[[1]](#footnote-1)

Diseño de la Arquitectura de Software para una App Móvil Orientada ofrecer servicios académicos en la Universidad Libre

Manrique. Christian, Infante. Jhojan, Becerra. Ángel, Vergara. Cesar, Castañeda. Jose

Resumen –

**"Se lo Tengo App" es una aplicación móvil creada por y para estudiantes de la Universidad Libre, destinada a centralizar servicios esenciales como alquiler de materiales, venta de libros, tutorías y transporte compartido. Su arquitectura se basa en un enfoque orientado a servicios, garantizando escalabilidad y mantenimiento eficiente. Utiliza un frontend desarrollado con React, HTML y CSS, alojado en Azure Static Web Apps, y un backend en .NET Core, desplegado en Azure App Service, para manejar la lógica de negocio. La base de datos SQL en Azure gestiona las relaciones complejas entre entidades como usuarios, ofertas, solicitudes y reportes. La integración a través de APIs asegura una comunicación fluida entre componentes, mientras que la seguridad y la capacidad de escalar permiten incorporar nuevas funcionalidades sin comprometer la estabilidad del sistema. Este diseño robusto y moderno no solo satisface las necesidades actuales de los estudiantes, sino que está preparado para futuras expansiones y mejoras, mejorando continuamente la experiencia estudiantil en la Universidad Libre.**

**Palabras Clave: Diseño del sistema, Arquitectura de Software, App Móvil,**

**Abstract**

**"Se lo Tengo App" is a mobile application created by and for students of Universidad Libre, aimed at centralizing essential services such as renting study materials, selling books, tutoring, and ridesharing. Its architecture is based on a service-oriented approach, ensuring scalability and efficient maintenance. It uses a frontend developed with React, HTML, and CSS, hosted on Azure Static Web Apps, and a backend in .NET Core, deployed on Azure App Service to handle business logic. The SQL database in Azure manages the complex relationships between entities such as users, offers, requests, and reports. Integration through APIs ensures smooth communication between components, while security and scalability allow for the incorporation of new features without compromising system stability. This robust and modern design not only meets the current needs of students but is also prepared for future expansions and improvements, continuously enhancing the student experience at Universidad Libre.**

**Keywords: Application, Diagrams, Entities, Classes, Models, solution, implementation, process, student, data, offer, service, publication**

# introduccion

En el contexto de la Universidad Libre, una institución comprometida con la formación integral y el desarrollo de sus estudiantes surge la iniciativa de crear una solución tecnológica que aborde las necesidades específicas de la comunidad estudiantil. "Se lo Tengo App" es una aplicación móvil concebida y diseñada por estudiantes para estudiantes, con el objetivo de facilitar el acceso y la gestión de recursos y servicios esenciales dentro del entorno universitario. Esta plataforma permitirá el alquiler de materiales de estudio, la venta de libros, la oferta de tutorías académicas y la organización de servicios de transporte compartido.

El diseño de la arquitectura de "Se lo Tengo App" se basa en un enfoque orientado a servicios, utilizando tecnologías modernas y prácticas de desarrollo que aseguran escalabilidad, mantenimiento y eficiencia. Para llevar a cabo este proyecto, se realizó un exhaustivo levantamiento de requisitos, que permitió establecer los componentes clave de la arquitectura del sistema. Estos incluyen la estructura de los datos y las relaciones entre las diferentes entidades involucradas, así como la lógica de negocio necesaria para gestionar las interacciones de los usuarios con la aplicación.

