

SISTEMA DE INFORMACION DE ASIGNACION Y CONTROL DE HORARIOS  
PARA LAS AULAS DE LA UNIVERSIDAD DE NARIÑO

ANDRES MAURICIO SANTACRUZ BURBANO

UNIVERSIDAD DE NARIÑO  
FACULTAD DE INGENIERIA  
PROGRAMA DE INGENIERIA DE SISTEMAS  
SAN JUAN DE PASTO  
2011

SISTEMA DE INFORMACION DE ASIGNACION Y CONTROL DE HORARIOS  
PARA LAS AULAS DE LA UNIVERSIDAD DE NARIÑO

ANDRES MAURICIO SANTACRUZ BURBANO

Trabajo de Grado presentado como requisito parcial  
para optar al título de Ingeniero de Sistemas

Asesor

Ing. Mg. Nelson Jaramillo Enríquez

UNIVERSIDAD DE NARIÑO  
FACULTAD DE INGENIERIA  
PROGRAMA DE INGENIERIA DE SISTEMAS  
SAN JUAN DE PASTO  
2011

NOTA DE ACEPTACION

---

---

---

---

---

Firma del Presidente del Jurado

---

Firma del Jurado

---

Firma del Jurado

San Juan de Pasto, 14 de septiembre de 2011

*“Las ideas y conclusiones aportadas en el Trabajo de Grado son responsabilidad exclusiva del autor.”*

*Artículo 1º del Acuerdo N°. 324 de octubre 11 de 1966, emanado del Honorable Consejo Directivo de la Universidad de Nariño.*

## CONTENIDO

	Pág.
INTRODUCCION .....	10
1. ANTECEDENTES .....	11
2. RECURSOS .....	13
2.1 RECURSOS HUMANOS .....	13
2.2 RECURSOS TECNOLÓGICOS .....	13
2.2.1 Hardware .....	13
2.2.2 Software.....	13
2.3 RECURSOS MATERIALES .....	14
2.4 RECURSOS FINANCIEROS .....	14
2.5 RECURSOS OPERATIVOS .....	14
3. MARCO TEORICO .....	15
3.1 SISTEMA DE INFORMACION .....	15
3.2 BASES DE DATOS .....	16
3.3 MICROSOFT VISUAL STUDIO .....	16
3.4 PROGRAMACION EXTREMA(EXTREME PROGRAMMING,XP) .....	18
3.4.1 Principios.....	18
3.4.2 Características Fundamentales .....	21
3.4.3 Fases .....	22

3.4.4	Historias de Usuario .....	24
3.4.5	Roles .....	24
3.5	ASP.NET .....	26
4.	METODOLOGIA .....	27
5.	DESARROLLO DE LA APLICACIÓN .....	29
5.1	HISTORIAS DE USUARIO Y TAREAS DE PROGRAMACION .....	30
5.2	CASOS DE USO DEL SISTEMA .....	47
5.2.1	Lista de actores .....	47
5.2.2	Casos de uso .....	48
5.3	DIAGRAMAS DE CASOS DE USO .....	62
5.3.1	Figura N° 1 .....	62
5.3.2	Figura N° 2 .....	63
5.3.3	Figura N° 3 .....	63
5.4	MODELO CONCEPTUAL .....	64
5.4.1	Figura N° 1 .....	64
5.4.2	Figura N° 2 .....	65
5.4.3	Figura N° 3 .....	66
5.4.4	Figura N° 4 .....	66
5.4.4	Figura N° 5 .....	67
5.5	DIAGRAMA DE CLASES .....	68
5.5.1	Figura N° 1 .....	68
5.5.2	Figura N° 2 .....	69
5.5.3	Figura N° 3 .....	69

5.6	DISEÑO DE LA BASE DE DATOS .....	70
5.6.1	Diagrama entidad-relación .....	71
5.6.2	Tablas de la base de datos .....	72
5.7	GLOSARIO .....	79
5.7.1	Casos de uso .....	79
5.7.2	Conceptos .....	81
6.	CONCLUSIONES .....	82
7.	RECOMENDACIONES .....	83
	BIBLIOGRAFIA .....	84

## **RESUMEN**

La Oficina de Planeación de la Universidad de Nariño, al comenzar cada periodo académico, realiza el proceso de asignación de franjas horarias académicas en las aulas de la universidad para los diferentes programas académicos.

Para ello se utilizan plantillas de Microsoft Excel lo que hace muy ineficiente la realización del proceso, además los recursos de hardware y software con los que cuenta la universidad no se han empleado para solucionar las dificultades presentadas y optimizar el proceso.

El presente trabajo consistió en tomar el proceso de asignación de horarios en las aulas y trasladarlo a una aplicación Web con la ayuda de la plataforma .Net de Microsoft, legalmente disponible en la universidad.

En este trabajo también se explica la metodología de desarrollo y los conceptos teóricos que se utilizaron en la construcción del software.

## **ABSTRACT**

The Office of Planning at the University of Nariño, the beginning of each academic period, makes the process of allocation of slots in academic halls of the university for academic programs.

This is done using Microsoft Excel templates which makes it very inefficient implementation of the process, and hardware resources and software with which the university has not been used to solve the difficulties presented and optimize the process.

The present work was to take the assignment process in the classroom schedule and remove to a Web application with the help of the platform. Net Microsoft, legally available in the university.

This paper also describes the development methodology and theoretical concepts that were used in the construction of software.

## INTRODUCCION

En la actualidad el manejo de la información ha pasado a ser un aspecto importante al momento de aumentar la eficiencia de las entidades en busca de alcanzar sus objetivos y metas. La adquisición de sistemas computacionales y el desarrollo de software en las entidades se han convertido en una excelente herramienta para administrar la información, con el fin de disminuir esfuerzos y ahorrar recursos.

Teniendo en cuenta lo anterior y las necesidades de la oficina de Planeación, y de la Universidad de Nariño en general, se propone el trabajo de grado -- Sistema de Información de asignación y control de horarios para las aulas de la Universidad de Nariño – en la modalidad aplicación, para solucionar las deficiencias y mejorar el procesamiento de la información.

El área de sistemas de la oficina de planeación y los secretarios académicos de cada facultad son los encargados de asignar los horarios para las diferentes facultades de la universidad teniendo en cuenta la disponibilidad de aulas en cada una de ellas, al comienzo de cada periodo académico. Este proceso se hace teniendo en cuenta la carga académica asignada para cada programa proveniente del sistema de información del centro de Informática. La introducción de estos datos se hace en una plantilla de Excel, lo que hace dispendioso la asignación de la información y dificulta la búsqueda de aulas disponibles para realizar otras actividades, así mismo se presentan errores de inserción de datos y cruces de horarios en las aulas. Debido a esto se planteó un sistema para que realice la asignación de aulas.

Se pretende que cada secretario de facultad sea el encargado de introducir la información referente a horarios de clase y que el área de sistemas de la oficina de Planeación sea la encargada de controlar dicha información y asignar aulas disponibles a quienes lo soliciten.

El sistema está formado por los siguientes módulos: módulo de manejo de aulas, modulo de información académica, módulo de asignación de espacios académicos, módulo de asignación de espacios extra clase y módulo de reportes.

Para la elaboración de la aplicación se utilizó Microsoft Visual Studio 2010 y el gestor de bases de datos PostgreSQL 9.0 para Windows.

## 1. ANTECEDENTES

La asignación de horarios en las aulas de la universidad no se ha determinado como un proceso general que involucre sistemas de información, sin embargo en el área de sistemas de la oficina de planeación se ha utilizado hojas y plantillas de Excel que por lo general son insuficientes para manejar los datos.

La información sobre aulas de las facultades y su disponibilidad se encuentra en archivos de Excel, con la cual se hace la asignación y control de horas a las diferentes materias. Por cada facultad y por cada programa se asigna un color con el cual se identifica que aula está ocupada o disponible a determinada hora. Las casillas en blanco representan las aulas en las que se puede asignar una actividad en una hora específica y que se llena por medio de un cuadro de texto.

La carencia de reportes obliga a que el encargado de manejar el proceso esté continuamente abriendo Excel para averiguar que aula esta libre, realizar la asignación en el espacio adecuado y posteriormente informar sobre la asignación a la persona interesada.

A nivel externo se han presentado trabajos de grado orientados a la asignación de horarios en las entidades universitarias utilizando algoritmos específicos y fórmulas matemáticas:

Por ejemplo, un trabajo presentado para optar por el título de magister del ingeniero José María Mejía de la Universidad del Norte en Barranquilla, el cual consistía en la aplicación de Algoritmos Evolutivos para la asignación de horarios de clases universitarias. En este trabajo se mostraban algunas técnicas de distribución de horarios a partir de timetabling<sup>1</sup> y un conjunto de pasos que incluían restricciones de tipo matemático.

Otro ejemplo, se encontró en un artículo publicado por Genaro Alfonso Ramón Rodríguez y María del Pilar Pozos Parra de la Universidad Juárez Autónoma de Tabasco, México, el cual además de las técnicas mencionadas anteriormente también hace referencia a la utilización de algoritmos de inteligencia artificial para la asignación de horarios no solo universitarios sino escolares.

---

<sup>1</sup> **Timetabling:** determina cuando una tarea o evento en particular debe ser llevado a cabo. Se utiliza diariamente en distintas áreas para regular diferentes operaciones donde sea necesaria una ubicación de una herramienta en un sitio determinado en una hora específica.

También cabe mencionar que la Universidad de Caldas, posee un Sistema Integrado de Gestión SIG el cual permite la actualización de espacios académicos en las aulas de la universidad por medio del internet, en particular, la utilización del correo electrónico.

A nivel local se destaca el trabajo de grado presentado por el Ing. Juan Pablo Gómez de la Universidad de Nariño, que consistía en un sistema de información web de registro y control para programación de cursos en el SENA, el cual aplicaba conceptos de programación web a un alto nivel incluyendo la utilización de bases de datos.

## **2. RECURSOS**

### **2.1 RECURSOS HUMANOS**

Según los roles descritos en la metodología Xp (Programación Extrema) se tiene lo siguiente:

- Ing. Diego Cruz: Encargado Área de sistemas de la oficina de Planeación. Gestor (Big Boss), Cliente
- Ing. Nelson Jaramillo (Asesor de Proyecto). Consultor
- Andrés Mauricio Santacruz estudiante Universidad de Nariño, desarrollador del proyecto. Programador, Tester.

### **2.2 RECURSOS TECNOLÓGICOS**

#### **2.2.1 Hardware:**

- Computador AMD Athlon™ 7750 Dual Core con 2Gb de memoria RAM, disco duro de 320 Gb SATA y quemador de DVD.
- Conexión a Internet
- Regulador de 100 Kw
- Memoria USB de 2Gb

#### **2.2.2 Software:**

- Sistema Operativo Windows 7 Ultimate SP
- Orígenes de datos ODBC
- Gestor de Base de Datos PostgreSQL™ 9.0
- Microsoft Visual Studio 2010™
- .Net Framework 4.0

### 2.3 RECURSOS MATERIALES

<b>Material</b>	<b>Detalle</b>	<b>Cantidad</b>
Papelería	Impresiones, Fotocopias y demás útiles de oficina	-
DVD	Proyecto final	1
Libros		2

### 2.4 RECURSOS FINANCIEROS

<b>Concepto</b>	<b>Costo</b>
Computador	\$ 1'100.000
Regulador	\$ 50.000
Internet	\$ 46.000
Papelería	\$ 300.000
DVD	\$ 1.500
Libros	\$ 100.000
<b>TOTAL</b>	<b>\$ 1'497.500</b>

### 2.5 RECURSOS OPERATIVOS

El sistema se implantará en el computador del encargado del área de sistemas de la oficina de planeación y en los equipos de los secretarios de facultad de la universidad. En un futuro se buscara la forma de montarlo en la página web de la universidad.

### 3. MARCO TEORICO

#### 3.1 SISTEMA DE INFORMACION <sup>2</sup>

Un **sistema de información** es un conjunto organizado de elementos, que pueden ser personas, datos, actividades o recursos materiales en general. Estos elementos interactúan entre sí para procesar información y distribuirla de manera adecuada en función de los objetivos de una organización.

El estudio de los sistemas de información surgió como una subdisciplina de las ciencias de la computación, con el objetivo de racionalizar la administración de la tecnología dentro de las organizaciones. El campo de estudio fue avanzando hasta pasar a ser parte de los estudios superiores dentro de la administración.

Desde un punto de vista empresarial, los sistemas de información pueden clasificarse de diversas formas. Existen, por ejemplo, **sistemas de procesamiento de transacciones** (que gestionan la información respecto a las transacciones producidas en una empresa), **sistemas de información gerencial** (para solucionar problemas empresariales en general), **sistemas de soporte a decisiones** (analizan las distintas variables de negocio para la toma de decisiones), **sistemas de información ejecutiva** (para los directivos), **sistemas de automatización de oficinas** (aplicaciones que ayudan en el trabajo administrativo) y **sistemas expertos** (que emulan el comportamiento de un especialista en un dominio concreto).

Cabe resaltar que el concepto de sistema de información suele ser utilizado como sinónimo de sistema de información informático, aunque no son lo mismo. Este último pertenece al campo de estudio de la tecnología de la información y puede formar parte de un sistema de información como recurso material. De todas formas, se dice que los sistemas de información tratan el desarrollo y la administración de la infraestructura tecnológica de una organización.

---

<sup>2</sup> **Sistema de Información.** Disponible en: <http://definicion.de/sistema-de-informacion/>. Consulta 14/03/2011.

### 3.2 BASES DE DATOS <sup>3</sup>

Se define una base de datos como una serie de datos organizados y relacionados entre sí, los cuales son recolectados y explotados por los sistemas de información de una empresa o negocio en particular

Desde el punto de vista informático, la base de datos es un sistema formado por un conjunto de datos almacenados en discos que permiten el acceso directo a ellos y un conjunto de programas que manipulen ese conjunto de datos.

Cada base de datos se compone de una o más tablas que guarda un conjunto de datos. Cada tabla tiene una o más **columnas** y **filas**. Las columnas guardan una parte de la información sobre cada elemento que queramos guardar en la tabla, cada fila de la tabla conforma un registro.

A los programas encargados de servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan se les llama **Sistemas Gestores de Bases de Datos (SGBD)**.

### 3.3 MICROSOFT VISUAL STUDIO <sup>4</sup>

**Microsoft Visual Studio** es un entorno de desarrollo integrado (IDE, por sus siglas en inglés) para sistemas operativos Windows. Soporta varios lenguajes de programación tales como Visual C++, Visual C#, Visual J#, ASP.NET y Visual Basic .NET, aunque actualmente se han desarrollado las extensiones necesarias para muchos otros.

Visual Studio permite a los desarrolladores crear aplicaciones, sitios y aplicaciones web, así como servicios web en cualquier entorno que soporte la plataforma .NET (a partir de la versión net 2002). Así se pueden crear aplicaciones que se intercomunican entre estaciones de trabajo, páginas web y dispositivos móviles.

---

<sup>3</sup> **Bases de Datos:** Disponible en: <http://www.maestrosdelweb.com/principiantes/%C2%BFque-son-las-bases-de-datos/>. Consulta 14/03/2011.

<sup>4</sup> **Microsoft Visual Studio:** Disponible en: [http://es.wikipedia.org/wiki/Microsoft\\_Visual\\_Studio](http://es.wikipedia.org/wiki/Microsoft_Visual_Studio). Consulta 14/03/2011.

