



FACULTAD DE INGENIERÍA Y CIENCIAS AGROPECUARIAS

IMPLEMENTACIÓN DEL MÓDULO DE SEGUIMIENTO DE TUTORÍA PARA  
TRABAJOS DE TITULACIÓN EN LA UDLA

Trabajo de Titulación presentado en conformidad con los requisitos establecidos  
para optar por el título de Ingenieros en Sistemas de Computación e Informática

Profesor Guía  
Juan José León

Autores  
Luis Fernando Sánchez Llerena  
Jurgen Gavino Zambrano Zambrano

Año  
2014

## DECLARACIÓN DEL PROFESOR GUÍA

“Declaro haber dirigido este trabajo a través de reuniones periódicas con los estudiante, orientando sus conocimientos y competencias para un adecuado desarrollo del tema escogido y dando cumplimiento a todas las disposiciones vigentes que regulan los Trabajos de Titulación”

.....  
Juan José León  
Ing. Msc. Sistemas  
1707506760

## DECLARACIÓN DE AUTORÍA DEL ESTUDIANTE

“Declaramos que este trabajo es original, de nuestra autoría, que se han citado las fuentes correspondientes y que en su ejecución se respetaron las disposiciones legales que protegen los derechos de autor vigentes”

.....  
Luis Fernando Sánchez Llerena  
1803780970

.....  
Jurgen Gavino Zambrano Zambrano  
1310740186

## AGRADECIMIENTO

Agradezco a todas las personas que me apoyaron durante este proceso de estudios; en especial a Dios, a mi familia, a mis profesores de carrera, a mi tutor Juan José León quien se dio tiempo para guiarnos durante el desarrollo de este trabajo, a mi compañero Luis Sánchez y a MANAFIAL.

Jurgen Zambrano

## AGRADECIMIENTO

Agradezco a todas aquellas personas que estuvieron a lo largo de este proceso de estudios; en especial a mi familia que siempre se encuentra apoyándome en cada paso que doy, a todos mis profesores de carrera que me supieron brindar su conocimiento, a mi tutor Juan José León quien con su conocimiento nos supo guiar durante el desarrollo de este proyecto de tesis, a mi compañero Jurgen Zambrano con quien tuve el honor de compartir la realización de este proyecto.

Luis Fernando Sánchez

## RESUMEN

Las tendencias tecnológicas actuales son de gran ayuda para el desarrollo de todo tipo de instituciones en todos sus ámbitos. Centrándonos netamente en la parte de educación es indudable la utilización de sistemas, ya existentes en el mercado, que cubren la parte administrativa y académica. Gracias a estos sistemas se puede garantizar que el estudiante, desde la comodidad de su casa, pueda matricularse e incluso cargar su horario

La Universidad de las Américas como una de las instituciones líderes en educación a nivel nacional y al tanto de las nuevas necesidades tecnológicas ha dispuesto el desarrollo de este sistema el cual sin duda alguna gestionará el registro de tutorías de proyectos de titulación de una manera digital, ágil y confiable.

Para la consecución de este sistema se ha utilizado una metodología de desarrollo ágil como SCRUM, un IDE de desarrollo completamente probado y destacado en el mercado como Visual Studio con características de integración a SQL SERVER 2008 R2 en su versión express la cual es gratuita, un patrón de diseño MVC que se ha convertido en un estándar aceptado por muchos en tiempo de desarrollo y diferentes tipos de diagramas UML como ayuda en el diseño del sistema.

Este sistema también tiene como propósito comenzar a automatizar completamente el proceso de desarrollo de trabajo de titulación en la Universidad de las Américas, fomentando de esta manera el desarrollo interno de aplicaciones y brindado la oportunidad a los estudiantes de contribuir con el desarrollo de la institución en el campo tecnológico.

## **ABSTRACT**

Current technology trends are helpful for the development of all types of institutions in all areas. Focusing purely on the part of education is undoubtedly the use of, existing systems on the market, covering administrative and academics. With these systems you can ensure that the student, from the comfort of your home, you can enroll and even upload your schedule

The University of the Americas as one of the leading education institutions nationwide and abreast of new technological needs has ordered the development of this system which undoubtedly manage the records of tutoring titling projects in a digital manner, fast and reliable.

To achieve this system has been used a agile development methodology such as SCRUM, an IDE development fully tested and outstanding in the market as Visual Studio integration features to SQL SERVER 2008 R2 version express which is free a MVC design pattern that has become a standard accepted by many in development time and different types of UML diagrams to aid in the system design.

This system also aims to start fully automate the process of developing work degree at the University of the Americas , thus enhancing the internal application development and provided an opportunity for students to contribute to the development of the institution the technological field.

## ÍNDICE

|  |    |
|--|----|
| Introducción .....   | 1  |
| 1. Metodologías De Desarrollo .....                                    | 5  |
| 1.1. Metodologías Tradicionales .....                                  | 5  |
| 1.1.1. Metodología RUP (Rational Unified Process) .....                | 6  |
| 1.1.2. Metodología MSF (Microsoft Solution Framework) .....            | 8  |
| 1.2. Metodologías Agiles.....  | 9  |
| 1.2.1 Extreme Programming (XP).....                                    | 11 |
| 1.2.2. Crystal Clear .....   | 11 |
| 1.2.3 Metodología Scrum.....   | 12 |
| 1.2.3.1. Roles .....   | 13 |
| 1.2.3.2. Reuniones .....   | 13 |
| 1.2.3.3. Sprint .....  | 13 |
| 1.2.3.4. Documentación En Scrum .....                                  | 14 |
| 1.2.3.5. Product Backlog .....   | 14 |
| 1.2.3.6. Sprint Backlog .....  | 14 |
| 1.3. Diagramas UML .....   | 16 |
| 1.3.1. Diagrama De Clases.....   | 16 |
| 1.3.2. Diagrama De Casos De Uso.....                                   | 20 |
| 1.3.3. Diagrama Entidad Relación .....                                 | 22 |
| 1.3.3.1. Por Cardinalidad (esta relación se da en ambos sentidos)..... | 23 |
| 1.3.3.2. Por modalidad obligatoria: .....                              | 24 |
| 1.3.3.3. Por modalidad optativa: .....                                 | 24 |
| 1.3.4 Diagrama de Actividades .....                                    | 24 |
| 1.3.4.1. Unión (Condición) y Ramificación.....                         | 26 |
| 1.3.4.2. Fork y Join .....   | 26 |
| 1.3.4.3. Eventos de tiempo.....  | 27 |
| 1.3.5. Diagrama de Despliegue .....                                    | 27 |
| 1.3.5.1. Nodo .....  | 28 |
| 1.3.5.2. Artefactos .....  | 28 |



|  |           |
|--|-----------|
| 1.3.5.3. Asociaciones .....                              | 28        |
| 1.3.6. Diagrama de Secuencia.....                        | 29        |
| 1.3.6.1. Línea de vida .....                             | 29        |
| 1.3.6.2. Mensajes .....                                  | 30        |
| 1.3.6.3. Ocurrencia en ejecución .....                   | 31        |
| 1.3.6.4. Mensaje Self.....                               | 31        |
| 1.3.6.5. Mensajes perdidos y encontrado .....            | 31        |
| 1.3.6.6. Inicio y final de línea de vida.....            | 32        |
| 1.3.6.7. Restricciones de duración y tiempo.....         | 32        |
| <b>2. TECNOLOGÍAS .....</b>                              | <b>34</b> |
| 2.1. NET Framework.....                                  | 34        |
| 2.2. Kit de Desarrollo de Software de Windows (SDK)..... | 35        |
| 2.3. IDE (Entorno de Desarrollo Integrado).....          | 35        |
| 2.4. MCV.....  | 36        |
| 2.5. Sistema Gestor de Base de Datos SGBD .....          | 38        |
| 2.5.1. Base de Datos Relacionales .....                  | 39        |
| 2.6. CSS .....   | 42        |
| 2.7. Reporteadores .....                                 | 43        |
| 2.7.1. Reporting Services.....                           | 43        |
| 2.8. Wireframe .....                                     | 43        |
| 2.9. Javascript .....                                    | 45        |
| <b>3. DESARROLLO DEL SISTEMA.....</b>                    | <b>46</b> |
| 3.1 Scrum .....  | 46        |
| 3.1.1 Datos Generales.....                               | 46        |
| 3.1.2 Product Backlog.....                               | 46        |
| 3.1.3 Iteraciones .....                                  | 48        |
| 3.1.4 Scrum Daily Meeting.....                           | 49        |
| 3.2. Análisis de Requerimientos.....                     | 50        |
| 3.2.1. Requerimientos no Funcionales.....                | 50        |
| 3.2.2. Requerimientos Funcionales. ....                  | 51        |

|   |    |
|---|----|
| 3.3. Diagramas de Casos de Uso. ....    | 53 |
| 3.4. Diagramas de Secuencia. ....       | 61 |
| 3.5. Diagramas de Actividades. ....     | 65 |
| 3.6. Diagrama de Clases. ....           | 72 |
| 3.7. Diagrama de Base de Datos. ....    | 73 |
| 4. Conclusiones y recomendaciones ..... | 74 |
| 4.1. Conclusiones .....                 | 74 |
| 4.2 Recomendaciones .....               | 75 |
| ANEXOS .....                            | 80 |

## **Introducción**

### **Antecedentes**

La Universidad de Las Américas ubicada en Av. de los Granados y Colimes esquina (Sede Norte), es una institución que inicio sus funciones hace aproximadamente 18 años, en este tiempo la Universidad se ha esforzado y dedicado a cumplir con los estándares más altos de educación tanto a nivel nacional como internacional, apegándose siempre a los reglamentos y normas de CES y el CEAACES.

La Universidad de Las Américas cuenta con un número considerable de alumnos en sus diferentes facultades como son: Facultad de Ingeniería y Ciencias Agropecuarias, Facultad de Ciencias Sociales, Facultad de Ciencias Económicas y Administrativas, Facultad de Comunicación, Facultad de Ciencias de la Salud, Facultad de Turismo y Hospitalidad, Facultad de Derecho, Facultad de Arquitectura. Y sus Escuelas: Escuela de Cine, Escuela de Música, Formación Integral.

Debido a la demanda de alumnos y el deseo de estar a la par de los desarrollos tecnológicos académicos actuales, la Universidad ha visto la necesidad de implementar un módulo de seguimiento de tutoría para los trabajos de titulación, que provea la información necesaria para tener el control y saber el verdadero estado de esta etapa del proceso de titulación, brinde la seguridad adecuada para que el proceso se realice de manera íntegra y reduzca costos operacionales.

### **Marco referencial**

En la actualidad la Universidad maneja en su totalidad el proceso de titulación mediante elementos físicos, como hojas, carpetas y a su vez completamente asistida por la persona encargada en secretaria académica quien constata las fechas de las reuniones, firmas de los involucrados, solicitudes presentadas, y que toda la documentación en conjunto este correcta para poder continuar con el proceso; y aún así en algunos casos

ocurren irregularidades en este proceso, lo que conlleva a un retraso en el desarrollo del proyecto de titulación debido a las diferentes investigaciones o comprobaciones que se deben llevar a cabo. Con la implementación de este módulo se brindará un medio para dar seguimiento de manera actualizada a cada proyecto, también se brindará el acceso en cualquier momento a la información por medio de la red, confianza de la información brindada, se podrá lograr un ahorro de tiempo considerable en el proceso de titulación, y también ayudará la conservación del medio ambiente mediante la reducción del gasto del papel.

### **Alcance**

El alcance de este proyecto de titulación es la implementación de un software que realice un seguimiento al Módulo de “Trabajo de Titulación” de los estudiantes de la Universidad de Las Américas, abarcando exclusivamente el seguimiento de las reuniones del profesor guía con el estudiante; conteniendo este proceso las reuniones respectivas, el registro de las actividades realizadas, visualización de las reuniones realizadas, generación de reportes del estado de los estudiantes y de los profesores dentro del proceso de desarrollo de Trabajo de Titulación, y finalmente la presentación del proyecto final dentro del plazo establecido.

Para la implementación de este software, se realizará un levantamiento de requerimientos con las personas involucradas (Secretaría Académica) analizando todas las funcionalidades que se desean cumplir con esta implementación, luego se comenzará con el diseño de la estructura de la implementación con un modelo de clases y de base de datos con herramientas como Visual Studio 2012 y SqlServer 2008, se procederá a la implementación del mismo con el desarrollo del código y finalmente se realizará pruebas ingresando datos y comprobando la funcionalidad.

Al terminar la implementación contendrá un método de acceso por intermedio de un login de usuario, para luego de esto poder registrar las reuniones, llenar

los formularios y/o solicitudes que el proceso amerita, llevar seguimiento tanto el alumno como el profesor de las tutorías.

### **Justificación**

Tomando en cuenta que la Universidad de Las Américas por su calidad académica se ha convertido en uno de los centros educativos de mayor afluencia de estudiantes de todo el país, tenemos como resultado el aumento de la demanda de procesos en el desarrollo del Proyecto de Titulación.

Por este motivo se ha visto necesario implementar este módulo del proceso de titulación, el que proporcionara una manera más ágil de realizar el seguimiento, dará una ayuda para controlar que se lleven a cabo las reuniones, emitirá reportes de la situación del proceso de titulación de estudiantes, se ahorrara gasto en equipos de oficina y finalmente se tendrá un control sobre este proceso para cada uno de los estudiantes y profesores involucrados.

### **Objetivo General**

Implementar un software que ayude al seguimiento de tutoría para Trabajos de Titulación en la UDLA, para agilizar el proceso y brindar mejor seguridad.

### **Objetivos específicos**

- Realizar un análisis y diseño del Sistema del Proceso de Titulación.
- Implementar reportes del seguimiento de tutorías.
- Implementar métodos o mecanismos para evitar cualquier tipo de registros de reuniones de tutorías falsos.

- Proporcionar a la Universidad información sobre el estado de este proceso dentro del proyecto de titulación.
- Brindar al estudiante una manera ágil de conocer su estado en este proceso.

### **Metodología a utilizar**

El proyecto utilizará un método experimental, debido a que este nos ayudará en la elaboración e implementación del software comenzando desde la captación de requerimientos e información, elaboración del diseño de implementación, utilizando herramientas tecnológicas adecuadas, siguiendo con la etapa de desarrollo del módulo en sí, mediante herramientas de desarrollo ajustadas a las necesidades y basándonos en los requisitos previamente obtenidos, para finalmente obtener los resultados deseados.

## 1. Metodologías De Desarrollo

En función de tener éxito en el desarrollo de software no basta con modelado y herramientas, también es necesaria la metodología de desarrollo.

Para tener una idea sobre el tipo de metodologías, se realiza la siguiente tabla comparativa:

Tabla 1. Tabla comparativa Metodologías Agiles vs Tradicionales.

| <b>METODOLOGÍAS AGILES</b>  | <b>METODOLOGÍAS TRADICIONALES</b>   |
|---|---|
| Abiertas a cambios durante el proyecto  | Mantienen resistencia al cambio en el ciclo del proyecto                    |
| Proceso menos controlado y más versátil en desarrollo                             | Procesos más controlados por políticas y normas                             |
| El cliente conforma el equipo de trabajo y/o desarrollo                           | El cliente intercambia ideas con el equipo de trabajo a través de reuniones |
| Se organiza grupos de menores de trabajo realizando actividades en un mismo lugar | Se tienen grupos mayores y trabajando de manera distribuida                 |
| Se procura llegar a la satisfacción del cliente por medio de la funcionalidad     | Se basa en los requerimientos del cliente.                                  |

### 1.1. Metodologías Tradicionales

Al principio el desarrollo del software era completamente artesanal y debido a la necesidad de mejorar este proceso, se tuvo que tomar metodologías de otras áreas y adaptarlas a este desarrollo, que fue dividido en etapas de manera secuencial, lo cual mejoró en algo el desarrollo del software.

Las principales metodologías tradicionales tienen RUP y MFS. Estas metodologías centran su atención en crear documentación, planificación y procesos del proyecto.

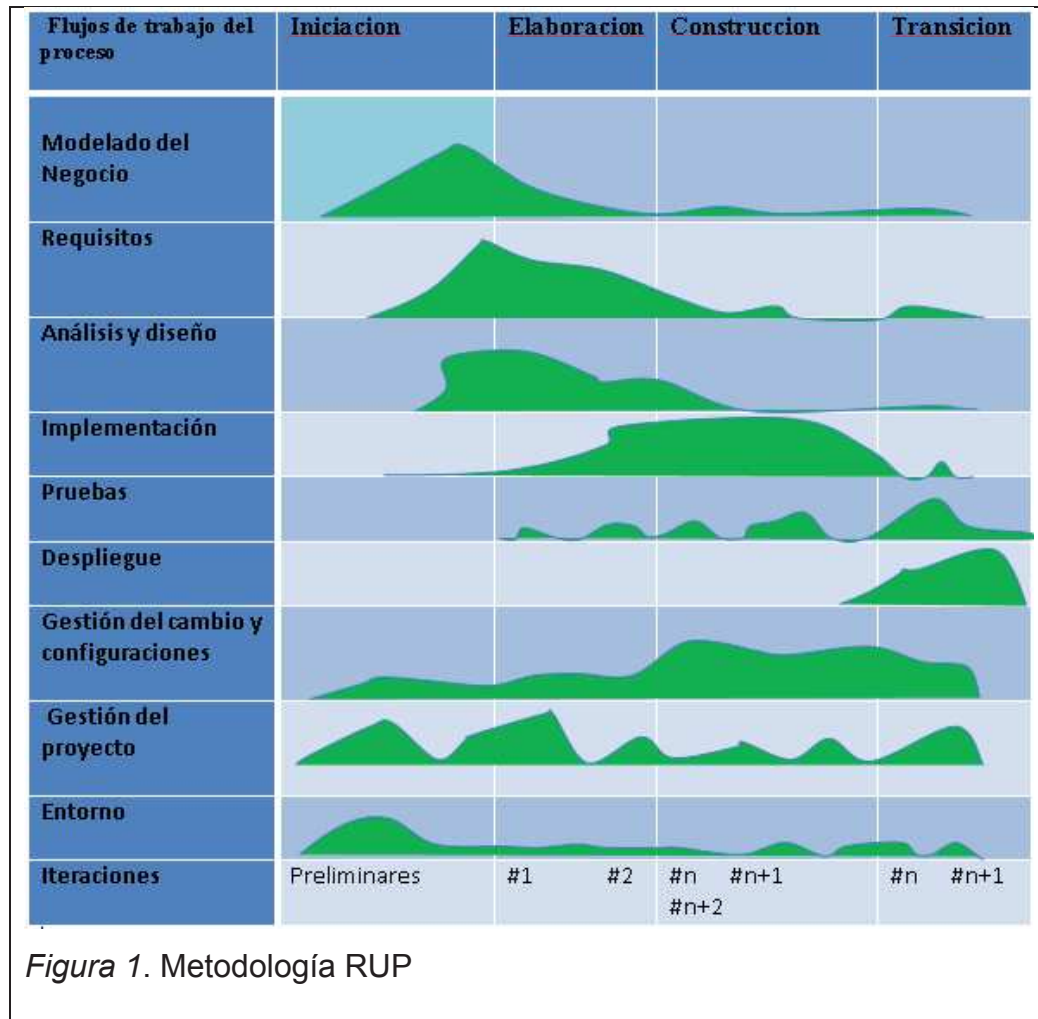
### **1.1.1. Metodología RUP (Rational Unified Process)**

Rup se encuentra enfocado en designar funciones y responsabilidades dentro de una estructura de desarrollo

Etapas

- Inicialización
- Elaboración
- Construcción
- Transición





### Ventajas

- Permite evaluar cada expresión, realizar cambios equánimes
- Es muy efectiva en planes de innovación.
- Esta metodología es sencilla, debido a que usa pasos sencillos (predecibles) al momento del desarrollo del software.
- Mantiene un seguimiento detallado de cada etapa.

### Desventajas

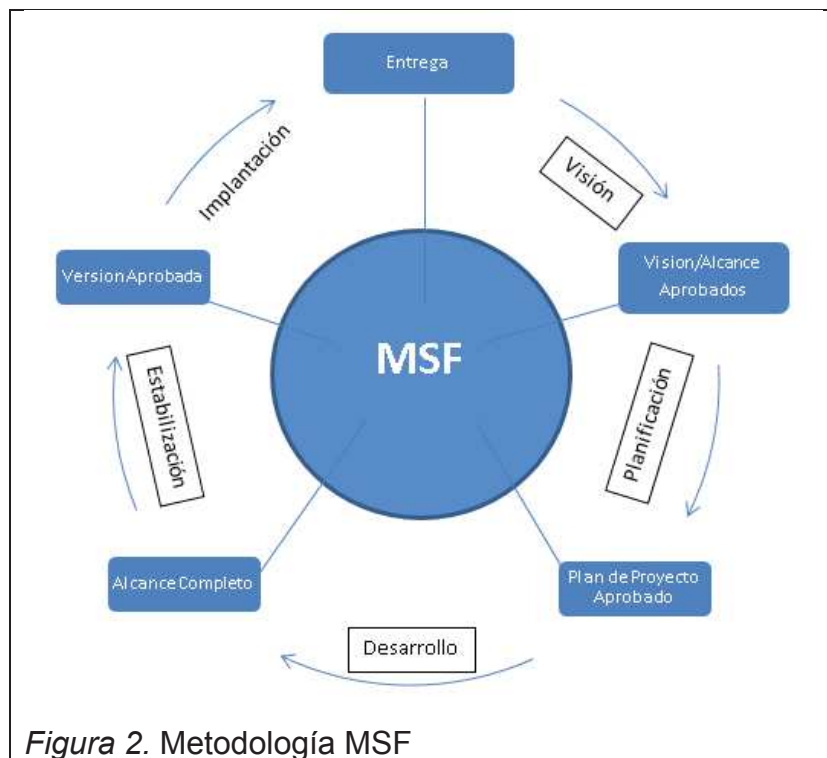
- La valoración de riesgos no es sencilla
- Demasiado flexible en algunos proyectos
- El cliente deberá tener la capacidad de detallar y comprender a un gran nivel el alcance del proyecto.

### 1.1.2. Metodología MSF (Microsoft Solution Framework)

Está diseñada en un conjunto de mejores prácticas en lo referente a administración de proyectos, es una metodología que gracias a varios modelos tiene la capacidad de adaptarse a cualquier tipo de tecnología.

Fases principales

- Visión y Alcances.
- Planificación.
- Desarrollo.
- Estabilización.
- Implantación.



## 1.2. Metodologías Ágiles

Este tipo de metodologías contribuyen a crear soluciones a medida para el entorno de desarrollo, simplificando las cosas, a pesar de esto no deja a un lado el asegurar la calidad del producto.

Las metodologías como tal valoran más a la persona, la participación del cliente y el desarrollo el software con repeticiones muy diminutas.

El punto de partida de las metodologías ágiles se puede apreciar con el manifiesto ágil, el que comienza enumerando las principales características de este desarrollo:

- A la persona y las repeticiones del grupo de trabajo sobre las tareas y herramientas.
- Crear un software que funcione más que crear un documento detallado.
- La cooperación entre cliente y desarrolladores es más primordial que el terminar el contrato.
- Dar prioridad a los nuevos cambios antes que proseguir exactamente un plan detallado.

Los valores anteriores ayudan a la redacción de los 12 principios del manifiesto ágil:

*I. La prioridad es satisfacer al cliente mediante tempranas y continuas entregas de software que le aporte un valor.*

*II. Dar la bienvenida a los cambios. Se capturan los cambios para que el cliente tenga una ventaja competitiva.*

*III. Entregar frecuentemente software que funcione desde un par de semanas a un par de meses, con el menor intervalo de tiempo posible entre entregas.*

*IV. La gente del negocio y los desarrolladores deben trabajar juntos a lo largo del proyecto.*

*V. Construir el proyecto en torno a individuos motivados. Darles el entorno y el apoyo que necesitan y confiar en ellos para conseguir finalizar el trabajo.*

*VI. El diálogo cara a cara es el método más eficiente y efectivo para comunicar información dentro de un equipo de desarrollo.*

*VII. El software que funciona es la medida principal de progreso.*

*VIII. Los procesos ágiles promueven un desarrollo sostenible. Los promotores, desarrolladores y usuarios deberían ser capaces de mantener una paz constante.*

*IX. La atención continua a la calidad técnica y al buen diseño mejora la agilidad.*

*X. La simplicidad es esencial.*

*XI. Las mejores arquitecturas, requisitos y diseños surgen de los equipos organizados por sí mismos.*

*XII. En intervalos regulares, el equipo reflexiona respecto a cómo llegar a ser más efectivo, y según esto ajusta su comportamiento.”*

¿Por qué usar metodologías ágiles?

Las metodologías están indicadas para proyectos con pocos requerimientos o cambiantes, además son muy útiles en equipos pequeños y dispuestos a resolver problemas concretos sin que esto sea una restricción para aplicarlas en desarrollo de grandes sistemas, también al dividir el trabajo en pequeños módulos disminuye la probabilidad de fallos y los costos bajan. Entre las principales ventajas están:

- Se puede cambiar los requisitos a lo largo del desarrollo del proyecto.
- Entrega de avances de software funcional en plazos breves y de manera continúa.
- Se trabaja de manera continua con el cliente y el equipo que desarrolla.

Algunas de las metodologías ágiles para desarrollo de software son:

### **1.2.1 Extreme Programming (XP)**

Fue la metodología pionera de este tipo y la fundadora para el resto de estas metodologías ágiles.

Esta metodología está centrada en asumir una elevada importancia en las relaciones ínter personales como clave para el éxito del proyecto, tomando muy en cuenta el trabajo en grupo, promoviendo el aprendizaje en el grupo de desarrolladores y un buen ambiente de trabajo. Se fundamenta con la retroalimentación constante con el cliente y el equipo de trabajo o desarrollo, simplicidad en lo implementado y disponibilidad absoluta a los cambios durante el desarrollo. Se puede definir como una metodología que se ajusta a desarrollos con requisitos muy cambiantes, con un alto riesgo. Las principales características son las siguientes:

Las historias de usuario: descripción breve por parte del cliente de las características del sistema.

- Roles XP: se detalla cada una de las funciones de las personas involucradas en el proyecto.
- Proceso XP: detalla el proceso de vida del proyecto en 6 etapas muy definidas: exploración, iteraciones, producción, mantenimiento, planificación de la entrega, muerte del proyecto.
- Prácticas XP: consiste en una serie de seguimiento a o reglas a fin de que el proceso evolutivo del proyecto funcione.

### **1.2.2. Crystal Clear**

Presenta un enfoque ágil tomando muy en cuenta la comunicación, y con mucha tolerancia hacia los entregables; maneja iteraciones cortas con retroalimentación constante entre usuarios y clientes lo que reduce la

posibilidad de productos intermedios. También define la necesidad de un usuario real para realizar las validaciones del desarrollo y requerimientos.

Los valores o propiedades de Crystal Clear son:

- Ambiente técnico con prueba automatizada
- Comunicación osmótica
- Foco
- Seguridad personal
- Mejora reflexiva
- Fácil acceso a usuarios expertos
- Entrega frecuente

### **1.2.3 Metodología Scrum**

Es la metodología ágil basada en la gestión de procesos iterativos e incrementales utilizada en ambientes de desarrollo de software.

Scrum es una de las metodologías ágiles más utilizada en la actualidad para el proceso de desarrollo de software, una de las características de esta metodología es la flexibilidad, calidad en cada entregable y satisfacción en el equipo de desarrollo. Se adapta fácilmente a los cambios que surgen en el proyecto además permite:

- Equipo multidisciplinario.
- Adaptabilidad al cambio.
- Liberación de entregables en cada iteración.

### **1.2.3.1. Roles**

Se deben identificar claramente las responsabilidades que intervienen en esta metodología para que el equipo de trabajo se pueda guiar en el desarrollo del producto.

Los roles principales que se manejan son:

- El Scrum Master es el encargado controlar que todos los procesos se realicen y ejecuten su trabajo a la par del ProductOwner.
- El productOwner es el que representa a los Stakeholders.
- El equipo de trabajo son los responsables de entregar el producto. El equipo tiene carácter multidisciplinario se encargan del Análisis, diseño, desarrollo, pruebas, documentación para el desarrollo.

### **1.2.3.2. Reuniones**

Diariamente se realiza una reunión de cada sprint sobre el estado en el que se encuentra el proyecto. Estas reuniones se las llama “daily standup” y no duran más de quince minutos y esta reunión debe generarse siempre en el mismo lugar y siempre a una hora determinada. Donde cada integrante da contestación a las siguientes preguntas:

- ¿Qué has empezado a hacer desde ayer?
- ¿Qué es lo que estás planeando realizar hoy?
- ¿Has tenido algún problema que te haya impedido alcanzar tu objetivo?

### **1.2.3.3. Sprint**

Dentro de esta metodología se definen reuniones entre un periodo de una a cuatro semanas estas son denominadas sprint para ajustar los tiempo de trabajo del equipo al final se debe mostrar un adelanto del producto para ser un entregable al cliente.

#### **1.2.3.4. Documentación En Scrum**

En esta metodología después de las reuniones se genera una lista de requisitos que se transforman en tareas y actividades para ser desarrolladas por los integrantes del grupo, estas listas son documentos dinámicos que pueden variar según las exigencias de cada uno de los clientes

#### **1.2.3.5. Product Backlog**

Son los requisitos o necesidades del sistema desde la perspectiva del cliente. Es la lista de usuario que contiene todas las tareas a ser resueltas, el responsable de esta lista se denomina Dueño del Producto.

#### **1.2.3.6. Sprint Backlog**

En este documento se detalla como el equipo de trabajo va a implementar los requisitos del sprint. Estas tareas son tomadas por cualquiera de los integrantes del equipo y no pueden durar más de 16 horas.



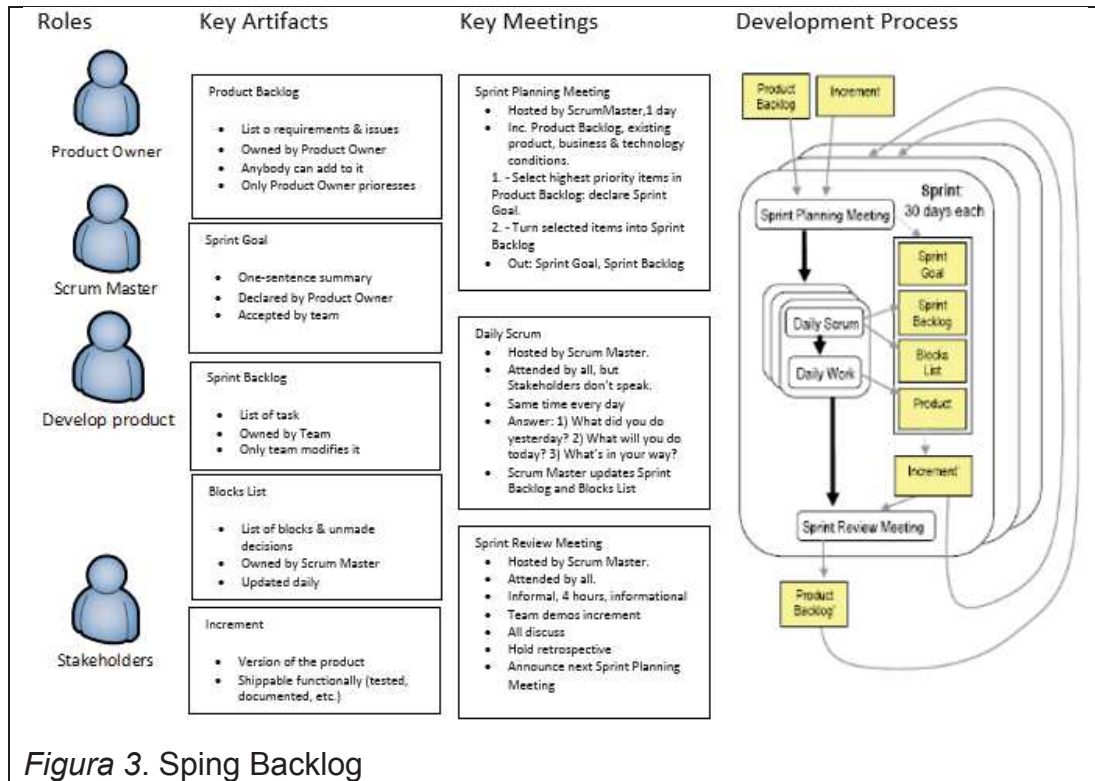


Figura 3. Sping Backlog

El ciclo de vida de Scrum se puede evidenciar en el siguiente gráfico:

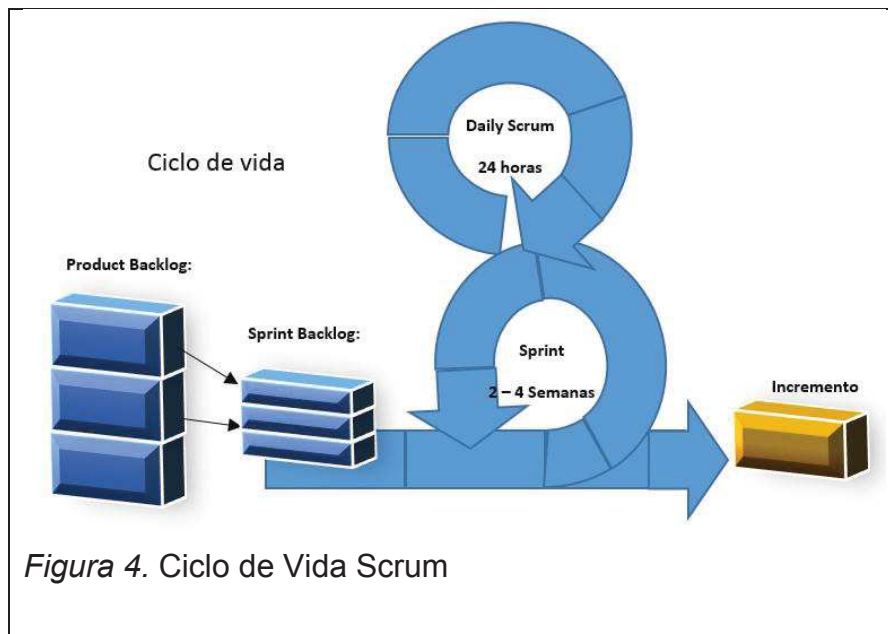


Figura 4. Ciclo de Vida Scrum

### 1.3. Diagramas UML

UML es el lenguaje de modelado más utilizado hoy en día, entre sus principales características están construir, visualizar, especificar, y documentar un sistemas que se encuentra en desarrollo. También utiliza su modelo para describir hipotéticamente lo que hará el sistema, pero no explica cómo implementarlo.

El UML cuenta con diversos esquemas con los cuales se indican los diferentes del proyecto.

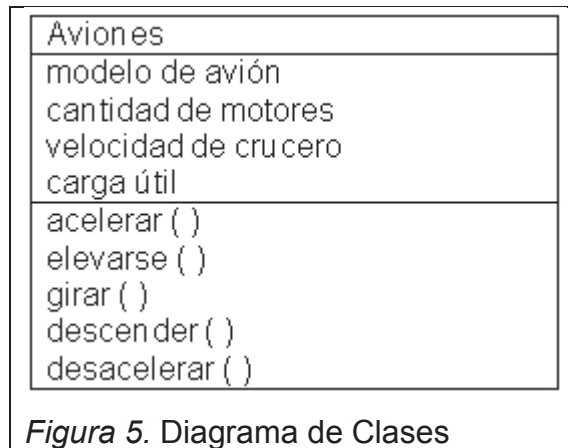
Los diagramas más importantes que se utilizaron para el desarrollo del proyecto serán los siguientes:

- **Diagrama De Clases**
- **Diagrama Entidad Relación**
- **Diagrama De Casos De Uso**
- **Diagrama de Secuencia**
- **Diagrama de Actividades**

#### 1.3.1. Diagrama De Clases

Describen la estructura estática del sistema, se representan a los objetos de un sistema. Un rectángulo es la manera de como representar a la clase y este se divide en tres partes y se asocian con otras clases mediante líneas que describen las uniones o la manera en que se vinculan entre sí.

El nombre de la clase se encuentra en la primera parte, los atributos de la misma en la segunda parte y las funciones o métodos de la clase en la tercera parte.

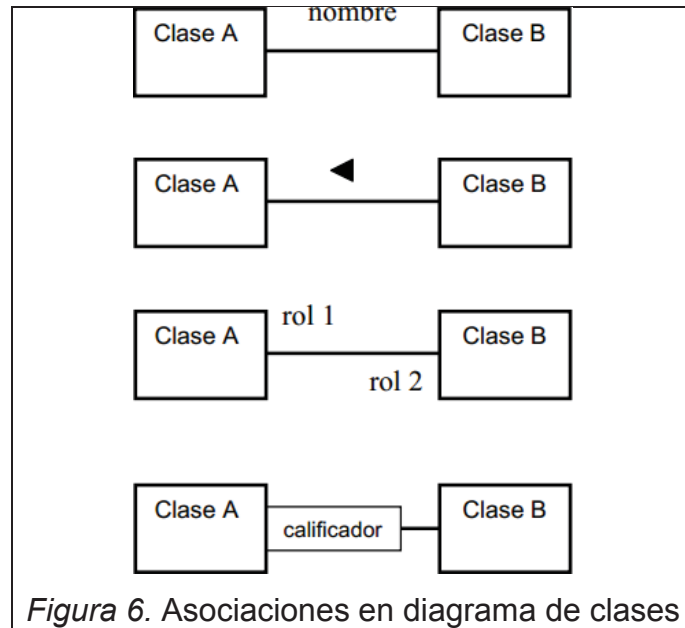


Los atributos son las características de la clase Avión en este caso, los atributos pueden ser públicos, protegidos y privados, los métodos o funciones también pueden ser públicos o privados

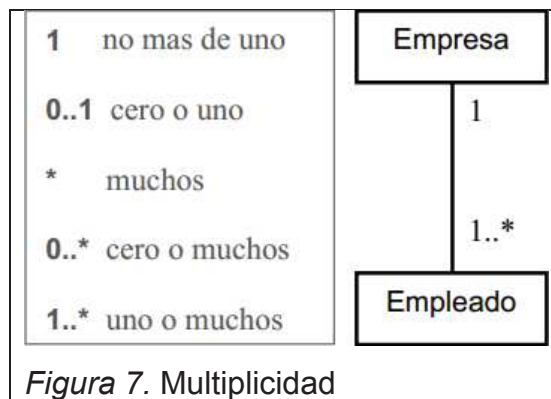
Asociaciones, representan las relaciones estáticas entre clases. El nombre va por sobre o por debajo de la línea que la simboliza.

Una flecha con relleno indica la dirección de la relación.

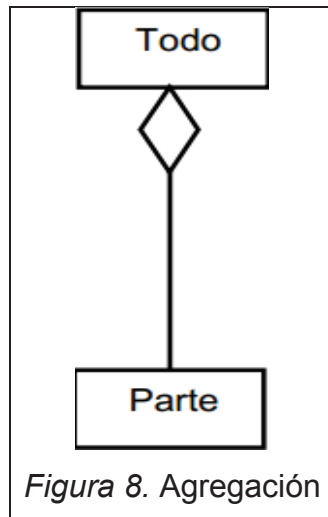
Los roles se colocan cerca del final de la asociación, los roles simbolizan la manera en que dos clases se ven entre sí. Cuando una asociación es calificada, el símbolo correspondiente se lo ubica al final de la asociación, contra la clase que hace de calificador.



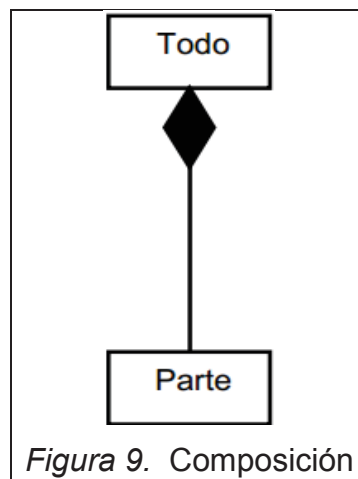
Multiplicidad, los signos utilizados para señalar la multiplicidad se los coloca cerca del final de una asociación. Estos signos o símbolos indican el número de instancias de una clase vinculada a una de las instancias de la otra clase.



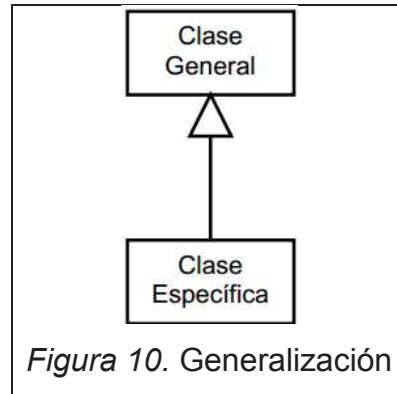
Agregación, es un nexo en el que la clase ejemplo **Todo** juega un rol más importante que la clase ejemplo **Parte**, pero ninguna de las dos clases son dependientes entre sí. Se diagrama con un rombo vacío contra la Clase **Todo** en el ejemplo.



Composición, es un tipo especial de agregación que indica una fuerte posesión de la Clase ejemplo **Todo**, a la Clase ejemplo **Parte**. Se la diagrama con un rombo relleno contra la clase que representa el todo.



