**Documento de requerimientos de software**

***Metro Cultural***

***Fecha:*** *05/09/2025*

**Tabla de contenido**

Historial de Versiones 3

Información del Proyecto 3

Aprobaciones 3

1. Propósito 4

2. Alcance del producto / Software 4

3. Referencias 4

4. Funcionalidades del producto 5

5. Clases y características de usuarios 5

6. Entorno operativo 5

7. Requerimientos funcionales 6

9.1. (Nombre de la funcionalidad 1) 6

9.2. (Nombre de la funcionalidad 2) 7

9.3. (Nombre de la funcionalidad N) 7

8. Reglas de negocio 8

9. Requerimientos de interfaces externas 9

9.1. Interfaces de usuario 9

9.2. Interfaces de hardware 9

9.3. Interfaces de software 9

9.4. Interfaces de comunicación 9

10. Requerimientos no funcionales 10

11. Otros requerimientos 11

12. Glosario 12

**Historial de Versiones**

| **Fecha** | **Versión** | **Autor** | **Organización** | **Descripción** |
| --- | --- | --- | --- | --- |
|  | 1.0 |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**Información del Proyecto**

| Empresa / Organización | DuocUC |
| --- | --- |
| Proyecto | Metro Cultural |
| Fecha de preparación | 15/08/2025 |
| Cliente | Ferrocarril metropolitano (Metro) |
| Patrocinador principal | Ferrocarril metropolitano (Metro) |
| Gerente / Líder de Proyecto | Felipe Gallardo |
| Gerente / Líder de Análisis de negocio y requerimientos | Felipe Gallardo |

**Aprobaciones**

| **Nombre y Apellido** | **Cargo** | **Departamento u Organización** | **Fecha** | **Firma** |
| --- | --- | --- | --- | --- |
| Felipe Gallardo | Scrum Master / Gestor del proyecto |  |  |  |
| Matias Millaqueo | Desarrollador / Diseñador / Tester |  |  |  |

1. **Propósito**

El propósito de este documento es especificar las pruebas que se realizarán para garantizar que la aplicación móvil Metro Cultural cumpla con todos los requisitos establecidos y funcione correctamente en el entorno de producción. El plan incluye las pruebas de funcionalidad, usabilidad, integración y seguridad de la aplicación.

1. **Alcance del producto / Software**

Propósito u objetivo general: Verificar que la aplicación Metro Cultural brinde información sobre eventos culturales cercanos a las estaciones del Metro de Santiago, de manera precisa y actualizada.

Beneficios: La aplicación mejorará la experiencia del usuario en el sistema de transporte público de Santiago, promoviendo el acceso a la cultura local y fortaleciendo la difusión de eventos culturales.

Objetivos: Asegurar que la aplicación cumpla con los requisitos funcionales y no funcionales especificados, que las interfaces sean intuitivas, y que la integración con servicios externos (como Google Maps y APIs culturales) sea correcta.

1. **Referencias**

[1.2.1 Plantilla Acta de Constitución.docx - Documentos de Google](https://docs.google.com/document/d/1_e-8SifHF7TNWU2uVDYgDVotV2NebJUL/edit#heading=h.gjdgxs)

[Fase1\_Definicion Proyecto APT.docx - Documentos de Google](https://docs.google.com/document/d/1rGhqXLtdP7njbwYuhKD-H33nagaQymmz/edit)

1. **Funcionalidades del producto**

A continuación se listan las funcionalidades clave que serán probadas:

-Búsqueda de Eventos Culturales por Estación

-Geolocalización y Mapas Integrados (Google Maps)

-Visualización de Detalles de los Eventos

-Recomendaciones Personalizadas según Ubicación

-Registro y Autenticación de Usuarios (si aplica)

1. **Clases y características de usuarios**

Usuario Regular

* Frecuencia de uso: Alta
* Características: Accede a eventos cercanos a estaciones del metro, interactúa con la aplicación de forma diaria.
* Funcionalidades relevantes: Búsqueda de eventos, ver detalles de los eventos.

Turista

* Frecuencia de uso: Moderada
* Características: Utiliza la app para conocer eventos culturales cercanos durante su visita a Santiago.
* Funcionalidades relevantes: Geolocalización, visualización de eventos cercanos.

1. **Entorno operativo**

#### Entorno Operativo

* **Plataforma**: Android, iOS
* **Sistemas operativos**: Android 10+, iOS 12+
* **Hardware**: Dispositivos móviles (smartphones y tabletas)
* **Requisitos de software**: Google Maps API, Google Places API, base de datos Supabase.

1. **Requerimientos funcionales**

**Búsqueda de eventos**

* **Descripción**: Los usuarios pueden buscar eventos culturales cercanos a estaciones de metro.
* **Prioridad**: Alta
* **Acciones iniciadas y comportamiento esperado**: El usuario ingresa en la estación de metro y se despliega una lista de eventos cercanos.
* **Requerimientos**:  
  + REQ-1: La búsqueda debe ser precisa y actualizada.
  + REQ-2: La aplicación debe mostrar eventos con ubicación y fecha.
  + REQ-3: Los datos deben ser obtenidos desde una base de datos en tiempo real.

**Geolocalización**

* **Descripción**: La app debe mostrar en un mapa los eventos cercanos a la ubicación actual del usuario.
* **Prioridad**: Alta
* **Acciones iniciadoras y comportamiento esperado**: El usuario activa la función de ubicación y se muestra el mapa con eventos cercanos.
* **Requerimientos**:  
  + REQ-4: El mapa debe ser interactivo y permitir ver los eventos cercanos.
  + REQ-5: Debe funcionar en modo offline para eventos ya cargados.

