SUNAT

**SISTEMA: API Gateway**

**SUBSISTEMA: Control Acceso OAuth**

Formato F2

Informe de Definición **Versión 1.0**

MDSI v3.0

**INTENDENCIA NACIONAL DE SISTEMAS DE INFORMACION**

**SUPERINTENDENCIA NACIONAL DE ADUANAS Y ADMINISTRACIÓN TRIBUTARIA**

**Lima - Perú**

CONTENIDO

[Registro de control de cambios al Documento 3](#_Toc522279909)

[Introducción 4](#_Toc522279910)

[Información General del Sistema 4](#_Toc522279911)

[1. ANÁLISIS DE REQUERIMIENTOS DEL SISTEMA DE INFORMACIÓN 5](#_Toc522279912)

[1.1 MODELAMIENTO DE REQUERIMIENTOS DEL SISTEMA DE INFORMACIÓN 5](#_Toc522279913)

[2. ANÁLISIS DEL SISTEMA DE INFORMACIÓN 10](#_Toc522279914)

[ANALISIS ORIENTADO A OBJETOS (ASI 2, ASI 3 y ASI 4) 10](#_Toc522279915)

[2.1 Análisis de los Casos de Uso (ASI 2) 11](#_Toc522279916)

[2.2 Análisis de Clases (ASI 3) 11](#_Toc522279917)

[2.3 Análisis de Paquetes (ASI 4) 11](#_Toc522279918)

[2.4 Elaboración del Modelo de Datos (ASI 5) 12](#_Toc522279919)

[2.5 Elaboración del Modelo de Procesos del Sistema del Sistema de Información(ASI 6) 12](#_Toc522279920)

[2.6 ESPECIFICACION DE INTERFACES CON OTROS SISTEMAS (ASI 7) 12](#_Toc522279921)

[2.7 ESPECIFICACION DE NECESIDADES DE MIGRACION DE DATOS Y CARGA INICIAL (ASI 8) 13](#_Toc522279922)

[2.8 VERIFICACION DE LOS MODELOS DE ANALISIS DEL SISTEMA DE INFORMACIÓN (ASI 9) 13](#_Toc522279923)

[2.9 DEFINICION DEL ALCANCE DEL PLAN DE PRUEBAS (ASI 10) 13](#_Toc522279924)

[3. DISEÑO DEL SISTEMA DE INFORMACIÓN 13](#_Toc522279925)

[1.1 DEFINICION DE LA ARQUITECTURA DEL SISTEMA (DSI 1) 13](#_Toc522279926)

[No Aplica. 15](#_Toc522279927)

[1.2 DISEÑO DE LA ARQUITECTURA DE SOPORTE (DSI 2) 15](#_Toc522279928)

[1.3 DISEÑO ORIENTADO A OBJETOS (DSI 3 Y DSI 4) 15](#_Toc522279929)

[1.4 DISEÑO ESTRUCTURADO (DSI 5) 15](#_Toc522279930)

[1.5 DISEÑO FÍSICO DE DATOS (DSI 6) 15](#_Toc522279931)

[1.6 GENERACION DE ESPECIFICACIONES DE CONSTRUCCION (DSI 8) 16](#_Toc522279932)

[1.7 DISEÑO DE LA MIGRACION Y CARGA INICIAL DE DATOS (DSI 10) 23](#_Toc522279933)

[1.8 ESPECIFICACION DE REQUERIMIENTOS DEDOCUMENTACION DE USUARIO (DSI 11) 23](#_Toc522279934)

[1.9 DEFINICION DEL PLAN DE IMPLANTACION (DSI 12) 23](#_Toc522279935)

# Registro de control de cambios al Documento

El F2 es un documento que permite modelar el sistema en el proceso de construcción, de igual manera sirve para el posterior mantenimiento y su operatividad durante el ciclo de vida. Se documenta de manera incremental o en paralelo, agilizando así el proceso de desarrollo

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Fecha** | **Versión** | **Descripción del cambio** | **Autor** | **Código (\*) Proyecto o requerimiento** | **Numero de pase** |
| 17/08/2018 | 1.0 | Primer despliegue del microservicio tecnologia3-seguridad-controlacceso-oauth2-microservice. | DAIA |  | PAS20181U210300155 |
| 22/10/2018 | 1.1 | - Se incluye nuevo end-point para autenticación de clientes externos  - Se independizan los certificados, ahora la firma de cada end-point tiene su propio certificado.  - Se modifica para que el tiempo de expiración del token se lea desde la BD y no desde un secret de k8s.  - Se devuelve en la audiencia del token dos nuevos valores:  indicador: que permite determinar si el “recurso” / “servicio” es público o privado  gt: expresión que permita identificar a qué tipo de “grant\_type” se le está otorgando acceso | DAIA |  | PAS20181U210300196 |

F2 - INFORME DE DEFINICIÓN

# Introducción

El presente documento, denominado Informe de Definición, tiene por objetivo especificar las características técnicas a implementar en el sistema de información y su posterior mantenimiento.

Este documento, contiene los elementos que se crean durante las fases de análisis de requerimientos y la fase de diseño del sistema de información según lo descrito en la Metodología de Desarrollo de Sistemas de Información - MDSI v3.0

# Información General del Sistema

|  |  |
| --- | --- |
| **Código del Sistema** |  |
| **Nombre del Sistema** | API Gateway |
| **Código del Subsistema** |  |
| **Nombre del Subsistema** | tecnologia3-seguridad-controlacceso-oauth2-microservice |
| **Código del Modulo** |  |
| **Nombre del Modulo** |  |
| **Coordinador del Grupo encargado del Sistema.** | DAIA |

# ANÁLISIS DE REQUERIMIENTOS DEL SISTEMA DE INFORMACIÓN

## MODELAMIENTO DE REQUERIMIENTOS DEL SISTEMA DE INFORMACIÓN

### Determinación del Alcance del Sistema de Información (ASI 1.1)

1. **Describir el alcance del sistema de información**

El sistema brinda los servicios de autenticación, autorización, generación y validación de token:

* Internet
* Intranet
* Extranet

1. **Elaborar el diagrama de contexto general del sistema**

1. Elaborar la trazabilidad delos procesos de negocio con el requerimiento informático de negocio

**No Aplica**

Matriz de trazabilidad (Esta información se llenará para los casos en que exista la información solicitada)

|  |  |  |  |
| --- | --- | --- | --- |
| Macro Proceso:  Proceso:  Subproceso: |  | | |
| Actividades | | Roles involucrados (actor) | Requerimiento informático de negocio |
|  | |  |  |
|  | |  |  |
|  | |  |  |

### Obtención de Requerimientos (ASI 1.2)

1. Elaborar la trazabilidad de Funciones-Roles

**No Aplica**

Se identifican a los roles involucrados en el requerimiento y se describe su participación e interés en el proceso y se detallan los roles a cumplir por cada uno de ellos. Esta identificación de roles permitirá la trazabilidad de actores de sistemas para el Diagrama de Casos de Uso.

|  |  |  |
| --- | --- | --- |
| Nro. | Roles involucrados | Funciones que realiza el rol |
|  |  |  |

1. Identificación de los Requerimientos Funcionales

**No Aplica**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Nro. | Nombre del Requisito Funcional | Descripción detallada | Criterio de Aceptación | Rin asociado | Impacto (1..4) |
|  |  |  |  |  |  |

1. Identificación de los Requerimientos No Funcionales

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Nro. | Característica | Nombre del Requisito  No Funcional | Descripción detallada | Rin asociado | Impacto (1..4) |
| 1 | Eficiencia | **RNF1** | Poseer la capacidad de ser distribuido (clustering), debiendo soportar hasta 1000 usuarios por minuto. |  | 4 |

### 

### Especificación de los Casos de Uso del Sistema (ASI 1.3)

#### Elaborar Trazabilidad Requerimiento- Casos de Uso

Se elabora la Matriz de Trazabilidad entre los Requerimientos de ASI 1.2 con los Casos de Uso identificados a fin de determinar que todas las funcionalidades cumplen con los Requerimientos solicitados por los Usuarios.

**No Aplica**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Requerimiento \ Caso de Uso | CU01- Autenticar por Internet | CU02-  Autenticar por Extranet | CU03-  Autenticar por Intranet | CU04-  Autenticar por DNI | CU05-Autenticar por Internet SOL |
|  |  |  |  |  |  |

#### Identificar Actores del Sistema

Se elabora el cuadro con la descripción de cada Actor del sistema

|  |  |  |
| --- | --- | --- |
| Núm. | Actor del sistema | Descripción |
|  | Usuario | Actor genérico que agrupa los actores: Externo, Contribuyente, Persona Natural y funcionario. |
|  | Externo | Organismo gubernamental que usa la infraestructura de SI de la SUNAT para agilizar sus operaciones de negocio. |
|  | Administrador Entidad | Tipo de usuario que tendrá acceso a las opciones de mantenimiento de roles y perfiles de funcionarios de su propia entidad |
|  | Funcionario Entidad | Tipo de usuario al que el usuario administrador le delega funciones de negocio |
| 3 | Contribuyente | Persona natural o jurídica que dispone de un número de ruc |
| 4 | Persona Natural DNI | Persona natural que se identifica solo con el número de DNI. |
| 5 | Funcionario | Funcionario de la SUNAT. |

