

**SISTEMA DE SOMBREADO DE PLAN DE ESTUDIOS UTB**

**JORGE ALBERTO RAMIREZ SERRATO**

**JHON J JARNEY GUERRERO MARTINEZ**

**UNIVERSIDAD TECNOLÓGICA DE BOLÍVAR**

**FACULTAD DE INGENIERÍA**

**DIRECCIÓN DE PROGRAMA DE INGENIERÍA DE SISTEMAS**

**MINOR EN DESARROLLO DE APLICACIONES .NET**

**CARTAGENA DE INDIAS, D. T Y C.**

**2012**

**SISTEMA DE SOMBREADO DE PLAN DE ESTUDIOS UTB**

**JORGE ALBERTO RAMIREZ SERRATO**

**JHON J JARNEY GUERRERO MARTINEZ**

**Proyecto presentado como requisito final para optar al título de Ingeniero de  
Sistema**

**DIRECTOR:**

**MOISÉS RAMÓN QUINTANA ÁLVAREZ**

**Director de Programa de Ingeniería de Sistemas**

**UNIVERSIDAD TECNOLÓGICA DE BOLÍVAR**

**FACULTAD DE INGENIERÍA**

**DIRECCIÓN DE PROGRAMA DE INGENIERÍA DE SISTEMAS**

**MINOR EN DESARROLLO DE APLICACIONES .NET**

**CARTAGENA DE INDIAS, D. T Y C.**

**2012**

**Nota de aceptación**

---

---

---

---

---

**Firma del presidente del jurado**

---

**Firma del jurado**

---

**Firma del jurado**

Cartagena de Indias, D. T y C. Junio de 2012

**Señores**

**COMITÉ CURRICULAR**

**UNIVERSIDAD TECNOLÓGICA DE BOLÍVAR**

Ciudad

Respetados señores:

Con todo el interés me dirijo a ustedes para presentar a su consideración, estudio y aprobación la monografía titulada “**SISTEMA DE SOMBREADO DE PLAN DE ESTUDIOS UTB**”, como requisito para obtener el título de Ingeniero de Sistemas.

Atentamente,



---

**JHON J JARNEY GUERRERO MARTINEZ**

Cartagena de Indias, D. T y C. Junio de 2012

**Señores**

**COMITÉ CURRICULAR**

**UNIVERSIDAD TECNOLÓGICA DE BOLÍVAR**

Ciudad

Respetados señores:

Con todo el interés me dirijo a ustedes para presentar a su consideración, estudio y aprobación la monografía titulada “**SISTEMA DE SOMBREADO DE PLAN DE ESTUDIOS UTB**”, como requisito para obtener el título de Ingeniero de Sistemas.

Atentamente,



---

**JORGE ALBERTO RODRIGUEZ SERRATO**

Cartagena de Indias, D. T y C. Junio de 2012

**Señores**

**COMITÉ CURRICULAR**

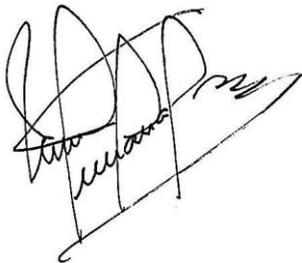
**UNIVERSIDAD TECNOLÓGICA DE BOLÍVAR**

Ciudad

Respetados señores:

Por medio de la presente me permito hacer entrega de la monografía titulada “**SISTEMA DE SOMBREADO DE PLAN DE ESTUDIOS UTB**”, para su estudio y evaluación, la cual fue realizada por los estudiantes **JORGE ALBERTO RAMIREZ SERRATO** y **JHON J JARNEY GUERRERO MARTINEZ** y de la cual acepto ser su director.

Atentamente,



---

**LIC. MOISES RAMON QUINTANA ALVAREZ**

Cartagena de Indias, D. T y C. Junio de 2012

## AUTORIZACIÓN

Yo, **JORGE ALBERTO RAMIREZ SERRATO**, identificada con cédula de ciudadanía número **1.143.355.825 de Cartagena**, autorizo a la Universidad Tecnológica de Bolívar, para hacer uso de mi trabajo de monografía y publicarlo en el catálogo on-line de la Biblioteca.



---

**JORGE ALBERTO RAMIREZ SERRATO**

Cartagena de Indias, D. T y C. Junio de 2012

## AUTORIZACIÓN

Yo, **JHON J JARNEY GUERRERO MARTINEZ**, identificada con cédula de ciudadanía número **1.137.222.859 de Cartagena**, autorizo a la Universidad Tecnológica de Bolívar, para hacer uso de mi trabajo de monografía y publicarlo en el catálogo on-line de la Biblioteca.

A handwritten signature in black ink, reading "Jhon J Jarney G.M.", positioned above a horizontal line.

---

**JHON J JARNEY GUERRERO MARTINEZ**

Cartagena de Indias, D. T y C. Junio de 2012

Dedicatoria de Jhon J Jarney Guerrero Martinez

**A Dios y a mi madre, por la quien  
estoy haciendo todo lo que hago.**

**A Bru por su gran apoyo en momentos difíciles.**

**Igualmente a mis profesores, a mis amigos,  
Quienes formaron de mi la persona que soy,  
y por ellos ser un buen profesional.**

Cartagena de Indias, D. T y C. Junio de 2012

Dedicatoria de Jorge Alberto Ramírez Serrato

**A mi mamá Ángela, a mi papá Jorge y mis hermanos**

**Sergio y Nicole por apoyarme, confiar en mí  
y hacer de mí un profesional, una mejor persona.**

## CONTENIDO

<b>INTRODUCCION</b> .....	XIV
<b>OBJETIVOS</b> .....	XVI
<b>OBJETIVOS GENERALES</b> .....	XVI
<b>OBJETIVOS ESPECIFICOS</b> .....	XVI
<b>1. CONTEXTUALIZACIÓN</b> .....	1
<b>1.1 DESCRIPCION DEL PROBLEMA</b> .....	1
<b>1.2 SOLUCIÓN DEL PROBLEMA</b> .....	2
<b>1.3 CARACTERISTICAS FUNCIONALES</b> .....	2
<b>1.4 CARACTERISTICAS NO FUNCIONALES</b> .....	3
<b>1.5 IMPACTO ESPERADO</b> .....	3
<b>1.6 EQUIPO DE TRABAJO</b> .....	4
<b>1.7 METODOLOGIA DE TRABAJO</b> .....	5
<b>1.8 PRODUCTO A ENTREGAR</b> .....	6
<b>1.9 CRONOGRAMA DE TRABAJO</b> .....	7
<b>2. ANALISIS DEL SOFTWARE</b> .....	8
<b>2.1 ESPECIFICACIONES DE REQUERIMIENTOS</b> .....	8
<b>2.1.1 USUARIOS DEL SISTEMA</b> .....	8
<b>2.1.2.1 REQUERIMIENTOS DEL SISTEMA</b> .....	8
<b>2.2 DIAGRAM DE CASOS DE USO</b> .....	11
<b>2.2.1 DETALLES DEL CASO DE USO</b> .....	12
<b>3. MODELO DE ANALISIS</b> .....	14
<b>3.1 MODELO DE COMPORTAMIENTO DEL SISTEMA</b> .....	14
<b>3.1.1 DIAGRAMA DE CLASES</b> .....	14
<b>3.1.2 DIAGRAMA DE SECUENCIA</b> .....	17
<b>4. MODELO DE DISEÑO</b> .....	18
<b>4.1 DIAGRAMA DE COMPONENTES</b> .....	21
<b>5. VALIDACIONES DEL SISTEMA</b> .....	22

