

Documento General de Requisitos

Grupo 6 - MyReminder

A 5 de mayo del 2021

1. HISTORIAL DE REVISIONES

Versión 1.0 del documento general de requisitos, fechado a día 24 de abril de 2021.

- Añadidos los pilares funcionales del producto, sus requisitos y sus descripciones.

Versión 2.0 del documento general de requisitos, fechado a día 5 de mayo de 2021.

- Incorporados los diagramas de requisitos y cerrados los requisitos principales.

2. INTRODUCCIÓN

2.1. Propósito del proyecto

El siguiente Documento General de Requisitos (DGR) recoge por escrito las necesidades que dan lugar a la iniciación del proyecto y que definirán el desarrollo del mismo. De esta forma, quedará totalmente documentado qué debe hacer la aplicación y cómo lo debe hacer, siendo tal un reflejo del acuerdo establecido entre el cliente y nuestro equipo.

2.2. Alcance y metas del proyecto

Nuestro producto software tiene como objetivo desarrollar una aplicación **segura, accesible y útil** para las grandes corporaciones que necesiten una plataforma común para sus empleados.

Con ella, se pretende que los trabajadores puedan tener dicha aplicación descargada y utilizarla para organizar sus citas y eventos. Así pues, la **organización** de eventos como reuniones, citas o presentaciones se verá **facilitada** en gran medida.

Además, no sólo se podrán crear citas de modo local en cada dispositivo, sino que estos estarán **sincronizados** de manera que, si un usuario crea un evento y desea invitar a otro, a este le llegará una notificación y aparecerá en el calendario de su aplicación.

En estos tiempos en los que actualmente todo se debe realizar a través de citas planificadas debido al COVID-19, es de gran ayuda para cualquier empresa tener una aplicación que las gestione y las muestre recogidas en un mismo lugar.

2.3. Visión general

MyReminder es una aplicación de gestión de citas y eventos orientada a entornos empresariales que trata de agrupar en un mismo portal toda la información relativa a la agenda corporativa.

A grandes rasgos, nuestra aplicación contará con un servidor que se encargará de sincronizar los distintos clientes, es decir, las aplicaciones de los distintos usuarios. Estos clientes, contarán con una interfaz gráfica que recogerá las interacciones con el usuario, pudiendo gestionar su agenda personal desde ella y recibiendo todas las notificaciones necesarias.

3. DESCRIPCIÓN GENERAL DEL PRODUCTO

3.1. Funciones del producto

Nuestra aplicación software hace uso de la arquitectura **cliente-servidor**, lo que permite una sincronización automática entre distintos usuarios haciéndola mucho más potente y práctica.

A grandes rasgos, nuestro producto debe poder realizar tres **acciones básicas**:

- Registrar nuevos usuarios en la plataforma.
- Crear nuevos eventos en un calendario / listado.
- Asignar usuarios (invitados) a eventos creados por otro usuario (host).

Las cuales se verán complementadas por otras tantas que añadan **nuevas funcionalidades**:

- Enviar una notificación automáticamente a dichos usuarios invitados.
- Notificar al usuario cuando se acerque alguna fecha marcada en el calendario.
- Mostrar información detallada de las citas y eventos creados.
- Capacidad de editar eventos ya creados o eliminarlos.

3.2. Suposiciones y dependencias

Los requisitos de un proyecto no son estáticos, por lo que estos pueden ir cambiando conforme se vaya desarrollando el producto. Por ello, es necesario tener en cuenta aquellos factores que pueden hacer que los requisitos se vean modificados.

El principal factor que debemos tener en cuenta es el **cliente**. Dado que nuestro objetivo es crear una aplicación que satisfaga sus necesidades, no podemos perder de vista su opinión al respecto.

Además, dado que la metodología empleada para el desarrollo del producto será la **Scrum**, habrá una continua comunicación con el cliente en la que se irá viendo como la aplicación cobra forma generando, en muchas ocasiones, un interés por introducir funcionalidades o mejoras que en un principio no se definieron.

En principio la aplicación está diseñada para un **número medio de usuarios**. Sin embargo, y más en los tiempos que nos ha tocado vivir, es posible que se produzca un notable aumento de usuarios que usen nuestra aplicación en la empresa. Esto implicaría aumentar la capacidad de la aplicación, por ejemplo, usando un servidor más potente o duplicándolo.

Tampoco debemos obviar la aparición de posibles **competidores** que ofrezcan en el mercado productos similares al nuestro. Esto puede dar lugar a posibles mejoras en la aplicación que no se tuvieron en cuenta inicialmente, tanto sugeridas por el cliente como por nosotros mismos, con el fin de conseguir un producto que destaque.

3.3. Restricciones

En cuanto a las restricciones debemos tener en cuenta dos factores a lo largo del desarrollo del proyecto: el **tiempo** empleado y el **cumplimiento de las normativas legales**.