A partir de la versión 2005 Microsoft ofrece gratuitamente las *Express Editions*. Estas son varias ediciones básicas separadas por lenguajes de programación o plataforma enfocadas para novatos y entusiastas. Estas ediciones son iguales al entorno de desarrollo comercial pero sin características avanzadas. Las ediciones que hay son:

- Visual Basic Express Edition
- Visual C# Express Edition
- Visual C++ Express Edition
- Visual J# Express Edition (Desapareció en Visual Studio 2008)
- Visual Web Developer Express Edition (para programar en ASP.NET)
- Visual F# (Apareció en Visual Studio 2010, es parecido al J#)

Adicionalmente, Microsoft ha puesto gratuitamente a disposición de todo el mundo una versión reducida de MS SQL Server llamada SQL Server Express Edition cuyas principales limitaciones son que no soporta bases de datos superiores a 4 GB de tamaño, únicamente utiliza un procesador y un Gb de RAM, y no cuenta con el Agente de SQL Server.

Visual Studio 2010 es la versión más reciente de esta herramienta, acompañada por .NET Framework 4.0. La fecha prevista para el lanzamiento de la versión final ha sido el 12 de abril de 2010.

Hasta ahora, uno de los mayores logros de la versión 2010 de Visual Studio ha sido el de incluir las herramientas para desarrollo de aplicaciones para Windows 7, tales como herramientas para el desarrollo de las características de Windows 7 (System.Windows.Shell) y la Ribbon Preview para WPF.

Entre sus más destacables características, se encuentran la capacidad para utilizar múltiples monitores, así como la posibilidad de desacoplar las ventanas de su sitio original y acoplarlas en otros sitios de la interfaz de trabajo. Además de esto, aparece una edición que compila las características de todas las ediciones comunes de Visual Studio: **Professional**, **Team Studio**, **Test**, conocida como **Visual Studio Ultimate**.

### 3.4 PROGRAMACION EXTREMA (EXTREME PROGRAMMING, XP) [1]

La programación extrema o XP es una metodología de desarrollo que se englobaría dentro de las denominadas metodologías ágiles en la que se da máxima prioridad a la obtención de resultados y reduce la burocracia que se produce al utilizar otras 'metodologías pesadas'.

Todo en el software cambia. Los requisitos cambian. El diseño cambia. El negocio cambia. La tecnología cambia. El equipo cambia. Los miembros del equipo cambian. El problema no es el cambio en sí mismo, puesto que se sabe que el cambio va a suceder; el problema es la incapacidad de adaptarse a dicho cambio cuando éste tiene lugar.

El autor de la XP es Kent Beck, entre otros, que con su larga experiencia como programador eligió las mejores características de las metodologías y profundizó en las relaciones de éstas y como se reforzaban unas a otras. Por tanto, la XP no se basa en principios nuevos, sino que todas, o casi todas, sus características ya se conocen dentro de la ingeniería del software, las cuales se complementan para minimizar los típicos problemas que pueden surgir en todo desarrollo de proyectos software.

#### 3.4.1. Principios [2]

**El principio de pruebas.** Se tiene que establecer un período de pruebas de aceptación del programa (llamado también *período de caja negra*) donde se definirán las entradas al sistema y los resultados esperados de estas entradas. Es muy recomendable automatizar estas pruebas para poder hacer varias simulaciones del sistema en funcionamiento. Para hacer estas simulaciones automatizadas, se pueden utilizar Ambientes de Prueba (*Unit testing frameworks*).

**Proceso de planificación.** En esta fase, el usuario tendrá que escribir sus necesidades, definiendo las actividades que realizará el sistema. Se creará un documento llamado Historias del usuario (*User Stories*). Entre 20 y 80 historias (todo dependiendo de la complejidad del problema) se consideran suficientes para formar el llamado Plan de Liberación, el cual define de forma específica los tiempos de entrega de la aplicación para recibir retroalimentación por parte del usuario. Por regla general, cada una de las Historias de usuario suelen necesitar de una a tres semanas de desarrollo.

Son muy importantes y tienen que ser una constante las reuniones periódicas durante esta fase de planificación. Estas pueden ser a diario, con todo el equipo de desarrollo para identificar problemas, proponer soluciones y señalar aquellos

puntos a los que se les ha de dar más importancia por su dificultad o por su punto crítico.

**Cliente In-situ. [3].** Un cliente real debe sentarse con el equipo de programadores, estar disponible para responder a sus preguntas, resolver discusiones y fijar las prioridades. Lo difícil es que el cliente nos confiera una persona que conozca a fondo el negocio para que se integre en el equipo. Estos elementos son muy valiosos, pero lo mejor para su negocio es tener un software pronto en funcionamiento, y esto no implica que el cliente no pueda realizar cualquier otro trabajo.

**Programación en parejas.** Uno de los principios más radicales y en el que la mayoría de gerentes de desarrollo pone sus dudas. Requiere que todos los programadores XP escriban su código en parejas, compartiendo una sola máquina. De acuerdo con los experimentos, este principio puede producir aplicaciones más buenas, de manera consistente, a iguales o menores costes.

**Integración continua.** Permite al equipo hacer un rápido progreso implementando las nuevas características del software. En lugar de crear *builds* (o versiones) estables de acuerdo a un cronograma establecido, los equipos de programadores XP pueden reunir su código y reconstruir el sistema varias veces al día. Esto reduce los problemas de integración comunes en proyectos largos y estilo cascada.

**Refactoring.** Permite a los equipos de programadores XP mejorar el diseño del sistema a través de todo el proceso de desarrollo. Los programadores evalúan continuamente el diseño y recodifican lo necesario. La finalidad es mantener un sistema enfocado a proveer el valor de negocio mediante la minimización del código duplicado y/o ineficiente.

**Entregas pequeñas.** Colocan un sistema sencillo en producción rápidamente que se actualiza de forma rápida y constante permitiendo que el verdadero valor de negocio del producto sea evaluado en un ambiente real. Estas entregas no pueden pasar las 2 o 3 semanas como máximo.

**Diseño simple.** Se basa en la filosofía de que el mayor valor de negocio es entregado por el programa más sencillo que cumpla los requerimientos. *Simple Design* se enfoca en proporcionar un sistema que cubra las necesidades inmediatas del cliente, ni más ni menos. Este proceso permite eliminar redundancias y rejuvenecer los diseños obsoletos de forma sencilla.

**Metáfora.** Desarrollada por los programadores al inicio del proyecto, define una historia de cómo funciona el sistema completo. XP estimula historias, que son breves descripciones de un trabajo de un sistema en lugar de los tradicionales diagramas y modelos UML (*Unified Modeling Language*). La metáfora expresa la visión evolutiva del proyecto que define el alcance y propósito del sistema.

Las tarjetas CRC (Clase, Responsabilidad y Colaboración) también ayudarán al equipo a definir actividades durante el diseño del sistema. Cada tarjeta representa una clase en la programación orientada a objetos y define sus responsabilidades (lo que ha de hacer) y las colaboraciones con las otras clases (cómo se comunica con ellas).

**Propiedad colectiva del código.** Un código con propiedad compartida. Nadie es el propietario de nada, todos son el propietario de todo. Este método difiere en mucho a los métodos tradicionales en los que un simple programador posee un conjunto de código. Los defensores de XP argumentan que mientras haya más gente trabajando en una pieza, menos errores aparecerán.

**Estándar de codificación.** Define la propiedad del código compartido así como las reglas para escribir y documentar el código y la comunicación entre diferentes piezas de código desarrolladas por diferentes equipos. Los programadores las han de seguir de tal manera que el código en el sistema se vea como si hubiera estado escrito por una sola persona.

### 3.4.2 Características Fundamentales [2]

Las características fundamentales del método son:

- **Desarrollo iterativo e incremental:** pequeñas mejoras, unas tras otras.
- **Pruebas unitarias continuas**, frecuentemente repetidas y automatizadas, incluyendo pruebas de regresión. Se aconseja escribir el código de la prueba antes de la codificación.
- **Programación en parejas:** se recomienda que las tareas de desarrollo se lleven a cabo por dos personas en una misma posición. Se supone que la mayor calidad del código escrito de esta manera – el código es revisado y discutido mientras se escribe- -es más importante que la posible pérdida de productividad inmediata.
- **Frecuente integración del equipo de programación con el cliente** o usuario. Se recomienda que un representante del cliente trabaje junto al equipo de desarrollo.
- **Corrección de todos los errores** antes de añadir nueva funcionalidad. Hacer entregas frecuentes.
- **Refactoring del código**, es decir, reescribir ciertas partes del código para aumentar su legibilidad y mantenibilidad pero sin modificar su comportamiento. Las pruebas han de garantizar que en el *refactoring* no se ha introducido ningún fallo.
- **Propiedad del código compartida:** en vez de dividir la responsabilidad en el desarrollo de cada módulo en grupos de trabajo distintos, este método promueve el que todo el personal pueda corregir y extender cualquier parte del proyecto. Las frecuentes pruebas de regresión garantizan que los posibles errores serán detectados.
- **Simplicidad en el código:** es la mejor manera de que las cosas funcionen. La programación extrema apuesta que es más sencillo hacer algo simple y

tener un poco de trabajo extra para cambiarlo si se requiere, que realizar algo complicado y quizás nunca utilizarlo.

La simplicidad y la comunicación son extraordinariamente complementarias. Con más comunicación resulta más fácil identificar qué se debe y qué no se debe hacer. Mientras más simple es el sistema, menos tendrá que comunicar sobre este, lo que lleva a una comunicación más completa, especialmente si se puede reducir el equipo de programadores.

### 3.4.3 Fases [2]

**Fase I: Exploración.** En esta fase, los clientes plantean a grandes rasgos las historias de usuario que son de interés para la primera entrega del producto. Al mismo tiempo el equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas que se utilizarán en el proyecto. Se prueba la tecnología y se exploran las posibilidades de la arquitectura del sistema construyendo un prototipo. La fase de exploración toma de pocas semanas a pocos meses, dependiendo del tamaño y familiaridad que tengan los programadores con la tecnología.

#### **Fase II: Planeamiento o Planificación de la Entrega (*Release*)**

En esta fase el cliente establece la prioridad de cada historia de usuario, y correspondientemente, los programadores realizan una estimación del esfuerzo necesario de cada una de ellas. Se toman acuerdos sobre el contenido de la primera entrega y se determina un cronograma en conjunto con el cliente. Una entrega debería obtenerse en no más de tres meses. Esta fase dura unos pocos días. Las estimaciones de esfuerzo asociado a la implementación de las historias la establecen los programadores utilizando como medida el punto.

Un punto, equivale a una semana ideal de programación. Las historias generalmente valen de 1 a 3 puntos. Por otra parte, el equipo de desarrollo mantiene un registro de la “velocidad” de desarrollo, establecida en puntos por iteración, basándose principalmente en la suma de puntos correspondientes a las historias de usuario que fueron terminadas en la última iteración. La planificación se puede realizar basándose en el tiempo o el alcance. La velocidad del proyecto es utilizada para establecer cuántas historias se pueden implementar antes de una

fecha determinada o cuánto tiempo tomará implementar un conjunto de historias. Al planificar por tiempo, se multiplica el número de iteraciones por la velocidad del proyecto, determinándose cuántos puntos se pueden completar. Al planificar según alcance del sistema, se divide la suma de puntos de las historias de usuario seleccionadas entre la velocidad del proyecto, obteniendo el número de iteraciones necesarias para su implementación

**Fase III: Iteraciones.** Esta fase incluye varias iteraciones sobre el sistema antes de ser entregado. El Plan de Entrega está compuesto por iteraciones de no más de tres semanas. En la primera iteración se puede intentar establecer una arquitectura del sistema que pueda ser utilizada durante el resto del proyecto. Esto se logra escogiendo las historias que fueren la creación de esta arquitectura, sin embargo, esto no siempre es posible ya que es el cliente quien decide qué historias se implementarán en cada iteración (para maximizar el valor de negocio). Al final de la última iteración el sistema estará listo para entrar en producción. Los elementos que deben tomarse en cuenta durante la elaboración del Plan de la Iteración son: historias de usuario no abordadas, velocidad del proyecto, pruebas de aceptación no superadas en la iteración anterior y tareas no terminadas en la iteración anterior. Todo el trabajo de la iteración es expresado en tareas de programación, cada una de ellas es asignada a un programador como responsable, pero llevadas a cabo por parejas de programadores.

**Fase IV: Producción.** La fase de producción requiere de pruebas adicionales y revisiones de rendimiento antes de que el sistema sea trasladado al entorno del cliente. Al mismo tiempo, se deben tomar decisiones sobre la inclusión de nuevas características a la versión actual, debido a cambios durante esta fase. Es posible que se rebaje el tiempo que toma cada iteración, de tres a una semana. Las ideas que han sido propuestas y las sugerencias son documentadas para su posterior implementación (por ejemplo, durante la fase de mantenimiento).

**Fase V: Mantenimiento.** Mientras la primera versión se encuentra en producción, el proyecto XP debe mantener el sistema en funcionamiento al mismo tiempo que desarrolla nuevas iteraciones. Para realizar esto se requiere de tareas de soporte para el cliente. De esta forma, la velocidad de desarrollo puede bajar después de la puesta del sistema en producción. La fase de mantenimiento puede requerir nuevo personal dentro del equipo y cambios en su estructura.

**Fase VI: Muerte del proyecto.** Es cuando el cliente no tiene más historias para ser incluidas en el sistema. Esto requiere que se satisfagan las necesidades del cliente en otros aspectos como rendimiento y confiabilidad del sistema. Se genera la documentación final del sistema y no se realizan más cambios en la arquitectura. La muerte del proyecto también ocurre cuando el sistema no genera los beneficios esperados por el cliente o cuando no hay presupuesto para mantenerlo.

**3.4.4 Historias de Usuario.** El primer paso de cualquier proyecto que siga la metodología X.P es definir las historias de usuario con el cliente. Las historias de usuario tienen la misma finalidad que los casos de uso pero con algunas diferencias: Constan de 3 ó 4 líneas escritas por el cliente en un lenguaje no técnico sin hacer mucho hincapié en los detalles; no se debe hablar ni de posibles algoritmos para su implementación ni de diseños de base de datos adecuados, etc. Son usadas para estimar tiempos de desarrollo de la parte de la aplicación que describen. También se utilizan en la fase de pruebas, para verificar si el programa cumple con lo que especifica la historia de usuario. Cuando llega la hora de implementar una historia de usuario, el cliente y los desarrolladores se reúnen para concretar y detallar lo que tiene que hacer dicha historia. El tiempo de desarrollo ideal para una historia de usuario es entre 1 y 3 semanas.

#### 3.4.5 Roles [2]

- **Programador:** El programador escribe las pruebas unitarias y produce el código del sistema. Debe existir una comunicación y coordinación adecuada entre los programadores y otros miembros del equipo.
- **Cliente:** El cliente escribe las historias de usuario y las pruebas funcionales para validar su implementación. Además, asigna la prioridad a las historias de usuario y decide cuáles se implementan en cada iteración centrándose en aportar mayor valor al negocio. El cliente es sólo uno dentro del proyecto pero puede corresponder a un interlocutor que está representando a varias personas que se verán afectadas por el sistema.
- **Encargado de pruebas (Tester):** El encargado de pruebas ayuda al cliente a escribir las pruebas funcionales. Ejecuta las pruebas regularmente,

difunde los resultados en el equipo y es responsable de las herramientas de soporte para pruebas.

