



CAMM

“CAJERO AUTOMÁTICO
MULTI MONEDA”

Miguel Yucra

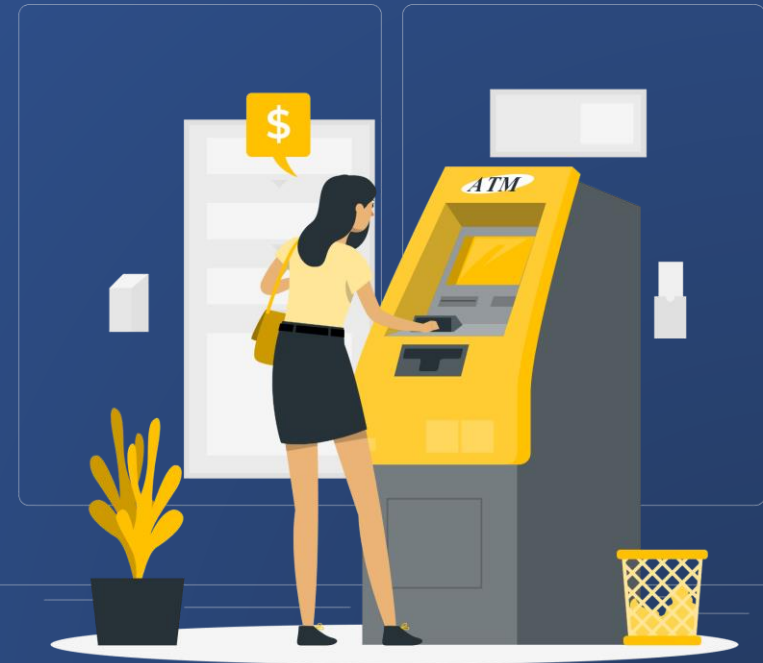
Enero, 2024

Agenda

- 1 Visión de arquitectura
- 2 Atributos de calidad
- 3 Escenarios
- 4 Vistas de arquitectura

1

Visión de Arquitectura





Contexto

El Banco **VIZZA** ha establecido una estrategia de modernización tecnológica para mejorar su plataforma de cajeros automáticos. La decisión de modernizar surge de la necesidad de adaptarse a las actuales demandas tecnológicas y permitir un escalado eficiente, ya que la infraestructura actual no satisface adecuadamente estas exigencias. Esta limitación impacta directamente en la capacidad del banco para atraer nuevos clientes, afectando, por ende, la rentabilidad del negocio.

Tras un exhaustivo análisis de impacto, se optó por llevar a cabo el desarrollo in-house del nuevo sistema. Además, se ha decidido seguir un enfoque de implementación incremental para maximizar la eficacia del proceso. El inicio de esta fase se centrará en los siguientes módulos clave:

- Gestión de cuentas
- Gestión de transacciones
- Gestión de tasas de cambio.

Este enfoque permitirá al Banco VIZZA adaptarse progresivamente a las nuevas exigencias del mercado, garantizando una transición suave y controlada hacia una plataforma más moderna y eficiente.

