# Especificación de requerimientos de software

# para

# Aplicación para Android de la FCEFyN

Version 1.0

Preparada por Jeremías Benítez

cmd

02/05/2017

Tabla de contenidos

Tabla de contenidos [ii](#__RefHeading___Toc20732946)

Historia de revisiones [iii](#__RefHeading___Toc20732947)

1. Introducción [1](#__RefHeading___Toc20732948)

1.1 Propósito [1](#__RefHeading___Toc20732949)

1.2 Audiencia objetivo y sugerencias de lectura [1](#__RefHeading___Toc20732951)

1.3 Scope del producto [1](#__RefHeading___Toc20732952)

1.4 Referencias [1](#__RefHeading___Toc20732953)

2. Overall Description [2](#__RefHeading___Toc20732954)

2.1 Product Perspective [2](#__RefHeading___Toc20732955)

2.2 Product Functions [2](#__RefHeading___Toc20732956)

2.3 User Classes and Characteristics [2](#__RefHeading___Toc20732957)

2.4 Operating Environment [2](#__RefHeading___Toc20732958)

2.5 Design and Implementation Constraints [2](#__RefHeading___Toc20732959)

2.6 User Documentation [2](#__RefHeading___Toc20732960)

2.7 Assumptions and Dependencies [3](#__RefHeading___Toc20732961)

3. External Interface Requirements [3](#__RefHeading___Toc20732962)

3.1 User Interfaces [3](#__RefHeading___Toc20732963)

3.2 Hardware Interfaces [3](#__RefHeading___Toc20732964)

3.3 Software Interfaces [3](#__RefHeading___Toc20732965)

3.4 Communications Interfaces [3](#__RefHeading___Toc20732966)

4. Domain Model [4](#__RefHeading___Toc20732967)

5. System Features (Use Cases) [4](#__RefHeading___Toc20732968)

5.1 Use Case 1 [4](#__RefHeading___Toc20732969)

5.1.1 Name: [4](#__RefHeading___Toc20732970)

5.1.2 Goal: [4](#__RefHeading___Toc20732971)

5.1.3 Input: [4](#__RefHeading___Toc20732972)

5.1.4 Output: [4](#__RefHeading___Toc20732973)

5.1.5 Main Scenario: [4](#__RefHeading___Toc20732974)

5.1.6 Pre-condition: [4](#__RefHeading___Toc20732975)

5.1.7 Steps: [4](#__RefHeading___Toc20732976)

5.1.8 Post-condition [4](#__RefHeading___Toc20732977)

5.1.9 Exceptional Scenario 1 [4](#__RefHeading___Toc20732978)

5.1.10 Example [4](#__RefHeading___Toc20732979)

5.2 Use Case 2 (and so on) [5](#__RefHeading___Toc20732980)

6. Other Nonfunctional Requirements [5](#__RefHeading___Toc20732981)

6.1 Performance Requirements [5](#__RefHeading___Toc20732982)

6.2 Safety Requirements [5](#__RefHeading___Toc20732983)

6.3 Security Requirements [5](#__RefHeading___Toc20732984)

6.4 Software Quality Attributes [5](#__RefHeading___Toc20732985)

7. Other Requirements [5](#__RefHeading___Toc20732986)

Appendix A: Glossary [6](#__RefHeading___Toc20732987)

Appendix B: Analysis Models [6](#__RefHeading___Toc20732988)

Appendix C: To Be Determined List [6](#__RefHeading___Toc20732989)

Historia de revisiones

|  |  |  |  |
| --- | --- | --- | --- |
| **Nombre** | **Fecha** | **Razón de los cambios** | **Version** |
| **Jeremías Benítez** | **02/05/17** | Creación del documento. | V1.0.0 |
|  |  |  |  |

# Introducción

## Propósito

El presente documento ha sido redactado para especificar en su totalidad los requerimientos de las versiones v1.X de la Aplicación para Android desarrollada para la Facultad de Ciencias Exactas, Físicas y Naturales (en adelante FCEFyN).

