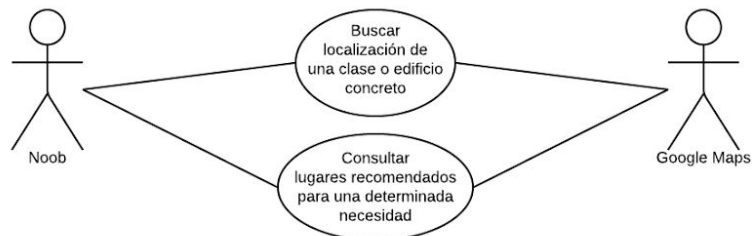
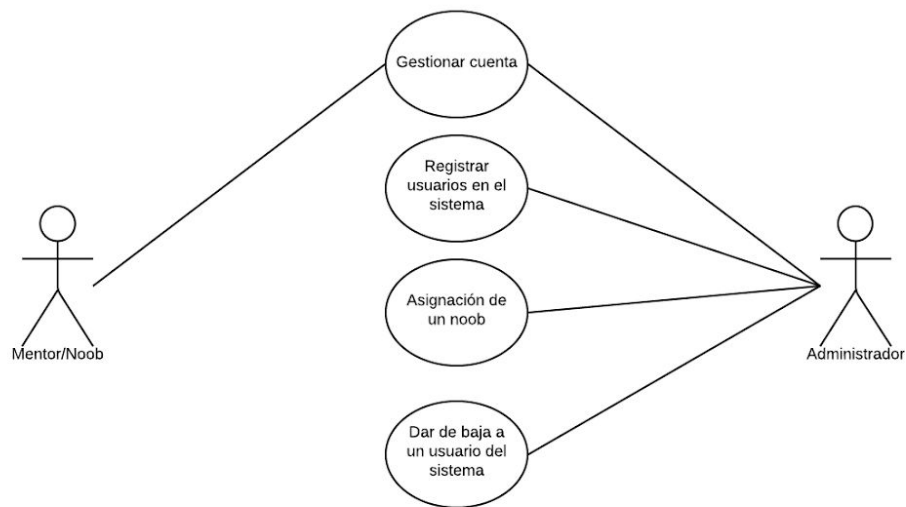
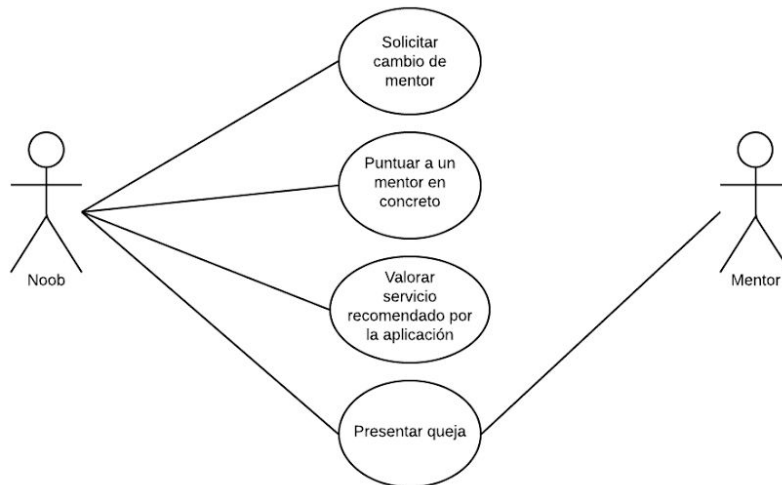
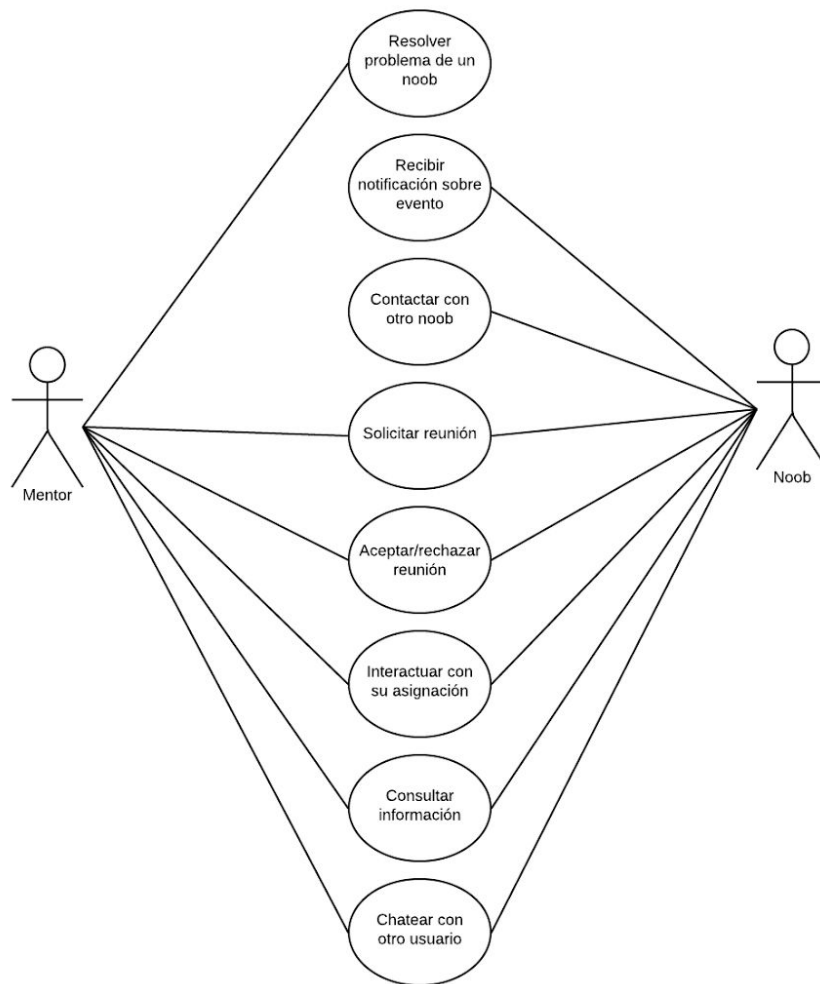


