

**ESCUELA POLITÉCNICA  
SUPERIOR DE CÓRDOBA**  
Universidad de Córdoba



**INGENIERÍA DEL SOFTWARE**

*Grado en Ingeniería Informática*

**Trabajo final de prácticas**

**Autores:** Gálvez Ruz Antonio, López Rubias Antonio

**DNI:** 32731194-D, 80164190J

**Correo:** i52garua@uco.es, i42lorua@uco.es



UNIVERSIDAD DE CÓRDOBA



## Índice

<b>1. Práctica 2</b>	4
1.1. Definición del problema	4
1.2. Extracción de requisitos	4
1.2.1. Requisitos funcionales	4
1.2.2. Requisitos no funcionales	4
1.2.3. Prioridad de requisitos funcionales	5
1.3. Historias de usuario	5
1.3.1. Listar alumno	5
1.3.2. Visualizar alumno	6
1.3.3. Editar alumno	6
1.3.4. Eliminar alumno	6
1.3.5. Insertar alumno	6
1.3.6. Guardar copias de seguridad	6
1.3.7. Cargar copias de seguridad	7
1.3.8. Mostrar resultados	7
1.3.9. Login	7
1.3.10. Registrarse	7
1.3.11. Guardar fichero	7
1.3.12. Cargar fichero	8
1.3.13. Buscar grupo	8
1.4. Casos de uso	8
1.4.1. CU - Listar alumnos	8
1.4.2. CU – Visualizar alumno	8
1.4.3. CU – Editar alumno	9
1.4.4. CU – Eliminar alumno	9
1.4.5. CU – Añadir alumno	9
1.4.6. CU – Guardar copias de seguridad	10
1.4.7. CU – Cargar copias de seguridad	10
1.4.8. CU – Buscar por grupo	11
1.4.9. CU – Iniciar sesión	11
1.4.10. CU – Registrarse	11
1.5. Diagrama de casos de uso	12
<b>2. Práctica 3</b>	12
2.1. Diagrama de clases	12
2.2. Diagrama de secuencia	13

2.2.1. Listar alumnos .....	13
2.2.2. Visualizar alumno .....	14
2.2.3. Iniciar sesión.....	14
2.2.4. Registrarse.....	15
2.2.5. Editar alumno .....	15
2.2.6. Eliminar alumno .....	16
2.2.7. Cargar copias de seguridad .....	16
2.2.8. Guardar copias de seguridad .....	17
2.2.9. Añadir alumno .....	17
2.2.10. Buscar grupo .....	18
<b>3. Práctica 4.....</b>	<b>18</b>
3.1. Metodología SCRUM .....	18
3.1.1. Product Backlog.....	18
3.1.2. Sprint Backlog.....	19
3.1.3. Burndown chart.....	20
3.2. Matrices de validación .....	21
<b>4. Bibliografía.....</b>	<b>22</b>

## 1. Práctica 2

### 1.1. Definición del problema

En este problema se plantea la realización de una agenda para la gestión de la asignatura Ingeniería del Software la cual permita la gestión del alumnado de las clases prácticas junto con los grupos de trabajo que cada uno forma y gestionada por cada profesor mediante un sistema de registro en la misma.

La aplicación constará, a grandes rasgos, de dos partes, un sistema de registro en la aplicación diferenciando entre dos tipos de profesores (coordinador y colaborador) y un menú el cual permita todas las operaciones posibles sobre los alumnos y grupos que serán expuestas a continuación en los requisitos.

### 1.2. Extracción de requisitos

En este documento recogemos toda la información recogida por parte de la entrevista con el cliente dividido en los diferentes requisitos.

#### 1.2.1. Requisitos funcionales

Definen una función del sistema software o sus componentes.

RF	Descripción
<b>RF-1</b>	El sistema debe poder listar los contactos de los alumnos (ordenado o por nombre o por DNI o por apellido o por curso más alto matriculado de forma ascendente o descendente)
<b>RF-2</b>	Se deberá poder visualizar los datos de los alumnos
<b>RF-3</b>	El sistema podrá buscar un alumnos por DNI, apellido o grupo
<b>RF-4</b>	El sistema debe poder editar los datos de un alumno
<b>RF-5</b>	El sistema debe permitir eliminar un alumno tanto por DNI como por apellido
<b>RF-6</b>	El sistema debe poder insertar los datos de alumnos
<b>RF-7</b>	El coordinador debe poder crear copias de seguridad, guardarlas y cargarlas
<b>RF-8</b>	El sistema mostrará la información mediante la línea de comandos o HTML o ambas
<b>RF-9</b>	El profesor coordinador debe poder dar de alta a los profesores ayudantes
<b>RF-10</b>	El usuario debe poder loguearse en el sistema

#### 1.2.2. Requisitos no funcionales

Atributos de calidad del sistema.

RNF	Descripción
<b>RNF-1</b>	El proyecto se tiene que implementar en C++
<b>RNF-2</b>	La documentación debe de realizarse en lenguaje Markdown
<b>RNF-3</b>	El sistema debe funcionar en Linux
<b>RNF-4</b>	El sistema deberá incorporar como interfaz gráfica la línea de comandos y como opcional cualquiera
<b>RNF-5</b>	La seguridad de los datos se delega al sistema operativo
<b>RNF-6</b>	Si se añade un DNI o correo ya existente, el software dará un aviso de que no se puede insertar o modificar dicho alumno

<b>RNF-7</b>	En caso de que una búsqueda por apellidos y aparezcan varios registros, dará opción a seleccionar el deseado
<b>RNF-8</b>	Al iniciar el programa se mostrará un menú
<b>RNF-9</b>	La capacidad máxima de alumnos registrados será de 150
<b>RNF-10</b>	Un grupo puede tener como máximo un líder
<b>RNF-11</b>	El sistema debe distinguir entre profesor coordinador y profesores ayudantes
<b>RNF-12</b>	Solo puede haber un profesor coordinador y está predefinido en el sistema como admin
<b>RNF-13</b>	Ningún profesor podrá cambiar su nombre de usuario en el sistema de login
<b>RNF-14</b>	El profesor coordinador no podrá cambiar su contraseña
<b>RNF-15</b>	En el fichero credenciales se guardará el rol de los profesores, su usuario y su contraseña
<b>RNF-16</b>	El fichero credenciales estará oculto y se creará al iniciar el sistema si no existe previamente
<b>RNF-17</b>	A la hora de registrarse nuevos profesores ayudantes se deben registrar con una contraseña si ya existen predefinidos en el fichero de credenciales
<b>RNF-18</b>	El sistema solo se utilizará para la asignatura de Ingeniería del Software de los alumnos

### 1.2.3. Prioridad de requisitos funcionales

Prioridad asignada a los requisitos ordenada de mayor a menor.

Prioridad	Requisitos
<b>P-1</b>	RF-6, RF-10
<b>P-2</b>	RF-7
<b>P-3</b>	RF-3, RF-9
<b>P-4</b>	RF-4, RF-5
<b>P-5</b>	RF-1, RF-2, RF-8

### 1.3. Historias de usuario

En esta primera fase de desarrollo software identificaremos las historias de usuario que son representaciones de requisitos escritos en una o dos frases utilizando el lenguaje común del usuario.

