL		
regdate	date	YES	NULL		

6. **Tabla: formulario_reserva:** Guarda la información correspondiente a las reservas realizadas a las correspondientes instalaciones y canchas.

Columnas Tabla formulario_reserva					
Field	Type	Null	Key	Default	Extra
id_formulario_reserva	int(11)	NO	PRI	NULL	auto_increment
fecha_solicitud	date	YES	NULL	NULL	
cancha	varchar(150)	YES	NULL	NULL	
personaresponsable	varchar(150)	YES	NULL	NULL	
tiporesponsable	varchar(150)	YES	NULL	NULL	
dependencia	varchar(150)	YES	NULL	NULL	
observaciones	varchar(300)	YES	NULL	NULL	
fechaInicioReserva	date	YES	NULL	NULL	
horaInicioReserva	varchar(150)	YES	NULL	NULL	

fechaFinReserva	date	YES	NULL	NULL	
horaFinReserva	varchar(150)	YES	NULL	NULL	
hestudiante	varchar(150)	YES	NULL	NULL	
mestudiante	varchar(150)	YES	NULL	NULL	
hdocente	varchar(150)	YES	NULL	NULL	
mdocente	varchar(150)	YES	NULL	NULL	
hadm	varchar(150)	YES	NULL	NULL	
madm	varchar(10)	YES	NULL	NULL	
persona_entrega	varchar(150)	YES	NULL	NULL	
a1	varchar(10)	YES	NULL	NULL	
c1	varchar(10)	YES	NULL	NULL	
d1	varchar(10)	YES	NULL	NULL	
e1	varchar(10)	YES	NULL	NULL	
a2	varchar(10)	YES	NULL	NULL	
c2	varchar(10)	YES	NULL	NULL	
d2	varchar(10)	YES	NULL	NULL	
e2	varchar(10)	YES	NULL	NULL	
a3	varchar(10)	YES	NULL	NULL	
c3	varchar(10)	YES	NULL	NULL	
d3	varchar(10)	YES	NULL	NULL	
e3	varchar(10)	YES	NULL	NULL	
a4	varchar(10)	YES	NULL	NULL	
c4	varchar(10)	YES	NULL	NULL	
d4	varchar(10)	YES	NULL	NULL	
e4	varchar(10)	YES	NULL	NULL	
observacionesMateriales	varchar(500)	YES	NULL	NULL	
usuarioid	int(11)	NO	NULL	NULL	
regdate	date	YES	NULL	NULL	

7. Tabla: titulación: Guarda la información de las respectivas titulaciones de la universidad.

Columnas Tabla titulación					
Field	Type	Null	Key	Default	Extra
id_titulacion	int(11)	NO	NULL	PRI	auto_increment
nombreCompleto	varchar(500)	YES	NULL		

nombreSistema	varchar(500)	YES	NULL		
---------------	--------------	-----	------	--	--

8. Tabla mes: Guarda los meses del año.

Columnas Tabla mes					
Field	Type	Null	Key	Default	Extra
id_mes	int(11)	NO	PRI	NULL	auto_increment
mes	varchar(250)	NO		NULL	
numero	varchar(10)	NO		NULL	

2.2. Script de la base de datos y las tablas

A continuación se muestra los scripts que se utilizaron para la creación de la base de datos con sus respectivas tablas.

```
SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
```

```
SET time_zone = "+00:00";
```

```
/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
```

```
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS
*/;
```

```
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
```

```
/*!40101 SET NAMES utf8 */;
```

```
--
```

```
-- Base de datos: `proyectoutpl`
```

```
--
```

```
-----
```

```
--
```

```
-- Estructura de tabla para la tabla `canchas`
```

```
--
```

```
CREATE TABLE IF NOT EXISTS `canchas` (
```

```
  `id_canchas` int(11) NOT NULL AUTO_INCREMENT,
```

```
  `nombre` varchar(100) COLLATE utf8_spanish_ci DEFAULT NULL,
```

```
  `capacidad` varchar(100) COLLATE utf8_spanish_ci DEFAULT NULL,
```

```
  `estado` varchar(100) COLLATE utf8_spanish_ci DEFAULT NULL,
```

```
  PRIMARY KEY (`id_canchas`)
```

```
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_spanish_ci
```

```
AUTO_INCREMENT=14 ;
```

--

-- Estructura de tabla para la tabla `formulario_material`

```
CREATE TABLE IF NOT EXISTS `formulario_material` (  
  `id_formulario_material` int(11) NOT NULL AUTO_INCREMENT,  
  `persona_entrega` varchar(150) COLLATE utf8_spanish_ci DEFAULT NULL,  
  `persona_recibe` varchar(150) COLLATE utf8_spanish_ci DEFAULT NULL,  
  `fecha_entrega` date DEFAULT NULL,  
  `hora_entrega` varchar(150) COLLATE utf8_spanish_ci DEFAULT NULL,  
  `fecha_devolucion` date DEFAULT NULL,  
  `hora_devolucion` varchar(150) COLLATE utf8_spanish_ci DEFAULT NULL,  
  `numero1` varchar(150) COLLATE utf8_spanish_ci DEFAULT NULL,  
  `articulo1` varchar(150) COLLATE utf8_spanish_ci DEFAULT NULL,  
  `descripcion1` varchar(150) COLLATE utf8_spanish_ci DEFAULT NULL,  
  `estado1` varchar(150) COLLATE utf8_spanish_ci DEFAULT NULL,  
  `numero2` varchar(150) COLLATE utf8_spanish_ci DEFAULT NULL,  
  `articulo2` varchar(150) COLLATE utf8_spanish_ci DEFAULT NULL,  
  `descripcion2` varchar(150) COLLATE utf8_spanish_ci DEFAULT NULL,  
  `estado2` varchar(150) COLLATE utf8_spanish_ci DEFAULT NULL,  
  `numero3` varchar(150) COLLATE utf8_spanish_ci DEFAULT NULL,  
  `articulo3` varchar(150) COLLATE utf8_spanish_ci DEFAULT NULL,  
  `descripcion3` varchar(150) COLLATE utf8_spanish_ci DEFAULT NULL,  
  `estado3` varchar(150) COLLATE utf8_spanish_ci DEFAULT NULL,  
  `numero4` varchar(150) COLLATE utf8_spanish_ci DEFAULT NULL,  
  `articulo4` varchar(150) COLLATE utf8_spanish_ci DEFAULT NULL,  
  `descripcion4` varchar(150) COLLATE utf8_spanish_ci DEFAULT NULL,  
  `estado4` varchar(150) COLLATE utf8_spanish_ci DEFAULT NULL,  
  `numero5` varchar(150) COLLATE utf8_spanish_ci DEFAULT NULL,  
  `articulo5` varchar(150) COLLATE utf8_spanish_ci DEFAULT NULL,  
  `descripcion5` varchar(150) COLLATE utf8_spanish_ci DEFAULT NULL,  
  `estado5` varchar(150) COLLATE utf8_spanish_ci DEFAULT NULL,  
  `id_usuario` int(11) NOT NULL,  
  `regdate` date DEFAULT NULL,  
  PRIMARY KEY (`id_formulario_material`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_spanish_ci  
AUTO_INCREMENT=10;
```

-- Estructura de tabla para la tabla `formulario_pedido`

```
CREATE TABLE IF NOT EXISTS `formulario_pedido` (  
  `id_formulario_pedido` int(11) NOT NULL AUTO_INCREMENT,  
  `fecha_solicitud` date DEFAULT NULL,  
  `cancha` varchar(150) COLLATE utf8_spanish_ci DEFAULT NULL,  
  `encargado1` varchar(150) COLLATE utf8_spanish_ci DEFAULT NULL,  
  `encargado2` varchar(150) COLLATE utf8_spanish_ci DEFAULT NULL,  
  `dependencia` varchar(150) COLLATE utf8_spanish_ci DEFAULT NULL,  
  `fecha_actividad` date DEFAULT NULL,  
  `horario` varchar(150) COLLATE utf8_spanish_ci DEFAULT NULL,  
  `observaciones` varchar(400) COLLATE utf8_spanish_ci DEFAULT NULL,  
  `fecha_entrega` date DEFAULT NULL,  
  `cedula1` varchar(150) COLLATE utf8_spanish_ci DEFAULT NULL,  
  `nombre1` varchar(300) COLLATE utf8_spanish_ci DEFAULT NULL,  
  `apellido1` varchar(150) COLLATE utf8_spanish_ci DEFAULT NULL,  
  `titulacion1` varchar(150) COLLATE utf8_spanish_ci DEFAULT NULL,  
  `id_usuario` int(11) NOT NULL,  
  `regdate` date DEFAULT NULL,  
  PRIMARY KEY (`id_formulario_pedido`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_spanish_ci  
AUTO_INCREMENT=9;
```

-- Estructura de tabla para la tabla `formulario_reserva`

```
CREATE TABLE IF NOT EXISTS `formulario_reserva` (  
  `id_formulario_reserva` int(11) NOT NULL AUTO_INCREMENT,  
  `fecha_solicitud` date DEFAULT NULL,  
  `cancha` varchar(150) COLLATE utf8_spanish_ci DEFAULT NULL,  
  `personaresponsable` varchar(150) COLLATE utf8_spanish_ci DEFAULT NULL,  
  `tiporesponsable` varchar(150) COLLATE utf8_spanish_ci DEFAULT NULL,  
  `dependencia` varchar(150) COLLATE utf8_spanish_ci DEFAULT NULL,
```



```

`observaciones` varchar(300) COLLATE utf8_spanish_ci DEFAULT NULL,
`fechaInicioReserva` date DEFAULT NULL,
`horaInicioReserva` varchar(150) COLLATE utf8_spanish_ci DEFAULT NULL,
`fechaFinReserva` date DEFAULT NULL,
`horaFinReserva` varchar(150) COLLATE utf8_spanish_ci DEFAULT NULL,
`hestudiante` varchar(150) COLLATE utf8_spanish_ci DEFAULT NULL,
`mestudiante` varchar(150) COLLATE utf8_spanish_ci DEFAULT NULL,
`hdocente` varchar(150) COLLATE utf8_spanish_ci DEFAULT NULL,
`mdocente` varchar(150) COLLATE utf8_spanish_ci DEFAULT NULL,
`hadm` varchar(150) COLLATE utf8_spanish_ci DEFAULT NULL,
`madm` varchar(10) COLLATE utf8_spanish_ci DEFAULT NULL,
`persona_entrega` varchar(150) COLLATE utf8_spanish_ci DEFAULT NULL,
`a1` varchar(10) COLLATE utf8_spanish_ci DEFAULT NULL,
`c1` varchar(10) COLLATE utf8_spanish_ci DEFAULT NULL,
`d1` varchar(10) COLLATE utf8_spanish_ci DEFAULT NULL,
`e1` varchar(10) COLLATE utf8_spanish_ci DEFAULT NULL,
`a2` varchar(10) COLLATE utf8_spanish_ci DEFAULT NULL,
`c2` varchar(10) COLLATE utf8_spanish_ci DEFAULT NULL,
`d2` varchar(10) COLLATE utf8_spanish_ci DEFAULT NULL,
`e2` varchar(10) COLLATE utf8_spanish_ci DEFAULT NULL,
`a3` varchar(10) COLLATE utf8_spanish_ci DEFAULT NULL,
`c3` varchar(10) COLLATE utf8_spanish_ci DEFAULT NULL,
`d3` varchar(10) COLLATE utf8_spanish_ci DEFAULT NULL,
`e3` varchar(10) COLLATE utf8_spanish_ci DEFAULT NULL,
`a4` varchar(10) COLLATE utf8_spanish_ci DEFAULT NULL,
`c4` varchar(10) COLLATE utf8_spanish_ci DEFAULT NULL,
`d4` varchar(10) COLLATE utf8_spanish_ci DEFAULT NULL,
`e4` varchar(10) COLLATE utf8_spanish_ci DEFAULT NULL,
`observacionesMateriales` varchar(500) COLLATE utf8_spanish_ci DEFAULT NULL,
`usuarioid` int(11) NOT NULL,
`regdate` date DEFAULT NULL,
PRIMARY KEY (`id_formulario_reserva`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_spanish_ci
AUTO_INCREMENT=9 ;

