

# Obtiene descarga Renovación de contrato (1,2,3,4,5,6,7,8) - [getRenovacionContrato]

## ? Descripción

Valida si un trabajador tiene pendiente la descarga de su **renovación de contrato**, lo cual puede bloquear el registro de marcaciones o solicitudes dentro del aplicativo.

El servicio evalúa fechas, estado del trabajador y el registro interno en base MySQL donde se guarda el historial de descargas realizadas.

## ? Endpoint

GET /get-renovacion-contrato/{empleado}

“ {empleado} ” corresponde al código del empleado (Employee.name en ERPNext).

## ? Seguridad

- Requiere que el cliente esté autenticado (según arquitectura de la app).
- Accede a ERP (Frappe) mediante un servicio interno (dbErp).

## ? Parámetros

Parámetro	Tipo	Requerido	Descripción
-----------	------	-----------	-------------

empleado	string	Sí	Código único del empleado en ERPNext.
----------	--------	----	---------------------------------------

## ? Responses

### ?? 200 – No corresponde validar renovación (fuera de fecha)

El proceso solo se ejecuta si **día**  $\geq$  26.

Si es antes, devuelve:

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

### ?? 200 – Validación NO necesaria (no aplica por antigüedad)

Si el empleado ingresó después del mes previo:

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

### ? 200 – Debe descargar el contrato de renovación (pendiente)

El empleado tiene renovación generada y habilitada, pero **no tiene registro de descarga**:

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

## ?? 200 – Renovación ya descargada

La base MySQL registra descarga previa:

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

## ?? 200 – Error controlado

Ante cualquier excepción:

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

# ? Flujo del Servicio

1. Verifica el **día actual**
  - Si día < 26 → no corresponde validar
2. Obtiene datos del empleado desde ERP ( `fecha_de_ingreso_real` )
3. Calcula:
  - Último día del mes anterior
  - Mes actual (texto)
4. Si el ingreso del empleado es **anterior al último día del mes previo**, aplica validación.
5. Consulta en ERP:
  - Solicitudes de renovación para este empleado
  - Con criterios:
    - Mes
    - Año
    - Código del empleado

- `renueva = "Si"`
  - documento validado
6. Si tiene renovación pendiente:
- Consulta base MySQL2 → tabla `historial_procesos_app`
  - Busca registro del proceso `descargaContratoRenovacion`
  - Si no existe → debe descargar renovación
7. Si ya está registrado → permitir normal funcionamiento.
- 

# ? Consultas utilizadas

## A. Obtener datos del empleado

```
SELECT fecha_de_ingreso_real  
FROM `tabEmployee`  
WHERE name = {empleado}
```

## B. Consulta de renovaciones

```
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}  
AND doc.año = {year}  
AND tab.codigo = {empleado}  
AND tab.renueva = 'Si'  
AND doc.estado_de_documento = 'Validado'
```

## C. Validación de descarga

```
SELECT *  
FROM historial_procesos_app  
WHERE empleado = {empleado}  
AND proceso IN ('descargaContratoRenovacion')
```

---

# ? Ejemplo de consumo (HTTP)

Request:

```
GET /get-renovacion-contrato/EMP-00045
```

Response (pendiente):

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

---

Revisión #6

Creado 2025-11-25 14:49:46 -05 por Elian

Actualizado 2025-11-26 16:04:18 -05 por Elian