imNoob - ESPECIFICACIÓN DE REQUISITOS DEL SOFTWARE

1. ESPECIFICACIÓN FUNCIONAL

1.1. Diagrama de casos de uso





Caso de uso UC001 - Gestionar cuenta

Caso de uso UC002 - Registrar usuarios en el sistema

Caso de uso UC003- Asignación de un noob

Caso de uso UC004 - Dar de baja a un usuario del sistema

Caso de uso UC005 - Interactuar con su asignación

Caso de uso UC006 - Resolver problema de un noob

Caso de uso UC007 - Contactar con otro noob

Caso de uso UC008 - Solicitar reunión

Caso de uso UC009 - Aceptar/rechazar reunión

Caso de uso UC010 - Recibir notificación sobre evento

Caso de uso UC011 - Consultar información

Caso de uso UC012 - Solicitar cambio de mentor

Caso de uso UC013 - Puntuar a un mentor en concreto

Caso de uso UC014 - Valorar servicio recomendado por la aplicación

Caso de uso UC015 - Presentar queja

Caso de uso UC016 - Buscar localización de una clase o edificio concreto

Caso de uso UC017 - Consultar lugares recomendados para una determinada necesidad

Caso de uso UC018 - Chatear con otro usuario

1.2. Descripción individual de los casos de uso

Caso de uso UC001 - Gestionar cuenta

Los usuarios del sistema deben poder modificar los diferentes ajustes de su cuenta o perfil, como la contraseña o las horas libres. También deben poder iniciar sesión o salir de esta.

Caso de uso UC002 - Registrar usuarios en el sistema

Los administradores deben poder registrar nuevos usuarios en el sistema.

Caso de uso UC003- Asignación de un noob

Los administradores deben poder añadir un noob a un mentor.

Caso de uso UC004 - Dar de baja a un usuario del sistema

Los administradores deben poder dar de baja a un usuario, ya sea noob o mentor, del sistema.

Caso de uso UC005 - Interactuar con su asignación

Los usuarios del sistema deben poder hablar con los diferentes alumnos con los que ha sido emparejado para poder concretar la realización de reuniones o para resolver pequeñas consultas.

Caso de uso UC006 - Resolver problema de un noob

Los usuarios del sistema cuyo rol sea el de mentor deben poder resolver una duda previamente hecha por su noob asignado.

Caso de uso UC007 - Contactar con otro noob

Los usuarios del sistema cuyo rol sea el de noob deben poder contactar con otro noob registrado del sistema, de su mismo centro educativo.

