|  |  |
| --- | --- |
| logo | Universidad de Córdoba |
|  |  |
|  |  |
|  |  |
| GRADO DE INGENIERÍA INFORMÁTICA | |
| INGENIERÍA DEL SOFTWARE | |
|  | |

|  |
| --- |
|  |
| DOCUMENTACIÓN PROCESO SCRUM DESARROLLO Appgestionalumnos g-36 |
|  |
|  |
| El objeto del presente documento es presentar la documentación del proceso de desarrollo basado en la metodología ágil SCRUM de la aplicación AppGestionAlumnos encargado por el profesorado de las prácticas de la asignatura Ingeniería del Software de 2º curso del Grado de Informática de la Universidad de Córdoba. |
|  |

|  |
| --- |
|  |
| Grupo 36, autores:  Cantero Alén, Rafael (Participa)  Freire Caballero, Carlos (Participa)  Herrera Poch, Fernando (Participa) |
| *Fecha: 22/12/2018* |
|  |

Índice de contenidos

1. [análisis y diseño del problema 5](#_Toc533099709)

[1.1 Definición del Problema 5](#_Toc533099710)

[1.2 Extracción de Requisitos 5](#_Toc533099711)

[1.3 Historias de Usuario 5](#_Toc533099712)

[1.4 Casos de Uso 5](#_Toc533099713)

[1.5 Diagrama de Clases 6](#_Toc533099714)

[1.6 Diagramas de Secuencia 6](#_Toc533099715)

1. [Desarrollo de la aplicación – metodología scrum 7](#_Toc533099716)

[2.1 Introducción 7](#_Toc533099717)

[2.2 Product BackLog 7](#_Toc533099718)

[2.3 Sprints Review meeting 7](#_Toc533099719)

[2.4 Sprint retrospective meeting 8](#_Toc533099720)

1. [cierre del sprint 9](#_Toc533099721)

[3.1 Pruebas de software 9](#_Toc533099722)

[3.2 Verificación de Requisitos 10](#_Toc533099723)

[3.2.1 Matrices de Verificación. 10](#_Toc533099724)

1. [Bibliografía y referencias Web 12](#_Toc533099725)

[Bibliografía 12](#_Toc533099726)

Índice de figuras

**Ilustración 1 Historia Uso 000 Acceso a la aplicación** 8

**Ilustración 2 Historia Uso 001 Insertar alumno** 9

**Ilustración 3 Historia Uso 002 Buscar alumno** 9

**Ilustración 4 Historia Uso 002 Buscar alumno** 9

**Ilustración 5 Matriz de Verificación Requisitos Funcionales** 18

**Ilustración 6 Matriz de Verificación Clases** 19

Índice de tablas

**Tabla 1 Requisitos funcionales** 7

|  |  |
| --- | --- |
|  |  |
|  | análisis y diseño del problema |
|  |  |

# Definición del Problema

Los profesores de la asignatura de prácticas nos plantean la necesidad de disponer de un sistema de gestión de los alumnos que cursan la asignatura. Con carácter general nos informan que les interesa mantener información de los grupos que se forman de prácticas indicando qué alumno es el representante o líder de la misma.

# Extracción de Requisitos

## Metodología de desarrollo y de extracción de requisitos

El desarrollo del sistema software se realiza de acuerdo a la Metodología Ágil Scrum.

La extracción de los requisitos se realizó en una primera reunión con todos los equipos de desarrollo en la que pudimos realizar preguntas para una adecuada comprensión de las necesidades del sistema a desarrollar. A lo largo de las sesiones prácticas y por correo electrónico se pudo afinar los detalles de los requisitos y tras el análisis de la información facilitada obtuvimos los requisitos funcionales y no funcionales de la aplicación. Los requisitos finales de la aplicación se obtuvieron de forma escalonada, de acuerdo a la realidad del desarrollo Ágil. Al inicio de la sesión práctica 3 se incorporaron nuevos requisitos funcionales de la aplicación.

Analizando los requisitos de usuario, las personas involucradas y los datos que se almacenarían de la aplicación pudimos obtener los requisitos del sistema. Realizamos una especificación de los mismos estableciendo su prioridad y creando sus correspondientes Historias de Usuario.

## Personal implicado.

**Profesores:** Gestionan y mantienen la base de datos de los alumnos. Formado por el coordinador y los ayudantes. Las funciones que realizan son las mismas, salvo que el coordinador puede realizar copias de seguridad externas.

**Alumnos:** Cursan la asignatura impartida por el profesor.

## Datos almacenados por la aplicación

El sistema almacenará los siguientes parámetros correspondientes a un alumno:

*DNI.*

Nombre.

*Apellidos.*

Fecha de nacimiento.

*Teléfono.*

Email Corporativo.

*Domicilio.*

Curso más alto en el que se ha matriculado.

*Nota.*

**Equipo al que pertenece.**

**Líder (Sí/No).**

Los indicados en negrita son datos no obligatorios.

## Requisitos del sistema

En primer lugar se presentan los requisitos funcionales extraídos para la aplicación:

**0 Acceso a la aplicación:** Los profesores podrán acceder a la aplicación mediante sus credenciales de la Universidad.

**1 Introducir un alumno:** Los profesores podrán introducir un nuevo alumno.

**2 Buscar un alumno:** El sistema deberá buscar un alumno o varios de acuerdo a unas directrices.

**3 Modificar un alumno:** Los profesores tendrán la opción de modificar los datos de un alumno.

**4 Gestión de líderes:** El sistema dará la posibilidad de gestionar los líderes de los grupos de acuerdo a las restricciones impuestas por el cliente (no pudiendo haber más de un líder por grupo).

**5 Mostrar alumnos:** El sistema podrá mostrar uno, varios o todos los alumnos.

**6 Eliminar un alumno:** El sistema permitirá borrar un alumno de la base de datos.

**7 Guardar base de datos:** El sistema dará la opción de guardar los datos introducidos en la aplicación en un fichero binario.

**8 Cargar base de datos:** El sistema permite cargar el fichero binario con los datos introducidos.

**9 Guardar copia de seguridad externa:** El sistema dará la opción de realizar una copia de seguridad en un servidor externo con los datos de la aplicación.

**10 Cargar copia de seguridad externa:** El sistema también dará la opción de cargar la copia de seguridad externa.

**11 Salir de la aplicación:** El sistema consultará si se desea guardar la base de datos o bien realizar la copia de seguridad externa, sólo para el caso del Coordinador.

Los requisitos funcionales 9 y 10 fueron incorporados por los profesores de la asignatura en la sesión 3 de prácticas.

Las prioridades establecidas para cada Requisito funcional fueron las que se indican en la siguiente tabla:



**Tabla 1 Requisitos funcionales**

Los requisitos no funcionales de la aplicación son

**1.** Sistema operativo Linux obligatorio, multiplataforma opcional.

**2.** Lenguaje de Programación C++.

**3.** Lenguaje de documentación Markdown.

**4.** Utilización de ficheros binarios para almacenar la información, tanto la de los alumnos como las credenciales de los profesores.

**5.** Utilización de Drive para almacenar la información en un servidor externo.

**6.** Máximo 150 alumnos.

**7.** Interfaz obligatoria línea de comandos, opcional gráfica.

**8.** Únicamente sirve para el año actual.

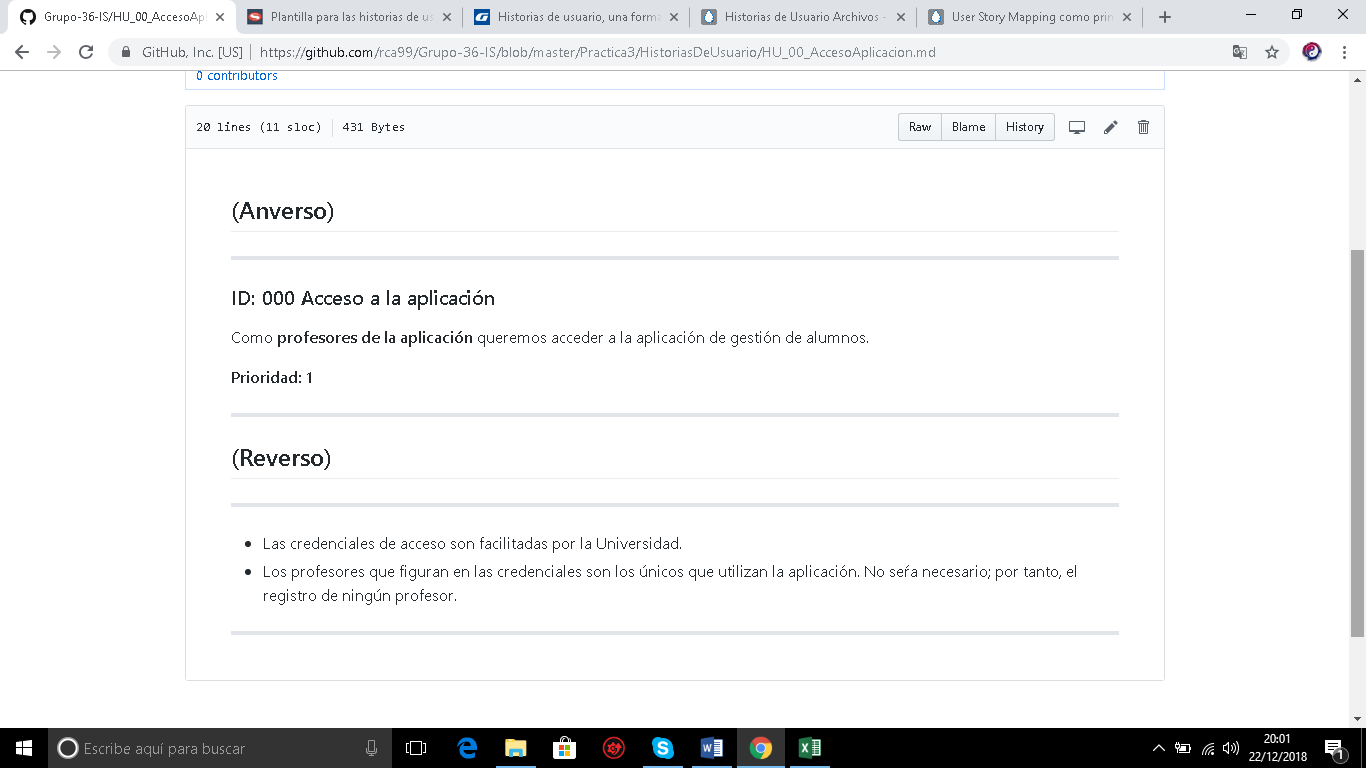
**9.** Tendrán acceso solo aquellos profesores que figuren en las credenciales cedidas por la Universidad que incluyen usuario, contraseña y rol.