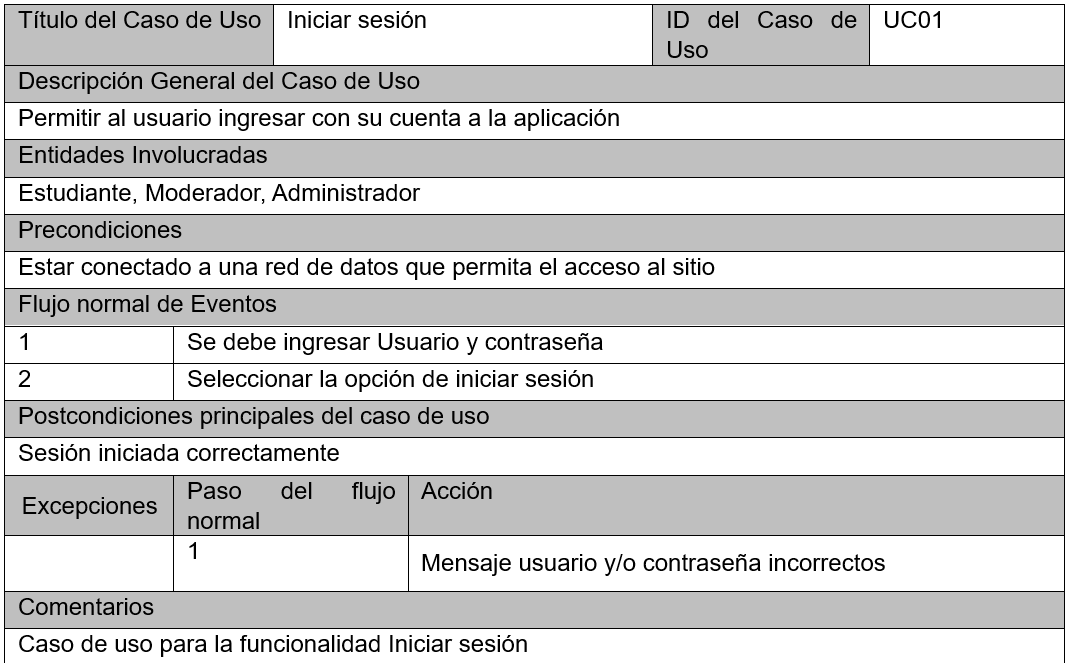
La implementación de la arquitectura se apoya en una infraestructura tecnológica avanzada, que incluye un frontend desarrollado con tecnologías web modernas, un backend robusto y una base de datos eficiente. Todo esto se integra a través de servicios que garantizan una comunicación fluida y segura entre los diferentes componentes del sistema, permitiendo que cada funcionalidad de la aplicación sea accesible de manera modular.

El enfoque orientado a servicios facilita la escalabilidad y el mantenimiento de la aplicación, asegurando que nuevas funcionalidades puedan ser integradas sin afectar el rendimiento o la estabilidad del sistema existente. Este diseño también garantiza que la aplicación sea capaz de manejar eficientemente el volumen de datos y las interacciones esperadas dentro de la comunidad estudiantil.

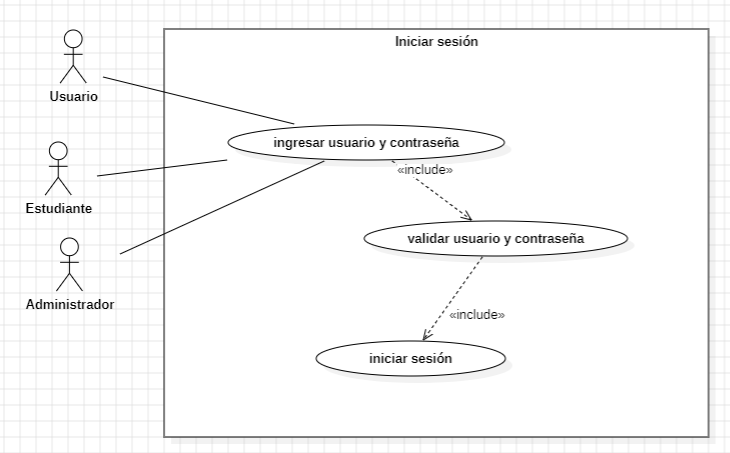
# Análisis y diseño de la arquitectura

## Diseño de Casos de uso

Dentro del proyecto "Se lo Tengo App", desarrollamos los casos de uso identificando actores como estudiantes y administradores, y definiendo objetivos específicos para cada uno. Describimos los flujos principales y alternativos de cada interacción, y establecimos precondiciones y postcondiciones. Este enfoque detallado nos permitió asegurar que todas las funcionalidades clave de la aplicación fueran bien implementadas y alineadas con las necesidades de los usuarios, mejorando así la experiencia estudiantil.