#### Elaborar diagrama de Actores del Sistema



#### Elaborar Diagrama de Casos de Uso del Sistema (Gráfico)

**No Aplica**

#### Especificar los Casos de Uso del Sistema

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1. Nombre del Caso de Uso del Sistema | | | CUS01 – Inicio de sesión de Usuario Intranet | |
| 2. Descripción del Caso de Uso | | | | |
| Permitir al funcionario iniciar sesión en el directorio interno de usuarios Sunat. | | | | |
| 3. Actor(es) | | | | |
| Funcionario | | | | |
| 4. Precondiciones | | | | |
| * Envío de datos a través de una aplicación web o móvil. * La aplicación debe enviar el parámetro grantType = password * La aplicación debe tener registrado un clientId y un clientSecret los cuales se enviarán como parámetros | | | | |
| 5. Post condiciones | | | | |
|  | | | | |
| 6. Flujo de eventos \* | | | | |
| Nro. | Acción del Actor | | | Respuesta del Sistema |
|  | El usuario envía sus credenciales a través de la aplicación:   * Usuario Intranet * Clave Intranet | | |  |
|  |  | | | El sistema realiza los siguientes pasos:  - Valida que el valor del parámetro grantType sea igual a “password”  \* Si el valor del parámetro es incorrecto se devuelve el mensaje “grant\_type no valido”  - Valida que los parámetros (grantType, clientId, clientSecret, Username y Password) tengan valores.  \* Si algún parámetro es nulo, se devuelve el mensaje "parámetros inválidos"  - Valida el clientId y el clientSecret (existencia, validez y estado: activo/inactivo)  \* Si falla la validación se devuelve el mensaje "cliente no autorizado"  - Autentica los datos del usuario (usuario y clave)  - Autoriza al usuario a los recursos que puede acceder  - Devuelve un token que permitirá a la aplicación invocar los servicios a los que el usuario está autorizado. |
| 7. Flujos alternativos / Sub-flujos | | | | |
| Nro. | Acción del Actor | | | Respuesta del Sistema |
|  |  | | |  |
| 8. Excepciones | | | | |
| Nro. | | Descripción | | |
|  | |  | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1. Nombre del Caso de Uso del Sistema | | | CUS02 – Inicio de sesión de Usuario Sol | |
| 2. Descripción del Caso de Uso | | | | |
| Permitir al contribuyente iniciar sesión en los aplicativos de Sunat. | | | | |
| 3. Actor(es) | | | | |
| Contribuyente | | | | |
| 4. Precondiciones | | | | |
| * Envío de datos a través de una aplicación web o móvil. * La aplicación debe enviar el parámetro grantType = password * La aplicación debe tener registrado un clientId y un clientSecret los cuales se enviarán como parametros | | | | |
| 5. Post condiciones | | | | |
|  | | | | |
| 6. Flujo de eventos \* | | | | |
| Nro. | Acción del Actor | | | Respuesta del Sistema |
|  | El usuario envía sus credenciales a través de la aplicación:   * RUC * Usuario Sol * Clave Sol   ó   * DNI * Clave Sol | | |  |
|  |  | | | El sistema realiza los siguientes pasos:  - Valida que el valor del parámetro grantType sea igual a “password”  \* Si el valor del parámetro es incorrecto se devuelve el mensaje “grant\_type no valido”  - Valida que los parámetros (grantType, clientId, clientSecret, Username y Password) tengan valores.  \* Si algún parámetro es nulo, se devuelve el mensaje "parámetros invalidos"  - Valida el clientId y el clientSecret (existencia, validez y estado: activo/inactivo)  \* Si falla la validación se devuelve el mensaje "cliente no autorizado"  - Autentica los datos del usuario (usuario y clave)  \* Si entra con DNI, lo convierte en RUC  \* Si falla la autenticación se devuelve el mensaje "Error en la autenticación del usuario."  - Autoriza al usuario a los recursos que puede acceder  \*Si falla la autorización se devuelve el mensaje "Error en la autorización del usuario."  - Devuelve un token que permitirá a la aplicación invocar los servicios a los que el usuario está autorizado. |
| 7. Flujos alternativos / Sub-flujos | | | | |
| Nro. | Acción del Actor | | | Respuesta del Sistema |
|  |  | | |  |
| 8. Excepciones | | | | |
| Nro. | | Descripción | | |
|  | |  | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1. Nombre del Caso de Uso del Sistema | | | CUS03 – Autenticación de cliente externo | |
| 2. Descripción del Caso de Uso | | | | |
| Permitir a clientes externos generar un token para interactuar con los servicios de Sunat. | | | | |
| 3. Actor(es) | | | | |
| Contribuyente | | | | |
| 4. Precondiciones | | | | |
| * Solicitud de token enviando un request a través de una aplicación externa. * La aplicación debe enviar el parámetro grantType = client\_credentials * La aplicación debe tener registrado un clientId y un clientSecret los cuales se enviarán como parámetros | | | | |
| 5. Post condiciones | | | | |
|  | | | | |
| 6. Flujo de eventos \* | | | | |
| Nro. | Acción del Actor | | | Respuesta del Sistema |
|  | Se envían los datos a través de la aplicación:   * client\_id * client\_secret | | |  |
|  |  | | | El sistema realiza los siguientes pasos:  - Valida que el valor del parámetro grantType sea igual a “client\_credentials”  \* Si el valor del parámetro es incorrecto se devuelve el mensaje “grant\_type no valido”  - Valida que los parámetros (grantType, clientId, clientSecret) tengan valores.  \* Si algún parámetro es nulo, se devuelve el mensaje "parametros invalidos"  - Valida el clientId y el clientSecret (existencia, validez y estado: activo/inactivo)  \* Si falla la validación se devuelve el mensaje "cliente no autorizado"  - Devuelve un token que permitirá a la aplicación invocar los servicios a los que el usuario está autorizado. |
| 7. Flujos alternativos / Sub-flujos | | | | |
| Nro. | Acción del Actor | | | Respuesta del Sistema |
|  |  | | |  |
| 8. Excepciones | | | | |
| Nro. | | Descripción | | |
|  | |  | | |

### Especificación de la Interface de Usuario (ASI 1.4)

Es un prototipo de las pantallas. Así mismo para explicar algunas características de cada pantalla se usará la siguiente tabla.

|  |  |
| --- | --- |
| Número |  |
| Propósito de la interface |  |
| Gráfica de la interface | |
|  | |

**No Aplica**

### Identificación de Perfiles (ASI 1.5)

|  |  |  |
| --- | --- | --- |
| Perfil | **SQS** | |
| Opciones a las que tiene acceso el perfil | | Tipo de Acceso |
|  | |  |

**No Aplica**

### Especificación del Comportamiento Dinámico de la Interface (ASI 1.6)

**No Aplica**

### Especificación de Formatos de Impresión (ASI 1.7)

**No Aplica**

### Obtención de servicios candidatos (ASI 1.8)

|  |  |  |  |
| --- | --- | --- | --- |
| Núm. | Nombre del servicio candidato | Descripción del candidato | Requerimiento funcional asociado |
| 1 |  |  |  |
| 2 |  |  |  |
| n |  |  |  |

**No Aplica**

# ANÁLISIS DEL SISTEMA DE INFORMACIÓN

## ANALISIS ORIENTADO A OBJETOS (ASI 2, ASI 3 y ASI 4)



## Análisis de los Casos de Uso (ASI 2)

|  |  |
| --- | --- |
| ASI 2.1: | Identificación de Clases Asociadas a un Caso de Uso |

1. **Identificarlos 3 tipos de clases de análisis**

Diagrama de Clases (Gráfico, lista de los 3 tipos de clases) <<\\dirección del modelo de clases en powerdesigner>>

**No Aplica.**

1. **Asignar atributos a las clases del tipo entidad** (Gráfico, clases del tipo entidad con atributos completos) <<\\dirección del modelo de clases en powerdesigner>>

**No Aplica.**

## 

## Análisis de Clases (ASI 3)

1. **Diccionario de clases**  (descripción de la clase)<<\\dirección del modelo de clases en powerdesigner>>

**No Aplica.**

**Clase Entidad**

|  |  |
| --- | --- |
| **Nombre de la clase** | **Descripción** |
|  |  |

**Clase control**

|  |  |
| --- | --- |
| **Nombre de la clase** | **Descripción** |
|  |  |

**Clase Interfaz**

|  |  |
| --- | --- |
| **Nombre de la clase** | **Descripción** |
|  |  |

1. **Diagrama de clases de análisis** (grafico de los 3 tipos de clases) <<\\direccion del modelo de clases en powerdesigner>> y el Diagrama de Transición de Estados (clases del tipo entidad que perduran en el tiempo) <<\\direccion del modelo de clases en powerdesigner>>. El Diagrama de Transición de Estados es obligatorio para nuevos requerimientos o casos de uso cuya especificación incluye objetos persistentes. Incluir en los casos de prueba verificar las transiciones válidas y verificar que no es posible las transiciones no válidas.

