# Especificación de requerimientos de software

# para

# FCEFyN maps

Version 1.0

Preparada por Jeremías Benítez

cmd

02/05/2017

Contenido

[Historia de revisiones 5](#_Toc482024398)

[1. Introducción 6](#_Toc482024399)

[1.1 Propósito 6](#_Toc482024400)

[1.2 Audiencia Objetivo y Sugerencias de Lectura 6](#_Toc482024401)

[1.3 Scope del Producto 6](#_Toc482024402)

[1.4 Referencias 6](#_Toc482024403)

[2. Descripción General 8](#_Toc482024404)

[2.1 Definición 8](#_Toc482024405)

[2.2 Funciones del Producto 8](#_Toc482024406)

[Diagrama de casos de uso para la aplicación. 9](#_Toc482024407)

[2.3 Clases de Usuario y Características 9](#_Toc482024408)

[2.4 Ambiente de Operación 9](#_Toc482024409)

[2.5 Restricciones de Diseño e Implementación 9](#_Toc482024410)

[2.6 Documentación de Usuario 10](#_Toc482024411)

[2.7 Suposiciones y Dependencias 10](#_Toc482024412)

[3. Listado de Requerimientos Funcionales 11](#_Toc482024413)

[4. Detalle de los Requerimientos 11](#_Toc482024414)

[4.1 REQ 1: Exploración 11](#_Toc482024415)

[4.1.1 Objetivo: 11](#_Toc482024416)

[4.1.2 Input: 11](#_Toc482024417)

[4.1.3 Output: 11](#_Toc482024418)

[4.1.4 Diagrama 11](#_Toc482024419)

[4.2 REQ 2: Recorrido más corto 11](#_Toc482024420)

[4.2.1 Objetivo: 11](#_Toc482024421)

[4.2.2 Input: 12](#_Toc482024422)

[4.2.3 Output: 12](#_Toc482024423)

[4.2.4 Caso Excepcional: 12](#_Toc482024424)

[4.2.5 Diagrama 12](#_Toc482024425)

[4.3 REQ 3: Búsqueda 13](#_Toc482024426)

[4.3.1 Objetivo: 13](#_Toc482024427)

[4.3.2 Input: 13](#_Toc482024428)

[4.3.3 Output: 14](#_Toc482024429)

[4.3.4 Caso Excepcional: 14](#_Toc482024430)

[4.3.5 Diagrama 14](#_Toc482024431)

[4.4 REQ 4: Zoom 14](#_Toc482024432)

[4.4.1 Objetivo: 14](#_Toc482024433)

[4.4.2 Input: 14](#_Toc482024434)

[4.4.3 Output: 15](#_Toc482024435)

[4.4.4 Diagrama 15](#_Toc482024436)

[5. Requerimientos No Funcionales 16](#_Toc482024437)

[5.1 Requerimientos de Performance 16](#_Toc482024438)

[5.2 Requerimientos de Calidad de Software 16](#_Toc482024439)

[6. Pruebas de sistema 17](#_Toc482024440)

[6.1 Pruega N°1 17](#_Toc482024441)

[6.1.1 Pre-condición: 17](#_Toc482024442)

[6.1.2 Pasos: 17](#_Toc482024443)

[6.1.3 Resultado 17](#_Toc482024444)

[6.2 Pruega N°2 17](#_Toc482024445)

[6.2.1 Pre-condición: 17](#_Toc482024446)

[6.2.2 Pasos: 17](#_Toc482024447)

[6.2.3 Resultado 17](#_Toc482024448)

[6.3 Prueba N°3: 17](#_Toc482024449)

[6.3.1 Pre-condiciones: 17](#_Toc482024450)

[6.3.2 Pasos: 17](#_Toc482024451)

[6.3.3 Resultado 18](#_Toc482024452)

[6.4 Prueba N°4: 18](#_Toc482024453)

[6.4.1 Pre-condiciones: 18](#_Toc482024454)

[6.4.2 Pasos: 18](#_Toc482024455)

[6.4.3 Resultado 18](#_Toc482024456)

[6.5 Prueba N°5: 18](#_Toc482024457)

[6.5.1 Pre-condiciones: 18](#_Toc482024458)

[6.5.2 Pasos: 18](#_Toc482024459)

[6.5.3 Resultado 18](#_Toc482024460)

[6.6 Prueba N°6 18](#_Toc482024461)

