

Agenda de Alumnos

Sistema para el registro de alumnos en un centro educativo

Realizado por:

- Noelia Hinojosa Sánchez
- Isabel María Ariza Velasco
- Óscar Godoy Calderón

Contenido

INTRODUCCIÓN	1
DESCRIPCIÓN DEL PROBLEMA.....	1
EXTRACCIÓN DE REQUISITOS	1
HISTORIAS DE USUARIO	4
CASOS DE USO.....	9
DIAGRAMA DE CLASES	20
DIAGRAMAS DE SECUENCIAS.....	20
METODOLOGÍA SCRUM.....	26
PRODUCT BACKLOG	26
SPRINT BACKLOG 1.....	37
SPRINT BACKLOG 2.....	38
BURNDOWN CHART	39
MATRICES DE VALIDACIÓN	40
MATRIZ DE REQUISITOS FUNCIONALES (RF) – CASOS DE USO (CU)	40
MATRIZ DE CASOS DE USO (CU) – CLASES	40

INTRODUCCIÓN

El proyecto que estamos realizando, en las prácticas de Ingeniería del Software, es un programa informático básico que consiste en una Agenda de alumnos donde cada uno de los profesores registrados en el sistema pueden llevar un registro de sus alumnos, guardando de ellos tanto sus datos personales como sus datos académicos. Esta Agenda es común para todos los cursos, almacenando en ella como máximo 150 alumnos.

Para esta última entrega del proyecto, vamos a integrar en un solo archivo las prácticas 2, 3 y 4, en las que se ha estado trabajando sobre este proyecto y en las que se ha llevado a cabo un estudio para la organización del trabajo a realizar y que el proceso sea lo más eficiente posible. Para ello, se han recogido los requisitos que debe cumplir el programa informático, además de la información detallada de cada requisito para la posterior realización de las historias de usuario y los distintos tipos de diagramas, como los de clase, los de casos de uso y de secuencia, que se han llevado a cabo para la definición técnica del problema.

DESCRIPCIÓN DEL PROBLEMA

Se desea realizar una aplicación informática para administrar a los alumnos de un centro educativo, dicho programa solo cuenta con una capacidad de 150 alumnos como máximo, por lo que se podrá emplear para centros pequeños.

Para poder usar el sistema, anteriormente se hace un registro de los profesores que van a acceder a los datos del alumnado concediendo a cada uno de ellos unos permisos distintos dependiendo de su posición (Coordinador o Ayudante).

Las opciones con las que se podrá trabajar serán las de insertado de nuevos alumnos, hasta alcanzar el cupo de la aplicación, buscar datos de dichos alumnos, ya sea por DNI, Apellidos o Equipo, listar todos los alumnos guardados y modificar datos o borrar alumnos. Además los profesores que sean Coordinadores tendrán las opciones de guardar copias de seguridad del listado de alumnos y de cargar dichas copias de seguridad.

EXTRACCIÓN DE REQUISITOS

- **Actores:**
 - **Principal:** Profesor
 - **Secundario:** Alumno

- **Datos del Alumno:**
 - DNI
 - Nombre
 - Apellidos

- Teléfono
- Email
- Dirección
- Curso
- Fecha de Nacimiento
- Equipo
- Líder

- **Datos del Profesor:**

- DNI
- Nombre
- Apellidos
- Teléfono
- Email
- Dirección
- Fecha de Nacimiento
- Usuario
- Contraseña
- Rol

- **Requisitos funcionales:**

- **De prioridad 1:**
 - Buscar Alumno
 - Insertar Alumno
 - Mostrar Listado de Alumnos
 - Guardar Copia de Seguridad
 - Iniciar Sesión Profesor (Login)
 - Registrar Profesor
 - Buscar Profesor
- **De prioridad 2:**
 - Mostrar Alumno
 - Cargar Copia de Seguridad

- **De prioridad 3:**
 - Modificar Alumno
 - Borrar Alumno
- **Requisitos no funcionales:**
 - Programación en C++
 - Documentación en lenguaje Markdown
 - Uso del sistema de control de versiones Git y la plataforma GitHub
 - Fichero binario de copia de seguridad
 - Máximo de alumnos: 150

HISTORIAS DE USUARIO

(ANVERSO)

ID: 001 Buscar Alumno

- El profesor puede consultar uno o varios alumnos de la Agenda, introduciendo en el sistema el DNI, el Apellido o el número del equipo al que pertenece.

Prioridad: 1

(REVERSO)

- El sistema recoge los datos del alumno/s buscado/s.
- Muestra un mensaje de error si el alumno/s no ha sido encontrado.

(ANVERSO)

ID: 002 Insertar Alumno

- El profesor podrá dar de alta a un nuevo alumno indicando de este tanto su información básica personal, así como cierta información académica. Toda la información debe ser ingresada de forma correcta, siendo la mayoría de campos obligatorios.

Prioridad: 2

(REVERSO)

- Da de alta correctamente a un usuario y verifica que sea registrado.
- Muestra un mensaje de error si los campos obligatorios están vacíos.
- Muestra un mensaje de error si se quiere insertar un usuario con un DNI o email ya registrados.
- Muestra un mensaje de error si el DNI, el Teléfono, el Email, la Fecha de Nacimiento, el Curso o el Equipo no tienen el formato correcto.

(ANVERSO)

ID: 003 Modificar Alumno

- El profesor podrá modificar los datos de un alumno. Toda la información debe ser ingresada de forma correcta, siendo la mayoría de campos obligatorios.

Prioridad: 3

(REVERSO)

- Modifica correctamente un usuario y verifica que los cambios se hayan registrado.
- Muestra un mensaje de error si los campos obligatorios están vacíos.
- El DNI no se puede modificar ya que es único para cada persona.
- El email no se puede modificar ya que solo puede haber uno y no se puede repetir.

(ANVERSO)

ID: 004 Borrar Alumno

- El profesor puede borrar un alumno de la Agenda introduciendo su DNI o su Apellido.

Prioridad: 3

(REVERSO)

- Válida la existencia del usuario que se desea borrar.
- Muestra un mensaje satisfactorio cuando el alumno ha sido borrado correctamente.
- Muestra un mensaje de error si no se ha borrado el alumno correctamente.
- Si existen varios alumnos con el mismo Apellido, se muestran todos ellos para elegir el que se desea eliminar.

(ANVERSO)

ID: 005 Mostrar Listado

- Muestra el listado de todos los alumnos, mostrando toda la información guardada de ellos.

Prioridad: 1

(REVERSO)

- Muestra el listado de alumnos por pantalla.
- Muestra un mensaje de error si la agenda está vacía.

(ANVERSO)

ID: 006 Mostrar Alumno

- El profesor podrá ver toda la información de un alumno o varios buscando por el DNI, los Apellidos o el número de equipo al que pertenece.

Prioridad: 2

(REVERSO)

- Muestra todos los datos de uno o varios alumnos, indicando el criterio de búsqueda que desea, por DNI, Apellidos o Equipo.

(ANVERSO)

ID: 007 Guardar Copia de Seguridad

- Hacer una copia de seguridad de todos los alumnos guardados.

Prioridad: 1

(REVERSO)

- El sistema realizará una copia de seguridad de todos los datos de los alumnos almacenados en la agenda guardándolos en un fichero binario.
- Si ya hay una copia de seguridad anterior y se desea hacer otra, se sobrescribe la información del fichero de copia de seguridad.

(ANVERSO)

ID: 008 Cargar Copia de Seguridad

- Poder recuperar la información si ocurre algún fallo.

Prioridad: 5

(REVERSO)

- Cargará el fichero binario de copia de seguridad con todos los datos guardados en la Agenda.
- Se mostrará un mensaje satisfactorio si se realizó correctamente.
- Se mostrará un mensaje de error si ocurrió algún fallo.

(ANVERSO)

ID: 009 Registrar Profesor

- El profesor podrá registrarse en el sistema. Sus datos se guardarán en un fichero externo binario.

Prioridad: 1

(REVERSO)

- Dar de alta a un profesor en el sistema.
- Muestra un mensaje de error si los campos obligatorios están vacíos.
- Muestra un mensaje de error si el profesor ya está registrado, se comprobará según el DNI introducido.

(ANVERSO)

ID: 010 Iniciar Sesión (Login)

- El profesor podrá iniciar sesión en el sistema para trabajar con la agenda.

Prioridad: 1

(REVERSO)

- Abrirá la sesión de un profesor y habilitará las opciones del menú correspondientes con su rol.
- Mostrará un mensaje de error si los datos del login no son correctos.
- Mostrará un mensaje de error si el profesor no está registrado.

