

UNIVERSIDAD DE CÓRDOBA
ESCUELA POLITÉCNICA SUPERIOR DE CÓRDOBA



GRADO EN INGENIERÍA INFORMÁTICA
INGENIERÍA DEL SOFTWARE
CURSO 2021/2022

Equipo 205
Práctica Final

Realizado por:
Martín del Río Jiménez
Kamal Abdelkader Haddu
Jorge Benítez Alonso

<https://uco-is2022-eq205.myjetbrains.com/>
<https://github.com/Martindrj/Equipo-205>

Córdoba, Diciembre de 2021

Índice

- 1. Introducción**
- 2. Objetivos**
- 3. Metodología**
- 4. Reuniones**
- 5. Análisis de requisitos**
 - 5.1. Entrevista**
 - 5.2. Tipos de requisitos**
 - 5.3. Requisitos del sistema**
- 6. Historias de usuario**
- 7. Casos de uso**
- 8. Diagrama de clases**
- 9. Matriz de trazabilidad**
 - 9.1 Requisitos funcionales frente a casos de uso**
 - 9.2 Casos de uso frente a clases**
- 10. Diagrama de secuencia**
- 11. Implementación**
- 12. Pruebas**
- 13. Bibliografía**

1. Introducción

Esta práctica consiste en el diseño de un sistema para la gestión de recursos informáticos dedicados a las tareas de investigación, que permita hacer reservas de máquinas y sus recursos con el fin de ejecutar experimentos y poder controlar una distribución de estos recursos de forma equitativa y justa entre todos los usuarios que la necesiten.

Para diseñar este sistema, el equipo de trabajo debe de plantear todo lo necesario de este previamente según las fases de desarrollo de un proyecto siguiendo la metodología expuesta en el punto 3 del documento.

A continuación se documentará y organizará el problema que se nos ha planteado y se resolverán todas las necesidades de este.

2. Objetivos

Como objetivo para esta práctica debemos resolver el problema mencionado anteriormente. Para lograr esta meta se creará un programa el cual se encargue de realizar las reservas de recursos para el equipo de investigación. Dicho programa se creará siguiendo unos requisitos, obtenidos mediante una reunión con el cliente, que se expondrán más adelante, mediante los cuales nos podremos guiar para crear un sistema totalmente funcional.

Además, otro objetivo de la práctica es adquirir conocimiento suficiente para ser capaz de organizar proyectos de software, para lo cual vamos a seguir una metodología que se explicará en el punto 3 de este documento. Posteriormente se analizará el problema y se organizará en los puntos posteriores.

3. Metodología

El equipo de trabajo seguirá la metodología SCRUM y la aplicará en la página de YouTrack en la cual se ha creado un proyecto para poder documentar nuestra tarea.

Esta metodología se centra en el buen trabajo colectivo para así obtener el mejor resultado posible del software. Para ello, se realizan reuniones cada cierto tiempo para así saber qué parte del proyecto se debe ejecutar a continuación y llevar un mejor seguimiento de lo ya hecho. En nuestro caso, estas reuniones conocidas como sprints las vamos a llevar a cabo cada semana, coincidiendo así con la clase de prácticas de la asignatura. Por otra parte, en esta metodología se realizan otro tipo de reuniones, las diarias, las cuales son de corta duración y sirven para comentar posibles problemas, dudas o trabajo que cada miembro del equipo piensa realizar.

También, se tiene un Scrum Master encargado de realizar el reconocimiento de requisitos del proyecto mediante la comunicación con el cliente y la organización de estos mismos. Para ello se realizan los siguientes puntos del documento en los cuales se dispone el trabajo realizado para esta práctica.

4. Reuniones

Para la realización de esta práctica, hemos repartido la carga de trabajo en diferentes reuniones comentadas en YouTrack:

- Reunión 1: Toma de contacto con el cliente mediante la cual recopilamos información para la realización del proyecto.
- Reunión 2: Organización de la información brindada por el cliente.
- Reunión 3: Revisión de requisitos junto al cliente y creación de las historias de usuario.
- Reunión 4 y 5: Creación de un cierto número de historias de usuario indispensables para el proyecto.
- Reunión 6: Subida a YouTrack de las historias de usuario consideradas como importantes.
- Reunión 7: Asignación de fechas y sprints a las historias de usuario y creación de los casos de uso.
- Reunión 8: Corrección de errores tras la revisión del cliente y adjuntar nuevos requisitos.
- Reunión 9: Corrección de requisitos funcionales añadidos al proyecto.
- Reunión 10: Revisión de requisitos funcionales modificados y continuación del proyecto con las historias de usuario.
- Reunión 11: Actualización de casos de uso tras la reunión con el cliente y las modificaciones de las historias de usuario.
- Reunión 12: Diagramas de caso de uso y diseño de la primera matriz de trazabilidad.
- Reunión 13: Diseño de las tablas de clase.
- Reunión 14: Se ha diseñado la segunda matriz y los diagramas de clases y secuencias mediante la herramienta Visual Paradigm.
- Reunión 15: El equipo va a diseñar el programa con los casos de uso 3, 5 y 2 (en ese orden de importancia).
- Reunión 16: El caso de uso 2 (Inicio de sesión) ha sido programado y ya se han realizado los tests pertinentes.
- Reunión 17: El equipo está intentando conectar Eclipse con Github, el cual está dando errores debido a fallos en el login.
- Reunión 18: Los casos de uso referentes a las reservas están en proceso de desarrollo.
- Reunión 19: La programación de los casos de uso va avanzando mientras se sube el programa actualizado en GitHub.
- Reunión 20: El CU2 junto al CU1 se han implementado al sistema y este funciona correctamente, mostrando máquinas disponibles a los usuarios y todos los datos de los usuarios a los administradores. El segundo commit se ha subido a Github.
- Reunión 21: Se ha finalizado la implementación del sistema.
- Reunión 22: Se han realizado pruebas en la implementación del sistema.

5. Análisis de requisitos

5.1. Entrevista

Para la realización del análisis de requisitos es necesario conocer los detalles del sistema que debemos crear. Para ello hay muchas formas diferentes de adquirir información del cliente. En nuestro caso se ha realizado una entrevista con el cliente en la cual los diferentes grupos de trabajo de la clase hemos llevado a cabo una serie de preguntas. Las respuestas a estas preguntas, tanto a las de nuestro equipo como las de otros, se anotaron para posteriormente analizarlas.

Una vez conocida la base de nuestro sistema y organizada según los tipos de requisitos que se explicarán y expondrán a continuación, somos capaces de seguir con la organización del proyecto.

5.2. Tipos de requisitos

Los diferentes tipos de requisitos sobre los cuales nos vamos a guiar para la creación del sistema de reservas de recursos para el grupo de investigación son:

-Requisitos funcionales: En este tipo se recogen los diferentes requisitos relacionados con el conjunto de funciones o servicios que el sistema debe proporcionar. Estos nos ayudarán a saber qué debe hacer el sistema en sus diferentes puntos.

-Requisitos de información: Estos requisitos son aquellos los cuales nos hacen referencia a cómo se va a administrar la información que el sistema va a estar usando para su correcto funcionamiento.

-Requisitos no funcionales: En este caso se indican las restricciones que va a tener el sistema y expresa cómo se va a comportar.

5.3. Requisitos del sistema

Requisitos funcionales

RF1. El sistema debe proporcionar la capacidad de dar de alta o de baja usuarios (habrá 3 tipos, usuario, administrador de usuarios y administrador de máquinas).

RF2. El sistema debe proporcionar la capacidad de iniciar sesión a cualquier usuario/administrador del sistema.

RF3. El sistema debe permitir que el administrador de usuarios asigne un mínimo y un máximo de recursos disponibles para cada usuario(número de CPUs, tiempo límite de reserva, número de reservas).

RF4. El sistema debe permitir que cada usuario pueda modificar sus datos personales(nombre, apellidos, DNI, contraseña, número de teléfono, fecha de nacimiento).

RF5. El sistema debe proporcionar la capacidad de gestionar los clusters al propietario de los mismos (añadir, modificar, borrar).

RF6. El sistema debe tener la posibilidad de añadir y borrar máquinas.

RF7. El sistema debe proporcionar la capacidad de crear reservas siempre y cuando se respeten los límites del usuario (número de CPUs, tiempo máximo de reserva, límite de reservas) como de la máquina (número de CPUs).

RF8. El sistema debe mostrar los recursos disponibles de la máquina seleccionada antes de realizar la reserva.