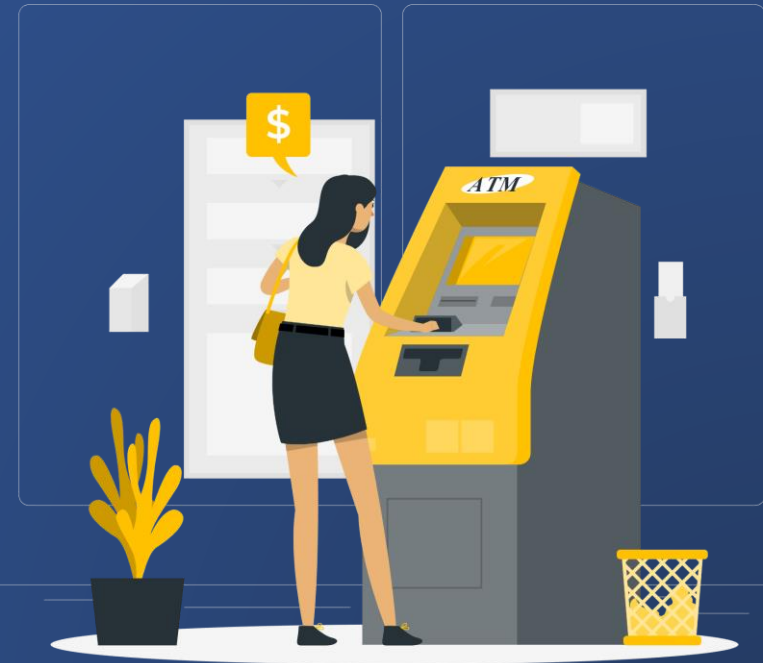


Visión de Arquitectura

Diseñar e implementar una solución tecnológica de cajero automático flexible y escalable brindando una experiencia de usuario mejorada y adaptada a las preferencias individuales, posicionándolo como un punto de innovación tecnológica. Esto contribuirá a la fidelización de los clientes y a la atracción de nuevos usuarios, fortaleciendo la posición competitiva del banco.

2

Atributos de Calidad



Atributos de Calidad

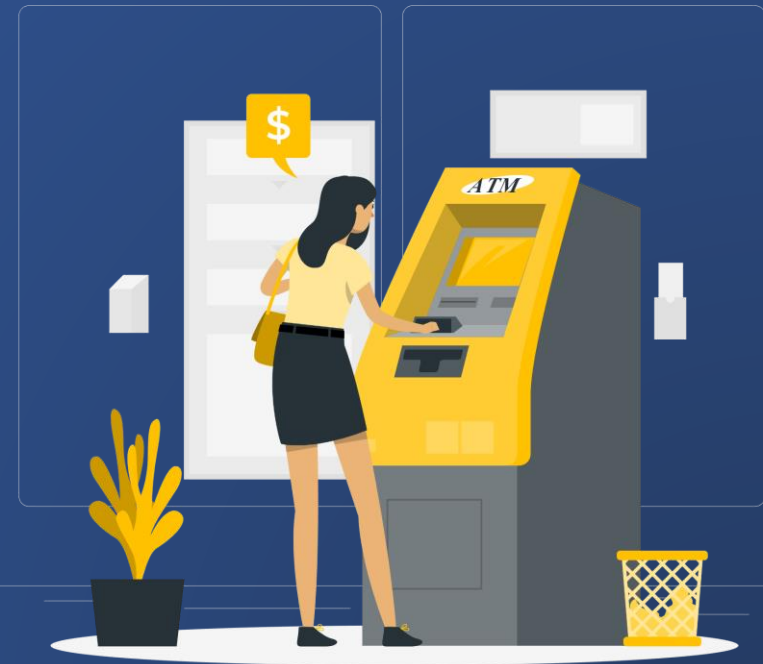
ATRIBUTO	DESCRIPCIÓN
Seguridad	El sistema debe mantener criterios de seguridad en un cajero automático debido a la naturaleza sensible de las transacciones financieras. Se deben implementar mecanismos robustos de autenticación (Oauth2) y autorización (JWT), además de protección contra amenazas de manipulación física.
Disponibilidad	El sistema debe mantener la disponibilidad continua del cajero automático para garantizar que los clientes puedan acceder a sus fondos en cualquier momento. Se utilizará técnicas como la replicación de datos, distribución geográfica (si es necesario) y la implementación de servidores en clúster para garantizar la disponibilidad continua incluso en situaciones de fallos.
Usabilidad	El sistema debe contar con una interfaz de usuario intuitiva y fácil de usar contribuye a una experiencia positiva del cliente. Realizar pruebas de usabilidad y recopilación de comentarios de usuarios para ajustar y mejorar continuamente la interfaz. Implementar opciones de personalización basadas en perfiles de usuario y preferencias.
Desempeño	El sistema debe contar con un buen rendimiento para minimizar los tiempos de espera durante las transacciones. Optimizar consultas de base de datos, utilizar almacenamiento en caché eficiente y aplicar técnicas de compresión para reducir el tiempo de respuesta. Escalar horizontalmente los componentes críticos para distribuir la carga y mantener un rendimiento consistente bajo cargas variables.
Escalabilidad	Diseñar la arquitectura para ser escalable horizontalmente, dividiendo funcionalidades en microservicios independientes y utilizando tecnologías de contenedores y orquestación de contenedores. Implementar soluciones de escalabilidad automática para adaptarse a las fluctuaciones en la carga de trabajo