## Audiencia objetivo y sugerencias de lectura

La audiencia objetivo de este SRS está compuesta por los siguientes individuos:

* Desarrolladores, quienes deben implementar el software basandosé en estos requerimientos.
* Testers, quienes deben asegurarse que dichos requerimientos sean validables y cuyos tests deben realizar dicha validación.
* Project Managers.
* Personal encargado de la redacción de la documentación.
* Ingenieros de la Calidad, que deben serciorarse que los requerimientos de calidad sean cumplidos.

Este documento comienza ofreciendo una visión de alto nivel de la Aplicación para Android de la FCEFyN. Luego se enumeran las restricciones de software y hardware, como así también las dependencias necesarias y cualquier información que brinde información que facilite la comprensión del documento.

La mayor parte del mismo se ocupa de listar los requerimientos. Cada uno de los mismo es descripto utilizando casos de uso.

Al final se encuentral los requerimientos no funcionales establecidos para este sistema.

## Scope del producto

La aplicación deberá ayudar tanto a alumnos como a profesores a desenvolverse en el edificio de la cede Ciudad Universitaria de la FCEFyN. A ambos les permitirá acceder a un mapa de la institución con las aulas propiamente identificadas. Además, el sistema les indicará, de forma visual, cuál es el camino más corto desde el aula o lugar identificado en el que se encuentre el usuario y el aula o lugar identificado al que desea ir. También permitirá conocer qué profesores tienen reservada el aula, o, si las hay, qué materias se dictan allí y en qué horarios. A los segundos les permitirá, además, reservar ya sean aulas específicas, o aulas que cumplan con ciertos requisitos (por ej. Cantidad máxima de alumnos, horarios hábiles, etc.). La idea es que el sistema le permita a los usuarios un manejo más eficiente del tiempo al permitir las averiguaciones pertinentes al mismo desde el celular del usuario, sin necesidad de estar el mismo en el edificio, como así también evitar incidentes de superposición de reservas de aulas.

El sistema no se funcionará con GPS, por lo que no será posible dar el recorrido desde la posición real del usuario hasta la posición destino.

## Referencias

Templates de documentos de requerimientos:

* <https://drive.google.com/file/d/0B6Lbceno346BVloxWUlfUUFRTG8/view>
* <https://drive.google.com/file/d/0B6Lbceno346BYU1wOXV4Q01WSkk/view>
* <http://www2.latech.edu/~box/ase/srs_template.doc>

Sommerville, I., (2011), *Ingeniería de software*, México, Pearson.

# Overall Description

## Product Perspective

<Describe the context and origin of the product being specified in this SRS. For example, state whether this product is a follow-on member of a product family, a replacement for certain existing systems, or a new, self-contained product. If the SRS defines a component of a larger system, relate the requirements of the larger system to the functionality of this software and identify interfaces between the two. A simple diagram that shows the major components of the overall system, subsystem interconnections, and external interfaces can be helpful.>

## Product Functions

<Summarize the major functions the product must perform or must let the user perform. Details will be provided in Section 4, so only a high level summary (such as a bullet list) is needed here. Organize the functions to make them understandable to any reader of the SRS. A picture of the major groups of related requirements and how they relate, such as a top level data flow diagram or object class diagram, is often effective.>

## User Classes and Characteristics

<Identify the various user classes that you anticipate will use this product. User classes may be differentiated based on frequency of use, subset of product functions used, technical expertise, security or privilege levels, educational level, or experience. Describe the pertinent characteristics of each user class. Certain requirements may pertain only to certain user classes. Distinguish the most important user classes for this product from those who are less important to satisfy.>

## Operating Environment

<Describe the environment in which the software will operate, including the hardware platform, operating system and versions, and any other software components or applications with which it must peacefully coexist.>

## Design and Implementation Constraints

<Describe any items or issues that will limit the options available to the developers. These might include: corporate or regulatory policies; hardware limitations (timing requirements, memory requirements); interfaces to other applications; specific technologies, tools, and databases to be used; parallel operations; language requirements; communications protocols; security considerations; design conventions or programming standards (for example, if the customer’s organization will be responsible for maintaining the delivered software).>

## User Documentation

<List the user documentation components (such as user manuals, on-line help, and tutorials) that will be delivered along with the software. Identify any known user documentation delivery formats or standards.>

