



**TRABAJO FIN DE CARRERA  
INGENIERÍA INFORMÁTICA  
TEMA ASP.NET CON WCF**

**MEMORIA  
APLICACIÓN: NET-ACADEMY**

**Software para la gestión de las actividades  
de una Academia.**

**Alumno: Pilar Hernando Botto**

**Tutor: Juan Carlos González Martín**

**Fecha: 02/03/2011 – 13/06/2011**



A mi familia, por su paciencia y apoyo en los momentos difíciles y por su confianza en mí.

## **INDICE DE CONTENIDO**

INDICE DE CONTENIDO.....	2
INDICE DE FIGURAS .....	4
1 INTRODUCCIÓN .....	6
2 JUSTIFICACIÓN Y OBJETIVOS DEL PROYECTO. ....	6
2.1 Justificación del Proyecto y contexto en el que se desarrolla. ....	6
2.2 Objetivos del proyecto.....	8
3 PLANIFICACIÓN INICIAL Y REAL .....	8
3.1 Relación de actividades .....	8
3.1.1 Calendario de actividades (diagrama de Gantt).....	9
3.2 Hitos principales .....	10
3.3 Descripción de los otros capítulos de la Memoria .....	10
4 SÍNTESIS DE LA DOCUMENTACIÓN DE ANÁLISIS, DISEÑO E IMPLEMENTACIÓN .....	11
4.1 Análisis .....	11
4.1.1 Recogida de requerimientos .....	11
4.1.2 MÓDULOS DEL PROYECTO.....	12
4.2 DIAGRAMAS DE CASOS DE USO .....	13
4.2.1 Paquete Actors .....	13
4.2.2 Casos de uso del paquete Gestión de Acceso .....	15
4.2.3 Casos de uso del Paquete Manager Admissions .....	17
4.2.4 Casos de uso del Paquete Manager and Explotation .....	19
4.3 DETALLE CASOS DE USO .....	28
4.3.1 PAQUETE GESTIÓN DE ACCESO .....	28
4.3.2 PAQUETE MANGER ADMISSIONS.....	29
4.3.3 PAQUETE MANAGER AND EXPLOTATION .....	31
4.4 DIAGRAMA DE CLASES DEL MODELO CONCEPTUAL .....	35
5 DISEÑO .....	38
5.1 DIAGRAMA DE LAS ARQUITECTURAS DE SOFTWARE Y HARDWARE	38
5.1.1 SOFTWARE.....	38
5.1.2 HARDWARE.....	38
5.2 DISEÑO DE LA BASE DE DATOS: MODELO LOGICO.....	39
5.3 INTERFAZ DE USUARIO.....	42
5.3.1 PANTALLA DE INICIO DE LA APLICACIÓN.....	42
5.3.2 Pantallas correspondientes al perfil Administrador. ....	43
5.3.3 Pantallas correspondientes al perfil Profesor .....	50

5.3.4	Pantallas correspondientes al perfil Alumno .....	52
6	IMPLEMENTACIÓN .....	53
6.1	CAPAS DE LA APLICACIÓN .....	53
6.1.1	Capa UI: Proyecto NetAcademy.WebApplycation .....	54
6.1.2	Capa WCF: Proyecto NetAcademy.WebServices .....	59
6.1.3	Capa Entities: NetAcademy.Entities .....	62
6.1.4	Capa DataBase: NetAcademy.DataBase .....	63
6.1.5	Capa DataAccess: NetAcademy.DataAccess .....	63
6.1.6	MANUAL DE INSTALACIÓN Y/O CONFIGURACIÓN.....	65
6.1.7	Pasos para poder ejecutar la aplicación en entorno de Desarrollo 65	
6.1.8	Pasos para poder ejecutar la aplicación en entorno Real.....	65
6.1.9	Manual de usuario .....	65
7	EVALUACIÓN .....	66
7.1	Objetivos conseguidos.....	66
7.2	Evaluación de costes .....	66
7.3	Trabajo futuro y recomendaciones de mejora .....	67
8	CONCLUSIONES.....	67
9	BIBLIOGRAFÍA .....	68

## INDICE DE FIGURAS

Figura 1: Estructura del marco de trabajo .NET .....	7
Figura 2: Esquema de las herramientas de Microsoft utilizadas .....	8
Figura 3: Tabla actividades y plazos del Proyecto.....	8
Figura 4: Diagrama de Gantt del Proyecto.....	9
Figura 5: Paquetes del Diagrama de Casos de Uso del Proyecto.....	13
Figura 6: Paquete Actors .....	13
Figura 7: Descripción del actor Administrador.....	14
Figura 8: Descripción del actor profesor.....	14
Figura 9: Descripción del actor Alumno.....	14
Figura 10: Paquete Gestión de Acceso .....	15
Figura 11: Descripción del caso de uso Acceso a la aplicación.....	16
Figura 12: Descripción del caso de uso Modificación contraseña.....	16
Figura 13: Descripción del caso de uso Alta usuario .....	17
Figura 14: Paquete Manager Admissions.....	17
Figura 15: Descripción de caso de uso Gestión Usuarios .....	18
Figura 16: Paquete Manager and Explotation.....	19
Figura 17: Descripción del caso de uso Gestión Aulas .....	20
Figura 18: Descripción del caso de uso Gestión Cursos .....	21
Figura 19: Descripción del caso de uso Gestión Matrículas .....	22
Figura 20: Descripción del caso de uso Gestión Horarios.....	23
Figura 21: Descripción del caso de uso Consultar Listados .....	24
Figura 22: Descripción del caso de uso Consultar Horario por Profesor .....	25
Figura 23: Descripción del caso de uso evaluar Alumno .....	26
Figura 24: Descripción del caso de uso Consultar horario por Alumno.....	27
Figura 25: Descripción del caso de uso Consultar Notas.....	27
Figura 26: Diagrama de flujo Acceso al Sistema .....	28
Figura 27: Diagrama de flujo de la Modificación de Contraseña.....	28
Figura 28: Diagrama de flujo Gestión de Usuarios.....	29
Figura 29: Diagrama de Flujo de editUsuario .....	30
Figura 30: Diagrama de flujo de DeleteUsuario .....	30
Figura 31: Diagrama de flujo GestionAulas.....	31
Figura 32: Diagrama de flujo GestiónCursos.....	31
Figura 33: Diagrama de Flujo GestionMatriculas .....	32
Figura 34: Diagrama de flujo GestionHorarios .....	32
Figura 35: Diagrama de flujo Consultar Listados .....	33
Figura 36: Diagrama de Flujo Consultar Horarios Profesor.....	33
Figura 37: Diagrama Flujo Evaluación Alumno .....	34
Figura 38: Diagrama de flujo Consulta Horario Alumno.....	34
Figura 39: Diagrama de flujo Consultar Notas por el Alumno .....	35
Figura 40: Diagrama Clases del modelo conceptual inicial .....	36
Figura 41: Diagrama Clases del modelo conceptual final .....	37
Figura 42: Diagrama arquitectura aplicación. ....	38
Figura 43: Modelo Lógico de la BBDD de la Aplicación inicial .....	40

Figura 44: Modelo Lógico de la BBDD final .....	41
Figura 45: Detalle de las claves primarias y foráneas de la Base de Datos Final .....	42
Figura 46: Pantalla de acceso a la aplicación .....	42
Figura 47: Pantalla con las opciones del perfil Administrador. ....	43
Figura 48: Ventana con las opciones del perfil Administrador de NET-Academy .....	43
Figura 49: Ventana de la Gestión de Usuarios de NET-Academy.....	44
Figura 50: Ventana Gestión de Usuarios, opción Añadir .....	45
Figura 51: Ventana Gestión de Usuarios opción Buscar.....	45
Figura 52: Ventana de la Gestión de Usuarios opciones Modificar/Eliminar	46
Figura 53: Ventana de Gestión de Aulas .....	46
Figura 54: Ventana Gestión de Aulas opción Buscar .....	47
Figura 55: Ventana Gestión de Horarios .....	47
Figura 56: Ventana Gestión de Cursos.....	48
Figura 57: Ventana Gestión de Cursos opción Buscar, Modificar , Eliminar	48
Figura 58: Ventana Gestión de Matrículas .....	49
Figura 59: Ventana Gestión de Listados perfil Administrador .....	49
Figura 60: Ventana inicial del perfil Profesor.....	50
Figura 61: Ventana Modificar Password Profesor .....	50
Figura 62: Ventana Evaluar Alumno perfil Profesor.....	51
Figura 63: Ventan Evaluar Alumno perfil Profesor opciones .....	51
Figura 64: Ventana de inicio de NET-Academy para el perfil de Alumno ....	52
Figura 65: Ventana de búsqueda de Horarios por Curso perfil Alumno .....	52
Figura 66: Ventana de Consulta notas del Alumno por curso .....	53
Figura 67: Capas de la Aplicación NET-Academy .....	53
Figura 68: Diseño de la Ventana Gestión de Usuario .....	54

## 1 INTRODUCCIÓN

La arquitectura SOA se está convirtiendo en una influencia de diseño importante en el software moderno. Se trata de una filosofía arquitectónica que fomenta la construcción o acoplamiento de grandes sistemas distribuidos a partir de extremos débilmente acoplados que exponen su funcionalidad a través de interfaces bien conocidas. WCF representa los principios estándar de esta arquitectura y ofrece un alto rendimiento y una máxima interoperabilidad.

El objeto de este Trabajo Fin de Carrera es crear una aplicación que gestione el funcionamiento de una Academia de Enseñanza, utilizando la plataforma .Net Framework 4.0, como entorno de programación se utilizará Microsoft Visual Studio 2010 (Ultimate) y por último la gestión de la persistencia de los datos se realiza con Microsoft SQL 2008.

NET-Academy podrá ser utilizada por tres perfiles distintos de usuarios, Administrador, Profesor y Alumno, cada uno de ellos tendrá acceso a determinadas funcionalidades. El perfil Administrador es el encargado de la gestión de los usuarios, aulas, horarios, cursos y matrículas, así como de la consulta a diferentes listados. El perfil Profesor es el encargado de la gestión de la evaluación de los Alumnos y también tiene acceso a la consulta de distintos listados y por último el perfil Alumno podrá consultar tanto los cursos como sus calificaciones.

La realización del proyecto se ha dividido en cuatro etapas, Plan de Trabajo, Análisis y Diseño, Implementación y por último la Memoria y el vídeo en el que se presenta el proyecto, cada una de las cuales ha finalizado con la entrega del producto correspondiente.


## 2 JUSTIFICACIÓN Y OBJETIVOS DEL PROYECTO.

### 2.1 Justificación del Proyecto y contexto en el que se desarrolla.


El proyecto NET\_Academy se plantea como la solución a la gestión de una academia de enseñanza, de modo que se puedan realizar las operaciones necesarias para llevarla a cabo.

Para ello se han tenido en cuenta los diferentes perfiles de los usuarios que van a utilizarla, dándoles el acceso correspondiente a las funcionalidades de la aplicación que les corresponden, el perfil Administrador será el encargado de la gestión de usuarios, aulas, horarios, cursos y matrículas, el perfil Profesor será el encargado de la evaluación de los alumnos y el perfil Alumno tendrá acceso a la consulta de cursos y notas. Tanto Administrador como Profesor también podrán acceder a la consulta de diferentes listados.

Para la realización del proyecto se utilizará:

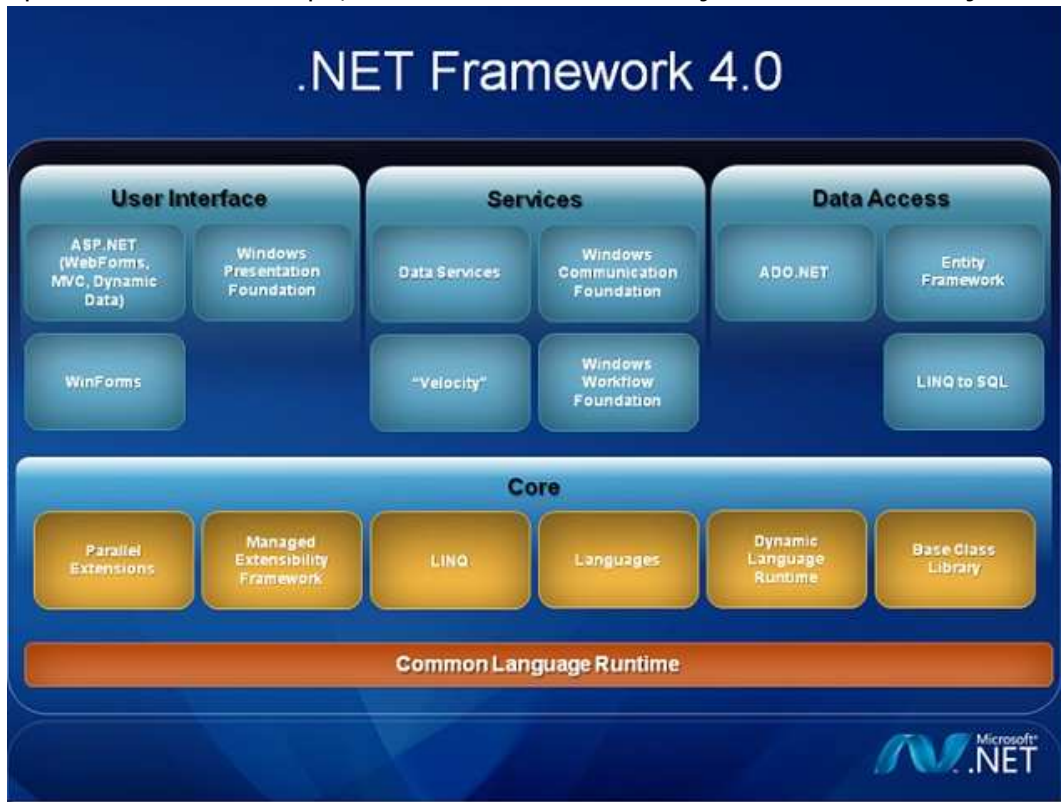
 La plataforma .NET Framework 4.0.

 Microsoft Visual Studio 2010(Ultimate) como entorno de programación.

 Como software de apoyo, para confeccionar los documentos así como los diferentes tipos de diagramas y pantallas del prototipo, se ha utilizado, procesador de textos Microsoft Word 2007, programa de diseño Microsoft Visio 2007, diagrama de Gantt, Microsoft Project 2007. Por último para la grabación del vídeo presentación, Microsoft Expression Encoder 4 Screen




Capture para grabarlo y Microsoft Expression Encoder 4 para montarlo y generar el archivo .avi

El .NET Framework es un conjunto de herramientas que facilitan la creación de aplicaciones de todo tipo, así como un entorno de ejecución donde se ejecutan.



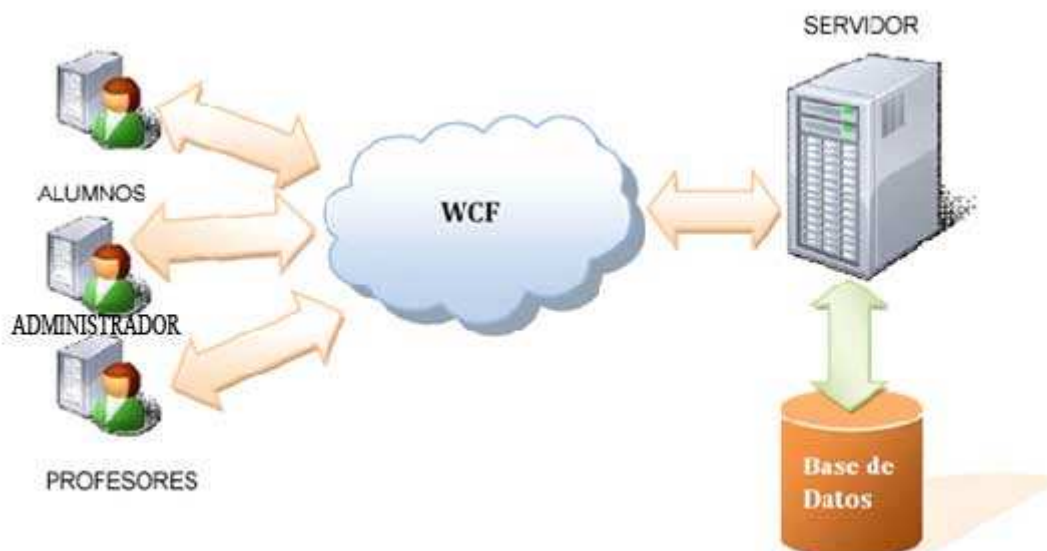
**Figura 1: Estructura del marco de trabajo .NET**

Para el desarrollo del software se utilizarán las herramientas de Microsoft usando la tecnología .NET, concretamente:

-  WCF (Windows Communications Foundation)
-  SQL Server 2008
-  ASP.NET

A continuación se muestra el esquema.





**Figura 2: Esquema de las herramientas de Microsoft utilizadas**

## 2.2 Objetivos del proyecto

El objetivo principal del proyecto es crear una aplicación para la gestión de una academia de enseñanza, desarrollada en ASP.NET y que utiliza servicios WCF para acceder a la lógica de negocio.

## 3 PLANIFICACIÓN INICIAL Y REAL

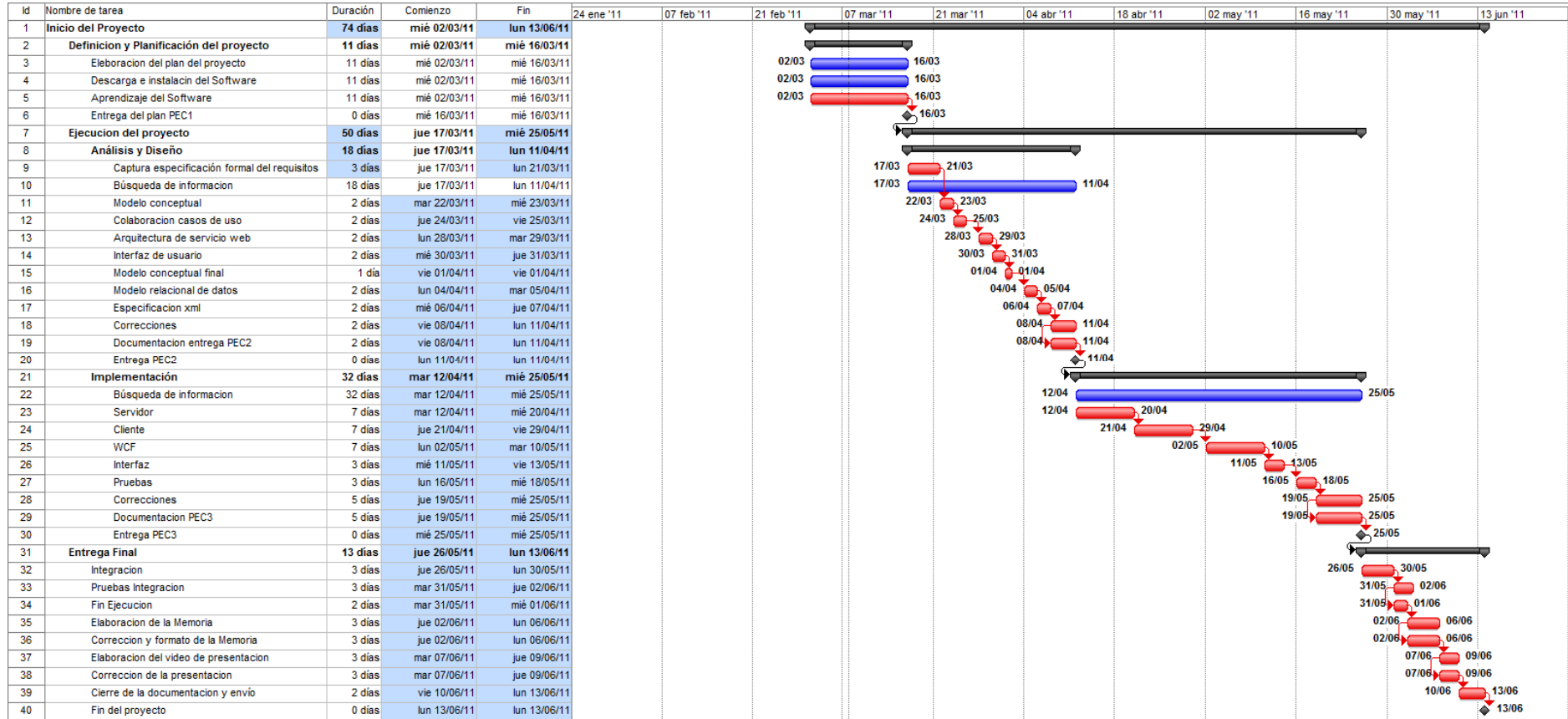
### 3.1 Relación de actividades

El proyecto consta de cuatro objetivos o tareas que se enumeran a continuación, el final de cada uno de ellos es un Hito del proyecto cuya fecha es inamovible, en el Diagrama de Gantt del apartado siguiente se detalla cada una de las Tareas. Así mismo se muestra en color Rojo el camino crítico y en color azul las tareas no críticas que se corresponden esencialmente a la de búsqueda de información y estudio.

TAREA	INICIO	FIN
Plan de Trabajo (PEC1)	02/03/2011	16/03/2011
Análisis y Diseño (PEC2)	17/03/2011	11/04/2011
Implementación (PEC3)	12/04/2011	23/05/2011
Memoria y Presentación virtual (Entrega Final)	24/05/2011	13/06/2011

**Figura 3: Tabla actividades y plazos del Proyecto.**






**3.1.1 Calendario de actividades (diagrama de Gantt)**



**Figura 4: Diagrama de Gantt del Proyecto**

### 3.2 Hitos principales




Según el calendario de trabajo presentado estaban previstos los siguientes hitos:


-  Inicio del proyecto, marcado por la fecha de inicio del semestre académico , 2 de marzo de 2011
-  Entrega PEC1: Presentación del Plan de Trabajo, que marcará el desarrollo posterior del proyecto.
-  Entrega PEC2: Análisis y Diseño. En este punto deben estar completamente acotados los requisitos y realizado en su totalidad el análisis y diseño de todos los módulos que comprende la entrega del proyecto. Estas actividades deben acompañarse de la documentación oportuna. Este hito es fundamental para el correcto avance del proyecto ya que es importante asegurarse de que a partir de este punto todos los esfuerzos se centrarán en la implementación y no habrá que volver atrás a redefinir o rediseñar ningún punto.
-  Entrega PEC3: Implementación. En este punto deben estar completamente implementados todos los módulos que componen el proyecto (Estas actividades comprenden una parte importante de investigación y estudio de las herramientas necesarias) y Este hito viene marcado también por el fin del desarrollo, incluida la fase de pruebas oportunas. Se incluye también la tarea de integración de los módulos desarrollados, si la hubiere, así como una carga de datos para la entrega final.
-  Fin de proyecto. Marcado por la entrega de la memoria y del video con la presentación del proyecto el 13 de junio de 2011

Estos hitos definen el camino crítico del proyecto, se han alcanzado todos en el tiempo marcado, por lo tanto no ha habido ningún cambio entre la planificación inicial y la final.

### 3.3 Descripción de los otros capítulos de la Memoria

A continuación se hace un pequeño resumen del resto de capítulos de la memoria:

-  **Análisis.** En este capítulo se especifican y explican los requerimientos funcionales de los diferentes módulos de la aplicación, mediante los diagramas de casos de uso y el diagrama de clases del modelo conceptual, así como los cambios realizados con respecto al primer documento presentado
-  **Diseño.** En este capítulo se detallan los paquetes de los que se costa la aplicación así como su arquitectura y los requerimientos de hardware necesarios para disponer de todas las funcionalidades de los distintos módulos del proyecto, se detalla el modelo lógico conceptual para la BBDD que necesita el programa y la interfaz de usuario, al igual que en el capítulo anterior se destacan los cambios realizados con respecto al documento inicial.
-  **Implementación.** Se detallan los aspectos más relevantes de la implementación.

 **Evaluación.** Se detallan los objetivos conseguidos, la evaluación de los costes, así como las posibles líneas de trabajo en el futuro y las recomendaciones de mejora.