Generalización, es otro tipo de nombre para la herencia. Es una relación entre dos clases en donde una Clase ejemplo **Específica** es una versión especializada de otra Clase ejemplo **General**.

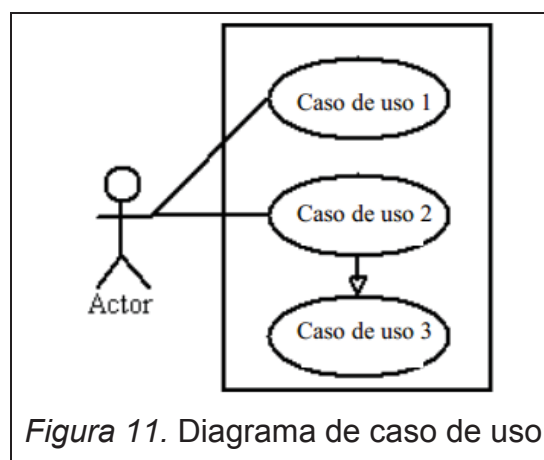


### 1.3.2. Diagrama De Casos De Uso

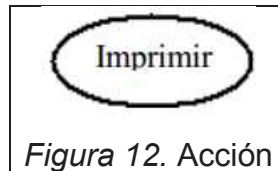
Se utilizan para describir todas las acciones que va a realizar el sistema desde la visión del usuario. Su valor está en adquirir los requerimientos del sistema desde la visión del usuario. Son las funciones o acciones del sistema para con los usuarios.

Para representar los casos de uso se utilizan los siguientes elementos:

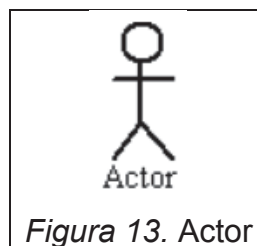
Sistema, el rectángulo simboliza el contorno del sistema que contiene los casos de uso. Los actores se los coloca fuera del contorno del sistema.



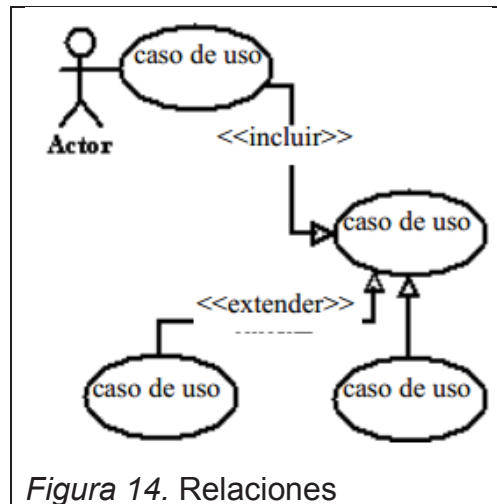
Casos de Uso, es alguna acción que puede realizar el sistema bajo la orden de un actor o de otro caso de uso, estos se los representa con óvalos. La etiqueta dentro del óvalo indica la función que tiene en el sistema.



Actores, son agentes exteriores al sistema que dado un rol van a interactuar con este, los actores son los usuarios o clientes de un sistema. Se representa mediante el dibujo de una persona.



Relaciones, la o las relaciones que existen entre el actor y el caso de uso, se la diagrama con una línea. Para las relaciones entre casos de uso, se utilizan flechas con etiquetas como **incluir** o **extender**. Una relación o nexo **incluir** señala que un caso de uso necesita de otro para cumplir una tarea. Una relación o nexo **extender** señala otras más opciones para un cierto caso de uso.



### 1.3.3. Diagrama Entidad Relación

Consiste en una apreciación del mundo real, a la vez se basa en una agrupación de objetos básicos llamados entidades y relaciones.

Entidad, definido como cualquier elemento abstracto o real, que existente en un ambiente determinado ó a su vez puede llegar a existir y del cual deseamos guardar su información.

Atributos, son propiedades o características propias de las entidades que toma un valor en una instancia de la entidad.

Clave Primaria, permiten identificar en forma única a una tabla, se lo utiliza cuando es necesario realizar una referencia de un registro específico de una entidad a otra entidad.

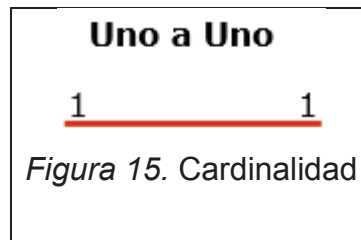
Clave foránea, es clave primaria en otra entidad con la cual se relaciona hacia otra tabla.

Relación, es la manera en la que interactúan dos o más entidades entre sí, se representan mediante una flecha. Se dividen en los siguientes tipos:

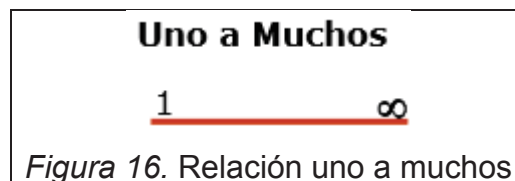


### 1.3.3.1. Por Cardinalidad (esta relación se da en ambos sentidos)

Relación Uno a Uno: Existe este tipo de relación cuando dos entidades solo se pueden relacionar mediante un registro. Este tipo de relación tiene la clave foránea en una de las dos entidades, se representa de la siguiente manera:

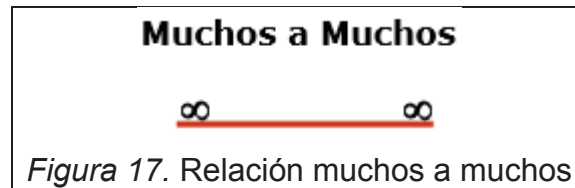


Relación Uno a Muchos: Existe este tipo de relación cuando un registro de la tabla principal se puede relacionar con más de un registro de la tabla secundaria y a su vez un registro de la tabla secundaria puede estar únicamente relacionado a un registro de la tabla principal. En este tipo de casos la clave foránea se la debe ubicar en la tabla secundaria, se representa de la siguiente manera:



Relación Muchos a Muchos: Existe este tipo de relación cuando múltiples registros de una tabla pueden estar relacionados con múltiples registros de la otra tabla. En este tipo de caso las dos tablas no pueden relacionarse directamente entre sí, se debe crear o añadir otra tabla entre ellas (Tabla de vinculación).

La clave primaria de esta nueva tabla se forma por la unión de las dos claves foráneas de las tablas que la conforman, se representa de la siguiente manera:



### 1.3.3.2. Por modalidad obligatoria:

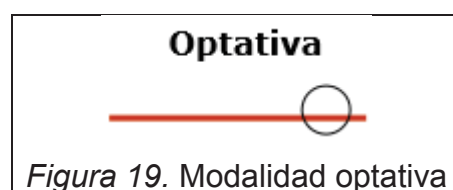
Partiendo de dos tablas relacionadas;

Cuando cada registro de la tabla principal debe al menos existir siempre un registro de la tabla secundaria asociada, se representa de la siguiente manera



### 1.3.3.3. Por modalidad optativa:

Cuando para todo registro de la tabla principal puede existir o no al menos un registro de la tabla secundaria asociado, se representa de la siguiente manera

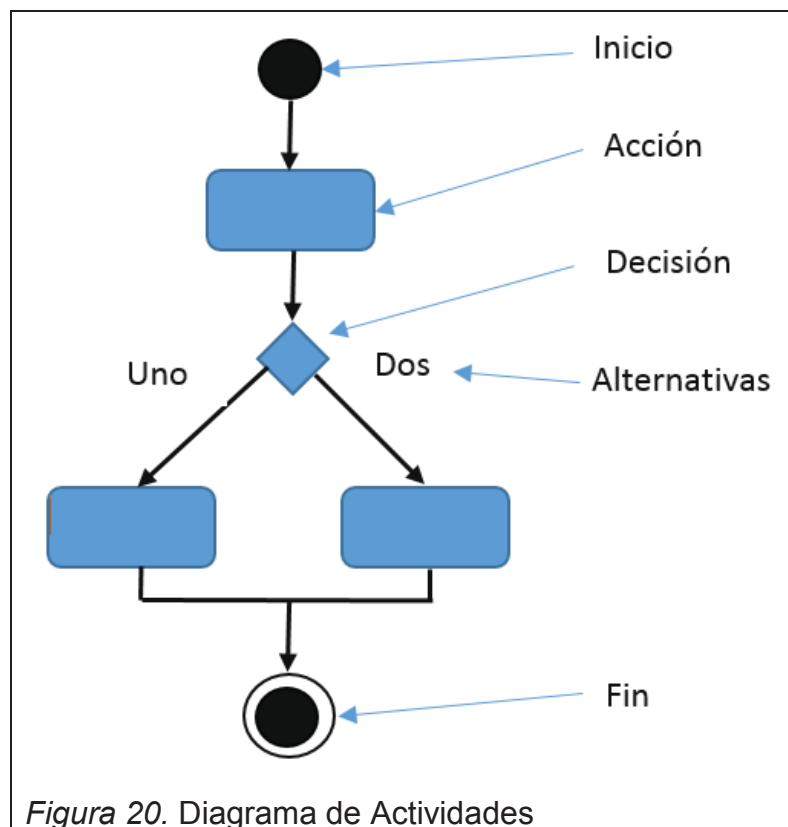


### 1.3.4 Diagrama de Actividades

El diagrama de actividades permite describir como un sistema implementa su funcionalidad, nos permite modelar el comportamiento dinámico de un

procedimiento, transacción o caso de uso el cual hace énfasis en el proceso que se encuentra llevándose a cabo en dicho momento.

Este tipo de diagramas es el que mejor se llega a comprender ya que son herederos directos de los diagramas de flujo, también heredan algunas características de los diagramas de estado, también de los diagramas de flujo de datos.



El diagrama de actividades tiene:

- Inicio.- Único
- Acciones.- Es un proceso sistemático.
- Decisiones.- Elección de una salida.
- Actividades.- Es un conjunto de acciones que modelan un proceso.
- Fin.- Uno o varios.

Ejemplo: Colocar usuario y password son actividades, el proceso es ingreso a un sistema.

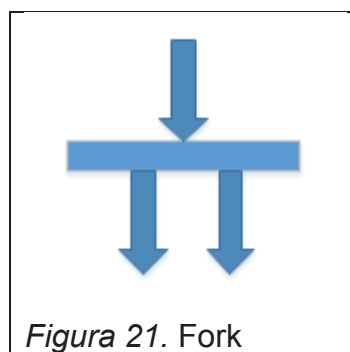
### 1.3.4.1. Unión (Condición) y Ramificación

Las decisiones que se presentan en este tipo de diagramas son las alternativas de flujo que va a tomar un diagrama en función de una condición, esta condición puede tomar uno o más posibles ramas que llegan a un punto de convergencia.

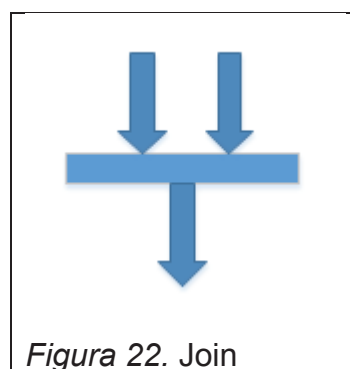
### 1.3.4.2. Fork y Join

Los forks y joins son usados para describir concurrencia entre acciones o actividades.

**Fork:** Indica que un conjunto de actividades puede realizarse en paralelo.



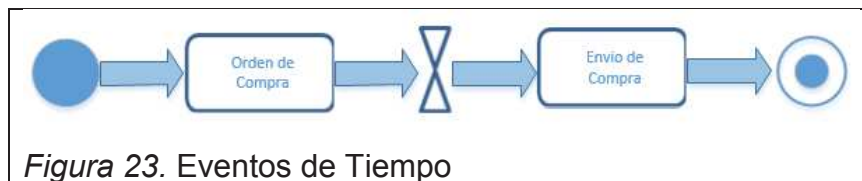
**Join:** Indica que para que continúe, deben haber finalizado todas las actividades concurrentes.



### 1.3.4.3. Eventos de tiempo.

Los eventos de tiempo pueden ser actividades temporizadas, retrasos, timeouts.

Un evento de tiempo puede ser el inicio de una actividad, así como también en un flujo de entrada la actividad temporizada representa una única acción tras la llegada del flujo.

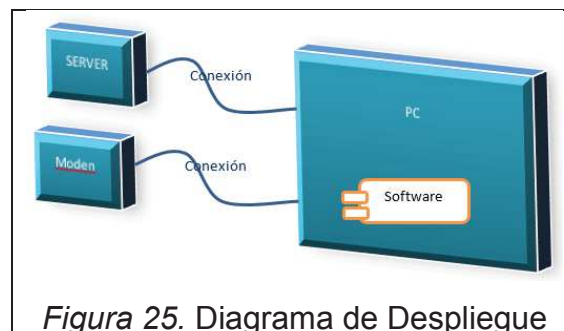


En un evento de tiempo sin flujo de entrada puede representar una activación repetitiva en el tiempo.



### 1.3.5. Diagrama de Despliegue

El diagrama de despliegue modela la arquitectura que existe entre el hardware y software del sistema.



### 1.3.5.1. Nodo

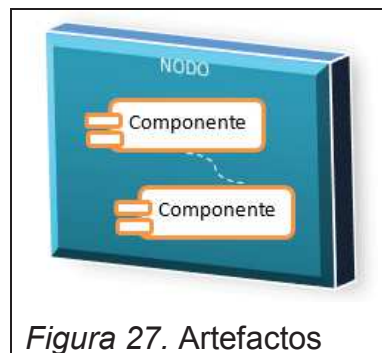
Es un elemento de hardware o software que representa un elemento, el nodo se lo representa mediante una caja en tres dimensiones.



*Figura 26. Nodo*

### 1.3.5.2. Artefactos

El artefacto es un producto del proceso de desarrollo, este puede incluir los modelos del proceso como: modelo de casos de uso, modelos de diseño, etc)



*Figura 27. Artefactos*

### 1.3.5.3. Asociaciones

La asociación representa la ruta de interacción entre nodos, mediante un tipo de protocolo de comunicación.



*Figura 28. Asociaciones*

### 1.3.6. Diagrama de Secuencia

Este diagrama muestra la manera cómo interactúan los objetos a través de líneas de vida a lo largo de un proceso, con sus respectivas iteraciones figuradas a manera de mensajes graficados con flechas desde la línea de vida de origen hacia el destino respectivamente. Nos muestran cuales objetos se comunican con otros. Los elementos de estos diagramas son:

#### 1.3.6.1. Línea de vida

Representa a un participante en particular, usualmente contiene un rectángulo que con el nombre del objeto.

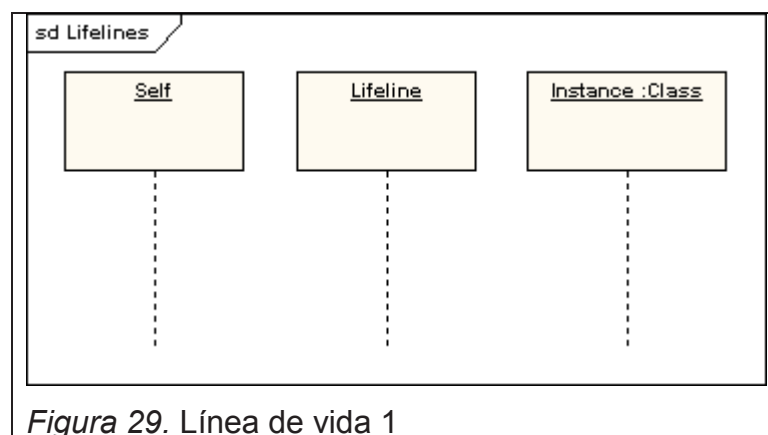
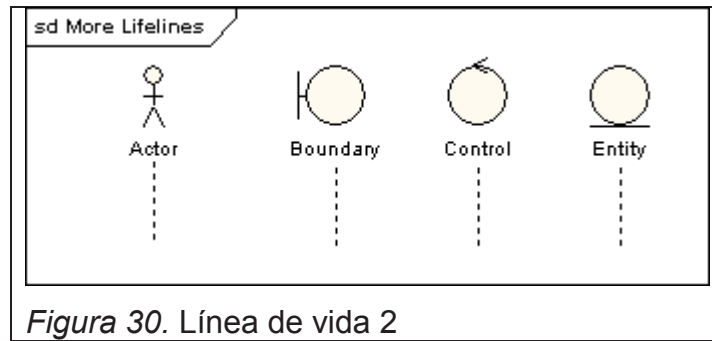


Figura 29. Línea de vida 1

En algunos casos en el diagrama de secuencia también se representa la línea de vida con el símbolo correspondiente al elemento actor que representa. Esto se da cuando el diagrama de secuencia se deriva de algún caso de uso, existen componentes como entidad, límite y control también pueden contener las líneas de vida.



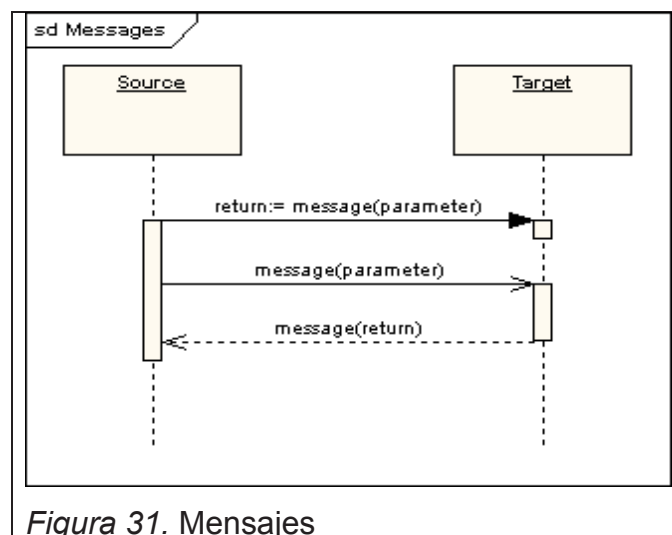
### 1.3.6.2. Mensajes

Se representan o diagraman como flechas, pueden ser completos, encontrados o perdidos; sincrónicos o asincrónicos: llamadas. En la siguiente figura se observa:

**Primero.-** Es un mensaje sincrónico (flecha con punta oscura), que tiene un mensaje de retorno.

**Segundo.-** Es un mensaje asincrónico (flecha con punta en línea),

**Tercero.-** Es un mensaje de retorno asincrónico (flecha con línea punteada)





### 1.3.6.3. Ocurrencia en ejecución

Se lo representan con un rectángulo delgado en la línea de vida, muestra la ocurrencia de ejecución o activación de un foco de control entre elementos, en la figura 31 se puede observar tres ocurrencias.

### 1.3.6.4. Mensaje Self

Representa una llamada recursiva de una operación o un método dentro del mismo objeto.

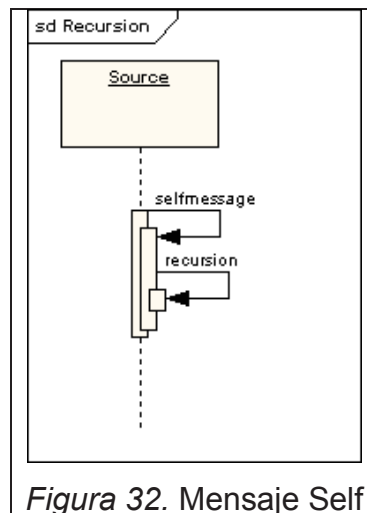


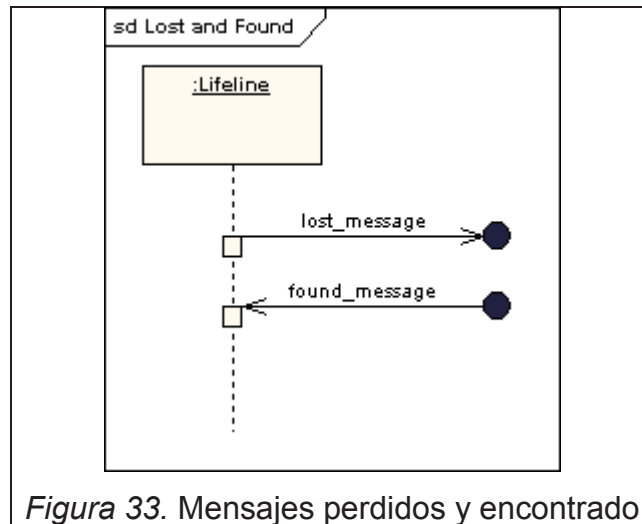
Figura 32. Mensaje Self

### 1.3.6.5. Mensajes perdidos y encontrado

**Mensajes Perdidos:** Son aquellos mensajes que fueron enviados pero no llegaron a su destino o que llegaron otro destino que no se muestra en el diagrama actual.

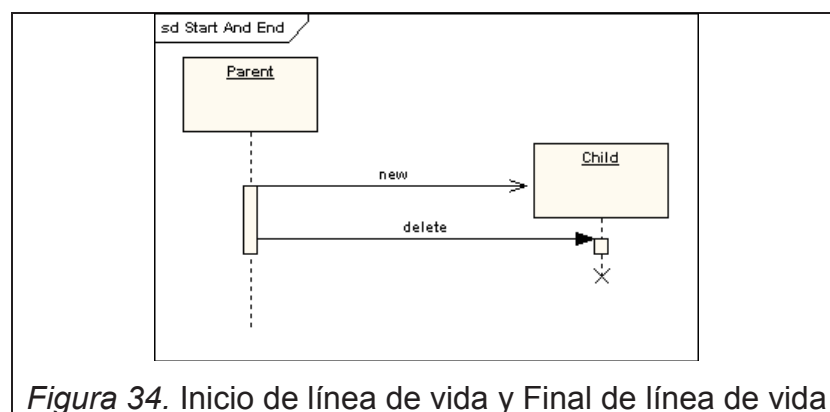
**Mensajes encontrados:** Son aquellos mensajes que llegan desde un remitente desconocido o que no consta en el diagrama actual.

Se representa yendo o llegando desde un elemento de punto final.



### 1.3.6.6. Inicio y final de línea de vida

Una línea de vida puede ser creada o destruida durante el proceso. En caso de destruir una línea de vida se lo termina mediante una cruz. En caso de crear una línea de vida se coloca a un nivel más bajo de la página que el símbolo del objeto que causo la creación.



### 1.3.6.7. Restricciones de duración y tiempo.

De manera original un mensaje se muestra con una línea horizontal; la línea de vida simboliza el paso de tiempo hacia abajo. Al representar una restricción para el mensaje este se lo muestra con una línea inclinada.

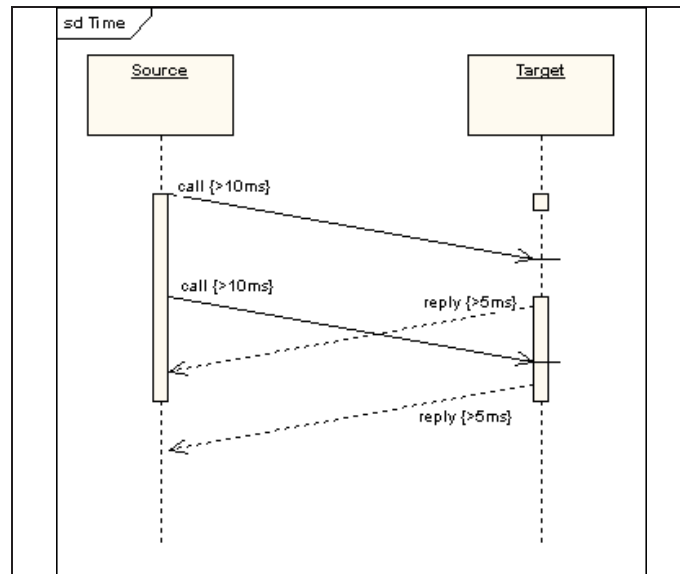


Figura 35. Restricciones de tiempo y duración

## 2. TECNOLOGÍAS

### 2.1. NET Framework

Un 'framework' es una ayuda o soporte, por medio de la cual se ayudan otras aplicaciones para ser desarrolladas; cuenta con programas y bibliotecas que ayudan a optimizar y mejorar el desarrollo.

.NET Framework es una plataforma de desarrollo para compilar aplicaciones de Windows, Windows Phone, Windows Server y Windows Azure. Se encuentra formado por la biblioteca de clases de .NET Framework las cuales incluyen clases e interfaces que son compatibles con la mayoría de tecnologías y Common Language Runtime (CLR).

- **Common Language Runtime (Entorno de Ejecución para Lenguajes):** Es la parte fundamental de .NetFramework, se lo puede describir como el administrador de código en tiempo de ejecución, y proporciona servicios fundamentales entre los cuales se encuentran administración de memoria, control de excepciones, administración de subprocesos, control de hilos. El proceso se basa en que CLR compila el código fuente de cualquier lenguaje soportados por .NET, lo convierte en MSIL (Lenguaje Intermedio de Microsoft) y finalmente antes de poder ejecutar el código se debe convertir el MSIL al código propio de la CPU esto se lo hace a través de un JIT (Just-In-time) que genera el código de maquina real que se ejecuta en la plataforma cliente. De esta manera se obtiene la ejecución final del código.
- **Biblioteca de clases de .NET Framework:** Se considera como una colección de tipos reutilizables, relacionadas directamente con el CLR; al estar orientada a objetos proporciona tipos que pueden derivar funciones reutilizables, con esto también se logra que los tipos puedan acoplarse fácilmente a nuevas características de .NetFramework. Con las clases orientada a objetos de esta biblioteca los tipos de .NetFramework permiten realizar operaciones comunes de programación como por

ejemplo administración de cadenas, recolección de datos, conectividad de base de datos y acceso a archivos.

Los siguientes servicios ofrece .Net Framework a las aplicaciones en ejecución:

- Administración de la memoria
- Sistema de tipos comunes
- Biblioteca de clases extensa
- Frameworks y tecnologías de desarrollo
- Interoperabilidad de lenguajes
- Compatibilidad de versiones
- Ejecución en paralelo
- Compatibilidad con múltiples versiones.

Esté ya se encuentra incluido en equipos con los sistemas operativos: Windows Vista, 7, 8, Server 2003, Server 2008 (También en R2) y Windows Server 2012(También en R2), en todos los sistemas operativos mencionados se puede instalar también manualmente sus respectivas versiones actualizadas desde el sitio oficial de Microsoft; y para Windows XP se debe instalar manualmente cualquiera de sus versiones disponibles.

## **2.2. Kit de Desarrollo de Software de Windows (SDK)**

El SDK es un conjunto de herramientas, ejemplos de código, documentación, compiladores, encabezados y bibliotecas que utilizan los desarrolladores para escribir, generar, probar e implementar aplicaciones en .Net Framework. Ofrece todas estas utilidades para Windows XP, Server 2003, Vista, Server 2008, 7 y Windows 8.

## **2.3. IDE (Entorno de Desarrollo Integrado)**

IDE es un entorno de programación que consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica (GUI).

**Visual Studio 2012** es un IDE que permite desarrollar proyectos web, de escritorio, entre otros; soporta varios lenguajes de programación, ofrece herramientas de desarrollo avanzadas, funciones de debuggin, funciones de bases de datos que permiten crear aplicaciones para distintas plataformas de manera rápida.

Ofrece herramientas mejoradas de utilización de datos como LINQ que facilita el armado de soluciones partiendo de analizar información y actuar en secuencia; a su vez también ofrece herramientas que facilitan la creación de aplicaciones integradas con varias plataformas, Office 2012, SQL Server 2008. Para la Web ASP.NET y AJAX y otras nuevas tecnologías que permiten a los desarrolladores crear aplicaciones verdaderamente eficientes e interactivas.

Los requisitos mínimos que debe tener un sistema para el correcto funcionamiento de este IDE son:

- Sistema Operativo: Windows 7 SP1(X86 y X64), Windows 8 (X86 y X64), Windows Server 2008 R2 SP1 (X64), WINDOWS SERVER 2012(X64).
- 1 GB de RAM (1,5 GB si se ejecuta en máquina virtual).
- 10 GB de espacio disponible en disco duro.
- Procesador a 1,6 Ghz o más rápido.
- Tarjeta de video compatible con DirectX 9 con una resolución de pantalla de 1024 x 768 o superior.

#### **2.4. MCV**

Es un patrón de diseño de arquitectura de software basado en tres capas; basado en la separación de los datos por un lado, la interfaz del usuario por otro y la lógica. Este patrón de diseño apareció por primera vez a fines de los 70's por parte de Trygve Reenskaug trabajando en Smalltalk-76 para la compañía Xerox Parc. Su uso radica sobre aplicaciones en su mayoría web, en donde su estructura quedaría formada por la vista como la página html, el modelo el Sistema de Gestión de Base de Datos (SGBD) y la lógica interna, y el controlador aquel que recibe los eventos y los resuelve.

En este patrón se encuentran inmersos los siguientes actores:

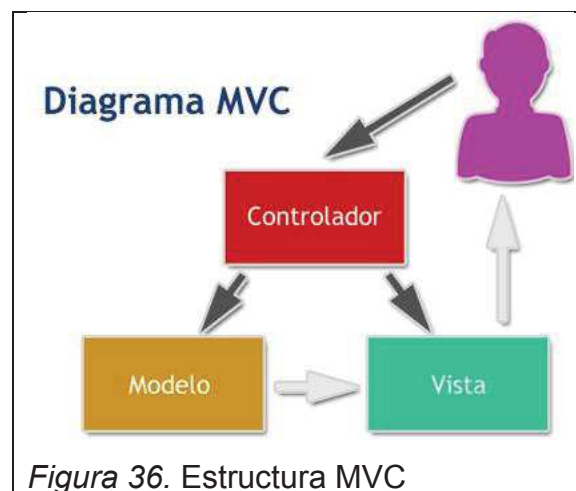
**El usuario**, es quien interactúa con el sistema, es decir genera una petición al controlador manipulando el modelo y recibe una vista como respuesta a esta petición.

**El modelo**, es donde se trabaja con los datos del sistema, este reacciona ante un pedido del controlador gestionando los datos y generando los datos que van a ser mostrados al usuario a través de la vista.

**La vista**, es la presentación gráfica del sistema al usuario, se muestran de manera entendible al usuario los datos solicitados por este a través del controlador y generados por la base de datos.

**El controlador**, administra los eventos generados por el usuario y actualiza el modelo; es el enlace entre las vistas y el modelo resolviendo las peticiones del usuario en el modelo para luego estas ser mostradas en la vista.

A continuación se muestra un diagrama del patrón MVC



La interpretación es la siguiente:

1. El usuario genera una petición al sistema; comúnmente esta tiene como propósito acceder a una página de nuestro sistema o realizar una determinada operación dentro de este. Esta petición es recibida por el controlador.

2. El controlador se pone en contacto tanto con el modelo como con la vista, al modelo le pide datos o le manda a realizar actualizaciones, a la vista le pide la salida correspondiente que va a hacer visualiza por el usuario una vez que se hayan terminado las operaciones según la lógica de negocio.
3. Para generar la salida, se puede dar el caso de que la vista solicite más información al modelo; en otras ocasiones el controlador será el encargado de solicitar los datos al modelo y enviarlos a la vista actuando como conector entre estos.
4. Finalmente la vista presenta la salida al usuario, aunque en algunos casos esa salida puede regresar al controlador siendo este el encargado de hacer el envío al cliente.

## 2.5. Sistema Gestor de Base de Datos SGBD

Se denomina SGBD a un sistema con el cual se puede gestionar la información alojada en una Base de Datos, en donde el gestionar puede ser considerada como cualquier operación que afecte directamente a la base de datos.

Un SGBD debe permitir:

- **Definir una Base de Datos:** Se trata de especificar las estructuras, tipos de datos y restricciones para los diferentes datos que se almacenarán.
- **Construir la Base de Datos:** Se refiere al proceso de almacenar los datos sobre un medio de almacenamiento controlado.
- **Manipular la Base de Datos:** Se refiere a realizar consultas, actualización, generación de informes.

Entre las principales características de un SGBD tenemos:

- Abstracción de la información.
- Control de la concurrencia.
- Respaldo y recuperación
- Consistencia



- Redundancia mínima
- Seguridad
- Integridad
- Independencia.
- Una Base de datos se considera a un banco de datos ordenados y relacionados entre sí, los cuales van a ser utilizados posteriormente de acuerdo a las necesidades del negocio.

### **2.5.1. Base de Datos Relacionales**

Este tipo de Bases de Datos se origina a través de la construcción de Tablas, Registros (filas), Campos (columnas), el fundamento de este tipo de Bases de Datos está en que las tablas se encuentran relacionadas mediante un campo en común en dos o más tablas llamadas claves primarias y claves foráneas.

Las Bases de Datos Relacionales se han implementado a través de los años evidenciando un buen rendimiento en aplicaciones de negocio y sistemas debido a las fortalezas que presenta como tipos de datos sencillos, protección elevada, consulta de datos fuertes; están también pasan por un proceso denominado `Normalización` de una base de datos mediante el cual se optimiza la utilización de la Base de Datos.

Entre las principales características de este tipo de Bases de datos tenemos:

- Se componen de varias tablas con sus respectivas relaciones.
- Inexistencia de dos tablas con nombres iguales.
- Cada tabla se encuentra formada por un conjunto de registros denominados filas y columnas.
- Las relaciones entre tabla padre y tabla hija se realizan a través de una clave primaria y clave foránea respectivamente.

- A las claves primarias se las reconoce como la clave principal dentro de una tabla, se encuentran en la tabla padre debiendo cumplir con la integridad de datos.
- Las claves foráneas se encuentran en la tabla hija, debiendo tener el mismo contenido de la clave primaria de la tabla padre; de esta manera se relacionan las tablas.

Entre los principales SGBD de este tipo de Base de Datos se encuentran: Oracle, PostgreSQL y SQL server

### **Oracle**

Es un Gestor de Base de Datos potente utilizado a nivel mundial, aunque a pesar de su gran potencia, solo es visto en grandes empresas por su elevado costo.

Esta almacenada físicamente en ficheros, la correspondencia entre los ficheros y las tablas se dan gracias a las estructuras internas de la BDD, que permiten que varios tipos de datos estén almacenados físicamente por separado; estas divisiones lógicas se hacen gracias a los llamados TableSpaces.

Entre las características más destacadas de Oracle están:

Soporte de transacciones (Select, Insert, Delete, Update, entre otras)

Apoya en el diseño y optimización del modelo de datos.

Soporte en la definición de estándares de diseños y nomenclatura de objetos.

Seguridad en el acceso de los datos debido a la gestión de privilegios.

Permite copias de seguridad

Escalabilidad y estabilidad.

### **POSTGRESQL**

Es un Gestor de Base de Datos objeto-relacional, con código fuente distribuido libremente y no hay costo asociado con la licencia.

Utiliza un modelo Cliente/Servidor y multiprocesos garantizando la estabilidad del sistema, puesto que de existir un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando; también incluye un completo modelo

de seguridad basado en el soporte de un protocolo de comunicación encriptado por SSL.

Entre las características destacadas de PostgreSQL están:

Instalación ilimitada.

Diseñado para ambiente de alto volumen

Estabilidad y confiabilidad

DBMS Objeto-Relacional siendo capaz de manejar complejas rutinas y reglas.

Integridad referencial la que se utiliza para garantizar la validez de los datos de la base.

### **SQL SERVER 2008 R2**

Es un sistema de administración de datos robusto, confiable y eficaz el cual permite obtener algunas características de manejo de datos, protección de datos de una base y alto rendimiento en ejecución para aplicaciones web, almacenes de datos locales entre otras. En si está diseñado para una implementación sencilla, una creación de prototipos rápida conservando siempre la integridad de los datos y asegurando disponibilidad de los datos de una manera eficiente, se integra fácilmente a infraestructuras de desarrollo como Visual Studio facilitando así la experiencia de creación de aplicaciones o sistemas en general.

Entre las características de SQL SERVER 2008 R2 están:

Optimiza las consultas.

Disponibilidad, confiabilidad y tolerancia a fallos.

Password y encriptación de datos en red para mejorar la seguridad.

Mayor seguridad en datos gracias a la existencia de un único ID de login.

Permite encriptar procedimientos almacenados para brindar seguridad e integridad de código de aplicación.

## Comparación de los SGDB

Cuadro Comparativo de los SGBD expuestos anteriormente:

Tabla 2. Cuadro comparativo entre BDD.

| CARACTERÍSTICA                              | ORACLE | POSTGRESQL | SQL SERVER |
|---|--------|------------|------------|
| Velocidad                                   | ✓      | ✓          | ✓          |
| Versión Express                             | ✓      |            | ✓          |
| Multiproceso                                | ✓      | ✓          | ✓          |
| Multi-hilo                                  | ✓      | ✓          | ✓          |
| Sistema de contraseñas                      | ✓      | ✓          | ✓          |
| Compatibilidad con autenticación de Windows |        |            | ✓          |
| Cifrado de información                      | ✓      | ✓          | ✓          |
| Utiliza Estándar SQL                        | ✓      | ✓          | ✓          |
| Conexión TCP/IP                             | ✓      | ✓          | ✓          |
| Conexión inmediata con Visual Studio        |        |            | ✓          |
| Concurrencia                                | ✓      | ✓          | ✓          |
| Compatibilidad con Active Directory         |        |            | ✓          |
| Integridad (Referencial)                    | ✓      | ✓          | ✓          |

Al hacer un análisis al cuadro anterior y revisar otro tipo de características se determinó realizar el sistema con el gestor de base de datos **SQL SERVER 2008 R2** en su versión EXPRESS la cual es gratis.

## 2.6. CSS

Es una tecnología que se basa en hojas de estilo las cuales nos permiten controlar la apariencia de un proyecto web, al principio los sitios web ponían más atención en el contenido que en la presentación, html como tal no se interesa en la apariencia del documento css hace expone la manera en la que los elementos son presentados al usuario. De una manera general CSS complementa al html o a una hoja web. Con CSS podemos editar diferentes

estilos de textos, tamaño, color, el lugar donde queremos poner texto e imágenes en las páginas. Se lo utiliza para dar un estilo diferente a documentos como HTML, XML y XHTML.

## **2.7. Reportadores**

Son herramientas que permite realizar reportes o informes desde una base de datos y presentarlos de una manera ordenada, esta a su vez puede ser exportada hacia documentos de texto.

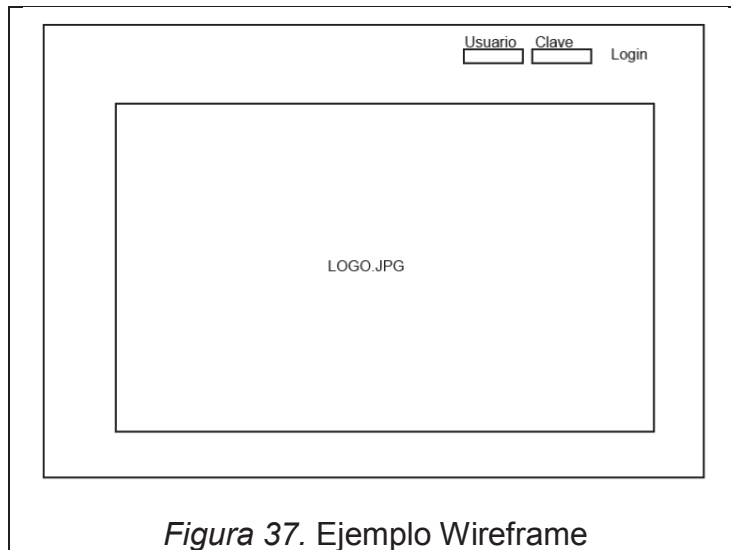
### **2.7.1. Reporting Services**

Es una herramienta que nos permite crear, implementar y administrar informes (reportes) interactivos, tabulares, gráficos o de forma libre. Esta herramienta funciona en un entorno Microsoft Visual Studio y se integra con SQL Server.

## **2.8. Wireframe**

Es una representación visual que muestra una idea del contenido de una pantalla, se lo conoce también como el esquema de una página o el esqueleto de la misma.

El objetivo principal de un wireframe se encuentra en representar con rasgos sencillos (en 2 -D) el diseño de una página web, la pantalla de una aplicación, etc.



La información que deben contener el/los wireframes es:

- Listado de contenido de las pantallas.
- Elementos cómo: cabeceras, listas, enlaces, imágenes, etc.
- Títulos, subtítulos, etc.
- Layout.
- Notas o detalles que nos muestren comportamientos de algunos de los elementos que contiene la página.

Ventajas de usar Wireframes:

- Tanto el cliente y el equipo se centran en el diseño de contenidos y no solamente en el diseño visual.
- Son fáciles de elaborarlos ya que solo muestran un bosquejo de una pantalla y los elementos que pueden contener.
- No necesitar tener
- Al definir una estructura previa al diseño y a la programación se puede evitar cambios posteriores que a su vez pueden generar costos y tiempos innecesario al proyecto

## 2.9. Javascript

Java script es un lenguaje de programación que permite crear pequeños programas, estos programas pueden ser insertados dentro de páginas WEB o en programas más grandes.

El lenguaje Javascript es Opensource gracias a esto puede ser utilizado por cualquier persona sin necesidad de comprar una licencia para su uso.

Con javascript se puede crear diferentes efectos que pueden interactuar con la página o programa con el usuario.

El lenguaje javascript es soportado por la mayoría de los navegadores como son: Internet Explorer, Google Chrome, Mozilla Firefox, Opera, etc.

