

UNIVERSIDAD PRIVADA DE TACNA

**FACULTAD DE INGENIERIA**

**Escuela Profesional de Ingeniería de Sistemas**

**Informe Final Proyecto *API\_SERVICIOS***

Curso: *Patrones de diseño de software*

Docente: *Ing. Patrick Cuadros*

|  |  |
| --- | --- |
| Integrantes: |  |
| ***Condori Loayza Helbert Andres*** | ***2020067571*** |
| ***Diego Andre Aranda Reyes*** | ***2019063855*** |
| ***Alvaro Gallegos Laucata*** | ***2017057608*** |
| ***Erick Mauricio Mamani Lima*** | ***2020066321*** |
| ***Frank Kevin Paz Huaychani***  ***Sebastian Cotrina Caceres*** | ***2019063321***  ***2020067569*** |

**Tacna – Perú**

***2023***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| CONTROL DE VERSIONES | | | | | |
| Versión | Hecha por | Revisada por | Aprobada por | Fecha | Motivo |
| 1.0 | MPV | ELV | ARV | 10/07/2023 | Versión Original |

**Índice General**

[**1.** **Antecedentes** 3](#_Toc139886664)

[**2.** **Planteamiento del Problema** 3](#_Toc139886665)

[a) Problema 3](#_Toc139886666)

[b) Justificación 3](#_Toc139886667)

[c) Alcance 4](#_Toc139886668)

[**3.** **Objetivos** 4](#_Toc139886669)

[a) Objetivo general: 4](#_Toc139886670)

[b) Objetivos específicos: 4](#_Toc139886671)

[**4.** **Marco Teórico** 4](#_Toc139886672)

[**5.** **Desarrollo de Solución** 5](#_Toc139886673)

[a) Análisis de Factibilidad (técnico, económica, operativa, social, legal, ambiental) 5](#_Toc139886674)

[b) Metodología de implementación (Documento de VISION, SRS, SAD) 7](#_Toc139886675)

[**6.** **Cronograma** 8](#_Toc139886676)

[**7.** **Conclusiones** 9](#_Toc139886677)

[**8.** **Anexos** 9](#_Toc139886678)

# **Antecedentes**

En la era digital actual, los bancos y las instituciones financieras están buscando constantemente formas de mejorar la eficiencia, la accesibilidad y la experiencia del cliente. Una forma de lograrlo es mediante el uso de API (Application Programming Interface) de servicios. Una API de servicios para un banco permite la integración de sistemas y aplicaciones de terceros para acceder y utilizar los servicios bancarios de manera segura y conveniente.

# **Planteamiento del Problema**

El problema que se plantea es la falta de una API de servicios robusta y completa en el banco. Esto limita la capacidad del banco para integrarse fácilmente con otras plataformas y aplicaciones, lo que puede resultar en una experiencia del cliente deficiente y una pérdida de oportunidades comerciales. Además, sin una API de servicios adecuada, el desarrollo de nuevas funcionalidades y la implementación de soluciones innovadoras se vuelven más difíciles y costosos.

## Problema

El problema principal es la falta de una API de servicios bancarios que proporcione una interfaz estandarizada, segura y de fácil uso para que otras aplicaciones y sistemas se integren con los servicios del banco. Esto limita la capacidad del banco para ofrecer servicios digitales de calidad, afectando la experiencia del cliente y restringiendo el potencial de crecimiento y competitividad.

## Justificación

La implementación de una API de servicios para el banco tiene varias justificaciones importantes. En primer lugar, permitiría al banco ofrecer una experiencia más integrada y conveniente a sus clientes, ya que podrían acceder a los servicios bancarios desde aplicaciones y plataformas de terceros que ya utilizan en su vida diaria. Esto mejoraría la satisfacción del cliente y aumentaría su lealtad hacia el banco.Además, una API de servicios robusta y completa facilitaría la colaboración con empresas fintech y otras instituciones financieras, permitiendo la creación de nuevas soluciones innovadoras y la exploración de oportunidades de negocio conjunto. También podría agilizar los procesos internos del banco al permitir la integración de sistemas y aplicaciones internas, mejorando la eficiencia operativa y reduciendo los costos.

## Alcance

El alcance de la API de servicios bancarios incluirá la creación de una interfaz segura y documentada que permita a terceros acceder a servicios bancarios clave, como consultas de saldo, transferencias, pagos, solicitud de productos financieros, entre otros. La API deberá garantizar la autenticación y autorización adecuada, así como la protección de la información confidencial de los clientes.

# **Objetivos**

## Objetivo general:

Implementar una API de servicios bancarios que permita la integración de sistemas y aplicaciones externas, mejorando la experiencia del cliente y fomentando la innovación y colaboración en el banco.

## Objetivos específicos:

* Diseñar y desarrollar una API de servicios bancarios segura y robusta.
* Documentar claramente la API y proporcionar ejemplos de uso.
* Establecer mecanismos de autenticación y autorización para garantizar la seguridad de la API.
* Integrar la API con los sistemas internos del banco.
* Realizar pruebas exhaustivas de la API para garantizar su funcionamiento correcto y confiabilidad.