**No Aplica.**

1. **Modelo conceptual de datos** (grafico de diagrama de clases) ) <<\\direccion del modelo de clases en powerdesigner>>

**No Aplica.**

## Análisis de Paquetes (ASI 4)

1. **Identificación de paquetes** (grafico) <<\\direccion del modelo de paquetes en powerdesigner>>

**No Aplica.**

1. **Identificación de servicios**(grafico) <<\\direccion del modelo de paquetes distinguiendo los servicios comunes en powerdesigner**>>**indicando la lista de servicios

**No Aplica.**

### ANALISIS ESTRUCTURADO (ASI 5 y ASI 6)

### 

## Elaboración del Modelo de Datos (ASI 5)

1. Elaborar el modelo conceptual de datos (Gráfico) <<\\direccion del diagrama entidad relación en powerdesigner**>>**

**No Aplica.**

1. Elaborar el modelo lógico de datos (Gráfico) <<\\direccion del diagrama entidad relación en powerdesigner**>>**

**No Aplica.**

1. Normalizar el modelo lógico (Gráfico) <<\\direccion del diagrama entidad relación en powerdesigner**>>**

**No Aplica.**

### 

## Elaboración del Modelo de Procesos del Sistema del Sistema de Información(ASI 6)

* 1. Identificar los elementos de un diagrama de flujo de datos

**No Aplica.**

* 1. Elaborar el diagrama de flujo de datos; Modelo de Procesos del Sistema (Gráfico) <<\\direccion del diagrama de flujo de datos en powerdesigner>>

**No Aplica.**

## ESPECIFICACION DE INTERFACES CON OTROS SISTEMAS (ASI 7)

a. Describir la interface de relación con otro sistema

**No Aplica.**

|  |  |
| --- | --- |
| Nombre del Sistema relacionado |  |
| Responsable del Sistema |  |
| Procesos del Sistema relacionado |  |
| Especificaciones Funcionales de las interfaces con el sistema |  |
| Formatos de los datos intercambiados |  |
| Tipo de intercambio |  |
| Frecuencia o periodicidad del intercambio |  |
| Evento que desencadena la interface |  |
| Validaciones y/o requisitos especiales de seguridad |  |
| Modificaciones Requeridas |  |

## ESPECIFICACION DE NECESIDADES DE MIGRACION DE DATOS Y CARGA INICIAL (ASI 8)

### Tarea ASI 8.1: Especificación de Necesidades de Migración de Datos y Carga Inicial

1. **Elaborar el plan de migración de datos y carga inicial de datos**

**No Aplica.**

|  |  |
| --- | --- |
| Relación de Tablas a Migrar |  |
| Estimación del espacio requerido |  |
| Prioridad de las cargas |  |
| Necesidades especiales |  |

## VERIFICACION DE LOS MODELOS DE ANALISIS DEL SISTEMA DE INFORMACIÓN (ASI 9)

### Tarea ASI 9.1: Verificación de la calidad técnica de los modelos

**No Aplica.**

### Tarea ASI 9.2: Análisis de consistencia entre modelos

**No Aplica.**

## DEFINICION DEL ALCANCE DEL PLAN DE PRUEBAS (ASI 10)

### Tarea ASI 10.1: Definición de requerimientos del entorno de pruebas

* 1. **Planificar los requerimientos del entorno de pruebas;**

**No Aplica.**

* 1. **Definición de las especificaciones para realizar las pruebas del sistema**

**No Aplica.**

### Tarea ASI 10.2: Definición de las pruebas de aceptación de sistemas

### No Aplica.

# DISEÑO DEL SISTEMA DE INFORMACIÓN

## DEFINICION DE LA ARQUITECTURA DEL SISTEMA (DSI 1)

### Definición de Niveles de Arquitectura (DSI 1.1)

**No Aplica.**

### Identificación de subsistemas de diseño (DSI 1.2)

**No Aplica.**

* **Catálogo de subsistemas**

|  |  |  |
| --- | --- | --- |
| Subsistema | Funcionalidad | Tipo (Específico, Genérico) |
|  |  |  |
|  |  |  |
|  |  |  |

* **Grafico de subsistemas**

### Especificación de Requerimientos de Diseño y Construcción (DSI 1.3)

**No Aplica.**

Requisitos No Funcionales

|  |  |  |  |
| --- | --- | --- | --- |
| Nro. | Nombre del Requisito  No Funcional | Descripción detallada | Importancia (1..4) |
| 1 |  |  |  |
| 2 |  |  |  |
| 3 |  |  |  |

* **Catálogo de Excepciones (DSI 1.3)**

|  |  |
| --- | --- |
| Tipo: |  |
| Descripción: | |
|  | |
| Condiciones Previas del Sistema de Información: | |
|  | |
| Elemento Afectado: |  |
| Respuesta del Sistema de Información: | |
|  | |

* **Entorno Tecnológico del Sistema. (DSI 1.3)**

|  |  |
| --- | --- |
| Elementos de la infraestructura | |
| Hardware | CPU 3 Core - 21GB de Ram – Disco 100GB |
| Software | Servidor Web: NGINX 1.4.7  Servidor de Aplicaciones: Jetty 9.3  Software de Infraestructura: Kubernetes 1.3.10 y Dockers 1.9  Sistema Gestor de Base de Datos: Informix 12.10  Máquina Virtual Java: Jdk 1.8  Servidor de Monitoreo: elasticsearch-2.3.4  logstash-2.3.4  Kibana 4.5.4  IBM MQ 8.4  Sistema Operativo: Red Hat Enterprise Linux 7.1+ |
| Comunicaciones | Protocolo: TCP-IP, HTTP  Topología: LAN |

* **Estimación de la Planificación de Capacidades (DSI 1.3)**

**No Aplica.**

|  |  |  |  |
| --- | --- | --- | --- |
| Dispositivo | Almacenamiento | Procesamiento | Comunicación |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

### Especificación de servicio (SOA) (DSI 1.4)

## No Aplica.

* **Definición de niveles de servicios (DSI 1.4)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Núm.** | **Servicio** | **Seguridad** | **Performance** |
|  |  |  |  |

