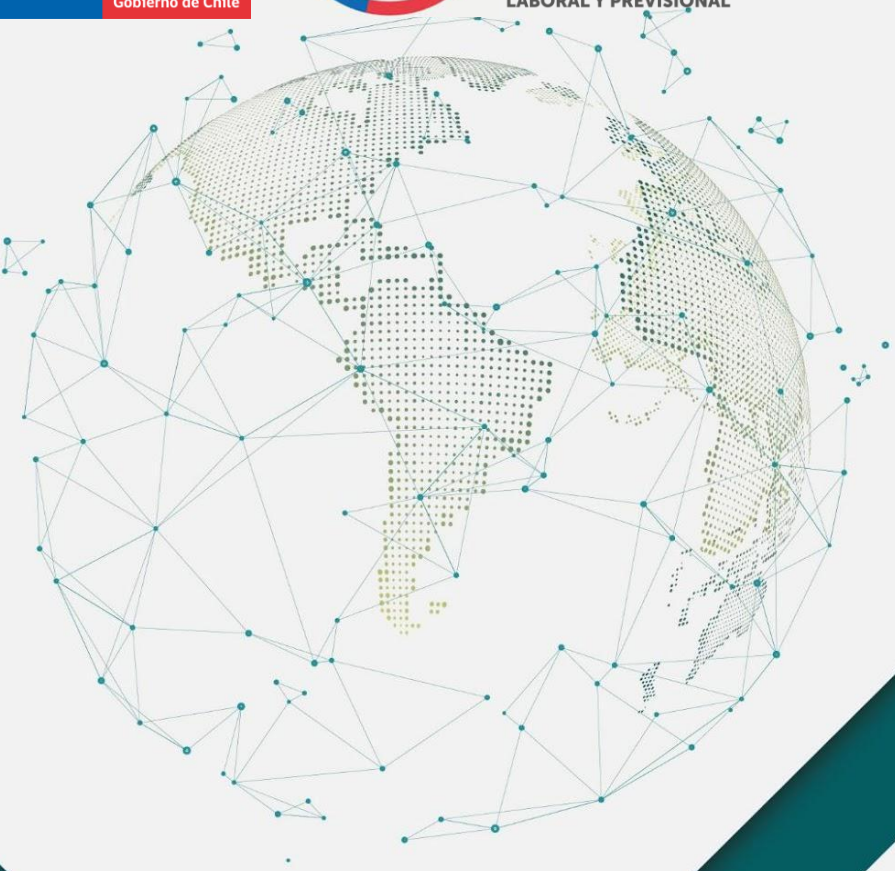
Subsecretaría de Previsión Social

Gobierno de Chile

Instituto de Previsión Social

NODO LABORAL Y PREVISIONAL



INSTRUCTIVO DE

CONSUMO NODO L&P

Coderhub

Versión	Fecha	Descripción de Cambios	Estado
1.0	06/10/2025	Primera versión de Instructivo	Borrador

