

**MÓDULO DE GESTIÓN PARA LA INSCRIPCIÓN DE MATERIAS EN
UNIMINUTO**

WILSON FABIÁN HERRERA

CRISTIAN CAMILO PÉREZ

CORPORACIÓN UNIVERSITARIA MINUTO DE DIOS

FACULTAD DE INGENIERÍA

DEPARTAMENTO DE INFORMÁTICA Y ELECTRÓNICA

PROGRAMA TECNOLOGÍA EN INFORMÁTICA

BOGOTÁ

I-2014

**MÓDULO DE GESTIÓN PARA LA INSCRIPCIÓN DE MATERIAS EN
UNIMINUTO**

WILSON FABIÁN HERRERA

CRISTIAN CAMILO PÉREZ

Trabajo de grado para optar al título de

Tecnólogos en informática

Asesor:

Ing. Luis Eduardo Pérez Peregrino

CORPORACIÓN UNIVERSITARIA MINUTO DE DIOS

FACULTAD DE INGENIERÍA

DEPARTAMENTO DE INFORMÁTICA Y ELECTRÓNICA

PROGRAMA TECNOLOGÍA EN INFORMÁTICA

BOGOTÁ

I-2014

Nota de aceptación

Firma del presidente del jurado

Firma del jurado N°1

Firma del jurado N°2

Firma del jurado N°3

Bogotá D.C. Julio de 2014

DEDICATORIA

Dedicado a nuestros padres y familiares, quienes nos apoyaron durante nuestro proceso de formación académica y siempre nos alentaron a seguir adelante sin importar las dificultades.

AGRADECIMIENTOS

Agradecemos a la Corporación Universitaria Minuto de Dios, por brindarnos la oportunidad de formarnos no solo académicamente, sino también humanamente, al departamento de tecnología en informática, el cual apoyó siempre nuestra formación académica, nos motivó a seguir adelante y fomentó nuestro amor por la informática con aspectos como las maratones de programación y los semilleros de investigación.

Agradecemos a nuestros padres, pues sin su ayuda y apoyo no hubiera sido posible lograr nuestro objetivo de terminar nuestra carrera de tecnología.

Agradecemos a todos los docentes de la institución, que con su dedicación, conocimiento y exigencia nos ayudaron a tener unas bases muy sólidas para la elaboración y desarrollo de este proyecto.

TABLA DE CONTENIDO

	Pág.
INTRODUCCIÓN	12
1.1 Título del Proyecto	13
1.2 Planteamiento del Problema	13
1.3 Alcance y Justificación	14
1.4 Objetivos.....	15
1.4.1 Objetivo General	15
1.4.2 Objetivos Específicos	15
2. INGENIERÍA DEL PROYECTO	16
2.1. Metodología y Modelo de Desarrollo.....	16
2.2. Patrón de arquitectura	20
3. ANÁLISIS Y DISEÑO	22
3.1. Definición de Requerimientos	22
3.1.1. Requerimientos funcionales.....	22
3.1.2. Requerimientos No funcionales	23
3.2. Descripción del Sistema Propuesto	24
3.2.1. Diagrama Entidad Relación	25
3.2.2. Diccionario de Datos	26
3.2.3. Actores y Componentes del Sistema Propuesto	28
3.2.4. Propuesta de Interfaz Gráfica.....	31
3.3. Diseño del Sistema Propuesto	35
3.3.1. Diagramas Estáticos	35
3.3.2. Diagrama de Clases.....	35

3.3.3.	Diagramas de Objetos	38
3.3.4.	Diagramas Dinámicos	39
3.3.5.	Diagramas de casos de uso	39
4.	DESARROLLO	51
5.	GLOSARIO	54
6.	CONCLUSIONES	55
7.	REFERENCIAS.....	56

LISTAS DE GRÁFICOS

	Pág.
Gráfico N° 1 - Etapas de desarrollo del proyecto	17
Gráfico N° 2 - Modelo Vista Controlador	20
Gráfico N° 3 - Diagrama Entidad Relación	26
Gráfico N° 4 - Iniciar / Crear Sesión	31
Gráfico N° 5 - Usuario no registrado.....	32
Gráfico N° 6 - Pensum (Materias Vistas).....	32
Gráfico N° 7 - Materias a inscribir	33
Gráfico N° 8 - Horarios	34
Gráfico N° 9 - Módulo Administrador	34
Gráfico N° 10 - Diagrama de Clases	36
Gráfico N° 11 - Modelo de Objetos	38
Gráfico N° 12 -Caso de Uso 1	40
Gráfico N° 13 - Caso de Uso 2	41
Gráfico N° 14 - Caso de Uso 3	42
Gráfico N° 15 - Caso de Uso 4	43
Gráfico N° 16 - Caso de Uso 5	44
Gráfico N° 17 - Caso de Uso 6	45
Gráfico N° 18 - Caso de Uso 7	46
Gráfico N° 19 - Caso de Uso 8	47
Gráfico N° 20 - Caso de Uso 9	48
Gráfico N° 21 - Caso de Uso 10	49
Gráfico N° 22 - Diagrama General Casos de Uso	50

LISTAS DE TABLAS

	Pág.
Tabla N° 1 - Entidad USUARIOS	26
Tabla N° 2 - Entidad CARRERA	27
Tabla N° 3 - Entidad MATERIAS_VISTAS	27
Tabla N° 4 - Entidad MATERIAS	27
Tabla N° 5 - Entidad NRC	28
Tabla N° 6 - Entidad ETADISTICAS	28
Tabla N° 7 - Descripción de las Clases	38
Tabla N° 8 - Caso de Uso 1	40
Tabla N° 9 - Caso de Uso 2	41
Tabla N° 10 - Caso de Uso 3	42
Tabla N° 11 - Caso de Uso 4	43
Tabla N° 12 - Caso de Uso 5	44
Tabla N° 13 - Caso de Uso 6	45
Tabla N° 14 - Caso de Uso 7	46
Tabla N° 15 - Caso de Uso 8	47
Tabla N° 16 - Caso de Uso 9	48
Tabla N° 17 - Caso de Uso 10	49

RESUMEN

El propósito del proyecto “Módulo de gestión para la inscripción de materias en Uniminuto”, es el de crear una herramienta orientada a la web, la cual ayude a los alumnos de Uniminuto en su proceso de inscripción de materias, al comienzo de cada semestre, brindándoles control sobre sus asignaturas vistas y pendientes, además de mostrar combinaciones de horarios para seleccionar el que mejor se ajuste a las necesidades de cada estudiante.

El proyecto comienza con un proceso de investigación del sistema actual y las posibles mejoras que necesita, posteriormente se hace un respectivo análisis, alcance, objetivos y se estiman los requerimientos. Para finalizar se selecciona un modelo de desarrollo, en este caso UP (Proceso Unificado), también se define que se va a utilizar el lenguaje unificado de modelamiento UML y se procede a su desarrollo.

ABSTRACT

The purpose of the " Módulo de gestión para la inscripción de materias en Uniminuto" project is to create a web-oriented tool, which helps Uniminuto students in the process of registration of subjects at the beginning of each semester by giving them control over their subjects, and outstanding views, and show schedules combinations to select the one that best fits the needs of each student.

The project begins with a process of investigation of the current system and possible improvements it need, then a respective analysis, scope, objectives and requirements is estimated. To end a development model is selected, in this case UP (Unified Process) is also defined to be the unified modeling Language using UML and proceeds to language development.

INTRODUCCIÓN

La Corporación Universitaria Minuto de Dios “Uniminuto” es una institución educativa que se diferencia de las demás en varios aspectos, uno de ellos es el brindar a sus estudiantes la posibilidad de conocer a personas de los distintos programas académicos presentes en la universidad, mediante las materias transversales o de núcleo común.

El proceso de inscripción de materias es realizado por los estudiantes al principio de cada semestre. Consiste en buscar las materias de su carrera y las materias del núcleo común (que se encuentran dispersas en distintos archivos Excel) y organizarlas de manera manual cuidando que no se cruce el horario de una materia con el horario de otra.

La necesidad de crear un sistema que organice los horarios automáticamente, surge a partir de los diversos problemas que tienen los estudiantes a la hora de hacer este proceso manualmente, como por ejemplo: cruce de horarios entre materias o cupos llenos, que obligan a los estudiantes a repetir el proceso.

El desarrollo del aplicativo se realizó utilizando las herramientas de software libre PHP y MYSQL, primero por motivos económicos del proyecto y segundo porque son herramientas que conocemos y hemos manejamos durante nuestra formación en la universidad.

El proyecto fue analizado, diseñado y desarrollado en seis meses para optar por el título de tecnología en informática para los dos autores.

1.1 Título del Proyecto

El título “Módulo de gestión para la inscripción de materias en Uniminuto”, se basa en las dos principales características del proyecto, la primera, que es una aplicación orientada a la web y la segunda, que es una herramienta de ayuda para la inscripción de materias en Uniminuto.

1.2 Planteamiento del Problema

Actualmente el proceso que debe hacer un estudiante de Uniminuto en la sede principal para inscribir una materia es muy complejo y toma mucho tiempo, ya que tiene que consultar en distintas páginas del portal web de Uniminuto los horarios disponibles de las asignaturas. El inconveniente principal es que la información se encuentra muy dispersa, es decir, si se desea inscribir las asignaturas de cuarto semestre para el programa de tecnología en informática se debe recurrir a 4 o 5 fuentes de archivos (hojas de cálculo), para seleccionar: las materias pertenecientes al programa, electivas, transversales, idiomas entre otras.

Una vez que el estudiante ha seleccionado las materias acorde a su horario disponible, debe ingresarlas en el módulo de gestión Génesis para formalizar la inscripción, en este punto se pueden presentar los siguientes problemas:

- Cruce de materias en un mismo bloque de horario
- Una o varias materias están sin cupo disponible para inscribir.
- Las asignaturas que no están disponibles en un bloque de hora, obligan al estudiante a repetir el proceso de organizar sus materias a inscribir nuevamente, es decir, volver a consultar los distintos archivos, buscar las materias y organizarlas. Este problema se puede presentar varias veces.

La complejidad del sistema actual se hace evidente, a causa de esto, muchos estudiantes de Uniminuto se ven obligados a pedir ayuda a compañeros, a recurrir

a su respectiva facultad, a inscribir materias en horarios distintos a los deseados e incluso en algunas ocasiones, a perder uno o más créditos.

A raíz de este problema surgen las siguientes preguntas:

¿Cómo organizar adecuadamente los horarios, a la hora de hacer la inscripción de materias en Uniminuto, sin tener que pasar por un tedioso proceso y repetirlo cuando se encuentra un inconveniente?

¿Cómo tener organizada la información referente a todas las materias que necesita un estudiante, a la hora de hacer la inscripción en un mismo lugar y no dispersa en distintos archivos?

¿Cómo evitar el repetir la inscripción parcial o total de asignaturas, cuando un NRC tiene el cupo lleno?

1.3 Alcance y Justificación

El proyecto busca generar una herramienta orientada a la web, que permita a los estudiantes de Uniminuto, realizar el proceso de inscripción de asignaturas más rápido y con menos dificultad que en la actualidad, brindando información acerca de las asignaturas vistas, no vistas y pendientes, ordenando los horarios del semestre según las necesidades de cada estudiante y suministrando filtros para el orden de estos datos.

En general la principal función de la herramienta será realizar los cálculos que permitan mostrar al estudiante, todas las posibles combinaciones de horarios disponibles, tomando en cuenta las asignaturas que el mismo estudiante selecciona, así mismo la herramienta suministrará filtros de datos, para generar un horario más ajustado a sus necesidades.

La herramienta, adicionalmente permitirá llevar un control de las asignaturas pendientes y tendrá la posibilidad de retirar un NRC y reorganizar los horarios, esto último en caso que se encuentre el cupo lleno para dicho código.

