BB Recaudaciones con Banred V0.01

Especificación Técnica



Índice de Contenidos

[**Introducción.** 3](#_Toc92278874)

[**Descripción del Proyecto** 3](#_Toc92278875)

[**Propiedades del Proyecto o Componente** 3](#_Toc92278876)

[**Tecnología: Componentes y Propiedades.** 5](#_Toc92278877)

[**Arquitectura** 8](#_Toc92278878)

[Principios y Patrones utilizados. 8](#_Toc92278879)

[Diagrama de Componentes 11](#_Toc92278880)

[**Componente Principal ()** 11](#_Toc92278881)

[**Componente FrontEnd – Web.** 11](#_Toc92278882)

[Seguridad. 13](#_Toc92278883)

[**ACEPTACIÓN DEL DOCUMENTO** 15](#_Toc92278884)

# **Introducción.**

El presente documento busca dar a conocer los detalles técnicos del componente de Recaudaciones con Banred, en su primera versión se inicia con el servicio de ANT, posterior se analiza el incremento de servicios.

# **Descripción del Proyecto**

El componente de recaudaciones se basa en microservicios que permitirá la Recaudación de servicios, se iniciara con el servicio de ANT, estos microservicios se comunicaran e integraran con la plataforma de Banred (Universal Payment.

Banred a través de su plataforma Universal Payment realizara la orquestación del servicio con los proveedores y enviara a realizar (Debito, Crédito, Reverso, Replica) a Bolivariano de acuerdo a la transacción solicitada por el cliente o proceso.

# **Propiedades del Proyecto o Componente**

|  |  |  |  |
| --- | --- | --- | --- |
| **Versión:** | 0.0.1 | | |
| **Nombre del Proyecto o Componente** | BB Recaudaciones con Banred | | |
| **Alias del Proyecto o Componente:** | BB ANT | | |
| **Custodio del Componente:** | Recaudaciones | | |
| **Objetivos del Componente:** | Recaudaciones de Servicios con banco Bolivariano. | | |
| **Fecha de Liberación:** | Sprint 01 Marzo 2022 | | |
| **Ruta en Repositorio:** | Por definir | | |
| **Metodología:** | Agil | | |
| **Tipo de Componente:** | Microservicio de Negocio Agnóstico | | |
| **Infraestructura** | On-premise | | |
| **S.O** | Openshift (CoreOS) | | |
| **Servidor Web** | N/A | | |
| **Arquitectura:** | Microservicios | | |
| **Topología:** | API REST | | |
| **FrontEnd** | N/A | | |
| **Middleware** | Microservicios | | |
| **BackEnd** | QUARKUS | | |
| **Base de Datos** | CNB: Relacional (Sysbase)  OTC: Relacional (Oracle) | | |
| **Control de Versionamiento** | TFS / Git | | |
| **DevOps** | N/A | | |
| **Inteligencia Artificial** | N/A | | |
| **Pruebas** | Desarrollo: Unitarias  Desarrollo/BANRED/ANT: Integración/Funcionamiento  Arquitectura/QA/DBA/Desarrollo: Rendimiento/Carga | | |
| **Información de Seguridad:** | **Tipo de Servicio:** | Público | |
| **Medio de Acceso** | **Dominio de Seguridad** | **Políticas** |
| REST/JSON | API GATEWAY | JWT |
| **Consumidores:** |  | | |
| **SLAs:** | |  |  | | --- | --- | | **Tiempo de respuesta** | 100 milisegundos | | **Disponibilidad del servicio** | 24/7 | | **%Transacciones Exitosas** |  | | | |
| **Información Adicional:** | Componentes a Desarrollar por parte de recaudaciones | | |