```

-- Estructura de tabla para la tabla `meses`

```
CREATE TABLE IF NOT EXISTS `meses` (  
  `id_mes` int(11) NOT NULL AUTO_INCREMENT,  
  `mes` varchar(250) CHARACTER SET utf8 COLLATE utf8_spanish_ci NOT NULL,  
  `numero` varchar(10) NOT NULL,  
  PRIMARY KEY (`id_mes`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=13 ;
```

--

```
INSERT INTO `meses` (`id_mes`, `mes`, `numero`) VALUES
```

```
(1, 'Enero', '01'),  
(2, 'Febrero', '02'),  
(3, 'Marzo', '03'),  
(4, 'Abril', '04'),  
(5, 'Mayo', '05'),  
(6, 'Junio', '06'),  
(7, 'Julio', '07'),  
(8, 'Agosto', '08'),  
(9, 'Septiembre', '09'),  
(10, 'Octubre', '10'),  
(11, 'Noviembre', '11'),  
(12, 'Diciembre', '12');
```

-- Estructura de tabla para la tabla `reservas`

```
CREATE TABLE IF NOT EXISTS `reservas` (  
  `id_reservas` int(11) NOT NULL AUTO_INCREMENT,  
  `nombre_canchas` varchar(100) CHARACTER SET utf8 COLLATE utf8_spanish_ci NOT  
NULL,  
  `start_date` datetime NOT NULL,  
  `end_date` datetime NOT NULL,  
  PRIMARY KEY (`id_reservas`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=33 ;
```

-- Estructura de tabla para la tabla `usuarios`

```

CREATE TABLE IF NOT EXISTS `usuarios` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `nombre` varchar(150) COLLATE utf8_spanish_ci DEFAULT NULL,
  `apellido` varchar(150) COLLATE utf8_spanish_ci DEFAULT NULL,
  `email` varchar(150) COLLATE utf8_spanish_ci DEFAULT NULL,
  `telefono` varchar(150) COLLATE utf8_spanish_ci DEFAULT NULL,
  `password` varchar(300) COLLATE utf8_spanish_ci DEFAULT NULL,
  `rol` varchar(150) COLLATE utf8_spanish_ci NOT NULL,
  `regdate` date DEFAULT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_spanish_ci
AUTO_INCREMENT=6 ;

--

/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
-----
-- Estructura de tabla para la tabla `titulacion`

CREATE TABLE IF NOT EXISTS `titulacion` (
  `id_titulacion` int(11) NOT NULL AUTO_INCREMENT,
  `nombreCompleto` varchar(500) COLLATE utf8_spanish_ci DEFAULT NULL,
  `nombreSistema` varchar(500) COLLATE utf8_spanish_ci DEFAULT NULL,
  PRIMARY KEY (`id_titulacion`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_spanish_ci
AUTO_INCREMENT=64 ;

-- Volcado de datos para la tabla `titulacion`

INSERT INTO `titulacion` (`id_titulacion`, `nombreCompleto`, `nombreSistema`) VALUES
(1, 'TITULACION ARQUITECTURA', 'Arquitectura'),
(2, 'TITULACION ARTES PLASTICAS Y DISE', 'Artes Plasticas y Dise'),
(3, 'TITULACION ELECTRONICA Y TELECOMUNICACIONES', 'Electronica y
Telecomunicaciones'),

```

- (4, 'TITULACION GEOLOGIA Y MINAS', 'Geologia y Minas'),
- (5, 'TITULACION INGENIERIA CIVIL', 'Ingenieria Civil'),
- (6, 'TITULACION INGENIERIA SISTEMAS INFORMATICOS', 'Sistemas Informaticos'),
- (7, 'TITULACION GESTION PUBLICA', 'Gestion Publica'),
- (8, 'POSTGRADOS AREA ADMINISTRATIVA UTPL', 'Postg. Administrativa'),
- (9, 'TITULACION DE GASTRONOMIA', 'Gastronomia'),
- (10, 'TITULACION ADMINISTRACION DE EMPRESAS', 'Administracion Empresas'),
- (11, 'TITULACION ECONOMIA', 'Economia'),
- (12, 'TITULACION ADMINISTRACION EN BANCA Y FINANZAS', 'Adminsiutracion Banca y Finanzas'),
- (13, 'TITULACION INGENIERIA EN CONTABILIDAD Y AUDITORIA', 'Contabilidad y Auditoria'),
- (14, 'TITULACION INGENIERIA EN ADMINISTRACION DE EMPRESAS TURISTICAS Y HOTELERAS', 'Administracion Turismo y Hoteleria'),
- (15, 'TITULACION ASISTENCIA GERENCIAL Y RELACIONES PUBLICAS', 'Asistencia Gerencial y Relaciones Publicas'),
- (16, 'TITULACION RELACIONES PUBLICAS', 'Relaciones Publicas'),
- (17, 'TITULACION ENFERMERIA', 'Enfermeria'),
- (18, 'TITULACION INGENIERIA INDUSTRIAL', 'Ingenieria Industrial'),
- (19, 'POSTGRADOS AREA TECNICA', 'Postg. Area Tecnica'),
- (20, 'POSTGRADOS AREA BIOLOGICA', 'Postg. Area Biologia'),
- (21, 'TITULACION DE INGENIERO EN ALIMENTOS', 'Ingenieria Alimentos'),
- (22, 'TITULACION BIOQUIMICA Y FARMACIA', 'Bioquimica y Farmacia'),
- (23, 'TITULACION GESTION AMBIENTAL', 'Gastion Ambiental'),
- (24, 'TITULACION INGENIERIA QUIMICA', 'Ingenieria Quimica'),
- (25, 'TITULACION MEDICINA', 'Medicina'),
- (26, 'TITULACION BIOLOGIA', 'Biologia'),
- (27, 'TITULACION INGENIERIA AGROPECUARIA', 'Ingenieria Agropecuaria'),
- (28, 'POSTGRADOS AREA SOCIOHUMANISTICA', 'Postg. Area Sociohumanistica'),
- (29, 'TITULACION CIENCIAS DE LA EDUCACION MENCION LENGUA Y LITERATURA', 'Educacion Lengua y Literatura'),
- (30, 'TITULACION DERECHO', 'Derecho'),
- (31, 'TITULACION PSICOLOGIA', 'Psicologia'),
- (32, 'TITULACION CIENCIAS DE LA EDUCACION MENCION EDUCACION BASICA', 'Educacion Mencion Ed. Basica'),

- (33, 'TITULACION CIENCIAS DE LA EDUCACION MENCION CIENCIAS HUMANAS Y RELIGIOSAS', 'Educacion Mencion Ed. Ciencias Humanas'),
- (34, 'TITULACION CIENCIAS DE LA EDUCACION MENCION INGLES', 'Educacion Mencion Ingles'),
- (35, 'TITULACION COMUNICACION SOCIAL', 'Comunicaci'),
- (36, 'TITULACION CIENCIAS DE LA EDUCACION MENCION EDUCACION INFANTIL', 'Educacion Mencion Ed. Social'),
- (37, 'TITULACION CIENCIAS DE LA EDUCACION MENCION FISICO MATEMATICAS', 'Educacion Mencion Fisico Matematico'),
- (38, 'TITULACION CIENCIAS DE LA EDUCACION MENCION QUIMICO BIOLOGICAS', 'Educacion Mencion Quimico Biologico');

```

/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;

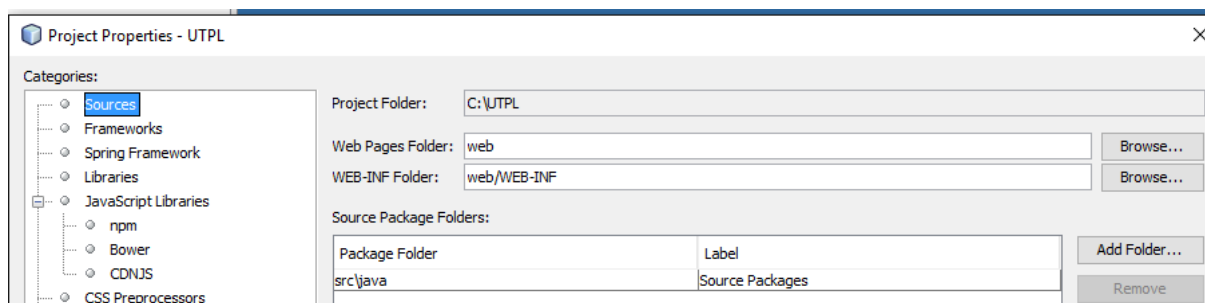
```

3. Revisión de la creación de los módulos de la aplicación.

Para la creación de los respectivos módulos y su funcionamiento de la aplicación es imprescindible que

- Esté instalado el servidor de base de datos MySQL,
- Crear la base de datos proyectoutpl.

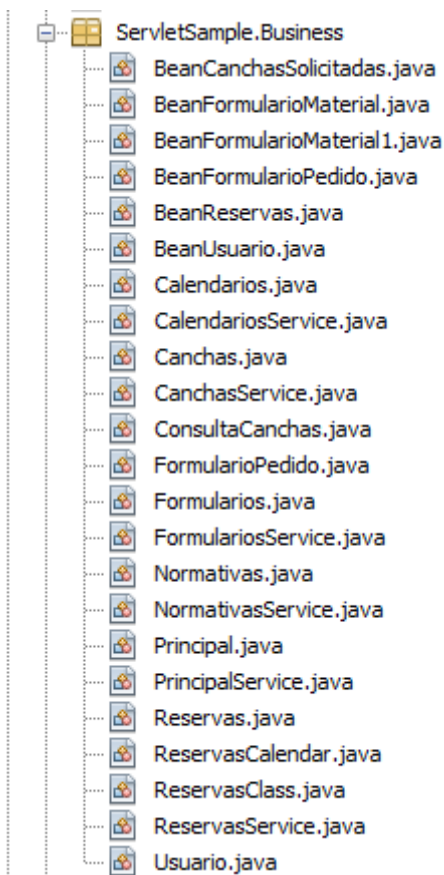
Podemos encontrar todo el código de la aplicación en la carpeta \UTPL\build\web, que incluye los archivos de cada uno de los respectivos módulos y de las capas del modelo MVC.



