

Cliente: BANCO BP

Documento de Arquitectura

Versión: 1.0
Fecha: 10-09-2024

	FORMATO			
	DOCUMENTO ARQUITECTURA			
	Código:	Versión:	Fecha:	

Documento aprobado por:

Elaborado por:	Aprobado por:
Juan Camilo García Hurtado _____ Arquitecto Cluod	 _____

Copyright
GFT Technologies

	FORMATO			
	DOCUMENTO ARQUITECTURA			
	Código:	Versión:	Fecha:	

Historial de Revisiones

Fecha	Versión	Autor:	Modificación
10/09/2024	1.0	Juan Camilo Garcia Hurtado	Creación de documento de arquitectura.

	FORMATO			
	DOCUMENTO ARQUITECTURA			
	Código:	Versión:	Fecha:	

Introducción

A lo largo de este documento se describen los diferentes elementos y las decisiones de diseño que constituyen la solución para el **Sistema de Banca BP** propuesta. Dadas las necesidades funcionales del solicitante, en este documento se muestran las consideraciones arquitectónicas que se deben tener en cuenta en el proceso de construcción de software.

Dicha arquitectura está basada en la definición de requerimientos y casos de uso, que reflejan las directrices del proceso de desarrollo y que se han extraído y consolidado de la documentación existente.

Así pues, la intención de este documento es capturar y comunicar las decisiones significativas de arquitectura que se tomaron para realizar el diseño del **sistema de Banca BP** y mostrar un camino factible para su implementación concreta. La solución en sí representa una aplicación ágil, segura y confiable para los usuarios, consumidores de la aplicación.

Alcance

Este documento es de interés de los solicitantes que están interesados en conocer la arquitectura de la solución propuesta. Para lo anterior, se proveen vistas que intentan brindar la información requerida desde diferentes ángulos. Así mismo, abarca gran parte de los diagramas elaborados en la fase de análisis, brindando un contexto y una explicación a sus relaciones.

Objetivo

Definir un diseño integral que soporte las funcionalidades requeridas, garantizando seguridad, cumplimiento normativo, alta disponibilidad, escalabilidad, y una experiencia de usuario óptima.

Propósito

Este documento de arquitectura provee una visión detallada de la arquitectura del sistema.

Su principal objetivo es capturar y expresar las decisiones más importantes que se han tomado sobre la arquitectura del sistema. Más específicamente, este documento describe y discute las soluciones de arquitectura que encaminaran las decisiones y el proceso de desarrollo durante el proyecto.

	FORMATO			
	DOCUMENTO ARQUITECTURA			
	Código:	Versión:	Fecha:	

Definiciones, Acrónimos y Abreviaturas

- **MVP producto mínimo viable**, Es una versión de un producto con las funciones suficientes para que lo puedan usar los primeros clientes que luego pueden proporcionar comentarios para el desarrollo futuro del producto, también se conoce como Arquitectura de Transición.
- **AWS Certificate Manager (ACM)**, Es un servicio de AWS que facilita la obtención, implementación y administración de certificados SSL/TLS para tus aplicaciones web que se ejecutan en AWS.
- **RDS Aurora Postgres**, es un servicio gestionado de base de datos que facilita la configuración, operación y escalado de bases de datos aurora en la nube de AWS.
- **VPC Virtual Private Cloud**, Amazon Virtual Private Cloud (Amazon VPC) le brinda control total sobre su entorno de redes virtuales, incluidas la ubicación de los recursos, la conectividad y la seguridad.
- **Amazon API Gateway**, Es un servicio de AWS para la creación, la publicación, el mantenimiento, el monitoreo y la protección de las API REST, HTTP y de WebSocket a cualquier escala.
- **Route 53**, Es un servicio de Sistema de Nombres de Dominio (DNS) escalable y altamente disponible ofrecido por Amazon Web Services (AWS).
- **WS EKS** Servicio administrado de Kubernetes que permite ejecutar aplicaciones basadas en contenedores.
- **Secrets Manager**, Permite almacenar, recuperar, controlar el acceso, rotar, auditar y monitorear datos confidenciales de manera central.
- **AWS Web Application Firewall**, Es un servicio de seguridad en la nube que ayuda a proteger las aplicaciones web de ataques comunes como la inyección de código SQL, el scripting entre sitios (XSS), ataques DDoS (de denegación de servicio distribuido) y otros..
- **Lambda Authorizer**, Un Lambda Authorizer es una función AWS Lambda que se utiliza para autenticar y autorizar solicitudes a las API en API Gateway.