Atributos de Calidad

ATRIBUTO	DESCRIPCIÓN
Integridad de datos	El sistema debe implementar mecanismos de validación de datos, restricciones de integridad en la base de datos y auditorías para asegurar que los datos almacenados sean precisos y no estén sujetos a manipulación no autorizada. Utilizar transacciones y controles de concurrencia para mantener la coherencia de los datos.
Mantenibilidad	El sistema debe adoptar principios de desarrollo limpio, modularidad y documentación detallada del código. Utilizar herramientas de CI/CD para facilitar despliegues y actualizaciones continuas. Aplicar patrones de diseño que permitan la fácil incorporación de nuevas características y la modificación de funcionalidades existentes.
Interoperabilidad	El sistema debe adoptar estándares de la industria para la comunicación entre sistemas. Utilizar APIs bien definidas y protocolos de comunicación abiertos. Implementar una arquitectura hexagonal para facilitar la integración con otros sistemas bancarios y servicios financieros.
Confiabilidad	El sistema debe implementar técnicas de gestión de errores, registro detallado y observabilidad para identificar y abordar proactivamente problemas potenciales. Utilizar pruebas unitarias, de integración y estáticas para garantizar la confiabilidad del sistema en diferentes escenarios.
Cumplimiento normativo	La solución debe cumplir con regulaciones y estándares del sector financiero. Y debe estar diseñado para operar de acuerdo con normativas de seguridad y privacidad para garantizar la confianza de los usuarios y cumplir con requisitos legales.

3

Escenarios



Escenarios

	Descripción
Escenario	Orquestación de servicios
Justificación	La orquestación de servicios en una arquitectura de microservicios proporciona una forma efectiva de coordinar y gestionar la complejidad de aplicaciones empresariales distribuidas, mejorando la agilidad, la escalabilidad y la resiliencia del sistema
Atributos de calidad	Escalabilidad y distribución
Estimulo	Solicitud de procesamiento a un servicio
Respuesta	Ejecución ordenada de los servicios
Decisiones de arquitectura	<p>Decisión Tomada: Uso de Kubernetes para la Orquestación de Servicios, para coordina microservicios.</p> <ul style="list-style-type: none">• Ventajas: Escalabilidad, gestión de recursos, y orquestación integrada con la plataforma de contenedores.• Desventajas: Curva de aprendizaje, dependencia de la infraestructura de contenedores.
Medición	Tiempo de respuesta de los microservicios

Escenarios

	Descripción
Escenario	Integración con terceros
Justificación	La integración con terceros en una arquitectura de microservicios no solo es una necesidad, sino una estrategia inteligente para aprovechar especializaciones externas, acelerar el desarrollo, mejorar la adaptabilidad y permitir que el equipo se enfoque en el núcleo de negocio.
Atributos de calidad	Adaptabilidad, escalabilidad y evolución
Estimulo	Solicitud de operaciones
Respuesta	Proceso ejecutado y concluido de acuerdo a las necesidades
Decisiones de arquitectura	<p>Decisión Tomada: se usa una combinación de opciones para mejorar la seguridad</p> <ul style="list-style-type: none">• Nat Gateway, para proteger nuestra red privada con las redes externas• VPN, para establecer una comunicación directa y segura• API Gateway, para centralizar y gestionar las integraciones, y adaptar fácilmente los cambios de terceros
Medición	<p>Tiempos de respuesta</p> <p>Operaciones correctas e incorrectas</p> <p>Costos de implementación</p>