<b>6. CONCLUSIONES .....</b>	<b>23</b>
<b>7. GLOSARIO.....</b>	<b>24</b>
<b>8.REFERENCIAS BIBLIOGRAFICAS .....</b>	<b>25</b>

## LISTA DE FIGURAS

ILUSTRACIÓN 1: MODELO BÁSICO DEL PATRÓN DE DISEÑO MVC [DISEÑO PROPIO].....	5
ILUSTRACIÓN 2: DIAGRAMA DE CASO DE USO DE LA INTERACCIÓN BÁSICA DEL SISTEMA [DISEÑO PROPIO].....	11
ILUSTRACIÓN 3: DIAGRAMA DE CLASES QUE HACEN PARTE DEL SISTEMA [DISEÑO PROPIO] .....	14
ILUSTRACIÓN 4: DIAGRAMA DE CLASES ADICIONAL QUE HACEN PARTE DEL SISTEMA PERO SOLO COMO AYUDA. [DISEÑO PROPIO] .....	15
ILUSTRACIÓN 5: DIAGRAMA DE SECUENCIA QUE REFLEJA EL FUNCIONAMIENTO DEL SISTEMA. [DISEÑO PROPIO].....	17
ILUSTRACIÓN 6: MAQUETADO DE DISEÑO DE LA INTERFAZ DE USUARIO [DISEÑO PROPIO] .....	18
ILUSTRACIÓN 7: DISEÑO DEL DIAGRAMA DE RELACIONES DE BASES DE DATOS [DISEÑO PROPIO] .....	19
ILUSTRACIÓN 8: DISEÑO DE DIAGRAMA DE COMPONENTES DEL SISTEMA. [DISEÑO PROPIO].....	21

## **ABREVIATURAS**

- *MVC*: Modelo Vista Controlador
- *ASP*: Paginas Activas de Servidor (Active Server Page)
- *HTML*: Lenguage de Marcado de HiperTexto (HyperText Markup Language)
- *CSS*: Hojas de Estilo en Cascada (Cascade Style Sheet)
- *AJAX*: JavaScript Asincronico y XML (Asynchronous JavaScript and XML)

## INTRODUCCIÓN

El propósito es desarrollar e implementar una aplicación web potencial basada principalmente en tecnología .NET, que permite visualizar de forma, fácil de entender visualmente el plan de estudios de los estudiantes.

El propósito principal es brindar una experiencia de usuario mucho mas agradable en cuanto a interpretación visual se refiere respecto al sistema actual. Para cualquier persona siempre es mas fácil interpretar datos de forma grafica que, de forma textual. Esto implica un beneficio al usuario ya que no tarda mucho tiempo en comprender los datos que se muestran.

El aspecto innovador de esta aplicación es la implementación de tecnologías actuales, tales como, ser una aplicación web .NET basada en MVC de ASPX implementadas en C#, HTML5, CSS3 y JavaScript para el manejo dinámico de la aplicación Web agregando características que enriquecen la experiencia de usuario tales como JQuery y AJAX y tecnología de servidor basada SQL Server de Microsoft, no perdiendo la flexible y capacidad de extensión, e incluso la integración con módulos adicionales para agregar funcionalidades nuevas.

Analíticamente las principales entidades afectadas con la implementación de esta aplicación, seria directamente los estudiantes, quienes ahorrarían tiempo en la interpretación de su situación académica actual. Indirectamente estarían todas aquellas entidades que tienen acceso a la información de los estudiantes o estén interesadas en acceder a esta información, tales como profesores, directivas de programas o asistentes académicos; brindándoles una interpretación mas clara de la situación actual del estudiante.

El sistema debe estar en la capacidad de brindar el sombreado completo del plan de estudios de todos y cada uno de los estudiantes registrados en el sistema actual SIRIUS y mostrar los detalles de cada una de sus asignaturas asociadas. Teniendo en cuenta las consideraciones tales como el periodo de ingreso del estudiante para el cual aplica su plan de estudio.

El equipo de desarrollo encargado de implementar esta aplicación esta conformado por Jorge Alberto Ramírez Serrato, programador web, bases de datos y diseñador de interfaz grafica y Jhon J Jarney Guerrero Martínez, programador .NET y documentador del sistema.

Esta misma aplicación web servirá de base para otros proyectos tales como “Intensión de Matricula” desarrollado por Luis Carlos Guzmán Almanza y David Leonardo Puerta Jaramillo y será un módulo de la aplicación “Control de Estudiantes” desarrollado por Carlos Andrés Bello Álvarez y Nadia Cordero Gómez.

## **OBJETIVOS**

### **Objetivos Generales**

Desarrollar una aplicación que muestre el plan de estudio en forma de malla curricular, que además brinde detalles de las materias aprobadas y en curso, con una interfaz limpia, agradable y fácil de usar, para mejorar la interpretación de la situación académica actual de los estudiantes con respecto a su avance académico.

### **Objetivos Específicos**

- Identificar las principales necesidades y funcionalidades que debe cumplir el sistema, para el análisis y diseño de la aplicación.
- Obtener un diseño previo del esquema del sistema con unos límites bien establecidos para la puesta en marcha del desarrollo de la aplicación.
- Desarrollar la aplicación basado en Tecnologías .NET de Microsoft y orientados a la Web.
- Entregar una aplicación funcional que cumpla con las necesidades y funcionalidades identificadas en el esquema y límites del proyecto.

## **1. CONTEXTUALIZACIÓN**

### **1.1 DESCRIPCIÓN DEL PROBLEMA**

El sistema actual de SIRUIS que brinda la universidad para la gestión y publicación de información académica relevante para los estudiantes y profesores, ofrece diversos servicios en línea que facilitan la consulta de información.

Entre sus servicios se encuentra, el de brindar el CárDEX (Información detalla del avance académico) del estudiantes y la matricula actual con sus respectivos detalles.

Para que un estudiante pueda acceder a verificar el avance académico actual, debe ingresar a la página del sistema SIRIUS, autenticarse y acceder a través de la siguiente ruta: Alumno y Ayuda Financiera, Registro de Alumnos, CárDEX Académico. Se selecciona luego el nivel y tipo de cardex que se desea visualizar.

A continuación aparece el historial académico del estudiante detallado, en forma textual.

Como proceso adicional también se puede visualizar el detalle de las materias actualmente matriculadas. A esta opción en el sistema SIRIUS actual se accede atreves de la ruta: Alumno y Ayuda Financiera, Registro de alumno, Detalles de Calificación. Se selecciona el periodo que se desea visualizar. Aparecerá el listado de cada una de las materias actuales y sus respectivos detalles.

