Contenido

[1. Introducción 2](#_Toc23250915)

[1.1. Propósito 2](#_Toc23250916)

[1.2. Ámbito del Sistema 2](#_Toc23250917)

[1.3. Definiciones, Acrónimos y Abreviaturas 2](#_Toc23250918)

[1.4. Referencias 2](#_Toc23250919)

[1.5. Visión General del Documento 3](#_Toc23250920)

[2. Descripción General 3](#_Toc23250921)

[2.1. Perspectiva del Producto 3](#_Toc23250922)

[2.2. Funciones del Producto 3](#_Toc23250923)

[2.3. Características de los Usuarios 4](#_Toc23250924)

[2.4. Restricciones 4](#_Toc23250925)

[2.5. Suposiciones y Dependencias 5](#_Toc23250926)

[2.6. Requisitos Futuros 6](#_Toc23250927)

[3. Requisitos Espec´ıficos 6](#_Toc23250928)

[3.1. Interfaces Externas 8](#_Toc23250929)

[3.2. Funciones 8](#_Toc23250930)

[3.3. Requisitos de Rendimiento 10](#_Toc23250931)

[3.4. Restricciones de Disen˜o 10](#_Toc23250932)

[3.5. Atributos del Sistema 10](#_Toc23250933)

[4. Ap´endices 10](#_Toc23250934)

# Introducción

En esta seccio´n se proporcionara´ una introducci´on a todo el documento de Especificacio´n de Requisitos Software(ERS). Consta de varias subsecciones: propo´sito, ambito del sistema, definiciones, referencias y visio´n general del documento.

## Propósito

Este documento se ha creado con el fin de dar conocer las decisiones que se han tomado para los requisitos del sistema HolApp, asi como dar a conocer los documentos a los cuales se hacen referencia aquí. Por último, esta especificación de requerimientos de software va dirigida al profesor Carlos Torres, quien es el encargado de la experiencia educativa “Desarrollo de sistemas de software”.

## Ámbito del Sistema

El nuevo sistema que se desarrollará lleva el nombre holApp, ya que es un sistema móvil de mensajería instantánea y su nombre hace alusión a un típico saludo que usamos cuando encontramos a un amigo, familiar, conocido, etcétera.

## Definiciones, Acrónimos y Abreviaturas

A continuación, se hará una lista de definiciones con el propósito de que quien lea este documento pueda comprender lo redactado aquí:

## Referencias

En esta subseccio´n se mostrar´a una lista completa de todos los documentos referenciados en la ERS.

## Visión General del Documento

Esta subseccio´n describe brevemente los contenidos y la organizaci´on del resto de la ERS.

# Descripción General

Este producto de software esta diseñado para ser un sistema distribuido, el cual contará con una laptop utilizada como servidor y será desarrollada en java para el lado del cliente y desplegado en teléfonos móviles únicamente con sistema operativo Android.

## Perspectiva del Producto

El sistema holApp, asi como todos sus componentes, trabajarán de forma totalmente independiente de cualquier otro sistema de software. Por lo tanto, no se requiere nombrar o vincular nuestro sistema con alguno en existencia.

## Funciones del Producto

Este sistema permitirá el uso de una cuenta de usuario, con el propósito de mantener la seguridad de su información, la cual se manejará dentro de la aplicación. Asimismo, holApp tendrá un inicio de sesión para la autenticación del usuario.

Por otro lado, se podrán gestionar fotografías dentro de la cuenta, así como reaccionar a estas, comentarlas y marcarlas como favoritas para poder visualizarlas posteriormente.

El cliente tambien tendrá acceso a un perfil dentro de su cuenta, donde podrá compartir una fotografía de perfil, un estado y su nombre de usuario. Será capaz de cambiar la fotografía de perfil y el estado.

Habrá una cuenta de moderador, el cual podrá revisar el contenido que suben los usuarios, permitiendo de esta forma eliminar el contenido o dar de baja definitiva una cuenta.

## Características de los Usuarios

Habrá dos tipos de usuario en el sistema holApp, el cliente y el moderador. A continuación, se describen sus características:

|  |  |
| --- | --- |
| **Cliente** | Se espera que este usuario tenga una edad máxima de 30 años, para que pueda hacer uso del sistema con fluidez. De lo contrario, se necesita que haya estado en contacto con aplicaciones en un teléfono móvil para que pueda comprender el uso de la aplicación. |
| **Moderador** | Este usuario debe tener conocimientos básicos en el uso de sistemas informáticos y su edad debe ser mayor a 18 años, ya que tendrá acceso a las cuentas de los clientes. |

## Restricciones

El sistema se deberá desarrollar en Java para el lado del cliente y se debe se puede elegir el uso de cualquier otro lenguaje para el servidor, pero está prohibido usar Java.

