

Obtener Usuario (3) - [getUser3]

? Descripción

Este servicio permite **obtener la información completa de un usuario**, ya sea **Empleado** o **Estudiante**, usando cualquiera de los siguientes identificadores:

- **Correo** (user_id)
- **DNI / Pasaporte** (passport_number)

La función consulta múltiples tablas del ERP y retorna la información enriquecida del usuario, incluyendo:

- Datos personales
- Datos laborales
- Contratos
- Imagen de usuario
- Estado actual
- Sucursal, puesto, departamento
- Datos adicionales (PETS, tipo de jornada, fondo de pensiones, etc.)

Si el usuario no es empleado, se intenta validar como estudiante.

El servicio aplica múltiples reglas de negocio internas sobre estado, tipo de usuario y permisos.

? Endpoint

POST /get-user-3

Body esperado

```
{  
  "username": "usuario@correo.com"  
}
```

O también:

```
{  
  "username": "12345678"  
}
```

? Seguridad

Este servicio requiere autenticación interna contra el ERP mediante `dbErp()` y `ServiceErp()` con token válido del sistema.

? Flujo del Servicio (Resumen Real)

1. Determinar tipo de búsqueda

Si `username` contiene @, se asume que es `user_id`.

Si no contiene, se usa como `passport_number` (DNI).

Se arma el filtro dinámicamente:

Caso	Campo filtrado
Correo	<code>emp.user_id</code>
DNI / Pasaporte	<code>emp.passport_number</code>

2. Buscar al usuario como *Empleado*

Consulta principal: `POST method/send-query-database`

Incluye joins a:

- `tabUser`
- `tabBranch`
- `tabContrato de Trabajo`
- `tabDepartment`

Si existe un registro válido:

- Se valida que NO esté inactivo.
- Se formatea fecha de nacimiento.
- Se determina si el contrato es renovación o primer contrato.

- Se normalizan campos (pets, cuenta haberes, vigencia del contrato, tipo_usuario = "Empleado").

☐ Si todo está correcto, el servicio termina aquí retornando la data del empleado.

3. Si NO existe empleado ? Buscar como *Estudiante*

Filtros dependen de correo o DNI.

Validaciones aplicadas:

- Debe estar habilitado (`enabled = 1`).
- Si no existe → acceso denegado.
- Si está deshabilitado → retorno con error.

Datos retornados incluyen:

- Nombre completo
 - Sucursal, departamento
 - Tipo de puesto (convertido desde convocatoria si aplica)
 - Imagen de usuario
 - Tipo_usuario = "Estudiante"
-

4. Retornar la data final

Si es empleado → retorna información laboral con contratos.

Si es estudiante → retorna información básica educativa.

? Response 200 – Ejemplo (Empleado)

```
{
  "valor": true,
  "msn": "Si hay data",
  "data": [{
    "name": "EMP-001",
    "employee_name": "Juan Pérez",
    "user_id": "juan.perez@empresa.com",
    "gender": "Male",
    "status": "Active",
    "tipo_usuario": "Empleado",
    "vigencia_contrato": "2024-01-01 hasta 2024-12-31",
    "contratacion": "renovacion",
    "pets": "0",
    "bank_ac_no": "1234567890"
  }]
}
```

? Posibles Errores

1. No existe como empleado ni estudiante

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

2. Empleado Inactivo

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

3. Usuario deshabilitado (estudiante)

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

4. Error al consultar ERP

```
{  
  "valor": false,  
  "msn": "Error al obtener la información",  
  "data": []  
}
```

? Tablas consultadas

Empleado

- tabEmployee
- tabUser
- tabBranch
- tabContrato de Trabajo
- tabDepartment

Estudiante

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

? Resultado Final (Estructura General)

Para empleados:

```
{
  "name": "EMP-001",
  "employee_name": "...",
  "user_id": "...",
  "gender": "...",
  "tipo_usuario": "Empleado",
  "contratacion": "primer_contrato | renovacion",
  "status": "Active | PreActivo | Inactive",
  "pets": "0",
  "vigencia_contrato": "...",
  "...otros campos"
}
```

Para estudiantes:

```
{
  "name": "ST-001",
  "employee_name": "...",
  "user_id": "...",
  "tipo_usuario": "Estudiante",
  "statusUser": 1,
  "...otros campos"
}
```

? Pseudocódigo del servicio

```
username = request.username

if username contains "@":
  filter by user_id
else:
  filter by passport_number

employee = buscarEmpleado()

if existe empleado:
  validar estado
  cargar contratos
  formatear datos
  return empleado
else:
  student = buscarEstudiante()

  si no existe o está deshabilitado:
    error

return student
```

Revisión #1

Creado 2025-11-25 15:16:02 -05 por Elian

Actualizado 2025-11-25 15:27:38 -05 por Elian