### 3. DESARROLLO DEL SISTEMA

#### 3.1 Scrum

Primero vamos a exponer algunas herramientas y diferentes datos que han sido utilizados en el desarrollo de este Trabajo de Titulación siguiendo la metodología ágil Scrum

##### 3.1.1 Datos Generales

Tabla 3. Datos las iteraciones del proyecto

| <b>Datos de Sprints</b> |                 |
|-------------------------|-----------------|
| Número de Sprints       | <b>40</b>       |
| Número real de Sprints  | <b>45</b>       |
| Duración de cada Sprint | <b>7 días</b>   |
| Horas de cada Sprint    | <b>28 Horas</b> |

Tabla 4. Roles

| <b>Roles</b>    |                                   |
|-----------------|-----------------------------------|
| <b>Rol</b>      | <b>Responsable</b>                |
| Patrocinadores  | UDLA                              |
| Administradores | Fernando Sanchez, Jurgen Zambrano |
| Product Owner   | UDLA                              |
| Scrum Master    | Juan José León                    |
| Desarrolladores | Fernando Sánchez, Jurgen Zambrano |

##### 3.1.2 Product Backlog

A continuación como ejemplo de la utilización de la metodología Scrum en el presente proyecto se coloca un fragmento del product backlog

Tabla 5. Muestra de 20 tareas dentro del product backlog



| ID Historia | Nombre de la Historia  | Fecha de Culminación | Tamaño (días) | Sprint | Comentarios |
|-------------|--|----------------------|---------------|--------|-------------|
| 1           | Planteamiento del capítulo 1   | 29/04/2013           | 1             | 1      |             |
| 2           | Redacción de antecedentes, marco referencial                                   | 30/04/2013           | 1             | 1      |             |
| 3           | Redacción de alcance y justificación   | 01/05/2013           | 1             | 1      |             |
| 4           | Redacción de objetivos generales y específicos                                 | 02/05/2013           | 1             | 1      |             |
| 5           | Redacción general de la Introducción   | 03/05/2013           | 1             | 1      |             |
| 6           | Revisión del capítulo 1  | 04/05/2013           | 0,5           | 1      |             |
| 7           | Corrección del capítulo 1  | 05/05/2013           | 1,5           | 1      |             |
| 8           | Planteamiento del capítulo 2   | 06/05/2013           | 1             | 2      |             |
| 9           | Investigación acerca de metodologías de desarrollo                             | 07/05/2013           | 1             | 2      |             |
| 10          | Elaboración de cuadro comparativo de metodologías de desarrollo                | 08/05/2013           | 1             | 2      |             |
| 11          | Investigación acerca de metodologías de desarrollo de software tradicionales   | 09/05/2013           | 1             | 2      |             |
| 12          | Redacción de lo investigado acerca de metodologías de desarrollo tradicionales | 10/05/2013           | 1             | 2      |             |
| 13          | Investigación acerca de metodologías de desarrollo de software ágiles          | 11/05/2013           | 1             | 2      |             |
| 14          | Redacción de lo investigado acerca de metodologías de desarrollo ágiles        | 12/05/2013           | 1             | 2      |             |
| 15          | Lectura e investigación acerca de metodología Scrum                            | 14/05/2013           | 1,5           | 3      |             |
| 16          | Redacción acerca de Metodología Scrum  | 15/05/2013           | 1             | 3      |             |
| 17          | Lectura e investigación acerca de Diagramas UML                                | 17/05/2013           | 1,5           | 3      |             |
| 18          | Redacción acerca de Diagramas UML  | 19/05/2013           | 2             | 3      |             |
| 19          | Revisión del capítulo 2  | 21/05/2013           | 1,5           | 4      |             |
| 20          | Corrección del capítulo 2  | 23/05/2013           | 2             | 4      |             |

### 3.1.3 Iteraciones

En el presente Trabajo de Titulación se realizaron 45 sprints, a continuación se presenta el segundo sprint como muestra de la utilización de la metodología Scrum

Tabla 6. Segunda iteración

#### Sprint 2

**Fecha Inicio**

: 06/05/2013

**Fecha Fin:** 12/05/2013

| Tarea  | Responsable                       | Estado    | Horas Planeadas | Horas Realizadas | Fecha Planeada fin | Fecha de Culminación |
|--|-----------------------------------|-----------|-----------------|------------------|--------------------|----------------------|
| Investigación de metodologías de desarrollo                                    | Fernando Sánchez, Jorgen Zambrano | Culminado | 8               | 4                | 08/05/2013         | 07/05/2013           |
| Elaboración de cuadro comparativo de metodologías de desarrollo                | Fernando Sánchez, Jorgen Zambrano | Culminado | 6               | 4                | 10/05/2013         | 08/05/2013           |
| Investigación acerca de metodologías de desarrollo de software tradicionales   | Fernando Sánchez, Jorgen Zambrano | Culminado | 8               | 4                | 12/05/2013         | 09/05/2013           |
| Redacción de lo investigado acerca de metodologías de desarrollo tradicionales | Fernando Sánchez, Jorgen Zambrano | Culminado | 6               | 4                | 14/05/2013         | 10/05/2013           |
| Investigación acerca de metodologías de desarrollo de software ágiles          | Fernando Sánchez, Jorgen Zambrano | Culminado | 8               | 4                | 16/05/2013         | 11/05/2013           |

|   |                                   |           |   |   |            |            |
|---|-----------------------------------|-----------|---|---|------------|------------|
| Redacción de lo investigado acerca de metodologías de desarrollo ágiles | Fernando Sánchez, Jurgén Zambrano | Culminado | 7 | 4 | 18/05/2013 | 12/05/2013 |
|---|-----------------------------------|-----------|---|---|------------|------------|

### 3.1.4 Scrum Daily Meeting

En el proceso de desarrollo del proyecto nos fue imposible reunirnos diariamente con el Scrum Master (Profesor Guía). Sin embargo a través de chats y archivos compartidos teníamos contacto de manera seguida.

Tabla 7. Resumen de Reuniones del segundo Sprint

Sprint 2

**Fecha**

**Inicio :** 06/05/2013

**Fecha**

**Fin :** 12/05/2013

| # de Historia | Fecha      | ¿Qué has hecho desde ayer?                                | ¿Qué vas a hacer hoy?   | ¿Existe algún impedimento? | Estado del Equipo | Estado Scrum Master |
|---------------|------------|---|---|----------------------------|-------------------|---------------------|
| 9             | 07/05/2013 | Plantear ideas acerca del capítulo 2                      | Investigar acerca de metodologías de desarrollo                           | Ninguno                    | Preparado         | OK                  |
| 10            | 08/05/2013 | Investigar acerca de metodologías de desarrollo           | Elaborar cuadro comparativo de metodologías de desarrollo                 | Ninguno                    | Preparado         | Ok                  |
| 11            | 09/05/2013 | Elaborar cuadro comparativo de metodologías de desarrollo | Investigar acerca de metodologías de desarrollo de software tradicionales | Ninguno                    | Preparado         | Ok                  |
| 12            | 10/05/2013 | Investigar acerca de                                      | Redactar acerca de  | Ninguno                    | Preparado         | Ok                  |

|    |            |  |  |         |           |    |
|----|------------|--|--|---------|-----------|----|
|    |            | metodologías de desarrollo de software tradicionales               | metodologías de desarrollo tradicionales                           |         |           |    |
| 13 | 11/05/2013 | Redactar acerca de metodologías de desarrollo tradicionales        | Investigar acerca de metodologías de desarrollo de software ágiles | Ninguno | Preparado | Ok |
| 14 | 12/05/2013 | Investigar acerca de metodologías de desarrollo de software ágiles | Redactar acerca de metodologías de desarrollo ágiles               | Ninguno | Preparado | Ok |

### 3.2. Análisis de Requerimientos.

En esta parte se determinaron los requerimientos funcionales, no funcionales, del sistema por medio de una reunión con la Ing. Lucia Toledo, coordinadora de Trabajo de Titulación de FICA de la UDLA, con el propósito de definir y fijar las condiciones sobre las cuales se va a desarrollar el sistema a fin de que este ayude con el control y optimización de las reuniones de tutorías de Trabajo de Titulación de la UDLA

#### 3.2.1. Requerimientos no Funcionales.

Son aquellas funciones que no realiza el sistema por medio de su desarrollo, pero que son necesarias para que este funcione correctamente; estos se listan a continuación:

- Solo se podrá ingresar al sistema dentro de la Universidad.
- El Sistema deberá regirse a los estándares del reglamento de titulación UDLA 2013.
- El sistema deberá ser amigable para el usuario en su interfaz.
- El tiempo de espera de los recursos del sistema debe ser mínimo y sostenido.

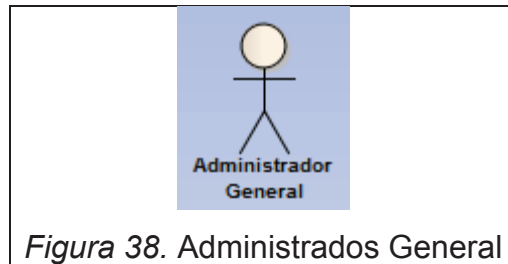
### 3.2.2. Requerimientos Funcionales.

Son aquellas funciones que va a realizar el sistema y para lo cual se ha desarrollado; estos se listan a continuación:

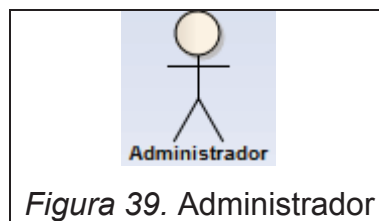
- Registro de reuniones de desarrollo de proyecto de titulación por parte de los profesores tutores
- Registro de reuniones de desarrollo de proyecto de titulación por parte de los alumnos que estén desarrollando su proyecto de titulación
- Creación de Proyectos de Titulación.
- Creación de profesores tutores de los proyectos de titulación.
- Creación de alumnos que van a desarrollar su proyecto de titulación.
- Asignación de estudiantes a un proyecto de titulación
- Asignación de profesores a un proyecto de titulación
- Generación de Reportes sobre reuniones de los proyectos de titulación.
- Acceso al sistema mediante login
- Paramétrica de aspectos relacionados al desarrollo del proyecto de titulación(ANEXO)

## Actores

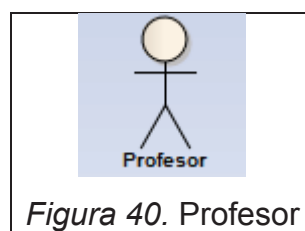
Administrador General, encargado de ingresar datos de alumnos, profesores, temas de Trabajo de Titulación.



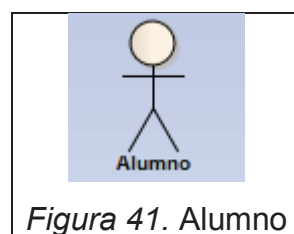
Administrador, encargada de generar reportes



Profesor, registra reuniones y habilitar el registro de alumno



Alumno, registra reuniones



### 3.3. Diagramas de Casos de Uso.

Diagrama de caso de uso para ingresar al sistema.

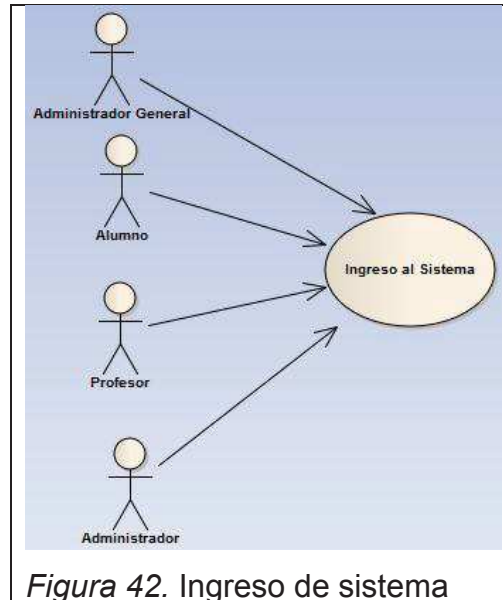


Figura 42. Ingreso de sistema

Casos de Uso Administrador General

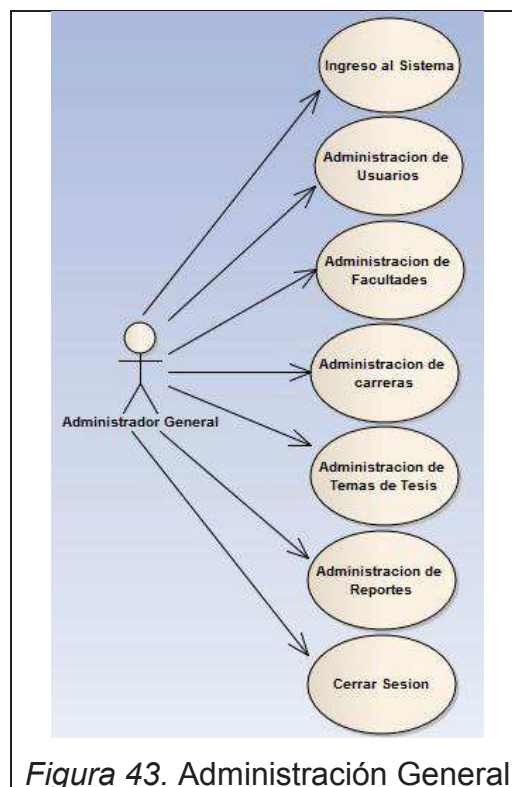


Figura 43. Administración General

## Casos de Uso Administrador

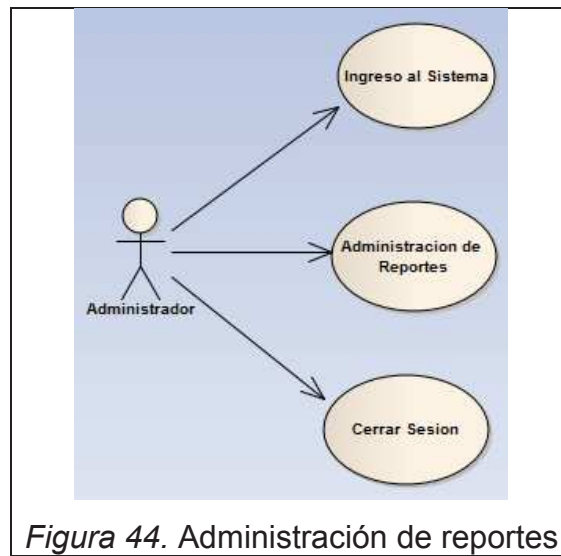


Figura 44. Administración de reportes

## Casos de Uso Profesor

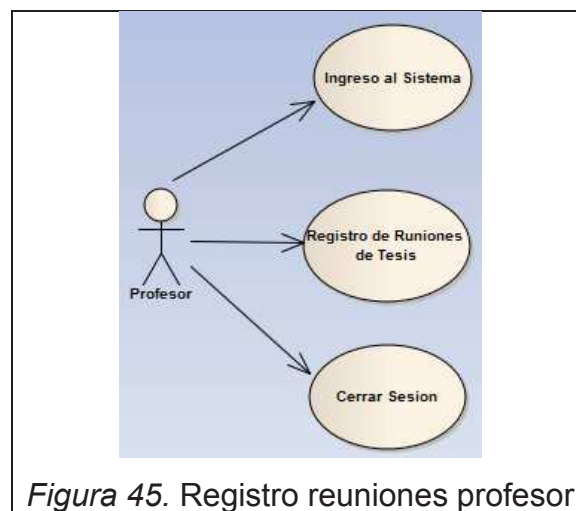


Figura 45. Registro reuniones profesor



## Casos de Uso Alumno

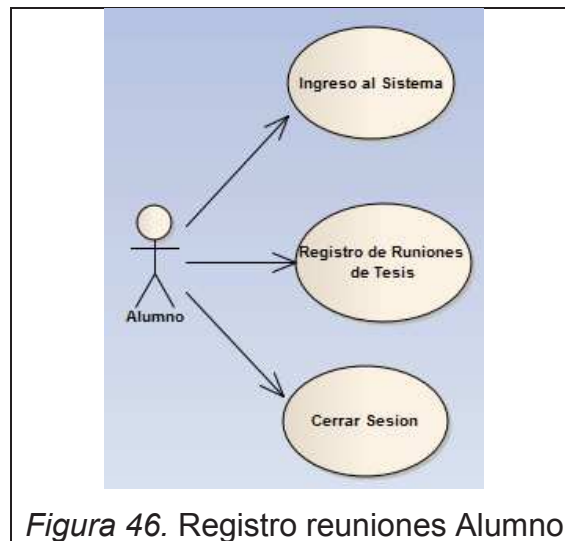


Figura 46. Registro reuniones Alumno

## Especificación de Casos de Uso

Tabla 8. Caso de Uso Ingreso Sistema

|   |
|---|
| Caso de Uso: Ingreso al Sistema   |
| Actores: Administrador General, Administrador, Profesor, Alumno   |
| Descripción: Permite a los actores ingresar al sistema, y visualizar los menús de acuerdo al perfil asignado  |
| Precondiciones: El actor debe estar registrado en el sistema por parte del Administrador General y tener un perfil asignado   |
| Flujo Normal:<br>El actor accede a la pantalla de ingreso al sistema.<br>El actor digita su nombre de usuario y contraseña del sistema.<br>El actor presiona el botón Ingresar.<br>Se muestra la pantalla con las opciones habilitadas dependiendo del perfil del actor |
| Flujo Alternativo:<br>1. El actor digita su nombre de usuario y contraseña de manera incorrecta.<br>2. El actor presiona el botón Ingresar.<br>3. Se muestra un aviso de validación de datos incorrectos  |
| Pos condiciones: Ingreso al sistema visualizando las opciones habilitadas de acuerdo al perfil  |

Tabla 9. Caso de uso Administración de usuarios

|  |
|--|
| Caso de Uso: Administración de Usuarios  |
| Actores: Administrador General   |
| Descripción: Permite al Administrador General Crear, Modificar y Eliminar a los Usuarios   |
| Precondiciones: Se ingrese al sistema con éxito  |
| Flujo Normal:<br>Crear Usuario:<br>1.1. El Administrador General selecciona la opción Usuarios<br>El sistema muestra la pantalla de Usuarios |

|  |
|--|
| <p>El Administrador General presiona el botón Crear Usuario.<br/> El sistema muestra una pantalla con los respectivos campos para ingresar datos.<br/> El Administrador General ingresa los datos requeridos, asigna un perfil al usuario y presiona el botón Crear.<br/> El sistema guarda la información y se actualiza la lista de usuarios.</p> <p>Modificar Usuario:</p> <ol style="list-style-type: none"> <li>2.1. El Administrador General selecciona la opción Usuarios.</li> <li>2.2. El sistema muestra la pantalla de Usuarios.</li> <li>2.3. El Administrador General presiona sobre la opción de Editar del usuario seleccionado.</li> <li>2.4. El sistema muestra una pantalla con los respectivos datos del usuario.</li> <li>2.5. El Administrador General modifica los datos necesarios y presiona el botón Guardar.<br/> El sistema guarda la información con los datos modificados y se actualiza la lista de usuarios.</li> </ol> <p>3. Eliminar Usuario:</p> <ol style="list-style-type: none"> <li>3.1. El Administrador General selecciona la opción Usuarios.</li> <li>3.2. El sistema muestra la pantalla de Usuarios.</li> <li>3.3. El Administrador General presiona sobre la opción de Eliminar del usuario seleccionado.</li> <li>3.4. El sistema elimina del registro al usuario y se actualiza la lista de usuarios</li> </ol> |
| <p>Flujo Alternativo:</p> <ol style="list-style-type: none"> <li>1. El Administrador General selecciona la opción incorrecta para crear o modificar</li> <li>2. El administrador General presiona el botón Cancelar</li> <li>3. El sistema muestra la lista de Usuarios</li> </ol>   |
| <p>Pos condiciones: Operación realizada correspondiente a Administración de Usuarios</p>   |

Tabla 10. Caso de Uso Administración de facultades

|   |
|---|
| Caso de Uso: Administración de Facultades   |
| Actores: Administrador General  |
| Descripción: Permite al Administrador General Crear, Modificar y Eliminar Facultades  |
| Precondiciones: Se ingrese al sistema con éxito   |
| <p>Flujo Normal:</p> <ol style="list-style-type: none"> <li>1. Crear Facultad: <ol style="list-style-type: none"> <li>1.1. El Administrador General selecciona la opción Facultad</li> <li>1.2 El sistema muestra la pantalla de Facultad.</li> <li>1.3 El Administrador General presiona el botón Crear Facultad.</li> <li>1.4 El sistema muestra una pantalla con los respectivos campos para ingresar datos.</li> <li>1.5 El Administrador General ingresa los datos requeridos y presiona el botón Crear.</li> <li>1.6 El sistema guarda la información y se actualiza la lista de Facultad.</li> </ol> </li> </ol> <p>Modificar Facultad:</p> <ol style="list-style-type: none"> <li>2.1. El Administrador General selecciona la opción Facultad.</li> <li>2.2. El sistema muestra la pantalla de Facultad.</li> <li>2.3. El Administrador General presiona sobre la opción de Editar de la Facultad seleccionada.</li> <li>2.4. El sistema muestra una pantalla con los respectivos datos de la facultad.</li> <li>2.5. El Administrador General modifica los datos necesarios y presiona el botón Guardar.</li> <li>2.6. El sistema guarda la información con los datos modificados y se actualiza la lista de facultades.</li> </ol> <p>3. Eliminar Facultad:</p> <ol style="list-style-type: none"> <li>3.1. El Administrador General selecciona la opción Facultad.</li> <li>3.2. El sistema muestra la pantalla de Facultad.</li> <li>3.3. El Administrador General presiona sobre la opción de Eliminar de la Facultad seleccionada.</li> <li>3.4. El sistema elimina el registro de la Facultad y se actualiza la lista de facultades</li> </ol> |
| <p>Flujo Alternativo:</p> <ol style="list-style-type: none"> <li>1. El Administrador General selecciona la opción incorrecta para crear o modificar</li> </ol>  |

|   |
|---|
| 2. El administrador General presiona el botón Cancelar                              |
| 3. El sistema muestra la lista de facultades  |
| Pos condiciones: Operación realizada correspondiente a Administración de Facultades |

Tabla 11. Administración de Carreras

|   |
|---|
| Caso de Uso: Administración de Carreras   |
| Actores: Administrador General  |
| Descripción: Permite al Administrador General Crear, Modificar y Eliminar Carreras  |
| Precondiciones: Se ingrese al sistema con éxito y se haya creado previamente Facultades.  |
| Flujo Normal:<br>1. Crear Carrera:<br>1.1. El Administrador General selecciona la opción Carrera<br>1.2 El sistema muestra la pantalla de Carrera.<br>1.3 El Administrador General presiona el botón Crear Carrera.<br>1.4 El sistema muestra una pantalla con los respectivos campos para ingresar datos.<br>1.5 El Administrador General ingresa los datos requeridos y presiona el botón Crear.<br>1.6 El sistema guarda la información y se actualiza la lista de Carrera.<br>Modificar Carrera:<br>2.1. El Administrador General selecciona la opción Carrera.<br>2.2. El sistema muestra la pantalla de Carrera.<br>2.3. El Administrador General presiona sobre la opción de Editar de la Carrera seleccionada.<br>2.4. El sistema muestra una pantalla con los respectivos datos de la Carrera.<br>2.5. El Administrador General modifica los datos necesarios y presiona el botón Guardar.<br>2.6. El sistema guarda la información con los datos modificados y se actualiza la lista de carreras.<br>3. Eliminar Facultad:<br>3.1. El Administrador General selecciona la opción Carrera.<br>3.2. El sistema muestra la pantalla de Carrera.<br>3.3. El Administrador General presiona sobre la opción de Eliminar de la Carrera seleccionada.<br>3.4. El sistema elimina el registro de la Carrera y se actualiza la lista de carreras |
| Flujo Alternativo:<br>1. El Administrador General selecciona la opción incorrecta para crear o modificar<br>2. El administrador General presiona el botón Cancelar<br>3. El sistema muestra la lista de carreras  |
| Pos condiciones: Operación realizada correspondiente a Administración de Carreras   |

Tabla 12. Caso de uso Administración de Tema de trabajo de Titulación

|   |
|---|
| Caso de Uso: Administración Temas de Trabajo de Titulación  |
| Actores: Administrador General  |
| Descripción: Permite al Administrador General Crear, Modificar y Eliminar Temas de Trabajo de Titulación y asignarle un profesor guía   |
| Precondiciones: Se ingrese al sistema con éxito y se haya creado previamente Alumnos y Profesores.  |
| Flujo Normal:<br>1. Crear Tema de Trabajo de Titulación:<br>1.1. El Administrador General selecciona la opción Tema<br>1.2 El sistema muestra la pantalla de Tema de Trabajo de Titulación.<br>1.3 El Administrador General presiona el botón Crear Tema de Trabajo de Titulación.<br>1.4 El sistema muestra una pantalla con los respectivos campos para ingresar datos.<br>1.5 El Administrador General ingresa los datos requeridos y presiona el botón Crear. |

|  |
|--|
| <p>1.6 El sistema guarda la información y se actualiza la lista de Tema de Trabajo de Titulación.</p> <p>Modificar Tema de Trabajo de Titulación:</p> <p>2.1. El Administrador General selecciona la opción Tema.</p> <p>2.2. El sistema muestra la pantalla de Tema de Trabajo de Titulación.</p> <p>2.3. El Administrador General presiona sobre la opción de Editar el Tema de Trabajo de Titulación seleccionado.</p> <p>2.4. El sistema muestra una pantalla con los respectivos datos del Tema de Trabajo de Titulación.</p> <p>2.5. El Administrador General modifica los datos necesarios y presiona el botón Guardar.</p> <p>2.6. El sistema guarda la información con los datos modificados y se actualiza la lista de Tema de Trabajo de Titulación.</p> <p>3. Eliminar Tema de Trabajo de Titulación:</p> <p>3.1. El Administrador General selecciona la opción Tema.</p> <p>3.2. El sistema muestra la pantalla de Tema de Trabajo de Titulación.</p> <p>3.3. El Administrador General presiona sobre la opción de Eliminar del Tema de Trabajo de Titulación seleccionado.</p> <p>3.4. El sistema elimina el registro del Tema de Trabajo de Titulación y se actualiza la lista de Tema de Trabajo de Titulación</p> |
| <p>Flujo Alternativo:</p> <p>1. El Administrador General selecciona la opción incorrecta para crear o modificar</p> <p>2. El administrador General presiona el botón Cancelar</p> <p>3. El sistema muestra la lista de Tema de Trabajo de Titulación</p>   |
| <p>Pos condiciones: Operación realizada correspondiente a Administración de Tema de Trabajo de Titulación</p>  |

Tabla 13. Caso de uso Administración de Tema Usuario

|  |
|--|
| Caso de Uso: Administración Tema Usuario   |
| Actores: Administrador General   |
| Descripción: Permite al Administrador General Asignar, Modificar y Eliminar Estudiantes en un tema de Trabajo de Titulación  |
| Precondiciones: Se ingrese al sistema con éxito y se haya creado previamente Alumnos, Profesores y Temas de Trabajo de Titulación.   |
| <p>Flujo Normal:</p> <p>1. Asignar Tema de Trabajo de Titulación a Estudiante:</p> <p>1.1. El Administrador General selecciona la opción Tema Usuario</p> <p>1.2 El sistema muestra la pantalla de Tema Usuario.</p> <p>1.3 El Administrador General presiona en Ingresar Nuevo.</p> <p>1.4 El sistema muestra una pantalla con los respectivos campos para ingresar datos.</p> <p>1.5 El Administrador General ingresa los datos requeridos y presiona el botón Ingresar.</p> <p>1.6 El sistema guarda la información y se actualiza la lista de Tema Usuario.</p> <p>Modificar Estudiante a Tema de Trabajo de Titulación:</p> <p>2.1. El Administrador General selecciona la opción Tema Usuario.</p> <p>2.2. El sistema muestra la pantalla de Tema Usuario.</p> <p>2.3. El Administrador General presiona sobre la opción de Editar el Tema de Trabajo de Titulación seleccionado.</p> <p>2.4. El sistema muestra una pantalla con los respectivos datos del Tema de Trabajo de Titulación y estudiante(s) de ese tema de Trabajo de Titulación.</p> <p>2.5. El Administrador General modifica los datos necesarios y presiona el botón Guardar.</p> <p>2.6. El sistema guarda la información con los datos modificados y se actualiza la lista de Tema de Trabajo de Titulación y Usuarios de ese tema.</p> <p>3. Eliminar Tema Usuario:</p> <p>3.1. El Administrador General selecciona la opción Tema Usuario.</p> <p>3.2. El sistema muestra la pantalla de Tema de Usuario.</p> <p>3.3. El Administrador General presiona sobre la opción de Eliminar del Tema de Trabajo de</p> |

|   |
|---|
| Titulación seleccionado.<br>3.4. El sistema elimina el registro del Tema de Trabajo de Titulación y se actualiza la lista de Tema de Trabajo de Titulación  |
| Flujo Alternativo:<br>1. El Administrador General selecciona la opción incorrecta para crear o modificar<br>2. El administrador General presiona el botón Cancelar<br>3. El sistema muestra la lista de Tema de Trabajo de Titulación y usuario(s) de ese tema. |
| Pos condiciones: Operación realizada correspondiente a Administración de Tema Usuario   |

Tabla 14. Caso de uso administración de reportes

|  |
|--|
| Caso de Uso: Administración de Reportes  |
| Actores: Administrador General, Administrador  |
| Descripción: Permite al Administrador General y al Administrador generar reportes generales e individuales de los Alumnos, Profesores, Temas de Trabajo de Titulación y Reuniones  |
| Precondiciones: Se ingrese al sistema con éxito  |
| Flujo Normal:<br>1. El Actor selecciona la opción Reportes<br>2. El sistema muestra la pantalla de Reportes.<br>3. El Actor escoge el tipo de reporte a generar.<br>4. El Actor ingresa los parámetros del reporte a generar.<br>5. El Actor presiona el botón Generar<br>6. El sistema despliega el reporte en pantalla   |
| Flujo Alternativo:<br>5.1. No hay datos que mostrar.<br>6.1. El actor presiona el botón Exportar.<br>6.1.1. El sistema muestra pantalla para seleccionar el formato del reporte a exportar.<br>6.1.2. El Actor escoge el formato del reporte y presiona Aceptar<br>6.1.3. Se muestra una pantalla para escoger donde se desea que se guarde el reporte.<br>6.1.4 El actor escoge donde se desea que se guarde el reporte y presiona aceptar<br>6.1.5 Se guarda el reporte en la ubicación y formato escogido |
| Pos condiciones: Despliegue en pantalla del reporte de acuerdo con las opciones seleccionadas  |

Tabla 15. Casos de Uso Registro de reuniones profesor

|   |
|---|
| Caso de Uso: Registro de Reuniones de Trabajo de Titulación del profesor  |
| Actores: Profesor   |
| Descripción: Permite registrar una Reunión de Trabajo de Titulación como válida y habilitar el registro de reunión de Trabajo de Titulación al alumno.  |
| Precondiciones: Se ingrese al sistema con éxito, se haya creado el Tema de Trabajo de Titulación y asignado alumno(s) al Tema de Trabajo de Titulación.   |
| Flujo Normal:<br>1. Registrar la Reunión de Trabajo de Titulación:<br>1.1 El sistema muestra la pantalla de Reunión de Trabajo de Titulación.<br>1.2 El profesor presiona el botón Registrar Reunión de Trabajo de Titulación.<br>1.3 El sistema muestra una pantalla con los respectivos campos para ingresar datos.<br>1.4 El profesor ingresa los datos requeridos, se toma la foto con el (los) estudiante(s) la guarda y la sube al sistema, confirma su presencia con un check y presiona el botón Guardar. |

|   |
|---|
| 1.5 El sistema guarda el registro de la reunión y habilita el registro de reunión del alumno.   |
| Flujo Alternativo:<br>1. El profesor no tiene asignados temas de Trabajo de Titulación.<br>2. No existe alumnos asignados al tema de Trabajo de Titulación. |
| Pos condiciones: Registro de Reunión de Trabajo de Titulación del profesor guardada con éxito en el sistema.  |

Tabla 16. Casos de Uso Registro de reuniones alumno

|  |
|--|
| Caso de Uso: Registro de Reuniones de Trabajo de Titulación del alumno   |
| Actores: Alumno  |
| Descripción: Permite registrar una Reunión de Trabajo de Titulación como valida  |
| Precondiciones: Se ingrese al sistema con éxito, se haya creado el Tema de Trabajo de Titulación y se haya creado el registro de reunión por parte del profesor  |
| Flujo Normal:<br>1. Crear la Reunión de Trabajo de Titulación:<br>1.1 El sistema muestra la pantalla de Reunión de Trabajo de Titulación.<br>1.2 El alumno presiona el botón Registrar Reunión de Trabajo de Titulación.<br>1.3 El sistema muestra una pantalla con el check de registro.<br>1.4 El alumno confirma su presencia con un check y presiona el botón Guardar.<br>1.5 El sistema guarda el registro de la reunión. |
| Flujo Alternativo:<br>1. El alumno no tiene asignado un tema de Trabajo de Titulación.<br>2. No se encuentra habilitado el check de registro de Reunión de Trabajo de Titulación.  |
| Pos condiciones: Registro de Reunión de Trabajo de Titulación del alumno guardada con éxito en el sistema  |