[6.6.1 Pre-condición: 18](#_Toc482024462)

[6.6.2 Pasos: 18](#_Toc482024463)

[6.6.3 Resultado 19](#_Toc482024464)

[6.7 Prueba N°7 19](#_Toc482024465)

[6.7.1 Pre-condición: 19](#_Toc482024466)

[6.7.2 Pasos:. 19](#_Toc482024467)

[6.7.3 Resultado 19](#_Toc482024468)

[6.8 Prueba N°8 19](#_Toc482024469)

[6.8.1 Pre-condición: 19](#_Toc482024470)

[6.8.2 Pasos:. 19](#_Toc482024471)

[6.8.3 Resultado 19](#_Toc482024472)

[6.9 Prueba N°9 20](#_Toc482024473)

[6.9.1 Pre-condición: 20](#_Toc482024474)

[6.9.2 Pasos:. 20](#_Toc482024475)

[6.9.3 Resultado 20](#_Toc482024476)

[6.10 Prueba N°9 20](#_Toc482024477)

[6.10.1 Pre-condición: 20](#_Toc482024478)

[6.10.2 Pasos:. 20](#_Toc482024479)

[6.10.3 Resultado 20](#_Toc482024480)

[7. Matriz de trazabilidad 21](#_Toc482024481)

Historia de revisiones

|  |  |  |  |
| --- | --- | --- | --- |
| **Nombre** | **Fecha** | **Razón de los cambios** | **Version** |
| Jeremías Benítez | 02/05/17 | Creación del documento. | V1.0.0 |
| Jeremías Benítez | 03/05/17 | Se agregó toda la sección 2.  Se cambió el nombre de la aplicación.  Se actualizó el índice. | V1.0.1 |
| Jeremías Benítez | 06/05/17 | Se eliminó lo relacionado con la base de datos.  Se agregaron los requisitos funcionales. | V1.1.0 |
| Jeremías Benítez | 07/05/17 | Agregado requerimiento funcional 4.  Agregados requerimientos no funcionales.  Agregados diagramas.  Corregido el índice. | V1.1.1 |
| Maximiliano Rao | 08/05/2017 | Modificacion y actualización de índice  Pruebas de sistema  Matriz de trazabilidad | V 1.1.2 |

# Introducción

## Propósito

El presente documento ha sido redactado para especificar en su totalidad los requerimientos de las versiones v1.X de la aplicación móvil FCEFyN maps.

## 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 FCEFyN maps. Luego se enumeran las restricciones de software y hardware, como así también las dependencias necesarias y cualquier dato 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, así como diagramas de secuencia donde sea necesario.

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. 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.

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

Gráficos:

* Casos de uso, diagramas de actividades: <https://yuml.me/diagram/scruffy/activity/draw>
* Diagramas de secuencias: <https://www.websequencediagrams.com/>

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

# Descripción General

Esta sección provee una descripción de alto nivel de la aplicación FCEFyN maps.

## Definición

La aplicación FCEFyN maps es un nuevo sistema, el cual busca ofrecer a sus usuarios (los cuales serán los alumnos, personal docente y no docente de la institución) una forma más simple de desarrollarse en el ambiente universitario, permitiendo la posibilidad de navegar mejor dentro de la institución.

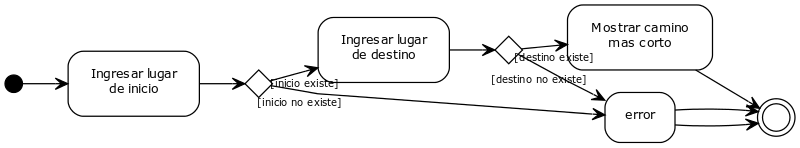
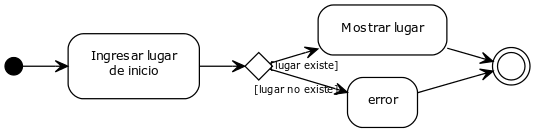
Diagrama de actividades para la acción de buscar camino más corto.

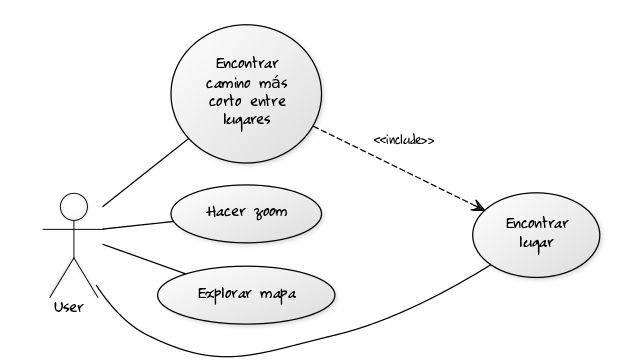
Diagrama de actividades para la acción de buscar un lugar.