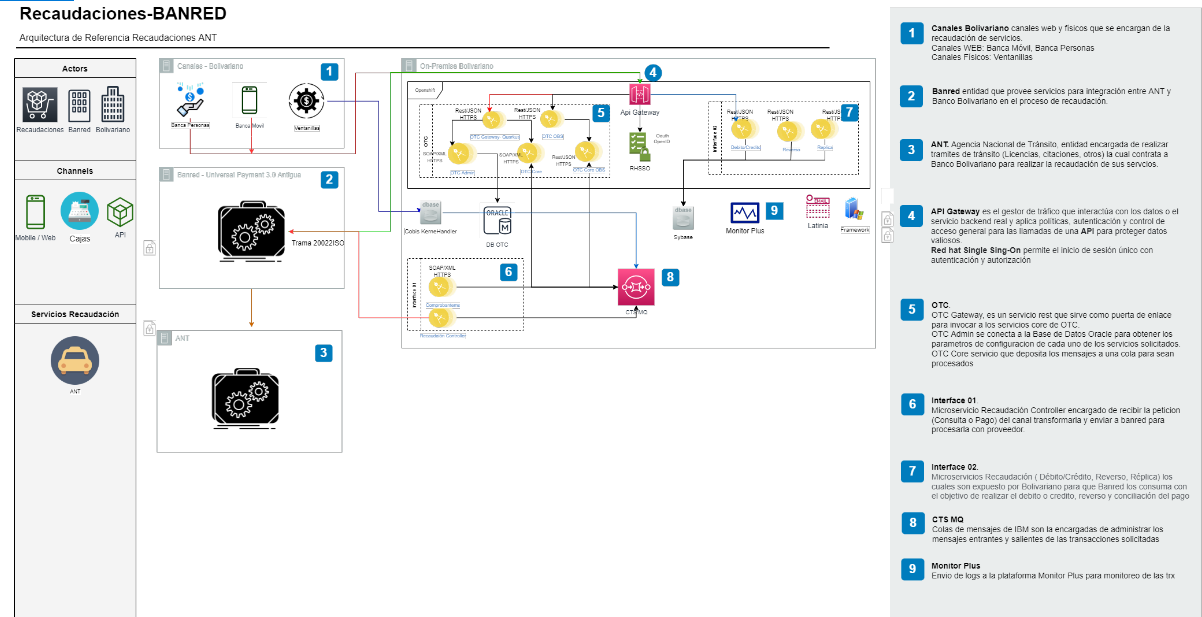
# **Tecnología: Componentes y Propiedades.**

|  |  |
| --- | --- |
|  | **Sybase**  **Sysbase** es un servicio de base de datos Relacional, es el core del banco. |
| **Micro servicios**  Quarkus es el framework basado en Java nativo de Kubernetes para desarrollo de micro servicios, propiedad de Red Hat. Beneficios orientado a reducir el tiempo, el consumo de memoria y disco . |  |
|  | **3Scale**  API Gateway es el gestor de tráfico que interactúa con los datos o el servicio backend real y aplica políticas, autenticación y control de acceso general para las llamadas de una **API** para proteger datos valiosos. |
| **RHSSO / Cognito**  permite el inicio de sesión único con autenticación y autorización |  |
| https://cdn-images-1.medium.com/max/512/1*BzD9VGIbTmQpANue7eC1Rw.png | **JSON**  Formato ligero de intercambio de datos, utilizado principalemente en peticiones REST,con estructura llave/valor, donde las llaves son únicas y sin caracteres especiales.  En el caso de los servicios REST se podrá recibir parámetros en este formato, ya que es mucho más ligero y no intervendría un proceso de parseo, lo cual disminuye considerablemente el uso de recursos. |
| **OTC**  Componete de transformacion y comunicación de XML entre ForntEnd y BanckEnd |  |
|  | **CTS**  Una cola es un contenedor de mensajes. Las aplicaciones empresariales que están conectadas al gestor de colas que aloja la cola pueden recuperar mensajes de la cola y transferirlos a la cola. |
| **Bus de Servicios**  Es un modelo de [arquitectura de software](https://es.wikipedia.org/wiki/Arquitectura_de_software) que gestiona la comunicación entre [servicios web](https://es.wikipedia.org/wiki/Servicios_web). Es un componente fundamental de la [Arquitectura Orientada a Servicios](https://es.wikipedia.org/wiki/Arquitectura_Orientada_a_Servicios). |  |
|  | Banred  BANRED provee soluciones e integra servicios especializados en el procesamiento de transacciones financieras,  compensación de cobros y pagos. |

# **Arquitectura**

El diseño del proyecto Recaudaciones está basado en la arquitectura de referencia Microservicios.