* **Diseño de mensajes (DSI 1.4)**

|  |  |  |
| --- | --- | --- |
| **Servicio** | **Mensaje entrada** | **Mensaje salida** |
|  |  |  |

* **Dependencias de servicio del modelo (DSI 1.4)**
* **Composición y flujos de servicios del modelo (DSI 1.4)**
* **Requisitos no funcionales (DSI 1.4)**

|  |  |  |
| --- | --- | --- |
| **Servicio** | **Tipo** | **Requerimiento No Funcional** |
|  |  |  |

### Diseño de Subsistemas (SOA)(DSI 1.5)

### No Aplica.

## DISEÑO DE LA ARQUITECTURA DE SOPORTE (DSI 2)

### Diseño de subsistemas de soporte (DSI 2.1)

### No Aplica.

## 

## DISEÑO ORIENTADO A OBJETOS (DSI 3 Y DSI 4)

### 

### Diseño de Casos de Uso Reales (DSI 3)

### No Aplica.

### Diseño de Clases (DSI 4)

### No Aplica.

## DISEÑO ESTRUCTURADO (DSI 5)

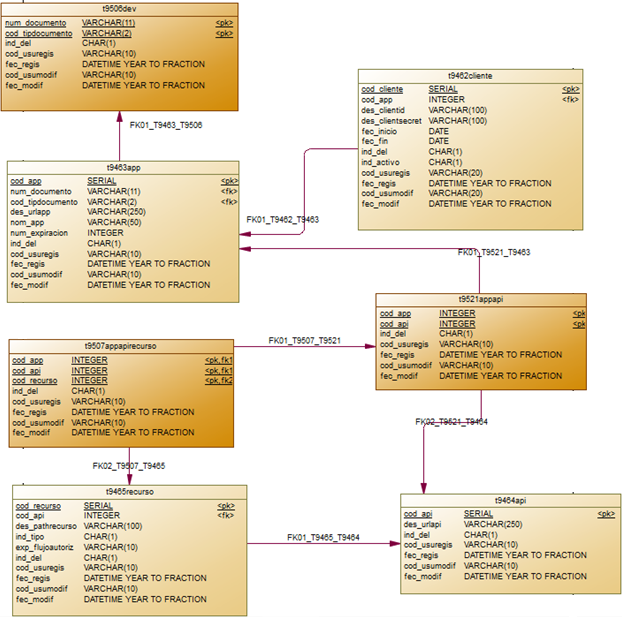
### 

### Diseño de la Arquitectura de Módulos del Sistema (DSI 5)

**No Aplica.**

## DISEÑO FÍSICO DE DATOS (DSI 6)

### Diseño del Modelo Físico de Datos (DSI 6.1)



### Especificación de los Caminos de Acceso a los Datos (DSI 6.2)

**No Aplica.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nro. | Camino | Tipo de Transacción | Frecuencia | Prioridad |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

### Optimización del Modelo Físico de Datos. (DSI 6.3)

**No Aplica.**

* **Formato de Estimación de Volúmenes de tablas**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Nro. | Tabla | Cantidad de registros anual | Tamaño del registro | Crecimiento anual (%) | Tipo de acceso |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

### Especificación de la Distribución de Datos. (DSI 6.4)

* **Formato de Asignación Esquema Físico de Datos a Nodos.**

|  |  |  |
| --- | --- | --- |
| Base de Datos | Tablas | Nodo |
|  |  |  |
|  |  |  |
|  |  |  |

## GENERACION DE ESPECIFICACIONES DE CONSTRUCCION (DSI 8)

### 

### Especificación del Entorno de Construcción (DSI 8.1)

**No Aplica.**

Plataforma del ambiente de desarrollo.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Especificación de la unidad de construcción | | | | |
| Nombre |  | | Acción |  |
| Paquete |  | | | |
| Método |  | | | |
| Parámetros de Entrada: |  | | | |
| Retorna |  | | | |
| EAR |  | | | |
| JAR/WAR |  | | | |
| Ruta |  | | | |
| Tipo |  | | | |
| Objetivo |  | | | |
| Especificación (seudocódigo) | | | | |
|  | | | | |
| Regla de Negocio (Opcional) | |  | | |

### Definición de Componentes y Subsistemas de Construcción (DSI 8.2)

### Gráfico Diagrama de Componentes y/o de Despliegue

### D:\Usuarios\hniquen\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.Outlook\T0HDZ998\Diagrama Arquitectura API Seguridad Produccion SUNAT v1-PRD Miraflores (3) (002).jpg

### D:\Usuarios\hniquen\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.Outlook\T0HDZ998\Diagrama Arquitectura API Seguridad Produccion SUNAT v1-PRD San Isidro (3) (002).jpg