Caso de uso UC008 - Solicitar reunión

Los usuarios del sistema deben poder solicitar una reunión con su respectivo mentor o sus noobs proponiendo una fecha y hora de la misma seleccionadas de entre las posibles indicadas en los perfiles.

Caso de uso UC009 - Aceptar/rechazar reunión

Los usuarios del sistema deben poder aceptar o rechazar una reunión solicitada en caso que en la fecha y hora indicada no le sea posible asistir.

Caso de uso UC010 - Recibir notificación sobre evento

Los usuarios del sistema cuyo rol sea el de noob deben poder recibir notificaciones de eventos del centro en el que estudian.

Caso de uso UC011 - Consultar información

Los usuarios del sistema deben poder consultar el nombre, la fotografía o las horas libres de los usuarios con los que ha sido asignado.

Caso de uso UC012 - Solicitar cambio de mentor

Los usuarios del sistema cuyo rol sea el de noob deben poder pedir un cambio de mentor.

Caso de uso UC013 - Puntuar a un mentor en concreto

Los usuarios del sistema cuyo rol sea el de noob deben poder dar una valoración y/o puntuación a su mentor asignado.

Caso de uso UC014 - Valorar servicio recomendado por la aplicación

Los usuarios del sistema cuyo rol sea el de noob deben poder valorar un servicio recomendado anteriormente por la aplicación.

Caso de uso UC015 - Presentar queja

Los usuarios del sistema deben poder presentar una queja formal de los alumnos con los que ha sido asignado en caso de un mal uso por parte de estos último de los recursos proporcionados.

Caso de uso UC016 - Buscar localización de una clase o edificio concreto

Los usuarios del sistema cuyo rol sea el de noob deben poder buscar en la propia aplicación la ubicación de un aula o edificio de su centro educativo.

Caso de uso UC017 - Consultar lugares recomendados para una determinada necesidad

Los usuarios del sistema cuyo rol sea el de noob deben poder elegir de entre una selección de necesidades o actividades para consultar un lugar recomendado para llevarla a cabo.

Caso de uso UC018 - Chatear con otro usuario

Los usuarios del sistema, ya sean mentores o noobs, deben tener la posibilidad de comunicarse con otros usuarios para realizar algún tipo de consulta.

Cabe remarcar que algunos de estos casos de uso deberán ser adaptados en función de cada centro adicional que adquiera nuestros servicios. Estos son todos aquellos relacionados con la gestión de usuarios y los que necesitan información diferencial de cada centro en particular. Principalmente, lo que conllevará más esfuerzo de esta adaptación es el acoplamiento de la aplicación con el sistema de automatrícula, en caso de que este exista, de la universidad en cuestión. De esta manera, en el momento de la matrícula de los estudiantes, estos podrán tener acceso a nuestro sistema.

En resumen, los casos de uso que requerirán una adaptación son los siguientes:

- Buscar localización de una clase o edificio concreto
- Consultar lugares recomendados para una determinada necesidad
- Registrar usuarios en el sistema

Por último, cabe destacar la posibilidad de que algún centro pueda requerir un caso de uso específico.

2. ESPECIFICACIÓN NO FUNCIONAL

#01 Apariencia.

- **Descripción:**

Queremos que nuestro sistema tenga un formato atractivo de manera que pueda gustar a los usuarios de la aplicación.

- **Justificación:**

El objetivo de este requisito es que nuestro sistema sea capaz de atraer al mayor número de usuarios de manera que estos estén dispuestos a utilizarlo de manera continua y facilitar así que el usuario pueda recordar nuestro sistema ya que les resultara estéticamente agradable y familiar.

- **Condición de satisfacción:**

Diremos que nuestro sistema cumple con este requisito si durante el desarrollo del mismo nos ajustamos a los patrones y estilos ya definidos por los sistemas operativos que vamos a utilizar. Además, durante la fase de testeo de la aplicación al menos un 75% de los encargados de hacer los tests llegan a la conclusión de que se trata de un sistema que posee un atractivo de cara al público bastante alto.

#02 Facilidad de uso

- **Descripción:**

La aplicación debe de ser fácil de usar para cualquier estudiante de universidad.