Diagrama de arquitectura general

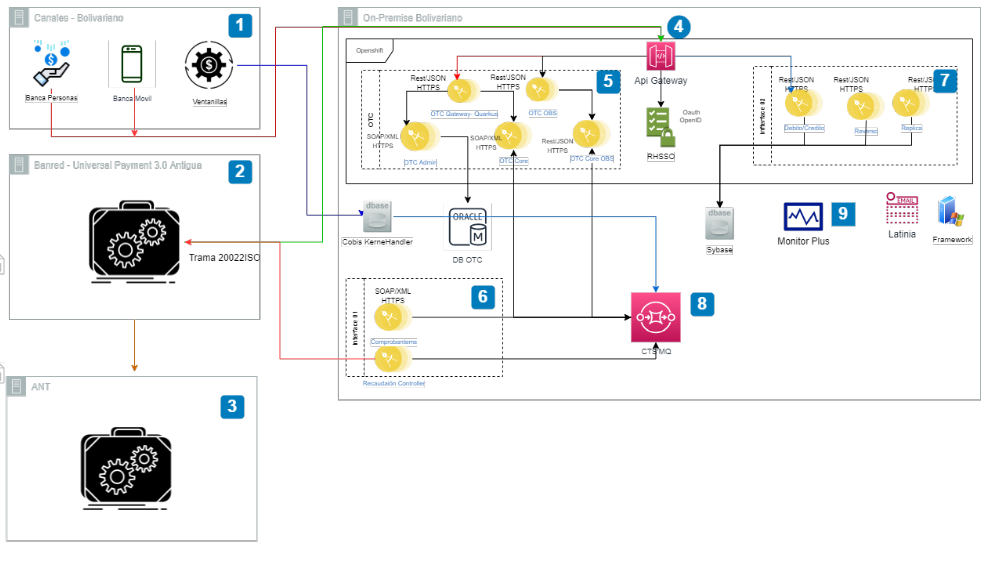


Principios y Patrones utilizados.

|  |  |
| --- | --- |
| PRINCIPIO | DESCRIPCION |
| Bajo acoplamiento | Este componente requiere de acoplamiento bajo consumo y se encuentra desacoplado de su entorno, es decir, puede ser desplegado dentro de cualquier instancia y servidor, independientemente de si este es físico o virtualizado. Alto desacoplamiento entre la exposición del servicio (contrato de servicio normalizado) y la lógica core del servicio (Patrón Orquestación, implementación del servicio). |
| ReuSABILIDAD | El servicio contiene y expresa la lógica agnóstica y se puede colocar como un recurso reutilizable. |
| Autonomía | El servicio ejerce un alto grado de control sobre su entorno de ejecución en tiempo de ejecución subyacente. El servicio desacopla el flujo de manejo de peticiones y la lógica core del servicio (basado en la arquitectura de referencia MSA), con el fin de maximizar la autonomía entre los componentes. |
| Circuit BREAKer | El patrón Circuit Breaker (Corto circuito) permite cerrar la comunicación con un determinado servicio cuando se ha detectado que está fallado. Así también detecta cuando se supera el estado de falla para abrir la comunicación. |
| MIcroservicios | El patrón Microservicios tiene como objetivo es poder desacoplar lo máximo posible todos los componentes |
| API Gateway | El patrón el API Gateway es la cara que damos a los clientes, y es la forma en que los clientes externos se comunicarán con nosotros por lo que es común que el API Gateway ofrezca servicios simples y de alto nivel que oculten la complejidad de nuestra arquitectura. |
| Access Token | Evitar el uso de usuario y password en cada petición al servidor y en su lugar enviar un Token, el cual es una cadena de caracteres sin aparente significado, pero que el servidor puede descifrar y comprobar la autenticidad del usuario. |
| Store and Forward | Este patrón es altamente utilizado, garantiza la entrega de los mensajes, los productores pueden enviar miles o millos de mensajes y el consumidor los irá tomando a medida que pueda procesarlos, de esta forma, evitamos asfixiar al consumidor con tantos mensajes. |
| Single Sign On | separar la lógica de autenticación de la aplicación como un componente independiente, que tenga como única responsabilidad la de autenticar a los usuarios. |

Diagrama de Componentes

Este Diagrama de Componentes ilustra las piezas de software, en este caso la estructura tecnológica del micro servicio, que conformará la solución BB Recaudaciones con Banred.