## Funciones del Producto

El sistema debe permitirle a los usuarios realizar las siguientes acciones:

**A todos los usuarios:**

* Encontrar el camino más corto para llegar desde la posición de inicio provista, hasta la de destino deseada.
* Acceder al mapa de la facultad, donde estarán listados todos los edificios, pisos y entrepisos de la cede correspondiente.



## Diagrama de casos de uso para la aplicación.

## Clases de Usuario y Características

Hay dos grupos principales de usuarios que se espera interactúen con este sistema. El primer grupo está compuesto por los Administradores de Sistema encargados de mantener la integridad de los datos y la estabilidad del sistema. Este grupo tiene un gran conocimiento técnico. Si interacción con el sistema es mínima y solamente necesaria para hacer backups de las bases de datos, como también para actualizar los datos según sea necesario.

El segundo grupo es el de los usuarios. Se espera que tengan un conocimiento promedio de manejo de aplicaciones móviles. La interacción de los mismos con el sistema va a depender de sus necesidades, pero podemos asumir que van a interactuar con el mismo por lo menos una vez por día.

## Ambiente de Operación

El software correrá en dispositivos móviles que operen con el sistema operativo Android 5 o superior.

## Restricciones de Diseño e Implementación

**Limitaciones de software:**

* El sistema necesitará Android versión 5 o superior.
* Andorid Studio 2.3 será usado para el desarrollo del software.

**Limitaciones de hardware:**

* Cada usuario necesitará un dispositivo móvil que corra Android 5 o superior.

## Documentación de Usuario

Junto con el sistema se proveerá un link que dirija al usuario a una página on-line donde se encuentre el manual de uso.

## Suposiciones y Dependencias

El sistema depende de Java y del sistema operativo móvil Android (versión 5 o superior) para funcionar.

# Listado de Requerimientos Funcionales

**REQ 1:** Exploración – el usuario debe poder recorrer los mapas disponibles.

**REQ 2:** Recorrido más corto – el sistema debe encontrar el camino más corto desde la posición indicada de inicio y la posición de llegada.

**REQ 3:** Búsqueda – el sistema debe poder localizar el aula u oficina indicada por el usuario y devolverle al usuario el piso en donde dicho lugar se encuentra. Además actualizará la vista del mapa, centrando dicho lugar.

**REQ 4:** Zoom – el usuario debe poder realizar zoom para aumentar o disminuir la escala del mapa.

**REQ 5:** Reserva de aulas – El usuario debe ser capaz de realizar reservas para las aulas en días y horarios deseados, siempre que las mismas estén libres.

**REQ 6:** Detalle de aulas – El usuario debe poder acceder a un resumen de las características de un aula específica.

**REQ 7:** Búsqueda por parámetros – El sistema debe listar aulas, filtrando dicha lista según parámetro ingresados por el usuario.

# Detalle de los Requerimientos

## REQ 1: Exploración

### Objetivo:

El usuario debe poder recorrer libremente los mapas de todos los pisos y edificios de la intitución que estén disponibles.

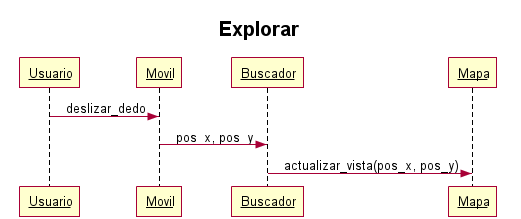
### Input:

Deslizamiento del dedo sobre la pantalla.

### Output:

En la pantalla del dispositivo móvil se deberá mostrar el mapa de la planta baja del edificio principal, el cual se deberá poder recorrer en cualquier dirección, así como también se podrá cambiar de piso ya sea mediante un menú deslizable que se encuentre a la izquierda de la pantalla o haciendo clic en los íconos correspondientes.

### Diagrama



## REQ 2: Recorrido más corto

### Objetivo:

El sistema debe encontrar el camino más corto desde la posición indicada de inicio y la posición de llegada. Esta opción estará disponible a través de un botón flotante ubicado en la esquina inferior derecha, el cual abrirá el menú de búsqueda.