Escenarios

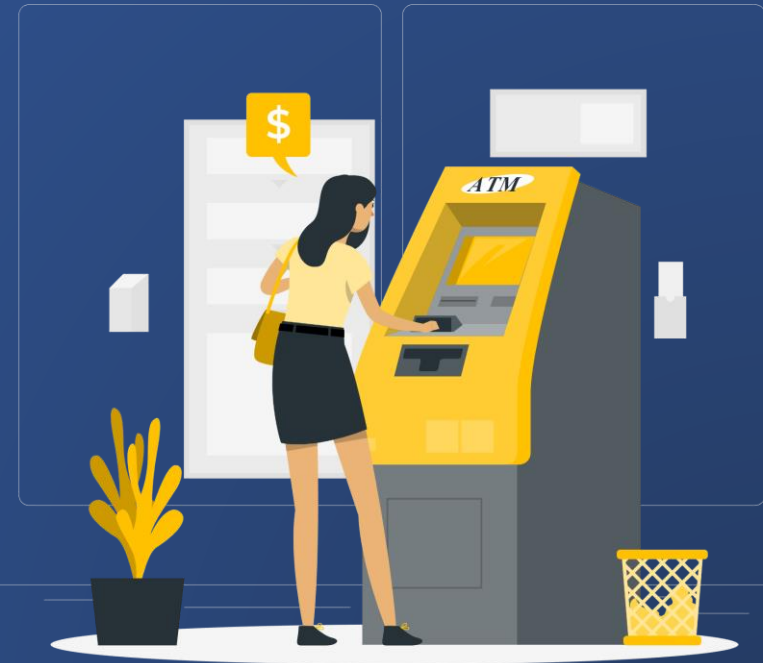
	Descripción
Escenario	Modelo de datos y pistas auditorias
Justificación	La implementación de un modelo de datos y pistas de auditoría en una arquitectura de microservicios con una base de datos compartida proporciona beneficios significativos en términos de consistencia de datos, trazabilidad, resolución de problemas y adaptabilidad a los cambios en el negocio.
Atributos de calidad	Cumplimiento normativo, consistencia de datos
Estimulo	Registro de datos en la base de datos
Respuesta	Datos guardados correctamente
Decisiones de arquitectura	Decisión Tomada: se utiliza las siguientes recomendaciones: <ul style="list-style-type: none">• Toda tabla no transaccional debe contar con su tabla gemela histórica• Toda tabla debe mantener al menos 5 campos de auditoria (fecha, usuario, ip, tipo y modulo)• Todo el modelo de base de datos debe mantener una integridad referencial