 **Conclusiones.**

 **Bibliografía.**

## 4 SÍNTESIS DE LA DOCUMENTACIÓN DE ANÁLISIS, DISEÑO E IMPLEMENTACIÓN

### 4.1 Análisis

#### 4.1.1 Recogida de requerimientos

La aplicación NET\_Academy será utilizada por tres perfiles diferentes de usuarios: Administrador, Profesor y Alumno. Cada uno de los cuales desarrollara una serie de roles dentro del sistema, en función de los cuales tendrán acceso a las diferentes funcionalidades de la aplicación. A continuación se detallan las funcionalidades de estos roles:

#### Administrador

- a) Podrá añadir, editar y eliminar usuarios
- b) Podrá añadir, editar y eliminar profesores
- c) Podrá añadir, editar y eliminar alumnos
- d) Podrá añadir, editar y eliminar aulas
- e) Podrá añadir, editar y eliminar cursos
- f) Podrá añadir , editar y eliminar matriculas
- g) Podrá añadir, editar y eliminar horarios
- h) Podrá sacar listados de: Usuarios, profesores, alumnos, aulas, cursos y matriculas

#### Profesor

- a) Podrá consultar los horarios de las aulas y del profesor
- b) Podrá buscar al alumno para añadir, modificar y eliminar las notas, así como poner la evaluación final
- c) Podrá añadir un examen, practica,...

#### Alumno

- a) Podrá consultar los horarios de las aulas y solicitar aula o consultar los horarios del profesor y solicitar profesor
- b) Podrá consultar las notas
- c) Podrá solicitar la revisión de una nota.

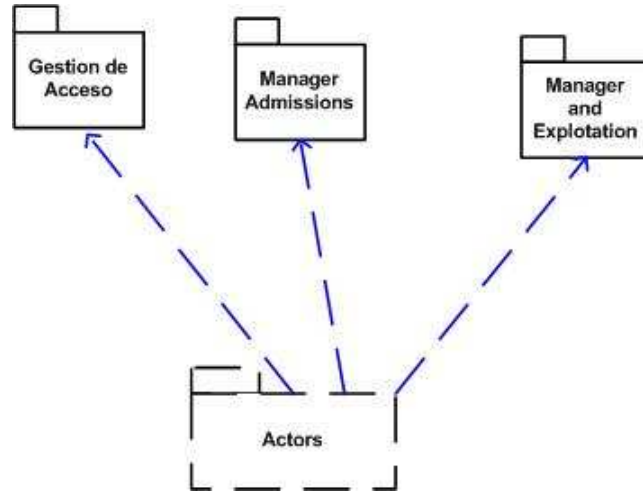
#### 4.1.2 MÓDULOS DEL PROYECTO

Para llevar a cabo la resolución del proyecto se dividirá en varios módulos, cuya relación variará dependiendo de sus características. A continuación se detalla su funcionamiento y sus funcionalidades:





MODULO	DESCRIPCIÓN
<b>GESTION DE ACCESO</b>	<ul style="list-style-type: none"> <li>✓ El usuario de NET-Academy se deberá de identificar para poder acceder a la aplicación.</li> <li>✓ Los usuarios se identificaran con un login y una contraseña asignada por el sistema</li> </ul>
<b>MANAGER ADMISSIONS</b>	<ul style="list-style-type: none"> <li>✓ El administrador dará de alta a los profesores y a los alumnos</li> <li>✓ Los usuarios tendrán unos datos comunes(nombre, apellidos,...) y otros datos concretos según su perfil</li> </ul>
<b>MANAGER AND EXPLOTATION</b>	<ul style="list-style-type: none"> <li>✓ El Administrador creará, modificará y eliminara las aulas. Para cada aula se especificará su capacidad y su localización</li> <li>✓ El Administrador creara, modificará y eliminara los cursos. Para cada curso se indicara el aula, horario, profesor, duración y número máximo de alumnos</li> <li>✓ El Administrador añadirá, modificara y eliminara las matrículas indicando el alumno y el curso</li> <li>✓ El administrador añadirá, modificará y eliminará los horarios. Para cada horario se especificara el aula, el curso, la hora de inicio, la hora de fin y el día de la semana</li> <li>✓ El administrador podrá consultar listados de profesores, alumnos, aulas, cursos, matrículas</li> <li>✓ Los profesores podrán consultar los horarios de las aulas y del profesor</li> <li>✓ Los profesores podrán añadir , modificar y eliminar las notas de los alumnos así como calcular la evaluación final</li> <li>✓ Los profesores podrán añadir exámenes, pruebas,...</li> <li>✓ Los alumnos podrán consultar los horarios de las aulas y solicitarla</li> <li>✓ Los alumnos podrán consultar los horarios de los profesores y solicitar una clase del profesor</li> <li>✓ Los alumnos podrán consultar las notas</li> <li>✓ Los alumnos podrán solicitar la revisión de una nota.</li> </ul>

## 4.2 DIAGRAMAS DE CASOS DE USO

El diagrama de casos de uso se ha dividido en paquetes para facilitar su comprensión. A continuación se muestra el diagrama y se describen las responsabilidades de cada paquete

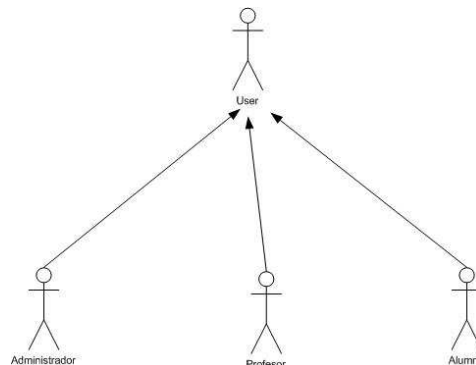


**Figura 5: Paquetes del Diagrama de Casos de Uso del Proyecto**

-  El paquete **Gestión de Acceso** contiene todos los casos de uso relacionado con la autenticación de los usuarios.
-  El paquete **Actors** contiene todos los actores que interactúan con el sistema para cumplir con sus objetivos.
-  El paquete **Manage and Explotation** contiene todos los casos de uso relacionado con el funcionamiento de las aulas, cursos, matriculas, alumnos y profesores.
-  El paquete **Manage admissions** contiene todos los casos de uso relacionado con la gestión de usuarios.

### 4.2.1 Paquete Actors

El paquete Actors contiene todos los actores que interactúan con el sistema para cumplir con sus objetivos. Un actor representa una entidad con algún comportamiento definido, como una persona (identificada por un rol), sistema informatizado u organización que está relacionado con el sistema que estamos modelando.



**Figura 6: Paquete Actors**

<b>Actor</b>	<i>Administrador</i>
<b>Autor</b>	Pilar Hernando Boto
<b>Fuentes</b>	Cliente
<b>Descripción</b>	El <i>Administrador</i> representa un rol del sistema que está encargado de dar de alta a los usuarios, cambiar su contraseña, introducir sus datos personales, adjudicarles el rol que les corresponde (administrador, profesor o alumno) ,gestionar las aulas, los cursos, los horarios, las matrículas y consultar los informes del sistema.
<b>Justificación</b>	El <i>Administrador</i> es un actor del sistema que participa de las historias de los casos de uso proporcionando y recibiendo información del sistema con el objetivo de administrar el sistema.

**Figura 7: Descripción del actor Administrador**

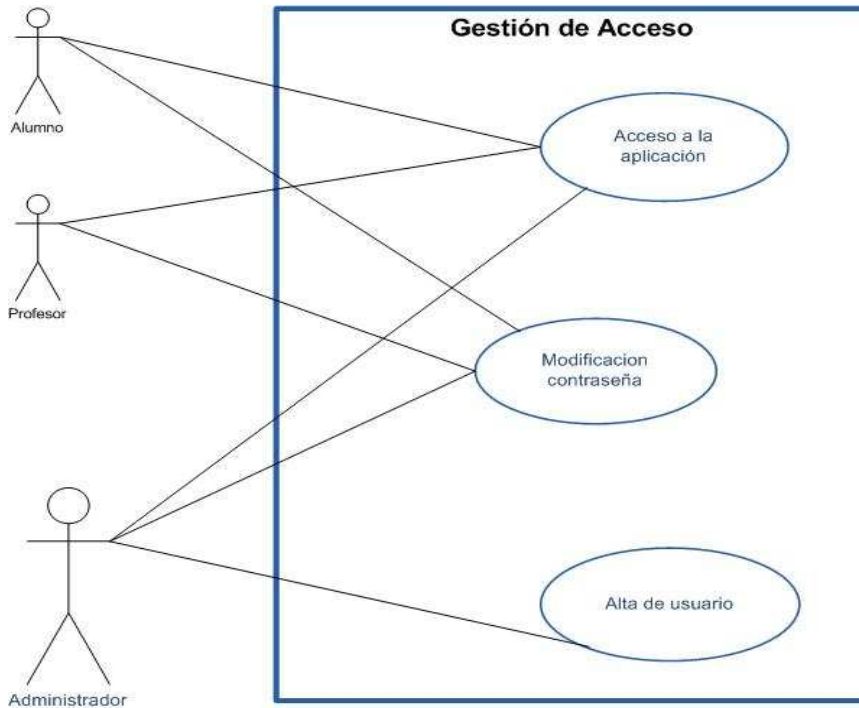
<b>Actor</b>	<i>Profesor</i>
<b>Autor</b>	Pilar Hernando Boto
<b>Fuentes</b>	Cliente
<b>Descripción</b>	El <i>Profesor</i> representa un rol del sistema que imparte las clases, crea los documentos para las pruebas de los alumnos y los evalúa. El <i>Profesor</i> usa el sistema para buscar las aulas, los horarios y a los alumnos, así como cambiar su contraseña.
<b>Justificación</b>	El <i>Profesor</i> es un actor del sistema que participa de las historias de los casos de uso proporcionando y recibiendo información del sistema para cumplir con sus objetivos: gestionar sus clases y evaluar a sus alumnos.

**Figura 8: Descripción del actor profesor**

<b>Actor</b>	<i>Alumno</i>
<b>Autores</b>	Pilar Hernando Boto
<b>Fuentes</b>	Cliente
<b>Descripción</b>	El <i>Alumno</i> representa un rol del sistema que disfruta del servicio de las clases en las que está matriculado. El <i>Alumno</i> usa el sistema para buscar las aulas y solicitar la admisión en las mismas, para buscar los profesores y solicitar la admisión en sus clases, para buscar sus notas y para solicitar la revisión de una nota, así como para cambiar su contraseña
<b>Justificación</b>	El <i>Alumno</i> es un actor del sistema que participa de las historias de los casos de uso proporcionando y recibiendo información del sistema para cumplir con sus objetivos: disfruta del servicio de las clases.

**Figura 9: Descripción del actor Alumno**

**4.2.2 Casos de uso del paquete Gestión de Acceso**



**Figura 10: Paquete Gestión de Acceso**

**4.2.2.1 Descripción de los casos de uso del paquete Gestión de Acceso**

<b>Caso de uso</b>	Acceso a la aplicación
<b>Paquete</b>	Gestión de Acceso
<b>Autores</b>	Pilar Hernando Botto
<b>Fuente</b>	Cliente
<b>Actores</b>	<i>Administrador, Profesor, Alumno</i>
<b>Descripción</b>	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un <i>Administrador, Profesor, Alumno</i> desea entrar en el sistema introduciendo su identificador y su contraseña. El <i>Administrador, Profesor, Alumno</i> solicita al sistema comenzar la sesión. El sistema solicita el identificador y la contraseña. El sistema comprueba que los datos del <i>Administrador, Profesor, Alumno</i> son correctos y comienza la sesión.
<b>Precondiciones</b>	- El <i>Administrador, Profesor o Alumno</i> están autenticados en el sistema
<b>Poscondiciones</b>	- El <i>Administrador, Profesor o Alumno</i> entran el sistema.
<b>Flujo principal</b>	1 El <i>Administrador, Profesor o Alumno</i> solicita al sistema comenzar la sesión. 2.El sistema solicita los siguientes datos: identificador y contraseña 3. El <i>Administrador, Profesor o Alumno</i> proporciona los datos requeridos y solicita al sistema el comienzo de la sesión. 4. El sistema comienza la sesión y le conduce a la pantalla adecuada según el perfil.
<b>Flujo alternativo</b>	<b>Cancelar operación.</b> En cualquier paso, si el <i>Administrador, Profesor o Alumno</i> solicita cancelar la operación, el sistema cancela la operación sin comenzar la



	sesión. <b>Error en los datos.</b> Si los datos no son correctos el sistema mostrará un mensaje de error indicándolo.
--	--

**Figura 11: Descripción del caso de uso Acceso a la aplicación**

<b>Caso de uso</b>	Modificación contraseña
<b>Paquete</b>	Gestión de Acceso
<b>Autores</b>	Pilar Hernando Botto
<b>Fuente</b>	Cliente
<b>Actores</b>	<i>Administrador, Profesor, Alumno</i>
<b>Descripción</b>	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un <i>Administrador, Profesor, Alumno</i> desea cambiar la contraseña. El <i>Administrador, Profesor, Alumno</i> solicita al sistema cambiar la contraseña. El sistema solicita la contraseña anterior y la contraseña nueva. El sistema registra el cambio de contraseña.
<b>Precondiciones</b>	- El <i>Administrador, Profesor o Alumno</i> están autenticados en el sistema
<b>Poscondiciones</b>	- El <i>Administrador, Profesor o Alumno</i> cambian la contraseña.
<b>Flujo principal</b>	<ol style="list-style-type: none"> <li>1 El <i>Administrador, Profesor o Alumno</i> solicita al sistema cambiar la contraseña.</li> <li>2. El sistema solicita los siguientes datos: contraseña antigua y contraseña nueva</li> <li>3. El <i>Administrador, Profesor o Alumno</i> proporciona los datos requeridos y solicita al sistema el cambio de contraseña.</li> <li>4. El sistema cambia la contraseña.</li> </ol>
<b>Flujo alternativo</b>	<b>Cancelar operación.</b> En cualquier paso, si el <i>Administrador, Profesor o Alumno</i> solicita cancelar la operación, el sistema cancela la operación sin cambiar la contraseña. <b>Error en los datos.</b> Si los datos no son correctos el sistema mostrará un mensaje de error indicándolo.

**Figura 12: Descripción del caso de uso Modificación contraseña**

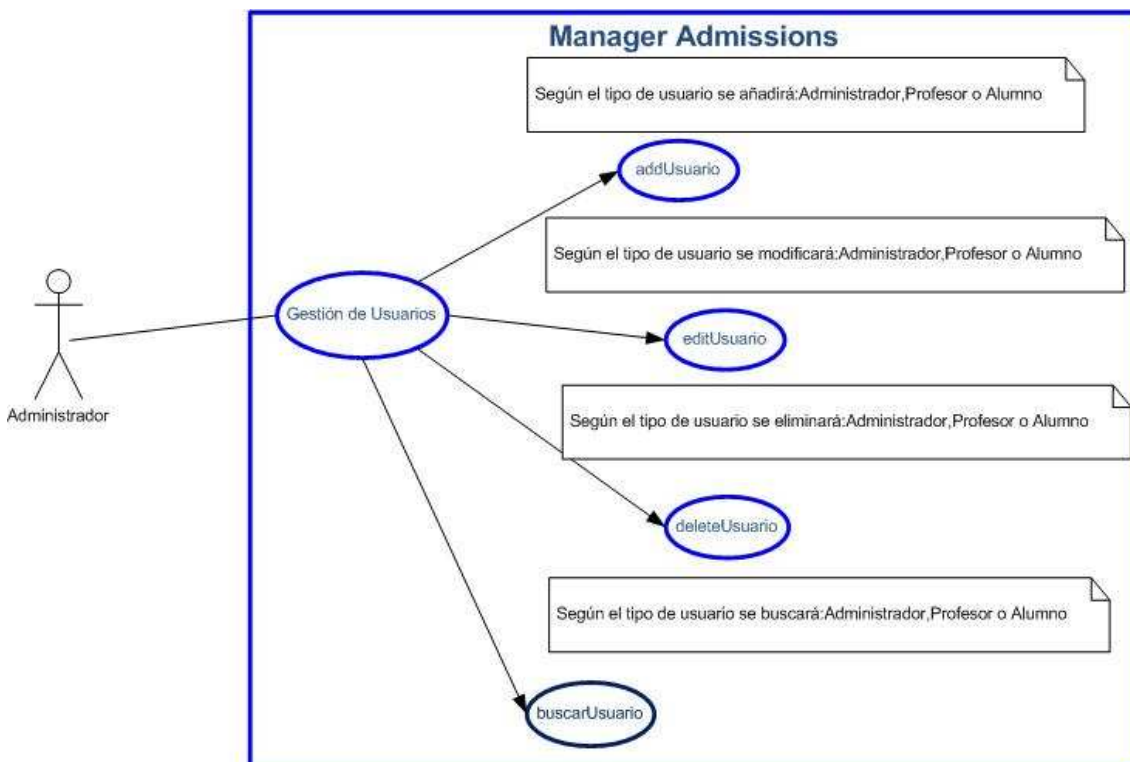
<b>Caso de uso</b>	Alta de usuario
<b>Paquete</b>	Gestión de Acceso
<b>Autores</b>	Pilar Hernando Botto
<b>Fuente</b>	Cliente
<b>Actores</b>	<i>Administrador</i>
<b>Descripción</b>	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un <i>Administrador</i> , desea dar de alta a un nuevo usuario. El <i>Administrador</i> , solicita al sistema dar de alta a un nuevo usuario. El sistema solicita el identificador y la contraseña del nuevo usuario. El sistema da de alta al nuevo usuario.
<b>Precondiciones</b>	- El <i>Administrador</i> están autenticados en el sistema
<b>Poscondiciones</b>	- El sistema da de alta al nuevo usuario.

<b>Flujo principal</b>	1 El <i>Administrador</i> solicita al sistema el alta de un usuario 2.El sistema solicita los siguientes datos: identificador y contraseña 3. El <i>Administrador</i> , proporciona los datos requeridos y solicita al sistema el alta de usuario. 4. El sistema da de alta al usuario
<b>Flujo alternativo</b>	<b>Cancelar operación.</b> En cualquier paso, si el <i>Administrador</i> , solicita cancelar la operación, el sistema cancela la operación sin comenzar la sesión. <b>Error en los datos.</b> Si los datos no son correctos el sistema mostrará un mensaje de error indicándolo.

**Figura 13: Descripción del caso de uso Alta usuario**

### 4.2.3 Casos de uso del Paquete Manager Admissions

En la primera versión los usuarios Administrador, Alumno y Profesor se trataban de forma independiente, en la versión final existe un solo caso de uso Gestión Usuarios, ya que se han unificado los anteriores de Gestión de Alumnos y Gestión de Profesores. Este cambio también afecta al diseño de la BBDD.



**Figura 14: Paquete Manager Admissions**

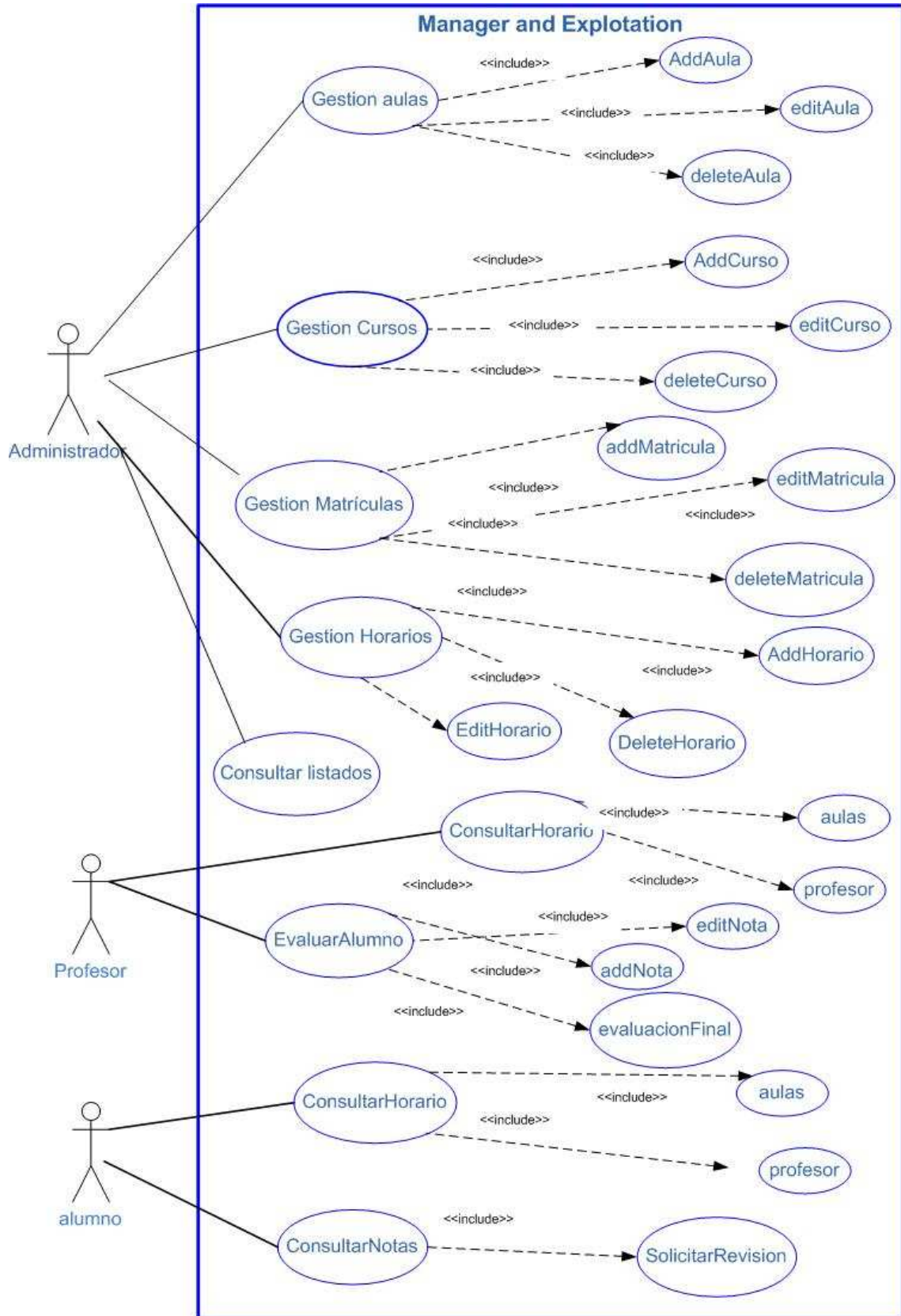
#### 4.2.3.1 Descripción de los casos de uso del Paquete Manager Admissions

<b>Caso de uso</b>	GestiónUsuarios
<b>Paquete</b>	Manager Admissions
<b>Autores</b>	Pilar Hernando Botto
<b>Fuente</b>	Cliente
<b>Actores</b>	<i>Administrador</i>

<b>Descripción</b>	<p>El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un <i>Administrador</i>, desea gestionar un. El <i>Administrador</i>, solicita al sistema que le muestre las opciones de la gestión de usuarios. El sistema muestra al <i>Administrador</i> las opciones de la gestión de usuarios: addUsuario, editUsuario , deleteUsuario y buscarUsuario</p>
<b>Precondiciones</b>	<ul style="list-style-type: none"> <li>- El <i>Administrador</i> están autenticados en el sistema.</li> </ul>
<b>Poscondiciones</b>	<ul style="list-style-type: none"> <li>- El sistema da de alta los cambios en los datos de los usuarios</li> </ul>
<b>Flujo principal</b>	<ol style="list-style-type: none"> <li>1. El <i>Administrador</i> seleccionará Gestión Usuarios y el sistema le mostrará la pantalla correspondiente con las siguientes opciones:             <ol style="list-style-type: none"> <li>1.1. addUsuario: el <i>Administrador</i> deberá introducir los datos comunes a todos los usuarios ( NIF, nombre, apellidos, dirección, población, teléfono y email), a continuación seleccionará el tipo de usuario (<i>Administrador</i>, <i>Profesor</i> o <i>Alumno</i>) y el sistema le mostrará los datos específicos de cada uno , una vez completados, el sistema le solicitará su aprobación .</li> <li>1.2. editUsuario: el <i>Administrador</i> introducirá el NIF o el nombre del usuario, así como el perfil al que pertenece, el sistema le mostrará los datos del usuario, el <i>Administrador</i> modificará los datos, el sistema solicitará la aprobación de los cambios.</li> <li>1.3. deleteUsuario: El <i>Administrador</i> introducirá el NIF del usuario o el nombre, así como el perfil al que pertenece el <i>Administrador</i> solicita al sistema la eliminación del usuario, el sistema le solicitará la aprobación de la operación.</li> <li>1.4. buscarUsuario: El <i>Administrador</i> introducirá el NIF del usuario o el nombre, así como el perfil al que corresponde el sistema le mostrará una lista con los datos de los usuarios que coinciden con las condiciones de búsqueda. El <i>Administrador</i> lo seleccionará y el sistema le mostrará todos los datos para que si es necesario el <i>Administrador</i> proceda a su modificación o a su eliminación, de ser así se procederá como se indica en los puntos 1.2 y 1.3</li> </ol> </li> <li>2. El sistema efectúa los cambios en la base de datos.</li> </ol>
<b>Flujo alternativo</b>	<p><b>Cancelar operación.</b> En cualquier paso, si el <i>Administrador</i>, solicita cancelar la operación, el sistema cancela la operación sin efectuar ningún cambio.</p> <p><b>Error en los datos.</b> Si los datos no son correctos el sistema mostrará un mensaje de error indicándolo.</p>