Hay que resaltar, que la herramienta no inscribirá las asignaturas, solamente mostrará combinaciones de horarios. No administrará los créditos disponibles de un estudiante, tan solo brindará el número de créditos que tiene cada combinación de horarios. No se conectará con el sistema génesis para saber si un código NRC tiene cupo lleno, únicamente ofrecerá la opción de retirar el NRC y reorganizar los horarios. Todo esto debido al corto tiempo del periodo académico y a la falta acceso al sistema génesis.

1.4 Objetivos

1.4.1 Objetivo General

Desarrollar e implementar un módulo de gestión, que permita a los estudiantes de Uniminuto, hacer más fácil y rápido el proceso de inscripción de asignaturas, mostrando distintas combinaciones de horarios y filtrándolas según sus necesidades. Adicionalmente, permitir llevar un control de las asignaturas vistas, no vistas y pendientes.

1.4.2 Objetivos Específicos

- Sistematizar el proceso de inscripción de asignaturas según los archivos existentes.
- Diseñar los modelos que representen la interacción del estudiante con el sistema para la creación de horarios.
- Optimizar el proceso de selección de asignaturas para la creación de horarios.

2. INGENIERÍA DEL PROYECTO

Para el desarrollo de este proyecto se tendrá en cuenta el diseño, planeación y metodología, los cuales darán las principales pautas en el desarrollo del software, con el objetivo de proporcionar una herramienta que cumpla con los estándares de calidad esperados y con las necesidades de los estudiantes, quienes serán los usuarios finales.

Para lograr este objetivo, el proyecto aplicará el modelo de desarrollo, que será descrito a continuación.

2.1. Metodología y Modelo de Desarrollo

EL modelo de desarrollo que se utilizará para el Módulo de Gestión de Horarios Uniminuto, será U.P. (Proceso Unificado), el cual se caracteriza por estar centrado en la arquitectura, en los casos de uso y además ser iterativo e incremental. Esto permitirá que el proyecto corrija todos los fallos y las posibles imperfecciones en cada iteración, además de ayudar a que la aplicación web, tenga un fácil mantenimiento y una posibilidad alta de evolución. Si bien UP no es solo un proceso, también resulta un marco de trabajo extensible que al ser iterativo e incremental permite la conexión de componentes a través de las interfaces para el desarrollo evolutivo del Software y así mitigar falencias a nivel funcional.

EL Proceso Unificado está compuesto de 4 fases denominadas Inicio, Elaboración, Construcción y Transición, cada fase a su vez contiene distintas iteraciones complementarias, que proporcionan un incremento funcional al Módulo de Gestión de Horarios Uniminuto, al tiempo que ayudan a la corrección de fallos y a un mejor desarrollo de cara al producto final.

El siguiente gráfico ilustra las distintas fases e iteraciones que tendrá el proyecto:

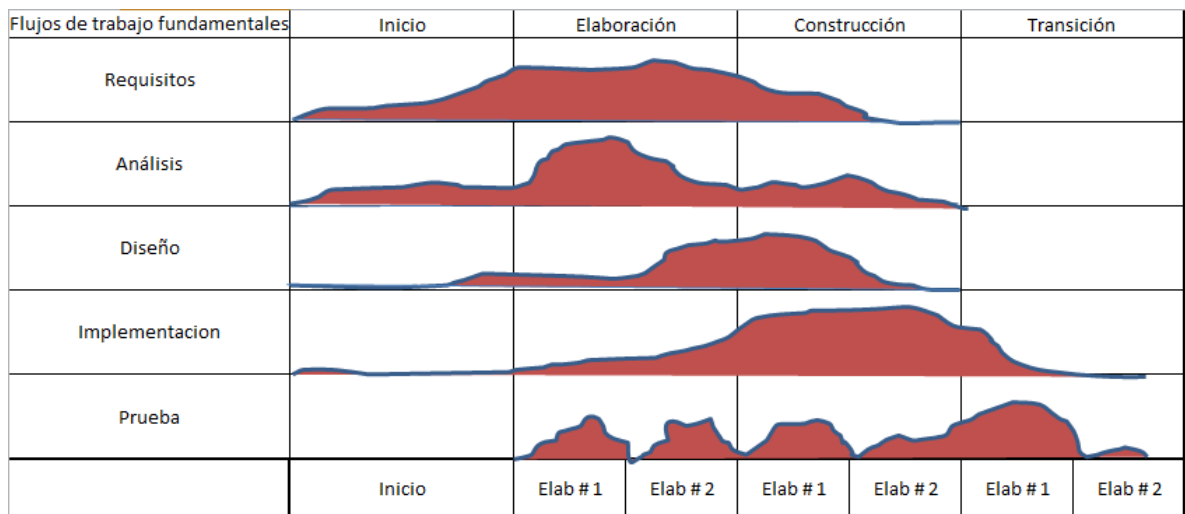


Gráfico N° 1 - Etapas de desarrollo del proyecto

En la fase de Inicio se define la factibilidad y viabilidad del proyecto, se identifican los actores que intervienen en el sistema que son: Usuario Externo y Sistema. Se determina el dominio y alcance reconociendo los deseos del Usuario como los requerimientos funcionales. De igual forma se establece que tipo de software se utiliza para suplir la necesidad del cliente.

En la etapa de Elaboración se hace indispensable identificar los riesgos y determinar la flexibilidad del sistema a nuevos requisitos, de esta forma se desarrollaran los siguientes ítems:

- Se elaboran los diagramas de Casos de Uso (CU) para la descripción de relación entre los actores anteriormente mencionados (Usuario Externo y Sistema).
- Se elaboran los diagramas UML para definir clases, propiedades y funcionalidades del sistema.
- Se estructura el diseño de la interfaz de usuario apropiada para los requisitos del usuario.

- Se estipula el control de acceso a la aplicación por parte del Usuario Externo.

En la etapa de Construcción se lleva a cabo lo siguiente:

- Se implementan las clases con sus funciones que permitirán organizar una selección de horarios.
- Se desarrolla la conexión con los objetos de la Base de Datos para llegar a un nivel de persistencia de datos apropiado para los requisitos del cliente.
- Se determinan los scripts principales que otorguen funcionalidad a todas las clases donde se puedan implementar.
- Se realizan las pruebas Unitarias e Integrales para observar funcionalidad y corregir errores.

Para terminar en la fase de transición se definen los entregables para el Usuario Externo, como los manuales técnico y de usuario, se realizan las pruebas funcionales, no funcionales y de usuarios a todos los módulos para evidenciar la correcta funcionalidad del sistema de desarrollo.

En la fase del diseño del sistema se decidió trabajar con UML. Puesto que UML es un lenguaje visual que permite especificar, construir y verificar sistemas. Independientemente de lenguajes, plataformas e implementaciones. Y por tanto nos ayudara a analizar adecuadamente los requerimientos, a hacer más efectivo su proceso de modelado y de desarrollo.

UML para trabajar la vista estructural nos facilita los diagramas estáticos:

- **Diagrama de clases:**

Este diagrama se utiliza para mostrar y delimitar las clases que modelarán el sistema y como se relacionan entre sí.

- **Diagrama de objetos:**
Estos diagramas se usan para representar, las instancias del diagrama de clases. Un objeto es una entidad la cual posee un estado, comportamiento y pertenece a una clase.
- **Diagrama de Componentes:**
El diagrama de componentes se enfoca hacia los elementos físicos del sistema y como estos se relacionan entre sí, representan a los distintos tipos de elementos que puede contener un sistema de información.
- **Diagrama de implementación:**
Este diagrama muestra el orden de implementación, colocándolo en forma de nodos, los cuales son elementos físicos del sistema que se muestran en tiempo de ejecución.

De los cuales el proyecto utilizará los diagramas de clases y de objetos.

Para la vista funcional UML proporciona los siguientes diagramas:

- **Diagrama de casos de uso:**
Este diagrama muestra los distintos usuarios del sistema, así como sus roles dentro del sistema, basándose en los requerimientos.
- **Diagrama de actividades:**
Muestra un flujo de actividades que se realizarán con sus respectivos resultados. Es útil para determinar las acciones en cada proceso y mostrar los flujos del sistema.
- **Diagrama de secuencia:**
Determina los distintos estados de un objeto en el sistema, durante el ciclo de vida del sistema.
- **Diagrama de estados:**
Este diagrama muestra la interacción del usuario con el sistema, así mismo como la interacción interna del sistema en sus distintos componentes.

De los cuales se tomará el diagrama de casos de uso, el cual nos ayudará a ver de manera gráfica los tipos de usuario, así como sus respectivas funciones en el sistema.

2.2. Patrón de arquitectura

Puesto que el proyecto está enfocado a crear una aplicación orientada a la web, la cual maneja una base de datos, unos procesos internos y una interfaz de usuario, el patrón de arquitectura que se utilizará será el Modelo Vista Controlador (MVC), el cual nos permite separar la lógica de la aplicación, la interfaz gráfica y los datos almacenados, brindando un código ordenado y de fácil mantenimiento, debido a que es muy usado y conocido por los desarrolladores de todo el mundo.

En la siguiente gráfica se ilustra este patrón:

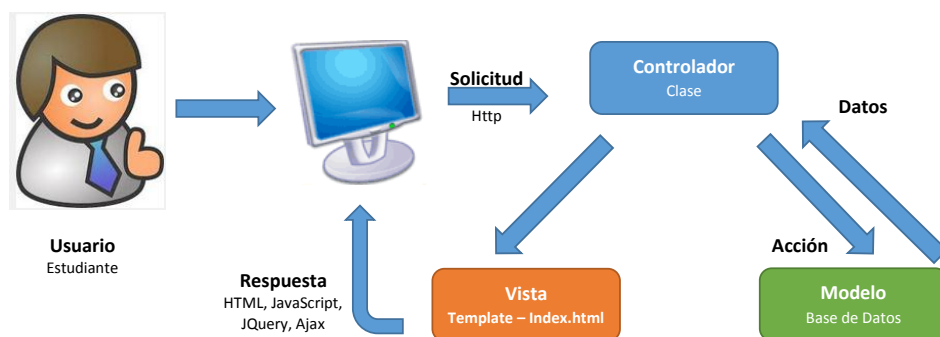


Gráfico N° 2 - Modelo Vista Controlador

Los componentes del gráfico 2 – Modelo Vista controlador, se detallan a continuación:

- **Navegador:** Es la herramienta sobre la cual funcionará el aplicativo.
- **Controlador:** Es el encargado de recibir las peticiones del usuario y llamar al Modelo para ejecutarlas.
- **Vista:** Se encarga de la interfaz gráfica, mostrar los horarios y el pensum, estas órdenes las recibe desde el Controlador.
- **Modelo:** Es la parte encargada de obtener los datos y procesarlos, retornando la información de los distintos horarios a mostrar.
- **Datos:** Base de datos donde se almacenará la información concerniente a los alumnos y sus asignaturas.

3. ANÁLISIS Y DISEÑO

Teniendo en cuenta las necesidades y principales problemas a los que el proyecto pretende dar solución, se aplica una fase de análisis del sistema, en la cual se hace un levantamiento de información con el fin de recolectar los principales requerimientos para el sistema propuesto. Con estos requerimientos ya definidos, se procede a la elaboración de un diseño acorde a las necesidades que el proyecto requiere.

3.1. Definición de Requerimientos

Luego del levantamiento de información, acerca de las necesidades que existen para hacer más fácil la inscripción de materias en Uniminuto, se establecieron los siguientes requerimientos que la aplicación debe suplir.

3.1.1. Requerimientos funcionales

Para definir un alcance y determinar un dominio de la funcionalidad del software se establecen los siguientes requerimientos en este proyecto:

1. Permitir que los usuarios ingresen con un ID y Contraseña.
2. Permitir que los usuarios creen una cuenta.
3. Permitir cargar los documentos de Excel con las asignaturas disponibles de las distintas facultades
4. Hacer pregunta de seguridad si el usuario olvida la clave.
5. Permitir ver estadísticas del sitio
6. Permitir eliminar o agregar materias al sitio.
7. Mostrar el pensum del alumno según su carrera.
8. Diferenciar entre usuario Estudiante y usuario Administrador.