#### 1.3.1. Listar alumno

Identificación	001 Listar alumnos
<b>Prioridad</b>	5
<b>Anverso</b>	El sistema debe permitir al usuario listar todos los alumnos que estén registrado en la aplicación.
<b>Reverso</b>	<ul style="list-style-type: none"> <li>Se desea visualizar todos los alumnos registrados.</li> <li>El listado se podrá ordenar ascendente o descendentemente.</li> <li>El listado se podrá ordenar por DNI, Apellido o curso más alto matriculado.</li> </ul>

### 1.3.2. Visualizar alumno

Identificación 002 Visualizar alumno	
<b>Prioridad</b>	5
<b>Anverso</b>	El sistema debe permitir al usuario visualizar los datos de un alumno.
<b>Reverso</b>	<ul style="list-style-type: none"><li>Se deben visualizar DNI, nombre, apellidos, teléfono, email, dirección postal, curso más alto matriculado, fecha de nacimiento, equipo al que pertenece y si es líder no.</li></ul>

### 1.3.3. Editar alumno

Identificación 003 Editar alumno	
<b>Prioridad</b>	4
<b>Anverso</b>	El sistema debe permitir al usuario la opción de editar un alumno.
<b>Reverso</b>	<ul style="list-style-type: none"><li>Se podrá modificar cualquier alumno y cada uno de sus campos.</li><li>Si se modifica el campo DNI o correo y se introduce alguno ya existente, ese valor no podrá ser cambiado a éste.</li></ul>

### 1.3.4. Eliminar alumno

Identificación 004 Eliminar alumno	
<b>Prioridad</b>	5
<b>Anverso</b>	El sistema debe permitir eliminar un alumno del listado.
<b>Reverso</b>	<ul style="list-style-type: none"><li>Se buscará el alumno por DNI o apellido, en caso de varios con el mismo apellido aparecerán todos estos.</li><li>Se seleccionará el alumno a eliminar y una vez confirmada la operación, éste desaparecerá del listado.</li></ul>

### 1.3.5. Insertar alumno

Identificación 005 Insertar alumno	
<b>Prioridad</b>	1
<b>Anverso</b>	El sistema debe poder insertar los datos de alumnos.
<b>Reverso</b>	<ul style="list-style-type: none"><li>Los campos obligatorios sobre la información de los contactos son todos menos equipo y si son líder o no.</li><li>No dejará introducir el alumno en el caso de haber un alumno con el mismo DNI o correo.</li></ul>

### 1.3.6. Guardar copias de seguridad

Identificación 006 Guardar copias de seguridad	
<b>Prioridad</b>	2
<b>Anverso</b>	El sistema debe permitir al coordinador crear copias de seguridad del listado de los alumnos.
<b>Reverso</b>	<ul style="list-style-type: none"><li>Se desea guardar en un fichero toda la información introducida con respecto a los alumnos.</li><li>La información de cada campo será insertada en el fichero separada por espacios.</li></ul>

### 1.3.7. Cargar copias de seguridad

Identificación 007 Cargar copias de seguridad	
<b>Prioridad</b>	2
<b>Anverso</b>	El sistema debe permitir al coordinador cargar las copias de seguridad del listado de los alumnos.
<b>Reverso</b>	<ul style="list-style-type: none"><li>Se desea cargar desde un fichero toda la información con respecto a los alumnos previamente guardada o introducida en texto plano.</li></ul>

### 1.3.8. Mostrar resultados

Identificación 008 Mostrar resultados	
<b>Prioridad</b>	5
<b>Anverso</b>	El sistema debe permitir mostrar los resultados de las búsquedas y diferentes operaciones.
<b>Reverso</b>	<ul style="list-style-type: none"><li>Los resultados podrán ser mostrados por línea de comandos, HTML o ambas opciones dependiendo de lo que el usuario decida.</li></ul>

### 1.3.9. Login

Identificación 009 Login	
<b>Prioridad</b>	6
<b>Anverso</b>	El sistema debe permitir al usuario loguearse usando su usuario y contraseña.
<b>Reverso</b>	<ul style="list-style-type: none"><li>Se desea que todos los usuarios registrados puedan loguearse en el sistema.</li><li>El login se comprobará a través de un fichero binario de credenciales.</li></ul>

### 1.3.10. Registrarse

Identificación 010 Registrarse	
<b>Prioridad</b>	6
<b>Anverso</b>	El sistema debe permitir al usuario registrarse sólo si previamente existe en el fichero de credenciales y no tiene una contraseña asignada
<b>Reverso</b>	<ul style="list-style-type: none"><li>Se desea que todos los usuarios existentes en el fichero de credenciales puedan registrarse en el sistema.</li><li>Sólo se podrán registrar si no poseen una contraseña asignada.</li><li>Al registrarse en el sistema podrán realizar el login con su usuario y contraseña asignados.</li></ul>

### 1.3.11. Guardar fichero

Identificación 011 Guardar fichero	
<b>Prioridad</b>	6
<b>Anverso</b>	El sistema debe permitir guardar en un fichero los datos obtenidos tras el uso de la aplicación.
<b>Reverso</b>	<ul style="list-style-type: none"><li>Se desea que los usuarios puedan utilizar esta opción para guardar los datos con los que han trabajado.</li><li>Esta opción podrá ser usada por todos los usuarios logueados del sistema.</li></ul>



### 1.3.12. Cargar fichero

Identificación		012 Cargar fichero
<b>Prioridad</b>		6
<b>Anverso</b>		El sistema debe permitir cargar de un fichero datos necesarios para el uso de la aplicación.
<b>Reverso</b>		<ul style="list-style-type: none"> <li>Se desea que los usuarios puedan utilizar esta opción para cargar los datos con los que van a trabajar.</li> <li>Esta opción podrá ser usada por todos los usuarios logueados del sistema.</li> </ul>

### 1.3.13. Buscar grupo

Identificación		013 Buscar grupo
<b>Prioridad</b>		6
<b>Anverso</b>		El sistema debe permitir al usuario buscar un grupo de alumnos existentes en la aplicación.
<b>Reverso</b>		<ul style="list-style-type: none"> <li>Se desea que los usuarios puedan utilizar esta opción para buscar por el número de grupo a los alumnos pertenecientes a dicho grupo.</li> </ul>

## 1.4. Casos de uso

En esta fase nos dedicaremos a la extracción de los casos de uso que son la descripción de una acción o una actividad dentro de nuestro problema a resolver.

### 1.4.1. CU - Listar alumnos

Identificación		001 CU - Listar alumnos
<b>Actores principales</b>		Usuario
<b>Actores secundarios</b>		Ninguno
<b>Breve descripción</b>		El sistema debe permitir al usuario listar todos los alumnos que estén registrados en la aplicación.
<b>Precondiciones</b>		Debe haber al menos un alumno registrado en el sistema.
<b>Flujo principal</b>		<ol style="list-style-type: none"> <li>El usuario elige la opción de listar alumnos.</li> <li>El sistema da a elegir el campo por el cual buscar.</li> <li>El usuario elige el orden de búsqueda.</li> <li>El usuario elige como visualizar el resultado.</li> </ol>
<b>Postcondiciones</b>		El sistema muestra todos los alumnos registrados.
<b>Flujos alternativos</b>		2.a. Si no existe ningún alumno registrado, el sistema mostrará un mensaje de advertencia.