RF9. El sistema debe ser capaz de comprobar que no se sobrepasen los límites mencionados anteriormente(**RF7**).

RF10. El sistema debe controlar la disponibilidad de recursos.

RF11. El sistema debe controlar la cantidad de recursos asignados por usuario.

RF12. El sistema debe permitir que se hagan múltiples reservas en la misma máquina siempre y cuando no se sobrepasen sus límites.

RF13. El sistema mostrará al usuario otra máquina del mismo cluster si no se puede realizar la reserva en una máquina por falta de recursos en esta(número de CPUs).

RF14. El sistema debe proporcionar la capacidad de modificar reservas al administrador de máquinas.

RF15. El sistema debe proporcionar la capacidad de borrar reservas.

RF16. El sistema debe generar un listado de usuarios y usarlo para su funcionamiento.

RF17. El sistema debe generar un listado de información sobre las máquinas y usarlo para su funcionamiento.

RF18. El sistema debe generar un listado de información sobre las reservas y usarlo para su funcionamiento.

Requisitos de información

RI1. El sistema debe guardar información sobre los usuarios(nombre, apellidos, DNI, correo, contraseña, número de teléfono, fecha de nacimiento y límites necesarios por el sistema como número de CPUs, tiempo máximo de reserva y número de reservas).

RI2. El sistema debe guardar información sobre los clusters(nombre, localización, propietario).

RI3. El sistema debe guardar información sobre las máquinas(tipo de cluster, nombre, cantidad de CPUs, características del hardware).

RI4. El sistema debe guardar información sobre las reservas (máquina, usuario, fecha de inicio y fin, CPUs, motivo de reserva).

RI5. El sistema debe guardar información sobre los recursos disponibles.

Requisitos no funcionales

RN1. El sistema debe controlar el inicio de sesión de usuarios mediante datos proporcionados a cada usuario.

RN2. El sistema debe brindar al usuario la capacidad de reservar recursos.

RN3. El sistema debe comprobar que no se excedan los límites.

RN4. El sistema debe tener un horario operativo (9:00-14:00 y 16:00-20:00 de lunes a viernes)

RN5. El sistema tendrá un periodo de mantenimiento.

RN6. El sistema debe garantizar que el uso de datos será privado por motivos de protección de datos.

6. Historias de usuario

Las historias de usuario nos permiten esquematizar y organizar las tareas que debe realizar nuestro programa, así como las funciones que este necesita. También nos indican la persona a la que la función va dirigida y cuales son las finalidades de estas dentro del programa. Las historias de usuario que hemos elegido son:

H1. Como administrador de usuarios **quiero** dar de alta a los usuarios **para** permitir que usen el sistema.

Como el sistema que se va a desarrollar estará reservado a un grupo de personas, los usuarios serán dados de alta por el administrador de usuarios con la información que se le solicite al miembro investigador que solicite usar el sistema. Además se distinguirá entre tres tipos de usuarios/roles diferentes, el usuario que usará el sistema, el administrador de usuarios y el administrador de máquinas. Estas dos últimas tendrán privilegios respecto a los usuarios ya que serán los encargados de hacer que el sistema funcione correctamente.

Responsable: Martín del Río

Criterios de validación:

-Todos los usuarios deben tener la información personal completa sobre el miembro del grupo investigador que la usará.

- Ninguna persona externa al grupo de investigación tiene acceso a la reserva de recursos.
- Hay tres tipos de usuarios con funcionalidades diferentes.

Conversaciones:

Se ha considerado que para dar de alta a los usuarios será obligatorio disponer de los siguientes datos del miembro investigador:

- Nombre
- Apellidos
- Correo de la UCO
- DNI
- Contraseña

Y como parámetros optativos:

- Número de teléfono
- Fecha de nacimiento

Además el administrador de usuarios deberá establecer los límites del usuario tales como número de CPUs que puede reservar, cantidad de reservas simultáneas que puede tener y tiempo límite que puede tener reservado.

Prioridad: Alta

Estimación en días: 2

Sprint: 5

H2. Como administrador de usuarios **quiero** modificar la información de los usuarios **para** establecer sus límites.

El administrador de usuarios debe ser capaz de modificar los datos que regulan el funcionamiento del sistema para cada usuario, asignando un límite de CPUs, tiempo máximo de reserva y número de reservas simultáneas, ya que estos serán diferentes para cada investigador en función de su rol dentro de la investigación.

Responsable: Martín Del Río

Criterios de validación:

- El administrador de usuarios puede modificar la información de cualquier usuario.
- El administrador de usuarios es capaz de modificar información privilegiada de los usuarios, tal como los límites de recursos(número de CPUs, tiempo máximo de reserva, número de reservas).
- Los límites serán establecidos por el administrador de usuario.

Conversaciones:

Este tipo de permisos será de uso exclusivo para los administradores de usuarios.

Prioridad: Media

Estimación en días: 2

Sprint: 6

H2.1. Como administrador de usuarios **quiero** modificar los datos de los usuarios **para** cambiar su información.

El administrador de usuarios debe ser capaz de modificar los datos personales de los usuarios.

Responsable: Martín del Río

Criterios de validación:

-El administrador de usuarios es capaz de modificar información personal de todos los usuarios.

-El administrador de usuarios es capaz de modificar información privilegiada de los usuarios, tal como los límites de recursos(H2).

-El sistema deberá comprobar que los datos requeridos para su correcto funcionamiento sean válidos antes de confirmar la modificación.

Conversaciones:

Se recomienda que, para confirmar la modificación de datos de usuario, aquellos que sean obligatorios estén adecuadamente introducidos.

Prioridad: Alta

Estimación en días: 1

Sprint: 5

H2.2. Como usuario quiero modificar mis datos para cambiar la información personal.

Debido a que el usuario es dado de alta por un administrador de usuarios, cada usuario debe tener la posibilidad de modificar la información personal disponible tal como nombre, apellidos, correo, DNI, contraseña, número de teléfono, fecha de nacimiento.

Responsable: Martín del Río

Criterios de validación:

-Cada usuario es capaz de modificar su información personal.

-El sistema deberá comprobar que los datos requeridos para su correcto funcionamiento sean válidos antes de confirmar la modificación.

Conversaciones:

Se recomienda que, para confirmar la modificación de datos, aquellos que son obligatorios deben estar adecuadamente introducidos.

Prioridad: Baja

Estimación en días: 1

Sprint: 6

H3. Como usuario quiero iniciar sesión para usar el sistema.

Para hacer que el sistema sea funcional se ha de permitir el inicio de sesión al usuario.

Responsable: Martín del Río

Criterios de validación:

-Cada usuario puede entrar al sistema con sus credenciales(correo y contraseña).

-Se puede controlar el acceso al sistema.

-Cada usuario que entra al sistema ha sido dado de alta por el administrador de usuarios.

Conversaciones:

Prioridad: Media

Estimación en días: 2

Sprint: 6

H3.1. Como administrador del sistema quiero iniciar sesión para poder administrarlo.

Para hacer que el sistema sea funcional se ha de permitir el inicio de sesión al administrador de usuarios y al administrador de máquinas.

Responsable: Martín del Río

Criterios de validación:

- Cada usuario puede entrar al sistema con sus credenciales(correo y contraseña).
- Se puede controlar el acceso al sistema.
- Cada usuario que entra al sistema ha sido dado de alta por el administrador de usuarios.

Conversaciones:

Prioridad: Media

Estimación en días: 2

Sprint: 6

H4. Como usuario quiero realizar reservas para la ejecución de experimentos.

Cuando el usuario entre en el sistema debe ser capaz de reservar recursos para la investigación pertinente. Para ello deberá introducir la información requerida por el sistema para que este compruebe la validez de la reserva.

Responsable: Kamal Abdelkader

Criterios de validación:

- El sistema comprueba que los datos introducidos por el usuario para realizar la reserva no sobrepasan el límite del usuario(número de CPUs, tiempo de reserva, número de reservas activas).
- El sistema comprueba que la máquina seleccionada tenga CPUs disponibles en el tiempo seleccionado por el usuario.
- En caso de que no se pueda realizar la reserva debido a los límites del usuario, el sistema mostrará un mensaje de error.
- En caso de que no se pueda realizar la reserva debido a los límites de la máquina, el sistema mostrará otra máquina del mismo cluster.
- Se debe de tener información sobre la reserva a nivel de usuario(número de CPUs ya reservadas, fecha de inicio y fin de la reserva, número de reservas activas, motivo de la reserva).