- **Justificación:**

Al ser una aplicación que, en principio, va dirigida a un público bastante concreto, como son los usuarios de las universidades, debe tener la intención de facilitarles la vida. Por lo tanto, es esencial que sea sencilla de usar para

personas familiarizadas con aplicaciones de este tipo ya que de este modo es más probable que sea el sistema favorito frente a otros del sector.

- **Condición de satisfacción:**

Podremos concluir que nuestra aplicación satisface el requisito si al hacer una encuesta anónima después de 2 semanas de familiarización, un 85% de los usuarios hacen uso de ella, cuando se requiere.

#03 Fiabilidad y disponibilidad.

- **Descripción:**

Es imperativo que nuestro sistema esté activo o en funcionamiento siempre (en el peor de los casos, la mayor parte del tiempo), evitando grandes caídas que puedan afectar a dicho funcionamiento.

- **Justificación:**

Con el objetivo de que nuestro sistema tenga éxito, es necesario que esté en disponible en cualquier momento para cada uno de nuestros usuarios y sobretodo que funcione a su máximo rendimiento. Además de que nuestro sistema debe evitar las caídas continuas, mientras esté operativo, tiene la obligación de prestar sus servicios de manera completa con el objetivo de evitar valoraciones desfavorables por parte de nuestros clientes y perder así su confianza depositada.

- **Condición de satisfacción:**

Diremos que nuestro sistema cumple con el requisito de fiabilidad y disponibilidad, si y solo si, al pasar las pruebas y tests pertinentes no observamos caídas importantes que superen el 5% del tiempo de funcionamiento. Y que en el momento de estar operativo observamos que nuestro sistema presta sus servicios al 100%.

#04 Aprendizaje.

- **Descripción:**

El producto debe ser usable para los usuarios sin la necesidad de un aprendizaje previo.

- **Justificación:**

La aplicación debe ser fácil de usar para los usuarios, de manera que puedan interactuar con esta de manera continua, favoreciendo así la voluntariedad de los clientes para hacer uso de la misma.

- **Condición de satisfacción:**

Diremos que nuestro sistema satisface el requisito de Aprendizaje si un 75% de los usuarios son capaces de usar el sistema sin necesidad de consultar ningún tipo de manual de usuario.

#05 Comprensibilidad y pulidez.

- **Descripción:**

El producto debe ser claro y pulido para que los usuarios que lo usen entiendan todo lo que se muestre y además no haya información innecesaria.

- **Justificación:**

La aplicación debe ser fácil de comprender, haciendo uso de símbolos y palabras que sean conocidos de manera natural por parte de los usuarios. Además, debe evitar mostrar al usuario detalles de su construcción ya que sólo quitarían comprensibilidad y pulidez.

- **Condición de satisfacción:**

Diremos que nuestro sistema satisface el requisito si el 90% de la información mostrada es la necesaria para la acción realizada y no se muestra jamás información interna del sistema.

#06 Accesibilidad.

- **Descripción:**

Es necesario que nuestro sistema tenga la capacidad de poder adaptarse a aquellas personas que poseen una discapacidad con el objetivo de que éstas puedan hacer uso de la aplicación.

- **Justificación:**

Con el objetivo de no perder una parte importante de los clientes potenciales, como son las personas que tienen alguna discapacidad, es necesario que nuestro sistema pueda proporcionar a dichas personas una experiencia satisfactoria dotando a nuestro sistema de funcionalidades con una facilidad tanto técnica como visual para poder alcanzar dicho objetivo.

- **Condición de satisfacción:**

Diremos que nuestro sistema cumple con el requisito de Accesibilidad, si y sólo si, después de realizar los tests y probar la aplicación, al menos un 70% de los usuarios que poseen alguna discapacidad siguen haciendo uso de ésta. De lo contrario diremos que no se cumple este requisito.

#07 Velocidad y latencia.

- **Descripción:**

Es necesario conseguir que nuestro sistema tenga una velocidad y latencia altamente aceptables de cara al público que el que hará uso de la misma.

- **Justificación:**

Con el objetivo de que los usuarios estén conformes con nuestro sistema y tengan una experiencia satisfactoria, la velocidad y latencia de nuestro sistema tienen que ser lo suficientemente aceptables de manera que estén a la altura de nuestros usuarios. Con esto evitaremos que estos no dejen de hacer uso de nuestra aplicación por motivos de disgusto o disconformidad.