9. Permitir seleccionar materias vistas y no vistas en el pensum para tener un registro.
10. Permitir seleccionar las materias a inscribir por el usuario Estudiante.
11. Mostrar todas las posibles combinaciones de horarios según las materias y espacios temporales escogidos el alumno.
12. Permitir retirar un NRC y reordenar las combinaciones de horarios sin este código.
13. Guardar en una base de datos la información relacionada con los usuarios, asignaturas, pensum y carreras de la universidad.
14. Mostrar el número de créditos de cada horario y materia.
15. Permitir cerrar sesión.
16. Realizar consultas de número de usuarios, nombres, roles, información personal.
17. Permitir la actualización de la información personal de los usuarios.

3.1.2. Requerimientos No funcionales

1. La aplicación debe estar montada en un servidor
2. El usuario debe tener una conexión a internet para usar la herramienta.
3. El sistema debe mantener la confidencialidad de las cuentas de la información personal de los usuarios.
4. El sistema debe tener tiempos de respuesta adecuados, teniendo en cuenta las prestaciones del servidor.
5. El sistema debe proporcionar información verídica acerca de las materias y horarios.
6. El sistema debe ser escalable y permitir el desarrollo de nuevas funcionalidades, afectando el código de la menor manera posible.
7. El sistema debe ser de fácil uso para el usuario final con menús y opciones claras y fácilmente localizables.

8. El sistema debe dar acceso solo a personas registradas y validar las claves teniendo en cuenta el rol del usuario.
9. El sistema debe proporcionar confidencialidad en cuanto a la información de los usuarios se refiere.
10. El sistema debe tener en cuenta al validar los formularios de creación y acceso, la obligatoriedad de los campos, la longitud de caracteres y el tipo de datos que son ingresados.
11. El sistema debe funcionar correctamente en los navegadores Internet Explorer v.7 o superior, Mozilla o Google Chrome.
12. El sistema debe garantizar que el diseño de consultas no afecta el rendimiento de la base de datos.
13. Contar con un reconocimiento de roles de usuarios y re direccionamiento a sus respectivos módulos, rápido e intuitivo.
14. El sistema debe haber terminado satisfactoriamente la fase de pruebas.

3.2. Descripción del Sistema Propuesto

- Módulo de Gestión de Horarios Uniminuto, es una aplicación orientada a la web, que permite a los estudiantes de Uniminuto, realizar un proceso de inscripción de materias mucho más amable que el usado actualmente. Permite de manera gráfica visualizar las posibles combinaciones de horarios que se puede seleccionar, según filtros diseñados para facilitar todo el proceso y hacerlo acorde a las necesidades de cada estudiante. La aplicación consta de cuatro módulos que se describen a continuación:
- Módulo Inicio, el cual permite dos acciones, una la creación de cuentas de usuario, diligenciando un formulario muy sencillo con nombre, contraseña, programa, jornada y semestre, la otra el ingreso al sistema ya sea usuario Estudiante o usuario Administrador.
- Módulo Pensum, en esta área la aplicación muestra al usuario Estudiante todas las materias que debe ver durante su carrera, agrupadas en tablas

por semestre y cada una con una casilla de chequeo para seleccionar las ya vistas, las cuales se mostrarán de un color distinto. El sistema guardará esta información para que solo haya que chequear una vez la lista.

- Módulo Materias, donde se mostrarán las materias pendientes por inscribir al usuario Estudiante, con su respectiva casilla de activación, además se podrá especificar un máximo de créditos permitido.
- Módulo Horarios, en el cual se mostrarán en forma de tablas con horarios todas las posibles combinaciones de horarios que el sistema propone, con los datos suministrados en el módulo Materias. Además este módulo presentará dos filtros: el primero para retirar un NRC, el cual esta formado por un campo de texto para escribir el código y un botón que reorganizará los horarios sin dicho código, y el segundo filtro será una tabla dinámica con formato de horario en la cual se puede seleccionar o deseleccionar espacios de horario, con su respectivo botón que organiza nuevamente los horarios.
- Módulo Administrador, este módulo permite a los usuarios con rol Administrador ingresar los archivos Excel de las distintas materias, eliminar materias del sistema, agregar nuevas materias, renombrar materias y ver estadísticas de la aplicación.

3.2.1. Diagrama Entidad Relación

El diagrama entidad relación representa una visión de los datos del sistema propuesto de forma global, representado por medio de entidades, las cuales son representaciones de objetos del mundo real con sus correspondientes atributos y las relaciones que se establecen entre las distintas entidades que componen el sistema de datos.

A continuación se muestra el diagrama de la base de datos, sus respectivas relaciones y la definición de las entidades que lo componen.

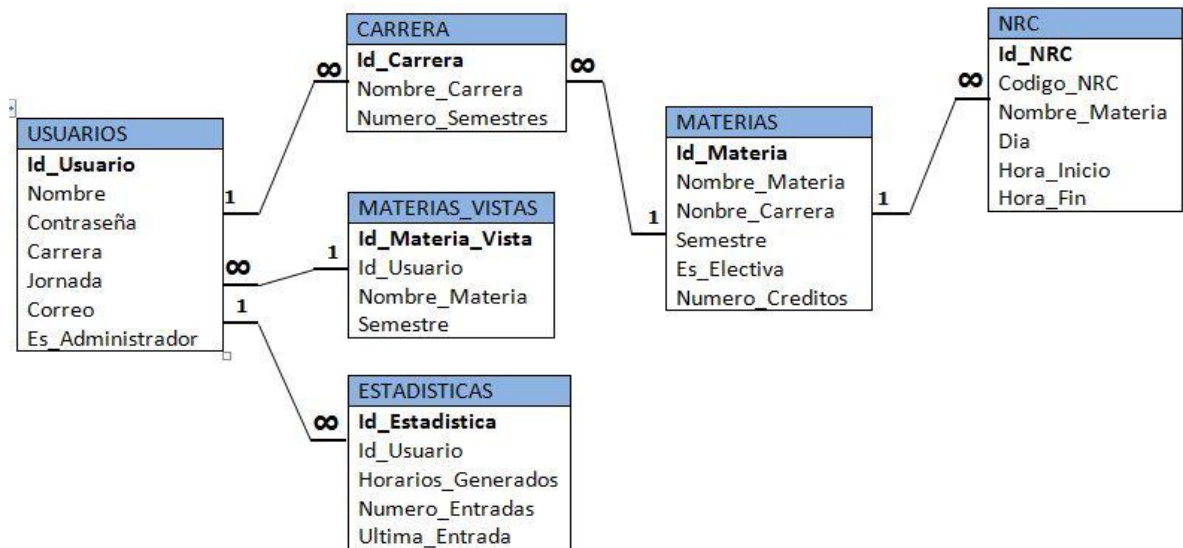


Gráfico N° 3 - Diagrama Entidad Relación

3.2.2. Diccionario de Datos

Las siguientes tablas muestran la descripción de las entidades, sus atributos y sus principales características

Nombre Entidad: USUARIOS		Código: 01	
Campo	Tipo	Tamaño	Descripción
Id_Usuario	Numérico	Entero largo	Llave primaria de la entidad USUARIOS
Nombre	Texto	50	Almacena el nombre de los usuarios.
Contraseña	Texto	20	Contraseña de los Usuarios.
Nombre_Carrera	Texto	30	Carrera que está cursando el Usuario
Jornada	Texto	10	Jornada en la cual estudia el Usuario
Correo	Texto	50	Correo del Usuario
Es_Administrador	Texto	8	Si es o no administrador
Tablas Relacionadas: CARRERA, MATERIAS_VISTAS		Campos Clave: Id_Usuario	

Tabla N° 1 - Entidad USUARIOS

Nombre Entidad: CARRERA		Código: 02	
Campo	Tipo	Tamaño	Descripción
Id_Carrera	Auto numérico	Entero largo	Llave primaria de la entidad CARRERA
Nombre_Carrera	Texto	30	Llave foránea USUARIOS.
Numero_Semestres	Numérico	20	Número de semestres que tiene la carrera.
Tablas Relacionadas: USUARIOS, MATERIAS.		Campos Clave: Id_Carrera, Nombre_Carrera.	

Tabla N° 2 - Entidad CARRERA

Nombre Entidad: MATERIAS_VISTAS		Código: 03	
Campo	Tipo	Tamaño	Descripción
Id_Materia_Vista	Auto numérico	Entero largo	Llave primaria de la entidad MATERIAS_VISTAS.
Id_Usuario	Numérico	Entero largo	Llave foránea USUARIOS.
Nombre_Materia	Texto	50	Número de semestres que tiene la carrera.
Semestre	Numérico	Entero largo	Semestre al que pertenece la materia
Tablas Relacionadas: USUARIOS.		Campos Clave: Id_Materia_Vista, Id_Usuario.	

Tabla N° 3 - Entidad MATERIAS_VISTAS

Nombre Entidad: MATERIAS		Código: 04	
Campo	Tipo	Tamaño	Descripción
Id_Materia	Auto numérico	Entero largo	Llave primaria de la entidad MATERIAS
Nombre_Materia	Texto	50	Almacena el nombre de La materia.
Nombre_Carrera	Texto	30	Llave foránea CARRERA.
Semestre	Numérico	Entero largo	Numero de semestre al cual pertenece la materia
Es_Electiva	Texto	8	Si es o no una materia electiva
Numero_Creditos	Numérico	Entero largo	Número de créditos de la materia
Tablas Relacionadas: CARRERA, NRC		Campos Clave: Id_Materia, Nombre_Carrera	

Tabla N° 4 - Entidad MATERIAS

Nombre Entidad: NRC		Código: 05	
Campo	Tipo	Tamaño	Descripción
Id_NRC	Auto numérico	Entero largo	Llave primaria de la entidad NRC
Codigo_NRC	Numérico	Entero largo	Almacena el nombre de los usuarios.
Nombre_Materia	Texto	50	Llave foránea MATERIAS.
Dia	Texto	30	Día de la semana que se dicta.
Hora_Inicio	Numérico	Entero largo	Hora de inicio de la clase
Hora_Fin	Numérico	Entero largo	Hora de cierre de la clase.
Tablas Relacionadas: MATERIAS		Campos Clave: Id_NRC, Nombre_Materia	

Tabla N° 5 - Entidad NRC

Nombre Entidad: ESTADISTICAS		Código: 06	
Campo	Tipo	Tamaño	Descripción
Id_Estadistica	Auto numérico	Entero largo	Llave primaria de la entidad ESTADISTICAS
Id_Usuario	Numérico	Entero largo	Llave foránea USUARIOS.
Horarios_Generados	Numérico	Entero largo	Número de horarios que se han generado.
Numero_Entradas	Numérico	Entero largo	Número de entradas que se han generado.
Ultima_Entrada	Fecha/Hora	8	Registro fecha y hora de la última entrada.
Tablas Relacionadas: USUARIOS		Campos Clave: Id_Estadistica, Id_Usuario	

Tabla N° 6 - Entidad ETADISTICAS

3.2.3. Actores y Componentes del Sistema Propuesto

Los actores del sistema son Administrador y Usuario Externo (Estudiante), a continuación se describen los permisos y funciones dentro del módulo de gestión

Usuario 1: Administrador

El administrador tiene acceso total a todas las pantallas del módulo de gestión y puede realizar las siguientes funciones.

