

Solicitudes de Pagos(Obtener Combos Conceptos (1)) - [getComboSolicitudPago]

📄 Descripción

Obtiene la lista de **conceptos permitidos** para un usuario específico, basándose en la tabla del ERP "**Validación de Pagos Gerencia**".

El servicio filtra los registros por el usuario ingresado y retorna un **array único de conceptos** asociados a dicho usuario.

Se utiliza para validar qué tipos de pagos puede solicitar un trabajador.

? Endpoint

POST /get-combo-solicitud-pago

? Request Body

Debe incluir el campo:

```
{
  "usuario": "correo_o_user_id"
}
```

? Validación

Si **no se envía usuario**, responde:

```
{
  "valor": true,
  "msn": "Debe de ingresar el usuario correctamente",
  "data": []
}
```

? Seguridad

El servicio usa el método interno `dbErp()` que requiere autenticación interna vía ERPNext. Por ello, se asume que el cliente ya cuenta con **token/credenciales válidas**.

? Flujo del Servicio (Explicación Real)

1. **Validar parámetro usuario**
 - Si está vacío → responde error de validación.
2. **Construir filtros para ERP**

```
WHERE usuario = <usuario>
```

3. **Consultar al ERP**

Usando:

```
tabValidacion de Pagos Gerencia
```

con el SQL:

- **SELECT** concepto
 - **FROM** `tabValidacion de Pagos Gerencia`
 - **WHERE** usuario = %(usuario)s
4. **Obtener resultados**
 - Se extrae la columna `concepto` de la respuesta.
 - Se eliminan duplicados mediante `array_unique`.
 5. **Retornar los conceptos permitidos.**
-

? Response 200 – Ejemplo

```
{
  "valor": true,
  "msn": "Se encontró los conceptos",
  "data": [
    "Bonificación",
    "Viáticos",
    "Movilidad",
    "Pago Extraordinario"
  ]
}
```

? Posibles Errores

1. Usuario no enviado

```
{
  "valor": true,
  "msn": "Debe de ingresar el usuario correctamente",
  "data": []
}
```

2. Usuario sin conceptos asignados

```
{
  "valor": true,
  "msn": "Se encontró los conceptos",
  "data": []
}
```

(El servicio no devuelve error — solo retorna lista vacía.)

3. Error del ERP

Si `dbErp()` falla internamente, el servicio responderá una lista vacía debido a su estructura actual.

? Tablas y Estructuras Usadas

Tabla: tabValidacion de Pagos Gerencia

Campos utilizados:

```
{  
  "usuario": "string",  
  "concepto": "string"  
}
```

? Lógica en Pseudocódigo

```
IF usuario está vacío:  
  return error  
  
filtros = { usuario: usuario }  
  
data = ERP.dbErp(  
  table = "Validacion de Pagos Gerencia",  
  select = "concepto",  
  where usuario = usuario  
)  
  
conceptos = extraer columna "concepto"  
conceptos = eliminar duplicados  
  
return { valor: true, data: conceptos }
```

Revisión #2

Creado 2025-11-26 15:26:45 -05 por Elian

Actualizado 2025-11-26 15:30:12 -05 por Elian