# **Marco Teórico**

En esta sección, se proporciona información relevante sobre los conceptos teóricos y tecnologías relacionadas con el desarrollo de la API de servicios bancarios.

**Patrón de diseño por servicios:**

El patrón de diseño por servicios es una estrategia arquitectónica que se utiliza para diseñar sistemas de software distribuidos. Se basa en la idea de dividir la funcionalidad en servicios independientes y autónomos, que pueden comunicarse entre sí a través de interfaces bien definidas. Cada servicio se centra en una tarea específica y puede ser desarrollado, probado y desplegado de manera independiente. Esto facilita la escalabilidad, el mantenimiento y la evolución del sistema en general.

**Patrones de microservicios:**

Los microservicios son un enfoque para el diseño de arquitecturas de software donde una aplicación se divide en componentes pequeños e independientes llamados microservicios. Cada microservicio se enfoca en una función o tarea específica y se comunica con otros microservicios a través de mecanismos como APIs RESTful. Los patrones de microservicios proporcionan pautas y buenas prácticas para desarrollar, desplegar y gestionar estos componentes de manera efectiva, incluyendo la escalabilidad, la tolerancia a fallos y la recuperación.

**API en Python Django:**

Python Django es un framework de desarrollo web de alto nivel que utiliza el lenguaje de programación Python. Proporciona una estructura y conjunto de herramientas para desarrollar aplicaciones web de manera eficiente y escalable. Al construir una API en Python Django, se puede aprovechar su potente sistema de enrutamiento, modelos de datos, vistas y autenticación para crear endpoints que expongan la funcionalidad de la aplicación de manera segura y consistente.

**Verificación de deudas asociadas a servicios básicos:**

La verificación de deudas asociadas a servicios básicos implica el acceso y consulta de la información sobre los saldos pendientes de pagos de servicios como agua, luz, educación, internet y telefonía. La API desarrollada en este proyecto permitirá a los usuarios acceder a esta información de manera rápida y precisa, mejorando la experiencia del cliente y optimizando las operaciones de las entidades financieras involucradas. Esto implica la integración con sistemas de facturación y gestión de pagos, así como la implementación de mecanismos de seguridad y autenticación adecuados.

# **Desarrollo de Solución**

En esta sección, se abordará el desarrollo de la solución para la API de Servicios, comenzando con el análisis de factibilidad en diferentes áreas clave.

## Análisis de Factibilidad (técnico, económica, operativa, social, legal, ambiental)

El análisis de factibilidad es fundamental para evaluar la viabilidad del proyecto desde diferentes perspectivas. A continuación, se presenta un análisis exhaustivo en diferentes aspectos

**Factibilidad Técnica:** Se llevó a cabo un análisis detallado de la factibilidad técnica para evaluar si el desarrollo de la API de Servicios era viable. Durante este análisis, se examinaron diversos aspectos, como el nivel de conocimiento del equipo de desarrollo, la disponibilidad de herramientas y tecnologías necesarias, y la capacidad de integración con los sistemas existentes. Además, se consideraron la escalabilidad y el rendimiento de la solución, asegurándose de que la API de Servicios pudiera manejar la carga esperada y cumplir con los requisitos de tiempo de respuesta establecidos. Este análisis exhaustivo permitió determinar si el desarrollo de la API de Servicios era técnicamente factible y proporcionó una base sólida para su implementación.

**Factibilidad Económica:** El análisis de factibilidad económica se centró en evaluar los costos y beneficios relacionados con el desarrollo e implementación de la API de Servicios. Se examinaron detalladamente los costos asociados con recursos humanos, infraestructura, licencias de software y mantenimiento a largo plazo. Se consideraron cuidadosamente los beneficios esperados, como la mejora en la eficiencia operativa, la reducción de costos en la gestión de usuarios y el impacto positivo en la satisfacción y retención de los clientes. También se llevó a cabo un análisis de retorno de inversión (ROI) para determinar la viabilidad financiera del proyecto. Este análisis proporcionó una evaluación integral de los aspectos económicos del desarrollo de la API de Servicios, permitiendo tomar decisiones fundamentadas sobre su implementación.

**Factibilidad Operativa:** Se evaluó la factibilidad operativa del desarrollo de la API de Servicios, considerando los recursos internos disponibles, como el personal y la infraestructura. Se analizó la capacidad del equipo de desarrollo para llevar a cabo el proyecto dentro de los plazos establecidos y cumplir con los objetivos establecidos. Se consideraron los procesos y procedimientos operativos existentes, asegurando que la API de Servicios se pueda integrar sin interrupciones significativas en la operación actual.

