

TRABAJO FINAL DE CARRERA

Sistema de gestión integral en I+D

MEMORIA - Ingeniería del software

Agradecimientos

Me gustaría dedicar este proyecto a todas aquellas personas que me han ayudado, aguantado y perdonado durante todo este largo periodo de tiempo. En especial se lo quiero dedicar a Laura que sin su paciencia, comprensión y apoyo, este proyecto no hubiese sido posible.

Control de cambios

Versión	Fecha	Comentarios	Autor
1	05/2012	Creación estructura del documento	Luis Conde García
2	06/2012	Homogeneizar documento	Luis Conde García
3	06/2012	Revisión y corrección de erratas	Luis Conde García

INDICE

1. Introducción	6
1.1. Justificación	6
2. Establecimiento del plan de trabajo.....	7
3. Arquitectura del sistema	10
4. Requisitos del sistema	12
4.1. Requisitos funcionales	12
4.1.1. Especificación requisitos funcionales	13
4.2. Requisitos no funcionales.....	15
4.2.1. Especificación requisitos no funcionales.....	15
5. Definición de actores	17
6. Definición de casos de uso.....	18
6.1. Documentación textual de los casos de uso.....	18
7. Diagrama de clases.....	32
7.1. Subsistema de administración	32
7.1.1. Diagrama de clases	33
7.1.2. Diagrama de clases gestoras	33
7.1.3. Diagrama de clases de excepción.....	34
7.1.4. Diagrama de clases frontera	34
7.1.5. Diagrama de secuencia	35
7.1.6. Diagrama de estado.....	36
7.1.7. Notación CRC de clases lógica de negocio	36
7.2. Subsistema de solicitudes	43
7.2.1. Diagrama de clases	44
7.2.2. Diagrama de clases gestoras	44
7.2.3. Diagrama de clases de excepción.....	44
7.2.4. Diagrama de clases frontera	45
7.2.5. Diagrama de secuencia	45
7.2.6. Diagrama de estado.....	46
7.2.7. Notación CRC de clases lógica de negocio	47
7.3. Subsistema de conexión.....	47
7.3.1. Diagrama de clases	48

7.3.2. Diagrama de clases gestoras	48
7.3.3. Diagrama de clases de excepción.....	48
7.3.4. Diagrama de clases frontera	49
7.3.5. Diagrama de secuencia	49
7.3.6. Diagrama de estado.....	50
7.3.7. Notación CRC de clases lógica de negocio	50
7.4. Subsistema de estadísticas.....	51
7.4.1. Diagrama de clases	52
7.4.2. Diagrama de clases gestoras	52
7.4.3. Diagrama de clases de excepción.....	52
7.4.4. Diagrama de clases frontera	53
7.4.5. Diagrama de secuencia	53
7.4.6. Diagrama de colaboración.....	54
7.4.7. Notación CRC de clases lógica de negocio	54
8. Diseño de la persistencia	56
8.1. Diagrama E/R.....	56
9. Interfaz de usuario.....	58
9.1. Subsistema de administración	58
9.2. Subsistema de solicitudes	64
9.3. Subsistema de conexión	65
9.4. Subsistema de estadísticas.....	67
10. Conclusiones	69
10.1. Conclusiones científicas	69
10.2. Conclusiones personales	69
Apéndice.....	70
I. Glosario de términos	70
II. Bibliografía	71

1. Introducción

“La verdad se corrompe tanto con la mentira como con el silencio”.

- Marco Tulio Cicerón (106 AC-43 AC)

El desarrollo tecnológico está produciendo importantes cambios en la estructura económica y social afectando a todos los sectores de la economía y de la sociedad.

La importancia de las tecnologías en una empresa es hoy en día crucial, ya que no solo permiten aumentar la producción y abaratar costes sino que también posibilita que pequeñas empresas locales tengan acceso a un mercado mundial que antes tenían vetado por su tamaño gracias a Internet.

Siendo conscientes de la importancia del desarrollo tecnológico, Telecomunicaciones S.A. nos ha encargado el desarrollo de un sistema de gestión integral que permita gestionar el laboratorio de I+D de la compañía.

Por ello, el objetivo de este documento será llevar a cabo todo el análisis y diseño técnico necesario para la fase de desarrollo del proyecto.

En primer lugar se definirá la estructura de la aplicación explicando detalladamente cada una de las partes en las que estará dividida, presentando los distintos diagramas de clases que definirán el proyecto incluyendo los detalles de cada clase en notación CRC.

A continuación se presentaran los diagramas de estados y de secuencia de la aplicación, terminando con un profundo análisis de la persistencia del sistema junto con el diseño de la interfaz de usuario del cliente (GUI)¹.

1.1. Justificación

El proyecto surge de la necesidad de dotar al laboratorio de I+D de una herramienta potente, flexible y escalable que permita controlar y gestionar todos los recursos del laboratorio, mejorando de esta manera la gestión del mismo.

En definitiva lo que se pretende con el proyecto eGestionLab es mejorar la eficiencia del laboratorio de I+D al poder hacer un mejor uso de los recursos del mismo, mejorando con ello el rendimiento y eficiencia del mismo.

¹ Siglas en inglés de “*graphical user interface*” y es referido a la *interfaz grafica de usuario*.

2. Establecimiento del plan de trabajo

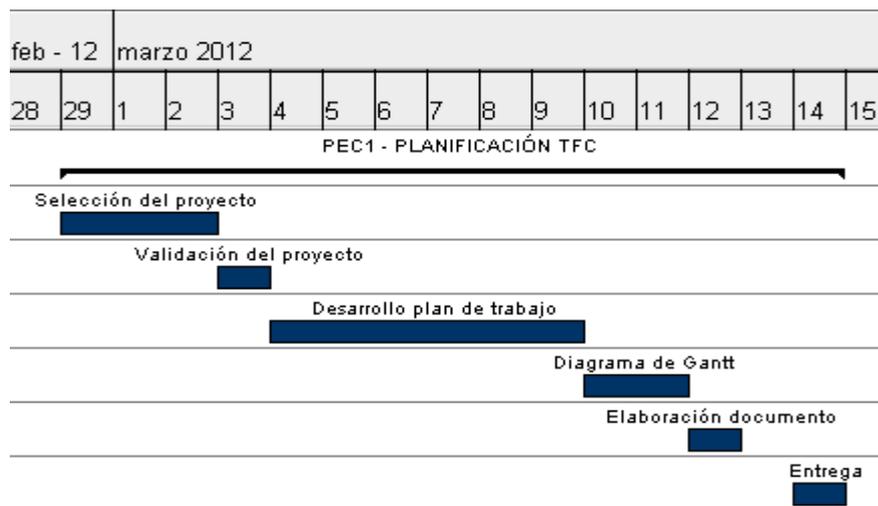
“El sabio puede sentarse en un hormiguero, pero sólo el necio se queda sentado en él”.

- Proverbio chino

A continuación se describirá el plan de trabajo previsto para el desarrollo de todo el proyecto donde se podrán ver las fechas claves del mismo.

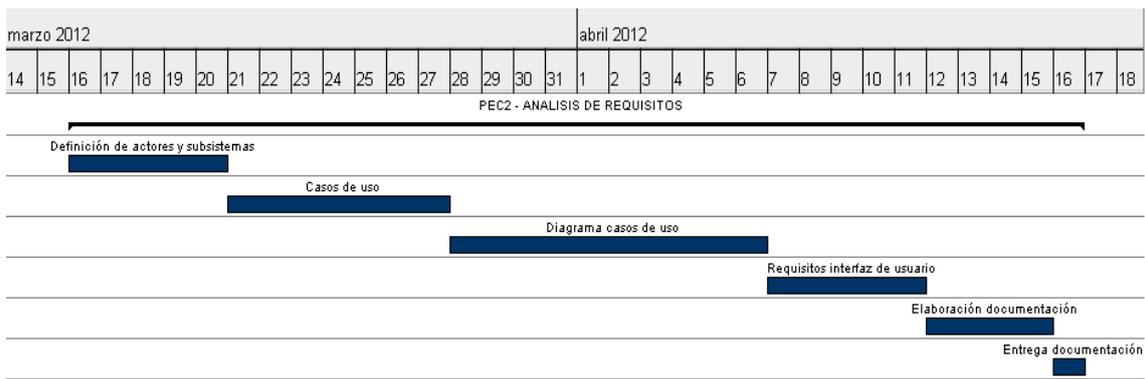
PEC 1

Tarea	Fecha inicio	Fecha fin
PEC1 – Planificación TFC	29/02/2012	14/03/2012
Selección del proyecto	29/02/2012	03/03/2012
Validación del proyecto	03/03/2012	03/03/2012
Desarrollo plan de trabajo	04/03/2012	10/03/2012
Diagrama de Gantt	10/03/2012	12/03/2012
Elaboración documentación	12/03/2012	13/03/2012
Entrega documentación	14/03/2012	14/03/2012



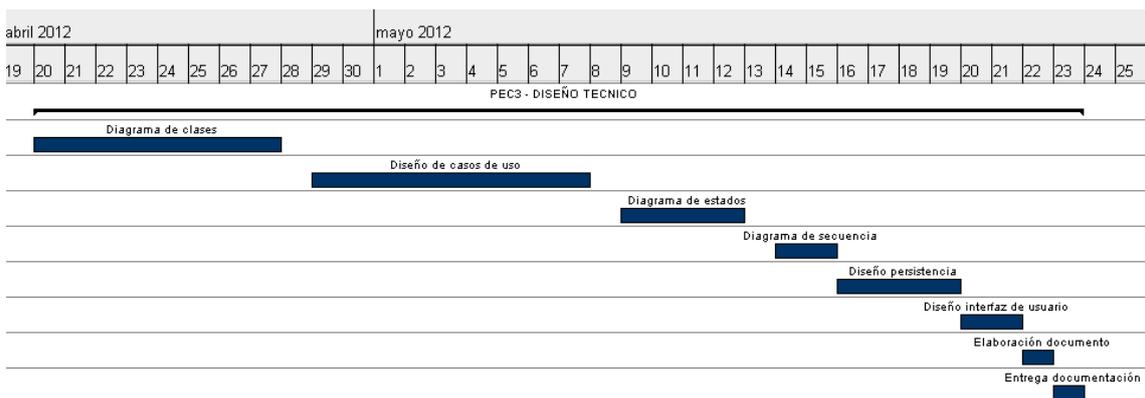
PEC 2

Tarea	Fecha inicio	Fecha fin
PEC2 – Análisis de requisitos	15/03/2012	18/04/2012
Definición de actores y subsistemas	16/03/2012	21/03/2012
Casos de uso	21/03/2012	28/03/2012
Diagrama casos de uso	29/03/2012	08/04/2012
Requisitos interfaz de usuario	08/04/2012	13/04/2012
Elaboración documentación	13/04/2012	17/04/2012
Entrega documentación	18/04/2012	18/04/2012



PEC 3

Tarea	Fecha inicio	Fecha fin
PEC3 – Diseño técnico	19/04/2012	23/05/2012
Diagrama de clases	20/04/2012	28/04/2012
Diseño de casos de uso	29/04/2012	08/05/2012
Diagrama de estados	09/05/2012	13/05/2012
Diagrama de secuencia	14/05/2012	16/05/2012
Diseño persistencia	16/05/2012	20/05/2012
Diseño interfaz de usuario	20/05/2012	22/05/2012
Elaboración documento	22/05/2012	23/05/2012
Entrega documentación	23/05/2012	23/05/2012



Memoria y presentación

Tarea	Fecha inicio	Fecha fin
MEMORIA Y PRESENTACIÓN	24/05/2012	06/06/2012
Elaboración de la memoria	25/05/2012	30/05/2012
Elaboración de la presentación	31/05/2012	06/06/2012
Entrega de la memoria y presentación	06/06/2012	06/06/2012



Tribunal virtual

Tarea	Fecha inicio	Fecha fin
TRIBUNAL VIRTUAL	18/06/2012	22/06/2012

3. Arquitectura del sistema

“Daría todo lo que sé, por la mitad de lo que ignoro”.

- René Descartes (1596-1650)

El proyecto eGestionLab permitirá desarrollar una aplicación que permitirá gestionar de manera integral un laboratorio de I+D de una empresa de telecomunicaciones e ingeniería informática.

El sistema permitirá entre otras cosas asignar a los investigadores los recursos necesarios para su trabajo y controlara el uso que se hace de los mismos.

El proyecto será desarrollado mediante la tecnología Microsoft .Net ² utilizando Microsoft Visual Studio 2010.

La aplicación contará con una arquitectura cliente-servidor donde la parte cliente se encargará de establecer y mantener la conexión contra la parte servidora y además se encargará de gestionar las peticiones de los usuarios contra el sistema.

Para la persistencia del sistema se utilizará el gestor de base de datos relacional orientado a objetos PostgreSQL ³.

La parte servidora gestionará toda la lógica de negocio y además se encargará entre otras cosas de gestionar y mantener la comunicación con el gestor de base de datos.

La aplicación estará dividida en cuatro subsistemas:

- **Subsistema de administración:** se encargará de todas las funciones relacionadas con la administración del sistema, como la gestión de los distintos recursos (alta, baja o modificación) y la de mantener los distintos parámetros de configuración del sistema.

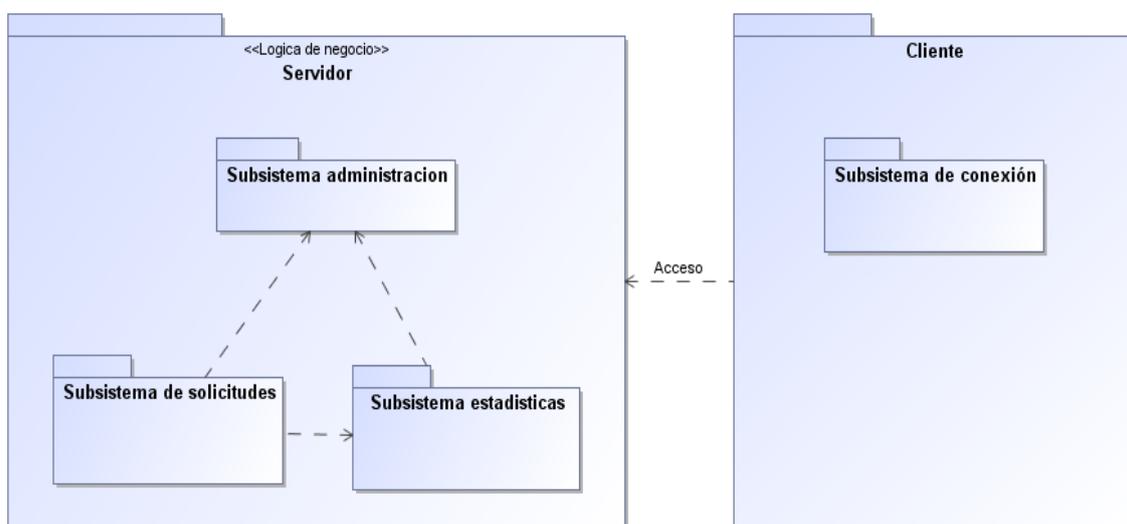
También se encargará de gestionar todas las funciones de seguridad, como activar y desactivar la auditoría de seguridad o gestionar los eventos de la misma. Además, se encargará de velar por la integridad de los datos almacenados en el gestor de base de datos así como del almacenamiento seguro de todas las credenciales de autenticación del sistema.

² Framework propiedad de Microsoft que permite el desarrollo de aplicaciones informáticas.

³ PostgreSQL, gestor de base de datos publicado bajo licencia BSD.

- **Subsistema de conexión:** se encargara de gestionar la conexión de red entre cliente y servidor así como de la autenticación de los usuarios contra el sistema, asignando a cada usuario autenticado un identificador único de usuario que será válido durante toda la sesión y que servirá entre otras cosas para asignar el rol del usuario y controlar que funciones puede desempeñar en el sistema.
- **Subsistema de estadísticas:** este se encargara de gestionar toda la información relativa al uso que hace cada uno de los usuarios registrados en el sistema durante un periodo de tiempo determinado. Dicha información servirá a posteriori para generar todo tipo de estadísticas donde figuraran entre otras cosas el número de reservas hecha, la duración de las mismas o el material defectuoso que el usuario ha devuelto. Dichas estadísticas podrán ser impresos o exportadas a cualquier otro formato como pdf, html o docx para su posterior tratamiento.
- **Subsistema de solicitudes:** este se encargara de gestionar todas las solicitudes y asignaciones de los usuarios, verificando antes, la disponibilidad de los mismos. Una vez asignados los recursos solicitados, el subsistema se encargara de las renovaciones y cancelaciones de las reservas, advirtiendole al usuario cada vez que el préstamo de un recurso esté a punto de expirar.

La aplicación eGestionLab estará distribuida según el siguiente diagrama de paquetes:



4. Requisitos del sistema

“Quiero saberlo todo. Y siempre me encuentro como antes, triste como la vida y resignado como la sabiduría”.

- Giovanni Papini (1881-1956)

Un requisito funcional define el comportamiento interno del software, como por ejemplo: cálculos, detalles técnicos, manipulación de datos y nos muestran como se aplicaran los casos de uso. En definitiva, los requisitos funcionales establecen el comportamiento del sistema.

Por otro lado, los requisitos no funcionales definen el diseño o la implementación.

En este apartado detallaremos cuales son los requisitos funcionales y no funcionales de la aplicación.

4.1. Requisitos funcionales

Requisitos de mantenimiento

- RQ.MAN.0001 – Almacenamiento de la información de líneas.
- RQ.MAN.0002 – Almacenamiento de la información de dispositivos.
- RQ.MAN.0003 – Almacenamiento de los usuarios del sistema.
- RQ.MAN.0004 – El administrador podra de dar alta/baja o modificar usuarios.

Requisitos de conexión y acceso

- RQ.CONX.0001 – Almacenamiento seguro de la contraseña
- RQ.CONX.0002 – Restablecimiento de la contraseña en caso de pérdida u olvido
- RQ.CONX.0003 – Acceso al sistema con certificados digitales
- RQ.CONX.0004 – Acceso mediante credenciales
- RQ.CONX.0005 – Comunicación con protocolos estándar de Internet para comunicación WAN
- RQ.CONX.0006 – Compatibilidad con el estándar IEEE 802.3

Requisitos de asignación

- RQ.ASIG.0001 – Almacenamiento de las reservas de puestos de trabajo.
- RQ.ASIG.0002 – Almacenamiento de las reservas de líneas de datos.
- RQ.ASIG.0003 – Almacenamiento de las reservas de material.
- RQ.ASIG.0004 – Solicitudes de usuarios.
- RQ.ASIG.0005 – Gestión de solicitudes.

Requisitos de consultas

- RQ.CON.0001 – Consultas del estado de las reservas de líneas de datos.
- RQ.CON.0002 – Consulta de estadísticas.
- RQ.CON.0003 – Consultas de las reservas de material.
- RQ.CON.0004 – Consultas de las reservas de las reservas de puestos de trabajo.
- RQ.CON.0005 – Consultas del administrador de los usuarios logados.
- RQ.CON.0006 – Consultar el estado de las cuentas de usuario.