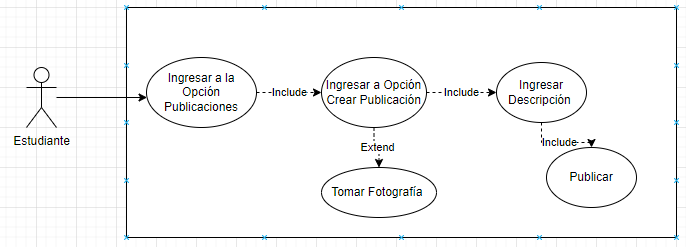
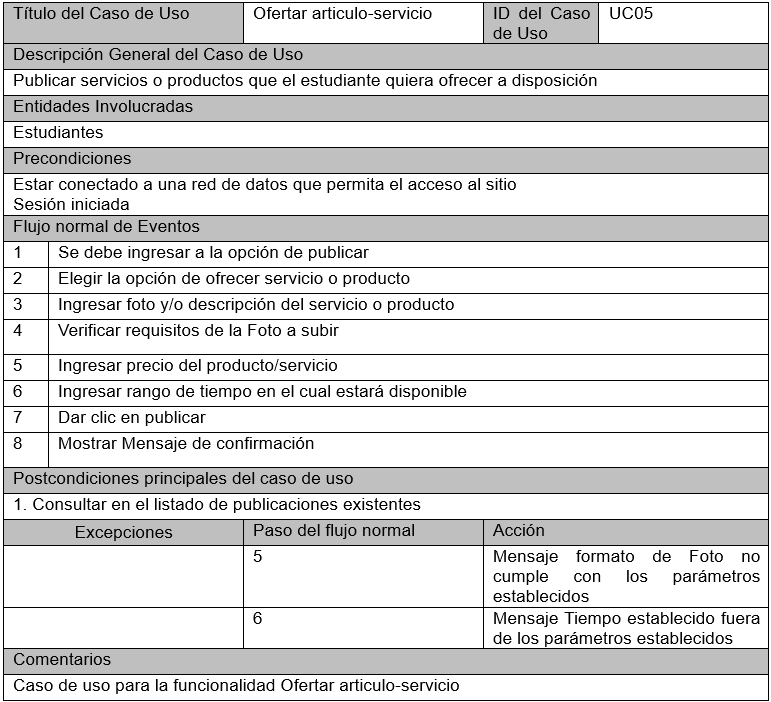