(Arias Chaves, 2007)

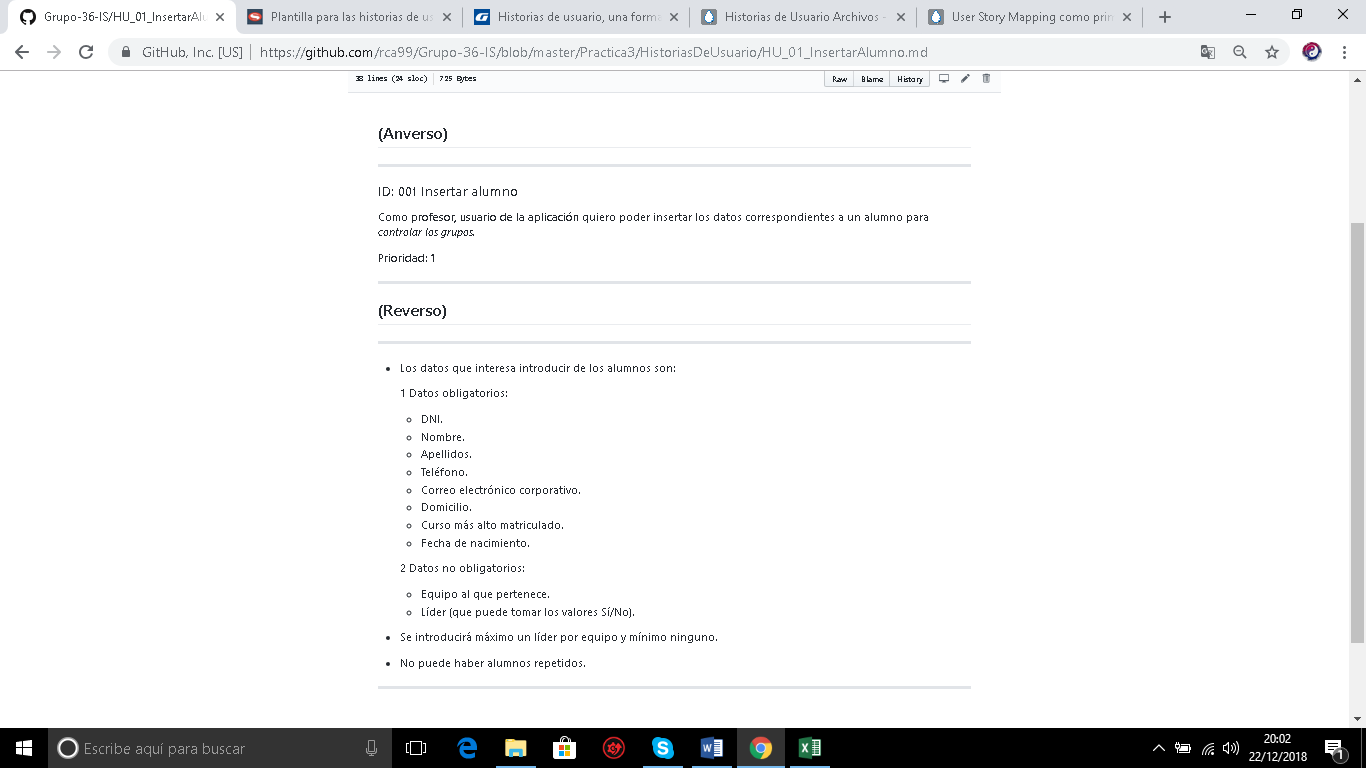
## Historias de Usuario

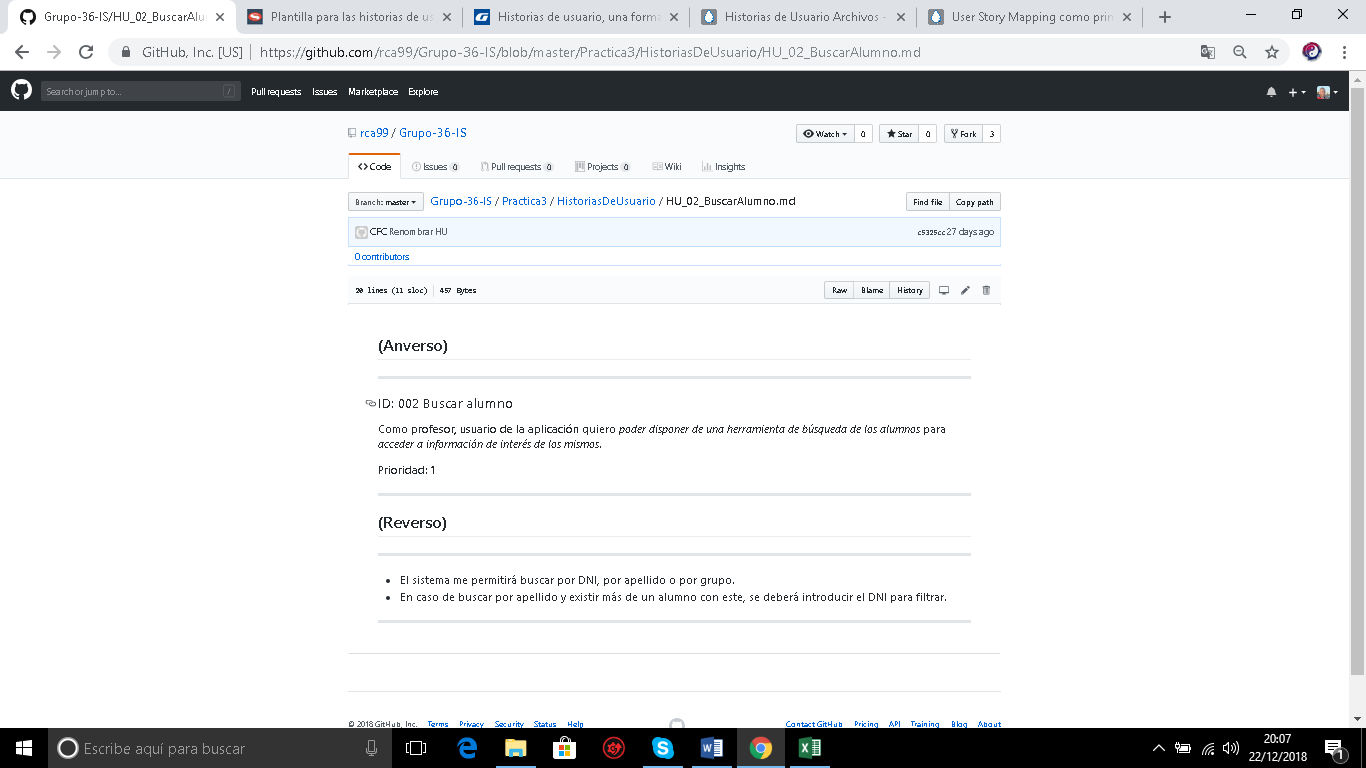
Una vez definidos los requisitos funcionales pasamos a definir las

Historias de usuario como elemento básico para aplicar la metología Scrum. En las mismas describiremos qué requisito dará cumplimiento y los criterios de validación de los mismos.