Requisitos varios

- RQ.VAR.0001 – Exportación de datos.
- RQ.VAR.0002 – Elección de idioma.
- RQ.VAR.0003 – Configuración aplicación.
- RQ.VAR.0004 – Usuario administrador predeterminado

4.1.1. Especificación requisitos funcionales

Requisitos de mantenimiento

Código	Descripción
RQ.MAN.0001 – Líneas	El sistema debe almacenar los datos de todas las líneas de datos en una base de datos.
RQ.MAN.0002 - Recursos	El sistema debe almacenar los datos de todos los recursos en una base de datos.
RQ.MAN.0003 - Usuarios	El sistema debe almacenar los datos de todos los usuarios en una base de datos.

Requisitos conexión y acceso

Código	Descripción
RQ.CONX.0001 – Contraseña	El sistema debe almacenar los datos de conexión en una base de datos.
RQ. CONX.0002 - Restablecer	El sistema debe permitir recuperar las credenciales de acceso en caso de pérdida u olvido.
RQ. CONX.0003 - Certificados	El sistema admitirá el acceso con certificados digitales.
RQ. CONX.0004 – Acceso	El sistema debe permitir el acceso de los usuarios mediante usuario y clave de acceso.
RQ. CONX.0005 – Comunicación	La comunicación con los clientes se basará en protocolos estándar de Internet.
RQ. CONX.0006 – Conexión LAN	La comunicación LAN será compatible con el estándar IEEE 802.3

Requisitos asignación

Código	Descripción
RQ.ASIG.0001 – Reserva puestos	El sistema debe almacenar todas las reservas de puesto de trabajo.
RQ. ASIG.0002 – Reserva líneas	El sistema debe almacenar todas las reservas de líneas de datos solicitadas por los usuarios.
RQ. ASIG.0003 – Reserva material	El sistema debe almacenar todas las asignaciones de material solicitado por los usuarios.
RQ. ASIG.0004 – Solicitudes	El sistema debe permitir a los usuarios hacer cualquier tipo de solicitud.
RQ. ASIG.0005 – Gestión solicitud	El administrador podrá gestionar (cancelar, aceptar, renovar) cualquier tipo de solicitud.

Requisitos de consulta

Código	Descripción
RQ.CON.0001 – Reservas líneas	El sistema debe permitir a los usuarios consultar el estado de sus reservas de líneas de datos.
RQ. CON.0002 – Estadísticas	El sistema debe permitir a los usuarios consultar las estadísticas globales del laboratorio.
RQ. CON.0003 – Reserva material	El sistema debe permitir a los usuarios consultar el estado de sus reservas de material.
RQ. CON.0004 – Puestos	El sistema debe permitir a los usuarios consultar el estado de sus reservas de puestos de trabajo.
RQ. CON.0005 – Accesos	El sistema debe permitir al administrador consultar los usuarios que están logados en el sistema.
RQ. CON.0006 – Reservas	Consultar el estado de las cuentas de usuario.

Requisitos varios

Código	Descripción
RQ.VAR.0001 – Exportar	El sistema debe permitir a los usuarios exportar todos los datos de asignación y reservas.
RQ. VAR.0002 – Idioma	El sistema debe permitir a los usuarios elegir el idioma de la aplicación entre español o inglés.
RQ. VAR.0003 – Configuración	El sistema debe permitir a los usuarios configurar los distintos parámetros de configuración de usuario.
RQ. VAR.0004 – Administrador	El sistema tendrá una cuenta de administrador por defecto y que no se pueda borrar.

4.2. Requisitos no funcionales

Requisitos compatibilidad

- RQ.COMP.0001 – Compatibilidad con sistemas Microsoft.
- RQ.COMP.0002 – Compatibilidad con .net Framework.
- RQ.COMP.0003 – Compatibilidad con i386.

Requisitos seguridad

- RQ.SEG.0001 – Bloqueo automatico de cuenta.
- RQ.SEG.0002 – Almacenamiento seguro de credenciales.
- RQ.SEG.0003 – Registro de auditoria de seguridad.
- RQ.SEG.0004 – Perfiles de usuario.

Requisitos rendimiento

- RQ.REN.0001 – Mostrar busquedas en cierto periodo de tiempo.
- RQ.REN.0002 – Pruebas de carga.
- RQ.REN.0003 – Pruebas de estres.
- RQ.REN.0004 – Pruebas de estabilidad.

4.2.1. Especificación requisitos no funcionales

Requisitos compatibilidad

Código	Descripción
RQ.COMP.0001 – Plataforma	El sistema debe ser compatible con Windows XP, Vista y 7.
RQ. COMP.0002 – .Net Framework	El sistema debe ser compatible con .Net Framework
RQ. COMP.0003 – Compatibilidad i386	El sistema debe ser compatible con los microprocesadores i386.

Requisitos seguridad

Código	Descripción
RQ.SEG.0001 – Bloqueo cuenta	El sistema bloqueara cualquier cuenta de usuario durante 5 minutos tras 4 intentos fallidos de autentificación.
RQ. SEG.0002 – Credenciales	El sistema almacenara de forma segura las credenciales de acceso de los usuarios.
RQ. SEG.0003 – Auditoria	El sistema tendrá un sistema de auditoría para hacer

	un seguimiento de los inicio de sesión en el sistema así como del borrado o actualización de datos.
RQ. SEG.0004 – Perfiles de usuario	Cada usuario tendrá un perfil que le autorizara a realizar distintas funciones dentro del sistema

Requisitos rendimiento

Código	Descripción
RQ.REN.0001 – Búsquedas	El sistema no deberá tardar más de 7 segundos en mostrar los resultados de una búsqueda. En caso de que los supere deberá cancelar la búsqueda y mostrar los resultados obtenidos
RQ. REN.0002 – Prueba de carga	Se realizara un conjunto de pruebas de carga para observar el comportamiento de una aplicación bajo una cantidad de peticiones esperada.
RQ. REN.0003 – Prueba de estrés	Se realizara un conjunto de pruebas de estrés para determinar la solidez de la aplicación en los momentos de carga extrema para determinar si la aplicación rendirá lo suficiente en caso de que la carga real supere a la carga esperada.
RQ. REN.0004 – Prueba estabilidad	Se realizara un conjunto de pruebas de estabilidad para determinar si la aplicación puede aguantar una carga esperada continuada.

5. Definición de actores

“El ignorante afirma, el sabio duda y reflexiona”.

- Aristóteles (384 AC-322 AC)

En este apartado identificaremos a los actores del sistema y explicaremos cuáles son las funciones de cada uno.

El primer actor con el que nos encontramos es el “usuario”. Este podrá hacer todo tipo de solicitudes (material, líneas de datos, espacios de trabajo, etc.) tras haberse validado. Además, podrá hacer un seguimiento de todas las solicitudes que ha hecho y tiene pendientes así como consultar todas las estadísticas de uso durante un periodo determinado de tiempo. Además, este también podrá cancelar o renovar cualquier solicitud en cualquier momento.

El siguiente actor con el que nos encontramos es el “administrador” que será una especificación de usuario. La principal función de este será la gestionar y administrar el sistema y la de hacer un seguimiento de todas las solicitudes.

Una de las primeras funciones del administrador será la de dar de alta en el sistema todos los recursos del laboratorio (usuarios, líneas de datos, material, puestos de trabajo, etc.) Después, una vez dado de alta a los usuarios, este les asignará un rol que definirá las funciones que este usuario puede hacer en el sistema.

Por otro lado, el administrador también se encargará de la seguridad de todo el sistema. Este se encargará de activar/desactivar la auditoría de seguridad, auditando en cualquier momento el evento que más llame su atención. También se encargará de hacer un seguimiento de todos los logs de inicio de sesión y de controlar el estado de las cuentas de usuario. Además de esto, también velará por la seguridad e integridad de los datos del gestor de base de datos, manteniéndole actualizado en todo momento y realizando copias de seguridad para asegurar la información del servidor en caso de pérdida o catástrofe con los datos.

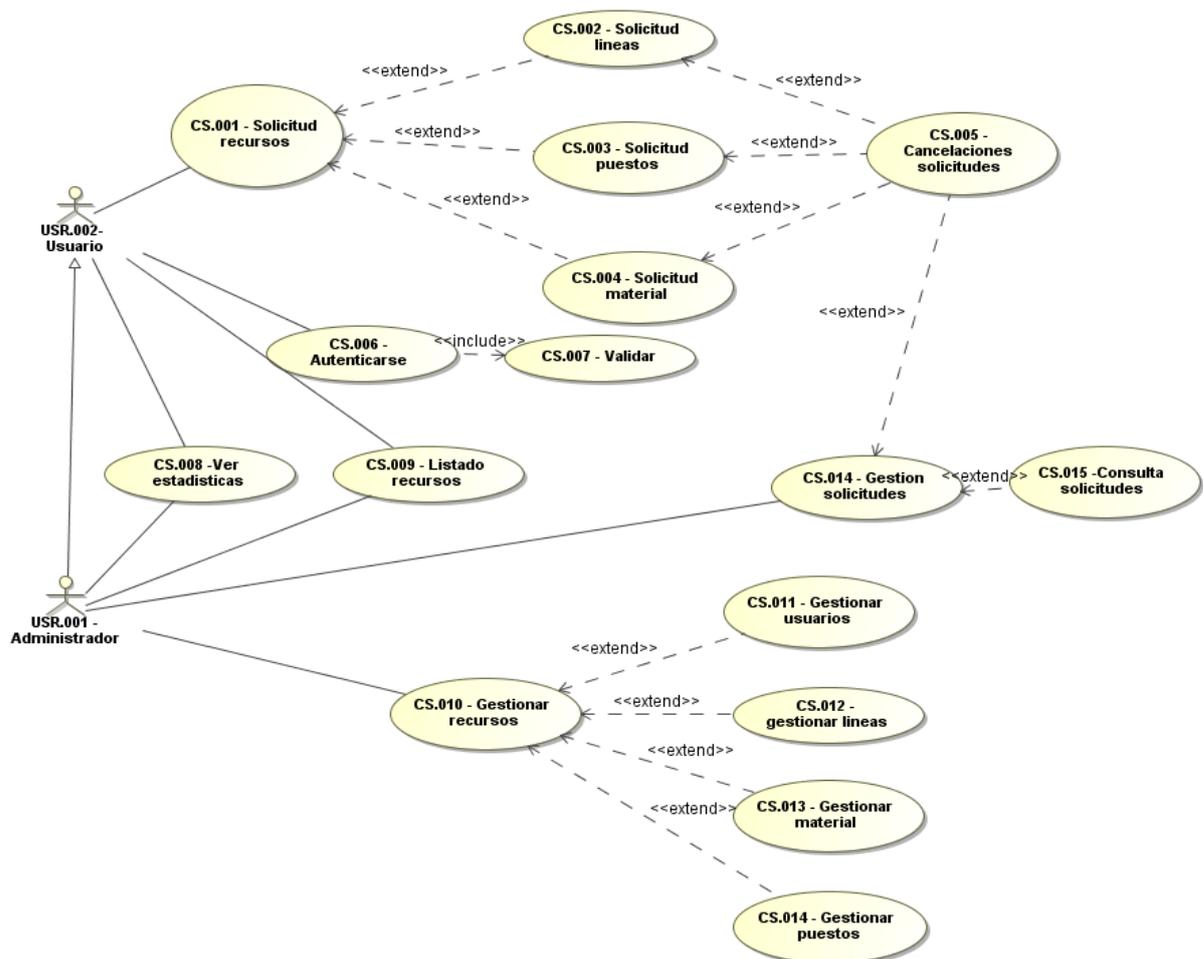
Otra de las funciones del administrador será la de hacer un seguimiento del uso que se hace de los recursos del laboratorio ya que este podrá consultar en tiempo real las estadísticas globales del sistema así como consultar las estadísticas de un usuario en particular o grupo de ellos, permitiendo de esta manera optimizar los recursos del laboratorio. Toda esta información podrá exportarla a cualquier formato para su presentación (pdf, html, xls, docx, etc) o bien almacenarla en la base de datos para su posterior análisis.

6. Definición de casos de uso

“El sabio puede cambiar de opinión.
El necio, nunca”.

- Immanuel Kant (1724-1804)

En este punto detallamos el diagrama de casos de uso. Este nos mostrara la relación del sistema con su entorno.



6.1. Documentación textual de los casos de uso

CASO DE USO: Alta de usuario – CS.011 -	
Funcionalidad	Añade un nuevo usuario al sistema
Actores	Administrador
Pre condición	El usuario no está dado de alta en la base de datos
Post condición	El usuario queda registrado en la base de datos
Flujo básico	<ul style="list-style-type: none"> El administrador introduce los datos solicitados por el sistema. El sistema asignara automáticamente un código único alfanumérico.

	<ul style="list-style-type: none"> • Se guardara toda la información en la base de datos. • El sistema mostrara un mensaje confirmación del alta.
Excepciones	<ul style="list-style-type: none"> • Si el usuario ya existe en la base de datos, el sistema mostrara un mensaje de error indicando que el usuario ya existe. • Si al dar de alta al usuario el administrador no rellena los campos obligatorios. El sistema mostrara un mensaje de error pidiendo que se rellenen todos los campos.

CASO DE USO: Baja de usuario – CS.011 -	
Funcionalidad	Eliminar una cuenta usuario en el sistema.
Actores	Administrador.
Pre condición	El usuario está dado de alta en la base de datos.
Post condición	El usuario queda eliminado de la base de datos.
Flujo básico	<ul style="list-style-type: none"> • El administrador hace una búsqueda para localizar la cuenta de usuario que desea borrar. • Una vez localizada la cuenta de usuario, se procederá a eliminarla pulsando el botón “Eliminar” tras confirmar el mensaje de confirmación. • El sistema mostrara un mensaje confirmación de la baja de la línea de datos.
Excepciones	<ul style="list-style-type: none"> • Si el usuario no existe en la base de datos, el sistema mostrara un mensaje de error indicando que el usuario no existe.
Aclaraciones	<ul style="list-style-type: none"> • Cuando se elimina a un usuario de la base de datos, todos los recursos que tenga asignados quedaran marcados como “Libres”. • Cuando se elimine una cuenta de usuario, esta no será borrada físicamente de los servidores, simplemente se marcara como inactiva.

CASO DE USO: Modificación de usuario – CS.011 -	
Funcionalidad	Actualizar los datos de un usuario en el sistema.
Actores	Administrador.
Pre condición	El usuario está dado de alta en la base de datos.
Post condición	Los nuevos datos de usuario han sido registrados en la base de datos.
Flujo básico	

	<ul style="list-style-type: none"> • El administrador hará una búsqueda para localizar al usuario en cuestión. • El sistema mostrara todos los datos del usuario. • El administrador modificara los datos necesarios. • Los datos modificados se guardaran en la base de datos. • Se mostrara un mensaje confirmando la actualización de los datos.
Excepciones	<ul style="list-style-type: none"> • Si el administrador introduce en algún campo un carácter no permitido, como por ejemplo un carácter alfabético en un campo fecha, el sistema mostrara un mensaje de error indicando el motivo del mismo. • Si el usuario intenta dejar en blanco un campo obligatorio le saldrá un mensaje de error indicando que rellene todos los campos obligatorios.

CASO DE USO: Consulta de usuarios – CS.011 -

Funcionalidad	Buscar los datos de un usuario en el sistema.
Actores	Administrador.
Pre condición	El usuario está dado de alta en la base de datos.
Post condición	Los datos del usuario no serán modificados.
Flujo básico	<ul style="list-style-type: none"> • El administrador seleccionara los filtros de búsqueda que mejor se adapten a sus necesidades. • El administrador hará una búsqueda para localizar al usuario en cuestión. • El sistema mostrara todos los datos del usuario.
Excepciones	<ul style="list-style-type: none"> • Si el sistema no puede mostrar algún resultado, este mostrara un mensaje de información para indicar que no se han encontrado ningún usuario con esos criterios de búsqueda.

CASO DE USO: Alta de línea de datos – CS.012 -

Funcionalidad	Añade un nueva línea de datos al sistema.
Actores	Administrador.
Pre condición	La línea de datos no está dada de alta en la base de datos.
Post condición	La línea de datos queda registrada en la base de datos.
Flujo básico	<ul style="list-style-type: none"> • El administrador introduce los datos solicitados por el sistema. • Se guardara toda la información en la base de datos. • El sistema mostrara un mensaje confirmación del alta.
Excepciones	<ul style="list-style-type: none"> • Si la línea ya está dada de alta en la base de datos, el

	<p>sistema mostrara un mensaje de error indicando que la línea ya existe.</p> <ul style="list-style-type: none"> • Si al dar de alta la línea de datos el administrador no rellena los campos obligatorios. El sistema mostrara un mensaje de error pidiendo que se rellenen todos los campos.
--	---

CASO DE USO: Baja de línea de datos – CS.012 -	
Funcionalidad	Elimina una línea de datos en el sistema.
Actores	Administrador.
Pre condición	La línea de datos está dada de alta en la base de datos.
Post condición	La línea de datos queda “eliminada” de la base de datos.
Flujo básico	<ul style="list-style-type: none"> • El administrador hace una búsqueda para localizar la línea de datos que desea borrar. • Una vez localizada la cuenta de usuario, se procederá a eliminarla pulsando el botón “Eliminar” tras confirmar el mensaje de confirmación. • El sistema mostrara un mensaje confirmación de la baja de la línea de datos.
Excepciones	<ul style="list-style-type: none"> • En caso de que la línea este reservada por un usuario saldrá un mensaje de error indicando que la línea está siendo usada y se cancelara la baja de la línea.
Aclaraciones	<ul style="list-style-type: none"> • Cuando se elimine una línea de datos, esta no será borrada físicamente de los servidores, simplemente se marcara como inactiva.

CASO DE USO: Modificación de línea de datos – CS.012 -	
Funcionalidad	Actualizar los datos de una línea de datos en el sistema.
Actores	Administrador.
Pre condición	La línea está dada de alta en la base de datos.
Post condición	Los nuevos datos de la línea han sido registrados en la base de datos.
Flujo básico	<ul style="list-style-type: none"> • El administrador hará una búsqueda para localizar la línea que desea modificar. • El administrador modificara los datos necesarios. • Los datos modificados se guardaran en la base de datos. • Se mostrara un mensaje confirmando la actualización de los datos.
Excepciones	

	<ul style="list-style-type: none"> Si el usuario intenta dejar en blanco un campo obligatorio le saldrá un mensaje de error indicando que rellene todos los campos obligatorios.
Aclaraciones	<ul style="list-style-type: none"> El campo "Numero de teléfono" no podrá ser modificado ya que este número deber ser único en la base de datos.

CASO DE USO: Consulta de líneas de datos – CS.012 -

Funcionalidad	Buscar los datos de una línea en el sistema.
Actores	Administrador.
Pre condición	La línea de datos está dada de alta en la base de datos.
Post condición	Los datos de la línea no serán modificados.
Flujo básico	<ul style="list-style-type: none"> El administrador seleccionara los filtros de búsqueda que mejor se adapten a sus necesidades. El administrador hará una búsqueda para localizar la línea de datos en cuestión. El sistema mostrara todos los datos de la línea.
Excepciones	<ul style="list-style-type: none"> Si el sistema no puede mostrar algún resultado, este mostrara un mensaje de información para indicar que no se han encontrado ninguna línea con esos criterios de búsqueda.