La fecha de entrega de la aplicación es fija e inamovible, es decir, los tiempos deben estar perfectamente definidos de manera que nos aseguremos de que el producto está listo en la fecha indicada. Por ello es conveniente dejar un margen prudencial que nos proteja ante cualquier retraso que se pueda ocasionar en alguno de los procesos.

Además, debemos tener en cuenta que nuestra aplicación funcionará a través de internet y almacenará datos de los usuarios, por lo que es necesario cumplir con el **Reglamento General de Protección de Datos**, que puede hacer que algunas funcionalidades se vean limitadas.

3.4. Requisitos futuros

Dado que lo principal es que la aplicación cumpla con la funcionalidad que el cliente le exige y que ha sido comentada en puntos anteriores, en este apartado se incluirán las mejoras que el usuario final no verá, pero que hará que consigamos un producto mucho más robusto y seguro.

Dado que tratamos con datos personales, un requisito futuro que se llevará a cabo si los tiempos lo permiten será el de conseguir una **comunicación cifrada** mediante el uso de claves públicas y privadas para intercambiar la clave simétrica final.

También es importante de cara a la robustez, almacenar toda la información cifrada en una base de datos que permita que esta permanezca intacta, aunque fallen otros servicios.

4. REQUISITOS ESPECÍFICOS

Requisitos Funcionales (RF)

1. RF. Inicio de sesión en la plataforma

- **1.1 RF. Registro seguro y completo en la plataforma**

La primera vez que un nuevo usuario se registre en MyReminder deberá proveer una serie de datos que incluirán su **email**, **dni**, **nombre de usuario** y **contraseña** que serán cifradas en la base de datos. Posteriormente se le habilitará el acceso a la plataforma.

- **1.2 RF. Autenticación en el inicio de sesión en la aplicación**

Cualquier usuario **registrado** deberá proporcionar o bien su email o bien su nombre de usuario al igual que su contraseña de su cuenta para poder acceder a su agenda.

- **1.3 RF. Cambio de contraseña, email y nombre de usuario en caso de pérdida**

Se le ofrece la opción al usuario ya registrado en la plataforma de **cambiar sus credenciales actuales** si así lo desea. Este proceso se realiza en la misma aplicación.

2. RF. Gestión de los eventos

- **2.1 RF. Creación de eventos nuevos**

Mediante la interfaz gráfica el usuario podrá determinar la **fecha**, **localización** y **asistentes** de un nuevo evento. Esto será completo cuando el invitado acepte la invitación al evento desde dentro de la propia aplicación.

- **2.2 RF. Visualización completa del evento**

El usuario puede seleccionar un evento para verlo en **profundidad**, esto es pudiendo acceder a sus características descritas con anterioridad.

- **2.3 RF. Invitación a otros usuarios a eventos propios**

Todo evento creado o en proceso de serlo debe ser proveído de una serie de **asistentes** que pueden ser agregados desde nuestro panel de contactos o desde el propio evento.

- **2.4 RF. Modificación o eliminación de eventos ya existentes**

Si un evento ha sido creado a priori, entonces el usuario tendrá una opción de **eliminar** dicho evento si así lo desea, en caso contrario lo podrá modificar a gusto del usuario

3 RF. Herramientas para gestionar o notificar usuarios

- **3.1 RF. Lista de contactos gestionada por el usuario**

El usuario tendrá la opción de gestionar su propia lista de contactos, esto quiere decir que en dicha lista podrá **administrar** los usuarios registrados con los que tenga amistad.

- **3.2 RF. Bandeja de entrada receptora de mensajes**

Aquellos mensajes de **invitación** o **recordatorios** de eventos que aún no han sido leídos se almacenarán en una bandeja de entrada para que el usuario pueda gestionarlos.

- **3.3 RF. Recordatorios y notificaciones personalizables**

El usuario tendrá una pestaña para recordar los eventos que se irán actualizando. También dispondrá de notificaciones personalizadas, con esto el usuario podrá **elegir** su propia preferencia, evadir eventos innecesarios y recordar los más importantes.

4 RF. Interfaz fácil y sencilla para facilitar la gestión

La fácil manipulación de los eventos y mensajes en la aplicación es vital para que nuestra aplicación resalte entre el resto. Para ello, diseñaremos una interfaz **intuitiva** que facilite su uso al usuario.

5 RF. Aplicación completamente funcional para ordenador

Diseñaremos un **logotipo** que además de crear una imagen para nuestro producto será el identificador de nuestra aplicación. Gracias a que está siendo implementada en JAVA puede ser utilizada en cualquier sistema operativo.

Requisitos no Funcionales (RNF)

6 RNF. Inicio de la aplicación

- **6.1 RNF. Datos personales durante el registro**

El usuario deberá introducir un **nombre de usuario** único, su dirección **email** de la empresa, su **dni** personal y una **contraseña** para poder registrarse en la aplicación.

- **6.2 RNF. Autenticación mediante usuario y contraseña.**

La **autenticación** se llevará a cabo mediante el **usuario** y **contraseña** definido anteriormente. Este procedimiento se repetirá cada vez que se abra la aplicación.