1. **(Nombre de la funcionalidad 1)**

En el título de la funcionalidad, se recomienda utilizar nombres lo más descriptivo posible para cada funcionalidad. No limitarse a nombrarlas “Funcionalidad 1”. Un buen ejemplo podría ser “Autorización de pedido de compra”.

Descripción: Descripción corta de la funcionalidad.

Prioridad: Nivel bajo, medio o alto de prioridad. Esta debe ser establecida por el área funcional.

Acciones iniciadoras y comportamiento esperado: Secuencia de acciones de usuario y respuestas esperadas del sistema para esta funcionalidad.

Requerimientos funcionales: Lista detallada de los requerimientos funcionales asociados a esta funcionalidad.

Para cada requerimiento funcional se establece como debe mostrarse el software y cuales comportamientos debe desempeñar para que el usuario pueda realizar la función que necesita.

Es recomendable incluir como el software debe responder a condiciones de error y entradas de datos inválidas.

### Registro y Autenticación de Usuarios

* **Descripción**: El sistema debe permitir a los usuarios registrarse y autenticarse mediante correo electrónico y contraseña.
* **Prioridad**: Media
* **Condiciones**: Validación de datos al registrarse, mostrar mensajes de error claros si los datos son inválidos.

### Búsqueda de Eventos Culturales

* **Descripción**: El sistema debe permitir a los usuarios buscar eventos culturales asociados a una estación del Metro.
* **Prioridad**: Alta
* **Condiciones**: La búsqueda debe actualizarse automáticamente con la base de datos en tiempo real.

### Geolocalización y Mapas

* **Descripción**: La aplicación debe mostrar en un mapa (Google Maps) los eventos culturales cercanos a la ubicación actual del usuario.
* **Prioridad**: Alta
* **Condiciones**: El mapa debe ser interactivo, permitir zoom y mostrar detalles de cada evento.

### Visualización de Detalles de Eventos

* **Descripción**: Los usuarios podrán acceder a la información detallada de cada evento (nombre, ubicación, fecha, descripción, imágenes).
* **Prioridad**: Alta

### Recomendaciones Personalizadas

* **Descripción**: El sistema debe recomendar eventos culturales en base a la estación seleccionada, ubicación actual y preferencias del usuario.
* **Prioridad**: Media

### Favoritos

* **Descripción**: Los usuarios registrados podrán marcar eventos como favoritos para consultarlos posteriormente.
* **Prioridad**: Media

### Integración con APIs Externas

* **Descripción**: El sistema debe integrarse con **Google Maps API, Google Places API** y APIs culturales (si están disponibles) para obtener información actualizada.
* **Prioridad**: Alta

### Gestión de Datos

* **Descripción**: El administrador podrá crear, leer, actualizar y eliminar (CRUD) eventos y estaciones desde un panel web.
* **Prioridad**: Alta

### Notificaciones

* **Descripción**: La aplicación debe enviar notificaciones push sobre eventos destacados o cercanos.
* **Prioridad**: Baja (opcional en la primera versión).

### Multilenguaje

* **Descripción**: La aplicación debe permitir cambiar el idioma entre español e inglés para atender a turistas.
* **Prioridad**: Media

1. **Reglas de negocio**

**Acceso a eventos**

Solo los usuarios registrados pueden guardar eventos como favoritos.

Los eventos solo se pueden ver si se encuentran en un radio de 5 km desde la estación seleccionada.

1. **Requerimientos de interfaces externas**
2. **Interfaces de usuario**

#### Requerimientos de Interfaces Externas

* **Interfaces de usuario**: Diseño UX/UI intuitivo y adaptado a dispositivos móviles.
* **Interfaces de hardware**: Compatible con smartphones Android y iOS.
* **Interfaces de software**: Integración con Google Maps API y Supabase.
* **Interfaces de comunicación**: Protocolo HTTPS para seguridad en las comunicaciones.

1. **Interfaces de hardware**

Información sobre cuales tipos de dispositivos soporta el sistema por ejemplo: Computadores, dispositivos móviles, impresoras, otros dispositivos.

Protocolos de comunicación que soporta.

Interacciones de datos y control entre el software y el hardware.

1. **Interfaces de software**

Aquí se describen las interacciones entre el software y otros componentes, incluyendo: Otros componentes de software y sistemas, y de ser aplicables bases de datos, sistemas operativos, herramientas, librerías, componentes de software comercial, entre otros.

1. **Interfaces de comunicación**

Requerimientos de las funciones de comunicación que requiere el producto, incluyendo email, navegadores web, protocolos de comunicación de red, formularios electrónicos, entre otros.

Incluye formatos de mensajería, estándares de comunicación (Ej. FTP, HTTP, etc.). Describir también requerimientos de encriptación y seguridad en las comunicaciones.

1. **Requerimientos no funcionales**

Los [requerimientos no funcionales](http://www.pmoinformatica.com/2013/01/requerimientos-no-funcionales-porque.html) son los que especifican criterios para evaluar la operación de un servicio de tecnología de información, en contraste con los requerimientos funcionales que especifican los comportamientos específicos.

Para ver algunos ejemplos de cómo se redactan los requerimientos no funcionales, te recomendamos el siguiente enlace:

> [Ejemplos de requerimientos no funcionales de software](http://www.pmoinformatica.com/2015/05/requerimientos-no-funcionales-ejemplos.html)

1. **Otros requerimientos**
2. **Glosario**