### Input:

El usuario deberá ingresar el lugar desde el cual desea partir, y el lugar hasta el cual desea llegar. Se entiende por lugar a cualquier aula, oficina, u otros que tengan asignada una identificación en el mapa.

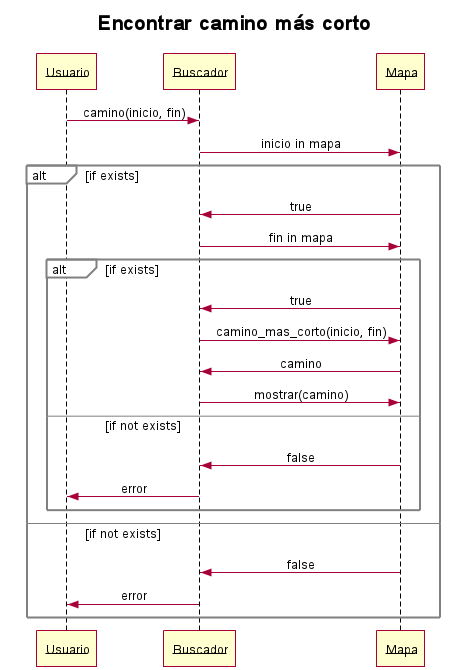
### Output:

Sobre el mapa mostrado en la pantalla principal se mostrará con una línea azul el camino que se deberá recorrer para llegar, recorriendo la menor distancia posible, hasta el lugar especificado por el usuario.

### Caso Excepcional:

Si alguno de los lugares provistos por el usuario no son encontrados por el sistema, el mismo emitirá un mensaje de error.

### Diagrama

******

## REQ 3: Búsqueda

### Objetivo:

El sistema debe poder localizar el aula u oficina indicada por el usuario y devolverle al usuario el piso en donde dicho lugar se encuentra. Además actualizará la vista del mapa, centrando dicho lugar.

### Input:

El usuario deberá ingresar el lugar del cual desea conocer la ubicación.

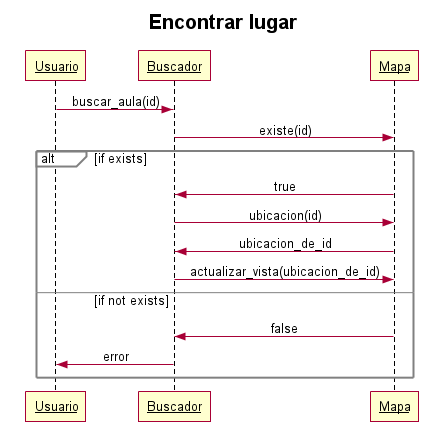
### Output:

Si el sistema tiene dicho lugar registrado deberá actualizar la vista del mapa para centrarla en dicho lugar. Deberá devolver también un mensaje donde se le indique textualmente al usuario en qué piso se encuentra.

### Caso Excepcional:

Si el lugar provisto no es encontrado por el sistema, se devolverá un mensaje de error y la vista principal permanecerá sin modificaciones.

### Diagrama

****

## REQ 4: Zoom

### Objetivo:

El sistema debe poder manejar zoom realizado por el usuario sobre el mapa, ampliando o reduciendo la escala del mismo según sea solicitado.

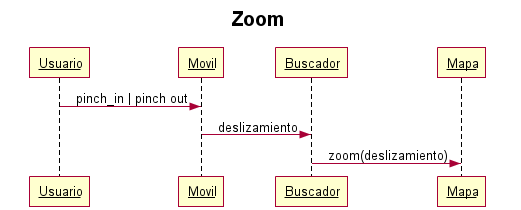
### Input:

El usuario realizará una acción de “pinching” con ambos dedos, ya sea en dirección entrante o saliente.

### Output:

La escala del mapa mostrado variará acorde a la dirección del “pinching”: si la misma es saliente respecto del centro del celular, la escala aumentará; si es, en cambio, entrante, la escala se reducirá-

### Diagrama

****

## REQ 5: Reserva de aulas

### Objetivo:

El usuario debe ser capaz de realizar reserva para aulas específicas en el día y horario que desee, dependiendo de la disponibilidad del aula en cuestión.

**Pre-condición:**

El usuario debe darle tap al botón de “Hacer reserva” ubicado en la vista del aula.

