

Desarrollo de microservicios y su integración mediante herramientas colaborativas y APIs.

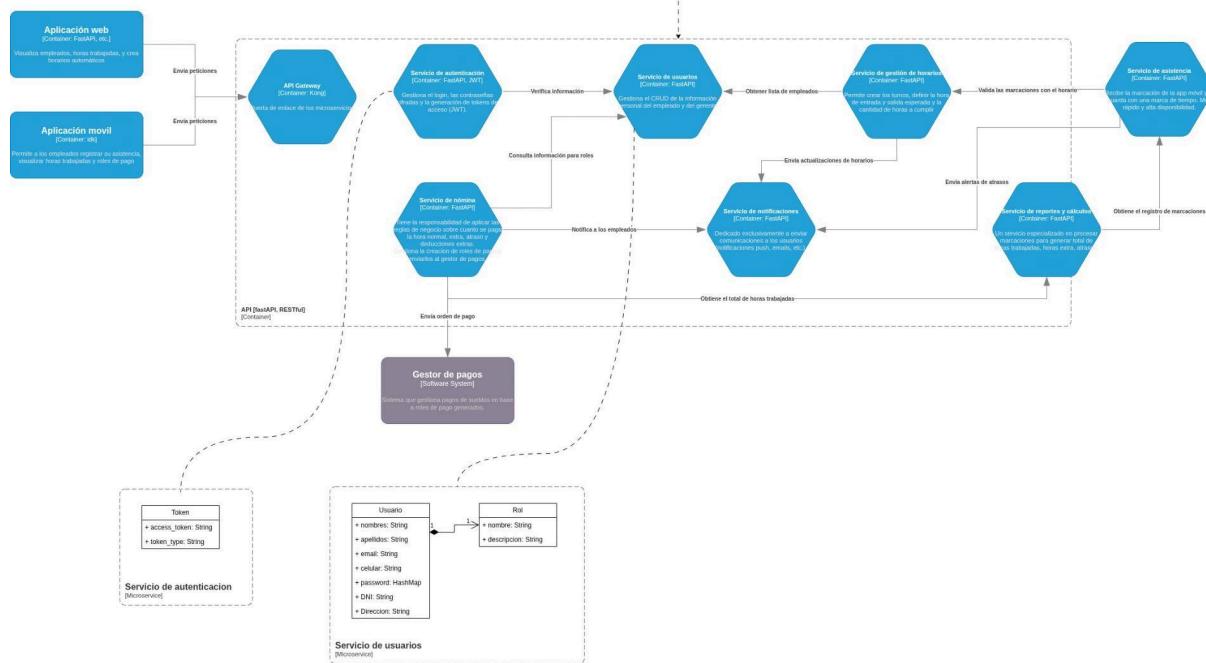
Integrantes: Alexis Vinicio Roman Avila, Josue Rodolfo Torres Guiracocha

Fecha de entrega: 26 de octubre de 2025

Desarrollo de microservicios y su integración mediante herramientas colaborativas y APIs.

Resumen del microservicio creado

Nombre del microservicio	Propósito	Relación con el sistema
Servicio de autenticación	Verificar la identidad del usuario (login) y en un futuro se planea gestionar los tokens de refresco para el logout y refresh.	Emite los tokens que los demás servicios verifican para conceder acceso.
Servicio de usuarios	Gestionar la información personal, credenciales y roles.	Le dice a Autenticación quién es quién y qué rol tiene.
Servicio de gestión de horarios	Crear turnos, asignar horas de entrada/salida esperadas y gestionar las reglas de los horarios.	Es consultado por Asistencia, para saber si alguien llega tarde y por Reportes.
Servicio de asistencia	Recibir y almacenar las marcaciones de los empleados desde la app móvil.	Provee los datos de marcaciones que alimentan al servicio de Reportes y Cálculos.
Servicio de nómina	Calcular y generar el rol de pago, aplicando reglas de negocio (pago por hora, extras) y enviarlo al gestor de pagos externo.	Gestiona el pago final consultando a Reportes por el total de horas y a Usuarios para los datos del empleado.
Servicio de notificaciones	Enviar comunicaciones a los usuarios (email, notificaciones push) sobre eventos importantes.	El mensajero
Servicio de reportes y cálculos	Procesar datos de marcaciones para generar información agregada: total de horas trabajadas, atrasos, etc.	Provee los cálculos ya digeridos que Nómina necesita para poder pagar.