CONTENIDO

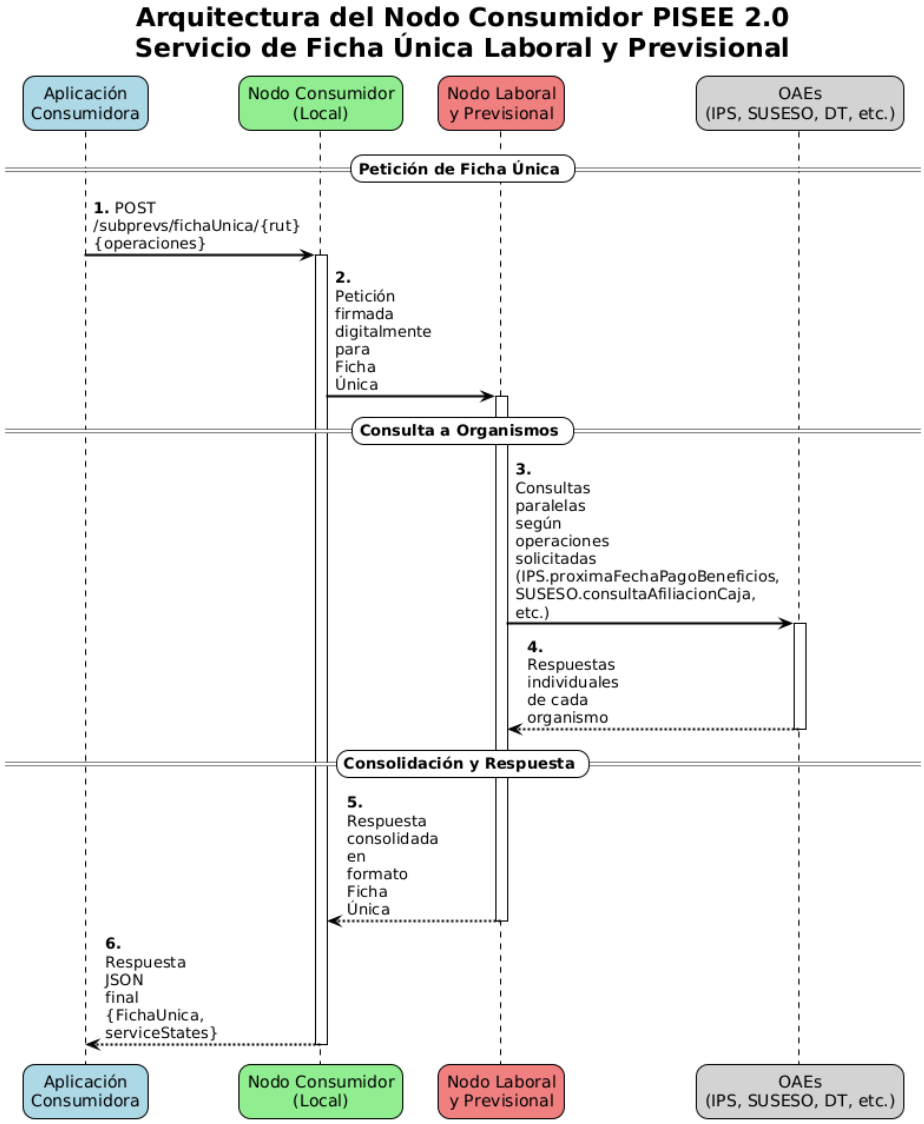
I. INTRODUCCIÓN

Este documento describe el proceso de instalación, configuración y consumo del servicio "Ficha Única" a través de un Nodo Consumidor que se conecta al Nodo Laboral y Previsional mediante la infraestructura PISEE. El Nodo Laboral y Previsional es un servicio especializado que consolida información de múltiples organismos del Estado del ámbito laboral y previsional (IPS, SUSESO, Dirección del Trabajo, etc.) en una única respuesta estructurada.

El objetivo de este manual es proporcionar una guía para los equipos de desarrollo que necesiten integrar este servicio en sus sistemas, permitiendo acceder de forma unificada a información laboral y previsional de los ciudadanos.

II. ARQUITECTURA GENERAL

Para consumir el servicio de Ficha Única, su aplicación interactúa con un Nodo Consumidor instalado localmente. Este nodo se encarga de gestionar la comunicación segura a través de la PISEE hacia el Nodo Laboral y Previsional, que es el proveedor especializado que consolida la información de los distintos organismos del Estado (OAE) del ámbito laboral y previsional.



El flujo de una petición es el siguiente:

1. La Aplicación Consumidora realiza una petición HTTP POST al Nodo Consumidor local, especificando el RUT y las operaciones deseadas (Fuentes de datos deseadas).
2. El Nodo Consumidor valida la petición, la firma digitalmente y la envía a través de la infraestructura PISEE. La comunicación es Punta a Punta y no utiliza a Gobierno Digital como intermediario.
3. El Nodo Laboral y Previsional recibe la petición y consulta a los organismos específicos (IPS, SUSESO, SP, etc.) según las operaciones solicitadas.
4. Cada OAE procesa la consulta y devuelve los datos al Nodo Laboral y Previsional.
5. El Nodo Laboral y Previsional consolida las respuestas de todos los organismos en un único documento de Ficha Única. La respuesta consolidada viaja de vuelta hacia el Nodo Consumidor.
6. Finalmente, el Nodo Consumidor entrega la respuesta JSON a la Aplicación Consumidora.