- **Condición de satisfacción:**

Diremos que nuestro sistema cumple con el requisito de Velocidad y latencia, si y sólo si, después de hacer los tests pertinentes, el sistema proporciona sus servicios y funcionalidades en un tiempo razonable, a una velocidad esperada por los usuarios y sin ningún contratiempo al menos un 90% de las veces.

#08 Capacidad.

- **Descripción:**

El objetivo de este requisito es que nuestro sistema tenga la capacidad de gestionar el tráfico de los usuarios sin ningún tipo de conflicto.

- **Justificación:**

Como no somos conscientes del número de usuarios que harán uso de nuestro sistema, éste debe estar preparado para saber manejar este tráfico. Para ello el sistema debe ser capaz de soportar algunos picos de volumen de usuarios de manera que el sistema continúe ofreciendo sus servicios sin ningún problema; de lo contrario nuestro sistema estaría fallando y dando un mal servicio a nuestros usuarios.

- **Condición de satisfacción:**

Nuestro sistema cumplirá con este requisito, si y sólo si, en una fase de pruebas es capaz de superar todos y cada uno de los tests a los cuales ha sido sometido.

#09 Mantenimiento.

- **Descripción:**

Es necesario que nuestro sistema pueda actualizarse cuando sea necesario, para poder corregir cualquier error en la aplicación.

- **Justificación:**

Si queremos brindar el mejor servicio a nuestros clientes, no podemos diseñar un sistema que contenga errores de aplicación. Para ello necesitamos poder actualizar y corregir dichos errores de manera continua, para evitar descontento entre nuestro público.

- **Condición de satisfacción:**

Podemos decir que nuestro sistema satisface el requisito de Mantenimiento, si al momento de notificar un error en la aplicación no tarda más de 1 día en arreglar dicho problema y presentar la nueva versión a los usuarios.

#10 Soporte.

- **Descripción:**

Nuestro sistema tiene que tener una atención al cliente siempre que éste requiera ayuda para resolver cualquier duda.

- **Justificación:**

Con el objetivo de dar una atención de calidad a nuestros usuarios, es imperativo que nuestro sistema tenga un sistema de atención al cliente para solucionar cualquier tipo de problema lo más pronto posible.

- **Condición de satisfacción:**

Diremos que nuestro sistema satisface el requisito de Soporte, si un usuario cualquiera puede realizar y resolver sus dudas en cualquier momento contactando con personal de soporte dedicado a dicha tarea.

#11 Adaptabilidad.

- **Descripción:**

Nuestro sistema tiene que tener la capacidad de poder adaptarse a sistemas operativos actuales.

- **Justificación:**

Con el objetivo de ofrecer un software lo más aceptable posible, éste tiene que ser adaptable, de manera que lo puedan usar el mayor número de personas.

- **Condición de satisfacción:**

Diremos que nuestro sistema satisface el requisito de Adaptabilidad si es compatible con cualquier versión de Android e IOs.

#12 Seguridad de acceso.

- **Descripción:**

El objetivo de este requisito es poder garantizar que los datos y diferentes funcionalidades de nuestro sistema en su totalidad, sólo puedan ser accesibles por aquellos usuarios que están autorizados para dicha acción.

- **Justificación:**

En nuestro sistema, como es normal, existen diferentes tipos de funcionalidades y algunas o muchas de ellas han sido diseñadas con el objetivo de que sean accesibles por determinados tipos de usuarios que interactúan con nuestro sistema, como por ejemplo un mentor tiene funcionalidades que un usuario “noob” no tiene. Por lo tanto para lograr un buen funcionamiento del sistema es necesario que cada tipo de usuario tenga acceso a sus funcionalidades específicas.

- **Condición de satisfacción:**

Diremos que nuestro sistema cumple con el requisito de acceso si éste está diseñado de tal forma que toda la información como por ejemplo: personal, privada, sensible y las diferentes funcionalidades del sistema únicamente son accesible por los usuarios que tengan autorización para ello.

#13 Privacidad.

- **Descripción:**

El objetivo de este requisito es que nuestro sistema pueda ofrecer a todos y cada uno de nuestros usuarios la privacidad necesaria en cuanto al uso de sus datos personales, así como también cumplir con las leyes que corresponden al tratamiento de éstos.

