**DOCUMENTO ARQUITECTURA DE LA SOLUCION**

**Universal Bank.**



Elaborado por:

Farid Eliecer Ureche López

Manuel Alejando Alcalá Bustos

|  |  |  |
| --- | --- | --- |
| **Versión** | **Fecha** | **Descripción de la modificación** |
| 0.1 | 6/11/2019 | Descripción Inicial |

Tabla de Contenido1. Introducción 4

1.1 Objetivo 4

1.2 Definiciones, Acrónimos y Abreviaturas 4

1.3 Referencias 4

1.4 Alcance 4

2. Contexto 5

2.1 Fundamentos de la solución 5

3. Drivers de Arquitectura 6

3.1 Objetivos de la arquitectura de la solución 6

3.2 Requerimientos Funcionales Significativos 6

3.3 Restricciones 7

3.4 Atributos de calidad 7

**3.4.1** **Descripción de atributos de calidad** 7

4. Vistas de arquitectura 8

4.1 Vistas de la arquitectura 8

4.1.1 Vista lógica 8

4.1.2 Vista de desarrollo 9

4.1.3 Vista de procesos 9

4.1.4 Vista física 10

Vista de casos de uso 10

4.1.5 10

4.1.6 Vista de datos 11

5. Riesgos de la arquitectura 12

6. ANEXO 12

6.1 Diagrama componentes aplicación SOBS, proveedores. 12

6.2 Diagrama de flujo ofertas a cotización. 12

# Introducción

Este documento muestra el proceso y resultado de la arquitectura de software que se implementara sobre el portal financiero en línea para el Universal Bank, teniendo en cuenta los distintos pasos de análisis y diseño relacionados en la creación de la arquitectura deseada.

El documento nos proporcionara todos los conceptos fundamentales o propiedades del sistema y su relación con su entorno. Esto nos lleva a una a la recopilación de todos los elementos y datos importantes para la generación de la arquitectura de la aplicación, partiendo desde las distintas vistas del modelo 4+1, pasando por los requerimientos (funcionales o no funcionales) y su adecuada priorización, generación de escenarios.

Las partes más importantes del desarrollo de la arquitectura se reflejarán en los diferentes apartados del documento, existen elementos con un alto acoplamiento, donde estos mostraran la transformación del proceso a lo largo del trabajo de diseño arquitectónico y una línea de tiempo de los cambios realizados.

## Objetivo

El documento nos permite tener una vista a grandes rasgos de la composición del portal financiero en línea para el Universal Bank. Mediante el documento, se busca la representación de diferentes vistas arquitecturales que describan los componentes que son de alta relevancia para entender el funcionamiento del sistema y poder trasmitir de manera clara, las decisiones arquitecturales que han sido tomadas para darle forma a los objetivos planteados.

## Definiciones, Acrónimos y Abreviaturas

PFL: Portal financiero en Línea.

CICS: Customer Information Control System.

## Referencias

Enterprise Integration Patterns - Introduction to Integration Styles. (2019). Retrieved 14 October 2019, from https://www.enterpriseintegrationpatterns.com/patterns/messaging/IntegrationStylesIntro.html

Microservices Pattern: Microservice Architecture pattern. (2019). Retrieved 14 October 2019, from https://microservices.io/patterns/microservices.html

SOA Patterns | Arcitura Patterns. (2019). Retrieved 14 October 2019, from https://patterns.arcitura.com/soa-patterns

## Alcance

Este documento de arquitectura de software busca la especificación de manera general de los componentes y sus interacciones que conforman la arquitectura del portal financiero en línea para el Universal Bank.

# Contexto

Universal Bank es un banco tradicional de tamaño mediano que ha adquirido una gama completa de capacidades de servicios financieros a través de una serie de adquisiciones. El banco actualmente tiene una presencia en línea limitada que está fragmentada a través de sus diversas divisiones. Como parte de su estrategia de expansión, Universal Bank ha decidido innovar en el mercado de la banca en línea proporcionando una serie de servicios de valor agregado además de una capacidad de gestión financiera totalmente integrada. Algunos bancos ofrecen actualmente, o están desarrollando la capacidad de ofrecer, banca en línea integrada en todas las cuentas (como ahorros, cheques y tarjetas de crédito). Pero ningún otro banco ofrece una gama completa de servicios de valor agregado, como asesoramiento financiero, análisis financiero y planificación, y planificación y presentación de impuestos. Además, Universal Bank no ha visto las sinergias que preveía de ofrecer una línea completa de productos. Esta falta de sinergia es causada por su incapacidad para realizar ventas cruzadas de manera efectiva en función de las relaciones existentes y el conocimiento del cliente.

## Fundamentos de la solución

Problema:

El universal bank requiere fortalecer la presencia en línea del banco. Convertir el canal en línea en un medio eficaz para agregar nuevos servicios a un costo relativamente bajo.

Por lo que se requiere:

* Opción de pago de facturas electrónicas al sistema bancario en línea actual.
* Permitir a los clientes programar pagos desde cuentas de ahorros y corrientes.