	FORMATO		
	DOCUMENTO ARQUITECTURA		
	Código:	Versión:	Fecha:

Representación Arquitectónica

La arquitectura definida para el [Sistema de Banca BP](#) se basa en los principios de construcción. De acuerdo con los principios, todos los servicios de negocio se apoyan en una lista de servicios nativos dentro de la arquitectura, que buscan principalmente resolver la funcionalidad básica.

Arquitectura de Referencia AWS

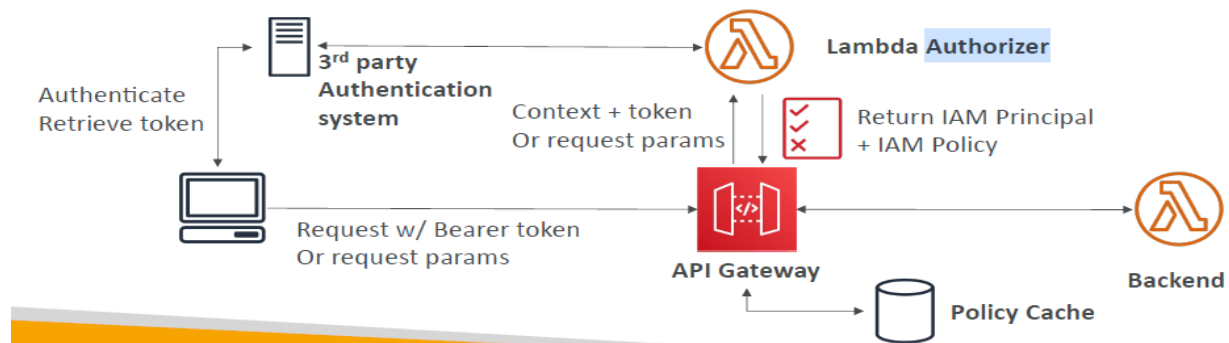
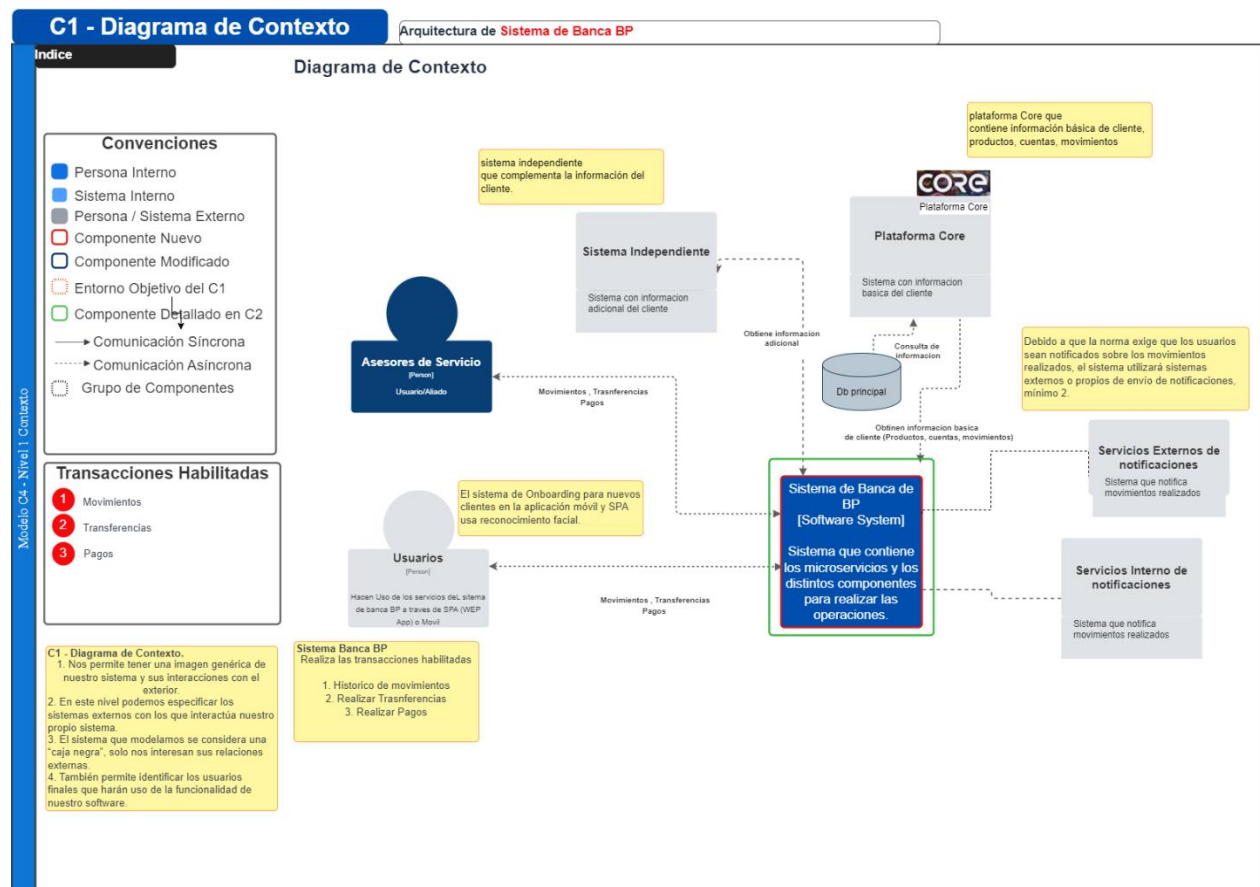