**Figura 15: Descripción de caso de uso Gestión Usuarios**

4.2.4 Casos de uso del Paquete Manager and Explotation



**Figura 16: Paquete Manager and Explotation**

4.2.4.1 Descripción de los casos de uso del Paquete Manager and Explotation

<b>Caso de uso</b>	Gestión Aulas
<b>Paquete</b>	Manager and Explotation
<b>Autores</b>	Pilar Hernando Botto
<b>Fuente</b>	Cliente
<b>Actores</b>	<i>Administrador</i>
<b>Descripción</b>	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un <i>Administrador</i> , desea gestionar un aula. El <i>Administrador</i> , solicita al sistema que le muestre las opciones de la gestión de aulas. El sistema muestra al <i>Administrador</i> las opciones de la gestión de aulas: addAula, editAula, buscarAula y deleteAula.
<b>Precondiciones</b>	- El <i>Administrador</i> está autenticado en el sistema
<b>Poscondiciones</b>	- El sistema da de alta los cambios en los datos de las aulas
<b>Flujo principal</b>	<ol style="list-style-type: none"> <li>1. El Administrador seleccionará una opción y el sistema le mostrará la pantalla correspondiente: <ol style="list-style-type: none"> <li>1.1. addAula: el <i>Administrador</i> deberá introducir los datos del aula (Identificador del Aula, localización, capacidad), el sistema le solicitará la aprobación de los datos.</li> <li>1.2. editAula: el Administrador introducirá el identificador del Aula, el sistema le mostrará los datos del aula, el Administrador modificará los datos, el sistema solicitará la aprobación de los cambios.</li> <li>1.3. deleteAula: El Administrador introducirá el identificador del aula, el <i>Administrador</i> solicita al sistema la eliminación del aula, el sistema le solicitará la aprobación de la operación.</li> <li>1.4. buscarAula: El Administrador introducirá el nombre del aula o su localización, el sistema le mostrará una tabla con la o las aulas que cumplan con los criterios, el Administrador podrá seleccionar una de las Aulas, el sistema le mostrará todos los datos del Aula seleccionada, el Administrador podrá visualizarlos o ejecutar los puntos 1.3 o 1.4. Si no se introducen criterios de búsqueda, el sistema mostrará el listado completo de las Aulas que actualmente están dadas de alta.</li> </ol> </li> <li>2. El sistema efectúa los cambios en la base de datos.</li> </ol>
<b>Flujo alternativo</b>	<p><b>Cancelar operación.</b> En cualquier paso, si el <i>Administrador</i>, solicita cancelar la operación, el sistema cancela la operación sin efectuar ningún cambio.</p> <p><b>Error en los datos.</b> Si los datos no son correctos el sistema mostrará un mensaje de error indicándolo.</p>

**Figura 17: Descripción del caso de uso Gestión Aulas**

<b>Caso de uso</b>	Gestión Cursos
<b>Paquete</b>	Manager and Explotation

<b>Autores</b>	Pilar Hernando Botto
<b>Fuente</b>	Cliente
<b>Actores</b>	<i>Administrador</i>
<b>Descripción</b>	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un <i>Administrador</i> , desea gestionar un curso. El <i>Administrador</i> , solicita al sistema que le muestre las opciones de la gestión de cursos. El sistema muestra al <i>Administrador</i> las opciones de la gestión de cursos: addCurso, editCurso y deleteCurso.
<b>Precondiciones</b>	- El <i>Administrador</i> está autenticado en el sistema
<b>Poscondiciones</b>	- El sistema da de alta los cambios en los datos de los cursos
<b>Flujo principal</b>	<ol style="list-style-type: none"> <li>1. El Administrador seleccionará una opción y el sistema le mostrará la pantalla correspondiente: <ol style="list-style-type: none"> <li>1.1. addCurso: el <i>Administrador</i> deberá introducir los datos del curso (Identificador del curso, nombre del curso, identificador del Aula, horario, profesor, duración, documentos), el sistema le solicitará la aprobación de los datos.</li> <li>1.2. editCurso: el Administrador introducirá el identificador del curso, el sistema le mostrará los datos del curso, el Administrador modificará los datos, el sistema solicitará la aprobación de los cambios.</li> <li>1.3. deleteCurso: El Administrador introducirá el identificador del curso, el <i>Administrador</i> solicita al sistema la eliminación del curso, el sistema le solicitará la aprobación de la operación.</li> <li>1.4. buscarCurso: El Administrador introducirá el nombre del curso , el sistema le mostrará una tabla con el o los cursos que cumplan con los criterios, el Administrador podrá seleccionar un de los Cursos, el sistema le mostrará todos los datos del Curso seleccionado, el Administrador podrá visualizarlos o ejecutar los puntos 1.3 o 1.4. Si no se introducen criterios de búsqueda, el sistema mostrará el listado completo de los Cursos que actualmente están dados de alta.</li> </ol> </li> <li>2. El sistema efectúa los cambios en la base de datos.</li> </ol>
<b>Flujo alternativo</b>	<p><b>Cancelar operación.</b> En cualquier paso, si el <i>Administrador</i>, solicita cancelar la operación, el sistema cancela la operación sin efectuar ningún cambio.</p> <p><b>Error en los datos.</b> Si los datos no son correctos el sistema mostrará un mensaje de error indicándolo.</p>

**Figura 18: Descripción del caso de uso Gestión Cursos**

<b>Caso de uso</b>	Gestión Matriculas
<b>Paquete</b>	Manager and Explotation

<b>Autores</b>	Pilar Hernando Botto
<b>Fuente</b>	Cliente
<b>Actores</b>	<i>Administrador</i>
<b>Descripción</b>	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un <i>Administrador</i> , desea gestionar una matrícula. El <i>Administrador</i> , solicita al sistema que le muestre las opciones de la gestión de matrículas. El sistema muestra al <i>Administrador</i> las opciones de la gestión de cursos: addMatricula, editMatricula y deleteMatricula.
<b>Precondiciones</b>	- El <i>Administrador</i> está autenticado en el sistema
<b>Poscondiciones</b>	- El sistema da de alta los cambios en los datos de las matriculas
<b>Flujo principal</b>	<ol style="list-style-type: none"> <li>1. El Administrador seleccionará una opción y el sistema le mostrará la pantalla correspondiente: <ol style="list-style-type: none"> <li>1.1. addMatricula: el <i>Administrador</i> deberá introducir los datos de la matrícula (Identificador de matrícula, Identificador del curso, NIF del alumno), el sistema le solicitará la aprobación de los datos.</li> <li>1.2. editMatricula: el Administrador introducirá el identificador de la matrícula, el sistema le mostrará los datos de la matrícula, el Administrador modificará los datos, el sistema solicitará la aprobación de los cambios.</li> <li>1.3. deleteMatricula: El Administrador introducirá el identificador de la matrícula, el <i>Administrador</i> solicita al sistema la eliminación de la matrícula, el sistema le solicitará la aprobación de la operación.</li> <li>1.4. buscarMatricula: El Administrador introducirá el nombre del Curso, el sistema le mostrará una tabla con las Matriculas que cumplan con los criterios, el Administrador podrá seleccionar una de las Matriculas, el sistema le mostrará todos los datos de la Matricula seleccionada, el Administrador podrá visualizarlos o ejecutar los puntos 1.3 o 1.4. Si no se introducen criterios de búsqueda, el sistema mostrará el listado completo de las Matriculas que actualmente están dadas de alta.</li> </ol> </li> <li>2. El sistema efectúa los cambios en la base de datos.</li> </ol>
<b>Flujo alternativo</b>	<p><b>Cancelar operación.</b> En cualquier paso, si el <i>Administrador</i>, solicita cancelar la operación, el sistema cancela la operación sin efectuar ningún cambio.</p> <p><b>Error en los datos.</b> Si los datos no son correctos el sistema mostrará un mensaje de error indicándolo.</p>

**Figura 19: Descripción del caso de uso Gestión Matrículas**

<b>Caso de uso</b>	Gestión horarios
<b>Paquete</b>	Manager and Explotation
<b>Autores</b>	Pilar Hernando Botto
<b>Fuente</b>	Cliente
<b>Actores</b>	<i>Administrador</i>
<b>Descripción</b>	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un <i>Administrador</i> , desea gestionar un horario. El <i>Administrador</i> , solicita al sistema que le muestre las opciones de la gestión de horarios. El sistema muestra al <i>Administrador</i> las opciones de la gestión de horarios: addHorario, editHorario y deleteHorario.
<b>Precondiciones</b>	- El <i>Administrador</i> está autenticado en el sistema
<b>Poscondiciones</b>	- El sistema da de alta los cambios en los datos del horario
<b>Flujo principal</b>	<ol style="list-style-type: none"> <li>1. El Administrador seleccionará una opción y el sistema le mostrará la pantalla correspondiente: <ol style="list-style-type: none"> <li>1.1. addHorario: el <i>Administrador</i> deberá introducir los datos del horario (Identificador de horario, Identificador del aula, identificador del curso, NIF del profesor), el sistema le solicitará la aprobación de los datos.</li> <li>1.2. editHorario: el Administrador introducirá el identificador del horario, el sistema le mostrará los datos del horario, el Administrador modificará los datos, el sistema solicitará la aprobación de los cambios.</li> <li>1.3. deleteHorario: El Administrador introducirá el identificador del horario, el <i>Administrador</i> solicita al sistema la eliminación del horario, el sistema le solicitará la aprobación de la operación.</li> <li>1.4. buscarHorario: El Administrador introducirá el nombre del aula o la hora de comienzo, el sistema le mostrará una tabla con el o los horarios que cumplan con los criterios, el Administrador podrá seleccionar un Horario, el sistema le mostrará todos los datos del Horario seleccionado, el Administrador podrá visualizarlos o ejecutar los puntos 1.3 o 1.4. Si no se introducen criterios de búsqueda, el sistema mostrará el listado completo de los Horarios que actualmente están dados de alta.</li> </ol> </li> <li>2. El sistema efectúa los cambios en la base de datos.</li> </ol>
<b>Flujo alternativo</b>	<p><b>Cancelar operación.</b> En cualquier paso, si el <i>Administrador</i>, solicita cancelar la operación, el sistema cancela la operación sin efectuar ningún cambio.</p> <p><b>Error en los datos.</b> Si los datos no son correctos el sistema mostrará un mensaje de error indicándolo.</p>

**Figura 20: Descripción del caso de uso Gestión Horarios**



<b>Caso de uso</b>	Consulta listados
<b>Paquete</b>	Manager and Explotation
<b>Autores</b>	Pilar Hernando Botto
<b>Fuente</b>	Cliente
<b>Actores</b>	Administrador
<b>Descripción</b>	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un Administrador, desea consultar un listado . El Administrador, solicita al sistema que le muestre los listados disponibles. El sistema muestra al Administrador la lista de los listados disponibles: Listado de alumnos, listado de profesores, listado de aulas, listado de cursos, listado de horarios, listado de matriculas
<b>Precondiciones</b>	- El Administrador está autenticado en el sistema
<b>Poscondiciones</b>	- El sistema muestra el listado.
<b>Flujo principal</b>	<ol style="list-style-type: none"> <li>1. El Administrador seleccionará una opción y el sistema le mostrará la pantalla correspondiente:             <ol style="list-style-type: none"> <li>1.1. Listado de alumnos: el Administrador seleccionará esta opción y el sistema le mostrará el listado.</li> <li>1.2. Listado de profesores: el Administrador seleccionará esta opción y el sistema le mostrará el listado.</li> <li>1.3. Listado de aulas: el Administrador seleccionará esta opción y el sistema le mostrará el listado.</li> <li>1.4. Listado de cursos: el Administrador seleccionará esta opción y el sistema le mostrará el listado.</li> <li>1.5. Listado de horarios: el Administrador seleccionará esta opción y el sistema le mostrará el listado.</li> <li>1.6. Listado de matrículas: el Administrador seleccionará esta opción y el sistema le mostrará el listado.</li> </ol> </li> <li>2. El sistema mostrará el listado.</li> </ol>
<b>Flujo alternativo</b>	<p>Cancelar operación. En cualquier paso, si el Administrador, solicita cancelar la operación, el sistema cancela la operación.</p> <p>Error en los datos. Si los datos no son correctos el sistema mostrará un mensaje de error indicándolo.</p>

**Figura 21: Descripción del caso de uso Consultar Listados**

<b>Caso de uso</b>	Consultar Horario por Profesor
<b>Paquete</b>	Manager and Explotation
<b>Autores</b>	Pilar Hernando Botto
<b>Fuente</b>	Cliente
<b>Actores</b>	<i>Profesor</i>
<b>Descripción</b>	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un <i>Profesor</i> , desea consultar un horario. El <i>Profesor</i> , solicita al sistema que le muestre las opciones de la consulta de horarios. El sistema muestra al <i>Profesor</i> las opciones de las consultas de horarios: Horarios de Aulas, Horarios de profesor.
<b>Precondiciones</b>	- El <i>Profesor</i> está autenticado en el sistema
<b>Poscondiciones</b>	- El sistema muestra el horario solicitado
<b>Flujo principal</b>	<ol style="list-style-type: none"> <li>1. El Profesor seleccionará una opción y el sistema le mostrará la pantalla correspondiente: <ol style="list-style-type: none"> <li>1.1. Horarios de Aulas: el <i>Profesor</i> deberá introducir el identificador del aula ,el sistema le solicitará la aprobación de los datos.</li> <li>1.2. Horarios de profesor: el Profesor introducirá el identificador del profesor, el sistema mostrará el horario.</li> </ol> </li> <li>2. El sistema muestra el horario solicitado.</li> </ol>
<b>Flujo alternativo</b>	<p><b>Cancelar operación.</b> En cualquier paso, si el <i>Profesor</i>, solicita cancelar la operación, el sistema cancela la operación.</p> <p><b>Error en los datos.</b> Si los datos no son correctos el sistema mostrará un mensaje de error indicándolo.</p>

**Figura 22: Descripción del caso de uso Consultar Horario por Profesor**

<b>Caso de uso</b>	Evaluar Alumno
<b>Paquete</b>	Manager and Explotation
<b>Autores</b>	Pilar Hernando Botto
<b>Fuente</b>	Cliente
<b>Actores</b>	<i>Profesor</i>
<b>Descripción</b>	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un <i>Profesor</i> , desea evaluar a un alumno. El <i>Profesor</i> , solicita al sistema que le muestre las opciones de la evaluación de alumnos. El sistema muestra al <i>Profesor</i> las opciones de la evaluación de alumnos: addNotas, deleteNotas, editNotas , evaluacionFinal, addItem.
<b>Precondiciones</b>	- El <i>Profesor</i> está autenticado en el sistema
<b>Poscondiciones</b>	- El sistema muestra da de alta los cambios en la evaluación del alumno.
<b>Flujo principal</b>	<ol style="list-style-type: none"> <li>1. El Profesor seleccionará una opción y el sistema le mostrará la pantalla correspondiente: <ol style="list-style-type: none"> <li>1.1. addNota: el <i>Profesor</i> deberá introducir el NIF del alumno, el sistema le mostrará los cursos del alumno, el profesor seleccionará un curso, el sistema le mostrará los datos del cursos, el profesor añadirá la nota, el sistema le pedirá la aprobación de</li> </ol> </li> </ol>

	<p>los cambios.</p> <p>1.2. editNota: el <i>Profesor</i> deberá introducir el NIF del alumno, el sistema le mostrará los cursos del alumno, el profesor seleccionará un curso, el sistema le mostrará los datos del curso, el profesor modificará la nota. El sistema solicitará la aprobación de los cambios.</p> <p>1.3. deleteNota: el <i>Profesor</i> deberá introducir el NIF del alumno, el sistema le mostrará los cursos del alumno, el profesor seleccionará un curso, el sistema le mostrará los datos del curso, el profesor eliminará la nota. El sistema solicitará la aprobación de los cambios.</p> <p>1.4. buscarAlumno: El Profesor introducirá el nombre del alumno, el sistema le mostrará una tabla con el o los Alumnos que cumplan con los criterios, el Administrador podrá seleccionar un Alumno, el sistema le mostrará todos los datos del Alumno seleccionado, el Administrador podrá visualizarlos o ejecutar los puntos 1.3 o 1.4. Si no se introducen criterios de búsqueda, el sistema mostrará el listado completo de los alumnos que actualmente están dados de alta.</p> <p>1.5. EvaluaciónFinal: el <i>Profesor</i> deberá introducir el NIF del alumno, el sistema le mostrará los cursos del alumno, el profesor seleccionará un curso, el sistema le mostrará los datos del curso, el profesor seleccionará la opción evaluación final. El sistema solicitará la aprobación de los cambios.</p> <p>1.6. addItem: el <i>Profesor</i> deberá introducir el identificador del curso, el sistema mostrará los datos del curso, el profesor añadirá un nuevo documento (examen, prueba de evaluación)</p>
<b>Flujo alternativo</b>	<p><b>Cancelar operación.</b> En cualquier paso, si el <i>Profesor</i>, solicita cancelar la operación, el sistema cancela la operación.</p> <p><b>Error en los datos.</b> Si los datos no son correctos el sistema mostrará un mensaje de error indicándolo.</p>

**Figura 23: Descripción del caso de uso evaluar Alumno**

<b>Caso de uso</b>	Consultar Horario por Alumno
<b>Paquete</b>	Manager and Exploitation
<b>Autores</b>	Pilar Hernando Botto
<b>Fuente</b>	Cliente
<b>Actores</b>	<i>Alumno</i>
<b>Descripción</b>	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un <i>Alumno</i> , desea consultar un horario. El <i>Alumno</i> , solicita al sistema que le muestre las opciones de la consulta de horarios. El sistema muestra al <i>Alumno</i> las opciones de las consultas de horarios: Horarios de Aulas, Horarios de profesor.

<b>Precondiciones</b>	- El <i>Alumno</i> está autenticado en el sistema
<b>Poscondiciones</b>	- El sistema muestra el horario solicitado
<b>Flujo principal</b>	<ol style="list-style-type: none"> <li>1. El <i>Alumno</i> seleccionará una opción y el sistema le mostrará la pantalla correspondiente: <ol style="list-style-type: none"> <li>1.1. Horarios de Aulas: el <i>Alumno</i> deberá introducir el identificador del aula, el sistema le solicitará la aprobación de los datos.</li> <li>1.2. Horarios de profesor: el <i>Alumno</i> introducirá el identificador del profesor, el sistema mostrará el horario.</li> </ol> </li> <li>2. El sistema muestra el horario solicitado.</li> </ol>
<b>Flujo alternativo</b>	<p><b>Cancelar operación.</b> En cualquier paso, si el <i>Alumno</i>, solicita cancelar la operación, el sistema cancela la operación.</p> <p><b>Error en los datos.</b> Si los datos no son correctos el sistema mostrará un mensaje de error indicándolo.</p>

**Figura 24: Descripción del caso de uso Consultar horario por Alumno**

<b>Caso de uso</b>	Consultar Notas
<b>Paquete</b>	Manager and Explotation
<b>Autores</b>	Pilar Hernando Botto
<b>Fuente</b>	Cliente
<b>Actores</b>	<i>Alumno</i>
<b>Descripción</b>	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un <i>Alumno</i> , desea consultar sus notas. El <i>Alumno</i> , solicita al sistema que le muestre las opciones de la consulta de notas. El sistema muestra al <i>Alumno</i> las opciones de las consultas de notas: Consulta Notas, Solicitar revisión.
<b>Precondiciones</b>	- El <i>Alumno</i> está autenticado en el sistema
<b>Poscondiciones</b>	- El sistema muestra las notas solicitadas
<b>Flujo principal</b>	<ol style="list-style-type: none"> <li>1. El <i>Alumno</i> seleccionará una opción y el sistema le mostrará la pantalla correspondiente: <ol style="list-style-type: none"> <li>1.1. Consulta de notas: el <i>Alumno</i> deberá introducir su NIF, el sistema le solicitará la aprobación de los datos. El sistema mostrara las notas del alumno</li> <li>1.2. Revisión de notas: el <i>Alumno</i> introducirá su NIF del profesor, el sistema le solicitará la aprobación de los datos. El sistema le mostrará el listado con las notas, el alumno seleccionará la nota para la revisión, el sistema le solicitará la aprobación de los datos.</li> </ol> </li> <li>2. El sistema muestra el resultado de la operación.</li> </ol>
<b>Flujo alternativo</b>	<p><b>Cancelar operación.</b> En cualquier paso, si el <i>Alumno</i>, solicita cancelar la operación, el sistema cancela la operación.</p> <p><b>Error en los datos.</b> Si los datos no son correctos el sistema mostrará un mensaje de error indicándolo.</p>

**Figura 25: Descripción del caso de uso Consultar Notas**

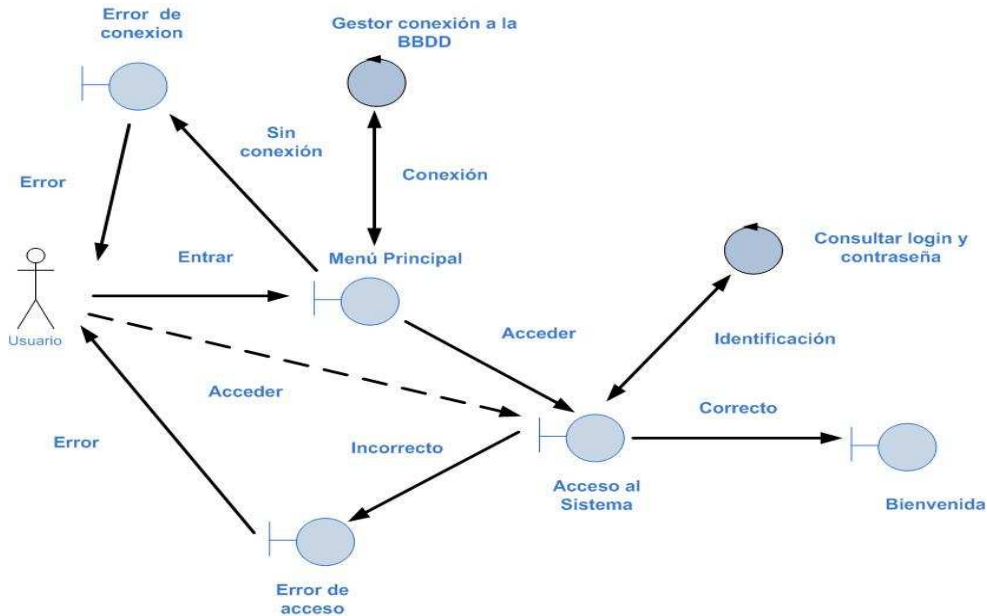
Se puede comprobar como en la versión final a todos los casos de uso del paquete se les ha añadido la opción buscar, desde la cual además de poder visualizar los

todos los datos del elemento, de pueden ejecutar las operaciones de modificación o eliminación del mismo.