CASO DE USO: Alta de recursos de almacén – CS.013 -

Funcionalidad	Registra la entrada de un nuevo recurso en el almacén.
Actores	Administrador.
Pre condición	El recurso no está dado de alta en la base de datos.
Post condición	El recurso queda registrado en la base de datos.
Flujo básico	<ul style="list-style-type: none"> El administrador introduce los datos solicitados por el sistema. Se guardara toda la información en la base de datos. El sistema mostrara un mensaje de confirmación del alta.
Excepciones	<ul style="list-style-type: none"> Si el recurso ya está dada de alta en la base de datos, el sistema mostrara un mensaje de error indicando que el recurso ya está registrado. Si al dar de alta el recurso el administrador no rellena los campos obligatorios. El sistema mostrara un mensaje de error pidiendo que se rellenen los campos obligatorios.

CASO DE USO: Baja de recursos de almacén – CS.013 -	
Funcionalidad	Eliminar un recurso/dispositivo del almacén.
Actores	Administrador.
Pre condición	El recurso está dado de alta en la base de datos.
Post condición	El recurso no está dado de alta en la base de datos.
Flujo básico	<ul style="list-style-type: none"> • El administrador hace una búsqueda para localizar el recurso que desea borrar. • Una vez localizado el recurso, se procederá a eliminarlo pulsando el botón “Eliminar” tras confirmar el mensaje de confirmación. • El sistema mostrara un mensaje confirmación de la baja de la línea de datos.
Excepciones	<ul style="list-style-type: none"> • En caso de que el recurso este reservado por un usuario saldrá un mensaje de error indicando que el dispositivo está siendo usado y se cancelara la baja del dispositivo.
Aclaraciones	<ul style="list-style-type: none"> • Cuando se de de baja un dispositivo del almacén, este no será borrado físicamente de los servidores, simplemente se marcara como inactivo.

CASO DE USO: Modificación de recursos de almacén – CS.013 -	
Funcionalidad	Actualizar los datos de un recurso de almacén.
Actores	Administrador.
Pre condición	El recurso está dado de alta en la base de datos.
Post condición	Los nuevos datos del recurso han sido registrados en la base de datos.
Flujo básico	<ul style="list-style-type: none"> • El administrador hará una búsqueda para localizar el recurso que desea modificar. • El administrador modificara los datos necesarios. • Los datos modificados se guardaran en la base de datos. • Se mostrara un mensaje confirmando la actualización de los datos.
Excepciones	<ul style="list-style-type: none"> • Si el usuario intenta dejar en blanco un campo obligatorio le saldrá un mensaje de error indicando que rellene todos los campos obligatorios.
Aclaraciones	<ul style="list-style-type: none"> • El campo “Numero de serie” no podrá ser modificado ya que este número deber ser único en la base de datos.

CASO DE USO: Consulta de recursos de almacén – CS.013 -	
Funcionalidad	Buscar los datos de los recursos de almacén.
Actores	Administrador.
Pre condición	El recurso está dado de alta en la base de datos.
Post condición	El recurso no será modificado.
Flujo básico	<ul style="list-style-type: none"> • El administrador seleccionara los filtros de búsqueda que mejor se adapten a sus necesidades. • El administrador hará una búsqueda para localizar el recurso en cuestión. • El sistema mostrara todos los datos del recurso almacenado en el almacén.
Excepciones	<ul style="list-style-type: none"> • Si el sistema no puede mostrar algún resultado, este mostrara un mensaje de información para indicar que no se han encontrado ningún recurso con esos criterios de búsqueda.

CASO DE USO: Alta de una reserva de espacio de trabajo – CS.014 -	
Funcionalidad	Da de alta una reserva de espacio de trabajo.
Actores	Administrador.
Pre condición	El espacio de trabajo no está reservado.
Post condición	El espacio de trabajo queda reservado por un usuario o grupo.
Flujo básico	<ul style="list-style-type: none"> • El administrador introduce los datos solicitados por el sistema. • Se guardara toda la información en la base de datos. • El sistema mostrara un mensaje de confirmación de la reserva.
Excepciones	<ul style="list-style-type: none"> • Si el puesto o espacio de trabajo ya está reservado para esas mismas fechas, el sistema mostrara un mensaje de error indicando que el espacio de trabajo ya esta reservado. • Si a la hora de hacer la reserva el administrador no rellena los campos obligatorios. El sistema mostrara un mensaje de error pidiendo que se rellenen todos los campos.

CASO DE USO: Cancelar una reserva de espacio de trabajo – CS.014-	
Funcionalidad	Cancela una reserva de espacio de trabajo.
Actores	Administrador.
Pre condición	El espacio de trabajo está reservado.
Post condición	El espacio de trabajo queda libre.
Flujo básico	<ul style="list-style-type: none"> • El administrador hace una búsqueda para localizar la reserva que desea cancelar.

	<ul style="list-style-type: none"> • Una vez localizada esta, se procederá a cancelarla pulsando el botón “Cancelar” tras confirmar el mensaje de confirmación. • El sistema mostrara un mensaje de confirmación de la cancelación.
Excepciones	<ul style="list-style-type: none"> • En caso de que el espacio este reservado por un usuario, saldrá un mensaje de error indicando que está siendo usado y se cancelara la baja.
Aclaraciones	<ul style="list-style-type: none"> • Todas las cancelaciones serán completamente eliminadas de la base de datos.

CASO DE USO: Modificación de una reserva de espacio de trabajo – CS.014 -

Funcionalidad	Modificar fechas de una reserva de espacio de trabajo.
Actores	Administrador.
Pre condición	La reserva está dada de alta en la base de datos.
Post condición	Los nuevos datos de la reserva han sido registrados en la base de datos.
Flujo básico	<ul style="list-style-type: none"> • El administrador hará una búsqueda para localizar la reserva que desea modificar. • El administrador modificara los datos necesarios. • Los datos modificados se guardaran en la base de datos. • Se mostrara un mensaje confirmando la actualización de los datos.
Excepciones	<ul style="list-style-type: none"> • Si el usuario intenta dejar en blanco un campo obligatorio le saldrá un mensaje de error indicando que rellene todos los campos obligatorios. • Si las nuevas fechas de reserva entran en conflicto con otra reserva, aparecerá un mensaje de error señalando que las nuevas fechas no están disponibles.

CASO DE USO: Consulta de una reserva de espacio de trabajo – CS.014 -

Funcionalidad	Buscar los datos una reserva de espacio de trabajo.
Actores	Administrador.
Pre condición	La reserva está dada de alta en la base de datos.
Post condición	La reserva no será modificada.
Flujo básico	<ul style="list-style-type: none"> • El administrador seleccionara los filtros de búsqueda que mejor se adapten a sus necesidades. • El administrador hará una búsqueda para localizar la reserva

	<p>en cuestión.</p> <ul style="list-style-type: none"> • El sistema mostrara todas las reservas almacenadas en el sistema.
Excepciones	<ul style="list-style-type: none"> • Si el sistema no puede mostrar algún resultado, este mostrara un mensaje de información para indicar que no se han encontrado ninguna reserva de puesto de trabajo con esos criterios de búsqueda.

CASO DE USO: Recuperación de la contraseña en caso de olvido – CS.011 -

Funcionalidad	Recuperación de la contraseña en caso de olvido
Actores	Usuario.
Pre condición	El usuario esta dado de alta en el sistema.
Post condición	La contraseña queda restablecida.
Flujo básico	<ul style="list-style-type: none"> • El usuario entra en el menú “Recuperación de contraseña” desde la ventana de inicio de sesión. • El usuario introduce su D.N.I • El sistema enviara un correo electrónico con la contraseña de acceso del usuario a la dirección de correo electrónico que tenga asignado el D.N.I introducido por el usuario. • El sistema mostrara un mensaje de confirmación de la operación.
Excepciones	<ul style="list-style-type: none"> • Si el D.N.I del usuario no aparece registrado en el sistema, aparecerá un mensaje de error indicando que el usuario no está dado de alta en el sistema. • Si el sistema no fuese capaz de enviar el e-mail por el motivo que sea, el sistema mostrara un mensaje error indicando el error.

CASO DE USO: Acceso al sistema con usuario y contraseña – CS.006 -

Funcionalidad	Acceder al sistema mediante validación con usuario y contraseña.
Actores	Usuarios
Pre condición	El usuario está dado de alta en la base de datos.
Post condición	El usuario queda validado en el sistema
Flujo básico	<ul style="list-style-type: none"> • El usuario arranca la aplicación y se va al menú “Inicio →Conectar” • Una vez aparecida la pantalla de solicitud de acceso, el usuario procederá a introducir el nombre de usuario y contraseña que le facilito el administrador cuando le dio de alta en el sistema.

	<ul style="list-style-type: none"> El sistema validara el acceso al usuario en caso de que sea posible.
Excepciones	<ul style="list-style-type: none"> En caso de que el usuario no esté dado de alta en el sistema, aparecerá un mensaje de error indicando que el usuario no está dado de alta en el sistema. En caso de que el usuario no rellene alguno de los campos de acceso (login o contraseña), el sistema mostrara un mensaje de error pidiéndole que introduzca todos los datos.

CASO DE USO: Acceso al sistema con certificado digital o D.N.I – CS.006 -

Funcionalidad	Acceder al sistema mediante certificado digital.
Actores	Usuarios.
Pre condición	El usuario está dado de alta en la base de datos.
Post condición	El usuario queda validado en el sistema
Flujo básico	<ul style="list-style-type: none"> El usuario arranca la aplicación y se va al menú “Inicio →Conectar” Una vez aparecida la pantalla de solicitud de acceso, el usuario marcara la casilla “Con certificado” y procederá a pasar su D.N.I. por el lector digital. El sistema validara el acceso al usuario en caso de que sea posible.
Excepciones	<ul style="list-style-type: none"> En caso de que el usuario no esté dado de alta en el sistema, aparecerá un mensaje de error indicando que el usuario no está dado de alta en el sistema.

CASO DE USO: consulta del estado de las cuentas de usuario – CS.009 -

Funcionalidad	Comprobar el estado de las cuentas de usuario.
Actores	Administrador.
Pre condición	La cuenta de usuario esta dado de alta en la base de datos.
Post condición	La cuenta de usuario quedara como esta o restablecida en caso de bloqueo
Flujo básico	<ul style="list-style-type: none"> El administrador seleccionara los filtros de búsqueda que mejor se adapten a sus necesidades. El administrador hará una búsqueda para localizar la cuenta de usuario que desea verificar. El sistema mostrara la cuenta de usuario almacenadas en el sistema. El administrador comprobara el campo “bloqueo” para asegurarse de que la cuenta esta desbloqueada.
Excepciones	

	<ul style="list-style-type: none"> Si el sistema no puede mostrar algún resultado, este mostrara un mensaje de información para indicar que no se han encontrado ninguna cuenta de usuario con esos criterios de búsqueda.
--	---

CASO DE USO: consulta de usuarios logados – CS.011 -	
Funcionalidad	Comprobar el número de usuarios logados en el sistema.
Actores	Administrador.
Pre condición	La cuenta de usuario está dada de alta en la base de datos y se encuentra logada en ese momento.
Post condición	Las cuentas de usuario no se modificaran.
Flujo básico	<ul style="list-style-type: none"> El administrador hará una búsqueda para localizar las cuentas de usuario que se encuentran logadas en ese momento, seleccionando la casilla “logados” en el menú de consultas de usuario. El sistema mostrara las cuentas de usuarios logadas en el sistema en ese mismo instante.
Excepciones	<ul style="list-style-type: none"> Si el sistema no puede mostrar algún resultado, este mostrara un mensaje de información para indicar que no se ha encontrado ninguna cuenta de usuario.

CASO DE USO: Asignaciones de usuario – CS.004 -	
Funcionalidad	Hacer una solicitud de equipamiento
Actores	Usuario.
Pre condición	El recurso está dado de alta en la base de datos.
Post condición	El recurso solicitado queda asignado al usuario.
Flujo básico	<ul style="list-style-type: none"> El usuario hará una búsqueda para localizar el recurso que desea solicitar aplicando los filtros de búsqueda necesarios. El usuario pulsara el botón “Asignar”. Se mostrara un mensaje confirmando la asignación del recurso solicitado.
Excepciones	<ul style="list-style-type: none"> Si el usuario intenta solicitar un recurso que ya está asignado a otro usuario, el sistema mostrara un mensaje de error indicando que el recurso ya está asignado.

CASO DE USO: Asignaciones de administrador – CS.004-	
Funcionalidad	Que el administrador pueda asignar un recurso de manera manual.
Actores	Administrador.
Pre condición	El recurso está dado de alta en la base de datos.
Post condición	El recurso solicitado queda asignado a un usuario.
Flujo básico	<ul style="list-style-type: none"> El administrador hará una búsqueda para localizar el recurso que desea asignar aplicando los filtros de búsqueda necesarios. El administrador pulsara el botón “Asignar a”. El administrador seleccionara de un combo al usuario que quiere asignar el recurso. Se mostrara un mensaje confirmando la asignación del recurso solicitado.
Excepciones	<ul style="list-style-type: none"> Si el administrador intenta asignar un recurso que ya está asignado a otro usuario, el sistema mostrara un mensaje de error indicando que el recurso ya está asignado.

CASO DE USO: Cancelar asignaciones – CS.005 -	
Funcionalidad	Cancelar cualquier asignación en curso.
Actores	Usuarios.
Pre condición	La solicitud esta aceptada.
Post condición	La solicitud queda anulada y el recurso queda liberado.
Flujo básico	<ul style="list-style-type: none"> El usuario hará una búsqueda para localizar la asignación que desea cancelar aplicando los filtros de búsqueda necesarios. El usuario pulsara el botón “Liberar”. El sistema mostrara un mensaje confirmando la cancelación del la asignación.
Excepciones	<ul style="list-style-type: none"> Si el usuario intenta cancelar una asignación de otro usuario, el sistema mostrara un mensaje de error denegándole la petición.

CASO DE USO: renovar asignaciones – CS.014 -	
Funcionalidad	Renovar cualquier asignación en curso.
Actores	Usuarios.
Pre condición	La asignación esta aceptada.
Post condición	La asignación queda renovada por un cierto periodo de tiempo.
Flujo básico	<ul style="list-style-type: none"> El usuario hará una búsqueda para localizar la asignación que

	<p>desea renovar aplicando los filtros de búsqueda necesarios.</p> <ul style="list-style-type: none"> • El usuario pulsara el botón “Renovar”. • El sistema le pedirá que introduzca la nueva fecha de finalización. • El sistema mostrara un mensaje confirmando la renovación de la asignación.
Excepciones	<ul style="list-style-type: none"> • Si el usuario intenta renovar la asignación dando una fecha que ya está reservada para otro usuario, el sistema mostrara un mensaje de error denegándole la renovación.

CASO DE USO: consulta de estadísticas de usuario – CS.008 -	
Funcionalidad	Consulta de estadísticas de usuario.
Actores	Usuarios.
Pre condición	El sistema tiene asignaciones.
Post condición	N/A
Flujo básico	<ul style="list-style-type: none"> • El usuario entrara en el menú “Estadísticas de usuario” y seleccionaran los filtros que mejor se adapten a sus necesidades. • El usuario introducirá el rango de fechas para generar las estadísticas. • El sistema generara las estadísticas y las mostrara por pantalla.
Excepciones	<ul style="list-style-type: none"> • Si el sistema no tiene registrado ningún dato en un rango determinado de fechas el sistema mostrara un mensaje de información indicando que en ese periodo no hay datos disponibles.

CASO DE USO: consulta de estadísticas globales – CS.008 -	
Funcionalidad	Consulta de estadísticas globales.
Actores	Administrador.
Pre condición	El sistema tiene asignaciones.
Post condición	N/A
Flujo básico	<ul style="list-style-type: none"> • El administrador entrara en el menú “Estadísticas globales” y seleccionaran los filtros que mejor se adapten a sus necesidades. • El administrador introducirá el rango de fechas para generar las estadísticas globales. • El sistema generara las estadísticas y las mostrara por pantalla.

Excepciones	<ul style="list-style-type: none"> Si el sistema no tiene registrado ningún dato en ese rango de fechas seleccionado el sistema mostrara un mensaje de información indicando que en ese periodo no hay datos disponibles.
--------------------	--

CASO DE USO: exportación de datos – CS.008 -

Funcionalidad	Exportar las estadísticas del sistema.
Actores	Administrador.
Pre condición	El administrador genera las estadísticas globales.
Post condición	Las estadísticas están en otro formato.
Flujo básico	<ul style="list-style-type: none"> El administrador entrara en el menú “Estadísticas globales” y seleccionaran los filtros que mejor se adapten a sus necesidades. El administrador introducirá el rango de fechas para generar las estadísticas globales. El sistema generara las estadísticas y las mostrara por pantalla. El administrador pulsara el botón “Exportar” y seleccionara el formato de exportación de datos. El sistema generara las estadísticas en el nuevo formato seleccionado.
Excepciones	<ul style="list-style-type: none"> Si el sistema no tiene registrado ningún dato en ese rango de fechas seleccionado el sistema mostrara un mensaje de información indicando que en ese periodo no hay datos disponibles. Si el sistema no puede guardar en el disco duro las estadísticas con el nuevo formato mostrara un mensaje error indicando que no tiene acceso al disco duro.

CASO DE USO: elección de idioma – CS.006 -

Funcionalidad	Elección del idioma en la interfaz de usuario.
Actores	Usuarios.
Pre condición	El usuario se encontrara en el menú “Conectar”
Post condición	La interfaz de usuario estará en el idioma elegido por el usuario.
Flujo básico	<ul style="list-style-type: none"> El usuario entrara en el menú “Inicio→Conectar” y seleccionara en el combo “Idioma” el idioma que desea. La interfaz de usuario cambiara al idioma seleccionado.
Excepciones	<ul style="list-style-type: none"> Si el sistema tuviese algún problema para acceder al fichero “properties”, el combo solo mostrara el idioma Español.

7. Diagrama de clases

- “Piensa como piensan los sabios,
mas habla como habla la gente sencilla.”*
- Aristóteles (384 AC-322 AC)

El Lenguaje Unificado de Modelado (UML)⁴ es un lenguaje de modelado de sistemas de software. Este es un lenguaje grafico que permite visualizar, modelar, construir y documentar un sistema.

UML 2.0 cuenta con 13 tipos diferentes de diagramas como por ejemplo el diagrama de clases, diagramas de secuencia, diagramas de estado, o de paquetes entre otros, aunque en nuestro proyecto nos centraremos en estos últimos. Estos nos permitirán observar los diferentes aspectos del sistema a desarrollar.

El diagrama de clases es un tipo de diagrama estático que se usa para describir la estructura de un sistema mostrando las clases, atributos, métodos y las relaciones que hay entre ellos. Este es el diagrama principal en las fases de análisis y diseño de una aplicación.

Los diagramas de secuencia son usados para modelar la interacción entre objetos a través del tiempo y se modela para cada caso de uso mientras que los diagramas de estado muestran el conjunto de estados por los que pasa un objeto durante su vida en respuesta a diferentes eventos.

En este apartado detallaremos los distintos diagramas de clases por cada subsistema de la aplicación así como los distintos diagramas de secuencia y de estados que definirán en buena parte al proyecto. En estos diagramas se han omitido todos los métodos y atributos de las clases para mejorar la legibilidad pero dichos métodos y atributos serán especificados en notación CRC.

7.1. Subsistema de administración

Este subsistema se encargara de mantener y gestionar todos los recursos de la aplicación, como por ejemplo los usuarios del sistema, las líneas de datos, los puestos de trabajo o incluso los distintos dispositivos del almacén, permitiendo darlos de alta o baja en el sistema así como la modificación de los mismos.