(ANVERSO)

ID: 011 Buscar Profesor

- El sistema realiza una búsqueda de un profesor por el DNI.

Prioridad: 1

(REVERSO)

- El sistema recoge los datos del profesor buscado.

CASOS DE USO

BUSCAR ALUMNO

ID: 001

Breve descripción: El sistema busca un alumno en la agenda.

Actor principal: Profesor

Actor secundario: Alumno

Precondiciones:

1. El profesor debe introducir un DNI, un Apellido o un Equipo.

Flujo principal:

1. El caso de uso empieza cuando el profesor desea buscar un alumno.
2. El sistema recoge los datos del alumno/s.

Postcondiciones:

1. Ninguna.

Flujos alternativos:

- 2.a. En caso de estar la agenda vacía, mostrará un mensaje de error.
- 2.b. Si no existe el alumno, el sistema muestra un mensaje de error.

INSERTAR ALUMNO

ID: 002

Breve descripción: El profesor añade un alumno nuevo a la agenda.

Actor principal: Profesor

Actor secundario: Alumno

Precondiciones:

1. El alumno no debe existir en la agenda.
2. No puede haber más de 150 alumnos en la agenda.

Flujo principal:

1. El caso de uso empieza cuando el profesor quiere insertar un nuevo alumno en la agenda.
2. El profesor introduce los datos del alumno.
3. El sistema hace una búsqueda del alumno.

Postcondiciones:

1. El sistema muestra un mensaje satisfactorio.

Flujos alternativos:

- 2.a. Si ya hay 150 alumnos, muestra un mensaje de error.
- 3.a. Si el alumno ya existe, es decir, el DNI ya está guardado, el sistema da la oportunidad de introducir otro alumno o cancelar la operación.
- 3.a. Si el email introducido ya está guardado en la Agenda, el sistema muestra un mensaje de error.
- 3.a. Si intenta añadir un alumno como líder a un equipo que ya lo tiene, el sistema guarda al alumno pero no lo pone como líder.

MODIFICAR ALUMNO

ID: 003

Breve descripción: El profesor modifica los datos del alumno.

Actor principal: Profesor

Actor secundario: Alumno

Precondiciones:

1. El alumno debe existir en el sistema.
2. El profesor debe introducir el DNI o el Apellido.

Flujo principal:

1. El caso de uso empieza cuando el profesor quiere modificar los datos de un alumno.
2. El profesor introduce el DNI o el Apellido del alumno que desea modificar.
3. El sistema recoge los datos del alumno/s.
4. El profesor modifica los datos del alumno.

Postcondiciones:

1. El sistema guarda los datos del alumno.

Flujos alternativos:

- 2.a. Si el alumno no existe, muestra un mensaje de error.
- 4.a. Si intenta añadir a un alumno como líder a un equipo que ya lo tiene asignado, el sistema modifica al alumno pero no lo pone como líder.

BORRAR ALUMNO

ID: 004

Breve descripción: El profesor borra un usuario del sistema.

Actor principal: Profesor

Actor secundario: Alumno

Precondiciones:

1. El profesor debe introducir un DNI o un Apellido.
2. El alumno debe existir en el sistema.

Flujo principal:

1. El caso de uso empieza cuando el profesor desea borrar un alumno.
2. El sistema recoge los datos del alumno/s.
3. El sistema preguntará al profesor si está seguro de borrar los datos del alumno.

Postcondiciones:

1. El sistema borra los datos del alumno.

Flujos alternativos:

- 2.a. En caso de estar la agenda vacía, mostrará un mensaje de error.
- 2.b. Si no existe el alumno, el sistema muestra un mensaje de error.
- 2.c. Si existen dos o más alumnos con el mismo apellido, el sistema muestra cual de ellos se quiere borrar.

MOSTRAR LISTADO

ID: 005

Breve descripción: El sistema muestra la lista de alumnos.

Actor principal: Profesor

Actor secundario: Alumno

Precondiciones:

1. Ninguna.

Flujo principal:

1. El caso de uso empieza cuando el profesor desea mostrar el listado de alumnos.
2. El sistema recoge todos los datos de los alumnos.

Postcondiciones:

1. El sistema muestra la lista por pantalla.

Flujos alternativos:

- 2.a. En caso de estar la agenda vacía, mostrará un mensaje de error.

MOSTRAR ALUMNO

ID: 006

Breve descripción: El sistema muestra la información de uno o varios alumnos.

Actor principal: Profesor

Actor secundario: Alumno

Precondiciones:

1. El profesor debe introducir un DNI, un Apellido o un Equipo.
2. El alumno debe existir en la agenda.

Flujo principal:

1. El caso de uso empieza cuando el profesor desea mostrar uno o varios alumno/s.
2. El sistema recoge los datos del alumno/s.

Postcondiciones:

1. El sistema muestra por pantalla los datos del alumno o alumnos.

Flujos alternativos:

- 2.a. En caso de estar la agenda vacía, mostrará un mensaje de error.
- 2.b. Si no existe el alumno, el sistema muestra un mensaje de error.

GUARDAR COPIA DE SEGURIDAD

ID: 007

Breve descripción: El sistema guarda los datos de todos los alumnos.

Actor principal: Profesor

Actor secundario: Alumno

Precondiciones:

1. La agenda no debe estar vacía.

Flujo principal:

1. El caso de uso empieza cuando el profesor quiere guardar una copia de seguridad de la agenda.
2. El sistema guarda la copia de seguridad.

Postcondiciones:

1. El sistema muestra un mensaje satisfactorio.

Flujos alternativos:

- 2.a. En caso de estar la agenda vacía, mostrará un mensaje de error.
- 2.b. En caso de no haber realizado la copia correctamente muestra un mensaje de error.

CARGAR COPIA DE SEGURIDAD

ID: 008

Breve descripción: El sistema recuperará los datos de los alumnos.

Actor principal: Profesor

Actor secundario: Alumno

Precondiciones:

1. La copia de seguridad debe existir.

Flujo principal:

1. El caso de uso empieza cuando el profesor necesita cargar la copia de seguridad.
2. El sistema carga la copia de seguridad.

Postcondiciones:

1. El sistema muestra un mensaje satisfactorio.

Flujos alternativos:

- 2.a. En caso de no existir la copia de seguridad, mostrará un mensaje de error.
- 2.b. En caso de no haber realizado la carga correctamente, muestra un mensaje de error.

REGISTRAR PROFESOR

ID: 009

Breve descripción: El profesor se registra en el sistema.

Actor principal: Profesor

Actor secundario: Alumno

Precondiciones:

1. El profesor no debe existir en el sistema.

Flujo principal:

1. El caso de uso empieza cuando el profesor quiere registrarse en el sistema.
2. El profesor indica sus datos.
3. El sistema hace una búsqueda del profesor.
4. El sistema guarda en el fichero binario los datos.

Postcondiciones:

1. El sistema muestra un mensaje satisfactorio.

Flujos alternativos:

- 2.a. Muestra un mensaje de error si los campos están vacíos o son incorrectos.
- 3.a. Si el profesor ya existe, el sistema muestra un mensaje indicándolo.
- 4.a. Muestra un mensaje de error si el fichero no se ha guardado de forma correcta.

INICIAR SESIÓN (LOGIN)

ID: 010

Breve descripción: El profesor inicia sesión en el sistema.

Actor principal: Profesor

Actor secundario: Alumno

Precondiciones:

1. El profesor debe estar guardado en el fichero binario.

Flujo principal:

1. El caso de uso empieza cuando el profesor quiere iniciar sesión en el sistema.
2. El sistema hace una búsqueda de las credenciales del profesor.
3. El sistema abre la sesión.

Postcondiciones:

1. Ninguna.

Flujos alternativos:

- 2.a. Muestra un mensaje de error si los campos están vacíos o son incorrectos.
- 2.b. Si el profesor no se encuentra, es decir, no está guardado en el fichero, el sistema muestra un mensaje de error.

BUSCAR PROFESOR

ID: 011

Breve descripción: El sistema busca un profesor en el fichero binario.

Actor principal: Profesor

Actor secundario: Alumno

Precondiciones:

1. Ninguna.

Flujo principal:

1. El caso de uso empieza cuando el profesor quiere registrarse en el sistema o iniciar sesión.
2. El sistema recoge los datos del profesor.