- Ingresar al sistema con usuario y contraseña
- Ingresar al menú principal de administrador
- Cargar fuentes de datos de horarios (archivos por lotes .csv)
- Consultar estadísticas del sitio
- Editar un programa académico y sus respectivas materias

Usuario 2: Estudiante

El estudiante tiene acceso a las siguientes funciones:

- Ingresar al sistema con usuario y contraseña
- Ingresar al módulo de Pensum(Materias vistas)
- Ingresar a Materias (Materias a inscribir)
- Ingresar a horarios

El sistema está dividido en 4 módulos principales:

Módulo Iniciar / Crear Sesión:

Formularios:

- Crear Usuario:
 - ID
 - Nombre
 - Contraseña
 - Correo

- Carrera
 - Jornada
 - Botón Enviar
- Ingresar:
 - ID
 - Contraseña
 - Botón Enviar
 - Olvidó su contraseña

Módulo Pensum (Materias Vistas)

Formularios:

- Materias que ya aprobaste:
 - Lista de Checkbox de asignaturas
 - Botón enviar

Módulo Materias (Materias Vistas)

Formularios:

- Materias a inscribir:
 - Lista de Checkbox de asignaturas
 - Botón enviar

Módulo Horarios

- Filtro de bloques de hora
- Consultas de horarios disponibles
- Botón Atrás

3.2.4. Propuesta de Interfaz Gráfica

El software está dividido en diferentes módulos para brindar una interfaz organizada y de fácil manejo al usuario. Cada módulo permitirá al usuario (Estudiante para este software) iniciar/crear sesión, seleccionar las materias vistas, seleccionar materias a cursar y visualizar horario.

El diseño presentado a continuación fue desarrollado con PHP como lenguaje de programación, html5 para el diseño de las interfaces gráficas y MySQL como sistema de almacenamiento de datos.

El módulo de gestión de horarios está compuesto por:

- Formulario Inicio/creación de sesión: Existen 2 formularios, el primero es de creación de usuario para los estudiantes que ingresan por primera vez al módulo de gestión, donde ingresarán información básica como ID, Nombre, Contraseña entre otros datos para crear el usuario. El segundo formulario es de inicio de sesión para los estudiantes que ya están creados en el sistema, de esta forma se validará el ingreso e informará al usuario si no existe en la base de datos. Además se mostrarán en la parte inferior las distintas opciones de la aplicación.

The image shows a screenshot of the 'GESTOR DE HORARIOS UNIMINUTO' web application. At the top, there is a blue header with the UNIMINUTO logo on the left and the text 'GESTOR DE HORARIOS UNIMINUTO' in the center. Below the header, the page is divided into two main sections: 'Crear usuario' on the left and 'Ingresar' on the right. The 'Crear usuario' form includes input fields for ID, Nombre, Contraseña, Correo, Carrera (a dropdown menu), and Jornada (a dropdown menu), with an 'Enviar' button at the bottom. The 'Ingresar' form includes input fields for ID and Contraseña, an 'Enviar' button, and a blue link labeled 'Olvido su clave'.

Gráfico N° 4 - Iniciar / Crear Sesión

Si el usuario no se encuentra registrado en el sistema se mostrará la alerta en el navegador.

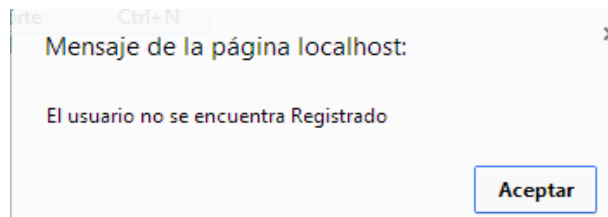


Gráfico N° 5 - Usuario no registrado

- Pensum (Materias Vistas): En esta parte el estudiante deberá marcar las materias que ya ha visto de su carrera. Para ello selecciona las casillas correspondientes de cada asignatura que se encuentran agrupadas por semestre.

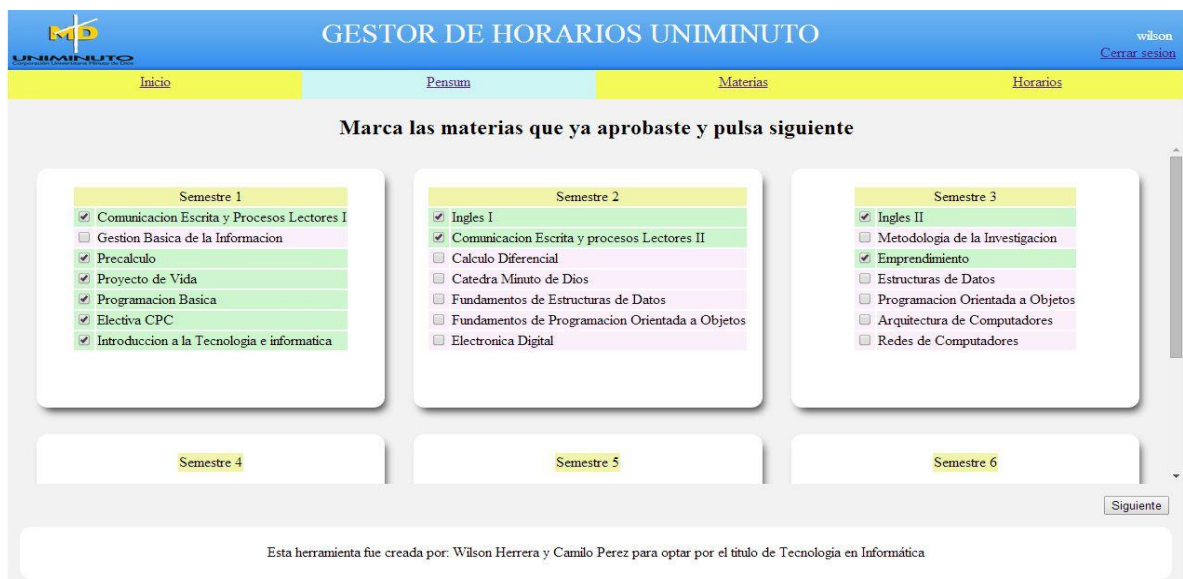


Gráfico N° 6 - Pensum (Materias Vistas)

- Materias (Materias a cursar): En esta sección el estudiante deberá seleccionar las materias que va a cursar para generar los respectivos

horarios. Para ello basta con marcar las casillas pertenecientes a cada asignatura que se encuentran agrupadas por semestre.

GESTOR DE HORARIOS UNIMINUTO

Inicio Pensum Materias Horarios

Wilson Cerrar sesion

Selecciona las materias que deseas inscribir y pulsa siguiente

Semestre 1

- Gestion Basica de la Informacion

Semestre 2

- Calculo Diferencial
- Catedra Minuto de Dios
- Fundamentos de Estructuras de Datos
- Fundamentos de Programacion Orientada a Objetos
- Electronica Digital

Semestre 3

- Metodologia de la Investigacion
- Estructuras de Datos
- Programacion Orientada a Objetos
- Arquitectura de Computadores
- Redes de Computadores

Siguiendo

Esta herramienta fue creada por: Wilson Herrera y Camilo Perez para optar por el titulo de Tecnologia en Informatica

Gráfico N° 7 - Materias a inscribir

- **Horarios:** En esta sección se podrá visualizar los horarios disponibles para escoger según las materias seleccionadas en la anterior sección. En la parte izquierda se mostrará un horario donde el usuario podrá marcar los espacios de hora disponibles para las materias que seleccionó, teniendo en cuenta que entre más espacios de hora seleccione, mayor será la probabilidad de generar más horarios con las asignaturas seleccionadas.

GESTOR DE HORARIOS UNIMINUTO wilson
Cerrar sesión

Inicio Pensum Materias Horarios

Horarios que puedes tomar

Espacios de hora para agregar o quitar

Horarífin	Lun	Mart	Mier	Juev	Vier	Sab
700-745						
745-830						
830-915						
915-1000						
1000-1045						
1045-1130						
1130-1215						
1215-1300						
1300-1345						
1345-1430						
1430-1515						
1515-1600						
1600-1645						

Retirar NRC:

Horario 1						
Hora	Lunes	Martes	Miercoles	Jueves	Viernes	Sabado
830-915						Redes de Computadores 2121
915-1000						
1045-1130						Redes de Computadores 2121
1130-1215						
1815-1900		Catedra Minuto de Dios 1111				
1900-1945						
2030-2115					Precalculo 333	
2115-2200						

Esta herramienta fue creada por: Wilson Herrera y Camilo Perez para optar por el título de Tecnología en Informática

Gráfico N° 8 - Horarios


- **Módulo Administrador:** Este módulo solo estará visible para los usuarios de tipo administrador que estén registrados en el sistema, allí podrá realizar tareas especiales como: Cargar fuentes de datos de horarios (archivos por lotes .csv), Consultar estadísticas del sitio, Editar un programa académico y sus respectivas materias

GESTOR DE HORARIOS UNIMINUTO Fabian
Cerrar sesión

Módulo administrador para tareas varias

Cargar excel

Seleccionar archivo | Ningún archivo seleccionado



Estadísticas del sitio

Número de usuarios activos:
324

Número de entradas totales:
5012

Número de horarios mostrados:
15003

Editar Programa academico

Tec. informatica

Materias

Redes

Precalculo

Geometria

Fisica electrica

Proyecto de vida

Catedra Minuto

Esta herramienta fue creada por: Wilson Herrera y Camilo Perez para optar por el título de Tecnología en Informática

Gráfico N° 9 - Módulo Administrador

3.3. Diseño del Sistema Propuesto

El diseño del sistema se va a fundamentar en dos vistas una estructural y una funcional. La herramienta que se va a trabajar es UML, la cual nos brindará los diagramas necesarios para fundamentar el modelo de desarrollo de acuerdo con las necesidades que el proyecto requiere.

3.3.1. Diagramas Estáticos

Los diagramas Estáticos se encargan de definir los elementos que componen el sistema y su estructura, estos elementos pueden ser clases, objetos, componentes entre otros.

3.3.2. Diagrama de Clases

El diagrama de clases que se muestra a continuación describe las distintas clases que posee el sistema con sus atributos métodos y relaciones.

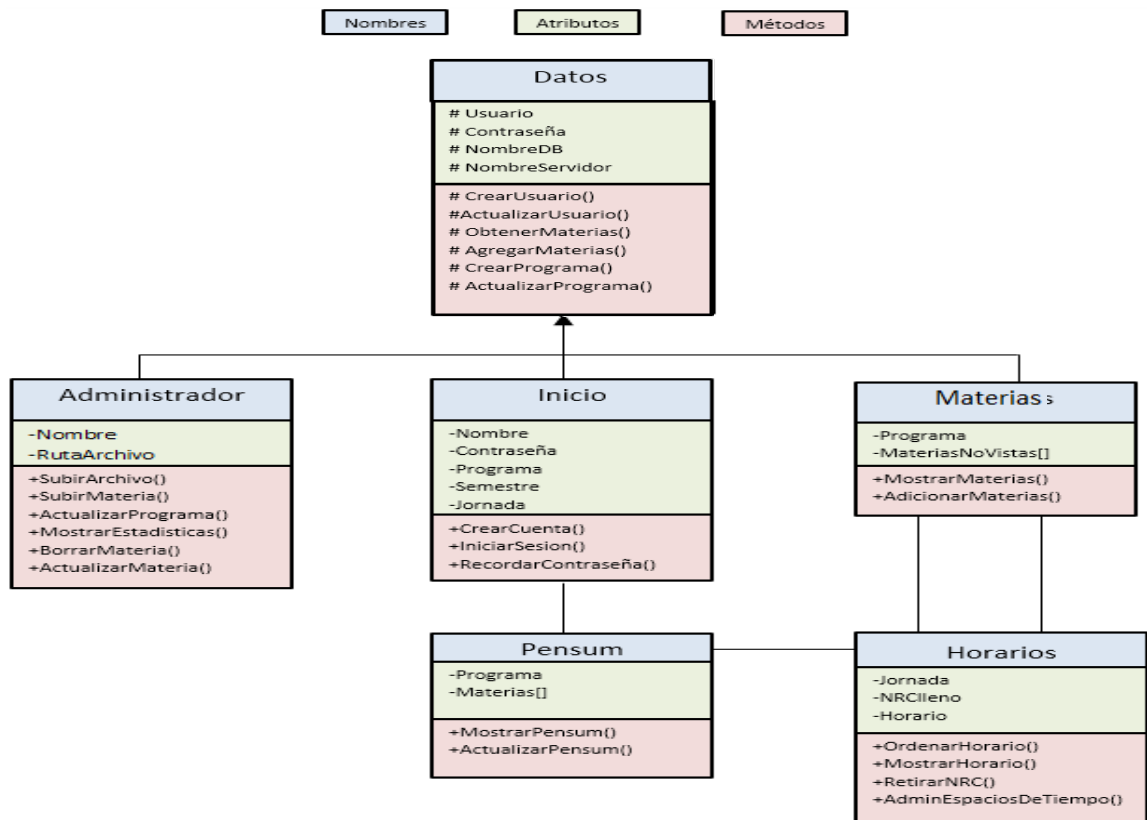


Gráfico N° 10 - Diagrama de Clases

Clase	Atributos	Métodos
Inicio	Nombre ->para guardar usuario, tipo string Contraseña ->para guardar usuario, tipo string Programa ->para guardar usuario, tipo string Semestre ->para guardar usuario, tipo entero Jornada -> mañana tarde o noche, tipo char(M,T,N)	CrearCuenta() ->captura los datos del usuario y crea una cuenta, si no existe ya el usuario IniciarSesion() ->valida que clase de usuario es e inicia la sesión respectiva RecordarContraseña() ->hace una pregunta de seguridad para restablecer contraseña
Pensum	Programa ->al que pertenece el estudiante tipo string Materias[] ->tipo array, con las	MostrarPensum() ->cálculos necesarios para llenar las tablas del pensum según corresponda

