

# Documento de Ingresos (Subir Documentos de ingresos (1)) - [documentosIngreso]

## ? Descripción

Registra y actualiza los **documentos obligatorios de ingreso** de un empleado dentro del ERP, así como su información bancaria y de fondo de pensiones.

Este servicio valida la información mínima necesaria antes de actualizar el documento Employee en el ERP.

☐ Es un servicio crítico para el proceso de onboarding, ya que sin este registro el empleado no continúa con otros procesos internos.

---

## ? Endpoint

POST `/documentos-ingreso`

---

## ? Parámetros de entrada (Request Body)

```
{
  "employee": "EMP-00123",
  "banco": "BCP",
  "dni": "12345678",
  "copia_dni_de_hijos": "url_file",
  "certificado_de_estudios": "url_file",
  "cv": "url_file",
  "copia_dni_de_conyuge": "url_file",
  "certificado_de_trabajo": "url_file",
  "certijoven": "url_file",
  "copia_de_recibo": "url_file",
  "fondo_pensiones": "AFP",
  "afiliado_fondo_pensiones": "SI"
}
```

## ? Seguridad

Requiere token válido del ERP (la autenticación se maneja internamente mediante `ServiceErp()`).

## ? Flujo del Servicio (Resumen Real)

1. **Valida que existan campos obligatorios:**
  - employee
  - banco
  - fondo\_pensiones
  - afiliado\_fondo\_pensiones ("SI" o "NO")
2. Construye un arreglo con los datos a actualizar:

```
$data = [
  "eleccion_banco" => banco,
  "elección_fondo_pensiones" => fondo_pensiones,
  "afiliado_fondo_pensiones" => afiliado_fondo_pensiones,
  ...
];
```

3. Si el fondo de pensiones es **ONP**, también actualiza:

```
"fondo_de_pensiones" => "ONP"
```

4. Si se envían documentos opcionales (DNI, copia de DNI de hijos, CV, certificado de trabajo, etc.), también se agregan al payload.
5. Realiza la actualización del empleado:

PUT /resource/Employee/{employee}

6. Devuelve la respuesta del ERP junto con los datos enviados.

---

## ? Response 200 – Ejemplo

```
{
  "valor": true,
  "msn": "Documentos registrados correctamente",
  "data": {
    "eleccion_banco": "BCP",
    "elección_fondo_pensiones": "AFP",
    "afiliado_fondo_pensiones": "SI",
    "dni": "12345678",
    "cv": "url_file",
    "certificado_de_trabajo": "url_file"
  }
}
```

---

## ? Posibles Errores

### 1. Falta el employee

```
{
  "valor": false,
  "msn": "Falta key employee"
}
```

### 2. Falta banco

```
{
  "valor": false,
  "msn": "El banco es obligatorio"
}
```

### 3. Falta fondo de pensiones

```
{
  "valor": false,
  "msn": "El fondo de pensiones es obligatorio"
}
```

## 4. Afiliado fondo pensiones inválido

```
{
  "valor": false,
  "msn": "Valor no permitido en Afiliado a Fondo de Pensiones"
}
```

## 5. Error del ERP al actualizar

```
{
  "valor": false,
  "msn": "Error al actualizar documentos",
  "data": { ...respuesta del ERP... }
}
```

# ? Data Model Usado

## Employee (PUT)

Campos actualizables:

```
{
  "eleccion_banco": "string",
  "elección_fondo_pensiones": "AFP|ONP",
  "afiliado_fondo_pensiones": "SI|NO",
  "dni": "string",
  "copia_dni_de_hijos": "string|fileurl",
  "certificado_de_estudios": "string|fileurl",
  "certijoven": "string|fileurl",
  "cv": "string|fileurl",
  "copia_dni_de_conyuge": "string|fileurl",
  "certificado_de_trabajo": "string|fileurl",
  "copia_de_recibo": "string|fileurl",
  "asignacion_familiar_2": 0|1,
  "fondo_de_pensiones": "AFP|ONP"
}
```

# ? Pseudocódigo de la lógica

```
if employee is empty → error
if banco is empty → error
if fondo_pensiones is empty → error
if afiliado_fondo_pensiones not in ("SI","NO") → error

data = {
  eleccion_banco: banco,
  elección_fondo_pensiones: fondo_pensiones,
  afiliado_fondo_pensiones: afiliado_fondo_pensiones
}

if fondo_pensiones == "ONP":
  data["fondo_de_pensiones"] = "ONP"

Agregar campos opcionales si existen

PUT Employee/{employee} with data

return respuesta del ERP + msn "Documentos registrados correctamente"
```

Revisión #1

Creado 2025-11-26 11:20:25 -05 por Elian

Actualizado 2025-11-26 11:24:37 -05 por Elian