## Assumptions and Dependencies

<List any assumed factors (as opposed to known facts) that could affect the requirements stated in the SRS. These could include third-party or commercial components that you plan to use, issues around the development or operating environment, or constraints. The project could be affected if these assumptions are incorrect, are not shared, or change. Also identify any dependencies the project has on external factors, such as software components that you intend to reuse from another project, unless they are already documented elsewhere (for example, in the vision and scope document or the project plan).>

# External Interface Requirements

## User Interfaces

<Describe the logical characteristics of each interface between the software product and the users. This may include sample screen images, any GUI standards or product family style guides that are to be followed, screen layout constraints, standard buttons and functions (e.g., help) that will appear on every screen, keyboard shortcuts, error message display standards, and so on. Define the software components for which a user interface is needed. Details of the user interface design should be documented in a separate user interface specification.>

## Hardware Interfaces

<Describe the logical and physical characteristics of each interface between the software product and the hardware components of the system. This may include the supported device types, the nature of the data and control interactions between the software and the hardware, and communication protocols to be used.>

## Software Interfaces

<Describe the connections between this product and other specific software components (name and version), including databases, operating systems, tools, libraries, and integrated commercial components. Identify the data items or messages coming into the system and going out and describe the purpose of each. Describe the services needed and the nature of communications. Refer to documents that describe detailed application programming interface protocols. Identify data that will be shared across software components. If the data sharing mechanism must be implemented in a specific way (for example, use of a global data area in a multitasking operating system), specify this as an implementation constraint.>

## Communications Interfaces

<Describe the requirements associated with any communications functions required by this product, including e-mail, web browser, network server communications protocols, electronic forms, and so on. Define any pertinent message formatting. Identify any communication standards that will be used, such as FTP or HTTP. Specify any communication security or encryption issues, data transfer rates, and synchronization mechanisms.>

# Domain Model

<Sometimes, this section is optional. However, it may be important to have it since domain model may give more useful as well>

# System Features (Use Cases)

<This template illustrates organizing the functional requirements for the product by system features, the major services provided by the product. You may prefer to organize this section by use case, mode of operation, user class, object class, functional hierarchy, or combinations of these, whatever makes the most logical sense for your product. You should Use-case diagram>

## Use Case 1

<Don’t really say “Use case 1.” State the feature name in just a few words.>

### Name:

### Goal:

### Input:

### Output:

### Main Scenario:

### Pre-condition:

### Steps:

#### Step1:

#### .

#### Step n:

### Post-condition

### Exceptional Scenario 1

### Example

## Use Case 2 (and so on)

# Other Nonfunctional Requirements

## Performance Requirements

<If there are performance requirements for the product under various circumstances, state them here and explain their rationale, to help the developers understand the intent and make suitable design choices. Specify the timing relationships for real time systems. Make such requirements as specific as possible. You may need to state performance requirements for individual functional requirements or features.>

## Safety Requirements

<Specify those requirements that are concerned with possible loss, damage, or harm that could result from the use of the product. Define any safeguards or actions that must be taken, as well as actions that must be prevented. Refer to any external policies or regulations that state safety issues that affect the product’s design or use. Define any safety certifications that must be satisfied.>

## Security Requirements

<Specify any requirements regarding security or privacy issues surrounding use of the product or protection of the data used or created by the product. Define any user identity authentication requirements. Refer to any external policies or regulations containing security issues that affect the product. Define any security or privacy certifications that must be satisfied.>

## Software Quality Attributes

<Specify any additional quality characteristics for the product that will be important to either the customers or the developers. Some to consider are: adaptability, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability. Write these to be specific, quantitative, and verifiable when possible. At the least, clarify the relative preferences for various attributes, such as ease of use over ease of learning.>

# Other Requirements

<Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project.>

Appendix A: Glossary

<Define all the terms necessary to properly interpret the SRS, including acronyms and abbreviations. You may wish to build a separate glossary that spans multiple projects or the entire organization, and just include terms specific to a single project in each SRS.>

Appendix B: Analysis Models

<Optionally, include any pertinent analysis models, such as data flow diagrams, class diagrams, state-transition diagrams, or entity-relationship diagrams.>

Appendix C: To Be Determined List

<Collect a numbered list of the TBD (to be determined) references that remain in the SRS so they can be tracked to closure.>