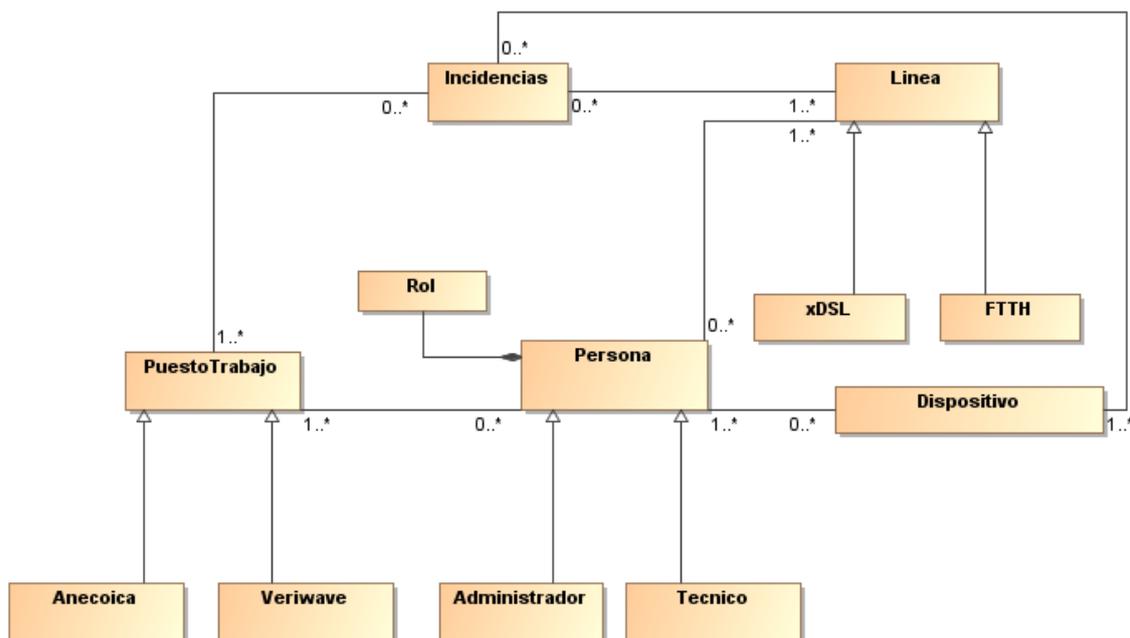
Además, también se encargara de supervisar toda la seguridad del sistema permitiendo habilitar/deshabilitar los distintos eventos de auditoría y hacer un seguimiento de todos los log de sucesos.

⁴ Por sus siglas en inglés Unified Modeling Language (UML)

Al subsistema de administración solo podrá ser accedido por un usuario con rol de Administrador.

7.1.1. Diagrama de clases

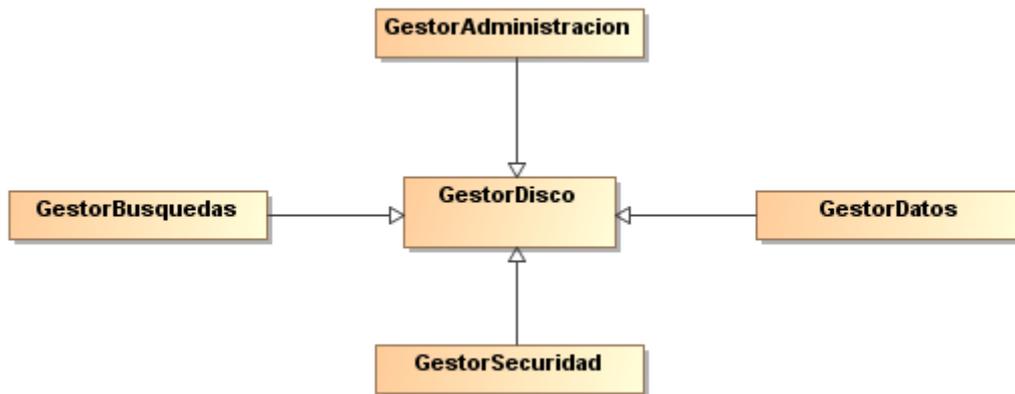
Las clases de negocio son las que llevan la lógica funcional del sistema. En este apartado detallaremos las clases relacionadas con el subsistema de administración.



7.1.2. Diagrama de clases gestoras

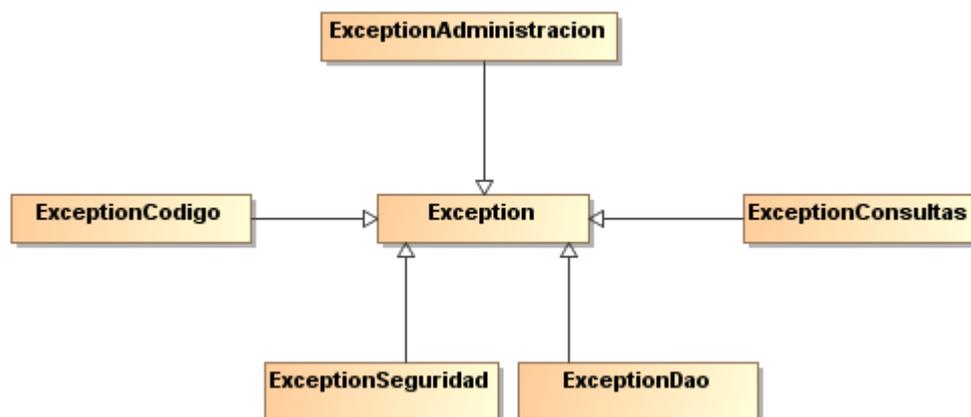
En este subapartado se detallara cuales son las clases gestoras del subsistema de administración y cuál es su jerarquía.

Estas clases se encargaran de atender cualquier tipo de petición por parte del administrador del sistema y en definitiva permitirán gestionar todo el subsistema de mantenimiento.



7.1.3. Diagrama de clases de excepción

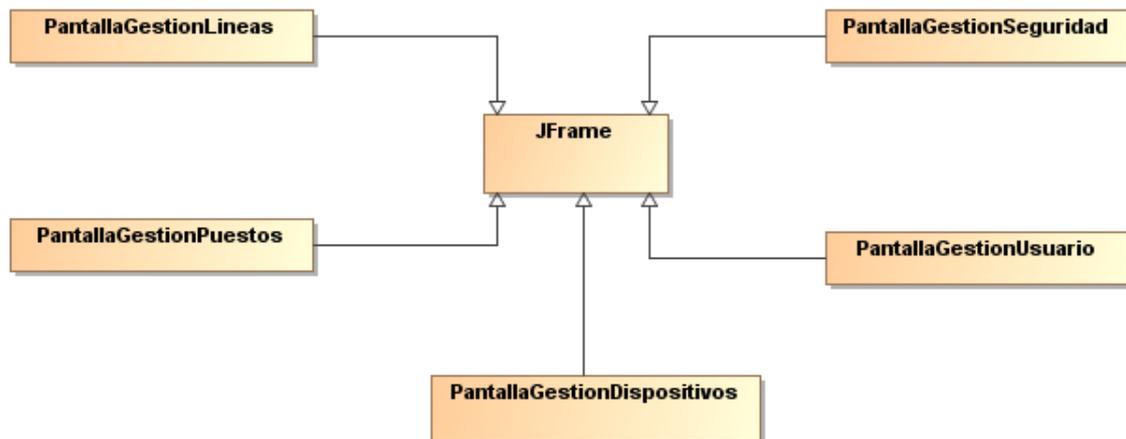
En este subapartado se detallarán todas aquellas clases del subsistema de administración que se encargaran de gestionar cualquier tipo de excepción que ocurra en el subsistema.



7.1.4. Diagrama de clases frontera

En este subapartado se detallarán todas aquellas clases frontera que intervienen en el subsistema de administración.

Las clases frontera se usan para modelar las interfaces del sistema y manejan la comunicación entre el contexto del sistema y el interior del sistema.

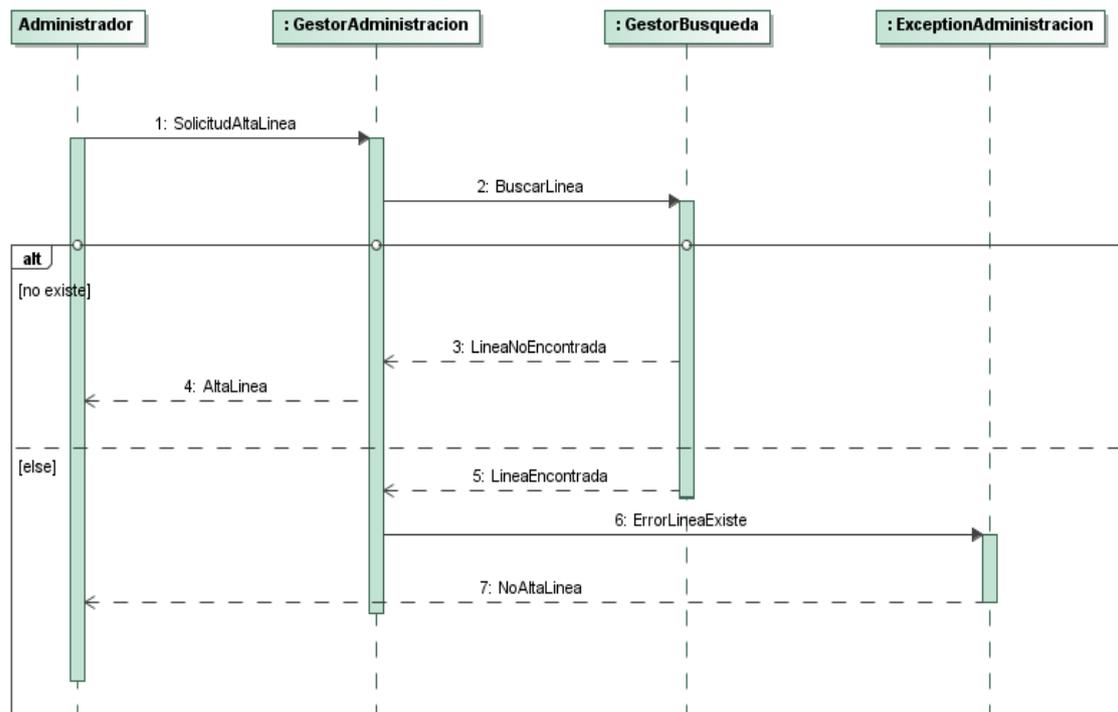


7.1.5. Diagrama de secuencia

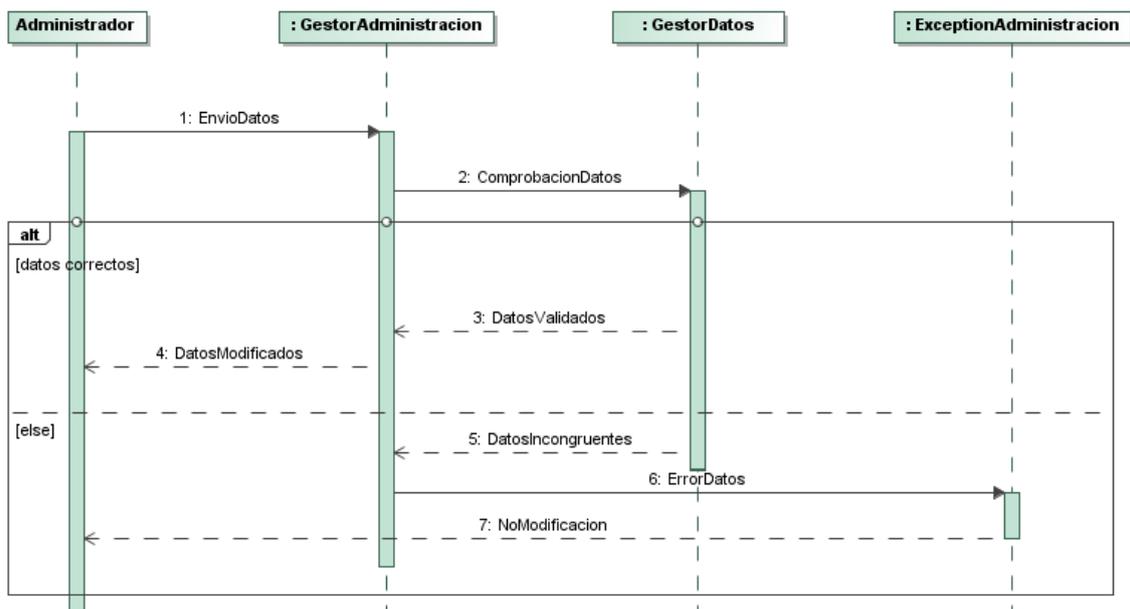
Los diagramas de secuencia muestra la forma en que los objetos se comunican entre sí en el transcurso del tiempo.

En este apartado detallaremos los diagramas de secuencia de los casos de uso alta y modificación de una línea de datos.

Alta de una línea:



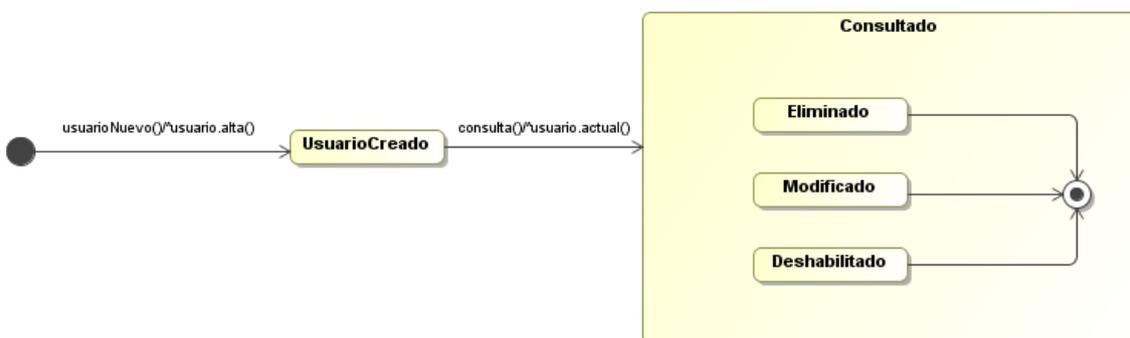
Modificación de una línea:



7.1.6. Diagrama de estado

Los diagramas de estado muestran los estados por los cuales pasa un objeto durante su vida en respuesta a diferentes eventos.

En este apartado detallaremos un diagrama de estado que muestra las situaciones por las que pasa una cuenta de usuario.



7.1.7. Notación CRC de clases lógica de negocio

En este subapartado detallaremos los métodos y atributos de todas las clases de lógica de negocio del subsistema de administración para completar la información de las mismas.

Clase: "Línea"	
Descripción	Representa a las líneas de datos

Tipo de clase	Clase principal
Responsabilidades	Gestionar la información de las líneas de datos
Características	Concreta y persistente
Constructores	+Linea();
Atributos	#idTipo:String #idReserva:int #perfil:String # datos:Datos #telefono:String #ipTv:String #ipDatos:String #ipPassword:String #estado:Boolean #observaciones:String
Métodos	+getIdTipos():String + setIdTipos(idLinea:String) +getPerfil():String + setPerfil(idLab:String) +getDatos():Datos +setDatos(datos:Datos) +getIdReserva():int +setIdReserva(reserva:int) +getTelefono():String + setTelefono(telefono:String) +getIpTv():String + setIpTv(ipTv:String) +getIpDatos():String + setIpDatos(ipDatos:String) +getEstado():boolean + setEstado(estado:boolean) +getObservaciones():String + setObservaciones(observaciones:String) +getPassword():String + setPassword(password:String)

Clase: "Xdsl"	
Descripción	Es una especificación de la clase "Linea" y representa a líneas xdsl (adsl, vdsl)
Tipo de clase	Clase principal
Responsabilidades	Gestionar la información de las líneas de datos adsl y vdsl
Características	Concreta y persistente
Constructores	+Xdsl();
Atributos	-idRecurso:int
Métodos	-altaLinea(telefono:String,ipTv:String,ipDatos:String, ,observaciones:String):boolean -bajaLinea(idRecurso):boolean -modificarLinea(idRecurso:int):boolean -buscarLinea(telefono:String):List<String>

	+getIdRecurso():int + setIdRecurso(idRecurso:int)
--	--

Clase: "FTTH"	
Descripción	Es una especificación de la clase "Linea" y representa a líneas de fibra óptica
Tipo de clase	Clase principal
Responsabilidades	Gestionar la información de las líneas de fibra óptica
Características	Concreta y persistente
Constructores	+Ftth();
Atributos	-idRecurso:int
Métodos	-altaLinea(idLinea:String,telefono:String,ipTv:String,ipDatos:String, -observaciones:String):boolean -bajaLinea(idLinea):boolean -modificarLinea(idLinea:String):boolean -buscarLinea(telefono:String):List<String> +get IdRecurso ():int + set IdRecurso (idLinea:int)

Clase: "Usuario"	
Descripción	Representa a los usuarios registrados en el sistema
Tipo de clase	Clase principal
Responsabilidades	Gestiona a los usuarios del sistema
Características	Concreta y persistente
Constructores	+usuario();
Atributos	#idUsuario:int #nombre:String #login:String #password:String #empresa:String #rol:String #observaciones:String #email:String #ciudad:String #pais:String # datos:Datos
Métodos	+getIdUsuario ():int + setIdUsuario (idUsuario:int) +getNombre ():String + setNombre (nombre:String) +getLogin():String + setLogin(login:String) +getPassword(): String + setPassword(password:String) +getEmpresa(): String + setEmpresa(empresa:String) +getRol(): String

	+ setRol(password:String) +getObservaciones():String + setObservaciones (ob:String) +getEmail:String + setEmail(email: String) +getCiudad():String + setCiudad(ciudad:String) +getPais():String + setPais(pais: String) +getDatos():Datos +setDatos(datos:Datos)
--	--

Clase: "Administrador"	
Descripción	Es una especificación de la clase "Usuario" y representa a los administradores del sistema
Tipo de clase	Clase principal
Responsabilidades	Gestionar la información de los administradores del sistema
Características	Concreta y persistente
Constructores	+administrador();
Atributos	-idAdmin:int -estado:boolean
Métodos	-altaAdmin(idUsuario:String):Boolean -bajaAdmin(idAdmin):Boolean -modificarAdmin(idAdmin:String):Boolean -buscarAdmin(idAdmin:String):List +getIdAdmin():int + setIdAdmin(idAdmin:int) +getEstado():Boolean + setEstado(estados:Boolean)

Clase: "Tecnico"	
Descripción	Es una especificación de la clase "Usuario" y representa a los técnicos registrados en el sistema
Tipo de clase	Clase principal
Responsabilidades	Gestionar la información de los técnicos del laboratorio
Características	Concreta y persistente
Constructores	+tecnico();
Atributos	-idTecnico:int -estado:boolean
Métodos	-altaTecnico(idUsuario:String):boolean -bajaTecnico (idTecnico):boolean -modificarTecnico (idTecnico:String):boolean -buscarTecnico (idTecnico:String):List<String> +getIdTecnico():int + setIdTecnico (idTecnico:int) +getEstado():boolean +setEstado(estados:boolean)