III. Requisitos Previos

Antes de proceder con la instalación, es indispensable cumplir con los siguientes requisitos:

1. **Habilitación del Organismo:** Su institución debe estar autorizada por la Secretaría de Gobierno Digital (SGD) y por la Subsecretaría de Previsión Social (SPS) para consumir el servicio *subprevs/fichaUnica*. Debe iniciar esta solicitud formalmente.
2. **Coordinación del Nodo:** Si su institución no cuenta con un Nodo de Interoperabilidad, debe coordinar su creación con Gobierno Digital. Se le solicitará una dirección IP o un DNS público desde donde se consumirá el servicio para configurar los permisos de acceso.
3. **Entorno de Servidor:** Se requiere un servidor con al menos 4Gb de RAM y 2 CPU. Las instrucciones de este manual están basadas en una distribución estándar de Linux. Asegúrese de tener los permisos necesarios para crear directorios y ejecutar aplicaciones.
4. **Software Requerido:** Debe tener el comando unzip instalado para descomprimir el paquete del Nodo.

Una vez completados estos pasos, la Subsecretaría de Previsión Social (SPS) le entregará un archivo **NodoFichaUnica.zip** preconfigurado con las credenciales de su institución.

IV. Instalación Nodo Consumidor OAE

4.1. Instalación del Nodo en el Servidor

Estos comandos deben ejecutarse en el servidor Linux donde operará el Nodo.

1. **Crear el Directorio** `/fichaunica` y **Descomprimir** el Paquete `NodoFichaUnica.zip` en dicha carpeta. ejecutando los siguientes comandos en el servidor:

```
mkdir -p /fichaunica
unzip NodoFichaUnica.zip -d /fichaunica
```

2. **Asignar Permisos de Ejecución:** Navegue a la carpeta donde descomprime el Nodo y asigne permisos de ejecución al aplicativo.

```
cd /fichaunica
sudo chmod +x ./NodoV2
```

3. **Iniciar el Nodo:** Ejecute el siguiente comando para iniciar.

```
./NodoV2 start
```

4.2.. CONFIGURACIÓN DEL ARCHIVO “CONFIG.JSON”

El archivo *config.json*, ubicado en el directorio de instalación (*/fichaunica*), contiene la configuración principal del Nodo. Para consumir el servicio de Ficha Única, debe añadir la siguiente configuración en la sección *consumidor*.

```
"consumidor":[
 {
       "nombre": "fichaUnica",
       "rutaLocal": "/subprevs/fichaUnica/{rut:.*}",
       "timeout": 10
 }
]
```

nombre: Identificador interno del servicio. Debe coincidir con el nombre del servicio que se está consumiendo.

rutaLocal: Define la ruta local a través de la cual su aplicación accederá al servicio en el Nodo. La expresión {rut:.*} captura el parámetro RUT de la URL. Esta ruta está configurada por Gobierno Digital en el catálogo de PISEE.

timeout: Tiempo máximo de espera (en segundos) antes de abortar la petición.

Una vez que modifique este archivo, es necesario reiniciar el Nodo para que los cambios surtan efecto.

```
cd /fichaunica
./NodoV2 stop
./NodoV2 start
```

V. ADMINISTRACIÓN DEL NODO

El Nodo se administra completamente a través de la línea de comandos. Para ejecutar los siguientes comandos, asegúrese de estar posicionado en el directorio de instalación del Nodo (ej./fichaunica).

```
cd /fichaunica
```