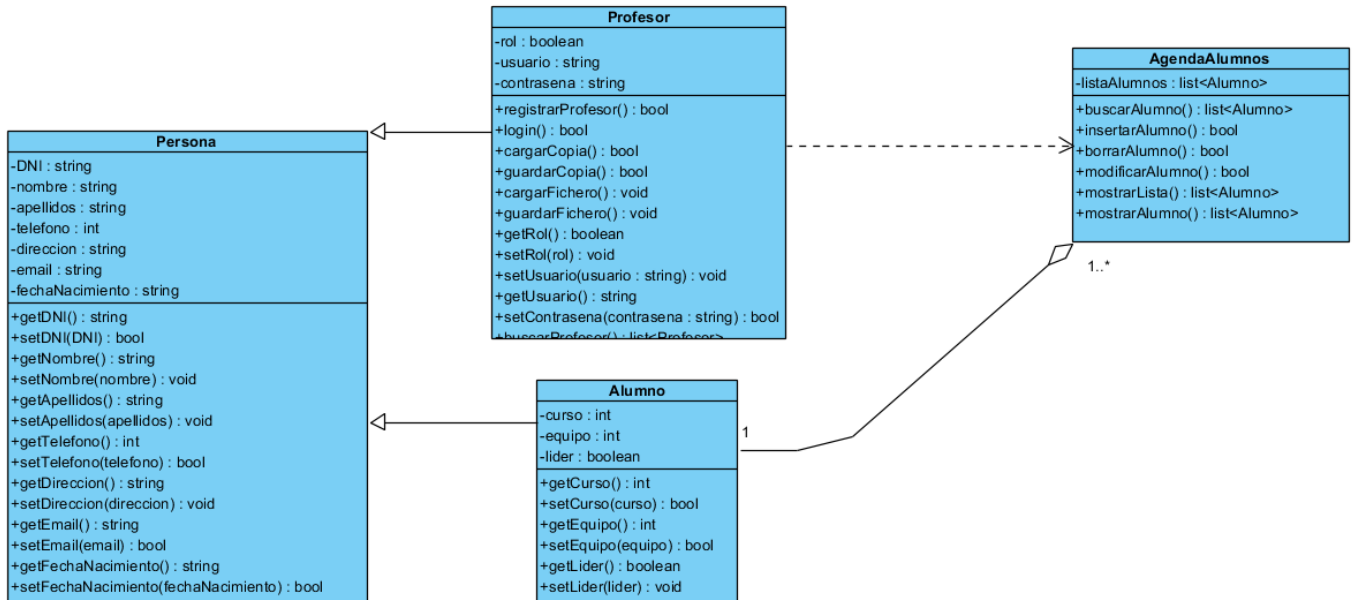
Postcondiciones:

2. El sistema realiza la operación indicada.

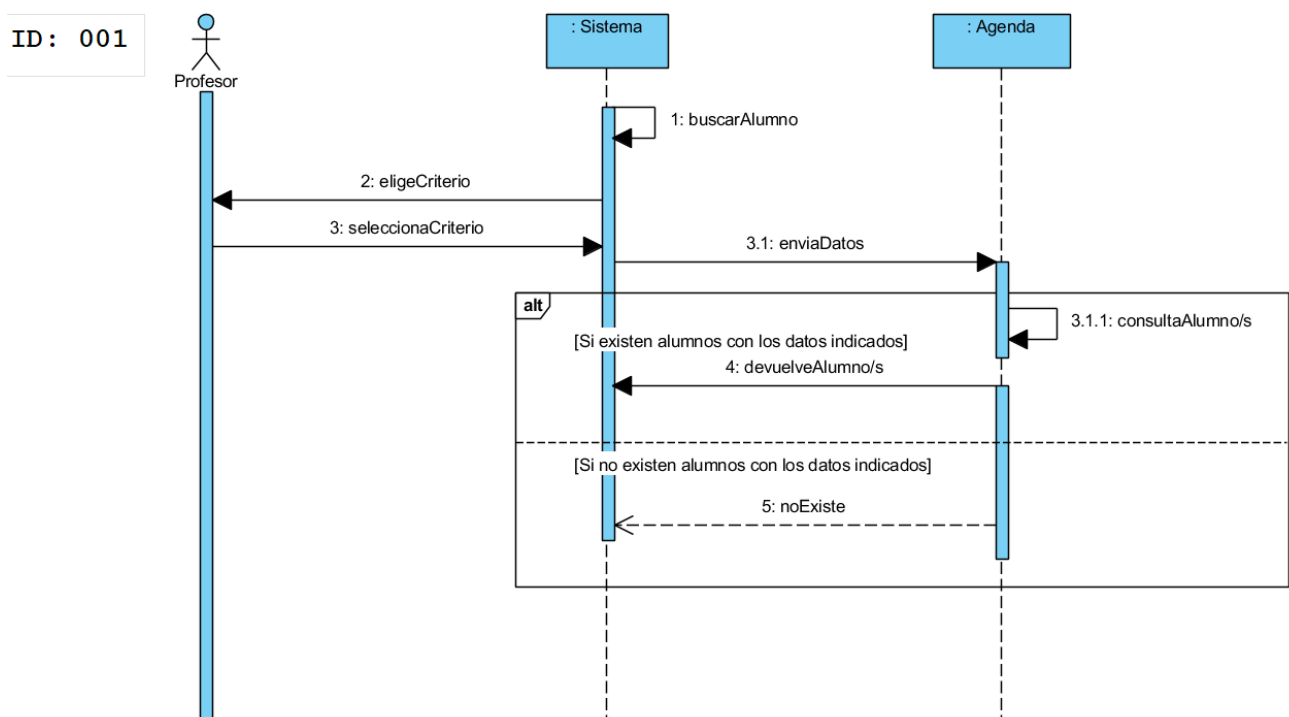
Flujos alternativos:

- 2.a. Si no se encuentra al profesor, en caso de intentar registrarlo mostrará un mensaje satisfactorio, en caso de intentar iniciar sesión mostrará un mensaje de error.