4. Explicación de la implementación

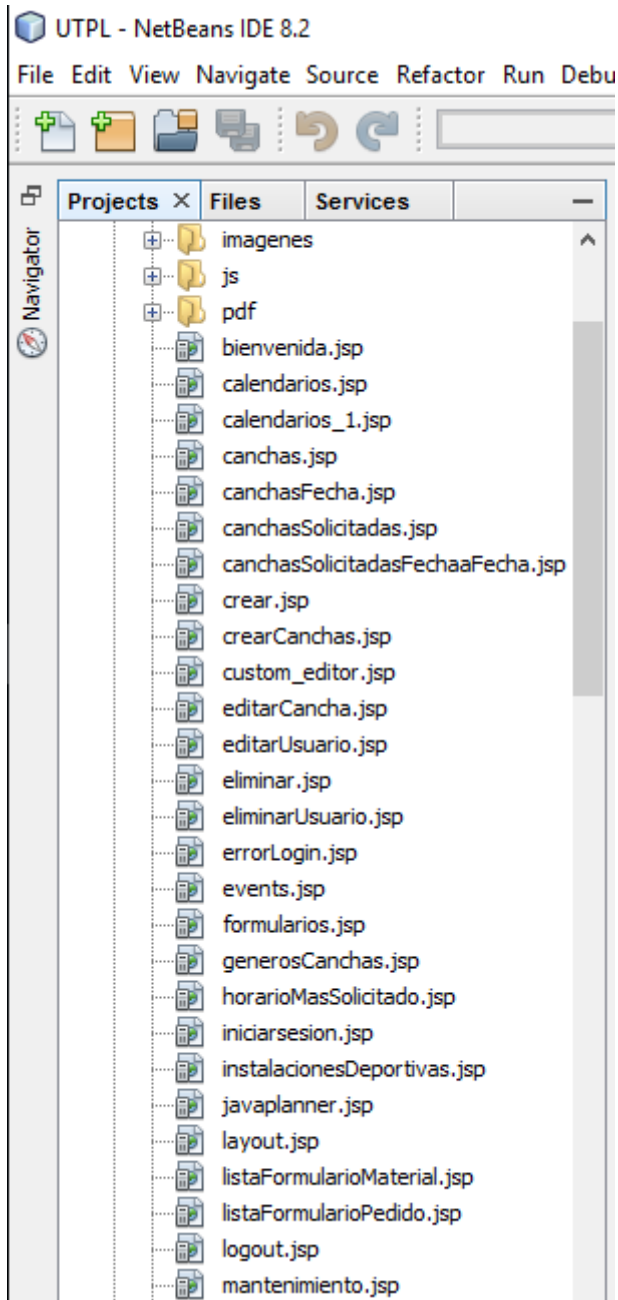
Para la implementación de la aplicación se utilizó la arquitectura MVC, y en función de los respectivos diagramas y en especial del de clases se crearon las funciones, clases, procedimientos.

Como se puede observar en las tres figuras siguientes están los respectivos archivos de las tres capas del modelo MVC, así logramos la separación del código como nos indica esta arquitectura, evitando así tener el denominado código espagueti. Con este modelo si se requiere hacer alguna revisión del contenido que queremos que se muestre en la aplicación no estamos obligados a revisar todo y cada uno de los archivos la aplicación



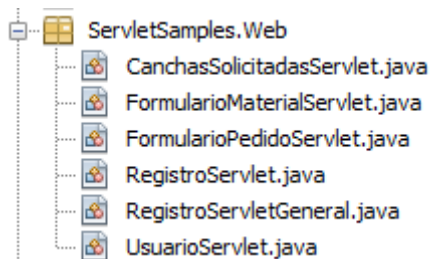
Archivos de la Capa Modelo

En esta parte es con la que trabajamos con los datos, es aquí donde mayormente se trabaja con los Select, Update, Insert, etc.



Archivos de la capa Vista

En esta capa se abarca el código de la aplicación que va a crear la visualización de las interfaces del usuario, en esta capa tenemos los códigos HTML que nos muestran las salidas de los datos, esta capa vamos a trabajar con los datos pero no vamos a realizar un acceso directo a ellos, ya que esta (Capa Vista) solicitara los datos a la capa del Modelo



Archivos de la Capa Controlador

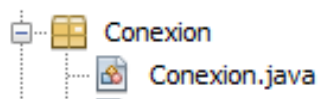
La capa Controlador incluye el código necesario para contestar a las acciones que se solicitan en la aplicación. Esta capa se la utiliza de enlace entre la capa de las Vista y la capa de los Modelos

4.1. Explicación de la capa de persistencia de la aplicación

En esta parte primeramente debemos tomar en cuenta que netbeans debe estar instalada la versión full, utilizamos el servidor swanserver para levantar la base de datos y tenerla arriba.

EL servidor Apache Tomcat se utiliza para la implementación de los servlets jsp;

Capa de persistencia es la parte de integración con la base de datos, donde tenemos el componente Conexión.java, que es clase que realiza la conexión con la base de datos..



Archivos de Conexión hacia la BD

A este archivo se hace referencia cuando es necesario establecer la conexión de la aplicación con la base de datos.

ANEXO 5
MANUAL DE INSTALACION

Introducción

El presente manual forma parte del conjunto de herramientas a ser utilizado en el Aplicativo para la Asignación de y prestamos de instalaciones deportivas de la UTPL, y nos guiara en el uso del programa que administra las diferentes bases de datos que lo conforman.

Este manual de instalación y configuración del aplicativo, da instrucciones claras y afables que facilitan el proceso de instalación.

Para la instalación del aplicativo se necesita de un equipo que entregue los servicios web, así como de base de datos.

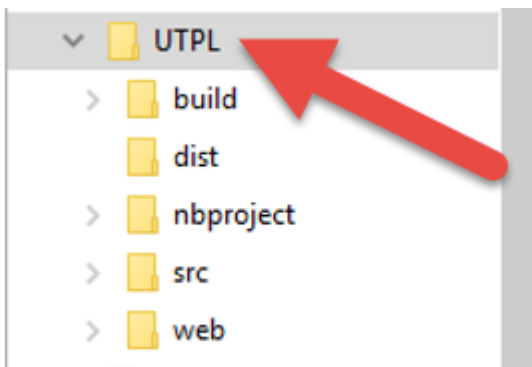
Requerimientos de hardware y software.

- Windows 7 o Windows Server 2008 (como básico) con las
 - 4 GB de memoria RAM
 - 3.1 GHz de velocidad de procesamiento
 - Disco Duro de 500 GB
 - Adaptador de red 100/1000.
 - Monitor, teclado, mouse.
 - Puertos USB.

Salida a internet, y conectado a la red LAN de la universidad.

Instalación y configuración.

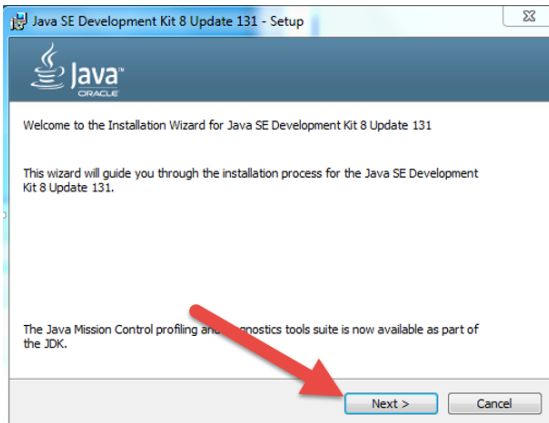
Se debe copiar el directorio que contiene todos los archivos de la aplicación en la unidad de disco duro del computador que contiene los archivos en la raíz del disco C:



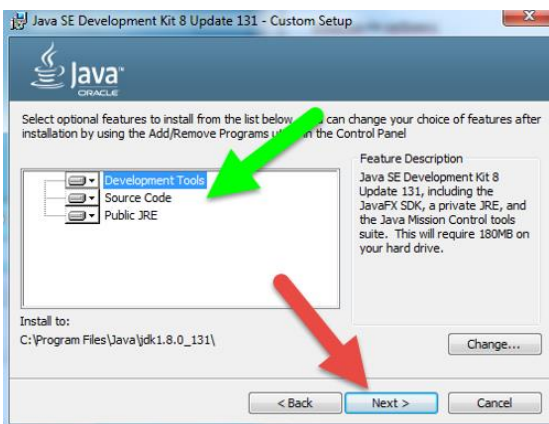
En primer lugar instalamos el Netbeans, previamente se debe tener instalado la máquina virtual de java y el JSK.

Instalación JDK

Instalación del paquete JDK para Windows



Damos clic en Next



Escogemos todas los features por defecto, y damos en Next



Una vez terminado el proceso cerramos el proceso de instalación.

Instalación del NetBeans

Después instalamos del JDK instalamos el Netbeans que es nuestra interfaz de desarrollo, instalamos la versión Full, la cual la podemos descargar directamente de la página.

NetBeans IDE 8.2 Download

8.1 | 8.2 | Development | Archive

Email address (optional):

Subscribe to newsletters: Monthly Weekly

NetBeans can contact me at this address

IDE Language: English Platform: Windows

Note: Greyed out technologies are not supported for this platform.

NetBeans IDE Download Bundles

Supported technologies *	Java SE	Java EE	HTML5/JavaScript	PHP	C/C++	All
NetBeans Platform SDK	•	•				•
Java SE	•					•
Java FX	•	•				•
Java EE		•				•
Java ME						•
HTML5/JavaScript		•	•	•		•
PHP			•	•		•
C/C++					•	•
Groovy						•
Java Card™ 3 Connected						•
Bundled servers						
GlassFish Server Open Source Edition 4.1.1		•				•
Apache Tomcat 8.0.27		•				•

Download buttons: Download, Download, Download x86, Download x86, Download x86, Download x64, Download x64, Download x64, Download

Free, 95 MB Free, 197 MB Free, 108 - 112 MB Free, 108 - 112 MB Free, 107 - 110 MB Free, 221 MB

Instalamos el netbeans

Una vez iniciada la instalación damos clic en Next

NetBeans IDE Installer

Configuring the Installer...