Conversaciones:

Se ha considerado que para realizar reservas se tenga que introducir la cantidad de recursos que ya estén asignados a ese usuario y el límite de tiempo que va a necesitar para la nueva reserva.

Prioridad: Alta

Estimación en días: 4

Sprint: 5

H5. Como administrador de máquinas quiero modificar las reservas para gestionar mejor el sistema.

Debido a que cada usuario tiene unos límites y hay una cantidad de máquinas que se pueden usar con límites de hardware en cada una de ellas, es necesario que el administrador de máquinas pueda modificar una reserva actual para tener un acceso a los recursos cuando sea necesario.

Responsable: Kamal Abdelkader

Criterios de validación:

- El administrador de máquinas tiene la opción de modificar una reserva de cualquier usuario.

-El administrador de máquinas tiene la opción de borrar una reserva de cualquier usuario.

Conversaciones:

Para la modificación de la reserva se ha de tener en cuenta que no se sobrepasen los límites del sistema ni del usuario antes de que la modificación se confirme.

Prioridad: Baja

Estimación en días: 2

Sprint: 6

H5.1 Como usuario **quiero** borrar mis reservas **para** poder reservar cuando me sea más rentable.

Debido a que existe un límite de recursos y tiempo para cada usuario es necesario poder borrar una reserva actual para tener un acceso a los recursos cuando sea necesario.

Responsable: Kamal Abdelkader

Criterios de validación:

-El usuario tiene la opción de borrar una reserva propia.

Conversaciones:

Para confirmar que se borró una reserva se debe de tener en cuenta que el usuario que la borre sea el mismo que la ha creado.

Prioridad: Baja

Estimación en días: 2

Sprint: 7

H6. Como administrador de usuarios **quiero** disponer de información de los usuarios **para** comprobar que cada uno dispone de la información necesaria por el sistema.

Debido a que el sistema requiere información personal de los usuarios para poder realizar las reservas, el administrador de usuarios necesita poder acceder a dicha información para hacer que se cumplan los límites del sistema.

Responsable: Jorge Benitez

Criterios de validación:

-El administrador de usuarios puede acceder a la información de cada usuario, tanto a su información personal como a los límites.

-Cada uno de los usuarios ha sido dado de alta por un administrador de usuarios previamente.

-En todo momento cada uno de los usuarios tiene toda la información requerida por el sistema.

Conversaciones:

Se debe de tener en consideración que esta operación sea exclusiva para el administrador de usuarios.

Prioridad: Media

Estimación en días: 2

Sprint: 5

H7. Como administrador de máquinas **quiero** disponer de información de las reservas **para** hacer que funcionen correctamente.

Para hacer que el sistema funcione adecuadamente el administrador de máquinas debe tener acceso a los datos sobre las reservas actuales y la información que proporcione el usuario a la hora de realizar una reserva (número de CPUs, fecha de reserva, número de reservas actuales), para así poder evitar que se sobrepasen los límites.

Responsable: Jorge Benitez

Criterios de validación:

- Cualquier reserva actual o pasada tendrá una descripción detallada del motivo por el cual se hace.
- Todas las reservas tendrán especificadas la cantidad de recursos y la máquina en la cual se hacen.
- Ningún usuario tendrá en algún momento una cantidad de recursos o tiempo superior al límite.

Conversaciones:

Se debe de tener en consideración que esta operación sea exclusiva para los administradores de máquinas.

Prioridad: Media

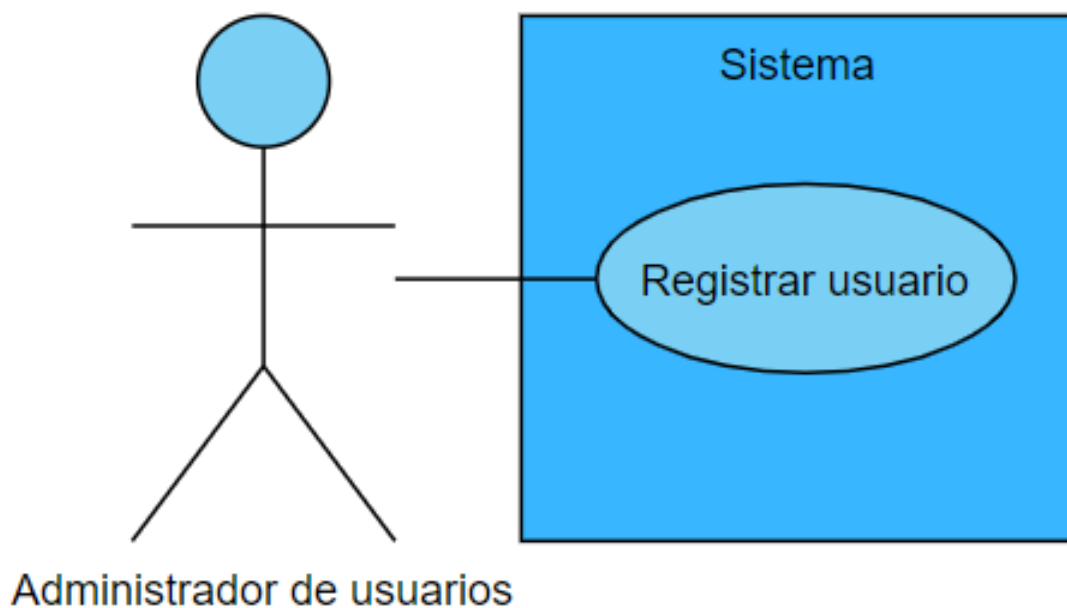
Estimación en días: 2

Sprint: 6

7. Casos de uso

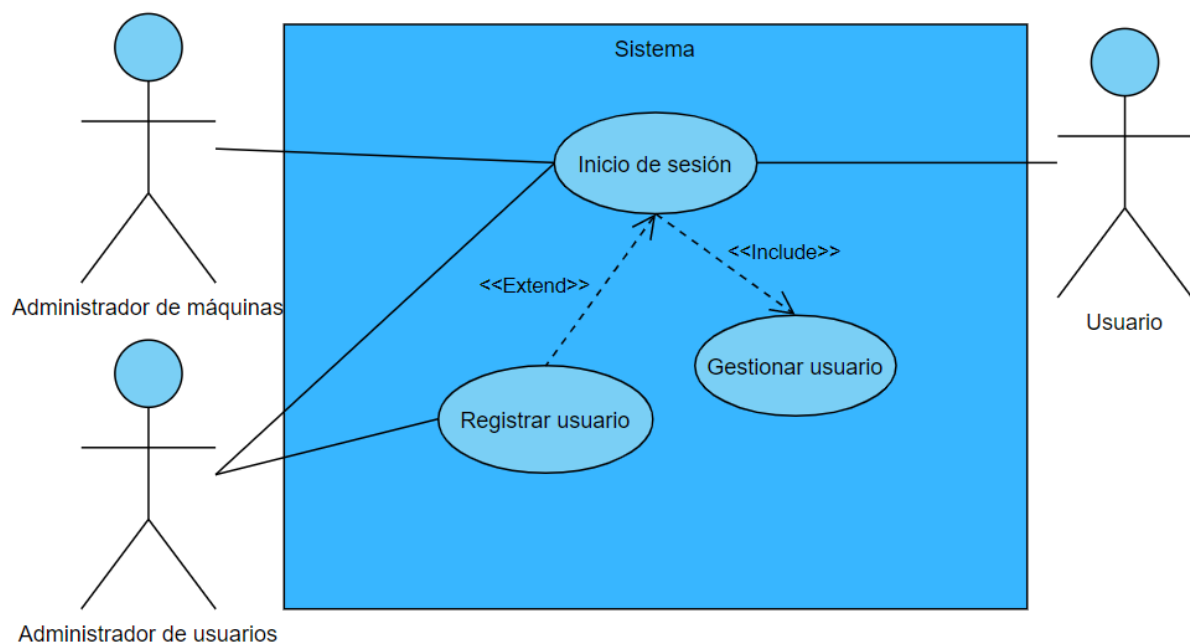
<u>CASO DE USO</u>	Registro de usuarios
<u>IDENTIFICADOR</u>	CU1
<u>OBJETIVO</u>	Brindar a los miembros del grupo investigador la posibilidad de usar el sistema.
<u>CONTEXTO</u>	Debido a que el sistema de reservas está restringido a un grupo de usuarios en concreto, los usuarios serán dados de alta por el administrador de usuarios con los datos personales que se recojan del miembro investigador.
<u>ACTOR PRINCIPAL</u>	Administrador de usuarios
<u>ESCENARIO PRINCIPAL</u>	<ol style="list-style-type: none">1. Un miembro del equipo de investigación solicita usar el programa.2. El administrador de usuarios recoge su información personal para dar de alta al usuario.3. El administrador de usuarios asigna unos límites de recursos disponibles

