

Análisis del Servicio "calculate" - OnlineKpi

Fecha de Creación: 28 de Octubre de 2025

Versión del Documento: 1.0

Endpoint: POST /srv/onlinekpi/calculate

Clase: cl.cgs.server.webservices.services.OnlineKpi

Método: calculatekpi(ParamDataOKpi param)

1. TABLAS DE BASE DE DATOS UTILIZADAS

El servicio consulta las siguientes tablas de la base de datos:

1.1. Tabla plantas

Propósito: Obtiene todas las plantas activas (no eliminadas) del sistema para realizar los cálculos de KPIs.

Columnas utilizadas:

- nickname - Identificador único de la planta
- eliminado - Para filtrar plantas activas (WHERE eliminado = false)
- mes_fiscal - Mes de inicio del año fiscal de la planta
- dia_corte - Día de corte del año fiscal
- hora_reporte - Hora de corte para los reportes
- nombre_rmes_file - Nombre del archivo RMES con la estructura de equipos

Consulta ejecutada:

```
plantasDao.findByEliminado(false)
```

Ubicación en código: SrvOnlineKpi.java , línea 76

1.2. Tabla ubicacion_tecnica

Propósito: Proporciona el mapeo entre los códigos RMES de equipos y los nombres SAP para mostrar información más amigable al usuario.

Columnas utilizadas:

- `nick_equipo_rmes` - Código/nickname del equipo en el sistema RMES
- `nombre_sap` - Nombre del equipo en el sistema SAP
- `nombre_equipo_rmes` - Nombre del equipo en RMES
- `nodo_padre_rmes` - Nodo padre del equipo en la jerarquía
- `nick_nodo_padre_rmes` - Nickname del nodo padre

Consulta ejecutada:

```
ubicacionTecnicaDao.findAll()
```

Uso: Se crea un diccionario HashMap que mapea `nickEquipoRmes` → `nombreSap` para traducir los códigos de equipos a nombres legibles.

Ubicación en código: `SrvOnlineKpi.java`, líneas 162-170

1.3. Tabla `server_config`

Propósito: Almacena la configuración del servicio OnlineKpi en formato JSON.

Columnas utilizadas:

- `id_server_config` - ID único del registro
- `name` - Nombre de la configuración (debe ser "ServerConfOnlineKpi")
- `value` - Valor JSON con la configuración completa

Configuraciones extraídas del JSON:

Parámetro	Tipo	Valor por Defecto	Descripción
<code>horaReporte</code>	String	"08:00:00"	Hora de corte para los cálculos diarios
<code>maxDayRange</code>	Integer	10	Máximo número de días permitidos en el período de consulta
<code>isDebug</code>	Boolean	false	Activa el modo debug con logging extendido
<code>ssImpact</code>	Array	[]	Configuración de impacto sistémico por subsistemas

Estructura JSON esperada en server_config.value :

```
{
  "horaReporte": "08:00:00",
  "maxDayRange": 10,
  "isDebug": false,
  "ssImpact": [
    {
      "rootNickName": "Planta-Concentradora",
      "nodes": ["Chancado-Primario", "Molienda", "Flotacion"]
    },
    {
      "rootNickName": "Planta-Electrica",
      "nodes": ["Generacion", "Distribucion"]
    }
  ]
}
```

Consulta ejecutada:

```
scc.findByName("ServerConfOnlineKpi")
```

Ubicación en código: ServerConfOnlineKpi.java , líneas 33-141

2. INDICADORES / KPIs CALCULADOS

El servicio calcula los siguientes KPIs para cada equipo, desglosados por día:

2.1. Tiempos (en horas)

KPI	Campo JSON	Descripción
Tiempo de Producción	productionTime	Tiempo de uso/producción del equipo en el intervalo
Tiempo de Producción FYTD	productionTimeFYTD	Tiempo de producción acumulado desde inicio del año fiscal
Tiempo Calendario	calendarTime	Duración total del intervalo en horas
Tiempo Disponible	availableTime	Tiempo disponible del equipo

2.2. Tiempos de Inactividad (Downtime)

Clasificación de las detenciones según su causa:

KPI	Campo JSON	Código	Descripción
Mantenimiento Programado	se	MP	MpTime - Detenciones por mantenimiento planificado
Detención Operacional	sp	DO	DoTime - Detenciones operacionales programadas
Mantenimiento Correctivo	ue	MC	McTime - Detenciones por fallas no programadas
Detención Operacional No Programada	up	DONP	DonpTime - Detenciones operacionales no planificadas
Detención Causa Externa	st	DCE	DceTime - Detenciones por causas externas
Tiempo Total de Inactividad	equipmentDowntime	-	Suma de SE+SP+UE+UP+ST