Como la información se muestra de forma textual, para quien consulta es difícil de entender o interpretar a primera vista y puede tomar más tiempo de lo necesario cuando se desea tener una certeza rápida de la situación académica actual.

## **1.2 SOLUCIÓN DEL PROBLEMA**

Teniendo en cuenta las necesidades de información para la consulta de la situación académica actual de los estudiantes, se brindará la misma la información presentada de forma grafica con una estructura de malla curricular de plan de estudio para garantizar la interpretación rápida y la satisfacción de sus usuarios, se hace necesario diseñar e implementar un Sistema de Sombreado de Plan de Estudio.

El objetivo fundamental de este software es solucionar los problemas identificados y expresados en el numeral anterior derivados del uso del sistema actual utilizado para realizar este proceso.

Dentro de las características previstas de esta aplicación, se proponen las siguientes:

## **1.3 CARACTERÍSTICAS FUNCIONALES**

El sistema permitirá:

- Mostrar información básica del estudiante al ingreso del sistema tal como, nombre del estudiante, programa a la que pertenece, situación académica actual, promedio ponderado acumulado (PPA) y estado.
- Visualizar de primera mano el plan de estudio completo del estudiante a rasgos generales.
- Proporcionar la opción de sombrear el plan de estudio del estudiante para verificar su avance académico visualmente.
- Visualizar los detalles de cada una de las materias que ha cursado y está cursando actualmente. Con la capacidad de poder diferencia entre cada una de ellas.
- Adicionalmente si un estudiante ha cursado más materias de los que están en su plan de estudio, el software debe estar en la capacidad de mostrar esos cursos adicionales.

#### **1.4 CARACTERÍSTICAS NO FUNCIONALES**

- El aplicativo Web debe ser desarrollado principalmente en ASP.NET con MVC3 y Razor.
- Uso del motor de base de datos SQL Server 2008 R2 como proveedor de datos.
- Debe ofrecer una sesión segura al estudiante para evitar filtro de información personal.
- Brindar un aspecto visual agradable al usuario (Experiencia de Usuario).
- Capacidad de extensibilidad con otros aplicativos.

#### **1.5 IMPACTO ESPERADO**

El impacto esperado de esta herramienta está relacionado con la agilidad en la lectura e interpretación de la situación académica actual de los estudiantes de la Universidad Tecnológica de Bolívar, para la mayoría de programas y principalmente enfocado en el programa de Ingeniería de Sistemas brindando:

- Información oportuna y confiable sobre:
  - Plan de estudios general del estudiante.
  - Materias cursadas que hacen parte de su plan de estudios.
  - Materias actualmente en curso.
  - Materias cursadas que no hacen parte de su plan de estudios.
- Mejora las gestiones relacionadas con los procesos de matrícula de los estudiantes.
- Servicio rápido y ágil a través de internet, brindando disponibilidad a cualquier hora.

## 1.6 EQUIPO DE TRABAJO

Analizando las características y exigencias de este proyecto, se propone el siguiente equipo de trabajo:

<b>CARGO</b>	<b>RESPONSABILIDAD</b>	<b>PERFIL</b>
<b>Arquitecto de Software</b>	Será responsable por elaborar el diseño arquitectónico de la herramienta. Su responsabilidad tendrá que ver con la tecnología a utilizar para alcanzar objetivos de diseño como desempeño, seguridad, facilidad de uso mediante el uso de Tecnología Microsoft .NET	Ingeniero de Sistemas con conocimientos y experiencia en el diseño de sistemas multinivel y multicapa, utilizando Tecnología Microsoft .NET
<b>Programador</b>	Codificar y desarrollar los distintos modelos conceptuales del sistema mediante el lenguaje de programación compatibles con .NET.	Experiencia en el desarrollo de aplicaciones usando el lenguaje de programación relacionados con .NET
<b>Administrador de Base de datos</b>	Será responsable de crear los diseños de almacenamiento de datos persistentes y sus respectivas consultas para acceder a ellas de manera óptima y eficiente.	Conocimiento de servidores SQL Server de Microsoft y características como creación de procedimientos almacenados, consultas óptimas, creación de tablas normalizadas.
<b>Diseñador Web</b>	Realizará la interfaz grafica Web basada en HTML5, CSS3 y JavaScript para brindar la mejor experiencia de usuario	Conocimientos avanzados en CSS3, XHTML y framwork de JavaScript como JQuery.

## 1.7 METODOLOGÍA DE TRABAJO

Para la elaboración del proyecto de sombreado de plan de estudio basado en los datos ofrecido por el sistema SIRIUS actual a través de un modelo de vistas definido, se definió como metodología de desarrollo al desarrollo de software XP (Extreme Programming) por la agilidad y los beneficios que esta ofrece. Esta como tal es muy utilizada para proyectos de corto plazo y que cuenta con poco personal de desarrollo y consiste principalmente en tener al cliente o usuario final como miembro del equipo para lograr obtener rápidamente los requisitos necesarios para finalizar el proyecto de manera exitosa.

Además se optó por aplicar la arquitectura de MVC para la implementación de esta aplicación.

El patrón de arquitectura Model-View-Controller (MVC) separa una aplicación en tres componentes principales: el modelo, la vista y el controlador. El marco ASP.NET MVC proporciona una alternativa al patrón de ASP.NET para crear aplicaciones Web basadas en MVC. El ASP.NET MVC es un framework de presentación liviana, altamente comprobable, que está integrado con características ASP.NET existentes, tales como las páginas maestras y autenticación basada en membresía.

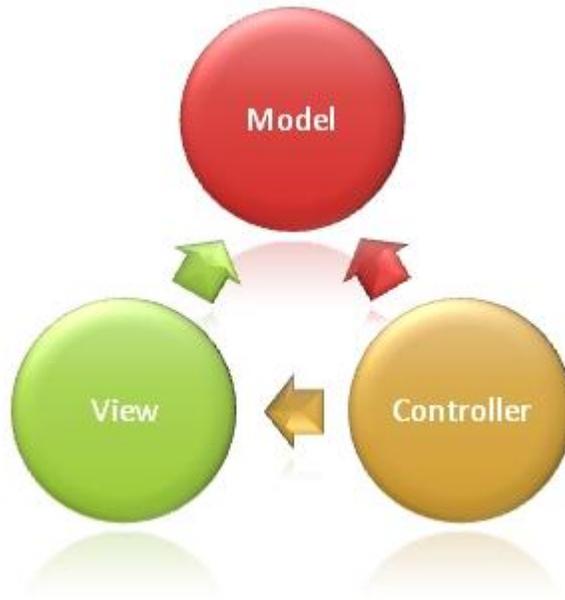


Ilustración 1: Modelo Básico del patrón de diseño MVC [Diseño Propio]

- **Modelos.** El modelo es la parte de la aplicación que implementa la lógica del dominio de aplicación de los datos. A menudo, los objetos del modelo se encargan de recuperar y almacenar el estado del modelo en una base de datos.