Please wait while NetBeans IDE installer initializes

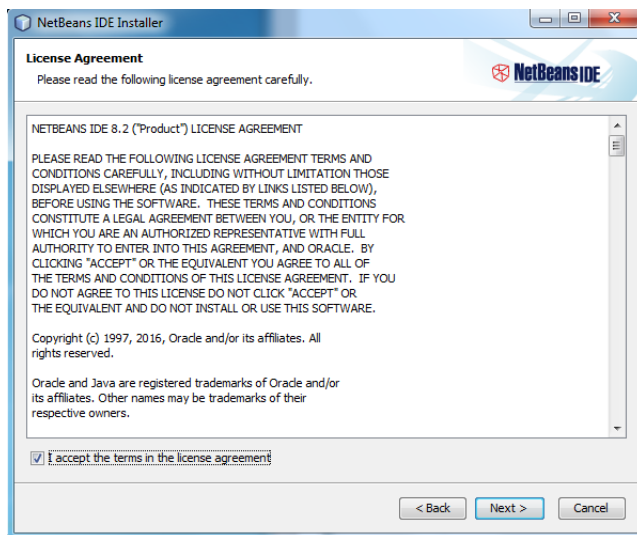
Configuring the Installer... 10%

Next > Cancel

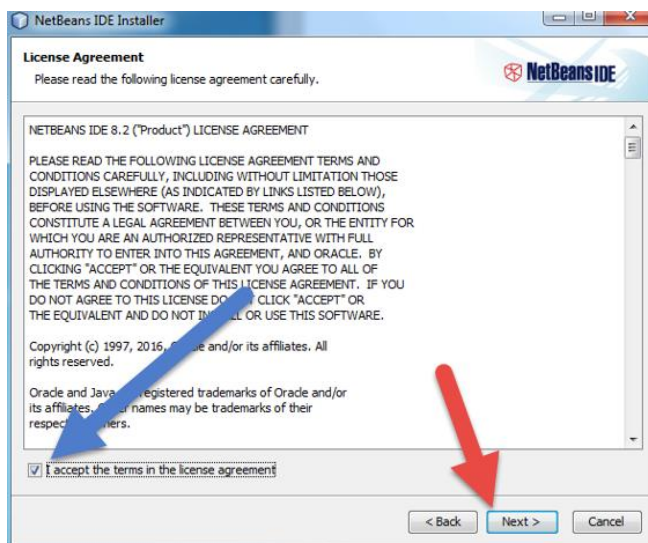
Damos click en Next

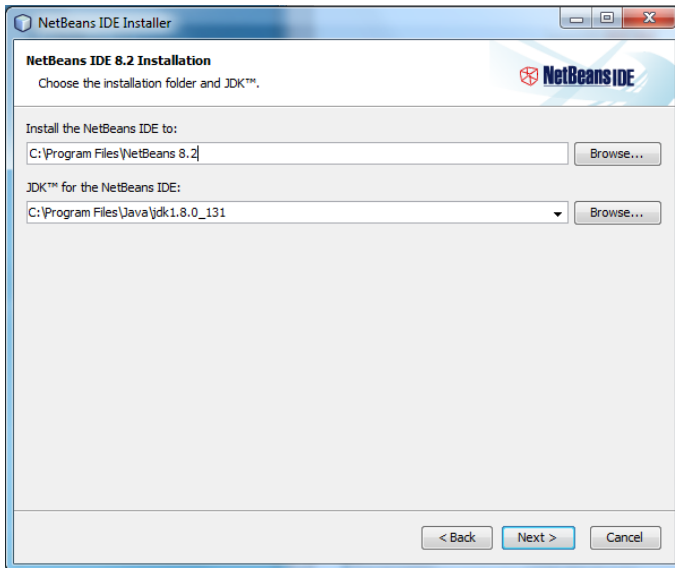


Aceptamos los términos de uso y damos Next

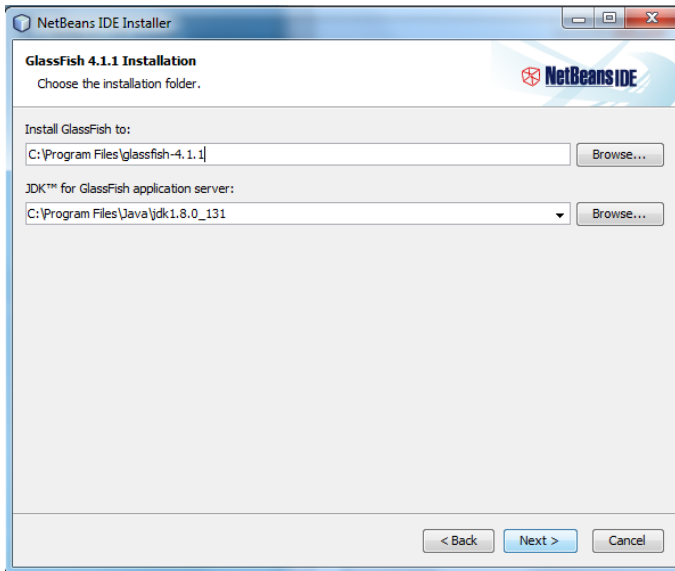


Dejamos por defecto y damos Next

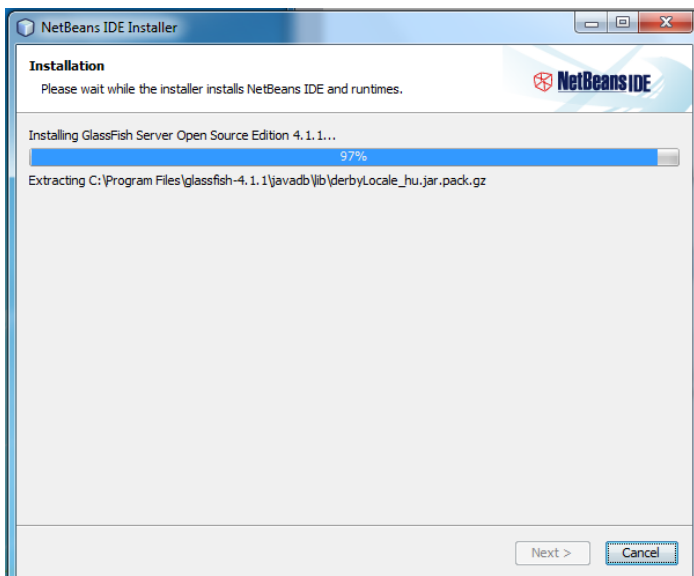




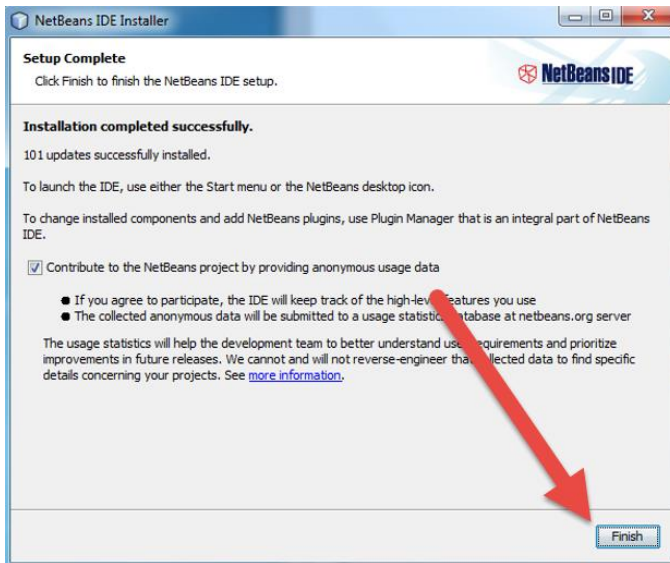
Next



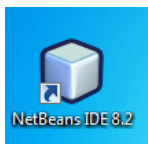
Next



Damos clic en Finish



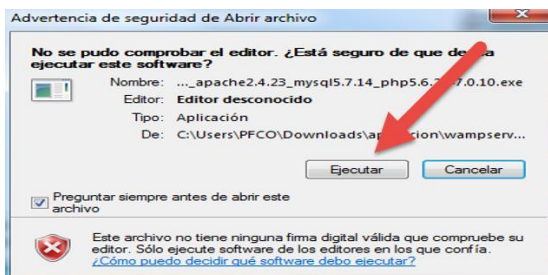
Ya tenemos ya instalado el Netbeans



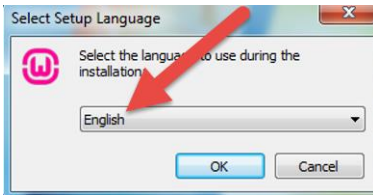
Instalación del WampServer

Instalación Wampserver que tiene las herramientas para la gestión de la base de datos.

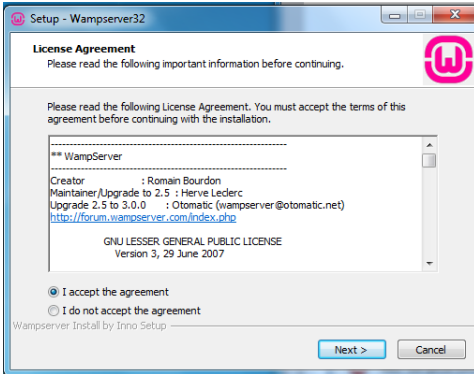
Ejecutamos el instalador



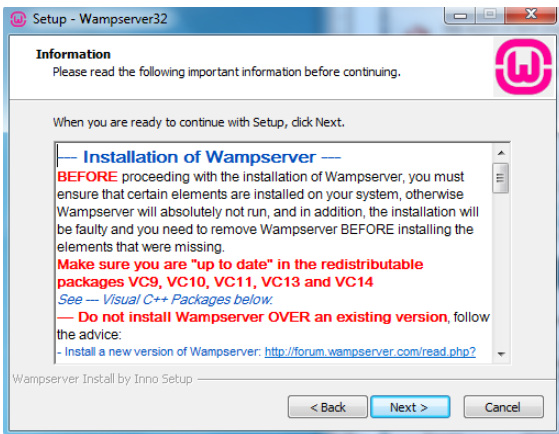
Escogemos el idioma en que queremos que se instale el programa y damos en OK (al no venir en español escogemos en idioma ingles)



Next

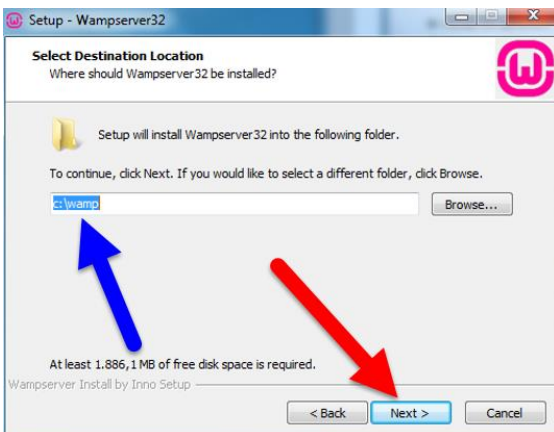


Next

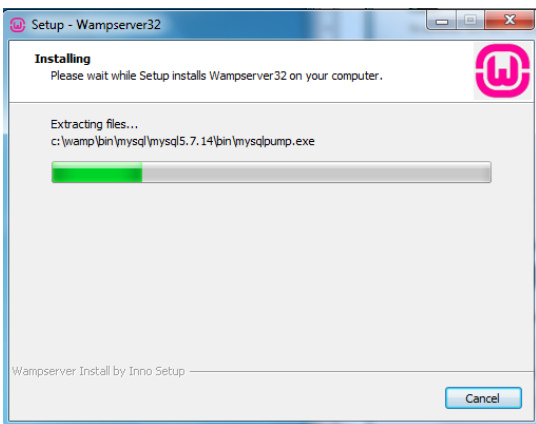
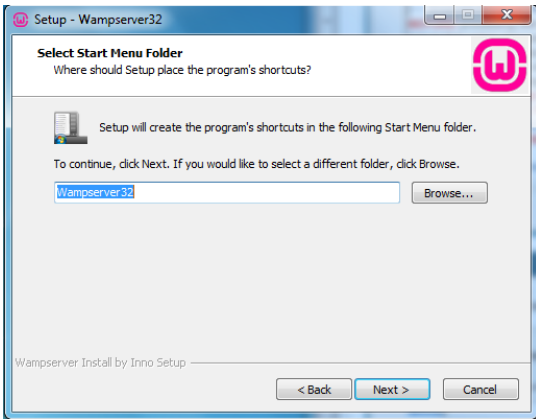


Next

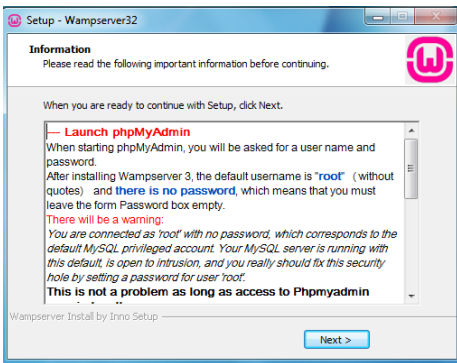
Es esta parte de la instalación escogemos la dirección donde se va a instalar, en este caso le dejamos la dirección por defecto que es la raíz del disco C:, y damos en Next.



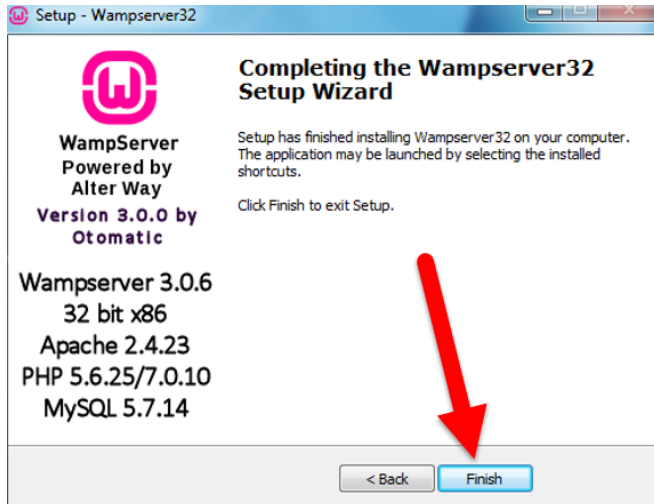
Next



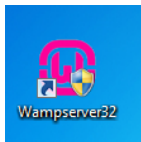
Next



Damos en Finish, para terminar la instalación.



