



**UNIVERSIDAD PRIVADA DE TACNA**

**FACULTAD DE INGENIERIA**

**Escuela Profesional de Ingeniería de Sistemas**

**“API BANCA - PRODUCTOS”**

Curso: Patrones de Software

Docente: Patrick Cuadros Quiroga

Integrantes:

***Limache Durand, Rodrigo Jeral***      ***(2017059278)***

**Tacna – Perú**  
**2023**

CONTROL DE VERSIONES					
Versión	Hecha por	Revisada por	Aprobada por	Fecha	Motivo
1.0	JCG-AVV-ARR	ELV	ELV	23/05/2023	Versión 1

# API BANCA - PRODUCTOS

**Versión 1.0**

CONTROL DE VERSIONES					
Versión	Hecha por	Revisada por	Aprobada por	Fecha	Motivo
1.0	Rodrigo Limache	PCQ	PCQ	17/06/2023	Versión 1

## INDICE GENERAL

INTRODUCCIÓN .....	¡Error! Marcador no definido.
I. Generalidades de la Empresa.....	4
1. Nombre del proyecto .....	4
2. Introducción.....	4
1. Marco Teórico.....	4
CONCLUSIONES.....	5

## **I. Generalidades de la Empresa**

### **1. Nombre del proyecto**

Api Productos V.2.0

### **2. Introducción**

En el mundo actual, la industria bancaria está experimentando una transformación digital sin precedentes. La creación de APIs se ha convertido en una práctica común para permitir la integración de servicios financieros con aplicaciones y sistemas externos. En este contexto, el lenguaje de programación C# se destaca como una excelente opción para desarrollar un API de Banca Productos.

C# es un lenguaje potente y versátil, ampliamente adoptado en la industria y con un fuerte respaldo por parte de Microsoft. Su sintaxis clara y orientada a objetos facilita el desarrollo de aplicaciones robustas y escalables. Además, el ecosistema .NET proporciona una amplia gama de bibliotecas y frameworks que agilizan el proceso de desarrollo y ofrecen soluciones probadas.

Al optar por C# para desarrollar tu API de Banca Productos, te beneficiarás de una gran cantidad de recursos y documentación disponible, así como una comunidad activa de desarrolladores dispuestos a compartir sus conocimientos y experiencias. Además, la estrecha integración de C# con el entorno Microsoft te permitirá aprovechar al máximo las tecnologías y herramientas existentes, como Azure y Visual Studio, para una implementación más eficiente y una administración simplificada.

La seguridad y confiabilidad son aspectos críticos en el ámbito bancario, y C# ofrece características y herramientas que ayudan a garantizar la protección de los datos sensibles. Con mecanismos de autenticación sólidos y la capacidad de implementar cifrado y prácticas de seguridad recomendadas, podrás asegurar la integridad de las transacciones y la confidencialidad de la información financiera de tus usuarios..

### **1. Marco Teórico**

A través de las indagaciones realizadas a la empresa de estacionamiento “San José” ubicado en Arica, se encontró falencias en los procesos que se cumplen en el momento de gestionar la información que corresponden a: vehículos, datos específicos de los empleados y clientes.

- **Lenguaje de programación C#:** El lenguaje de programación C# es ampliamente utilizado en el desarrollo de aplicaciones en el entorno Microsoft. Proporciona una sintaxis clara y orientada a objetos, lo que lo convierte en una opción popular para desarrollar aplicaciones en el ecosistema .NET.
- **API (Interfaz de Programación de Aplicaciones):** Un API es un conjunto de reglas y protocolos que permite la interacción entre diferentes aplicaciones y servicios. En el contexto de la banca de productos, un API bancario proporciona funciones y servicios específicos relacionados con cuentas, transacciones, pagos, transferencias y otras operaciones financieras.
- **HTTP (Protocolo de Transferencia de Hipertexto):** El protocolo HTTP es el protocolo base utilizado para la comunicación en la web. Los APIs bancarios suelen utilizar solicitudes HTTP (como GET, POST, PUT, DELETE) para realizar operaciones y recibir respuestas.

- Cliente HTTP: Para interactuar con el API de Banca Productos, se utiliza un cliente HTTP. En C#, la clase HttpClient proporciona funcionalidades para enviar solicitudes HTTP a un servidor y recibir respuestas.
- Autenticación y seguridad: Los APIs bancarios suelen requerir autenticación para garantizar la seguridad y protección de los datos financieros. Las claves de API, tokens de acceso u otros mecanismos de autenticación se utilizan para validar las solicitudes y controlar el acceso a los servicios bancarios.
- Serialización de datos: En el contexto de las solicitudes y respuestas de un API, la serialización de datos es importante. En C#, se utiliza frecuentemente JSON (JavaScript Object Notation) para serializar y deserializar objetos entre el formato de objeto en memoria y el formato de intercambio de datos.
- Manejo de errores: Es crucial implementar un manejo adecuado de errores al interactuar con un API bancario. Esto implica capturar y gestionar excepciones, así como verificar los códigos de respuesta HTTP para determinar si la solicitud se ha completado correctamente o si ha ocurrido algún error.
- Documentación del API: Es esencial revisar la documentación oficial del API de Banca Productos. Esta documentación proporcionará detalles sobre los endpoints disponibles, los parámetros requeridos, las estructuras de datos de entrada y salida, así como las mejores prácticas recomendadas para su uso.

## CONCLUSIONES

Se definió el porque trabajar de esta manera el proyecto:

- Potencia y versatilidad de C#: C# es un lenguaje de programación poderoso y flexible que brinda una amplia gama de funcionalidades para el desarrollo de APIs. Su sintaxis clara y orientada a objetos facilita la creación de soluciones robustas y escalables.
- Respaldo de Microsoft y comunidad activa: C# cuenta con el respaldo de Microsoft y una comunidad de desarrolladores activa que comparte conocimientos y experiencia. Esto significa que encontrarás una gran cantidad de recursos, documentación y soporte para ayudarte en el desarrollo de tu API de Banca Productos.
- Integración con el ecosistema Microsoft: Si tu proyecto se basa en el entorno Microsoft, utilizar C# te permitirá aprovechar las tecnologías y herramientas existentes, como Azure y Visual Studio. Esto simplificará el proceso de desarrollo y te brindará ventajas adicionales en términos de administración y despliegue.
- Seguridad y confiabilidad: La seguridad y confiabilidad son fundamentales en el ámbito bancario. C# ofrece características y herramientas que te ayudarán a garantizar la protección de los datos sensibles y a implementar prácticas de seguridad sólidas.