### 4.3 DETALLE CASOS DE USO

#### 4.3.1 PAQUETE GESTIÓN DE ACCESO

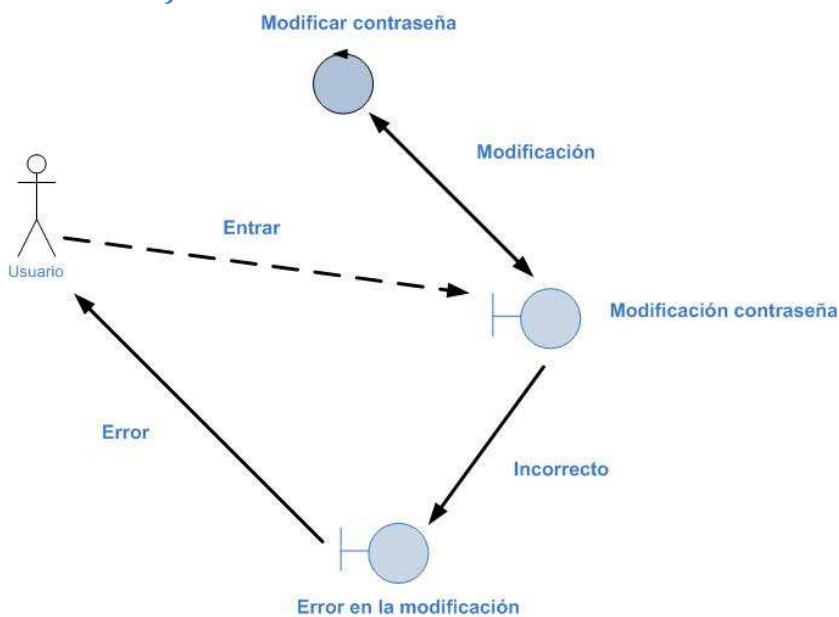
##### 4.3.1.1 ACCESO A LA APLICACIÓN



**Figura 26: Diagrama de flujo Acceso al Sistema**

El primer proceso que debe realizar cualquier tipo de usuario (Administrador, Profesor o Alumno) es identificarse en el Sistema. Si la conexión a la BBDD no es posible o si el usuario no se identifica correctamente se mostrará la correspondiente pantalla de error, si todo es correcto se le dará acceso a la aplicación.

##### 4.3.1.2 Modificación Contraseña



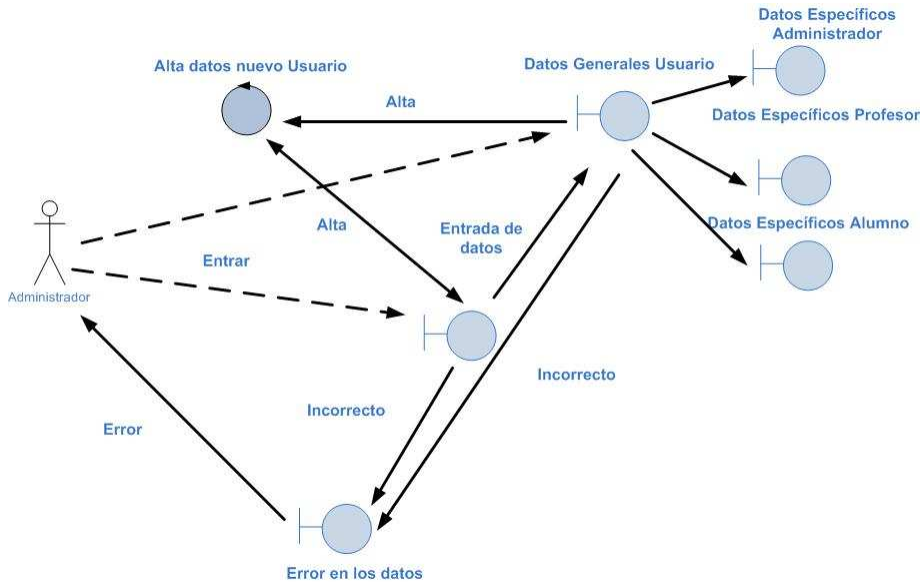
**Figura 27: Diagrama de flujo de la Modificación de Contraseña**

### 4.3.2 PAQUETE MANGER ADMISSIONS

#### 4.3.2.1 GESTION DE UUARIOS

En la versión final ha variado este diagrama, a continuación se muestra la versión definitiva.

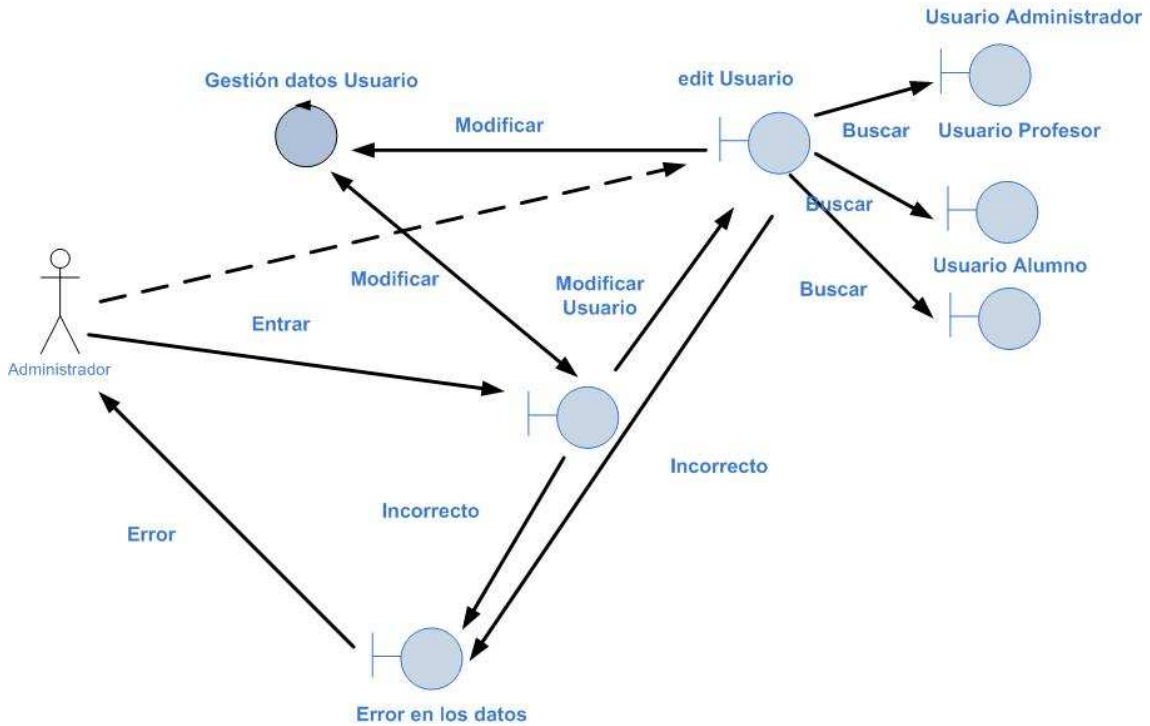
##### 4.3.2.1.1 AddUsuario



**Figura 28: Diagrama de flujo Gestión de Usuarios.**

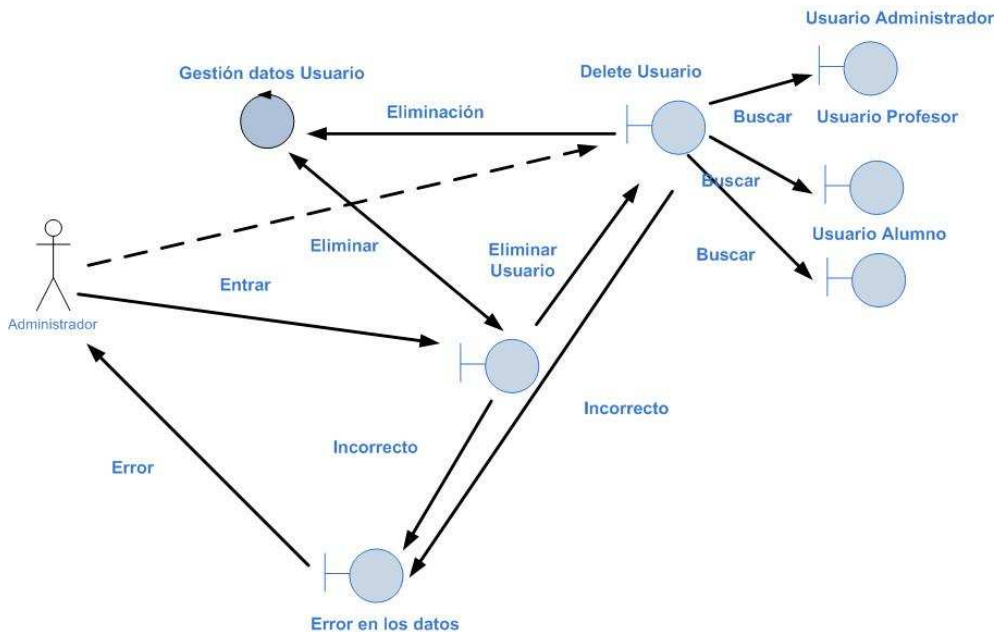
En primer lugar se introducirán los datos comunes a cualquier usuario (nombre, apellidos, dirección, teléfono, mail,...) y una vez introducidos se seleccionará el perfil del usuario, dependiendo del mismo la aplicación mostrará los datos específicos que le corresponden.

4.3.2.1.2 EditUsuario



**Figura 29: Diagrama de Flujo de editUsuario**

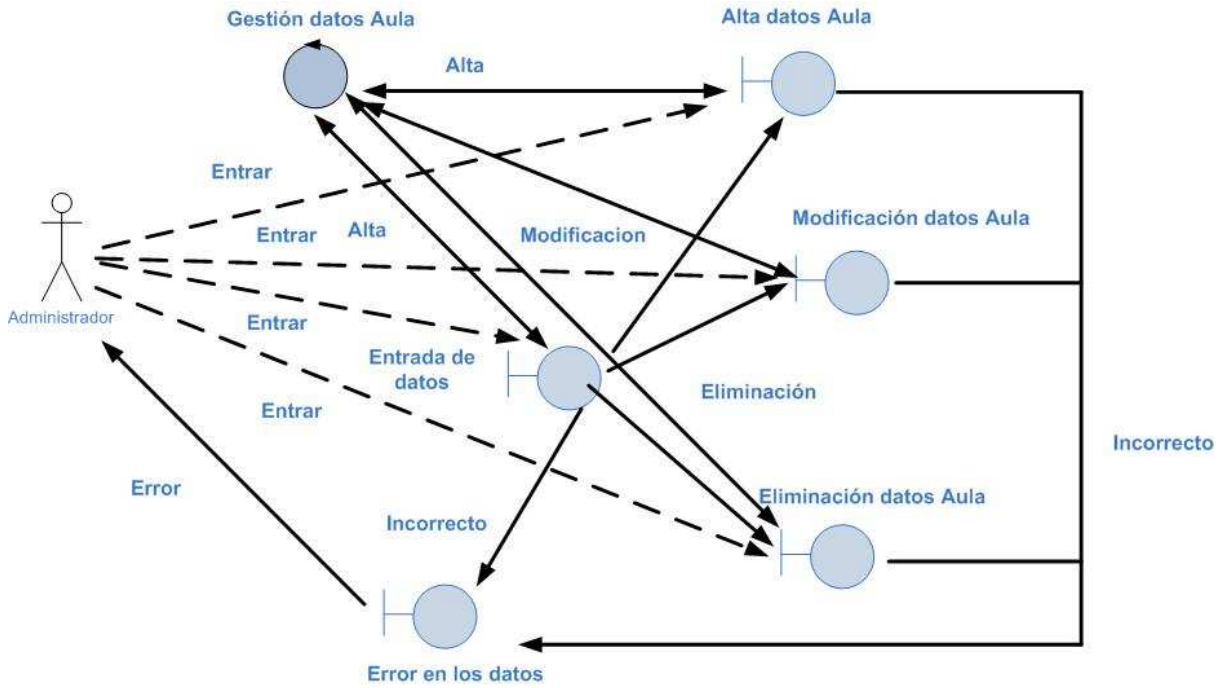
4.3.2.1.3 DeleteUsuario



**Figura 30: Diagrama de flujo de DeleteUsuario**

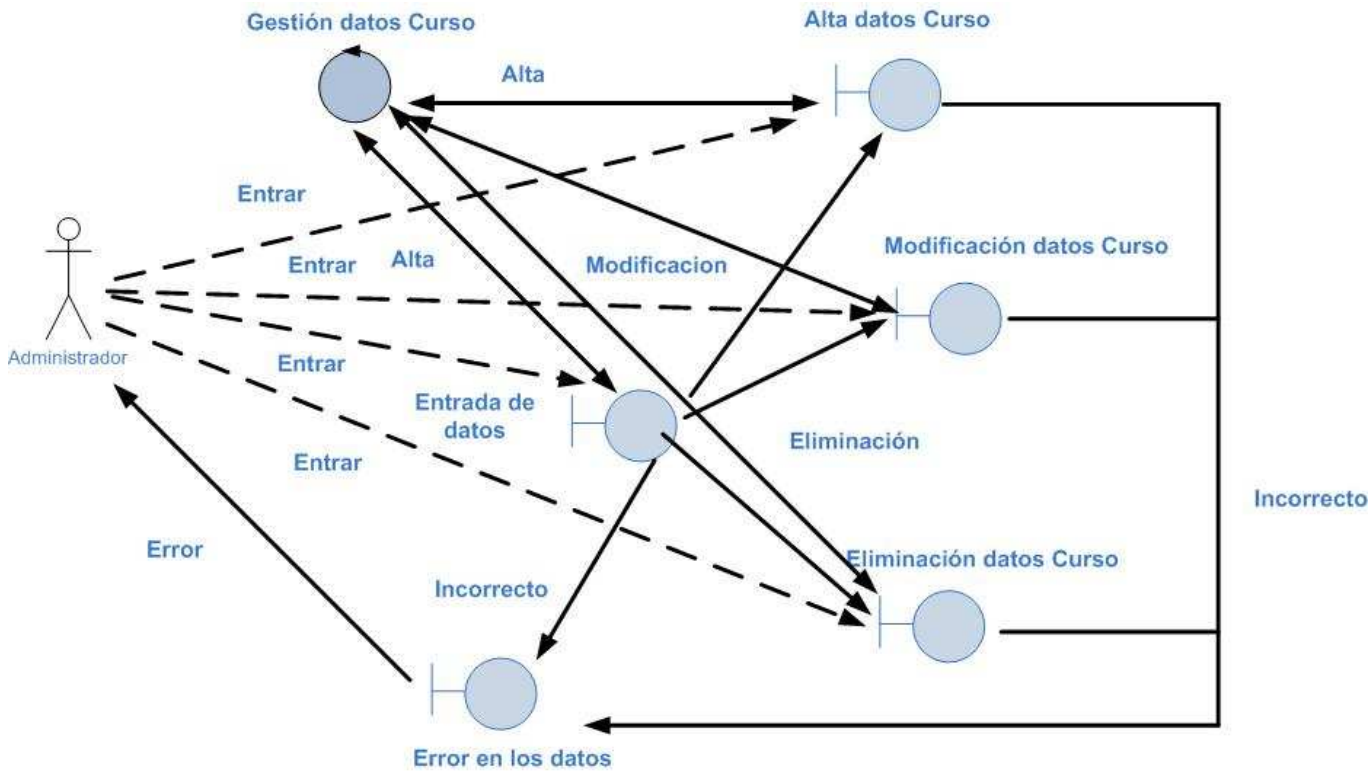
**4.3.3 PAQUETE MANAGER AND EXPLOTATION**

**4.3.3.1 GESTIÓN AULAS**



**Figura 31: Diagrama de flujo GestionAulas**

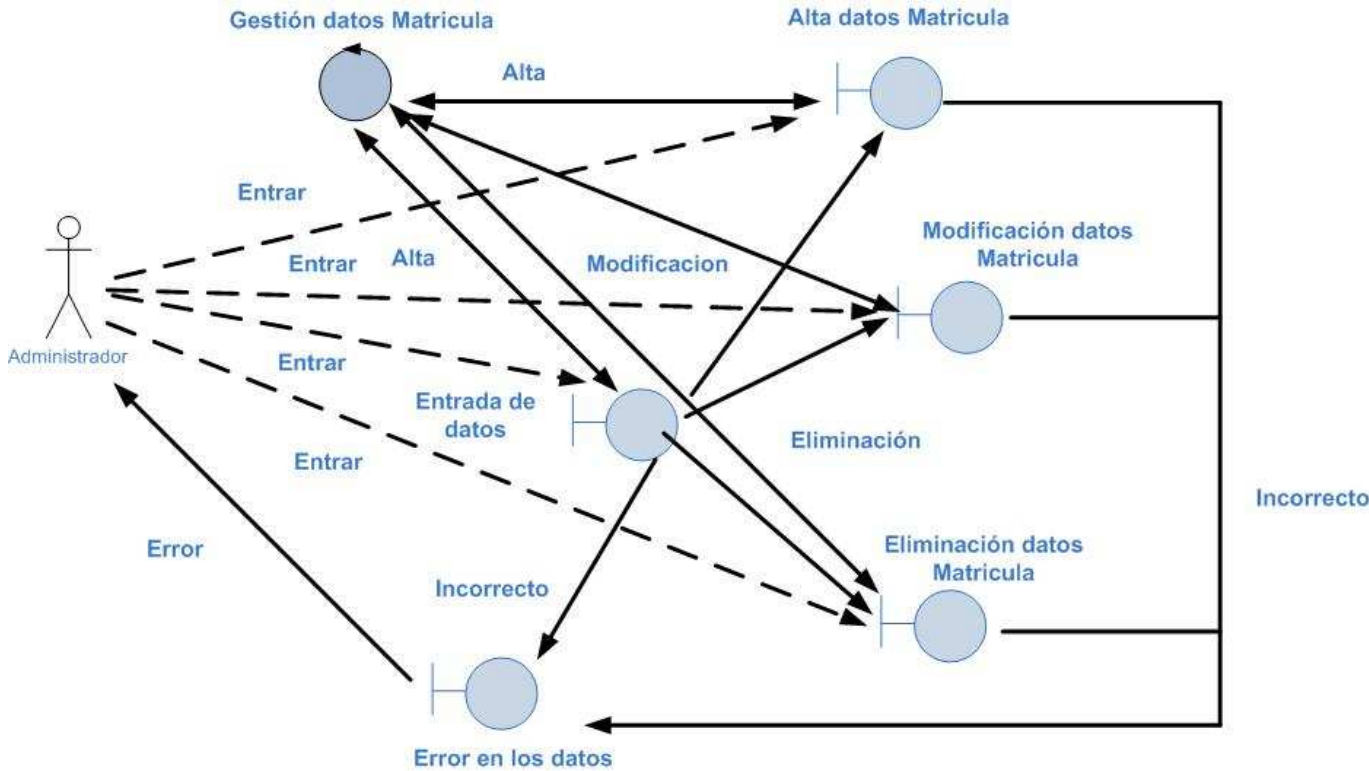
**4.3.3.2 GESTIÓN CURSOS**



**Figura 32: Diagrama de flujo GestiónCursos**

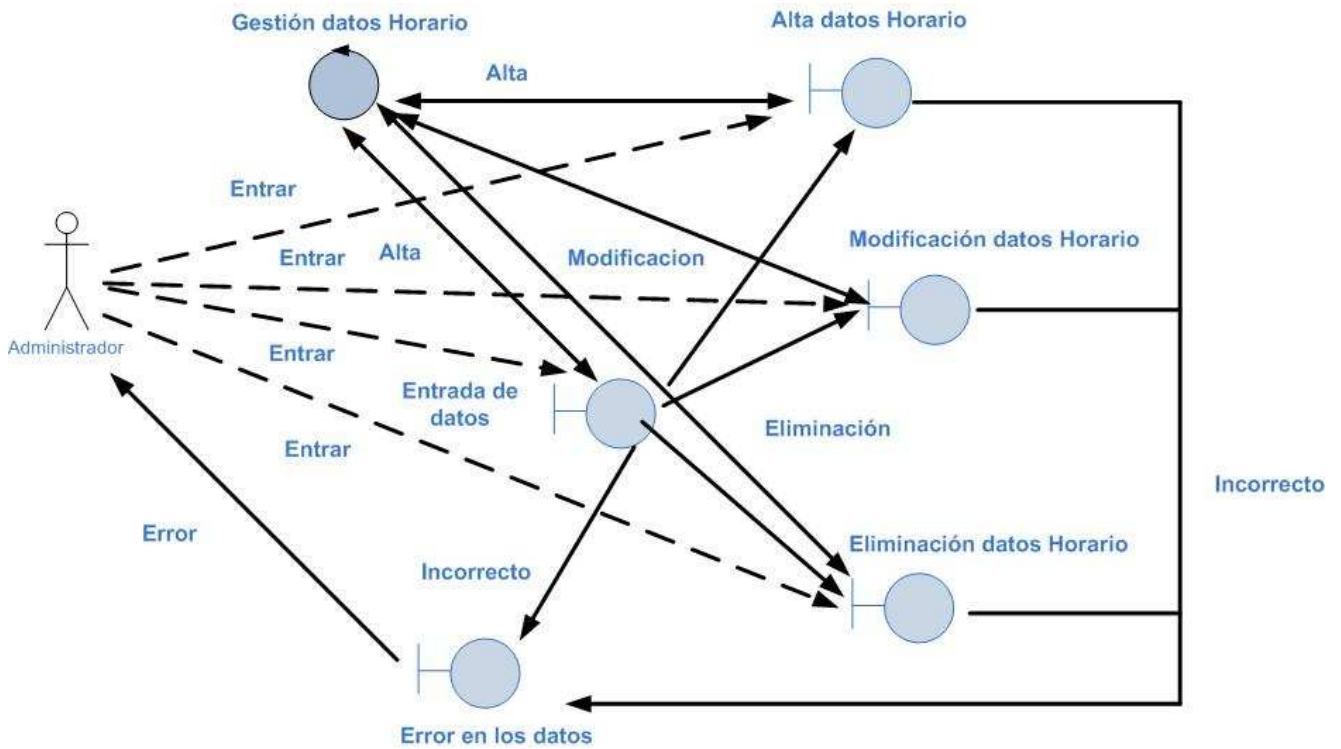


**4.3.3.3 Gestión Matriculas**



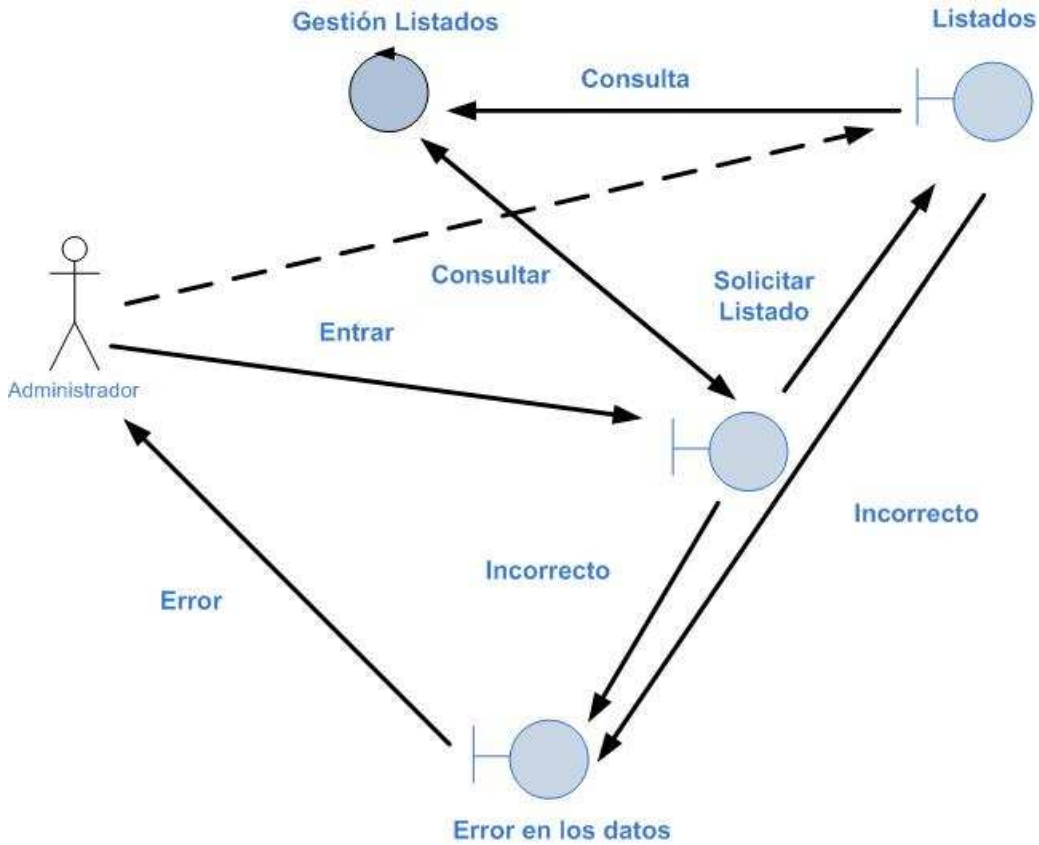
**Figura 33: Diagrama de Flujo GestionMatriculas**