En pequeñas aplicaciones, el modelo es a menudo una separación conceptual en lugar de una física. Por ejemplo, si la aplicación sólo lee un conjunto de datos y los envía a la vista, la aplicación no tiene una capa de modelo físico. En ese caso, el conjunto de datos asume el papel de un objeto de modelo.

- **Vistas.** Las vistas son los componentes que muestran la interfaz de usuario de aplicación (UI). Normalmente, esta interfaz de usuario se crea desde los datos del modelo.
- **Controladores.** Los controladores son los componentes que manejan la interacción con el usuario, trabajan con el modelo y finalmente selecciona una vista para representarla en interfaz de usuario. En una aplicación MVC, la vista sólo muestra información; el controlador maneja y responde a la entrada del usuario.

## 1.8 PRODUCTOS A ENTREGAR

De acuerdo con la propuesta presente, el proyecto debe arrojar los siguientes productos:

1. Aplicativo Web que funcionará como módulo del proyecto “Aplicativo para Control de Estudiante UTB”.
2. Código fuente de la aplicación.
3. Este documento que brinda una vista general del sistema que se desarrolla.

## 1.9 CRONOGRAMA DE TRABAJO

El cronograma de trabajo que se propone en la siguiente tabla, está basado en la metodología anteriormente propuesta.

<b>Nombre de la actividad</b>	<b>Fecha de Inicio</b>	<b>Fecha de Terminación</b>	<b>Duración</b>	<b>Producto a entregar</b>
<b>Definición y especificación del sistemas</b>	4 de Junio	9 de Junio	5 días	Documento de especificación de requerimientos del sistemas
<b>Diseño del sistemas</b>	13 de Junio	16 de Junio	3 días	Documento de diseño del sistemas
<b>Diseño físico del sistemas</b>	17 de Junio	31 de Junio	3 semanas	Los fuentes del sistema
<b>Pruebas del sistema</b>	23 de Junio	31 de Junio	2 semanas	Documento de pruebas del sistema
<b>Instalación del sistema y puesta en marcha</b>	6 de Julio	7 de Julio	2 días	El sistema desplegado en los servidores ofrecidos por la Universidad, mas concretamente de la facultad de Ingeniería de Sistema.

## 2. ANALISIS DEL SOFTWARE

### 2.1 ESPECIFICACIÓN DE REQUERIMIENTO

#### 2.1.1 USUARIOS DEL SISTEMA

Los principales usuario del sistema del sombrado del plan de estudio son:

- Estudiantes
- Profesores

Para todos los usuarios cumplirá la misma funcionalidad, para la que fue diseñado el software; presentar de forma grafica, con estilo de malla curricular el plan de estudio del estudiante y un sombreado que indica el progreso. Además brindar una vista mas clara de la situación actual de dicho estudiante.

#### 2.1.2 REQUERIMIENTOS FUNCIONALES

##### 2.1.2.1 REQUERIMIENTOS DEL SISTEMA

<b>ID</b>	<b>R1</b>
<b>Descripción</b>	<b>Autenticación de usuario</b>
<b>Descripción Detallada</b>	El sistema no autenticará el usuario, solo será capaz de tomar los datos de usuario que vendrán autenticados de una fuente externa (Web Service).
<b>Entrada</b>	Datos de usuario validados desde otra fuente (Web Service).
<b>Salida</b>	Ninguna
<b>Prioridad</b>	Baja

<b>ID</b>	<b>R1.1</b>
<b>Descripción</b>	<b>Solicitud de información de usuario</b>
<b>Descripción Detallada</b>	Luego de que el usuario es recibido como autentico desde la fuente externa, se procede a obtener la información básica del usuario para una primera vista.
<b>Entrada</b>	Datos de usuario validados desde otra fuente(Web Service).
<b>Salida</b>	Datos básicos del usuario: Nombre completo, carrera a la que pertenece, su situación académica actual, promedio ponderado acumulado, estado.
<b>Prioridad</b>	Baja

<b>ID</b>	<b>R2</b>
<b>Descripción</b>	<b>Carga del plan de estudios</b>
<b>Descripción Detallada</b>	Luego de tener la información básica del usuario, se procede a la carga del plan de estudios académico completo del estudiante.
<b>Entrada</b>	Datos de usuario validados desde otra fuente (Web Service).
<b>Salida</b>	Plan de estudios completo general y estructurado del estudiante.
<b>Prioridad</b>	Alta

<b>ID</b>	<b>R3</b>
<b>Descripción</b>	<b>Sombreado del plan de estudios</b>
<b>Descripción Detallada</b>	Luego de la carga del plan de estudios se procede brinda la opción de sombrear según los siguiente criterios: Si el estudiante ya vio la materia y aprobó (color verde), las materias que actualmente esta cursando el estudiante (color azul) y las que aun no ha aprobado (color gris).
<b>Entrada</b>	Ninguna
<b>Salida</b>	Plan de estudios sombreado según los criterios antes mencionados.
<b>Prioridad</b>	Alta

<b>ID</b>	<b>R4</b>
<b>Descripción</b>	<b>Detalles de materias</b>
<b>Descripción Detallada</b>	El usuario tiene la capacidad de conocer los detalles de cada una de las materias que ha cursado y esta cursando pero no de las que no ha cursado.
<b>Entrada</b>	Código de materia a consultar
<b>Salida</b>	Detalles de la materia
<b>Prioridad</b>	Alta

<b>ID</b>	R5
<b>Descripción</b>	<b>Salida del Sistema</b>
<b>Descripción Detallada</b>	Permite al usuario salir del sistema de forma segura evitando que otros puedan consultar su información.
<b>Entrada</b>	Ninguna
<b>Salida</b>	Ninguna
<b>Prioridad</b>	Alta

## 2.2 DIAGRAMA DE CASOS DE USO

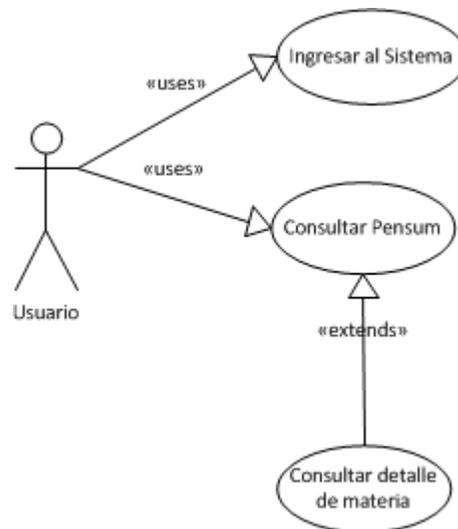


Ilustración 2: Diagrama de caso de uso de la interacción básica del sistema  
[Diseño propio]

## 2.2.1 DETALLES DEL CASO DE USO

<b>ID:</b> CU001
<b>Nombre:</b> Consultar plan de estudios.
<b>Prioridad:</b> Alta.
<b>Satisface Requerimientos:</b> R1.1, R2, R3, R4,
<b>Descripción:</b> El sistema permite al usuario visualizar de primera vista su plan de estudios general y el detalle de las materias que ha cursado o esta cursando.
<b>Actor Principal:</b> <ul style="list-style-type: none"><li>• Estudiante</li></ul>
<b>Actor Secundario:</b> <ul style="list-style-type: none"><li>• Profesores</li><li>• Asistentes Académicos</li></ul>
<b>Precondiciones:</b> <ul style="list-style-type: none"><li>• El servidor está disponible.</li><li>• El usuario debe haber iniciado el sistema.</li><li>• La conexión con la Base de Datos está activa.</li><li>• El usuario debe indicar que quiere visualizar su sombrero.</li></ul>
<b>Post Condición:</b> Visualizo plan de estudios de forma correcta.
<b>Objetos Frontera:</b> Interfaz de sombreado de plan de estudios estudiante <b>Objetos Entidad:</b> Estudiante, plan de estudios, Materias <b>Objetos Control:</b> Validaciones y lógica de construcción del plan de estudios del estudiante a partir de los datos de la base de datos.