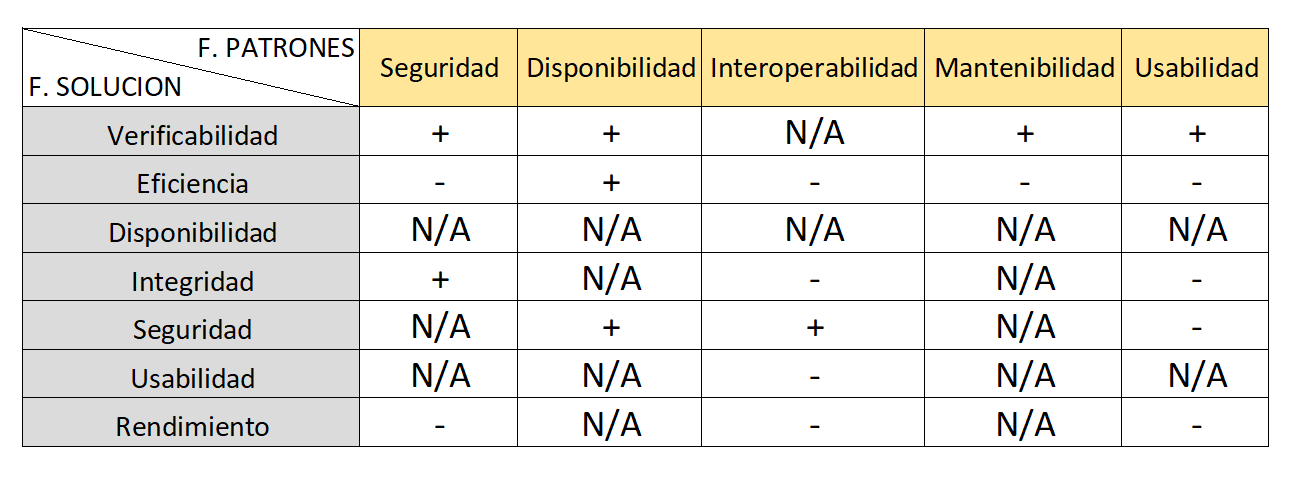
Patrón nuclear

* Microservice Architecture
  + Permitir muchos desarrolladores trabajando en la misma aplicación.
  + Curva de aprendizaje corta.
  + Facilitar en el mantenimiento.
  + Facilitar la implementación continua de la aplicación.
  + Favorecer la escalabilidad y disponibilidad.

Patrones complementarios

* API Gateway
  + Proporcionar un solo punto de comunicación entre las aplicaciones y los servicios.
  + Proporcionar conexión a diferentes tipos de dispositivos o clientes de los servicios.
  + Ofrecer estabilidad en las aplicaciones clientes en los cambios de los servicios.
  + Los servicios pueden usar un conjunto diverso de protocolos, algunos de los cuales pueden no ser compatibles con la web
* Audit logging
  + Capturar las acciones realizadas por un usuario al interactuar con los servicios.
* Single Service Instance per Host
  + Independencia con la variedad de lenguajes, frameware y versiones de framewares, utilizadas en el desarrollo de microservicios.
  + Permitir la utilización de múltiples instancias de servicio para rendimiento y disponibilidad.
  + Permitir que el servicio pueda ser desplegable y escalable independientemente.
  + Proporcionar agilidad al construir e implementar un servicio.
  + Permitir una implementación confiable.
* Access token
  + Certificar que un usuario esté autorizado para realizar una operación.
* 3rd Party Registration
  + Permitir que a las instancias de servicio registrarse al inicio y desligarse al apagar.
* Server-side service discovery
  + El número de instancias de servicios y sus ubicaciones cambia dinámicamente.
  + Las máquinas virtuales y los contenedores suelen tener asignadas direcciones IP dinámicas.
  + El número de instancias de servicios puede variar dinámicamente. Por ejemplo, un grupo de autoescalado EC2 ajusta el número de instancias en función de la carga.

**TRADE OFF**



Consecuencias:

Realizar la aplicación bajo la arquitectura de microservicios, nos permite tener un excelente manejo de los cambios realizados sobre los servicios sin afectar las aplicaciones clientes que los consuman, se puede ofrecer una integración con diversas aplicaciones echas en cualquier lenguaje de programación, se alcanza un bajo acoplamiento entre los componentes del sistema. Se obtiene alta disponibilidad, tolerancia a fallos.

# Drivers de Arquitectura

## Objetivos de la arquitectura de la solución

* Implementar mecanismos de confidencialidad, integridad y no repudio para el 99% de las transacciones realizadas por los clientes en el PFL a enero 2020, para que sea amigable, rápida y confiable la experiencia de generación de pagos de facturas y servicios.