### 1.4.2. CU – Visualizar alumno

Identificación		002 CU - Visualizar alumno
<b>Actores principales</b>		Usuario
<b>Actores secundarios</b>		Ninguno
<b>Breve descripción</b>		El sistema debe permitir al usuario visualizar los datos de un alumno.
<b>Precondiciones</b>		Debe haber al menos un alumno registrado en el sistema.
<b>Flujo principal</b>		<ol style="list-style-type: none"> <li>El usuario elige la opción de visualizar alumno.</li> </ol>

	<ol style="list-style-type: none"> <li>2. El sistema da a elegir entre visualizar por DNI o por Apellido.</li> <li>3. Se elegirá el orden de visualización.</li> <li>4. El usuario elige como visualizar el resultado.</li> </ol>
<b>Postcondiciones</b>	El sistema muestra los datos del alumno.
<b>Flujos alternativos</b>	2.a. Si no existe el alumno mostrará un mensaje de advertencia

#### 1.4.3. CU – Editar alumno

Identificación	003 CU - Editar alumno
<b>Actores principales</b>	Usuario
<b>Actores secundarios</b>	Ninguno
<b>Breve descripción</b>	El sistema debe permitir al usuario la opción de editar un alumno.
<b>Precondiciones</b>	El alumno a editar debe existir en el sistema.
<b>Flujo principal</b>	<ol style="list-style-type: none"> <li>1. Se introducirá el DNI del alumno a editar.</li> <li>2. Se visualizará los datos actuales de dicho alumno.</li> <li>3. El usuario seleccionará el campo a editar y lo podrá modificar.</li> <li>4. Seleccionará la opción de fin de edición para terminar la operación.</li> </ol>
<b>Postcondiciones</b>	El alumno se modificará.
<b>Flujos alternativos</b>	<ol style="list-style-type: none"> <li>3.a. No se podrá modificar el campo del DNI.</li> <li>3.b. Se puede modificar el campo de Correo, pero si se introduce uno ya existente, éste no podrá ser modificado por el actual.</li> <li>3.c. No se podrá asignar como líder a un alumno si en su grupo ya existe otro líder.</li> </ol>

#### 1.4.4. CU – Eliminar alumno

Identificación	004 CU - Eliminar alumno
<b>Actores principales</b>	Usuario
<b>Actores secundarios</b>	Ninguno
<b>Breve descripción</b>	El sistema debe permitir eliminar un alumno del listado.
<b>Precondiciones</b>	El alumno a eliminar debe existir en el sistema.
<b>Flujo principal</b>	<ol style="list-style-type: none"> <li>1. Se introducirá el DNI o apellido del alumno a eliminar.</li> <li>2. Se visualizará los datos actuales de dicho alumno.</li> <li>3. El usuario seleccionará confirmará dicha operación.</li> <li>4. Seleccionará la opción de fin de edición para terminar la operación.</li> </ol>
<b>Postcondiciones</b>	El alumno se eliminará.
<b>Flujos alternativos</b>	1.a. Si se introduce un apellido y hay varios registros con el mismo, aparecerá una lista con todos los alumnos con ese apellido y se seleccionará el deseado a eliminar.

#### 1.4.5. CU – Añadir alumno

Identificación	005 CU - Añadir alumno
<b>Actores principales</b>	Usuario
<b>Actores secundarios</b>	Ninguno
<b>Breve descripción</b>	El sistema debe poder insertar los datos de alumnos.

<b>Precondiciones</b>	El alumno a introducir no puede existir en la lista.
<b>Flujo principal</b>	<ol style="list-style-type: none"> <li>1. Se comprueba el número de alumnos en el sistema.</li> <li>2. Se introducirá el DNI del alumno</li> <li>3. Se introducirá el correo del alumno</li> <li>4. Se introducirá el Apellido, Teléfono, Dirección, Curso más alto matriculado, Fecha nacimiento, Grupo de trabajo y si es o no Líder del grupo.</li> </ol>
<b>Postcondiciones</b>	El alumno se insertará en la lista.
<b>Flujos alternativos</b>	<ol style="list-style-type: none"> <li>1.a. Si hay 150 alumnos registrados se mostrará un mensaje de advertencia.</li> <li>2.a. Una vez introducido el DNI, el sistema buscará si existe ese DNI para que, en ese caso, no se siga introduciendo más datos.</li> <li>3.a. Una vez introducido el correo, el sistema buscará si existe ese correo para que en ese caso, no se siga introduciendo más datos.</li> </ol>

#### 1.4.6. CU – Guardar copias de seguridad

<b>Identificación</b> 006 CU – Guardar copias de seguridad	
<b>Actores principales</b>	Usuario
<b>Actores secundarios</b>	Ninguno
<b>Breve descripción</b>	El sistema debe permitir al usuario guardar en ficheros las copias de seguridad del listado de los alumnos.
<b>Precondiciones</b>	Debe haber al menos un alumno registrado en el sistema.
<b>Flujo principal</b>	<ol style="list-style-type: none"> <li>1. El usuario elige la opción de guardar copia de seguridad.</li> <li>2. El sistema pide el nombre del fichero en el cual guardar la información.</li> <li>3. El sistema crea el fichero y guarda la información en él.</li> </ol>
<b>Postcondiciones</b>	El sistema muestra los datos del alumno.
<b>Flujos alternativos</b>	3.a. Si ya existe un fichero con dicho nombre se sobrescribirá la información.

#### 1.4.7. CU – Cargar copias de seguridad

<b>Identificación</b> 007 CU – Cargar copias de seguridad	
<b>Actores principales</b>	Usuario
<b>Actores secundarios</b>	Ninguno
<b>Breve descripción</b>	El sistema debe permitir al usuario cargar desde un fichero las copias de seguridad del listado de los alumnos.
<b>Precondiciones</b>	Debe existir el fichero a cargar.
<b>Flujo principal</b>	<ol style="list-style-type: none"> <li>1. El usuario elige la opción de cargar copia de seguridad.</li> <li>2. El sistema pide el nombre del fichero desde el cual cargar la información.</li> <li>3. El sistema elimina la información almacenada en ese hasta ese momento y carga la nueva almacenada en el fichero.</li> </ol>
<b>Postcondiciones</b>	El sistema muestra los datos del alumno.
<b>Flujos alternativos</b>	3.a. Si no existe un fichero con dicho nombre se mostrará un mensaje de advertencia.

#### 1.4.8. CU – Buscar por grupo

Identificación		008 CU – Buscar por grupo
Actores principales		Usuario
Actores secundarios		Ninguno
Breve descripción		El sistema debe permitir al usuario buscar alumnos por el grupo al que pertenecen.
Precondiciones		Debe haber al menos un alumno registrado en el sistema.
Flujo principal		<ol style="list-style-type: none"> <li>1. El usuario elige la opción buscar por grupo.</li> <li>2. El sistema pide el número del grupo a buscar.</li> </ol>
Postcondiciones		El sistema muestra los componentes del grupo.
Flujos alternativos		2.a. Si no existe el grupo a buscar se mostrará un mensaje de advertencia.