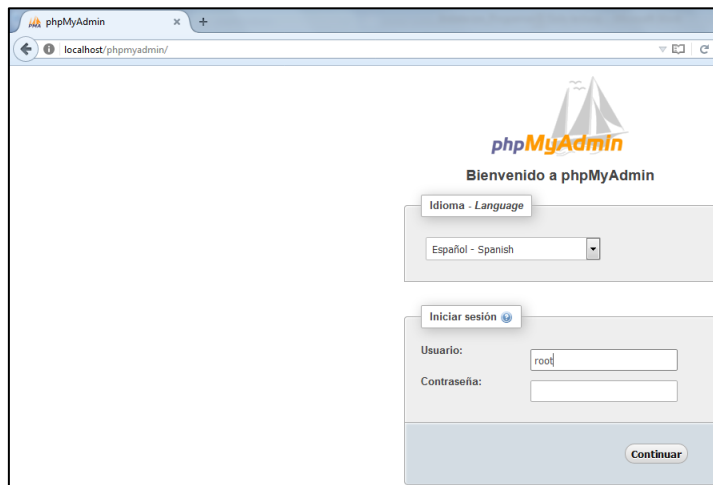
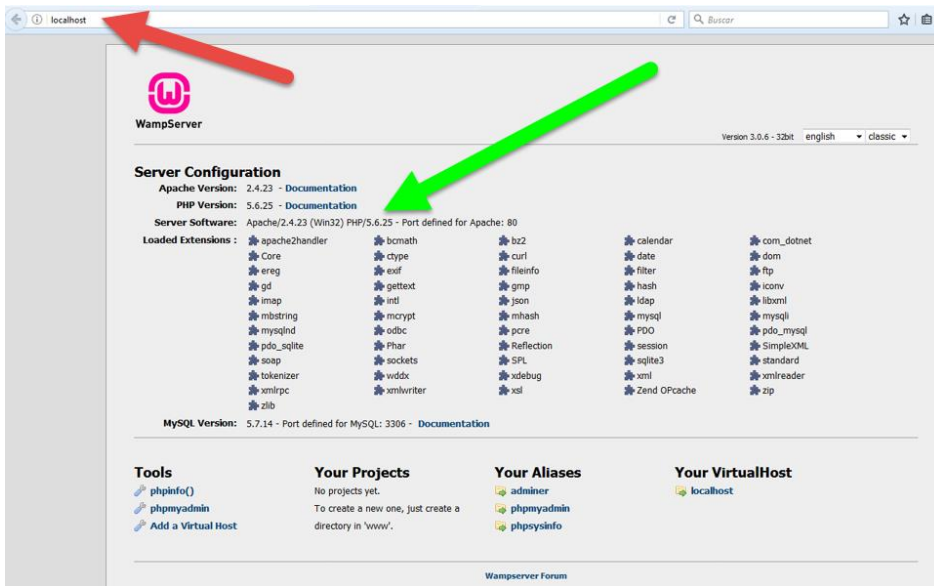
Y tenemos instalado el Wampserver



Cuando ya está instalado y configurado aparece el icono de color verde, esto nos indica que los servicios están corriendo con normalidad, si este icono se presenta en color naranja o rojo, hay que volver a instalar.



Validamos que podemos ingresar al localhost, ya al phpmyadmin



Poner clave al usuario root

Cambiamos la clave al usuario root, por seguridad de la aplicación y de nuestra base de datos



Cambio de contraseña

Sin contraseña

Contraseña:

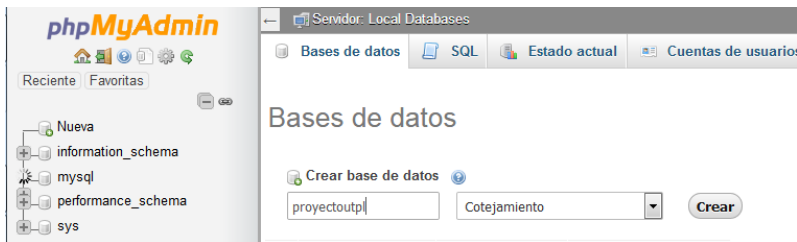
•••••

Debe volver a escribir:

•••••

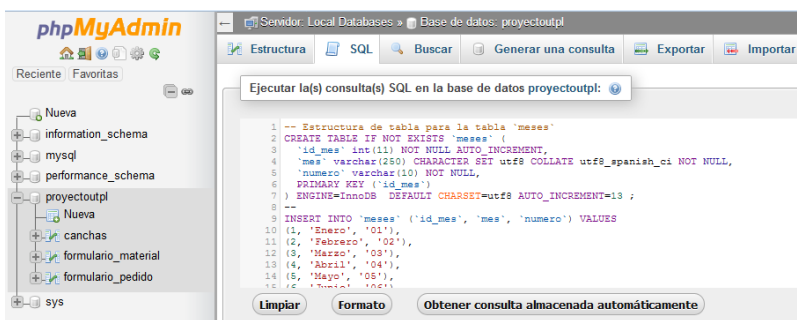
Creamos la base de datos

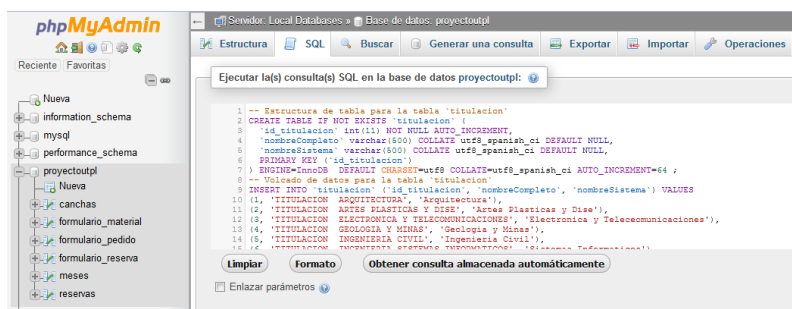
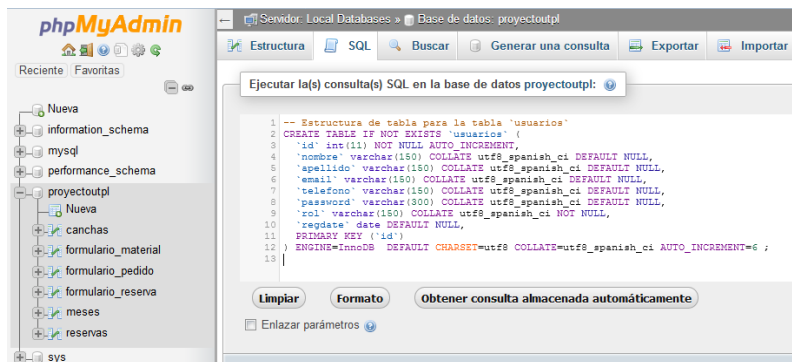
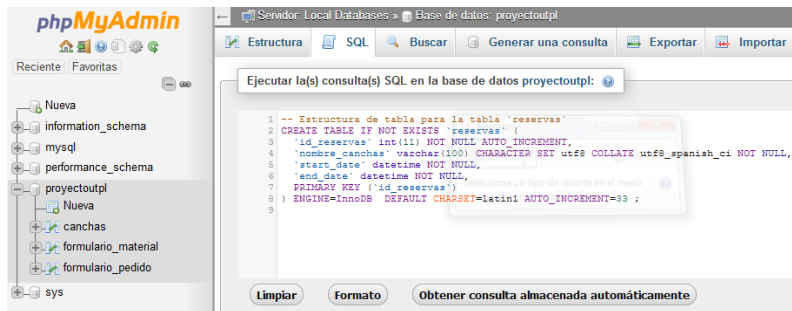
Para que empiece a operar la aplicación debe estar creada la base de datos donde se va a guardar la información.



Creación de las tablas de la base de datos.

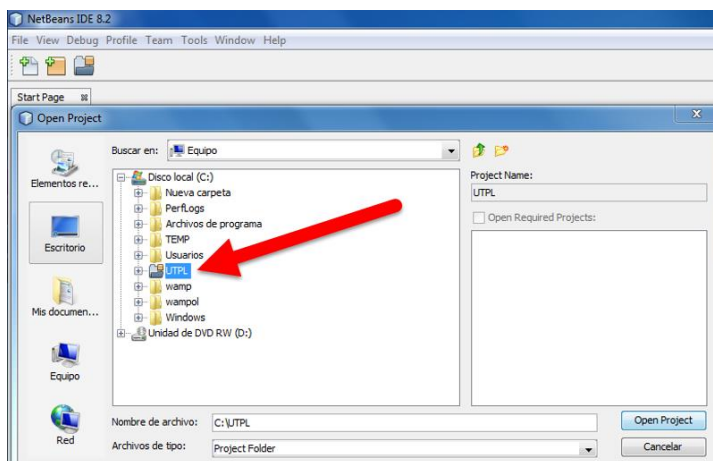
Y procedemos a crear las tablas del aplicativo.





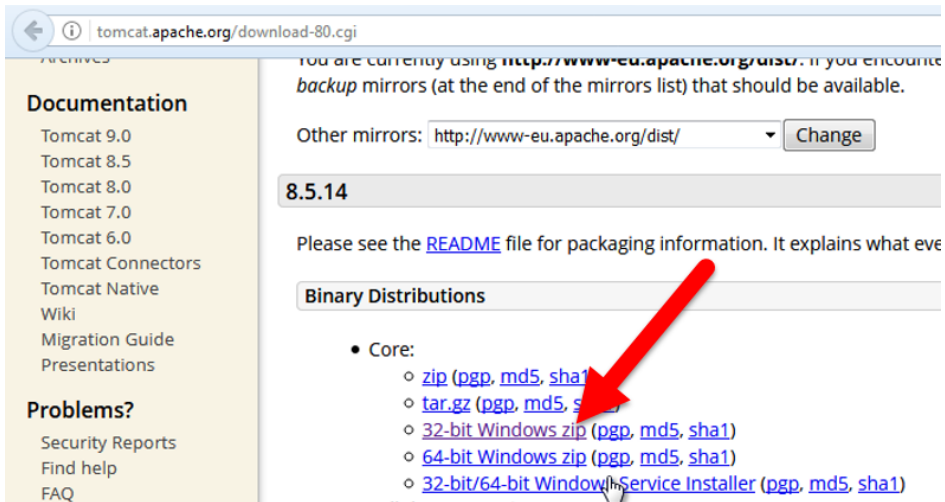
Configuración el Netbeans

Como copiamos previamente el directorio de los archivos de nuestro aplicativo, procedemos a realizar la configuración del NetBeans, donde seleccionamos la carpeta de nuestro proyecto.

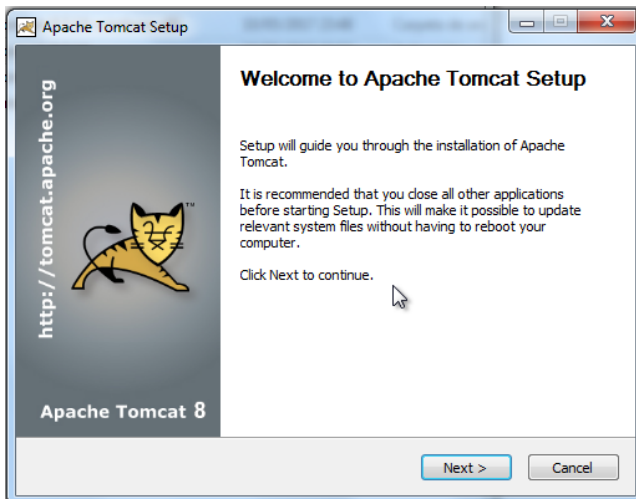


Instalación del Web Server Tomcat

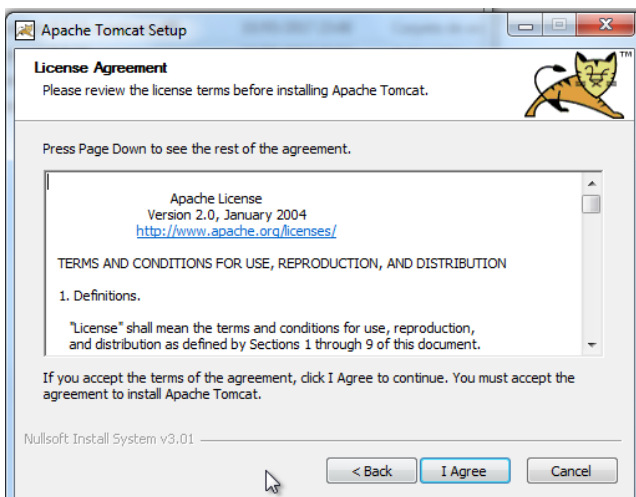
Para esto se descarga el servidor Web Tomcat directamente de su página Web.