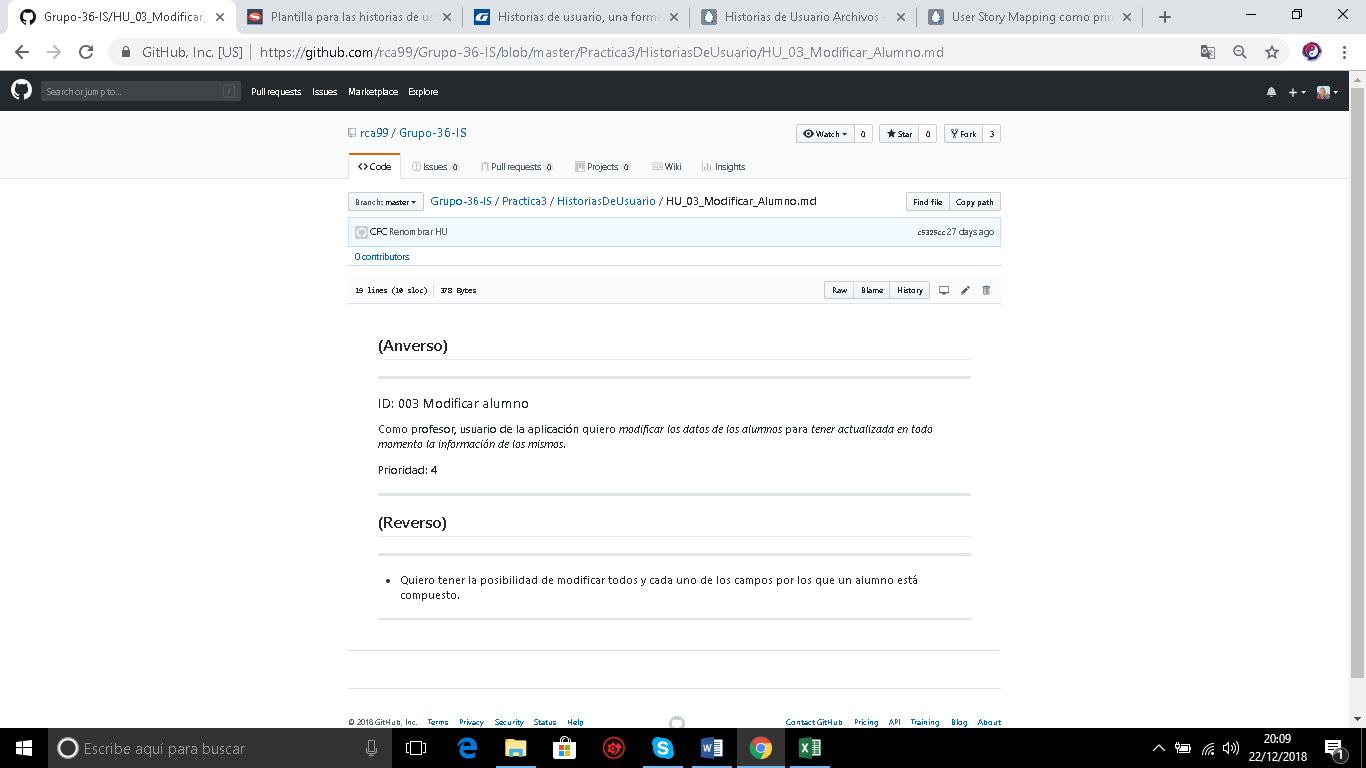
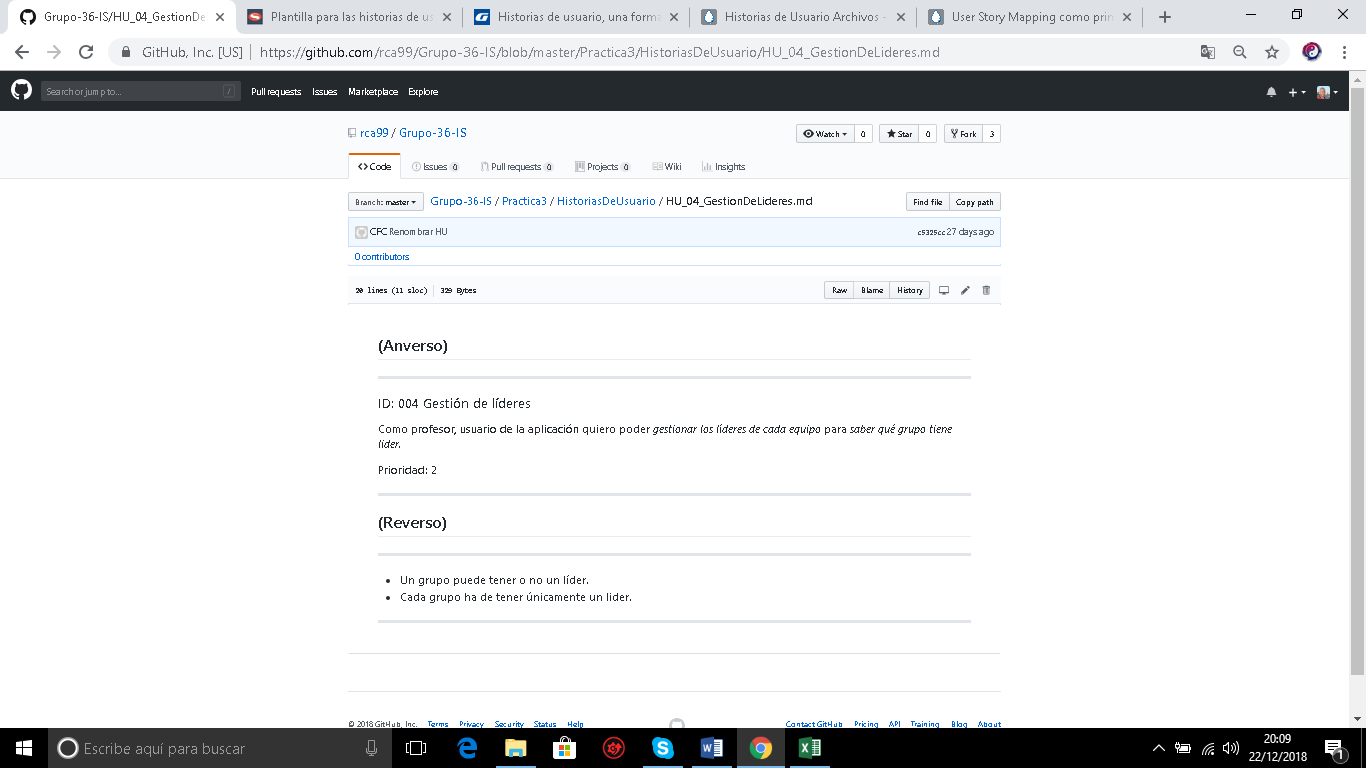


**Ilustración 1 Historia Uso 000 Acceso a la aplicación**



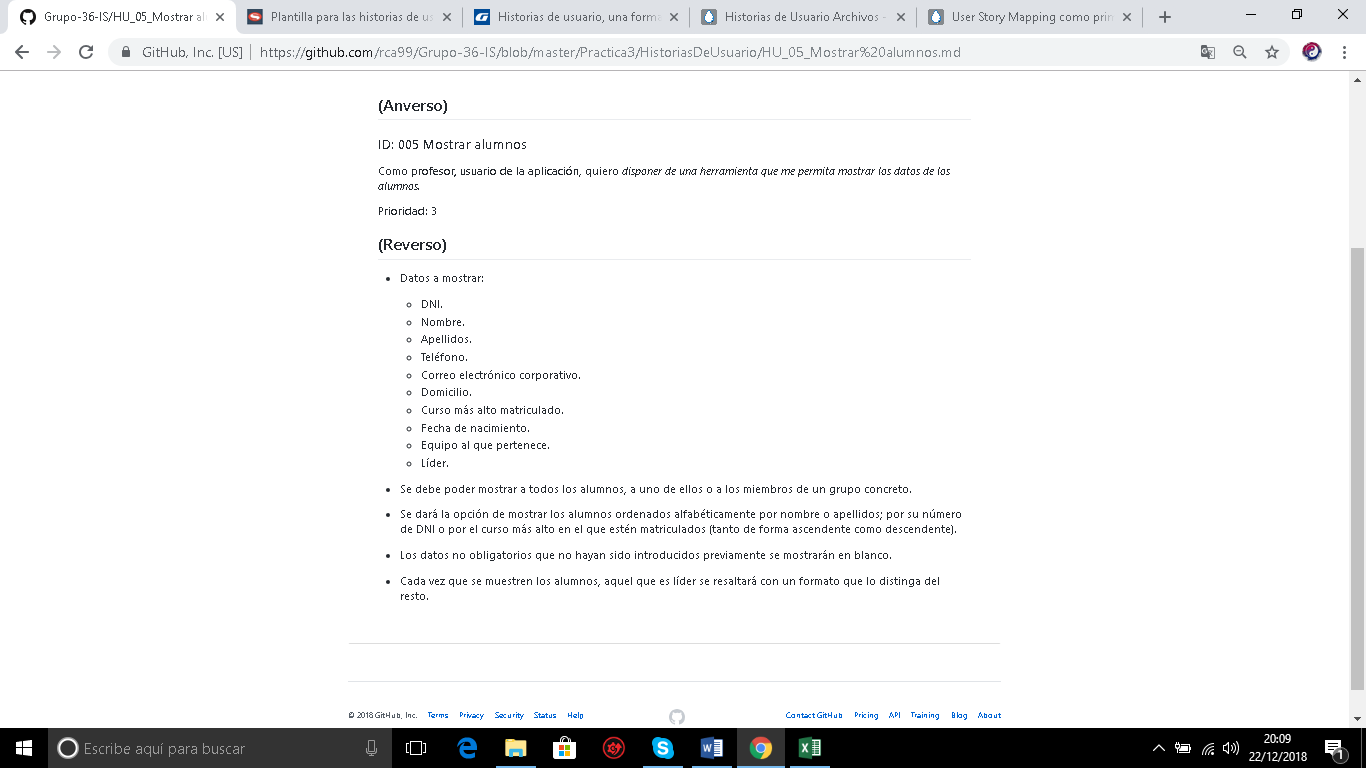
**Ilustración 2 Historia Uso 001 Insertar alumno**

