

**Informe Ejecutivo  
Se lo Tengo App**

Universidad Libre  
Facultad de Ingeniería  
Programa de Ingeniería de Sistemas

Autores: Ángel Yesid Becerra Herrera, Jhojan Infante Montaño, Christian Camilo Manrique Molano, Cesar Camilo Vergara Castillo, Jose Andrés Castañeda

Docente: Mg. Arellys de Jesús Correa Rodríguez

Bogotá D.C.  
2025

# Tabla de Imágenes

Imagen 1 – Diagrama de Casos de Uso

Imagen 2 – Diagrama de Clases

Imagen 3 – Diagrama Entidad-Relación

Imagen 4 – Diagrama de Arquitectura en Azure

Imagen 5 – Prototipo UI/UX

# Tabla de Tablas

Tabla 1 – Cronograma resumido del Proyecto

Tabla 2 – Matriz RACI resumida del Proyecto

# Resumen Ejecutivo

El proyecto “Se lo Tengo App” surge como respuesta a una problemática común en la comunidad estudiantil de la Universidad Libre: la dispersión de la información y la ausencia de un canal centralizado para acceder a servicios esenciales. La solución consiste en el desarrollo de una aplicación móvil y web que centralice la oferta y demanda de servicios estudiantiles como transporte compartido, tutorías y materiales de estudio. La arquitectura técnica estará basada en un enfoque orientado a servicios (SOA), con frontend en React, backend en .NET Core y base de datos SQL en Azure, asegurando escalabilidad y seguridad.

# Introducción

El entorno académico actual exige soluciones digitales que mejoren la eficiencia y accesibilidad de los recursos. En la Universidad Libre, los estudiantes han manifestado la necesidad de contar con una plataforma que unifique servicios cotidianos en un solo canal. “Se lo Tengo App” busca transformar la experiencia universitaria ofreciendo una solución innovadora alineada con la misión institucional y las necesidades reales de la comunidad estudiantil.

# Objetivos

## Objetivo General

Desarrollar una aplicación móvil y web para la Universidad Libre que centralice la gestión de servicios académicos y de apoyo estudiantil, optimizando la interacción entre usuarios y fortaleciendo la experiencia universitaria.

## Objetivos Específicos

1. Diseñar una arquitectura de software escalable, modular y segura.  
2. Implementar funcionalidades que permitan publicar, solicitar y gestionar servicios.  
3. Integrar mecanismos de autenticación confiables y accesibles.  
4. Incorporar calificaciones y reportes que fortalezcan la confianza entre usuarios.  
5. Definir y aplicar un plan de proyecto con cronograma, costos y roles claramente establecidos.

# Alcance

El proyecto contempla un MVP que incluye funcionalidades de registro de usuarios, publicación y búsqueda de servicios, confirmación de solicitudes y calificaciones, además de un módulo de administración. En fases posteriores se integrarán pasarela de pagos, publicidad y funciones premium.

# Límites y Delimitaciones

El alcance inicial se restringe a la comunidad de la Universidad Libre. No se incluirán servicios externos en la fase inicial y las funcionalidades avanzadas como geolocalización o pagos se desarrollarán en iteraciones futuras.

# Metodología

## Enfoques y Procesos

Se emplea la metodología ágil Scrum, con iteraciones quincenales. Se definieron historias de usuario y casos de uso como base del backlog, permitiendo priorizar funcionalidades de acuerdo con las necesidades del público objetivo.

## Aplicación de las Etapas

### 1. Etapa de Inicio

- Identificación del problema.  
- Benchmarking de soluciones similares.  
- Elaboración del modelo Canvas y propuesta de valor.

### 2. Etapa de Planificación

- Definición de objetivos alineados a la misión institucional.  
- Identificación del público objetivo: estudiantes, docentes y administrativos.  
- Análisis de la competencia y apps similares.  
- Requisitos funcionales y técnicos.  
- Creación de plan de proyecto: cronograma, RACI y precedencias.  
- Herramientas y recursos: Azure, React, .NET Core, Figma, GitHub.  
- Propuesta al cliente y validación del plan.  
- Resultado: cronograma de 16 semanas con costos OPEX y CAPEX.

## Tabla 1 – Cronograma resumido del Proyecto

|  |  |  |  |
| --- | --- | --- | --- |
| Actividad | Rol Responsable | Predecesora | Duración |
| Identificación de la necesidad y problema | BP, PM | - | 2 días |
| Benchmarking de soluciones | DM, BP | 1.1 | 3 días |
| Arquitectura técnica | DM, SA | 3.3 | 4 días |
| Construcción de módulos funcionales | DEV, PM | 4.3, 4.4 | 15 días |
| Pruebas de integración y regresión | QA, DEV | 5.3, 5.4 | 6 días |
| Go-Live en producción | DM, DevOps | 6.3 | 2 días |

## Tabla 2 – Matriz RACI resumida del Proyecto

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Actividad | PM | DM | DEV | QA | BP |
| Identificación de la necesidad y problema | A | C | - | - | R |
| Arquitectura técnica | C | A/R | C | - | I |
| Construcción de módulos | C | R | A/R | - | I |
| Pruebas de integración | I | C | C | A/R | I |
| UAT | C | I | - | R | A |

# Anexos

A continuación, se presentan los diagramas y prototipos relacionados con el proyecto.

Imagen 1 – Diagrama / Prototipo 1

Imagen 2 – Diagrama / Prototipo 2

Imagen 3 – Diagrama / Prototipo 3

Imagen 4 – Diagrama / Prototipo 4

Imagen 5 – Diagrama / Prototipo 5

# Bibliografía

Fowler, M. Patterns of Enterprise Application Architecture. Addison-Wesley, 2002.  
Bass, L., Clements, P., Kazman, R. Software Architecture in Practice. Addison-Wesley, 2012.  
Jacobson, I., Booch, G., Rumbaugh, J. The Unified Software Development Process. Addison-Wesley, 1999.  
McConnell, S. Code Complete. Microsoft Press, 2004.  
Documentación del proyecto: Idea de Proyecto, Parcial Ing. Software II, Trabajo Final, Prototipo y Arquitectura.