	<p>para dicho usuario en función del rol dentro del grupo de investigación.</p> <p>4. El administrador de usuarios notifica al usuario que ya puede usar el sistema.</p>
<u>EXTENSIONES</u>	<p>1. Si no se han recopilado todos los datos necesarios del usuario no se le podrá dar de alta en el sistema.</p>



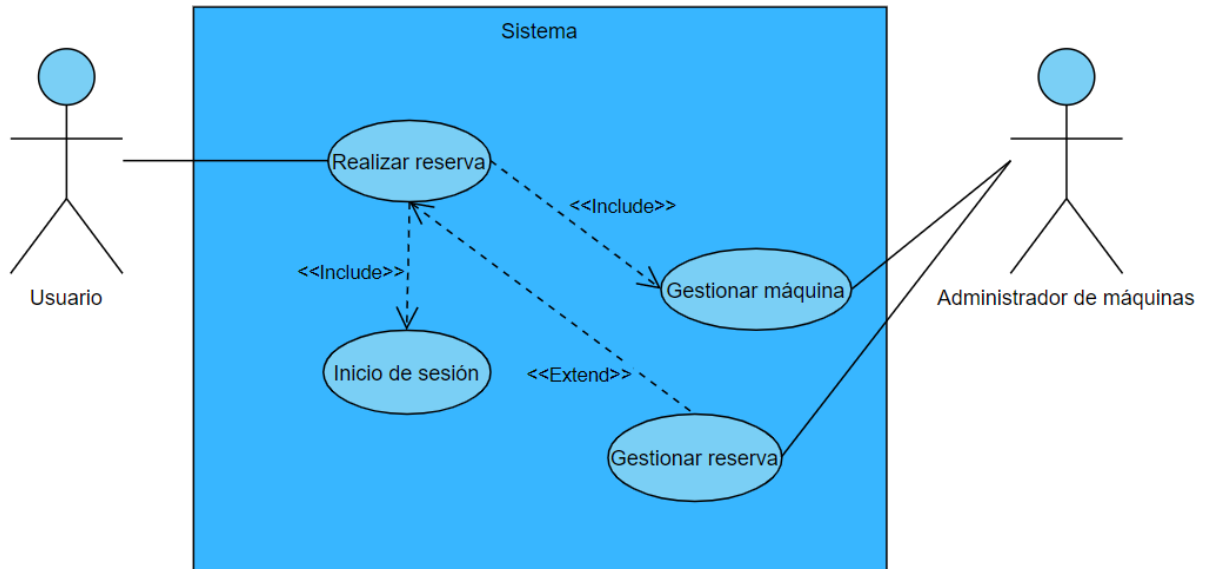
<u>CASO DE USO</u>	Inicio de sesión
<u>IDENTIFICADOR</u>	CU2
<u>OBJETIVO</u>	Todos los usuarios que quieran usar el sistema y previamente han proporcionado sus datos al administrador de usuarios deben ser capaces de acceder al sistema.
<u>CONTEXTO</u>	Para acceder al sistema se deben de introducir las credenciales requeridas por el sistema(correo y contraseña), las cuales han sido proporcionadas por el administrador de usuarios.
<u>ACTOR PRINCIPAL</u>	Usuario y administradores

<u>ESCENARIO PRINCIPAL</u>	<ol style="list-style-type: none"> 1. Cualquier usuario o administrador intenta acceder al sistema. 2. Introduce los datos requeridos por el sistema, tales como correo y contraseña. 3. El sistema comprueba que los datos introducidos corresponden a los de algún usuario del mismo. 4. Se confirma el inicio de sesión.
<u>EXTENSIONES</u>	<ol style="list-style-type: none"> 1. En caso de que los datos introducidos no correspondan a ningún usuario del sistema, este devolverá un mensaje de error y no se podrá acceder.



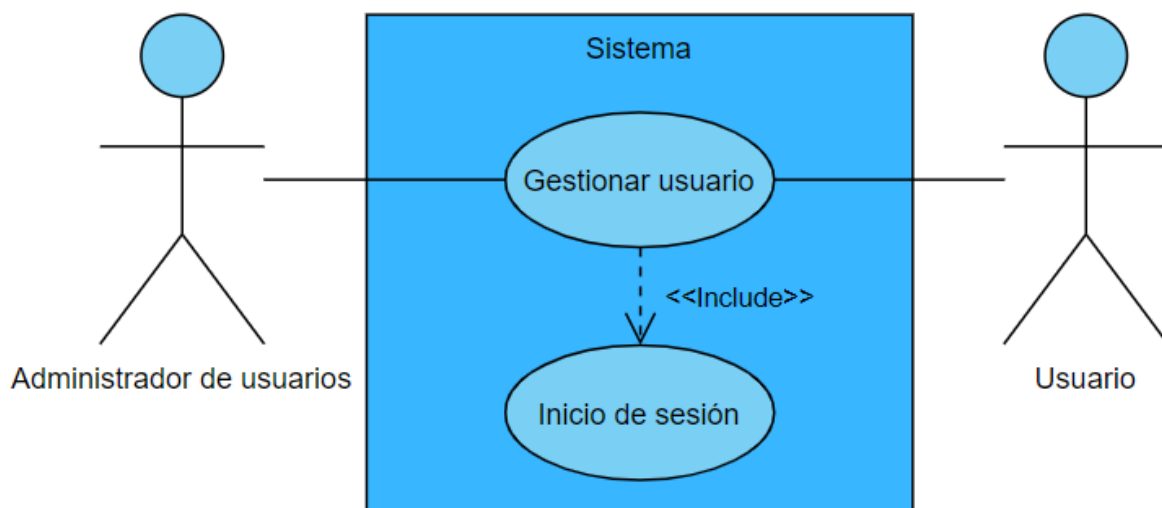
<u>CASO DE USO</u>	Realizar reserva
<u>IDENTIFICADOR</u>	CU3
<u>OBJETIVO</u>	La reserva debe quedar registrada en el sistema y tener una fecha posterior a la actual.
<u>CONTEXTO</u>	La reserva de máquinas se ha de realizar cumpliendo con todos los límites, tanto de

	<p>usuario como de máquina.</p> <p>Además se debe de complementar toda la información requerida por el sistema para realizar la reserva y esta debe quedar guardada.</p>
<u>ACTOR PRINCIPAL</u>	Usuario
<u>ESCENARIO PRINCIPAL</u>	<ol style="list-style-type: none"> 1. Un usuario rellena los datos solicitados para realizar la reserva. 2. El sistema comprueba que no se sobrepasan los límites del usuario(número de CPUs, tiempo límite de reservas, cantidad de reservas simultáneas). 3. El sistema comprueba los límites de la máquina(necesita tener CPUs suficientes disponibles en la fecha que solicite el usuario). 4. Si todo funciona correctamente, el sistema realiza la reserva y actualiza los datos de la máquina y del usuario.
<u>EXTENSIONES</u>	<ol style="list-style-type: none"> 1. El sistema no puede comprobar los límites de usuario o de máquina y devuelve un error. 2. El sistema comprueba los datos del usuario y notifica que no puede realizar la reserva debido a que sobrepasa sus límites. 3. El sistema comprueba los datos de la máquina y notifica que no se puede realizar la reserva debido a que la máquina seleccionada no tiene disponible suficientes recursos durante el periodo seleccionado y muestra otra máquina.