**Ilustración 3 Historia Uso 002 Buscar alumno**

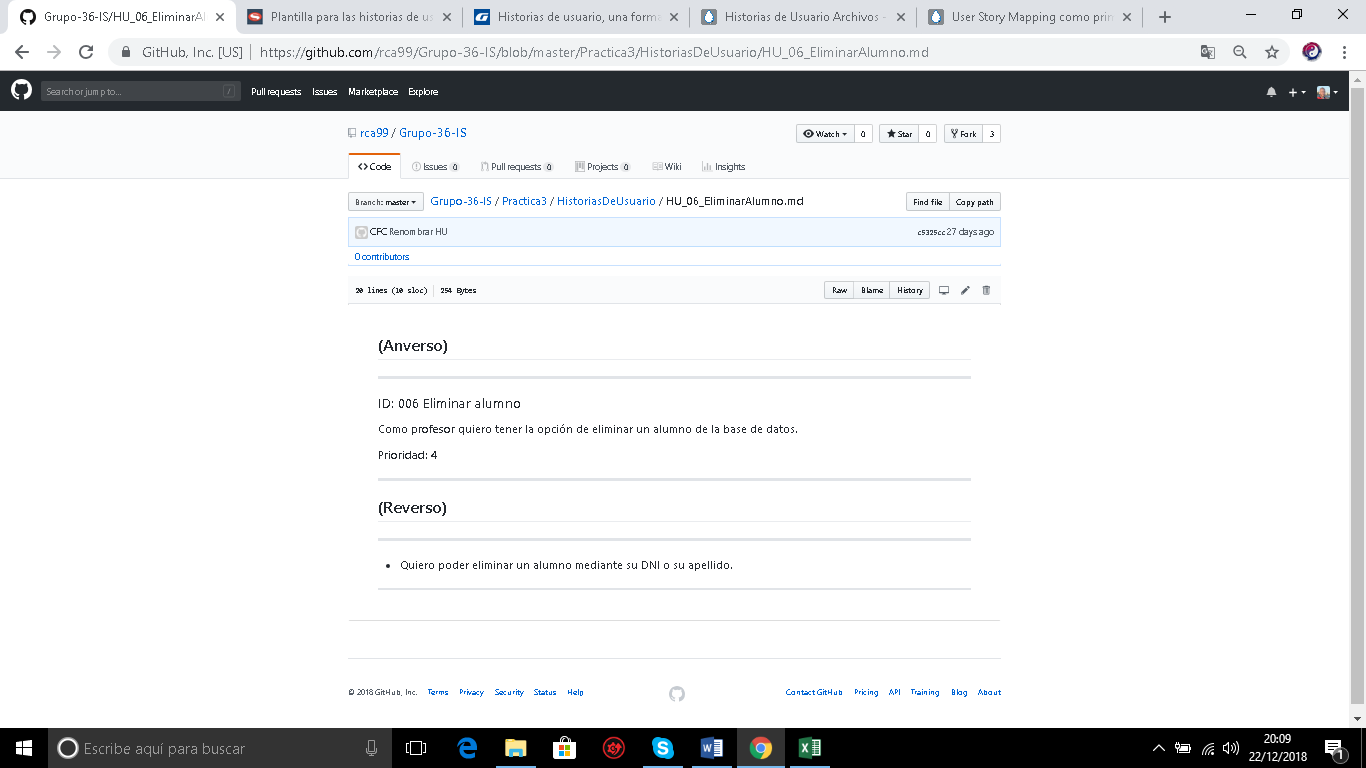


**Ilustración 5 Historia Uso 004 Gestión de Líderes**

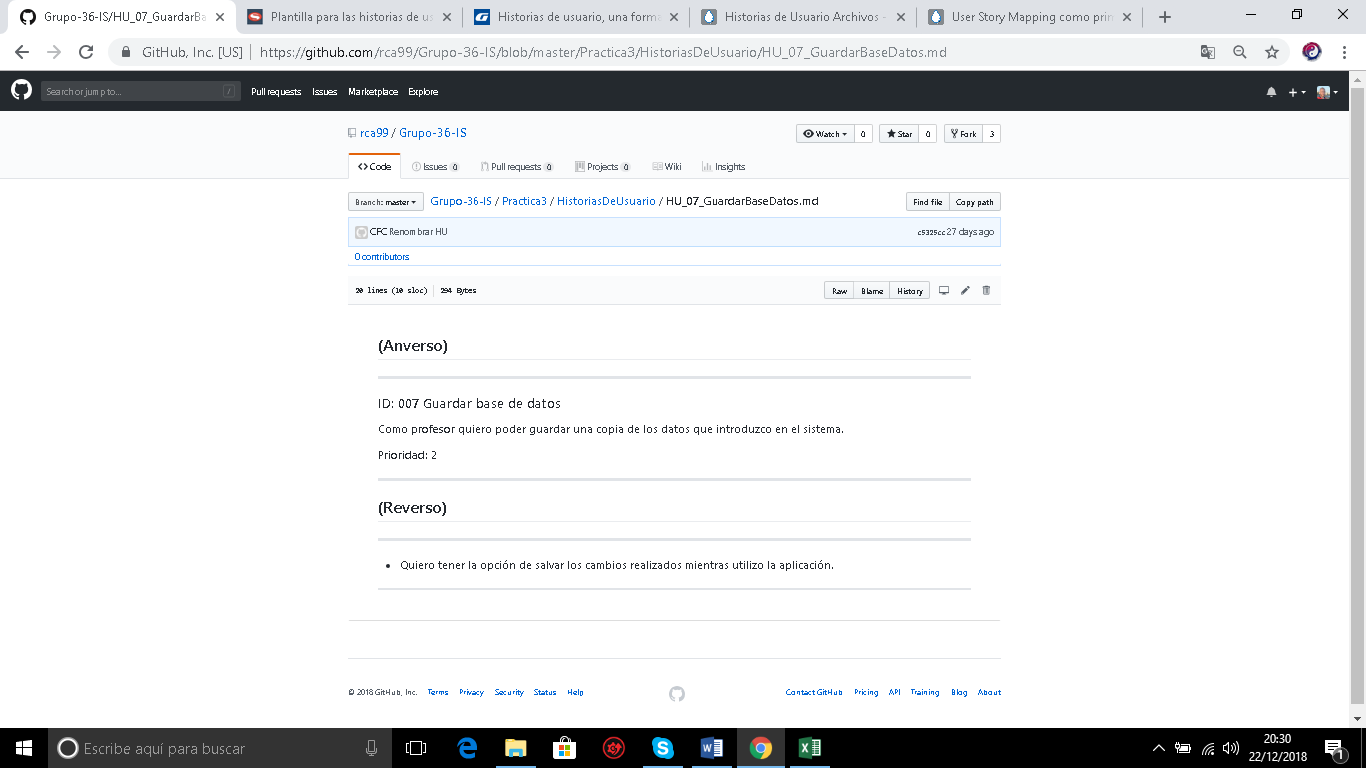
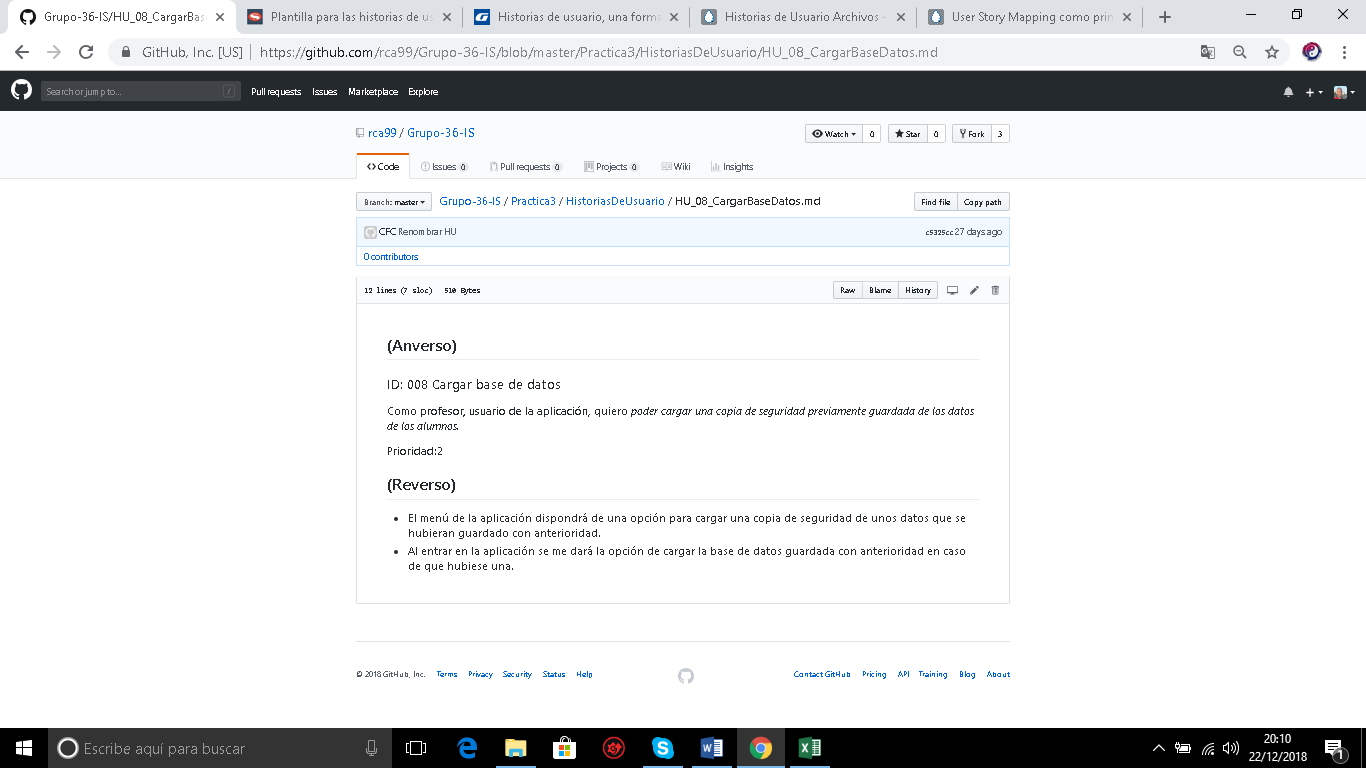
**Ilustración 4 Historia Uso 003 Modificar alumno**