	materias del programa del estudiante	ActualizarPensum() ->guarda las materias ya vistas por el estudiante y actualiza las tablas con colores distintos para las ya vistas
Horarios	<p>Jornada->mañana tarde o noche, tipo char(M,T,N)</p> <p>NRClleno->para deshabilitar NRC</p> <p>Horario->con las distintas materias de los horarios, tipo table</p>	<p>OrdenarHorario()->ordena los horarios según las materias y filtros dados</p> <p>MostrarHorario()->organiza los horarios en tablas de tipo horario</p> <p>RetirarNRC()->retira NRC y reordena las tablas</p> <p>AdminEspaciosDeTiempo()->reordena las tablas de horarios según los espacios de horario seleccionados</p>
Materias	<p>Programa-> al que pertenece el estudiante tipo string</p> <p>MateriasNoVistas[]->tipo array, con las materias que desea ver el estudiante</p>	<p>MostrarMaterias()->muestra las materias pendientes del estudiante</p> <p>AdicionarMaterias()->crea un array con las materias seleccionadas y se lo pasa al módulo horarios</p>
Administradora	<p>Nombre->nombre del administrador, tipo string</p> <p>RutaArchivo->ruta de donde se va subir el archivo con las materias</p>	<p>SubirArchivo()->monta y procesa un archivo en el sistema</p> <p>SubirMateria()->sube una materia a la base de datos</p> <p>ActualizarPrograma()->actualiza un programa y sus materias asociadas en la base de datos</p> <p>MostrarEstadisticas()->hace una consulta en la base de datos de los valores requeridos</p> <p>BorrarMateria()->borra una materia de la base de datos</p> <p>ActualizarMateria()->actualiza materia en la base de datos</p>
Datos	Usuario ->usuario de la base de datos	CrearUsuario() ->crea un usuario en la base de datos

	<p>Contraseña->contraseña de la base de datos</p> <p>NombreDB->nombre de la base de datos</p> <p>NombreServidor->nombre del servidor que aloja la base de datos</p>	<p>ActualizarUsuario()->actualiza un usuario en la base de datos</p> <p>ObtenerMaterias()->consulta materias según parámetros en la base de datos</p> <p>AgregarMaterias()->crea materias en la base de datos</p> <p>CrearPrograma()->crea un nuevo programa y asocia materias en la base de datos</p> <p>ActualizarPrograma()-> actualiza un programa en la base de datos</p>
--	---	--

Tabla N° 7 - Descripción de las Clases

3.3.3. Diagramas de Objetos

El siguiente diagrama de objetos muestra los distintos objetos que componen el sistema, así como las relaciones entre ellos.

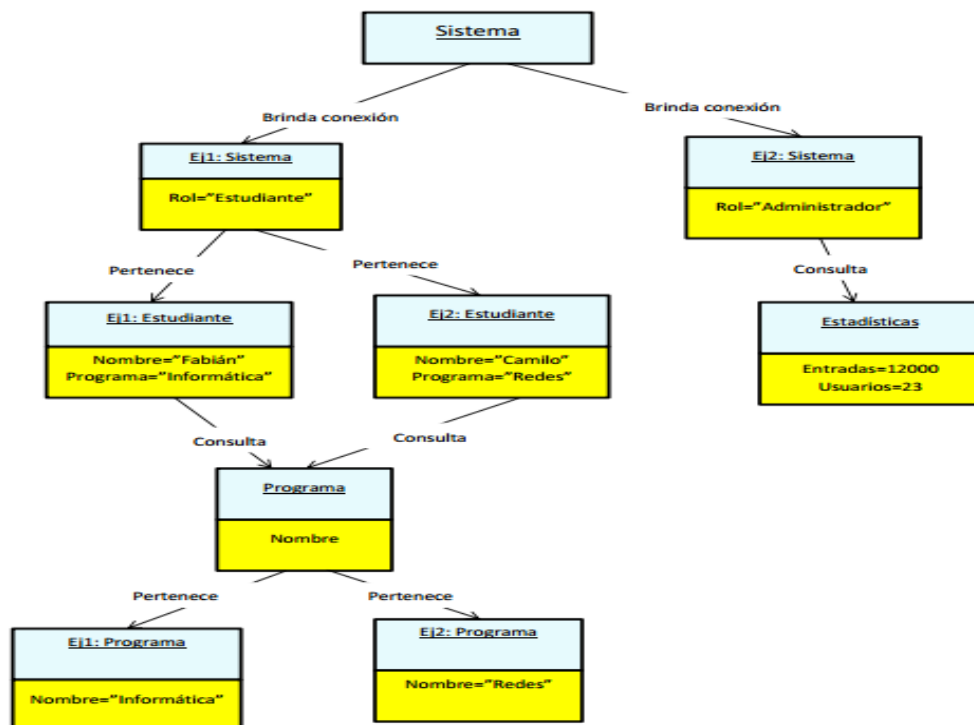


Gráfico N° 11 - Modelo de Objetos

Definición de los objetos:

Sistema: Contiene el tipo de usuario en el sistema, sus permisos y los módulos a los que tiene acceso.

Programa: contiene el nombre del programa académico y sus materias asociadas.

Estadísticas: Solo para los administradores, contiene el número de usuarios que tiene el sitio, el número de entradas y otras estadísticas.

3.3.4. Diagramas Dinámicos

Los diagramas dinámicos son representaciones del sistema en tiempo de ejecución, muestran las interrelaciones y los cambios de los objetos a lo largo del tiempo.

3.3.5. Diagramas de casos de uso

En el diagrama de casos de uso se muestran los dos roles que tiene el sistema, usuario estudiante y administrador, con sus respectivas acciones en la aplicación.

Caso de uso 1: Creación de usuario

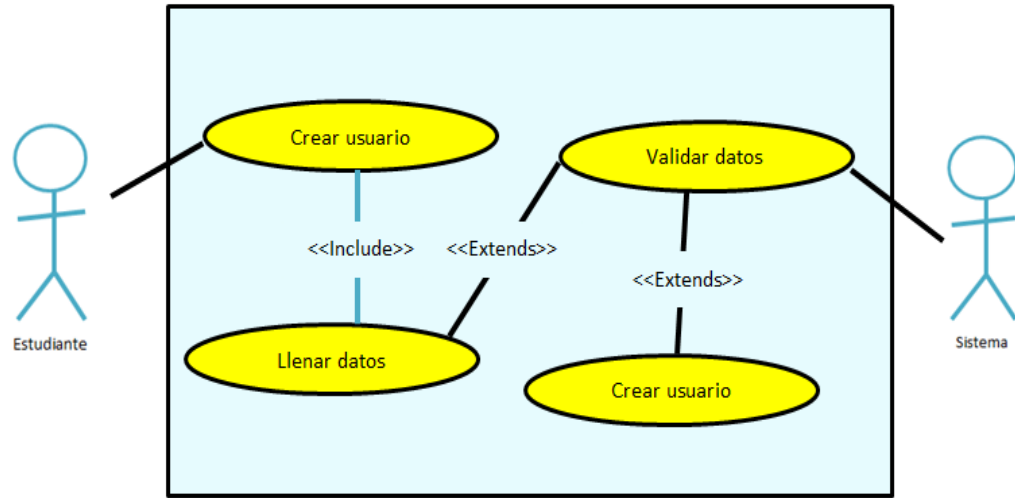


Gráfico N° 12 - Caso de Uso 1

Nombre:	Creación de usuario / CU-1	
Actor:	Estudiante	
Descripción:	Proceso que debe realizar un estudiante para crear una cuenta a su nombre.	
Flujo Principal:	Eventos ACTOR	Eventos SISTEMA
	1. Diligenciar formulario con: nombre, contraseña, programa académico, y semestre.	1. Muestra formulario y botón, muestra datos obligatorios y en el campo de contraseña los caracteres son ocultos.
	2. Clic en el botón crear.	2. Valida si el usuario no existe en la base de datos.
		3. crea al usuario en la base de datos con rol estudiante.
Alternativa:	1. Diligenciar formulario con: nombre, contraseña, programa académico, y semestre.	1. Muestra formulario y botón, muestra datos obligatorios y en el campo de contraseña los caracteres son ocultos.
	2. Clic en el botón crear.	2. Valida si el usuario existe en la base de datos.
		3. Muestra un mensaje indicando que el nombre de usuario ya existe.
Precondición:	El estudiante debe ingresar sus datos y no estar previamente inscrito.	
Poscondición:	El estudiante crea exitosamente su cuenta de usuario.	
Presunción:	La base de datos de registro al usuario con todos sus datos.	

Tabla N° 8 - Caso de Uso 1

Caso de uso 2: Ingresar al sistema

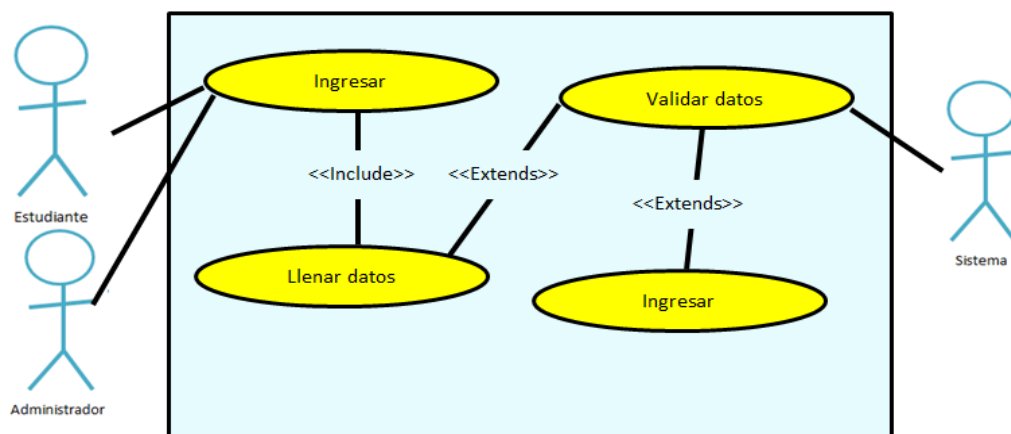


Gráfico N° 13 - Caso de Uso 2

Nombre:	Ingresar al sistema / CU-2	
Actor:	Estudiante/Administrador	
Descripción:	Proceso que debe realizar un estudiante o un administrador para ingresar al sistema.	
Flujo Principal:	Eventos ACTOR	Eventos SISTEMA
	1. Diligenciar formulario con: nombre, contraseña.	1. Muestra formulario y botón.
	2. Clic en el botón entrar.	2. Valida si el usuario existe en la base de datos y su rol en el sistema.
		3. Crea una sesión e ingresa a su respectivo módulo tomando en cuenta el rol del usuario.
Alternativa:	1. Diligenciar formulario con: nombre, contraseña.	1. Muestra formulario y botón.
	2. Clic en el botón entrar.	2. Valida si el usuario no existe en la base de datos.
		3. Si no existe en la base de datos muestra mensaje "debe crear usuario" y si existe y la contraseña es errónea "la contraseña es errónea".
Precondición:	El estudiante o administrador deben estar previamente inscritos.	
Poscondición:	Se inicia sesión y se muestra el módulo correspondiente al rol del usuario.	
Presunción:	El sistema crea una sesión con los datos del usuario.	