## Requerimientos Funcionales Significativos

* Implementar un portal de banca en línea que permita a los clientes pagar facturas en línea desde sus cuentas.
* Implementar integraciones con el sistema actual del banco.
* Implementar integración con sistema de gestión CRM.
* Implementar integración con pasarelas de pago externa.
* Utilizar Sistema de autenticación servidor de directorios.

## Restricciones

* Todas las transacciones relacionadas con la cuenta utilizarán el sistema actual, que reside en un IBM mainframe utilizando (CICS).
* El sistema de bancos en línea residirá en el centro de datos corporativo en Seattle, Washington. Se conectará al centro de datos de un banco adquirido en Los Angeles, California a través de una línea privada arrendada.
* Las tasas de transacción globales del sistema, los usuarios simultáneos y el tiempo de respuesta deben cumplir con el uso proyectado del primer año más un factor de seguridad de ingeniería de 3 veces (o tres veces el uso proyectado del primer año) para manejar la carga de ráfaga.
* El sistema debe cumplir o exceder el acuerdo de nivel de servicio (SLA) para nuestro sistema en línea actual.

## Atributos de calidad

### **Descripción de atributos de calidad**

#### Seguridad

* Confidencialidad: Garantizar que los datos de los clientes y proveedores viajen desde la aplicación hasta su persistencia en base de datos sin ser accedidos de ninguna forma.
* Integridad: Garantizar la autenticidad de los mensajes mediante mecanismos de tokenización. Asegurar que los datos a procesar son los enviados por los clientes y proveedores, con el fin de evitar fraudes.
* No repudio: Se debe llevar una traza de cada evento, consumo de servicio y acción realizada desde la aplicación. Permite obtener información para análisis de fallos y comportamiento de los clientes y proveedores sobre la aplicación.
* Autenticidad: Garantizar la identidad de un cliente o un proveedor.

#### Disponibilidad / Escalabilidad

Disponibilidad: Mantener operatividad de la aplicación un 99.7%, garantizando la experiencia en el proceso de cotización y oferta sobre la misma.

#### Interoperabilidad

* Interoperabilidad: Permitir que varias aplicaciones sin importer el lenguaje de programación en el que están escrito puedan interactuar con las funcionalidades de negocio definidas en la herramienta.

#### Mantenibilidad/Manejabilidad

* Modularidad: Integraciones de tecnologías, herramientas y desarrollos para desacoplar la estructura de la aplicación. Con el fin de aumentar la granularidad de los componentes.

#### Facilidad de soporte

* Capacidad para ser modificado: Permitir que los cambios realizados en cualquier capa no afecten significativamente el desempeño y funcionamiento de la aplicación.

#### Usabilidad

* Capacidad para ser usado: Proporcionar a los clientes opciones de fácil uso.
* Protección contra errores de usuario: Estructurar la navegación dentro de la aplicacón con el fin de evitar al usuario tomar caminos que provoquen errores que deriven malas experiencias o errores en la generación de las cotizaciones y/o ofertas por parte de proveedores.
* Estética de la interfaz de usuario: Proporcionar interfaces que permitan a los usuarios una buena experiencia en su proceso de cotización/oferta y visualización de productos.

# Vistas de arquitectura

## Vistas de la arquitectura

### Vista lógica

**Imagen que contiene captura de pantalla, texto

Descripción generada automáticamente**

### Vista de desarrollo

### Vista de procesos

### Vista física

### Vista de casos de uso

# Ver pagos programados

Imagen que contiene texto, mapa

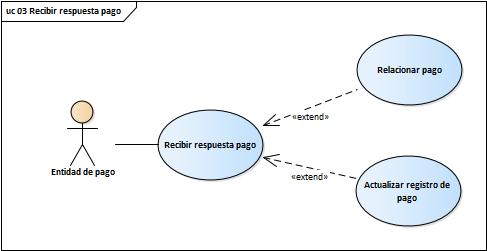
Descripción generada automáticamente

# Ejecutar pago programado

Imagen que contiene texto, mapa

Descripción generada automáticamente

# Recibir respuesta de pago



# Programar pagos

Imagen que contiene texto, mapa

Descripción generada automáticamente

### Vista de datos

# Riesgos de la arquitectura

|  |  |  |  |
| --- | --- | --- | --- |
| **Riesgos** | **Estrategia de mitigación** | **Plan de contingencia** | **Impacto** |
| Ataques de inyección de mensajería corrupta a los servicios de la aplicación SOBS. | Se implementa tokenización por JWT, permitiendo validar la autenticidad del mensaje. | Reversar TX encontradas en este caso. | MUY ALTO |
| Algunos servicios podrían exceder la capacidad en consumo de recursos. | Se realiza una revisión y optimización de los servicios que exceden la capacidad de consumo de recursos. | Aumentar capacidad de escalamiento en nube. | MEDIO |
| Alto costo de los servicios en la nube, cuenta con recursos limitados. | Se contratan servicios básicos con la opción de crecimiento escalado manual. | Aumentar capacidad de los planes en el momento requerido. | MEDIO |
|  |  |  |  |

# ANEXO