### Elaboración de especificaciones de construcción (DSI 8.3)

**tecnologia3-seguridad-controlacceso-backend**

|  |  |  |  |
| --- | --- | --- | --- |
| Especificación de la unidad de construcción | | | |
| Nombre | OAuth2ClientesSolRestService.java | Acción |  |
| Nuevo/Modifica/Reusable | Métodos (endpoints) necesarios para el proceso de Login. | | |
| Tipo | JAVA | | |
| Objetivo | Login de usuarios sol y devolución de token. | | |
| Especificación (seudocódigo) | | | |
| @Path(OAuth2ClientesSolRestService.***INTEGRADOR\_PATH***)  **public** **class** OAuth2ClientesSolRestService {  @Inject  **private** AutenticacionService autenticacionService;  @Inject  **private** AutorizacionService autorizacionService;  @Inject  **private** GeneradorToken generadorToken;  **public** **static** **final** String ***INTEGRADOR\_PATH*** = "/v1/clientessol";  @POST  @Path("/{client\_id}/oauth2/token/")  @Consumes(MediaType.***APPLICATION\_FORM\_URLENCODED***)  @Produces({ MediaType.***APPLICATION\_JSON*** })  **public** Response tokenEndpoint(@PathParam(Constantes.***CLIENT\_ID***) String pathClientId,  @FormParam(Constantes.***GRANT\_TYPE***) String grantType, @FormParam(Constantes.***SCOPE***) String scope,  @FormParam(Constantes.***CLIENT\_ID***) String clientId, @FormParam(Constantes.***CLIENT\_SECRET***) String clientSecret,  @FormParam(Constantes.***USERNAME***) String username, @FormParam(Constantes.***PASSWORD***) String password) **throws** Exception {  **if** (!grantType.equals(Constantes.***PASSWORD***)) {  OAuthResponse response = OAuthASResponse.*errorResponse*(HttpServletResponse.***SC\_BAD\_REQUEST***)  .setError(OAuthError.TokenResponse.***INVALID\_GRANT***).setErrorDescription(Constantes.***GRANT\_TYPE\_ERROR***)  .buildJSONMessage();  **return** Response.*status*(response.getResponseStatus()).entity(response.getBody()).build();  } **else** **if** (clientId.equals(**null**) && clientSecret.equals(**null**) && username.equals(**null**)  && password.equals(**null**)) {  OAuthResponse response = OAuthASResponse.*errorResponse*(HttpServletResponse.***SC\_BAD\_REQUEST***)  .setError(OAuthError.TokenResponse.***INVALID\_REQUEST***).setErrorDescription(Constantes.***INVALID\_PARAM\_ERROR***)  .buildJSONMessage();  **return** Response.*status*(response.getResponseStatus()).entity(response.getBody()).build();  } **else** **if** (!generadorToken.validarCliente(clientId, clientSecret)) {  OAuthResponse response = OAuthASResponse.*errorResponse*(HttpServletResponse.***SC\_BAD\_REQUEST***)  .setError(OAuthError.TokenResponse.***UNAUTHORIZED\_CLIENT***).setErrorDescription(Constantes.***CLIENT\_AUTHORIZATION\_ERROR***)  .buildJSONMessage();  **return** Response.*status*(response.getResponseStatus()).entity(response.getBody()).build();  } **else** {  // try {  // \*\*\*\*\* Autenticacion \*\*\*\*\*\*\*\*\*  EntradaBean entBean = **new** EntradaBean();  // si es dni  **if** (username.length() == 8) {  entBean.setParDni(username);  entBean.setParTipo(Constantes.***TIPO\_DOCUMENTO\_1***);  // si es ruc  } **else** {  entBean.setSunatUser(username);  entBean.setParTipo(Constantes.***TIPO\_DOCUMENTO\_2***);  }  entBean.setPassword(password);  RespAutBean autBean = autenticacionService.autenticarUsuario(entBean);  // final Response.ResponseBuilder responseBuilder =  // Response.status(HttpServletResponse.SC\_FOUND);  // \*\*\*\*\*\* Verificar si hubo Error \*\*\*\*\*\*\*  **if** (!Constantes.***EXITO***.equals(autBean.getCodigo())) {  OAuthResponse response = OAuthASResponse.*errorResponse*(HttpServletResponse.***SC\_BAD\_REQUEST***)  .setError(OAuthError.CodeResponse.***ACCESS\_DENIED***)  .setErrorDescription(Constantes.***AUTENTICATION\_ERROR***).buildJSONMessage();  **return** Response.*status*(response.getResponseStatus()).entity(response.getBody()).build();  }  // \*\*\*\*\*\* Autorizacion \*\*\*\*\*\*\*\*\*  // EntradaAutBean entAutBean = new EntradaAutBean();  // entAutBean.setParDni(entBean.getParDni());  // entAutBean.setParTipo(entBean.getParTipo());  // entAutBean.setPassword(entBean.getPassword());  // entAutBean.setSunatUser(entBean.getSunatUser());  // entAutBean.setUrlOrig(originalUrl+"?state="+state);  RespAutorBean autorBean = autorizacionService.autorizarUsuario(autBean.getUsuarioBean());  // \*\*\*\*\*\* Verificar si hubo Error \*\*\*\*\*\*\*  **if** (autorBean == **null** || (autorBean != **null** && autorBean.getUsuarioBean() == **null**)) {  OAuthResponse response = OAuthASResponse.*errorResponse*(HttpServletResponse.***SC\_BAD\_REQUEST***)  .setError(OAuthError.CodeResponse.***UNAUTHORIZED\_CLIENT***)  .setErrorDescription(Constantes.***AUTHORIZATION\_ERROR***).buildJSONMessage();  **return** Response.*status*(response.getResponseStatus()).entity(response.getBody()).build();  }  // \*\*\*\*\*\* Generacion Token \*\*\*\*\*\*\*\*\*  String[] token = generadorToken.generarSignedJWT(clientId, grantType, ***INTEGRADOR\_PATH***, autorBean.getUsuarioBean(),  autorBean.getPerfiles());  OAuthResponse response = OAuthASResponse.*tokenResponse*(HttpServletResponse.***SC\_OK***).setAccessToken(token[0])  .setTokenType(token[1]).setExpiresIn(token[2]).buildJSONMessage();  **return** Response.*status*(response.getResponseStatus()).entity(response.getBody()).build();  }  }  } | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Especificación de la unidad de construcción | | | |
| Nombre | OAuth2ClientesSunatRestService.java | Acción |  |
| Nuevo/Modifica/Reusable | Métodos (endpoints) necesarios para el proceso de Login para la aplicación de escritorio. | | |
| Tipo | JAVA | | |
| Objetivo | Login de usuarios sunat (intranet) y devolución de token. | | |