- **Encargado de seguimiento (*Tracker*):** El encargado de seguimiento proporciona realimentación al equipo en el proceso XP. Su responsabilidad es verificar el grado de acierto entre las estimaciones realizadas y el tiempo real dedicado, comunicando los resultados para mejorar futuras estimaciones.  
También realiza el seguimiento del progreso de cada iteración y evalúa si los objetivos son alcanzables con las restricciones de tiempo y recursos presentes. Determina cuándo es necesario realizar algún cambio para lograr los objetivos de cada iteración.
- **Entrenador (*Coach*):** Es responsable del proceso global. Es necesario que conozca a fondo el proceso XP para proveer guías a los miembros del equipo de forma que se apliquen las prácticas XP y se siga el proceso correctamente.
- **Consultor:** Es un miembro externo del equipo con un conocimiento específico en algún tema necesario para el proyecto. Guía al equipo para resolver un problema específico.
- **Gestor (*Big boss*):** Es el vínculo entre clientes y programadores, ayuda a que el equipo trabaje efectivamente creando las condiciones adecuadas. Su labor esencial es de coordinación.

### 3.5 ASP.NET<sup>5</sup>

Microsoft introdujo esta tecnología llamada Active Server Pages en diciembre de 1996, por lo que no es nada nueva. Es parte del Internet Information Server (IIS) desde la versión 3.0 y es una tecnología de páginas activas que permite el uso de diferentes scripts y componentes en conjunto con el tradicional HTML para mostrar páginas generadas dinámicamente, traduciendo la definición de **Microsoft**. *“Las Active Server Pages son un ambiente de aplicación abierto y gratuito en el que se puede combinar código HTML, scripts y componentes ActiveX del servidor para crear soluciones dinámicas y poderosas para el web”.*

El principio de la tecnología ASP es el VBScript, pero existe otra diversidad de lenguajes de programación que pueden ser utilizados como lo es Perl, JScript, etc.

El ASP es una tecnología dinámica funcionando del lado del servidor, lo que significa que cuando el usuario solicita un documento ASP, las instrucciones de programación dentro del script son ejecutadas para enviar al navegador únicamente el código HTML resultante. La ventaja principal de las tecnologías dependientes del servidor radica en la seguridad que tiene el programador sobre su código, ya que éste se encuentra únicamente en los archivos del servidor que al ser solicitado a través del web, es ejecutado, por lo que los usuario no tienen acceso más que a la página resultante en su navegador.

---

<sup>5</sup> Asp.Net: Disponible en: <http://www.maestrosdelweb.com/editorial/aspintro/>. Consulta 19/08/20

## 4. METODOLOGIA

La metodología de desarrollo escogida para la construcción del sistema propuesto, es la Programación Extrema (eXtreme Programming o Xp) debido a que se han encontrado aspectos importantes que se adecuan a las necesidades del proyecto. Uno de ellos y el más importante para esta metodología es mejorar las relaciones interpersonales como clave para el desarrollo de software, promoviendo el trabajo en equipo y el aprendizaje de los participantes en el proyecto, así como también hacer entregas periódicas de software funcional.

Para tal fin se cuenta con la colaboración del Ing. Diego Cruz encargado del área de sistemas de la oficina de planeación.

La creación del sistema web se realizará en las siguientes etapas:

- **Observación del estado actual y recolección de datos:**

En esta etapa se conocerá la situación actual de la asignación y control de horarios, se observaran los posibles errores y dificultades generadas en la ejecución de este proceso y por último se tomaran los datos necesarios para la construcción del software provenientes de la Oficina de Planeación de la Universidad.

- **Análisis de información recolectada:**

En esta etapa se analizará la información obtenida en la primera para verificar su validez, prioridad y su grado de importancia para empezar a estructurar los módulos que conforman el sistema.

- **Estructuración del sistema:**

En esta etapa se esquematizará el desarrollo del sistema web a través de la creación de los módulos que lo conforman con todos sus componentes de programación por medio del lenguaje de alto nivel .Net, así como también la validación y manejo de datos a través del gestor de bases de datos PostgreSQL.

- **Diseño e implementación del sistema:**

En esta etapa se diseñará una interfaz gráfica para el manejo del software amigable y de fácil uso por parte los usuarios, además se llevará a cabo la implementación del sistema para realizar las primeras pruebas de funcionamiento en un solo equipo.

- **Pruebas de funcionamiento del sistema:**

Aquí se realizaran las pruebas al sistema pero con la utilización de un entorno web, para verificar el funcionamiento del mismo, simulando una situación real de asignación y control de espacios académicos en las aulas, incluyendo la generación de reportes.

- **Optimización de código:**

En esta etapa se harán mejoras al código del programa para detectar posibles fallos en el sistema y corregirlos para un mejor funcionamiento.

- **Análisis de resultados:**

En donde se analizará los resultados obtenidos de las pruebas hechas al sistema después de mirar su comportamiento ante una situación real para determinar si se logro el objetivo principal.

- **Arreglos al sistema:**

Aquí se harán los últimos ajustes al sistema para su implantación.

- **Documentación:**

Esta etapa se realizará en el transcurso del proyecto con el fin de entregar al final del mismo un informe general de lo que se hizo para realizar el sistema.

## 5. DESARROLLO DE LA APLICACION

El sistema está constituido por los siguientes módulos:

**Módulo de Manejo de Aulas:** Este módulo permite controlar la información de las aulas de cada bloque de la universidad teniendo en cuenta su capacidad de estudiantes y sus características. Realiza los procesos de actualización de las aulas ya existentes y de inserción de nuevos bloques y aulas.

**Módulo de Información Académica:** Este módulo permite cargar la información de profesores de un determinado programa académico que dictaran clase en el semestre con sus respectivas materias. Este proceso se realiza a través de archivos planos o de texto. Además tiene la opción de ingresar nuevos programas académicos.

**Módulo de Asignación de espacios académicos:** En este módulo se lleva a cabo el proceso de asignación de espacios académicos en las diferentes aulas de la universidad por cada programa académico.

**Módulo de asignación de espacios extraclase:** En este módulo se pueden asignar espacios en las aulas destinados a otras actividades como seminarios, cursos, etc.

**Módulo de reportes:** Se encarga de la construcción de los reportes de espacios académicos, espacios extraclase y de utilización de aulas.

Para la toma de requerimientos y construcción de los módulos del sistema se utilizo las historias de usuario, pertenecientes a la metodología Xp (Programación Extrema), además de la aplicación del UML con los casos de uso, el modelo conceptual y el diagrama de clases.

## 5.1 HISTORIAS DE USUARIO Y TAREAS DE PROGRAMACION

HISTORIA DE USUARIO	
<b>Numero:</b> 1	<b>Usuario:</b> Administrador del sistema y secretarios de Facultad.
<b>Nombre historia:</b> registro de usuarios	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Alta
<b>Puntos estimados:</b>	<b>Iteración asignada:</b> 1
<b>Programador responsable:</b> Andrés Mauricio Santacruz	
<b>Descripción:</b> Se determina los nombres y contraseñas de los usuarios que pueden acceder al sistema, además de ingresarlos en la base de datos en la tabla de usuarios.	
<b>Observaciones:</b>	

TAREA DE PROGRAMACIÓN DE LA HISTORIA DE USUARIO # 1	
<b>Numero de tarea:</b> 1	<b>Numero de historia:</b> 1
<b>Nombre de tarea:</b> Interfaz de acceso al sistema	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b>
<b>Fecha inicio:</b> -	<b>Fecha fin:</b> -
<b>Programador responsable:</b> Andrés Mauricio Santacruz	
<b>Descripción:</b> Diseñar una ventana que permita ingresar el nombre y la contraseña de un usuario.	
<b>Observaciones:</b>	

<b>TAREA DE PROGRAMACIÓN DE LA HISTORIA DE USUARIO # 1</b>	
<b>Numero de tarea:</b> 2	<b>Numero de historia:</b> 1
<b>Nombre de tarea:</b> Manejo de operaciones de registro de usuarios	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b>
<b>Fecha inicio:</b> -	<b>Fecha fin:</b> -
<b>Programador responsable:</b> Andrés Mauricio Santacruz	
<b>Descripción:</b> Determinar las sentencias SQL y la codificación para consultar los usuarios con sus respectivas contraseñas disponibles en la base de datos.	
<b>Observaciones:</b>	

<b>HISTORIA DE USUARIO</b>	
<b>Numero:</b> 2	<b>Usuario:</b> Administrador del sistema
<b>Nombre historia:</b> manejo de aulas	
<b>Prioridad en negocio:</b> Media	<b>Riesgo en desarrollo:</b> Media
<b>Puntos estimados:</b>	<b>Iteración asignada:</b> 2
<b>Programador responsable:</b> Andrés Mauricio Santacruz	
<b>Descripción:</b> Se visualiza e ingresa la información correspondiente a las aulas de cada bloque de la universidad contenidas en la base de datos.	
<b>Observaciones:</b> <ul style="list-style-type: none"> <li>- Un bloque tiene un código y un nombre, además está formado por varias aulas.</li> <li>- Un aula tiene varias características como código, descripción, área, etc.</li> </ul>	

TAREA DE PROGRAMACIÓN DE LA HISTORIA DE USUARIO # 2	
<b>Numero de tarea:</b> 1	<b>Numero de historia:</b> 2
<b>Nombre de tarea:</b> Interfaz de visualización de información de las aulas.	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b>
<b>Fecha inicio:</b> -	<b>Fecha fin:</b> -
<b>Programador responsable:</b> Andrés Mauricio Santacruz	
<b>Descripción:</b> Diseñar una ventana que permita visualizar las características correspondientes a una determinada aula como son descripción, capacidad, área y disponibilidad.	
<b>Observaciones:</b>	

TAREA DE PROGRAMACIÓN DE LA HISTORIA DE USUARIO # 2	
<b>Numero de tarea:</b> 3	<b>Numero de historia:</b> 2
<b>Nombre de tarea:</b> Interfaz de inserción de nuevos bloques de aulas.	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b>
<b>Fecha inicio:</b> -	<b>Fecha fin:</b> -
<b>Programador responsable:</b> Andrés Mauricio Santacruz	
<b>Descripción:</b> Diseñar una ventana que permita ingresar nuevos bloques de aulas que se construyan en la Universidad.	
<b>Observaciones:</b>	

TAREA DE PROGRAMACIÓN DE LA HISTORIA DE USUARIO # 2	
<b>Numero de tarea:</b> 4	<b>Numero de historia:</b> 2
<b>Nombre de tarea:</b> Interfaz de inserción de nuevas aulas en cada bloque.	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b>
<b>Fecha inicio:</b> -	<b>Fecha fin:</b> -
<b>Programador responsable:</b> Andrés Mauricio Santacruz	
<b>Descripción:</b> Diseñar una ventana que permita ingresar las características de las nuevas aulas correspondientes a cada nuevo bloque.	
<b>Observaciones:</b>	

TAREA DE PROGRAMACIÓN DE LA HISTORIA DE USUARIO # 2	
<b>Numero de tarea:</b> 5	<b>Numero de historia:</b> 2
<b>Nombre de tarea:</b> Interfaz de eliminación de aulas.	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b>
<b>Fecha inicio:</b> -	<b>Fecha fin:</b> -
<b>Programador responsable:</b> Andrés Mauricio Santacruz	
<b>Descripción:</b> Diseñar una ventana que permita eliminar un aula que ya no se utilice de un bloque determinado.	
<b>Observaciones:</b>	

<b>TAREA DE PROGRAMACIÓN DE LA HISTORIA DE USUARIO # 2</b>	
<b>Numero de tarea:</b> 6	<b>Numero de historia:</b> 2
<b>Nombre de tarea:</b> Manejo de operaciones de visualización, actualización y eliminación de aulas, inserción de nuevos bloques y nuevas aulas.	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b>
<b>Fecha inicio:</b> -	<b>Fecha fin:</b> -
<b>Programador responsable:</b> Andrés Mauricio Santacruz	
<b>Descripción:</b> Determinar las sentencias SQL y la codificación para consultar, actualizar y eliminar las características de cada aula, así como también la adición de nuevos bloques y aulas con sus respectivas características.	
<b>Observaciones:</b>	

<b>HISTORIA DE USUARIO</b>	
<b>Numero:</b> 3	<b>Usuario:</b> Secretarios de Facultad
<b>Nombre historia:</b> Manejo de asignación de espacios académicos	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Alta
<b>Puntos estimados:</b>	<b>Iteración asignada:</b> 3
<b>Programador responsable:</b> Andrés Mauricio Santacruz	
<b>Descripción:</b> Se asigna espacios académicos para cada programa de la universidad teniendo en cuenta ciertos parámetros como el bloque, aula, semestre, hora inicio, hora fin, etc., y luego se ingresa dicha información en la base de datos.	
<b>Observaciones:</b> <ul style="list-style-type: none"> <li>- Espacio académico es una franja horaria de clase que se asigna en un aula de la Universidad para un programa determinado.</li> </ul>	

TAREA DE PROGRAMACIÓN DE LA HISTORIA DE USUARIO # 3	
<b>Numero de tarea:</b> 1	<b>Numero de historia:</b> 3
<b>Nombre de tarea:</b> Interfaz de elección de programa y semestre	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b>
<b>Fecha inicio:</b> -	<b>Fecha fin:</b> -
<b>Programador responsable:</b> Andrés Mauricio Santacruz	
<b>Descripción:</b> Diseñar una ventana que permita seleccionar la facultad, el programa y el semestre para la asignación de espacios académicos.	
<b>Observaciones:</b>	

TAREA DE PROGRAMACIÓN DE LA HISTORIA DE USUARIO # 3	
<b>Numero de tarea:</b> 2	<b>Numero de historia:</b> 3
<b>Nombre de tarea:</b> Interfaz de asignación de espacios académicos	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b>
<b>Fecha inicio:</b> -	<b>Fecha fin:</b> -
<b>Programador responsable:</b> Andrés Mauricio Santacruz	
<b>Descripción:</b> Diseñar una ventana que permita escoger el bloque, el aula, el día la hora de inicio, la hora final y el grupo; además de seleccionar el docente y la materia para asignar un determinado espacio académico.	
<b>Observaciones:</b>	

<b>TAREA DE PROGRAMACIÓN DE LA HISTORIA DE USUARIO # 3</b>	
<b>Numero de tarea:</b> 3	<b>Numero de historia:</b> 3
<b>Nombre de tarea:</b> Interfaz de visualización de espacios académicos	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b>
<b>Fecha inicio:</b> -	<b>Fecha fin:</b> -
<b>Programador responsable:</b> Andrés Mauricio Santacruz	
<b>Descripción:</b> Diseñar una ventana que permita mostrar los espacios que se han asignado por el momento en un determinado programa y en un determinado semestre.	
<b>Observaciones:</b>	