Medición	Numero de tablas con pistas de auditoria Numero de tablas con tablas históricas vs tablas que no guardan historia

Escenarios

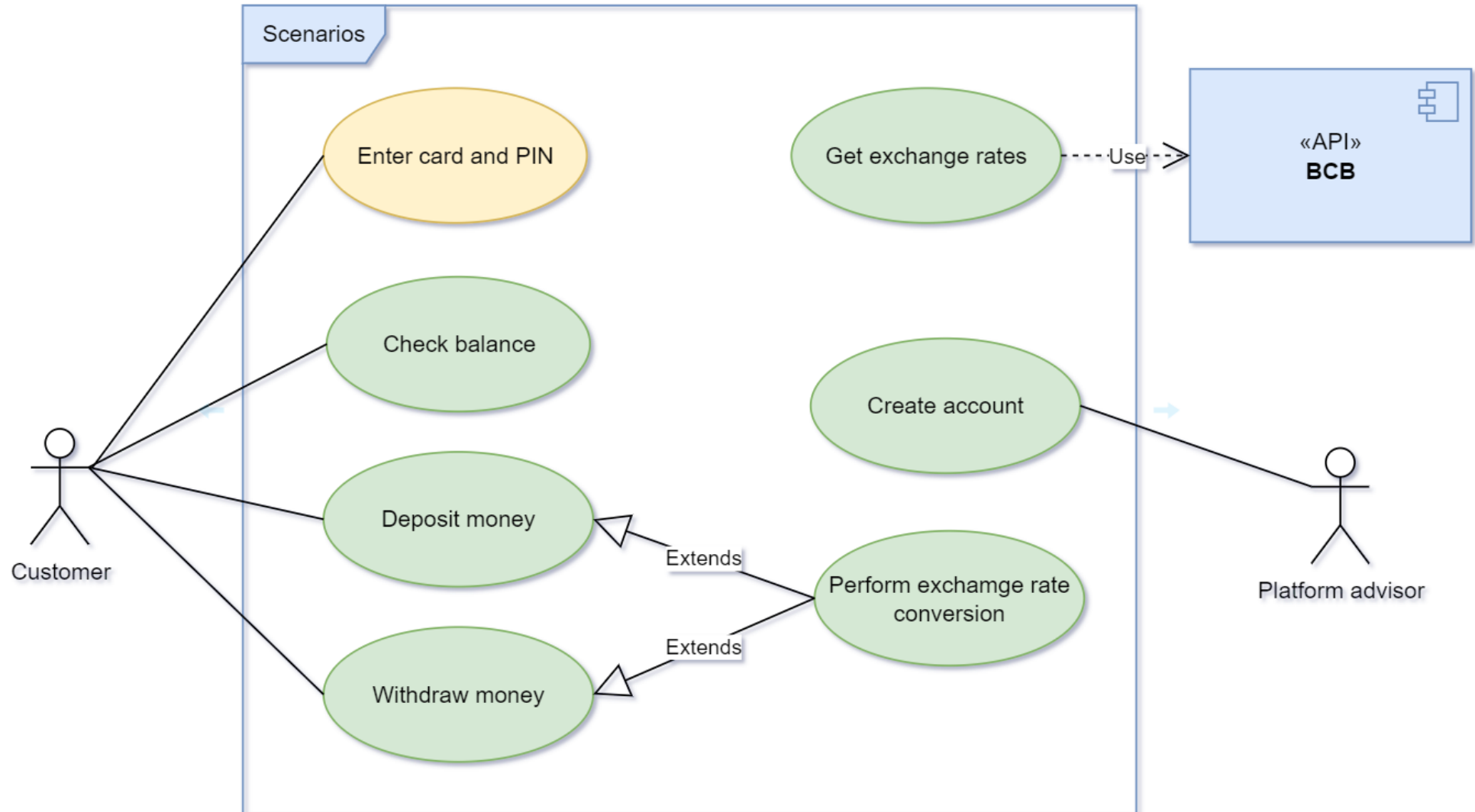
	Descripción
Escenario	Patrón de Diseño de software Arquitectura Hexagonal
Justificación	La adopción de la arquitectura hexagonal en un entorno de microservicios proporciona una serie de beneficios alineados con nuestros objetivos arquitectónicos. El desacoplamiento resultante mejora la flexibilidad, la testabilidad y la capacidad de adaptarse a cambios en servicios externos. Además, la estructura concéntrica facilita la comprensión del sistema y la colaboración en el equipo de desarrollo.
Atributos de calidad	Desacoplamiento, flexibilidad, testeabilidad
Estimulo	Nueva solicitud de desarrollo de un microservicio
Respuesta	Desarrollo de microservicio concluido
Decisiones de arquitectura	Decisión Tomada: <ul style="list-style-type: none">• Adoptar el patrón de diseño "Arquitectura Hexagonal"
Medición	Numero de microservicios desarrollados con Arquitectura Hexagonal

