

# Obtener Usuario (1,2,3,4) - [getUser3]

## ? Descripción

Retorna la información del usuario ingresado en la app (empleado o estudiante).

El servicio consulta primero la tabla **Employee** del ERP.

Si no existe o no está activo, consulta la tabla **Student**.

Finalmente, devuelve la información personal, laboral/educativa, permisos y datos complementarios.

Funciona como un **servicio maestro de login/datos del usuario** para la aplicación móvil.

---

## ? Endpoint

POST /get-user3

“ El nombre real depende del archivo `routes/api.php`.

---

## ? Seguridad

- No se ve autenticación explícita, pero las consultas al ERP (`dbErp`, `ServiceErp`) requieren token/credenciales internas.
  - Se recomienda protegerlo con autenticación JWT o middleware.
- 

## ? Request Body

```
{  
  "username": "string"  
}
```

## Campos

| Campo    | Tipo   | Requerido | Descripción   |
|----------|--------|-----------|---|
| username | string | Sí        | Email o DNI/pasaporte. Si contiene "@" se asume email, de lo contrario documento. |

# ? Lógica Interna Detallada

## 1?? Determinar si buscar por email o documento

- Si el parámetro incluye "@" → buscar por `emp.user_id`
- Sino → buscar por `emp.passport_number`

## 2?? Consultar información del empleado en ERP

Consulta SQL con múltiples JOIN:

### Tablas involucradas:

- `tabEmployee` (principal)
- `tabUser`
- `tabBranch`
- `tabContrato de Trabajo`
- `tabDepartment`

Si no encuentra empleado → pasa a modo estudiante.

## ? Errores de empleado antes de consultar estudiante:

- No `valor` en la respuesta
  - `response` vacío  
→ continuar con búsqueda en Student
- 

## 3?? Consultar información del estudiante (fallback)

Si el usuario no es empleado, consulta en:

- `tabStudent`
- `tabUser`
- `tabBranch`
- `tabJob Applicant`
- `tabRequerimiento de Personal`
- `tabDesignation`
- `tabDepartment`

## Validaciones de estudiante:

- Si no existe →

```
{ "valor": false, "msn": "Sin Estudiante: ...", "data": [] }
```

- Si el usuario está deshabilitado (`us.enabled = 0`) →

```
{ "valor": false, "msn": "Usuario Deshabilitado...", "data": [...] }
```

Si todo está correcto → retorna data del estudiante.

---

## 4?? Procesamiento final de empleado

### Validaciones

- Si `status = Inactive` → usuario no puede usar la app

## Ajustes en la data:

- Formato de fecha de nacimiento → `mm-dd`
- Si contrato indeterminado → reemplazar texto
- Detectar si es renovación o primer contrato (ServiceErp)
- Normalizar campos:
  - `bank_ac_no`
  - `pets`
  - `tipo_usuario = "Empleado"`

---

## 5?? Respuesta final (Empleado o Estudiante)

Retorna la data enriquecida del usuario.

---

## ? Responses

### ?? 200 – Usuario encontrado

```
{
  "valor": true,
  "msn": "Si hay data",
  "data": [ ... ]
}
```

---

### ? Usuario no es empleado ni estudiante

```
{
  "valor": false,
  "msn": "Este usuario no es empleado",
  "data": []
}
```

---

## ? Usuario estudiante sin permisos

```
{
  "valor": false,
  "msn": "Sin Estudiante: Su usuario no tiene permisos para acceder al aplicativo. Contactar con soporte.",
  "data": []
}
```

---

## ? Usuario deshabilitado

```
{
  "valor": false,
  "msn": "Usuario Deshabilitado: Comuníquese con su administrador",
  "data": [ ... ]
}
```

---

## ? Empleado inactivo

```
{
  "valor": false,
  "msn": "Empleado Inactivo: Comuníquese con su administrador",
  "data": [ ... ]
}
```

---

## ? Schemas

Request

```
{  
  "username": "string"  
}
```

## Response (general)

```
{  
  "valor": "boolean",  
  "msn": "string",  
  "data": "array"  
}
```

## ? Ejemplo de uso (curl)

```
curl -X POST https://midominio.com/api/get-user3 \  
-H "Content-Type: application/json" \  
-d '{"username": "empleado@empresa.com"}'
```

## ?? Servicios ERP usados internamente

### 1. **send-query-database**

Método POST que ejecuta consultas SQL personalizadas en ERPNext.

### 2. **resource/Contrato de Trabajo**

Consulta REST para obtener contratos de trabajo.

# ? Observaciones técnicas (reales)

- Código mezcla SQL dinámico en body con filtros JSON → validar inyección.
  - Se usa `LIMIT 1` en employee, pero la consulta de contratos tiene `limit=None`.
  - Campos duplicados en `sql_query` (ej. `emp.bank_ac_no` aparece 2 veces).
  - Es un servicio **muy grande**, podría dividirse por responsabilidad.
- 

Revisión #2

Creado 2025-11-24 16:57:47 -05 por Elian

Actualizado 2025-11-26 11:43:57 -05 por Elian