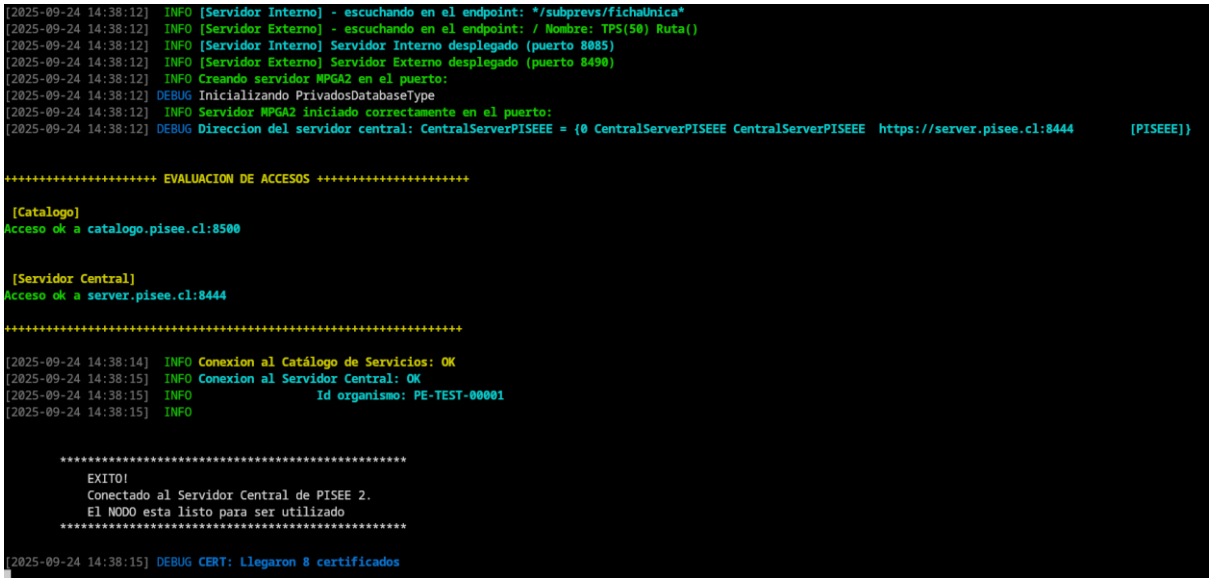
5.1. INICIAR EL NODO

Este comando inicia el proceso del Nodo en segundo plano.

```
./NodoV2 start
```

Al ejecutarse, verá los logs de arranque en la consola. Una inicialización exitosa mostrará mensajes que indican que el servidor interno se ha iniciado y está escuchando en el puerto configurado (por defecto, 8084).

En caso de que la inicialización sea exitosa, los registros deberían mostrarse de la siguiente forma:



5.2. Detener el Nodo

Este comando detiene el proceso del Nodo.

```
./NodoV2 stop
```

El comando no produce ninguna salida en la consola si se ejecuta correctamente.

5.3. Verificar el Estado

Permite obtener un resumen del estado de ejecución del Nodo.

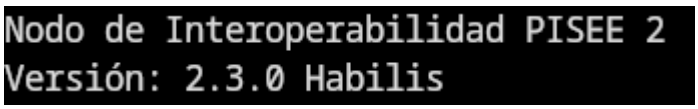
```
./NodoV2 estado
```

Muestra un resumen con los indicadores de estado de ejecución del Nodo. Este comando permite verificar si el Nodo está activo y si tiene conexión con sus dependencias, como la base de datos local, el catálogo de servicios y el servidor central de PISEE. También informa sobre los puertos de comunicación interno y externo.

5.4. Revisar la Versión

```
./NodoV2 version
```

Despliega la versión específica del software del Nodo que se está ejecutando, por ejemplo, "Versión: 2.0.0 Habilis".



5.5. Gestión de Logs

El sistema de registro de actividad (logging) del Nodo permite monitorear su funcionamiento y diagnosticar problemas. Toda la configuración se centraliza en el objeto `log` dentro del archivo principal **config.json**.
1. Nivel de Registro

El parámetro level controla el detalle de la información que se registrará. Los valores posibles son:

- "ERROR": Captura exclusivamente fallos críticos.
- "INFO": Opción balanceada que incluye eventos importantes, advertencias y errores. Se recomienda su uso en entornos de producción.
- "DEBUG": Modo detallado que registra cada paso del proceso. Se recomienda para desarrollo y depuración.

2. Destino de los Registros

El parámetro output especifica dónde se almacenarán los logs generados. Para que salga por la consola, el valor output debe ser stdout.