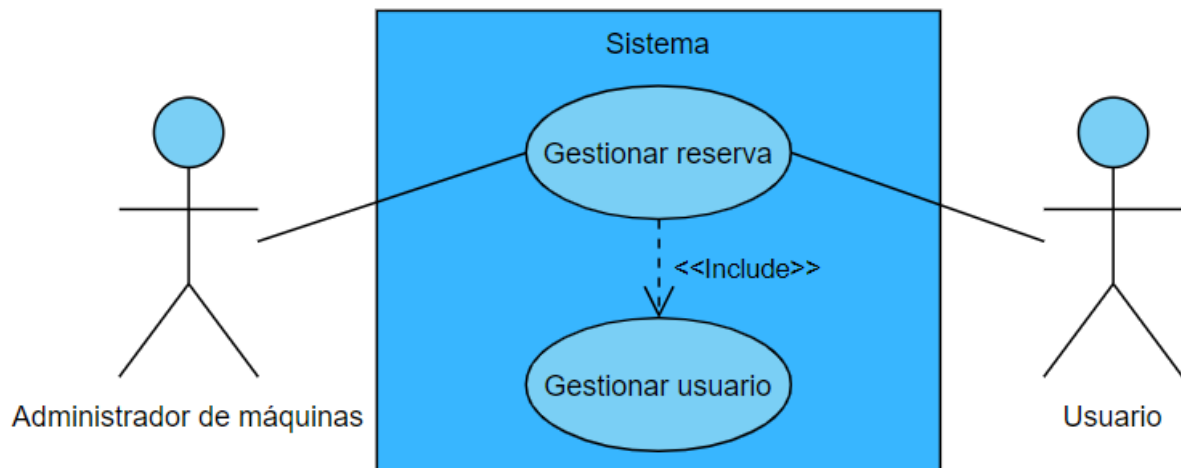
<u>CASO DE USO</u>	Gestionar usuario
<u>IDENTIFICADOR</u>	CU4
<u>OBJETIVO</u>	El administrador de usuarios debe ser capaz de comprobar los datos de los usuarios. Cada usuario debe ser capaz de modificar sus datos personales.
<u>CONTEXTO</u>	Debido a que cada usuario tiene unos límites impuestos por el administrador de usuarios, este debe ser capaz de acceder a estos datos que se encuentren dentro del sistema. De este modo se podrán hacer las comprobaciones necesarias respecto a los usuarios. Además cada usuario debe tener la posibilidad de modificar su información personal.
<u>ACTOR PRINCIPAL</u>	Administrador de usuarios y usuarios
<u>ESCENARIO PRINCIPAL</u>	Para administrador de usuarios

	<ol style="list-style-type: none"> 1. El administrador de usuarios intenta ver los datos de cualquier usuario. 2. El sistema accede al almacenamiento de datos y muestra los datos de todos los usuarios. 3. El administrador de usuarios selecciona uno de los usuarios en caso de que quiera modificar sus datos. <p style="text-align: center;">Para usuario</p> <ol style="list-style-type: none"> 1. El usuario intenta ver sus datos personales. 2. El sistema accede al almacenamiento de datos y muestra sus datos personales. 3. El usuario tiene la posibilidad de modificar estos datos.
<u>EXTENSIONES</u>	<ol style="list-style-type: none"> 1. El sistema no es capaz de acceder a los datos de los usuarios por lo cual muestra un mensaje de error. 2. El sistema no es capaz de mostrar los datos del usuario que se ha seleccionado en concreto. 3. El sistema no es capaz de actualizar los datos de usuario.



<u>CASO DE USO</u>	Gestionar reserva
---------------------------	-------------------

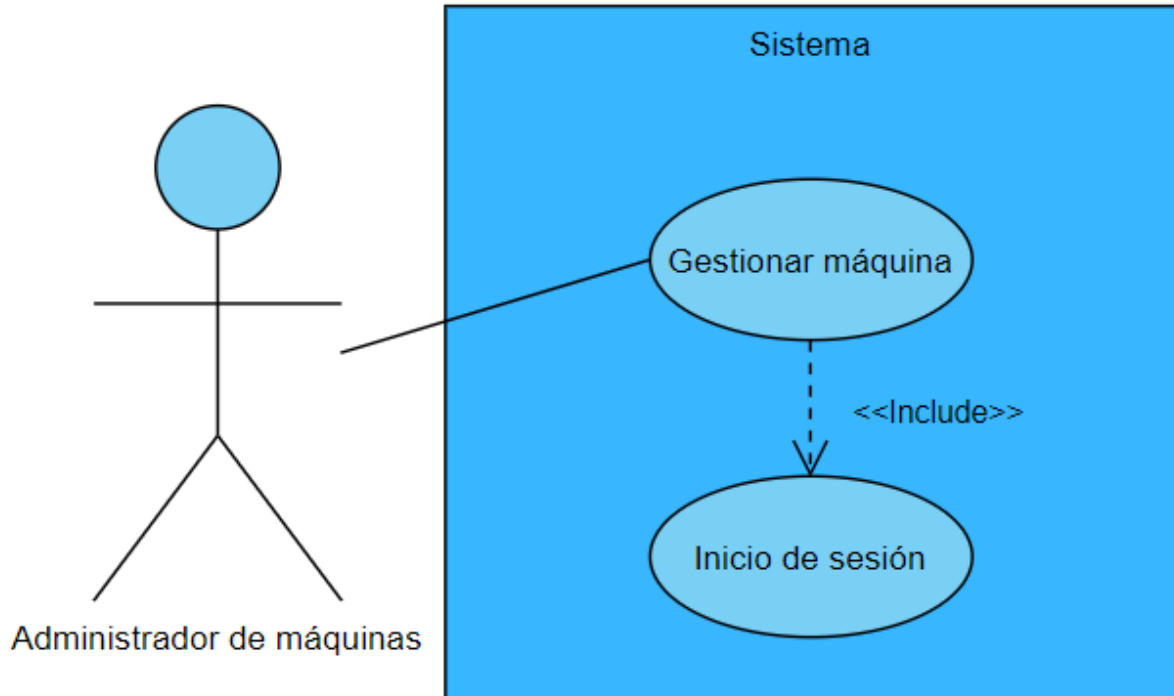
<u>IDENTIFICADOR</u>	CU5
<u>OBJETIVO</u>	Un administrador de máquinas debe ser capaz de gestionar las reservas que se han realizado.
<u>CONTEXTO</u>	Debido a que cada máquina tiene unos límites(número de CPUs), el administrador de máquinas debe ser capaz de acceder a los datos de las reservas para gestionar los límites de reservas.
<u>ACTOR PRINCIPAL</u>	Administrador de máquinas
<u>ESCENARIO PRINCIPAL</u>	<ol style="list-style-type: none"> 1. El administrador de máquinas intenta ver los datos de una reserva. 2. El sistema accede al almacenamiento de datos y muestra los datos de todas las reservas. 3. El administrador de máquinas selecciona una reserva en caso de que quiera modificar sus datos. 4. El sistema comprueba que los datos tras la modificación son correctos(los datos necesarios para una reserva están completos y no entran en conflicto con otras reservas). 5. El sistema confirma que se ha realizado la modificación.
<u>EXTENSIONES</u>	<ol style="list-style-type: none"> 1. El sistema no es capaz de acceder a los datos de las reservas por lo cual muestra un mensaje de error. 2. El sistema comprueba que los nuevos datos son incorrectos por lo cual no modifica la reserva y devuelve un error.



<u>CASO DE USO</u>	Gestionar máquina
<u>IDENTIFICADOR</u>	CU6
<u>OBJETIVO</u>	Un administrador de máquinas debe ser capaz de gestionar los datos pertenecientes a una máquina.
<u>CONTEXTO</u>	Debido a que cada máquina tiene unos recursos disponibles diferentes(nombre, número de CPUs, ubicación, propietario) el administrador de máquinas debe ser capaz de gestionar dicha máquina para acceder a sus datos.
<u>ACTOR PRINCIPAL</u>	Administrador de máquinas
<u>ESCENARIO PRINCIPAL</u>	<ol style="list-style-type: none"> 1. El administrador de máquinas intenta acceder a los datos de las máquinas. 2. El sistema muestra un listado de información sobre las máquinas existentes en los diferentes tipos de cluster.

EXTENSIONES

1. El sistema no es capaz de acceder a los datos de las máquinas por lo cual muestra un mensaje de error.



8. Diagrama de clases

Clase 1 Usuario			
Esta clase nos permitirá almacenar la información personal del usuario, así como sus límites.			
Atributos			
-	nombre	String	Nombre del usuario.
-	apellidos	String	Apellidos del usuario.
-	correo	String	Correo de la UCO.
-	DNI	String	DNI
-	password	String	Contraseña requerida para iniciar sesión.