**4.3.3.4 GESTIÓN HORARIOS**



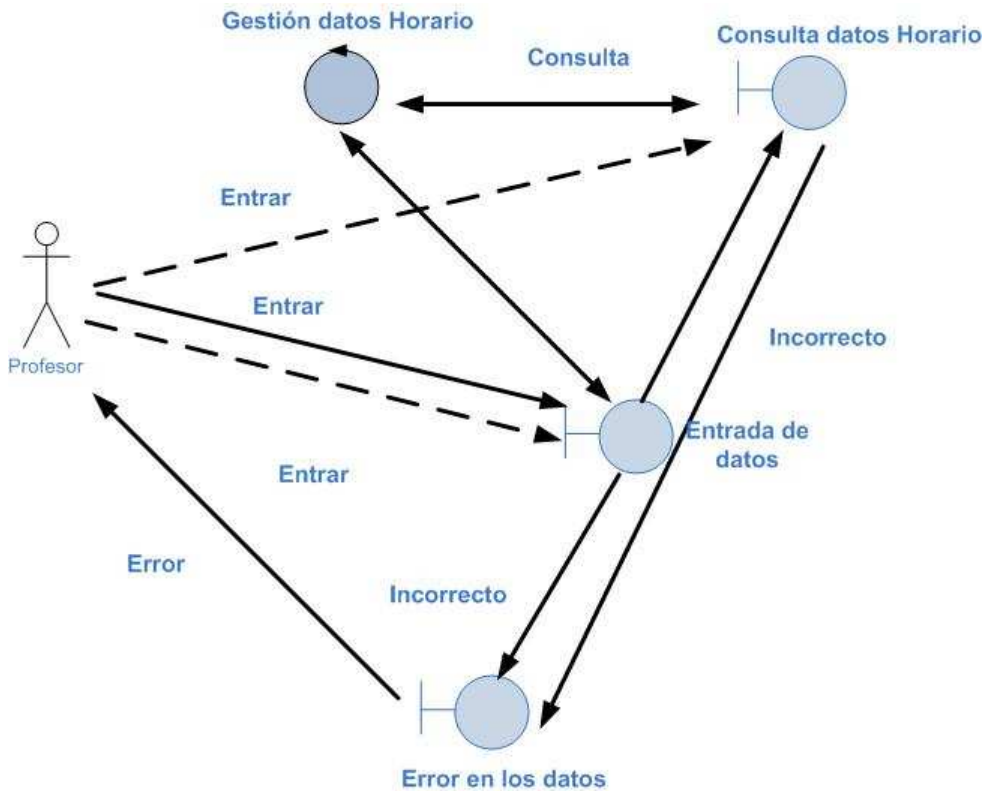
**Figura 34: Diagrama de flujo GestionHorarios**

**4.3.3.5 CONSULTAR LISTADOS**



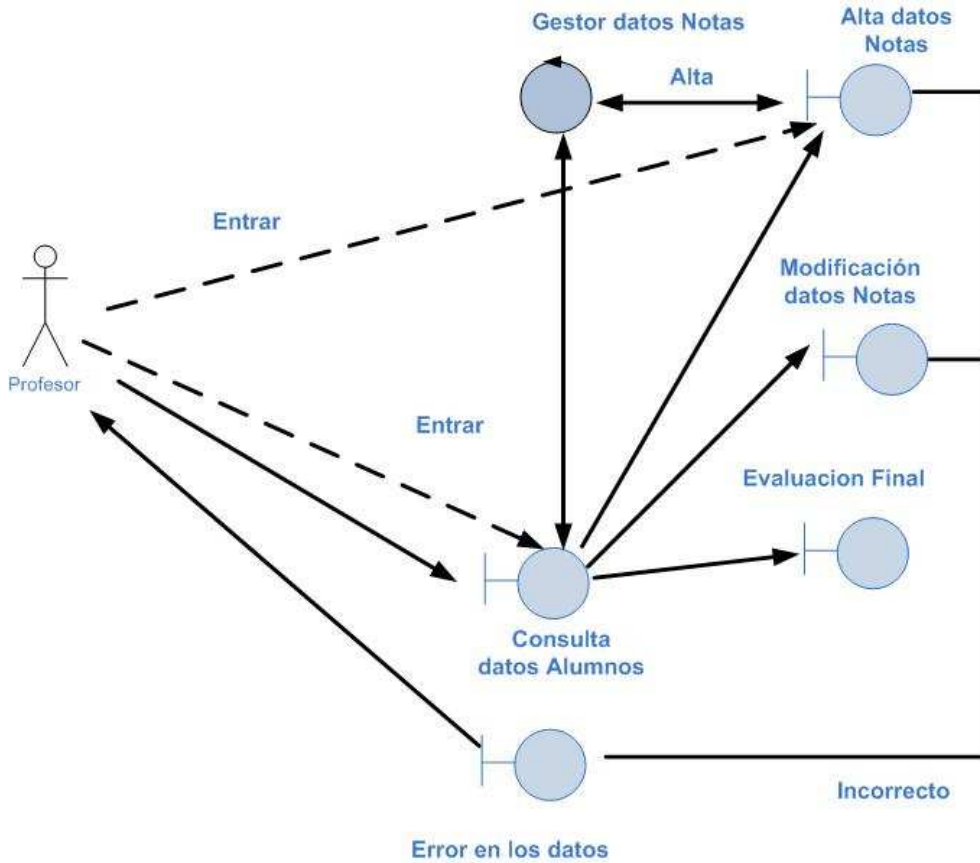
**Figura 35: Diagrama de flujo Consultar Listados**

**4.3.3.6 Consultar Horarios Profesor**



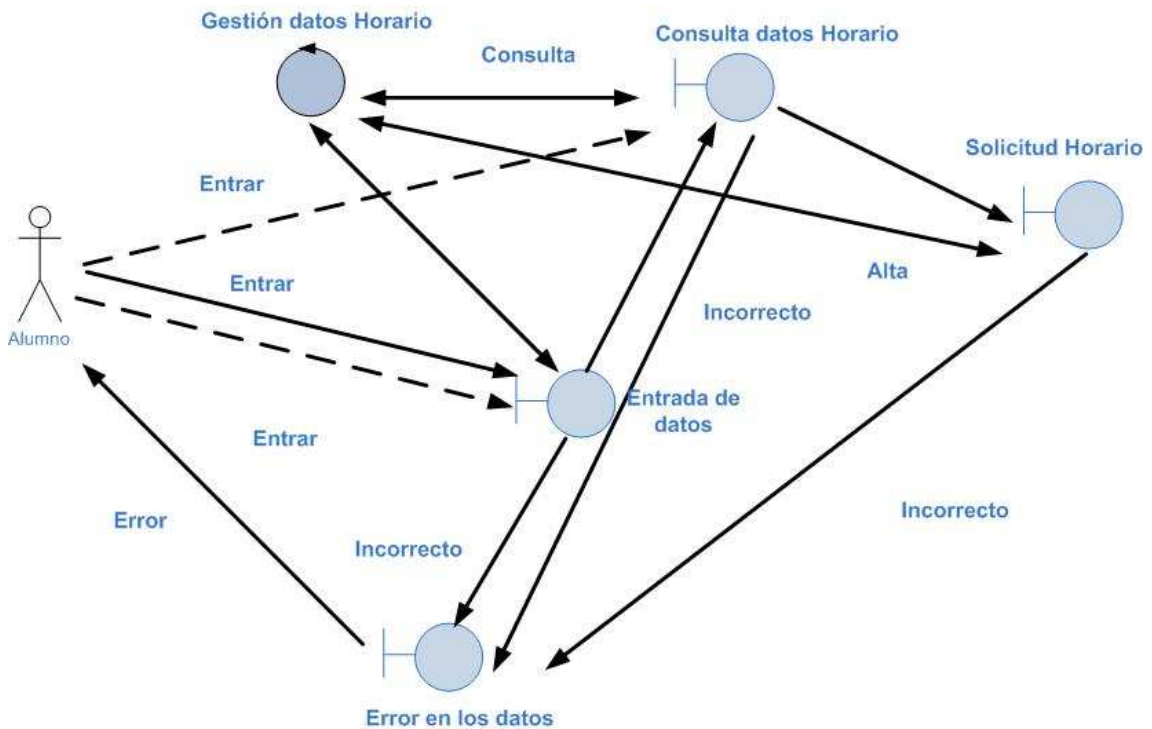
**Figura 36: Diagrama de Flujo Consultar Horarios Profesor**

4.3.3.7 Evaluación Alumno



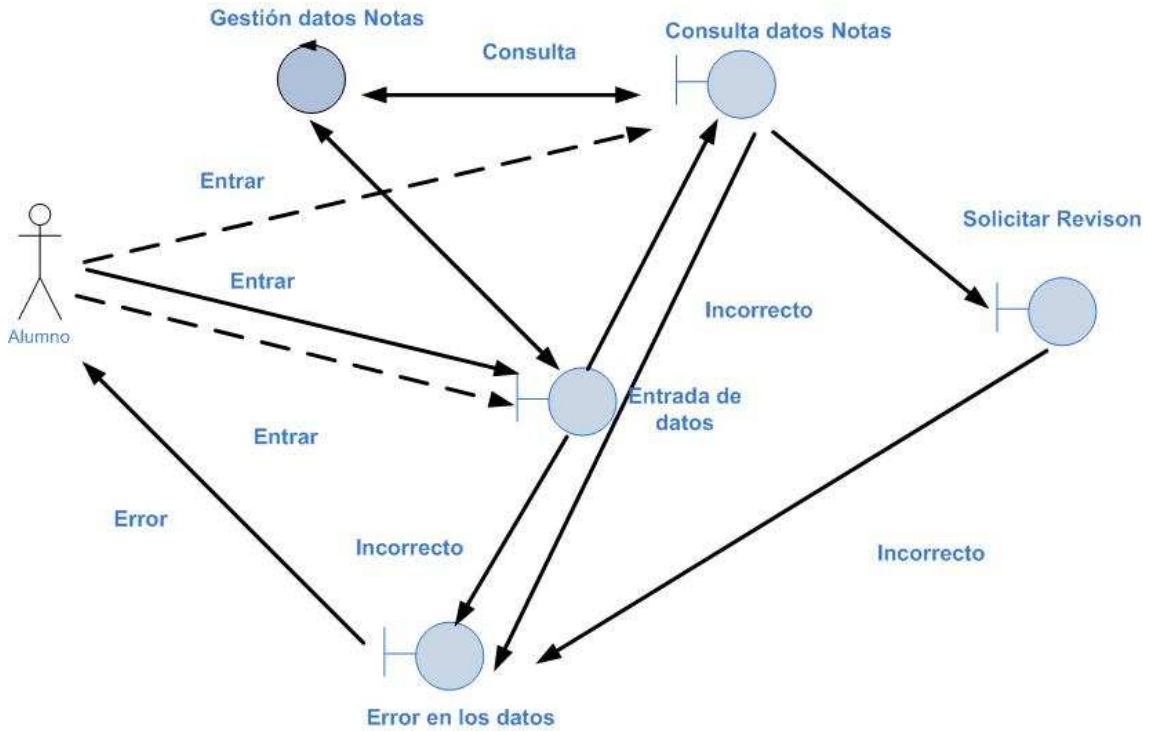
**Figura 37: Diagrama Flujo Evaluación Alumno**

4.3.3.8 Consulta Horarios Alumno



**Figura 38: Diagrama de flujo Consulta Horario Alumno**

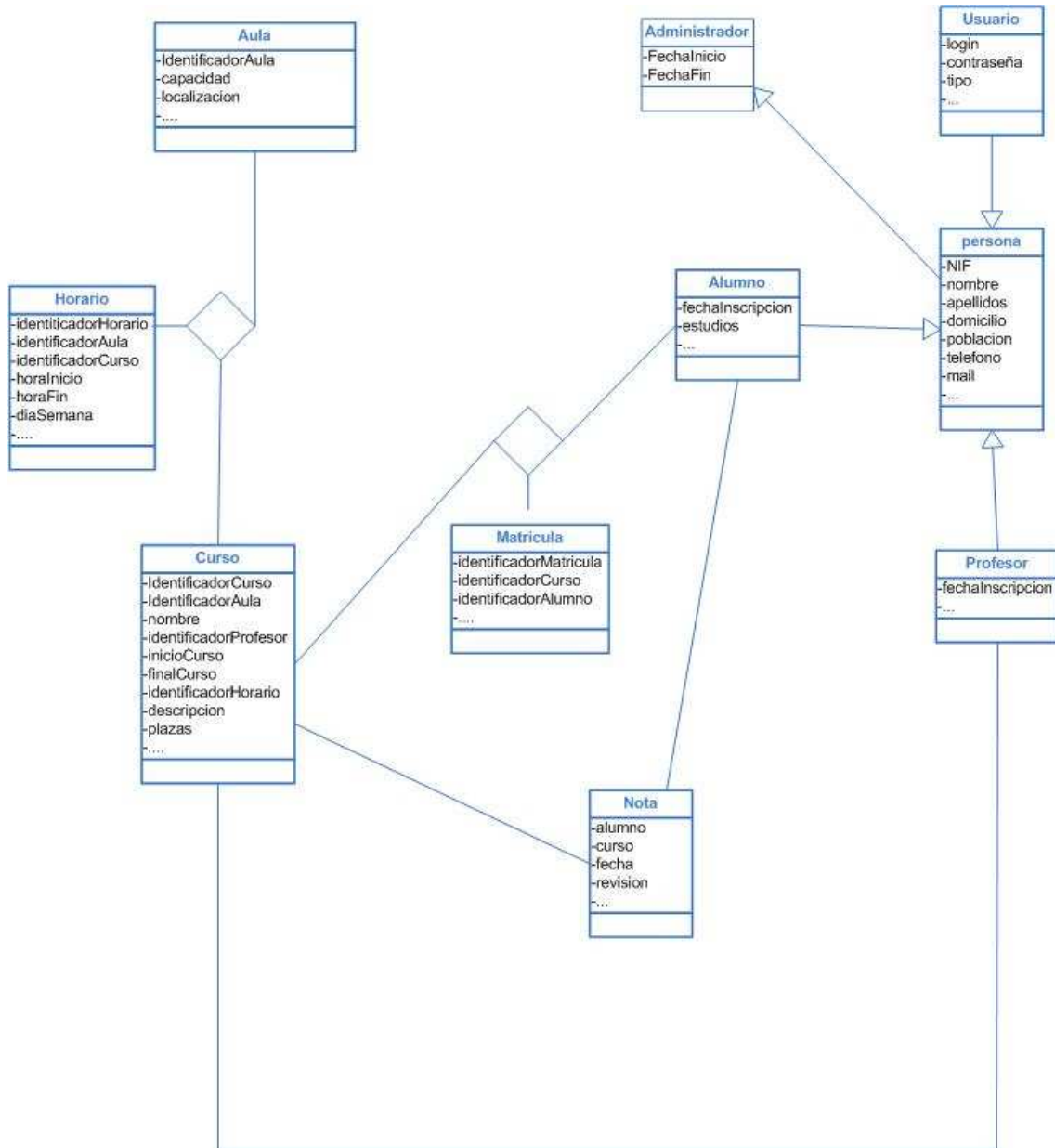
#### 4.3.3.9 Consulta Notas Alumno



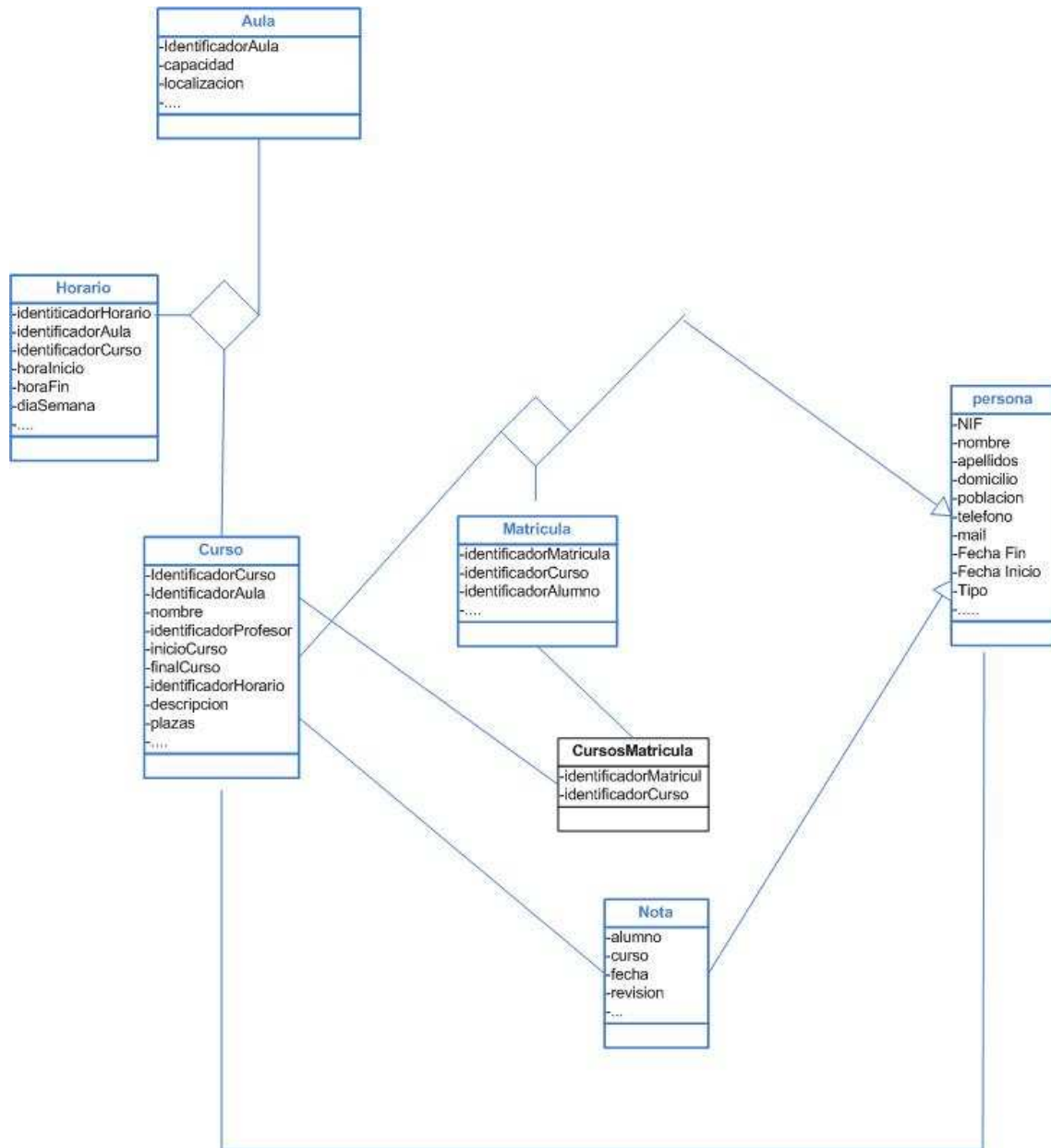
**Figura 39: Diagrama de flujo Consultar Notas por el Alumno**

#### 4.4 DIAGRAMA DE CLASES DEL MODELO CONCEPTUAL

El modelo de conceptual es una representación de los conceptos significativos del modelo de negocio del sistema que participan en los casos de uso, mediante un diagrama de clases, este diagrama ha variado respecto a la versión primera, han desaparecido las clases Administrador, Alumno, Profesor y Usuario quedando solo la clase Persona que reúne todos los atributos generales y también dependiendo del tipo de usuario los específicos de cada uno de los tipos de usuarios, también se ha añadido una nueva clase CursosMatrícula para poder gestionar las matriculas de cada curso , a continuación se muestran ambos diagramas para poder compararlos y ver las diferencias que existen entre ellos.