<b>TAREA DE PROGRAMACIÓN DE LA HISTORIA DE USUARIO # 3</b>	
<b>Numero de tarea:</b> 4	<b>Numero de historia:</b> 3
<b>Nombre de tarea:</b> Validación de datos en la asignación de espacios académicos	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b>
<b>Fecha inicio:</b> -	<b>Fecha fin:</b> -
<b>Programador responsable:</b> Andrés Mauricio Santacruz	
<b>Descripción:</b> Validar la información de cada espacio académico ingresado para que no se asigne a otro programa evitando así el cruce de horarios en las aulas. Además se debe validar que el docente asignado a una determinada hora no se asigne en la misma hora en otro programa diferente.	
<b>Observaciones:</b>	

<b>TAREA DE PROGRAMACIÓN DE LA HISTORIA DE USUARIO # 3</b>	
<b>Numero de tarea:</b> 5	<b>Numero de historia:</b> 3
<b>Nombre de tarea:</b> Interfaz de eliminación de espacios académicos	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b>
<b>Fecha inicio:</b> -	<b>Fecha fin:</b> -
<b>Programador responsable:</b> Andrés Mauricio Santacruz	
<b>Descripción:</b> Diseñar una ventana que permita eliminar un espacio académico que se asigno por error o que ya no se vaya a utilizar.	
<b>Observaciones:</b>	

<b>TAREA DE PROGRAMACIÓN DE LA HISTORIA DE USUARIO # 3</b>	
<b>Numero de tarea:</b> 6	<b>Numero de historia:</b> 3
<b>Nombre de tarea:</b> Manejo de operaciones de visualización, adición y eliminación de espacios académicos.	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b>
<b>Fecha inicio:</b> -	<b>Fecha fin:</b> -
<b>Programador responsable:</b> Andrés Mauricio Santacruz	
<b>Descripción:</b> Determinar las sentencias SQL y la codificación para consultar, adicionar y eliminar los espacios académicos para cada programa de la universidad, teniendo en cuenta sus respectivas características.	
<b>Observaciones:</b>	

<b>HISTORIA DE USUARIO</b>	
<b>Numero:</b> 4	<b>Usuario:</b> Administrador del Sistema.
<b>Nombre historia:</b> Inserción de nuevos programas académicos.	
<b>Prioridad en negocio:</b> Baja	<b>Riesgo en desarrollo:</b> Baja
<b>Puntos estimados:</b>	<b>Iteración asignada:</b> 4
<b>Programador responsable:</b> Andrés Mauricio Santacruz	
<b>Descripción:</b> Se ingresa la información de nuevos programas académicos al sistema.	
<b>Observaciones:</b>	

<b>TAREA DE PROGRAMACIÓN DE LA HISTORIA DE USUARIO # 4</b>	
<b>Numero de tarea:</b> 1	<b>Numero de historia:</b> 4
<b>Nombre de tarea:</b> Interfaz de adición de nuevos programas académicos.	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b>
<b>Fecha inicio:</b> -	<b>Fecha fin:</b> -
<b>Programador responsable:</b> Andrés Mauricio Santacruz	
<b>Descripción:</b> Diseñar una ventana que permita ingresar nuevos programas académicos que se ofrezcan en la Universidad.	
<b>Observaciones:</b>	

TAREA DE PROGRAMACIÓN DE LA HISTORIA DE USUARIO # 4	
<b>Numero de tarea:</b> 2	<b>Numero de historia:</b> 4
<b>Nombre de tarea:</b> Manejo de operaciones de adición de programas académicos.	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b>
<b>Fecha inicio:</b> -	<b>Fecha fin:</b> -
<b>Programador responsable:</b> Andrés Mauricio Santacruz	
<b>Descripción:</b> Determinar las sentencias SQL y la codificación para adicionar nuevos programas académicos que se creen en la Universidad en la base de datos.	
<b>Observaciones:</b>	

HISTORIA DE USUARIO	
<b>Numero:</b> 5	<b>Usuario:</b> Secretarios de Facultad.
<b>Nombre historia:</b> Inserción de Información Académica.	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Alta
<b>Puntos estimados:</b>	<b>Iteración asignada:</b> 5
<b>Programador responsable:</b> Andrés Mauricio Santacruz	
<b>Descripción:</b> Se ingresa la información correspondiente a profesores y materias de cada programa para asignar los espacios académicos.	
<b>Observaciones:</b>	

<b>TAREA DE PROGRAMACIÓN DE LA HISTORIA DE USUARIO # 5</b>	
<b>Numero de tarea:</b> 1	<b>Numero de historia:</b> 5
<b>Nombre de tarea:</b> Interfaz de ingreso de información académica al sistema.	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b>
<b>Fecha inicio:</b> -	<b>Fecha fin:</b> -
<b>Programador responsable:</b> Andrés Mauricio Santacruz	
<b>Descripción:</b> Diseñar una ventana que permita ingresar los datos correspondientes a profesores y materias de cada programa a la base de datos provenientes de archivos planos (.txt).	
<b>Observaciones:</b>	

<b>TAREA DE PROGRAMACIÓN DE LA HISTORIA DE USUARIO # 5</b>	
<b>Numero de tarea:</b> 2	<b>Numero de historia:</b> 5
<b>Nombre de tarea:</b> Manejo de operaciones de inserción de información académica al sistema.	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b>
<b>Fecha inicio:</b> -	<b>Fecha fin:</b> -
<b>Programador responsable:</b> Andrés Mauricio Santacruz	
<b>Descripción:</b> Determinar las sentencias SQL y la codificación para cargar en la base de datos la información correspondiente a los profesores de cada programa y sus respectivas materias.	
<b>Observaciones:</b>	

<b>HISTORIA DE USUARIO</b>	
<b>Numero:</b> 6	<b>Usuario:</b> Administrador del sistema
<b>Nombre historia:</b> Manejo de asignación de espacios extraclase.	
<b>Prioridad en negocio:</b> Baja	<b>Riesgo en desarrollo:</b> Bajo
<b>Puntos estimados:</b>	<b>Iteración asignada:</b> 6
<b>Programador responsable:</b> Andrés Mauricio Santacruz	
<b>Descripción:</b> Se asigna espacios extracurriculares para ciertas actividades programadas por la oficina de planeación que se realicen en una determinada aula de la universidad, que no son de carácter académico.	
<b>Observaciones:</b> <ul style="list-style-type: none"> <li>- Espacio extraclase es una franja horaria que se asigna en un aula de la Universidad para un actividad específica como seminarios, cursos, etc.</li> </ul>	

<b>TAREA DE PROGRAMACIÓN DE LA HISTORIA DE USUARIO # 6</b>	
<b>Numero de tarea:</b> 1	<b>Numero de historia:</b> 6
<b>Nombre de tarea:</b> Interfaz de asignación de espacios extraclase.	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b>
<b>Fecha inicio:</b> -	<b>Fecha fin:</b> -
<b>Programador responsable:</b> Andrés Mauricio Santacruz	
<b>Descripción:</b> Diseñar una ventana que permita ingresar el bloque, el aula, la hora inicio, la hora fin, la fecha, la actividad y el responsable para asignar un espacio extracurricular.	
<b>Observaciones:</b>	

<b>TAREA DE PROGRAMACIÓN DE LA HISTORIA DE USUARIO # 6</b>	
<b>Numero de tarea:</b> 2	<b>Numero de historia:</b> 6
<b>Nombre de tarea:</b> Interfaz de eliminación de espacios extra clase.	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b>
<b>Fecha inicio:</b> -	<b>Fecha fin:</b> -
<b>Programador responsable:</b> Andrés Mauricio Santacruz	
<b>Descripción:</b> Diseñar una ventana que permita eliminar un espacio extracurricular que se asignó por error o que ya no se vaya a utilizar.	
<b>Observaciones:</b>	

<b>TAREA DE PROGRAMACIÓN DE LA HISTORIA DE USUARIO # 6</b>	
<b>Numero de tarea:</b> 3	<b>Numero de historia:</b> 6
<b>Nombre de tarea:</b> Manejo de operaciones de adición y eliminación de espacios extra clase.	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b>
<b>Fecha inicio:</b> -	<b>Fecha fin:</b> -
<b>Programador responsable:</b> Andrés Mauricio Santacruz	
<b>Descripción:</b> Determinar las sentencias SQL y la codificación para adicionar y eliminar los espacios extracurriculares destinados a ciertas actividades que se programen en la oficina de planeación de la universidad.	
<b>Observaciones:</b>	

<b>HISTORIA DE USUARIO</b>	
<b>Numero:</b> 7	<b>Usuario:</b> Administrador del sistema y secretarios de Facultad.
<b>Nombre historia:</b> Manejo de reportes.	
<b>Prioridad en negocio:</b> Media	<b>Riesgo en desarrollo:</b> Medio
<b>Puntos estimados:</b>	<b>Iteración asignada:</b> 7
<b>Programador responsable:</b> Andrés Mauricio Santacruz	
<b>Descripción:</b> Se construyen los reportes correspondientes tanto de los espacios académicos asignados por programa y semestre, como de los espacios extraclase programados	
<b>Observaciones:</b>	

<b>TAREA DE PROGRAMACIÓN DE LA HISTORIA DE USUARIO # 7</b>	
<b>Numero de tarea:</b> 1	<b>Numero de historia:</b> 7
<b>Nombre de tarea:</b> Interfaz de elección de programa y semestre para generar reporte académico.	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b>
<b>Fecha inicio:</b> -	<b>Fecha fin:</b> -
<b>Programador responsable:</b> Andrés Mauricio Santacruz	
<b>Descripción:</b> Diseñar una ventana que permita seleccionar la facultad, el programa y el semestre para la construcción de un reporte académico determinado.	
<b>Observaciones:</b>	

TAREA DE PROGRAMACIÓN DE LA HISTORIA DE USUARIO # 7	
<b>Numero de tarea:</b> 2	<b>Numero de historia:</b> 7
<b>Nombre de tarea:</b> Interfaz de visualización del reporte de espacios académicos.	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b>
<b>Fecha inicio:</b> -	<b>Fecha fin:</b> -
<b>Programador responsable:</b> Andrés Mauricio Santacruz	
<b>Descripción:</b> Diseñar una ventana que permita mostrar el reporte de espacios académicos asignados en un determinado programa y semestre.	
<b>Observaciones:</b>	

TAREA DE PROGRAMACIÓN DE LA HISTORIA DE USUARIO # 7	
<b>Numero de tarea:</b> 3	<b>Numero de historia:</b> 7
<b>Nombre de tarea:</b> Interfaz de visualización del reporte de espacios extraclase.	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b>
<b>Fecha inicio:</b> -	<b>Fecha fin:</b> -
<b>Programador responsable:</b> Andrés Mauricio Santacruz	
<b>Descripción:</b> Diseñar una ventana que permita mostrar el reporte de espacios extracurriculares asignados para ciertas actividades programadas por la Oficina de Planeación de la Universidad.	
<b>Observaciones:</b>	

TAREA DE PROGRAMACIÓN DE LA HISTORIA DE USUARIO # 7	
<b>Numero de tarea:</b> 4	<b>Numero de historia:</b> 7
<b>Nombre de tarea:</b> Interfaz de elección de bloque y aula para generar reporte de utilización de aulas.	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b>
<b>Fecha inicio:</b> -	<b>Fecha fin:</b> -
<b>Programador responsable:</b> Andrés Mauricio Santacruz	
<b>Descripción:</b> Diseñar una ventana que permita seleccionar el bloque y un aula determinada para la construcción del reporte de utilización de aulas.	
<b>Observaciones:</b>	

TAREA DE PROGRAMACIÓN DE LA HISTORIA DE USUARIO # 7	
<b>Numero de tarea:</b> 5	<b>Numero de historia:</b> 7
<b>Nombre de tarea:</b> Interfaz de visualización del reporte de utilización de aulas.	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b>
<b>Fecha inicio:</b> -	<b>Fecha fin:</b> -
<b>Programador responsable:</b> Andrés Mauricio Santacruz	
<b>Descripción:</b> Diseñar una ventana que permita mostrar el reporte de utilización de aulas para un aula determinada.	
<b>Observaciones:</b>	

<b>TAREA DE PROGRAMACIÓN DE LA HISTORIA DE USUARIO # 7</b>	
<b>Numero de tarea:</b> 6	<b>Numero de historia:</b> 7
<b>Nombre de tarea:</b> Manejo de operaciones de visualización y creación de reportes de espacios académicos, espacios extra clase y utilización de aulas.	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b>
<b>Fecha inicio:</b> -	<b>Fecha fin:</b> -
<b>Programador responsable:</b> Andrés Mauricio Santacruz	
<b>Descripción:</b> Determinar las sentencias SQL y la codificación para construir los reportes de los espacios académicos asignados por programa y semestre, así como el reporte de los espacios extracurriculares asignados por la Oficina de Planeación de la Universidad y los reportes de utilización de aulas.	
<b>Observaciones:</b>	

## 5.2 CASOS DE USO DEL SISTEMA

### 5.2.1 Lista de actores

<b>Nombre:</b>	<b>Administrador del sistema</b>
<b>Descripción:</b>	Es la persona encargada de manejar lo referente a la asignación de espacios académicos y extraclase en las aulas de la universidad, así como de controlar dicha información. Tiene acceso a toda la funcionalidad del sistema.
<b>Objetivos:</b>	Maneja la información de aulas, administra la asignación de espacios académicos, crea espacios extraclase y maneja la información de reportes.

<b>Nombre:</b>	<b>Secretarios de Facultad</b>
<b>Descripción:</b>	Son las personas encargadas de asignar los espacios académicos para los programas que hacen parte de su facultad teniendo en cuenta la carga académica dada. Deben tener conocimientos sobre la programación de horarios.
<b>Objetivos:</b>	Tiene acceso restringido a los recursos del sistema, solo puede manejar la asignación de espacios académicos y la información de reportes.