**Ilustración 6 Historia Uso 005 Mostrar alumno**



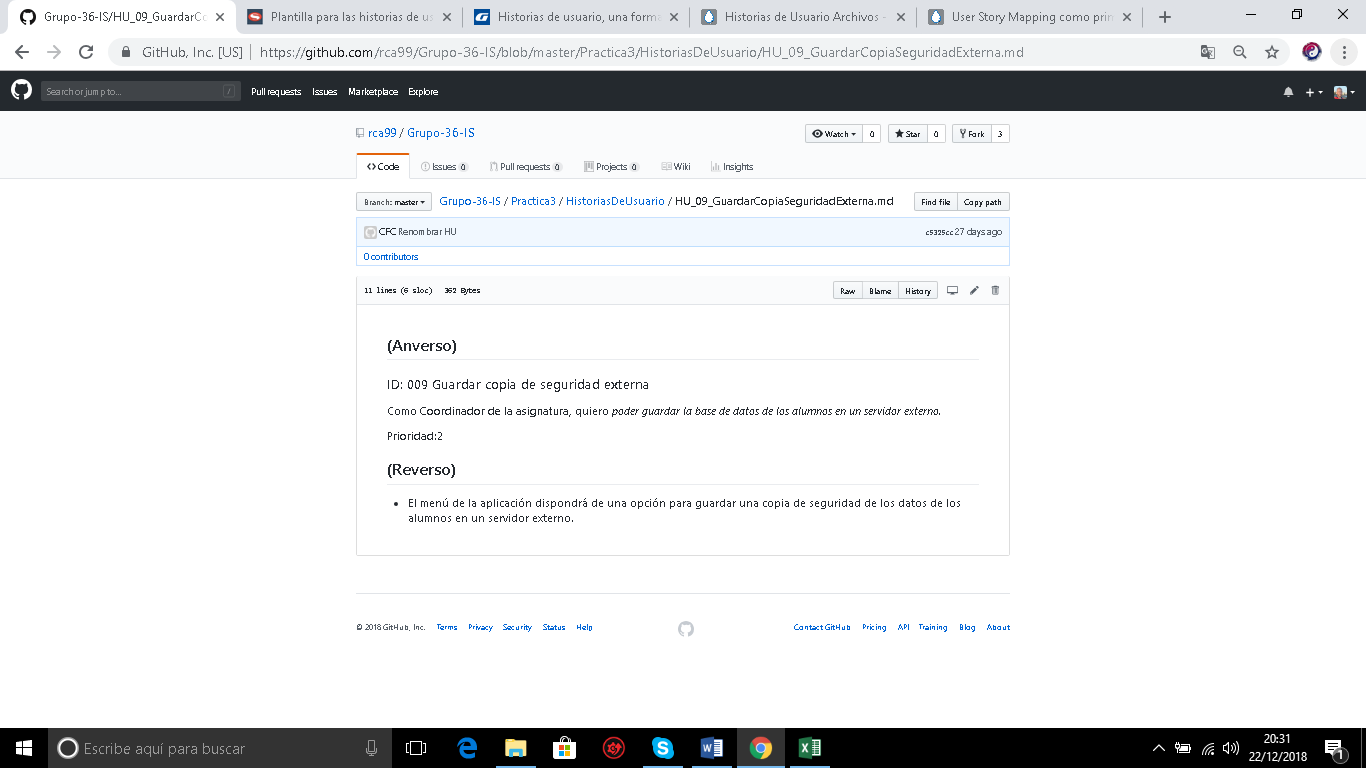
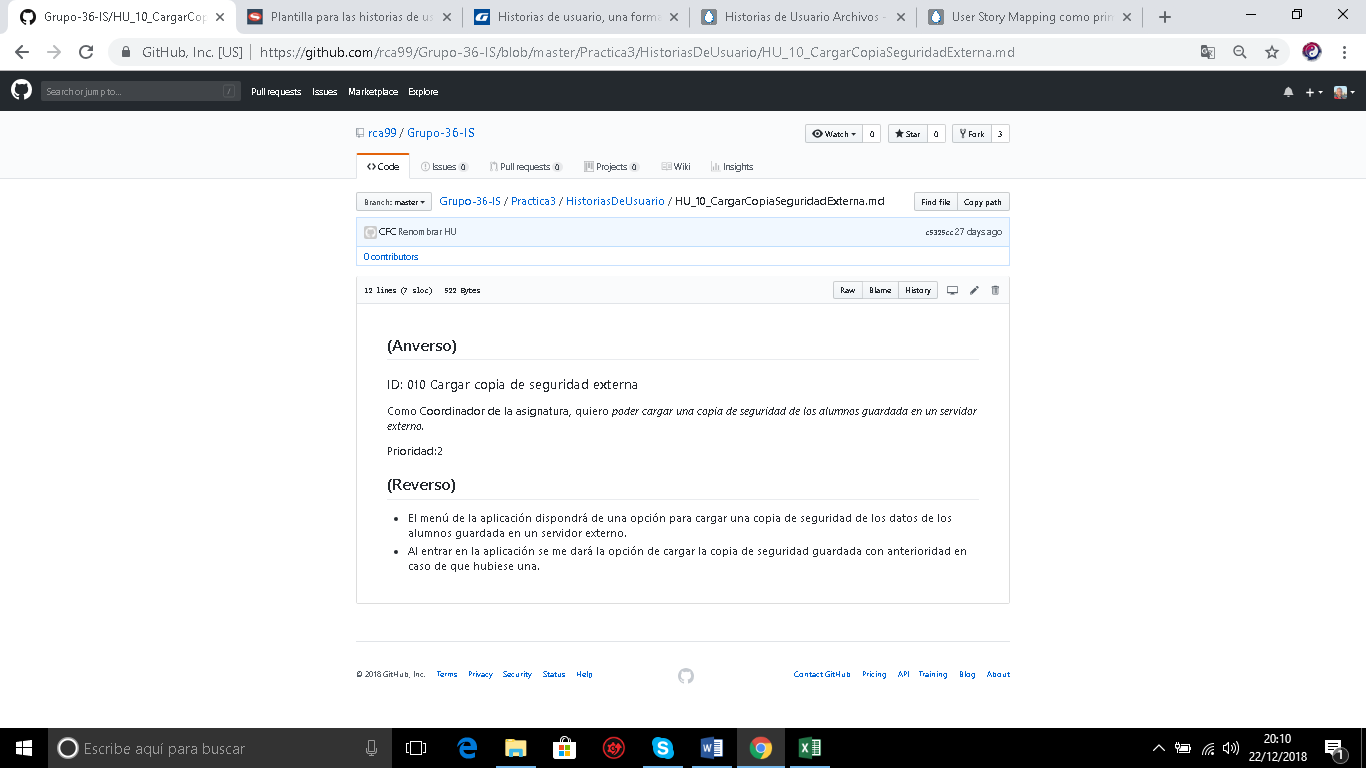
**Ilustración 7 Historia Uso 006 Eliminar alumno**



**Ilustración 8 Historia Uso 007 Guardar base de datos**

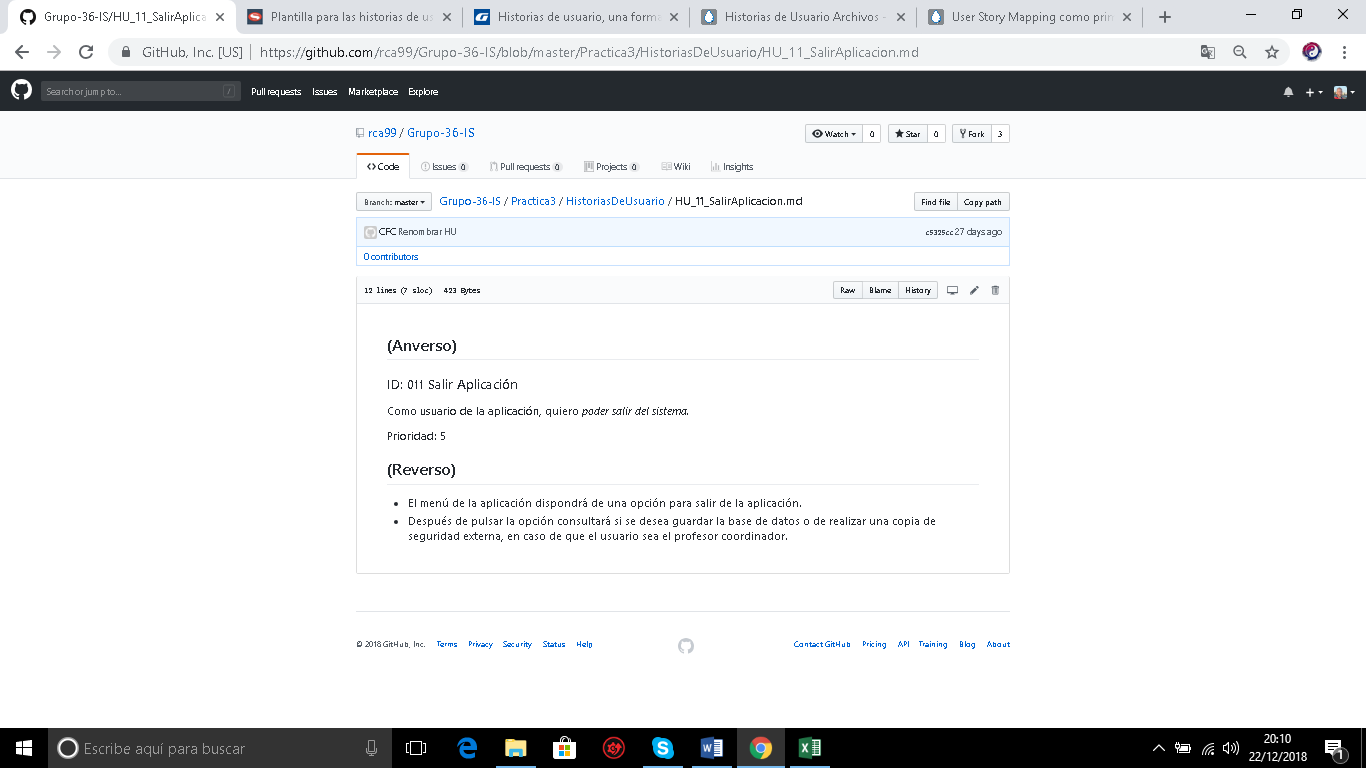
**Ilustración 9 Historia Uso 008 Cargar base de datos**