- **Justificación:**

En nuestro sistema se va a contener la información privada de cada uno de nuestros usuarios, por este motivo es obligatorio que podamos asegurar la protección necesaria de todo ello y garantizar que solo los usuarios autorizados tendrán acceso a estos datos. Además es imperativo también garantizar que los usuarios que tengan acceso a los datos personales de otro tipo de usuario hagan un uso correcto y estrictamente legal de éstos, de manera que así podamos dar una imagen de seguridad de cara al público y facilitar el uso de nuestro sistema.

- **Condición de satisfacción:**

Diremos que nuestro sistema satisface el requisito de Privacidad, si y sólo si, durante las fases de pruebas no se encuentre ninguna forma de acceder a datos restringidos para cada tipo de usuario.

#14 Personalización e internacionalización.

- **Descripción:**

La aplicación debe ofrecer a cada usuario un uso personalizado.

- **Justificación:**

Al ser una aplicación que va dirigida a diferentes tipos de usuarios (mentores, “noobs”,...), es importante que cada usuario pueda personalizar su uso y entorno según sus necesidades.

- **Condición de satisfacción:**

Podemos decir que nuestro sistema satisface el requisito cuando una vez puesto en marcha, el grado de satisfacción de los usuarios en cuanto a la personalización llegue al 70%.

#15 Requisitos para la interfaz con sistemas adyacentes.

- **Descripción:**

Nuestro sistema tiene que ser capaz de adaptarse a los cambios y/o actualizaciones de los sistemas adyacentes con los que interactúa.

- **Justificación:**

Como nuestro sistema es dependiente de sistemas externos, como por ejemplo Google Maps, el cual nos permite realizar la localización de edificios de un centro educativo en concreto. Es imperativo que nuestro sistema siempre use las versiones que están a la vanguardia de este sistema externo.

- **Condición de satisfacción:**

Diremos que nuestro sistema satisface el requisito de Requisitos para la interfaz con sistemas adyacentes, si es compatible con las últimas 4 versiones del sistema externo Google Maps.

#16 Inmunidad.

- **Descripción:**

Nuestro sistema tiene que tener la capacidad de defenderse de cualquier software malicioso.

- **Justificación:**

Con el objetivo de poder ofrecer un servicio de calidad y exitoso, es necesario garantizar la seguridad de nuestro sistema, de manera que no se puedan dar situaciones como por ejemplo, robos de datos o virus de cualquier tipo.

- **Condición de satisfacción:**

Diremos que nuestro sistema satisface el requisito de Inmunidad, si es capaz de evitar, frenar y responder a los ataques que puedan ocurrir provenientes de software maliciosos en el mínimo tiempo posible.

#17 Requisitos de cumplimiento legal.

- **Descripción:**

Nuestro sistema y sus futuras actualizaciones tienen que cumplir con las leyes que las involucren

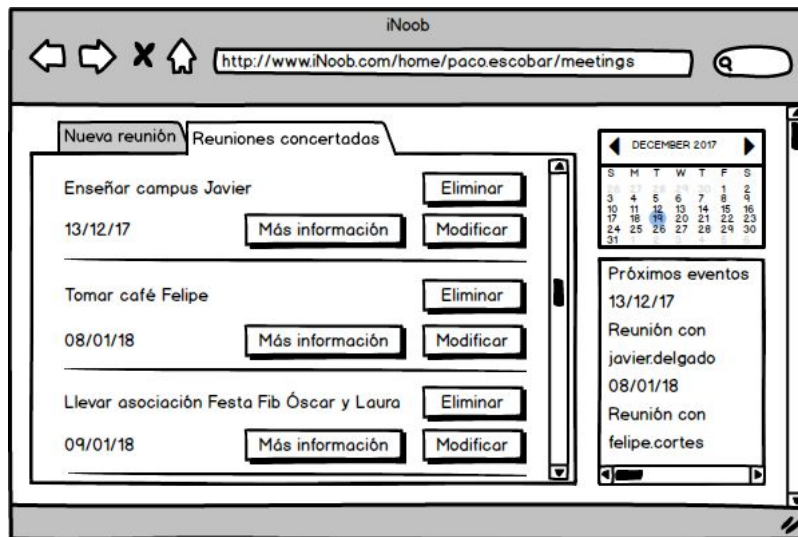
- **Justificación:**

Todos los usuarios de nuestro sistema, son susceptibles de aceptar las condiciones de uso y de servicio de manera que se pueda tener una relación legal.