Para imprimir los logs directamente en la terminal o la salida estándar del sistema, configure el archivo de la siguiente manera:

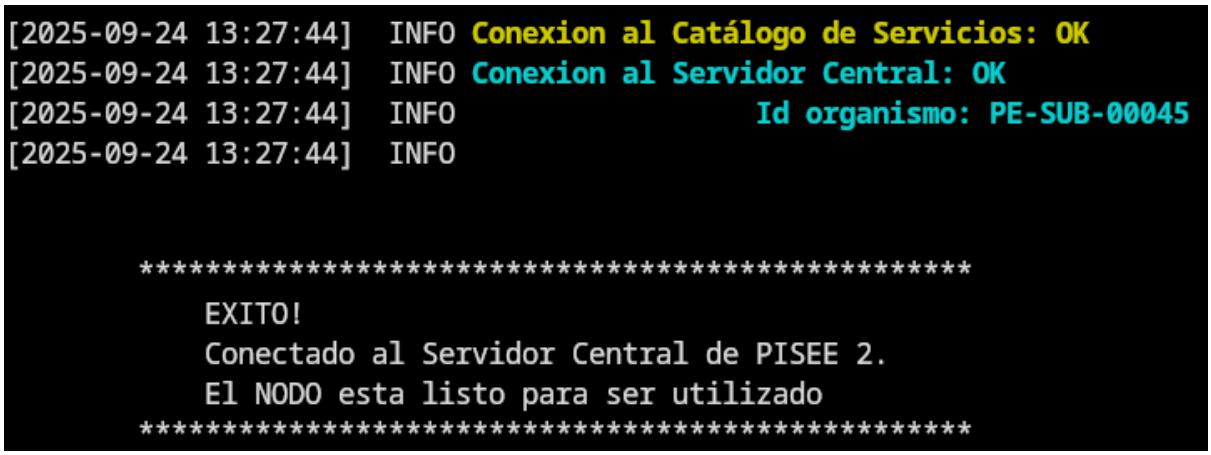
```
"log": {
  "level": "INFO",
  "output": "stdout"
}
```

Para almacenar los logs de forma persistente en un archivo, especifique la ruta absoluta en el parámetro `output`. Por ejemplo, para guardar los registros en `/var/log/nodo-lyp.log`:

```
"log": {
  "level": "INFO",
  "output": "/var/log/nodo-lyp.log"
```

}

Una vez que el Nodo esté en funcionamiento, puede verificar la correcta conexión con el servicio en revisando el log previamente configurado:



VI. USO DEL SERVICIO: PETICIÓN DE FICHA ÚNICA

Para consultar la Ficha Única, su aplicación debe realizar una petición HTTP POST al Nodo de Interoperabilidad local. Este método permite especificar dinámicamente qué fuentes de datos (operaciones) desea consultar para un RUT específico.

6.1. ESTRUCTURA DE LA PETICIÓN POST

Endpoint:

- **Método:** POST
- **URL:** http://<IP_NODO>:<PUERTO_INTERNO>/subprevs/fichaUnica/{rut}

Componentes de la URL:

- **<IP_NODO>:** La dirección IP del servidor donde instaló el Nodo. Si la petición se realiza desde la misma máquina, puede usar localhost.
- **<PUERTO_INTERNO>:**El puerto del servidor interno del Nodo, configurado en config.json (por defecto: 8084).
- **{rut}:** El RUT del ciudadano a consultar. El rut se debe agregar sin puntos, con guión y con dígito verificador.

Ejemplo de URL completa para localhost:

- http://localhost:8084/subprevs/fichaUnica/12345678-9

Cabeceras (Headers):

La petición debe incluir la siguiente cabecera para indicar que el cuerpo del mensaje está en formato JSON:

- Content-Type: application/json

Cuerpo de la Petición (Body):

El cuerpo debe ser un objeto JSON con la siguiente estructura:

```
{
  "operaciones": [
      "CODIGO_SERVICIO_1",
      "CODIGO_SERVICIO_2"
  ],
  "procedimiento": "identificador_procedimiento",
  "cpat": "codigo_cpat"
}
```

operaciones: Un arreglo (lista) de strings que contiene los nombres de los servicios específicos que se desean consultar en paralelo. Se pueden incluir uno o más servicios de la siguiente lista:

- **SUSESO.consultaAfiliacionCaja**

- **SUSESO.consultaAfiliacionMutual**
- **SUSESO.SIAGFConsultaCausanteSimple**
- **IPS.proximaFechaPagoBeneficios**
- **IPS.obtenerresolucionAF**
- **DT.finiquitoElectronico**
- **DT.libroRemuneraciones1**
- **DT.libroRemuneraciones2**
- **DT.registroContratoDeTrabajo**
- **SPENSIONES.ConsultaCotizaciones**

6.2. Ejemplo de Petición con cURL

A continuación se muestran ejemplos de cómo consumir el servicio utilizando cURL desde la terminal. Puedes solicitar uno o más servicios en la lista "operaciones".

Para efectos de la demostración se usarán los siguientes parámetros de la url y se consultarán directamente desde el servidor donde está instalado el nodo. Además, se envía adjunto los JSON asociados a la respuesta de cada ejemplo.

6.2.1. Ejemplo 1: Consultar un solo servicio

Este cURL consulta la próxima fecha de pago de beneficios en la OAE IPS:

```
curl -X POST "http://localhost:8084/subprevs/fichaUnica/1-9" \
  -H "Content-Type: application/json" \
  -d '{
       "operaciones": [
             "IPS.proximaFechaPagoBeneficios"
       ],
       "procedimiento": "foo",
       "cpat": "ABC123"
  }'
```

6.2.2. Ejemplo 2: Consultar múltiples servicios de una misma institución

Este comando consulta tres servicios de la OAE SUSESO para un RUT:

```
curl -X POST "http://localhost:8084/subprevs/fichaUnica/1-9" \
  -H "Content-Type: application/json" \
  -d '{
       "operaciones": [
          "SUSESO.consultaAfiliacionCaja",
          "SUSESO.consultaAfiliacionMutual",
```

```
            "SUSESO.SIAGFConsultaCausanteSimple"
        ],
        "procedimiento": "foo",
        "cpat": "bar"
    }'
```

6.2.3. Ejemplo 3: Consultar múltiples servicios de distintas instituciones

Este comando consulta servicios de las OAES IPS y SUSESO simultáneamente para un RUT:

```
curl -X POST "http://localhost:8084/subprevs/fichaUnica/1-9" \
  -H "Content-Type: application/json" \
  -d '{
        "operaciones": [
            "IPS.proximaFechaPagoBeneficios",
            "SUSESO.consultaAfiliacionCaja",
            "SUSESO.SIAGFConsultaCausanteSimple"
        ],
        "procedimiento": "foo",
        "cpat": "bar"
    }'
```

6.3. Ejemplo de petición con Python

El siguiente script muestra cómo realizar la petición usando la librería requests en Python. El script es autocontenido y puede ser ejecutado directamente.

```
import requests
import json

def obtener_ficha_unica(rut, operaciones, procedimiento="online", cpat="ABC123"):
	"""
	Realiza una petición POST al Nodo para obtener datos específicos
	de la Ficha Única de una persona.
	"""
	# URL del servicio en el Nodo local
	url = f"http://localhost:8084/subprevs/fichaUnica/{rut}"

	# Datos que se enviarán en el cuerpo de la petición
	payload = {
	   "operaciones": operaciones,
	   "procedimiento": procedimiento,
	   "cpat": cpat
	}

	# Cabeceras de la petición
	headers = {
	   "Content-Type": "application/json"
	}

	try:
	# Realizar la petición POST
	respuesta = requests.post(url, data=json.dumps(payload), headers=headers)

	# Verificar si la petición fue exitosa (código de estado 200)
	if respuesta.status_code == 200:
		return respuesta.json()
	else:
		return {
		"error": True,
```

```
            "status_code": respuesta.status_code,
            "mensaje": respuesta.text
            }
    except requests.exceptions.RequestException as e:
    return {
            "error": True,
            "mensaje": f"Error de conexión: {e}"
    }

# Ejemplo de uso
if __name__ == "__main__":
    rut_a_consultar = "1-9"  # Reemplazar con un RUT válido
    servicios_a_consultar = [
       "IPS.proximaFechaPagoBeneficios",
       "SUSESO.consultaAfiliacionCaja"
    ]

    datos_ficha = obtener_ficha_unica(rut_a_consultar, servicios_a_consultar)

    if datos_ficha and not datos_ficha.get("error"):
        print("Datos de la Ficha Única:")
        print(json.dumps(datos_ficha, indent=2, ensure_ascii=False))
    else:
        print("No se pudieron obtener los datos de la Ficha Única.")
        print(datos_ficha)
```