-	telefono	Int	Número de teléfono.
-	fechaNacimiento	String	Fecha de nacimiento con formato DD/MM/YYYY
-	limiteCPUs	Int	Cantidad de CPUs que puede reservar.
-	limiteReservas	Int	Cantidad de reservas simultáneas que puede tener el usuario.
-	limiteTiempo	Float	Cantidad de horas disponibles para reservar por el usuario.
-	numeroCPUs	int	Cantidad de CPUs ya ocupadas por el usuario.
-	numeroReservas	int	Cantidad de reservas ya ocupadas por el usuario.
-	numeroTiempo	Float	Cantidad de tiempo ya ocupado por el usuario.
-	rol	String	Determina qué tipo de rol tiene dentro del sistema(usuario, administrador de usuarios, administrador de máquinas)
Operaciones			
+	iniciarSesion	bool	Permite al usuario iniciar sesión en el sistema.
+	verDatos	void	Muestra los datos personales del usuario.
+	modificarDatos	bool	Modifica los datos personales del propio usuario.
+	mostrarReservas	void	Muestra los datos de las reservas activas del usuario.

Clase 2	Administrador de usuarios
<p>Esta clase hereda de la clase Usuario, ya que el administrador de usuarios sigue siendo un usuario pero tiene más funcionalidad. Este se encarga de administrar los usuarios por lo cual necesita tener acceso a ellos.</p>	

Atributos			
Operaciones			
+	añadirUsuario	bool	Permite dar de alta a usuarios en el sistema.
+	modificarUsuario(String correo)	bool	Permite modificar los datos personales de cualquier usuario.
+	asignarLimitesUsuario(String correo)	bool	Permite asignar los límites para cada usuario(número de CPUs, tiempo de reserva, número de reservas simultáneas).
+	verUsuarios	void	Muestra los datos de todos los usuarios.
+	verUsuario(String correo)	void	Muestra los datos de un usuario.

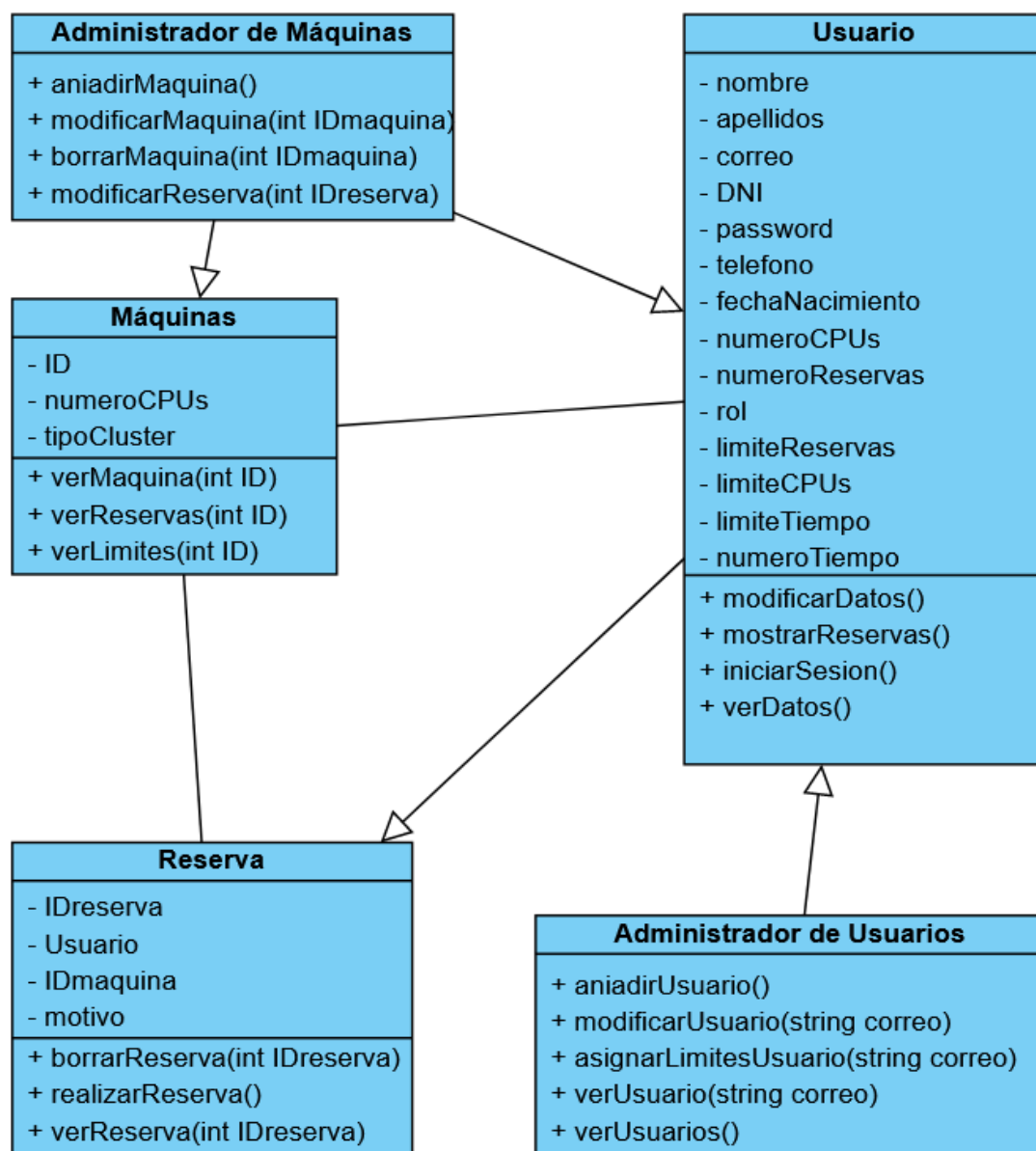
Clase 3 Administrador de máquinas			
Esta clase hereda de la clase Usuario, ya que el administrador de máquinas sigue siendo un usuario pero tiene más funcionalidad. Este se encarga de administrar las máquinas por lo cual necesita tener acceso a ellas.			
Atributos			
Operaciones			
+	añadirMáquina	bool	Permite añadir máquinas al sistema.
+	modificarMáquina(int IDmáquina)	bool	Permite modificar los datos de una máquina.
+	borrarMáquina(int ID)	bool	Permite borrar máquinas del sistema.
+	modificarRes	bool	Modifica una reserva (esto solo puede realizarlo el

	erva(int IDreserva)		administrador de usuarios).
--	------------------------	--	-----------------------------

Clase 4 Máquina			
Esta clase almacena la información de las máquinas, así como sus recursos disponibles y las reservas realizadas.			
Atributos			
-	ID	int	Número de identificación de la máquina en el cluster.
-	numeroCPUs	int	Número de CPUs de la máquina.
-	tipoCluster	int	Tipo del cluster al que pertenece la máquina.
Operaciones			
+	verMaquina(int ID)	void	Muestra los datos de una máquina.
+	verReservas(int ID)	void	Muestra las reservas de la máquina en el sistema.
+	verLimites(int ID)	void	Muestra los límites de hardware de la máquina(número de CPUs).

Clase 5 Reserva			
Esta clase almacena las reservas realizadas, así como la máquina en la que se realiza la reserva y el usuario que la solicita.			
Atributos			
-	IDreserva	int	Número de identificación de la reserva.
-	IDmaquina	int	ID de la máquina en la cual se realiza la reserva.

-	usuario	String	Correo del usuario que realiza la reserva.
-	motivo	String	Motivo por el cual se realiza la reserva.
Operaciones			
+	realizarReserva	bool	Permite realizar una reserva al usuario.
+	borrarReserva (int IDreserva)	bool	Permite al usuario borrar una reserva.
+	verReserva(int IDreserva)	void	Muestra los datos asociados a una reserva.



