

# Asignación Familiar(Solicitar asignacion familiar (1)) - [store]

## ? Descripción

Este servicio registra una **Solicitud de Asignación Familiar** para un empleado, validando previamente que:

1. Se hayan enviado las **dos imágenes del DNI** (anverso y reverso).
2. El empleado tenga un género válido y registrado.
3. No exista ya una asignación familiar duplicada para el mismo hijo (DNI del menor).
4. No existan más de dos solicitudes asociadas al mismo DNI de hijo (límite máximo permitido).

El registro se almacena directamente en el ERP mediante el recurso **Solicitud Asignacion Familiar**.

---

## ? Endpoint

**POST** `/solicitud-asignacion-familiar`

---

## ? Request Body

```
{
  "dni": "string",
  "fecha_hijo": "YYYY-MM-DD",
  "edad": 10,
  "empleado": "EMP-0001",
  "img_dni": "ruta/archivo1.png",
  "reverso": "ruta/archivo2.png",
  "nombre": "Nombre del hijo",
  "afiliacion_al_essalud": "SI"
}
```

---

# ? Seguridad

- ✓ Requiere autenticación interna vía `ServiceErp()` del ERP.
  - ✓ Se validan datos del empleado antes de crear la solicitud.
- 

# ? Flujo del Servicio (Resumen)

## 1. Validación de archivos obligatorios

Verifica que se envíen `img_dni` y `reverso`.

Si falta uno → retorna error.

## 2. Obtiene datos del empleado

Usa el método `verifyEmployeeGender($empleado)` para traer:

- género
- fecha de ingreso real

## 3. Revisa si ya existe una solicitud similar

Busca solicitudes para:

- el mismo DNI del hijo
- la misma fecha de ingreso del empleado

```
GET Solicitud Asignacion Familiar
```

```
filters: dni_hijo, fecha_ingreso
```

## 4. Valida duplicidades según reglas

- Si existe **1 solicitud previa**, compara géneros:
  - Si el género coincide →  ya tiene asignación familiar.
- Si existen **2 solicitudes previas** →  no puede registrar más.

## 5. Crea la nueva solicitud

Realiza un POST hacia: `POST /resource/Solicitud Asignacion Familiar`

Con los campos:

```
{
  "id_empleado": "...",
  "nombre_hijo": "...",
  "dni_hijo": "...",
  "edad_hijo": "...",
  "fecha_hijo": "...",
  "dni_doc": "...",
  "reverso": "...",
  "afiliacion_al_essalud": "..."
}
```

## 6. Retorna respuesta final

---

# ? Response 200 – Ejemplos

## ? Registro exitoso

```
{  
  "valor": true,  
  "msn": "Asignacion familiar registrada con exito"  
}
```

## ? Faltan imágenes del DNI

```
{  
  "valor": false,  
  "msn": "Debe de enviar la 2 imagenes del DNI"  
}
```

## ? Asignación ya existente para el DNI del menor

```
{  
  "valor": false,  
  "msn": "Este dni ya cuenta con una asignacion familiar"  
}
```

## ? DNI ya usado en dos asignaciones previas

```
{  
  "valor": false,  
  "msn": "Este dni ya cuenta con dos asignaciones familiares"  
}
```

## ? Error al registrar en el ERP

```
{  
  "valor": false,  
  "msn": "Ocurrió un error al registrar la solicitud de asignacion familiar"  
}
```

---

## ? Schemas utilizados

## ? Solicitud Asignacion Familiar (POST)

```
{
  "id_empleado": "string",
  "nombre_hijo": "string",
  "dni_hijo": "string",
  "edad_hijo": number,
  "fecha_hijo": "YYYY-MM-DD",
  "dni_doc": "string",
  "reverso": "string",
  "afiliacion_al_essalud": "SI/NO"
}
```

## ? Solicitud Asignacion Familiar (GET/FILTER)

```
{
  "name": "string",
  "id_empleado": "string",
  "dni_hijo": "string",
  "fecha_ingreso": "YYYY-MM-DD"
}
```

## ? Datos del empleado (verifyEmployeeGender)

```
{
  "gender": "M/F",
  "fecha_de_ingreso_real": "YYYY-MM-DD"
}
```

---

## ? Lógica en pseudocódigo

```
if falta img_dni o reverso:
    return error

genderEmployee = verifyEmployeeGender(empleado)

solicitudesPrevias = Buscar solicitudes por dni_hijo y fecha_ingreso

if count == 1:
    otherGender = verifyEmployeeGender(solicitudesPrevias.id_empleado)
    if same gender:
        return error: ya tiene asignación

if count >= 2:
    return error: límite alcanzado

body = datos de asignación familiar

guardar = POST Solicitud Asignacion Familiar

if error:
    return error

return éxito
```

---

Revisión #1

Creado 2025-11-26 15:14:15 -05 por Elian

Actualizado 2025-11-26 15:17:43 -05 por Elian