## **Componente Principal ()**

El componente principal lo integran entre Banred (BackEnd)) y Bolivaraino (FrontEnd y BackEnd)

## **Componente FrontEnd – Web.**

El componente de FrontEnd lo provee Bolivariano con los canales web y físicos hacia el cliente.

Estos canales están construidos y no ameritan cambios al menos que se defina solicitar algún nuevo parámetro por servicio.

**Componente BackEnd.**

El componente BanckEnd es compartido entre Banred y Bolivariano donde se realizarán microservicios que exponga y consuman servicios de Banred, se detallan los componentes utilizados.

* **Canales Bolivariano** canales web y físicos que se encargan de la recaudación de servicios.
* Canales WEB: Banca Móvil, Banca Personas
* Canales Físicos: Ventanillas
* **Banred** entidad que provee servicios para integración entre ANT y Banco Bolivariano en el proceso de recaudación.
* **ANT**. Agencia Nacional de Tránsito, entidad encargada de realizar trámites de tránsito (Licencias, citaciones, otros) la cual contrata a Banco Bolivariano para realizar la recaudación de sus servicios.
* **API Gateway** es el gestor de tráfico que interactúa con los datos o el servicio backend real y aplica políticas, autenticación y control de acceso general para las llamadas de una API para proteger datos valiosos.
* **Red hat Single Sing-On** permite el inicio de sesión único con autenticación y autorización
* **OTC Gateway**, es un servicio rest que sirve como puerta de enlace para invocar a los servicios core de OTC.
* **OTC Admin** se conecta a la Base de Datos Oracle para obtener los parámetros de configuración de cada uno de los servicios solicitados.
* **OTC Core** servicio que deposita los mensajes a una cola para sean procesados
* **Interface 01**.
* **Microservicio Recaudación Controller** encargado de recibir la petición (Consulta o Pago) del canal transformarla y enviar a banred para procesarla con proveedor.
* **Interface 02.**
* **Microservicios Recaudación** (Débito/Crédito, Reverso, Réplica) los cuales son expuesto por Bolivariano para que Banred los consuma con el objetivo de realizar el débito o crédito, reverso y conciliación del pago
* **CTS MQ** Colas de mensajes de IBM son encargadas de administrar los mensajes entrantes y salientes de las transacciones solicitadas
* **Monitor Plus** Envió de logs a la plataforma Monitor Plus para monitoreo de las trx

Seguridad.

Las políticas de seguridad a aplicar en cada dominio de seguridad de los medios de accesos definidos para el servicio, basado en el tipo de servicio público o privado

|  |  |  |
| --- | --- | --- |
| Medio de Acceso | Dominios de Seguridad | Políticas |
| SOAP | Disponibilidad | Aplicar patrones de regulación y articulación. |
| Confidencialidad | Exponer el servicio por protocolo HTTPS |
| Aplicar estándar WS-Policy o WS-Security para cifrado de mensaje. |
| Integridad | Exponer el servicio por protocolo HTTPS |
| Aplicar estándar WS-Policy o WS-Security para firmado de mensaje (mensaje firmado) |
| Autenticidad | Aplicar estándar WS-Policy o WS-Security para firmado de mensaje (mensaje firmado) |
| REST | Disponibilidad | Aplicar patrones de regulación y articulación. |
| Confidencialidad | Exponer el servicio por protocolo HTTPS |
| Aplicar estándar JWT JSON Web token |
| Integridad | Exponer el servicio por protocolo HTTPS para cifrado de mensaje. |
| Aplicar estándar JWT JSON Web token para firmado de mensaje (mensaje firmado) |
| Autenticidad | Aplicar autenticación HTTP con usuario y clave a nivel de servidor Web. En caso de que la plataforma lo permita y los consumidores lo soporten, aplicar autenticidad por medio del estándar OAuth |

# **ACEPTACIÓN DEL DOCUMENTO**

Responsables de los diferentes departamentos funcionales técnicos:

|  |  |
| --- | --- |
| **HOJA DE APROBACIÓN** | |
| **AUTORIZACIÓN** | **FIRMA DE APROBACIÓN** |
| **Líder Técnico Banco Bolivariano** |  |
| **Líder de Proyecto Interno** |  |
| **Líder de Proyecto (Nombre Empresa Proveedor)** |  |