4

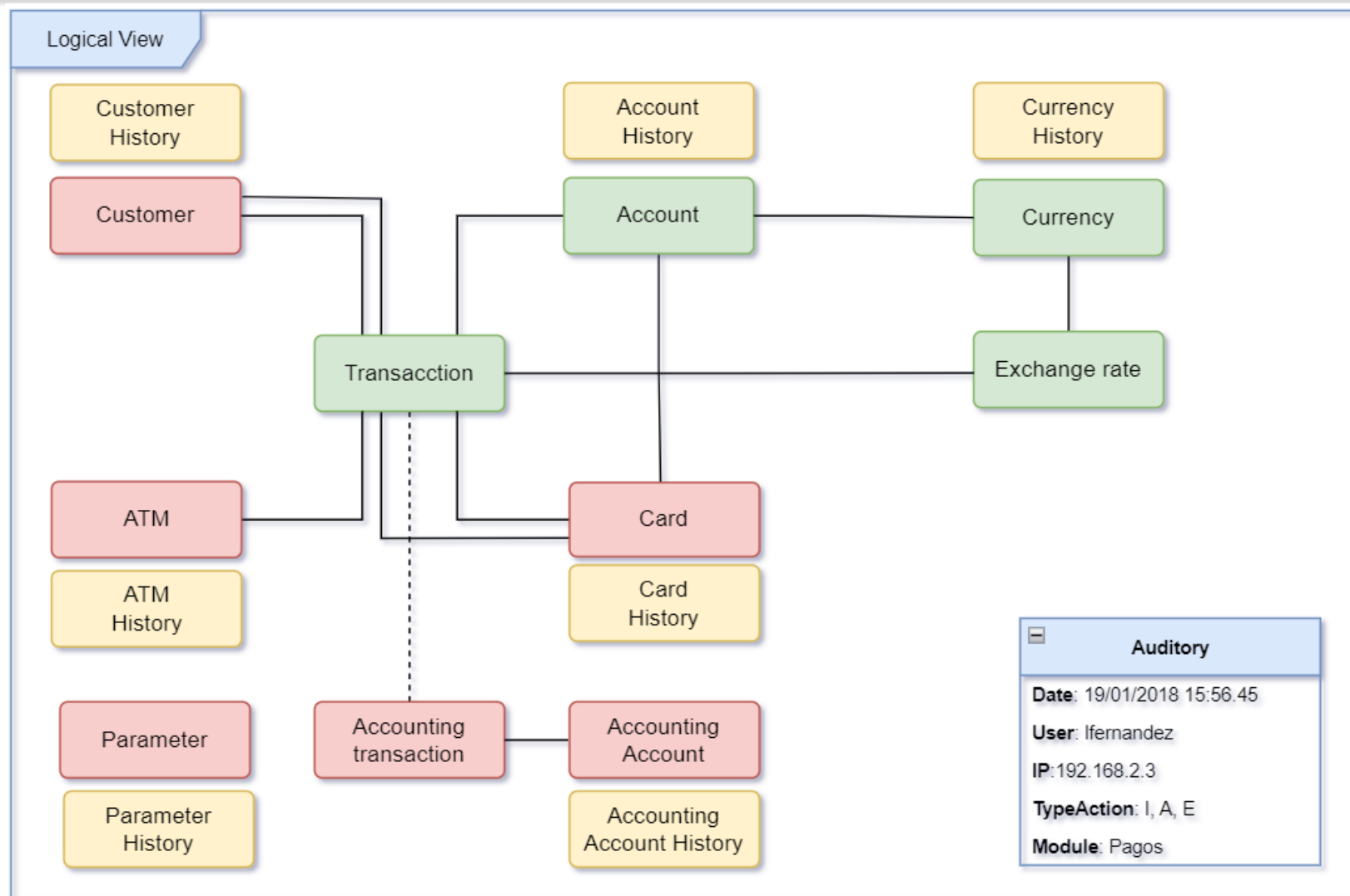
Vistas de Arquitectura



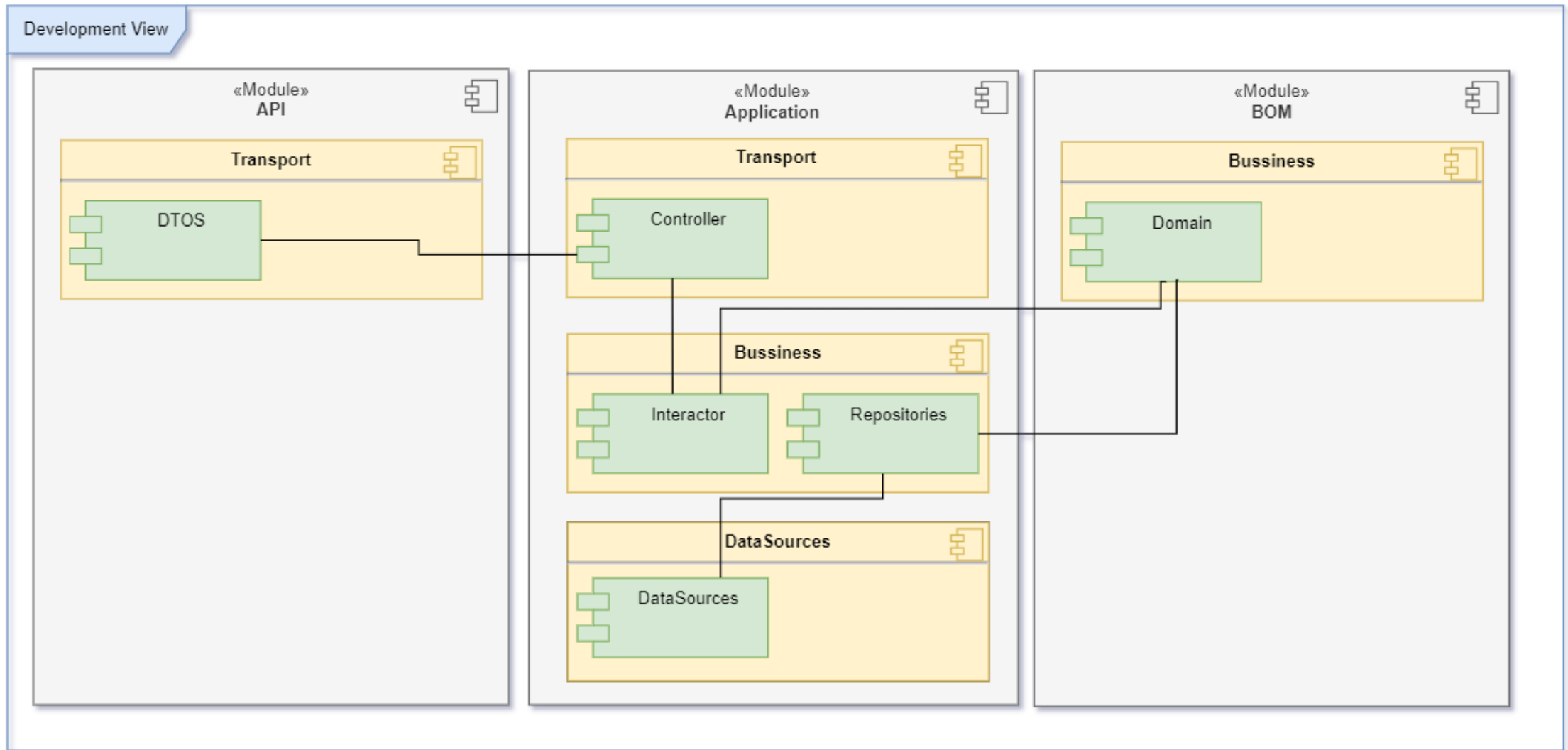