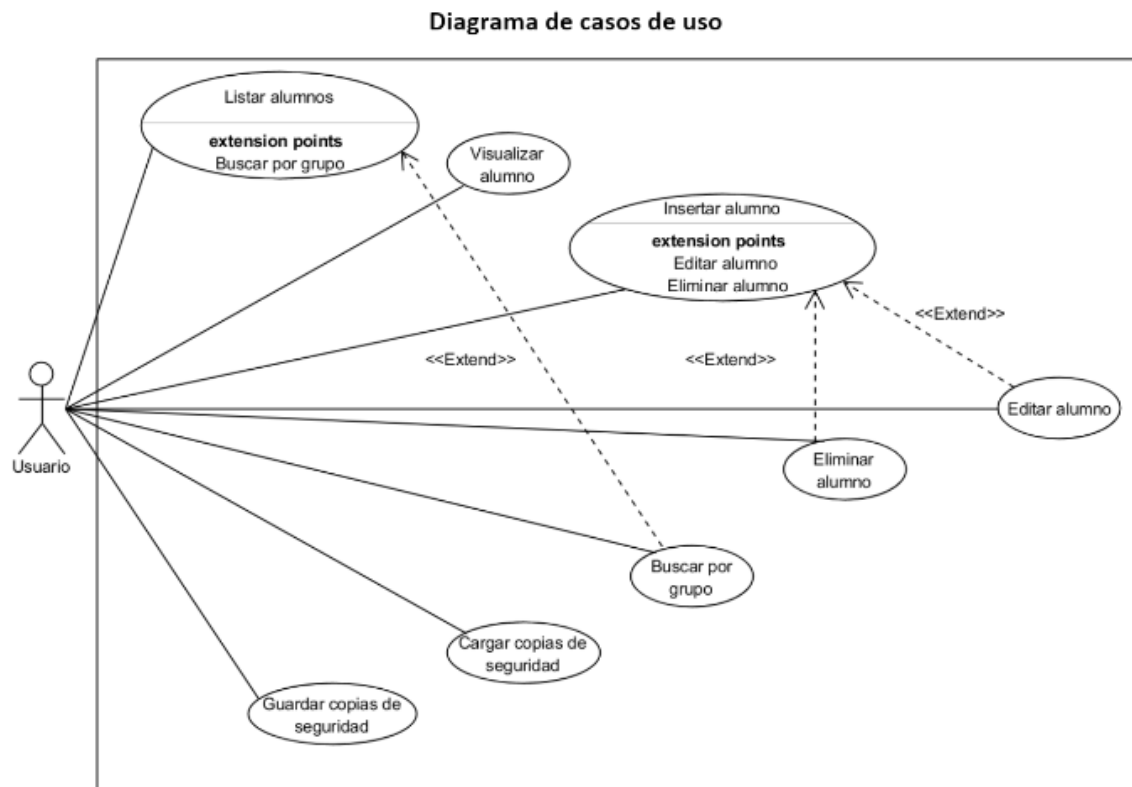
#### 1.4.9. CU – Iniciar sesión

Identificación		009 CU – Iniciar sesión
Actores principales		Usuario
Actores secundarios		Ninguno
Breve descripción		El sistema debe permitir al usuario la opción de loguearse en el sistema.
Precondiciones		Debe estar registrado en el fichero de credenciales.
Flujo principal		<ol style="list-style-type: none"> <li>1. El usuario elige la opción de iniciar sesión.</li> <li>2. El sistema pide el nombre del profesor y la contraseña.</li> </ol>
Postcondiciones		El sistema accede al menú principal y se podrá trabajar con la agenda.
Flujos alternativos		2.a. Si no existe el profesor mostrará un mensaje de advertencia.

#### 1.4.10. CU – Registrarse

Identificación		010 CU – Registrarse
Actores principales		Usuario
Actores secundarios		Ninguno
Breve descripción		El sistema debe permitir al usuario la opción de registrarse en el sistema.
Precondiciones		Debe estar registrado el nombre del profesor en el fichero de credenciales.
Flujo principal		<ol style="list-style-type: none"> <li>1. El usuario elige la opción de registrarse.</li> <li>2. El sistema pide el nombre del profesor.</li> <li>3. El sistema pide la contraseña nueva a asignar al profesor.</li> </ol>
Postcondiciones		El sistema vuelve a menu principal para poder iniciar sesión.
Flujos alternativos		2.a. Si no existe el profesor mostrará un mensaje de advertencia. 2.b. Si existe el profesor y tiene contraseña mostrará un mensaje de advertencia.

## 1.5. Diagrama de casos de uso

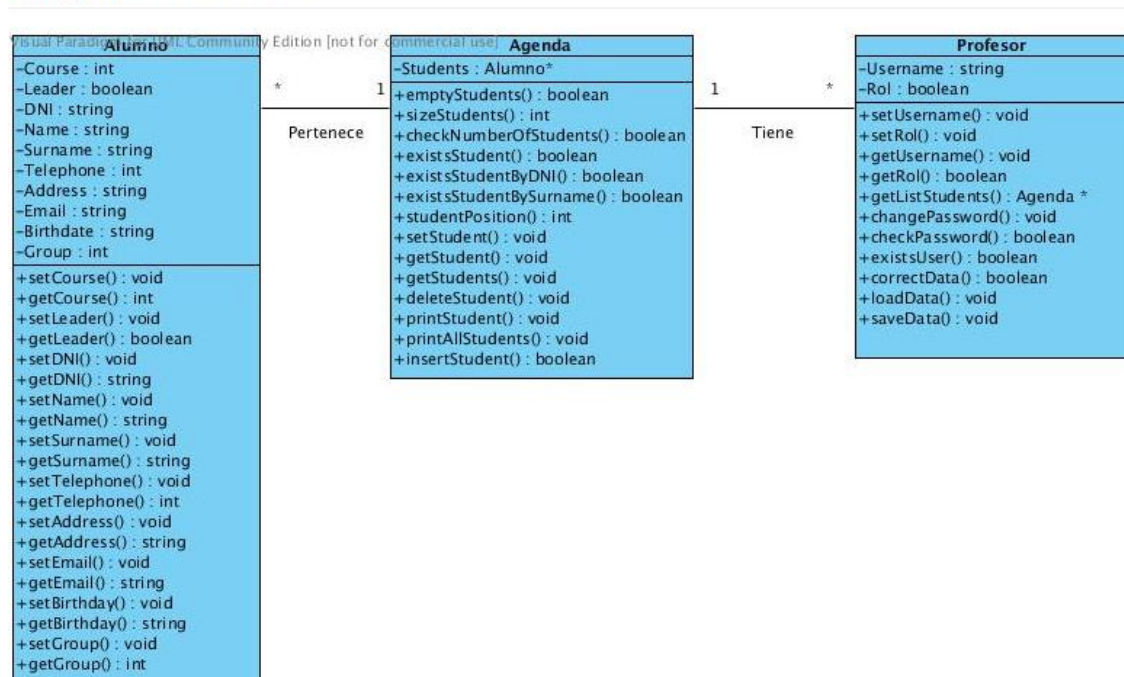


## 2. Práctica 3

### 2.1. Diagrama de clases

En esta fase del problema describiremos la estructura del sistema mediante un diagrama de clases y su explicación:

### Diagrama de clase



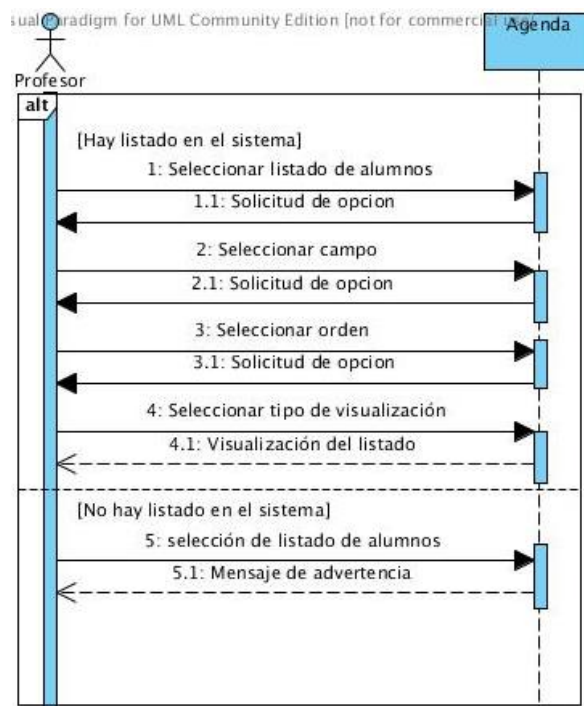
<b>Clases</b>	<ul style="list-style-type: none"> <li>• <b>Alumno</b></li> <li>• <b>Agenda</b></li> <li>• <b>Profesor</b></li> </ul>
<b>Restricciones</b>	<ul style="list-style-type: none"> <li>• A una agenda pertenecen muchos alumnos los cuales estarán almacenados en un vector de alumnos.</li> <li>• Un alumno pertenece a una agenda.</li> <li>• Los profesores tienen una agenda la cual puede ser administrada por los diferentes profesores, ya tengan rol de coordinadores o de ayudantes.</li> </ul>
<b>Métodos</b>	<ul style="list-style-type: none"> <li>• En la clase Alumno tendremos los métodos set y get de los atributos del mismo.</li> <li>• En la case Agenda los métodos de inserción, edición, eliminación y diferentes comprobaciones para realizar dichas operaciones.</li> <li>• En la clase Profesor tendremos los métodos para obtener dicha agenda y poder realizar copias de seguridad, cargar las mismas mediante ficheros y comprobaciones a la hora del logueo a la aplicación.</li> </ul>

## 2.2. Diagrama de secuencia

En este apartado realizamos los diagramas de secuencia que corresponden con el problema.

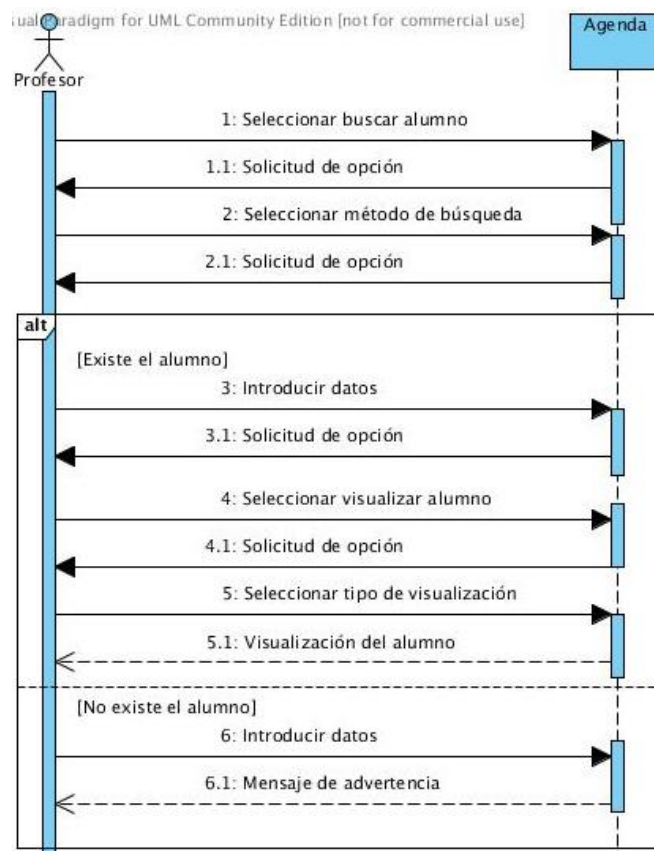
### 2.2.1. Listar alumnos

En este diagrama se explican los pasos que se llevan a cabo al listar a los alumnos en el sistema.



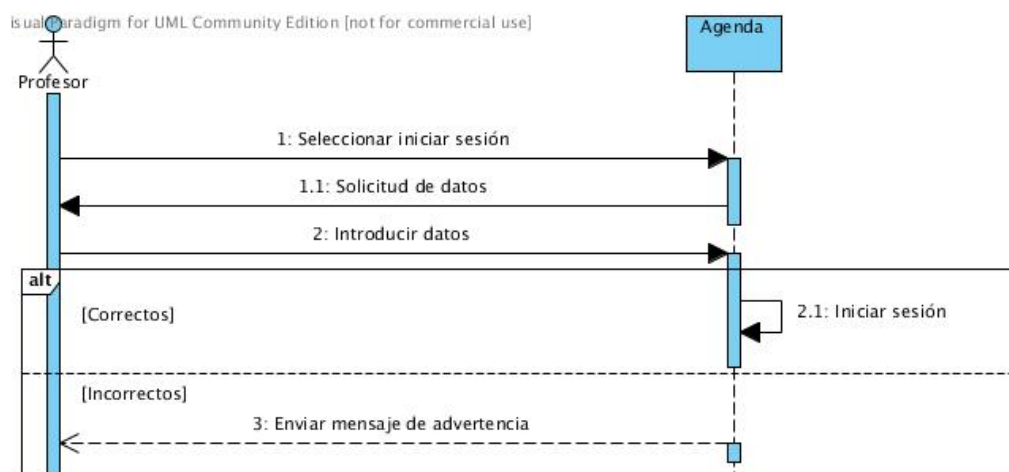
### 2.2.2. Visualizar alumno

En este diagrama se explican los pasos que se llevan a cabo al visualizar un alumno en el sistema.



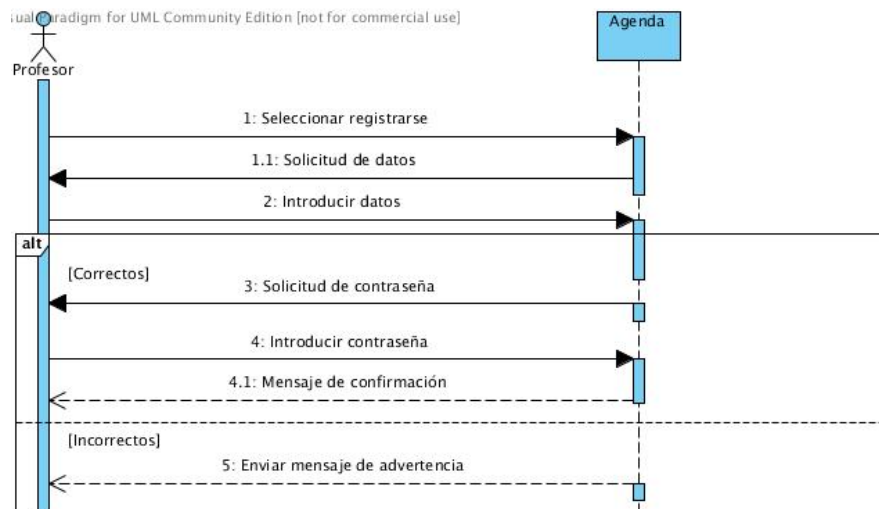
### 2.2.3. Iniciar sesión

En este diagrama se explican los pasos que se llevan a cabo al iniciar sesión en el sistema.



