

## DESCRIPCION DEL COMPONENTE prueba.intempo

**prueba-intempo** es un proyecto Maven desarrollado en Spring-Boot, sobre el IDE IntelliJ IDEA, con conexión a una base de datos MySQL, con persistencia en BD con JPA, cuenta con conectores a servicios SOAP y REST, se implementó la librería de Swagger-UI que permite analizar todo el proyecto y auto documentar todos los servicios, todos los servicios son de arquitectura REST.

Cuenta con los siguientes servicios:

Context-path: /prueba/intempo

Ping: /ping

Verifica el estado del componente

Body: null

Resultado exitoso: "It's alive Prueba Intempo"

Validación de Usuarios: /validateUser

Para la validación de usuarios se creó una estructura de base de datos donde se registran los usuarios de la plataforma, este servicio valida la existencia del usuario.

Body: { "user": "master", "password": "1234" }

Resultado exitoso: { "id": "1", "name": "Master User", "estado": "exitoso" }

Resultado fallido: { "id": "0", "estado": "Fallida" }

Lista de Convenios: /listConvenios

La lista de convenios se cargaron en una estructura de base de datos, este servicio retorna la lista de convenios.

Body: null

Resultado exitoso:

```
[
  { "id": 1, "name": "Agua", "compensar": "S" },
  { "id": 2, "name": "Gas", "compensar": "S" },
  { "id": 3, "name": "Luz", "compensar": "N" },
  { "id": 4, "name": "Telefonía", "compensar": "N" }
]
```

Lista de formas de pago: /listFormasPago

La lista de formas de pago se cargaron en una estructura de base de datos, este servicio retorna la lista de formas de pago.

Body: null

Resultado exitoso:

```
[
  { "id": 1, "description": "Efectivo" },
  { "id": 2, "description": "Tarjeta Debito" },
  { "id": 3, "description": "Tarjeta Credito" }
]
```

#### Consulta de referencia: /referenciaFactura

Para la consulta de referencia de facturas, encontramos tres situaciones que tienen que ver con los convenios; para el convenio de “Gas”, se implementó el consumo de un servicio SOAP, para el cual en la documentación venía el WSDL, el cual se monto en el componente y se recuperaron los métodos de fabrica, este servicio no fue posible consumirlo desde el SoapUI, ni por su puesto desde el componente, por tanto todas las peticiones salen fallidas.

Para el convenio de “Agua”, se implementó la conexión a un servicio REST, el cual recupera la información de la factura del servicio expuesto y la retorna.

Para los otros convenios al no tener un servicio de consulta, se simulo la respuesta y se devuelve como “Exitosa”.

Es de aclarar que se valida la integridad de la información del convenio. Por tanto, si en la petición viene un convenio que no exista, se rechaza la petición.

Body: { "convenio":"Gas", "referenciaFactura": "123456" }

Resultado exitoso:

```
{ "code": "0", "referenciaFactura": "123456", "totalPagar": 63686.67990612061,
  "message": "Exitoso" }
```

Resultado fallido:

```
{ "code": "99", "referenciaFactura": "0", "totalPagar": 0.0, "message": "Fallido" }
```

#### Pago de Factura: /pagoFactura

Al igual que en la consulta de referencia de factura, se tienen los tres escenarios de acuerdo con el convenio, para el convenio de “Gas”, servicio SOAP siempre sale “Fallido”, para convenio de “Agua”, servicio REST, se consume el servicio y se retorna la respuesta, para los otros convenios, se simula el consumo, y siempre retorna exitoso.

Este servicio también valida la existencia del convenio, y registra la información de la transacción en una estructura de la base de datos.

Body: { "convenio":"Agua", "referenciaFactura": "123456", "valorFactura":  
28829.137901375125 }

Resultado exitoso:

```
{ "code": "0", "referenciaFactura": "123456", "totalPagar": 28829.137901375125,
  "message": "Factura Pagada Exitosamente" }
```

Resultado fallido:

```
{ "code": "99", "referenciaFactura": "0", "totalPagar": 0.0, "message": "Fallido" }
```

#### Compensacion de Factura: /compensacionFactura

Al igual que en la consulta de referencia de factura, se tienen los tres escenarios de acuerdo con el convenio, para el convenio de "Gas", servicio SOAP siempre sale "Fallido", para convenio de "Agua", servicio REST, se consume el servicio y se retorna la respuesta, para los otros convenios, se simula el consumo, y siempre retorna exitoso.

Este servicio también valida la existencia del convenio, y registra la información de la transacción en una estructura de la base de datos.

Body: { "convenio": "Agua", "referenciaFactura": "123456", "valorFactura": 28829.137901375125 }

Resultado exitoso:

{ "code": "0", "referenciaFactura": "123456", "totalPagar": 28829.137901375125, "message": "Factura Compensada Exitosamente" }

Resultado fallido:

{ "code": "99", "referenciaFactura": "0", "totalPagar": 0.0, "message": "Fallido" }

## SWAGGER-UI: /swagger-ui.html

Se implementó la librería swagger-ui que permite analizar todo el proyecto y auto documentar todos los servicios.

The top screenshot displays the Swagger UI dashboard for 'Servicios Banco ABC'. It lists several API endpoints under the 'prueba-controller' group:

- POST /compensacionFactura compensacionFactura
- GET /ListConvenios listConvenios
- GET /ListFormasPago listFormasPago
- POST /pagoFactura pagoFactura
- GET /ping ping
- POST /referenciaFactura referenciaFactura
- POST /validateUser validateUser

The bottom screenshot provides a detailed view of the 'POST /compensacionFactura' endpoint. It shows the following details:

- Parameters:** A single parameter named 'request' of type 'request' (body) is required. The example value is a JSON object: 

```
{  "comenta": "string",  "referenciaFactura": "string",  "referenciaFactura": "string",  "referenciaFactura": "string"}
```

. The parameter content type is set to 'application/json'.
- Responses:** The response content type is set to 'JSON'. The following table lists the response codes and their descriptions:

Code	Description
200	OK
201	Created
401	Unauthorized
403	Forbidden

## DIAGRAMA DE COMPONENTES