### 3.4. Diagramas de Secuencia.

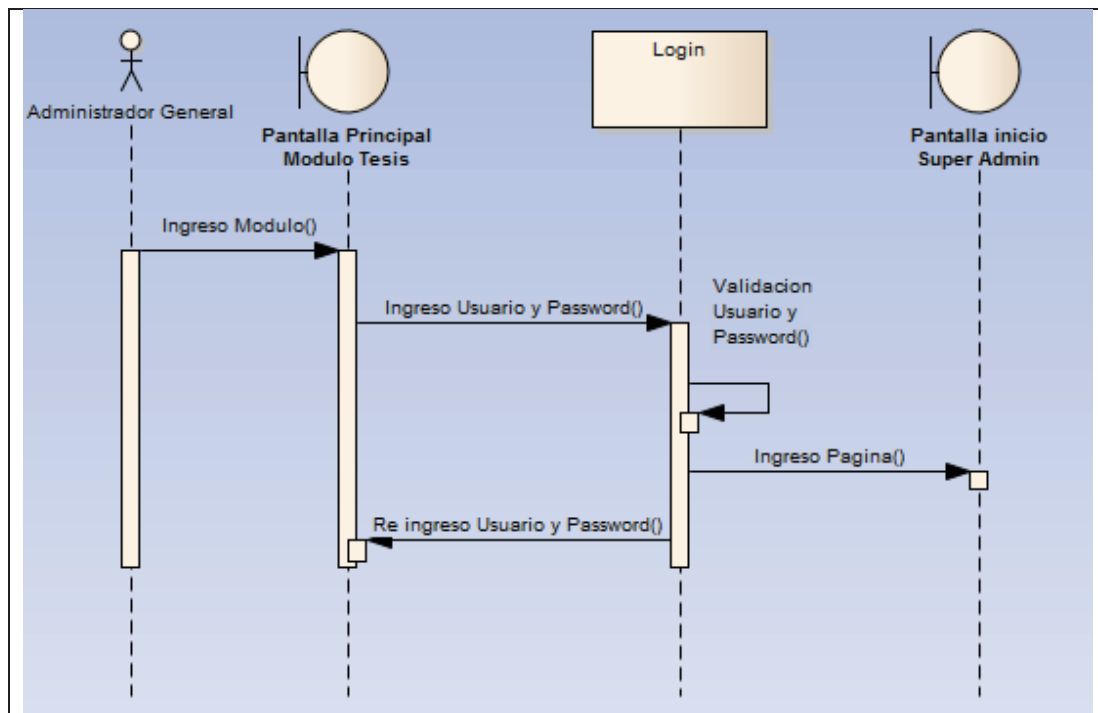


Figura 47. Diagrama de secuencia Login Administrador General

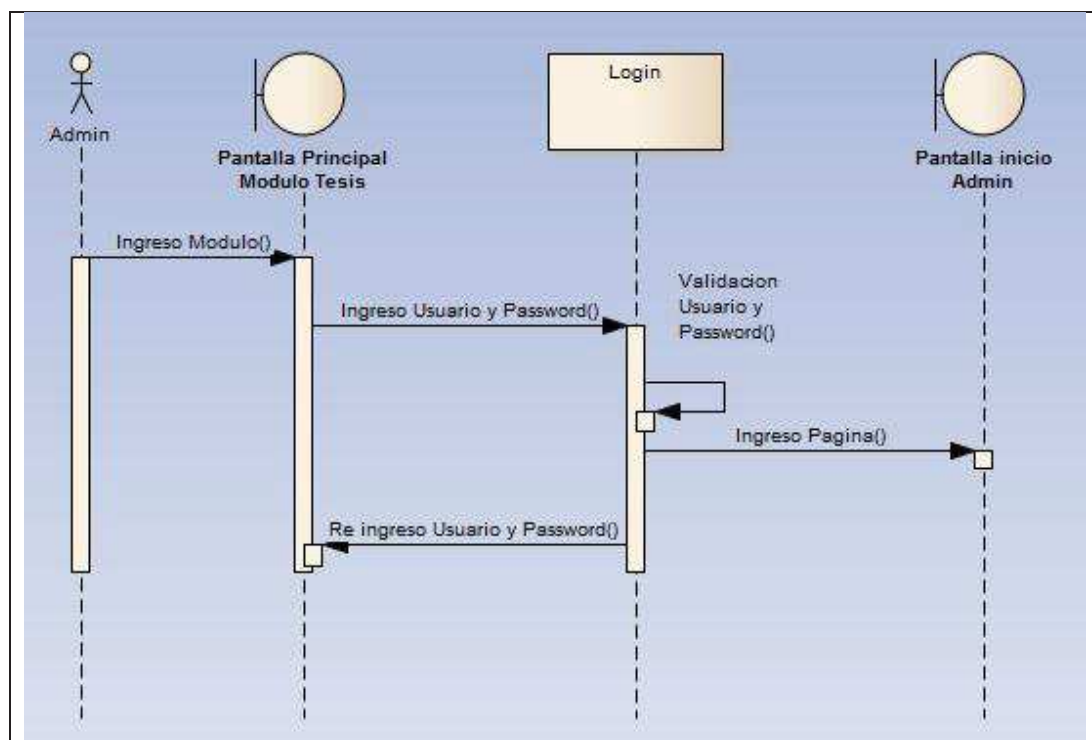


Figura 48. Diagrama de secuencia Login Administrador de Reportes

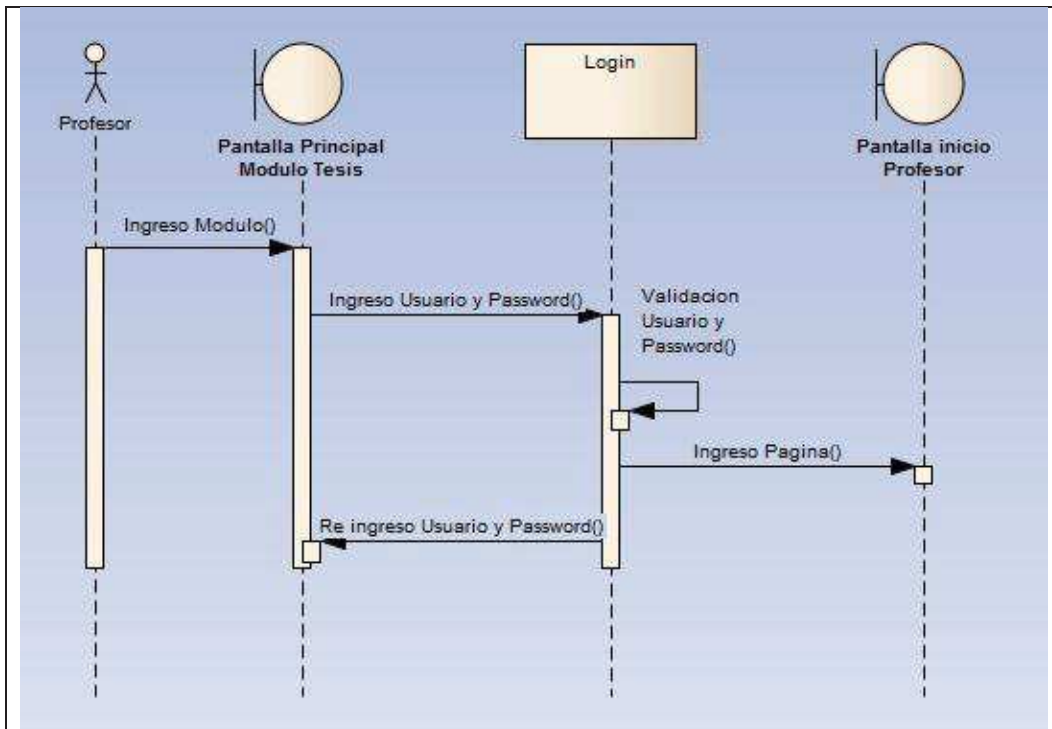


Figura 49. Diagrama de secuencia Login Profesor

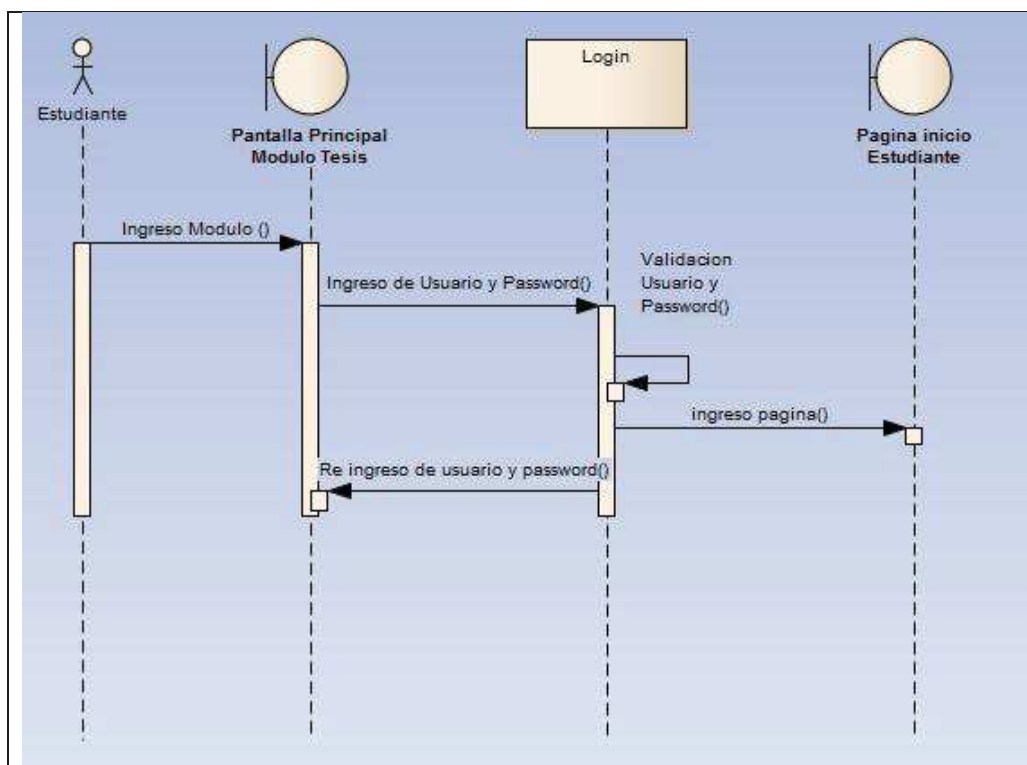


Figura 50. Diagrama de secuencia Login Alumno



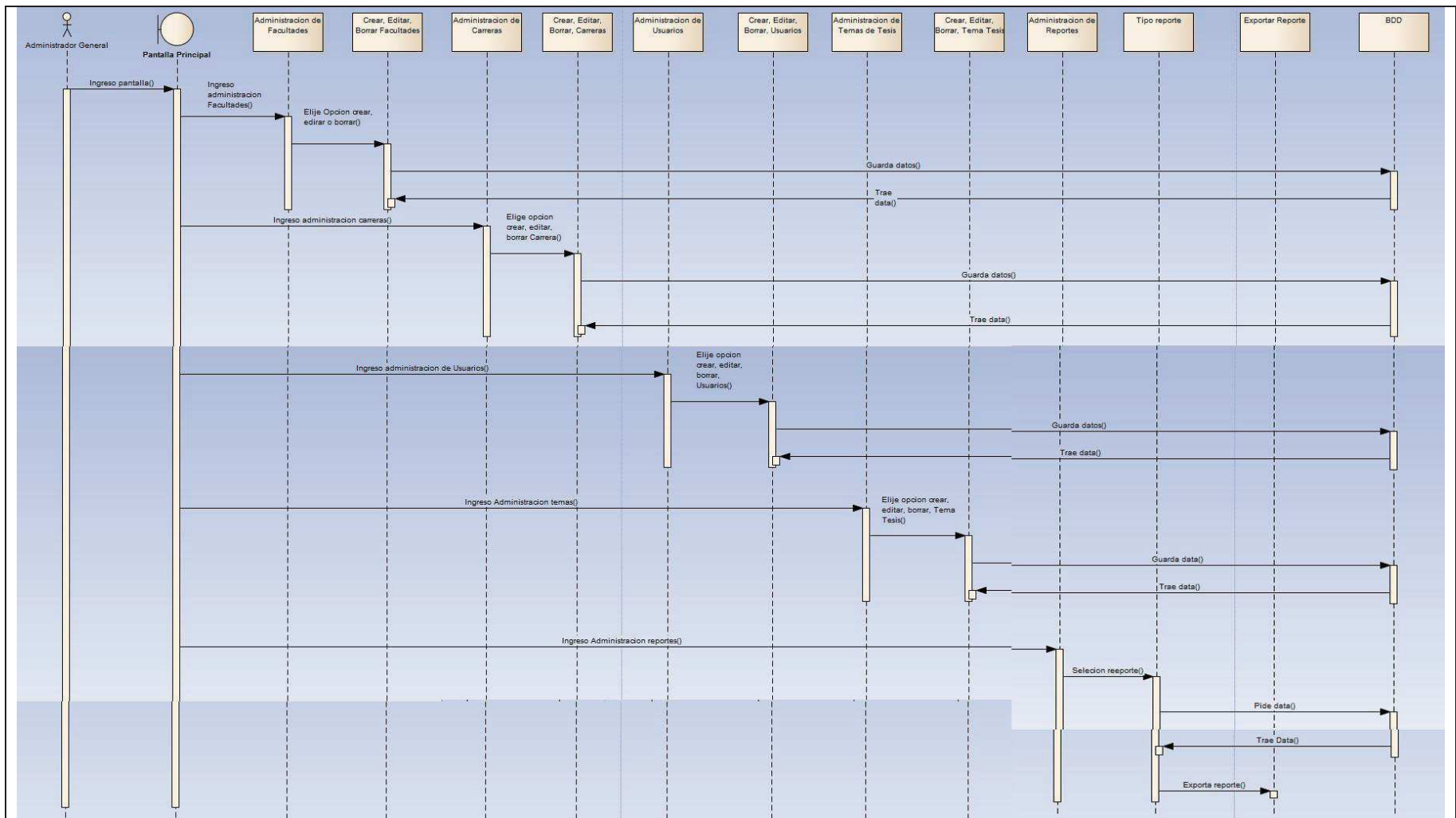


Figura 51. Diagrama Modulo administración general

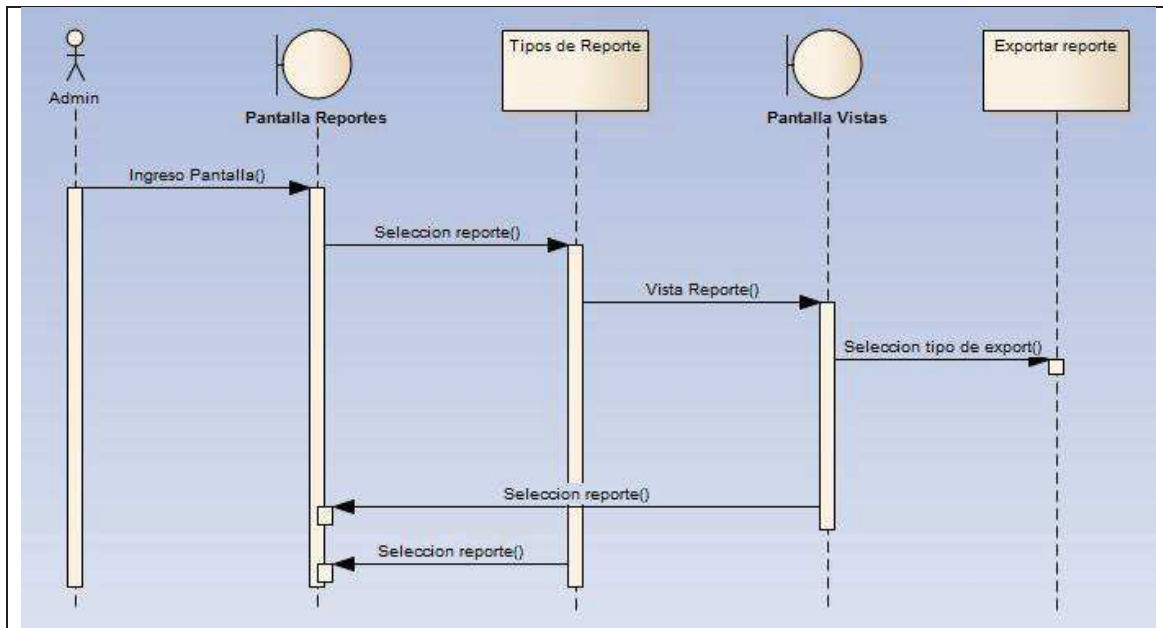


Figura 52. Módulo de secuencia Administración de Reportes

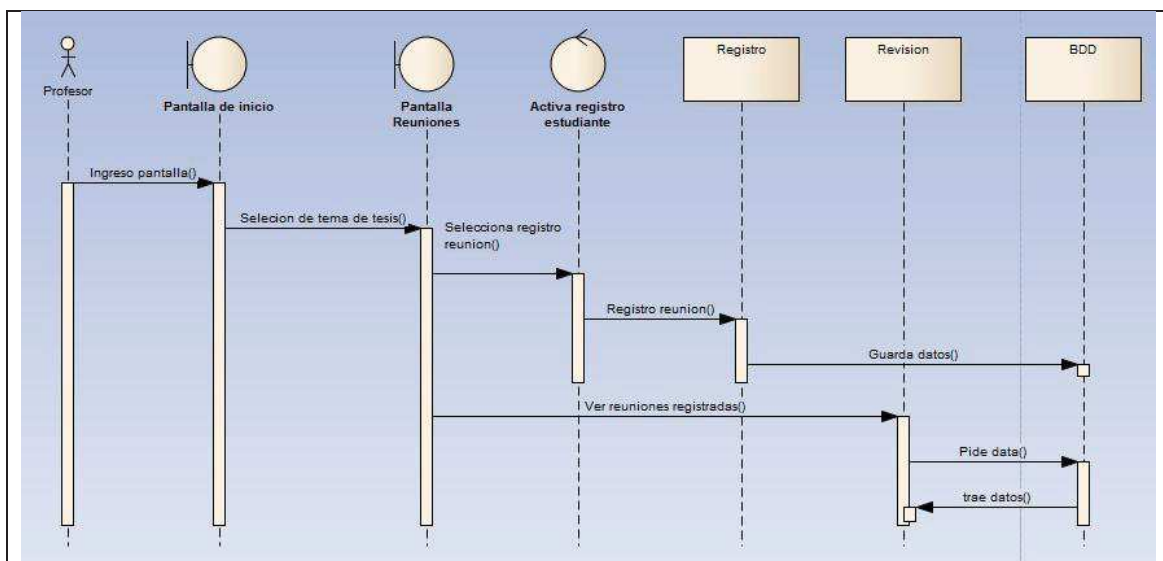


Figura 53. Diagrama de secuencia registro reuniones profesor

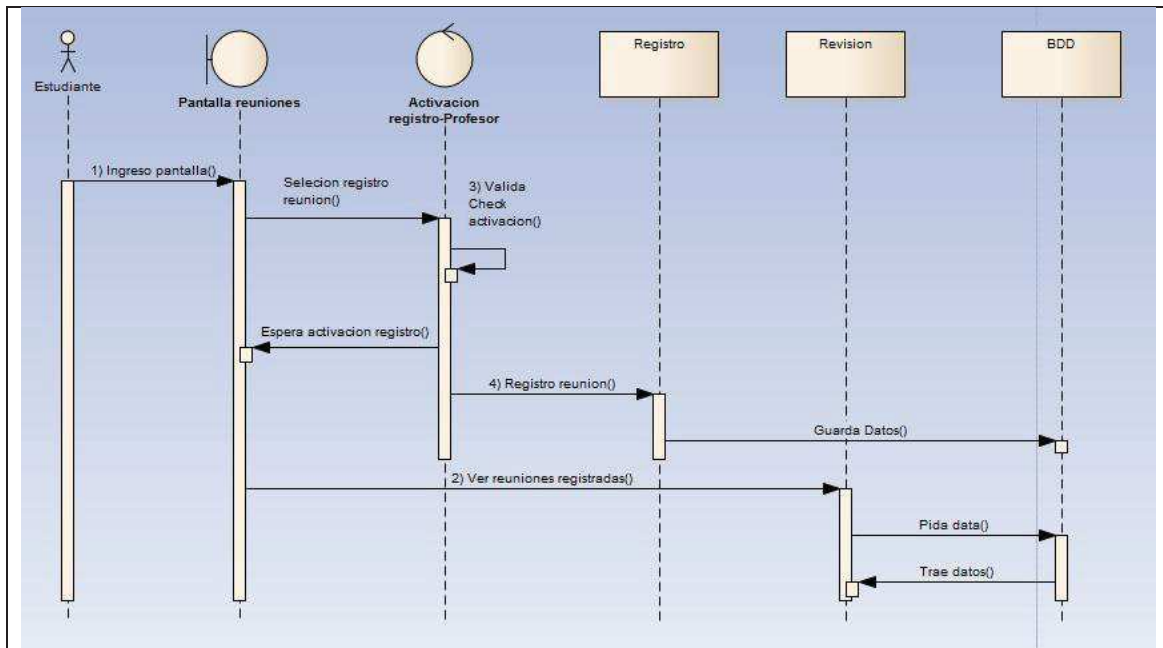


Figura 54. Diagrama de secuencia registro de reuniones Alumno

### 3.5. Diagramas de Actividades.

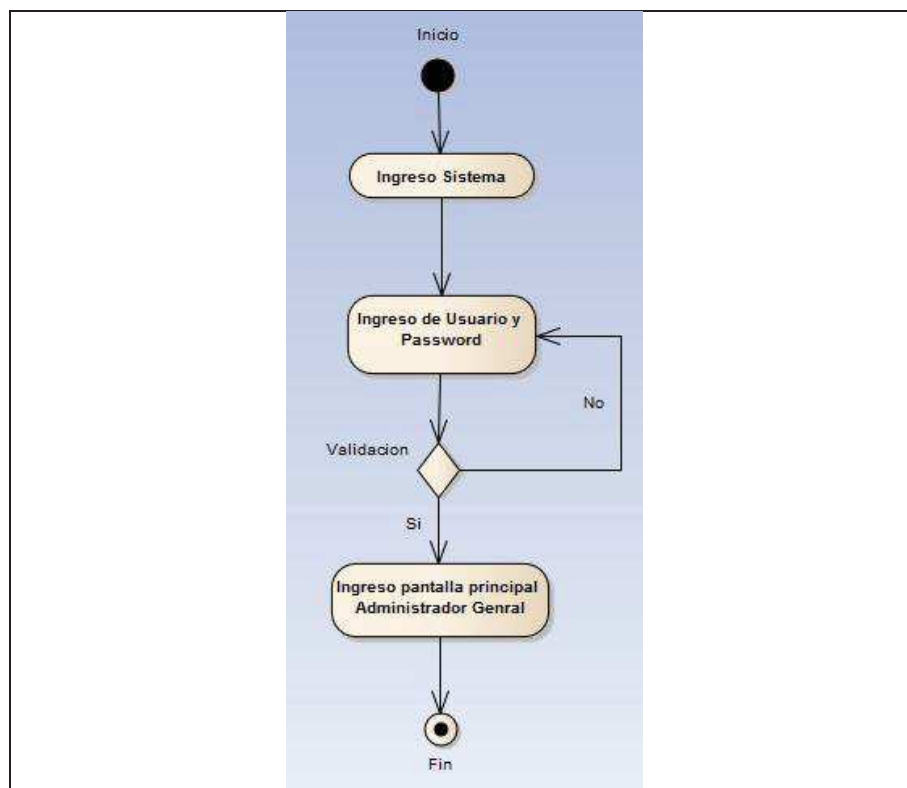


Figura 55. Diagrama de actividad Login Administrador general

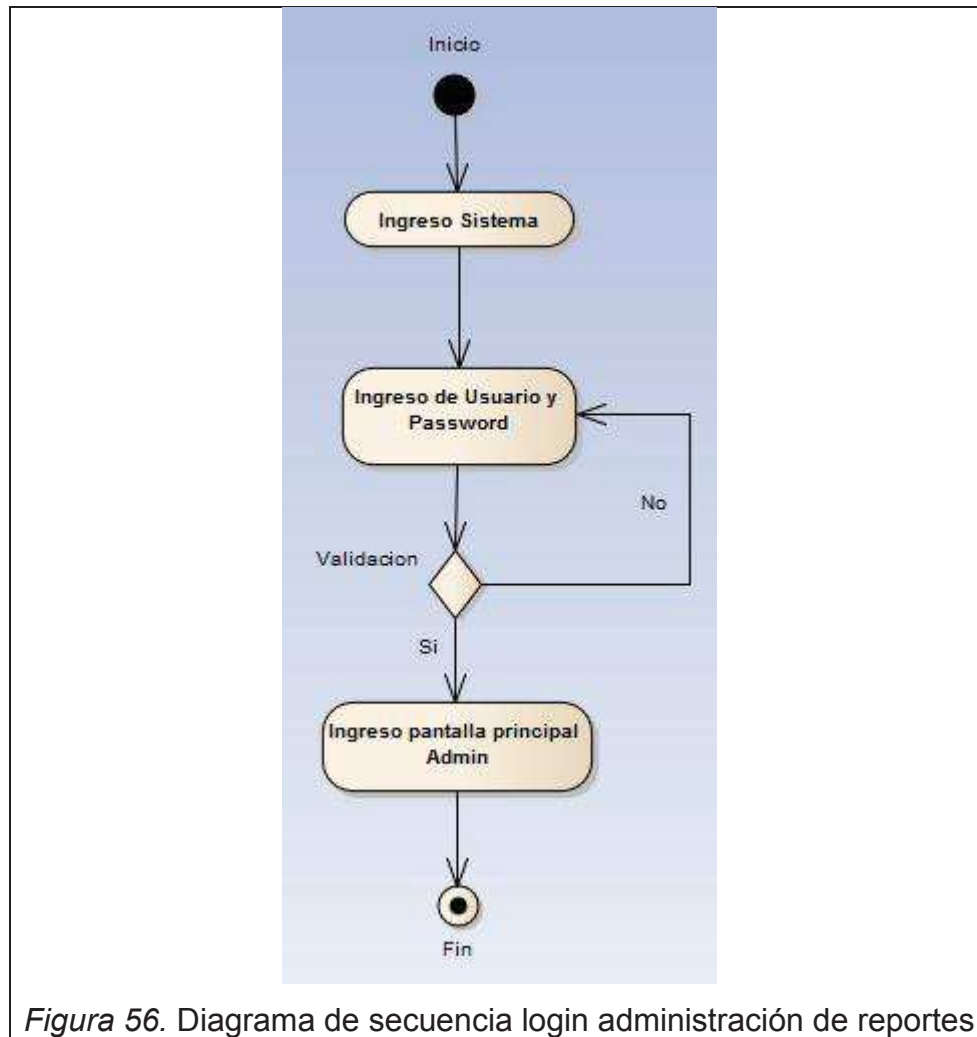


Figura 56. Diagrama de secuencia login administración de reportes

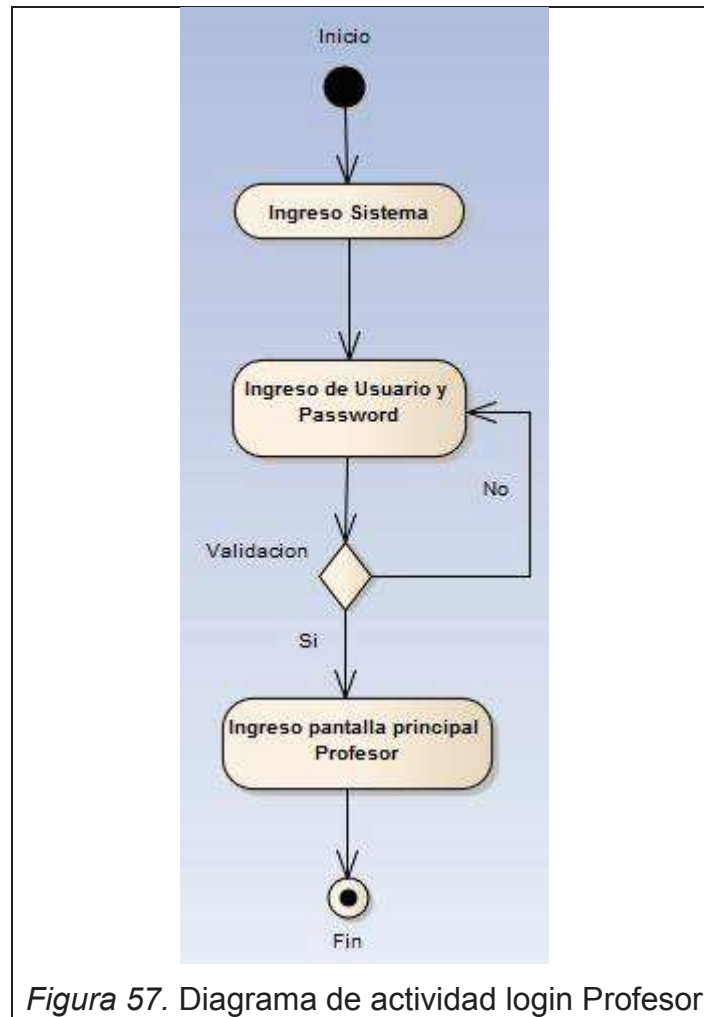


Figura 57. Diagrama de actividad login Profesor

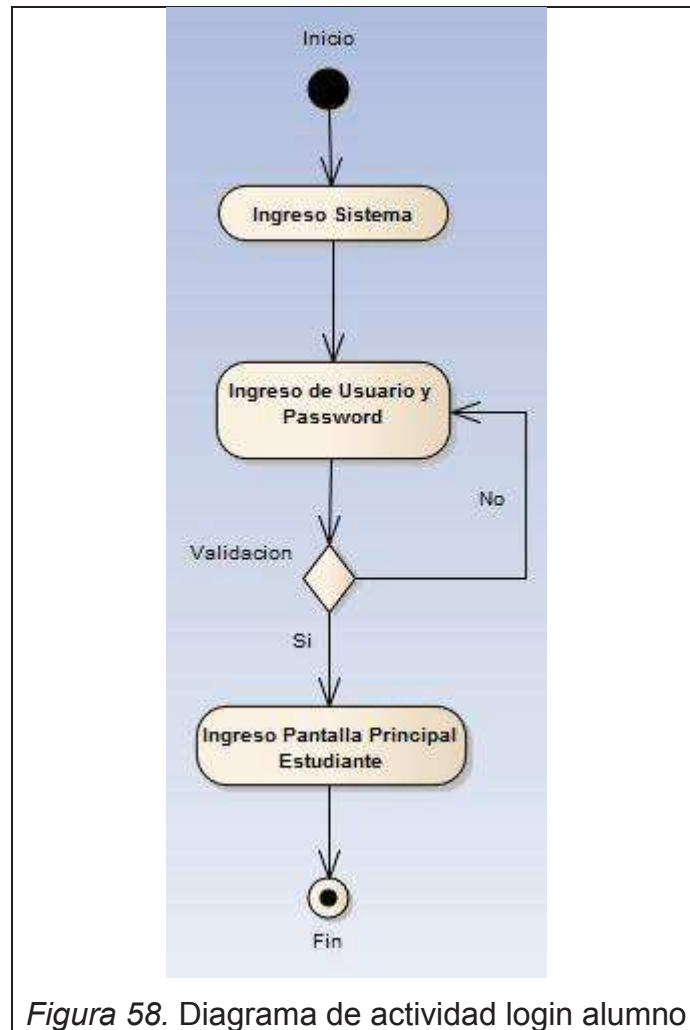


Figura 58. Diagrama de actividad login alumno

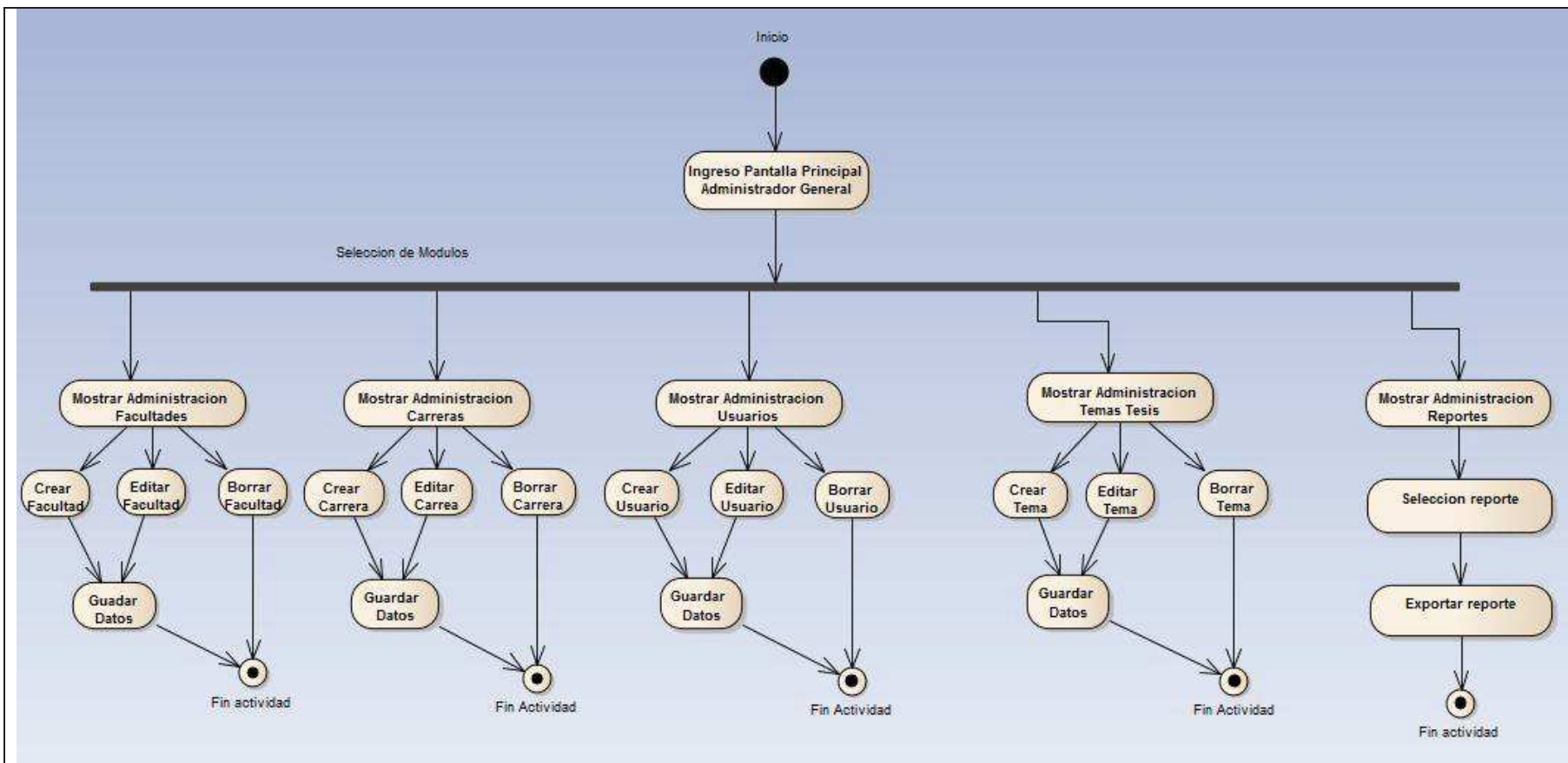


Figura 59. Diagrama de actividad Administrador general

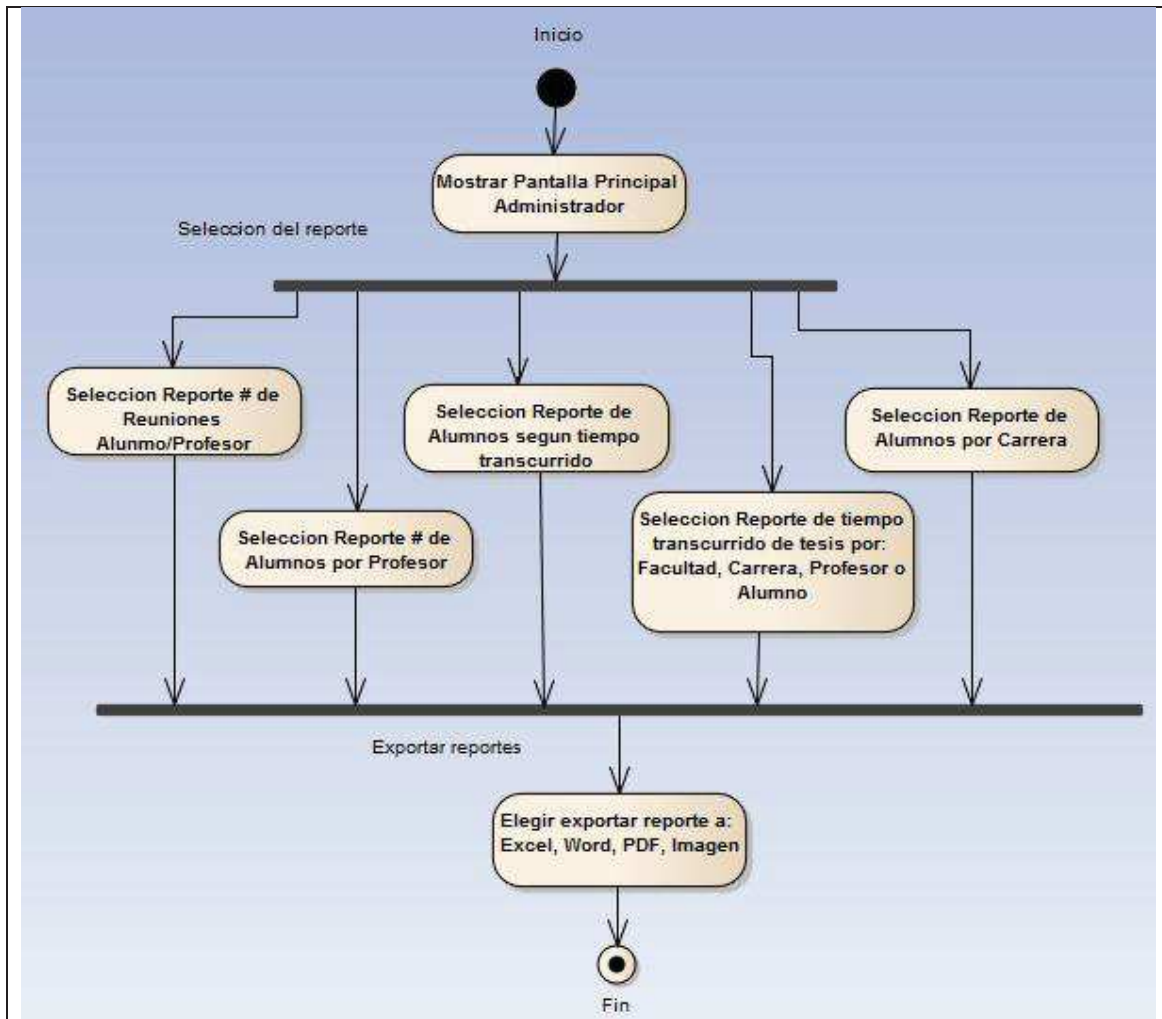


Figura 60. Diagrama de actividad administrador de reportes



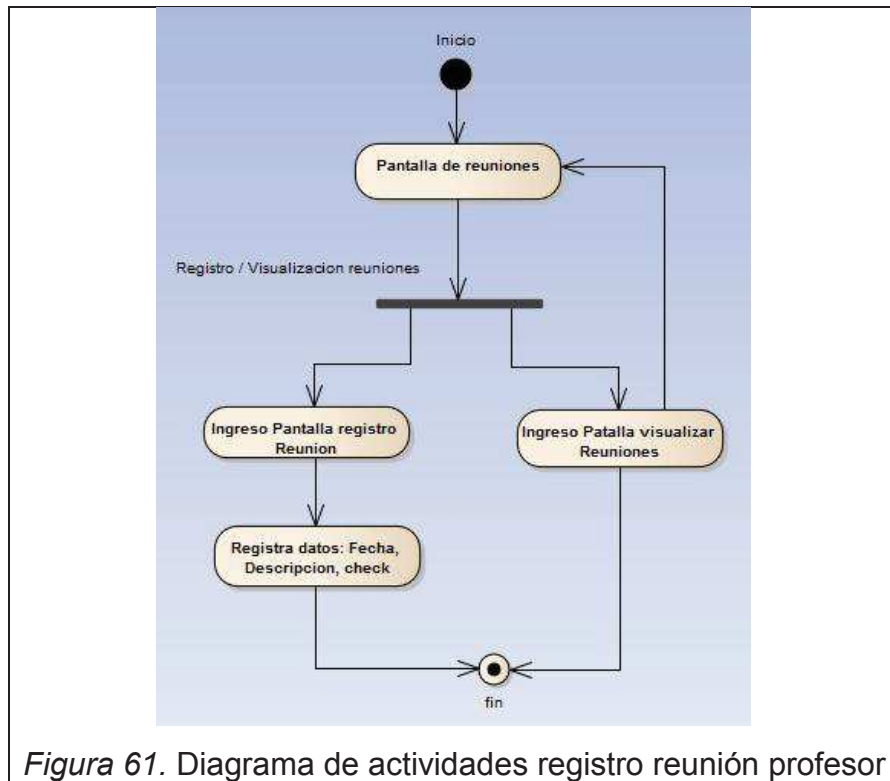


Figura 61. Diagrama de actividades registro reunión profesor

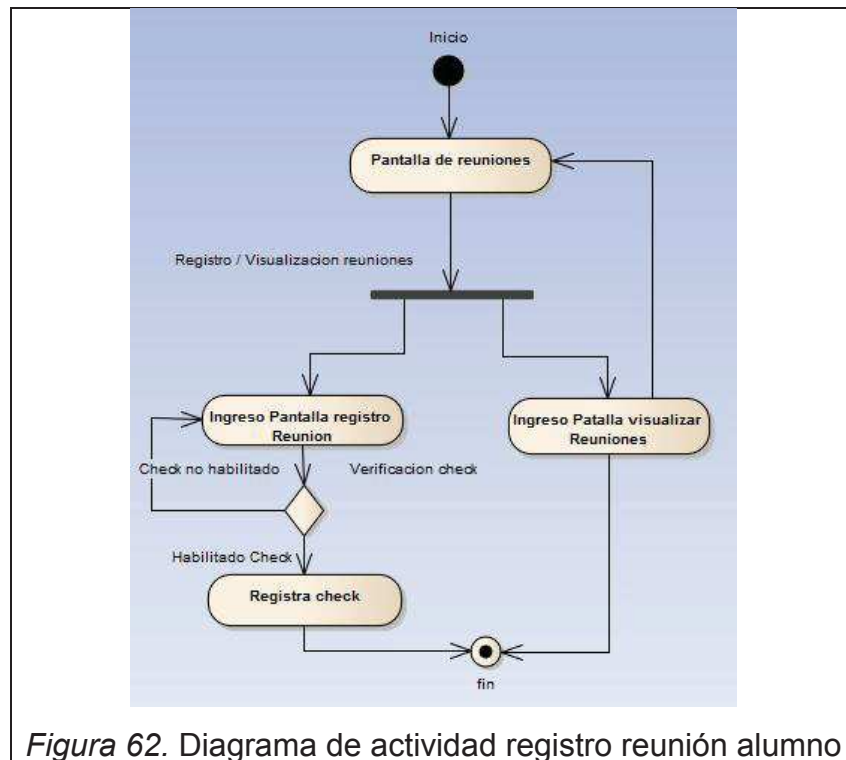


Figura 62. Diagrama de actividad registro reunión alumno

### 3.6. Diagrama de Clases.

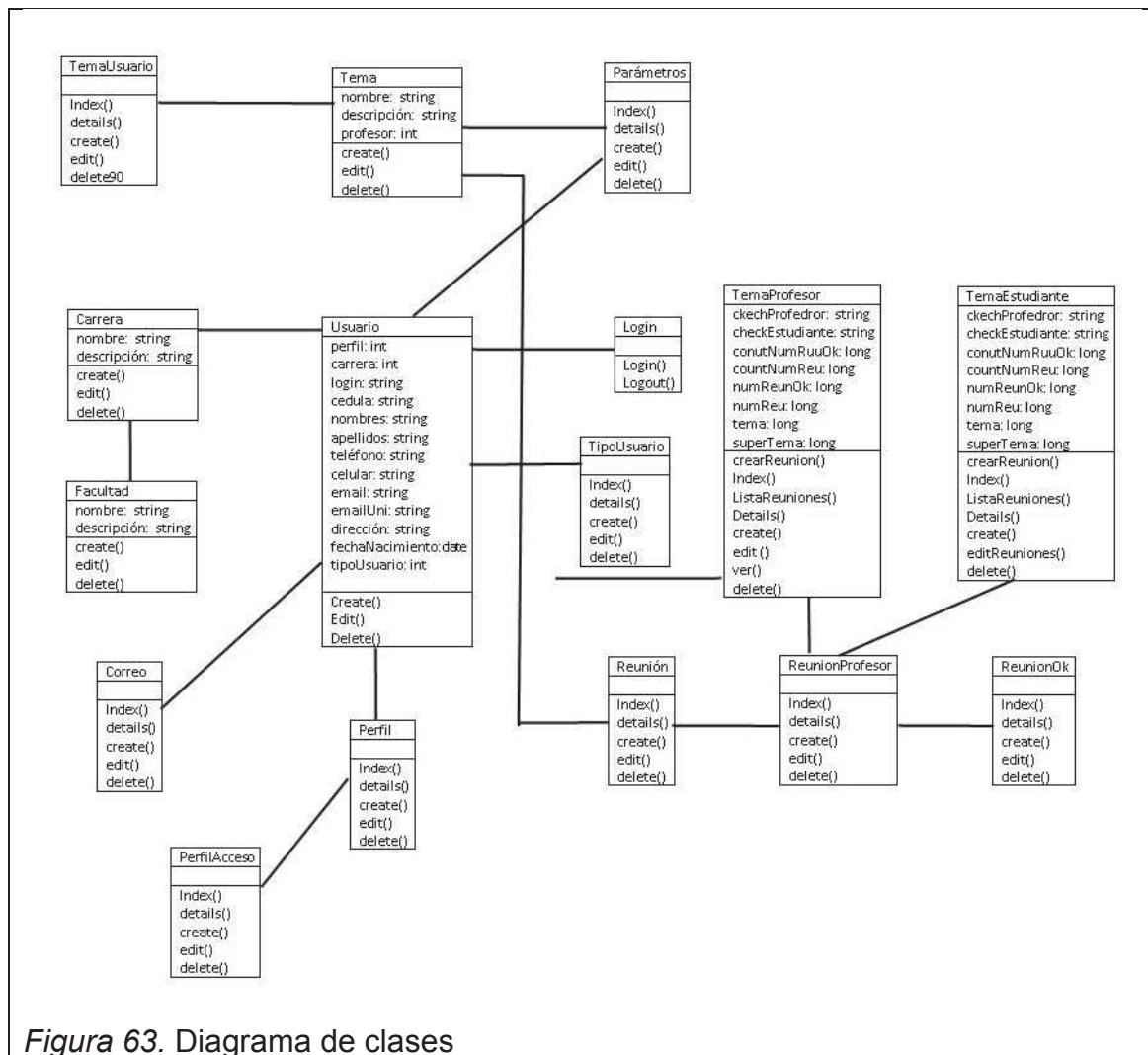


Figura 63. Diagrama de clases

### 3.7. Diagrama de Base de Datos.

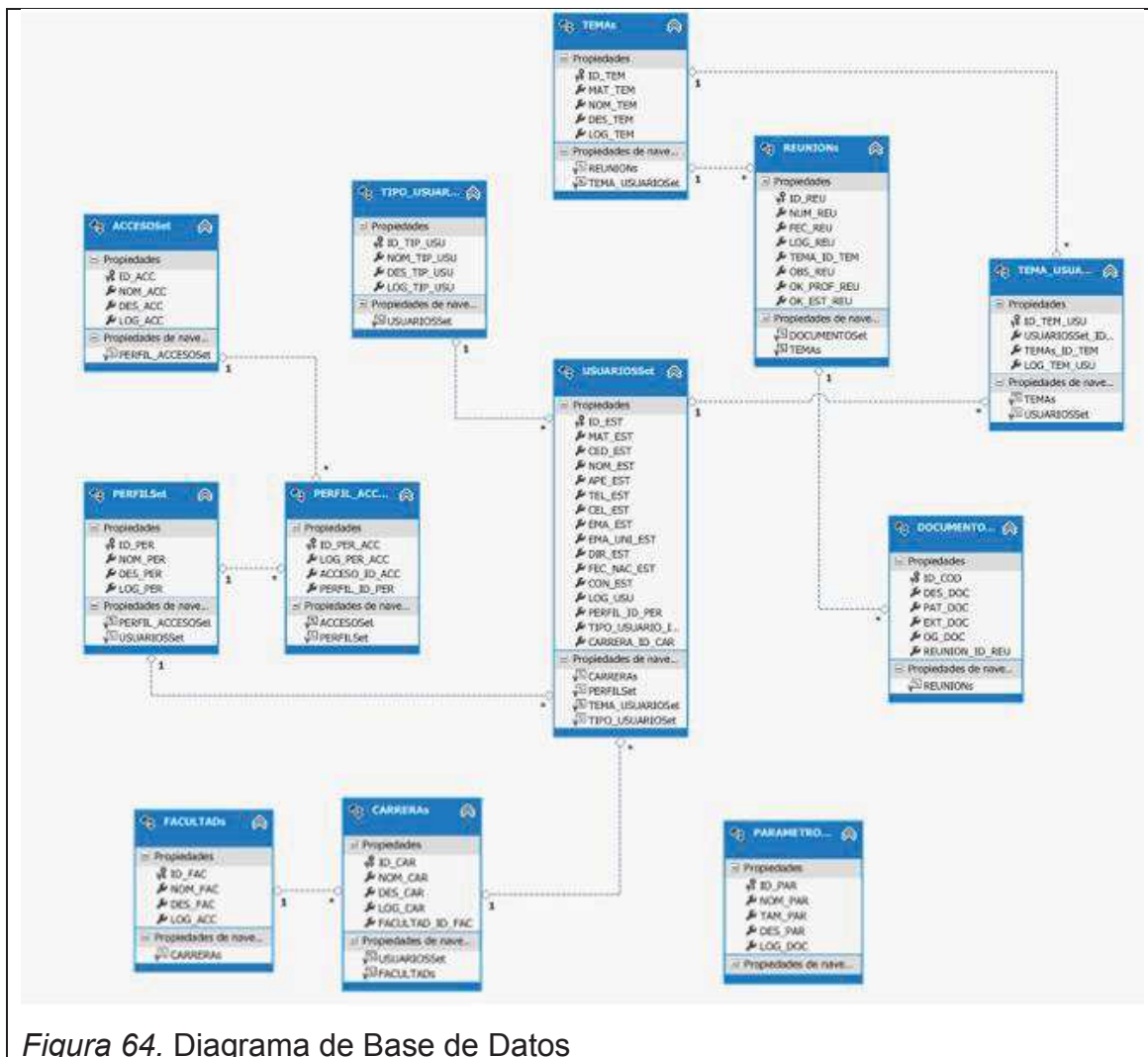


Figura 64. Diagrama de Base de Datos

## 4. Conclusiones y recomendaciones

### 4.1. Conclusiones

- Usar metodologías ágiles como Scrum permite llevar una guía de las diferentes actividades que envuelven al desarrollo de software, debido a la documentación generada en donde se da a conocer todo lo que se está haciendo y lo que se va a hacer a lo largo del proyecto.
- Un framework como .NetFramework unido al IDE Visual Studio permite al desarrollador la creación de código funcional, dejando que estos se encarguen de la mayoría de aspectos administrables del desarrollo.
- Los wireframes son de gran ayuda en el tiempo de desarrollo para dar una idea de la interfaz gráfica que se desea presentar a los usuarios obteniendo así de una manera rápida las vistas del sistema aun sin codificar.
- Con la utilización del patrón MVC se logra conseguir una manera ordenada de desarrollo separando la lógica del negocio de la lógica de programación, además de poder realizar cambios sin tener que afectar la integridad e independencia del sistema.
- Al realizar los diagramas de casos de uso de un sistema de manera correcta se genera una bitácora de cada uno de los flujos del sistema facilitando así el desarrollo.