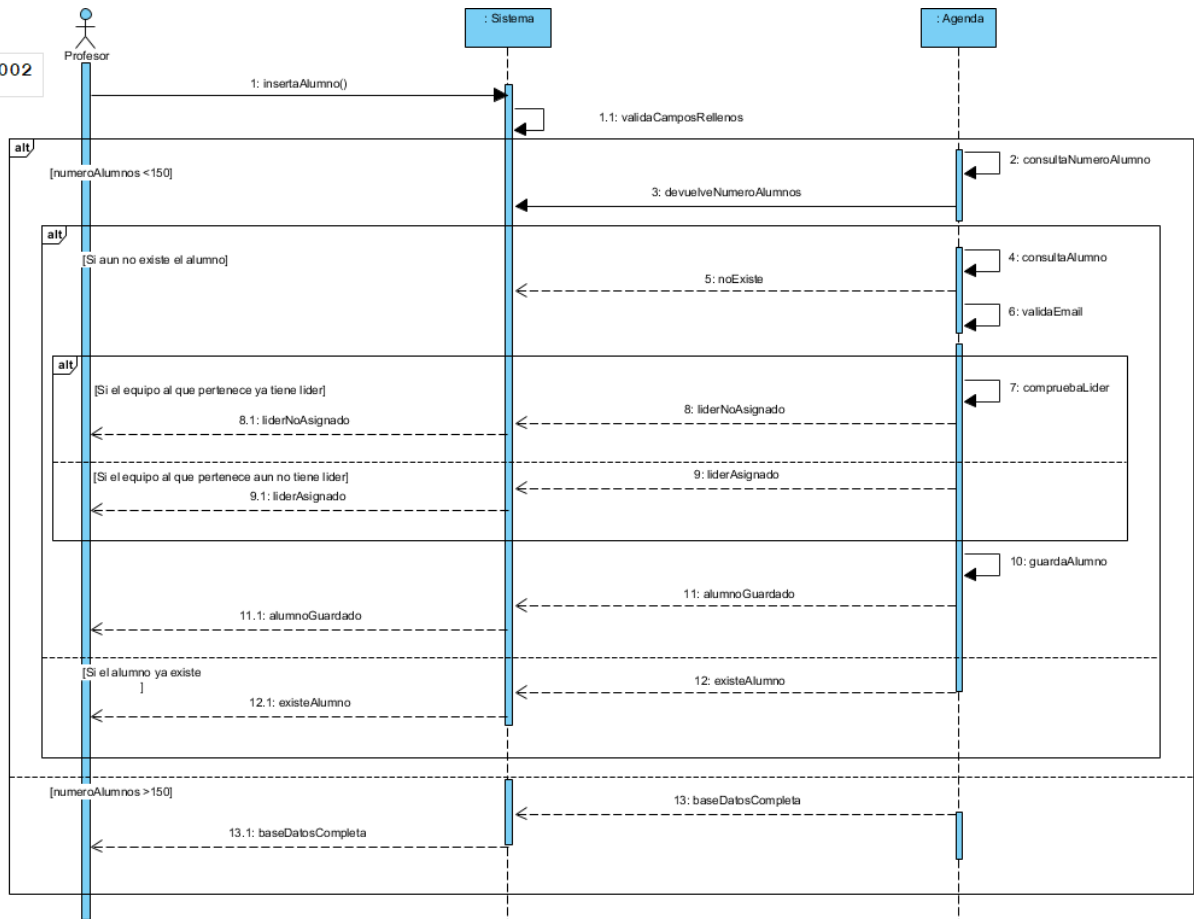
DIAGRAMA DE CLASES



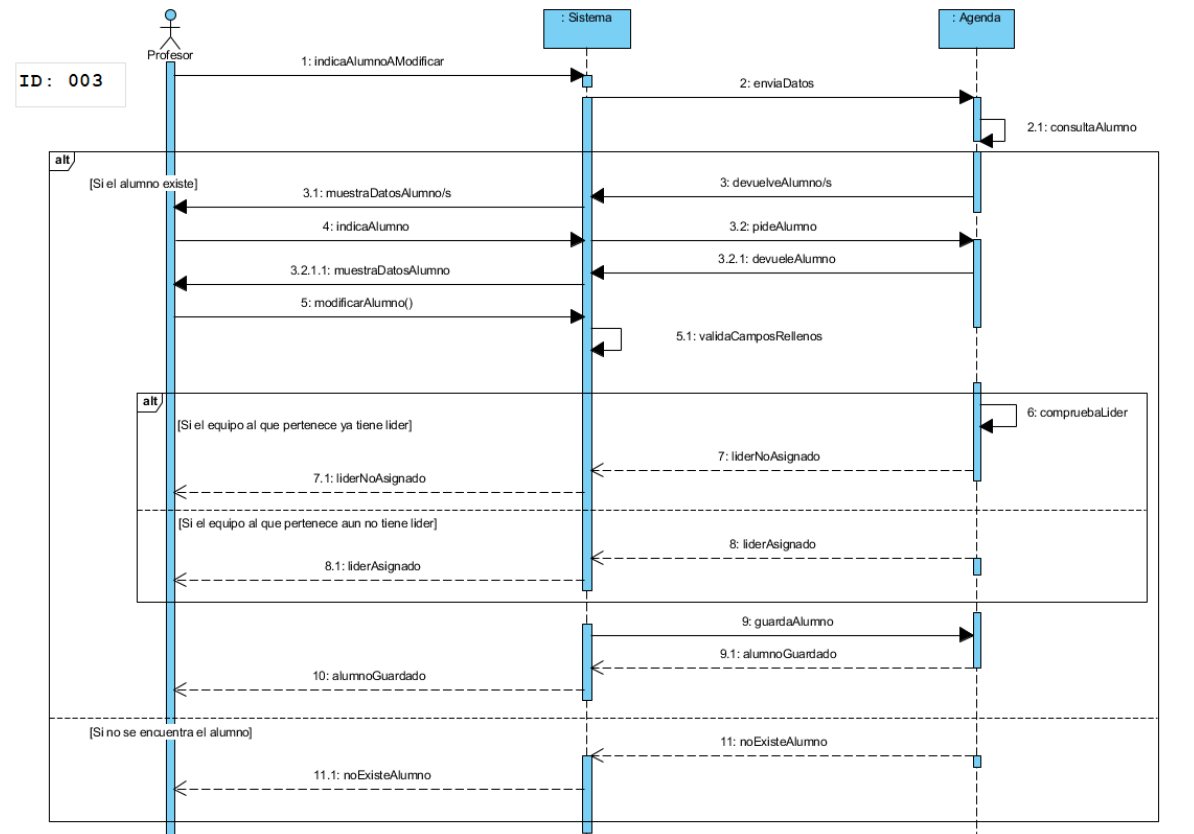
DIAGRAMAS DE SECUENCIAS



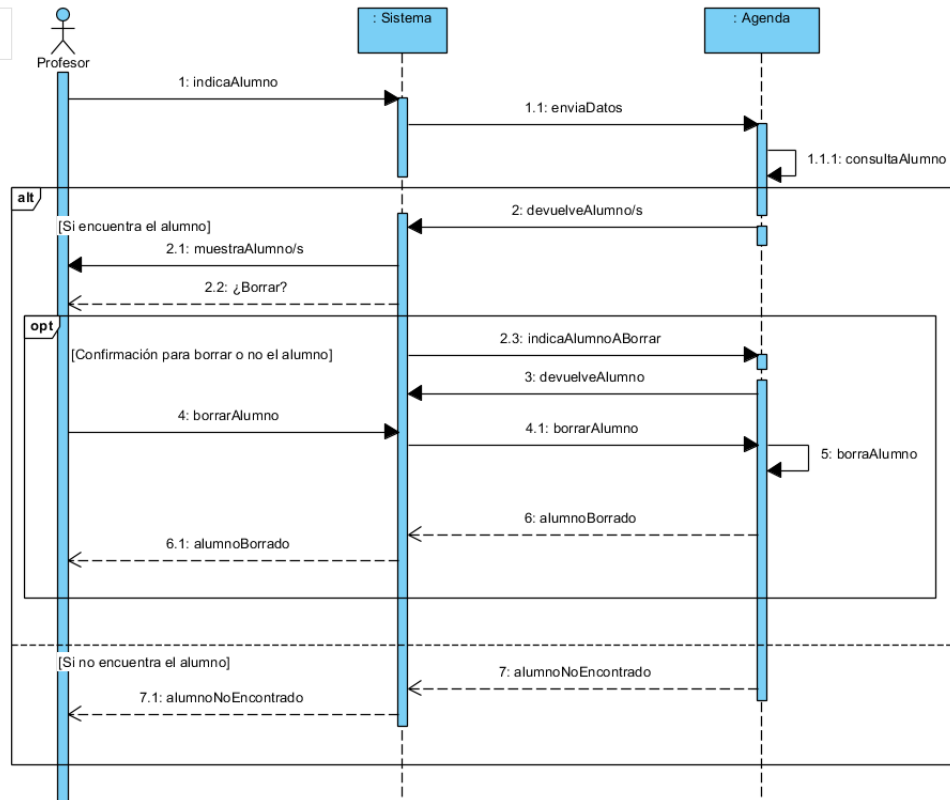
ID: 002



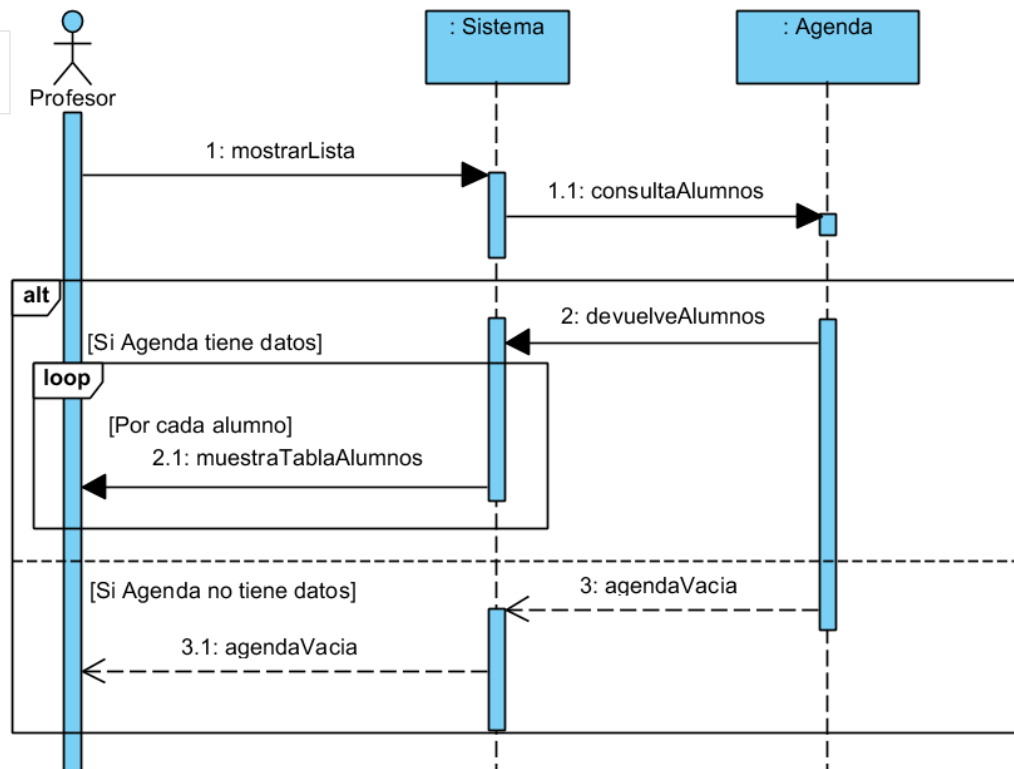
ID: 003



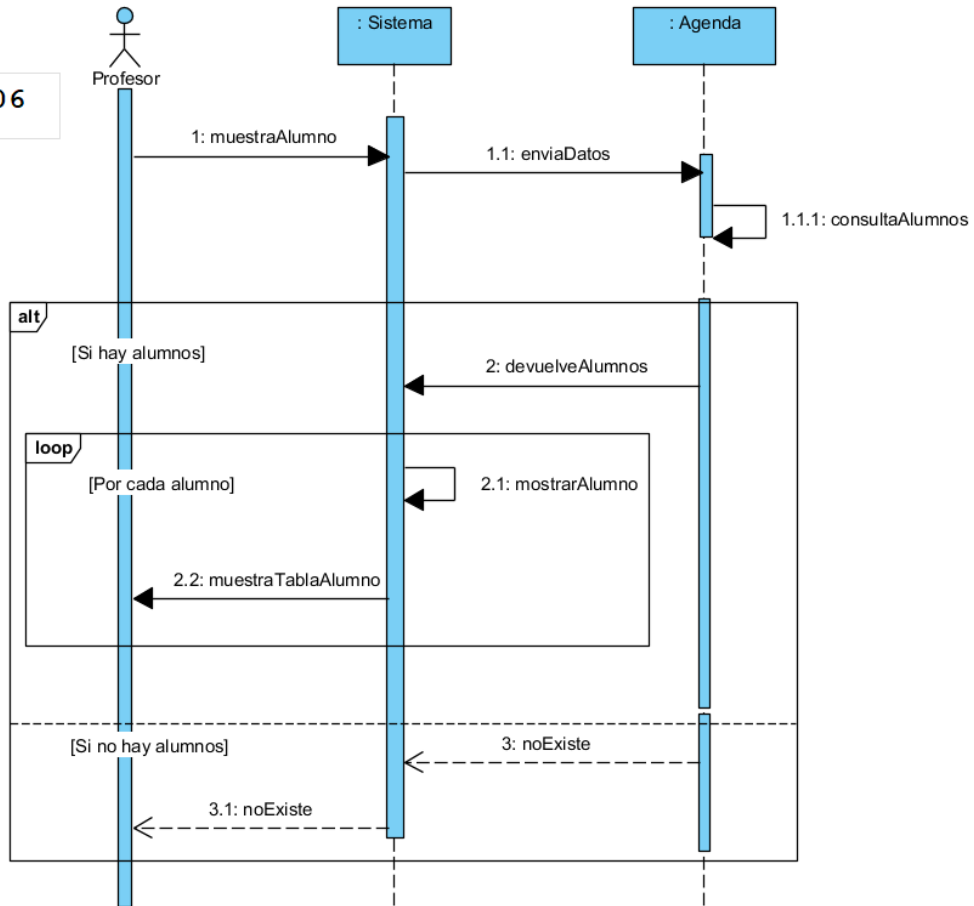
ID: 004



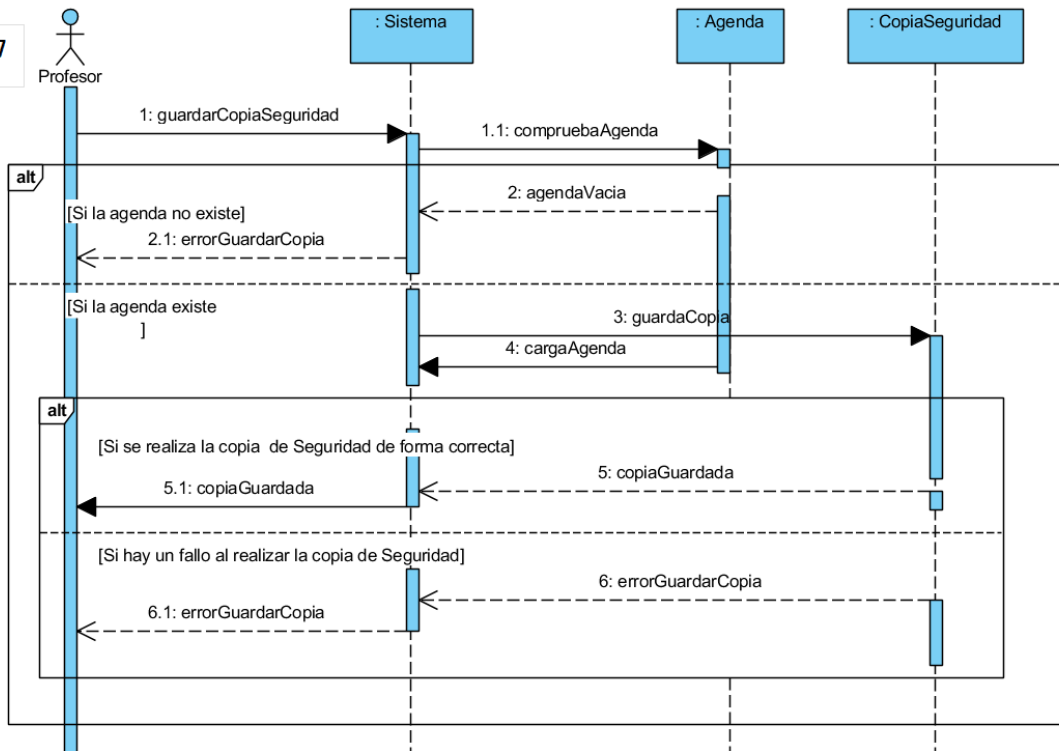
ID: 005



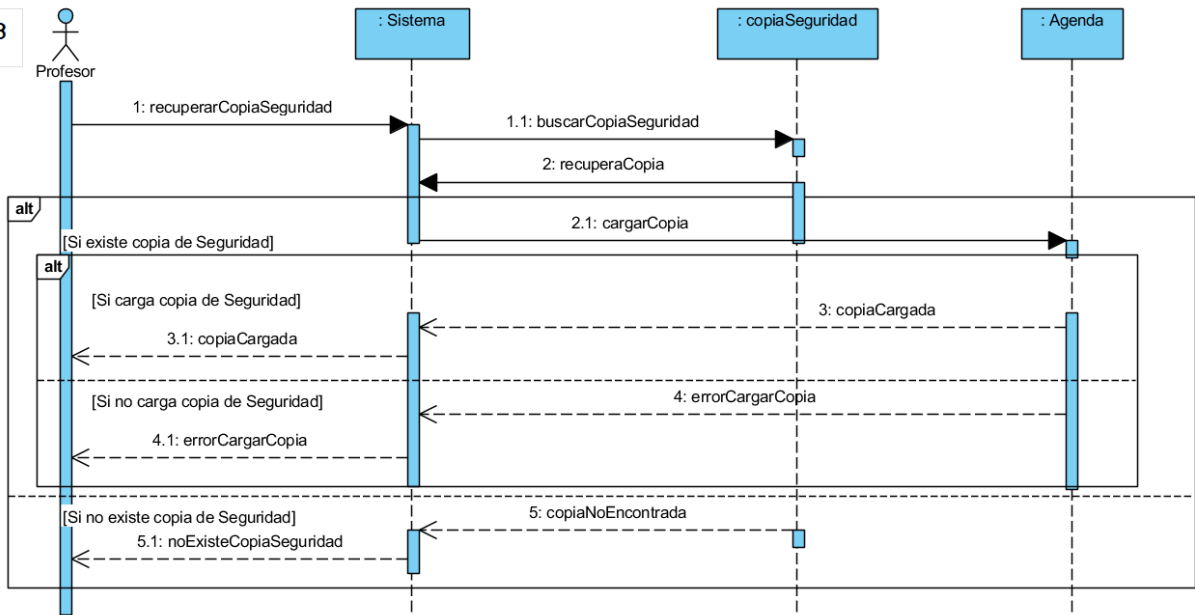
ID: 006



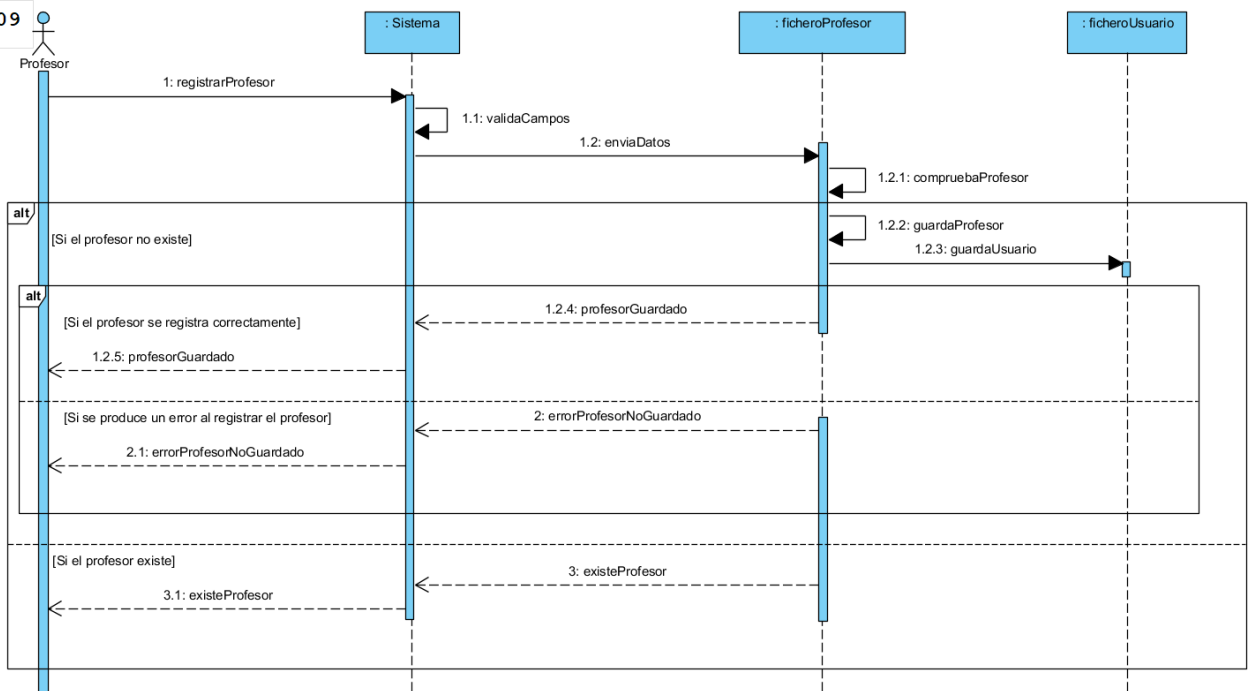
ID: 007



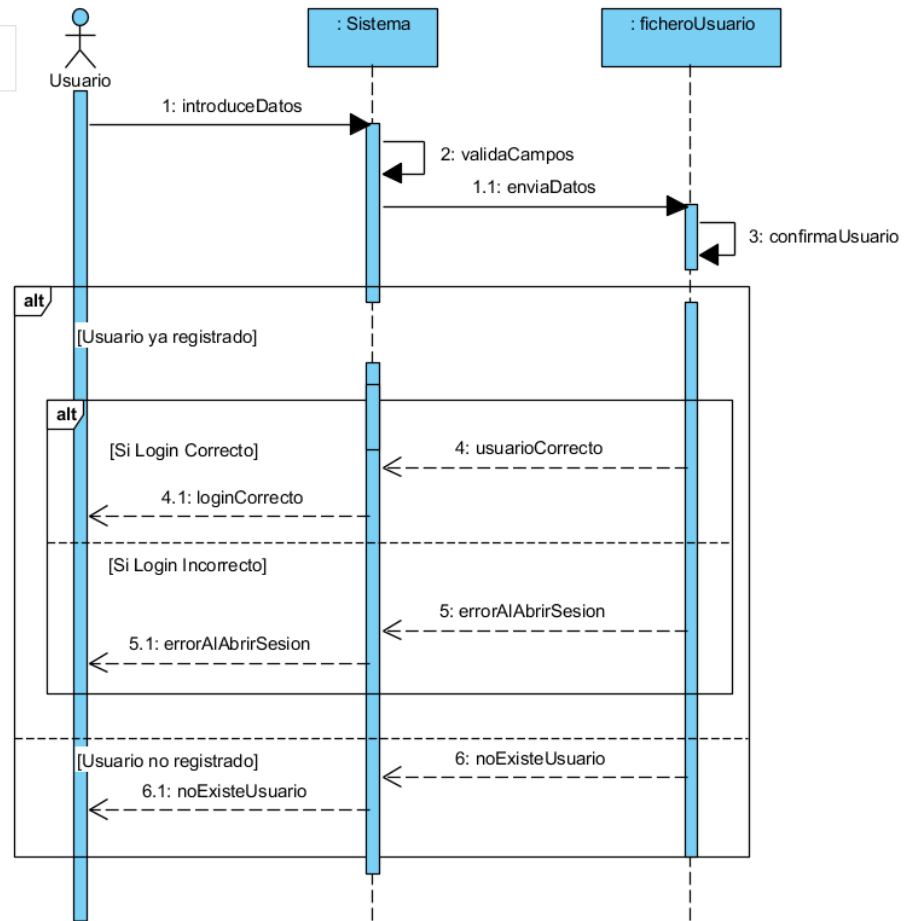
ID: 008



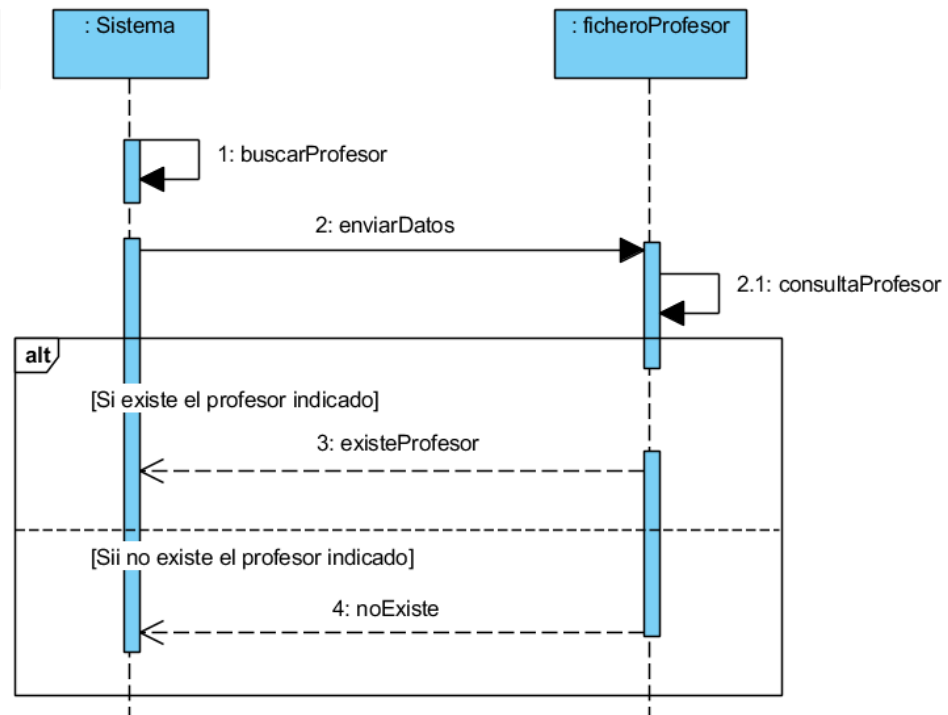
ID: 009



ID: 010



ID: 011



METODOLOGÍA SCRUM

PRODUCT BACKLOG

(ANVERSO)

ID: 001 Buscar Alumno

- El profesor puede consultar uno o varios alumnos de la Agenda, introduciendo en el sistema el DNI, el Apellido o el número del equipo al que pertenece.

Prioridad: 1

Prioridad de implementación: 0

Tiempo estimado de implementación: 3 horas

(REVERSO)

- El sistema recoge los datos del alumno/s buscado/s.
- Muestra un mensaje de error si el alumno/s no ha sido encontrado.

(ANVERSO)

ID: 011 Buscar Profesor

- El sistema realiza una búsqueda de un profesor por el DNI.

Prioridad: 1

Prioridad de implementación: 0

Tiempo estimado de implementación: 1 horas y 30 minutos

(REVERSO)

- El sistema recoge los datos del profesor buscado.

(ANVERSO)

ID: 002 Insertar Alumno

- El profesor podrá dar de alta a un nuevo alumno indicando de este tanto su información básica personal, así como cierta información académica. Toda la información debe ser ingresada de forma correcta, siendo la mayoría de campos obligatorios.