## 5.2.2 Casos de uso

<b>CASO DE USO: Ingresar al sistema</b>
<b>ACTORES:</b> Administrador del sistema y secretarios de facultad
<b>PROPOSITO:</b> acceder a los módulos del sistema mediante un nombre y una contraseña.
<b>RESUMEN:</b> El Administrador del sistema o secretario de facultad ingresa al sistema.
<b>CURSO NORMAL DE LOS EVENTOS</b> <ol style="list-style-type: none"><li>1. El Administrador del sistema o secretario de facultad ingresa su nombre y contraseña y luego da click en el botón aceptar. El sistema muestra una ventana con 2 marcos, el marco superior que contiene el menú principal y el marco inferior en el cual se muestra la opción escogida en el menú.</li></ol>
<b>FLUJOS ALTERNATIVOS</b> <ol style="list-style-type: none"><li>1. El nombre de usuario o contraseña es incorrecto, el sistema muestra un error.</li></ol>
<b>PRECONDICIONES</b>
<b>POSTCONDICIONES</b> <ul style="list-style-type: none"><li>• En una ventana se tiene acceso a los diversos módulos del sistema.</li></ul>

<b>CASO DE USO: Manejar aulas</b>
<b>ACTORES:</b> Administrador del sistema
<b>PROPOSITO:</b> Administrar la información correspondiente a las aulas de la universidad por cada bloque.
<b>RESUMEN:</b> El Administrador del sistema ingresa al sistema, en esta parte puede actualizar información de un aula determinada.
<b>CURSO NORMAL DE LOS EVENTOS</b> <ol style="list-style-type: none"> <li>1. El sistema carga una ventana donde se puede observar la información de una determinada aula seleccionando de una lista desplegable el bloque, y de una lista el aula; la información incluye características como código, descripción, capacidad de estudiantes, área y disponibilidad.</li> <li>2. El Administrador del sistema para actualizar la información de una determinada aula da click en el botón actualizar.</li> </ol>
<b>FLUJOS ALTERNATIVOS</b>
<b>PRECONDICIONES</b> <ul style="list-style-type: none"> <li>• El Administrador del sistema debe haber ingresado al sistema con su nombre y contraseña.</li> </ul>
<b>POSTCONDICIONES</b> <ul style="list-style-type: none"> <li>• Se actualiza el sistema con los cambios hechos en un aula.</li> </ul>

<b>CASO DE USO: Agregar bloque</b>
<b>ACTORES:</b> Administrador del sistema
<b>PROPOSITO:</b> Adicionar los datos de un nuevo bloque al sistema.
<b>RESUMEN:</b> El Administrador del sistema ingresa al sistema, en esta parte puede agregar un nuevo bloque de aulas.
<b>CURSO NORMAL DE LOS EVENTOS</b>

<p>1. El Administrador del sistema para crear un nuevo bloque da click en el botón Agregar Bloque.</p> <p>1.1 El sistema muestra una ventana donde se permite ingresar los datos del nuevo bloque; al dar click en el botón guardar se adiciona la información en la base de datos.</p>
<p><b>FLUJOS ALTERNATIVOS</b></p> <p>1. En el caso de que no se llenen todos los datos que están en la ventana, el sistema muestra un error.</p>
<p><b>PRECONDICIONES</b></p> <ul style="list-style-type: none"> <li>• El Administrador del sistema debe haber ingresado al sistema con su nombre y contraseña.</li> </ul>
<p><b>POSTCONDICIONES</b></p> <ul style="list-style-type: none"> <li>• Se actualiza el sistema con la creación de un nuevo bloque</li> </ul>

<p><b>CASO DE USO: Agregar Aula</b></p>
<p><b>ACTORES:</b> Administrador del sistema</p>
<p><b>PROPOSITO:</b> Adicionar los datos de una nueva aula al sistema.</p>
<p><b>RESUMEN:</b> El Administrador del sistema ingresa al sistema, en esta parte puede agregar las aulas de un bloque ya creado.</p>
<p><b>CURSO NORMAL DE LOS EVENTOS</b></p> <p>1. El Administrador del sistema para crear una nueva aula da click en el botón Agregar Aula.</p> <p>1.1 El sistema muestra una ventana donde se permite ingresar los datos de la nueva aula; al dar click en el botón aceptar se adiciona la información en la base de datos.</p>
<p><b>FLUJOS ALTERNATIVOS</b></p> <p>1. En el caso de que no se llenen todos los datos que están en la ventana, el sistema muestra un error.</p>
<p><b>PRECONDICIONES</b></p>

- El Administrador del sistema debe haber ingresado al sistema con su nombre y contraseña.
- El Administrador del sistema debe haber ingresado previamente a la ventana de Agregar Bloque.

**POSTCONDICIONES**

- Se actualiza el sistema con la creación de una nueva aula.

**CASO DE USO: Eliminar Aula**

**ACTORES:** Administrador del sistema

**PROPOSITO:** Eliminar los datos de un aula determinada.

**RESUMEN:** El Administrador del sistema ingresa al sistema, en esta parte puede eliminar aulas de un bloque determinado.

**CURSO NORMAL DE LOS EVENTOS**

1. El Administrador del sistema para eliminar un aula elige la opción del menú "Eliminar Aula".
  - 1.1 El sistema muestra una ventana donde se permite seleccionar el bloque y el aula que se desea eliminar; al dar click en el botón "Eliminar" se elimina la información del aula en la base de datos.

**FLUJOS ALTERNATIVOS**

**PRECONDICIONES**

- El Administrador del sistema debe haber ingresado al sistema con su nombre y contraseña.

**POSTCONDICIONES**

- Se actualiza el sistema con la eliminación de un aula determinada.

<b>CASO DE USO: Manejar espacios académicos</b>
<b>ACTORES:</b> Secretarios de facultad
<b>PROPOSITO:</b> Seleccionar la facultad, el programa y el semestre para la asignación de los respectivos espacios académicos. Asignar los espacios académicos correspondientes a cada programa de cada facultad por semestre teniendo en cuenta la carga académica y las aulas disponibles
<b>RESUMEN:</b> El secretario de facultad ingresa al sistema, en esta parte puede seleccionar la facultad, el programa y el semestre para asignar espacios académicos.
<b>CURSO NORMAL DE LOS EVENTOS</b> <ol style="list-style-type: none"> <li>1. El sistema carga una ventana donde se puede seleccionar la facultad y el semestre de unas listas desplegadas, así como seleccionar el programa de una lista de acuerdo a la facultad y el periodo académico.</li> <li>2. Al dar click en el botón asignar espacio se muestra una ventana donde se pueden asignar los espacios de acuerdo a los datos seleccionados anteriormente.</li> </ol>
<b>FLUJOS ALTERNATIVOS</b> <ol style="list-style-type: none"> <li>1. En el caso de que no se llenen todos los datos que están en la ventana, el sistema muestra un error.</li> </ol>
<b>PRECONDICIONES</b> <ul style="list-style-type: none"> <li>• El secretario de facultad debe haber ingresado al sistema con su nombre y contraseña.</li> </ul>
<b>POSTCONDICIONES</b> <ul style="list-style-type: none"> <li>• Se actualiza el sistema con dicha información.</li> <li>• Se accede a la ventana de asignación de espacios académicos.</li> </ul>

<b>CASO DE USO: Asignar espacios académicos</b>
<b>ACTORES:</b> Secretarios de facultad
<b>PROPOSITO:</b> Asignar los espacios académicos correspondientes a cada programa de cada facultad por semestre teniendo en cuenta la información de carga académica y las aulas disponibles
<b>RESUMEN:</b> El secretario de facultad ingresa al sistema, en esta parte puede adicionar los espacios académicos de acuerdo a la facultad, el programa y el semestre seleccionados en la ventana anterior.
<p><b>CURSO NORMAL DE LOS EVENTOS</b></p> <ol style="list-style-type: none"> <li>1. El sistema carga una ventana donde se ingresa la información necesaria para agregar un espacio académico, como por ejemplo, el bloque, el aula, el día, la hora inicio, la hora fin, etc. Además carga la lista de profesores asignados a ese semestre. <ol style="list-style-type: none"> <li>1.1 Al dar click en el botón Cargar Materia se muestran en la lista las materias que dicta el profesor que se ha seleccionado anteriormente.</li> <li>1.2 El secretario de facultad para adicionar un espacio da click en el botón Aceptar y se almacena en la base de datos.</li> </ol> </li> <li>2. El secretario de facultad para mirar los espacios que ya han sido asignados da click en la opción del menú "Ver Espacios Ocupados". <ol style="list-style-type: none"> <li>2.1 El sistema carga una ventana donde se observa en una grilla los espacios ocupados hasta el momento.</li> </ol> </li> </ol>
<p><b>FLUJOS ALTERNATIVOS</b></p> <ol style="list-style-type: none"> <li>1. En el caso de que no se llenen todos los datos que están en la ventana, el sistema muestra un error.</li> <li>2. Si un espacio que ya ha sido asignado se repite, el sistema muestra un error.</li> <li>3. Si un profesor que ya tiene asignado un espacio se lo vuelve a introducir en el mismo espacio, el sistema muestra un error.</li> <li>4. Si no se ha seleccionado un profesor antes de cargar la materia, el sistema muestra un error.</li> </ol>
<p><b>PRECONDICIONES</b></p> <ul style="list-style-type: none"> <li>• El secretario de facultad debe haber ingresado al sistema con su nombre y contraseña.</li> <li>• El secretario de facultad debe haber seleccionado la facultad, el programa y el semestre en la ventana anterior.</li> </ul>

**POSTCONDICIONES**

- Se actualiza el sistema con los espacios académicos asignados.

**CASO DE USO: Eliminar Espacios Académicos****ACTORES:** Secretario de Facultad**PROPOSITO:** Eliminar los datos de un espacio académico asignado.**RESUMEN:** El Secretario de Facultad ingresa al sistema, en esta parte puede eliminar los espacios académicos que crea convenientes.**CURSO NORMAL DE LOS EVENTOS**

1. El Secretario de Facultad para eliminar un espacio académico elige la opción del menú “Eliminar Espacio Académico”.
  - 1.1 El sistema muestra una ventana donde se permite seleccionar el bloque, el aula, el día, la hora de inicio y la hora final del espacio académico que se desea eliminar; al dar click en el botón “Eliminar” se elimina la información del espacio académico en la base de datos.

**FLUJOS ALTERNATIVOS****PRECONDICIONES**

- El Secretario de Facultad debe haber ingresado al sistema con su nombre y contraseña.

**POSTCONDICIONES**

- Se actualiza el sistema con la eliminación de un espacio académico.

<b>CASO DE USO: Agregar Nuevos Programas</b>
<b>ACTORES:</b> Administrador del sistema.
<b>PROPOSITO:</b> Adicionar nuevos programas académicos al sistema.
<b>RESUMEN:</b> El administrador del sistema ingresa al sistema, en esta parte puede agregar nuevos programas académicos.
<b>CURSO NORMAL DE LOS EVENTOS</b> <ol style="list-style-type: none"> <li>1. El sistema muestra una ventana donde se permite ingresar los datos para crear un nuevo programa; al dar click en el botón aceptar se adiciona la información en la base de datos.</li> </ol>
<b>FLUJOS ALTERNATIVOS</b> <ol style="list-style-type: none"> <li>1. En el caso de que no se llenen todos los datos que están en la ventana, el sistema muestra un error.</li> </ol>
<b>PRECONDICIONES</b> <ul style="list-style-type: none"> <li>• El secretario de facultad debe haber ingresado al sistema con su nombre y contraseña</li> </ul>
<b>POSTCONDICIONES</b> <ul style="list-style-type: none"> <li>• Se actualiza el sistema con la creación de un nuevo programa académico.</li> </ul>

<b>CASO DE USO: Cargar información académica</b>
<b>ACTORES:</b> Secretarios de facultad
<b>PROPOSITO:</b> Ingresar los datos de profesores y materias al sistema.
<b>RESUMEN:</b> El secretario de facultad ingresa al sistema, en esta parte puede cargar la información de profesores que dictaran clase en ese semestre con sus respectivas materias a través de archivos *.txt.
<b>CURSO NORMAL DE LOS EVENTOS</b> <ol style="list-style-type: none"> <li>1. El sistema muestra una ventana donde se permite ingresar la información académica;</li> </ol>

<p>1.1 al dar click en el botón Examinar se escoge el archivo .txt que se desea abrir.</p> <p>1.2 Después de seleccionar el archivo se da click en el botón Cargar para mostrar los datos en una grilla.</p> <p>1.3 Se da click en el botón Guardar en BD para ingresar la información en la base de datos.</p>
<p><b>FLUJOS ALTERNATIVOS</b></p> <ol style="list-style-type: none"> <li>1. En el caso de que no se cargue la información en la grilla, el sistema muestra un error.</li> <li>2. Si no se puede guardar la información en la base de datos, el sistema muestra un error.</li> </ol>
<p><b>PRECONDICIONES</b></p> <ul style="list-style-type: none"> <li>• El secretario de facultad debe haber ingresado al sistema con su nombre y contraseña.</li> </ul>
<p><b>POSTCONDICIONES</b></p> <ul style="list-style-type: none"> <li>• Se actualiza el sistema con el ingreso de la información académica.</li> </ul>

<p><b>CASO DE USO: Asignar espacios extraclase</b></p>
<p><b>ACTORES:</b> Administrador del sistema</p>
<p><b>PROPOSITO:</b> Asignar espacios extracurriculares en las aulas destinados a otras actividades que no tienen que ver con los programas académicos de la universidad.</p>
<p><b>RESUMEN:</b> El Administrador del sistema ingresa al sistema, en esta parte puede adicionar espacios extraclase de acuerdo a la disponibilidad de aulas.</p>
<p><b>CURSO NORMAL DE LOS EVENTOS</b></p> <ol style="list-style-type: none"> <li>1. El sistema carga una ventana donde se ingresa la información necesaria para agregar un espacio extraclase, como por ejemplo, el bloque, el aula, la fecha, la hora inicio, la hora fin, etc. <ol style="list-style-type: none"> <li>1.1 El secretario de facultad para adicionar un espacio da click en el botón Adicionar y se almacena en la base de datos.</li> </ol> </li> <li>2. El Administrador del sistema para mirar los espacios que están ocupados da click en la opción del menú “Ver Espacios no Disponibles”.</li> </ol>

2.1 El sistema carga una ventana donde se observa en una grilla los espacios ocupados hasta el momento.
<b>FLUJOS ALTERNATIVOS</b> 1. En el caso de que no se llenen todos los datos que están en la ventana, el sistema muestra un error.
<b>PRECONDICIONES</b> <ul style="list-style-type: none"> <li>• El Administrador del sistema debe haber ingresado al sistema con su nombre y contraseña.</li> <li>• El Administrador del sistema debe haber observado con anticipación los espacios no disponibles.</li> </ul>
<b>POSTCONDICIONES</b> <ul style="list-style-type: none"> <li>• Se actualiza el sistema con los espacios extracurriculares asignados.</li> <li>• Se accede a la ventana para mirar los espacios no disponibles.</li> </ul>

<b>CASO DE USO: Eliminar Espacios Extra clase</b>
<b>ACTORES:</b> Administrador del sistema
<b>PROPOSITO:</b> Eliminar los datos de un espacio extra clase asignado.
<b>RESUMEN:</b> El Administrador del sistema ingresa al sistema, en esta parte puede eliminar los espacios extra clase que crea convenientes.
<b>CURSO NORMAL DE LOS EVENTOS</b> 1. El Administrador del sistema para eliminar un espacio extra clase elige la opción del menú “Eliminar Espacio Extra clase”. 1.1 El sistema muestra una ventana donde se permite seleccionar el bloque, el aula, la fecha, la hora de inicio y la hora final del espacio extra clase que se desea eliminar; al dar click en el botón “Eliminar” se elimina la información del espacio extra clase en la base de datos.
<b>FLUJOS ALTERNATIVOS</b>

<p><b>PRECONDICIONES</b></p> <ul style="list-style-type: none"> <li>• El Administrador del sistema debe haber ingresado al sistema con su nombre y contraseña.</li> </ul>
<p><b>POSTCONDICIONES</b></p> <ul style="list-style-type: none"> <li>• Se actualiza el sistema con la eliminación de un espacio extra clase.</li> </ul>