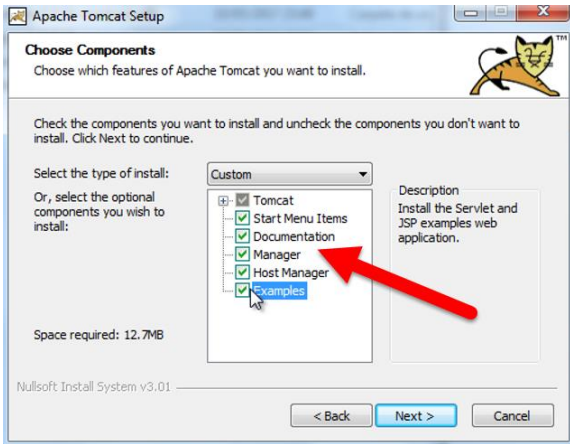
Y después de la descarga se procede a instalar.



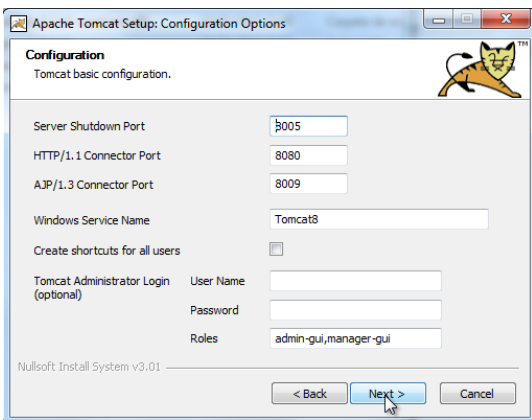
Next



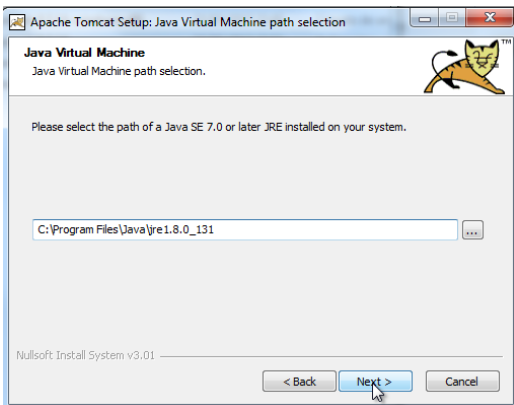
Escogemos los componentes por defecto y damos en Next



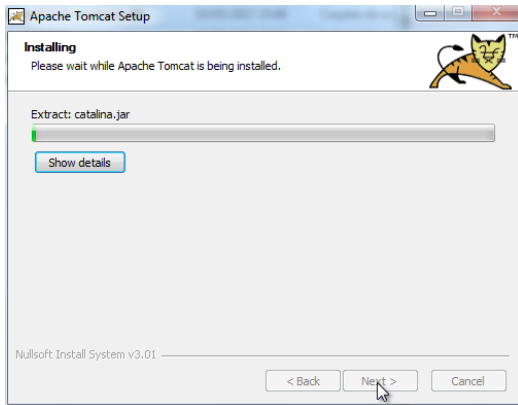
Next



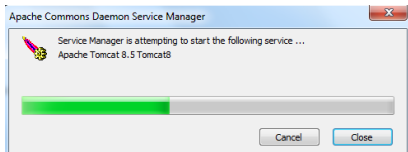
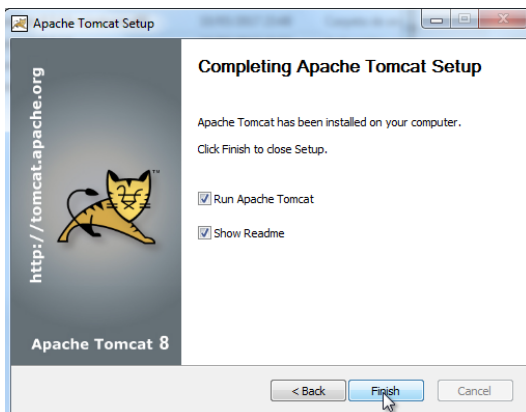
Next



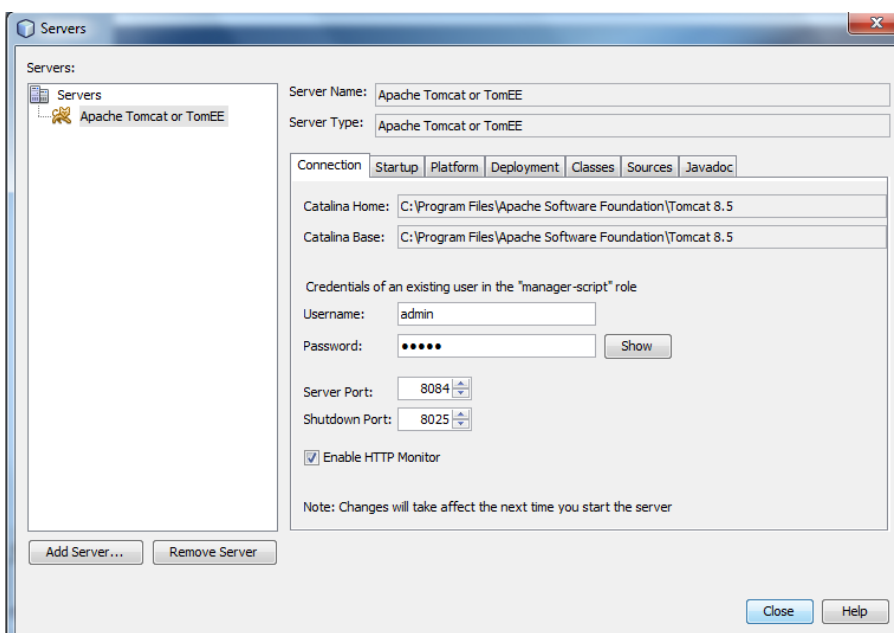
Next



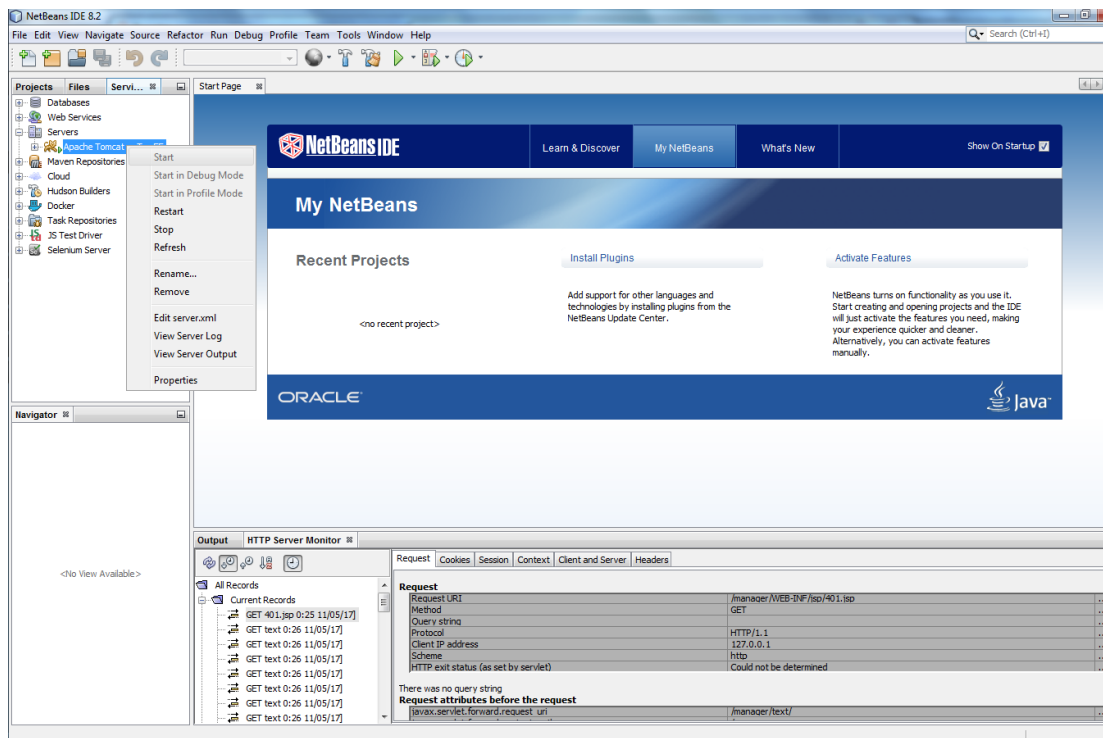
Finish



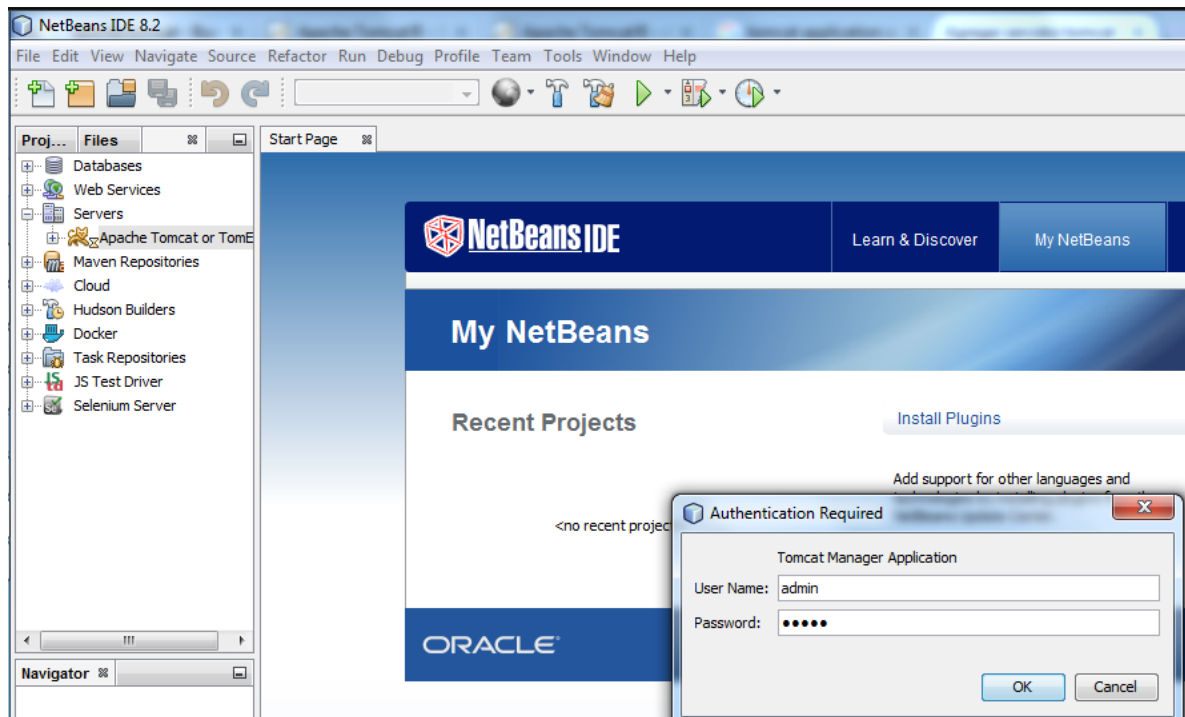
Una vez descargado e instalado el servidor web todo por default, no se realiza configuraciones adicionales, de esta manera se agrega el servidor web a nuestro netbeans.