2.3. Indicadores de Rendimiento (Porcentajes/Ratios)

KPI	Campo JSON	Fórmula	Descripción
Disponibilidad	availability	Avail	Porcentaje de disponibilidad del equipo
Utilización Efectiva	utilization	EffectiveUsage	Porcentaje de utilización efectiva
Utilización General	overallUtilization	productionTime / calendarTime	Ratio de uso vs tiempo total

2.4. Impacto Sistémico

KPI	Campo JSON	Descripción
Impacto en Proceso	processDowntime	Tiempo de inactividad del equipo que impacta al proceso completo (calculado según configuración <code>ssImpact</code>)

2.5. Metadatos del Equipo

Campo	Descripción
codigoRmes	Nickname/código del equipo en el sistema RMES
nombreEquipoRmes	Nombre del equipo (obtenido de ubicación técnica SAP si existe)
inicio	Fecha/hora de inicio del intervalo (formato: "dd-MM-yyyy hh:mm:ss.SSS")
fin	Fecha/hora de fin del intervalo (formato: "dd-MM-yyyy hh:mm:ss.SSS")

2.6. Estructura de la Respuesta

```
{
  "inicioPeriodo": "01-10-2025",
  "finPeriodo": "10-10-2025",
  "equipments": [
    {
      "equipmentName": "Chancador Primario 01",
      "data": [
        {
          "inicio": "01-10-2025 08:00:00.000",
          "fin": "02-10-2025 08:00:00.000",
          "codigoRmes": "2110-CR-001",
          "nombreEquipoRmes": "Chancador Primario 01",
          "productionTime": 20.5,
          "productionTimeFYTD": 4890.3,
          "se": 1.5,
          "sp": 0.5,
          "ue": 1.0,
          "up": 0.3,
          "st": 0.2,
          "calendarTime": 24.0,
          "availableTime": 22.5,
          "overallUtilization": 0.854,
          "availability": 0.938,
          "utilization": 0.911,
        }
      ]
    }
  ]
}
```

```

        "equipmentDowntime": 3.5,
        "processDowntime": 2.1
    },
    {
        "inicio": "02-10-2025 08:00:00.000",
        "fin": "03-10-2025 08:00:00.000",
        ...
    }
]
},
{
    "equipmentName": "Molino SAG 01",
    "data": [...]
}
]
}

```

Ubicación en código:

- Cálculo de KPIs: `ResponseEquipmentDataCreator.java` , líneas 56-111
- Estructura de respuesta: `ResponseOnlineKpiCalculate.java`

3. RESTRICCIONES DEL PERÍODO DE CÁLCULO

El servicio aplica las siguientes validaciones sobre las fechas ingresadas:

3.1. Formato de Fechas

Elemento	Formato Requerido	Ejemplo	Observaciones
Fecha	dd-MM-yyyy	"28-10-2025"	Día-Mes-Año con guiones
Hora	hh:mm:ss	"08:00:00"	Formato 24 horas
Año	yyyy	"2025"	4 dígitos obligatorios

Validación: El parser de fechas es estricto (`setLenient(false)`), por lo que fechas como "32-01-2025" serán rechazadas.

3.2. Restricciones Temporales

3.2.1. Fechas Diferentes

Validación: `fechaInicio` NO puede ser igual a `fechaTermino`

Error:

"`fechaInicio` y `fechaTermino` son iguales"

Código: `ValidatorDate.java`, línea 125

3.2.2. Orden Correcto

Validación: `fechaInicio` debe ser ANTERIOR a `fechaTermino`

Error:

"La fecha de `fechaTermino`: [fecha] es anterior a la fecha de `fechaInicio`: [fecha]"

Código: `ValidatorDate.java`, líneas 137-140

3.2.3. No Permitir Fechas Futuras

Validación: Tanto `fechaInicio` como `fechaTermino` deben ser menores o iguales a la fecha actual (ajustada al final del día)

Errores:

"La fecha de `fechaInicio`: [fecha] es posterior a la fecha actual, esto no es permitido."

"La fecha de `fechaTermino`: [fecha] es posterior a la fecha actual, esto no es permitido."