Diagrama C4 - Indice

Indice	
Arquitectura de Sistema de Banca BP	<p>Sistema de Banca BP</p> <p>#1 CRITERIOS DE ACEPTACIÓN (Resultado esperado) - Responsable Arquitecto - Juan Camilo García Hurtado</p> <p>Se requiere entregable donde se analice la arquitectura objetivo de la solución con el Sistema de Banca BP, donde podamos visualizar por estructura de negocio cual es el tratamiento que se debe dar a las transacciones, teniendo en cuenta el cumplimiento de normativo, lograr trazabilidad en el proceso y optimizar las actividades que actualmente se realizan en las transacciones.</p> <p>Diseño de la arquitectura objetivo release 1 donde se debe ver:</p> <ul style="list-style-type: none"> Diagrama de Contexto C1 - Vista de Negocio Diagrama de Contenedor C2 - Arquitectura de Referencia Diagrama de Componente C3 - Arquitectura de Despliegue Los mecanismos de autenticación OAuth2.0 Entrega y recepción de información a través de APIs y Microservicios
1. Diagrama de Contexto C1	
2. Diagrama de Contenedor C2	
3. Diagrama de Componentes C3	
4. Arq-Despliegue-HA-DR	

			FORMATO		
			DOCUMENTO ARQUITECTURA		
			Código:	Versión:	Fecha:

Diagrama de Contexto



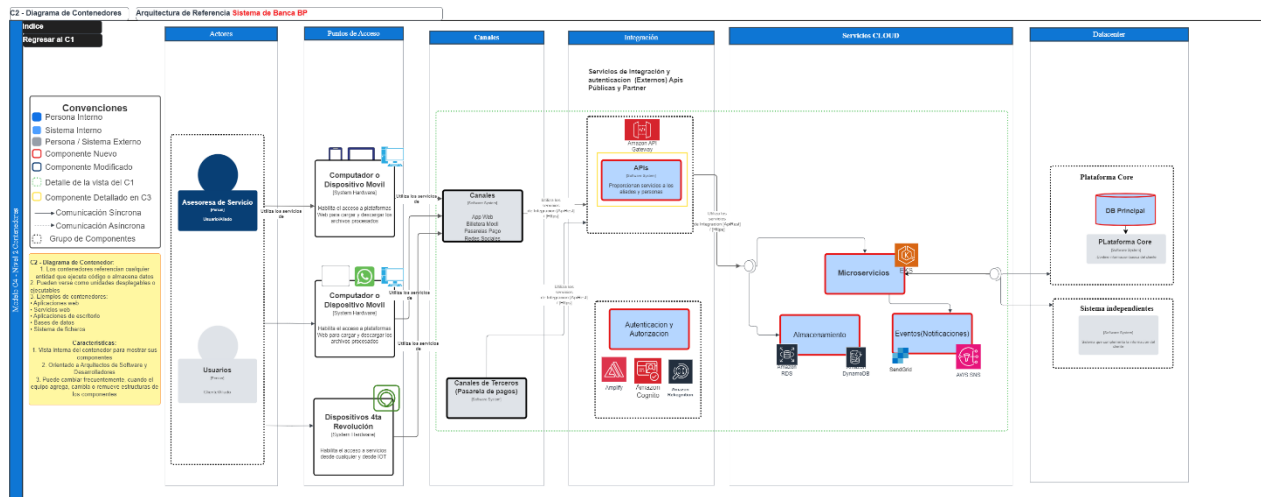
En el diagrama anterior se muestra la arquitectura a nivel de una Vista de Negocio en la cual se contemplan:

- ☐ **Usuarios finales (clientes):** Accederán al sistema mediante aplicaciones web SPA y móviles para consultar movimientos, realizar transferencias, y pagos interbancarios.
- ☐ **Plataforma Core:** Sistema principal de BP donde se almacena la información básica del cliente y movimientos.
- ☐ **Sistema Independiente:** Provee información detallada del cliente cuando sea necesario.
- ☐ **Servicios Externos:** Se conectarán para notificaciones, autenticación mediante Oauth2.0, y servicios de transferencia interbancaria.

FORMATO		
DOCUMENTO ARQUITECTURA		
Código:	Versión:	Fecha:

- ❑ **Reguladores:** Requisitos de normativas financieras y protección de datos.

Diagrama de Contenedor



Este diagrama de contenedor refleja una arquitectura distribuida y desacoplada, con alta disponibilidad y escalabilidad mediante el uso de contenedores en EKS y servicios administrados de AWS.

Los usuarios finales acceden al sistema mediante aplicaciones web o móviles. Estas interacciones se dirigen a través de un API Gateway que expone los servicios de microservicios desplegados en EKS. Los microservicios manejan operaciones críticas como transferencias, pagos y consultas de movimientos.

El sistema también está integrado con servicios de notificación, como SendGrid para correos externo y AWS SNS para notificaciones internas. Todo el sistema está respaldado por bases de datos relacionales y NoSQL, garantizando alta disponibilidad, resiliencia y velocidad en las operaciones.

FORMATO			
DOCUMENTO ARQUITECTURA			
Código:	Versión:	Fecha:	

Diagrama de componentes

Este diagrama de componentes detalla cómo los diferentes servicios y sistemas se interrelacionan, destacando la integración con API Gateway, la orquestación de microservicios, y la utilización de servicios como AWS Amplify, Amazon Cognito y Amazon Rekognition para la autenticación. Además, se resalta la integración con canales externos para facilitar pagos y el cumplimiento de requisitos normativos de notificación.