**Ilustración 9 Historia Uso 008 Cargar base de datos**



**Ilustración 10 Historia Uso 009 Guardar copia de seguridad externa**

**Ilustración 11 Historia Uso 010 Cargar copia de seguridad externa**

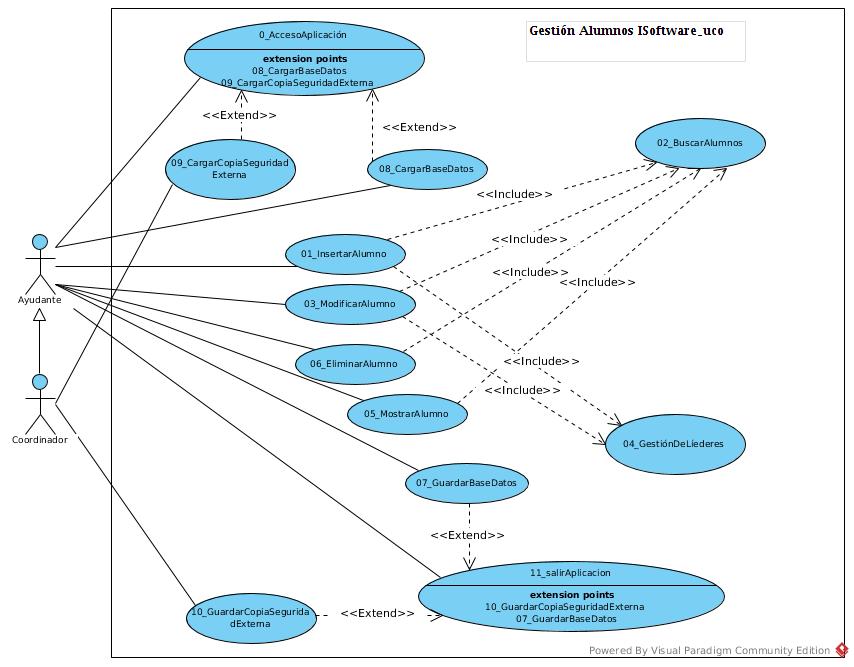


**Ilustración 12 Historia Uso 011 Salir Aplicación**

## Casos de Uso

Aunque desde la metodología Ágil, podríamos partir de las Historias de Usuario para desarrollar nuestro sistema, pasamos a describir los casos de Uso para servirnos de apoyo en la implementación del mismo.

Del análisis del Diagrama de Casos de Uso que se presenta a continuación podemos observar como en nuestro sistema encontramos a los ayudantes y coordinador de la asignatura relacionados mediante una relación de herencia y en el que el coodinador de la asignatura hereda los atributos y funciones propias del ayudante y además extiende sus funcionalidades a las propias definidas en los requisitos de guardar y cargar la base de datos externa. Para cada historia de usuario definimos un caso de uso y en ellos se pueden establecer relaciones de inclusión y extensión entre algunos de ellos. Relaciones de inclusión se dan entre los casos de uso Insertar, Modificar, Eliminar y Mostrar alumno con el caso de uso Buscar Alumno y también entre Insertar alumno y Modificar alumno con el caso de uso Gestión de Líderes. Las relaciones de extensión se dan en el caso de uso de Acceso a la aplicación en el que opcionalmente se puede bien cargar la base de datos o cargar la base de datos externa y en el caso de uso de salir de la aplicación el que opcionalmente se puede bien guardar la base de datos o la guardar la base de datos externa.



**Ilustración 12 Diagrama de Casos de Uso**

Se pasa a continuación a definir cada uno de los Casos de Uso a desarrollar en el sistema.

Acceso a la aplicación **ID: 00**

**Breve descripción: Los profesores podrán acceder a la aplicación mediante sus credenciales de la Universidad.**

**Actores principales: Profesor coordinador y profesores ayudantes**

**Actores secundarios: -**

### Precondiciones

* Se dispone de acceso a las credenciales de los profesores (usuario, contraseña, asignaturas que cursan y papel que desempeñan en la misma, coordinador o ayudante).

### Flujo principal

1 El caso de uso comienza cuando el usuario necesita acceder a la aplicación.

2 El profesor introducirá su usuario y contraseña habitual de moodle.

3 El sistema comprobará si existe el usuario y que la contraseña introducida sea correcta.

4 Opcionalmente, el sistema consultará si se desea cargar la base de datos de alumnos en caso de existir.

4.1 Además, si es coordinador también dará la posibilidad de cargar la copia de seguridad externa en caso de existir.

### Postcondiciones

A Se accede al menú principal de la aplicación.

B Muestra un mensaje de bienvenida al profesor indicando su nombre por pantalla.

C En caso de cargar la base de datos o copia de seguridad muestra el número de alumnos actuales que se hayan registrados.

## Flujos alternativos

3.1 Si no se introducen las credenciales correctas el sistema informa al usuario y da tres oportunidades de acceso. En caso de no introducir las credenciales correctas en estas tres ocasiones, se sale de la aplicación.

4.1 En caso de no haber sido posible cargar la base de datos mostrará un mensaje informando de la causa.

4.1.1 No existe base de datos o copia de seguridad que cargar.

4.1.2 Ha habido un error a la hora de procesar el archivo de base de datos o copia de seguridad.

Insertar Alumno **ID: 1**

**Breve descripción: El profesor podrá introducir los datos de un alumno.**

**Actores principales: Profesor.**

**Actores segundarios: Alumno.**

### Precondiciones

* El alumno a insertar no puede existir en el sistema.
* El número de alumnos del sistema será menor a 150.

### Flujo principal

1 El caso de uso comienza cuando el usuario necesita introducir un alumno.

2 El sistema comprueba el número de alumnos que existen en el sistema.

3 El sistema recoge los datos de dni, nombre y apellidos para comprobar si el alumno existe en el sistema

4 Se comprobará que el alumno no existe ya en el sistema.

4.1 Si el alumno no existe en el sistema se introducen los datos obligatorios y aquellos opcionales que se desean indicar.

5 El sistema comprueba si el grupo que ha elegido el alumno tiene ya lider.

5.1 Si el grupo que ha elegido el alumno no tiene líder permite asignar este alumno como líder del grupo

6 El sistema introduce el alumno.

7 El sistema comprueba la inserción del alumno en el sistema.

### Postcondiciones

**A** El sistema se actualiza con la introducción del alumno.

**B** El sistema muestra un mensaje indicando que la inserción ha sido correcta mostrando el nombre y dni del alumno introducido.

## Flujos alternativos

2.1 Si en el sistema hay 150 alumnos dados de alta, se muestra un error indicando que no se puede introducir el alumno.

4.1 Si no introduce los datos obligatorios muestra un mensaje de error, indicando los datos que faltan por introducir.

4.2 Por defecto, el campo líder valdrá NO.

4.3 Si el usuario introduce que un alumno es líder entonces deberá indicar obligatoriamente el grupo al que pertenece.

5.1 El sistema no permite la inserción de un alumno que ya existe en el sistema y muestra un mensaje de error.

6.1 El sistema no permite indicar que un alumno es líder de un grupo que ya tenga líder.

7.1 Si el sistema no ha podido realizar la inserción del alumno indica que esta no ha podido ser realizada y que vuelva a intentarlo.

Buscar Alumno **ID: 2**

**Breve descripción: El profesor podrá buscar los alumnos introducidos en la base de datos por DNI, apellidos o grupo.**

**Actores principales: Profesor**

**Actores segundarios: Alumno**

### Precondiciones

* La aplicación debe tener alumnos dados de alta.

### Flujo principal

1 El caso de uso comienza cuando se necesita buscar un alumno.

2 El sistema comprueba si hay alumnos dados de alta en el sistema.

3 El sistema recoge los datos necesarios (DNI, apellidos o grupo) para buscar.

4 El sistema busca el alumno por DNI, apellidos o grupo.

### Postcondiciones

**A** El sistema devuelve el alumno o alumnos encontrados.

**B** En caso de devolver los alumnos del grupo, el sistema indicará si el grupo tiene líder o no lo tiene.

## Flujos alternativos

2.1 Si no hay alumnos en la base de datos, el sistema consultará al profesor si ha cargado la base de datos.

2.1.1 Si ha cargado la base de datos, indicará que no hay alumnos y que proceda a insertar alumnos antes de buscarlos.

2.1.2 Si no ha cargado la base de datos, se le indica que la cargue.

3.1 Se comprueba que los datos introducidos son del tipo y formato correcto, si no lo son se indica que se introduzcan adecuadamente.

4.1 En caso de no existir el alumno muestra un mensaje indicando que el alumno no existe en la base de datos y se regresa al paso 1.