La aplicación para clientes debe ser móvil y se debe usar un API Rest para la autenticación de usuarios a través de tokens en el servidor para mayor confiabilidad y este permitirá el almacenamiento de fotos y chats.

## Suposiciones y Dependencias

Esta subsecci´on de la ERS describir´a aquellos factores que, si cambian, pueden afectar a los requisitos. Por ejemplo, los requisitos pueden presuponer una cierta organizacio´n de ciertas unidades de la empresa, o pueden presuponer que el sistema correr´a sobre cierto sistema operativo. Si cambian dichos detalles en la organizacio´n de la empresa, o si cambian ciertos detalles t´ecnicos, como el sistema operativo, puede ser necesario revisar y cambiar los requisitos.

## Requisitos Futuros

Esta subsecci´on esbozara´ futuras mejoras al sistema, que podra´n analizarse e implementarse en un futuro.

# Requisitos Espec´ıficos

Esta seccio´n contiene los requisitos a un nivel de detalle suficiente como para permitir a los disen˜adores disen˜ar un sistema que satisfaga estos requisitos, y que permita al equipo de pruebas planificar y realizar las pruebas que demuestren si el sistema satisface, o no, los requisitos. Todo requisito aqu´ı especificado describir´a comportamientos externos del sistema, perceptibles por parte de los usuarios, operadores y otros sistemas. Esta es la seccio´n ma´s larga e importante de la ERS. Debera´n aplicarse los siguientes principios:

El documento deber´ıa ser perfectamente legible por personas de muy distintas formaciones e intereses.

Deber´an referenciarse aquellos documentos relevantes que poseen alguna influencia sobre los requisitos.

Todo requisito deber´a ser un´ıvocamente identificable mediante algu´n co´digo o sistema de numeraci´on adecuado.

Lo ideal, aunque en la pra´ctica no siempre realizable, es que los requisitos posean las siguientes caracter´ısticas:

* Correccion: La ERS es correcta si y s´olo si todo requisito que figura aqu´ı (y que ser´a implementado en el sistema) refleja alguna necesidad real. La correccio´n de la ERS implica que el sistema implementado sera´ el sistema deseado.
* No ambiguos: Cada requisito tiene una sola interpretacio´n. Para eliminar la ambigu¨edad inherente a los requisitos expresados en lenguaje natural, se deber´an utilizar gra´ficos o notaciones formales. En el caso de utilizar t´erminos que, habitualmente, poseen ma´s de una interpretaci´on, se definir´an con precisi´on en el glosario.
* Completos: Todos los requisitos relevantes han sido incluidos en la ERS. Conviene incluir todas las posibles respuestas del sistema a los datos de entrada, tanto v´alidos como no va´lidos.
* Consistentes: Los requisitos no pueden ser contradictorios. Un conjunto de requisitos contradictorio no es implementable.
* Clasificados: Normalmente, no todos los requisitos son igual de importantes. Los requisitos pueden clasificarse por importancia (esenciales, condicionales u opcionales) o por estabilidad (cambios que se espera que afecten al requisito). Esto sirve, ante todo, para no emplear excesivos recursos en implementar requisitos no esenciales.
* Verificables: La ERS es verificable si y so´lo si todos sus requisitos son verificables. Un requisito es verificable (testeable) si existe un proceso finito y no costoso para demostrar que el sistema cumple con el requisito. Un requisito ambiguo no es, en general, verificable. Expresiones como a veces, bien, adecuado, etc. introducen ambigu¨edad en los requisitos. Requisitos como “en caso de accidente la nube t´oxica no se extendera´ ma´s all´a de 25Km” no es verificable por el alto costo que conlleva.
* Modificables: La ERS es modificable si y s´olo si se encuentra estructurada de forma que los cambios a los requisitos pueden realizarse de forma fa´cil, completa y consistente. La utilizacio´n de herramientas autom´aticas de gesti´on de requisitos (por ejemplo RequisitePro o Doors) facilitan enormemente esta tarea.
* Trazables: La ERS es trazable si se conoce el origen de cada requisito y se facilita la referencia de cada requisito a los componentes del disen˜o y de la implementacio´n. La trazabilidad hacia atra´s indica el origen (documento, persona, etc.) de cada requisito. La trazabilidad hacia delante de un requisito R indica qu´e componentes del sistema son los que realizan el requisito R.