**Figura 40: Diagrama Clases del modelo conceptual inicial**

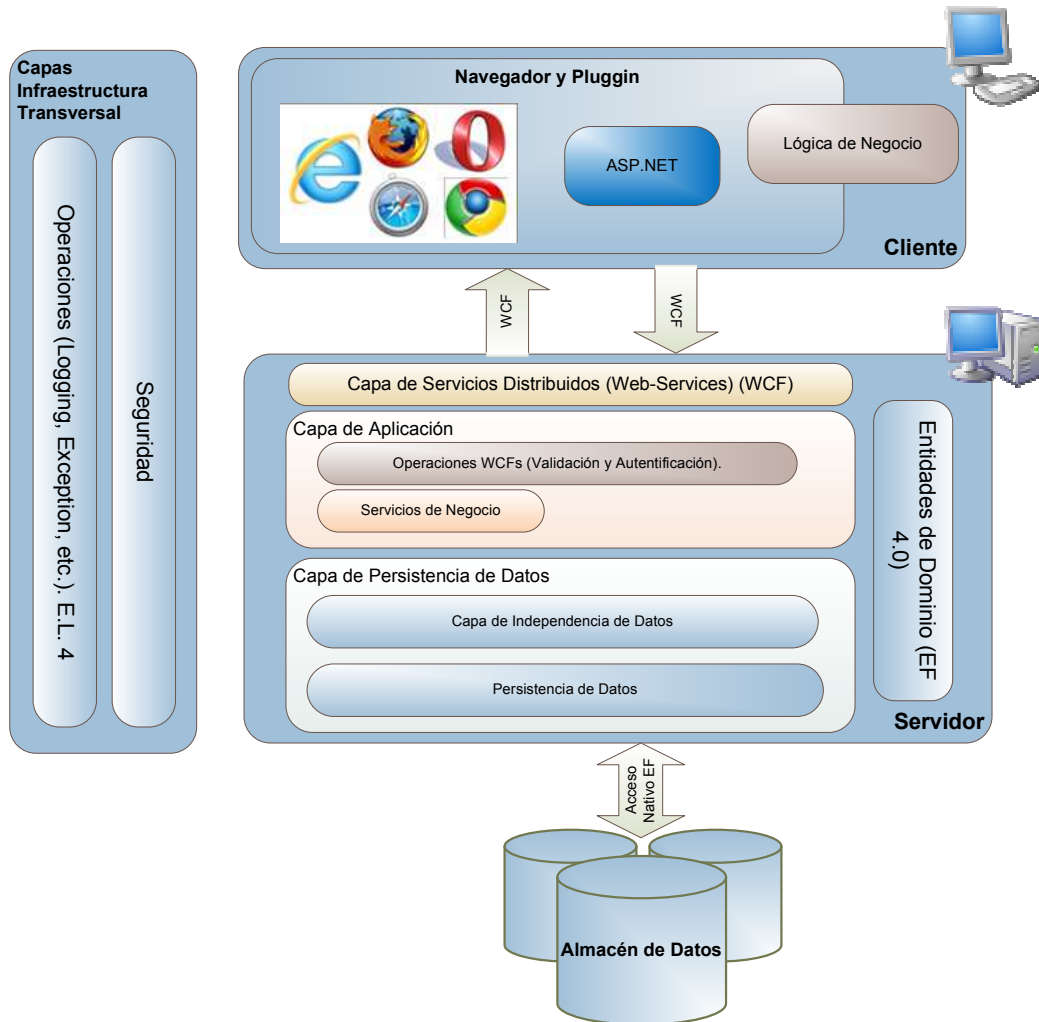


**Figura 41: Diagrama Clases del modelo conceptual final**

## 5 DISEÑO

### 5.1 DIAGRAMA DE LAS ARQUITECTURAS DE SOFTWARE Y HARDWARE

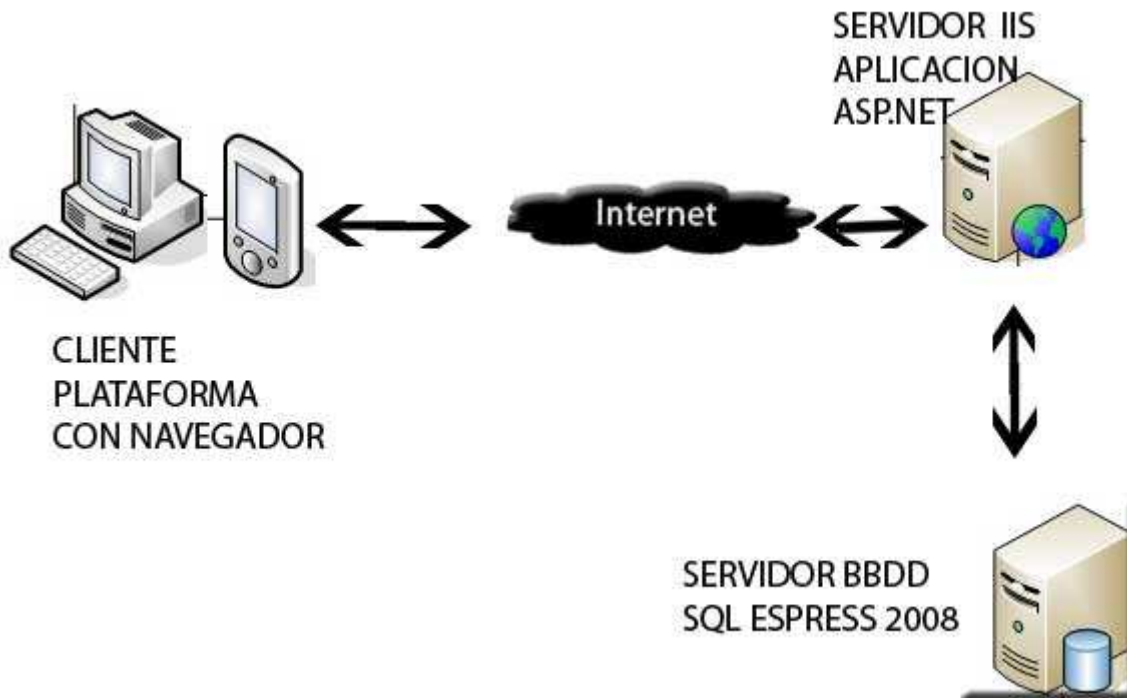
#### 5.1.1 SOFTWARE



**Figura 42: Diagrama arquitectura aplicación.**

#### 5.1.2 HARDWARE

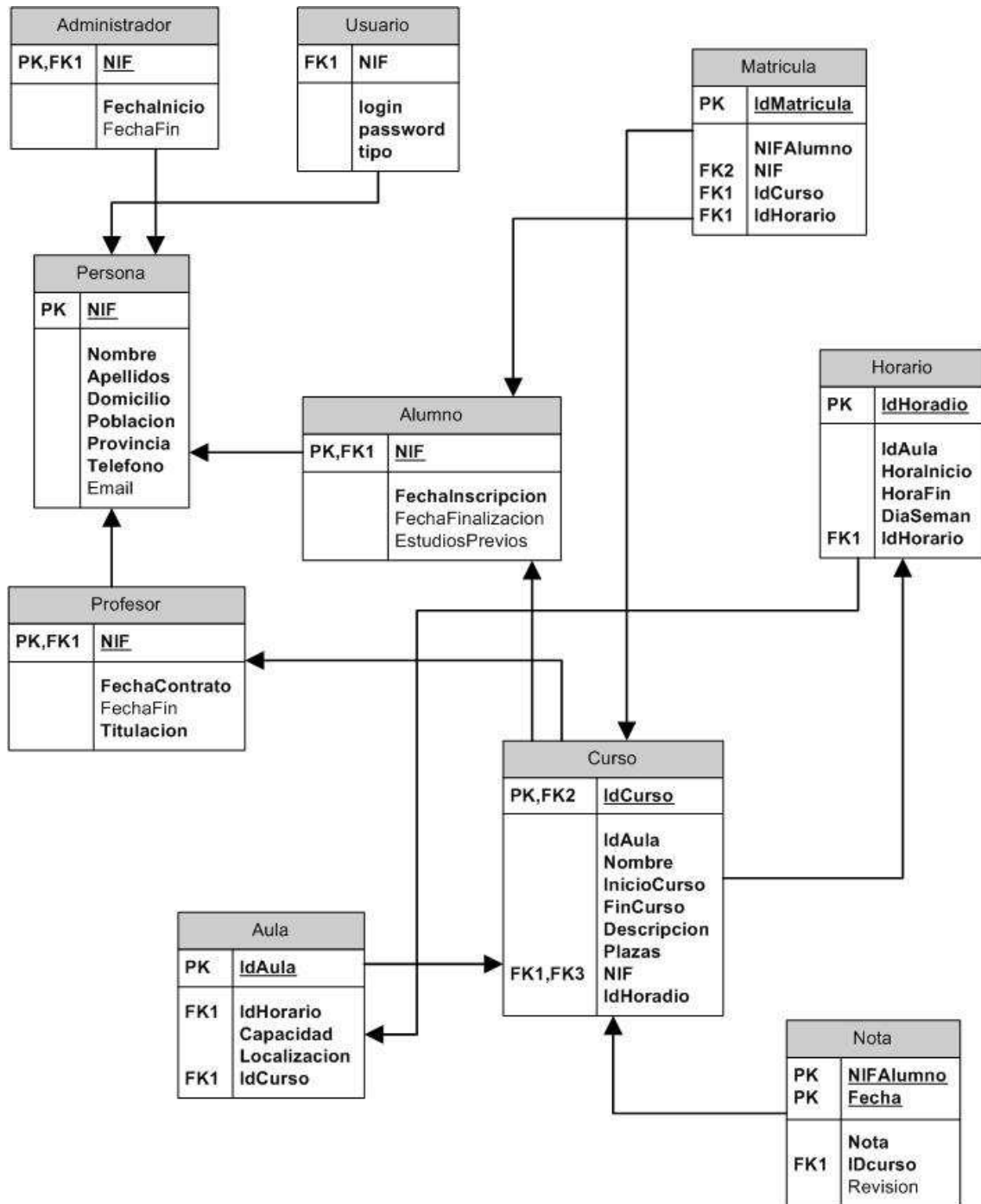
Consistirá en un servidor web con Windows server 2008 y con IIS, asp.net se comunicará con el servidor de BBDD que tendrá también Windows server 2008 con SQL Express 2008, el cliente podrá ser cualquier plataforma que disponga de un navegador y se comunicará vía internet con el servidor web, a continuación se muestra el esquema.



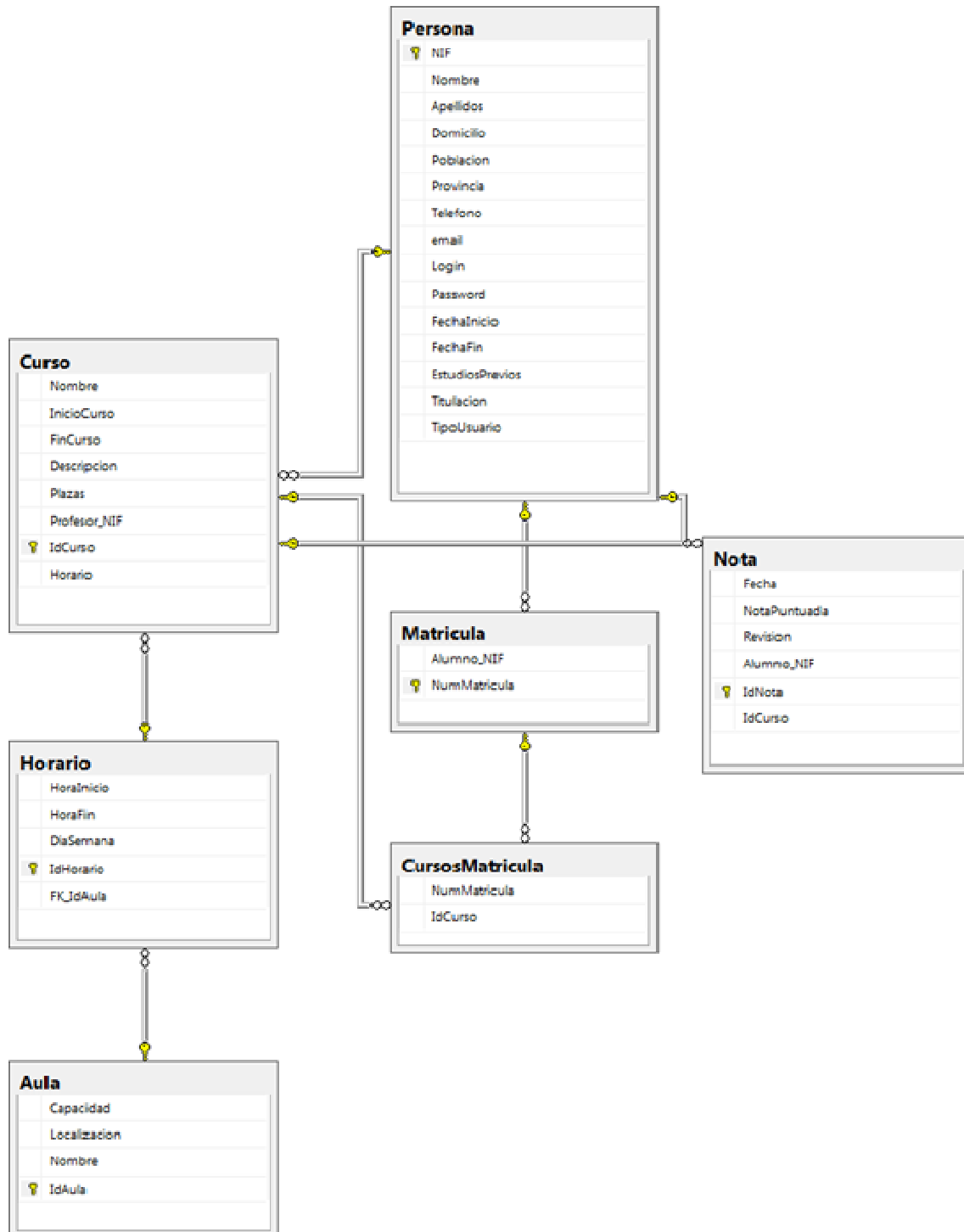
## 5.2 DISEÑO DE LA BASE DE DATOS: MODELO LOGICO

En base al modelo conceptual del apartado 4.4 DIAGRAMA DE CLASES DEL MODELO CONCEPTUAL se crea el siguiente modelo Lógico que será el que se aplicará para la generación de la BBDD de la aplicación, como consecuencia de los cambios realizados en el mismo también ha cambiado el diseño de la BBDD final, a continuación se muestran ambos para poder compararlos y ver las diferencias que existen entre la versión inicial y la final.

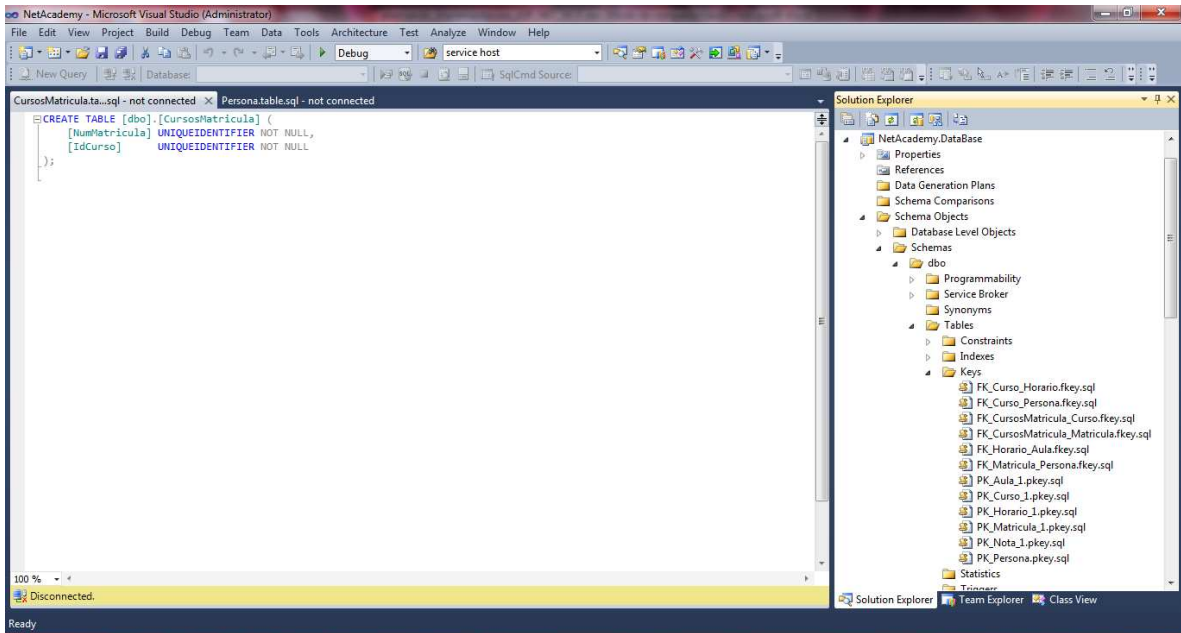




**Figura 43: Modelo Lógico de la BBDD de la Aplicación inicial**



**Figura 44: Modelo Lógico de la BBDD final**



**Figura 45: Detalle de las claves primarias y foráneas de la Base de Datos Final**

### 5.3 INTERFAZ DE USUARIO

A continuación se muestran las pantallas reales de la aplicación. Se presentan en el orden lógico de utilización, por lo tanto se podría utilizar como un posible Manual de Usuario, como se indica en la correspondiente sección de este documento ([ir a manual de usuario](#))

#### 5.3.1 PANTALLA DE INICIO DE LA APLICACIÓN

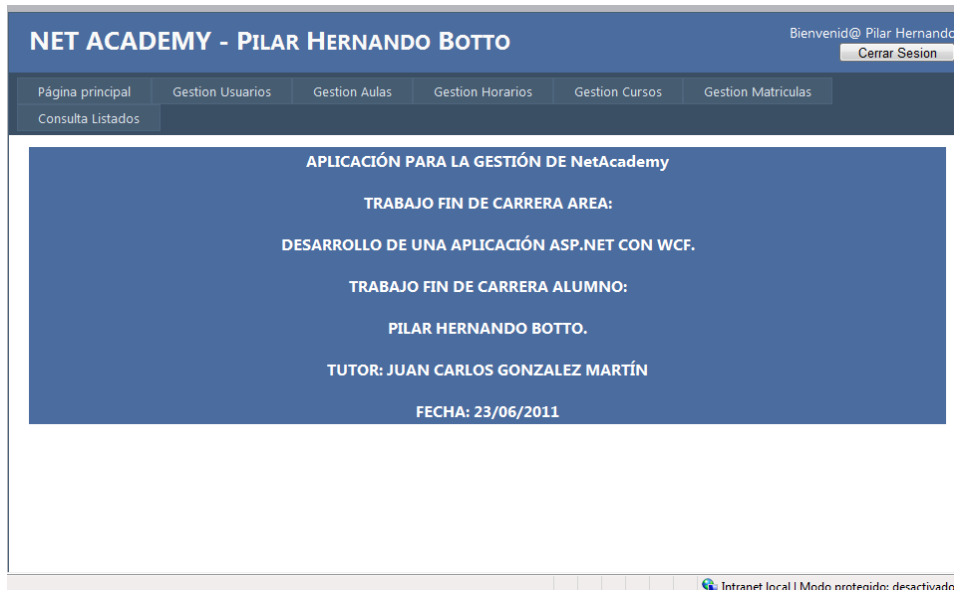
Para poder iniciar la aplicación se debe identificar el usuario. La primera vez que se pone en marcha el programa pedirá al usuario que se identifique, este usuario será el primer Administrador.



**Figura 46: Pantalla de acceso a la aplicación**

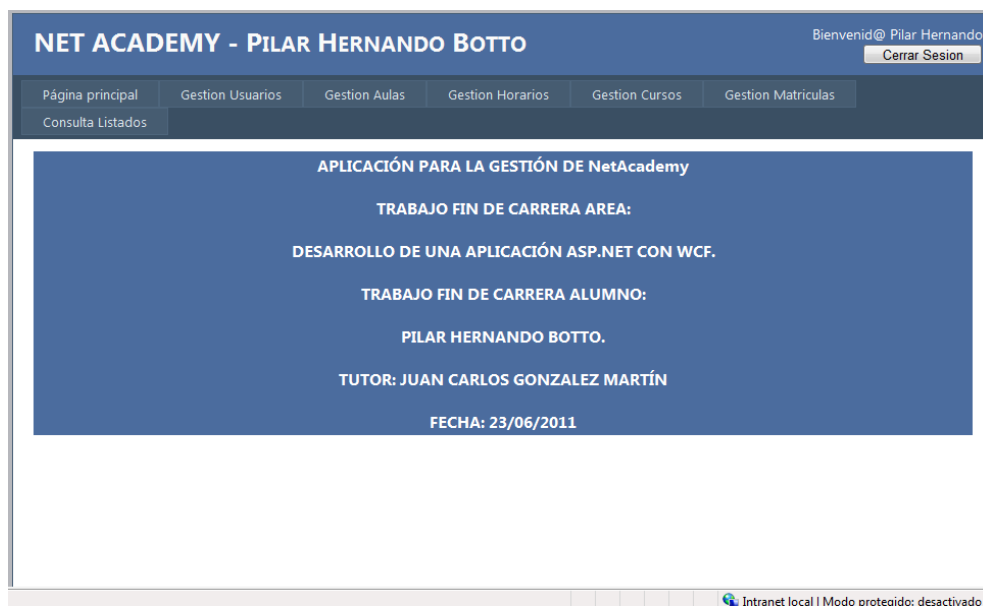
Una vez identificado el usuario, dependiendo de su perfil se mostrará las pantallas de menú correspondientes al Administrador, Profesor o Alumno.

### 5.3.2 Pantallas correspondientes al perfil Administrador.



**Figura 47: Pantalla con las opciones del perfil Administrador.**

Una vez identificado como Administrador se accederá a la pantalla que contiene las pestañas con las funcionalidades correspondientes, bastará con seleccionarla para poder tener acceso a ellas y ejecutarlas. El usuario Administrador debe tener en cuenta que, los campos con los identificadores de las tablas no aparecen ya que se generan de forma automática y son de solo lectura, por lo tanto no se considera necesario que sean visibles, de este modo se evitan posibles errores a la hora de introducir los datos.



**Figura 48: Ventana con las opciones del perfil Administrador de NET-Academy**

El Administrador comenzará el proceso dando de alta a los usuarios, para llevarlo a cabo seleccionará la pestaña Gestión Usuarios que le llevará a la siguiente pantalla y procederá según los pasos que se enumeran a continuación:

NET ACADEMY - PILAR HERNANDO BOTTO Bienvenid@ Pilar Hernando Cerrar Sesión

Página principal Gestion Usuarios Gestion Aulas Gestion Horarios Gestion Cursos Gestion Matriculas Consulta Listados

NOMBRE  TIPO DE USUARIO  Alumno  
 Profesor  
 Administrador

NIF

Buscar

**DATOS PERSONALES DEL USUARIO**

NIF

NOMBRE

APELLIDO

DIRECCION

POBLACION

PROVINCIA

TELEFONO

EMAIL

TIPO DE USUARIO  Alumno  
 Profesor  
 Administrador

FECHA DE ALTA

FECHA DE BAJA

LOGIN

PASSWORD

Limpiar Añadir Modificar Eliminar

**Figura 49: Ventana de la Gestión de Usuarios de NET-Academy**

1. Introducirá los datos personales del usuario.
2. Seleccionará el tipo de usuario, el sistema le mostrará los datos específicos correspondientes al perfil seleccionado para poder introducirlos.
3. Pulsará el botón Añadir, si todo es correcto la aplicación mostrará un mensaje notificándolo e insertará en la tabla el registro correspondiente, de lo contrario saltará la excepción correspondiente.
4. El usuario podrá en cualquier momento pulsar el botón Limpiar si necesita introducir datos nuevos.

**NET ACADEMY - PILAR HERNANDO BOTTO** Bienvenid@ Pilar Hernando [Cerrar Sesión](#)

[Página principal](#) [Gestion Usuarios](#) [Gestion Aulas](#) [Gestion Horarios](#) [Gestion Cursos](#) [Gestion Matriculas](#) [Consulta Listados](#)

NOMBRE:   
 NIF:  TIPO DE USUARIO:  Alumno  Profesor  Administrador

**DATOS PERSONALES DEL USUARIO**

NIF:   
 NOMBRE:   
 APELLIDO:   
 DIRECCION:   
 POBLACION:   
 PROVINCIA:   
 TELEFONO:   
 EMAIL:   
 TIPO DE USUARIO:  Alumno  Profesor  Administrador  
 FECHA DE ALTA:   
 FECHA DE BAJA:   
 LOGIN:   
 PASSWORD:

**DATOS EXTENDIDOS DE PROFESOR**

TITULACIÓN:

**Figura 50: Ventana Gestión de Usuarios, opción Añadir**

Si desea visualizar todos los usuarios registrados, deberá primero seleccionar el tipo de usuario y a continuación pulsar el botón Buscar

**NET ACADEMY - PILAR HERNANDO BOTTO** Bienvenid@ Pilar Hernando [Cerrar Sesión](#)

[Página principal](#) [Gestion Usuarios](#) [Gestion Aulas](#) [Gestion Horarios](#) [Gestion Cursos](#) [Gestion Matriculas](#) [Consulta Listados](#)

**INFORMACION DE USUARIOS**

	Nombre	Apellidos	NIF
<a href="#">Seleccionar</a>	nicolas	Gomez Serrano	12345678b

NOMBRE:   
 NIF:  TIPO DE USUARIO:  Alumno  Profesor  Administrador

**DATOS PERSONALES DEL USUARIO**

NIF:   
 NOMBRE:   
 APELLIDO:   
 DIRECCION:   
 POBLACION:   
 PROVINCIA:   
 TELEFONO:   
 EMAIL:   
 TIPO DE USUARIO:  Alumno  Profesor  Administrador  
 FECHA DE ALTA:   
 FECHA DE BAJA:   
 LOGIN:   
 PASSWORD:   
**DATOS EXTENDIDOS DE PROFESOR**

TITULACIÓN:

**Figura 51: Ventana Gestión de Usuarios opción Buscar**

Si desea visualizar, modificar o eliminar todos los datos de un usuario, deberá de selecciona el usuario dentro de la tabla con los resultados de la búsqueda y a continuación se mostrarán los datos para poder visualizarlos, modificarlos o eliminarlos.

The screenshot shows the 'INFORMACION DE USUARIOS' window. At the top, there are navigation tabs: 'Página principal', 'Gestión Usuarios', 'Gestión Aulas', 'Gestión Horarios', 'Gestión Cursos', 'Gestión Matriculas', and 'Consulta Listados'. The main content area is titled 'INFORMACION DE USUARIOS' and contains a table with the following data:

Seleccionar	Nombre	Apellidos	NIF
<input type="checkbox"/>	nicolas	Gomez Serrano	12345678b

Below the table, there are input fields for 'NOMBRE' and 'NIF', and a 'TIPO DE USUARIO' dropdown menu with options: 'Alumno', 'Profesor', and 'Administrador'. A 'Buscar' button is located below these fields.