*Tabla 1 - CU Iniciar sesión*

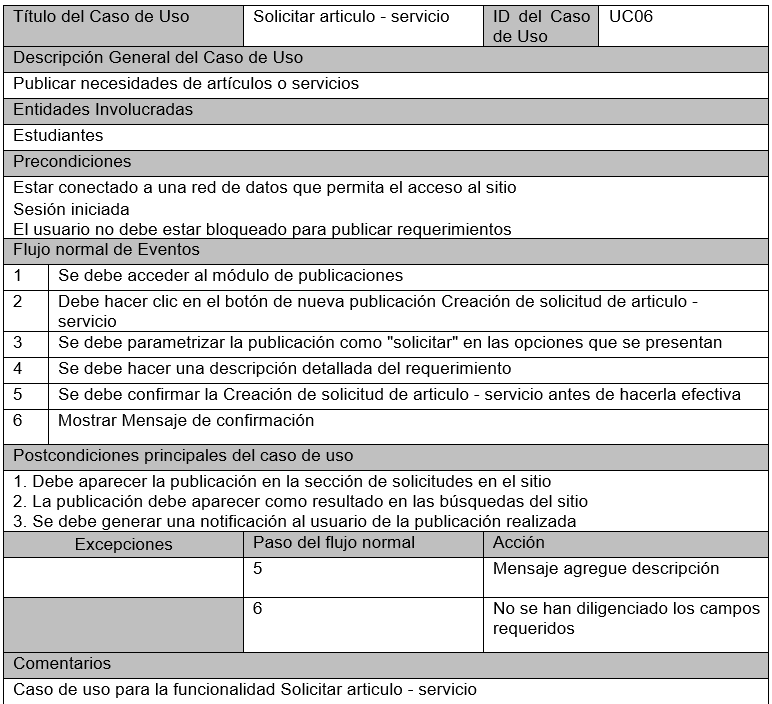


*Ilustración 1 - Diagrama 1*

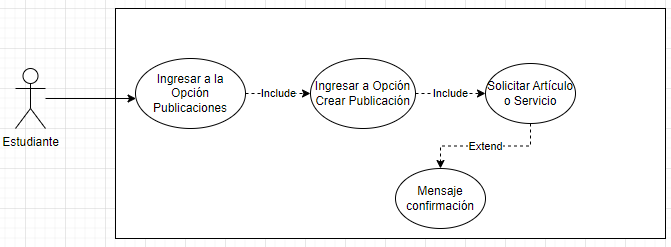
*Tabla 2 - CU Ofertar Art/Serv*



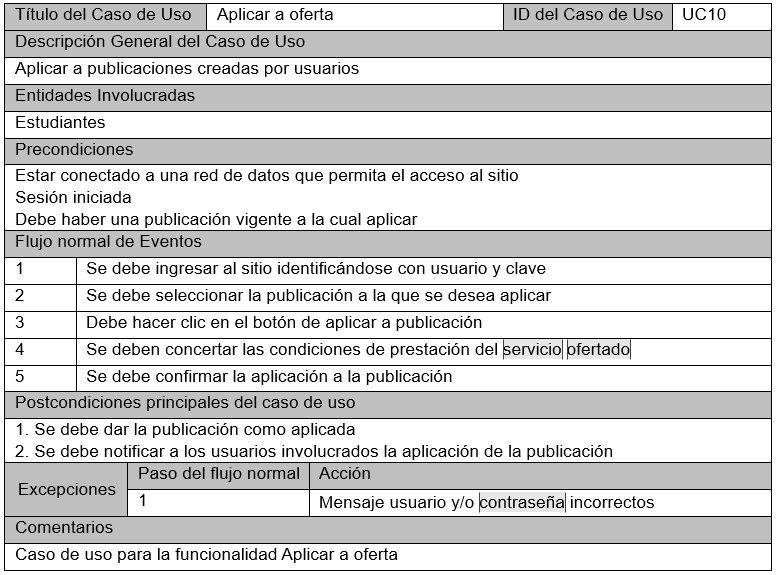
*Ilustración 2 - Diagrama 2*



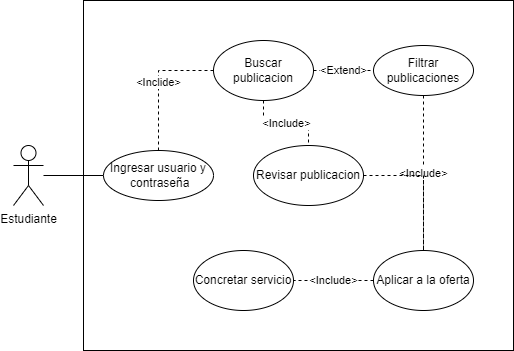
*Tabla 3 - CU Solicitar articulo*



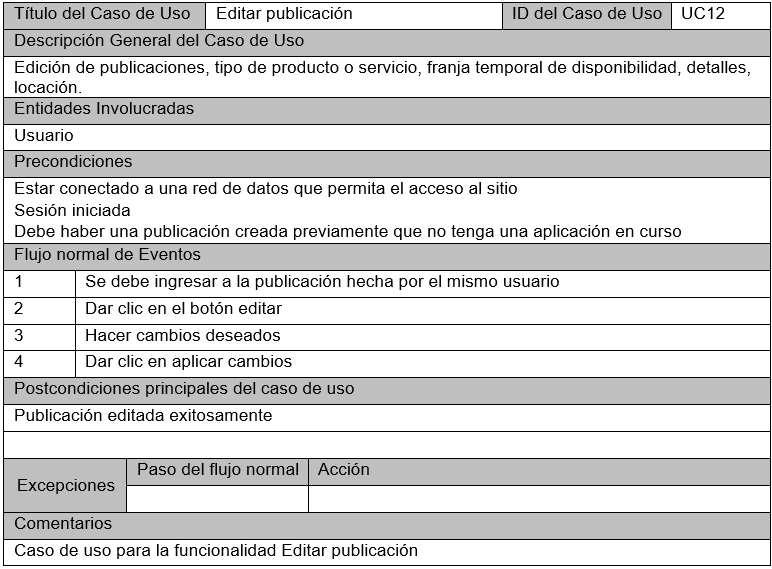
*Ilustración 3 - Diagrama 3*



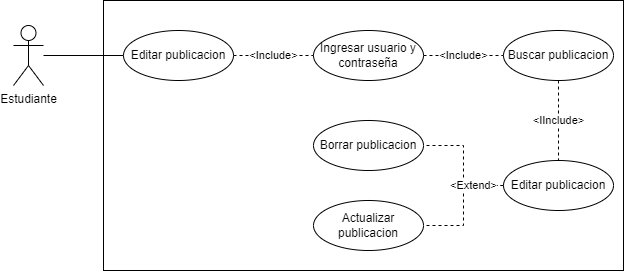
*Tabla 4 - CU Aplicar a oferta*



*Ilustración 4 - Diagrama 4*



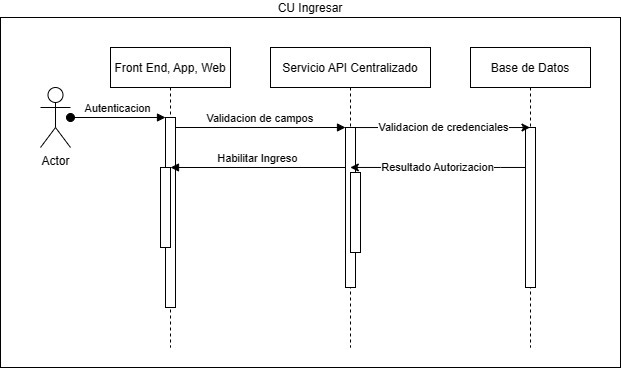
*Tabla 5 - CU Editar publicacion*



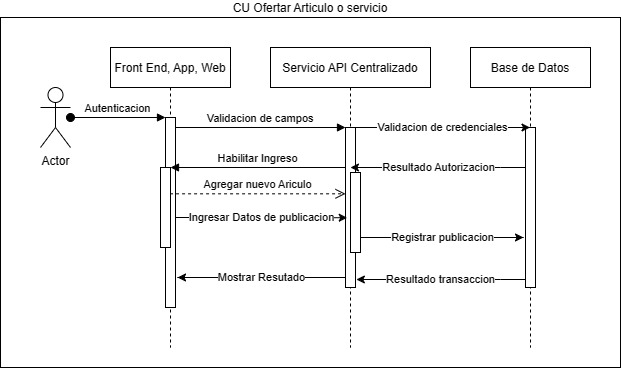
*Ilustración 5 - Diagrama 5*

## Diagramas de Secuencia

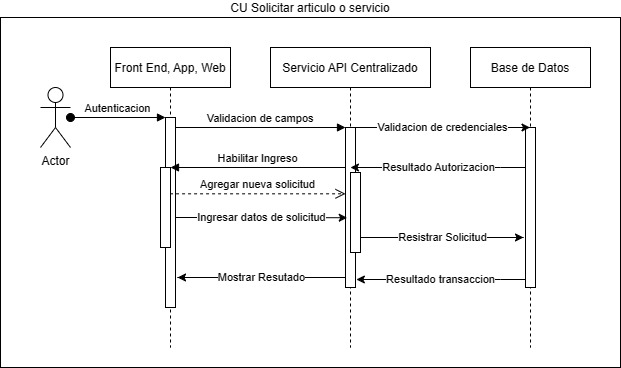