<p><b>CASO DE USO: Manejar reportes</b></p>
<p><b>ACTORES:</b> Administrador del sistema y secretarios de facultad</p>
<p><b>PROPOSITO:</b> Generar los reportes de espacios académicos y extraclase asignados en las aulas de la universidad, además de los reportes de utilización de aulas.</p>
<p><b>RESUMEN:</b> El Administrador del sistema o secretario de facultad ingresa al sistema, en esta parte puede elegir que reporte desea generar, de espacios académicos y/o espacios extraclase o de utilización de aulas.</p>
<p><b>CURSO NORMAL DE LOS EVENTOS</b></p> <ol style="list-style-type: none"> <li>1. En el marco de Menu Principal se puede escoger la opción del reporte que se quiere generar. <ol style="list-style-type: none"> <li>1.1 Al escoger la opción de “Espacios Académicos” accede a la ventana para generar reportes de ese tipo.</li> <li>1.2 Al escoger la opción de “Espacios extraclase” genera el reporte de ese tipo.</li> <li>1.3 Al escoger la opción de “Utilización de Aulas” accede a la ventana para generar reportes de ese tipo.</li> </ol> </li> </ol>
<p><b>FLUJOS ALTERNATIVOS</b></p>
<p><b>PRECONDICIONES</b></p> <ul style="list-style-type: none"> <li>• El Administrador del sistema o secretario de facultad debe haber ingresado al sistema con su nombre y contraseña.</li> </ul>
<p><b>POSTCONDICIONES</b></p> <ul style="list-style-type: none"> <li>• Se actualiza el sistema con la generación de reportes.</li> </ul>

<b>CASO DE USO: Generar reportes de espacios académicos</b>
<b>ACTORES:</b> Administrador del sistema y secretarios de facultad
<b>PROPOSITO:</b> Generar los reportes de espacios académicos asignados en las aulas de la universidad por facultad, programa y semestre.
<b>RESUMEN:</b> El Administrador del sistema o secretario de facultad ingresa al sistema, en esta parte puede seleccionar la facultad y el semestre de una lista desplegable, así como el programa de una lista, para generar el reporte de los espacios asignados que concuerdan con los datos seleccionados.
<b>CURSO NORMAL DE LOS EVENTOS</b> <ol style="list-style-type: none"> <li>1. El sistema carga una ventana donde se selecciona la facultad, el programa y el semestre para generar un reporte. <ol style="list-style-type: none"> <li>1.1 Al dar click en el botón aceptar se almacena en una tabla de la base de datos los espacios asignados para luego crear el reporte especificado.</li> </ol> </li> </ol>
<b>FLUJOS ALTERNATIVOS</b> <ol style="list-style-type: none"> <li>1. En el caso de que no se llenen todos los datos que están en la ventana, el sistema muestra un error.</li> </ol>
<b>PRECONDICIONES</b> <ul style="list-style-type: none"> <li>• El Administrador del sistema o secretario de facultad debe haber ingresado al sistema con su nombre y contraseña.</li> <li>• El Administrador del sistema o secretario de facultad debe haber elegido la opción de espacios académicos.</li> </ul>
<b>POSTCONDICIONES</b> <ul style="list-style-type: none"> <li>• Se actualiza el sistema con la generación de reportes de espacios académicos.</li> </ul>

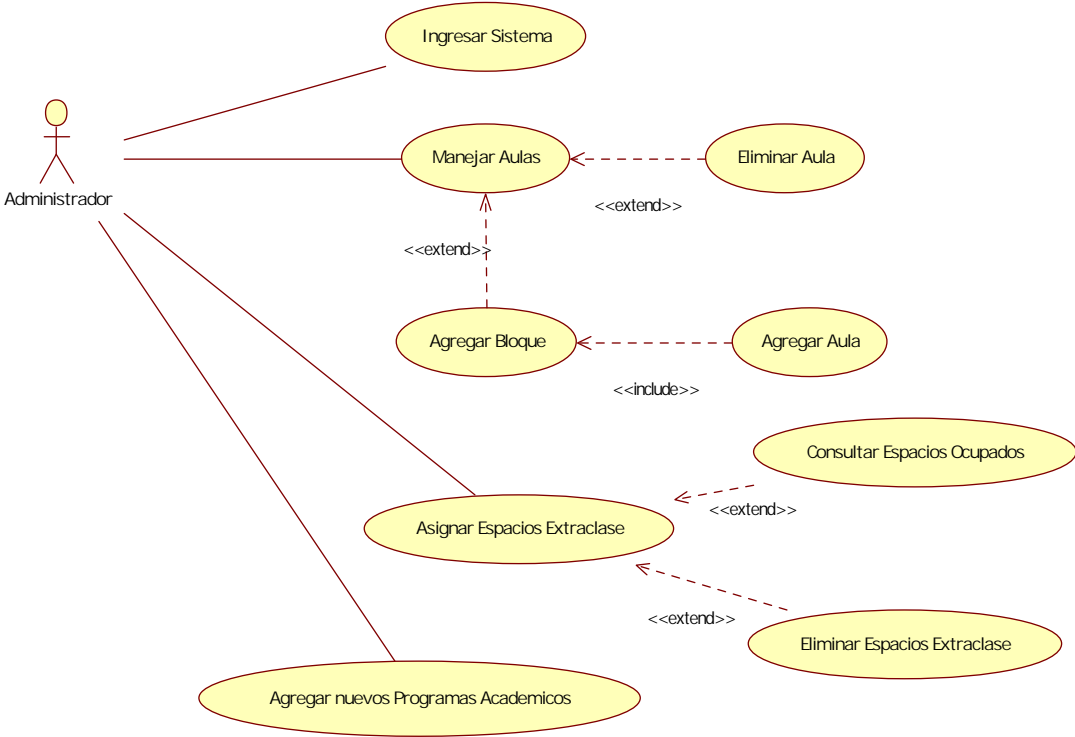
<b>CASO DE USO: Generar reportes de espacios extraclase</b>
<b>ACTORES:</b> Administrador del sistema y secretarios de facultad
<b>PROPOSITO:</b> Generar los reportes de espacios extracurriculares asignados en las aulas de la universidad.
<b>RESUMEN:</b> El Administrador del sistema o secretario de facultad ingresa al sistema, en esta parte puede generar el reporte de espacios extracurriculares.
<b>CURSO NORMAL DE LOS EVENTOS</b> 1. El sistema carga una ventana donde se puede observar los espacios extraclase asignados. .
<b>FLUJOS ALTERNATIVOS</b>
<b>PRECONDICIONES</b> <ul style="list-style-type: none"> <li>• El Administrador del sistema o secretario de facultad debe haber elegido la opción de espacios extraclase.</li> </ul>
<b>POSTCONDICIONES</b> <ul style="list-style-type: none"> <li>• Se actualiza el sistema con la generación de reportes de espacios extraclase.</li> </ul>

<b>CASO DE USO: Generar reportes de utilización de aulas</b>
<b>ACTORES:</b> Administrador del sistema y secretarios de facultad
<b>PROPOSITO:</b> Generar los reportes de utilización de aulas por bloque y aula.
<b>RESUMEN:</b> El Administrador del sistema o secretario de facultad ingresa al sistema, en esta parte puede seleccionar el bloque y el aula de unas listas desplegadas para generar el reporte de utilización de aulas.
<b>CURSO NORMAL DE LOS EVENTOS</b> 1. El sistema carga una ventana donde se permite seleccionar el bloque y un aula para generar un reporte. 1.2 Al dar click en el botón aceptar se carga una ventana donde se puede observar el reporte especificado.

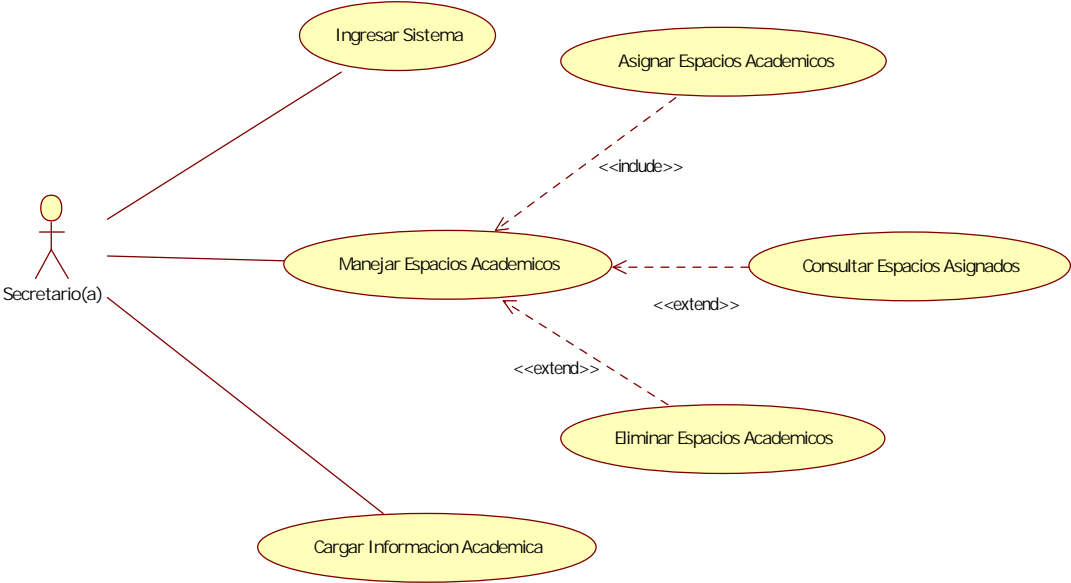
<b>FLUJOS ALTERNATIVOS</b>
<b>PRECONDICIONES</b> <ul style="list-style-type: none"><li>• El Administrador del sistema o secretario de facultad debe haber ingresado al sistema con su nombre y contraseña.</li><li>• El Administrador del sistema o secretario de facultad debe haber elegido la opción de utilización de aulas.</li></ul>
<b>POSTCONDICIONES</b> <ul style="list-style-type: none"><li>• Se actualiza el sistema con la generación de reportes de utilización de aulas.</li></ul>

