

INSTRADROID: ANÁLISIS Y ESPECIFICACIÓN DE REQUISITOS

1. Requisitos funcionales

Un requisito funcional¹ define una función del sistema de software o sus componentes. Una función es descrita como un conjunto de entradas, comportamientos y salidas. Se trata de funcionalidades específicas que un sistema debe cumplir. Estos requisitos establecen los comportamientos que tiene el software que vamos a desarrollar.

En el caso de la aplicación Intradoid, los requisitos funcionales serían los siguientes:

RF1 → El sistema debe permitir al usuario ver una línea de tiempo con las fotografías de los usuarios a los que sigue.

RF2 → El usuario debe poder “dar me gusta” a la fotografía, así como efectuar un comentario sobre la misma.

RF3 → El usuario debe poder ver en tiempo real los comentarios y me gustas que otros usuarios han realizado sobre la publicación.

RF4 → El usuario debe poder ver en un mapa la localización en la que fue subida la fotografía.

RF5 → El usuario debe poder subir una foto a Instadroid, echándola con la cámara o subiéndola desde la galería de su teléfono.

RF6 → El usuario debe poder editar la fotografía antes de subirla, pudiendo aplicar distintos filtros o efectos.

RF7 → El sistema debe permitir al usuario ver una sección con la información de su perfil.

RF8 → El usuario debe poder cambiar su información del perfil, como su nombre, fotografía de perfil y contraseña.

RF9 → El usuario debe poder mostrar su código QR único de usuario.

RF10 → El usuario debe poder agregar a otros usuarios a su lista de amigos a través del escaneo del código QR único de usuario de los demás usuarios.

RF11 → El sistema debe contar con una sección de búsqueda.

RF12 → El usuario debe poder buscar publicaciones por hashtag o usuarios por nombre de usuario.

RF13 → El usuario debe poder agregar a otro usuario mediante su búsqueda y seguimiento.

¹ https://es.wikipedia.org/wiki/Requisito_funcional

- RF14 → El usuario debería poder tener una conversación privada con otros usuarios.
- RF15 → El usuario debe poder identificarse con sus credenciales de otras redes sociales o servicios.
- RF16 → El usuario debe poder recibir notificaciones desde la aplicación.
- RF17 → El usuario debe poder aceptar las peticiones de amistad que se le envían.
- RF18 → El usuario debe poder usar la aplicación en cualquier momento y lugar.
- RF19 → El sistema debe controlar que las peticiones son todas seguras.
- RF20 → El sistema debe facilitar un sistema de login.
- RF21 → El sistema debe facilitar un sistema de registro para los usuarios que no quieran identificarse con los medios expuestos anteriormente.
- RF22 → El usuario debe poder añadir un título y hashtags a su publicación.
- RF23 → La aplicación cliente del sistema debe seguir los cánones de diseño marcados por cada plataforma.
- RF24 → La aplicación debe funcionar en cualquier dispositivo móvil del mercado, sea cuál sea su sistema operativo.
- RF25 → La experiencia del usuario debe de ser perfecta y debe sentirse cómodo usando la aplicación.
- RF26 → El sistema debe de ser escalable, por lo que se debe de actualizar y mejorar constantemente mientras dure el ciclo de vida de la aplicación.
- RF27 → La navegación en la aplicación cliente debe incluir una pantalla de publicaciones, una pantalla de perfil, una pantalla en la que se pueda subir una publicación y, por último, una pantalla para realizar búsquedas de usuarios y publicaciones.
- RF28 → El usuario debe poder editar (cambiar el título) a una publicación subida.
- RF29 → El usuario debe poder borrar una publicación subida.

2. Requisitos no funcionales

Un requisito no funcional² es aquel que impone restricciones en el diseño o la implementación como restricciones en el diseño o estándares de calidad. Son propiedades o cualidades que el producto debe tener.

En el caso de Intradoid, los requisitos no funcionales son los siguientes:

² https://www.ecured.cu/Requisitos_no_funcionales

RNF1 → El cliente móvil del sistema debe ser desarrollado con algún lenguaje o framework que le permita ser una aplicación multiplataforma, tales como React-Native, Native Script, Electron o similares.

RNF2 → Todos los datos que maneja la aplicación deberán almacenarse remotamente en un servidor.

RNF3 → El consumo de estos datos se hará a través de una API REST propia desarrollada con SpringBoot o similares. También se puede usar Firebase para esta parte de backend.

RNF4 → El acceso al consumo de estos datos del servidor debe estar securizado bien con JWT si se opta por una API REST propia o mediante el sistema de autenticación de Firebase si se opta por esta última opción.

RNF5 → El almacenamiento de datos se realizará en una base de datos que podrá ser SQL (Postgres), NoSQL (MongoDB o Firebase) o ambas.

RNF6 → El acceso a estos datos y su publicación en las vistas debe de ser lo más rápido posible.

RNF7 → En el caso de optar por una API REST propia, la misma debe de ser completamente testeable sin necesidad de una aplicación cliente, con soluciones como POSTMAN o similares.

RNF8 → El servicio web debe estar desplegado pudiendo utilizar Heroku, Docker, AWS o similar con dicho fin.

RNF9 → La gama de colores de la aplicación especificada por el cliente debe contener azul, blanco y verdes agua sin saturación.

RNF10 → Los métodos de autenticación del usuario son su email y contraseña, número de teléfono, Google, Facebook y Twitter.

3. Requisitos de información

Los requisitos de información son aquellos que representan entidades e información relevante con las que el producto software va a operar³.

En el caso de Instadroid, los requisitos de información con los que vamos a trabajar son los siguientes:

RI1 → *Usuario*: Representa a cada una de las personas que utilizan la aplicación. Como información importante del mismo encontramos:

- Id de usuario

³ <https://www.infor.uva.es/~mlaguna/is1/apuntes/2-requisitos.pdf>

- Nombre
- Apellidos
- Fecha de nacimiento
- Nick
- Token de autenticación
- Email

RI2 → *Publicación*: Representa las fotografías que suben los usuarios a Instadroid, junto con sus comentarios y me gustas. De las mismas, vamos a trabajar con estos datos:

- Id de publicación
- Fotografía
- Comentarios (se relaciona con el RI5)
- Número de me gustas
- Posición
- Datos del usuario que la ha subido

RI3 → *Mensaje*: Representa los mensajes que se mandan los usuarios en sus conversaciones privadas. De los mismos la información que vamos a tratar es:

- Id de mensaje
- Emisor (se relaciona con el RI1)
- Destinatario (se relaciona con el RI1)

RI4 → *Amigo*: Representa las relaciones de amistad entre usuarios, de tal forma que los datos a tratar son:

- Id usuario 1 (se relaciona con el RI1)
- Id usuario 2 (se relaciona con el RI1)

RI5 → *Comentario*: Representa un comentario que hace un usuario en una publicación, por lo que los datos a tratar en este caso son:

- Id publicación (se relaciona con el RI2)
- Contenido