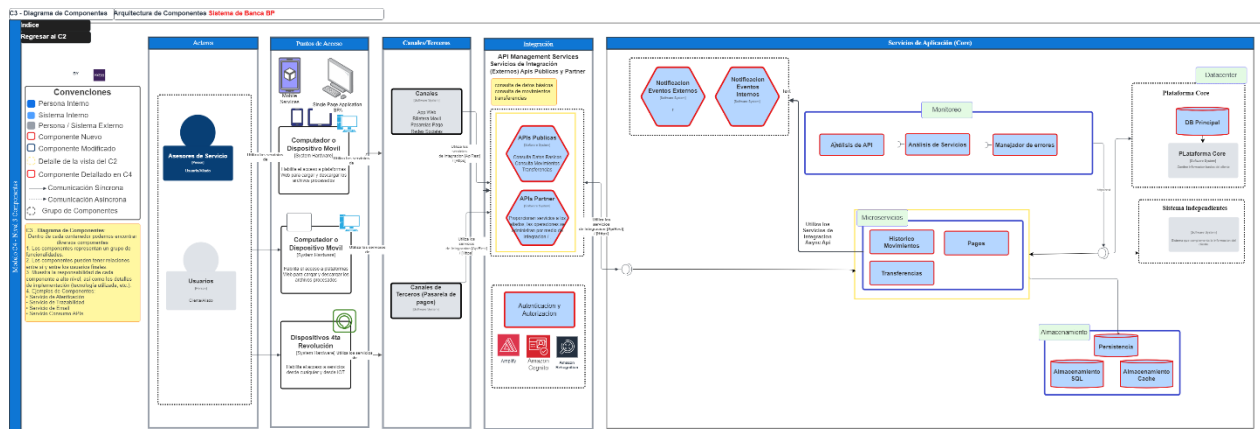


Diagrama de despliegue

En el siguiente diagrama se muestra una arquitectura de despliegue propuesta para soluciones containerizadas y administradas mediante servicios como EKS (Elastic Kubernetes Service) los cuales funcionan como servicios de orquestación de contenedores de AWS.

Esta arquitectura Garantiza, alta disponibilidad (HA), tolerancia a fallos, recuperación ante desastres (DR), Seguridad y Monitoreo, Excelencia operativa y auto-healing.

Dicha solución en nube, cuenta con una capa de autenticación y autorización en servicios serverless, los cuales ofrecen una combinación de beneficios enfocados a la simplicidad, escalabilidad, eficiencia de costos y rapidez de despliegue.

A nivel de backend se cuenta con una solución en Kubernetes la cual ofrece una combinación poderosa de flexibilidad, escalabilidad, seguridad y eficiencia, permitiendo desplegar y gestionar aplicaciones de manera efectiva en la nube. La integración profunda con los servicios de AWS potencia aún más las capacidades de Kubernetes, proporcionando una infraestructura robusta y altamente eficiente.

	FORMATO			
	DOCUMENTO ARQUITECTURA			
	Código:	Versión:	Fecha:	

Flujo de Autenticación Propuesto (OAuth2.0 Authorization Code Flow)

Se recomienda usar el Authorization Code Flow para mayor seguridad, ya que evita exponer los tokens directamente en el cliente. Este flujo es el más adecuado para aplicaciones SPA y móviles, y se complementa con autenticación multifactor.

Normativas

- ISO 27001: Estándar internacional de seguridad de la información.
- GDPR / Ley de Protección de Datos: Se debe garantizar que los datos de los usuarios sean manejados de forma segura y que existan mecanismos para que los usuarios puedan acceder y eliminar su información.
- PCI-DSS: Si se manejan pagos y transacciones financieras, el sistema debe cumplir con este estándar de seguridad para el manejo de datos de tarjetas.

VERSIÓN DEL FORMATO

Información de la versión

Revisado por		Fecha	
Aprobado por		Fecha	