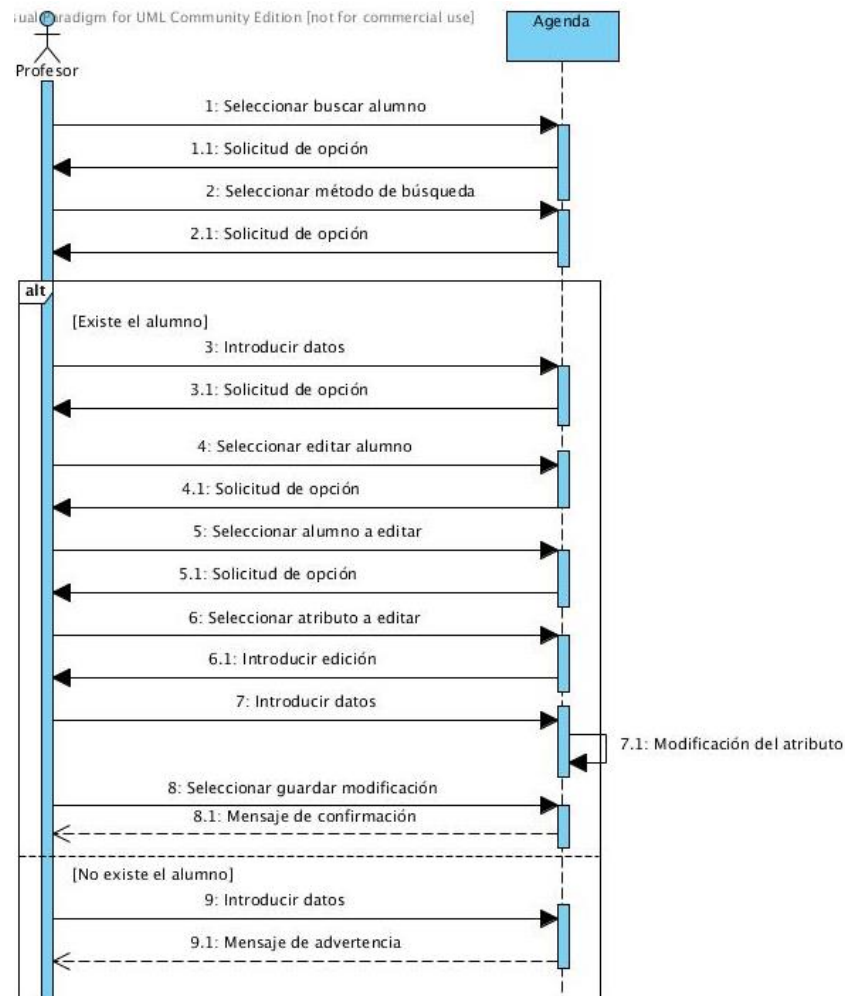
#### 2.2.4. Registrarse

En este diagrama se explican los pasos que se llevan a cabo al registrarse en el sistema.



#### 2.2.5. Editar alumno

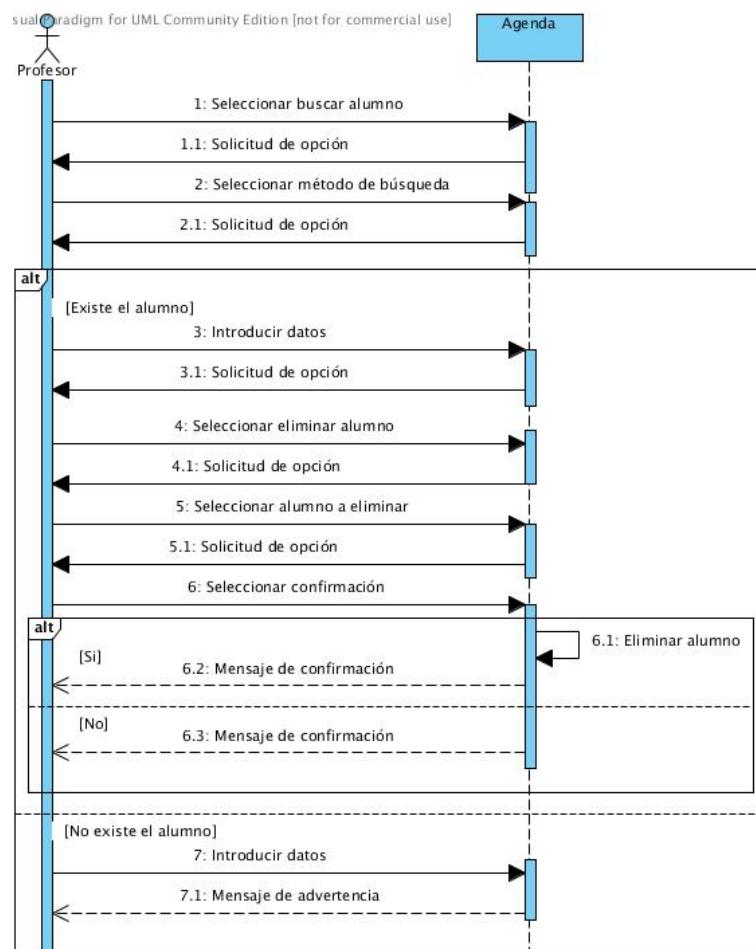
En este diagrama se explican los pasos que se llevan a cabo al editar un alumno en el sistema.





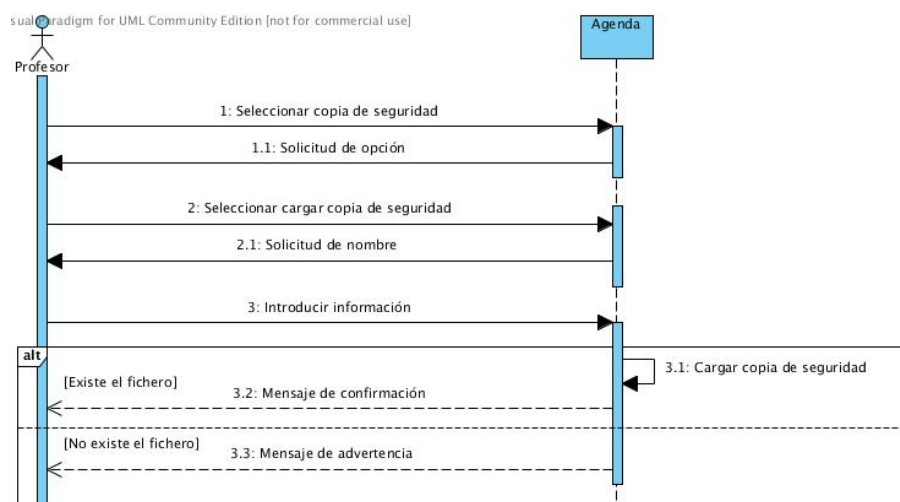
### 2.2.6. Eliminar alumno

En este diagrama se explican los pasos que se llevan a cabo al eliminar un alumno en el sistema.



