

Diseño de Arquitectura de Microservicios para Sistema de Planillas

Mynor Rolando Masaya Juárez 201213423

1. Introducción

El presente documento describe el diseño de una arquitectura basada en microservicios para optimizar el sistema de planillas de la empresa, que actualmente opera bajo un esquema monolítico y presenta problemas de rendimiento en temporadas de alta demanda. La nueva solución busca mejorar la escalabilidad y eficiencia del proceso.

2. Arquitectura del Sistema

El sistema se dividirá en los siguientes microservicios:

1. Servicio de Autenticación:

- Integración con OAuth de la empresa (tokens de 12 horas).
- Autenticación complementaria con manejo de permisos.

2. Servicio de Carga de Planillas:

- Recepción y procesamiento de archivos CSV.
- Validaciones internas de reglajes.

3. Servicio de Aprobación de Planillas:

- Flujo de aprobación en tres pasos por diferentes usuarios.

4. Servicio de Almacenamiento:

- Gestión del almacenamiento de los archivos CSV en la nube o un servidor FTP.

5. Servicio de Historial:

- Consulta de planillas procesadas y contenido almacenado.
- Posibilidad de descarga de las planillas.

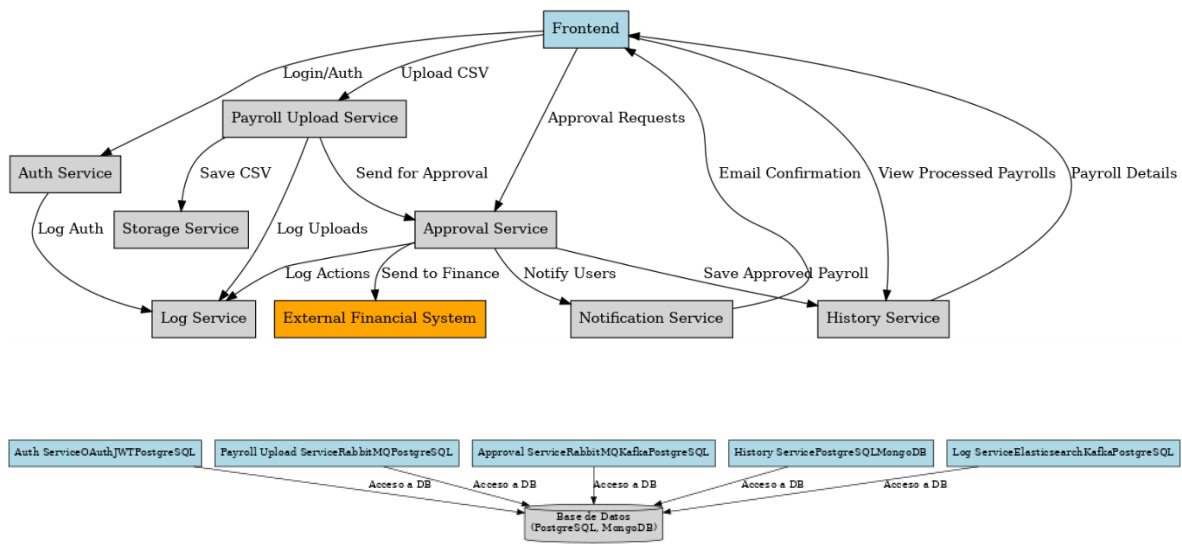
6. Servicio de Notificaciones:

- Envío de correos a empleados cuando la planilla es aprobada.

7. Servicio de Logs:

- Registro de eventos en el sistema para auditoría y control.

2.1 Diagrama de Arquitectura



3. Descripción de la Solución

El nuevo sistema se diseña para soportar grandes volúmenes de transacciones sin comprometer el rendimiento. Se basa en la modularidad de microservicios, permitiendo escalabilidad independiente de cada componente. La comunicación entre microservicios se realizará a través de APIs REST.

3.1 Tecnologías Sugeridas

- **Backend:** Node.js (NestJS) o Python (FastAPI/Django).
- **Base de Datos:** PostgreSQL o MongoDB.
- **Autenticación:** OAuth + JWT.
- **Almacenamiento de Planillas:** Amazon S3, Google Cloud Storage o FTP.
- **Notificaciones:** AWS SES, SendGrid o SMTP.
- **Gestión de Logs:** ELK Stack (Elasticsearch, Logstash, Kibana) o base de datos específica.

4. Diseño de Microservicios

Cada microservicio estará diseñado para realizar tareas específicas y se comunicará a través de APIs REST. A continuación, se describen las funcionalidades de cada uno:

4.1 Servicio de Autenticación

- Autenticación mediante OAuth de la empresa.