### **Flujo de Eventos**

Al momento del ingreso al sistema el usuario pasa por un proceso de autenticación con su código. Este proceso se realiza a través de un servicio de autenticación de usuario de un sistema externo que proporciona la validación y verificación del usuario.

Para acceder al aplicativo el usuario después de autenticado ingresa mediante un link a la opción de sombreado del plan de estudios del estudiante.

Una vez el usuario ingrese, inmediatamente podrá obtener la información de su plan de estudios académico actual, refiriéndose solo a las materias que se encuentran en el plan de estudios general del estudiante.

La función principal del sistema se ejecuta al momento de hacer click en el botón sombreado, donde se cargan las materias que ha cursado el estudiante y las que está cursando actualmente si es que tiene materias en curso, mostrando los detalles más relevantes de la materia.

### **Flujo alternativo**

Adicionalmente, debido a la posibilidad de poder cursar materias que no son propias de la carrera o adicionales a esta, el aplicativo está en la capacidad de poder mostrar esas materias que no son propias del plan de estudios al que pertenece el estudiante, en un listado a parte con sus respectivos detalles.

### 3. MODELO DE ANÁLISIS

#### 3.1 MODELO DE COMPORTAMIENTO DEL SISTEMA

##### 3.1.1 DIAGRAMA DE CLASES

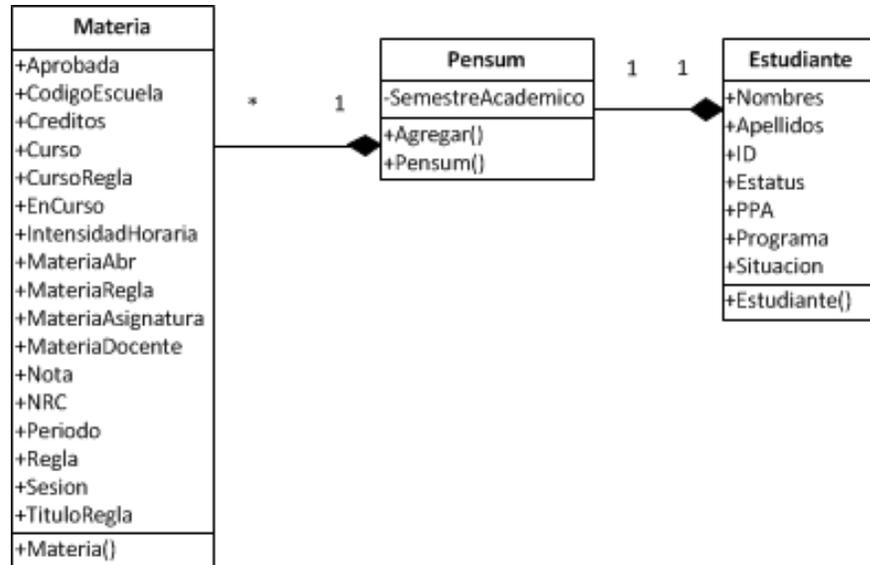


Ilustración 3: Diagrama de clases que hacen parte del sistema [Diseño propio]

A continuación una descripción textual detalla de cada una de las clases mostradas en el gráfico anterior, para ser más claro en sus relaciones y se más descriptivos en los métodos que son implementados:

Clase **Estudiante**: Es una clase que representa a la entidad estudiante que almacena todos los datos básicos del estudiante, no implementa métodos relevantes. Hace parte del componente de lógica de negocio de la arquitectura de MVC.

Clase **Materia**: Es una clase que representa la información básica de una materia, no implementa métodos relevantes. Hace parte del componente de lógica de negocio de la arquitectura de MVC.

Clase **Pensum**: Es una clase que representa el plan de estudios completo de un estudiante tiene una relación de agregación con la clase materia, y tiene métodos relevantes para agregar dinámicamente materias para formar el plan de estudios personalizado según sea el caso de cada estudiante. Hace parte del componente de lógica de negocio de la arquitectura de MVC.

- Método **Agregar**: Este método permite agregar materias al plan de estudios tantas como sean necesarias y se rige sobre una lógica de no repetición de materias agregadas.

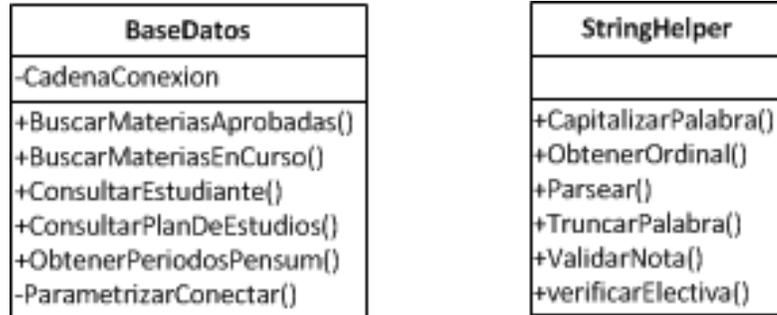


Ilustración 4: Diagrama de clases adicional que hacen parte del sistema pero solo como ayuda. [Diseño propio]

Clase **BaseDatos**: Es una clase estática que almacena todos los métodos necesarios para el acceso a datos de la aplicación, y maneja toda la lógica respecto a la construcción del plan de estudios, preparándola antes de ser enviada a la capa de la vista. Hace parte del componente de lógica de negocio de la arquitectura de MVC.

