

Backlog de Historias de Usuario - Prueba Técnica Parameta

Este documento define las historias de usuario necesarias para la implementación de la solución de vinculación de empleados, abarcando infraestructura, lógica de negocio y comunicación legacy.

HU-01: Configuración de Infraestructura Serverless (AWS SAM)

Título: Despliegue de Infraestructura como Código (IaC) usando AWS SAM **Prioridad:** Alta

Estimación: 3 Puntos

Descripción

Como **DevOps / Desarrollador Senior**, Quiero definir la infraestructura del proyecto utilizando un template de AWS SAM, Para poder desplegar de manera automatizada y reproducible el servicio REST y el Mock SOAP en un entorno AWS.

Criterios de Aceptación

- Definición del Template:** El archivo `template.yaml` debe existir y ser válido según la especificación de AWS SAM.
- Recurso API Gateway:** Se debe definir un API Gateway que exponga el endpoint `/api/vinculacion` mediante el método GET .
- Función Lambda (Core):** Debe definirse la función Lambda para el servicio Spring Boot con al menos 512MB de memoria y tiempo de espera de 30s.
- Función Lambda (Mock):** Debe definirse una segunda función Lambda ligera (Node.js o Python) que simule el servicio SOAP Legacy.
- Variables de Entorno:** La URL del Mock SOAP debe injectarse dinámicamente en la Lambda del servicio Core a través de variables de entorno.
- Despliegue Local:** El comando `sam local start-api` debe levantar ambos servicios correctamente en la máquina local.

HU-02: Lógica de Negocio y Testing Unitario

Título: Implementación de Reglas de Vinculación y Pruebas Unitarias **Prioridad:** Alta

Estimación: 5 Puntos

Descripción

Como Desarrollador Backend, Quiero implementar la lógica de validación de empleados y cálculo de fechas en el servicio Spring Boot, Para asegurar que solo se vinculen empleados mayores de edad y se calculen correctamente sus tiempos de vinculación.

Criterios de Aceptación

1. **Cálculo de Fechas:** El servicio debe calcular con precisión:
 - La edad actual del empleado basada en su fecha de nacimiento.
 - El tiempo de vinculación (años, meses, días) basado en la fecha de ingreso a la compañía.
2. **Validación de Mayoría de Edad:** El sistema debe rechazar (lanzar excepción controlada) cualquier solicitud donde la edad calculada sea menor a 18 años.
3. **Formato de Respuesta:** El DTO de respuesta debe incluir la estructura JSON requerida con: tiempo de vinculación y edad actual.
4. **Validación de Inputs:** Los campos obligatorios (Nombres, Apellidos, Tipo Documento, Número Documento, Fecha Nacimiento, Fecha Vinculación, Cargo, Salario) no deben ser nulos ni vacíos.
5. **Testing (Cobertura):** Se deben incluir pruebas unitarias usando **JUnit 5 y Mockito** con una cobertura de código superior al 80%.
6. **Casos de Prueba Mínimos:**
 - *Happy Path:* Empleado > 18 años con fechas válidas.
 - *Edge Case:* Empleado con exactamente 18 años hoy.
 - *Error Case:* Empleado menor de edad.
 - *Error Case:* Fecha de vinculación futura (imposible).

HU-03: Integración Cliente SOAP (Legacy Adapter)

Título: Desarrollo de Cliente SOAP y Conexión con Mock **Prioridad:** Media **Estimación:** 5 Puntos

Descripción

Como Desarrollador Backend, Quiero configurar un cliente SOAP dentro de la aplicación Spring Boot, Para enviar la información del empleado validado al sistema legacy simulado (Mock).

Criterios de Aceptación

1. **Estructura XML:** La aplicación debe ser capaz de serializar el objeto `Empleado` en un sobre SOAP XML válido.
2. **Configuración del Cliente:** Implementar un `WebServiceTemplate` (o cliente similar) configurado para apuntar a la URL del Mock definida en las variables de entorno.
3. **Mapeo de Respuesta:** El sistema debe ser capaz de deserializar la respuesta del Mock (XML) y confirmar si la operación fue exitosa.
4. **Manejo de Errores de Conexión:** Si el Mock no responde o da timeout, el servicio REST debe capturar la excepción y responder un mensaje de error amigable al usuario (HTTP 503 o 500 según corresponda).