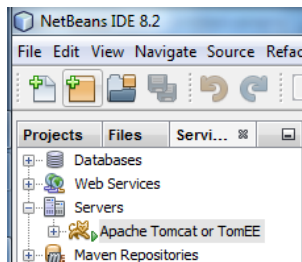
Una vez agregado el servidor damos clic en Start para iniciar el servicio quedando como muestra la siguiente figura, cuando ya está configurado.



Una vez que se inicia este servicio nos pide el usuario y la clave de configuración en nuestro caso es el usuario: *admin*.



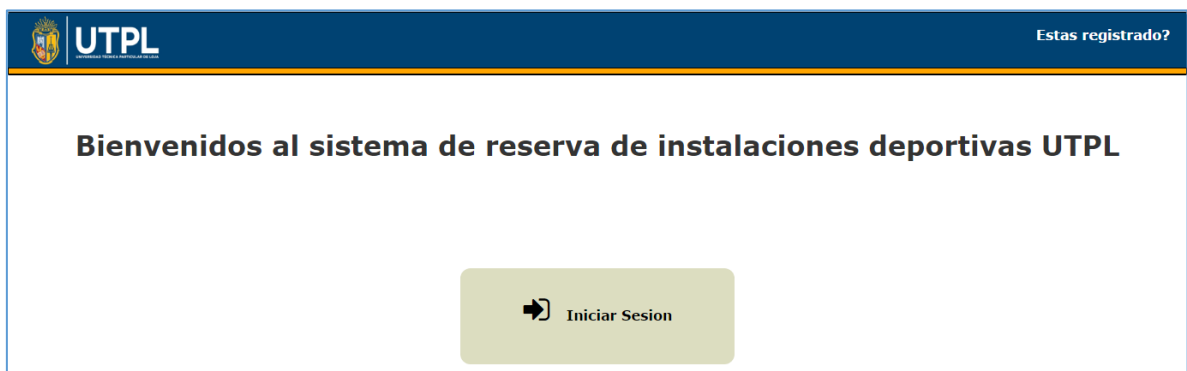
Cuando se inicia este servicio y nuestro servidor está listo para operar aparece como en la figura siguiente, el icono del servidor Tomcat con una pequeña flecha verde indicando que se está ejecutando y que esta iniciado.



Hecho todo esto damos en el botón de correr la aplicación desde el Run del Netbean



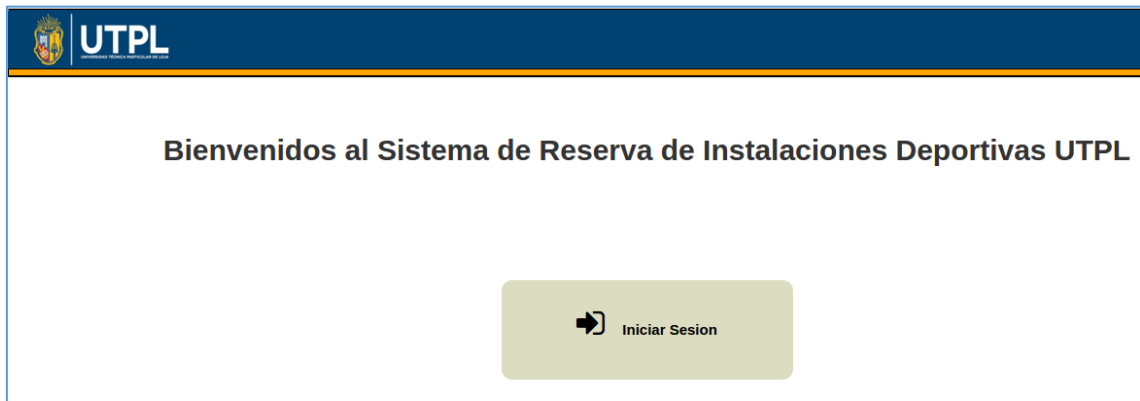
Se ejecuta y se compilan los archivos del código y se levanta la aplicación.



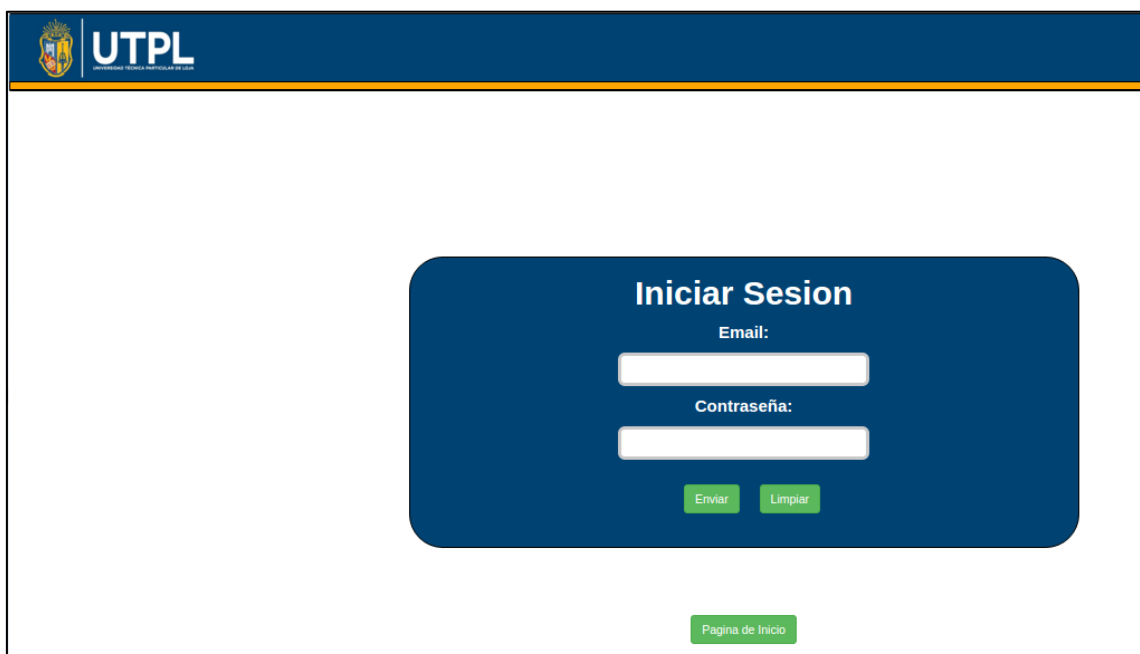
ANEXO 6:
MANUAL DE USUARIO

Manual de Usuario

Al acceder al sistema de reservas de las instalaciones de la UTPL se presenta la siguiente pantalla.



En esta parte solo ingresan usuarios que han sido previamente registrados, por el administrador del sistema. Una vez registrado será posible ingresar al sistema solamente con el email y la contraseña.





Página de inicio

En la página de inicio se tiene la facilidad de ingresar a las 7 opciones principales que son:

Lista de canchas: Para verificar las áreas o canchas que dispone la universidad

Comprobar disponibilidad: Permite visualizar la disponibilidad de las diferentes áreas que se quiere reservar.

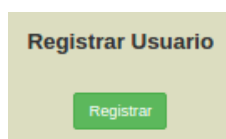
Reservar Cancha: Se encuentra el proceso de reserva de cancha, disponible solamente para el coordinador y administrador, los usuarios que no son administradores ni coordinadores solo pueden visualizar las reservaciones.

Normativa: en este elemento podemos visualizar la normativa de prestamo de las instalaciones.

Formularios: En esta parte del sistema podemos ingresar a los formularios digitales que se deben llenar para completar la reserva de las instalaciones.

Registrar Usuario: a esta parte solo puede acceder al administrador del sistema para poder registrar los usuario del sistema.

Reportes: Nos presentan tres tipos de reportes; Canchas mas reservadas, Pedodos realizados por usuario; Canchas reservadas por fecha.



En esta parte se registra a los usuarios del sistemas, a los administradores, coordinadores (pueden realizar reservas), estudiantes o usuarios normales (pueden visualizar las reservas).

Registro de Usuario

Nombre:	<input type="text"/>
Apellido:	<input type="text"/>
Email:	<input type="text" value="mail@example.com"/>
Telefono:	<input type="text"/>
Password:	<input type="password"/>
Permiso:	Seleccionar Permiso ▼

[Registrar](#)

[Pagina Principal](#)

Lista de Canchas

[Lista](#)

En esta parte se presenta la lista de las instalaciones con que cuenta la universidad.

Instalaciones Deportivas UTPL

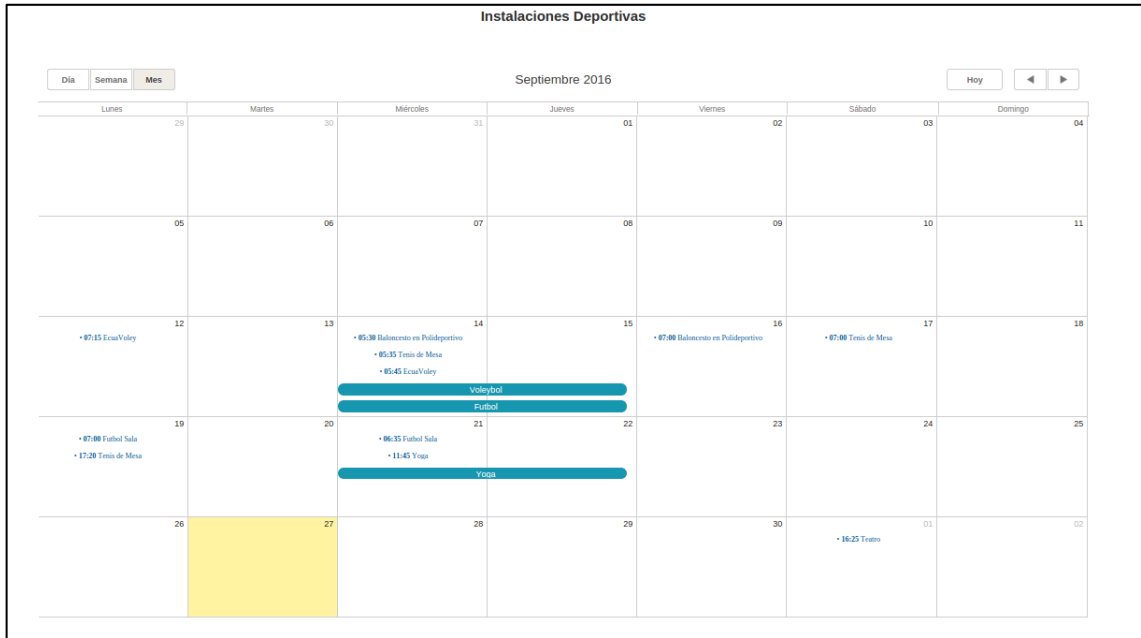
Nombre de la Cancha	Capacidad	Estado
Atletismo	1	S/I
Baile	10	S/I
Baloncesto en Cancha Exterior	6	S/I
Baloncesto en el Polideportivo	8	S/I
Ecuavoley	6	S/I
Futbol	18	S/I
Futbol Sala	10	S/I
Teatro	4	S/I
Voleibol	8	S/I
Tenis de Mesa	2	S/I
Yoga	4	S/I

[Volver](#)