Clase: "Rol"	
Descripción	La clase se usa para almacenar los distintos rol de los usuarios
Tipo de clase	Clase principal
Responsabilidades	Gestionar los roles de los usuarios
Características	Concreta y persistente
Constructores	+rol(usuario:User);
Atributos	-idRol:int -idUserario:int
Métodos	-altaRol():boolean -bajaRol(idRol):boolean -modificarRol(idRol:int):boolean -buscarRol(idRol:int):List +getIdRol():int +setIdRol(rol:int) +getIdUsuario():int +setIdUsuario(idUsuario:int)

Clase: "PuestoTrabajo"	
Descripción	Representa distintos puestos de trabajo
Tipo de clase	Clase principal
Responsabilidades	Gestionar la información de los distintos puestos de trabajo que se pueden reservar
Características	Concreta y persistente
Constructores	+PuestosTrabajo();
Atributos	#idReserva:int #datos:Datos #nombre:String #fechaInicio:Date #fechaFin:Date #equipamiento:String #laboratorio:String #proyecto:String #observaciones:String
Métodos	+getReserva():Reserva + setReserva (reserva:Reserva) +getNombre():String + setNombre(nombre:String) +getFechaInicio():Date + setFechaInicio (FechaInicio:Date) +getFechaFin():Date + setFechaFin (FechaFin:Date) +getEquipamiento():String + setEquipamiento(equipamiento:String) +getLaboratorio():String + setLaboratorio(lab:String) +getProyecto():String

	+ setProyecto(proyecto:String) + getObservaciones():String + setObservaciones(ob:String)
--	--

Clase: "Anecoica"	
Descripción	Representa el puesto de trabajo en la cámara anecoica
Tipo de clase	Clase principal
Responsabilidades	Gestionar la información del puesto de trabajo en la cámara anecoica.
Características	Concreta
Constructores	+Anecoica();
Atributos	#idRecurso:int #estado:boolean
Métodos	-altaReserva():Boolean -bajaReserva (idReserva):Boolean -modificarReserva (idReserva:Reserva):Boolean -buscarReserva (idReserva:Reserva):List<String> +get IdRecurso ():int + setIdRecurso(puesto:int) +getEstado ():boolean + setEstado (estado:boolean)

Clase: "Veriwave"	
Descripción	Representa el puesto de trabajo en el dispositivo Veriwave
Tipo de clase	Clase principal
Responsabilidades	Gestionar las reservas del puesto de trabajo.
Características	Concreta y persistente
Constructores	+Veriwave();
Atributos	#idRecurso:int #estado:boolean
Métodos	-altaReserva():Boolean -bajaReserva (idReserva):Boolean -modificarReserva (idReserva:Reserva):Boolean -buscarReserva (idReserva:Reserva):List<String> +get IdRecurso ():int + setIdRecurso(recurso:int) +getEstado ():boolean + setEstado (estado:boolean) +getNombre ():String + setNombre(nombre:String) +getFechaInicio():Date + setFechaInicio (FechaInicio:Date) +getFechaFin():Date + setFechaFin (FechaFin:Date) +getEquipamiento():String + setEquipamiento(equipamiento:String) +getLaboratorio():String + setLaboratorio(lab:String)

	<pre>+getProyecto():String + setProyecto(proyecto:String) +getObservaciones():String + setObservaciones(ob:String)</pre>
--	--

Clase: "Dispositivo"	
Descripción	Representa los distintos dispositivos del laboratorio listos para prestar.
Tipo de clase	Clase principal
Responsabilidades	Gestionar la información de los distintos dispositivos del laboratorio
Características	Concreta y persistente
Constructores	+Dispositivo();
Atributos	<pre>-idReserva:int -idRecurso:Int -tipo:String -marca:String -modelo:String -muestras:int -numeroSerie:String -fechaEntrada:Date -estado:Boolean - datos:Datos</pre>
Métodos	<pre>-altaDispositivo():boolean -bajaDispositivo (idReserva):boolean -modificarDispositivo (idReserva:Reserva):Boolean -buscarDispositivo (idReserva:Reserva):List +getEstado ():boolean + setEstado (estado:boolean) +getIdReserva():int + setIdReserva (reserva:int) +getIdDispositivo():int + setIdDispositivo (dispositivo:int) +getMarca ():String +setMarca (nombre:String) +getModelo ():String + setModelo(modelo:String) +getMuestras():int + setMuestras (muestras:int) +getFechaEntrada():Date + setFechaEntrada(FechaEntrada:Date) +getEstado():Boolean +setEstado(estado:Boolean) +getDatos():Datos +setDatos(datos:Datos) +toString():String</pre>

Clase: "Incidencias"	
Descripción	La principal que representa a las incidencias abiertas en el sistema.
Tipo de clase	Clase principal.
Responsabilidades	Gestionar las averías de los recursos del sistema.
Características	Concreta y persistente
Constructores	+Incidencias();
Atributos	-idIncidencia:int -estado:Boolean -fechaInicio:Date -fechaFin:Date -descripcion:String -resolucion:String -propietario>User
Métodos	altaIncidencia(estados:boolean,fechaInicio:Date,fechaFin:Date,descripcion:String,resolucion:String,propietario>User):boolean -bajaIncidencia (idIncidencia):boolean -modificarIncidencia (idIncidencia:int):boolean -buscarIncidencia (idIncidencia:int):List<String> +get idIncidencia ():int + set idIncidencia (idLinea:int) +getFechaInicio():Date + setFechaInicio(fecha:Date) +getFechaFin():Date + setFechaFin(fecha:Date) +getEstado():Boolean +setEstado(estados:Boolean) +getResolucion():String +setResolucion(resolucion:String) +toString():String

7.2. Subsistema de solicitudes

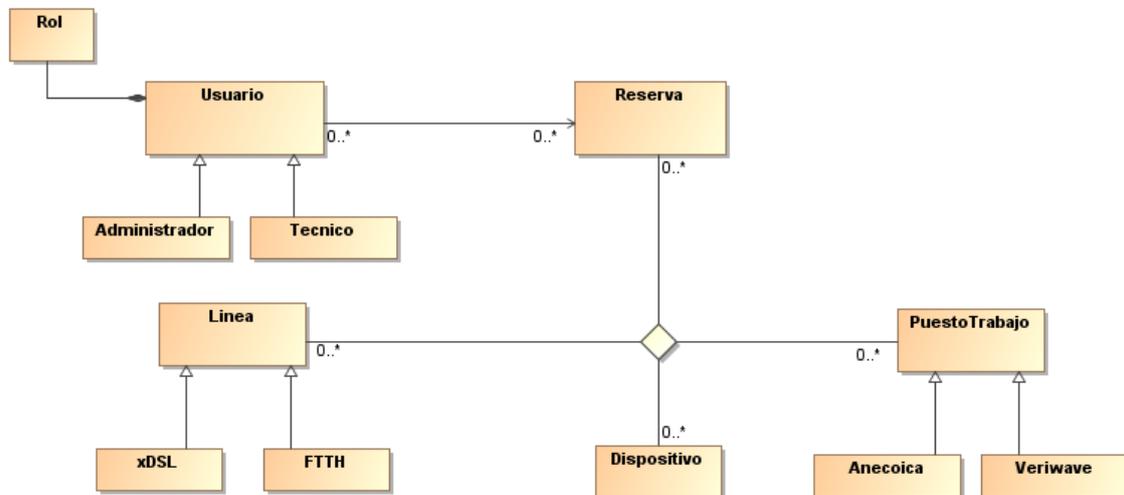
Este subsistema se encargara de gestionar y mantener todas las solicitudes y asignaciones de los usuarios, permitiendo darlos de alta, baja, modificarlos o incluso controlar los recursos asignados en todo momento.

También se encargara de controlar el estado de los dispositivos una vez hayan sido devueltos y de velar por las fechas reservadas ya que en caso de que una solicitud este cerca de su vencimiento, avisara al usuario para que proceda con su renovación en caso de que sea necesario.

Las solicitudes de recursos serán hechas por los usuarios del sistema aunque los administradores tendrán la capacidad de asignar manualmente cualquier tipo de recurso que le sea solicitado.

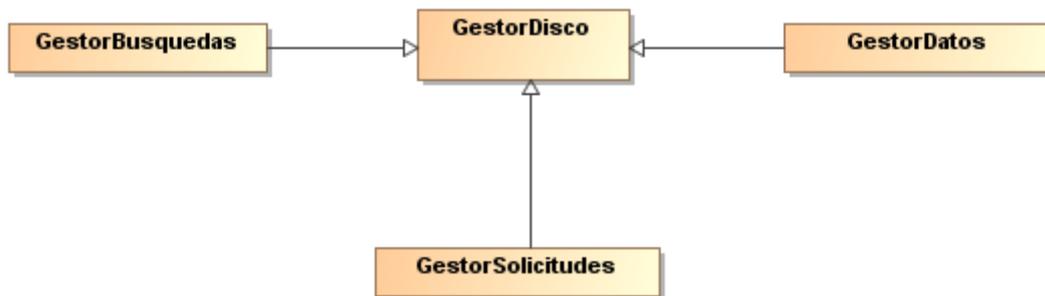
7.2.1. Diagrama de clases

En este subapartado se detallaran las clases relacionadas con el subsistema de solicitudes obviando en un primer momento los métodos y atributos para detallarlos más adelante en las fichas CRC.



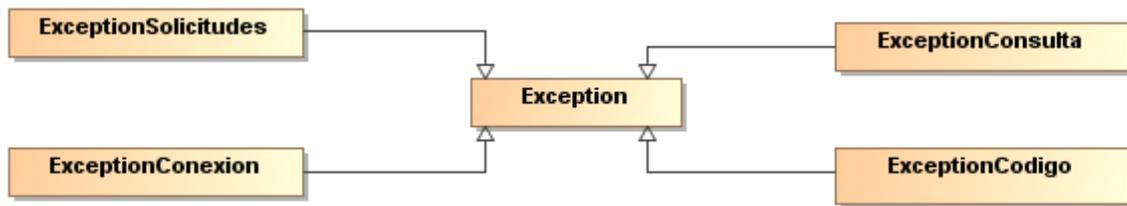
7.2.2. Diagrama de clases gestoras

En este subapartado detallaremos las clases gestoras relacionadas con el subsistema de solicitudes. Estas clases se encargaran de gestionar y tratar todas las solicitudes que hagan los usuarios al sistema.



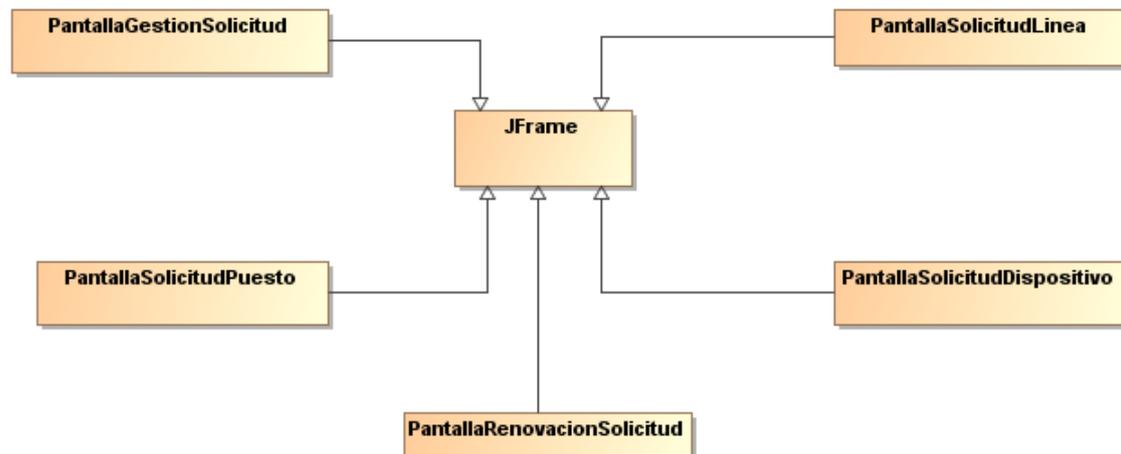
7.2.3. Diagrama de clases de excepción

En este subapartado se detallaran las distintas clases de excepción que están relacionadas con el subsistema de solicitudes, encargándose cada una de ellas de gestionar cualquier tipo de problema que surja durante la ejecución de la aplicación, evitando de esta manera que la aplicación se quede colgada ante cualquier problema.



7.2.4. Diagrama de clases frontera

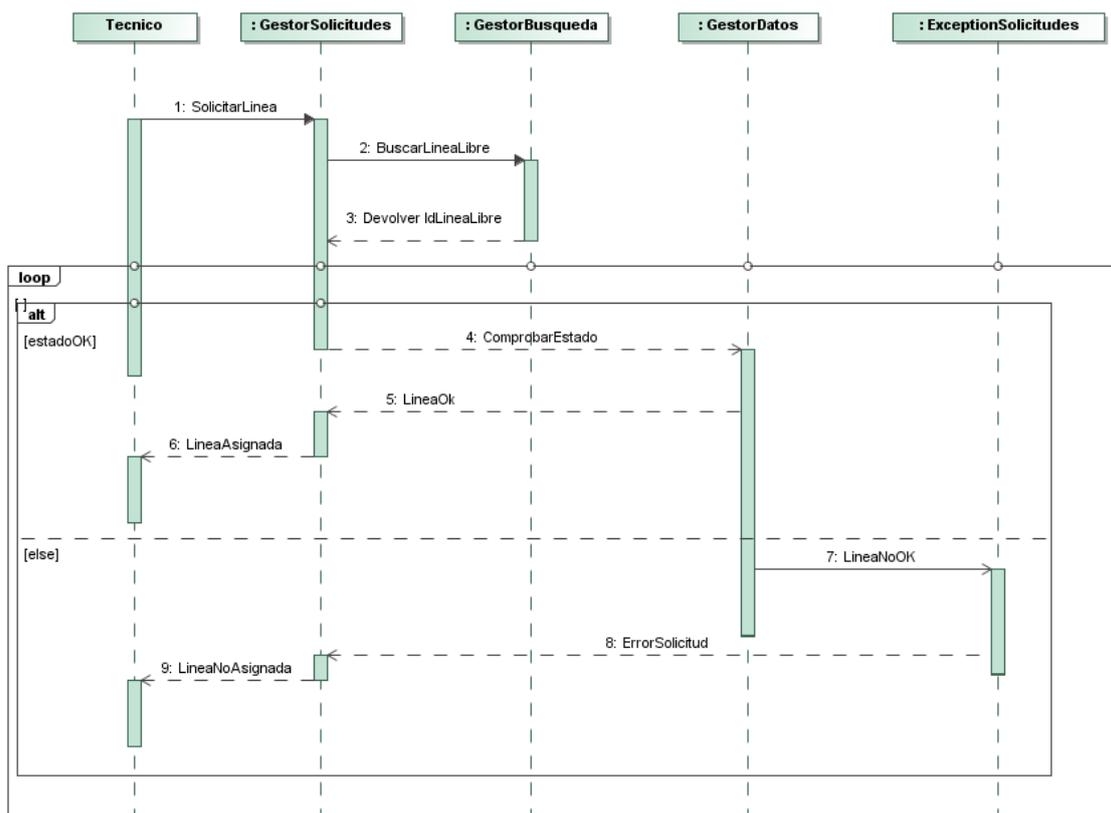
En este subapartado detallaremos las distintas clases frontera que conforman el subsistema de solicitudes, mostrando en un tema posterior el diseño de las mismas.



7.2.5. Diagrama de secuencia

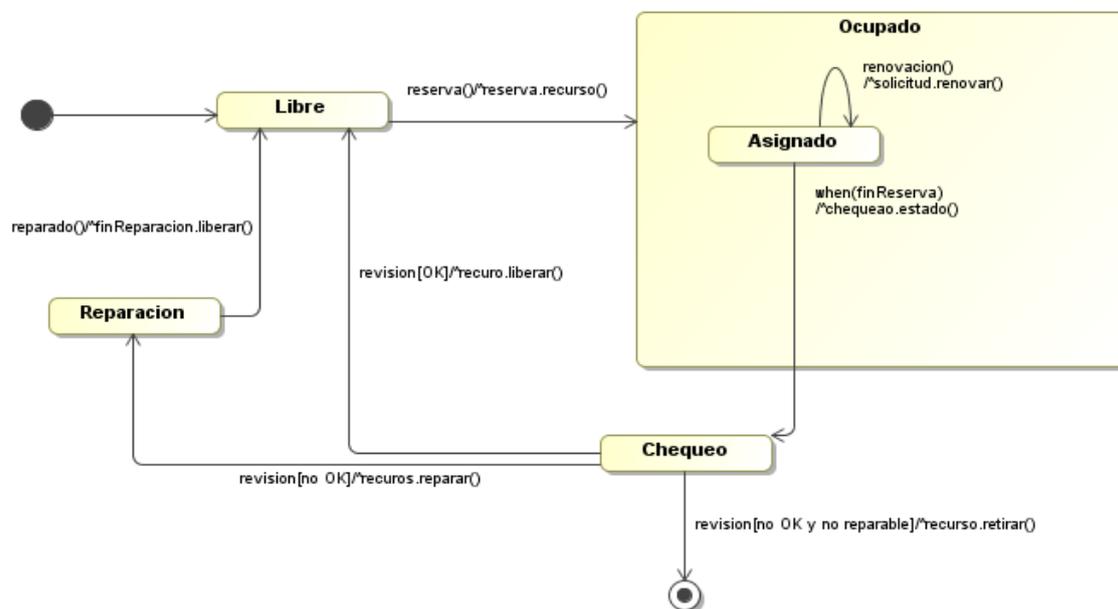
En este subapartado detallaremos el diagrama de secuencia del caso de uso “solicitud de una línea de datos”.

En el podremos observar como interaccionan los objetos entre sí a lo largo del tiempo cuando un usuario solicita una línea de datos.



7.2.6. Diagrama de estado

En este subapartado se detallara el diagrama de estado que mostrara los estados por los que pasa un objeto al realizar una solicitud y reserva de recursos.



7.2.7. Notación CRC de clases lógica de negocio

Clase: "Reserva"	
Descripción	Representa las distintas solicitudes y asignaciones de los usuarios.
Tipo de clase	Clase principal
Responsabilidades	Gestionar la información de las reservas de recursos.
Características	Concreta y persistente
Constructores	+solicitud(idReserva:int,idUsuario:User);
Atributos	-idReserva:Int -idRecurso:Int -fechaSolicitud:Date -idUsuario:User -estado:Boolean -log:List - datos:Datos
Métodos	-chequear(idUsuario:int,idRecurso:int):boolean -agregar (idSolicitud):boolean -renovar(fecha:Date,idRecurso:int):boolean -cancelar(idReserva):boolean -listar(idreserva:int):List +getIdReserva ():int + setIdReserva (reserva:int) +get idRecurso ():int + set idRecurso (idrecurso:int) +getFechaSolicitud ():Date + setFechaSolicitud (usuario:User) +get IdUsuario ():User + set IdUsuario (usuario:User) +getEstado():Boolean + setEstado (estado:Boolean) +getDatos():Datos +setDatos(datos:Datos)

7.3. Subsistema de conexión

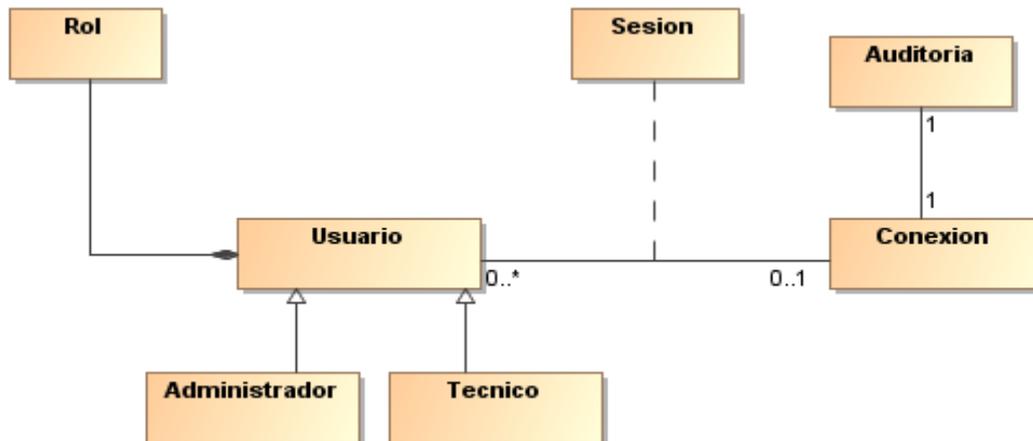
Una de las principales tareas de este subsistema será la de gestionar la comunicación por red entre el cliente y el servidor.