## 4.2 Recomendaciones

- Al realizar sistemas para clientes particulares es recomendable tener en claro la documentación que se genera para cumplir las funcionalidades que desea el cliente de acuerdo con el alcance del sistema, de ser necesario realizar varias entrevistas para evitar cambios futuros que puedan generar un retraso en el correcto desarrollo.
- Es muy importante definir y tener claro desde el principio del diseño del sistema la base de datos con sus respectivas tablas y relaciones, más aun si es un sistema como este basado en una base de datos relacional en donde un cambio de último momento afectaría a todo el modelo.
- Realizar un análisis del proyecto de manera organizada y enfocados en los objetivos a cumplir, usando por ejemplo los diagramas UML con el fin de no tener retrasos o dificultades a futuro.
- Al realizar sistemas por primera vez o utilizar tecnologías nuevas para el desarrollador se debe tomar en cuenta un estudio previo de las diferentes opciones existentes y así determinar la más adecuada para el desarrollo y entendimiento para el desarrollador se trata.
- Se debería considerar también las nuevas tecnologías en un desarrollo a futuro de una nueva versión del sistema como por ejemplo confirmación de acceso a través de código QR o control de acceso mediante huella dactilar.
- Continuar con el desarrollo de los demás procesos que integran el todo del desarrollo de proyecto de titulación, y de esta manera tener ya cubierto el proceso total de desarrollo de proyectos de titulación de la UDLA de una manera sistematizada.

## REFERENCIAS

- aalbertovargasc (s.f.). Rational Unified Process. Recuperado el 29 de mayo del 2013 de <http://aalbertovargasc.files.wordpress.com/2014/03/metodopesadesrup1.pdf>
- Albaladejo, X. (s.f.). Qué es SCRUM. Recuperado el 01 de junio del 2013 de <http://www.proyectosagiles.org/que-es-scrum>
- alegsa.(s.f.). Definicion de SGBD. Recuperado el 16 de agosto del 2013 de <http://www.alegsa.com.ar/Dic/sghd.php>
- Arturi, F. (2012). Introduccion a ASP.NET MVC. Recuperado el 01 de septiembre del 2013 de <http://www.arturi.eu/introduccion-a-asp-net-mvc/>
- bernardolimachi (2012). METODOLOGIA RUP. Recuperado el 29 de mayo del 2013 de <http://www.slideshare.net/bernardolimachi/metodologia-rup-14288208>
- Canós, J., Letelier, P. y Penadés, M. (s.f.). Metodologias Agiles en el Desarrollo de Software. Recuperado el 17 de mayo del 2013 de [http://noqualityinside.com.ar/nqi/nqifiles/XP\\_Agil.pdf](http://noqualityinside.com.ar/nqi/nqifiles/XP_Agil.pdf)
- Clifton, M., Dunlap, J. (2003). What is SCRUM?. Recuperado el 4 de junio del 2013 de <http://www.codeproject.com/Articles/4798/What-is-SCRUM>
- clubdelsuran. (s.f.). Diagramas del UML. Recuperado el 15 de junio del 2013 de [http://www.clubdelsuran.com.ar/site/materiales/proyecto/diagramas\\_del\\_uml.pdf](http://www.clubdelsuran.com.ar/site/materiales/proyecto/diagramas_del_uml.pdf)
- Cuesta, P., Amores, P., Peris, S. (2009). Lenguaje Unificado de Modelado. Recuperado el 15 de junio del 2013 de <http://www.slideshare.net/e1da4/diagramas-uml>
- Garcia, D. (2013). INTRODUCCION A ASP.NET MVC4. Recuperado el 15 de noviembre del 2013 de <http://danielggarcia.wordpress.com/2013/11/11/introduccion-a-asp-net-mvc-4/comment-page-1/>
- Garat, P. (2012). Crystal Clear. Recuperado el 01 de junio del 2013 de <http://www.slideshare.net/PacoGarat/crystal-clear-11805930>
- González, H. (2012). Cuadro comparativo distintos SGBD. Recuperado el 15 de agosto del 2013 de <http://carloshdz-basedatos.blogspot.com/2012/08/taller-base-datos.html>

- Gutierrez, D. (2011). Metodos de Desarrollo de Software. Recuperado el 22 de mayo del 2013 de [http://www.codecompiling.net/files/slides/IS\\_clase\\_13\\_metodos\\_y\\_procesos.pdf](http://www.codecompiling.net/files/slides/IS_clase_13_metodos_y_procesos.pdf)
- Henao, C. (2013). Ejemplo Modelo Vista Controlador. Recuperado el 11 de agosto del 2013 de <http://codejavu.blogspot.com/2013/06/ejemplo-modelo-vista-controlador.html>
- Hernández, D. (s.f.). Introducción al UML. Recuperado el 16 de junio del 2013 de [http://www.magma.com.ni/~jorge/upoli\\_uml/refs/Introduccion\\_UML.pdf](http://www.magma.com.ni/~jorge/upoli_uml/refs/Introduccion_UML.pdf)
- Herrarte Sanchez, P. (2013). Patron MVC. Modelo Vista Controlador. Recuperado el 15 de agosto del 2013 de <http://www.devjoker.com/contenidos/catss/525/Patron-MVC-Modelo-Vista-Controlador.aspx>
- Intuitive (s.f.). Microsoft Solution Framework. Recuperado el 29 de mayo del 2013 de <http://www.intuitive-bi.com.ar/nuestra-metodologia.html>
- lab. (s.f.). Patron de arquitectura Modelo Vista Controlador (MVC). Recuperado el 07 de agosto del 2013 de <http://www.lab.inf.uc3m.es/~a0080802/RAI/mvc.html>
- Linares, F., Velasco, J. Querales, E., Graterol, R., Figueroa, P., Noruega, G., Ponte, J. (2012). METODOLOGIA XP. Recuperado el 31 de mayo del 2013 de <http://ingsoftware072301.obolog.es/metodologia-xp-2012877>
- Microsoft. (s.f.). Información general acerca de .NET Framework. Recuperado el 19 de julio del 2013 de [http://msdn.microsoft.com/es-es/library/zw4w595w\(v=vs.110\).aspx](http://msdn.microsoft.com/es-es/library/zw4w595w(v=vs.110).aspx)
- Microsoft. (s.f.). .NET Framework 4.5. Recuperado el 18 de julio del 2013 de [http://msdn.microsoft.com/es-es/library/w0x726c2\(v=vs.110\).aspx](http://msdn.microsoft.com/es-es/library/w0x726c2(v=vs.110).aspx)
- Microsoft. (2006). Tutoriales de Reporting Services. Recuperado el 02 de septiembre del 2013 de [http://technet.microsoft.com/es-es/library/ms170246\(v=sql.90\).aspx](http://technet.microsoft.com/es-es/library/ms170246(v=sql.90).aspx)
- Orgaz González, A., Sánchez Palomino, J. (2013). Metodología SCRUM. Recuperado el 03 de junio del 2013 de <http://www.slideshare.net/despenao/metodologa-agile-scrum>
- Palacio, J. (2006). El modelo Scrum. Recuperado el 03 de junio del 2013 de [http://www.navegapolis.net/files/s/NST-010\\_01.pdf](http://www.navegapolis.net/files/s/NST-010_01.pdf)

- Quintero, G. (2013). MSF-Microsoft Solution Framework. Recuperado el 28 de mayo del 2013 de <http://prezi.com/jptdohl6v5nk/msf-microsoft-solution-framework/>
- Quiñones Moreno, L. (2013). Empezando con MC4 PARTE 1. Recuperado el 13 de septiembre del 2013 de <http://leonardoquinonesm.wordpress.com/2013/05/28/empezando-con-mvc-4-parte-1/>
- Romero, H. (2012). Metodologías de Desarrollo. Recuperado el 15 de mayo del 2013 de <http://www.slideshare.net/MeneRomero/metodologias-de-desarrollo>
- Ruiz Suarez, H. (2012). Tecnología Innovadora RUP. Recuperado el 30 de mayo del 2013 de <http://tecnologiarpup.blogspot.com/2012/11/cuales-son-las-ventajas-y-desventajas.html>
- Sanchez Ubaldo, F. (2012). Creación y Despliegue de la Aplicación ASP.NET en MVC 4. Recuperado el 01 de septiembre del 2013 de <http://felixthecatsos.blogspot.com/2012/12/creacion-y-despliegue-de-la-aplicacion.html>
- Shemmer, NS. (2013). ASP.NET MVC 4 - Part [1] – Introduction. Recuperado el 02 de septiembre del 2013 de <http://www.codeproject.com/Articles/470107/ASP-NET-MVC-Part-Introduction>
- SOFTENG SOFTWARE ENGINEERS (s.f.). Metodología Scrum. Recuperado el 01 de junio del 2013 de <http://www.softeng.es/es-es/empresa/metodologias-de-trabajo/metodologia-scrum.html>
- Ticona, S. (s.f.). METODOLOGIAS TRADICIONALES, METODOLOGIAS AGILES, METODOLOGIAS PARA JUEGOS, METODOLOGIAS EDUCATIVAS Y METODOLOGIAS PARA APLICACIONES MOVILES. Recuperado el 28 de mayo del 2013 de <http://tallerinf281.wikispaces.com/file/view/METODOLOG%C3%8DAS+TRADICIONALES.pdf>
- Trigas Gallego, M. (s.f.). Metodología Scrum. Recuperado el 03 de junio del 2013 de [http://openaccess.uoc.edu/webapps/o2/bitstream/10609/17885/1/mtrigasTF\\_C0612memoria.pdf](http://openaccess.uoc.edu/webapps/o2/bitstream/10609/17885/1/mtrigasTF_C0612memoria.pdf)
- um. (s.f.). Sistemas de Gestión de Bases de datos y SIG. Recuperado el 15 de agosto del 2013 de [http://www.um.es/geograf/sigmur/sigpdf/temario\\_9.pdf](http://www.um.es/geograf/sigmur/sigpdf/temario_9.pdf)



Universidad de las Americas. (2013). Pregrados modalidad diurna. Recuperado el 30 de abril de <http://www.udla.edu.ec/carreras/programas-academicos/pregrados-diurna/>

Universidad de las Americas. (2013). Pregrados modalidad diurna. Recuperado el 30 de abril de <http://www.udla.edu.ec/carreras/programas-academicos/tecnologias/>

Universidad Union Bolivariana (s.f.). Ingenieria de Software. Recuperado el 01 de junio del 2013 de [http://ingenieriadesoftware.mex.tl/52753\\_XP---Extreme-Programing.html](http://ingenieriadesoftware.mex.tl/52753_XP---Extreme-Programing.html)

Uñoja, R. H. (2012). METODOLOGIAS DE DESARROLLO DE SOFTWARE TRADICIONALES VS AGILES. Recuperado el 13 de mayo del 2013 de <http://masteringenieriasoft.blogspot.com/2012/04/metodologias-de-desarrollo-de-software.html>

Vargas, D. (2012). METODOLOGIAS DE DESARROLLO DE SOFTWARE TADICIONALES VERSUS METODOLOGIAS AGILES. Recuperado el 31 de de mayo del 2013 de <http://danithzavargas.blogspot.com/2012/04/1.html>

Villamarín, A. (2013). Modelo Vista Controlador adaptado a la Web. Recuperado el 08 de agosto del 2013 de <http://ant.onio.org/2013/04/10/modelo-vista-controlador-adaptado-a-la-web.html>

W3school.(s.f.). CSS Tutorial. Recuperado el 28 de agosto del 2013 de <http://www.w3schools.com/css/>

Yago, P. (2012). ASP MVC 4 Web API. Primeros pasos. Recuperado el 15 de septiembre del 2013 de [http://www.yagoperez.com/asp\\_mvc\\_4\\_web\\_api\\_primeros\\_pasos/](http://www.yagoperez.com/asp_mvc_4_web_api_primeros_pasos/)

## **ANEXOS**

## **ANEXO 1: PRUEBAS Y REUNIÓN**

Se procedió a realizar varias pruebas para establecer el buen funcionamiento del sistema, de las que surgieron errores de diferentes ámbitos como:

- Asignación de más de un alumno a un tema de Trabajo de Titulación
- Carga de más de un tema de Trabajo de Titulación a un profesor.
- Registro de check por parte del (los) alumno(s).
- Edición de parámetros.

Los cuales se procedieron a corregir, luego se realizaron nuevas pruebas para comprobar el éxito de las soluciones.

Además también se mantuvo una reunión con Pamela Espinoza y Estalin Armijos como delegados de secretaria académica de la Universidad de las Américas para probar la funcionalidad del software, de esta surgieron valiosos comentarios de modificación de algunos aspectos así como también la posibilidad de continuar con el desarrollo ya sea mejorando o aumentando ciertas características del sistema debido a que esta es recién una primera versión.

## ANEXO 2: WIREFRAMES

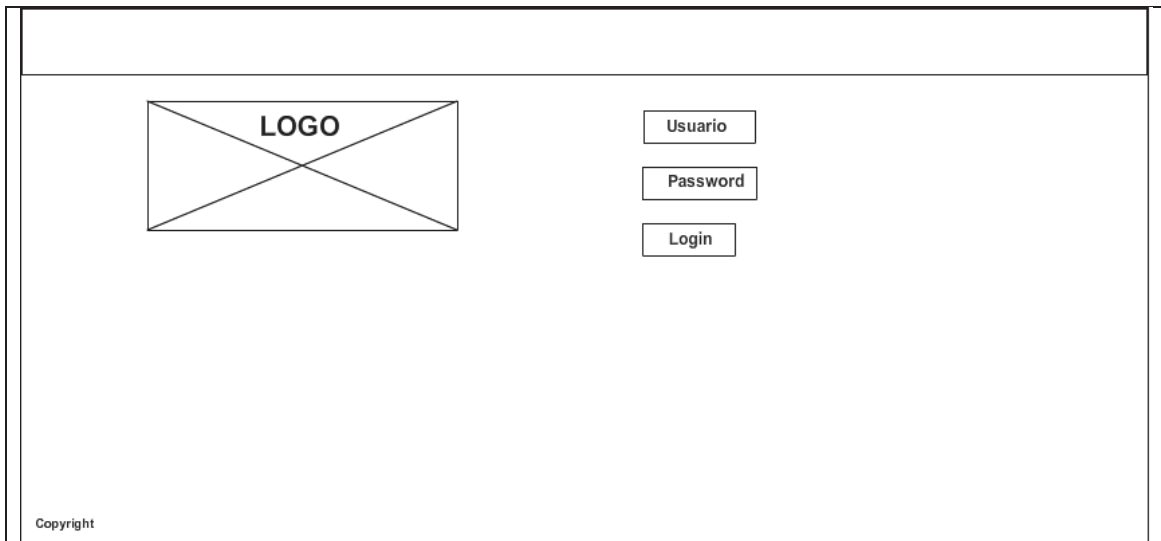


Figura 65. Pantalla para entrar al sistema

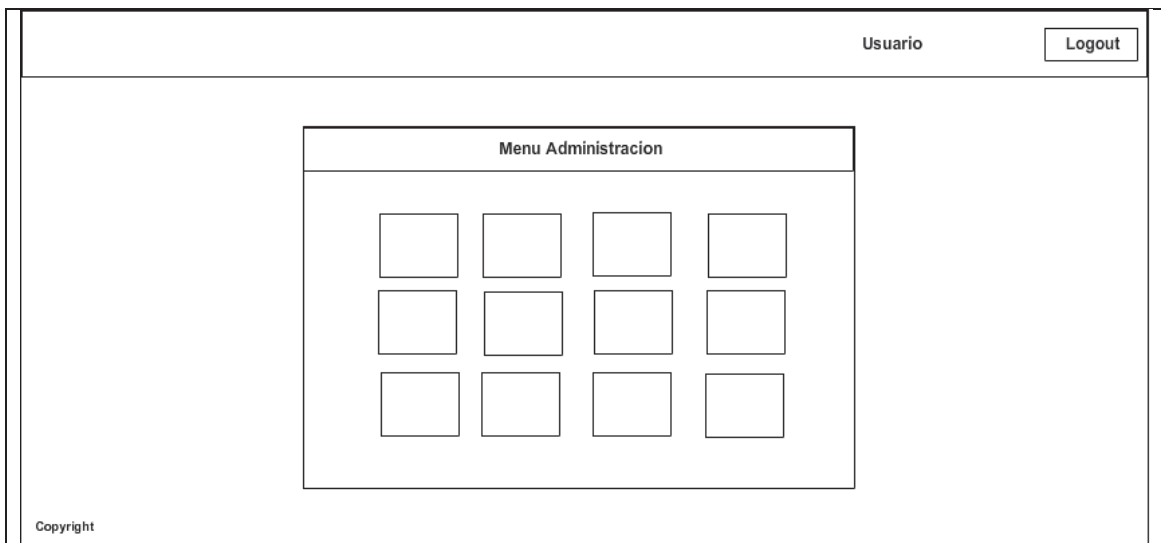
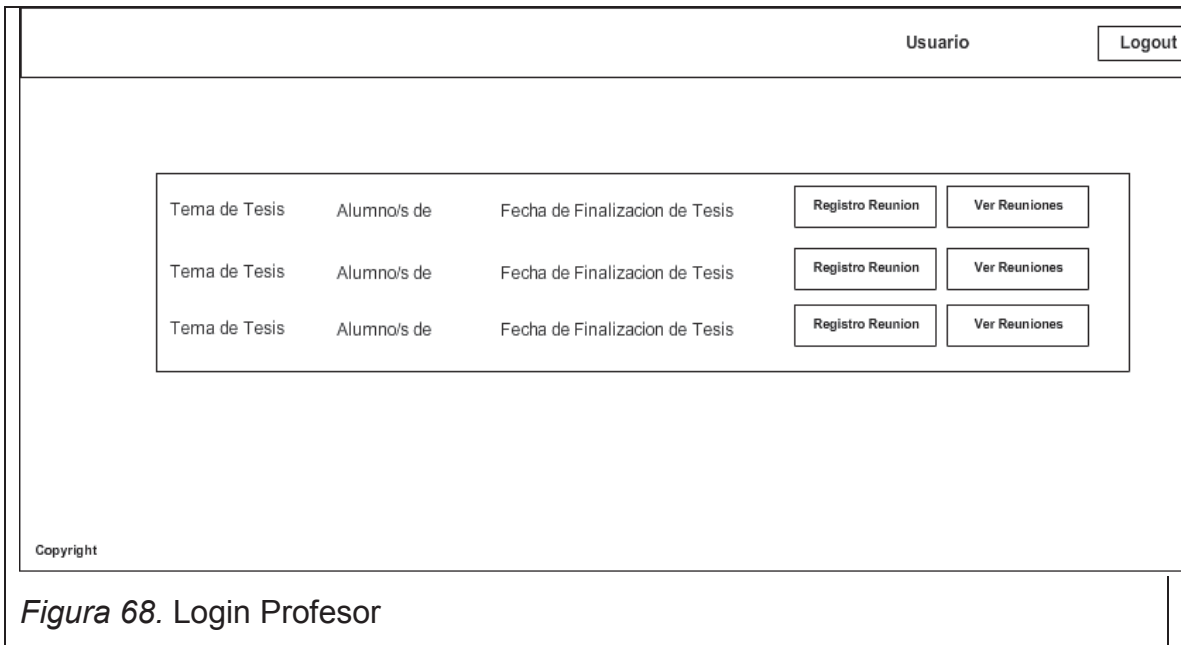
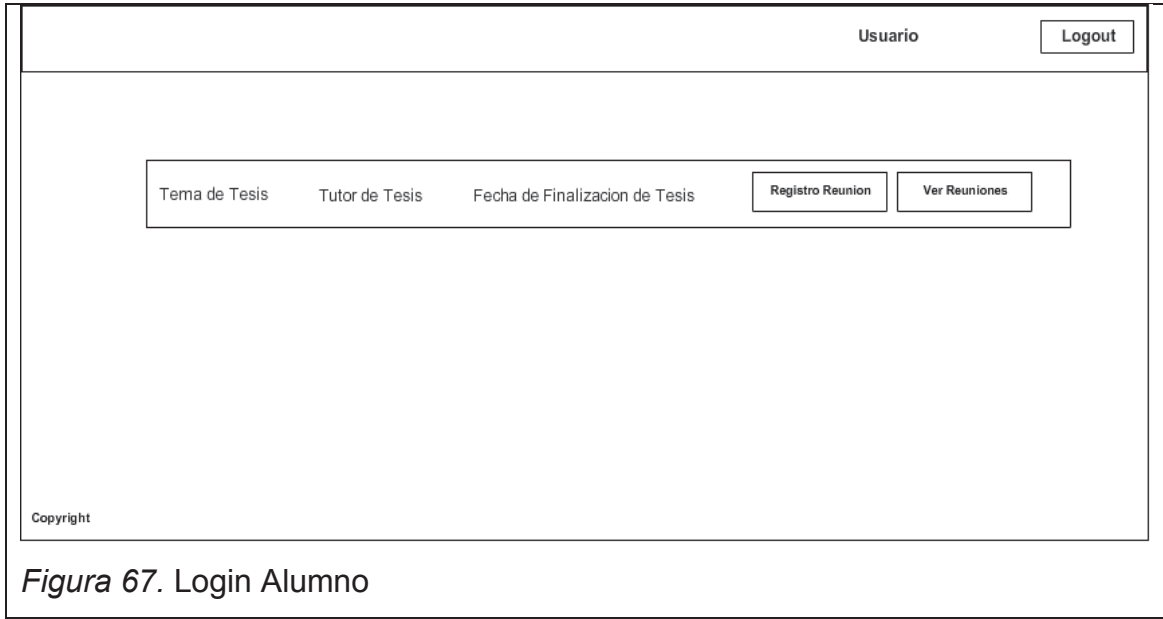


Figura 66. Login Admin



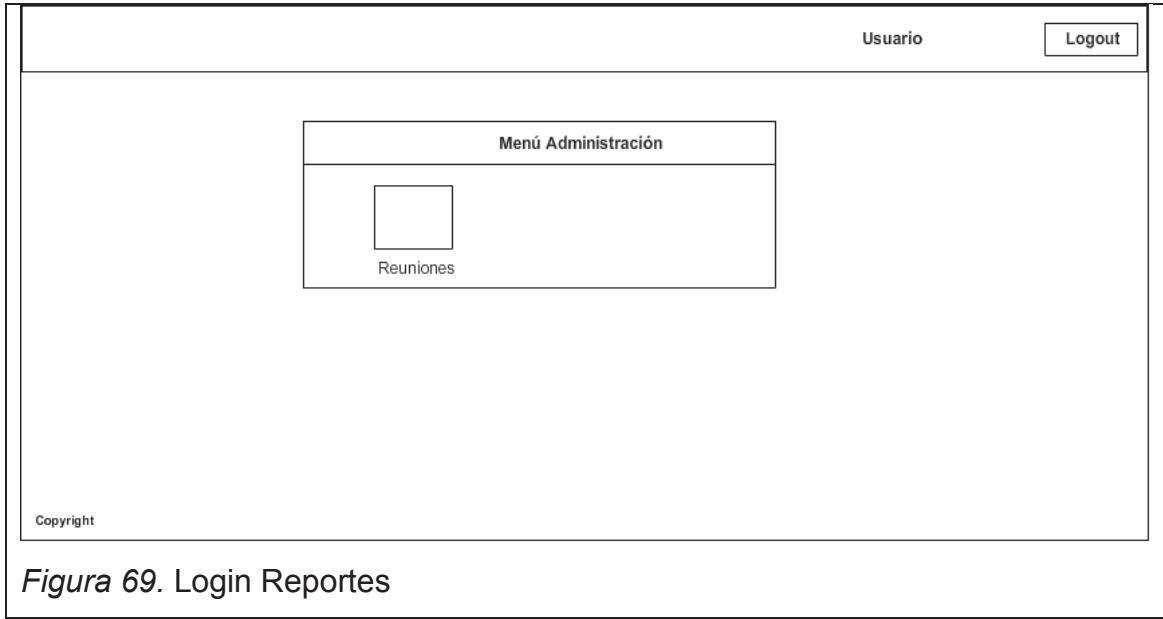


Figura 69. Login Reportes

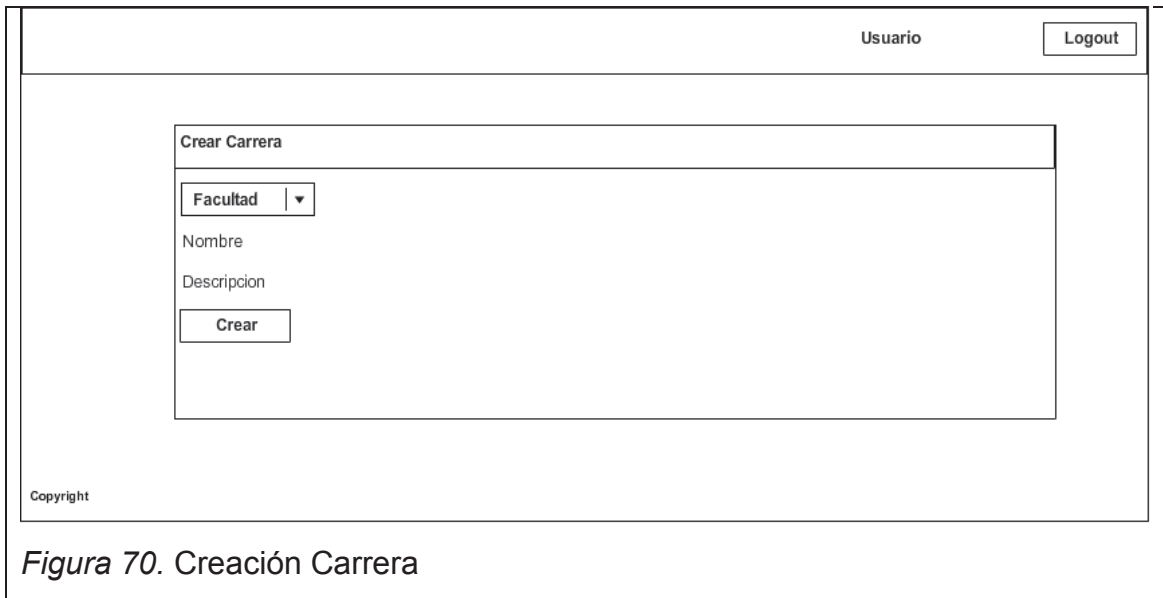


Figura 70. Creación Carrera

Usuario Logout

**Crear Facultad**

Nombre

Descripción

Copyright

*Figura 71. Creación Facultad*

Usuario Logout

**Crear Tema**

Nombre

Descripción

Fecha de Inicio

▼

Copyright

*Figura 72. Creación Tema de Proyecto de Titulación*

Usuario Logout

**Crear Usuario**

Tipo Usuario ▾

Carrera ▾

Login

Numero de Cedula

Nombres

Apellidos

Telefono casa

Celular

Email personal

Email Universidad

Direccion

Fecha de Nacimiento

Perfil ▾

Crear

Copyright

Figura 73. Creación de Usuarios

Usuario Logout

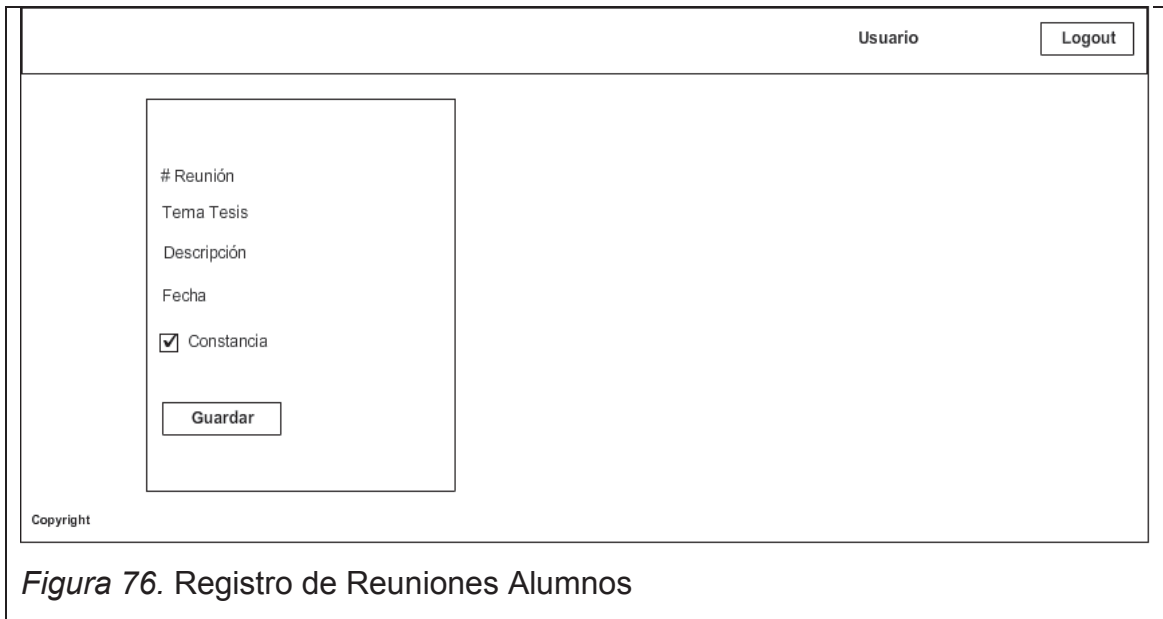
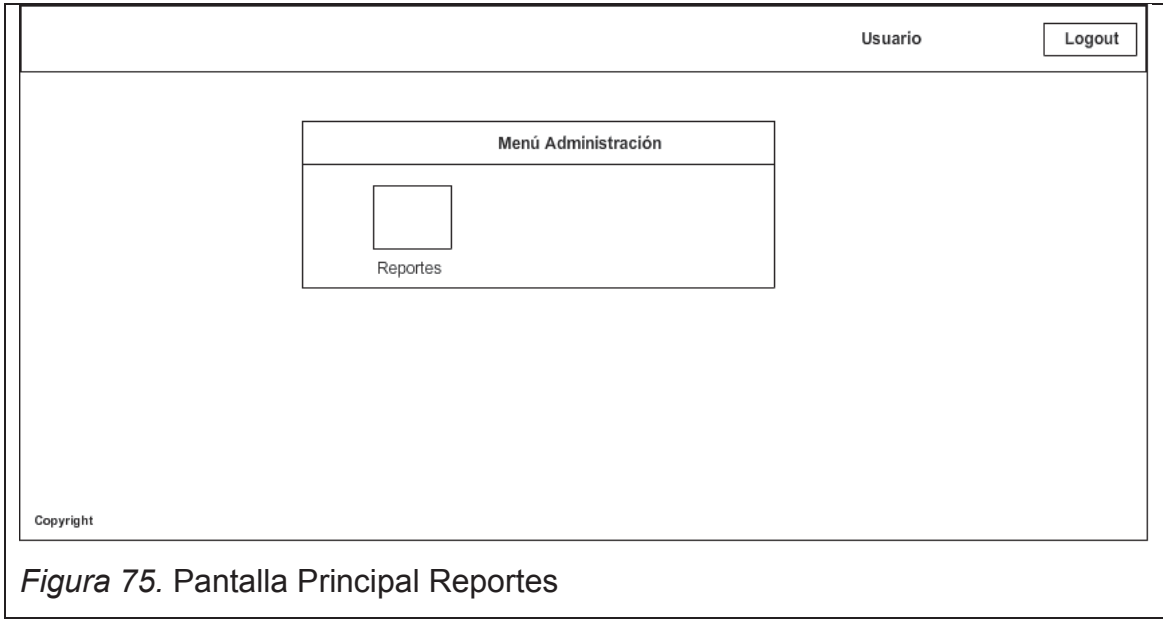
**Lista Reuniones**

| # Reunion | Fecha de reunion | Tema | Observacion | Ok Profesor | Estudiante | Ok Estudiante | Imagen |
|-----------|------------------|------|-------------|-------------|------------|---------------|--------|
|-----------|------------------|------|-------------|-------------|------------|---------------|--------|

Copyright

Figura 74. Listar Reuniones Profesor





Usuario Logout

Estudiante 1  
Estudiante 2

Numero de reuniones validas: ?

# Reunión

Tema Tesis

Descripción

Fecha

Constancia

.....

Copyright

*Figura 77. Registro de Reuniones Profesores*

Usuario Logout

Unir Tema a estudiante

Tema ▾

Estudiante ▾

Copyright

*Figura 78. Ingresar Tema Usuario*

## ANEXO 3: MANUAL DE INSTALACIÓN DE HERRAMIENTAS

En el siguiente manual se explica detalladamente la manera de instalación de las herramientas requeridas para que el sistema funcione correctamente, así como también se detalla los requerimientos de hardware.

### Requisitos SQL Server 2008

- .NET Framework 3.5 SP1 o .NET Framework 4.0 recomendado la versión 4.0.
- SQL Server Native Client  
Microsoft Windows Installer 4.5
- Sistema Operativo:

Windows Server 2003 x64

Windows Server 2003 SP2 de 64 bits x64 Datacenter

Windows Server 2003 SP2 de 64 bits x64 Enterprise

Windows Server 2003 SP2 de 64 bits x64 Standard

Windows Server 2003 R2 SP2 de 64 bits x64 Datacenter

Windows Server 2003 R2 SP2 de 64 bits x64 Enterprise

Windows Server 2003 R2 SP2 de 64 bits x64 Standard

Windows Vista SP2 Ultimate x64

Windows Vista SP2 Home Premium x64

Windows Vista SP2 Home Basic x64

Windows Vista SP2 Enterprise x64

Windows Vista SP2 Business x64

Windows Server 2008 SP2 de 64 bits x64 Datacenter

Windows Server 2008 SP2 de 64 bits x64 Datacenter sin Hyper-V

Windows Server 2008 SP2 de 64 bits x64 Enterprise

Windows Server 2008 SP2 de 64 bits x64 Enterprise sin Hyper-V

Windows Server 2008 SP2 de 64 bits x64 Standard

Windows Server 2008 SP2 de 64 bits x64 Standard sin Hyper-V

Windows Server 2008 SP2 de 64 bits x64 Web

Windows Server 2008 SP2 de 64 bits x64 Foundation Server

Windows 7 x64 Ultimate

Windows 7 x64 Home Premium

Windows 7 x64 Home Basic

Windows 7 x64 Enterprise

Windows 7 x64 Professional

Windows Server 2008 R2 de 64 bits x64 Datacenter

Windows Server 2008 R2 de 64 bits x64 Enterprise

Windows Server 2008 R2 de 64 bits x64 Standard

Windows Server 2008 R2 de 64 bits x64 Web

Windows Server 2008 R2 x64 para Windows Essential Server Solutions

Windows Server 2008 R2 de 64 bits x64 Foundation Server

- RAM:

- Mínimo: 512 MB
- Recomendado: 1 GB
- Máximo: 1 GB para Motor de base de datos.
- Procesador:
  - Tipo de procesador:
    - Mínimo: AMD Opteron, AMD Athlon 64, Intel Xeon compatible con Intel EM64T, Intel Pentium IV compatible con EM64T
  - Velocidad del procesador:
    - Mínimo: 1,4 GHz
    - Recomendado: 2,0 GHz o más

### **Requisitos Visual Studio 2012**

- Sistemas operativos compatibles
  - Windows 7 SP1 (x86 y x64)
  - Windows 8 (x86 y x64)
  - Windows 8.1 (x86 y x64)
  - Windows Server 2008 R2 SP1 (x64)
  - Windows Server 2012 (x64)
  - Windows Server 2012 R2 (x64)
- Internet Explorer 10
- Requisitos de hardware
  - Procesador a 1,6 GHz o superior
  - 1 GB de RAM (1,5 GB si se ejecuta en una máquina virtual)
  - 4 GB de espacio disponible en el disco duro
  - Disco duro de 5400 RPM
  - Tarjeta de vídeo compatible con DirectX 9 con resolución de pantalla de 1024 x 768 o más.