Dentro del proyecto "Se lo Tengo App", creamos diagramas de secuencia para modelar la interacción entre los diferentes componentes del sistema y los actores, como estudiantes y administradores. Estos diagramas detallan el flujo de mensajes y la lógica de cada proceso clave, asegurando una comprensión clara de cómo se llevan a cabo las operaciones. Este enfoque nos permitió identificar posibles mejoras y garantizar que la implementación fuera coherente y eficiente, optimizando la experiencia de usuario en la aplicación.



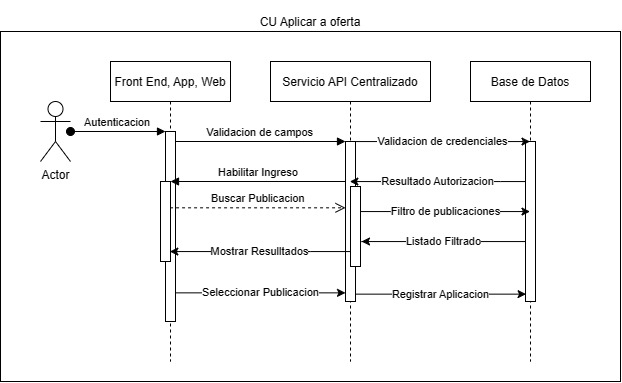
*Ilustración 6 - Secuencia ingresar*



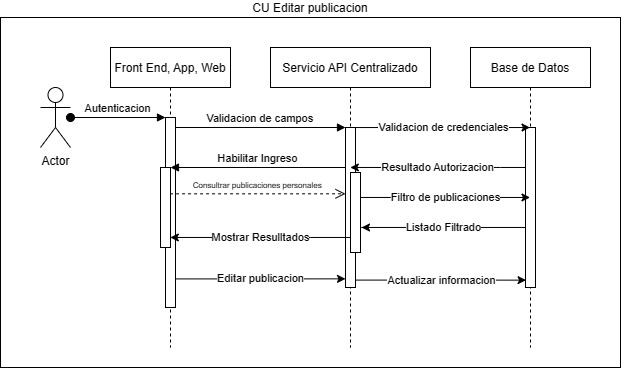
*Ilustración 7 - Secuencia Ofertar Art/Serv*