Casos de uso



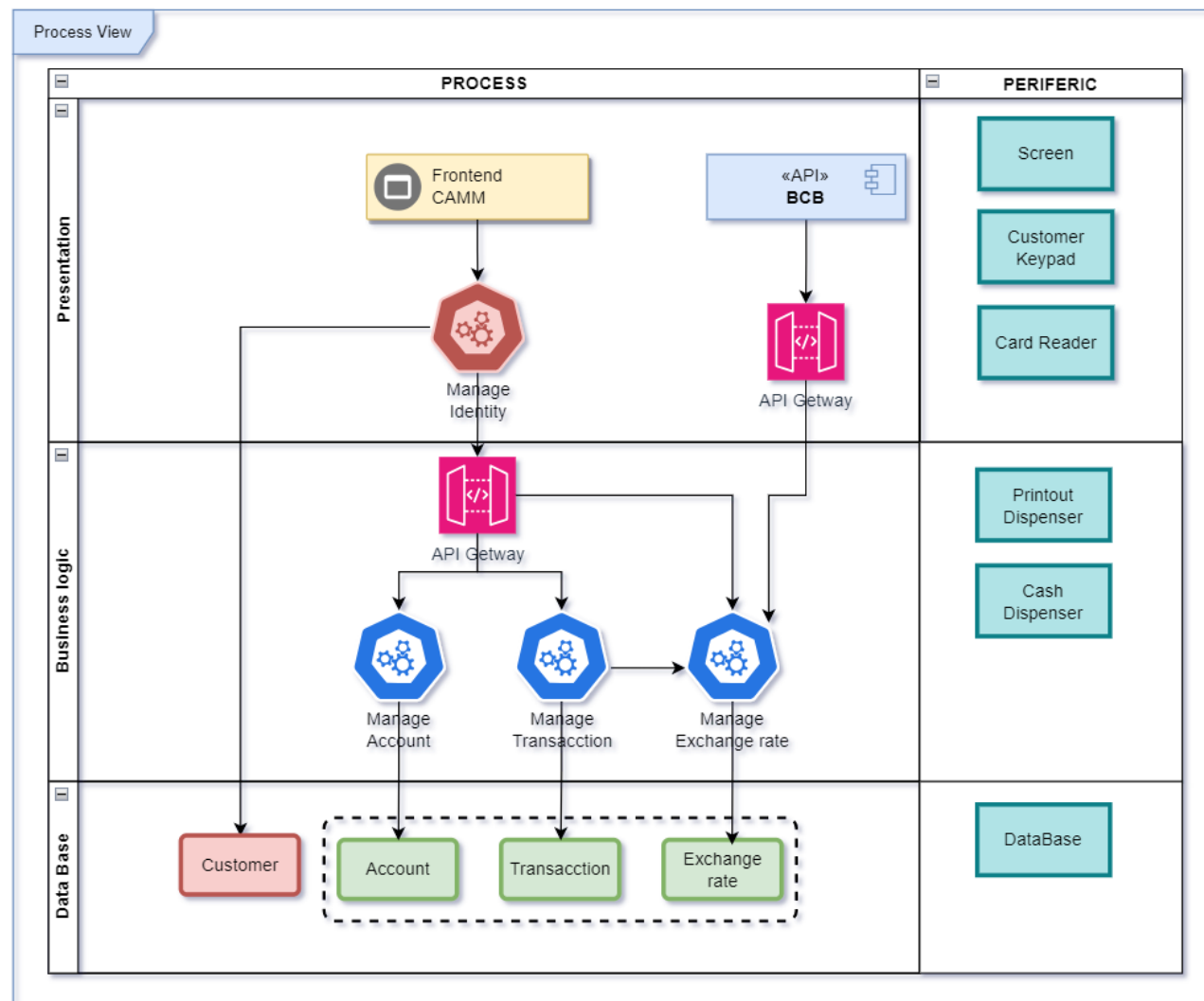
Vista lógica



Vista de Desarrollo

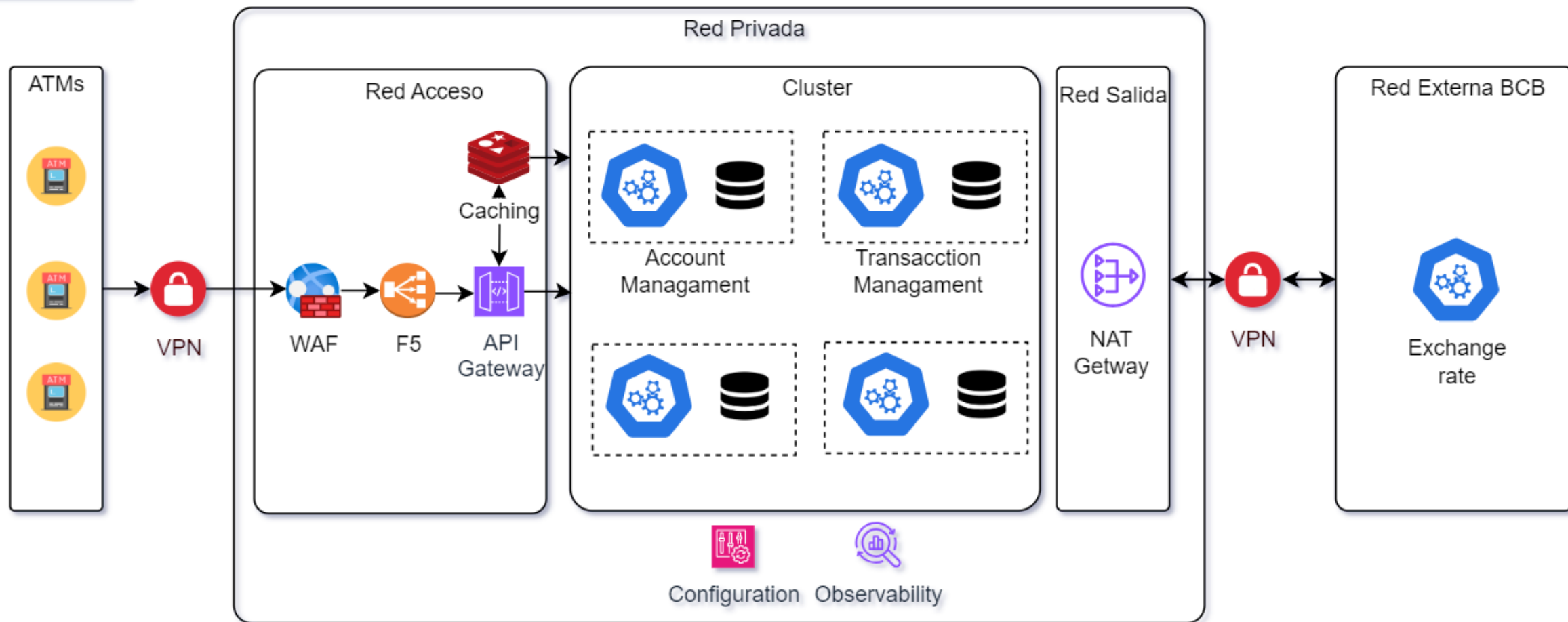


Vista de Proceso



Vista Física

Physical View



ii Gracias !!

Miguel Yucra

Enero, 2024

