**GUIA DE GENERACION DE PARÁMETROS PARA LOS CLIENTES QUE USARAN EL SERVICIO**

**JUNIO, 2024**

|  |  |  |  |
| --- | --- | --- | --- |
| **Nombres y Apellidos / Cargo** | | **Firma** | **Fecha** |
| Elaborado por | Richard Bendezu Villena |  | 19/06/2024 |
| Revisado por |  |  |  |

INDICE

[1. REQUISITOS 4](#_Toc135864051)

[2. GENERACION DE PARÁMETROS PARA GENERAR TOKEN 4](#_Toc135864052)

**HISTORIAL DE VERSIONES**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Fecha de Elaboración | Versión | Elaborado por | Descripción | Revisado por | Fecha de Revisión |
| 19/06/2024 | 1.0 |  | Primera versión del entregable |  |  |
|  |  |  |  |  |  |

|  |  |
| --- | --- |
| **Código** | SDSI-EXJUQ-P-00010-2024 |
| **Sistema / Módulo / Aplicación** | GENERAR LOS PARAMETROS USUARIO, CLAVE, CÓDIGO ROL Y CÓDIGO CLIENTE |
| **Documento(s) asociado(s)** | GLPI 334944 |

# OBJETIVO

Obtener los parámetros necesarios para los clientes que usaran el servicio.

# REQUISITOS

* Se debe contar con el java instalado jdk 8
* Se debe contar con el utilitario de encriptación Encryptador.jar

# GENERAR PARÁMETROS PARA OBTENER TOKEN DE AUTORIZACIÓN

|  |  |
| --- | --- |
| 1. Configuración de base de datos | |
| 1.- Obtener el key que se utilizara a lo largo de la encriptación y desencriptación. | El key es designado por el encargado del servidor de aplicaciones:   * Para encriptar todo referente a la seguridad del servicio usar la propiedad configuracion.seguridad.secretToken |
| 2.- Obtener los valores de usuario, clave, código cliente y código de rol. | - Estos valores pueden ser obtenidos del DBA quien realizo la ejecución de los scripts a la base de datos de Seguridad.  - Tambien se puede obtener los valores al ejecutar el siguiente script en la bd seguridad de postgresql  **SELECT** **DISTINCT**  u.c\_usuario **as** "usuario",c.c\_cliente **as** "codigo cliente",c.x\_cliente,r.c\_rol **as** "codigo rol",r.x\_descripcion  **FROM** seguridad.mae\_rol\_usuario ru  **INNER** **JOIN** seguridad.mae\_rol r **on** ru.n\_rol=r.n\_rol  **INNER** **JOIN** seguridad.mae\_operacion op **ON** op.n\_rol=r.n\_rol  **INNER** **JOIN** seguridad.mae\_aplicativo ap **ON** ap.n\_aplicativo=op.n\_aplicativo  **INNER** **JOIN** seguridad.mae\_usuario u **ON** ru.n\_usuario=u.n\_usuario  **INNER** **JOIN** seguridad.mae\_cliente c **ON** c.n\_cliente=u.n\_cliente  **WHERE** u.l\_activo = '1' **AND** ru.l\_activo = '1' **AND** r.l\_activo = '1'  **AND** op.l\_activo = '1' **AND** ap.l\_activo = '1' **AND** c.l\_activo = '1'  **AND** ap.c\_aplicativo = 'EJQWS'  Sin embargo la clave debe ser la que designo el DBA antes de encriptar y registrar. |
| 3.- Encriptar el usuario y clave obtenidos. | * Con el utilitario Encryptador.jar y usando la opción “Encriptar” encriptar el usuario y clave. |
| 4.- Remitir los parámetros obtenidos. | - Remitir como **clave:valor** lo siguientes datos a quien lo requiera para poder obtener el token de autorización.  username: (valor del usuario encriptado en el punto 3) password: (valor de la clave encriptada en el punto 3)  codigoCliente: (valor del código de cliente sin encriptar, punto 2)  codigoRol: (valor del código del rol sin encriptar,punto 2) |