9. Matriz de trazabilidad

9.1 Requisitos funcionales frente a casos de uso

	CU1	CU2	CU3	CU4	CU5	CU6
RF1	X			X		
RF2		X		X		
RF3				X		
RF4				X		

RF5						X
RF6						X
RF7			X			
RF8			X		X	X
RF9			X	X		
RF10			X		X	X
RF11			X	X		
RF12			X			
RF13			X			X
RF14					X	
RF15					X	
RF16	X	X	X	X		
RF17			X			X
RF18			X		X	

9.2 Casos de uso frente a clases

	CL1	CL2	CL3	CL4	CL5
CU1		x			
CU2	x	x	x		
CU3	x				x
CU4		x			

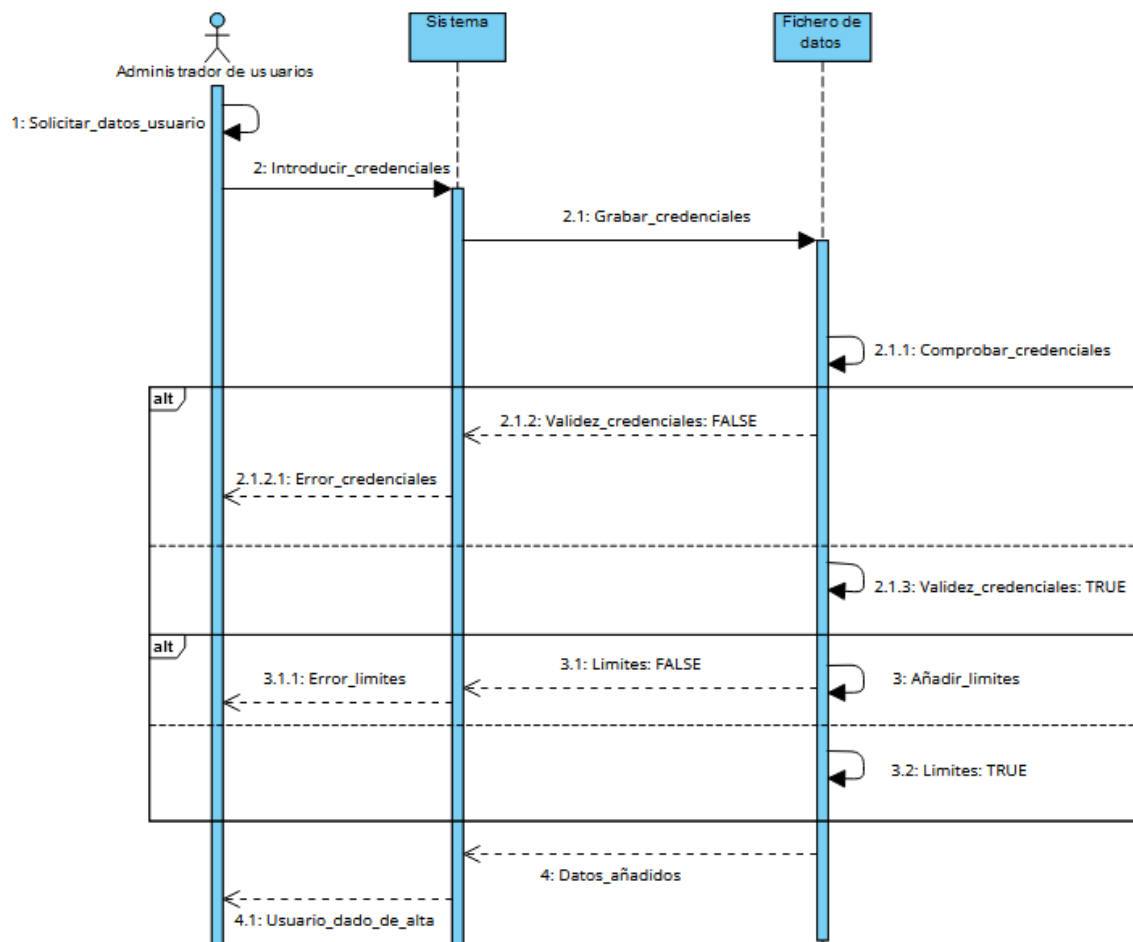
CU5		x			x
CU6			x	x	

10. Diagrama de secuencia

Los diagramas de secuencia nos ayudan a entender el procedimiento que hará el sistema para llevar a cabo los casos de uso de una forma visual. Para ello los crearemos con Visual Paradigm.

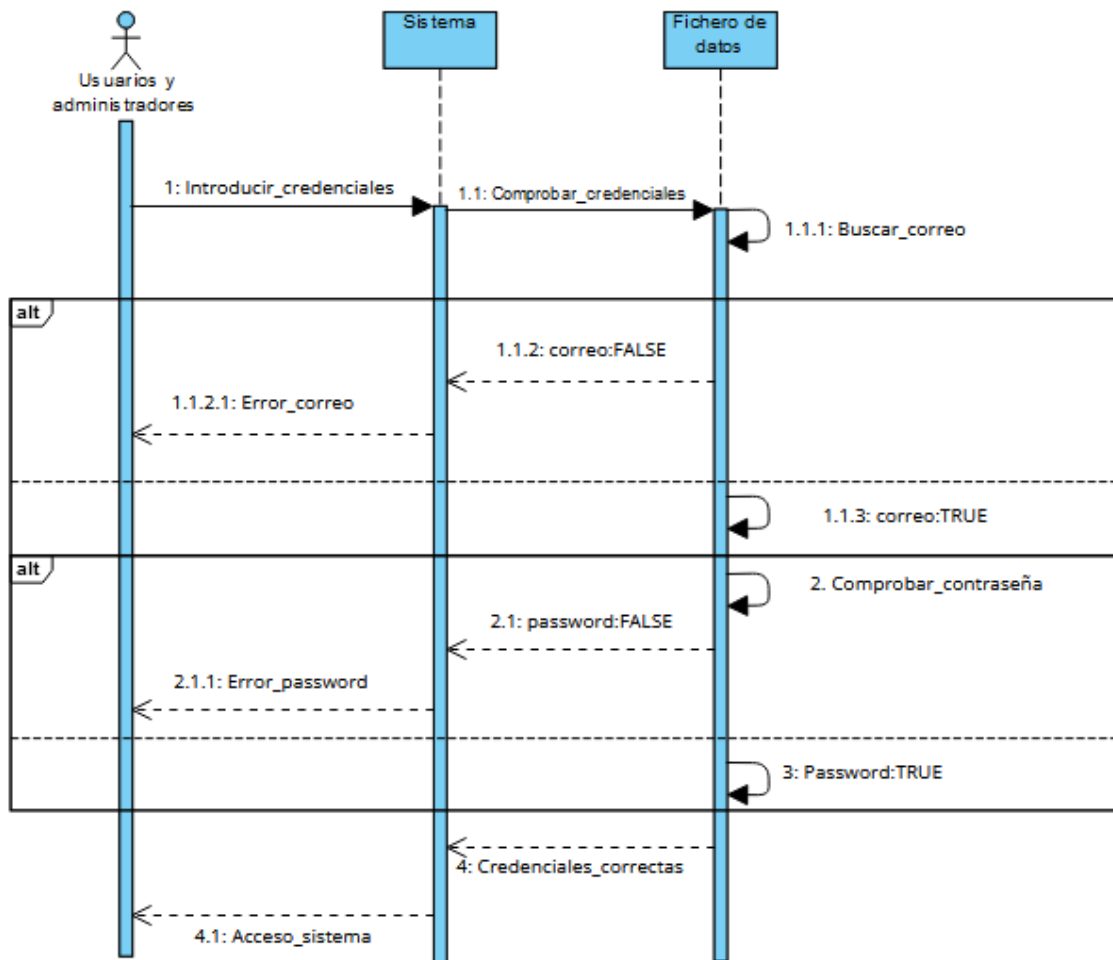
- **Diagrama de secuencia con el caso de uso 1 -Registro de usuarios**

Para el primer caso de uso, el administrador de usuarios como responsable es el encargado de interactuar con el sistema. Debe de recopilar la información personal requerida por el sistema(**H1**) e introducirla al fichero de datos de los usuarios. Posteriormente debe introducir los límites de cada usuario(**H2**).