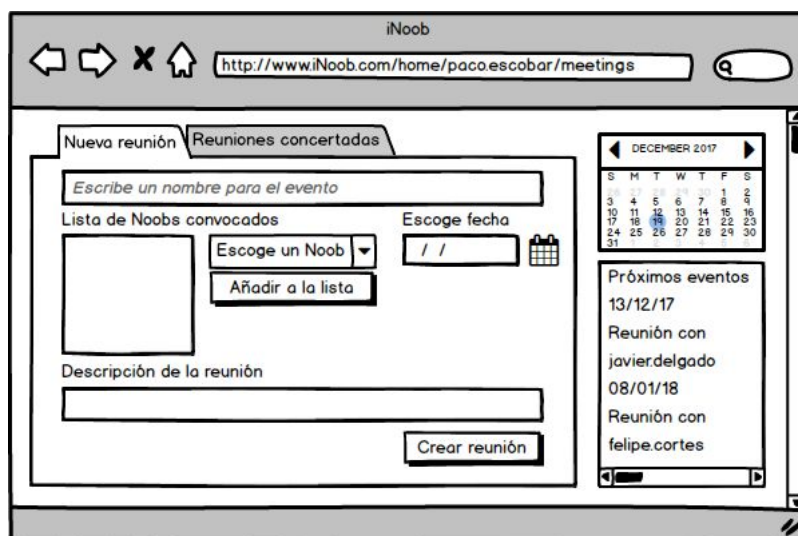
- **Condición de satisfacción:**

Diremos que nuestro sistema satisface el criterio de cumplimiento legal, si toda actualización del mismo es revisada convenientemente en términos legales para no estar abiertos a cometer cualquier tipo de infracción.

3. MOCK-UPS

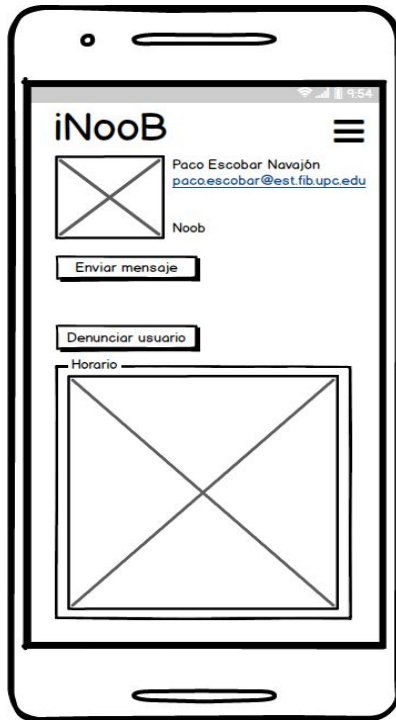


Versión navegador: Ventana de gestión de reuniones concertadas desde la perspectiva del mentor. Muestra una lista con todas las reuniones concertadas y da la opción de obtener más información, modificarlas e incluso eliminarlas. Adicionalmente, contiene un calendario con una lista de fechas de próximas reuniones.



Versión navegador: Ventana de creación de nuevas reuniones desde la perspectiva del mentor. Muestra el menú de creación de nuevas reuniones. Deja escribir un nombre y descripción de la reunión, elegir una fecha y noobs a los que

convocar. Al igual que la vista anterior, muestra un calendario y una lista de fechas de próximas reuniones.



Versión smartphone: Ventana de información de usuario. Da información básica del usuario, como su nombre, fotografía (la utilizada en el campus virtual de la universidad), correo electrónico y horario de clases. También da la posibilidad de enviar un mensaje privado o denunciar al usuario.

4. RECURSOS

[1] “Información de casos de uso”

https://es.wikipedia.org/wiki/Caso_de_uso

[2] “Especificación de los requisitos no funcionales, Volere”

<http://www.volere.co.uk/template.htm>

<http://www11.informatik.uni-erlangen.de/Lehre/SS2015/PR-SWE/Material/volere-template.pdf>

[3] “Identificación de requisitos”

<https://sites.google.com/site/metodologiareq/capitulo-ii/tecnicas-para-identificar-requisitos-funcionales-y-no-funcionales>

[4] “Mock-ups”

<https://support.balsamiq.com/resources/wireframestogo/>