Además, también se encargara de la validación de los usuarios en el sistema mediante un nombre y una contraseña o bien mediante certificado digital. También se encargara de gestionar la desconexión de cada uno de ellos después de un cierto periodo de inactividad.

Por último, este subsistema también gestionara la asignación del rol a cada usuario, controlando en todo momento que tareas puedes desempeñar cada uno ellos en el sistema.

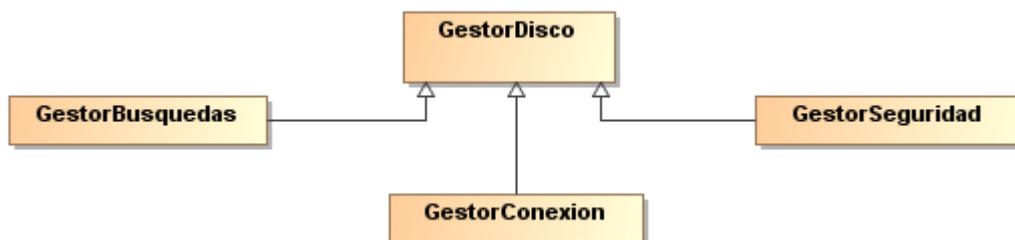
7.3.1. Diagrama de clases

A continuación detallaremos el diagrama de clases relacionado con este subsistema.



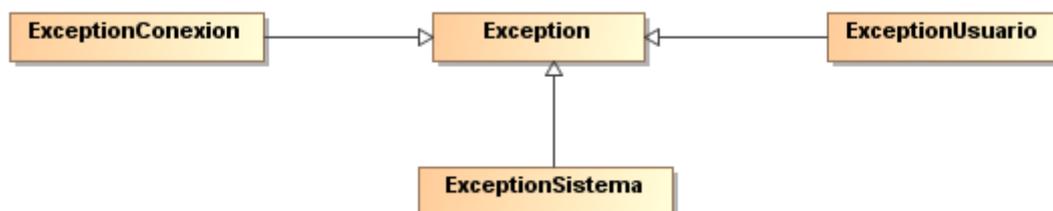
7.3.2. Diagrama de clases gestoras

En este apartado detallaremos el diagrama de clases gestoras que nos mostrara que clases se encargaran de gestionar el subsistema de conexión.



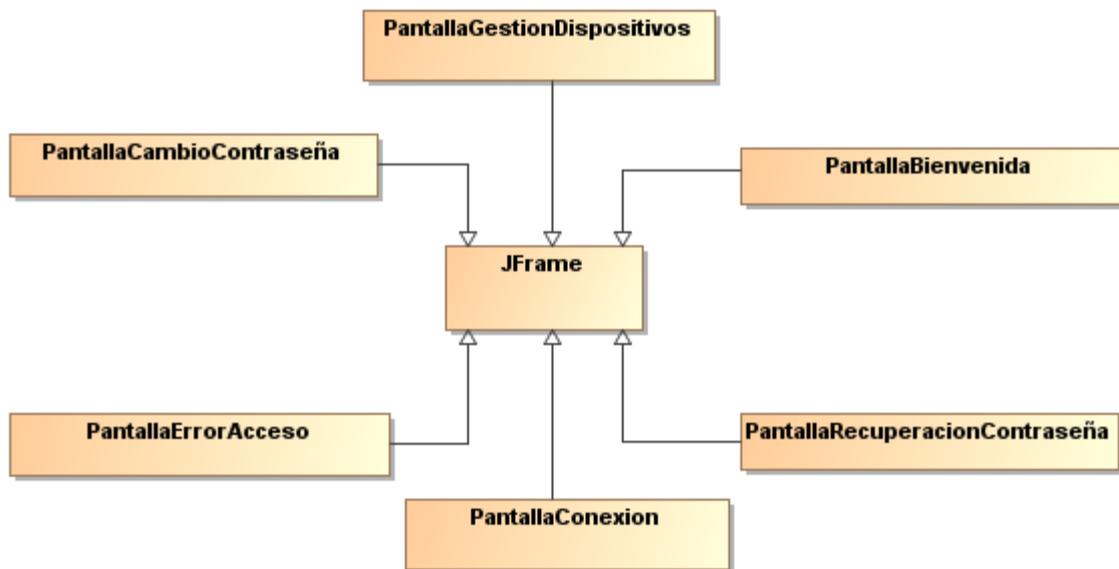
7.3.3. Diagrama de clases de excepción

En este apartado se detallaran las clases de excepción que participan en el subsistema de conexión. Estas clases serán las responsables de tratar cualquier tipo de excepción que ocurra en el subsistema.



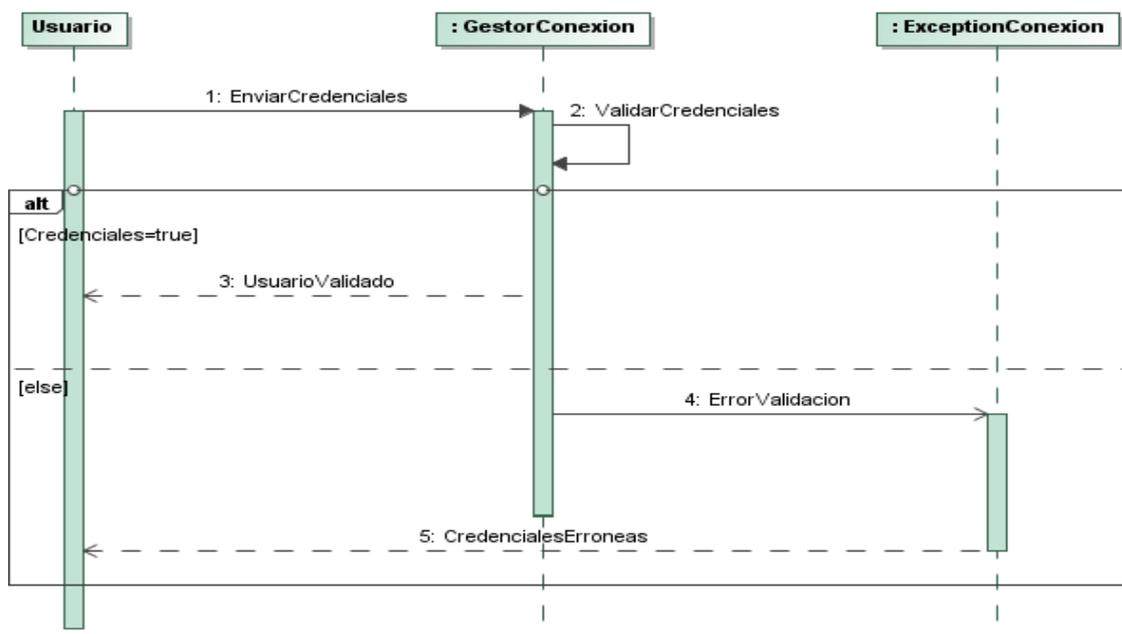
7.3.4. Diagrama de clases frontera

Como hemos explicado antes, las clases frontera representan las interfaces de usuario. En este subapartado detallaremos las clases frontera del actual subsistema.

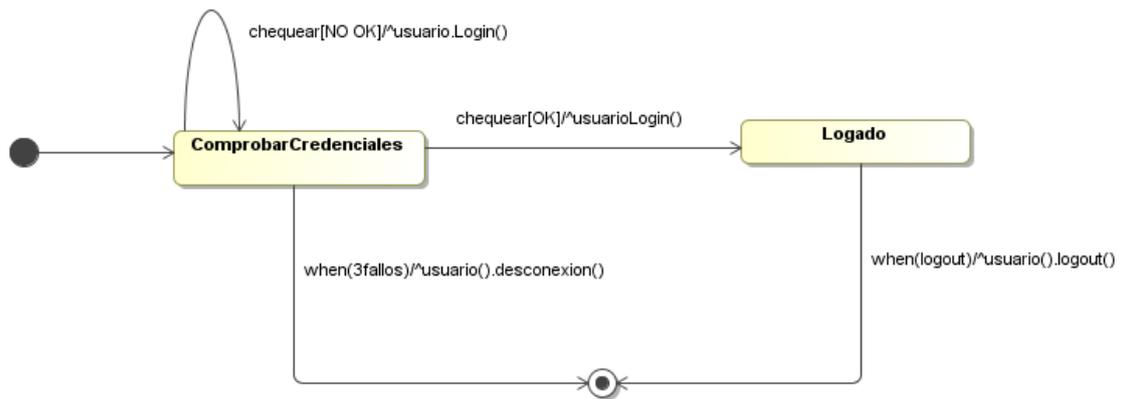


7.3.5. Diagrama de secuencia

En este subapartado detallaremos el diagrama de secuencia que nos mostrara la interacción de un conjunto de objetos en el caso de uso de validación de un usuario en el sistema mediante nombre de usuario y contraseña.



7.3.6. Diagrama de estado



7.3.7. Notación CRC de clases lógica de negocio

Clase: "Sesion"	
Descripción	Clase para gestionar las sesiones de los usuarios.
Tipo de clase	Clase principal
Responsabilidades	La clase se usa para almacenar la sesiones activas de los usuarios
Características	Concreta y persistente
Constructures	+Sesion(idConexion,idUsuario:User);
Atributos	-idSesion:Int -idUsuario:Int
Métodos	-chequearEstado():boolean

Clase: "Conexion"	
Descripción	Clase encargada de establecer las sesiones
Tipo de clase	Clase principal
Responsabilidades	La clase se usa para gestionar el inicio de sesión de los usuarios en el sistema.
Características	Concreta y persistente
Constructures	+Conexion(idusuario:int);
Atributos	-idConexion:int -idUsuario:int
Métodos	-conectar(idusuario:int):boolean -desconectar(idusuario:int):boolean

Clase: "Auditoria"	
Descripción	Clase encargada de auditar el sistema
Tipo de clase	Clase principal
Responsabilidades	La clase se encarga de registrar todas las sesiones establecidas contra el servidor.
Características	Concreta y persistente

Constructures	+Auditoria(idConexion:int,fechaInicio:Date,fechaFin:Date);
Atributos	-idConexion:int -idUsuario:int
Métodos	-registrar(idusuario:int,fechaInicio:Date,fechaFin:Date):Boolean -recuperar():String -chequear():Boolean

7.4. Subsistema de estadísticas

El subsistema de estadísticas se encarga de gestionar y tratar toda la información que se genera con el funcionamiento de la aplicación. Este subsistema solo podrá ser accedido por los usuarios con el rol de administrador del sistema.

Este subsistema permite hacer un seguimiento de todo lo que acontece en el sistema para luego poder adoptar las mejores decisiones en función del uso que se hace del laboratorio y de sus recursos.

El subsistema de estadísticas generara tres tipos de estadísticas:

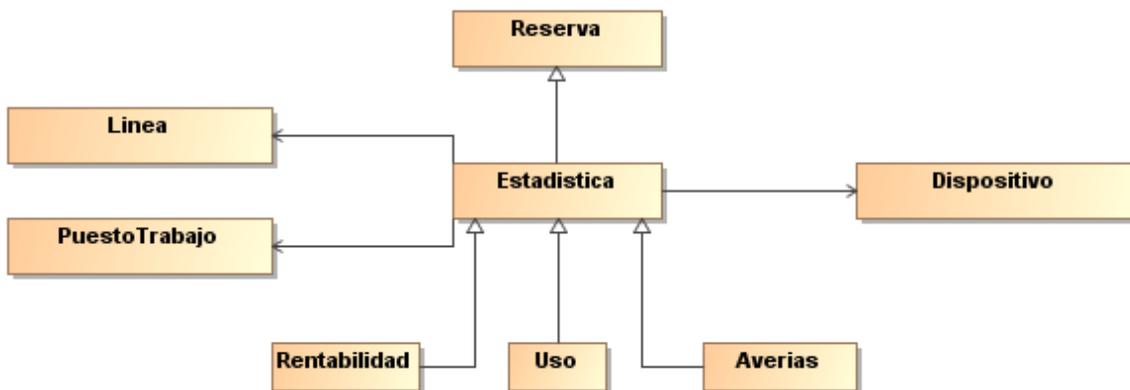
1. **Estadísticas de rentabilidad:** estas detallaran la rentabilidad de cada recurso del laboratorio dividiendo el coste neto de mantenimiento entre el beneficio que generan. De esta manera podremos saber que recursos son más rentables y cuales son deficitarios permitiendo de esta manera optimizar el coste del laboratorio.
2. **Estadísticas de uso:** estas detallaran que recursos del laboratorio son más y menos demandados. Además, también nos mostrara que usuarios hacen un mayor número de solicitudes, renovaciones o cancelaciones, mostrándonos de esta manera el uso que hace cada usuario de los recursos que se les asigna. De esta manera podremos adoptar las medidas necesarias para optimizar los recursos del mismo ya que por ejemplo podremos aumentar o disminuir el número de unidades de un recurso en función de su uso y demanda.
3. **Estadísticas de averías:** estas nos informaran sobre el numero de averías que se dan en las líneas de datos, mostrándonos datos como que líneas han sufrido mas averías en un periodo de tiempo determinado o cuánto tiempo se han tardado en resolverlas. De esta manera podremos dar de baja las líneas más problemáticas.

Toda la información generada por el subsistema de estadísticas podrá ser mostrada según los diferentes filtros del sistema permitiéndonos por ejemplo ordenar la información de mayor a menor y viceversa, limitar el número de datos a mostrar u ordenar por algún otro criterio como mostrar solo los recursos más o menos rentables o más demandados.

Además, una vez generadas los informes estadísticos podremos exportarlos a otros formatos como pdf, html o docx, además de imprimirlos o mostrarlos solo por pantalla.

7.4.1. Diagrama de clases

En este subapartado detallaremos las clases principales que participan en el subsistema de estadísticas omitiendo todas aquellas innecesarias o que ya han sido detalladas en puntos anteriores.



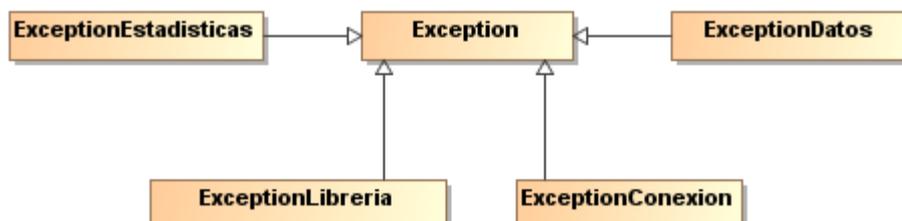
7.4.2. Diagrama de clases gestoras

En este subapartado detallaremos las clases gestoras del subsistema de estadísticas.



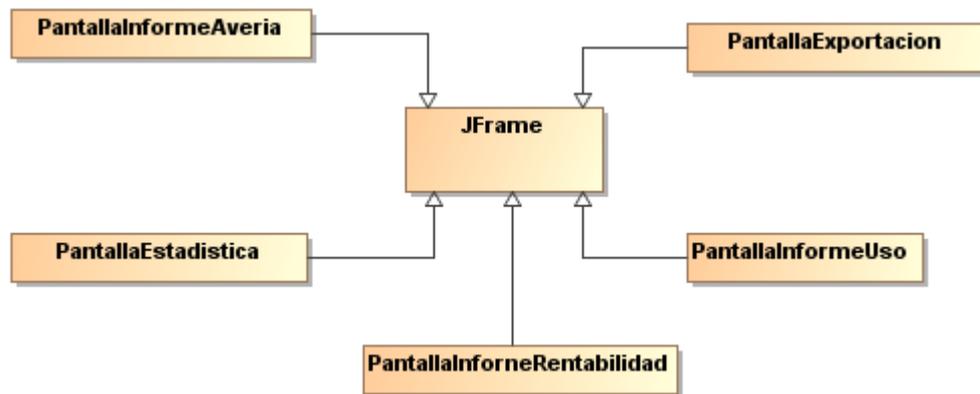
7.4.3. Diagrama de clases de excepción

En este subapartado detallaremos el diagrama de clases de excepción que se encargaran de tratar cualquier problema que surja durante la ejecución de dicho subsistema.



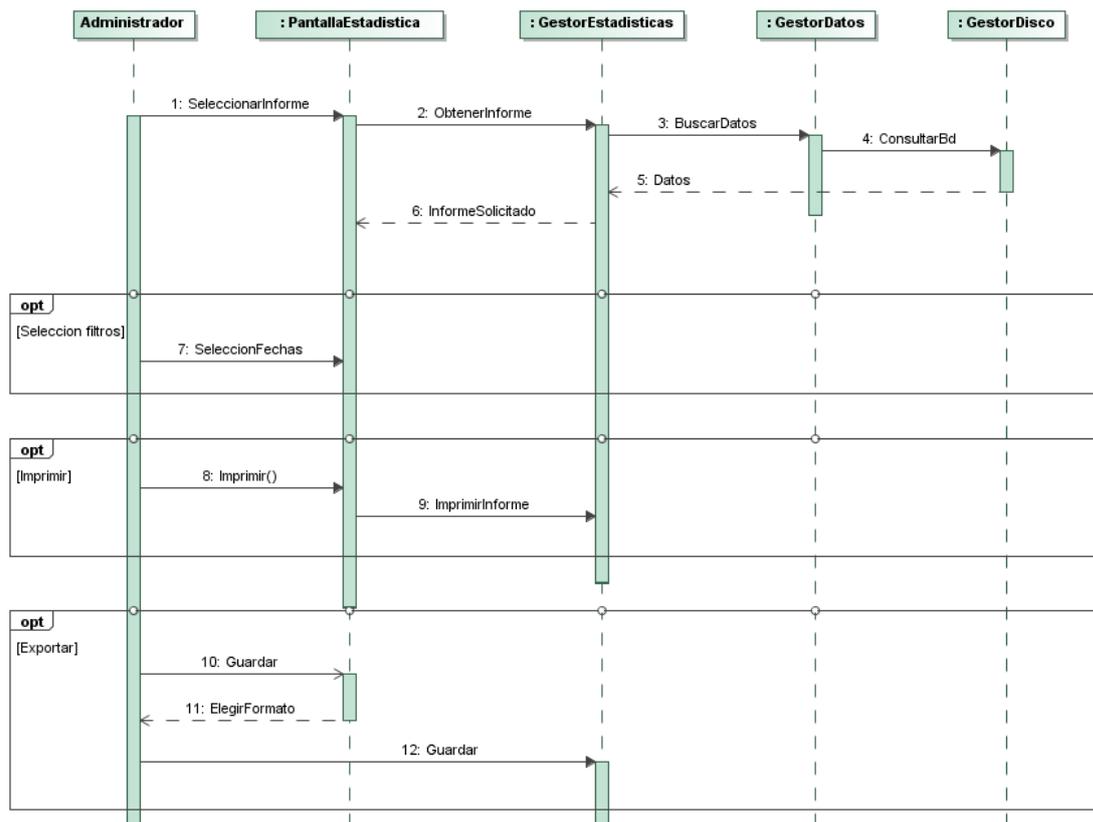
7.4.4. Diagrama de clases frontera

En este subapartado detallaremos el diagrama de clases frontera del subsistema de estadísticas. Mostrando todas aquellas interfaces que tendrá el subsistema.