*Ilustración 8 - Secuencia solicitar Art/Serv*



*Ilustración 9 - Secuencia Aplicar*

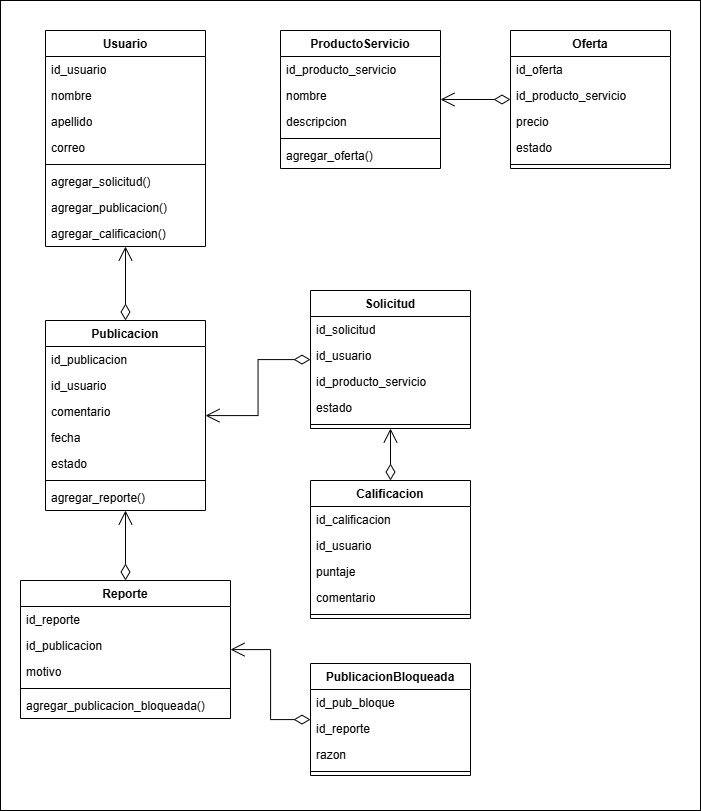


*Ilustración 10 - Secuencia Editar publicacion*

## Diseño de la arquitectura de la solución.

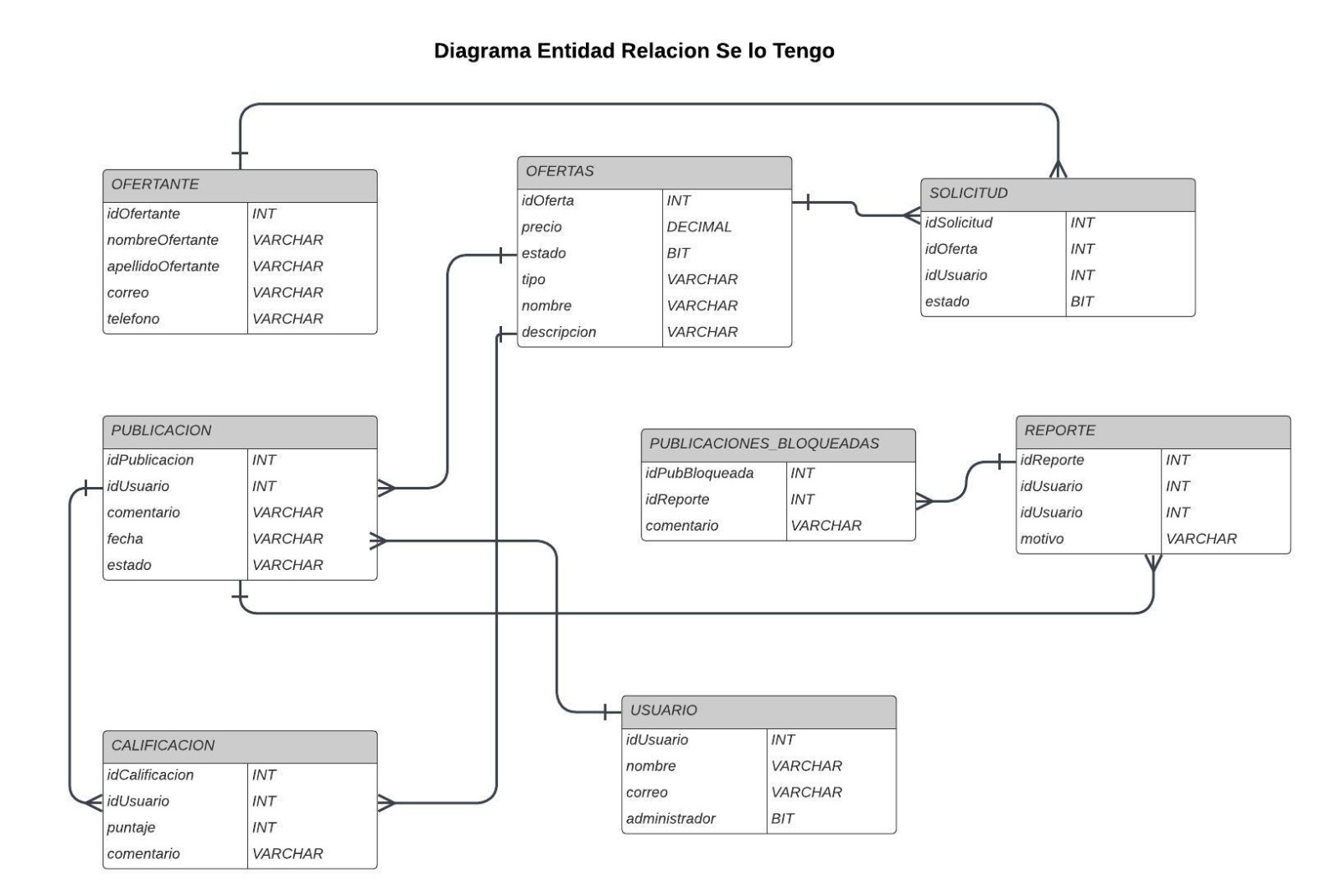
El diagrama de clases de "Se lo Tengo App" muestra que los usuarios pueden agregar solicitudes, publicaciones y calificaciones. Producto Servicio se asocia con Oferta, que incluye precio y estado. Las solicitudes se vinculan a usuarios y productos o servicios, con un estado que indica su progreso. Las publicaciones registran comentarios y fechas, y pueden ser reportadas y bloqueadas. Las calificaciones asignan puntajes y comentarios a los usuarios.