| Especificación (seudocódigo) | | | |
| @Path(OAuth2ClientesSunatRestService.***INTEGRADOR\_PATH***)  **public** **class** OAuth2ClientesSunatRestService {  @Inject  **private** GeneradorToken generadorToken;  @Inject  **private** AutenticacionLdapService autenticacionLdapService;  @Inject  **private** AutorizacionLdapService autorizacionLdapService;  **public** **static** **final** String ***INTEGRADOR\_PATH*** = "/v1/clientessunat";  @POST  @Path("/{client\_id}/oauth2/token/")  @Consumes(MediaType.***APPLICATION\_FORM\_URLENCODED***)  @Produces({ MediaType.***APPLICATION\_JSON*** })  **public** Response tokenEndpoint(@PathParam(Constantes.***CLIENT\_ID***) String pathClientId,  @FormParam(Constantes.***GRANT\_TYPE***) String grantType, @FormParam(Constantes.***SCOPE***) String scope,  @FormParam(Constantes.***CLIENT\_ID***) String clientId, @FormParam(Constantes.***CLIENT\_SECRET***) String clientSecret,  @FormParam(Constantes.***USERNAME***) String username, @FormParam(Constantes.***PASSWORD***) String password) **throws** Exception {  **if** (!grantType.equals(Constantes.***PASSWORD***)) {  OAuthResponse response = OAuthASResponse.*errorResponse*(HttpServletResponse.***SC\_BAD\_REQUEST***)  .setError(OAuthError.TokenResponse.***INVALID\_GRANT***).setErrorDescription(Constantes.***GRANT\_TYPE\_ERROR***)  .buildJSONMessage();  **return** Response.*status*(response.getResponseStatus()).entity(response.getBody()).build();  } **else** **if** (clientId.equals(**null**) && clientSecret.equals(**null**) && username.equals(**null**)  && password.equals(**null**)) {  OAuthResponse response = OAuthASResponse.*errorResponse*(HttpServletResponse.***SC\_BAD\_REQUEST***)  .setError(OAuthError.TokenResponse.***INVALID\_REQUEST***).setErrorDescription(Constantes.***INVALID\_PARAM\_ERROR***)  .buildJSONMessage();  **return** Response.*status*(response.getResponseStatus()).entity(response.getBody()).build();  } **else** **if** (!generadorToken.validarClienteIntranet(clientId, clientSecret)) {  OAuthResponse response = OAuthASResponse.*errorResponse*(HttpServletResponse.***SC\_BAD\_REQUEST***)  .setError(OAuthError.TokenResponse.***UNAUTHORIZED\_CLIENT***).setErrorDescription(Constantes.***CLIENT\_AUTHORIZATION\_ERROR***)  .buildJSONMessage();  **return** Response.*status*(response.getResponseStatus()).entity(response.getBody()).build();  } **else** {  // ---------- PRUBA PARA LDAP ---------- //  // \*\*\*\*\* Autenticacion \*\*\*\*\*\*\*\*\*  UsuarioBean user = autenticacionLdapService.autenticarUsuario(username, password);  // \*\*\*\*\* Autorizacion \*\*\*\*\*\*\*\*\*  RespAutorBean autorBean = autorizacionLdapService.autorizarUsuario(user);  String[] token = generadorToken.generarSignedJWTIntranet(clientId, grantType, ***INTEGRADOR\_PATH***, autorBean.getUsuarioBean(),  autorBean.getPerfiles());      OAuthResponse response = OAuthASResponse.*tokenResponse*(HttpServletResponse.***SC\_OK***).setAccessToken(token[0])  .setTokenType(token[1]).setExpiresIn(token[2]).buildJSONMessage();  **return** Response.*status*(response.getResponseStatus()).entity(response.getBody()).build();  }  }  } | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Especificación de la unidad de construcción | | | |
| Nombre | OAuth2ClientesExtranetRestService.java | Acción |  |
| Nuevo/Modifica/Reusable | Nuevo: métodos (endpoints) necesarios para el proceso de Login para la aplicación de escritorio. | | |
| Tipo | JAVA | | |
| Objetivo | Login de clientes externos y devolución de token. | | |
| Especificación (seudocódigo) | | | |
| @Path(OAuth2ClientesExtranetRestService.***INTEGRADOR\_PATH***)  **public** **class** OAuth2ClientesExtranetRestService {  @Inject  **private** GeneradorToken generadorToken;  @Inject  **private** UtilLog utilLog;  **public** **static** **final** String ***INTEGRADOR\_PATH*** = "/v1/clientesextranet";  @POST  @Path("/{client\_id}/oauth2/token/")  @Consumes(MediaType.***APPLICATION\_FORM\_URLENCODED***)  @Produces({ MediaType.***APPLICATION\_JSON*** })  **public** Response tokenEndpoint(@PathParam(Constantes.***CLIENT\_ID***) String pathClientId,  @FormParam(Constantes.***GRANT\_TYPE***) String grantType, @FormParam(Constantes.***SCOPE***) String scope,  @FormParam(Constantes.***CLIENT\_ID***) String clientId, @FormParam(Constantes.***CLIENT\_SECRET***) String clientSecret) **throws** Exception {    // \*\*\*\*\*\* Validar parámetros \*\*\*\*\*\*\*\*\*  **if** (!grantType.equals(Constantes.***GRANT\_TYPE\_EXTRANET***)) {  OAuthResponse response = OAuthASResponse.*errorResponse*(HttpServletResponse.***SC\_BAD\_REQUEST***)  .setError(OAuthError.TokenResponse.***INVALID\_GRANT***).setErrorDescription(Constantes.***GRANT\_TYPE\_ERROR***)  .buildJSONMessage();  **return** Response.*status*(response.getResponseStatus()).entity(response.getBody()).build();  } **else** **if** (clientId.equals(**null**) && clientSecret.equals(**null**)) {  OAuthResponse response = OAuthASResponse.*errorResponse*(HttpServletResponse.***SC\_BAD\_REQUEST***)  .setError(OAuthError.TokenResponse.***INVALID\_REQUEST***).setErrorDescription(Constantes.***INVALID\_PARAM\_ERROR***)  .buildJSONMessage();  **return** Response.*status*(response.getResponseStatus()).entity(response.getBody()).build();  } **else** **if** (!generadorToken.validarClienteExtranet(clientId, clientSecret)) {  OAuthResponse response = OAuthASResponse.*errorResponse*(HttpServletResponse.***SC\_BAD\_REQUEST***)  .setError(OAuthError.TokenResponse.***UNAUTHORIZED\_CLIENT***).setErrorDescription(Constantes.***CLIENT\_AUTHORIZATION\_ERROR***)  .buildJSONMessage();  **return** Response.