The 'DATOS PERSONALES DEL USUARIO' section contains a form with the following fields:

- NIF: 12345678b
- NOMBRE: nicolas
- APELLIDO: Gomez Serrano
- DIRECCION: Ligonde 4
- POBLACION: Madrid
- PROVINCIA: Madrid
- TELEFONO: 91555555
- EMAIL: nico@uoc.edu
- TIPO DE USUARIO: Alumno (selected), Profesor, Administrador
- FECHA DE ALTA: 22/05/2011 0:00:00
- FECHA DE BAJA: 31/12/9999 0:00:00
- LOGIN: nico
- PASSWORD: 1234

The 'DATOS EXTENDIDOS DE PROFESOR' section has a 'TITULACIÓN' field. At the bottom, there are buttons for 'Limpiar', 'Añadir', 'Modificar', and 'Eliminar'.

**Figura 52: Ventana de la Gestión de Usuarios opciones Modificar/Eliminar**

Para acceder a la gestión de las Aulas seleccionará la pestaña Gestión Aulas y se le mostrará la siguiente pantalla, desde la cual podrá Buscar, Añadir, Modificar o Eliminar un Aula. Al realizar cualquiera de estas operaciones, si se hacen de forma correcta, el sistema mostrará un mensaje indicándoselo al usuario, si se produce un error saltará la correspondiente excepción.

The screenshot shows the 'GESTION AULAS' window. At the top, there are navigation tabs: 'Página principal', 'Gestion Usuarios', 'Gestion Aulas', 'Gestion Horarios', 'Gestion Cursos', and 'Gestion Matriculas'. The main content area is titled 'NET ACADEMY - PILAR HERNANDO BOTTO' and includes a 'Bienvenid@ Pilar Hernando' message and a 'Cerrar Sesion' button.

Below the navigation tabs, there is a search bar with the following fields:

Nombre	Capacidad	Localización
<input type="text"/>	<input type="text"/>	<input type="text"/>

A 'Buscar' button is located below the search bar.

The 'GESTION AULAS' section contains a form with the following fields:

- Nombre: Aula1
- Capacidad: 15
- Localización: Planta 2

At the bottom, there are buttons for 'Añadir Aula', 'Eliminar', and 'Modificar'.

**Figura 53: Ventana de Gestión de Aulas**

Para visualizar todas las aulas o las que se correspondan con las opciones de búsqueda: Nombre, capacidad o localización se procederá de igual forma que con los usuarios. Se seguirá también el mismo proceso para Eliminar o Modificar.

**Figura 54: Ventana Gestión de Aulas opción Buscar**

Para acceder a la gestión de los Horarios se seleccionará la pestaña Gestión de Horarios, en este caso se muestra un desplegable con la lista de todas las Aulas para que se pueda seleccionar una, por lo demás su utilización es semejante a la de Gestión Aulas, también aquí si todo es correcto se le mostrará al usuario el mensaje correspondiente de conformidad y si no lo es saltará la excepción.

**Figura 55: Ventana Gestión de Horarios**

La gestión de los cursos se realizará desde la pestaña Gestión Cursos, tanto los profesores como los horarios se seleccionan desde una lista de opciones. Al igual que en las ventanas anteriores se mostrará el mensaje de éxito de la operación o se capturará la excepción.



**NET ACADEMY - PILAR HERNANDO BOTTO** Bienvenid@ Pilar Hernando [Cerrar Sesión](#)

[Página principal](#) [Gestion Usuarios](#) [Gestion Aulas](#) [Gestion Horarios](#) [Gestion Cursos](#) [Gestion Matriculas](#)  
[Consulta Listados](#)

Nombre Curso	Profesor	Inicio curso	Número de plazas
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

**DATOS DE LOS CURSOS**

Nombre Curso:

Identificador Profesor:

Identificador Horario:

Inicio Curso:

Fin Curso:

Número de plazas:

Descripción:

**Figura 56: Ventana Gestión de Cursos**

La opción Buscar mostrará todos los cursos que están dados de alta en la Base de Datos, si se introducen uno o varios de los criterios de búsqueda, mostrará los que cumplen con dichos criterios. Una vez mostrado el resultado al seleccionar uno de los resultados se mostrarán los datos correspondientes, de ese modo se podrá si es necesario Modificarlos o Eliminarlos, mostrando el correspondiente mensaje de éxito o capturando la excepción.

**NET ACADEMY - PILAR HERNANDO BOTTO** Bienvenid@ Pilar Hernando [Cerrar Sesión](#)

[Página principal](#) [Gestion Usuarios](#) [Gestion Aulas](#) [Gestion Horarios](#) [Gestion Cursos](#) [Gestion Matriculas](#)  
[Consulta Listados](#)

**INFORMACION CURSOS**

	Nombre Curso	Profesor	Inicio Curso	Numero plazas
<a href="#">Seleccionar</a>	IA	nicolas Gomez Serrano	01/01/2011	12

Nombre Curso	Profesor	Inicio curso	Número de plazas
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

**DATOS DE LOS CURSOS**

Nombre Curso:

Identificador Profesor:

Identificador Horario:

Inicio Curso:

Fin Curso:

Número de plazas:

Descripción:

**Figura 57: Ventana Gestión de Cursos opción Buscar, Modificar, Eliminar**

Para gestionar las matriculas se seleccionará la pestaña GestiónMatriculas, en la misma aparecerá un listado con los alumnos y los cursos dados de alta para que se seleccione el alumno y lo o los cursos correspondientes en los que se va a

matricular, una vez seleccionados se pulsará el botón Añadir para incorporar los datos a la Base de datos. Si se necesita Buscar la matrícula de un alumno se introducirá su NIF y se pulsará el botón Buscar, si no se introduce ningún NIF el botón Buscar mostrará la lista de todos los alumnos, al seleccionar uno se mostrarán en la parte inferior sus correspondientes matriculas. También se podrán Modificar o Eliminar los datos de una matrícula.



**Figura 58: Ventana Gestión de Matriculas**

Para acceder a consultar los listados se deberá seleccionar la pestaña Consultar Listados, desde esta pantalla se deberá seleccionar el tipo de listado a consultar y pulsar el botón Consultar.



Nombre	Descripción	Fecha de Inicio	Plazas
IA	Inteligencia Artificial	01/01/2011 0:00:00	12

**Figura 59: Ventana Gestión de Listados perfil Administrador**

### 5.3.3 Pantallas correspondientes al perfil Profesor

Si el perfil de usuario es Profesor, la aplicación le mostrará la siguiente pantalla con las opciones predeterminadas al mismo. Desde la pestaña Cambio Password podrá modificar el password original asignado por el Administrador al darle de alta, desde la pestaña Consultar Horarios podrá visualizar los horarios correspondientes a un Curso y desde la pestaña Evaluar Alumno podrá introducir las notas correspondientes al curso de un Alumno. La forma de actuar con ellas es semejante a la de las ventanas del perfil Administrador.



**Figura 60: Ventana inicial del perfil Profesor**



**Figura 61: Ventana Modificar Password Profesor**

**NET ACADEMY - PILAR HERNANDO BOTTO** Bienvenid@ nicolas Gomez Serrano [Cerrar Sesión](#)

[Página principal](#) [Cambio Password](#) [Consultar Horarios](#) [Evaluar Alumno](#)

**LISTADO DE HORARIOS**

Aula	Hora de Inicio	Día de la semana
Aula1	10:00:00	Lunes

Curso  [Buscar](#)

**Figura 62: Ventana Evaluar Alumno perfil Profesor**

**NET ACADEMY - PILAR HERNANDO BOTTO** Bienvenid@ nicolas Gomez Serrano [Cerrar Sesión](#)

[Página principal](#) [Cambio Password](#) [Consultar Horarios](#) [Evaluar Alumno](#)

**DETALLE NOTAS ALUMNOS**

	Alumno	Nota	Fecha	Curso
<a href="#">Seleccionar</a>	Telmo Rodriguez	7	15/06/2011 0:00:00	IA
<a href="#">Seleccionar</a>	Telmo Rodriguez	9	23/05/2011 0:00:00	IA
<a href="#">Seleccionar</a>	Telmo Rodriguez	6	12/06/2011 0:00:00	IA

Alumno  Curso  Fecha

[Buscar](#)

**EVALUAR ALUMNO**

Alumno

Nombre Curso

Nota

Fecha

Evaluacion Final

[Añadir](#) [Modificar](#) [Eliminar](#)

**Figura 63: Ventana Evaluar Alumno perfil Profesor opciones Buscar/Modificar/Eliminar**

### 5.3.4 Pantallas correspondientes al perfil Alumno

Si el perfil es el de usuario Alumno se abrirá la siguiente ventana desde la que podrá ir a la Página principal, Cambiar el Password asignado por el Administrador al darle de alta, Consultar Horarios y Consultar Notas. La forma de interactuar es igual que el resto de las ventanas de los otros perfiles de usuario.



**Figura 64: Ventana de inicio de NET-Academy para el perfil de Alumno**



**Figura 65: Ventana de búsqueda de Horarios por Curso perfil Alumno**



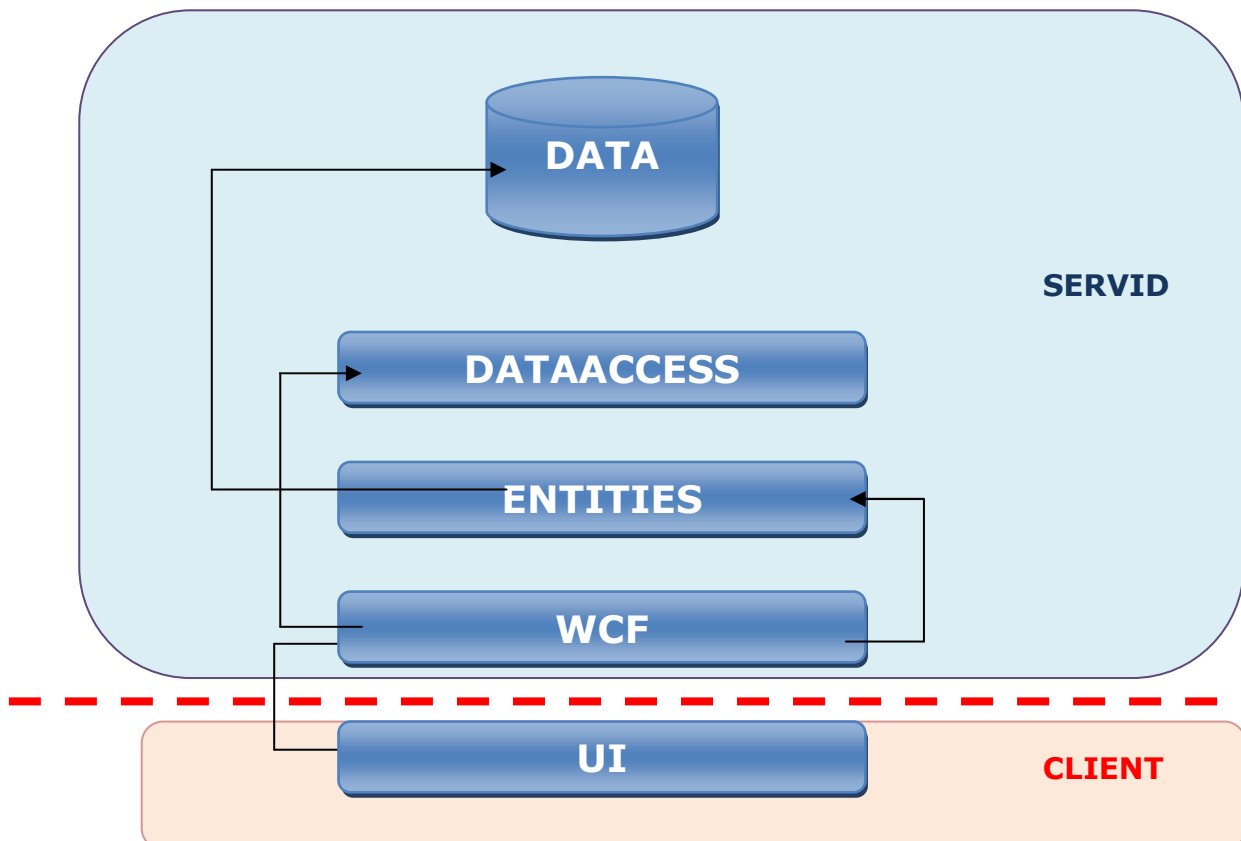
**Figura 66: Ventana de Consulta notas del Alumno por curso**

## 6 IMPLEMENTACIÓN

La implementación se ha realizado con Microsoft Visual Estudio 2010 (Ultimate), debido a las limitaciones de la versión profesional a la hora de manejar proyectos de base de datos.

### 6.1 CAPAS DE LA APLICACIÓN

La aplicación consta de las siguientes capas, cada una de las cuales constituye un proyecto dentro de la solución NetAcademy.



### 6.1.1 Capa UI: Proyecto NetAcademy.WebApplication .

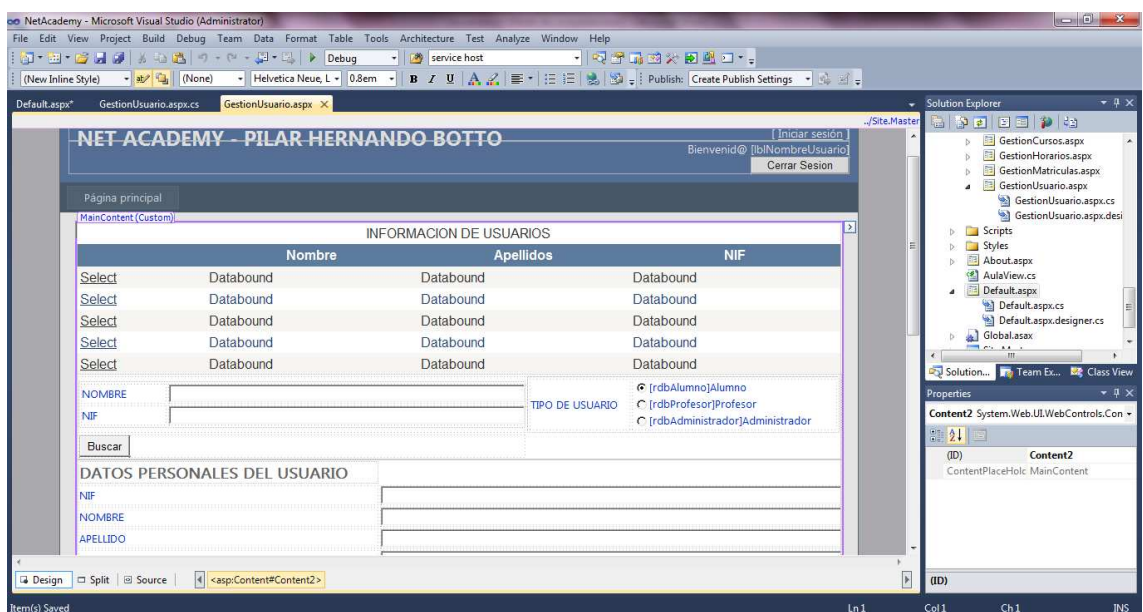
En esta capa contiene la Interfaz de Usuario de la aplicación, las páginas constan de un archivo con el código .aspx que es el que ve el usuario (código HTML) y otro .aspx.cs con el código .net que es el encargado de realizar las validaciones y las llamadas al servicio.

La página de inicio de la misma es Default.aspx, el resto está organizado en tres carpetas, Account, Gestión y Consultas. Todas las paginas están contenidas en una Master site que proporciona el marco de ejecución a toda la aplicación. Dicha Master site es la encargada de mostrar los acceso a las distintas opciones de la aplicación, organizadas en función del rol del usuario logado en la misma. Dentro de las carpetas, en la primera está la página que se muestra al usuario cuando introduce los datos erróneos desde la página de inicio, login.aspx y la página ModificarPassword.aspx que permite el cambio de password. En la carpeta Gestión se encuentran todas las páginas que permiten acceder a ejecutar las funcionalidades de gestión de la aplicación, tanto las que realiza el Administrador (Gestión Usuarios, Aulas, Cursos, Horarios y Matrículas) como la que realiza el Profesor (Evaluar Alumno). Por último en la carpeta Consulta se encuentran las páginas para poder acceder a consultar los Horarios (Administrador, Alumno y Profesor), los Listados (Administrador) y las Notas de los Alumnos (Profesor y Alumno). La página Default.aspx es la encargada de comprobar si el login y password del usuario son correctos, si lo son, dependiendo de su tipo le mostrará la ventana correspondiente con las diferentes funcionalidades a las que tiene acceso, si no lo son, le mostrará un mensaje de error y le devolverá a la ventana de login.aspx, para que vuelva a identificarse.

Para poder consumir los servicios de WCF se necesita añadir a la carpeta Service References el servicio ServiceNetAcademy.

Como la estructura de todas ellas es semejante, para explicar su funcionamiento se ha seleccionado GestionUsuario, por ser una de las que reúne prácticamente todos los tipos de componentes que se utilizan.

#### 6.1.1.1 Componentes de GestionUsuario.



**Figura 68: Diseño de la Ventana Gestión de Usuario**

- A. La cabecera de la ventana muestra a la izquierda el nombre de la aplicación, a la derecha muestra la opción de Iniciar o Cerrar Sesión dependiendo de la situación actual, en el caso de cerrar sesión nos devolvería a la página de inicio de la aplicación. también muestra un mensaje de bienvenida con el nombre del usuario actual
- B. Dependiendo de la ventana, se mostrarán tantas pestañas como distintas funcionalidades a las que tenga acceso el usuario actual, pudiendo siempre volver a la Página principal
- C. Un `GridView` mostrará los resultados de las búsquedas, en este caso, los datos del usuario/os que coincidan en todo o en parte con los criterios, si no se introduce ningún criterio se mostrarán los datos de todos los usuarios del tipo seleccionado en los `RadioButton`. Para poder cargar el contenido se llama al método `Page_Load(object sender, EventArgs e)` de la clase

```
protected void Page_Load(object sender, EventArgs e)
{
    table.Columns.Add("Nombre");
    table.Columns.Add("Apellidos");
    table.Columns.Add("NIF");

    GridView1.DataKeyNames = new string[] { "NIF" };
}
```

- D. El botón Buscar es el que contiene el evento `onclick="btnBuscar_Click"`, que llama al método del mismo nombre que se encuentra en la clase `public partial class GestionUsuario : System.Web.UI.Page` que implementa la interfaz, este método es el encargado de obtener los usuarios según el tipo seleccionado y de llenar la tabla del `GridView`

```
protected void btnBuscar_Click(object sender, EventArgs e)
{
    TipoUsuario _nivel = TipoUsuario.Alumno;

    if (this.rdbAdministrador.Checked) _nivel =
TipoUsuario.Administrador;
    if (this.rdbAlumno.Checked) _nivel = TipoUsuario.Alumno;
    if (this.rdbProfesor.Checked) _nivel = TipoUsuario.Profesor;

    usuarios = gestionUsuarios.ObtenerUsuarios(this.txtBuscarNIF.Text,
this.txtBuscarNombre.Text, _nivel);
    table.Clear();
    foreach (Persona p in usuarios)
        table.Rows.Add(new object[] { p.Nombre , p.Apellidos, p.NIF});
    ClearFields();
    GridView1.DataSource = table;
    GridView1.DataBind();
}
```

- E. Cuando se selecciona uno de los resultados de la búsqueda , se mostrarán los datos en los cuadros de texto , llamando al método `xxxx_SelectedIndexChanged(object sender, EventArgs e)` de la clase correspondiente, donde `xxxx` es el nombre que se le ha dado al `GridView`, este método efectúa una consulta a la base de datos pasando como parámetro , el que lee del `GridView` y escribiendo los resultados en los correspondientes `TextBox`

```
protected void GridView1_SelectedIndexChanged(object sender, EventArgs e)
```



```

{
    Persona p = (from d in usuarios
                where d.NIF ==
(string)this.GridView1.SelectedValue
                select d).First();

    this.txtNombreUsuario.Text = p.Nombre;
    this.txtApellidoUsuario.Text = p.Apellidos;
    this.txtNIFUsuario.Text = p.NIF;
    this.txtProvinciaUsuario.Text = p.Provincia;
    this.txtEmailUsuario.Text = p.Email;
    this.txtDireccionUsuario.Text = p.Domicilio;
    this.txtLoginUsuario.Text = p.Login;
    this.txtPassUsuario.Text = p.Password;
    this.txtPoblacionUsuario.Text = p.Poblacion;
    this.txtTelefonoUsuario.Text = p.Telefono;
    this.txtFechaAlta.Text = p.FechaInicio.ToString();
    this.txtFechaBaja.Text = p.FechaFin.ToString();

    if (p.TipoUsuario == TipoUsuario.Administrador)
    {
        this.rbtTipoAdministrador.Checked= true;
        this.PanelDatosAlumno.Visible = false;
        this.PanelDatosProfesor.Visible = false;
    }
    if (p.TipoUsuario == TipoUsuario.Profesor)
    {
        this.rbtTipoProfesor.Checked = true;
        this.txtTitulacion.Text = p.Titulacion;
        this.PanelDatosAlumno.Visible = false;
        this.PanelDatosProfesor.Visible = true;
    }
    if (p.TipoUsuario == TipoUsuario.Alumno)
    {
        this.rbtTipoAlumno.Checked = true;
        this.txtEstudiosPrevios.Text = p.EstudiosPrevios;
        this.PanelDatosAlumno.Visible = true;
        this.PanelDatosProfesor.Visible = false;
    }
}
}

```

F. Cuando es necesario introducir datos por pantalla se utilizan

```
asp:TextBox ID="txtNIFUsuario" runat="server"
```

G. En esta ventana, se tiene que seleccionar el tipo de usuario, dependiendo del mismo se mostrará un panel diferente en la zona inferior, este comportamiento, que también se utiliza para mostrar un tipo u otro de informe, se logra con la propiedad visible que tiene el objeto panel.

H. Los botones Añadir contienen el evento `<asp:Button ID="btnAnadir" runat="server" Text="Añadir" onclick="btnAnadir_Click" />`

Al igual que con el botón Buscar llama al método del mismo nombre de la clase, que es el encargado de introducir en la tabla correspondiente de la base de datos, el contenido de los cuadros de texto de la pantalla

```

protected void btnAnadir_Click(object sender, EventArgs e)
{
    try
    {
        Persona persona = CrearUsuario();
        gestionUsuarios.AnadirUsuario(persona);
        Helpers.ShowMessage("Usuario creado correctamente!");
    }
}

```

```

        catch (FaultException<NetAcademyDataException> ex)
        {
            Helpers.ShowMessage(ex.Reason.ToString());
        }
    }
    private Persona CrearUsuario()
    {
        Persona persona = new Persona();
        persona.Nombre = this.txtNombreUsuario.Text;
        persona.Apellidos = this.txtApellidoUsuario.Text;
        persona.NIF = this.txtNIFUsuario.Text;
        persona.Provincia = this.txtProvinciaUsuario.Text;
        persona.Email = this.txtEmailUsuario.Text;
        persona.Domicilio = this.txtDireccionUsuario.Text;
        persona.Login = this.txtLoginUsuario.Text;
        persona.Password = this.txtPassUsuario.Text;
        persona.Poblacion = this.txtPoblacionUsuario.Text;
        persona.Telefono = this.txtTelefonoUsuario.Text;
        persona.FechaInicio = DateTime.Now;
        persona.FechaFin = DateTime.MaxValue;
        persona.Titulacion = string.Empty;
        persona.EstudiosPrevios = string.Empty;

        if (rbtTipoAdministrador.Checked)
        {
            persona.TipoUsuario = TipoUsuario.Administrador;
        }
        else if (rbtTipoProfesor.Checked)
        {
            persona.TipoUsuario = TipoUsuario.Profesor;
            persona.Titulacion = this.txtTitulacion.Text;
        }
        else
        {
            persona.TipoUsuario = TipoUsuario.Alumno;
            persona.EstudiosPrevios = this.txtEstudiosPrevios.Text;
        }
        return persona;
    }
    protected void btnModificar_Click(object sender, EventArgs e)
    {
        try
        {
            Persona persona = CrearUsuario();
            gestionUsuarios.ModificarUsuario(persona.NIF, persona);
            Helpers.ShowMessage("Usuario modificado correctamente!");
        }
        catch (FaultException<NetAcademyDataException> ex)
        {
            Helpers.ShowMessage(ex.Reason.ToString());
        }
    }
}

```

Si la operación finaliza con éxito el usuario recibirá un mensaje de confirmación, si no se ha realizado se lanza una excepción.