### Input:

El usuario debe ingresar la fecha y el horario, determinando así el momento en el que necesita usar el aula.

### Output:

Se mostrará un mensaje indicando que la acción se realizó correctamente, y se podrá ver la reserva registrada en la lista de reservas para esa aula y en la lista de reservas del usuario que la registre.

### Caso Excepcional:

En caso de que dicha reserva se superponga con otra ya realizada, se mostrará un mensaje de error, y se ignorará la reserva.

### **Diagrama**

## REQ 6: Detalle de las aulas

### Objetivo:

El usuario debe poder acceder a un resumen de las características de un aula específica al darle tap a la misma, ya sea en el mapa o en alguna lista que se le pueda presentar bajo otro requisito.

**Pre-condición:**

El usuario dará tap en el nombre de un aula.

### Input:

N/A

### Output:

Se mostrará una nueva ventana donde se listarán en la mitad superior de la misma las características del aula (cantidad de asientos, si tiene proyector o no, etc.); y en la parte inferior un listado con las reservas realizadas sobre la misma, donde cada entrada de la lista constará de el nombre de quien realizó la reserva, la fecha para la cual reservó, y los tiempo de inicio y de fin de dicha reserva.

En caso de no tener ninguna reserva registrada para el aula, la lista se mostrará vacía.

### Caso Excepcional:

N/A

### **Diagrama**

## REQ 7: Búsqueda por parámetros

### Objetivo:

El sistema debe listar aulas, filtrando dicha lista según los parámetros ingresados por el usuario.

**Pre-condición:**

El usuario ingresará a la ventana de Búsqueda por parámetros

### Input:

El usuario ingresará un número en el parámetro de “Cantidad mínima de alumnos”, el cuál deberá coincidir con el número de alumnos que espera tener en la clase. Debe también seleccionar si desea que se listen aulas con proyector y/o con pizarra blanca y/o con computadoras. Esto lo hará mediante radio buttons.

### Output:

Se mostrará una lista con las aulas que cumplan con dicho criterio. En caso de no existir ninguna, la lista se mostrará vacía.

### Caso Excepcional:

En el caso de que alguno de los parámetros sea incorrecto, se mostrará un mensaje de error por un par de segundos, y el focus se devolverá al parámetro que no sea válido.

### **Diagrama**

# Requerimientos No Funcionales

## Requerimientos de Performance

* El sistema debe encontrar la ruta entre dos puntos, o devolver un error si uno de los puntos no es válido, en menos de 0.5 segundos. Esto es así para evitar que la app caiga en desuso por no ser práctica.
* La interfaz de usuario debe ser lo más intuitiva que sea posible, para así volver útil la aplicación a personas mayores.

## Requerimientos de Calidad de Software

**Confiabilidad:**

* El sistema debe estar disponible un 95% de las veces que el usuario lo intente usar.

**Portabilidad:**

* El sistema sólo debe ser portable a otros dispositivos que corran el mismo sistema operativo.

**Facilidad de uso:**

* El sistema no debe requerir entrenamiento especial para ser operado por un usuario promedio.
* El sistema debe requerir entrenamiento de no más de una sesión de 10 minutos para ser operado por un usuario con bajo nivel de manejo de aplicaciones.
* El sistema debe poseer una sola ventana de ayuda, más un link a un manual de usuario on-line.

# Pruebas de sistema

## Prueba N°1

### Pre-condición:

Teniendo la aplicación abierta en la pantalla principal (mapa)

### Pasos:

#### Paso 1: Poner dos dedos sobre el mapa sobre la zona que se desea aumentar el zoom.

#### Paso 2: Realizar un “pinching” en una dirección saliente sobre el mapa.

### Resultado

El mapa debe aumentar el zoom en la zona sobre la cual hicimos “pinching”, si repetimos los pasos debe aumentar el zoom paulatinamente.

## Prueba N°2

### Pre-condición:

Teniendo la aplicación abierta en la pantalla principal (mapa)

### Pasos:

#### Paso 1: Poner dos dedos sobre el mapa sobre la zona que se desea reducir el zoom.

#### Paso 2: Realizar un “pinching” en una dirección entrante sobre el mapa.

### Resultado

El mapa debe reducir el zoom en la zona sobre la cual hicimos “pinching”, si repetimos los pasos debe reducir el zoom paulatinamente.

## Prueba N°3:

### Pre-condiciones:

Tener el programa abierto.