4.2 En caso de que el grupo no tenga alumnos asociados, se muestra un mensaje indicando de que el grupo está vacío.

4.3 En caso de que exista más de un alumno con el mismo apellido, el sistema indica al profesor que introduzca el DNI para filtrar.

Modificar Alumno **ID: 3**

**Breve descripción: El profesor podrá modificar los datos de un alumno.**

**Actores principales: Profesor.**

**Actores segundarios: Alumno.**

### Precondiciones

* El alumno a modificar debe estar almacenado en el sistema.

### Flujo principal

1 El caso de uso da comienzo cuando el profesor necesita modificar un alumno.

2 El sistema pregunta si se desea buscar por DNI o por apellido.

3 El sistema busca al alumno en cuestión por DNI o por apellido.

4 El sistema muestra los datos del alumno, dando la opción de modificar aquellos que se deseen.

4.1 Si se mofica el campo de líder, el sistema comprobará que el grupo al que pertenece el alumno tenga líder o no.

### Postcondiciones

**A** El sistema se actualiza con la nueva información del alumno.

**B** El sistema muestra un mensaje indicando que la modificación ha sido correcta.

## Flujos alternativos

3.1 El sistema no permite la modificación de un alumno que no se encuentra almacenado en la aplicación.

4.1 Si se desean dejar en blanco alguno de los campos obligatorios, el sistema no permitirá guardar los cambios volviendo al punto 4 del proceso.

4.1.1 Si se desea modificar el campo líder, indicando que es líder de un grupo que ya lo tiene, muestra un mensaje de error.

Gestión de Líderes **ID: 4**

**Breve descripción: El sistema verifica si un alumno puede ser o no líder del grupo.**

**Actores principales: Profesor.**

**Actores segundarios: Alumno.**

### Precondiciones

* El alumno estará identificado en el sistema con su DNI, nombre, apellidos y grupo al que pertenece.

### Flujo principal

1 El caso de uso comienza cuando el profesor desea indicar si un alumno es líder de un grupo.

2 El sistema recibe el grupo con el que se quiere interactuar.

3 El sistema busca los alumnos que pertenecen a dicho grupo.

4 El sistema comprueba si algún alumno del grupo es líder.

### Postcondiciones

**A** El sistema indica si un grupo tiene líder o no lo tiene.

## Flujos alternativos

Mostrar Alumno **ID: 5**

**Breve descripción: El profesor podrá mostrar los datos de un alumno.**

**Actores principales: Profesor.**

**Actores segundarios: Alumno.**

### Precondiciones

* El alumno a mostrar debe existir en el sistema.

### Flujo principal

1 El caso de uso comienza cuando el usuario necesita mostrar un alumno.

2 El sistema da la opción de mostrar a todos los alumnos dados de alta en el sistema, a uno en particular, o a los pertenecientes a un grupo.

3 El sistema dará la opción de ordenar a los alumnos mostrados por orden alfabético (nombre o apellidos), por su DNI, o por el curso más alto en el que estén matriculados; tanto de forma ascendente como descendente.

4 El sistema muestra los datos del alumno o alumnos según los criterios seleccionados.

### Postcondiciones

**A** El sistema debe mostrar en blanco los datos del alumno no obligatorios que no hayan sido introducidos previamente.

## Flujos alternativos

2.1 Si el sistema no encuentra a un alumno al que se ha pedido mostrar al buscarlo en el sistema, dará un mensaje de error.

3.1 Si hay dos alumnos que posean la misma información en el campo según el que mostrar a los alumnos ordenados (por ejemplo, el curso más alto en el que se esté matriculado), se mostrarán ordenados según otro de los criterios disponibles, bien por apellido o por DNI en última instancia.

Eliminar Alumno **ID: 6**

**Breve descripción: El profesor podrá eliminar un alumno de la base de datos.**

**Actores principales: Profesor.**

**Actores segundarios: Alumno.**

### Precondiciones

* El alumno que será eliminado debe existir en el sistema.

### Flujo principal

1 El caso de uso se iniciará en el momento que el profesor desee eliminar un alumno.

2 El sistema pregunta si se desea eliminar por DNI o por apellido.

3 El sistema busca al alumno en cuestión por DNI o por apellido.

4 Se elimina al alumno del sistema.

### Postcondiciones

**A** El sistema actualiza la base de datos eliminando al alumno.

**B** Se muestra un mensaje de éxito, indicando la correcta eliminación del alumno.

## Flujos alternativos

* 1. El sistema no permite la eliminación de un alumno que no existe en la aplicación.

Guardar Copia de Seguridad **ID: 7**

**Breve descripción: El profesor podrá realizar una copia de seguridad de los alumnos registrados en el sistema.**

**Actores principales: Profesor.**

**Actores segundarios: Alumno.**

### Precondiciones

* La aplicación debe tener alumnos dados de alta.
* Los datos se guardarán en un archivo binario.

### Flujo principal

1 El caso de uso comienza cuando el usuario requiere realizar copia de seguridad.

2 El sistema comprueba si hay alumnos dados de alta.

3 El sistema actualiza los datos de los alumnos en el fichero binario que almacena la base de datos.

4 El sistema comprueba que la base de datos se ha guardado correctamente.

### Postcondiciones

**A** El sistema muestra un mensaje indicando que los datos se han guardado correctamente.

**B** En caso de no existir alumnos el sistema informa que no es necesaria la copia de seguridad.

## Flujos alternativos

2.1 Si no hay alumnos en la base de datos, el sistema consulta al profesor si ha cargado la base de datos.

2.1.1 Si ha cargado la base de datos, indicará que no hay alumnos y que proceda a insertar alumnos antes de guardar la base de datos.

2.1.2 Si no ha cargado la base de datos, se le indica que la cargue.

3.1 Si no existe el fichero binario se procederá a crearlo.

4.1. Si no se insertan correctamente los datos da un mensaje de error, indicando que vuelva a guardar la base de datos.

Cargar Copia de Seguridad **ID: 8**

**Breve descripción: El profesor podrá cargar una copia de seguridad de los datos de los alumnos registrados.**

**Actores principales: Profesor.**

**Actores segundarios: Alumno.**

### Precondiciones

* La copia de seguridad a cargar debe haber sido previamente guardada.

### Flujo principal

1 El caso de uso comienza cuando el usuario necesita cargar una copia de seguridad.

2 El sistema comprueba si existe el fichero de base de datos.

3 El sistema carga la base de datos.

4 El sistema comprueba que la base de datos se ha cargado correctamente.

### Postcondiciones

**A** El sistema muestra un mensaje de confirmación de la carga y de cuál de las copias de seguridad existentes ha sido cargada.

**B** El sistema permite al usuario sobrescribir los datos de esa copia de seguridad.

## Flujos alternativos

2.1 Si no existe fichero de base de datos, el sistema pedirá al usuario que debe guardar antes la base de datos.

4.1 Si el sistema no es capaz de cargar la base de datos, dará un mensaje de error.

Guardar Copia de Seguridad Externa **ID: 09**

**Breve descripción: El Coordinador podrá guardar la copia de seguridad externa en un servidor remoto.**

**Actores principales: Coordinador.**

**Actores segundarios: Alumno.**

### Precondiciones

* Debe existir previamente una base de datos de la cual crear la copia de seguridad externa.
* Debe haber una conexión a internet activa para acceder al repositorio remoto.

### Flujo principal

1 El caso de uso comienza cuando el usuario requiere guardar la copia de seguridad externa

2 El sistema comprueba si existe el fichero de base de datos que guardar.

3 El sistema conecta con el repositorio remoto.

4 El sistema envía el fichero de base de datos al fichero remoto y guarda la copia de seguridad externa.

### Postcondiciones

**A** El sistema muestra un mensaje de confirmación de realización de la copia de seguridad externa.

## Flujos alternativos

2.1 Si no hay fichero de base de datos, el sistema indicará al usuario que cree una copia de seguridad antes de realizar la copia de seguridad externa.

3.1 Si el sistema no es capaz de conectar con el repositorio remoto informa con un mensaje de error.

4.1 Si el sistema no realiza la copia de seguridad externa da un mensaje de error.

Cargar Copia de Seguridad Externa **ID: 10**

**Breve descripción: El Coordinador de la asignatura podrá cargar una copia de seguridad externa de los datos de los alumnos registrados.**

**Actores principales: Coordinador de la asignatura.**

**Actores segundarios: Alumno.**

### Precondiciones

* La copia de seguridad externa a cargar debe haber sido previamente guardada.
* Debe haber una conexión a internet activa para acceder al repositorio remoto.

### Flujo principal

1 El caso de uso comienza cuando el usuario necesita cargar una copia de seguridad externa

2 El sistema carga la copia de seguridad externa.

3 El sistema comprueba que la base de datos se ha cargado correctamente.

### Postcondiciones

**A** El sistema muestra un mensaje de confirmación de la carga y de cuál de las copias de seguridad existentes ha sido cargada.

**B** El sistema permite al usuario sobrescribir los datos de esa copia de seguridad.

## Flujos alternativos

2.1 Si no hay copias de seguridad externas guardadas, el sistema pedirá al usuario que cree una copia de seguridad.

3.1 Si el sistema no es capaz de cargar la copia de seguridad da un mensaje de error, indicando que no ha sido posible cargar esa copia de seguridad.

Salir Aplicación **ID: 11**

**Breve descripción: El profesor podrá salir de la aplicación.**

**Actores principales: Profesor.**

**Actores segundarios: Alumno.**

### Precondiciones

* .

### Flujo principal

1 El caso de uso comienza cuando el usuario necesita salir de la aplicación.

2 El sistema solicita confirmación para salir de la aplicación

3 El sistema consulta si se quiere guardar la base de datos y/o de realizar la copia de seguridad externa en caso de ser el Coordinador.