- Administración de permisos internos mediante JWT.

4.2 Servicio de Carga de Planillas

- Recepción de archivos CSV.
- Validación de datos.

4.3 Servicio de Aprobación

- Proceso de aprobación de tres pasos.
- Control de estado de cada planilla.

4.4 Servicio de Almacenamiento

- Gestión de archivos en la nube o FTP.
- Seguridad en el acceso a los archivos.

4.5 Servicio de Historial

- Consulta de registros de planillas procesadas.
- Acceso y descarga de planillas.

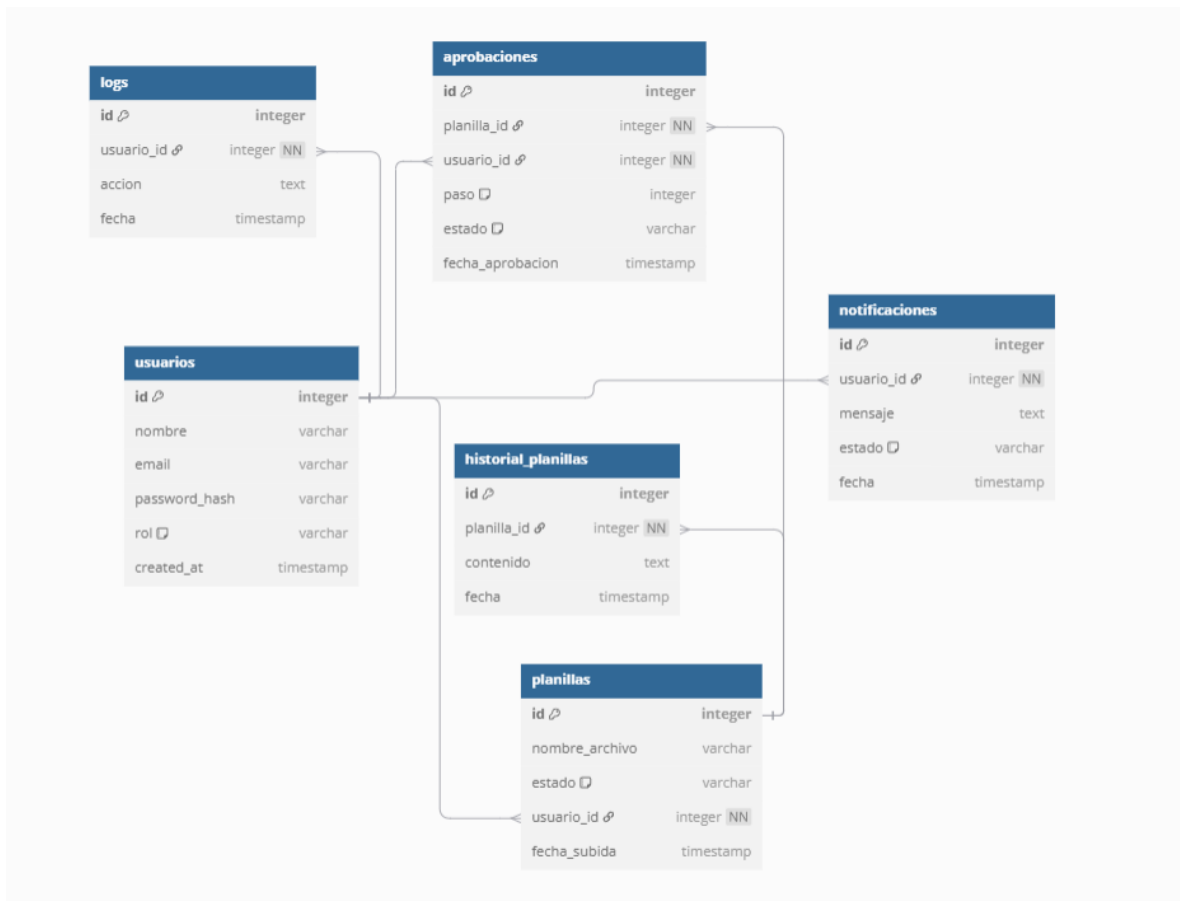
4.6 Servicio de Notificaciones

- Envío de correos electrónicos automáticos al aprobar una planilla.

4.7 Servicio de Logs

- Registro de eventos del sistema.
- Herramientas para consulta y auditoría.

5. Diagrama ER de la Base de Datos



6. Conclusión

El diseño propuesto permite una migración eficiente del sistema monolítico a una arquitectura de microservicios, mejorando el rendimiento, escalabilidad y mantenibilidad del sistema de planillas.