### 5.3 DIAGRAMAS DE CASOS DE USO

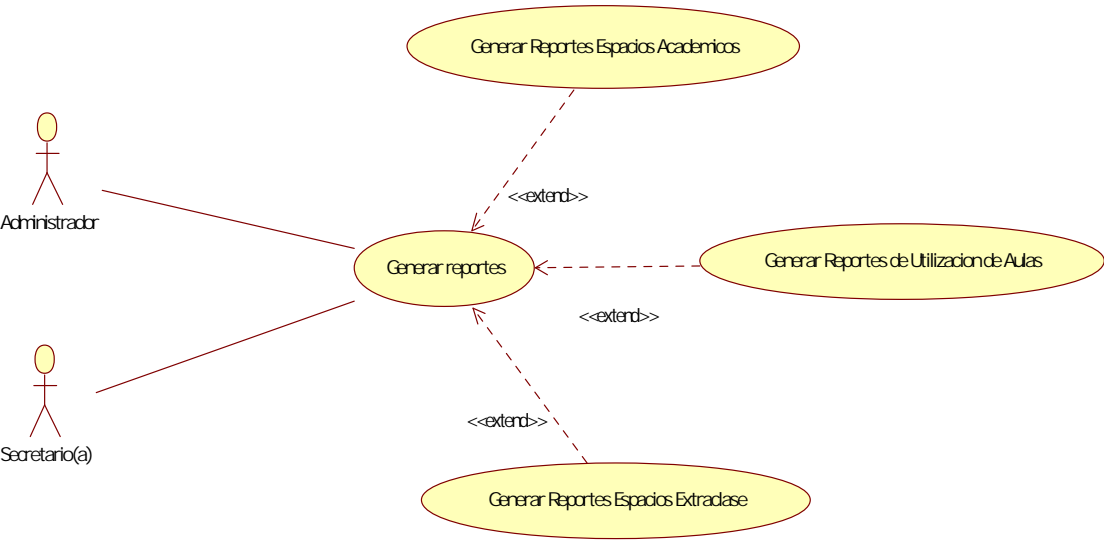
#### 5.3.1 Figura N° 1



5.3.2 Figura N° 2

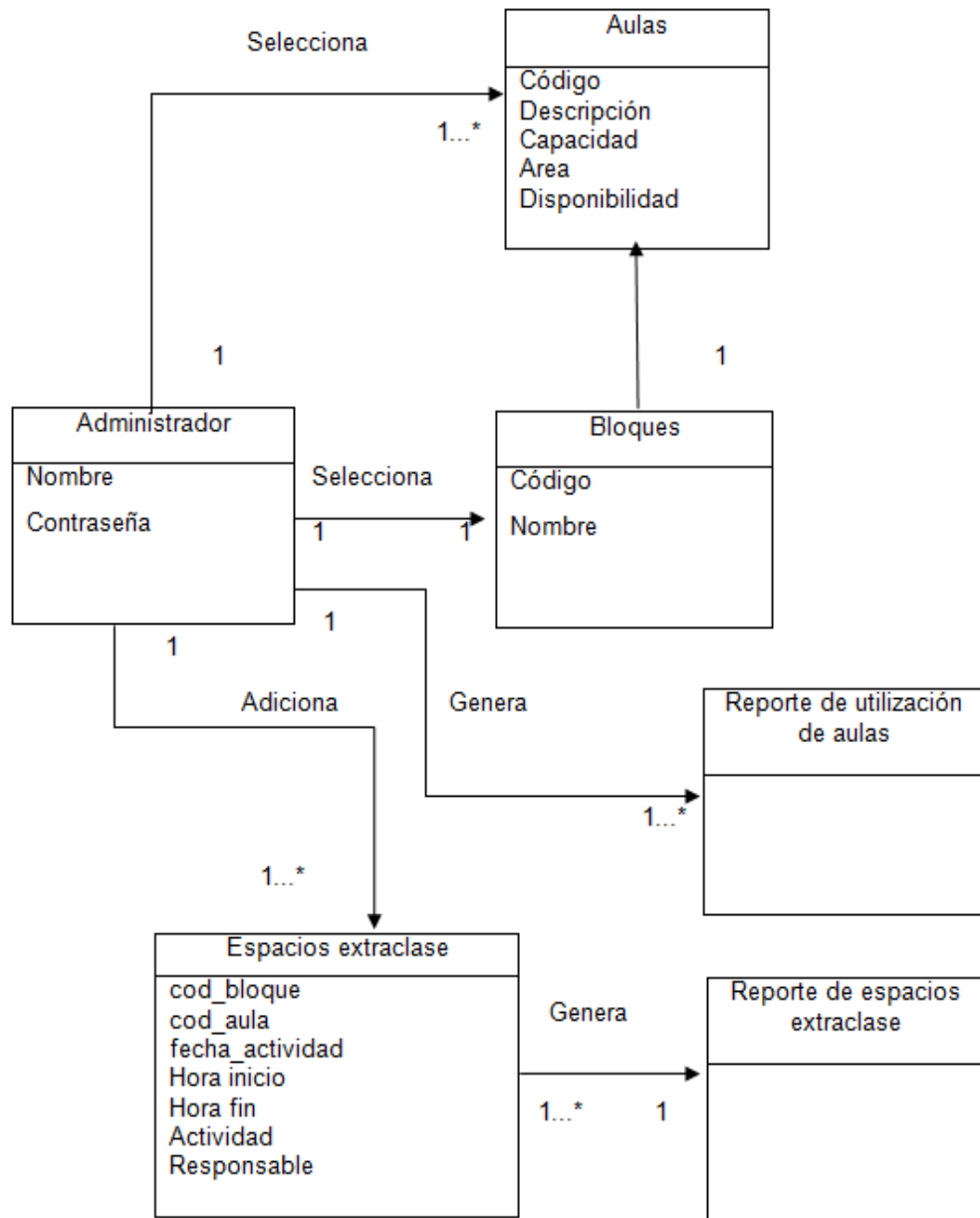


5.3.3 Figura N° 3

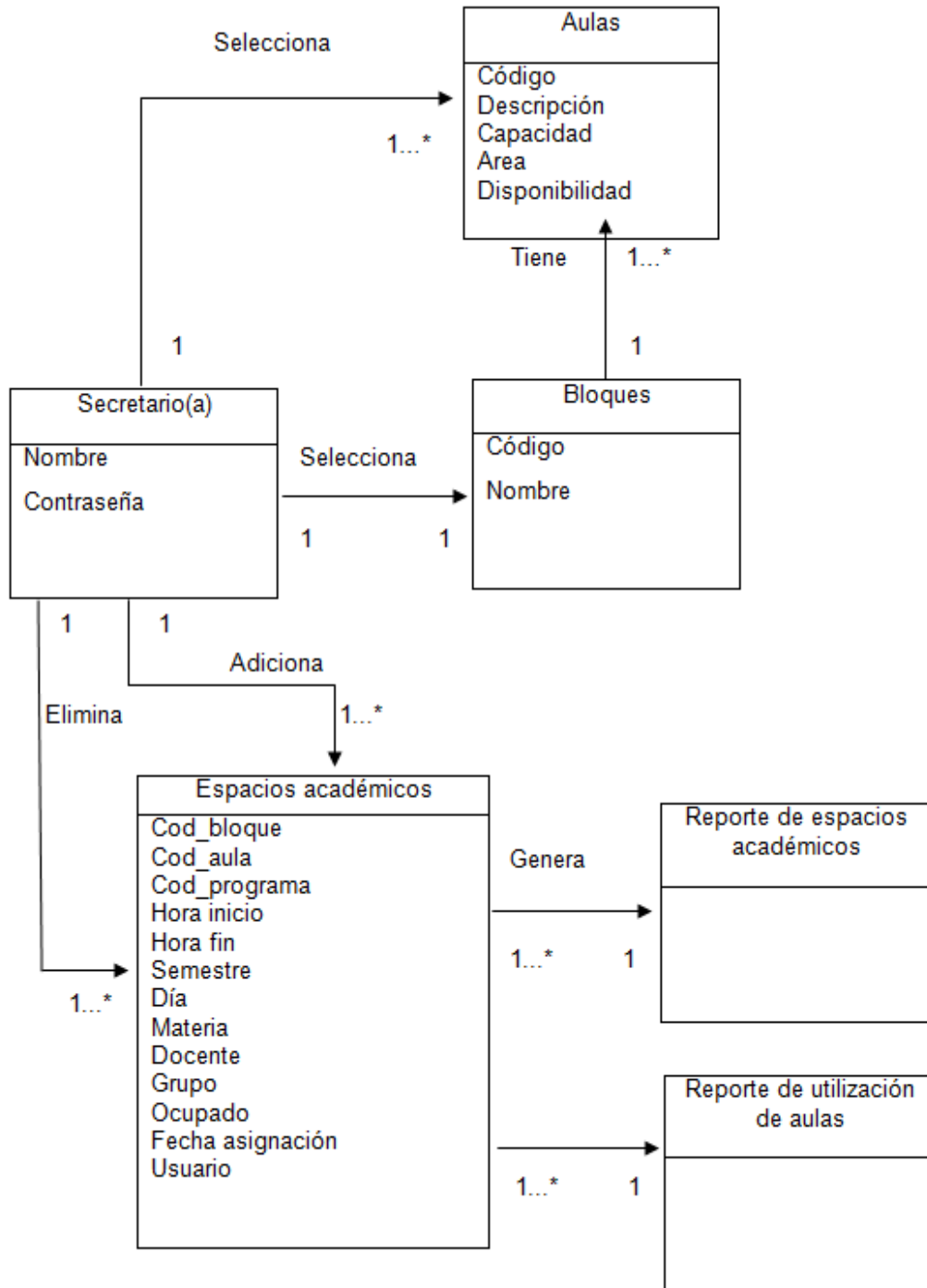


## 5.4 MODELO CONCEPTUAL

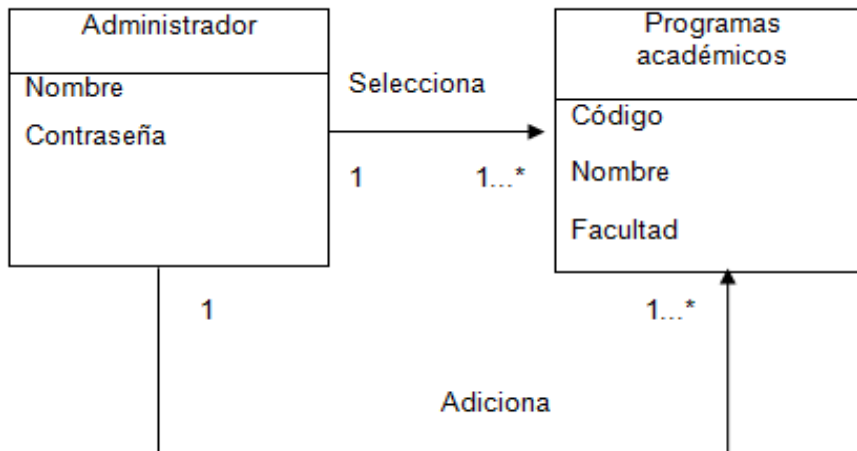
### 5.4.1 Figura N° 1



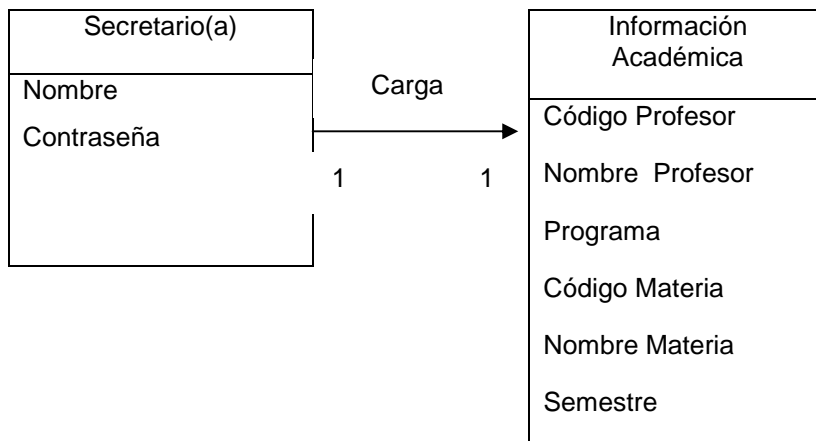
5.4.2 Figura N° 2



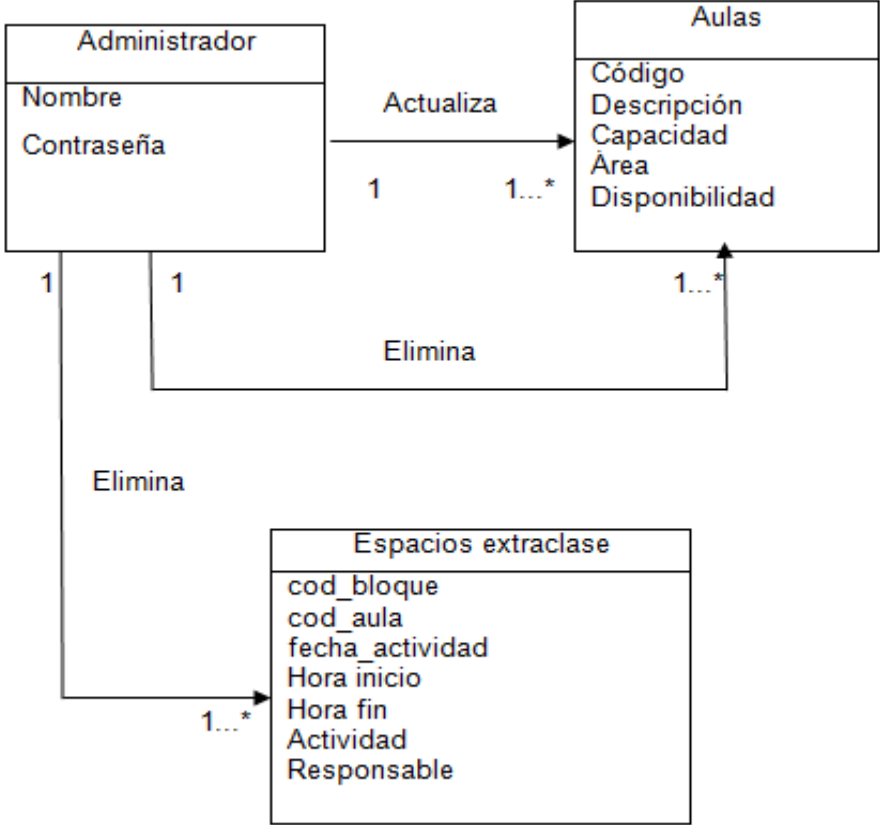
### 5.4.3 Figura N° 3



### 5.4.4 Figura N° 4

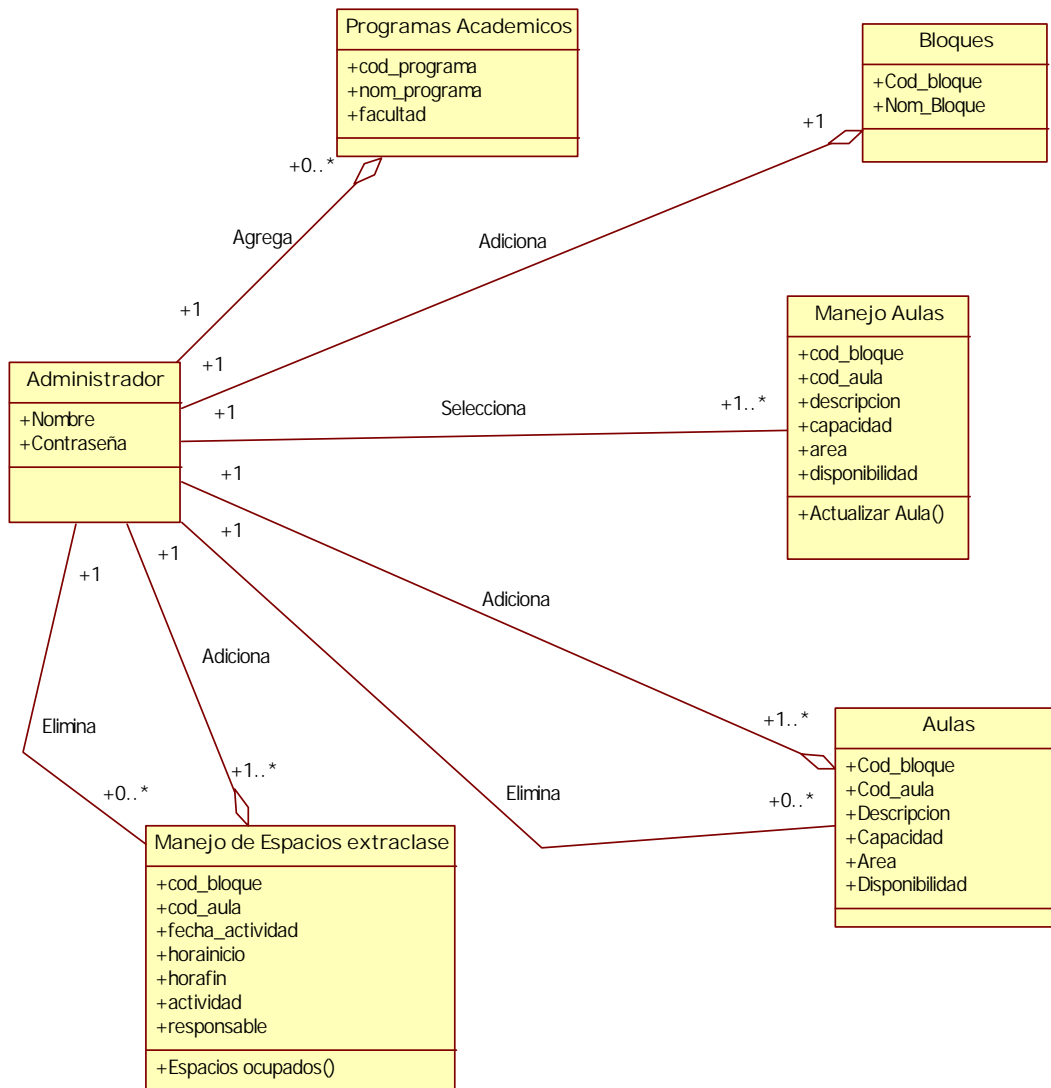


5.4.5 Figura N°5

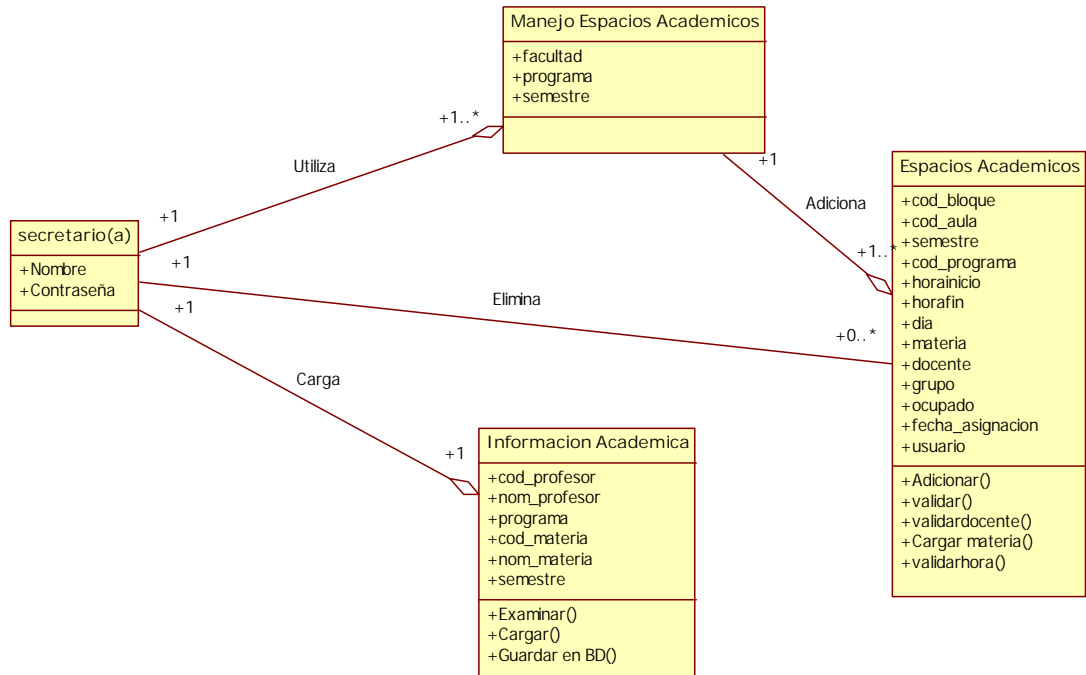


## 5.5 DIAGRAMA DE CLASES

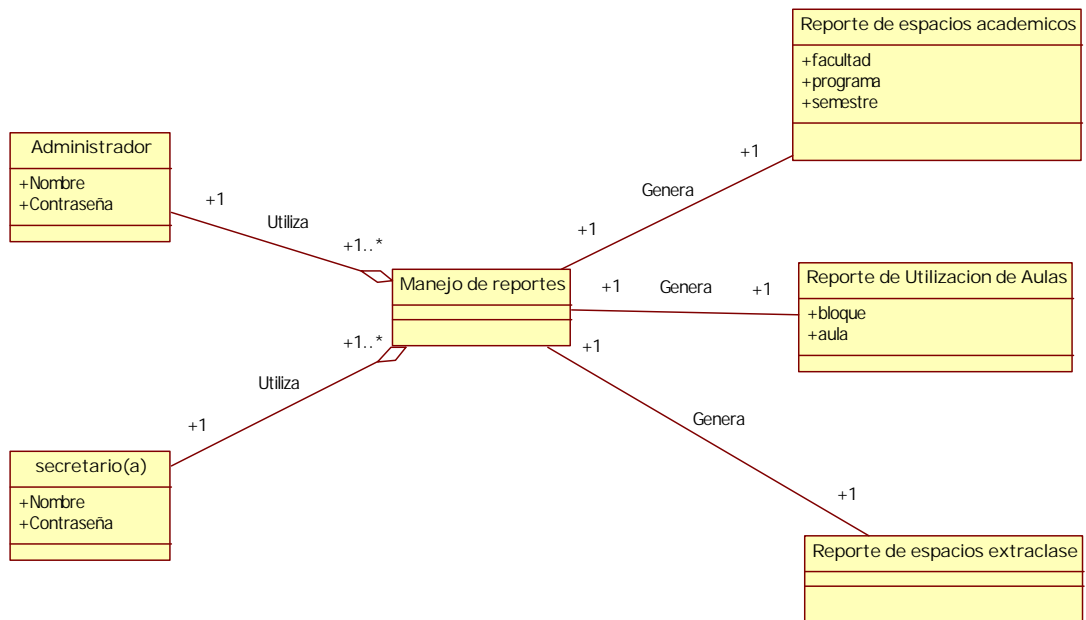
### 5.5.1 Figura N° 1



### 5.5.2 Figura N° 2



### 5.5.3 Figura N° 3



## 5.6 DISEÑO DE LA BASE DE DATOS

Para el diseño de la base de datos del sistema desarrollado se utilizaron los conceptos de las bases de datos relacionales debido a que son las de mayor difusión, y a que son más simples y eficaces para representar y manipular datos.