VII. MANEJO DE LA RESPUESTA

La respuesta a una petición es un objeto JSON que consolida la información de las fuentes consultadas.

7.1 ESTRUCTURA DE UNA RESPUESTA

Una respuesta exitosa se compone de tres partes principales:

1. **Response Header:** Contiene el estado general de la petición.
2. **Response Body (FichaUnica):** Contiene los datos retornados por cada servicio solicitado.
3. **Response Body (serviceStates):** Detalla el estado individual de cada uno de los servicios consultados.

Response Header

```
Response Header: {
  "Status": {
    "code": 200,
    "message": "OK",
  },
  "Content-Type": application/json
}
```

Response Body (FichaUnica)

```
{
  "FichaUnica":{
    "IPS": {
      "obtenerResolucionAF": {...},
      "proximaFechaPagoBeneficios": {...}
    },
    "SP": {
      "cotizacionesPrevisionales": {...}
    },
    "SUSESO":{
```

```
      "ConsultaAfiliacionCaja":{...},
      "ConsultaAfiliacionMutual":{...}
    },
    "DT": {
      "registroTerminoContratoTrabajo": {... }
    },
  },
```


Response Body (serviceStates)


```
  "serviceStates": {
    "IPS":{
      "obtenerResolucionAF":{
        "code": 200,
        "message": "OK - Solicitud exitosa"
      },
      "proximaFechaPagoBeneficios":{
        "code": 200,
        "message": "OK - Solicitud exitosa"
      }
    },
    "SP":{
      "ultimasCotizacionesPrevisionales":{...}
    },
    "SUSESO":{
```

```
      "ConsultaAfiliacionCaja":{...},
      "ConsultaAfiliacionMutual":{...}
    },
    "DT":{
      "registroContratoTrabajo":{...}
    }
  }
}
```


En caso de que una petición no pueda ser procesada correctamente, la respuesta incluirá los códigos de error correspondientes.

7.2 Códigos de Estado y Errores

Interpretar correctamente los códigos de estado es importante para manejar los diferentes escenarios.

7.2.1. Códigos del Response Header

Estos códigos, basados en los estándares HTTP, resumen el resultado general de la petición.

Status Code	Significado	Acción Recomendada
200 OK	Todas las fuentes de datos consultadas respondieron exitosamente.	Procesar la información contenida en FichaUnica.
207 Multi-Status	Algunas fuentes respondieron correctamente, pero otras fallaron.	**Revisar el objeto** *serviceStates* para determinar qué servicios fallaron y por qué. La información de los servicios exitosos estará disponible en *FichaUnica*.
400 Bad Request	La petición está mal formada. Puede ser un JSON inválido o datos incorrectos.	Verificar la sintaxis del JSON de la petición y el formato de los datos enviados (especialmente el RUT).
401 Unauthorized	No está autorizado para consumir el servicio.	Contactar a SPS para verificar que su institución y la IP del servidor estén correctamente habilitadas.
429 Too Many Requests	Se ha excedido la tasa de peticiones permitida.	Reducir la frecuencia de las llamadas al servicio.
503 Service Unavailable	El servicio de Ficha Única no está disponible en este momento.	Reintentar la petición más tarde.

7.2.2. Códigos de serviceStates

Este objeto provee un estado granular para cada servicio solicitado, permitiendo un manejo de errores más preciso.

Code	Significado	Causa Común
200 OK	El servicio específico respondió correctamente.	-
400 Bad Request	Datos inválidos para ese servicio en particular.	El RUT puede no ser válido para el sistema de la institución proveedora.

