

Informe de verificación — DeadApp (backend)

Fecha: 24-11-2025

Resumen rápido

- Estado de la API: Funcionando en `http://localhost:3002`.
- Tests unitarios/integración (Jest): PASADOS (1 suite, 1 test).
- Endpoints verificados:
 - `GET /health` → { "status": "OK" }
 - `GET /api/status` → { "status": "OK", "mongoState": 1 }
 - `GET /api/boveda/activos` → Lista (vacía o con activos). No devuelve contraseñas en texto plano.
 - `GET /api/boveda/activos/:id/secreto` → Devuelve el secreto descifrado para un activo específico (requiere rol `usuario_titular`).

Cambios y correcciones aplicadas

- `src/infrastructure/http/middleware/auth.middleware.ts`: Añadido fallback de desarrollo cuando no hay Keycloak configurado. Permite usar las cabeceras `x-dev-user` y `x-dev-roles` para simular autenticación en dev.
- `src/application/use-cases/GestionarBovedaUseCase.ts`: Ya NO se desencriptan las contraseñas en la lista de activos; se añadió `obtenerSecretoActivo(userId, activoId)` para recuperar el secreto de forma controlada.
- `src/infrastructure/http/controllers/BovedaController.ts` y `routes/boveda.routes.ts`: Añadido endpoint `GET /api/boveda/activos/:id/secreto` (solo titular) para revelar el secreto.
- Tests: Añadido `jest.global-setup.js` y mitigaciones para Windows con `mongodb-memory-server`.
- CI: `.github/workflows/ci.yml` añadido para ejecutar tests en Ubuntu (evita problemas Windows en CI).

Estado de pruebas

- Resultado observado al ejecutar `npm test`: PASS — 1 suite, 1 test. Se muestra una advertencia por la estrategia elegida para evitar `EPERM` en Windows:

"Running on Windows: skipping mongodb-memory-server.stop() to avoid EPERM issues."

Esto deja el proceso binario de `mongod` abierto localmente; en CI (Ubuntu) no ocurre. Recomendación: ejecutar tests en CI o aplicar `globalTeardown` si deseas que localmente se termine el proceso con `taskkill`.

Comandos para reproducir (PowerShell)

- Arrancar backend (recomendado en puerto 3002):

```
Set-Location 'C:\Users\usuko\Desktop\MitestasmentoDigital\DeadApp'  
$env:PORT = '3002'  
$env:NODE_ENV = 'development'  
npm run dev
```

- Verificar health (sin auth):

```
Invoke-RestMethod -Uri 'http://localhost:3002/health' -Method GET
```

Salida esperada:

```
{ "status": "OK" }
```

3. Verificar status (usar dev auth headers si Keycloak no está configurado):

```
Invoke-RestMethod -Uri 'http://localhost:3002/api/status' -Method GET -Headers @{ 'x-dev-user'='test-user'; 'x-dev-roles'='usuario_titular' }
```

Salida esperada:

```
{ "status": "OK", "mongoState": 1 }
```

4. Crear un activo de prueba (POST):

```
$body = @{ plataforma='TestPlatform'; usuarioCuenta='user.test@example.com'; password='S3cret!Perf'; notas='Activo de prueba'; categoria='OTRO' } | ConvertTo-Json  
Invoke-RestMethod -Uri 'http://localhost:3002/api/boveda/activos' -Method Post -Body $body -  
ContentType 'application/json' -Headers @{ 'x-dev-user'='test-user'; 'x-dev-roles'='usuario_titular' }
```

Salida esperada:

```
{ "message": "Activo digital guardado exitosamente" }
```

5. Leer activos (no revela contraseñas en claro):

```
Invoke-RestMethod -Uri 'http://localhost:3002/api/boveda/activos' -Method Get -Headers @{ 'x-dev-user'='test-user'; 'x-dev-roles'='usuario_titular' }
```

Salida esperada: objeto(s) de activo donde `passwordCifrada` contiene el texto cifrado (no el password en claro).

6. Recuperar secreto de un activo (solo si conoces el `id`):

```
Invoke-RestMethod -Uri 'http://localhost:3002/api/boveda/activos/<id>/secreto' -Method Get -  
Headers @{ 'x-dev-user'='test-user'; 'x-dev-roles'='usuario_titular' }
```

Salida esperada:

```
{ "secreto": "S3cret!Perf" }
```

Qué capturar en pantalla (pasos para el pantallazo que te piden)

- Una terminal con `npm test` mostrando PASS.
- Una terminal con `Invoke-RestMethod 'http://localhost:3002/health'` mostrando `{ "status": "OK" }`.
- Una terminal con `Invoke-RestMethod 'http://localhost:3002/api/status'` mostrando `mongoState: 1`.
- (Opcional) Una terminal con la respuesta de `GET /api/boveda/activos` y otra con `GET /api/boveda/activos/<id>/secreto` que demuestre que el secreto se obtiene mediante endpoint dedicado.

Notas finales y acciones pendientes (opcionales)

- Si prefieres que Jest no muestre el "Jest did not exit..." localmente, puedo añadir `jest.global-teardown.js` que intente `taskkill` a procesos `mongod` en Windows. Dímelo y lo implemento.
- Si quieres, genero un `REPORT.pdf` o una captura (no puedo tomar capturas desde tu máquina). Puedo crear un archivo `REPORT.md` (este) y tú haces el pantallazo desde tu equipo.

Fin del informe.