3 La aplicación se cierra.

### Postcondiciones

**A** El sistema muestra un mensaje de salida al profesor y se cierra la aplicación.

## Flujos alternativos

2.1 Si se indica que no se quiere salir de la aplicación vuelve al menú principal de la aplicación.

3.1 Si no es posible cerrar la aplicación en ese momento se muestra un mensaje de error.

# a de Clases

# Diagramas de Secuencia

|  |  |
| --- | --- |
|  |  |
|  | Desarrollo de la aplicación – metodología scrum |
|  |  |

(ALFARO-HERRERA, 2017, págs. 38-42)

# Introducción

# Product BackLog

# Sprints Review meeting

# Sprint retrospective meeting

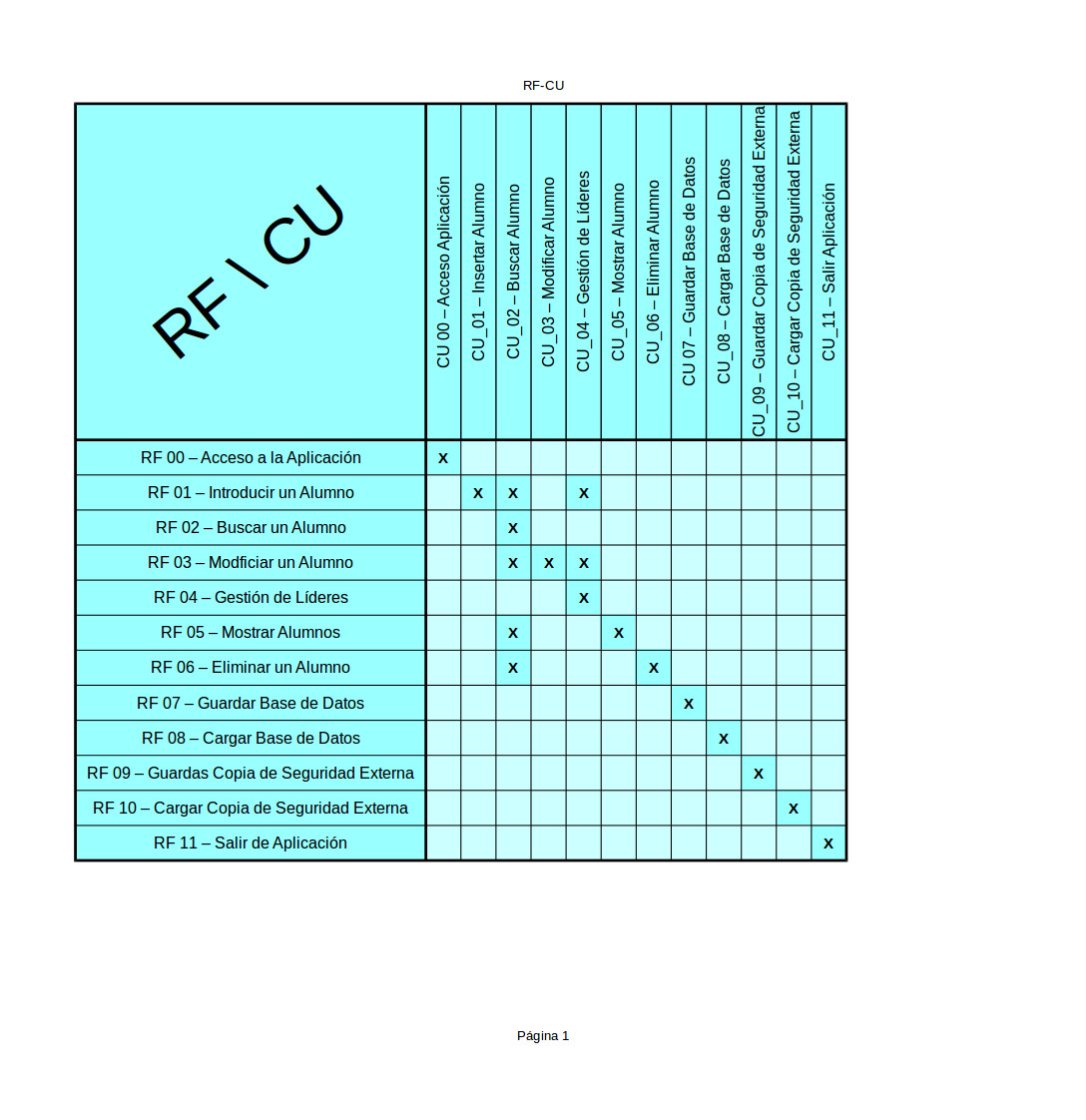
El objeto de la reunión de retrospectiva es mejorar la aplicación de la metodología Scrum por parte del equipo de trabajo. Cada sprint debe terminar con el Sprint Retrospective. Dada la naturaleza del proyecto y su temporalidad se realizará una única reunión para evaluar el resultado final del proceso de Scrum destacando los aspectos en los que se pudiera mejorar para las siguientes ocasiones en las que nos enfretemos al desarrollo de un sistema software basado en la metodología ágil Scrum.

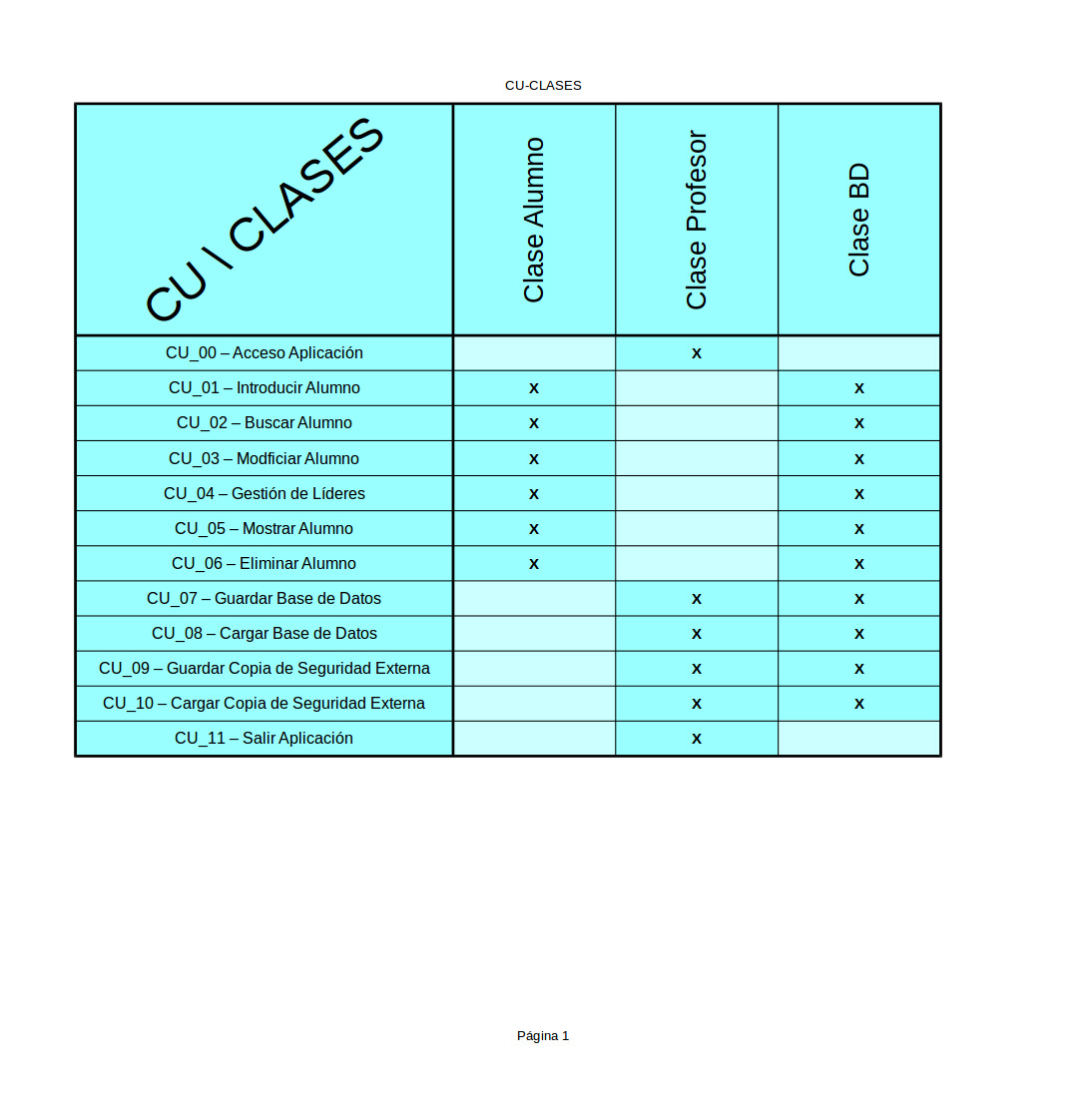
|  |  |
| --- | --- |
|  |  |
|  | cierre del sprint |
|  |  |

# Pruebas de software

# Verificación de Requisitos

## Matrices de Verificación.

**Ilustración 4 Matriz de Verificación Requisitos Funcionales**



**Ilustración 5 Matriz de Verificación Clases**

|  |  |
| --- | --- |
|  |  |
|  | Bibliografía y referencias Web |
|  |  |

ALFARO-HERRERA, J. C. (2017). Scrum como metodología para proyectos de redes. *Revista de Tecnologías Computacionales*, 38-42.

Arias Chaves, M. (2007). La ingeniería de requerimientos y su importancia en el desarrollo de proyectos de software. *Revista InterSedes*.