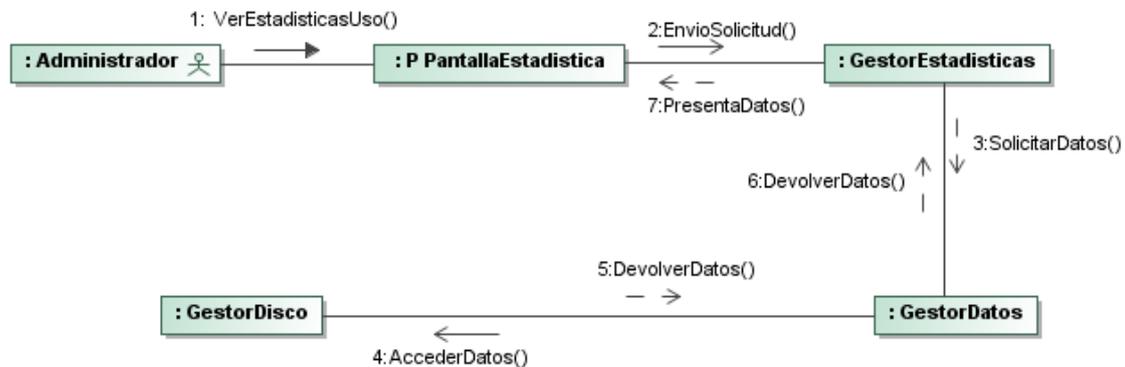
7.4.5. Diagrama de secuencia

En este subapartado detallaremos el diagrama de secuencia del caso de uso: “Generar informe estadístico”



7.4.6. Diagrama de colaboración

En este apartado detallaremos el diagrama de colaboración que nos mostrara la interacción de los objetos del sistema mediante mensajes cuando el administrador solicita un informe estadístico del uso de recursos.



7.4.7. Notación CRC de clases lógica de negocio

Clase: "Estadística"	
Descripción	Clase perteneciente al subsistema de estadísticas.
Tipo de clase	Clase principal
Responsabilidades	Gestionar la información de los distintos recursos del laboratorio para generar las distintas estadísticas del sistema.
Características	Concreta y persistente
Constructores	+estadistica();
Atributos	#idEstadistica: Int #countInformes: Int #imprimir: Boolean #guardar(): Boolean #datos: Datos
Métodos	#exportar(): Boolean +getIdEstadistica(): int +setIdEstadistica(estadistica: int) +getCountInformes(): int +setCountInformes(count: int) +getDatos(): Datos +setDatos(datos: Datos)

Clase: "Rentabilidad"	
Descripción	Clase perteneciente al subsistema de estadísticas que hereda de la clase "Estadística".
Tipo de clase	Clase principal
Responsabilidades	Gestionar la información relativa a la rentabilidad de los recursos del laboratorio.

Características	Concreta y persistente
Constructures	+rentabilidad();
Atributos	-idInforme:Int -idRentabilidad:Int
Métodos	-generar(datos:Datos):List -toString(datos:Datos):String +getIdInforme ():int + setIdInforme (informe:int) +get IdRentabilidad ():int + set IdRentabilidad (count:int)

Clase: "Uso"	
Descripción	Clase perteneciente al subsistema de estadísticas que hereda de la clase "Estadística".
Tipo de clase	Clase principal
Responsabilidades	Gestionar la información relativa al uso de los recursos del laboratorio.
Características	Concreta y persistente
Constructures	+Uso();
Atributos	-idInforme:Int -idUso:Int
Métodos	-generar(datos:Datos):List -toString(datos:Datos):String +getIdInforme ():int + setIdInforme (informe:int) +get IdUso ():int + set IdUso(count:int)

Clase: "Averia"	
Descripción	Clase perteneciente al subsistema de estadísticas que hereda de la clase "Estadística".
Tipo de clase	Clase principal
Responsabilidades	Gestionar la información relativa a averías o roturas de los recursos del laboratorio.
Características	Concreta y persistente
Constructures	+Averia();
Atributos	-idInforme:Int -idAveria:Int
Métodos	-generar(datos:Datos):List -toString(datos:Datos):String +getIdInforme ():int + setIdInforme (informe:int) +get IdAveria():int + set IdAveria(count:int)

8. Diseño de la persistencia

“Cuando veáis a un hombre sabio, pensad en igualar sus virtudes. Cuando veáis un hombre desprovisto de virtud, examinaos vosotros mismos”.

- Confucio (551 AC-478 AC)

La persistencia es la capacidad de preservar la información de un objeto de forma permanente, guardando esta en algún tipo de soporte para poderla recuperar cuando sea necesario, haciendo que esta perdure a lo largo del tiempo.

El modelo entidad-relación E/R ⁵ fue ideado por Peter Chen en el año 1976 y se trata de un modelo que sirve para crear modelos conceptuales de bases de datos.

Dicho modelo debe estar compuesto de:

- Entidades.
- Atributos.
- Relaciones.
- Cardinalidad.
- Llaves.

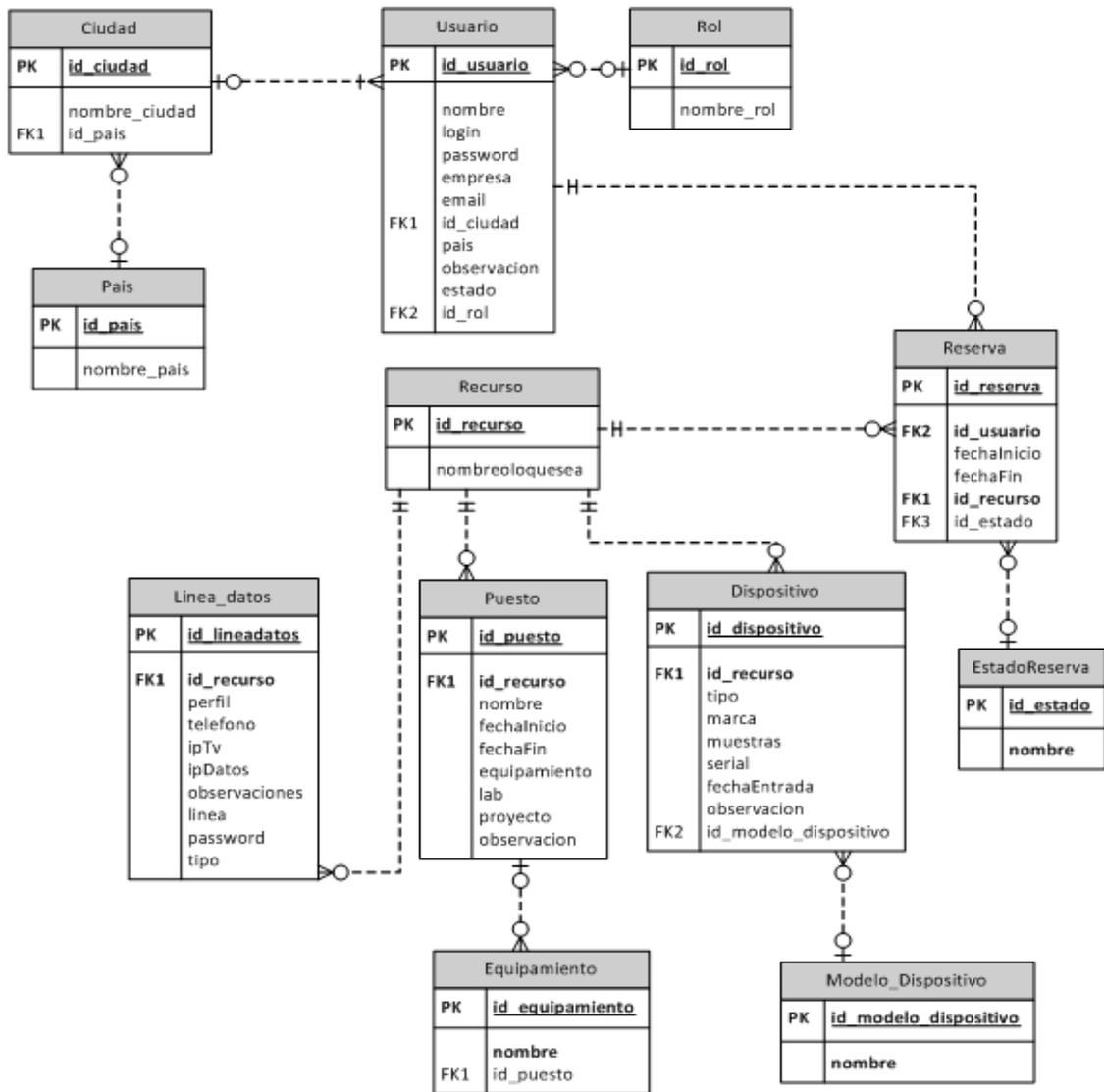
A continuación, detallaremos el diseño de la persistencia del proyecto eGestioLab, detallando el modelado de datos con el diagrama E/R.

8.1. Diagrama E/R

Un diagrama o modelo entidad-relación es una herramienta para el modelado de datos. Esta nos permite representar las entidades junto con sus interrelaciones y propiedades.

A continuación detallaremos el diagrama de entidad-relación del proyecto que estamos desarrollando donde nos mostrara todo el modelado de datos de la aplicación, mostrando en el las distintas relaciones entre sí, así como las distintas propiedades del mismo.

⁵ E/R, siglas en inglés de Entity relationship.



9. Interfaz de usuario

“La virtud no consiste en abstenerse del vicio, sino en no desearlo”.

- George Bernard Shaw (1856-1950)

Las interfaces de usuario son el punto de unión entre el usuario y la aplicación permitiendo a este interactuar con el sistema a través de imágenes y objetos gráficos.

Si estudiamos la etimología de la palabra “interfaz” encontramos una palabra compuesta por dos vocablos:

“**Inter**” que proviene del latín “inter”, y cuyo significado es “entre” o “en medio”, y “**Faz**” proveniente también del latín facies y cuyo significado es “superficie o vista de una cosa”. Por lo tanto, una traducción literal del concepto de interfaz atendiendo a su etimología podría ser “superficie mediadora o superficie en medio de dos cosas”

Dentro de las interfaces de usuario podemos distinguir tres tipos:

1. **Interfaz de software:** encargada de entregar la información sobre los procesos y herramientas de control a través de la pantalla.
2. **Interfaz de hardware:** estos son dispositivos utilizados para ingresar y procesar los datos como por ejemplo, el ratón, el teclado y la pantalla.
3. **Interfaz software-hardware:** permite a la maquina entender el idioma humano y al mismo tiempo permite al humano entender el lenguaje de la maquina.

La interfaz de usuario (GUI) del proyecto eGestionLab está inspirado en el estándar **IEEE Recommended Practice for Software Requirements Specification ANSI/IEEE 830 1998** y *cumplirá con las normativas de AENOR UNE 139801 EX y AENOR UNE 139802* para personas con problemas de accesibilidad, tal como se especifico en la fase de análisis de requisitos.

Además, la interfaz de usuario deberá ser de manejo fácil e intuitivo y deberá presentar un alto grado de usabilidad.

A continuación se mostraran las interfaces de usuario más destacadas por cada subsistema.

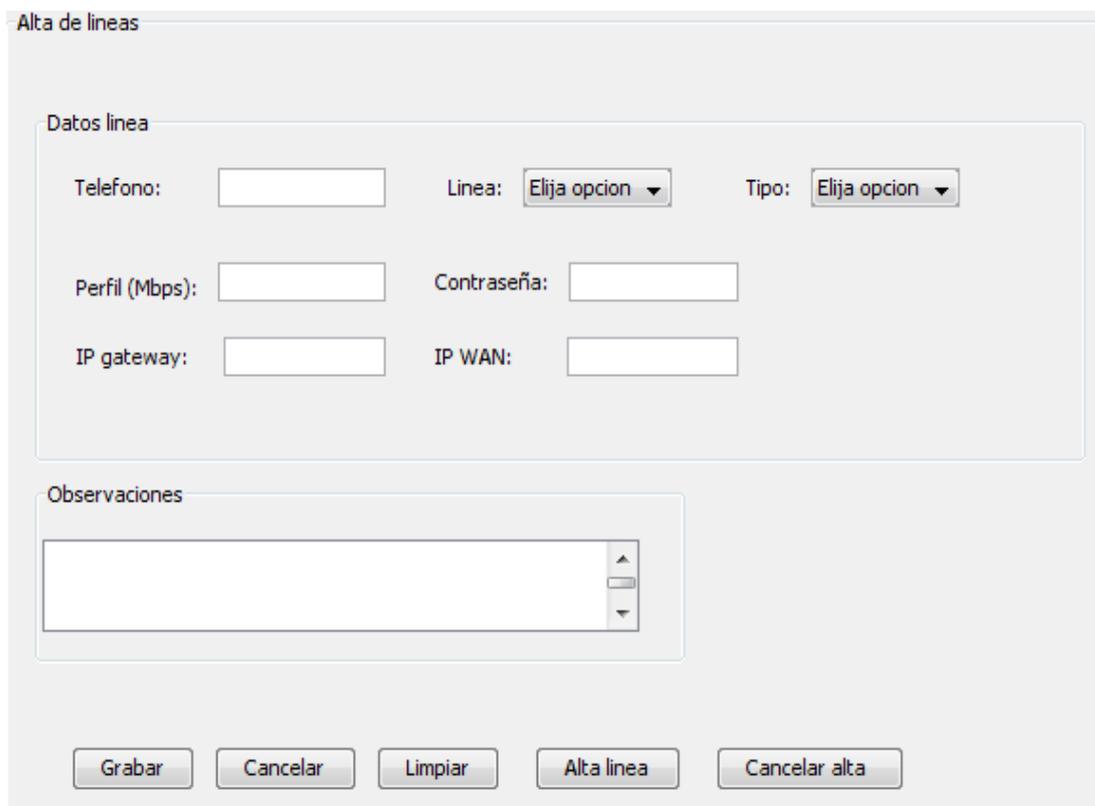
9.1. Subsistema de administración

En este subsistema contaremos con las distintas interfaces que nos permitirán administrar la aplicación.

El sistema contará con la siguiente barra de herramientas que permitirá acceder a las distintas funciones de la aplicación según sea el rol del usuario:

Inicio Gestion Solicitudes Herramientas Ayuda

Para gestionar el alta de líneas el administrador contará con la siguiente interfaz:



Para gestionar cualquier línea en general, el administrador dispondrá de una interfaz desde donde podrá hacer cualquier tipo de gestión con ellas, como la de buscar cualquier línea registrada en el sistema haciendo uso de los distintos filtros de búsqueda, eliminar las líneas necesarias o incluso modificar los datos de cualquiera de ellas. Además de estas funciones, el administrador también podrá asignar a los usuarios de manera manual cualquier línea que le sea solicitada.

Para finalizar, el administrador contará con un campo de observaciones que le permitirá registrar cualquier tipo de información relevante que vaya surgiendo a lo largo del tiempo.

La interfaz de usuario para gestionar las líneas de datos es la siguiente:

Gestión de líneas

Busqueda

Línea: Buscar servicio:

Asignación:

Planta Maqueta Ambos

Filtros

Línea Servicio Asignación Todo

Estado

Libre
 Asignada

Datos

Para gestionar los distintos dispositivos con los que cuenta el laboratorio, el administrador contará con una interfaz única desde donde podrá hacer cualquier tipo de gestión.

Como podremos observar en la siguiente figura, la primera acción que se puede llevar a cabo desde esta interfaz es la de dar de alta en el sistema cualquier tipo de dispositivo que entre en el almacén y que sea apto para asignar a cualquier usuario que lo necesite.

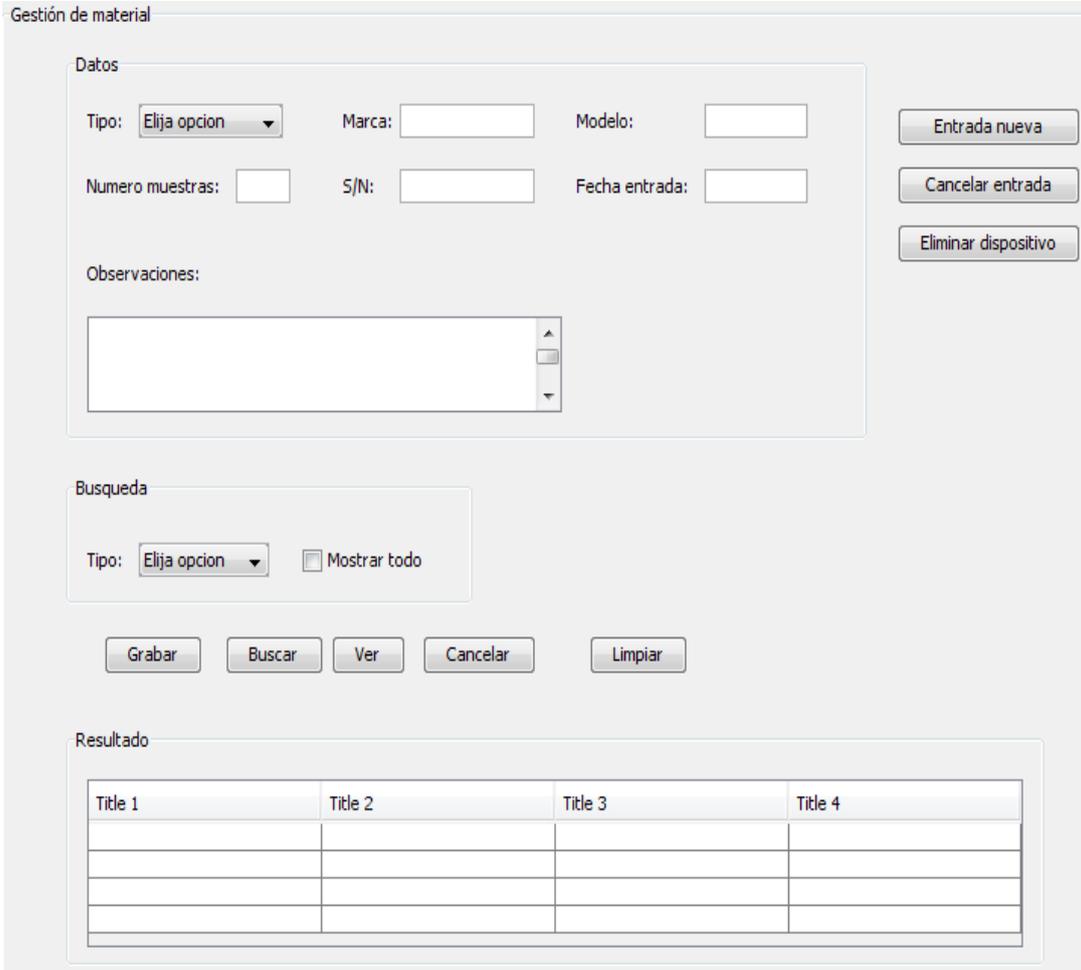
A continuación, el administrador podrá buscar cualquier dispositivo que haya sido dado de alta en el sistema permitiéndole borrarlo del mismo (borrado lógico). Para hacer la búsqueda, el administrador contará con dos filtros para hacer la búsqueda más rápida, por un lado podrá buscar “todos” los dispositivos del sistema o bien podrá buscar solo los dispositivos de un determinado tipo, como por ejemplo, routers⁶, conmutadores o switches⁷, STB⁸, analizadores de video, o incluso adaptadores ATA⁹.