Tabla N° 9 - Caso de Uso 2

Caso de uso 3: Llenar pensum con las materias ya vistas

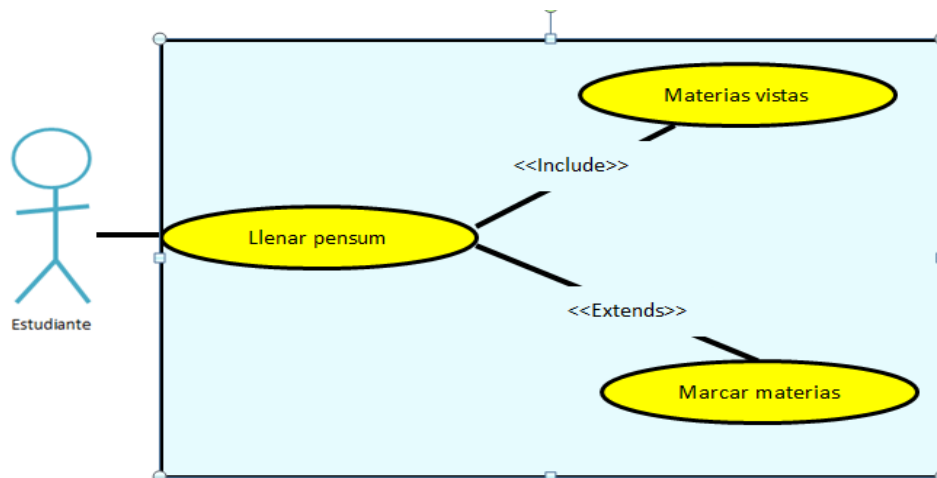


Gráfico N° 14 - Caso de Uso 3

Nombre:	Llenar pensum con las materias ya vistas / CU-3	
Actor:	Estudiante	
Descripción:	Proceso que debe realizar un estudiante para actualizar su pensum con las materias ya vistas y aprobadas.	
Flujo Principal:	Eventos ACTOR	Eventos SISTEMA
	1. marcar en las tablas de semestres las materias ya vistas y aprobadas.	1. Muestra tablas de los semestres según el programa y sus materias correspondientes, cada una con una casilla de verificación.
	2. Clic en el botón enviar.	2. Valida cuales materias marco el usuario.
		3. muestra en distinto color las materias ya vistas por el estudiante.
Precondición:	El estudiante debe tener un programa académico asociado.	
Poscondición:	El estudiante visualiza su pensum actualizado.	
Presunción:	El sistema actualiza en la base de datos las materias ya vistas por el estudiante.	

Tabla N° 10 - Caso de Uso 3

Caso de uso 4: Seleccionar materias a inscribir

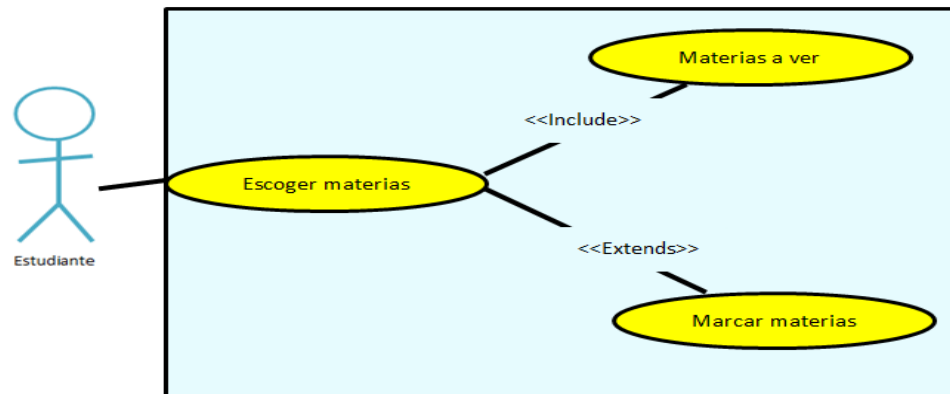


Gráfico N° 15 - Caso de Uso 4

Nombre:	Seleccionar materias a inscribir / CU-4	
Actor:	Estudiante	
Descripción:	Proceso que debe realizar un estudiante para seleccionar las materias que desea inscribir.	
Flujo Principal:	Eventos ACTOR	Eventos SISTEMA
	1. marcar en las tablas de semestres las materias que desea inscribir.	1. Muestra tablas de los semestres según el programa y las materias pendientes, cada una con una casilla de verificación.
	2. Clic en el botón enviar.	2. Valida cuales materias marco el usuario.
		3. Muestra por medio de tablas las posibles combinaciones de horarios con las materias seleccionadas.
Alternativa:	1. No marcar en las tablas de semestres las materias ya vistas y aprobadas.	1. Muestra tablas de los semestres según el programa y sus materias correspondientes, cada una con una casilla de verificación.
	2. Clic en el botón enviar.	2. Valida cuales materias marco el usuario.
		3. Muestra un mensaje indicando que se deben marcar las materias.
Precondición:	El estudiante debe marcar las materias que desea inscribir.	
Poscondición:	Se muestran las combinaciones de horarios.	
Presunción:	El sistema procesa los datos para mostrar las combinaciones de horarios sin cruces de horarios ni repetición de materias.	

Tabla N° 11 - Caso de Uso 4

Caso de uso 5: Mostrar combinaciones de horarios

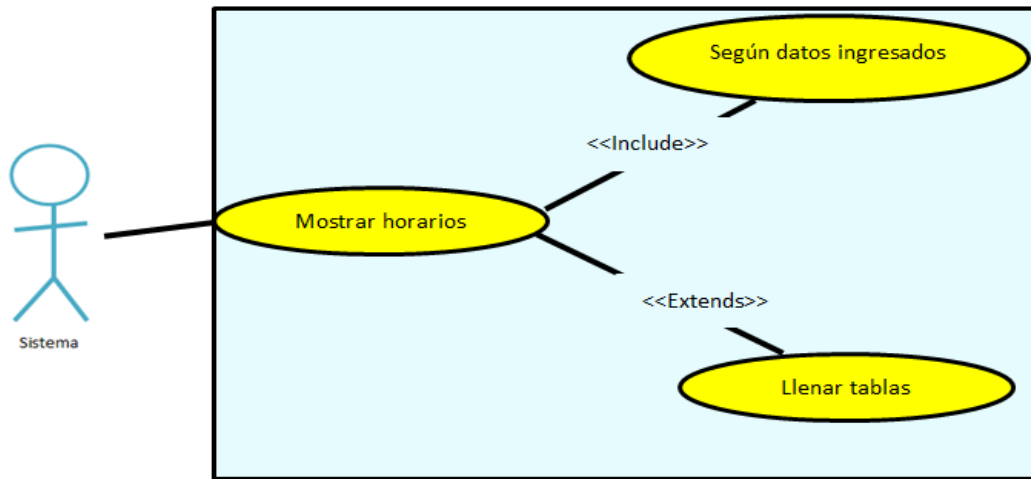


Gráfico N° 16 - Caso de Uso 5

Nombre:	Mostrar combinaciones de horarios / CU-5	
Actor:	Sistema	
Descripción:	Proceso que debe realizar un estudiante para actualizar su pensum con las materias ya vistas y aprobadas.	
Flujo Principal:	Eventos ACTOR	Eventos SISTEMA
	1. Recibe datos de las materias que se deben mostrar.	1. Se hacen los cálculos respectivos con las materias según hora y NRC y nombre de materia. 2. muestra las combinaciones de horarios en tablas.
Precondición:	Se debe haber seleccionado las materias a inscribir.	
Poscondición:	Se muestran las combinaciones de horarios.	
Presunción:	El sistema muestra mediante tablas las combinaciones de horarios sin cruce de NRC u hora.	

Tabla N° 12 - Caso de Uso 5

Caso de uso 6: Filtrar horarios por espacios de horas específicos

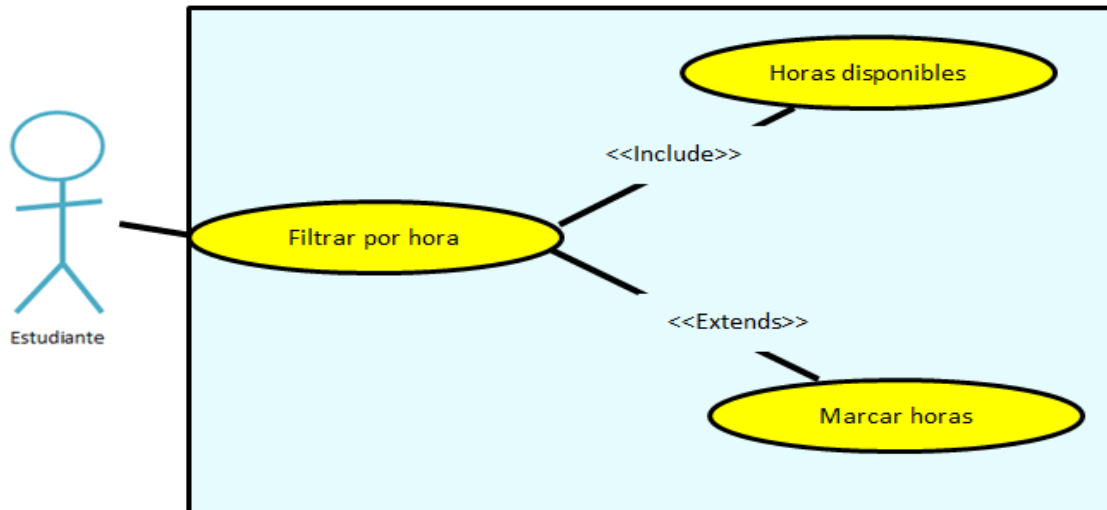


Gráfico N° 17 - Caso de Uso 6

Nombre:	Filtrar horarios por espacios de horas específicos / CU-6	
Actor:	Estudiante	
Descripción:	Proceso que debe realizar un estudiante para quitar o agregar espacios de tiempo específicos, en la configuración de horarios.	
Flujo Principal:	Eventos ACTOR	Eventos SISTEMA
	1. marcar en la tabla de horarios disponibles, los espacios de tiempo que desea agregar o quitar.	1. Muestra la tabla de horarios disponibles.
	2. Clic en el botón enviar.	2. Valida espacios de hora habilitados por el usuario.
		3. Reorganiza las tablas de combinaciones de horarios según la configuración.
Precondición:	El estudiante debe seleccionar los espacios de hora según sus necesidades.	
Poscondición:	El estudiante visualiza las combinaciones de horarios actualizadas.	
Presunción:	El sistema actualiza las tablas de combinaciones sin los espacios de tiempo deshabilitados.	