7 RNF. Aplicación multiplataforma

Con el fin de conseguir una aplicación **multiplataforma** que se pueda ejecutar en cualquier equipo, se usará el lenguaje **Java** como base de la implementación.

8 RNF. Organización de la interfaz.

Al autenticarte, la interfaz gráfica mostrará un **calendario** como vista principal. Accediendo a cada una de las fechas la vista cambiará y obtendrá más detalles sobre los eventos de dichas fechas. También es posible visualizarla como una **lista de eventos**.

9 RNF. Aplicación compatible con modo oscuro

Como suele venir siendo común en la mayoría de aplicaciones, resulta **agradable a la vista** poder cambiar la visualización del panel de la aplicación a un color más oscuro.

10 RNF. Crear un calendario y mostrar sus propiedades

- **10.1 RNF. Creación de citas en el calendario.**

Para crear una **cita** o **evento**, es necesario acceder al calendario en el día deseado y pulsar la opción de la ventana que permite **añadir citas**.

- **10.2 RNF. Invitación a otros usuarios desde el calendario.**

Para ello se puede hacer a la hora de crear la cita o bien posteriormente, seleccionando la opción que permite invitar a otros usuarios desde el propio evento.

11 RNF. Notificaciones automáticas

La aplicación comprobará en el servidor si el usuario al que se desea invitar existe y, si así ocurre, se le envía un **mensaje automático** a su aplicación generando una **alerta** en su interfaz gráfica informándole de la invitación.

12 RNF. Agenda de contactos del usuario

Se creará una opción que permita al usuario **agregar a otros usuarios** a su agenda de contactos de manera que no tenga que recordar el nombre de usuario de aquellos que haya guardado.

13 RNF. Recordatorio mediante alarma

Se creará una **alarma** que, una hora antes de que ocurra un evento, avise al usuario mediante una **notificación emergente** que será recogida en la bandeja de entrada.

14 RNF. Información de las citas en la nueva ventana

Para mostrar la información de una cita, se debe pulsar el botón correspondiente que desplegará automáticamente una nueva ventana con la cita o el evento descrito más detalladamente.

15 RNF. Sincronización de los usuarios

- 15.1 RNF. Arquitectura cliente-servidor

Para conseguir que los calendarios de los distintos usuarios se sincronicen automáticamente, se hará uso del modelo cliente-servidor de manera que el servidor sirva de pasarela entre los distintos clientes.

16 RNF. Protección de datos personales

- 16.1 RNF Privacidad de la información de los usuarios

La información privada de nuestros usuarios será almacenada tras un cifrado en una base de datos **NoSQL MongoDB** de manera que no se vea afectada si otros servicios fallan.

- 16.2 RNF. Seguridad de la información

Se cifrarán las **credenciales**, las **comunicaciones** y **eventos**, así como el almacenamiento en la base de datos, para evitar el robo de información e incrementar la privacidad.

5. DIAGRAMAS (APÉNDICES ADICIONALES)

Modelo generado en MagicDraw

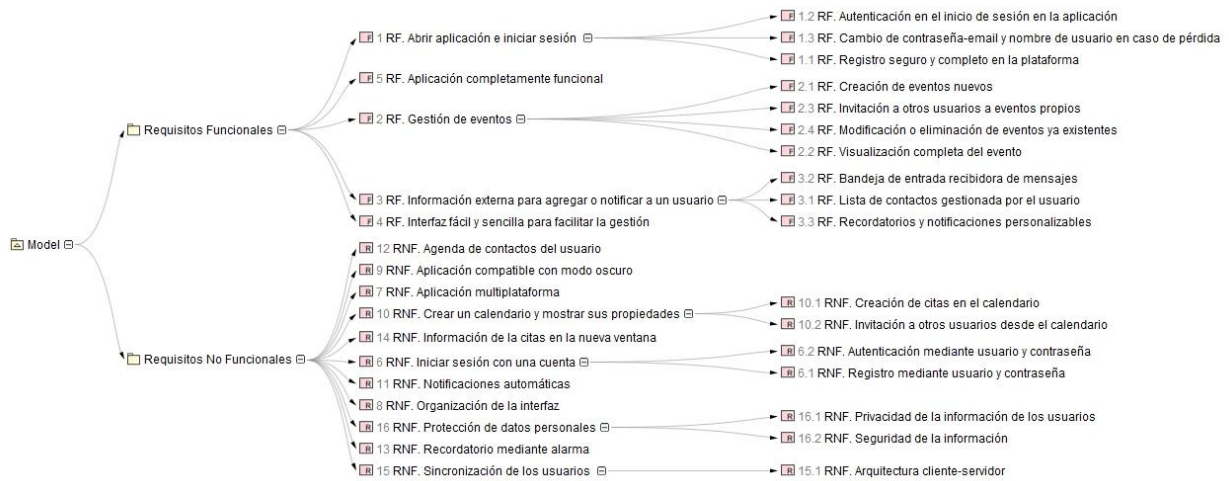


Figura 1. Modelo