

Obtiene descarga Renovación de contrato (3) - [getRenovacionContrato]

📄 Descripción

Valida si un empleado debe **descargar la renovación de contrato** antes de poder registrar marcaciones o realizar solicitudes dentro del aplicativo.

El servicio evalúa:

- La fecha actual (solo aplica desde el día **26 de cada mes**).
- La fecha de ingreso del empleado.
- Si el empleado tiene un proceso de renovación registrado y validado en ERP.
- Si ya existe un registro de descarga previo en `historial_procesos_app`.

Este servicio actúa como una **regla de negocio obligatoria** para habilitar o bloquear el uso del aplicativo.

? Endpoint

POST /get-renovacion-contrato

📄 Recibe un único parámetro:

```
{  
  "empleado": "EMP-0001"  
}
```

? Seguridad

- Requiere credenciales autorizadas para consumir la API del ERP.
- Utiliza internamente `dbErp()` para ejecutar consultas SQL seguras contra ERPNext.
- Accede a `mysql2` para verificar registros de descargas.

? Flujo del Servicio (Resumen Real)

1?? Validación inicial por fecha

- Solo permite validar renovación desde el **día 26** del mes.
Si es antes:

```
{  
  "status": true,  
  "msn": "No es el día correcto para la renovación de contrato"  
}
```

2?? Obtener fecha real de ingreso del empleado

Consulta al ERP:

```
SELECT fecha_de_ingreso_real  
FROM `tabEmployee`  
WHERE name = <empleado>
```

Si no existe → el servicio simplemente finaliza sin error.

3?? Determinar si el empleado ya debería tener renovación

- Calcula el último día del **mes anterior**.
- Si el ingreso es anterior a ese período, el empleado califica para renovación.

4?? Consultar renovación en ERP (contratos validados)

Busca registros de:

- Mes actual (en texto: Enero, Febrero...)
- Año del mes previo
- Código del empleado
- `renueva = Si`
- `estado_de_documento = Validado`

Consulta ejecutada:

```
SELECT doc.name
FROM `tabSolicitud de Renovaciones` doc
LEFT JOIN `tabTrabajadores pendiente de renovar` tab ON doc.name = tab.parent
WHERE
  doc.data_12 = <mes_actual_texto> AND
  doc.año = <year> AND
  tab.codigo = <empleado> AND
  tab.renueva = 'Si' AND
  doc.estado_de_documento = 'Validado'
```

5?? Validar si el empleado YA descargó la renovación

Consulta en base de datos interna:

```
historial_procesos_app
WHERE proceso = 'descargaContratoRenovacion'
AND empleado = <empleado>
```

- Si **NO la descargó** → Bloquea acciones del app:

```
{
  "status": false,
  "msn": "Para poder registrar su marcación o realizar alguna solicitud, descargar su renovación de contrato"
}
```

6?? Si todo está correcto

Retorna éxito:

```
{
  "status": true,
  "msn": "Se ha descargado el contrato de trabajo"
}
```

? Request Body

```
{  
  "empleado": "EMP-0001"  
}
```

? Response 200 – Ejemplos

? Caso correcto (ya descargó renovación)

```
{  
  "status": true,  
  "msn": "Se ha descargado el contrato de trabajo"  
}
```

? Debe descargar renovación antes de usar el app

```
{  
  "status": false,  
  "msn": "Para poder registrar su marcación o realizar alguna solicitud, descargar su renovación de contrato"  
}
```

? Aún no es día válido

```
{  
  "status": true,  
  "msn": "No es el día correcto para la renovación de contrato"  
}
```

? Posibles Errores

| Error | Respuesta |
|-------|-----------|
|-------|-----------|

| | |
|----------------------------------|---|
| Día inválido para la operación | "No es el día correcto para la renovación de contrato" |
| Empleado sin renovación validada | "Para poder registrar su marcación..." |
| Error inesperado en ejecución | Se devuelve mensaje genérico, pero siempre con status: true |
| No existe el empleado | El flujo simplemente finaliza sin error explícito |

? Consultas y estructuras usadas

? ERP — Employee

Campos consultados:

```
{  
  "fecha_de_ingreso_real": "date"  
}
```

? ERP — Solicitud de Renovaciones

Consulta combinada mediante JOIN:

- data_12 (mes)
- año
- codigo
- renueva
- estado_de_documento

? Base interna MySQL (mysql2)

Tabla: historial_procesos_app

Campo crítico: proceso = 'descargaContratoRenovacion'

? Lógica en Pseudo-código

```
if (dia_actual < 26)
    return "No es el día correcto"

empleado = GET fecha_ingreso_real FROM Employeee

if fecha_ingreso < ultimo_dia_mes_anterior:
    renovacion = GET renovacion_validada FROM SolicitudRenovaciones

    if renovacion existe:
        descarga = buscar en historial_procesos_app

        if no hay descarga:
            return "Debe descargar renovación"

return "OK"
```

Revisión #1

Creado 2025-11-25 16:31:19 -05 por Elian

Actualizado 2025-11-25 16:35:30 -05 por Elian