Flujo de Autenticación y Uso de JWT - BabyTrackMaster

Este documento describe el flujo de autenticación y el uso de tokens JWT dentro de la arquitectura de microservicios de la aplicación BabyTrackMaster. Incluye el rol del servicio **api-usuarios**, la propagación del token a través del **api-gateway** y el **api-bff**, y la validación en cada microservicio de negocio. La información se basa en la documentación técnica, funcional y el plan del proyecto.

1. Resumen del flujo de autenticación

1. El usuario se autentica mediante el servicio **api-usuarios**, enviando sus credenciales a través del **api-gateway**. 2. Si las credenciales son correctas, **api-usuarios** genera un token JWT firmado y lo devuelve al frontend. 3. El frontend almacena el JWT (por ejemplo, en *localStorage*) y lo incluye en la cabecera *Authorization: Bearer <token>* en todas las peticiones posteriores. 4. El **api-gateway** enruta las peticiones y propaga el JWT a los microservicios correspondientes. 5. El **api-bff** valida el JWT, extrae datos como *userId* y roles, y reenvía el token a los microservicios internos que consulta para el Dashboard. 6. Cada microservicio de negocio valida el JWT con la misma clave secreta configurada (vía Config Server o variables de entorno).

2. Validación y seguridad en cada componente

• **api-usuarios**: genera y valida JWT, gestiona registro y login de usuarios. • **api-gateway**: enruta y puede realizar validación ligera del JWT para filtrar peticiones inválidas. • **api-bff**: valida el JWT y lo reenvía a los microservicios internos para conservar el contexto de usuario. • **Microservicios de negocio**: validan el JWT en cada petición y aplican reglas de autorización basadas en *userId* y roles. • **Servicios internos** (config-server, log-service, backup-sync-service): no requieren JWT si están en red interna segura, pero deben protegerse con otros mecanismos (ACLs, Basic Auth, mTLS).

3. Endpoints públicos y privados

• **Públicos**: /auth/register, /auth/login (y opcionalmente /actuator/health con acceso restringido). • **Privados**: todos los endpoints de CRUD y consulta de datos del bebé (cuidados, gastos, hitos, diario, citas, rutinas, perfil).

4. Beneficios del uso de JWT en esta arquitectura

• No es necesario mantener sesión en memoria en el servidor. • Escalable: cualquier instancia de un microservicio puede validar el JWT sin dependencia de un estado central. • Seguridad consistente en todos los microservicios. • Facilidad para incluir información relevante (roles, ids) en el token.

5. Checklist de implementación

- Generar JWT en api-usuarios con firma HS256, incluyendo sub (userId), roles y fecha de expiración.
- Configurar la misma clave secreta en todos los microservicios (vía Config Server o variables de entorno).
- Añadir filtros de Spring Security para extraer y validar el token en cada servicio de negocio.
- Configurar api-gateway para propagar la cabecera Authorization.
- En api-bff, reinyectar el JWT al llamar a otros microservicios (por ejemplo, con Feign Interceptor).
- Proteger Swagger y Actuator en producción.