Una base de datos relacional es una base de datos en donde todos los datos visibles al usuario están organizados estrictamente como tablas de valores, y en donde todas las operaciones de la base de datos operan sobre estas tablas. Estas bases de datos son percibidas por los usuarios como una colección de relaciones normalizadas de diversos grados que varían con el tiempo.

En términos tradicionales una relación se asemeja a un archivo, una tupla a un registro, y un atributo a un campo. Pero estas correspondencias son aproximadas, en el mejor de los casos. Una relación no debe considerarse como “solo un archivo”, sino más bien como un archivo disciplinado, siendo el resultado de esta disciplina una simplificación considerable de las estructuras de datos con las cuales debe interactuar el usuario, lo cual a su vez simplifica los operadores requeridos para manejar esas estructuras.



## 5.6.2 Tablas de la base de datos

<b>Usuarios</b>				
<b>Descripción:</b> Tabla donde se encuentran los nombres y contraseñas de los usuarios del sistema.				
<b>Atributo</b>	<b>Tipo</b>	<b>Tamaño(bytes)</b>	<b>Llave</b>	<b>Descripción</b>
cod_usuario	numérico	2	primaria	Código del usuario
nom_usuario	carácter	20		Nombre del usuario
passwd_usuario	carácter	20		Contraseña del usuario

<b>Bloques</b>				
<b>Descripción:</b> Tabla donde se almacenan los bloques de aulas de la universidad.				
<b>Atributo</b>	<b>Tipo</b>	<b>Tamaño(bytes)</b>	<b>Llave</b>	<b>Descripción</b>
cod_bloque	entero	4	primaria	Código del bloque
nom_bloque	carácter	15		Nombre del bloque

<b>Aulas</b>				
<b>Descripción:</b> Tabla donde se guardan las características de las aulas de la Universidad.				
<b>Atributo</b>	<b>Tipo</b>	<b>Tamaño(bytes)</b>	<b>Llave</b>	<b>Descripción</b>
cod_bloque	entero	4	primaria, foránea	Código del bloque
cod_aula	entero	4	primaria	Código del aula
descripcion	carácter	20		Tipo de aula(auditorio, clases, dibujo)
capacidad	entero	4		Número máximo de estudiantes en una aula
area	carácter	15		Area en metros cuadrados de una aula
disponibilidad	Booleano	1		Determina si una aula esta disponible o no

<b>Espacios académicos</b>				
<b>Descripción:</b> Tabla donde se almacena toda la información correspondiente a la asignación de espacios académicos.				
<b>Atributo</b>	<b>Tipo</b>	<b>Tamaño(bytes)</b>	<b>Llave</b>	<b>Descripción</b>
cod_bloque	entero	4	Primaria, foránea	Código del bloque
cod_aula	entero	4	Primaria, foránea	Código del aula
nom_programa	carácter	20		Nombre del programa académico

semestre	entero	4	primaria	semestre del programa académico
hora inicio	carácter	15	primaria	Hora en la q inicia el espacio académico
hora fin	carácter	15	primaria	Hora en la q finaliza el espacio académico
dia	carácter	10	primaria	Día en el cual se asigna un espacio académico
materia	carácter	20		Materia que se dicta en un espacio académico
docente	carácter	20		Profesor que dicta la materia
grupo	entero	4		Grupo al cual se asigna un espacio académico
ocupado	Booleano	1		Determina si un espacio académico ya esta ocupado
fecha_asignacion	fecha	8		Fecha en la que fue asignado un espacio académico
nom_programa	carácter	20		Programa académico en el que fue asignado un espacio
usuario	carácter	30		Usuario que asigna el espacio academico

periodo	carácter	10		Periodo académico en el que se asignó el espacio
---------	----------	----	--	--

<b>Espacios extracurriculares</b>				
<b>Descripción:</b> Tabla donde se almacena toda la información correspondiente a la asignación de espacios extraclase.				
<b>Atributo</b>	<b>Tipo</b>	<b>Tamaño(bytes)</b>	<b>Llave</b>	<b>Descripción</b>
cod_bloque	entero	4		Código del bloque
cod_aula	entero	4		Código del aula
Fecha_actividad	fecha	8	Primaria	Fecha que se asigna para la actividad del espacio extraclase
actividad	carácter	20		Actividad que se asigna en el espacio extraclase
hora_inicio	carácter	10	Primaria	Hora en la que inicia el espacio extraclase
hora_fin	carácter	15	Primaria	Hora en la que finaliza el espacio extraclase
responsable	carácter	20		Persona encargada de realizar la actividad

<b>Programas</b>				
<b>Descripción:</b> Tabla donde se encuentran los programas académicos de la universidad.				
<b>Atributo</b>	<b>Tipo</b>	<b>Tamaño(bytes)</b>	<b>Llave</b>	<b>Descripción</b>
cod_programa	carácter	4	primaria	Código del programa
nom_programa	carácter	30		Nombre del programa
facultad	carácter	30		Facultad a la cual pertenece el programa

<b>Profesor Materia</b>				
<b>Descripción:</b> Tabla donde se almacena la información de profesores de cada programa y sus respectivas materias.				
<b>Atributo</b>	<b>Tipo</b>	<b>Tamaño(bytes)</b>	<b>Llave</b>	<b>Descripción</b>
cod_profesor	carácter	5	primaria	Código del profesor
nom_profesor	carácter	20		Nombre del profesor
programa	carácter	30		Nombre del programa
cod_materia	carácter	4	primaria	Código de la materia
nom_materia	carácter	30		Nombre de la materia
semestre	entero	4		Semestre en el que se dicta la materia

<b>Temporal</b>				
<b>Descripción:</b> Tabla donde se almacena los datos para construir los reportes de espacios académicos.				
<b>Atributo</b>	<b>Tipo</b>	<b>Tamaño(bytes)</b>	<b>Llave</b>	<b>Descripción</b>
bloque	entero	4		Código del bloque
aula	entero	4		Código del aula
horaini	carácter	10		Hora en la q inicia el espacio académico
horaf	carácter	10		Hora en la q finaliza el espacio académico
dia	carácter	10		Día en el cual se asigna un espacio académico
materia	carácter	20		Materia que se dicta en un espacio académico
docente	carácter	20		Profesor que dicta la materia
grupo	carácter	2		Grupo al cual se asigna un espacio académico

<b>Trabajadores</b>				
<b>Descripción:</b> Tabla donde se encuentran las personas que trabajan en la universidad.				
<b>Atributo</b>	<b>Tipo</b>	<b>Tamaño(bytes)</b>	<b>Llave</b>	<b>Descripción</b>
Id_trabajador	carácter	15	primaria	Nº de identificación del trabajador
nom_trabajador	carácter	30		Nombre del trabajador
cargo	carácter	20		Cargo del trabajador
habilitado	Booleano			Indica si el trabajador esta actualmente laborando

## 5.7 GLOSARIO

### 5.7.1 Casos de uso

<b>Termino</b>	<b>Categoría</b>	<b>Descripción</b>
Ingresar al sistema	Caso de uso	Proceso de autenticación de usuarios del sistema.
Manejar Aulas	Caso de uso	Proceso de administración y actualización de la información de las aulas de la universidad.
Agregar bloque	Caso de uso	Proceso de inserción de nuevos bloques de aulas al sistema.
Agregar aula	Caso de uso	Proceso de inserción de nuevas aulas por bloque al sistema.
Eliminar aula	Caso de uso	Proceso de eliminación de un aula determinada.
Asignar espacios extraclase.	Caso de uso	Proceso de asignación de espacios extracurriculares en las aulas.
Eliminar espacios extraclase.	Caso de uso	Proceso de eliminación de espacios extracurriculares en las aulas.
Manejar espacios académicos.	Caso de uso	Proceso mediante el cual se selecciona el programa y el semestre para asignar espacios académicos.
Asignar espacios académicos.	Caso de uso	Proceso de asignación de espacios académicos en las aulas.
Eliminar espacios académicos.	Caso de uso	Proceso de eliminación de espacios académicos en las aulas.

Agregar programas académicos.	Caso de uso	Proceso de inserción de nuevos programas académicos al sistema.
Generar reportes.	Caso de uso	Proceso mediante el cual se selecciona el tipo de reporte que se desea generar.
Generar reportes de espacios académicos.	Caso de uso	Proceso de generación del reporte de espacios académicos.
Generar reportes de espacios extraclase.	Caso de uso	Proceso de generación del reporte de espacios extraclase.
Generar reportes de utilización de aulas.	Caso de uso	Proceso de generación del reporte de utilización de determinadas aulas.

## 5.7.2 Conceptos

<b>Termino</b>	<b>Categoría</b>	<b>Descripción</b>
Secretario	Concepto	Persona encargada de la asignación de espacios académicos.
Administrador	Concepto	Persona encargada de la administración de aulas y de la asignación de espacios extraclase.
Bloque	Concepto	Espacio físico el cual contiene varias aulas de clase.
Aula	Concepto	Espacio físico donde se asigna un espacio académico o extraclase.
Espacio académico	Concepto	Franja horaria en la cual se dicta una materia de un semestre perteneciente a un programa académico.
Espacio extraclase	Concepto	Franja horaria que se designa para una actividad diferente a la académica.
Programa académico	Concepto	Carrera que se ofrece en la universidad.
Reporte de espacios académicos.	Concepto	Formato que muestra los espacios académicos asignados por programa y semestre.
Reporte de espacios extraclase.	Concepto	Formato que muestra los espacios extraclase asignados.

## 6. CONCLUSIONES

- La metodología xP (extreme programming) es muy adecuada, ágil y rápida para el desarrollo de software, sin embargo, crea aplicaciones de poca documentación, por lo que es necesario utilizar herramientas como el UML (Lenguaje de Modelado Unificado).
- El sistema desarrollado anteriormente es una gran ayuda para la administración de las aulas en la Universidad y para la asignación de horarios en los programas académicos.
- A lo largo de la construcción del software se miró las ventajas del Visual Studio 2010 en el desarrollo de aplicaciones complejas y manejo de bases de datos. La interfaz de programación es muy sencilla de manejar y ofrece una ayuda muy eficaz para los usuarios.
- El módulo de reportes es de gran utilidad ya que genera los reportes de forma muy rápida y permite imprimirlos o exportarlos hacia otros programas como Excel o Word.

## **7. RECOMENDACIONES**

- Capacitar a los usuarios sobre el manejo del sistema.
- Utilizar el Sistema Operativo Windows 7 para un mejor funcionamiento del sistema.
- Investigar la forma de acoplarlo con el sistema de carga académica y el sistema de horarios de la universidad.
- Crear correctamente la conexión ODBC con la base de datos.

## BIBLIOGRAFIA

1. Introducción a la programación extrema. 2007. Obtenido el 11 de agosto de 2010 desde <http://www.davidvalverde.com/blog/introduccion-a-la-programacion-extrema-xp/>
2. DESARROLLO ÁGIL DE SOFTWARE – CASO PROGRAMACIÓN EXTREMA – XP. 2009. Obtenido el 11 de agosto de 2010 desde <http://www ldc.usb.ve/~abianc/materias/ci4713/metodologiasagiles.pdf>
3. Una explicación de la programación extrema (XP). 2003. Obtenido el 12 de agosto de 2010 desde <http://www.willydev.net/descargas/prev/ExplicaXp.pdf>
4. Newkirk. Programación Extrema En La Práctica, Editorial Pearson (Año: 2002, 1ª edición), ISBN: 8478290575.
5. Beck, Kent. UNA EXPLICACION DE LA PROGRAMACION EXTREMA, Editorial Pearson, 216 págs., ISBN: 84-7829-055-9.
6. Asignación de horarios universitario: un enfoque deterministico. 2009. Obtenido el 11 de octubre de 2010 desde <http://1577308628427254787-a-1802744773732722657-s-sites.googlegroups.com/site/gennaroalphonse/home/articulo.pdf?>
7. Sistema Integrado de Gestión. 2010. Obtenido el 11 de octubre de 2010 desde <http://sig.ucaldas.edu.co/gestionDocumental/vistaDetalleInstructivo.php?codDoc=Mjg5&versionDoc=2&codProceso=RA>
8. Asignación de Horarios de Clases Universitarias Mediante Algoritmos Evolutivos. 2008. Obtenido el 13 de octubre de 2010 desde <http://manglar.uninorte.edu.co/bitstream/10584/80/1/84032706.pdf>

9. TIMETABLING. 2005. Obtenido el 13 de octubre de 2010 desde <http://bdigital.eafit.edu.co/bdigital/PROYECTO/P005.12C112/capitulo1.pdf>
10. Sistema de Información. 2008. Obtenido el 14 de marzo de 2011 desde <http://definicion.de/sistema-de-informacion/>.
11. Bases de datos. 2007. Obtenido el 14 de marzo de 2011 desde <http://www.maestrosdelweb.com/principiantes/%C2%BFque-son-las-bases-de-datos/>
12. Microsoft Visual Studio. 2010. Obtenido el 14 de marzo de 2011 desde [http://es.wikipedia.org/wiki/Microsoft\\_VisualStudio](http://es.wikipedia.org/wiki/Microsoft_VisualStudio)
13. Que es el Asp? 2001. Obtenido el 19 de agosto de 2011 desde <http://www.maestrosdelweb.com/editorial/aspintro/>