### **Instalación SQL Server 2008 R2 Express**

1. Dependiendo del idioma y arquitectura del Sistema Operativo (Inglés – Español, x86 o x64) ejecutar el instalador correspondiente al mismo:
  - SQLEXPRTWT\_x64\_ESN.exe – OS Español 64 Bits
  - SQLEXPRTWT\_x86\_ESN.exe – OS Español 32 Bits
  - SQLEXPRTWT\_x64\_ENU.exe – OS Español 64 Bits
  - SQLEXPRTWT\_x86\_ENU.exe – OS Español 32 Bits



Figura 79. Inicio de programa de instalación

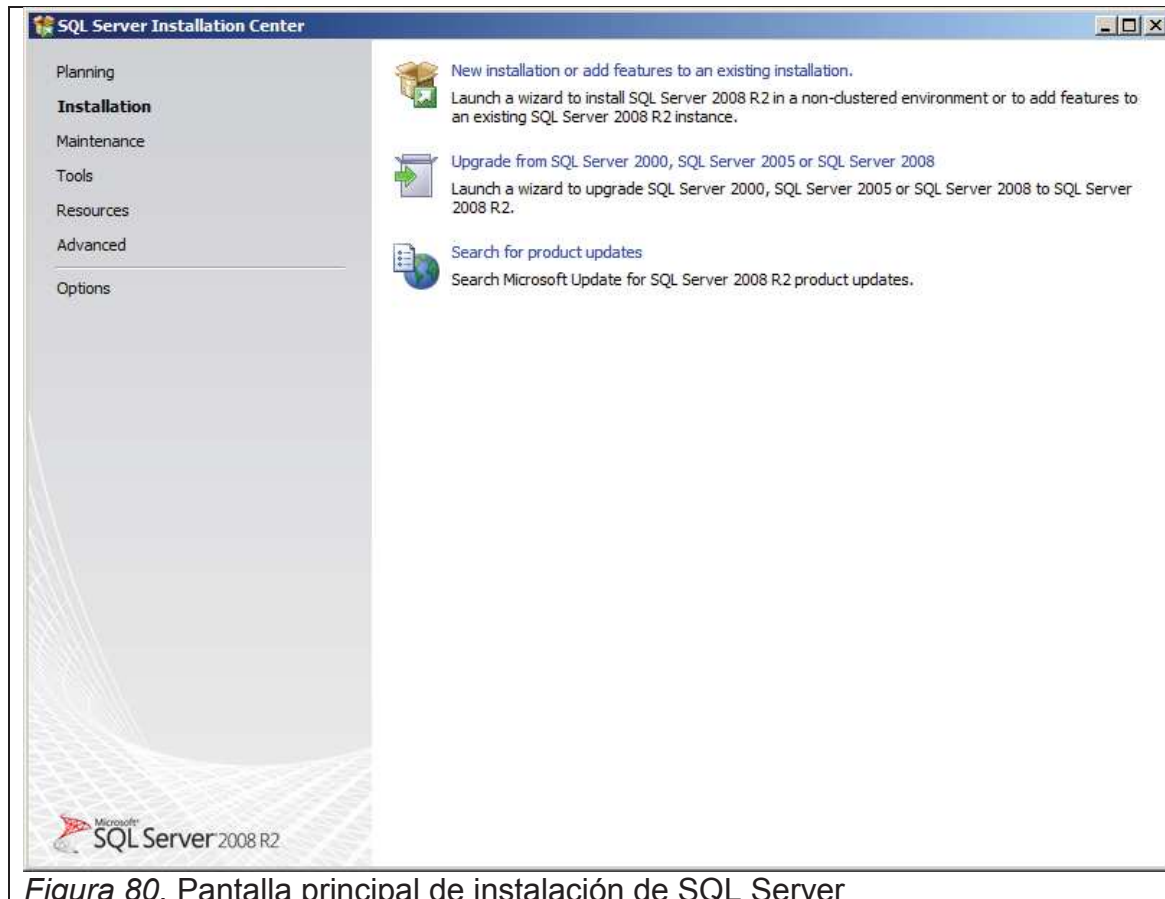


Figura 80. Pantalla principal de instalación de SQL Server

2. Click en **New installation or add features an existing installation**



Figura 81. Nueva instalación

3. Marcamos **I accept the license terms.** Y damos click en **Next.**

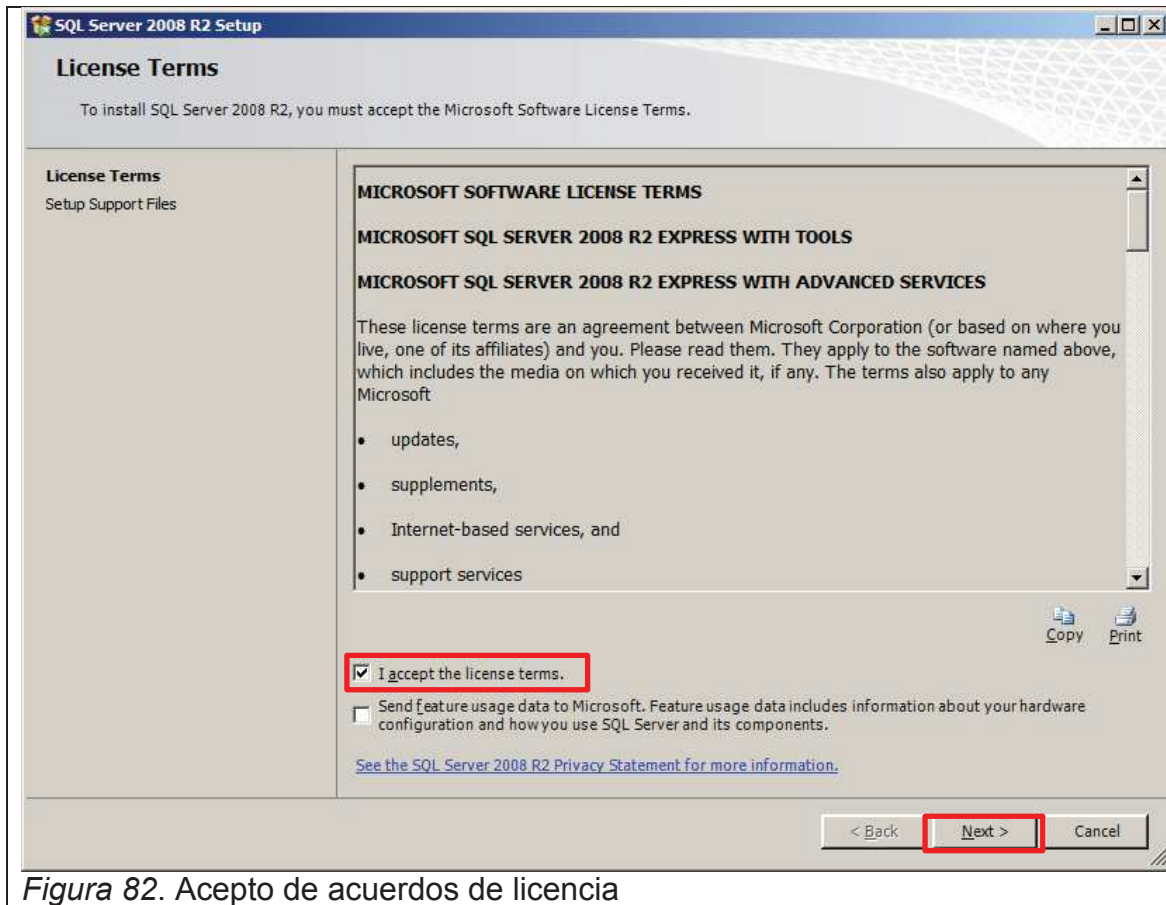


Figura 82. Acepto de acuerdos de licencia

4. En caso de pedir que se reinicie el equipo hacerlo al final de la instalación y damos click en **OK.**

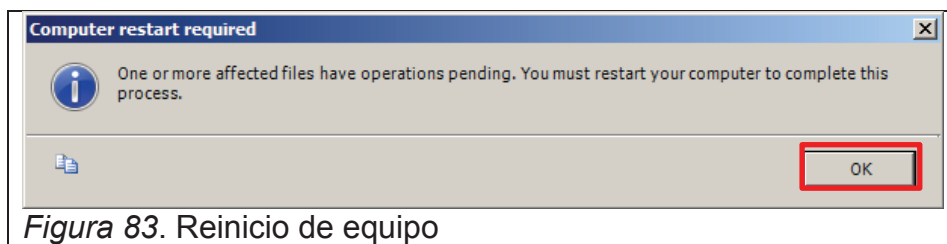


Figura 83. Reinicio de equipo

5. Dejamos marcado lo que viene por defecto y damos **Next**

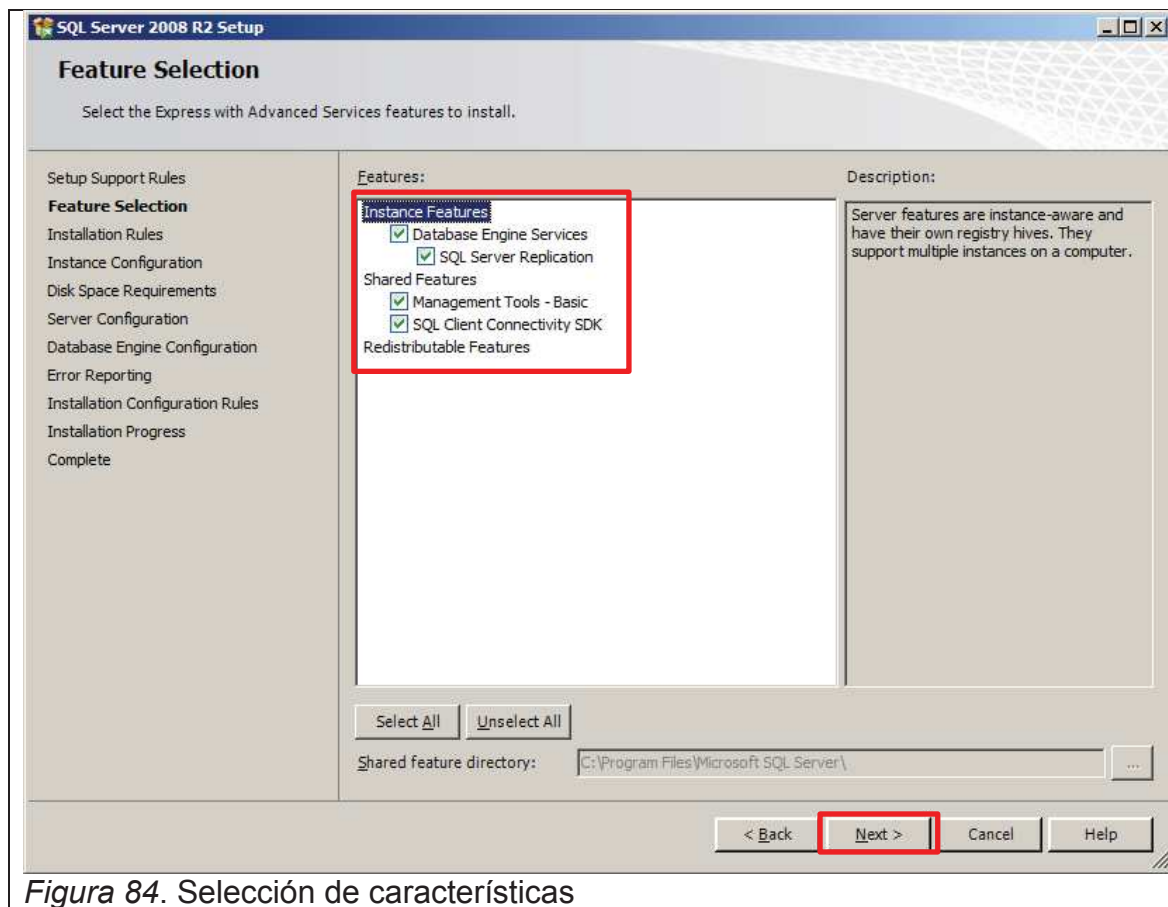


Figura 84. Selección de características



- Colocamos un nombre a la instancia de la base de datos, en este caso colocaremos TESIS y damos click en **Next**

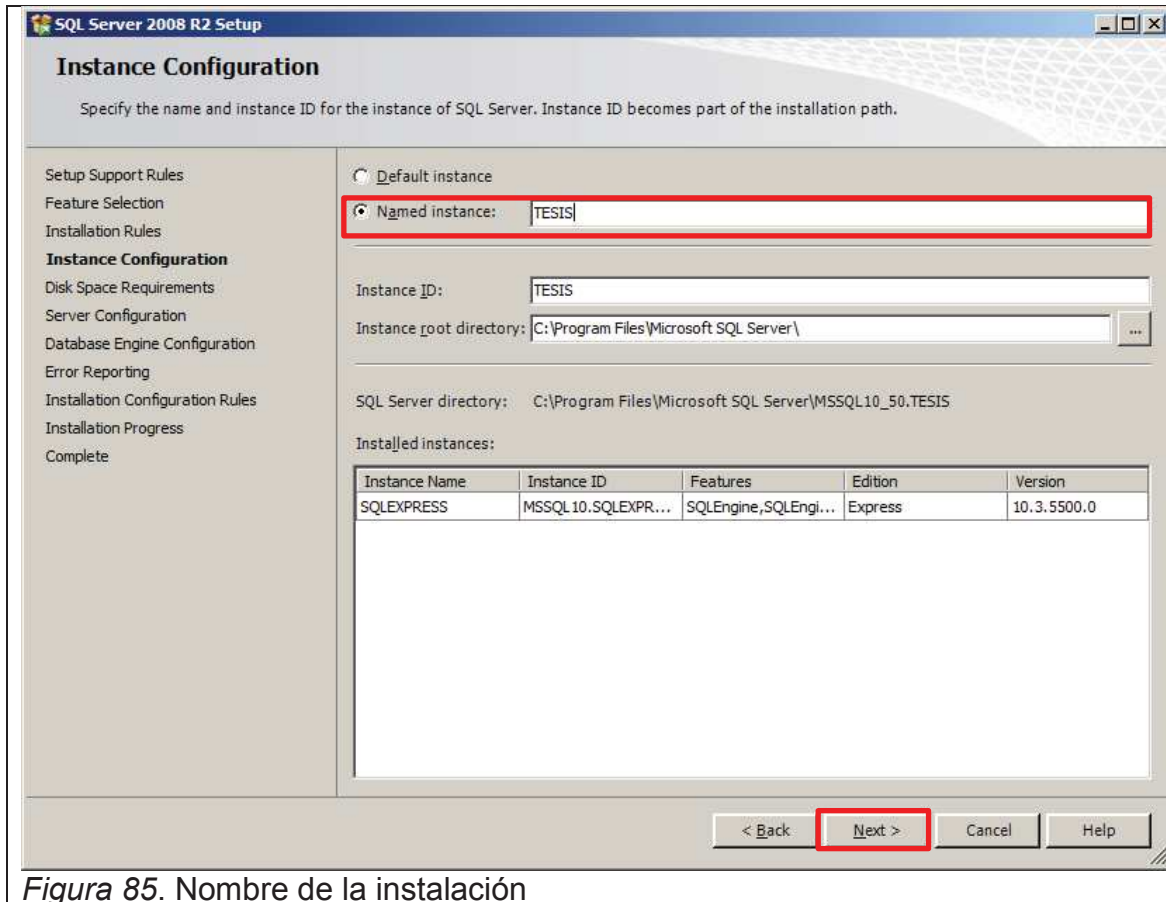


Figura 85. Nombre de la instalación

7. Cambiar el Account Name de SQL Server Database Engine a **NT AUTHORITY\SYSTEM** y cambiar **Startup Type** a **Automatic**, en **SQL Server Browser** cambiar **Startup Type** a **Automatic**, dar click en **Next**.

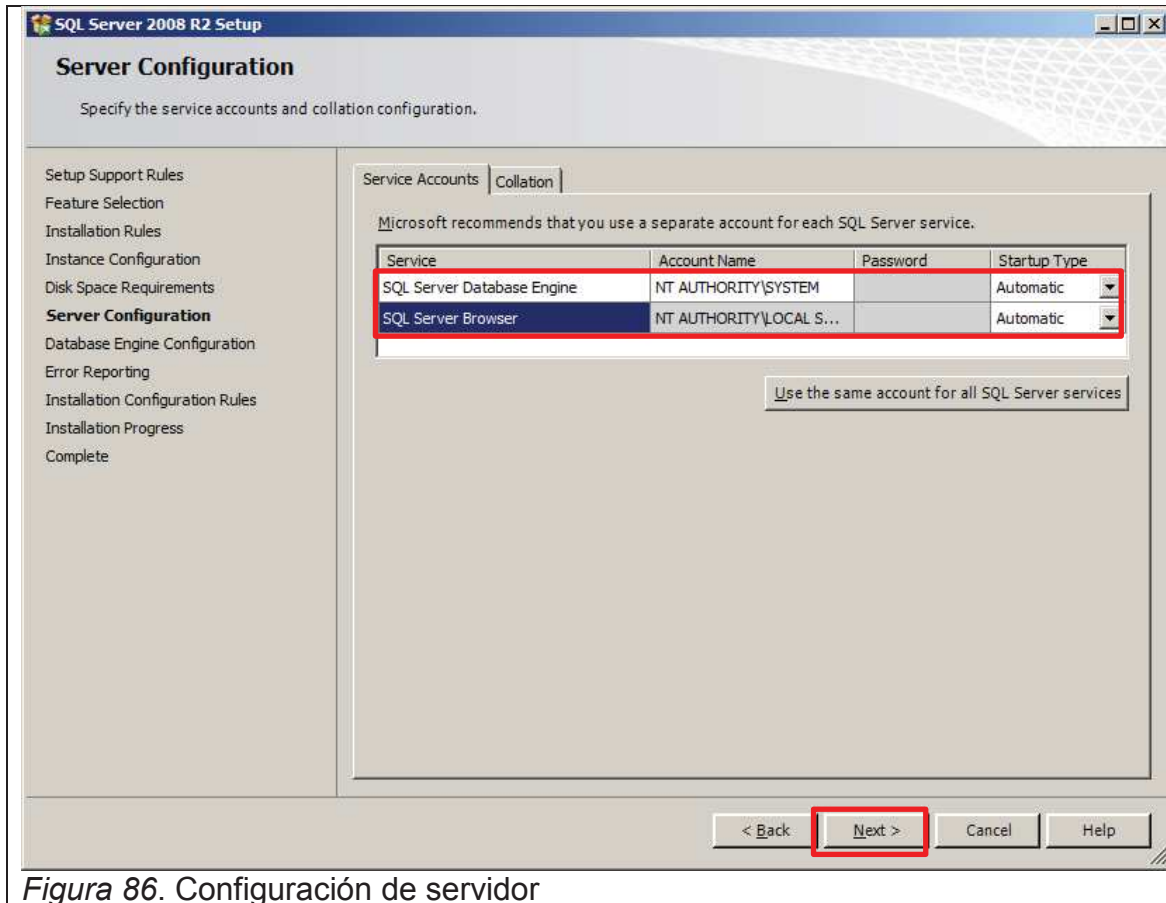


Figura 86. Configuración de servidor

8. Dejar todo por defecto y dar click en **Next**.

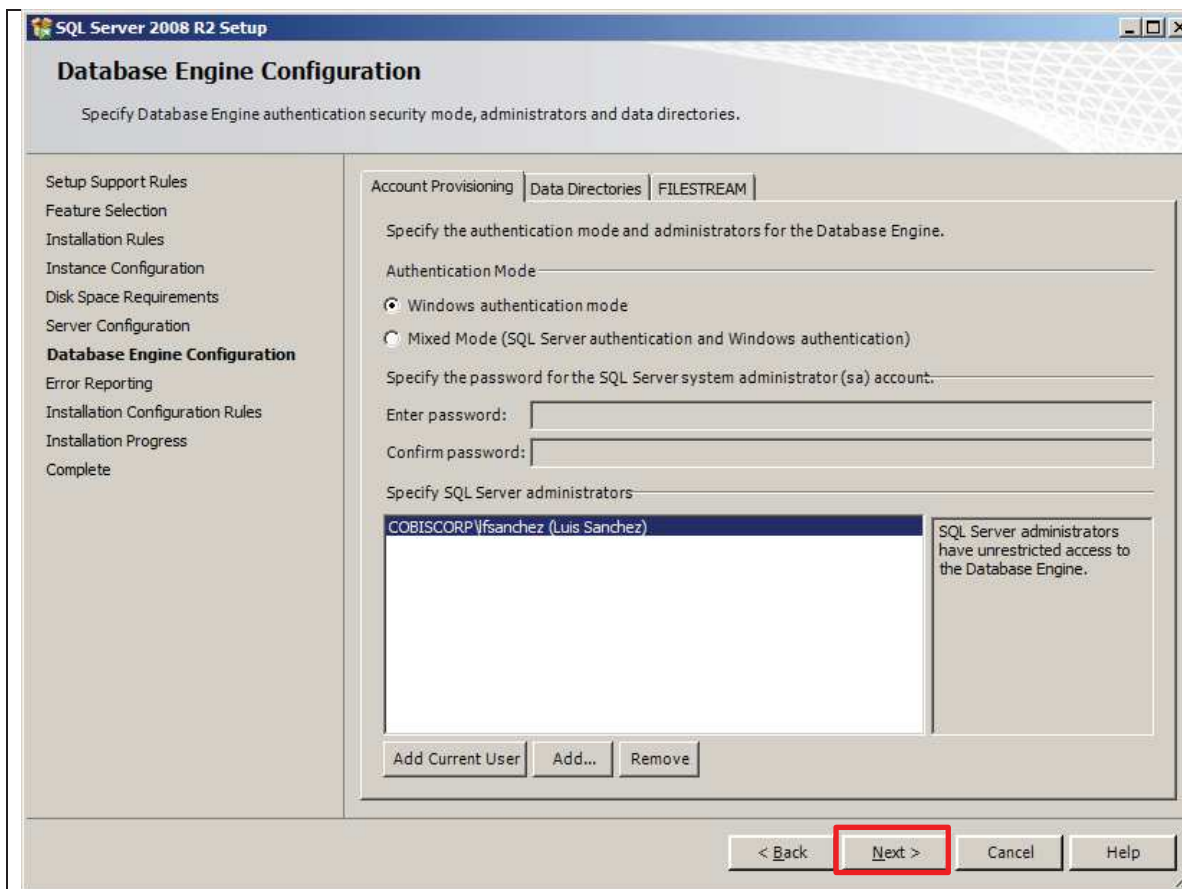


Figura 87. Configuración de motor de base de datos

9. Dejamos todo por defecto y damos **Next**.

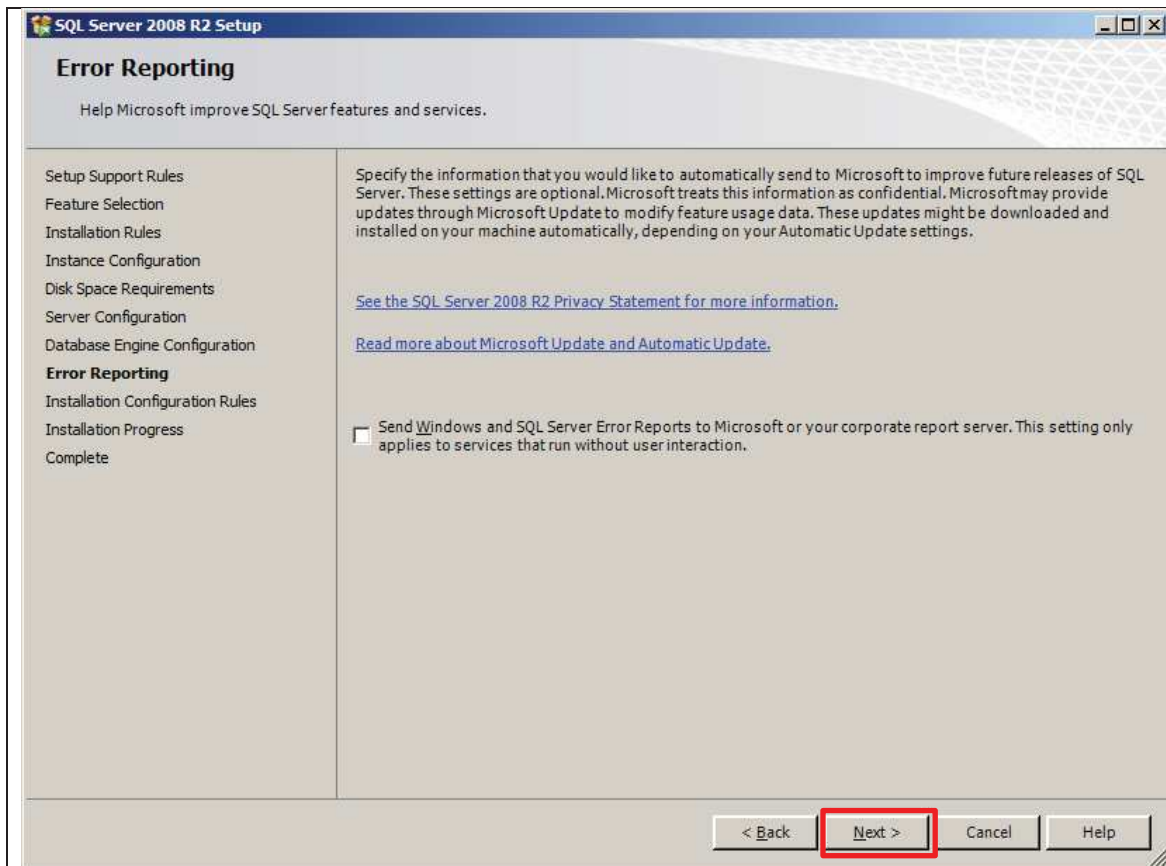


Figura 88. Reporte de errores

10. Terminada la instalación dar click en **Ok**.

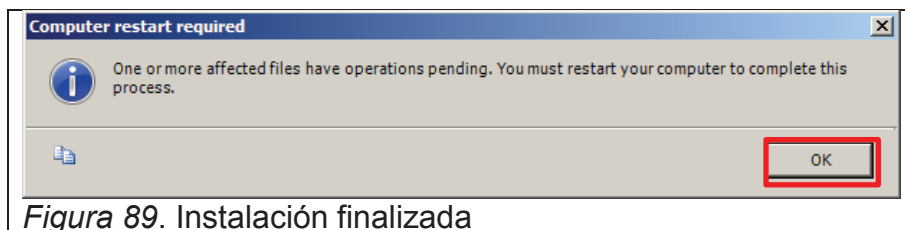


Figura 89. Instalación finalizada

11. Damos click en **Close**.

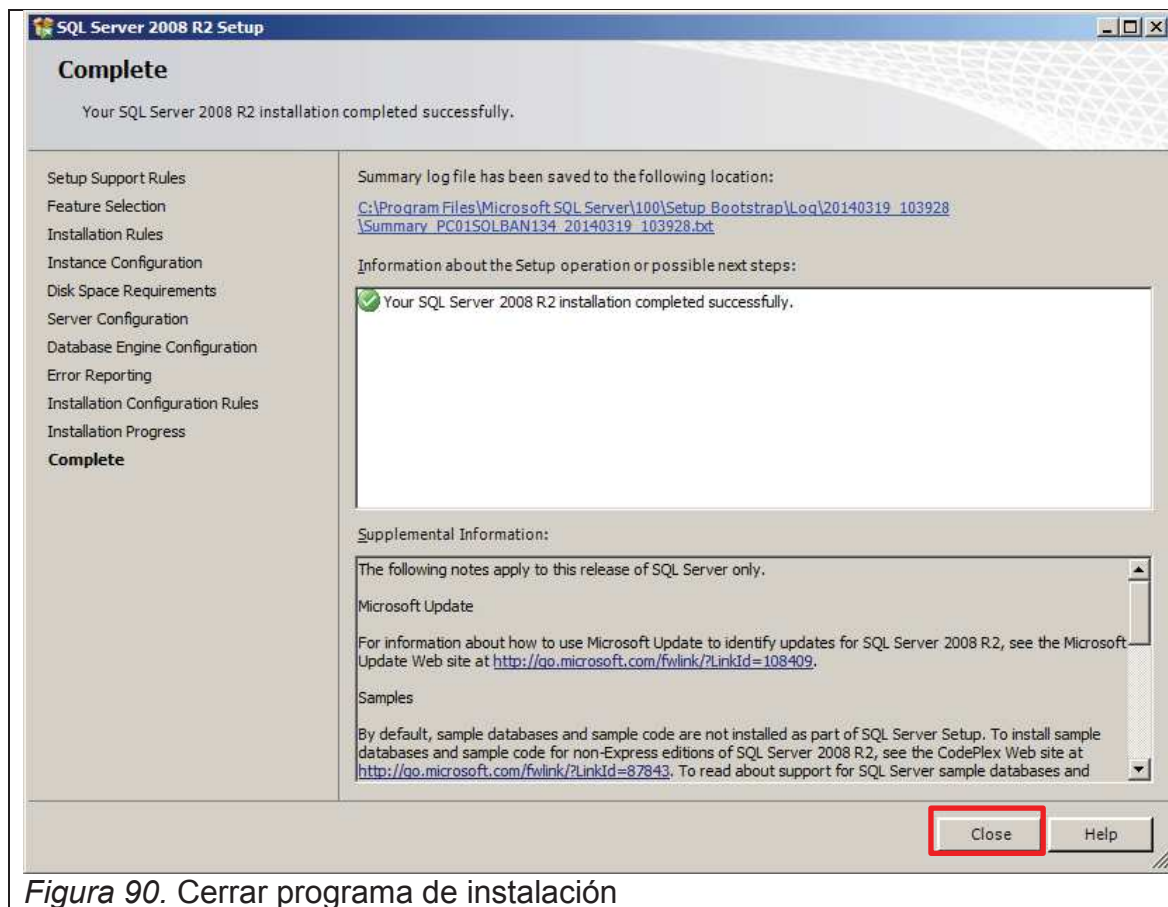


Figura 90. Cerrar programa de instalación

12. Una vez terminada la Instalación reiniciar el equipo.

13. Una vez reiniciado el equipo nos dirigimos a **Start(Inicio)/All Programs(Todos los Programas)/Microsoft SQL Server 2008 R2**, damos click en **SQL Server Management Studio**.

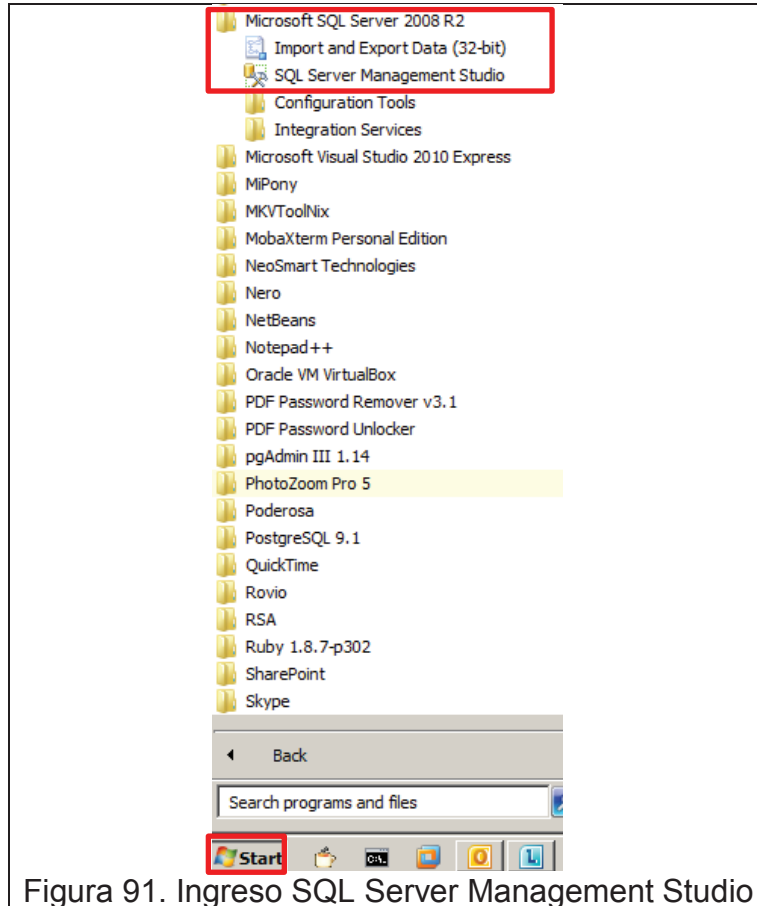


Figura 91. Ingreso SQL Server Management Studio

## 14. Damos click en **Connect**

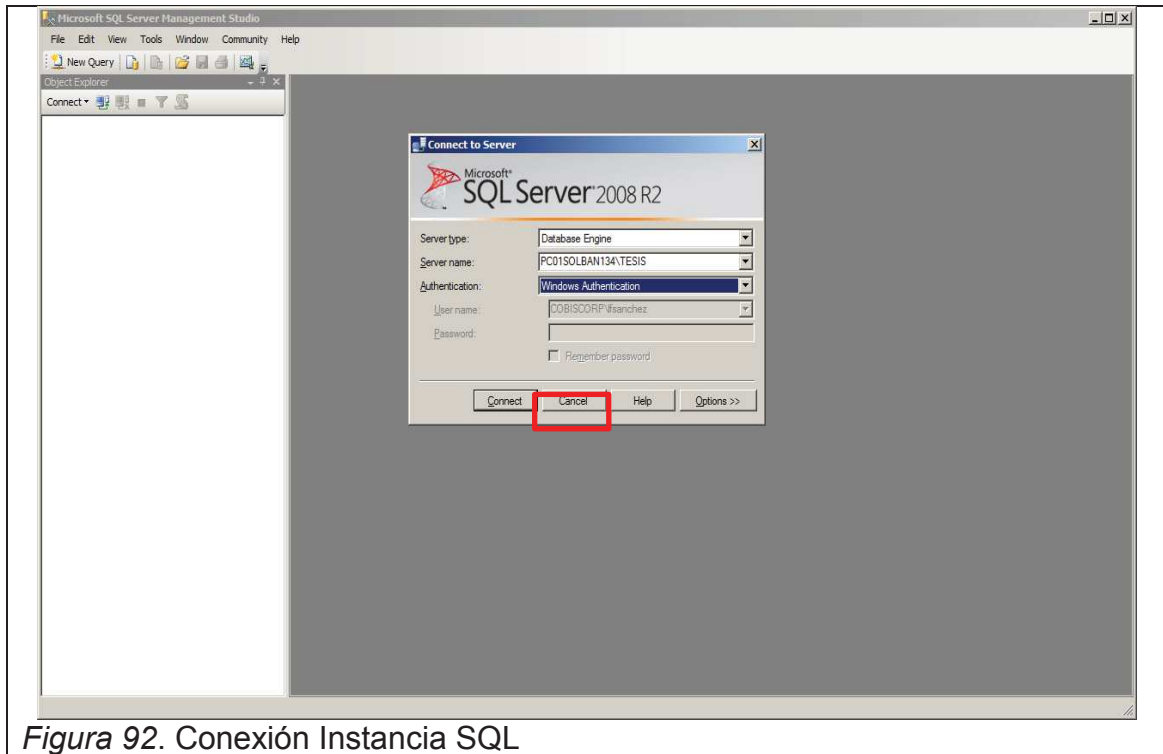


Figura 92. Conexión Instancia SQL





1. Una vez Grabada o Mantada la imagen (es\_visual\_studio\_ultimate\_2012\_x86\_dvd\_920958.iso) ejecutar el instalador, marcar **Acepto términos y condiciones de la licencia** y dar click en **Siguiente**.



Figura 94. Pantalla de instalación de Visual Studio

2. Dejamos por defecto las opciones marcadas y damos en **INSTALAR**



Figura 95. Características a instalar

3. Una vez terminada la instalación damos click en **INICIAR**.



Figura 96. Inicio de Instalación

- Una vez iniciado el programa seleccionar el entorno que se desee usar, en este caso seleccionaremos **configuración general de desarrollo** y damos click en Iniciar Visual Studio.

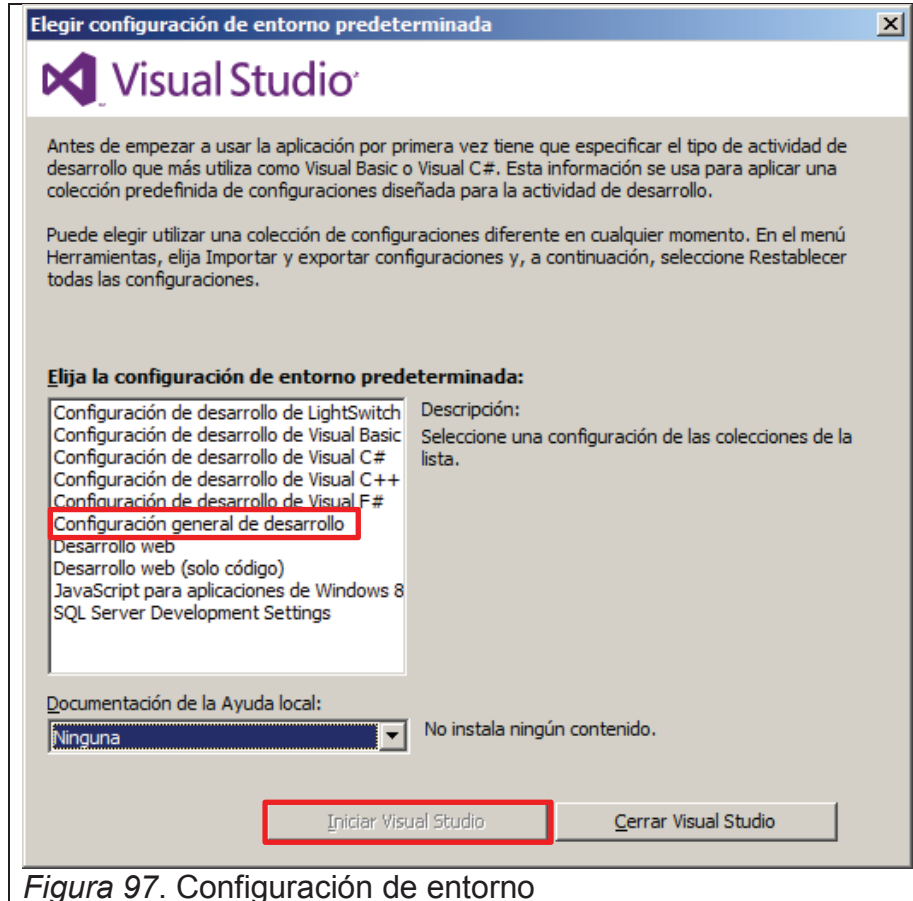


Figura 97. Configuración de entorno

5. Una vez iniciado por completo el Visual Studio nos mostrara la siguiente pantalla.

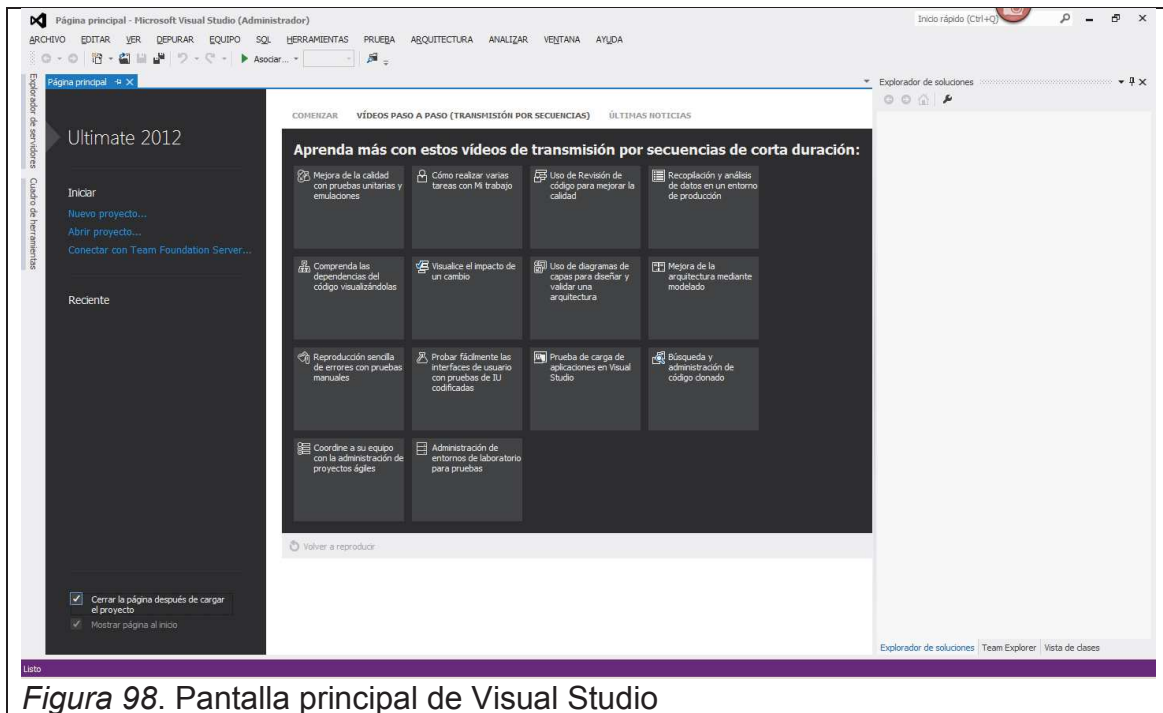


Figura 98. Pantalla principal de Visual Studio

## Configuración de Base de Datos.

1. Abrir el Microsoft SQL Server Management Studio y conectarse a la instancia creada anteriormente.

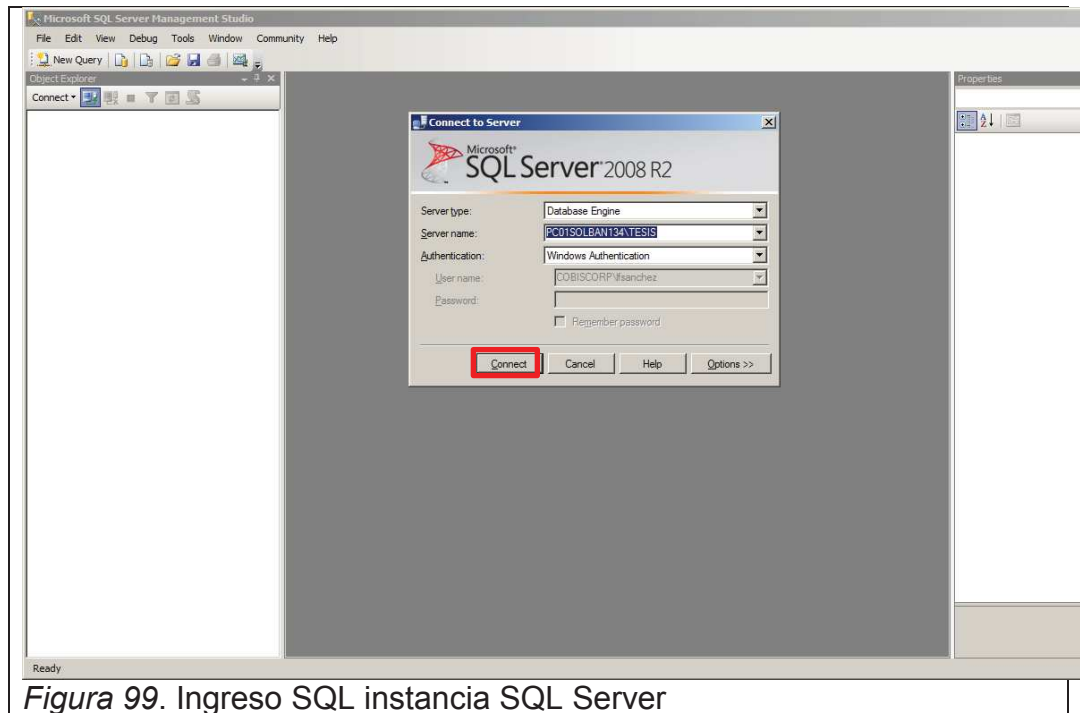


Figura 99. Ingreso SQL instancia SQL Server

2. Una vez ingresado en el Management Studio, damos click derecho en **Databases**, y damos click en **Restore Database**

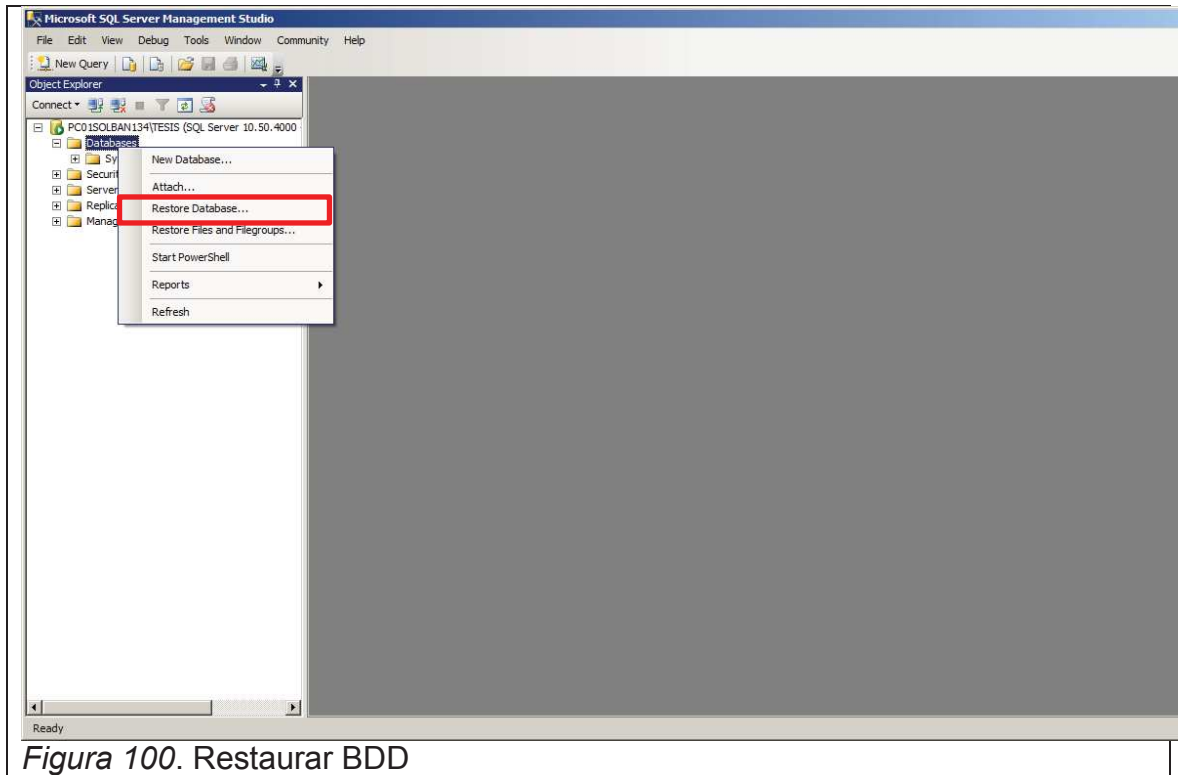


Figura 100. Restaurar BDD

3. Seleccionamos **From device**, damos click en los tres puntos (...).

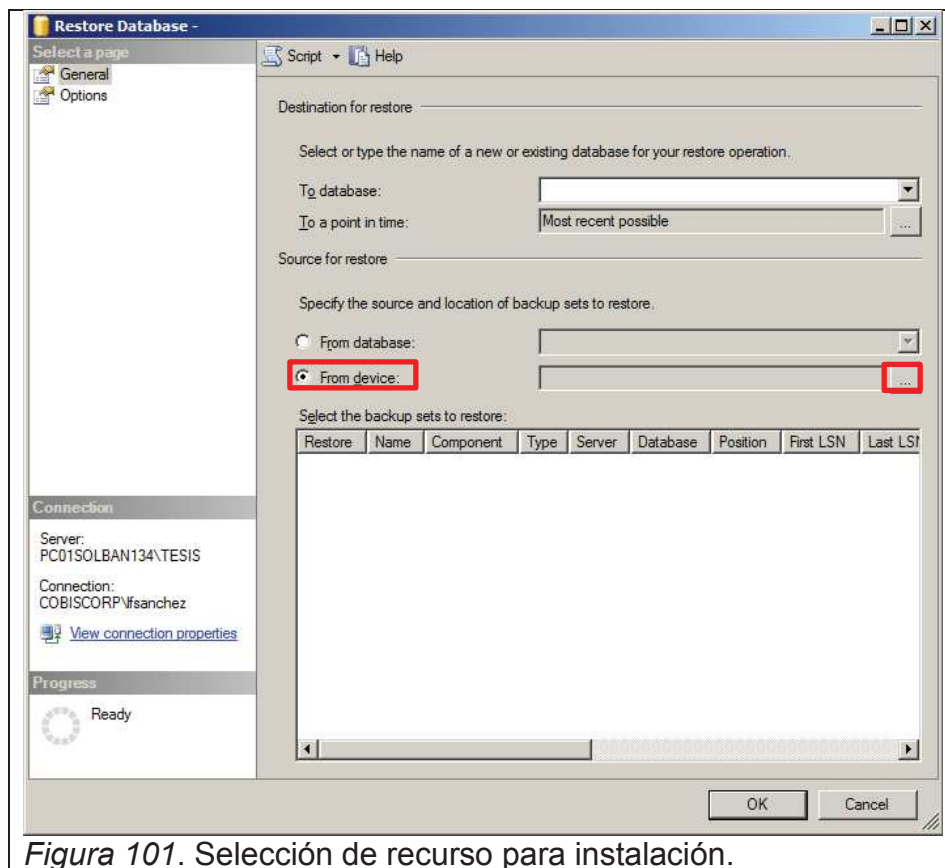


Figura 101. Selección de recurso para instalación.