Tabla N° 13 - Caso de Uso 6

Caso de uso 7: Llenar tablas con el pensum de la materia correspondiente

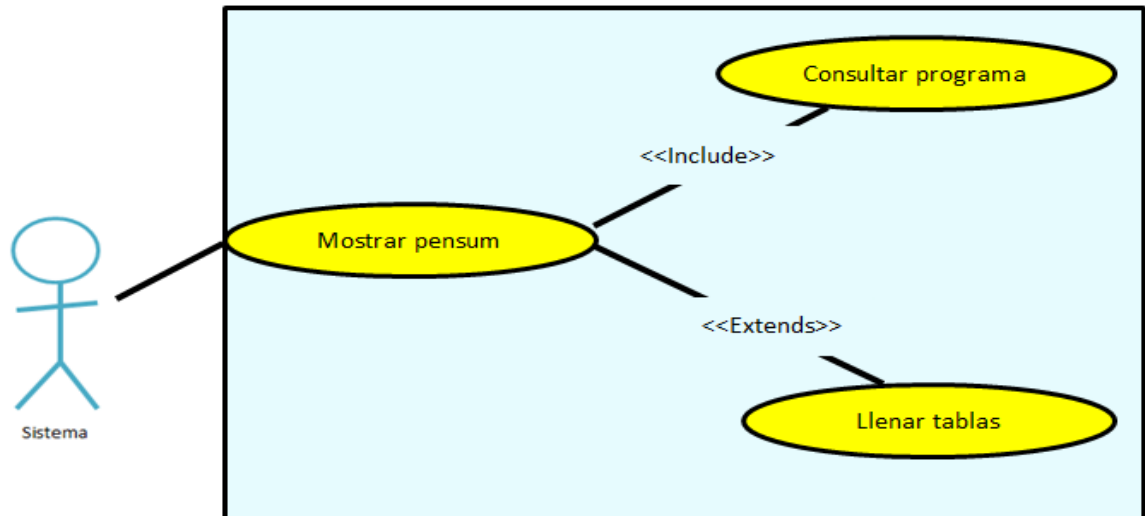


Gráfico N° 18 - Caso de Uso 7

Nombre:	Llenar tablas con el pensum de la materia correspondiente / CU-7	
Actor:	Sistema	
Descripción:	Proceso que debe realizar el sistema para mostrar en el módulo pensum las materias y semestres correspondientes a el programa del estudiante que ha iniciado sesión.	
Flujo Principal:	Eventos ACTOR	Eventos SISTEMA
	1. Recibe el programa y las materias vistas por el estudiante.	1. Procesar la información recibida.
		2. Mostrar las tablas de los semestres con un color distinto para las materias ya vistas.
Precondición:	La base de datos debe tener un programa con sus materias asociadas.	
Poscondición:	El sistema muestra el pensum en forma de tablas, una por cada semestre.	
Presunción:	El sistema realiza las operaciones necesarias para mostrar el pensum, por materias vistas y pendientes.	

Tabla N° 14 - Caso de Uso 7

Caso de uso 8: Retirar NRC de las combinaciones de horarios

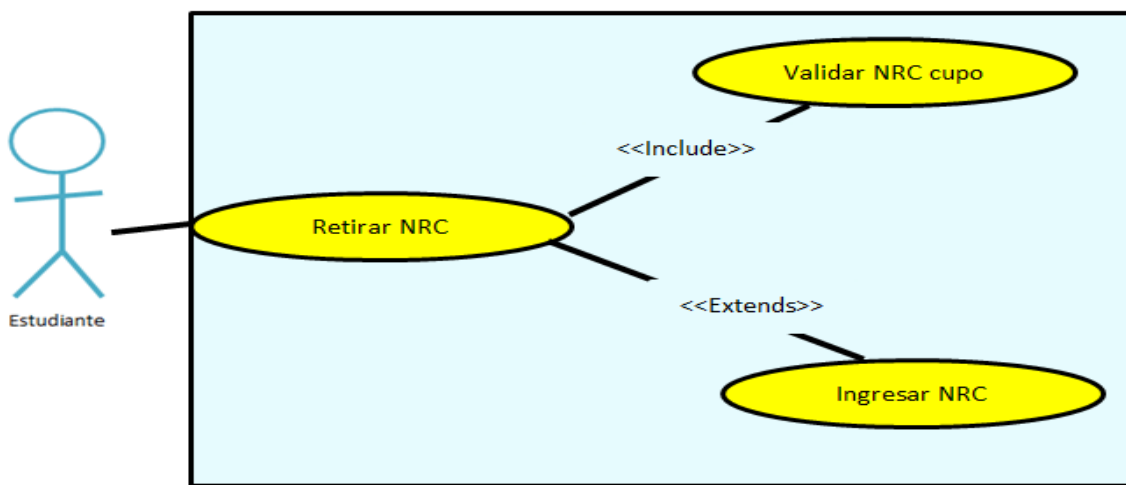


Gráfico N° 19 - Caso de Uso 8

Nombre:	Retirar NRC de las combinaciones de horarios / CU-8	
Actor:	Estudiante	
Descripción:	Proceso que debe realizar un estudiante retirar un NRC de las combinaciones de horarios.	
Flujo Principal:	Eventos ACTOR	Eventos SISTEMA
	1. escribir en el espacio de texto el código NRC.	1. Muestra espacio de texto y botón enviar.
	2. Clic en el botón enviar.	2. Valida si NRC existe.
		3. Muestra por medio de tablas las posibles combinaciones de horarios con las materias seleccionadas sin el NRC retirado.
Alternativa:	1. escribir en el espacio de texto el código NRC.	1. Muestra espacio de texto y botón enviar.
	2. Clic en el botón enviar.	2. Valida si no NRC existe.
		3. Muestra un mensaje indicando que el NRC no existe.
Precondición:	El estudiante debe ingresar NRC que desea retirar.	
Poscondición:	Se muestran las combinaciones de horarios sin el NRC seleccionado por el estudiante.	
Presunción:	El sistema procesa los datos para mostrar las combinaciones de horarios sin cruces de horarios ni repetición de materias.	

Tabla N° 15 - Caso de Uso 8

Caso de uso 9: Consultar estadísticas del sitio

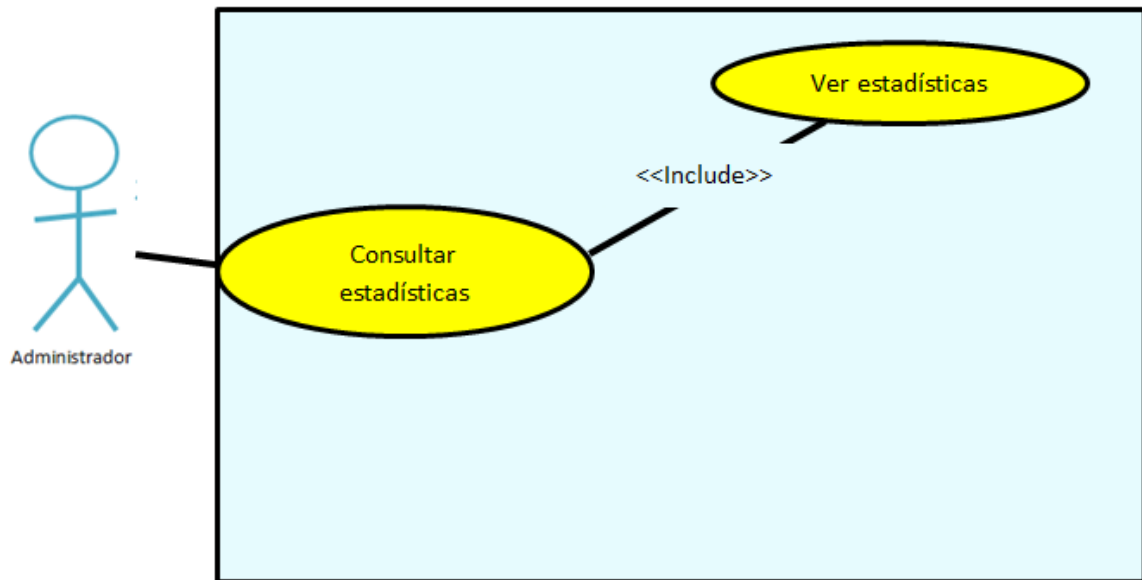


Gráfico N° 20 - Caso de Uso 9

Nombre:	Consultar estadísticas del sitio / CU-9	
Actor:	Administrador	
Descripción:	Proceso que debe realizar un administrador para ver las estadísticas del sitio.	
Flujo Principal:	Eventos ACTOR	Eventos SISTEMA
	1. Dar clic en el botón ver estadísticas.	1. Buscar en la base de datos los datos relacionados con las estadísticas. 2. Mostrar las estadísticas de número de usuarios y número de entradas al sitio.
Precondición:	El administrador debe iniciar sesión.	
Poscondición:	El sistema muestra las estadísticas.	
Presunción:	El sistema recurre a la base de datos y muestra estadísticas reales.	

Tabla N° 16 - Caso de Uso 9

Caso de uso 10: Actualizar programa académico y sus materias

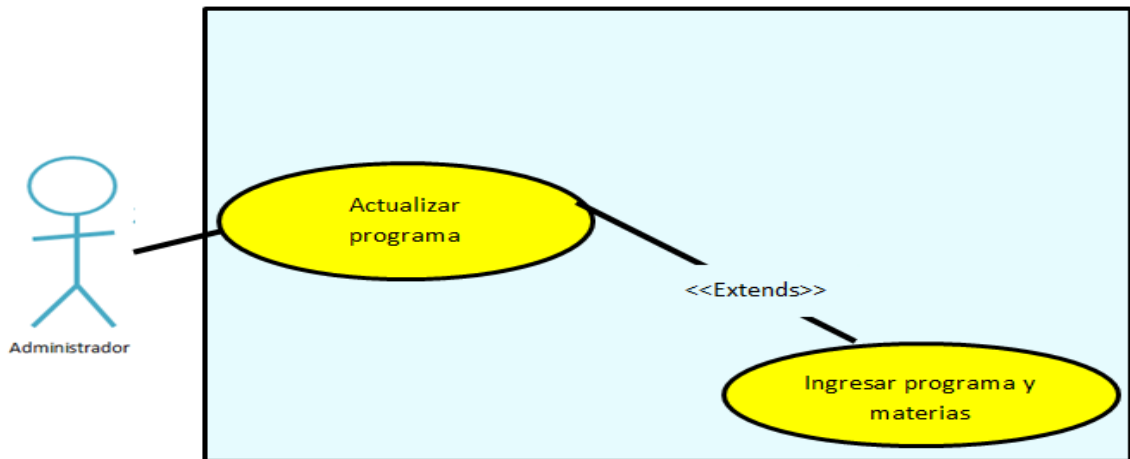


Gráfico N° 21 - Caso de Uso 10

Nombre:	Actualizar programa académico y sus materias / CU-10	
Actor:	Administrador	
Descripción:	Proceso que debe realizar un administrador para actualizar las materias de un programa académico específico.	
Flujo Principal:	Eventos ACTOR	Eventos SISTEMA
	1. Escribir el nombre del programa.	1. Muestra espacio de texto y botón enviar.
	2. Clic en el botón enviar.	2. Valida si el programa existe y mostrar las materias correspondientes al programa en espacios editables.
	3. actualizar las materias que se desea	3. Muestra un mensaje indicando que el programa fue editado correctamente.
Alternativa:	1. Escribir el nombre del programa.	1. Muestra espacio de texto y botón enviar.
	2. Clic en el botón enviar.	2. Valida si el programa no existe.
		3. Muestra un mensaje indicando que el programa no existe.
Precondición:	El administrador debe ingresar al sistema.	
Poscondición:	Se muestran un mensaje indicando que el programa fue editado correctamente.	
Presunción:	El sistema procesa los datos y edita en la base de datos el programa con las materias correspondientes.	