Integración mediante APIs

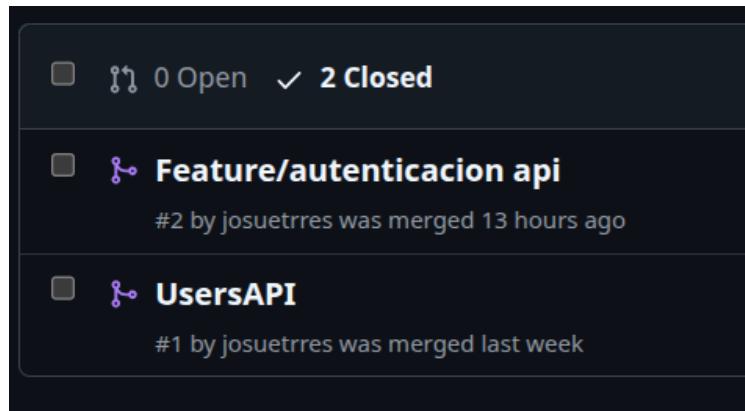
Los microservicios se comunicaran con el exterior mediante un API Gateway implementado con la herramienta Kong. La elección de esta herramienta se basa en la facilidad de uso y la automatización y abstracción de procesos de seguridad, etc.

```
└── kong.yaml
  1   _format_version: "3.0"
  2
  3   services:
  4     - name: auth-service
  5       url: http://servicio-autenticacion:8000
  6     - name: user-service
  7       url: http://servicio-usuarios:8000
  8
  9   routes:
 10     - name: auth-route
 11       service: auth-service
 12       paths:
 13         - /auth
 14       strip_path: false
 15
 16     - name: user-route
 17       service: user-service
 18       paths:
 19         - /users
 20       strip_path: false
 21
```

Código de configuración de Kong

Herramientas colaborativas utilizadas

Para la coordinación del equipo se utilizaron herramientas de comunicación como en nuestro caso whatsapp, debido a que somos un grupo de 2 personas, y un repositorio en github donde se comparte los avances que cada integrante tiene en el proyecto.



Pull request solicitados en el transcurso del proyecto

Buenas prácticas y aprendizajes

Las buenas prácticas que se implementaron en el transcurso del proyecto son respecto a los flujos de trabajo en Git, la implementación de ramas feature, permite el desarrollo independiente de cada parte del software y además del desarrollo por cada integrante del proyecto sin la necesidad de el cruce de commits.

Creemos que la modularidad permite un desarrollo más flexible , robusto y escalable a largo plazo, sin embargo se vuelve más tediosa la tarea de implementar cada módulo como un ente independiente.

Preguntas de reflexión.

¿Qué ventajas observas en la integración mediante APIs REST respecto a un monolito tradicional?

La principal ventaja radica en la modularización de las funcionalidades del sistema, ya que en un monolito normal estas trabajan en un solo módulo lo cual entorpece el desarrollo ágil , la experimentación o actualización puesto que se tendría que volver a desplegar nuevamente todo. Las Apis Rest, en cambio, permiten que los servicios funcionen independientemente y facilitan todas las acciones de desarrollo anteriormente mencionadas , además de dar paso a integración con otros servicios de manera segura.

¿Cómo aportan las herramientas colaborativas a la gestión técnica de los microservicios?

Las herramientas colaborativas permiten tener un flujo de trabajo más fluido con los compañeros, permiten la comunicación asertiva entre cada uno de los desarrolladores, lo que genera mejores resultados.

¿Qué riesgos existen al distribuir un sistema en varios microservicios y cómo pueden mitigarse?

Existen riesgos de sufrir latencias al comunicar los servicios, ya que al comunicarse externamente siempre será más lento que al comunicarse internamente. Se puede mitigar eligiendo las tecnologías óptimas y que se adapten a la velocidad que queremos brindar. Otro riesgo es el factor de ataque, ya que al ser varias aplicaciones independientes deben implementar mecanismos de seguridad en cada una de ellas. Se puede mitigar implementando una arquitectura que brinde garantías para posibles ataques.