Código: `ValidatorDate.java`, líneas 129-136

3.2.4. Rango Máximo de Días

Validación: El período seleccionado no puede superar el valor configurado en `maxDayRange`

Valor por defecto: 10 días

Configuración: Se puede modificar en la tabla `server_config` (nombre="ServerConfOnlineKpi", campo JSON `maxDayRange`)

Deshabilitación: Configurar `maxDayRange = -1` para permitir cualquier rango

Error:

"El período seleccionado supera los `[maxDayRange]` días"

Cálculo: El servicio divide el período en intervalos de 1 día y cuenta la cantidad. Por ejemplo:

- Desde "01-10-2025" hasta "11-10-2025" = 10 intervalos de días → **PERMITIDO** (si `maxDayRange=10`)
- Desde "01-10-2025" hasta "12-10-2025" = 11 intervalos de días → **RECHAZADO** (si `maxDayRange=10`)

Código: `ValidatorDate.java`, líneas 141-145

3.3. Otras Validaciones

3.3.1. Validación de Plantas

Requisito: Debe existir al menos una planta activa en la base de datos (donde `eliminado = false`)

Validación: `ValidatorPlanta.validate(plantas)`

Código: `SrvOnlineKpi.java`, línea 80

3.3.2. Validación de Token

Requisito: El token enviado debe ser válido y no estar caducado

Validación: `tokenValidator.isValid(param.getToken())`

Error:

"Token invalido, no existe en la base de datos o bien se encuentra caducado"

Código: `OnlineKpi.java`, líneas 37-41

3.4. Resumen de Validaciones

```
VALIDACIONES DEL SERVICIO calculate
```

1. Token válido y no caducado
2. Formato de fechas: dd-MM-yyyy
3. Año con 4 dígitos (yyyy)
4. fechaInicio ≠ fechaTermino
5. fechaInicio < fechaTermino
6. fechaInicio ≤ fecha actual
7. fechaTermino ≤ fecha actual
8. Rango de días ≤ maxDayRange (default: 10)
9. Existe al menos una planta activa

Ubicación en código: ValidatorDate.java , líneas 107-146

4. EJEMPLO DE USO

Request

```
POST /srv/onlinekpi/calculate HTTP/1.1  
Content-Type: application/json
```

```
{  
  "token": "abc123xyz789",  
  "fechaInicio": "01-10-2025",  
  "fechaTermino": "10-10-2025"  
}
```

Response Exitosa (200 OK)

```
{  
  "inicioPeriodo": "01-10-2025",  
  "finPeriodo": "10-10-2025",  
  "equipments": [  
    {  
      "equipmentName": "Chancador Primario 01",  
      "data": [  
        {  
          "inicio": "01-10-2025 08:00:00.000",  
          "fin": "02-10-2025 08:00:00.000",  
          "codigoRmes": "2110-CR-001",  
          "nombreEquipoRmes": "Chancador Primario 01",
```

```
    "productionTime": 20.5,
    "productionTimeFYTD": 4890.3,
    "se": 1.5,
    "sp": 0.5,
    "ue": 1.0,
    "up": 0.3,
    "st": 0.2,
    "calendarTime": 24.0,
    "availableTime": 22.5,
    "overallUtilization": 0.854,
    "availability": 0.938,
    "utilization": 0.911,
    "equipmentDowntime": 3.5,
    "processDowntime": 2.1
  }
]
}
```

Response con Error (500)

"Token invalido, no existe en la base de datos o bien se encuentra caducado"

o

"El período seleccionado supera los 10 días"

5. NOTAS IMPORTANTES

1. **Cálculo por Día:** El servicio siempre divide el período en intervalos de 1 día completo (24 horas).
2. **Año Fiscal:** El cálculo de FYTD (Fiscal Year To Date) usa el año fiscal de la **primera planta** encontrada. Si hay plantas con diferentes años fiscales, se debe modificar el algoritmo.
3. **Impacto Sistémico:** Solo se calcula si hay configuración de `ssImpact` en `server_config`. Si no existe, `processDowntime` será 0.
4. **Hora de Corte:** La hora configurada en `horaReporte` se aplica a todas las fechas para determinar los límites de los intervalos diarios.

5. **Ubicación Técnica:** El diccionario de ubicación técnica es opcional. Si no existe un mapeo para un equipo, se usa el nombre del equipo desde el archivo RMES.
6. **Performance:** El tiempo de procesamiento se registra en los logs. Para períodos largos (10 días) con muchos equipos, el procesamiento puede tomar varios segundos o minutos.