- **Diagrama de secuencia con el caso de uso 2 -Inicio de sesión**

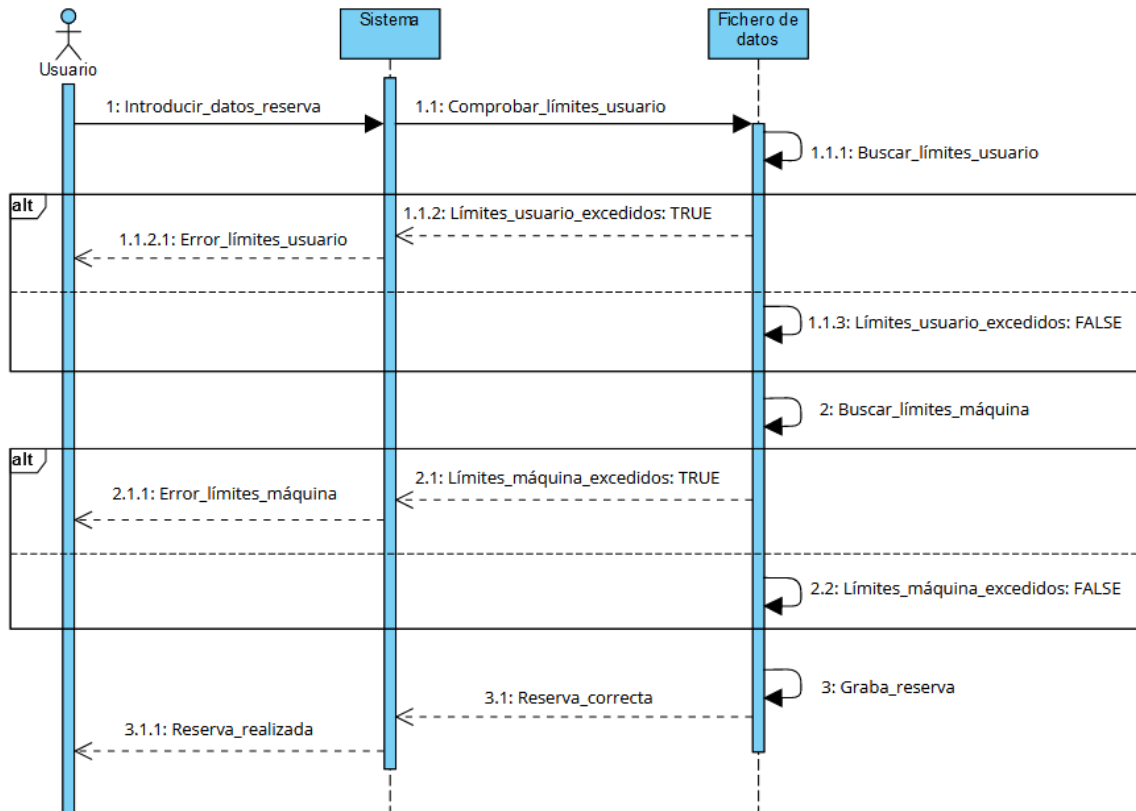
Este caso de uso consiste en el inicio de sesión para usar el sistema. Para ello, tanto usuarios como administradores deben de introducir sus credenciales(correo y contraseña) y el sistema debe comprobar en el fichero de datos que estos correspondan a un usuario del sistema. Tras hacer esto, el sistema tiene dos modos de actuar, cuando las credenciales son correctas permitirá el acceso al sistema pero, en caso contrario, devolverá un error de correo o de contraseña ya que no corresponden a un usuario.



- **Diagrama de secuencia con el caso de uso 3 -Realizar reserva**

En este diagrama de secuencia se ha diseñado lo que sería el realizar una reserva. Para que esto ocurra, el usuario debe introducir los datos necesarios para la reserva (el motivo, máquina a reservar, fecha de inicio y fin de la reserva, cantidad de CPUs que quiere reservar). Posteriormente, el sistema comprobará que el usuario no sobrepasa sus límites para reservar añadiendo los nuevos datos de la reserva que quiere realizar. En caso de tener disponibilidad, comprobará que la máquina seleccionada tenga CPUs disponibles en la fecha

seleccionada. En caso de que todo sea correcto, se actualizarán en los ficheros de datos los recursos reservados a nivel de usuario y de máquina y se grabará la reserva. El sistema procederá a confirmar la reserva al usuario.



11. Implementación

Para la implementación del sistema hemos usado la IDE de eclipse orientada para el lenguaje de C++. Además hemos usado GitHub para tener un seguimiento sobre las actualizaciones del proyecto a la hora de hacer su implementación. El repositorio de GitHub está organizado de la forma más cómoda posible y se ha invitado al cliente con el fin de que este pueda comprobar el progreso de desarrollo del sistema. Se adjunta enlace del repositorio del equipo:

<https://github.com/Martindrj/Equipo-205>

12. Pruebas

Para comprobar que nuestro programa funciona correctamente tal y como las peticiones del cliente, a continuación se muestran algunas de las pruebas realizadas mediante googletest:

Pruebas unitarias

ID	1	Fecha	18/12/2021
Título de la prueba	Maquina: setDatos		
Responsable de la prueba	Kamal Abdelkader Haddu		
Descripción			
Se ha comprobado el funcionamiento de la función setDatos perteneciente a la clase de Maquina.			
Prerrequisitos			
El usuario que realice esta acción debe ser un administrador de máquinas.			
Entradas			
ID, tipoCluster, LimiteCPUs, NumeroCPUs			
Acciones a realizar			
Se va a crear una máquina mediante los valores de entrada y se va a guardar en el fichero de máquinas.			
Salida esperada			
True			
Salida obtenida			
True			
Observaciones			

ID	2	Fecha	19/12/2021
Historia de usuario	H6		
Título de la prueba	Usuario: getDatosByEmail		

Responsable de la prueba	Jorge Benítez Alonso
Descripción	
Se ha comprobado el funcionamiento de la función <code>getDatosByEmail</code> perteneciente a la clase de <code>Usuario</code> .	
Prerrequisitos	
El usuario que realice esta acción debe ser un administrador de usuarios. El correo introducido pertenece a un usuario del sistema y está guardado en el fichero de usuarios.	
Entradas	
correo	
Acciones a realizar	
Se recorre el fichero de usuarios buscando que el correo introducido coincida con el de algún usuario.	
Salida esperada	
True	
Salida obtenida	
True	
Observaciones	

ID	3	Fecha	19/12/2021
Historia de usuario	H6		
Título de la prueba	Usuario: <code>getDatosByEmail</code> segunda prueba		

Responsable de la prueba	Jorge Benítez Alonso
Descripción	
Se ha comprobado el funcionamiento de la función <code>getDatosByEmail</code> perteneciente a la clase de <code>Usuario</code> .	
Prerrequisitos	
El usuario que realice esta acción debe ser un administrador de usuarios. El correo introducido no pertenece a ningún usuario del sistema.	
Entradas	
correo	
Acciones a realizar	
Se recorre el fichero de usuarios buscando que el correo introducido no coincida con el de algún usuario.	
Salida esperada	
False	
Salida obtenida	
False	
Observaciones	

ID	4	Fecha	19/12/2021
Título de la prueba	Usuario: comprobarLimites		
Responsable de la prueba	Martín del Río Jiménez		

Descripción
Se ha comprobado el funcionamiento de la función comprobarLimites perteneciente a la clase de Usuario.
Prerrequisitos
El usuario debe estar registrado en el sistema previamente. El usuario está intentando realizar una reserva.
Entradas
us, numeroCPUs, numeroDias
Acciones a realizar
Se comprueba que el usuario no va a sobrepasar sus límites en el momento de solicitar una nueva reserva.
Salida esperada
True
Salida obtenida
True
Observaciones

ID	5	Fecha	16/12/2021
Título de la prueba	Maquina: comprobarLimites		
Responsable de la prueba	Martín del Río Jiménez		

Descripción
Se ha comprobado el funcionamiento de la función comprobarLimites perteneciente a la clase de Maquina.
Prerrequisitos
La máquina debe estar registrada en el sistema previamente. El usuario está intentando realizar una reserva.
Entradas
numeroCPUs
Acciones a realizar
Se comprueba que los límites de la máquina no se van a sobrepasar en el momento de realizar una nueva reserva
Salida esperada
True
Salida obtenida
True
Observaciones

ID	6	Fecha	20/12/2021
Historia de usuario	H4		
Título de la prueba	Reserva: realizarReserva		

Responsable de la prueba	Kamal Abdelkader Haddu
Descripción	
Se ha comprobado el funcionamiento de la función realizarReserva perteneciente a la clase de Reserva.	
Prerrequisitos	
El usuario debe estar registrado en el sistema previamente. La máquina sobre la cual se realiza la reserva debe estar disponible.	
Entradas	
us, maq numeroCPUs, diaInicio, mesInicio, anioInicio, diaFin, mesFin, anioFin, numeroDias.	
Acciones a realizar	
Se pide por pantalla los datos sobre la nueva reserva. Se comprueba que no se sobrepasen los límites del usuario que solicita la reserva ni los límites de la máquina en la cual se intenta realizar. Guardar la información de la reserva en el fichero de reservas.	
Salida esperada	
False	
Salida obtenida	
False	
Observaciones	

13. Bibliografía

Moodle de la Universidad de Córdoba:

<https://moodle.uco.es/m2122> (Hemos utilizado los apuntes de Ingeniería de Software como el de Programación Orientada a Objetos para aprender a usar el lenguaje C++)

Página web para el diseño de los Casos de Uso y los diagramas:

<https://online.visual-paradigm.com/>

Página de YouTrack para el seguimiento del proyecto

<https://uco-is2022-eq205.myjetbrains.com/>

GitHub

<https://github.com>

<https://github.com/Martindrj/Equipo-205>

<https://stackoverflow.com/questions/6219878/stack-overflow-c> (El apartado de

C++)