Prioridad: 2

Prioridad de implementación: 1

Tiempo estimado de implementación: 2 horas

(REVERSO)

- Da de alta correctamente a un usuario y verifica que sea registrado.
- Muestra un mensaje de error si los campos obligatorios están vacíos.
- Muestra un mensaje de error si se quiere insertar un usuario con un DNI o email ya registrados.
- Muestra un mensaje de error si el DNI, el Teléfono, el Email, la Fecha de Nacimiento, el Curso o el Equipo no tienen el formato correcto.

(ANVERSO)

ID: 009 Registrar Profesor

- El profesor podrá registrarse en el sistema. Sus datos se guardarán en un fichero externo binario.

Prioridad: 1

Prioridad de implementación: 1

Tiempo estimado de implementación: 2 horas

(REVERSO)

- Dar de alta a un profesor en el sistema.
- Muestra un mensaje de error si los campos obligatorios están vacíos.
- Muestra un mensaje de error si el profesor ya está registrado, se comprobará según el DNI introducido.

(ANVERSO)

ID: 005 Mostrar Listado

- Muestra el listado de todos los alumnos, mostrando toda la información guardada de ellos.

Prioridad: 1

Prioridad de implementación: 2

Tiempo estimado de implementación: 1 horas y 30 minutos

(REVERSO)

- Muestra el listado de alumnos por pantalla.
- Muestra un mensaje de error si la agenda está vacía.

(ANVERSO)

ID: 006 Mostrar Alumno

- El profesor podrá ver toda la información de un alumno o varios buscando por el DNI, los Apellidos o el número de equipo al que pertenece.

Prioridad: 2

Prioridad de implementación: 2

Tiempo estimado de implementación: 1 horas y 30 minutos

(REVERSO)

- Muestra todos los datos de uno o varios alumnos, indicando el criterio de búsqueda que desea, por DNI, Apellidos o Equipo.

(ANVERSO)

ID: 010 Iniciar Sesión (Login)

- El profesor podrá iniciar sesión en el sistema para trabajar con la agenda.

Prioridad: 1

Prioridad de implementación: 2

Tiempo estimado de implementación: 3 horas

(REVERSO)

- Abrirá la sesión de un profesor y habilitará las opciones del menú correspondientes con su rol.
- Mostrará un mensaje de error si los datos del login no son correctos.
- Mostrará un mensaje de error si el profesor no está registrado.

(ANVERSO)

ID: 003 Modificar Alumno

- El profesor podrá modificar los datos de un alumno. Toda la información debe ser ingresada de forma correcta, siendo la mayoría de campos obligatorios.

Prioridad: 3

Prioridad de implementación: 3

Tiempo estimado de implementación: 1 horas

(REVERSO)

- Modifica correctamente un usuario y verifica que los cambios se hayan registrado.
- Muestra un mensaje de error si los campos obligatorios están vacíos.
- El DNI no se puede modificar ya que es único para cada persona.
- El email no se puede modificar ya que solo puede haber uno y no se puede repetir.

(ANVERSO)

ID: 004 Borrar Alumno

- El profesor puede borrar un alumno de la Agenda introduciendo su DNI o su Apellido.

Prioridad: 3

Prioridad de implementación: 3

Tiempo estimado de implementación: 2 horas

(REVERSO)

- Válida la existencia del usuario que se desea borrar.
- Muestra un mensaje satisfactorio cuando el alumno ha sido borrado correctamente.
- Muestra un mensaje de error si no se ha borrado el alumno correctamente.
- Si existen varios alumnos con el mismo Apellido, se muestran todos ellos para elegir el que se desea eliminar.

(ANVERSO)

ID: 007 Guardar Copia de Seguridad

- Hacer una copia de seguridad de todos los alumnos guardados.

Prioridad: 1

Prioridad de implementación: 4

Tiempo estimado de implementación: 2 horas

(REVERSO)

- El sistema realizará una copia de seguridad de todos los datos de los alumnos almacenados en la agenda guardándolos en un fichero binario.
- Si ya hay una copia de seguridad anterior y se desea hacer otra, se sobrescribe la información del fichero de copia de seguridad.

(ANVERSO)

ID: 008 Cargar Copia de Seguridad

- Poder recuperar la información si ocurre algún fallo.

Prioridad: 5

Prioridad de implementación: 5

Tiempo estimado de implementación: 2 horas

(REVERSO)

- Cargará el fichero binario de copia de seguridad con todos los datos guardados en la Agenda.
- Se mostrará un mensaje satisfactorio si se realizó correctamente.
- Se mostrará un mensaje de error si ocurrió algún fallo.

FUNCIONES AUXILIARES

- **comprobarExistenciaDNI**
 - **Prioridad de implementación: 0**
 - **Tiempo estimado de implementación: 1 hora**
- **getCurso**
 - **Prioridad de implementación: 1**
 - **Tiempo estimado de implementación: 5 minutos**
- **setCurso**
 - **Prioridad de implementación: 1**
 - **Tiempo estimado de implementación: 5 minutos**
- **getEquipo**
 - **Prioridad de implementación: 1**
 - **Tiempo estimado de implementación: 5 minutos**

- **setEquipo**
 - Prioridad de implementación: 1
 - Tiempo estimado de implementación: 5 minutos
- **getLider**
 - Prioridad de implementación: 1
 - Tiempo estimado de implementación: 5 minutos
- **setLider**
 - Prioridad de implementación: 1
 - Tiempo estimado de implementación: 5 minutos
- **getDNI**
 - Prioridad de implementación: 1
 - Tiempo estimado de implementación: 5 minutos
- **setDNI**
 - Prioridad de implementación: 1
 - Tiempo estimado de implementación: 30 minutos
- **getNombre**
 - Prioridad de implementación: 1
 - Tiempo estimado de implementación: 5 minutos
- **setNombre**
 - Prioridad de implementación: 1
 - Tiempo estimado de implementación: 5 minutos

- **getApellidos**
 - Prioridad de implementación: 1
 - Tiempo estimado de implementación: 5 minutos
- **setApellidos**
 - Prioridad de implementación: 1
 - Tiempo estimado de implementación: 5 minutos
- **getTelefono**
 - Prioridad de implementación: 1
 - Tiempo estimado de implementación: 5 minutos
- **setTelefono**
 - Prioridad de implementación: 1
 - Tiempo estimado de implementación: 30 minutos
- **getDireccion**
 - Prioridad de implementación: 1
 - Tiempo estimado de implementación: 5 minutos
- **setDireccion**
 - Prioridad de implementación: 1
 - Tiempo estimado de implementación: 5 minutos
- **getEmail**
 - Prioridad de implementación: 1
 - Tiempo estimado de implementación: 5 minutos

- **setEmail**
 - Prioridad de implementación: 1
 - Tiempo estimado de implementación: 30 minutos
- **getFechaNacimiento**
 - Prioridad de implementación: 1
 - Tiempo estimado de implementación: 5 minutos
- **setFechaNacimiento**
 - Prioridad de implementación: 1
 - Tiempo estimado de implementación: 30 minutos
- **getRol**
 - Prioridad de implementación: 1
 - Tiempo estimado de implementación: 5 minutos
- **setRol**
 - Prioridad de implementación: 1
 - Tiempo estimado de implementación: 5 minutos
- **getUsuario**
 - Prioridad de implementación: 1
 - Tiempo estimado de implementación: 5 minutos
- **setUsuario**
 - Prioridad de implementación: 1
 - Tiempo estimado de implementación: 30 minutos

- **getContrasena**
 - Prioridad de implementación: 1
 - Tiempo estimado de implementación: 5 minutos

- **setContrasena**
 - Prioridad de implementación: 1
 - Tiempo estimado de implementación: 30 minutos

- **comprobarLider**
 - Prioridad de implementación: 1
 - Tiempo estimado de implementación: 1 hora

- **comprobarEmail**
 - Prioridad de implementación: 1
 - Tiempo estimado de implementación: 1 hora

- **escribirDatos**
 - Prioridad de implementación: 1
 - Tiempo estimado de implementación: 2 horas

- **leerlinea**
 - Prioridad de implementación: 1
 - Tiempo estimado de implementación: 1 hora

- **letraDNI**
 - Prioridad de implementación: 1
 - Tiempo estimado de implementación: 30 minutos