- Campo **CadenaConexion**: Este campo de la clase almacena la cadena de conexión proporcionada en el archivo Web.config del servidor IIS. Se usa para realizar el acceso al servidor de base de datos de SQL Sever.
- Método **ConsultarEstudiante**: Verifica la existencia de un estudiante en la base de datos y trae toda la información básica del estudiante en un objeto tipo **Estudiante**. En caso contrario devolvería null.
- Método **ConsultarPlanDeEstudio**: Consulta el PLAN DE ESTUDIOS completo del estudiante y lo almacena en un objeto tipo **Pensum**.
- Método **BuscarMateriasAprobadas**: Una vez obtenido y cargado el PLAN DE ESTUDIOS del estudiante en objeto tipo **Pensum**, este método se encarga de modificar la información de las materias contenidas en el PLAN DE ESTUDIOS para las materias aprobadas por el estudiante.
- Método **BuscarMateriasEnCurso**: Una vez obtenido y cargado el PLAN DE ESTUDIOS del estudiante en objeto tipo **Pensum**, y con sus respectivas materias aprobadas, este método se encarga de modificar la información de las materias contenidas en el PLAN DE ESTUDIOS para las materias en curso del estudiante.
- Método **ObtenerPeriodosPensum**: Este método obtienen todos los periodos para los cuales son aplicables los PLAN DE ESTUDIOS.

- Método **ParametrizarConectar**: Este método privado, solo es usado en la clase para **BaseDatos** para parametrizar y conectar con la base de datos y ejecutar los procedimientos almacenados que sean necesarios.

Clase **StringHelper**: Esta clase es una clase de métodos de extensión que ayudan al manejo de cadenas. Por lo general los registros de la base de datos están en palabras todas en mayúsculas; esta clase permite formatear las cadenas y brindar un mejor aspecto a los datos que vienen de la base de datos para ser presentado.

- Método **CapitalizarPalabra**:  
Coloca todas las iniciales de una palabra en mayúscula.
- Método **ObtenerOrdinal**: Obtiene el cardinal de un número ingresado.
- Método **Parsear**: Método creado para convertir los datos de la base de datos en cadenas.
- Método **TruncarPalabra**:  
Método para recortar palabras dependiendo de ciertos criterios.
- Método **ValidarNota**:  
Método creado para validar una nota cuando es de tipo.
- Método **VerificarElectiva**: Valida dependiendo del código y la regla si es una electiva o no.

### 3.1.2 DIAGRAMA DE SECUENCIA

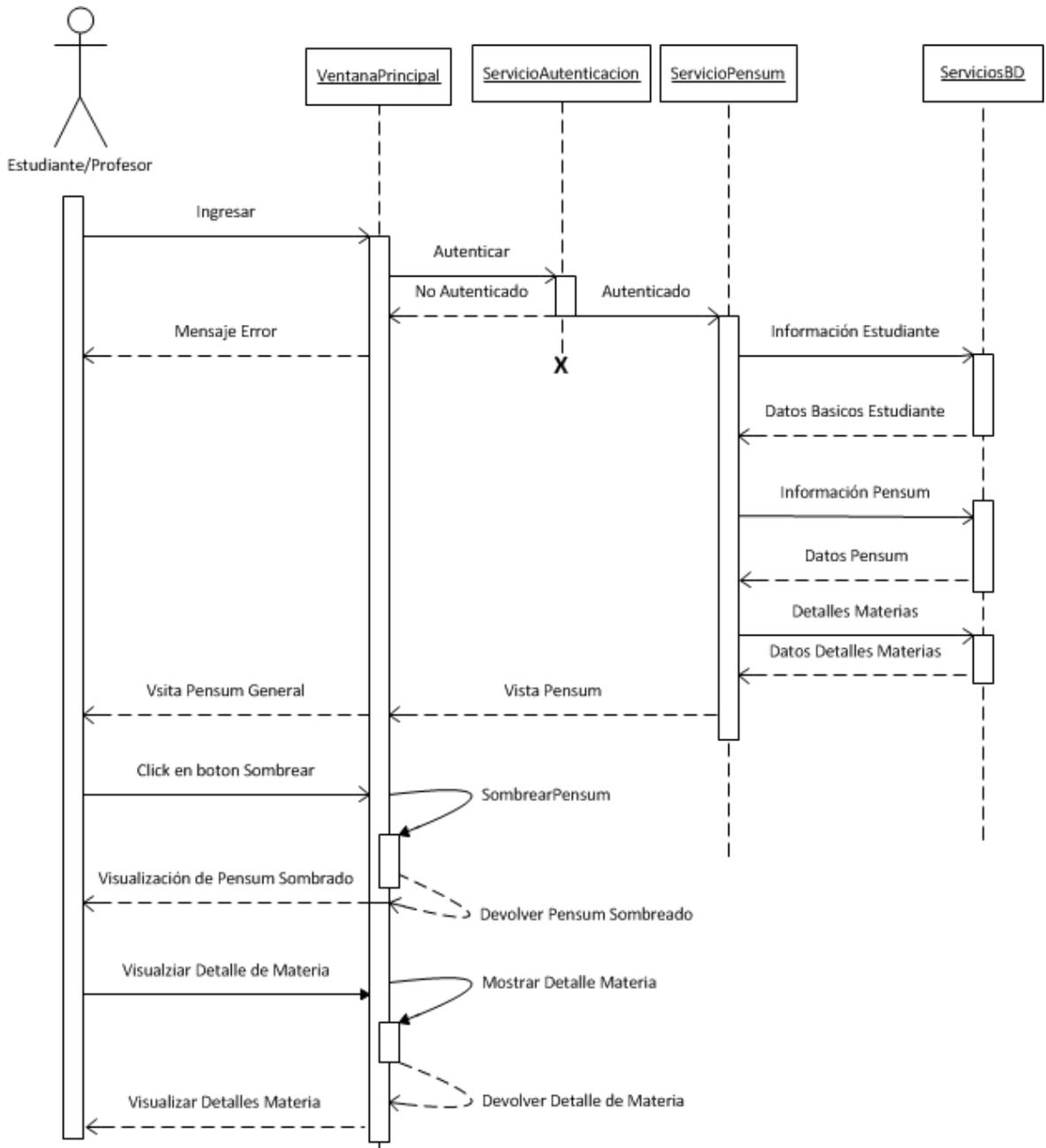


Ilustración 5: Diagrama de secuencia que refleja el funcionamiento del sistema.  
[Diseño propio]

## 4. MODELO DE DISEÑO

Pensando en una interfaz de usuario agradable y en relación a las tendencias actuales de diseño, se concreto que el resultado del software debería estar diseñado de la siguiente estructura en forma visual:

Universidad Tecnológica de Bolívar		Sombrear Pensum								
© NOMBRE ADRIAN ANDRES MORELOS HENRIQUEZ		1er Semestre	2do Semestre	3er Semestre	4to Semestre	5to Semestre	6mo Semestre	7mo Semestre	8vo Semestre	9no Semestre
	PROGRAMA <b>Ingeniería De Sistemas</b>	Química CBAS3503	Algebra Lineal CBAS2403	Matematica Discreta ISIS2403	Metodos Numericos CBAS2453	Entorno Economico ECON1533	Electiva Empresarial ISISEA13	Gestion De Proyectos AEMP1733	Inteligencia Artificial ISIS2443	Practica Profesional ISIS2903
	Situación <b>Situacion Academica Normal</b>	Calculo I CBAS2413	Calculo II CBAS2423	Calculo III CBAS2433	Ecuaciones Diferenciales Y CBAS2443	Teoria De La ISIS2413	Lenguajes De Programacion ISIS2423	Sistemas Dinamicos Y ISIS2723	Etica Y Profesionalismo CHUM1333	Programacion Java ISIS053
	PROMEDIO ACUMULADO <b>375</b>	Algoritmos ISIS1403	Fisica I CBAS3403	Fisica II CBAS3413	Estructura De Datos ISIS1433	Arquitectura Del Computador IETR2413	Modelos Cuantitativos ISIS2433	Sistemas Operativos ISIS3713	Electiva De Huma. ISIS4033	
	ESTADO <b>Activo</b>	Seminario De Ing. ISIS4403	Programacion I ISIS1413	Programacion II ISIS1423	Ingles IV CHUM1933	Base De Datos ISIS4413	Comunicacion Y Redes ISIS3703	Ingenieria De Software ISIS2713	Electiva De Ingenieria ISIS5023	
		Ingles I CHUM1903	Ingles II CHUM1913	Ingles III CHUM1923	Historia Social Y ISIS4023	Fisica III CBAS3423	Ingenieria De Software ISIS2703	Constitucion Politica Y CHUM1323	Electiva De Ingenieria ISIS5033	
		Habilidades Del Pensamiento CHUM1203	Composicion De Textos CHUM1213	Leer La Tierra ISIS4013	Estadistica I CBAS2603	Ingles V CHUM1943	Ingles VI CHUM1953	Electiva De Ingenieria ISIS5013	Electiva De Ingenieria ISIS5043	

Ilustración 6: Maquetado de diseño de la interfaz de usuario [Diseño propio]

Basándonos en los requerimientos, la primera fase de desarrollo se basa en la creación de procedimientos almacenados que ofrezcan la información de nuestro interés.

La fuente de datos básica proporcionada por la Universidad, fueron unas vistas de la base de datos del sistema de SIRIUS actual.

En nuestro caso solo necesitaremos, básicamente la información del estudiante, su plan de estudios académico en relación con su carrera y las materias relacionadas a su plan de estudios.

Se diseño el siguiente diagrama de base de base de datos y sus relaciones para obtener dicha información.

Para ser más específicos, se crearon unas script que generaron las tablas en SQL Server que son copias exactas de las vistas proporcionadas para trabajar. Esta implementación se realizó debido a que es mas rápido hacer consultas sobre las tablas del mismo motor de bases de datos, que hacerlas sobre las vistas que se proporcionan desde el servicio externo que nos ofrece el sistema actual de SIRIUS. Además de saber que la información que manejan dichas vistas son de actualización periódica que es de antemano conocida.



Para la consulta y las relaciones en las tablas se implementaron los siguientes Procedimientos Almacenados:

- **SP\_Sombreado\_ActualizarTablas:** Es un procedimiento almacenado que se ejecuta como una tarea programa de cada cierto periodo de tiempo. De modo que actualice las tablas que son copias de las vistas ofrecidas.
- **SP\_Sombreado\_ConsultarEstudiante:** Consulta la información básica del estudiante como el nombre, la carrera, el promedio ponderado acumulado. Esta información es obtenida de la tabla **ESTUDIANTES\_**.
- **SP\_Sombreado\_ObtenerMateriasAprobadas:** Se toma la relación entre la tabla **ESTUDIANTES\_**, **PENSUM\_ESTUDIANTES\_**, **AREAS\_**, **BIBLIOTECAS\_AREAS** y **CARDEX\_** para obtener las materias aprobadas por el estudiante.
- **SP\_Sombreado\_ObtenerMateriasEnCurso:** Se toma la relación entre la tabla **ESTUDIANTES\_**, **PENSUM\_ESTUDIANTES\_**, **AREAS\_**, **BIBLIOTECAS\_AREAS** y **MATRICULA\_** para obtener las materias en curso del estudiante.
- **SP\_Sombreado\_ObtenerPensumPorCodigoEstudiante:** Obtiene el plan de estudios completo del estudiante, sin detalles específicos de las materias de las tablas **AREAS\_** y **BIBLIOTECAS\_AREAS\_**.
- **SP\_Sombreado\_ReporteMateriasCursadas:** Obtiene la información necesaria para imprimir un reporte con las materias cursadas.
- **SP\_Sombreado\_ReporteMateriasEnCurso:** Obtiene la información necesaria para imprimir un reporte con las materias en curso.

## 4.1 DIAGRAMA DE COMPONENTES

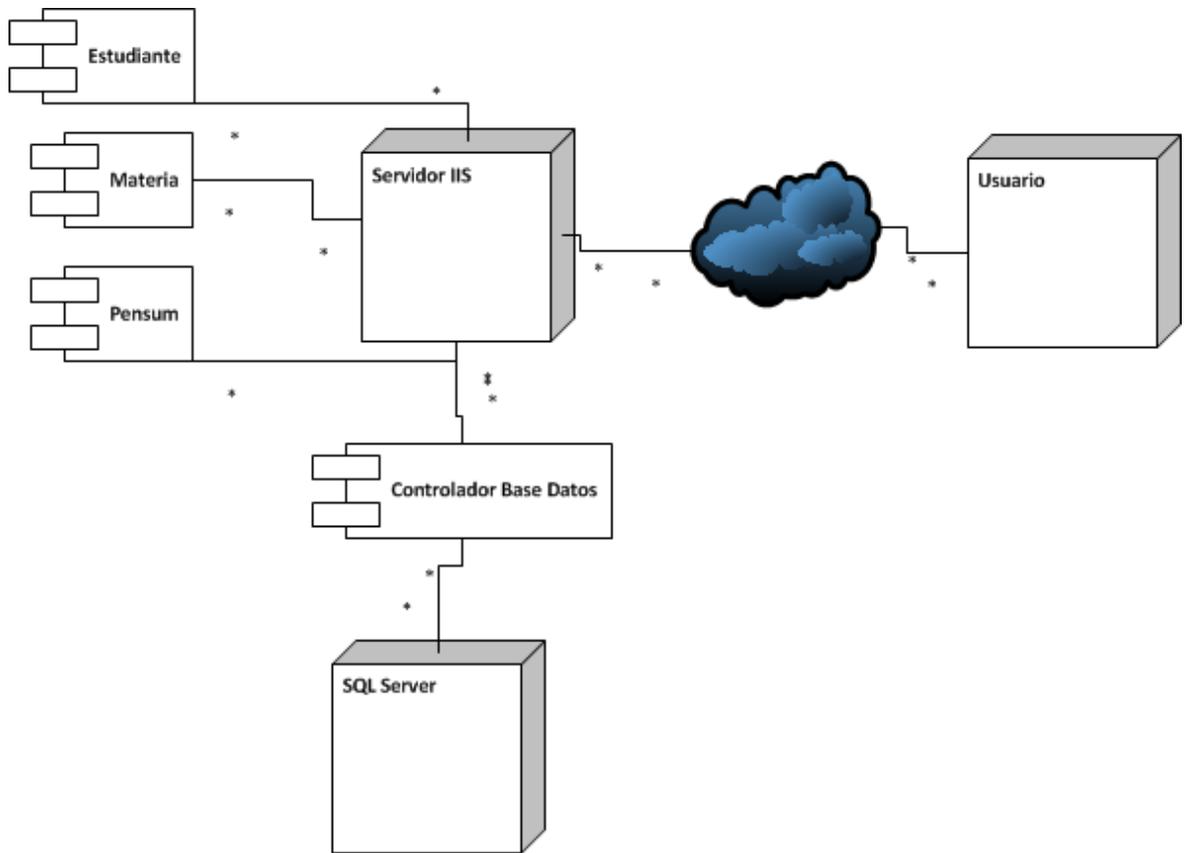


Ilustración 8: Diseño de Diagrama de Componentes del sistema. [Diseño Propio]  
Muestra los componentes del sistema y sus relaciones en el posible despliegue.