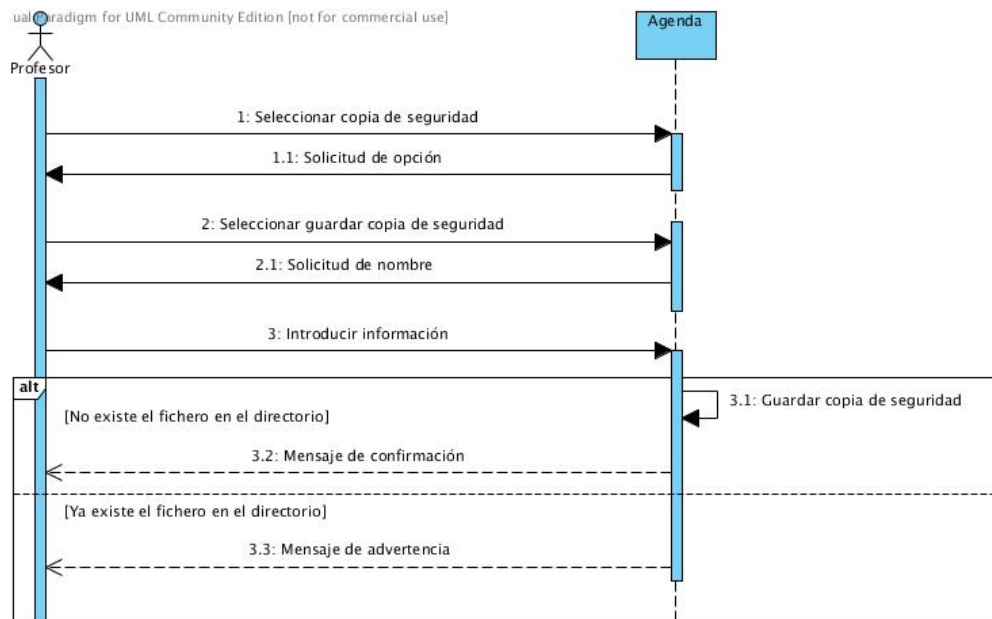
### 2.2.7. Cargar copias de seguridad

En este diagrama se explican los pasos que se llevan a cabo cargar una copia de seguridad en el sistema.



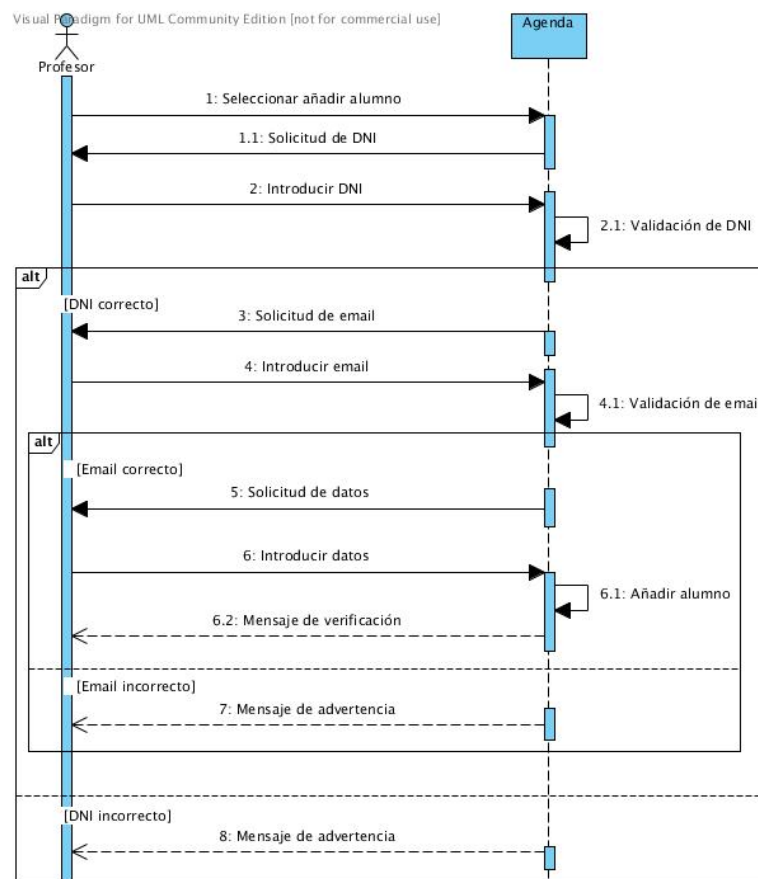
### 2.2.8. Guardar copias de seguridad

En este diagrama se explican los pasos que se llevan a cabo al guardar una copia de seguridad en el sistema.



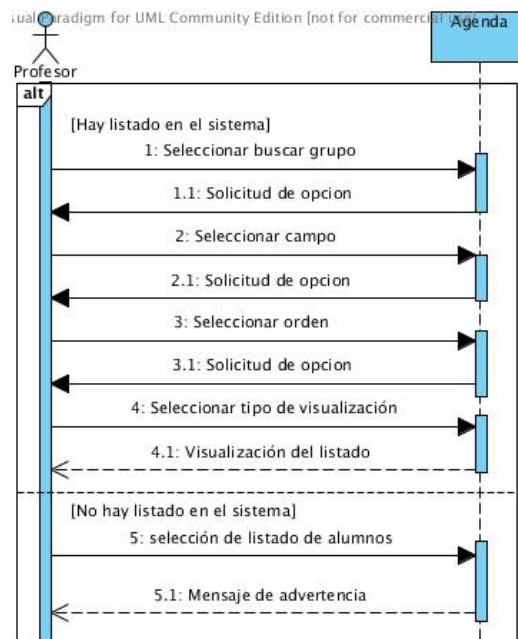
### 2.2.9. Añadir alumno

En este diagrama se explican los pasos que se llevan a cabo al introducir un alumno en el sistema.



### 2.2.10. Buscar grupo

En este diagrama se explican los pasos que se llevan a cabo al buscar por grupo en el sistema.



## 3. Práctica 4

### 3.1. Metodología SCRUM

Es una metodología ágil que se basa en entregas parciales y frecuentes de un producto para obtener resultados con rapidez. Es incremental, cuyos objetivos varían debido a que están poco definidos.

#### 3.1.1. Product Backlog

Contiene todas las funcionalidades ordenadas por prioridades. Es la agrupación de todas las historias de usuario del producto. Ordenados por prioridad de mayor a menor quedaría así.

PRIORIDAD 0	
Login	➤ ID: 9 ➤ Tiempo: 20 minutos
Registrarse	➤ ID: 10 ➤ Tiempo 20 minutos
PRIORIDAD 1	
Insertar alumno	➤ ID: 5 ➤ Tiempo: 30 minutos
Guardar fichero	➤ ID: 11 ➤ Tiempo 1 hora
PRIORIDAD 2	
Guardar copias de seguridad	➤ ID: 6 ➤ Tiempo: 1 hora

Cargar copias de seguridad	➤ ID: 7 ➤ Tiempo 1 hora
Cargar fichero	➤ ID: 12 ➤ Tiempo 1 hora
<b>PRIORIDAD 3</b>	
Listar alumnos	➤ ID: 1 ➤ Tiempo: 1 hora
Eliminar alumno	➤ ID: 4 ➤ Tiempo 40 minutos
Mostrar resultado	➤ ID: 8 ➤ Tiempo 2 horas
<b>PRIORIDAD 4</b>	
Visualizar alumno	➤ ID: 2 ➤ Tiempo: 30 minutos
Editar alumnos	➤ ID: 3 ➤ Tiempo 2 horas
<b>PRIORIDAD 5</b>	
Buscar grupo	➤ ID: 13 ➤ Tiempo: 1 hora
<b>PRIORIDAD 6</b>	
Funciones auxiliares	➤ ID: - ➤ Tiempo: 5 horas
<b>Tiempo total necesario: 17h y 20 min</b>	

### 3.1.2. Sprint Backlog

Funcionalidad a desarrollar en un sprint determinado.