- I. Los botones Modificar tienen también el evento onClick que llama al método correspondiente de la clase para modificar los campos de la tabla en la base de datos.

```

protected void btnModificar_Click(object sender, EventArgs e)
{
    try
    {

```

```

        Persona persona = CrearUsuario();
        gestionUsuarios.ModificarUsuario(persona.NIF, persona);
        Helpers.ShowMessage("Usuario modificado correctamente!");
    }
    catch (FaultException<NetAcademyDataException> ex)
    {
        Helpers.ShowMessage(ex.Reason.ToString());
    }
}

```

Si la operación se realiza con éxito se lo comunica al usuario con un mensaje, de lo contrario se captura la excepción.

- J. Los botones Limpiar llaman al método ClearFields() de la clase para vaciar los TextBox

```

private void ClearFields()
{
    this.txtNombreUsuario.Text = string.Empty;
    this.txtApellidoUsuario.Text = string.Empty;
    this.txtNIFUsuario.Text = string.Empty;
    this.txtProvinciaUsuario.Text = string.Empty;
    this.txtEmailUsuario.Text = string.Empty;
    this.txtDireccionUsuario.Text = string.Empty;
    this.txtLoginUsuario.Text = string.Empty;
    this.txtPassUsuario.Text = string.Empty;
    this.txtPoblacionUsuario.Text = string.Empty;
    this.txtTelefonoUsuario.Text = string.Empty;
    this.txtFechaAlta.Text = string.Empty;
    this.txtFechaBaja.Text = string.Empty;
    this.txtTitulacion.Text = string.Empty;
    this.txtEstudiosPrevios.Text = string.Empty;
    this.rbtTipoAdministrador.Checked = false;
    this.rbtTipoAlumno.Checked = false;
    this.rbtTipoProfesor.Checked = false;
}

protected void btnLimpiarCampos_Click(object sender, EventArgs e)
{
    ClearFields();
}

```

- K. Los botones Eliminar llaman al método correspondiente de la clase para eliminar el registro de la tabla correspondiente en la base de datos. Para realizar la búsqueda de usuario dentro de la colección de Personas empleamos LINQ to Entities para realizar la consulta.

```

protected void btnEliminar_Click(object sender, EventArgs e)
{
    try
    {
        gestionUsuarios.EliminarUsuario((from d in usuarios
                                         where d.NIF ==
                                         (string)this.GridView1.SelectedValue
                                         select d).First().NIF);
        ClearFields();
        Helpers.ShowMessage("Usuario eliminado correctamente!");
    }
    catch (FaultException<NetAcademyDataException> ex)
    {
        Helpers.ShowMessage(ex.Reason.ToString());
    }
}

```

Al igual que con la opción de añadir, si se realiza con éxito, se mostrará un mensaje notificándolo, si no se lanza una excepción.

L. Cuando es necesario mostrar un listado, en el caso de tablas que están relacionadas, como por ejemplo la tabla Curso y la tabla Horario, se utilizan **DropDownList**, que se rellenan utilizando el método ObtenerTodosxxxx() donde xxxx es el elemento concreto que se quiere obtener

```
protected void Page_Load(object sender, EventArgs e)
{
    table.Columns.Add("Id");
    table.Columns.Add("Nombre");
    table.Columns.Add("Profesor");
    table.Columns.Add("Aula");
    table.Columns.Add("Horario");
    table.Columns.Add("InicioCurso");
    table.Columns.Add("FinCurso");
    table.Columns.Add("Plazas");
    GridViewCursos.DataKeyNames = new string[] { "Id" };

    if (!IsPostBack)
    {
        List<Horario> horarios =
serviceNetAcademy.ObtenerTodasHorarios();
        dListHorariosCurso.DataSource = horarios;
        dListHorariosCurso.DataTextField = "DiaSemana";
        dListHorariosCurso.DataBind();
        List<Persona> profesores =
serviceNetAcademy.ObtenerTodosUsuarios(TipoUsuario.Profesor);
        dlistProfesores.DataSource = profesores;
        dlistProfesores.DataTextField = "nombre";
        dlistProfesores.DataBind();
    }
}
```

## 6.1.2 Capa WCF: Proyecto NetAcademy.WebServices

En esta capa se ha creado una carpeta Services que contiene tanto la interfaz **IServiveNETAcademy.cs** como la clase **ServiceNETAcademy.svc** que la implementa, también se ha creado una carpeta Excepciones que contiene la clase **NetAcademyDataException**

Para capturar las excepciones que se producen cuando se utiliza un método que puede provocar inconsistencias en la Base de Datos, como por ejemplo infringir la integridad referencial entre dos tablas, por lo tanto se utilizará en los métodos Añadir, Modificar y Eliminar

### 6.1.2.1 Dentro de la interfaz **IServiveNETAcademy.cs** se encuentran:

El **ServiceContract**, operaciones que ofrece el servicio ( **OpertionContract**) y **DataContract** que describe como son los datos compuestos añadidos a las entidades para poder comunicarse con la capa IU. Para hacer más legible el código, los he agrupado por #región, por ejemplo a continuación se muestran los métodos correspondientes a la #region correspondiente a los **USUARIOS**

```
[ServiceContract]
public interface IServicioNETAcademy
{
    #region USUARIOS
    [OperationContract]
    List<Persona> ObtenerUsuarios(string nif, string Nombre, TipoUsuario
tipousuario);
    [OperationContract]
```

```
List<Persona> ObtenerTodosUsuarios(TipoUsuario tipousuario);
[OperationContract]
void ModificarUsuario(string nif, Persona usuario);
[OperationContract]
void EliminarUsuario(string nif);
[OperationContract]
void AnadirUsuario(Persona usuario);
#endregion
```

El DataContract con los EnumMember, para la clase TipoCredenciales ,es la que se encarga de comprobar si el usuario es correcto

```
[DataContract]
public class TipoCredenciales
{
    private bool esUsuarioValido;
    private TipoErrorIdentificacion error;
    private TipoUsuario tipoUsuario;
    private string nombreUsuario;

    public TipoCredenciales()
    {
        esUsuarioValido = false;
    }

    [DataMember]
    public bool EsUsuarioValido
    {
        get { return esUsuarioValido; }
        set { esUsuarioValido = value; }
    }
}
```

.....  
...

#### 6.1.2.2 Dentro de la clase ServiveNETAcademy.cs.svc

Están implementados todos los métodos que contiene la interfaz IServiveNETAcademy.cs, al igual que se ha hecho en la interfaz, se han agrupado por #region para hacer más legible el código. Estos métodos son prácticamente los mismos para todos, a continuación se muestran los correspondientes a los cursos #region CURSOS

//Método que obtiene la lista de los cursos cuando el usuario los está buscando según los criterios que aparecen como argumentos del método, para ello debe acceder a la capa DATAACCESS

```
public List<Curso> ObtenerCursos(string nombre, DateTime inicio, int
plazas)
{
    List<Curso> res = new List<Curso>();
    using (CursosDataAccess cur = new CursosDataAccess())
    {
        res = cur.Obtener(nombre, inicio, plazas);
    }
    return res;
}
```

//Método que devuelve una lista con todos los cursos existentes, sirve para mostrar en los campos del tipo DropDownList, los valores correspondientes, también accede a la capa DATAACCESS

```
public List<Curso> ObtenerTodosCursos()
{
```

```
List<Curso> res = new List<Curso>();  
using (CursosDataAccess cur = new CursosDataAccess())  
{  
    res = cur.ObtenerTodos();  
}  
return res;  
}
```

//Método que devuelve una instancia de tipo Curso correspondiente al argumento Guid id, que es el campo identificador de la tabla, en este caso de la tabla Curso

```
public Curso ObtenerCurso(Guid id)  
{  
    Curso res;  
    using (CursosDataAccess cur = new CursosDataAccess())  
    {  
        res = cur.ObtenerPorId(id);  
    }  
    return res;  
}
```

//Metodo que modifica un registro de la tabla Cusro, recibe como argumentos el id del curso de tipo Guid y el objeto curso. Utiliza la clase CusrosDataAccess para el acceso a la BBDD, al ser un método que modifica se captura la excepción.

```
public void ModificarCurso(Guid id, Curso curso)  
{  
    using (CursosDataAccess cur = new CursosDataAccess())  
    {  
        try{  
            cur.Modificar(id, curso);  
        }  
        catch(Exception ex)  
        {  
            NetAcademyDataException exception = new  
NetAcademyDataException();  
            exception.Reason = "Error al Modificar el curso con id " +  
id.ToString() + " - " + ex.Message.ToString();  
            throw new FaultException<NetAcademyDataException>(exception,  
exception.Reason);  
        }  
    }  
}
```

//Método que elimina el registro de una tabla, recibe como parámetro el campo identificador del registro, utiliza la clase CursosDataAcces para acceder a la BBDD, al ser un método que modifica la BBDD se captura la excepción.

```
public void EliminarCurso(Guid id)  
{  
    using (CursosDataAccess cur = new CursosDataAccess())  
    {  
        try  
        {  
            cur.Eliminar(id);  
        }  
        catch(Exception ex)  
        {  
            NetAcademyDataException exception = new  
NetAcademyDataException();  
            exception.Reason = "Error al Eliminar el curso con id " +  
id.ToString() + " - " + ex.Message.ToString();  
            throw new FaultException<NetAcademyDataException>(exception,  
exception.Reason);  
        }  
    }  
}
```

```

    }
}
}

```

//Método que añade un registro a una tabla, en este caso un curso nuevo. Accede a la BBDD a través de la clase CursosDataAccess, se captura la excepción en caso de error.

```

public void AnadirCurso(Curso curso)
{
    using (CursosDataAccess cur = new CursosDataAccess())
    {
        try
        {
            cur.Anadir(curso);
        }
        catch(Exception ex)
        {
            NetAcademyDataException exception = new
NetAcademyDataException();
            exception.Reason = "Error al Añadir el curso con id " +
curso.IdCurso.ToString() + " - " + ex.Message.ToString();
            throw new FaultException<NetAcademyDataException>(exception,
exception.Reason);
        }
    }
}
#endregion

```

### 6.1.3 Capa Entities: NetAcademy.Entities

En un principio se había desarrollado la implementación en base a Entityes Framework, pero se encontraron problemas dentro del componente al utilizar herencia y relaciones múltiples entre las distintas tablas. Es un problema reportado por Microsoft por lo que se decidió un cambio en la implementación al modelo tradicional con entidades y capa de acceso a datos.

Las Entities son la transcripción a objetos de las tablas de la base de datos, en esta capa están las clases que representan a estos objetos, recogiendo sus atributos y los métodos get y set correspondientes, por ejemplo este es el código de la clase Aula

```

public class Aula
{
    #region Propiedades
    private Guid idAula = Guid.NewGuid();

    public Guid IdAula
    {
        get { return idAula; }
        set { idAula = value; }
    }

    private int capacidad;

    public int Capacidad
    {
        get { return capacidad; }
        set { capacidad = value; }
    }

    private string localizacion;

```

```

public string Localizacion
{
    get { return localizacion; }
    set { localizacion = value; }
}

private string nombre;

public string Nombre
{
    get { return nombre; }
    set { nombre = value; }
}
#endregion
}

```

#### 6.1.4 Capa DataBase: NetAcademy.DataBase

En esta capa se encuentra la definición de la Base de Datos, el diseño de la misma se ha detallado en el apartado Diseño de este documento, si se desea consultar seguir el siguiente hipervínculo [CONSULTAR MODELO BASE DE DATOS DE NET-Academy](#)

#### 6.1.5 Capa DataAccess: NetAcademy.DataAccess

Esta capa es la que conecta con la Base de Datos y contiene los métodos para poder interactuar con ella, implementa la interfaz IDisposable para poder liberar memoria y la conexión con la base de datos, a continuación se muestra como ejemplo el código de la clase [AulasDataAccess](#)

```

public class AulasDataAccess:IDisposable
{
    SqlConnection conn;

    public AulasDataAccess()
    {
        // Establecemos la conexión
        conn = new
SqlConnection(ConfigurationManager.ConnectionStrings["ModeloDatosContainer"].Conn
ectionString);
        conn.Open();
    }
    // Método que devuelve una lista con todas las Aulas que existen en la base de
datos, ejecutando una consulta a la misma
    public List<Aula> ObtenerTodos()
    {
        List<Aula> res = new List<Aula>();
        string command = string.Empty;
        command = "SELECT * FROM Aula";
        SqlCommand cmd = new SqlCommand(command, conn);
        SqlDataReader _reader = cmd.ExecuteReader();

        while (_reader.Read())
        {
            Aula _p = Construir(_reader);
            res.Add(_p);
        }
        return res;
    }
    // Método que devuelve un objeto de la clase Aula ejecutando una consulta de
selección a la base de datos donde el criterio es el identificador del Aula

```



```

public Aula Obtener(Guid id)
{
    List<Aula> res = new List<Aula>();
    string command = string.Empty;
    command = "SELECT * FROM Aula WHERE IdAula = '" + id + "'";
    SqlCommand cmd = new SqlCommand(command, conn);
    SqlDataReader _reader = cmd.ExecuteReader();

    while (_reader.Read())
    {
        Aula _p = Construir(_reader);
        res.Add(_p);
    }
    return res[0];
}

// Método que crea un objeto de la clase Aula con los atributos leídos
private Aula Construir(SqlDataReader _reader)
{
    Aula a = new Aula();
    a.Capacidad=(int)_reader["Capacidad"];
    a.IdAula=(Guid)_reader["IdAula"];
    a.Localizacion=_reader["Localizacion"] as string;
    a.Nombre=_reader["Nombre"] as string;
    return a;
}

// Método que devuelve una lista con todas las aulas que cumplen con los
argumentos pasados: nombre, capacidad y site, ejecutando una consulta de
selección a la basa de datos
public List<Aula> Obtener(string nombre, int capacidad, string site)
{
    List<Aula> res = new List<Aula>();
    string command = string.Empty;
    command = "SELECT * FROM Aula WHERE Nombre LIKE '%" + nombre +
        "' OR Capacidad = " + capacidad + " OR Localizacion LIKE
'" +
        site + "%'";
    SqlCommand cmd = new SqlCommand(command, conn);
    SqlDataReader _reader = cmd.ExecuteReader();

    while (_reader.Read())
    {
        Aula _p = Construir(_reader);
        res.Add(_p);
    }
    return res;
}

// Método que modifica un registro de la tabla Aula, recibe como argumentos el id
y el nombre del Aula y ejecuta la consulta de UPDATE
public void Modificar(Guid id, Aula aula)
{
    SqlCommand comm = new SqlCommand("UPDATE Aula SET Capacidad = " +
        aula.Capacidad + ", Localizacion = '" +
        aula.Localizacion.Trim() + "', Nombre = '" +
        aula.Nombre.Trim() + "' WHERE IdAula = " + id, conn);
    comm.ExecuteNonQuery();
}




// Método que elimina un registro de la tabla Aula cuyo id es pasado como
parámetro, mediante la consulta de Eliminacion sobre la tabla Aula cuyo criterio
es el id de la misma
public void Eliminar(Guid id)
{

```

```
SqlCommand comm = new SqlCommand("DELETE FROM Aula WHERE IdAula = '"  
+  
    id + "'", conn);  
comm.ExecuteNonQuery();  
}  
// Método que añade un nuevo registro a la tabla Aula mediante la sentencia  
// INSERT a la que se le pasan todos los valores de los campos de la tabla.  
public void Anadir(Aula aula)  
{  
    SqlCommand comm = new SqlCommand("INSERT INTO Aula VALUES (" +  
        aula.Capacidad + ", '" +  
        aula.Localizacion.Trim() + "', '" +  
        aula.Nombre.Trim() + "', '" +  
        aula.IdAula + "'" + ")", conn);  
    comm.ExecuteNonQuery();  
}  
// Se cierra la conexión y liberamos la memoria  
public void Dispose()  
{  
    conn.Close();  
    conn.Dispose();  
}  
}
```

### 6.1.6 MANUAL DE INSTALACIÓN Y/O CONFIGURACIÓN

Requerimientos de la aplicación:

-  Microsoft Visual Studio 2010
-  Microsoft SQL Server 2010 Express (o superior)
-  .NET Framework 4.0

### 6.1.7 Pasos para poder ejecutar la aplicación en entorno de Desarrollo

- Paso 1: Ejecución de script de creación de base de datos, fichero \Scripts de Base de datos\NetAcademyBaseDatos.sql, para la creación de base de datos y la carga inicial, se crearán los siguientes usuarios:
  - Administrador; nombre usuario = **pilar** password **1234**
  - Profesor; nombre usuario = **nico** password **1234**
  - Alumno; nombre usuario = **telmo** password **1234**
- Paso 2: Abrir la solución de Visual Studio, fichero NetAcademy.sln, y presionar ejecutar (F5).

### 6.1.8 Pasos para poder ejecutar la aplicación en entorno Real

No se ha podido conseguir la creación de un paquete de instalación de la aplicación de forma automática porque el software de creación de los mismos, de carácter gratuito, que se ha obtenido no ha funcionado.

### 6.1.9 Manual de usuario






El manual de usuario se corresponde con la secuencia de pantallas que se detallan en el apartado correspondiente al capítulo DISEÑO de este documento, si se desea consultar seguir el hipervínculo siguiente

[CONSULTAR MANUAL DE USUARIO](#)

## 7 EVALUACIÓN

### 7.1 Objetivos conseguidos

Al finalizar el proyecto se han obtenido los siguientes documentos:

-  Documento Plan de Trabajo (PEC1) :documento que recoge los objetivos, los requerimientos y la temporización inicial del proyecto.
-  Documento de Análisis y Diseño (PEC2): documento que recoge de manera formal, los requerimientos de la fase anterior, los casos de uso de los actores de Imismo, el modelo conceptual inicial de clases, el diseño inicial de la BBDD y los diseños de los prototipos de la interfaz de usuario.
-  Implementación(PEC3):entrega del código fuente del proyecto, desarrollado en el entorno Visual Studio 2010, junto con la Base de Datos , ambos operativos para poder realizar las pruebas de test oportunas, un documento en el que se comentan los aspectos mas relevantes de la fase y un Manualde usuario.
-  Memoria Final : Memoria del Proyecto.
-  Presentación Virtual: Presentación complementaria de la memoria en formato vídeo en la que se incluye una demo de la aplicación.

En dichos documentos, dependiendo de la fase en concreto, se han utilizado las herramientas necesarias del entorno .NET, WCF y SQL. El resultado de todas ellas es NET-Academy, aplicación para la gestión de una academia de enseñanza.

Los objetivos obtenidos, creo son bastante satisfactorios ya se cumplen la mayoría de los requisitos especificados en los objetivos del proyecto, como son el alta, modificación y eliminación de los diferentes perfiles de usuario, Administrador, Profesor y Alumno, la modificación del password de dichos usuarios, el alta, modificación y eliminación de aulas, cursos, horarios y matriculas, la evaluación de los Alumnos por parte de los Profesores y la obtención de los diferentes listados dependiendo del perfil del usuario que los solicite.

También se han gestionado los mensajes de validación de las operaciones que pueden producir cambios en la Base de Datos, como son las inserciones, modificaciones y borrado de registros, de cualquiera de las tablas y capturando el error en caso de no ser correctas.

Se ha añadido una nueva operación de búsqueda que permite mostrar determinados datos de la entidad a buscar, si se realiza sin ningún criterio , mostrará la totalidad de los elementos de dicha entidad y si se introducen uno o varios de los posibles criterios de búsqueda, mostrará aquellos registros que coinciden con ellos.

Ahora bien, debido al limitado tiempo del que se ha dispuesto para el desarrollo de la aplicación y al problema con la capa de datos indicado anteriormente, no se han podido implementar todas las funcionalidades que se propusieron al comienzo del proyecto, por lo tanto es evidente que se pueden introducir mejoras, mejoras que se detallan más específicamente en el apartado Trabajo futuro y recomendaciones de mejora de este documento.

### 7.2 Evaluación de costes

Por tratarse de un trabajo académico la justificación económica del proyecto no es valorable. En principio, el proyecto no tiene costos imputables más allá del esfuerzo a realizar.

### 7.3 Trabajo futuro y recomendaciones de mejora

Dado que esta es la primera vez que realizo un proyecto de esta dimensión con la tecnología .NET, el relativamente escaso tiempo del que he dispuesto para la realización del proyecto y los problemas que han ido surgiendo a lo largo de su desarrollo, hay aspectos del mismo que no están implementados, como por ejemplo:

- A. Falta por implementar la comprobación de la introducción de determinados datos en pantalla en formato correcto, como por ejemplo verificar si una dirección de correo electrónico o un número de teléfono tienen el formato correcto.
- B. Falta comprobar si un curso tiene el número máximo de alumnos.
- C. Falta comprobar si un usuario ya está dado de alta en el sistema y que un usuario no pueda darse de alta con perfiles incompatibles, como por ejemplo un usuario no puede ser a la vez Alumno y Profesor.
- D. No se ha podido crear el paquete de instalación al no disponer del software adecuado para ello.
- E. Falta implementar la subida de documentos, como por ejemplo pruebas de evaluación, por parte de los profesores.

Evidentemente estos puntos que no han sido implementados en esta versión son recomendaciones de futuras mejoras.

## 8 CONCLUSIONES

Este ha sido mi primer proyecto de importancia utilizando la tecnología .NET y el entorno de desarrollo de Microsoft Visual 2010, esto me ha supuesto un reto importante, que creo he conseguido superar satisfactoriamente, no tanto por la calidad de la aplicación (siempre mejorable), como por haber sido capaz de realizarlo y por los conocimientos adquiridos, prueba de ello es que estoy empezando a utilizarlo en otros desarrollos.

El TFC me ha permitido seguir todas las fases propias de un proyecto profesional de desarrollo de software (análisis, diseño e implementación), tanto a las correspondientes a UI, como a las correspondientes al desarrollo del código. De este modo he podido comprobar como cada fase alimenta al resto de forma que, el proyecto no es algo estático, que a medida que se desarrollan las diferentes fases, es cuando surgen los problemas, ya sea porque haya que realizar cambios en el diseño o por comportamientos del entorno de programación, de hecho he tenido que realizar cambios en la fase de implementación con respecto a la gestión de la capa de datos, como he indicado anteriormente, este cambio ha supuesto un trabajo extra, ya que he tenido que replantear y rehacer toda la gestión de dicha capa, que probablemente haya repercutido a la hora de dejar sin implementar algunas funcionalidades.

El motivo de seleccionar el área .Net para realizar este TFC, fue la importancia que creo que tiene en el mundo laboral y profesional la plataforma de Microsoft y el valor añadido que implica conocerla a la hora de progresar profesionalmente. A lo largo

de los estudios de Informática en la UOC, se ha trabajado siempre con la plataforma de Sun Microsystems y ahora que he tenido la oportunidad de conocer la otra gran plataforma de desarrollo no he querido desaprovecharla.

## 9 BIBLIOGRAFÍA

1. **Jordi Conesa Caralt, Angels Rius Gavidia, Jordi Ceballos, David Gañán Jimenez.** *Introducción a .NET.* Barcelona : Editorial UOC, 2010.
2. **Microsoft.** <http://www.microsoft.com/spain/formacion/default.mspx>.  
<http://social.msdn.microsoft.com/Forums/es-es/categories/>. [En línea]
3. **Chase, Nicolas.** *Active Server Pages 3.0.* Madrid : Prentice Hall, 2008.
4. <http://www.dotnetonweb.com/category/dot-net-tutorials/visual-studio-2010-tutorials/>. [En línea]
5. **Profesorado de la UOC. Asignatura:** *Metodología y Gestión de Proyectos Informáticos.* s.l.Editorial : UOC.
6. — **Profesorado de la UOC. Asignatura:** *Bases de Datos I.* s.l.Editorial: UOC.
7. — **Profesorado de la UOC. Asignatura:** *Bases de Datos II.* s.l. Editorial: UOC.
8. — **Profesorado de la UOC. Asignatura:** *Ingeniería del Software y Sistemas Distribuídos.* s.l. Editorial: UOC.
9. **UOC.** <http://openaccess.uoc.edu/webapps/o2/?i.launcher>. *Repositorio Institucional de la UOC.* [En línea]
10. **Microsoft.** <http://msdn.microsoft.com/en-us/VS2010TrainingCourse>. [En línea]