- **verificaDNI**
 - Prioridad de implementación: 1
 - Tiempo estimado de implementación: 1 hora
- **compruebaFecha**
 - Prioridad de implementación: 1
 - Tiempo estimado de implementación: 1 hora
- **validarEmail**
 - Prioridad de implementación: 1
 - Tiempo estimado de implementación: 1 hora
- **mostrarListado**
 - Prioridad de implementación: 2
 - Tiempo estimado de implementación: 1 hora
- **modificarDatos**
 - Prioridad de implementación: 3
 - Tiempo estimado de implementación: 1 hora y 30 minutos
- **convertirBool**
 - Prioridad de implementación: 4
 - Tiempo estimado de implementación: 30 minutos
- **convertirBoolLider**
 - Prioridad de implementación: 4
 - Tiempo estimado de implementación: 30 minutos

- **convertirStringLider**
 - **Prioridad de implementación:** 4
 - **Tiempo estimado de implementación:** 30 minutos

SPRINT BACKLOG 1

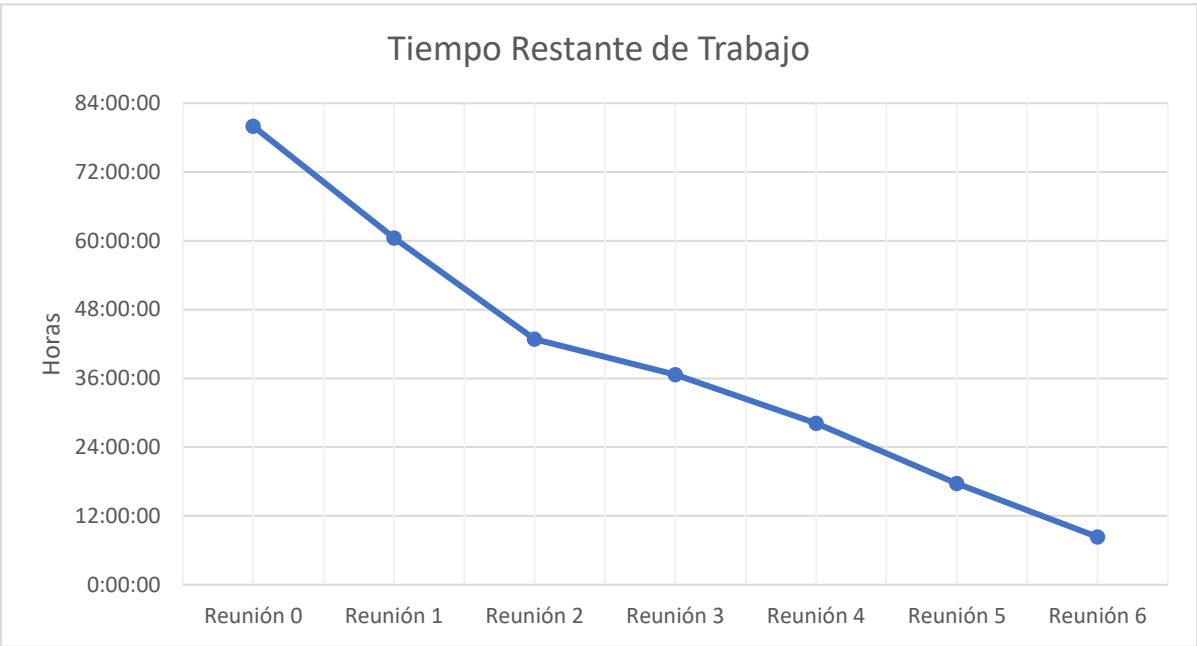
- **Isabel María Ariza Velasco**, un total de 15 horas y 30 minutos
 - **5 horas:** Dejar en funcionamiento la función de *Buscar Alumno*.
 - **6 horas:** Dejar en funcionamiento la opción de *Insertar Alumnos* en la Agenda, incluyendo la función auxiliar de *EscribirDatos*.
 - **1 hora:** Opción de *Guardar Copia de Seguridad* de la Agenda.
 - **2 horas:** Funciones auxiliares para las comprobaciones: *comprobarExistenciaDNI*, *comprobarLider*, *comprobarEmail*, *compruebaFecha*.
 - **30 minutos:** Funciones auxiliares para convertir cadenas: *convertirBoolLider*, *convertirStringLider*.
 - **10 minutos:** Menú del criterio de búsqueda.
 - **35 minutos:** Funciones auxiliares: *verificaDNI*, *validarEmail*, *letraDNI*.
 - **15 minutos:** Algunos métodos set: *setDNI*, *setEmail*, *setFechaNacimiento*.
- **Óscar Godoy Calderón**, un total de 15 horas y 45 minutos
 - **10 minutos:** Dejar en funcionamiento la *Carga de una Copia de Seguridad*.
 - **3 horas y 30 minutos:** La opción *Mostrar Listado* terminada, incluyendo la función auxiliar de *mostrarLista*.
 - **7 horas:** Opción *Modificar Alumno* terminada, incluyendo la función auxiliar de *modificarDatos*.
 - **5 horas:** Opción *Borrar Alumno* en funcionamiento.
 - **5 minutos:** Función auxiliar de *leerlinea* terminada.
- **Noelia Hinojosa Sánchez**, un total de 12 horas y 5 minutos
 - **1 hora y 25 minutos:** Algunos métodos get y set: *getDNI*, *getNombre*, *getApellidos*, *getTelefono*, *getDireccion*, *getEmail*, *getFechaNacimiento*, *getRol*, *getUsuario*, *getContrasena*, *setNombre*, *setApellidos*, *setTelefono*, *setDireccion*, *setRol*, *setUsuario*, *setContrasena*.
 - **1 hora y 30 minutos:** Función para la búsqueda de un profesor guardado finalizada.
 - **2 horas:** Función para añadir un nuevo profesor al sistema terminada.

- **1 hora:** Función que muestra un alumno o alumnos concretos finalizada.
- **1 hora:** Función para el inicio de la sesión de un profesor concreto en funcionamiento.
- **10 minutos:** Dejar en funcionamiento los menús de la aplicación, tanto el principal como el secundario.
- **5 horas:** Función *MAIN* de la aplicación terminado para probar definitivamente el programa.

SPRINT BACKLOG 2

- **Isabel María Ariza Velasco**, un total de
 - **6 horas:** Investigación y pruebas de la interfaz gráfica en QT Creator.
 - **3 horas y 30 minutos:** Agregarle a las funciones anteriormente realizadas la interfaz gráfica.
 - **3 horas:** Correcciones del código.
- **Óscar Godoy Calderón**, un total de
 - **1 hora:** Corrección de errores en la función de *Borrar Alumno*.
 - **3 horas:** Agregarle a las funciones anteriormente realizadas la interfaz gráfica.
 - **1 hora y 30 minutos:** Correcciones del código.
- **Noelia Hinojosa Sánchez**, un total de
 - **1 hora y 30 minutos:** Arreglos en el *MAIN* para la prueba del programa.
 - **4 horas:** Agregarle a las funciones anteriormente realizadas la interfaz gráfica.
 - **3 horas y 30 minutos:** Correcciones del código.

BURNDOWN CHART



MATRICES DE VALIDACIÓN

MATRIZ DE REQUISITOS FUNCIONALES (RF) – CASOS DE USO (CU)

	RF 001 buscar alumno	RF 002 insertar alumno	RF 003 modificar alumno	RF 004 borrar alumno	RF 005 mostrar lista	RF 006 mostrar alumno	RF 007 guardar copia	RF 008 cargar copia	RF 009 registrar profesor	RF 010 login	RF 011 buscar profesor
Caso Uso 001 buscarAlumno	X										
Caso Uso 002 insertarAlumno	X	X									
Caso Uso 003 modificarAlumno	X		X			X					
Caso Uso 004 borrarAlumno	X			X		X					
Caso Uso 005 mostrarLista					X						
Caso Uso 006 mostrarAlumno	X					X					
Caso Uso 007 guardarCopia							X				
Caso Uso 008 cargarCopia								X			
Caso Uso 009 registrarUsuario									X		X
Caso Uso 010 login										X	X
Caso Uso 011 buscarProfesor											X

MATRIZ DE CASOS DE USO (CU) – CLASES

	CL. Profesor	CL. Alumno	CL. AgendaAlumnos
Caso Uso 001 buscarAlumno		X	X
Caso Uso 002 insertarAlumno		X	X
Caso Uso 003 modificarAlumno		X	X
Caso Uso 004 borrarAlumno		X	X
Caso Uso 005 mostrarLista		X	X
Caso Uso 006 mostrarAlumno		X	X
Caso Uso 007 guardarCopia	X		
Caso Uso 008 cargarCopia	X		
Caso Uso 009 registrarUsuario	X		
Caso Uso 010 login	X		
Caso Uso 011 buscarProfesor	X		