4. Damos click en Add.

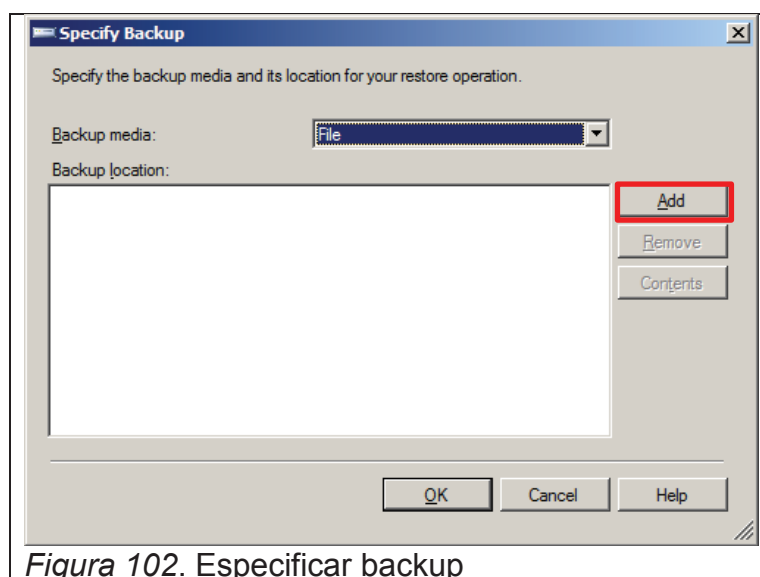


Figura 102. Especificar backup



5. Buscamos el respaldo proporcionado (udla2 limpia.bak) en la ruta donde se lo haya copiado y damos click en **OK**.

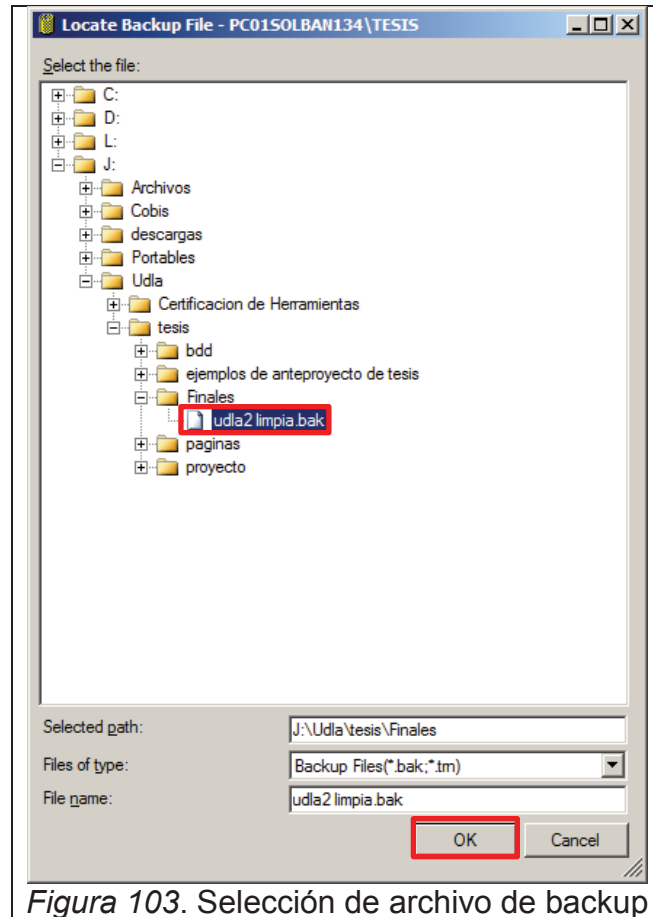


Figura 103. Selección de archivo de backup

6. Damos click en **OK**

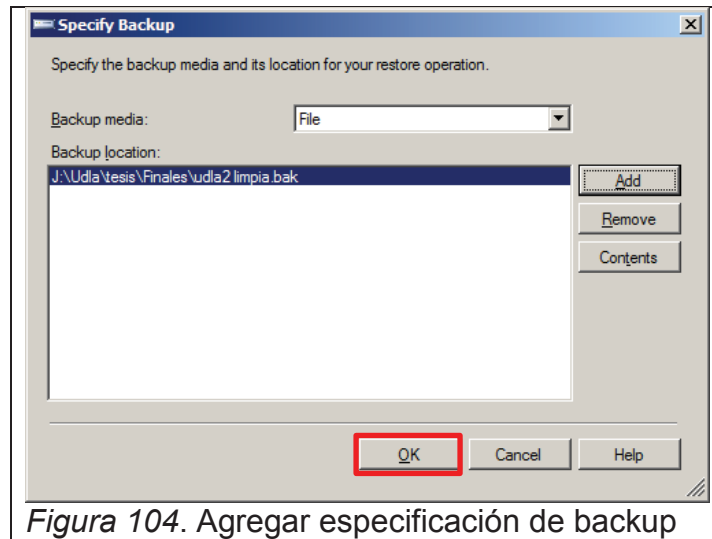


Figura 104. Agregar especificación de backup

7. Marcamos el respaldo el respaldo, escogemos la base **udla2** en **To database** y damos click en **Options**

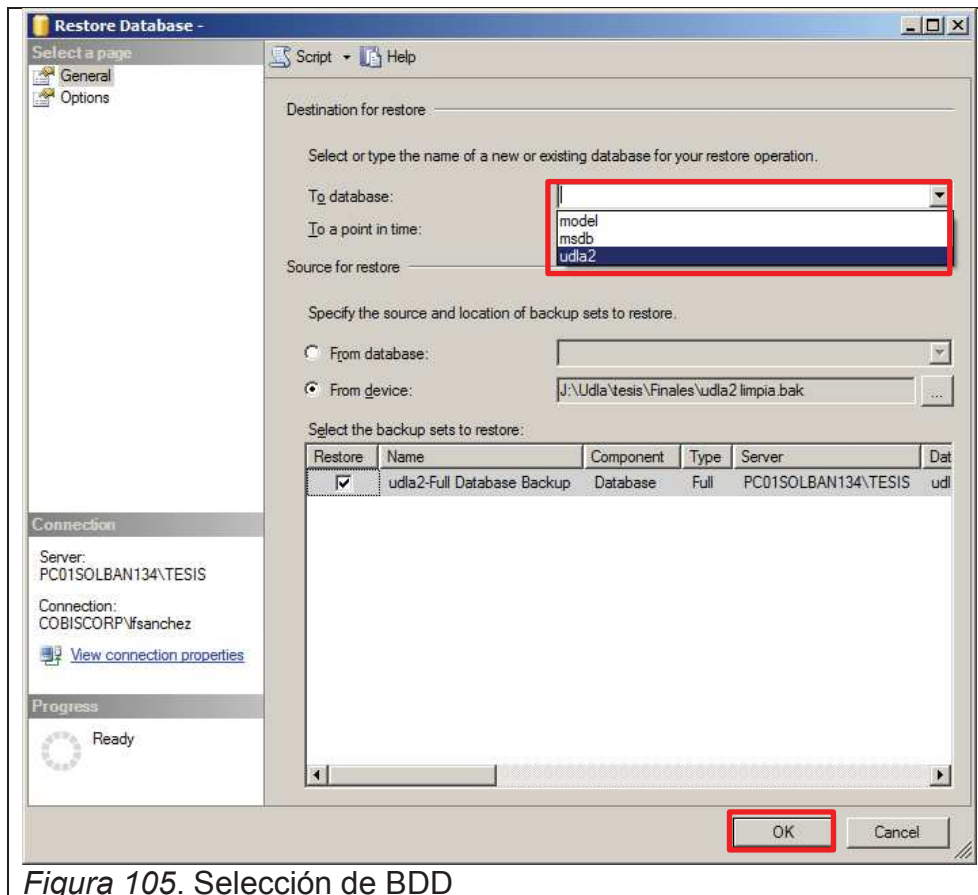


Figura 105. Selección de BDD

8. Marcamos **Overwrite the existing database (WITH REPLACE)**, y damos click en **OK**.

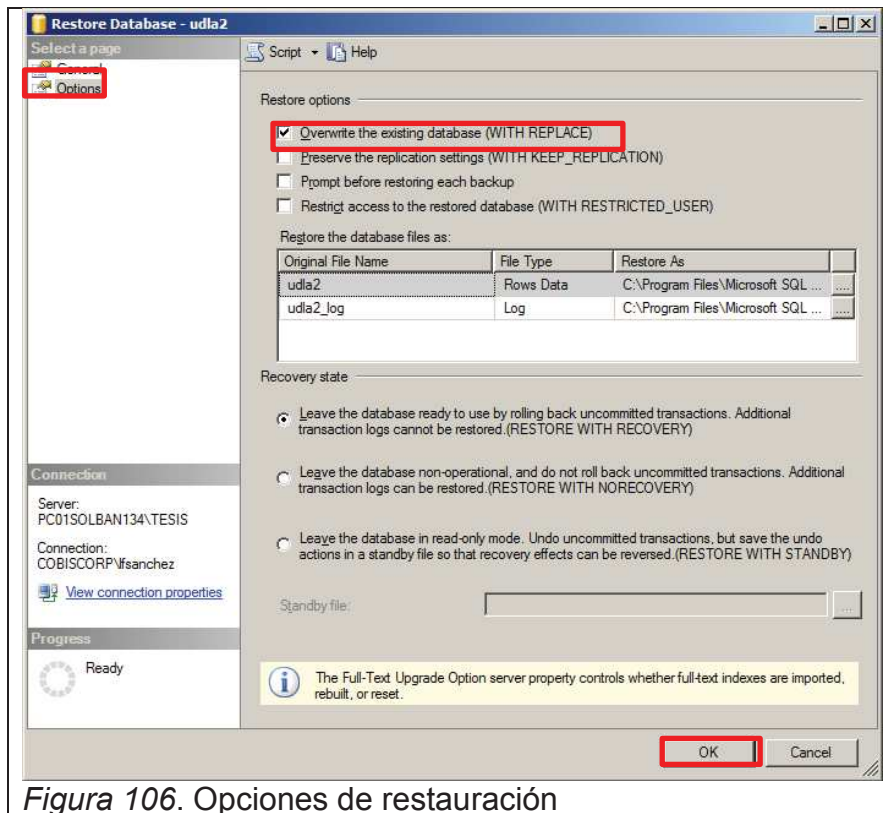


Figura 106. Opciones de restauración

9. Una terminada la restauración saldrá el siguiente mensaje y damos click en **OK**.

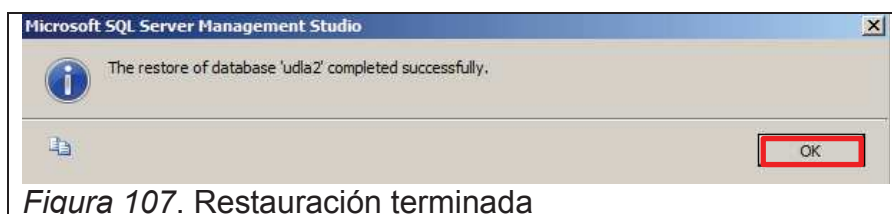


Figura 107. Restauración terminada

10. Una vez terminado el proceso tenemos la base de datos para las pruebas del Proyecto de tesis.

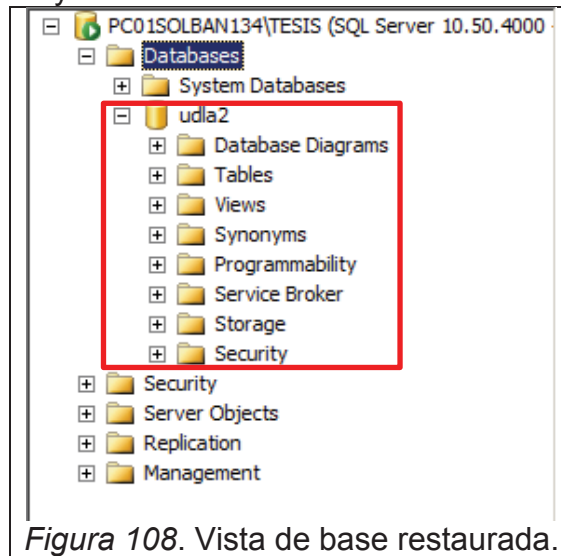


Figura 108. Vista de base restaurada.

### Configuración de proyecto de Visual Studio

1. Ingresamos a la carpeta que contiene el proyecto de Visual Studio (**Tesis**) y ejecutamos el archivo **Tesis12.sln**

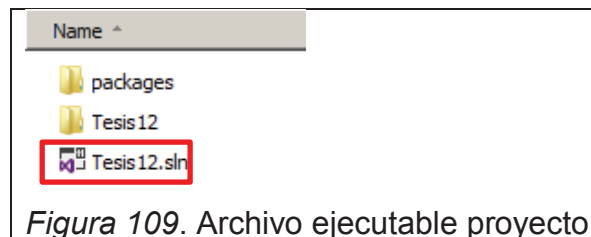


Figura 109. Archivo ejecutable proyecto

2. Una vez abierto el proyecto nos dirigimos al **Explorador de soluciones**, y buscamos el archivo

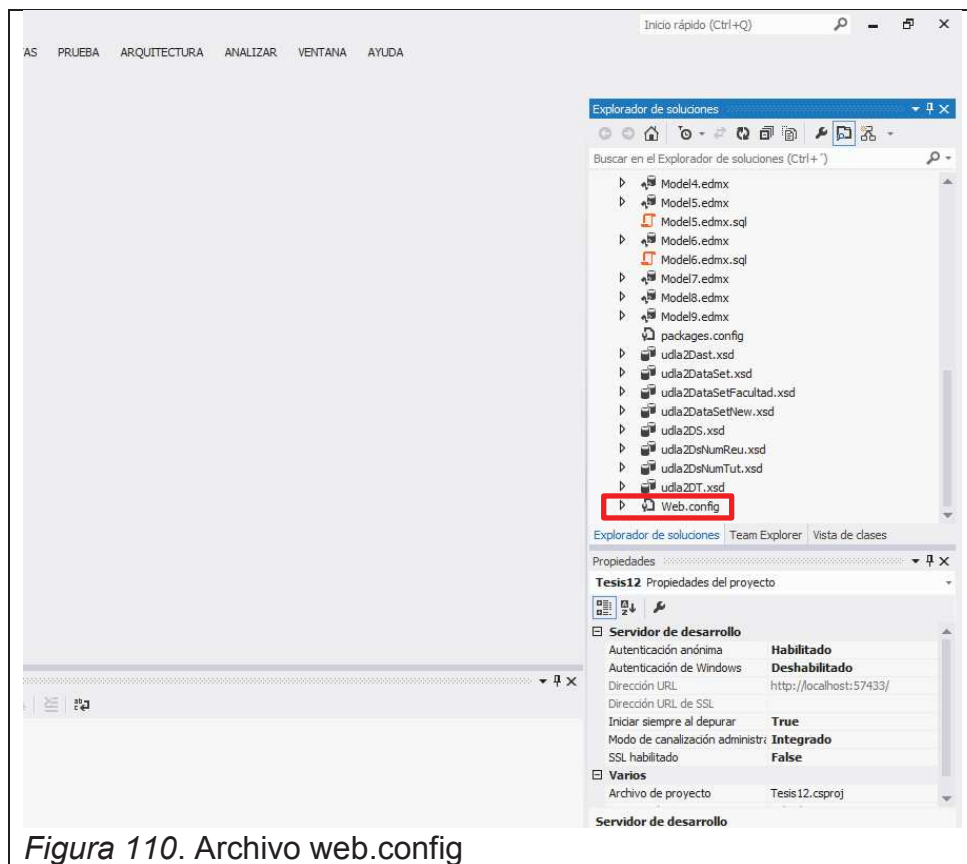


Figura 110. Archivo web.config

- Una vez abierto el archivo, buscamos el **connectionString** señalamos el valor que se encuentra en el tag "**Data Source=** ", ejecutamos la combinación de teclas **Ctrl + H** colocamos el nombre la instancia de base de datos creada anteriormente, en este caso el nombre es **PC01SOLBAN134\TESIS** en el segundo **textbox** y presionamos el boto **Reemplazar todo (Alt + A)**.

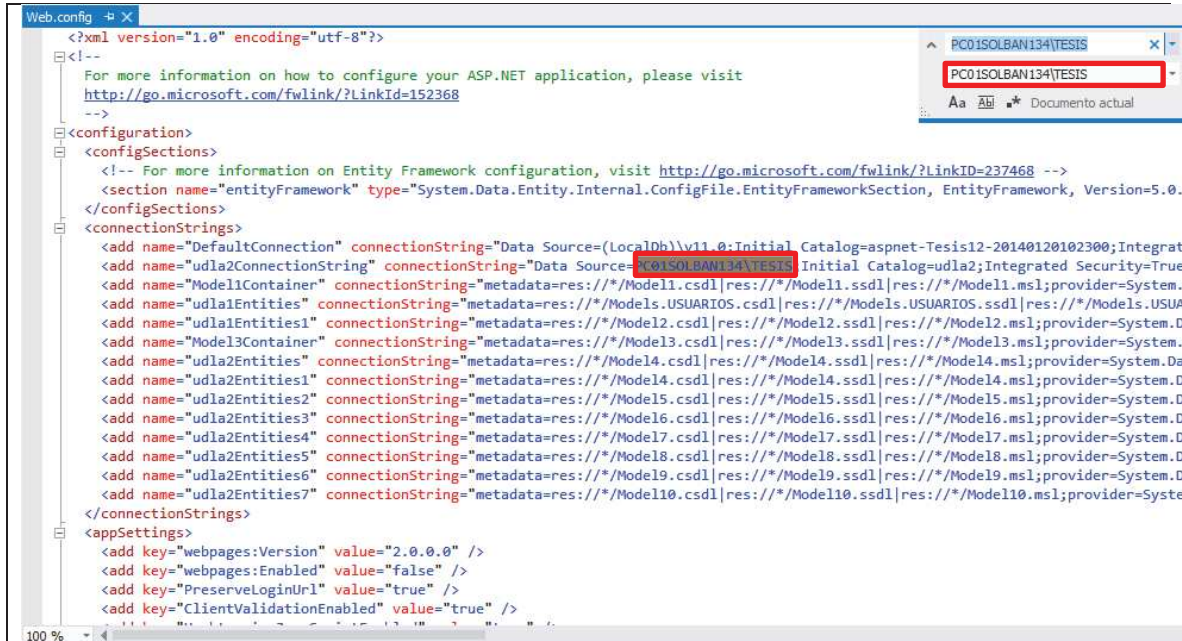


Figura 111. Edición de archivo web.config

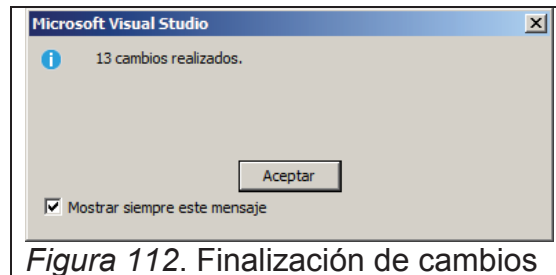


Figura 112. Finalización de cambios

- Una vez terminado el reemplazo de este valor guardamos los cambios realizados

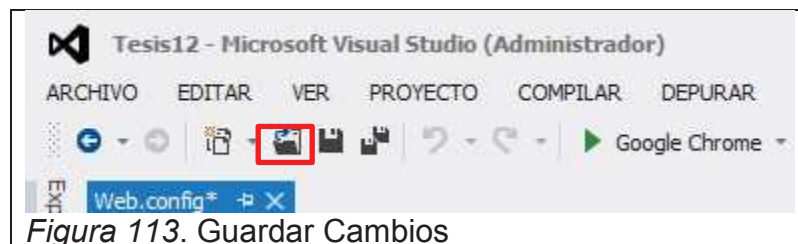
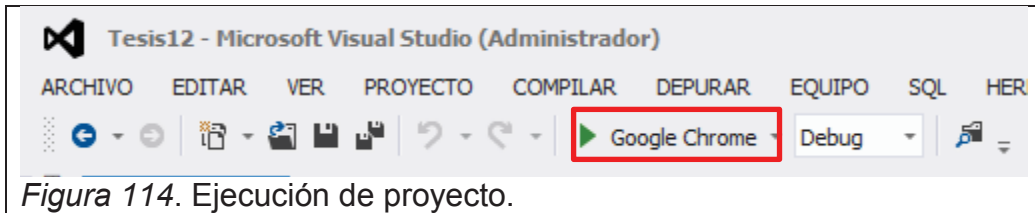


Figura 113. Guardar Cambios

5. Una vez guardados los cambios procedemos a ejecutar el proyecto y esperamos hasta que se habrá el navegador.



6. Una vez iniciado el proyecto nos mostrara una pantalla como la siguiente y se podrá realizar las pruebas necesarias.



## **ANEXO 4: MANUAL DE USUARIO**

El siguiente manual se ha realizado con la finalidad de detallar las opciones más importantes del sistema brindándole al usuario una guía de cómo proceder para utilizar cada una de estas.

UNIVERSIDAD DE LAS AMÉRICAS UDLA

MANUAL DE USUARIO

SISTEMA DE REGISTRO DE REUNIONES DE TESIS

FERNANDO SÁNCHEZ LLERENA

JURGEN ZAMBRANO ZAMBRANO

Fecha de creación: Marzo, 2014

Versión 1.0



## INGRESO AL SISTEMA (ADMINISTRADOR)

Se refiere a la manera de como ingresar al sistema desde la pantalla de inicio.

Esta pantalla será la de ingreso al sistema



Figura 116. Pantalla de login

Se ingresaran los siguientes datos para poder ingresar al sistema:

- Numero matricula/Nombre de usuario: Es el nombre o numero dado por el administrador con el cual el usuario se identifica en el sistema
- Password: Es el número de cedula del usuario que desea ingresar al sistema.

## MENÚ PRINCIPAL

Se refiere a la Administración total del sistema

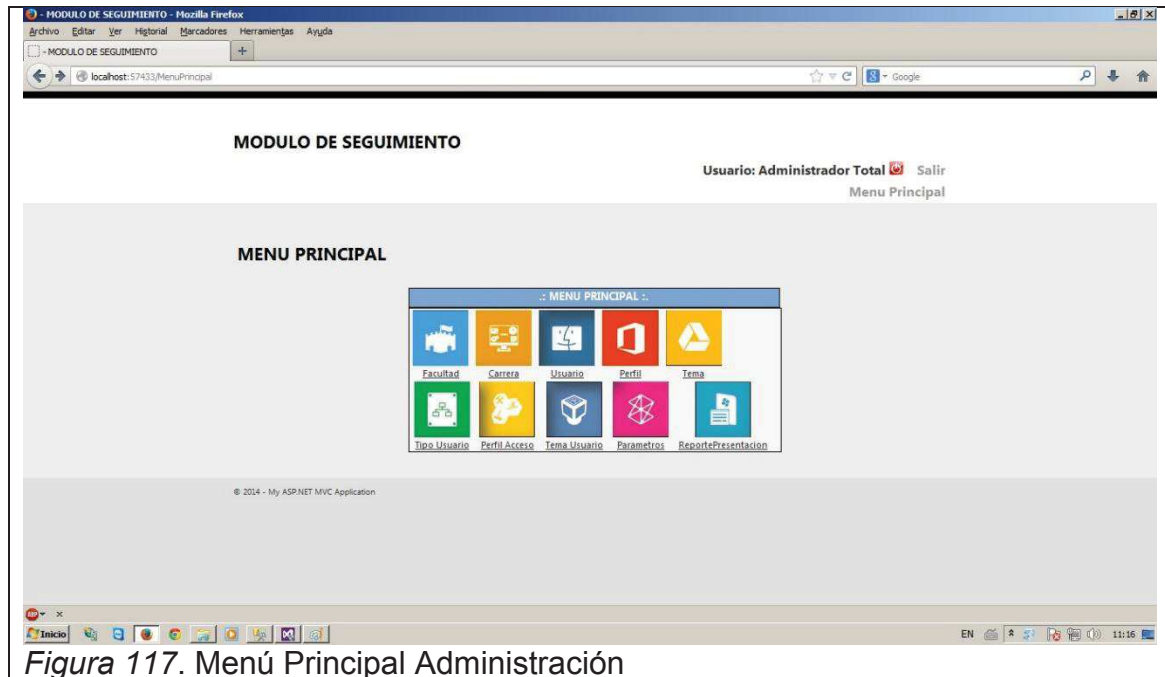


Figura 117. Menú Principal Administración

## USUARIO

En esta opción se podrá administrar los datos personales de los usuarios que utilizaran el sistema

Tus Descargas Sin Límites - | x Index - MODULO DE SEGUIM... x

localhost:57433/Estudiante

Aplicaciones Tesis

## MODULO DE SEGUIMIENTO

Usuario: **Administrador sistemas**  Salir  
 Menu Principal

### Index

[Crear Usuario](#)

| CARRERA   | TIPO USUARIO | NUMERO DE MATRICULA | NUMERO DE CEDULA | NOMBRES APELLIDOS | TELEFONO CASA | CELULAR | CORREO PERSONAL  | CORREO UNIVERSIDAD | DIRECCION | FECHA NACIMIENTO | CONTRASEÑA | PER     |
|---|--------------|---------------------|------------------|-------------------|---------------|---------|------------------|--------------------|-----------|------------------|------------|---------|
| Ingeniería en Sistemas de Computación e Informática | PROFESOR     | jleon               | 1234567890       | Juan Leon         | 222222        | 222222  | admin@udlanet.ec | admin@udlanet.ec   | UDLA      | 0                | XXXXX      | ACC MEC |
| Ingeniería en Sistemas de Computación e Informática | PROFESOR     | mgalarza            | 1234567890       | Profesor 2        | 222222        | 222222  | admin@udlanet.ec | admin@udlanet.ec   | UDLA      | 0                | XXXXX      | ACC MEC |
| Ingeniería en                                       |              |                     |                  |                   |               |         |                  |                    |           |                  |            |         |

localhost:57433/Usuario/Logout

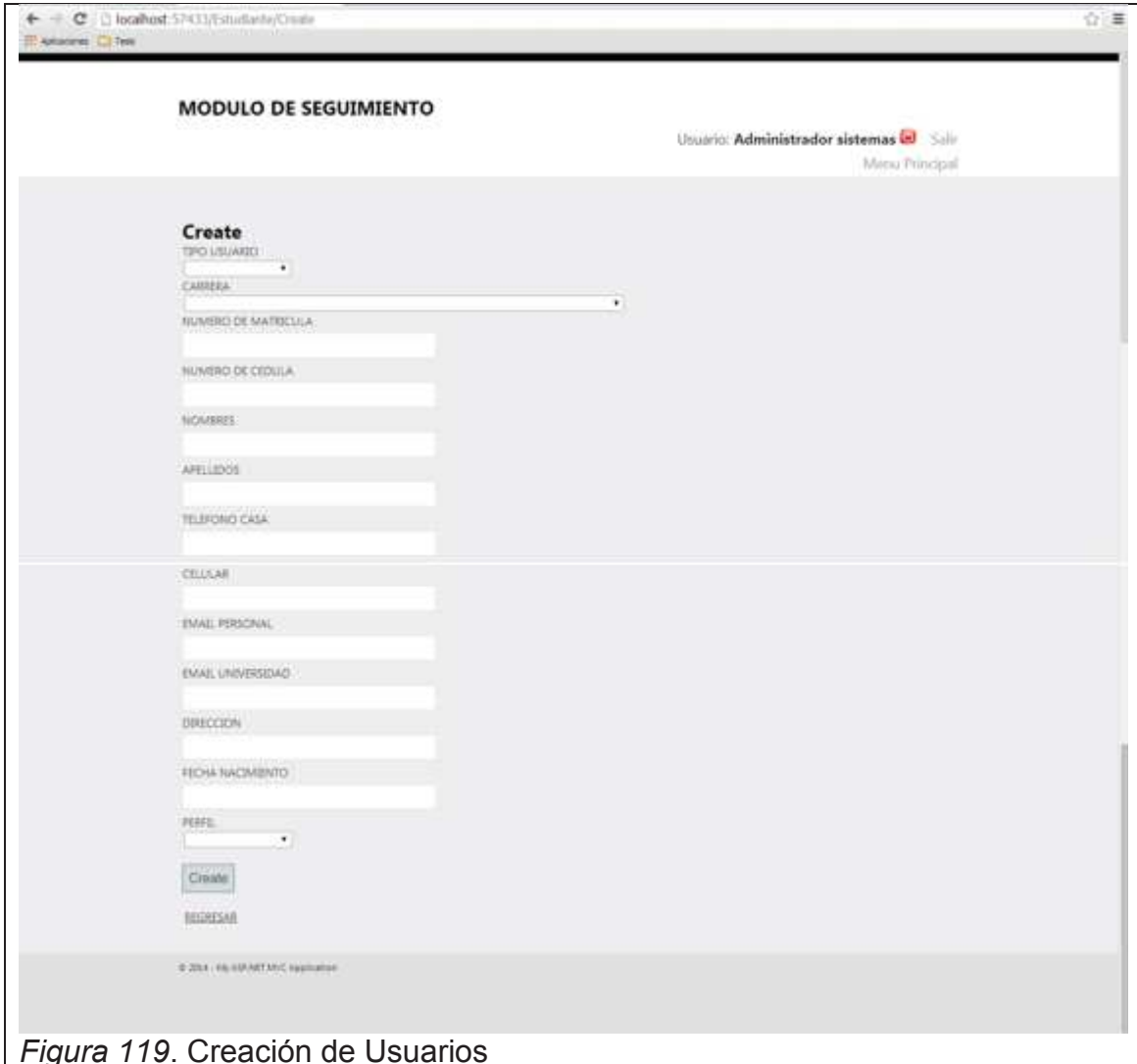
Inicio  EN 7:33

Figura 118. Opción Usuario

Esta opción permite realizar las siguientes acciones:

## Crear un nuevo Usuario

Al dar clic en esta acción podremos ingresar un nuevo usuario, procedemos a ingresar la información del usuario, esta es:



The screenshot shows a web browser window with the URL `localhost:57433/Estudiante/Create`. The page title is "MODULO DE SEGUIMIENTO". In the top right corner, it displays "Usuario: Administrador sistemas" with a "Salir" link and a "Menu Principal" link. The main content area is titled "Create" and contains the following form fields:

- TIPO USUARIO (dropdown menu)
- CARRERA (dropdown menu)
- NUMERO DE MATRICULA (text input)
- NUMERO DE CEDULA (text input)
- NOMBRES (text input)
- APellidos (text input)
- TELEFONO CASA (text input)
- CELULAR (text input)
- EMAIL PERSONAL (text input)
- EMAIL UNIVERSIDAD (text input)
- DIRECCION (text input)
- FECHA NACIMIENTO (text input)
- PERFE (dropdown menu)

At the bottom of the form, there is a "Create" button and a "REGRESAR" link. The footer of the page reads "© 2014 - All Rights Reserved".

Figura 119. Creación de Usuarios

- Tipo de usuario: Se refiere al tipo de usuario en la aplicación, sea este; administrador, profesor, alumno.
- Carrera: Se refiere a la carrera del estudiante (Si el usuario es profesor o administrador en este campo se escogerá como NO DEFINIDO).
- Número de matrícula / Nombre de usuario: Se refiere a la identificación del usuario dentro del sistema.
- Numero de Cedula: Se refiere al número de cedula del usuario el cual será la contraseña de este en el sistema.

- Nombres: Se ingresan los nombres del usuario
- Apellidos: Se ingresan los apellidos del usuario
- Teléfono casa: Se ingresa el número telefónico del domicilio del usuario.
- Celular: Se ingresa el numero celular del usuario
- Email Personal: Se ingresa la dirección de correo electrónico personal del usuario
- Email Universidad: Se ingresa la dirección de correo electrónico de la universidad del usuario.
- Direccion: Se ingresa la dirección domiciliaria del usuario
- Fecha Nacimiento: Se ingresa la fecha de nacimiento del usuario en el siguiente formato: DD/MM/AAAA
- Perfil: Se refiere al perfil que se le otorga al usuario dentro del sistema

Para grabar el nuevo Usuario se debe dar clic en el botón Crear.

#### *Actualizar un Usuario*

Para poder actualizar un Usuario se debe seleccionar el siguiente botón de la derecha del usuario, una vez seleccionado la información de este aparece, luego se procede a realizar las modificaciones pertinentes y finalmente se da clic en el botón Save.

#### *Eliminar un Usuario*

Para poder eliminar un Usuario se debe seleccionar el siguiente botón de la derecha del usuario y con esto se eliminara el usuario.

## TEMA

En esta opción se podrá administrar los temas de tesis en el sistema

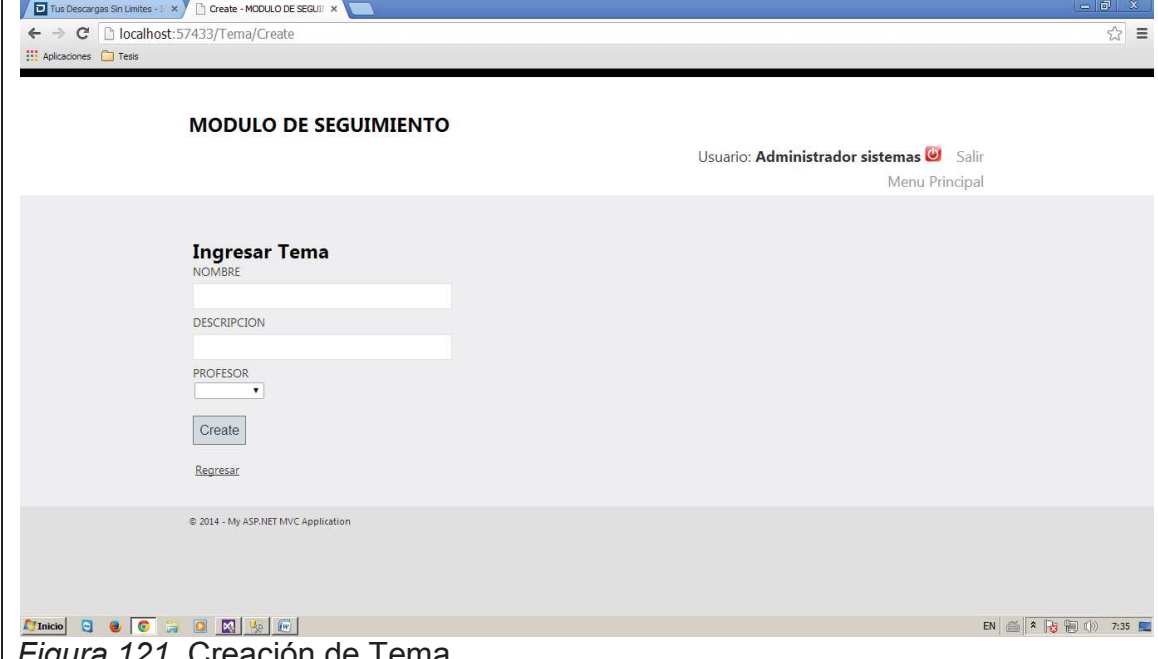


Figura 120. Opción Tema

Esta opción permite realizar las siguientes acciones:

## Ingresar Nuevo Tema

Al dar clic en esta acción podremos ingresar un nuevo Tema de Tesis, procedemos a ingresar la información del Tema de Tesis, esta es:



The screenshot shows a web browser window with the URL `localhost:57433/Tema/Create`. The page title is "MODULO DE SEGUIMIENTO". The user is logged in as "Administrador sistemas" with a "Salir" link and a "Menu Principal" link. The main content area is titled "Ingresar Tema" and contains the following form fields:

- NOMBRE: A text input field.
- DESCRIPCION: A text input field.
- PROFESOR: A dropdown menu.
- Buttons: "Create" and "Regresar".