## Interfaces Externas

Se describir´an los requisitos que afecten a la interfaz de usuario, interfaz con otros sistemas (hardware y software) e interfaces de comunicaciones.

## Funciones

Esta subsecci´on (quiza´ la ma´s larga del documento) debera´ especificar todas aquellas acciones (funciones) que debera´ llevar a cabo el software. Normalmente (aunque no siempre), son aquellas acciones expresables como “el sistema deber´a ...”. Si se considera necesario, podr´an utilizarse notaciones gra´ficas y tablas, pero siempre supeditadas al lenguaje natural, y no al rev´es.

Es importante tener en cuenta que, en 1983, el Est´andar de IEEE 830 establec´ıa que las funciones deber´ıan expresarse como una jerarqu´ıa funcional (en paralelo con los DFDs propuestos por el an´alisis estructurado). Pero el Esta´ndar de IEEE 830, en sus u´ltimas versiones, ya permite organizar esta subseccio´n de mu´ltiples formas, y sugiere, entre otras, las siguientes:

Por tipos de usuario: Distintos usuarios poseen distintos requisitos. Para cada clase de usuario que exista en la organizaci´on, se especificar´an los requisitos funcionales que le afecten o tengan mayor relacio´n con sus tareas.

Por objetos: Los objetos son entidades del mundo real que sera´n reflejadas en el sistema. Para cada objeto, se detallar´an sus atributos y sus funciones. Los objetos pueden agruparse en clases. Esta organizacio´n de la ERS no quiere decir que el disen˜o del sistema siga el paradigma de Orientacio´n a Objetos.

Por objetivos: Un objetivo es un servicio que se desea que ofrezca el sistema y que requiere una determinada entrada para obtener su resultado. Para cada objetivo o subobjetivo que se persiga con el sistema, se detallar´an las funciones que permitan llevarlo a cabo.

Por est´ımulos: Se especificara´n los posibles est´ımulos que recibe el sistema y las funciones relacionadas con dicho est´ımulo.

Por jerarqu´ıa funcional: Si ninguna de las anteriores alternativas resulta de ayuda, la funcionalidad del sistema se especificar´a como una jerarqu´ıa de funciones que comparten entradas, salidas o datos internos. Se detallara´n las funciones (entrada, proceso, salida) y las subfunciones del sistema. Esto no implica que el disen˜o del sistema deba realizarse segu´n el paradigma de Disen˜o Estructurado.

Para organizar esta subseccio´n de la ERS se elegir´a alguna de las anteriores alternativas, o incluso alguna otra que se considere ma´s conveniente.

Deber´a, eso s´ı, justificarse el porqu´e de tal eleccio´n.

*4 APENDICES´*

## Requisitos de Rendimiento

Se detallar´an los requisitos relacionados con la carga que se espera tenga que soportar el sistema. Por ejemplo, el nu´mero de terminales, el nu´mero esperado de usuarios simultaneamente conectados, nu´mero de transacciones por segundo que deber´a soportar el sistema, etc.

Tambi´en, si es necesario, se especificara´n lo requisitos de datos, es decir, aquellos requisitos que afecten a la informacio´n que se guardara´ en la base de datos. Por ejemplo, la frecuencia de uso, las capacidades de acceso y la cantidad de registros que se espera almacenar (decenas, cientos, miles o millones).

## Restricciones de Disen˜o

Todo aquello que restrinja las decisiones relativas al disen˜o de la aplicacio´n: Restricciones de otros esta´ndares, limitaciones del hardware, etc.

## Atributos del Sistema

Se detallar´an los atributos de calidad (las “ilities”) del sistema: Fiabilidad, mantenibilidad, portabilidad, y, muy importante, la seguridad. Debera´ especificarse qu´e tipos de usuario est´an autorizados, o no, a realizar ciertas tareas, y c´omo se implementara´n los mecanismos de seguridad (por ejemplo, por medio de un *login* y una *password*).

**3.6. Otros Requisitos**

Cualquier otro requisito que no encaje en otra seccio´n.

# Ap´endices

Pueden contener todo tipo de informacio´n relevante para la ERS pero que, propiamente, no forme parte de la ERS. Por ejemplo:

1. Formatos de entrada/salida de datos, por pantalla o en listados.
2. Resultados de an´alisis de costes.
3. Restricciones acerca del lenguaje de programaci´on.