<b>1ª Entrega</b>	
<b>PRIORIDAD 0</b>	
Login	➤ ID: 9 ➤ Tiempo: 20 minutos ➤ Encargado: Gávez Ruz
Registrarse	➤ ID: 10 ➤ Tiempo 20 minutos ➤ Encargado: Gávez Ruz
<b>PRIORIDAD 1</b>	
Insertar alumno	➤ ID: 5 ➤ Tiempo: 30 minutos ➤ Encargado: López Rubias
Guardar fichero	➤ ID: 11 ➤ Tiempo 1 hora ➤ Encargado: López Rubias
<b>PRIORIDAD 2</b>	
Guardar copias de seguridad	➤ ID: 6 ➤ Tiempo: 1 hora ➤ Encargado: López Rubias
Cargar copias de seguridad	➤ ID: 7 ➤ Tiempo 1 hora ➤ Encargado: Gávez Ruz
<b>2ª Entrega</b>	

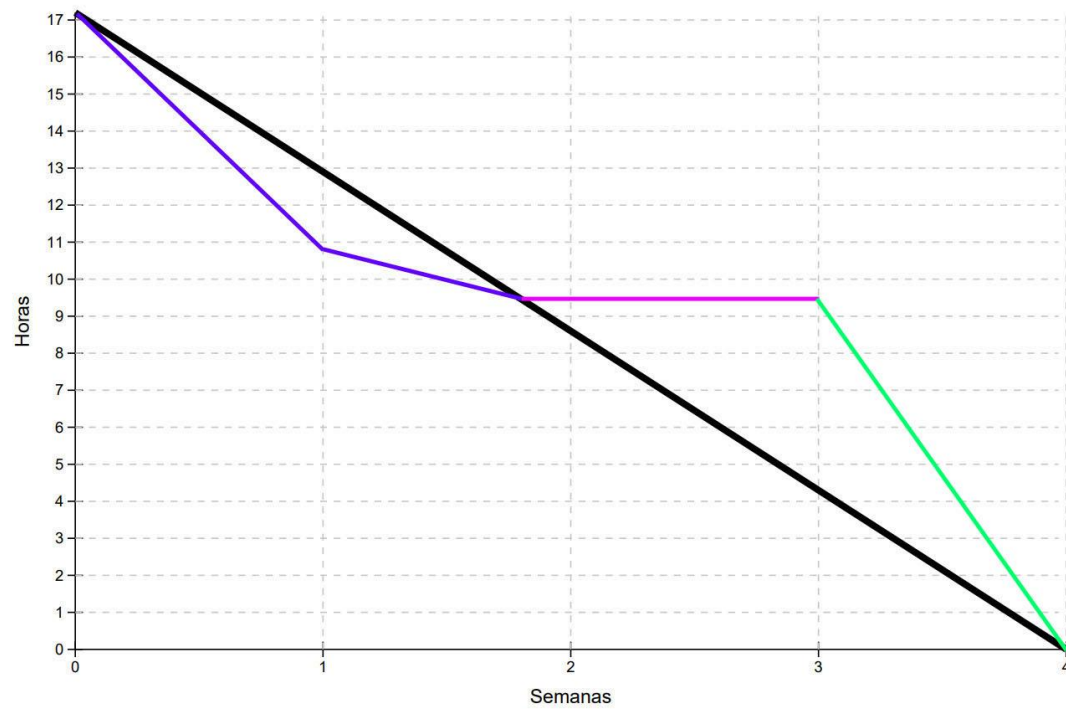
<b>Cargar fichero</b>	<ul style="list-style-type: none"> <li>➤ ID: 12</li> <li>➤ Tiempo 1 hora</li> <li>➤ Encargado: Gávez Ruz</li> </ul>
<b>PRIORIDAD 3</b>	
<b>Listar alumnos</b>	<ul style="list-style-type: none"> <li>➤ ID: 1</li> <li>➤ Tiempo: 1 hora</li> <li>➤ Encargado: López Rubias</li> </ul>
<b>Eliminar alumno</b>	<ul style="list-style-type: none"> <li>➤ ID: 4</li> <li>➤ Tiempo 40 minutos</li> <li>➤ Encargado: López Rubias</li> </ul>
<b>Mostrar resultado</b>	<ul style="list-style-type: none"> <li>➤ ID: 8</li> <li>➤ Tiempo 2 horas</li> <li>➤ Encargado: Gávez Ruz</li> </ul>
<b>PRIORIDAD 4</b>	
<b>Visualizar alumno</b>	<ul style="list-style-type: none"> <li>➤ ID: 2</li> <li>➤ Tiempo: 30 minutos</li> <li>➤ Encargado: López Rubias</li> </ul>
<b>Editar alumnos</b>	<ul style="list-style-type: none"> <li>➤ ID: 3</li> <li>➤ Tiempo 2 horas</li> <li>➤ Encargado: Gávez Ruz</li> </ul>
<b>PRIORIDAD 5</b>	
<b>Buscar grupo</b>	<ul style="list-style-type: none"> <li>➤ ID: 13</li> <li>➤ Tiempo: 1 hora</li> <li>➤ Encargado: López Rubias</li> </ul>
<b>PRIORIDAD 6</b>	
<b>Funciones auxiliares</b>	<ul style="list-style-type: none"> <li>➤ ID: -</li> <li>➤ Tiempo: 5 horas</li> <li>➤ Encargado: Gávez Ruz y López Rubias</li> </ul>
<b>Tiempo total necesario: 17h y 20 min</b>	

### 3.1.3. Burndown chart

Grafico que muestra cantidad de trabajo hecho durante las cuatro semanas sobre la implementación del software solicitado.

- En azul observamos el trabajo realizado durante el primer sprint en un tiempo de dos semanas.
- En rosa lo realizado en una semana, es decir, en la tercera semana de trabajo de programación.
- En verde el resultado final una vez realizado todo en la última semana.

17.20



### 3.2. Matrices de validación

Requisitos funcionales					Casos de uso					
RF-1					CU-1					
RF-2					CU-2					
RF-3					CU-1, CU-8					
RF-4					CU-3					
RF-5					CU-4					
RF-6					CU-5					
RF-7					CU-6, CU-7					
RF-8					CU-2					
RF-9					CU-10					
RF-10					CU-9					

	CU-1	CU-2	CU-3	CU-4	CU-5	CU-6	CU-7	CU-8	CU-9	CU-10
RF-1	X									
RF-2		X								
RF-3	X							X		
RF-4			X							
RF-5				X						
RF-6					X					
RF-7						X	X			
RF-8		X								
RF-9										X
RF-10									X	

## 4. Bibliografía

[Moodle] Ingeniería del software prácticas temas del 1 al 4:

[https://moodle.uco.es/m1819/pluginfile.php/211210/mod\\_resource/content/4/P4\\_Ingenieria\\_Software\\_Presentacion.pdf](https://moodle.uco.es/m1819/pluginfile.php/211210/mod_resource/content/4/P4_Ingenieria_Software_Presentacion.pdf)

[https://moodle.uco.es/m1819/pluginfile.php/194752/mod\\_resource/content/3/P3\\_Ingenieria\\_Software\\_Presentacion.pdf](https://moodle.uco.es/m1819/pluginfile.php/194752/mod_resource/content/3/P3_Ingenieria_Software_Presentacion.pdf)

[https://moodle.uco.es/m1819/pluginfile.php/166887/mod\\_resource/content/1/P2\\_Ingenieria\\_Software\\_Presentacion.pdf](https://moodle.uco.es/m1819/pluginfile.php/166887/mod_resource/content/1/P2_Ingenieria_Software_Presentacion.pdf)

[https://moodle.uco.es/m1819/pluginfile.php/110517/mod\\_resource/content/2/P1\\_Ingenieria\\_Software\\_Presentacion.pdf](https://moodle.uco.es/m1819/pluginfile.php/110517/mod_resource/content/2/P1_Ingenieria_Software_Presentacion.pdf)