At the bottom of the page, there is a copyright notice: "© 2014 - My ASP.NET MVC Application". The Windows taskbar is visible at the bottom of the browser window, showing the Start button, several application icons, and the system tray with the time 7:35.

Figura 121. Creación de Tema

- Nombre: Se ingresa el nombre del Tema de Tesis.
- Descripción: Se ingresa una pequeña descripción del Tema de Tesis ingresado.
- Profesor: Se refiere al profesor que será el tutor de dicha tesis.

Para grabar el nuevo Tema de Tesis se debe dar clic en el botón Crear.

Luego aparecerá la siguiente pantalla con la información del Tema de Tesis anteriormente creado



Figura 122. Tema creado

### *Actualizar un Tema de Tesis*

Para poder actualizar un Tema de Tesis se debe seleccionar el siguiente botón de la derecha del Tema de Tesis, una vez seleccionado la información de este aparece, luego se procede a realizar las modificaciones pertinentes y finalmente se da clic en el botón Save.

### *Eliminar un Tema de Tesis*

Para poder eliminar un Tema de Tesis se debe seleccionar el siguiente botón de la derecha del Tema de Tesis y con esto se eliminara el Tema de Tesis.



## TEMAUSUARIO

En esta opción asignara a los estudiantes que van a desarrollar un Tema de Tesis



Figura 123. Opción Tema Usuario

Esta opción permite realizar las siguientes acciones:

## Ingresar Nuevo

Al dar clic en esta acción podremos asignar estudiante(s) a un tema de tesis, procedemos a ingresar la información, esta es:

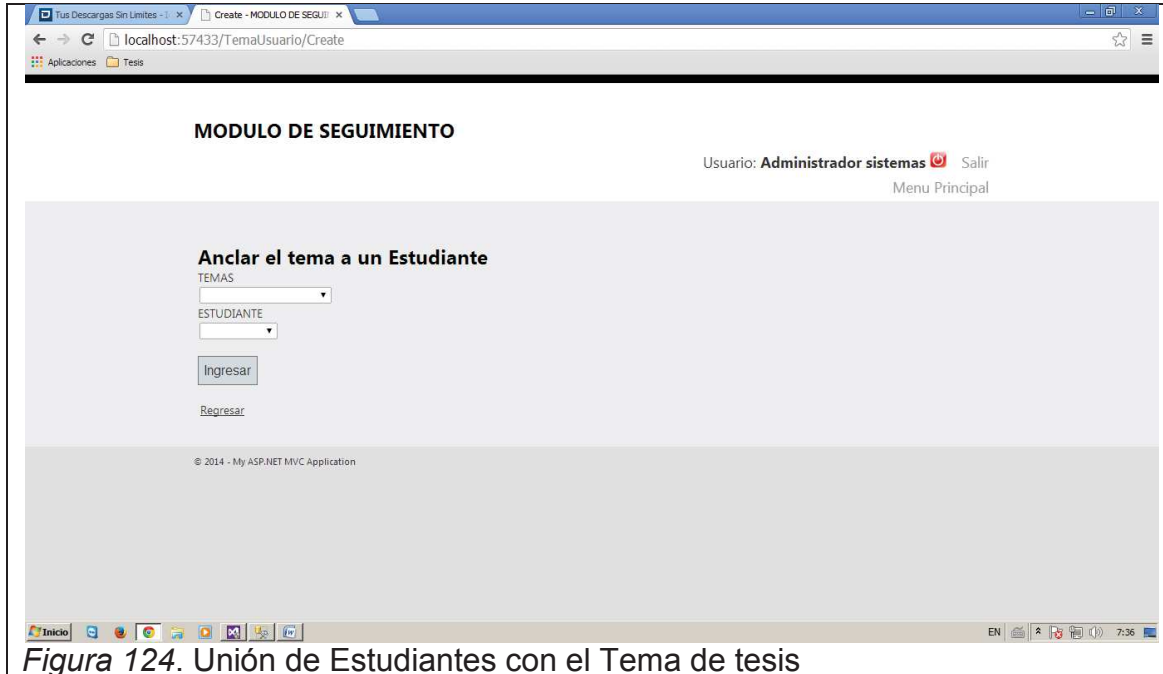


Figura 124. Unión de Estudiantes con el Tema de tesis

- Temas: Se escoge el Tema de Tesis
- Estudiante: Se asigna el estudiante al tema de tesis

Nota: Si se desea ingresar más de un estudiante se deberá repetir el proceso escogiendo a cada uno de estos con el mismo tema de tesis.

## INGRESO AL SISTEMA (PROFESOR)

Se refiere a la manera de como ingresar al sistema desde la pantalla de inicio.

Esta pantalla será la de ingreso al sistema

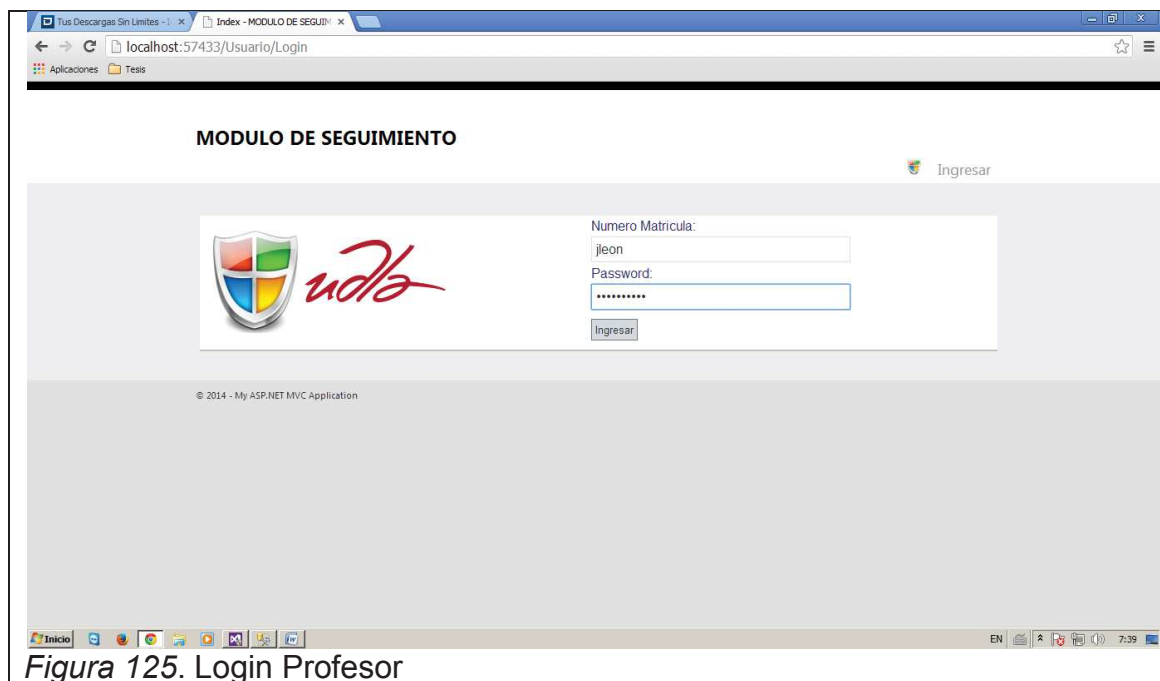


Figura 125. Login Profesor

Se ingresaran los siguientes datos para poder ingresar al sistema:

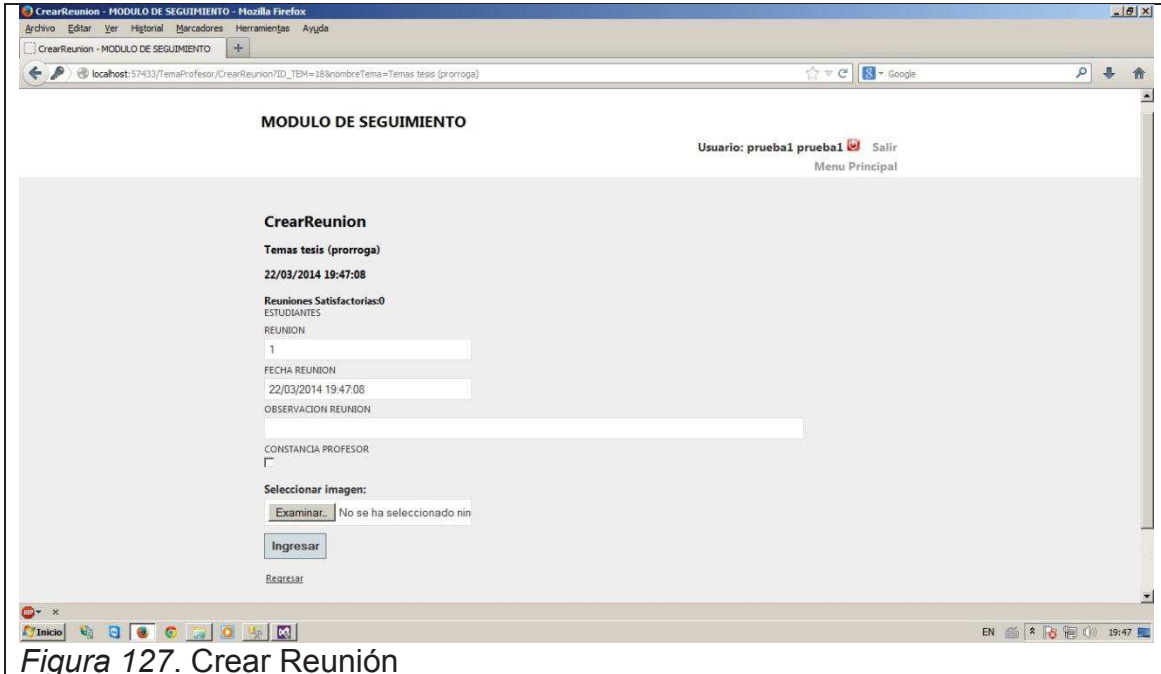
- Numero matricula/Nombre de usuario: Es el nombre o número dado por el administrador con el cual el usuario se identifica en el sistema.
- Password: Es el número de cedula del usuario que desea ingresar al sistema



Esta opción permite realizar las siguientes acciones:

## Crear una Reunión

Al dar clic en esta acción podremos crear una nueva reunión, procedemos a registrar la información de la Reunión, esta es:



The screenshot shows a web browser window with the following content:

- Page title: **MODULO DE SEGUIMIENTO**
- User information: **Usuario: prueba1 prueba1** with a 'Salir' link and a 'Menu Principal' link.
- Section: **CrearReunion**
- Context: **Temas tesis (prorroga)**
- Date and time: **22/03/2014 19:47:08**
- Count: **Reuniones Satisfactorias:0**
- Form fields:
  - REUNION**: Input field containing '1'
  - FECHA REUNION**: Input field containing '22/03/2014 19:47:08'
  - OBSERVACION REUNION**: Large empty text area
  - CONSTANCIA PROFESOR**: Checkbox (unchecked)
  - Seleccionar imagen:** Includes an 'Examinar...' button and the text 'No se ha seleccionado nin'
  - Ingresar**: Button
  - Regresar**: Link

Figura 127. Crear Reunión

- Reunión: Se debe ingresar el número de la Reunión (1-16)
- Fecha Reunión: Se refiere a la fecha y hora en la que se registra la reunión.
- Observación Reunión: Se debe ingresar la actividad realizada en la reunión.
- Seleccionar Imagen: Se carga la imagen de la foto tomada anteriormente.
- Constancia Profesor: Se marca la casilla para que la reunión quede grabada como satisfactoria.

Finalmente se da clic sobre Ingresar para que la reunión quede guardada y habilitada para el registro del estudiante.

## Ver una Reunión

Al dar clic en esta opción se podrán visualizar todas las reuniones creadas por el profesor por cada uno de los temas de tesis asignados.



The screenshot shows a web browser window with the URL `localhost:57433/TemaProfesor/Ver?ID_TEM=13`. The page title is "MODULO DE SEGUIMIENTO". The user is identified as "Juan Leon" with a "Salir" link and a "Menu Principal" link. Below this is a section titled "LISTA REUNIONES" containing a table with the following data:

| #REUNION | FECHA REUNION      | TEMA                  | OBSERVACION            | OK PROFESOR | OK ESTUDIANTE |
|----------|--------------------|-----------------------|------------------------|-------------|---------------|
| 1        | 19/02/2014 7:43:04 | Modulo de seguimiento | prueba ingreso reunion | ok          |               |

At the bottom of the page, there is a copyright notice: "© 2014 - My ASP.NET MVC Application". The Windows taskbar at the bottom shows the system tray with the time 7:43.

Figura 128. Vista de reunión Creada

## INGRESO AL SISTEMA (ESTUDIANTE)

Se refiere a la manera de como ingresar al sistema desde la pantalla de inicio.

Esta pantalla será la de ingreso al sistema

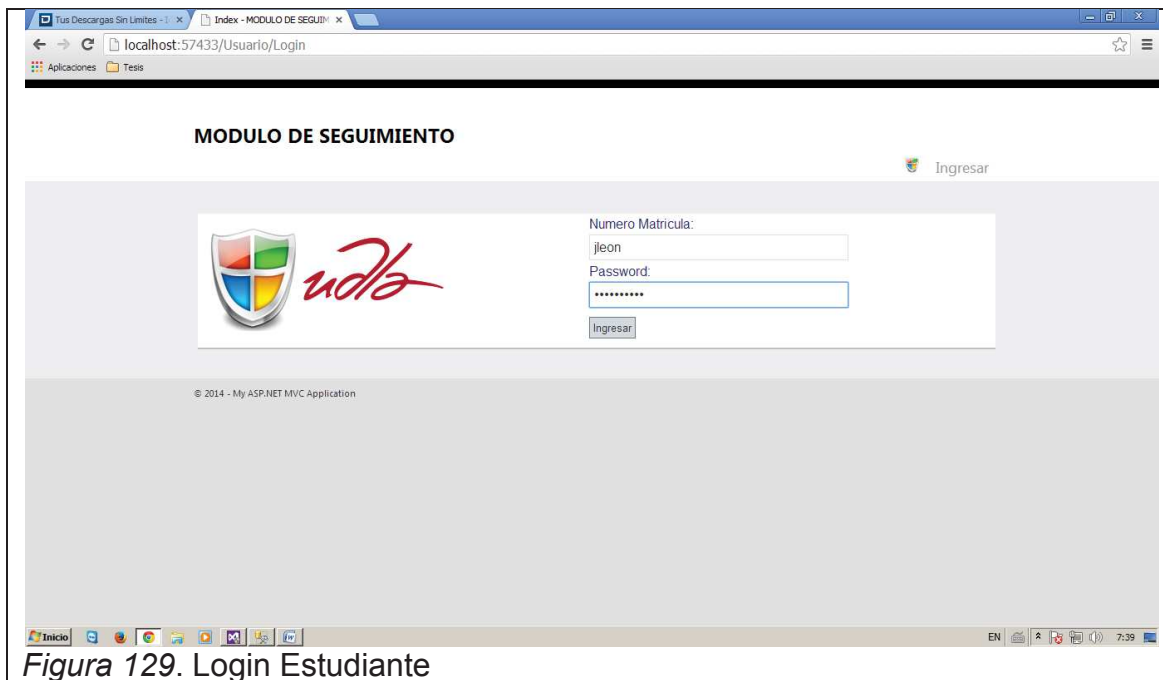


Figura 129. Login Estudiante

Se ingresaran los siguientes datos para poder ingresar al sistema:

- Numero matricula/Nombre de usuario: Es el nombre o número dado por el administrador con el cual el usuario se identifica en el sistema.
- Password: Es el número de cedula del usuario que desea ingresar al sistema

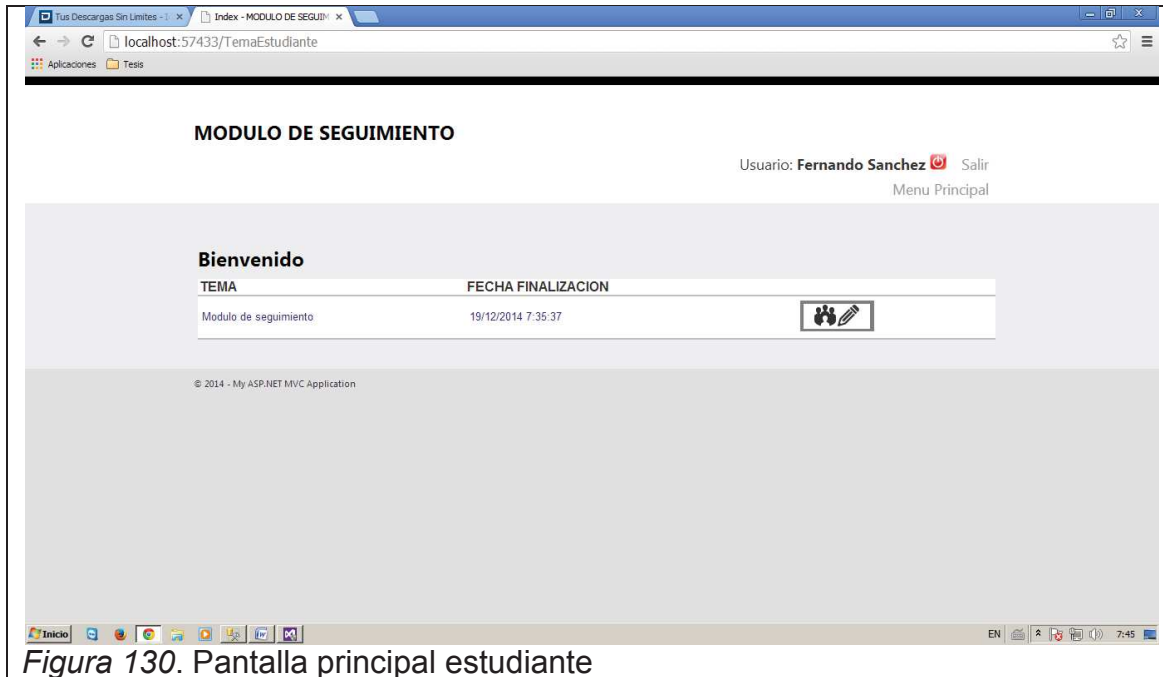


Figura 130. Pantalla principal estudiante

Esta opción permite realizar las siguientes acciones:



## Registrar una Reunión

Al dar clic en esta opción nos aparece la siguiente pantalla, la cual nos muestra la información de la reunión ingresada previamente por el profesor



Figura 131. Lista de reuniones.

Luego de esto damos clic sobre la figura

Nos aparecerá la siguiente pantalla



Figura 132. Registro de reunión.

Aquí solamente debemos de marcar la casilla de Estudiante y luego dar clic en Guardar.

Luego de esto la reunion queda guardada como se muestra en la figura a continuacion.



The screenshot shows a web browser window with the URL `localhost:57433/TemaEstudiante/ListaReuniones?ID_TEM=13&nombreTema=Modulo%20de%20seguimiento`. The page title is "MODULO DE SEGUIMIENTO". The user is identified as "Fernando Sanchez" with a "Salir" link and a "Menu Principal" link. Below this, there is a section titled "ListaReuniones" containing a table with the following data:

| REUNION | FECHA              | OBSERVACION            | TUTOR | ESTUDIANTE |
|---------|--------------------|------------------------|-------|------------|
| 1       | 19/02/2014 7:43:04 | prueba ingreso reunion | ok    | ok         |

At the bottom of the page, there is a copyright notice: "© 2014 - My ASP.NET MVC Application". The Windows taskbar at the bottom shows the system tray with the time 7:46.

Figura 133. Reunión guardada.

## PARAMETROS

Esta opción nos sirve para administrar ciertas condiciones que podrían cambiar con el tiempo

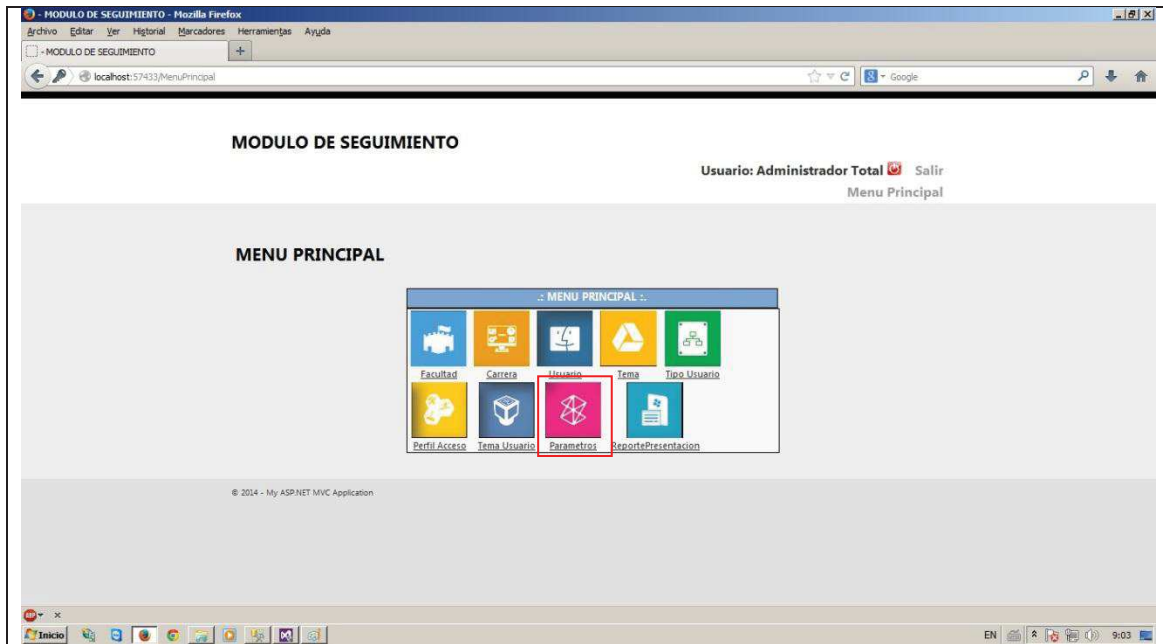


Figura 134. Opción Parámetros

Esta opción nos permitirá editar los diferentes parámetros del sistema:

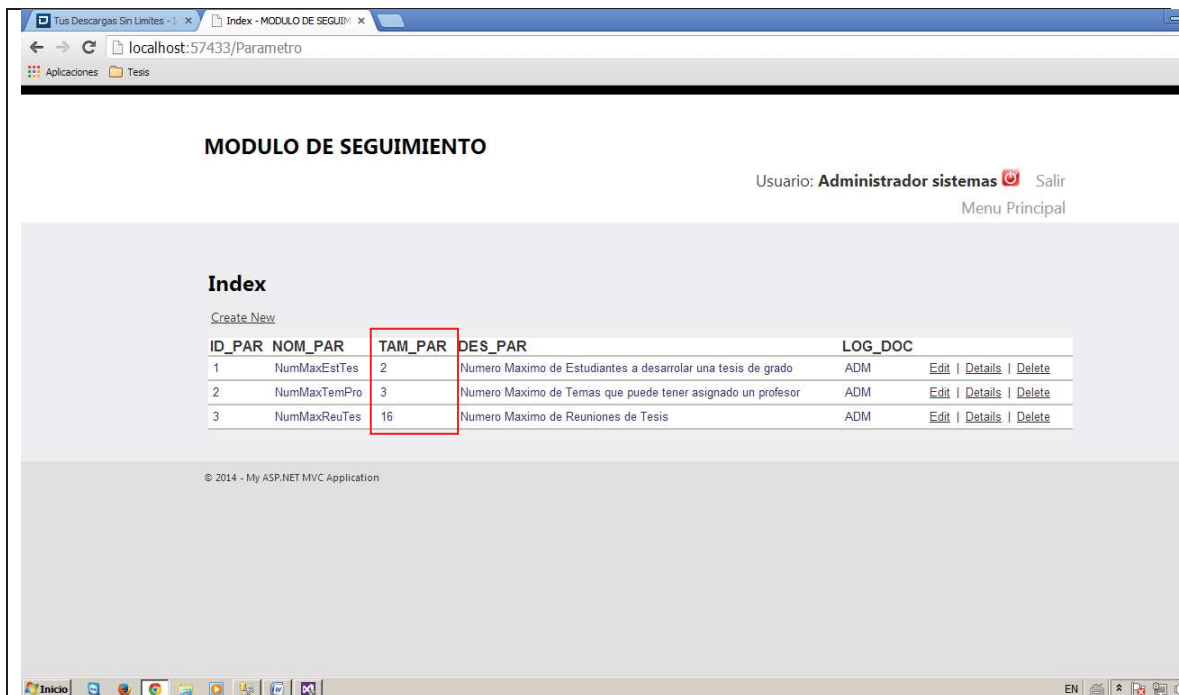


Figura 135. Pantalla de parámetros

En caso de editar alguno de estos parámetros solo de debe realizar el cambio del Tamaño del parámetro, el resto de campos no deben ser editados

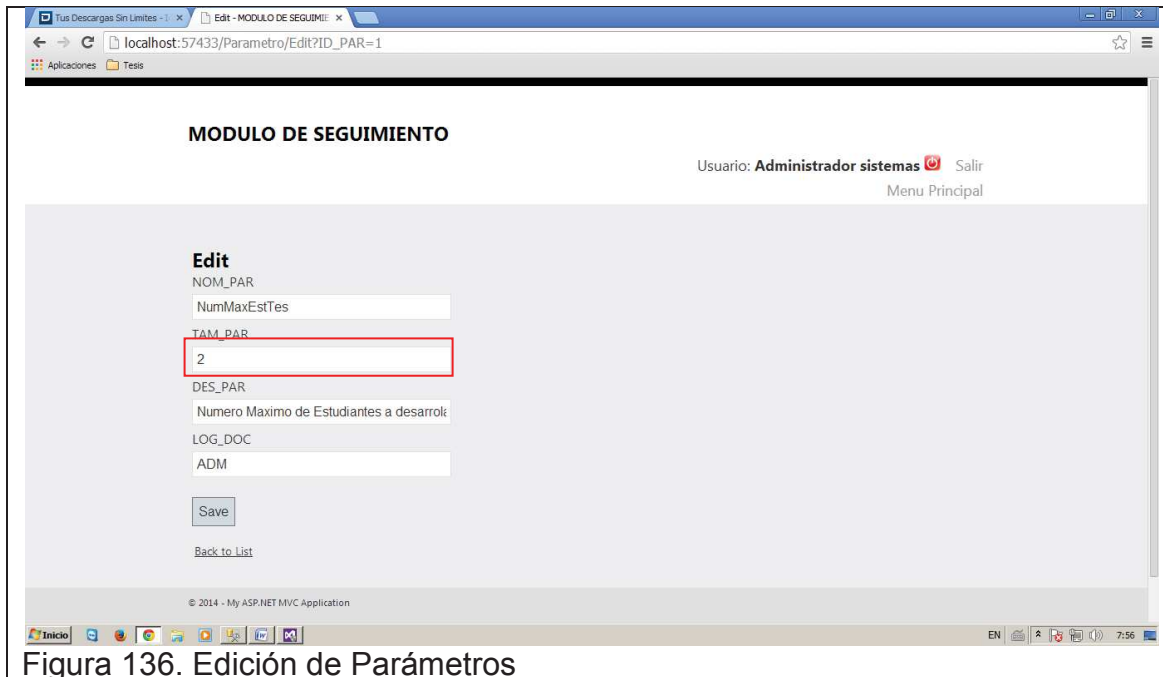


Figura 136. Edición de Parámetros

## REPORTE PRESENTACIÓN

Esta opción nos sirve para generar los diferentes reportes del sistema y permite exportarlos

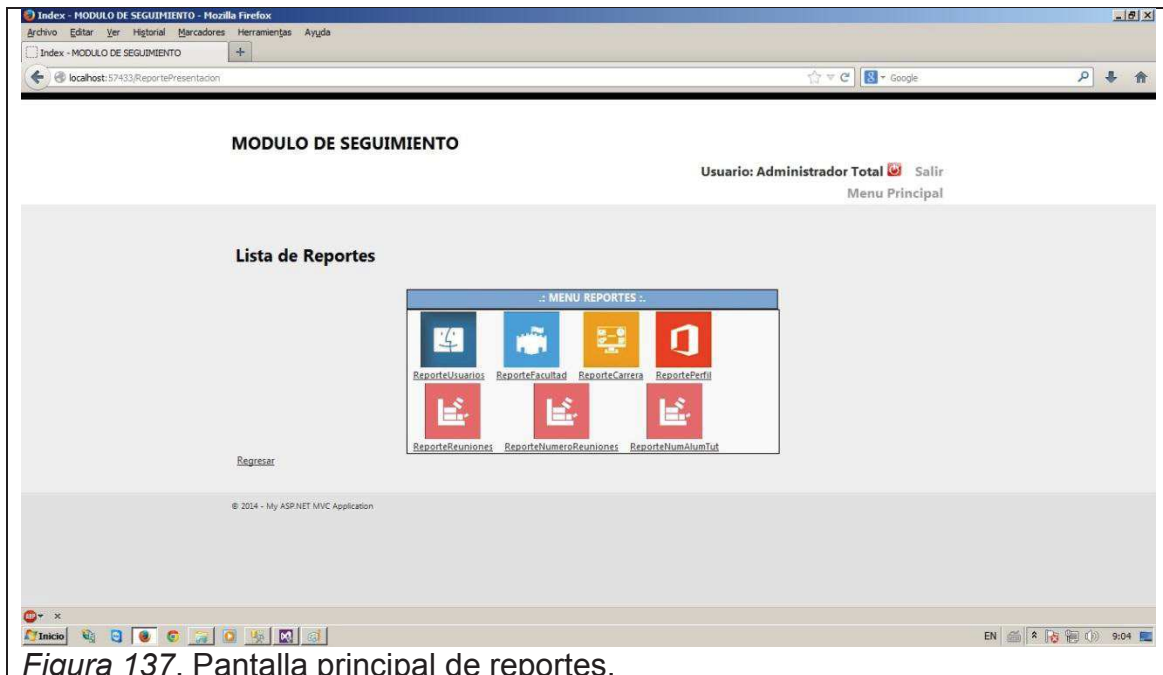


Figura 137. Pantalla principal de reportes.

Se selecciona el reporte a generar y aparece la siguiente pantalla con los datos requeridos

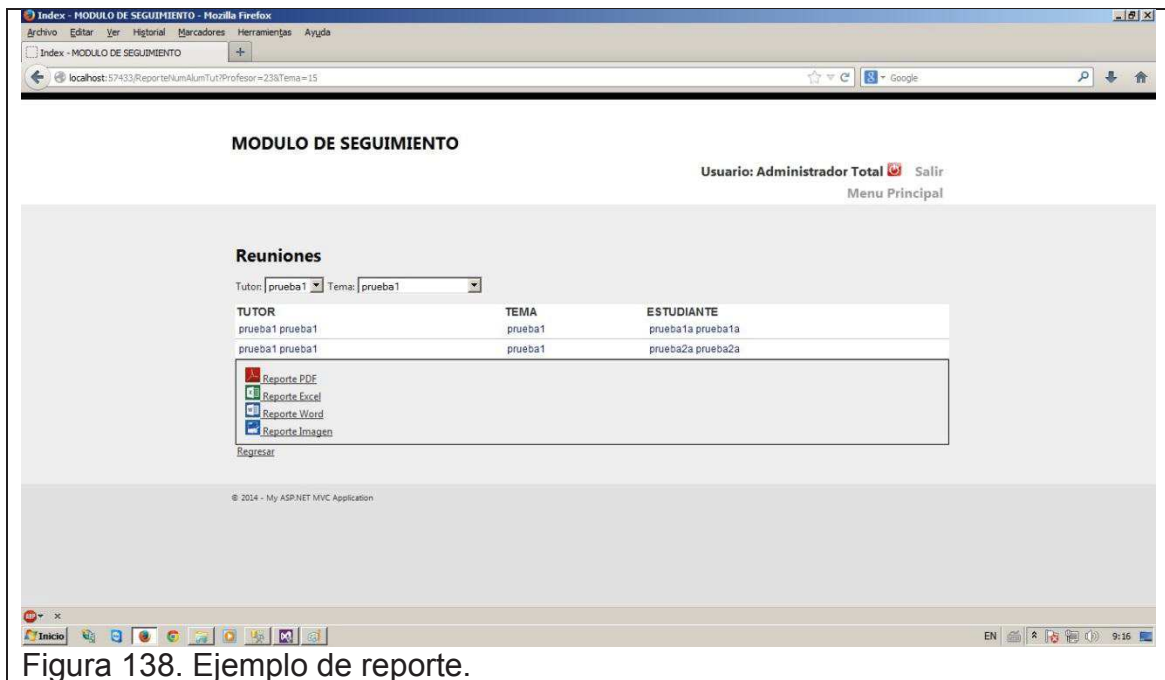


Figura 138. Ejemplo de reporte.

En esta pantalla se filtra los datos ya sea por tutor o por tema de tesis y también se puede exportar en los formatos mostrados.

## MENÚ TIPO USUARIO

Esta opción nos sirve para crear los tipos de Usuario que pueden ingresar al sistema para poder identificarlos de manera general




Figura 139. Menú TipoUsiario

Esta opción permite realizar las siguientes acciones:

### *Crear un nuevo Tipo Usuario*

Al dar clic en esta acción podremos ingresar un nuevo tipo usuario, procedemos a ingresar la información del usuario, esta es:



The screenshot shows a web browser window with the address bar displaying 'localhost:57433/TipoUsuario/Create'. The page content includes a header with the title 'MODULO DE SEGUIMIENTO' and user information 'Usuario: Administrador Total' with a 'Salir' button and a 'Menu Principal' link. The main content area is titled 'Crear Tipo Usuario' and contains three input fields: 'NOM\_TIP\_USU', 'DES\_TIP\_USU', and 'LOG\_TIP\_USU'. Below the fields is a 'Crear' button and a 'Regresar' link.

*Figura 140. Crear un Tipo de Usuario*

- **NOM\_TIP\_USU:** Se refiere al nombre general que se le desea dar a este nuevo tipo de usuario.
- **DES\_TIP\_USU:** Se refiere a una descripción que se le desea dar al nuevo tipo de usuario.
- **LOG\_TIP\_USU:** Se refiere al log del creador del tipo usuario en este campo siempre debe ir **ADM**.

Para grabar el nuevo Tipo Usuario se debe dar clic en el botón Crear.

### *Actualizar un Tipo Usuario*

Para poder actualizar un Tipo Usuario se debe seleccionar el siguiente botón de la derecha del tipo usuario, una vez seleccionado la información de este aparece, luego se procede a realizar las modificaciones pertinentes y finalmente se da clic en el botón Guardar.

### *Eliminar un Tipo Usuario*

Para poder eliminar un Tipo Usuario se debe seleccionar el siguiente botón de la derecha del tipo usuario y con esto se eliminara el tipo usuario

## MENÚ PERFIL

Esta opción nos sirve para crear perfiles de acuerdo al Tipo Usuario creado y determinar luego los accesos de cada uno de estos

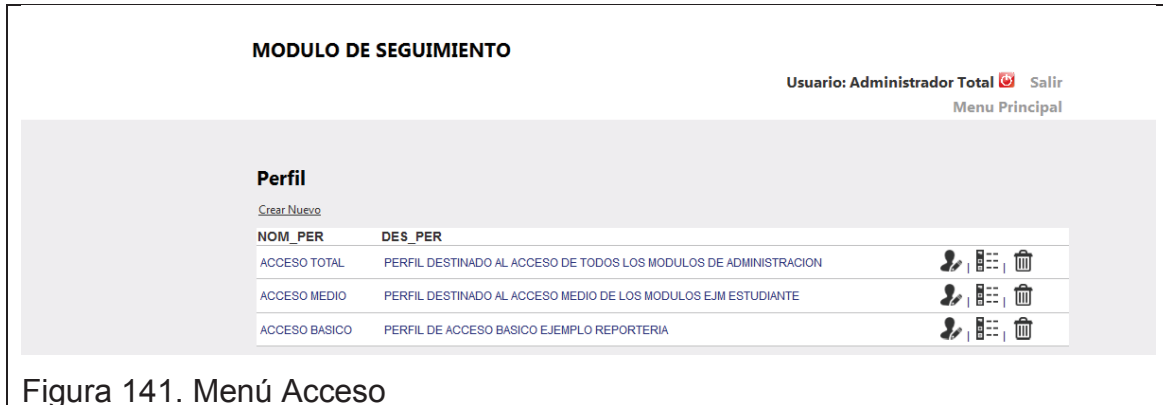


Figura 141. Menú Acceso

Esta opción permite realizar las siguientes acciones:

### *Crear un nuevo Perfil*

Al dar clic en esta acción podremos ingresar un nuevo perfil, procedemos a ingresar la información del perfil, esta es:



Figura 142. Crear Acceso

- NOMBRE: Se refiere al nombre que se le desea dar al perfil
- Descripción: Se refiere a una descripción que se le desee dar al nuevo perfil.

Para grabar el nuevo Perfil se debe dar clic en el botón Ingresar.

### Actualizar un Perfil

Para poder actualizar un Perfil se debe seleccionar el siguiente botón de la derecha del Perfil, una vez seleccionado la información de este aparece, luego se procede a realizar las modificaciones pertinentes y finalmente se da clic en el botón Guardar.

### Eliminar un Perfil

Para poder eliminar un Perfil se debe seleccionar el siguiente botón de la derecha del perfil y con esto se eliminara el perfil.

## MENU PERFIL ACCESO

Esta opción nos sirve para habilitar cada una de las opciones del sistema a determinados perfiles de acuerdo a la necesidad



**MODULO DE SEGUIMIENTO**

Usuario: **Administrador Total**  Salir  
Menu Principal

**Perfil Acceso**

[Create Nuevo](#)

| PERFIL       | ACCESO         |    |
|--------------|----------------|--|
| ACCESO TOTAL | ADMINISTRACION |    |
| ACCESO TOTAL | ADMINISTRACION |    |
| ACCESO TOTAL | Carrera        |    |
| ACCESO TOTAL | Estudiante     |    |
| ACCESO TOTAL | Facultad       |    |
| ACCESO TOTAL | Perfil         |    |

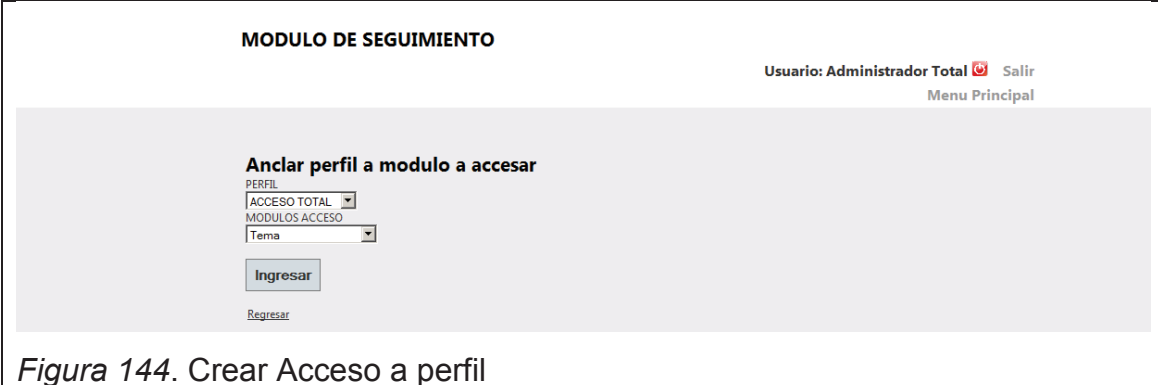
Figura 143. Menú Perfil Acceso

Esta opción permite realizar las siguientes acciones:



### *Crear un nuevo Perfil Acceso*

Al dar clic en esta acción podremos ingresar un nuevo perfil acceso, procedemos a ingresar la información del perfil acceso, esta es:



The screenshot shows a web interface titled "MODULO DE SEGUIMIENTO". In the top right corner, it displays "Usuario: Administrador Total" with a red lock icon and "Salir", and below it, "Menu Principal". The main content area is a light gray box with the heading "Anclar perfil a modulo a acceder". Below this heading are two dropdown menus: "PERFIL" with "ACCESO TOTAL" selected, and "MODULOS ACCESO" with "Tema" selected. There is a blue "Ingresar" button and a "Regresar" link below it.

*Figura 144. Crear Acceso a perfil*

- Perfil: Se refiere al perfil creado anteriormente.
- Modulo Acceso: Se refiere a las opciones que se desean habilitar para el perfil.

### *Actualizar un Perfil Acceso*

Para poder actualizar un Perfil Acceso se debe seleccionar el siguiente botón de la derecha del Perfil Acceso, una vez seleccionado la información de este aparece, luego se procede a realizar las modificaciones pertinentes y finalmente se da clic en el botón Guardar.

### *Eliminar un Perfil Acceso*

Para poder eliminar un Perfil Acceso se debe seleccionar el siguiente botón de la derecha del perfil y con esto se eliminara el perfil acceso.