*1) Diagrama de clases:* El diagrama de clases mostrado en la ilustración 11 describe la estructura de "Se lo Tengo App". Los actores principales incluyen usuarios que pueden agregar solicitudes, publicaciones y calificaciones. La clase Producto Servicio se asocia con la clase Oferta, que contiene detalles específicos como precio y estado. Las solicitudes se vinculan tanto a usuarios como a productos o servicios, con un estado que indica su progreso. Las publicaciones, que pueden ser reportadas, registran comentarios y fechas. Los reportes pueden llevar a que publicaciones sean bloqueadas. Las calificaciones se asignan a usuarios, proporcionando un puntaje y comentarios que reflejan la calidad del servicio o producto ofrecido.



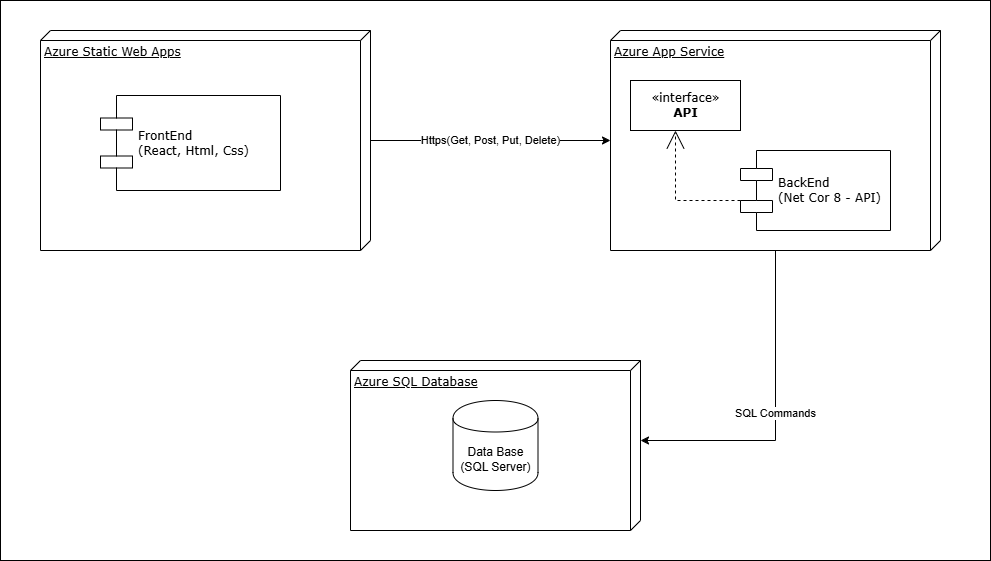
*Ilustración 11 - Diagrama de clases*

2) *Diagrama de entidad relación*: El diagrama de entidad-relación de "Se lo Tengo App" (Ilustración 12) muestra las principales entidades y sus relaciones. Los Ofertantes proporcionan Ofertas de productos o servicios, que pueden ser solicitadas a través de Solicitudes. Las Publicaciones registran comentarios sobre las ofertas, y pueden ser evaluadas mediante Calificaciones. Los Usuarios interactúan con el sistema publicando, solicitando y calificando. Los Reportes gestionan problemas en las publicaciones, y estas pueden ser Bloqueadas si es necesario. Esta estructura permite una gestión integral y eficiente de las interacciones y datos dentro de la aplicación.