*status*(response.getResponseStatus()).entity(response.getBody()).build();    } **else** **if** (!generadorToken.validarAudiencia(scope)) {  OAuthResponse response = OAuthASResponse.*errorResponse*(HttpServletResponse.***SC\_BAD\_REQUEST***)  .setError(OAuthError.TokenResponse.***INVALID\_SCOPE***).setErrorDescription(Constantes.***SCOPE\_ERROR***)  .buildJSONMessage();  **return** Response.*status*(response.getResponseStatus()).entity(response.getBody()).build();    } **else** {    // \*\*\*\*\*\* Generación Token \*\*\*\*\*\*\*\*\*  utilLog.~~imprimirLog~~(ConstantesUtils.***LEVEL\_INFO***, "OAUTH2 ==>> [OAuth2ClientesExtranetRestService] - Se inicia la generacion del token => generarSignedJWTExtranet()");  String[] token = generadorToken.generarSignedJWTExtranet(clientId, clientSecret,grantType,scope, ***INTEGRADOR\_PATH***);  OAuthResponse response = OAuthASResponse.*tokenResponse*(HttpServletResponse.***SC\_OK***).setAccessToken(token[0])  .setTokenType(token[1]).setExpiresIn(token[2]).buildJSONMessage();  **return** Response.*status*(response.getResponseStatus()).entity(response.getBody()).build();  }  }  } | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Especificación de la unidad de construcción | | | |
| Nombre | OAuthConfig.java | Acción |  |
| Nuevo/Modifica/Reusable | Nuevo: Bean para el archivo de configuración | | |
| Tipo | JAVA | | |
| Objetivo | Cargar la variables que se utilizan para la configuración del token y conexión ldap. | | |
| Especificación (seudocódigo) | | | |
| **public** **class** OAuthConfig **extends** MicroserviceConfig {  **private** **static** OAuthConfig *config*;  **private** String urlAgenteGestorSesion;  **private** JWTConfig jwtconfig;  **private** LdapConfig ldapConfig;  **public** **static** OAuthConfig getConfig() {  **return** *config*;  }  **public** **void** loadConfig() {  OAuthConfig.*config* = **this**;  }  **public** String getUrlAgenteGestorSesion() {  **return** urlAgenteGestorSesion;  }  **public** **void** setUrlAgenteGestorSesion(String urlAgenteGestorSesion) {  **this**.urlAgenteGestorSesion = urlAgenteGestorSesion;  }  **public** JWTConfig getJwtconfig() {  **return** jwtconfig;  }  **public** **void** setJwtconfig(JWTConfig jwtconfig) {  **this**.jwtconfig = jwtconfig;  }  **public** LdapConfig getLdapConfig() {  **return** ldapConfig;  }  **public** **void** setLdapConfig(LdapConfig ldapConfig) {  **this**.ldapConfig = ldapConfig;  }  } | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Especificación de la unidad de construcción | | | |
| Nombre | ControlAccesoApplication.java | Acción |  |
| Nuevo/Modifica/Reusable | Modifica: Se añade la clase OAuth2DesktopRestService y se modifica el Bean de configuración OAuthConfig. | | |
| Tipo | JAVA | | |
| Objetivo | Crear los endpoints necesarios para la aplicación de escritorio.  Añadir como variable la URL del gestor de sesión en el archivo config.yaml | | |
| Especificación (seudocódigo) | | | |
| **public** **class** ControlAccesoApplication **extends** SunatApplication<OAuthConfig> {    @Override  **public** **void** onRun(OAuthConfig myConfig, Environment environment) **throws** Exception {  environment.jersey().register(~~ValidationViolationExceptionMapper~~.**class**);  environment.jersey().register(~~ObjectNotFoundExceptionMapper~~.**class**);  environment.jersey().register(~~GenericExceptionMapper~~.**class**);  environment.jersey().register(OAuth2ClientesSolRestService.**class**);  environment.jersey().register(OAuth2ClientesSunatRestService.**class**);  environment.jersey().register(OAuth2ClientesExtranetRestService.**class**);  myConfig.loadConfig();  }    @Override  **public** **void** onInitialize (Bootstrap<OAuthConfig> bootstrap) {  //Empty implementation for default  bootstrap.addBundle(**new** ViewBundle<OAuthConfig>());  }    **public** **static** **void** main(String... params) **throws** Exception {  **new** ControlAccesoApplication().run(params);  }  } | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Especificación de la unidad de construcción | | | |
| Nombre | Config.yaml | Acción |  |
| Nuevo/Modifica/Reusable | Modifica: añade la URL del agente gestor sesión como variable. | | |
| Tipo | YAML | | |
| Objetivo | - | | |
| Especificación (seudocódigo) | | | |
| jwtconfig:  defaultIssuer: ${JWT\_DEFAULT\_ISSUER}  privateKey: ${JWT\_PRIVATE\_KEY}  privateKeyExtranet: ${JWT\_PRIVATE\_KEY\_EXTRA}  privateKeyIntranet: ${JWT\_PRIVATE\_KEY\_INTRA} | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Especificación de la unidad de construcción | | | |
| Nombre | gradle.properties | Acción |  |
| Nuevo/Modifica/Reusable | Modifica: se modifica la versión y código del pase. | | |
| Tipo | GRADLE | | |
| Objetivo | - | | |
| Especificación (seudocódigo) | | | |
| version=1.0.0-4  paseNro='PAS20181U210300196'  releaseNro=1 | | | |

### Especificación de componentes (SOA) (DSI 8.4)

**No Aplica.**

## DISEÑO DE LA MIGRACION Y CARGA INICIAL DE DATOS (DSI 10)

**No Aplica.**

* Especificación del Entorno de Migración y Carga Inicial de Datos.
* Definición de Procedimientos de Migración y Carga Inicial.
* Diseño detallado de Módulos de Migración y Carga Inicial.
* Especificación Técnica de las Pruebas de Migración y Carga Inicial

## ESPECIFICACION DE REQUERIMIENTOS DEDOCUMENTACION DE USUARIO (DSI 11)

**No Aplica.**

|  |  |  |
| --- | --- | --- |
| Nro. | Documento | Descripción |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

## DEFINICION DEL PLAN DE IMPLANTACION (DSI 12)

**No Aplica.**

|  |  |  |
| --- | --- | --- |
| Nro. | Requisitos de Implantación | Descripción |
|  |  |  |
|  |  |  |