## 5. VALIDACIONES DEL SISTEMA

Para la verificación continua de los resultados esperados por la aplicación, se optó por la implementación de la metodología de desarrollo basado en pruebas. Desarrollo basado en pruebas (TDD) es una técnica avanzada de la utilización de pruebas automatizadas de unidad para impulsar el diseño de software y la fuerza de desacoplamiento de las dependencias. El resultado del uso de esta práctica es un conjunto completo de pruebas unitarias que se pueden ejecutar en cualquier momento para proporcionar información que el software todavía está trabajando. Esta técnica está fuertemente enfatizada por aquellos que utilizan metodologías ágiles de desarrollo.

El lema de desarrollo basado en pruebas es "Rojo, Verde, Refactor".

- Rojo: Crear una prueba y hacer que falle.
- Verde: Hacer pasar la prueba por cualquier medio necesario.
- Refactorizar: Cambie el código para eliminar la duplicación en su proyecto y para mejorar el diseño al tiempo que garantiza que todas las pruebas todavía pasen.

El ciclo de Rojo / Verde / Refactor se repite con mucha rapidez para cada nueva unidad de código.

Se planteó un escenario de pruebas basado en los datos obtenidos de la base de datos que es la base de la ejecución del programa. Como los datos son de una actualización poco periódica se optaron por tomar una muestra de pruebas de datos que de ante mano se sabe se debe obtener y no cambiarán por un cierto periodo de tiempo. Basado en esta metodología se procedió a plantear las pruebas de la siguiente manera.

Consistencia de datos:

- Se examinaron y escogieron aleatoriamente datos de estudiantes. Estos son los datos esperados.
- Se implementaron las siguientes clases para la carga de información:
  - **Estudiante**
  - **Materia**
  - **Pensum**
- Se implementó una clase de acceso a datos (**BaseDatos**) independiente que proporciona los datos que se van a probar.

## 6. CONCLUSIONES

En el proceso de desarrollo de software se deben tener claro cuales son las funcionalidades que debe cumplir a cabalidad la aplicación. Para eso se deben entender desde un comienzo las necesidades que el cliente o usuario final tiene. El propósito del software es cumplir completamente esa funcionalidad y además debe dar la oportunidad de ser extendido en un futuro.

El primer proceso a cumplir en el desarrollo de un aplicativo software es el análisis de requerimientos y funcionalidades que debe abarcar dicho software. Esta es una fase crucial ya que define un bosquejo del producto software esperado. Al final se debe tener claro que funcionalidad debe ejecutar el programa para empezar a desarrollarlo.

A continuación se empieza con un análisis del “*como*” se debe construir la aplicación. El principal producto derivado de esta actividad conlleva a una amplia documentación de software, que plasma la arquitectura del mismo y todas sus funcionalidades.

Acto seguido se procede a codificar, es un proceso basado en las actividades de requerimientos y análisis y que empieza a formar el producto software esperado, con funcionalidades que cumplan los requerimientos y se amolden al modelo de análisis.

Paralelo al acto de desarrollo esta el proceso de pruebas, que garantiza que los resultados de cada funcionalidad esperada por el software sean los deseados, garantizado así, datos confiables.

Finalmente se tiene un producto que cumple con las funcionalidades y requerimientos deseados, genera datos confiables y esta documentado para continuar con su desarrollo previo. En cuanto al proceso de calidad, solo es garantizado por el proceso de desarrollo basad en pruebas que, aunque no es una forma de asegurar la calidad en el software, garantiza la consistencia en la información.

En todo el desarrollo de la aplicación hay siempre dificultades en cualquiera de las etapas, como por ejemplo, la curva de aprendizaje con respecto a nuevas tecnologías, desarrollo o implementación de algunos procesos o dudas respecto a como trabajar con nuevas herramientas. Todos estos inconvenientes atribuyen al crecimiento profesional e intelectual del grupo de trabajo.

## 7. GLOSARIO

**.NET:** es un framework de Microsoft que hace un énfasis en la transparencia de redes, con independencia de plataforma de hardware y que permita un rápido desarrollo de aplicaciones. Mas información en <http://www.microsoft.com/net/>

**Modelo Vista Controlador:** es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de negocio en tres componentes distintos.

**Active Server Pages:** también conocido como **ASP clásico**, es una tecnología de Microsoft del tipo "lado del servidor" para páginas web generadas dinámicamente. Mas información en <http://www.asp.net/>.

**C#:** es un lenguaje de programación orientado a objetos desarrollado y estandarizado por Microsoft como parte de su plataforma .NET.

**HyperText Markup Language:** versión 5 es la quinta revisión importante del lenguaje básico de la World Wide Web, HTML. Mas información en <http://www.w3.org/TR/html5/>

**CSS:** es un lenguaje usado para definir la presentación de un documento estructurado escrito en HTML o XML. Mas información <http://www.w3.org/Style/CSS/>

**JavaScript:** es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico. Mas información en <https://developer.mozilla.org/es/JavaScript>

**jQuery:** es una biblioteca de JavaScript, que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX a páginas web. Más información en [http://docs.jquery.com/Main\\_Page](http://docs.jquery.com/Main_Page)

**Asynchronous JavaScript And XML:** es una técnica de desarrollo web para crear aplicaciones interactivas o RIA (*Rich Internet Applications*). Estas aplicaciones se ejecutan en el cliente, es decir, en el navegador de los usuarios mientras se mantiene la comunicación asíncrona con el servidor en segundo plano.

**Microsoft SQL Server:** Es un sistema para la gestión de bases de datos producido por Microsoft basado en el modelo relacional.

**Web Service:** es una Tecnología que utiliza un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones.

## **8. REFERENCIAS BIBLIOGRAFICAS**

Griffiths Ian, Adams Matthew, and Liberty Jesse. Programming C# 4.0. Sexta Edición. O'Reilly, Agosto 2010.

Flanagan David. JavaScript The Definitive Guide. Sexta Edición. O'Reilly. Marzo 2011.

Galloway Jon, Haack Phil, Wilson Brad, Allen K. Scott, ASP:NET MVC 3. Indianapolis. John Wiley & Sons, Inc.

Spaanjaars Imar, Beginning ASP.NET 4 in C# and VB. Indianapolis. John Wiley & Sons, Inc.