*Ilustración 12 - Diagrama entidad relación*

3) *Diagrama de implementación/Arquitectura*: El diagrama muestra la arquitectura de implementación de "Se lo Tengo App" en Azure. El Frontend, desarrollado con React, HTML y CSS, se aloja en Azure Static Web Apps. Este interactúa con el Backend, implementado en .NET Core y desplegado en Azure App Service, a través de llamadas HTTPS (GET, POST, PUT, DELETE). El backend maneja la lógica de negocio y se comunica con la Azure SQL Data base mediante comandos SQL para gestionar los datos. Esta arquitectura orientada a servicios asegura una integración eficiente y escalabilidad, permitiendo una gestión robusta y segura de la aplicación.



*Ilustración 13 - Diagrama de implementacion*

IX. CONCLUSIÓN

A lo largo del análisis y la redacción, hemos profundizado en el desarrollo y la arquitectura de "Se lo Tengo App", una aplicación móvil diseñada por estudiantes para estudiantes de la Universidad Libre. Esta aplicación surge con el objetivo de centralizar y facilitar el acceso a diversos servicios esenciales como el alquiler de materiales de estudio, la venta de libros, las tutorías académicas y el transporte compartido. La solución propuesta aborda directamente las dificultades que enfrentan los estudiantes al intentar acceder a estos recursos, ofreciendo una plataforma unificada y eficiente.

El diseño de la arquitectura de la aplicación se fundamenta en un enfoque orientado a servicios, lo que garantiza su escalabilidad y mantenimiento a largo plazo. La integración de tecnologías modernas como React para el frontend, .NET Core para el backend y Azure SQL para la base de datos, asegura una comunicación fluida entre los distintos componentes del sistema. Este enfoque modular y orientado a servicios permite la incorporación de nuevas funcionalidades sin comprometer la estabilidad del sistema, adaptándose a las necesidades cambiantes de la comunidad estudiantil.

La creación de diagramas de clases y entidades fue esencial para definir la lógica de negocio y la estructura de la base de datos, respectivamente. Estos diagramas permitieron identificar las relaciones entre los diferentes elementos del sistema, como usuarios, ofertas, solicitudes, publicaciones y reportes, asegurando una gestión coherente y eficiente de la información. Asimismo, los diagramas de implementación y secuencia ayudaron a visualizar y planificar la interacción entre los distintos componentes del sistema, proporcionando una hoja de ruta clara para el desarrollo y la integración.

La metodología utilizada para desarrollar los casos de uso fue fundamental para entender y documentar cómo los usuarios interactuarían con la aplicación. Al identificar actores y objetivos específicos, y describir los flujos de interacción principales y alternativos, se logró una comprensión detallada de las operaciones clave de la aplicación. Esto no solo facilitó el desarrollo técnico, sino que también garantizó que la solución estuviera alineada con las necesidades reales de los usuarios.

La solución integral presentada por "Se lo Tengo App" no solo resuelve problemas actuales, sino que también está preparada para futuras expansiones y mejoras. El uso de una arquitectura moderna, el diseño detallado de la lógica de negocio y la estructura de datos, y la planificación meticulosa de la implementación, aseguran que la aplicación proporcione una experiencia óptima y continua para los estudiantes de la Universidad Libre. Este proyecto es un ejemplo claro de cómo una solución tecnológica bien diseñada puede transformar y mejorar significativamente la experiencia educativa y personal de los estudiantes.

References

1. M. Fowler, Patterns of Enterprise Application Architecture. Addison-Wesley, 2002.
2. L. Bass, P. Clements, and R. Kazman, Software Architecture in Practice, 3rd ed. Addison-Wesley, 2012.
3. I. Jacobson, G. Booch, and J. Rumbaugh, The Unified Software Development Process. Addison-Wesley, 1999.
4. A. Cockburn, Writing Effective Use Cases. Addison-Wesley, 2000.
5. S. McConnell, Code Complete: A Practical Handbook of Software Construction, 2nd ed. Microsoft Press, 2004.
6. "Unified Modeling Language (UML) Resource Page," OMG, [Online]. Available: https://www.omg.org/uml/. [Accessed: 01-Jun-2024].

1. [↑](#footnote-ref-1)