### Pasos:

#### Paso 1: Dar tap al botón de búsqueda, ubicado en el menú de la izquierda.

#### Paso 2 : Ingresar el aula o sitio que desea buscar.

#### Paso 3: Hacer clic sobre el botón 'Encontrar'.

### Resultado

Si el sistema tiene dicho lugar registrado deberá actualizar la vista del mapa para centrarla en dicho lugar. Deberá devolver también un mensaje donde se le indique textualmente al usuario en qué piso se encuentra.

## Prueba N°4:

### Pre-condiciones:

Tener el programa abierto.

### Pasos:

#### Paso 1: Dar tap al botón de Generación de recorrido, ubicado en el menú de la izquierda.

### Resultado

Se debe abrir una sub-ventana donde permita ingresar “Origen” y “Destino” también se puede visualizar el botón “Obtener recorrido”.

## Prueba N°5:

### Pre-condiciones:

Tener el programa abierto en la pantalla principal (mapa).

### Pasos:

#### Paso 1: Deslice un solo dedos sobre el mapa en cualquier dirección.

### Resultado

El mapa debe moverse en la dirección contraria a la cual se desliza el dedo.

## Prueba N°6

### Pre-condición:

Tener la aplicación abierta.

### Pasos:

#### Paso 1: Dar tap al botón de generación de recorrido.

#### Paso 2: Ingrese el “Origen” o deje el por defecto “Entrada” y ingrese el “Destino”

#### Paso 3: Hacer clic sobre el botón 'Obtener recorrido'.

### Resultado

Si el sistema tiene ambos lugares registrados deberá actualizar la vista del mapa para centrarla en dicho lugar. Sobre el mapa mostrado en la pantalla principal se mostrará con una línea azul el camino que se deberá recorrer para llegar, recorriendo la menor distancia posible, hasta el lugar especificado por el usuario.

## Prueba N°7

### Pre-condición:

Tener la aplicación abierta en la pantalla principal.

### Pasos:.

#### Paso 1: Dirigirse al menú deslizable en la parte superior izquierda.

#### Paso 2: Dar tap sobre el menú.

#### Paso 3: Escoger el piso donde deseemos ir.

### Resultado

Deberá actualizar la vista del mapa de cada piso según el usuario elija en el menú.

## Prueba N°8

### Pre-condición:

Tener la aplicación abierta en la pantalla principal.

### Pasos:.

#### Paso 1: Dar tap al botón de búsqueda, ubicado en el menú de la izquierda.

#### Paso 2 : Ingresar el aula o sitio que no exista.

#### Paso 3: Hacer clic sobre el botón 'Encontrar'.

### Resultado

Deberá mostrar un mensaje de error indicando que no existen el aula o sitio.

## Prueba N°9

### Pre-condición:

Tener la aplicación abierta en la pantalla principal.

### Pasos:.

#### Paso 1: Dar tap al botón de generación de recorrido.

#### Paso 2: Ingrese un Origen y/o Destino que no exista.

#### Paso 3: Hacer clic sobre el botón 'Obtener recorrido'.

### Resultado

Deberá mostrar un mensaje de error indicando que el Origen y/o Destino no existe.

## Prueba N°10

### Pre-condición:

Tener la aplicación abierta en la pantalla principal.

### Pasos:.

#### Paso 1: Dar tap al botón Busqueda..

### Resultado

Deberá mostrar una ventana donde se podrá visualizar el campo para completar con el aula o sitio a buscar y un botón “Encontrar”.

# Matriz de trazabilidad

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **PRUEBA N°1** | **PRUEBA N°2** | **PRUEBA N°3** | **PRUEBA N°4** | **PRUEBA N°5** | **PRUEBA N°6** | **PRUEBA N°7** | **PRUEBA N°8** | **PRUEBA N°9** | **PRUEBA N°10** |
| **REQ1** | **X** | **X** |  |  | **X** |  | **X** |  |  |  |
| **REQ2** |  |  | **X** |  | **X** | **X** |  |  | **X** |  |
| **REQ3** |  |  |  | **X** |  |  |  | **X** |  | **X** |
| **REQ4** | **X** | **X** |  |  |  |  |  |  |  |  |

# Sanity Test

Las pruebas que forman parte de este testo son:

* Prueba N°3
* Prueba N°5
* Prueba N°6
* Prueba N°8
* Prueba N°9