401 Unauthorized	No autorizado para acceder a ese recurso específico.	La configuración de permisos para ese servicio puede ser incorrecta.
403 Forbidden	Acceso prohibido al recurso.	Similar a 401, pero indica un rechazo definitivo por reglas de negocio.
404 Not Found	El recurso o la persona no fue encontrada en la fuente de datos.	El RUT no tiene registros para la consulta realizada en esa institución.
500 Internal Server Error	Error interno en el servidor de la institución proveedora.	Un problema en el sistema de origen impidió procesar la solicitud.
503 Service Unavailable	El servicio de la institución proveedora no está disponible.	El sistema de origen está en mantenimiento o fuera de línea.
504 Gateway Timeout	El Nodo no recibió una respuesta a tiempo desde la institución proveedora.	El sistema de origen está sobrecargado o experimentando lentitud.

VIII. TROUBLESHOOTING Y PREGUNTAS FRECUENTES

Esta sección aborda algunos de los problemas más comunes que pueden surgir durante la instalación y el uso del Nodo.

Vale la pena hacer referencia a los manuales de PISEE para información más detallada:

- https://digital.gob.cl/transformacion-digital/estandares-y-guias/guia-tecnica-de-interoperabilidad-pisee-2/
- https://digital.gob.cl/transformacion-digital/estandares-y-guias/

P1: AL REALIZAR UNA PETICIÓN, RECIBO UN ERROR DE "CONNECTION TIMED OUT" O "CONNECTION REFUSED" Y/O NO PUEDO CONECTARME AL SERVIDOR CENTRAL O AL CATÁLOGO DE PISEE.

- **Causa Posible 1: El Nodo no está en ejecución.**
  - **Solución:** Conéctese al servidor y ejecute cd /fichaunica && ./NodoV2 estado. Si no está activo, inícielo con ./NodoV2 start.
- **Causa Posible 2: Firewall.**
  - **Solución:** Asegúrese de que el puerto del servidor interno del Nodo (por defecto 8084) esté abierto en el firewall del servidor y que sea accesible desde la máquina que origina la petición.
- **Causa Posible 3: Dirección IP o puerto incorrectos.**
  - **Solución:** Verifique que la URL de la petición (http://<IP_NODO>:<PUERTO_INTERNO>) sea la correcta.
- **Causa Posible 4: Problema con los certificados.**
    - **Solución:** Solicitar nuevamente los certificados del Nodo a instalar a SGD.

P2: Recibo un error 400 Bad Request.

- **Causa Posible 1: Formato de RUT incorrecto.**
  - **Solución:** Verifique que el RUT en la URL cumpla estrictamente con el formato 12345678-9 (sin puntos, con guión).
- **Causa Posible 2: JSON mal formado.**
  - **Solución:** Valide la sintaxis del cuerpo JSON de su petición. Un error común es una coma extra al final de una lista.

P3: Recibo un error 401 Unauthorized.

- **Causa Posible:** La IP de su servidor no está autorizada.
  - **Solución:** Este error casi siempre significa que la IP pública del servidor desde donde se conecta el Nodo no está en la lista de IPs permitidas en el catálogo de PISEE. Contacte a Gobierno Digital o SPS para verificar y, si es necesario, registrar la IP correcta.

P4: El Nodo no se inicia o se detiene inmediatamente.

- **Causa Posible: Error en config.json.**
  - **Solución:** Revise el archivo *config.json* en busca de errores de sintaxis. Un JSON inválido impedirá que el Nodo se inicie. Revise los logs del Nodo (configurados en la misma *config.json*) para obtener pistas sobre el error.

P5: ¿Puedo ejecutar el Nodo en un contenedor como Docker?

Sí. De hecho el nodo Proveedor se encuentra en un contenedor Docker. El Nodo es un binario autocontenido, por lo que es compatible con la contenerización. Deberá asegurarse de:

1. Copiar el binario y la configuración al contenedor.
2. Exponer el puerto interno del Nodo (ej. 8084) al exterior del contenedor.
3. Asegurarse de que el contenedor tenga conectividad de red para alcanzar los servicios de PISEE.