⁶ Anglicismo que se refiere a un dispositivo de hardware usado para la interconexión de redes informáticas

⁷ Dispositivo digital que permite la interconexión de redes de computadoras y que opera en la capa de enlace de datos del modelo OSI. Su función es interconectar dos o más segmentos de red

Para finalizar, el administrador podrá modificar los datos de cualquier dispositivo que este dado de alta en el sistema o añadir cualquier observación que crea oportuna.

La interfaz de gestión de material será la siguiente:



Gestión de material

Datos

Tipo: Marca: Modelo:

Numero muestras: S/N: Fecha entrada:

Observaciones:

Busqueda

Tipo: Mostrar todo

Resultado

Title 1	Title 2	Title 3	Title 4

Para gestionar los distintos puestos de trabajo en el laboratorio el administrador contara con la siguiente interfaz:

⁸ Del ingles Set-Top-Box, es un dispositivo encargado de la recepción y decodificación de señal analógica o digital.

⁹ Dispositivo electrónico que convierte una señal analógica como la voz en paquetes de datos capaces de enviarse a través de una red ip, permitiendo de esta manera hacer llamadas de vozip a través de un teléfono analógico.

Gestión reserva de puestos

Datos

Nombre: Fecha inicio: Fecha fin:

Equipamiento: Laboratorio: Proyecto:

Responsable:

Observaciones:

Busqueda

Nombre Mostrar todo Mostrar historico

Resultado

Como podemos observar en la anterior interfaz, las características de esta son iguales a las anteriores ya que el administrador podrá gestionar todo el apartado de reservas de puesto desde una única.

Para empezar, el administrador podrá dar de alta en el sistema cualquier nuevo puesto que pueda ser asignado a los usuarios del laboratorio, como por ejemplo, cámaras anecoicas¹⁰, simuladores de red, sondas de red¹¹.

Una vez dados de alta, el administrador podrá gestionarlos de la misma manera que el resto de recursos del laboratorio, pudiendo eliminar o modificar cualquier puesto de trabajo dado de alta en el sistema.

Para gestionar a los usuarios del sistema, el administrador contara también con otra interfaz de usuario desde donde podrá gestionar de manera integral todo lo relacionado con los usuarios del sistema.

La interfaz de gestión de usuarios es:

¹⁰ Sala especialmente diseñada para absorber el sonido que incide sobre las paredes, suelo y techo anulando de esta manera todos los efectos de eco y reverberación de sonido. Son apropiadas para todo tipo de pruebas WI-FI.

¹¹ Dispositivo que permite monitorizar y analizar todo el tráfico de red, permitiendo detectar cualquier anomalía que se dé en esta.

Gestión de usuarios

Datos usuario

Nombre: Login: Contraseña:

Empresa: Rol: Repetir contraseña:

e-mail: Ciudad: Pais:

Observaciones:

Busqueda

Login: Usuarios logados Mostrar todos

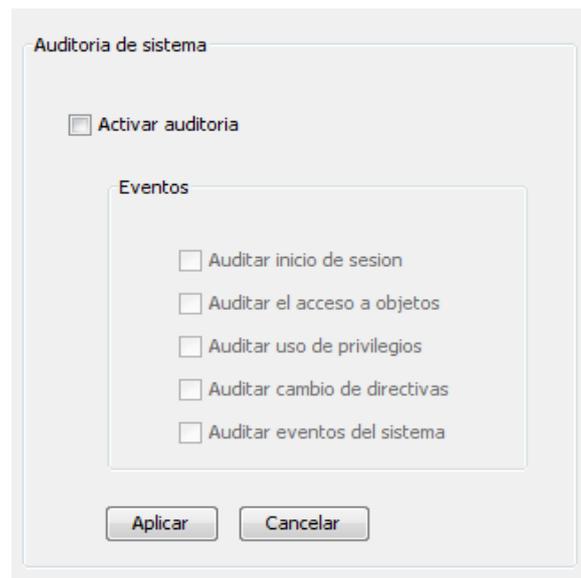
Datos

Title 1	Title 2	Title 3	Title 4

Como podemos observar en la anterior interfaz, la política es la misma que en las anteriores ya que desde una sola interfaz el administrador podrá dar de alta, borrar, modificar o eliminar a cualquier usuario que este dado de alta en el sistema.

Además de esto, el administrador también podrá “Desconectar” a cualquier usuario que se haya quedado logado o bloqueado en el sistema.

Para gestionar las funciones de auditoría de la aplicación usaremos la siguiente interfaz:



En esta podemos observar que el administrador podrá activar o desactivar la auditoria de seguridad según sus necesidades permitiéndole elegir los eventos de seguridad que necesite.

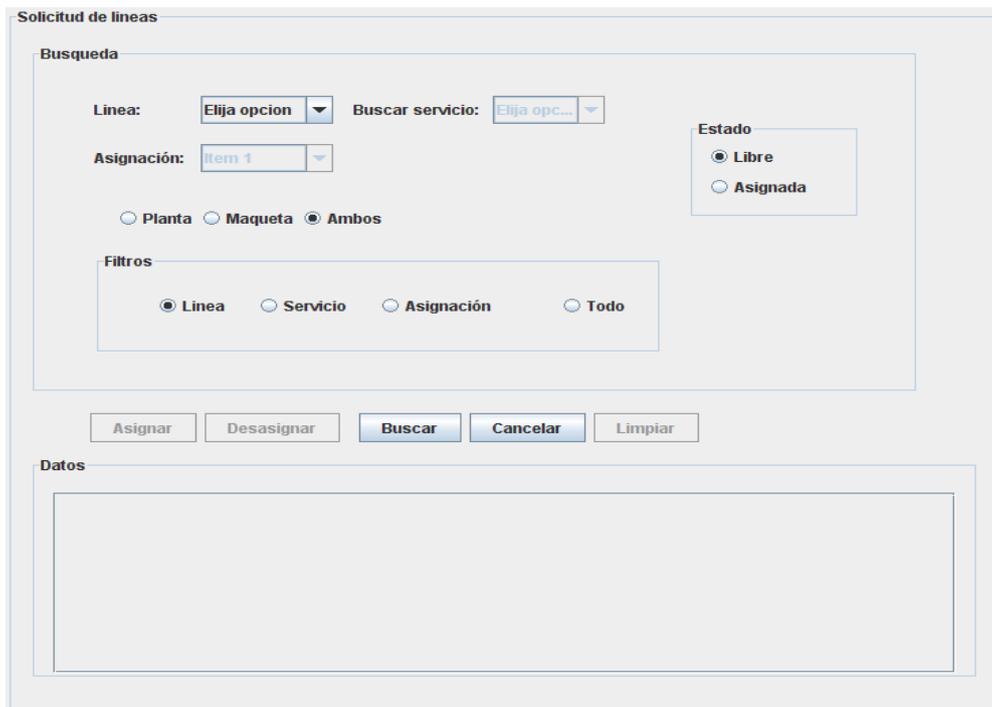
Los eventos que el sistema puede auditar son los siguientes:

- **Inicio de sesión:** el administrador podrá hacer un seguimiento de todos los inicios de sesión validos o fallidos.
- **Acceso a objetos:** el administrador podrá hacer un seguimiento del acceso a todos los objetos que hace el usuario en el sistema, como por ejemplo ficheros o impresoras.
- **Uso de privilegios:** el administrador podrá hacer un seguimiento del uso de los privilegios que hace cada uno de los administradores del sistema para garantizar el buen uso del sistema por parte de estos.
- **Cambio de directivas:** el administrador podrá registrar cualquier cambio de directivas que haga un administrador.
- **Eventos del sistema:** el administrador podrá hacer un seguimiento de los distintos eventos de sistema que la aplicación pueda generar como bloqueos de cuentas o cuelgues de aplicación.

9.2. Subsistema de solicitudes

Los usuarios del sistema contarán con una sola interfaz desde donde podrán solicitar la asignación de cualquier línea de datos así como gestionar las asignaciones que tiene en curso.

La interfaz de solicitud de líneas será la siguiente:



9.3. Subsistema de conexión

En este subapartado detallaremos las interfaces del subsistema de conexión más relevantes.

Estas permitirán al usuario desde la validación en el sistema hasta la recuperación de las credenciales en caso de olvido y detallaremos también los distintos mensajes de error que devolverá el sistema en caso de que haya un error en la validación.

Pantalla de login

Para que un usuario pueda acceder a la aplicación, antes deberá estar registrado en ella y haberse validado en el sistema a través de un nombre de usuario y contraseña o bien mediante un certificado digital. Cuando un usuario se intente logar ¹²en el sistema, verá la siguiente pantalla:

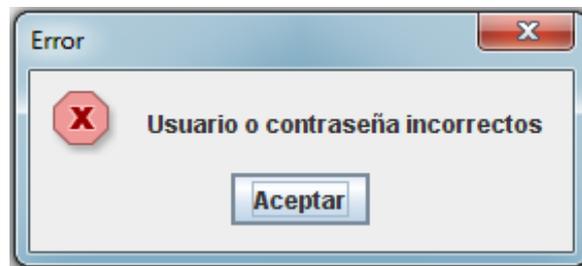
¹² Acción que permite acceder a un sistema restringido mediante la introducción de un nombre de usuario y una contraseña.



Como podemos observar en la interfaz anterior, el usuario también podrá elegir el idioma en el que desea ver la aplicación.

Pantalla de errores

En caso de que al introducir alguna de las credenciales, alguna de esta sea errónea, al usuario le aparecerá el siguiente mensaje de error:



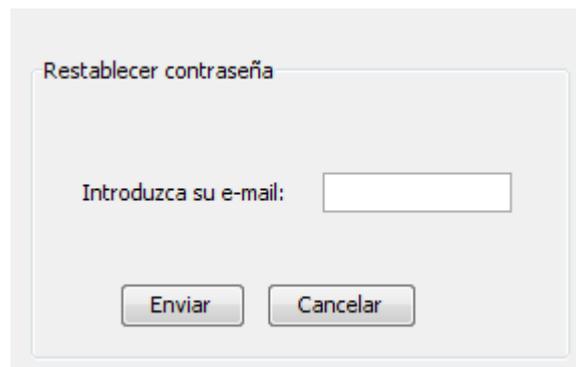
En caso de que el usuario no haya introducido todos los datos necesarios para la validación, a este le aparecerá el siguiente mensaje de error:



Recuperación de credenciales

En caso de pérdida u olvido de la contraseña, el sistema enviara al email suministrado por el usuario sus nuevas credenciales. Cabe destacar que para que el sistema envíe las credenciales por email, la dirección suministrada deberá estar antes registrada en el sistema.

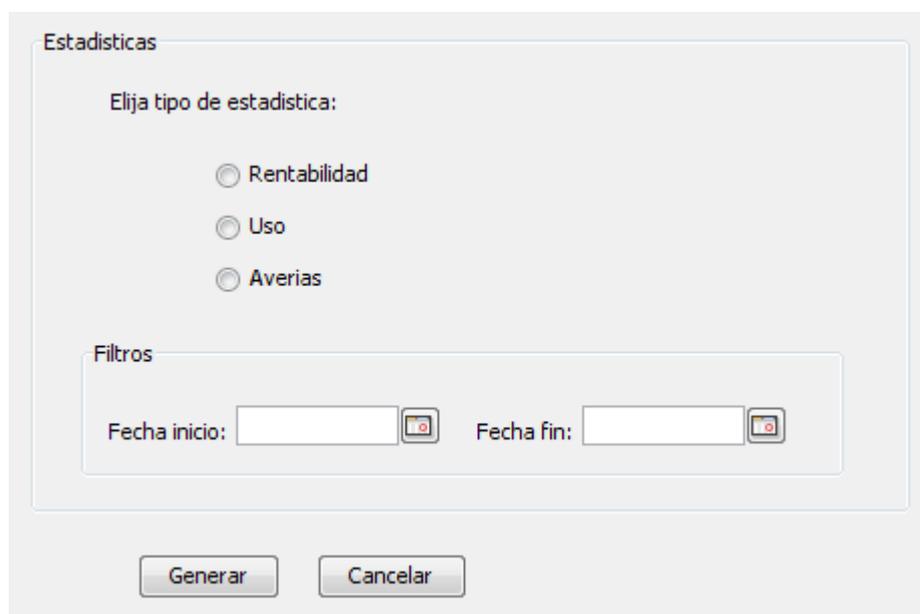
Para restablecer la contraseña, el usuario lo hará a través de la siguiente interfaz:



9.4. Subsistema de estadísticas

En este subapartado mostraremos las interfaces más relevantes del subsistema de estadísticas.

El administrador podrá generar las distintas estadísticas que genera el sistema a través de la siguiente interfaz:



Como podemos observar en la figura anterior, el administrador podrá generar los tres tipos de estadísticas explicadas en el punto anterior. Además de ello, el administrador podrá generar las estadísticas entre un rango de fechas.

Una vez generadas las estadísticas, el sistema las mostrara de la siguiente manera:

Informe averias

Uso por linea

Fecha: 12/02/2012 - 12/05/2012

Exportar

Telefono	Linea	Averias/mes	Averia año	% disponibilidad
915332318	TRIO	2	24	67,01%
915231456	DUO	1,08	13	50,2%
912245617	DUO	0,58	7	23,3%
914709876	DUO	0,16	2	15,02%

10. Conclusiones

“El amor y el deseo son las alas del espíritu de las grandes hazañas”.

- Johann Wolfgang Goethe (1749-1832).

10.1. Conclusiones científicas

El objetivo de todo proyecto técnico es satisfacer las necesidades del cliente, desarrollando un producto que cumpla con todas sus expectativas.

Para la consecución del proyecto se ha desarrollado un plan de trabajo validado por el consultor del proyecto y que ha sido llevado a cabo respetando todos los hitos marcados y validados por él.

El resultado final del proyecto ha sido que se ha desarrollado un producto potente, robusto y escalable que cumple con todas las expectativas del cliente en tiempo y formas, abriéndonos la puerta en participar en futuros proyectos profesionales.

10.2. Conclusiones personales

A nivel personal, la participación en este proyecto ha supuesto todo un reto para mí.

El proyecto final de carrera es la culminación de un largo y duro trabajo durante muchos años en los que he tenido que combinar estudios y trabajo, obligándome a renunciar o dejar de lado muchos otros proyectos personales.

Sin embargo, a pesar de todos los sacrificios y vicisitudes vividas, creo que la recompensa final merece la pena ya que no solo me ha hecho crecer profesionalmente sino a nivel personal.

Por ello, para terminar, puedo asegurar que la experiencia de esta asignatura ha sido totalmente enriquecedora.

Apéndice

I. Glosario de términos

A

Actores, son los que interactúan o hacen uso del sistema, 17

AENOR UNE 139801, normativa que establece los requisitos de accesibilidad al ordenador para personas con problemas de accesibilidad, 59

AENOR UNE 139802, normativa que establece los requisitos de accesibilidad al software, 59

C

Casos de uso, es un diagrama de comportamiento basado en UML, 18

Clases frontera, clases que representan a la interfaz de usuario, 34

D

Diagramas de estado, diagrama de modelado de sistemas que muestra el conjunto de estados por los que pasa un objeto, 32

Diagramas de secuencia, diagrama de modelado de sistemas que nos muestra la interacción entre objetos, 32

E

E/R, diagrama de modelado de datos que se utiliza para diseñar la persistencia de la aplicación, 57

G

GUI, en inglés gráfico user interface es un programa informático que actúa de interfaz de usuario utilizando un conjunto de imágenes y objetos gráficos, 6

I

I+D, término abreviado de Investigación y desarrollo y hace referencia a la investigación en ciencias aplicadas o bien a ciencia básica, 6

i386, tipo de arquitectura de un microprocesador, 15

IEEE 802.3, Primer estándar que definió a Internet, 13

L

LAN, siglas de Local Area Network y es una red de ámbito local, 13

Login, acción que permite autenticarse en el sistema para acceder a él, 27

Logs, término anglosajón que se refiere a un fichero que se usa para guardar información, 17

N

Net, es un Framework de Microsoft desarrollado por este que permite un rápido desarrollo de aplicaciones informáticas, 10

Net Framework, marco de trabajo diseñado para facilitar el desarrollo de software, 15

Notación CRC, sistema de notación que se usa para representar los métodos y atributos de un diagrama de clases, 6

P

Persistencia, es la acción para preservar la información de forma permanente, 57

PostgreSQL, gestor de base de datos relacional orientada a objetos y publicada bajo licencia BSD, 10

U

UML, es un lenguaje gráfico para visualizar, especificar y documentar cada una de las partes que comprende el desarrollo de software, 32

II. Bibliografía

P. Coad y E. Yourdon. Object Oriented Analysis. Prentice Hall, 1991. 22, 107, 348, 353.

Englewood Cliffs, Open Software Foundation (Editor). OSF/Motif Style Guide: Revisión 1.2. Prentice Hall International, Nueva Jersey, EE.UU. 1993.

Aenor, Aplicaciones informáticas para personas con discapacidad. Requisitos de accesibilidad al ordenador, Sep 2003. Disponible en la Universidad de la Coruña: <http://www.udc.es/fcs/es/web-to/terapia/asignaturas/toyafam/08tema/UNE139802-2003.pdf>

IEEE – “*IEEE Recommended Practice for Software Requirements Specifications*”. Disponible en: <http://standards.ieee.org/findstds/standard/830-1998.html>

Molich, R., y Nielsen, J., Heuristic evaluation of user interfaces, Proceedings of the ACM CHI’90 Conference, pp. 249-256, 1990.

Mar del sur, Wikipedia. Diseño Lógico de Bases de Datos Modelos de Datos. Disponible en: <http://bd.eui.upm.es/BD/docbd/tema/tema2.pdf>

UA31. Wikipedia. *Base teórica de modelo entidad-relación*. Disponible en: http://es.wikipedia.org/wiki/Modelo_entidad-relaci%C3%B3n

EmausBot . Wikipedia. *Lenguaje Unificado de Modelado*. Disponible en: http://es.wikipedia.org/wiki/Lenguaje_Unificado_de_Modelado

Microsoft. Visio Tutorial. Disponible en: <http://msoffice.us>

Tigris.org, Open Source Software Engineering Tools. ArgoUml. Disponible en: <http://argouml.tigris.org/>

Universidad de Chile. “*Modelado de clases*”. Disponible en: <http://www.dcc.uchile.cl/~psalinas/uml/modelo.html>