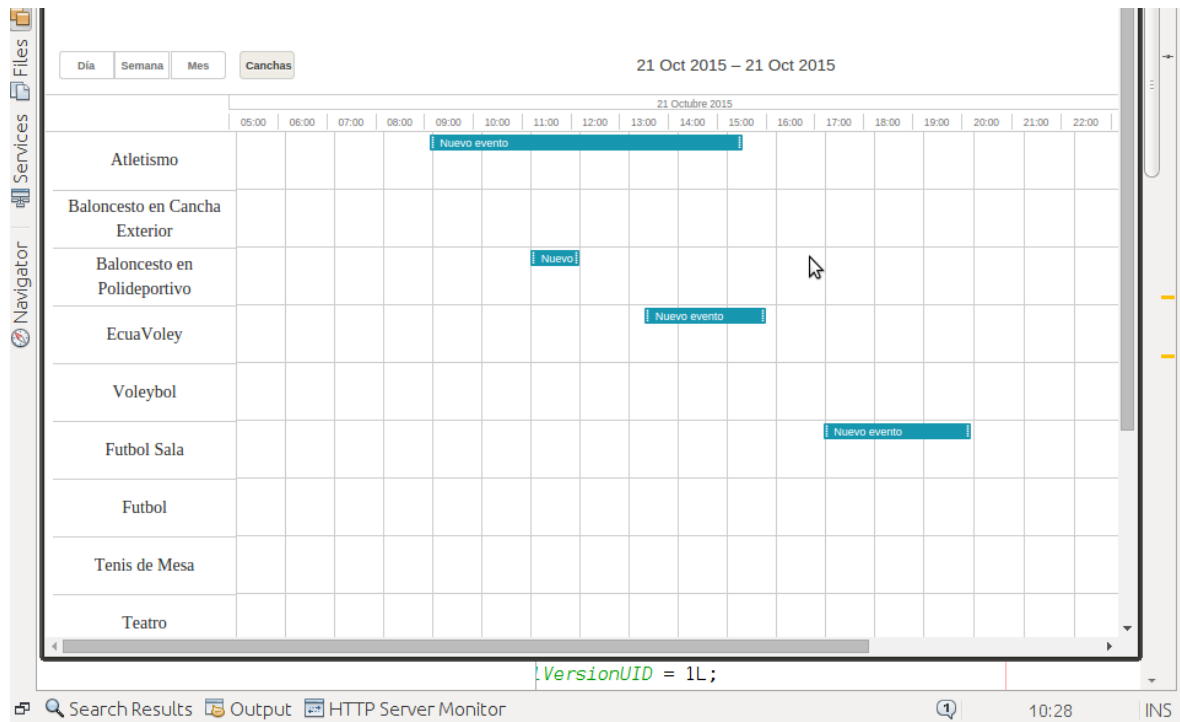
Comprobar Disponibilidad

[Comprobar](#)

Al ingresar a la opción verificar disponibilidad se puede observar las reservar para las diferentes canchas como sigue:



El sistema permite verificar la disponibilidad de las áreas por día, semana, mes o por área señalando en la parte superior izquierda, donde se podrá seleccionar como sigue:



Comprobación por fecha:

Comprobar Disponibilidad de Instalaciones Deportivas

Día Semana Mes Canchas

Octubre 2015

Lunes	Martes	Miércoles	Jueves	Viernes	Sábado	Domingo
28	29	30	01	02	03	04
05	06	07	08	09	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25

Reservar Cancha

Reservar

En esta parte se realiza la reserva de las instalaciones

Instalaciones Deportivas

Día Semana Mes Canchas

Septiembre 2016

Hoy ◀ ▶

Lunes	Martes	Miércoles	Jueves	Viernes	Sábado	Domingo
29	30	31	01	02	03	04
05	06	07	08	09	10	11
• 07:15 EcuaVoley		• 05:30 Baloncesto en Polideportivo • 05:35 Tenis de Mesa • 05:45 EcuaVoley		• 07:00 Baloncesto en Polideportivo	• 07:00 Tenis de Mesa	
		Voleybol Futbol				
• 07:00 Futbol Sala • 17:20 Tenis de Mesa		• 06:35 Futbol Sala • 11:45 Yoga				
		Yoga				
26	27	28	29	30	01	02
					• 16:25 Tenis	

Formularios

Formulario

Tenemos el área de formularios que nos sirve para llenar información de las personas que solicitan las instalaciones y si es el caso de los materiales que solicitan.

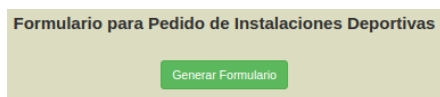
Dentro de los formularios tenemos las siguientes opciones:

- Formulario para pedido de las Instalaciones deportivas
- Formulario de préstamo de Material deportivo.
- Lista de pedido de Instalaciones deportivas.
- Lista de préstamo de Material deportivo.



Para registrar una nueva reserva el coordinador seleccionará reservar cancha y se deberá ingresar los siguientes datos.

La instalación se presenta de una lista previamente establecida como sigue:



Formulario para pedido de instalaciones deportivas

Fecha de solicitud: Instalacion solicitada:

Nombre persona responsable: Responsable:

Dependencia y/o titulacion: Observaciones:

Fecha inicio reserva: Hora inicio reserva:

Fecha fin reserva: Hora de devolucion:

Participantes

Estudiantes

Hombres: Mujeres:

Docentes

Hombres: Mujeres:

Administrativo

Hombres: Mujeres:

Prestamo de material deportivo

Nombre de la persona que entrega:

Articulo	Cantidad	Descripcion	Estado implemento
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text" value=""/>
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text" value=""/>
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text" value=""/>

Observaciones entrega de materiales

Lista de Pedido de Instalaciones Deportivas

Lista de Pedidos Instalaciones Deportivas UTPL

Fecha Solicitud	Instalacion	Persona Encargada	Encargado	Dependencia	Fecha Actividad	Horario	Observaciones	Fecha de Entrega
2016-03-23	Polideportivo-Voleibol	Fabio Paredes	Administrativo	Administrativo	2016-03-15	7:00 am a 07:00 pm	Ninguna	2016-03-22
2016-09-21	Polideportivo-Baloncesto	Fabio Paredes	Administrativo	Administrativa	2016-09-23	7:00 am a 9:00 am	asdsdasdsa	2016-09-23
2016-09-21	Polideportivo-Baloncesto	Fabio Paredes	Administrativo	Administrativa	2016-09-23	7:00 am a 9:00 am	asdsdasdsa	2016-09-23
2016-09-21	Polideportivo-Baloncesto	Fabio Paredes	Administrativo	Administrativa	2016-09-23	7:00 am a 9:00 am	asdsdasdsa	2016-09-23
2016-09-21	Polideportivo-Baloncesto	Fabio Paredes	Administrativo	Administrativa	2016-09-23	7:00 am a 9:00 am	asdsdasdsa	2016-09-23
2016-09-21	Canchas Externas-Baloncesto	asdasd	Administrativo	asdasd	2016-09-21	asdasdsad	asdasdasd	2016-08-30
2016-09-23	Voleibol	asdasdasd	Estudiante	asdasd	2016-09-14	asdasd	asdasd	2016-09-20
2016-09-21	Baile	asdasd	Estudiante	asdasdsa	2016-09-21	asdsad	sadasd	2016-09-21

Volver

Lista de Pedidos Instalaciones Deportivas UTPL

Persona que Entrega	Persona que Recibe	Fecha de Entrega	Hora de Entrega	Fecha de Devolucion	Hora de Devolucion
Fabio Paredes	Ana	2016-02-26	123456	2015-10-13	123456
Fabio	Jose	2016-03-09	7:00 am	2016-03-18	7:00 pm
Jose Perez	Pedro Perez	2016-09-22	07:00 am	2016-09-22	12:00 pm
Jose Perez	Pedro Perez	2016-09-22	07:00 am	2016-09-22	12:00 pm
Jose Perez	Pedro Perez	2016-09-22	07:00 am	2016-09-22	12:00 pm
asdas	sadasdsa	2016-09-06	qeqw	2016-09-13	wqeqwe
Jose	Pedro	2016-09-20	09:00 am	2016-09-22	12:00 pm
Pedro	Fabio	2016-09-22	07:00 am	2016-09-22	15:00 pm
asdsads	asdasdasd	2016-09-21	19:00 pm	2016-09-28	19:00 pm

Volver

Reportes

Reportes

En esta parte del sistema podemos obtener tres tipos de reportes para llevar datos estadísticos de las reservas.

Reportes

Canchas mas reservadas por mes IR	Canchas reservadas por fecha IR
Pedidos realizados por usuario IR	Hombres y mujeres que utilizan las canchas IR
Horarios mas solicitados de canchas IR	

Volver

Canchas mas Reservadas

IR

Reportes Canchas mas Reservadas por Mes

Seleccionar Mes:

Seleccionar Mes ▼

- Seleccionar Mes
- Enero
- Febrero
- Marzo
- Abril
- Mayo
- Junio
- Julio
- Agosto
- Septiembre
- Octubre
- Noviembre
- Diciembre

Canchas mas Reservadas por mes Mes 09

Nombre Cancha	Numero de veces Solicitada
Tenis de Mesa	3
Yoga	2
Baloncesto en Polideportivo	2
Ecuavoley	2
Futbol Sala	2
Voleybol	1
Futbol	1

Volver

Pedidos Realizados por Usuario

IR

Reportes

Pedido de canchas por Usuarios

Seleccionar Mes:

Seleccionar Mes ▼

- Seleccionar Mes
- Enero
- Febrero
- Marzo
- Abril
- Mayo
- Junio
- Julio
- Agosto
- Septiembre
- Octubre
- Noviembre
- Diciembre

Pedido de Cancha por Mes

Mes 09

Fecha Solicitud	Cancha	Encargado	Usuario que Realizo Solicitud
2016-09-21	Polideportivo-Baloncesto	Fabio Paredes	pedro perez
2016-09-21	Polideportivo-Baloncesto	Fabio Paredes	ana guerra
2016-09-21	Polideportivo-Baloncesto	Fabio Paredes	jose perez
2016-09-21	Polideportivo-Baloncesto	Fabio Paredes	pedro perez
2016-09-21	Canchas Externas-Baloncesto	asdasd	pedro perez
2016-09-23	Voleibol	asdasdasd	ana guerra
2016-09-21	Baile	asdasd	jose perez

[Volver](#)

Canchas Reservadas por Fecha

[IR](#)

Reportes

Canchas Reservadas Por Fechas

Desde:

hasta:

September 2016

Sun	Mon	Tue	Wed	Thu	Fri	Sat
28	29	30	31	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	1

Canchas Reservadas por Fecha

Desde 2016-09-01 Hasta 2016-09-30

Nombre de la Cancha	Inicio de la Reserva	Fin de la Reserva
Baloncesto en Polideportivo	2016-09-16 07:00:00.0	2016-09-16 18:25:00.0
Ecuavoley	2016-09-14 05:45:00.0	2016-09-14 23:35:00.0
Baloncesto en Polideportivo	2016-09-14 05:30:00.0	2016-09-14 20:25:00.0
Tenis de Mesa	2016-09-14 05:35:00.0	2016-09-14 14:10:00.0
Tenis de Mesa	2016-09-17 07:00:00.0	2016-09-17 07:05:00.0
Voleybol	2016-09-14 11:45:00.0	2016-09-15 02:30:00.0
Futbol	2016-09-14 21:35:00.0	2016-09-15 03:20:00.0
Ecuavoley	2016-09-12 07:15:00.0	2016-09-12 20:30:00.0
Futbol Sala	2016-09-19 07:00:00.0	2016-09-19 17:05:00.0
Yoga	2016-09-21 12:35:00.0	2016-09-22 04:10:00.0
Tenis de Mesa	2016-09-19 17:20:00.0	2016-09-19 21:55:00.0
Futbol Sala	2016-09-21 06:35:00.0	2016-09-21 16:40:00.0
Yoga	2016-09-21 11:45:00.0	2016-09-21 20:20:00.0

[Volver](#)