**Factibilidad Social:** La factibilidad social se enfocó en evaluar el impacto y la aceptación de la API de Servicios en el contexto social. Se consideraron las necesidades y expectativas de los usuarios finales, así como los requisitos de cumplimiento normativo y las regulaciones de protección de datos personales. Se evaluó el impacto en la experiencia del usuario, la confianza en la plataforma y la aceptación por parte de los usuarios

**Factibilidad Legal:** Se llevó a cabo un análisis de factibilidad legal para garantizar el cumplimiento de las leyes y regulaciones aplicables al desarrollo de la API de Servicios. Esto incluye normativas financieras, protección de datos personales, seguridad de la información y otras regulaciones relevantes. Es fundamental realizar un análisis exhaustivo de los requisitos legales pertinentes y asegurarse de que la API cumpla con ellos, implementando las medidas necesarias para proteger la privacidad y seguridad de los datos, así como garantizar la confidencialidad y el cumplimiento de los estándares legales y regulatorios vigentes. Además, se deben considerar los acuerdos de licencia de software y las restricciones de propiedad intelectual para evitar posibles conflictos legales.

**Factibilidad Ambiental:** Se realizó un análisis de factibilidad ambiental para evaluar los posibles impactos ambientales derivados del desarrollo y la operación de la API de Servicios. Se consideraron aspectos como el consumo de recursos, la generación de residuos y la adopción de prácticas sostenibles. Se buscaron oportunidades para minimizar el impacto ambiental y promover la responsabilidad social corporativa.

## Metodología de implementación (Documento de VISION, SRS, SAD)

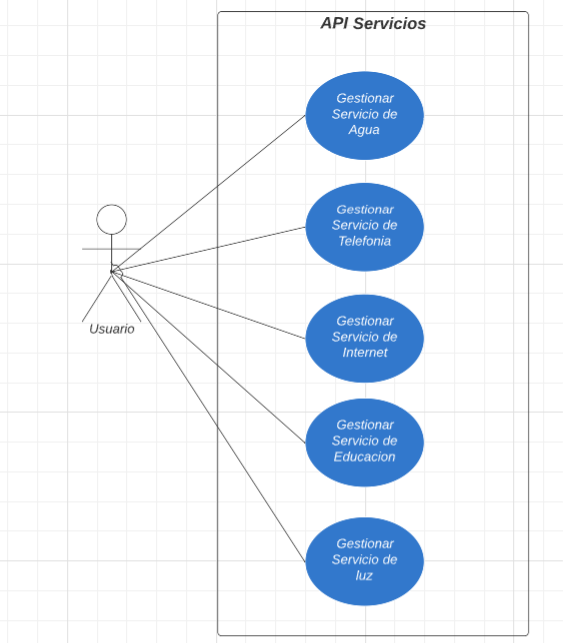
En esta sección, se describirán las metodologías y documentos utilizados durante la implementación de la API de Usuarios.

**Documento de Visión:** El Documento de Visión es un documento que describe la visión general del proyecto y los objetivos que se pretenden lograr con la implementación de la API de Servicios. Proporciona una descripción detallada de las funcionalidades esperadas, los requisitos del sistema y los beneficios que se esperan obtener. Además, establece los límites y alcance del proyecto, identificando los principales interesados y las restricciones a tener en cuenta durante el desarrollo.

**Especificación de Requisitos de Software (SRS):** La Especificación de Requisitos de Software (SRS) es un documento detallado que describe los requisitos funcionales y no funcionales de la API de Servicios. Proporciona una lista exhaustiva de los casos de uso, los escenarios de interacción y las características específicas que se deben implementar. También incluye restricciones técnicas, requisitos de rendimiento, seguridad y calidad. El SRS sirve como una guía para el desarrollo, asegurando que se cumplan todos los requisitos establecidos.

**Diseño de Arquitectura de Software (SAD):** El Diseño de Arquitectura de Software (SAD) describe la estructura y organización de la API de Servicios. Incluye la arquitectura de componentes, los patrones de diseño utilizados, la interacción entre los módulos y los flujos de datos. También aborda consideraciones de rendimiento, seguridad y escalabilidad. El SAD es esencial para garantizar un diseño eficiente y modular que cumpla con los requisitos.

DIAGRAMA DE CASOS DE USO



# **Cronograma**

Gráfico, Escala de tiempo

Descripción generada automáticamente

# **Conclusiones**

La implementación de una API de servicios bancarios proporciona numerosos beneficios, tanto para el banco como para sus clientes y socios comerciales. Permite una mayor integración y conveniencia para los clientes, mejora la eficiencia operativa y brinda oportunidades para la innovación y la colaboración en el sector financiero. Con una API bien diseñada y segura, el banco puede posicionarse como líder en servicios digitales y adaptarse a las demandas cambiantes del mercado.

# **Anexos**

Anexo 1. Principios de Diseño

Anexo 2. Patrones de Diseños Estructurales, de Comportamiento, Creacionales

Anexo 3. Patrones de Diseño de presentación o de microservicios

Anexo 4. <https://github.com/UPT-FAING-EPIS/proyecto-unidad-iii-pds-api_servicios>