Tabla N° 17 - Caso de Uso 10

Diagrama de casos de uso general del sistema

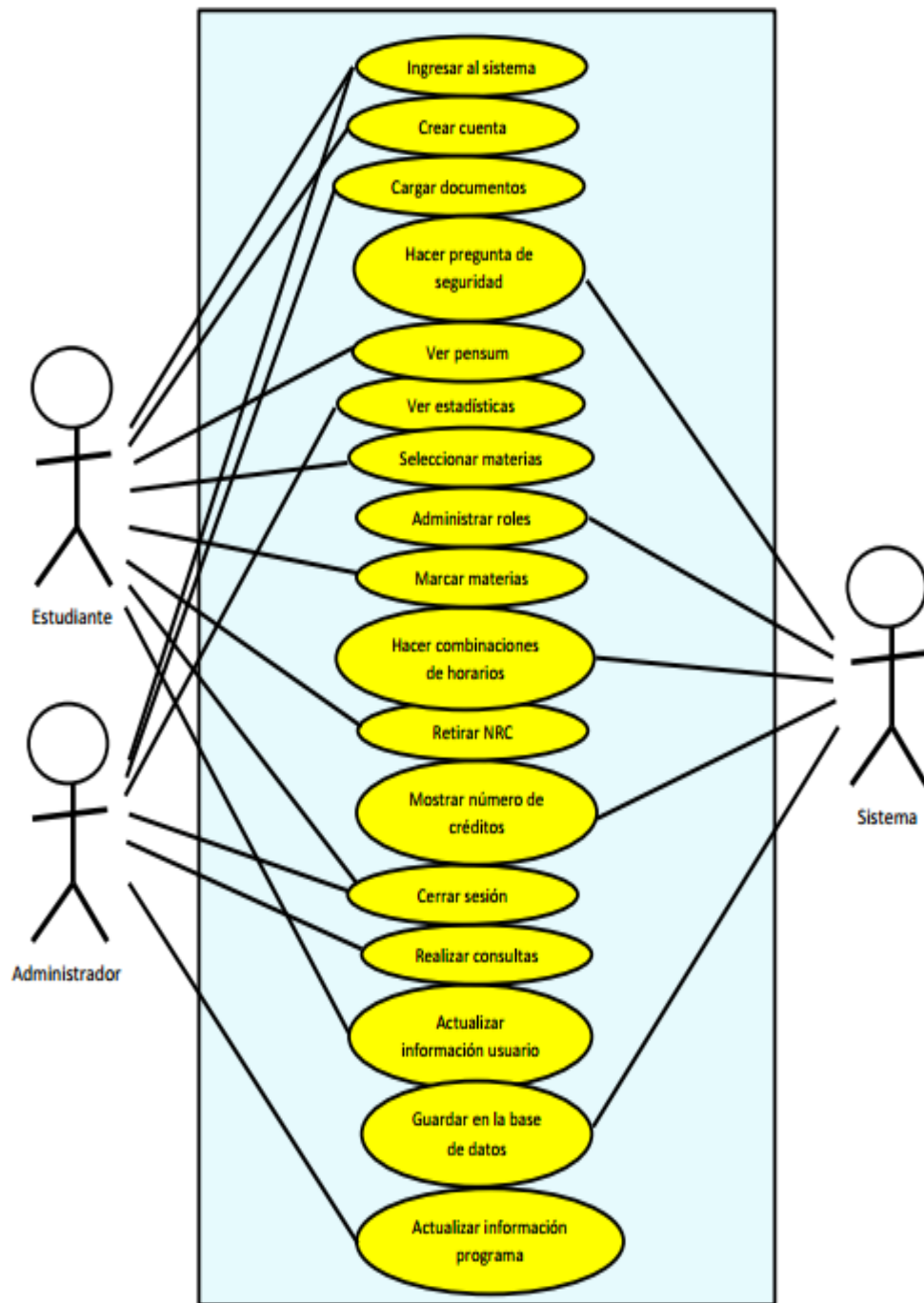


Gráfico N° 22 - Diagrama General Casos de Uso

4. DESARROLLO

El desarrollo se lleva a cabo acorde con el análisis, diseño y el estudio realizado previamente, cumpliendo con todos los requerimientos funcionales y no funcionales, además de mantener un alto estándar en lo que a programación se refiere, con un código fuente lo más optimizado en rendimiento, de fácil mantenimiento y con un esquema que permita y facilite su evolución con miras a nuevas opciones futuras.

4.1. Especificaciones técnicas

A continuación se describen las características a nivel de software y hardware que serán usadas para el desarrollo, las pruebas y los requisitos mínimos que el sistema necesita para un correcto funcionamiento.

4.2. Software

El desarrollo y pruebas se harán completamente con herramientas de software libre, que serán de gran ayuda debido a que no conllevan un valor económico utilizarlas. Estas herramientas son:

- HTML: se utilizará para elaborar la interface gráfica de la aplicación.
- PHP: Lenguaje de programación encargado de la parte lógica de la aplicación en el servidor y de las consultas en la base de datos.
- JavaScript: Lenguaje de programación dinámico que se encargará de las validaciones a nivel de usuario – navegador.
- XAMPP: Herramienta que suministra un motor de base de datos MYSQL y un servidor APACHE de manera local, para hacer las pruebas sin tener que recurrir a un servidor físico.
- APACHE: servidor que nos proporciona una respuesta en el navegador a través de la solicitud del lenguaje PHP.
- Netbeans 7.4: Es el entorno de desarrollo integrado que nos facilita la estructuración del código de los distintos módulos.

Las características mínimas de software, necesarias para utilizar la aplicación con un correcto desempeño son:

- Conexión a Internet de 512k o superior
- Navegador web: Mozilla Firefox, Google Chrome, Internet Explorer v.7 o superior.
- Sistema Operativo: Windows XP o superior, Linux, Mac.

4.3. Hardware

Para facilitar la etapa de desarrollo del aplicativo se utilizará un equipo de altas prestaciones y para las pruebas funcionales se utilizarán equipos con distintas configuraciones para garantizar el correcto funcionamiento de la aplicación el mayor número de dispositivos posibles. Las características físicas del equipo, en el cual se hará el desarrollo del aplicativo son:

- Procesador: AMD 8150 a 3.6Ghz.
- Memoria RAM: 8Gb.
- Disco duro: 1Tb

Las especificaciones de los equipos de pruebas serán, además del equipo de desarrollo las siguientes:

Especificaciones equipo medianas prestaciones:

- Procesador: Intel core i3 a 2.6Ghz.
- Memoria RAM: 2Gb.
- Disco duro: 500Mb

Especificaciones equipo medianas prestaciones:

- Procesador: Intel dual core a 2.1Ghz.
- Memoria RAM: 512Mb.
- Disco duro: 125Mb

Para finalizar las especificaciones mínimas que debe tener un equipo para ejecutar el aplicativo son:

- Procesador: Intel, Amd, Qualcomm, Mediatek, de 800Mhz o superior
- Memoria RAM: 512Mb o superior
- Teclado y Mouse

5. GLOSARIO

Módulo de Gestión: Aplicación orientada a la web, en la cual se permite gestionar información de distintas maneras.

Sistema Génesis: Sistema oficial de la Corporación Universitaria Minuto de Dios, para la inscripción de materias, todo lo relacionado con las notas de los estudiantes, asuntos de financiación y otros aspectos institucionales.

NRC: Código que se le da a una materia en la Corporación Universitaria Minuto de Dios. Una materia puede tener asociados varios códigos NRC.

UML: Lenguaje Unificado de Modelado. Es un estándar a nivel mundial, en lo que se refiere al modelado de software.

PHP: Es un lenguaje de programación interpretado para crear páginas web dinámicas, su uso no es local, por lo que se requiere un servidor para su funcionamiento.

MYSQL: Es un motor de bases de datos libre, el cual puede ser utilizado por cualquier persona. Maneja varias herramientas para el manejo de datos.

Servidor: Es una aplicación informática que provee servicios a otras aplicaciones denominadas Clientes.

Módulo: Es una sección de un programa informático, que se crea de forma independiente pero que interactúa con las demás secciones del programa.

Base de Datos: Es un conjunto de datos pertenecientes a un mismo contexto y almacenados para un posterior uso.

Evolución: En el contexto del desarrollo del software, se refiere a la capacidad de un programa o aplicación de mejorar o incluir nuevas opciones.

Rol: Se refiere a los permisos que tiene determinado usuario en un sistema.

6. CONCLUSIONES

Luego de realizar los procesos de análisis de requerimientos, diseños y prototipos del sistema del módulo de gestión para la inscripción de asignaturas se concluyó que:

- Para lograr un rendimiento mayor es necesario la integración con el sistema Génesis de Uniminuto y así poder realizar la inscripción de asignaturas.
- No fue posible acceder a Génesis en el tiempo de desarrollo del módulo por cuestión de permisos y políticas de privacidad.
- Los diseños propuestos logran realizar los procesos de selección de asignaturas para la creación de horarios adecuadamente.
- Se desarrolló el módulo independiente del sistema génesis logrando una administración de asignaturas por estudiante en un propio sistema de almacenamiento.
- El módulo de gestión se diseñó para facilitar y optimizar el proceso de inscripción de asignaturas.

7. REFERENCIAS

- Alvares, M. (2010, Octubre 15). *Instalación de un servidor Apache*. Recuperado Mayo 8, 2014, de <http://tecnologia.uncomo.com/articulo/como-instalar-un-servidor-apache-en-windows-23819.html>
- Berzal, F. (2010, Marzo 5). *Relaciones entre clases, Diagrama de clases UML*. Recuperado Marzo 20, 2014, de <http://elvex.ugr.es/decsai/java/pdf/3C-Relaciones>
- Freeman, J. (2009, Octubre 12). *Como escribir una introducción y una conclusión en un documento*. Recuperado Mayo 8, 2014, de http://www.ehowenespanol.com/escribir-introduccion-conclusion-documento-estilo-apa-como_30313/
- Haya, M., Franch, X., & Mayol, E. (2010, Abril 6). *Uso de los diagramas de actividades*. Recuperado Abril 5, 2014, de http://wer.inf.puc-rio.br/WERpapers/artigos/artigos_WER04/M
- López, A. (2009, Agosto 5). *Como mantener el modelo vista controlador en una aplicación orientada a la web*. Recuperado Mayo 5, 2014, de <http://biblioteca.uniminuto.edu/ojs/index.php/Inventum/article/view/132>
- López, S. (2008, Agosto 12). *Insertar código PHP en HTML*. Recuperado Marzo 17, 2014, de <http://aprendeaprogramarenphpya.com/blog/%C2%BFsabias-que-existen-cuatro-formas-distintas>
- Montoya, J. (2009, Marzo 9). *Ejemplos de diagramas UML, interfaces gráficas de usuario, y usos del UML en la ingeniería inversa*. Recuperado Febrero 2, 2014, de <http://www.monografias.com/trabajos67/diagramas-uml/diagramas-uml.shtml>

- Mosqueira, E. (2008, Septiembre 4). *El proceso unificado de desarrollo de software*. Recuperado Marzo 3, 2014, de http://quegrande.org/apuntes/EI/OPT/PAI/teoria/07-08/tema_2_-_el_proceso_unificado
- Riesco, D. (2009, enero 14). *Diagrama de clases y objetos*. Recuperado Marzo 20, 2014, de <http://www.sel.unsl.edu.ar/licenciatura/ingsoft2/UML-DiagramaClaseObjeto>
- Sarmiento, M. (2010, Mayo 4). *Diagrama de estados*. Recuperado Marzo 21, 2014, de <http://markblogs-markmendoza.blogspot.com/2010/12/diagramas-de-estado.html>
- Scott, F. (2011, Julio 8). *El desarrollo de software como ingeniería de software*. Recuperado Mayo 12, 2014, de <http://web.usbmed.edu.co/usbmed/fing/v2n2/v2n2a1>
- Sierra, M. (2010, Octubre 15). *Cuáles son los principales tipos de servidores*. Recuperado Marzo 28, 2014, de http://www.aprenderaprogramar.com/index.php?option=com_content&view=article&id=542:que-es-un-servidor
- Tresancos, J. (2013, Mayo 16). *JQuery Datepicker, manual de uso avanzado*. Recuperado Mayo 22, 2014, de <http://www.desarrolloweb.com/articulos/jquery-ui-datepicker-avanzado.htm>

