

Pasos para configurar las claves en GitHub Secrets:

y si Configuro una tarea de despliegue continuo Flutter/Dart con actions de Github para generar una apk... eso sería posible hacerlo sin exponer las claves? porque al generar la apk alguien podría obtenerlas de alguna manera????

Sí, es posible configurar un pipeline de despliegue continuo para generar una APK de Flutter/Dart en GitHub Actions sin exponer las claves API, pero es importante tomar medidas adicionales para protegerlas dentro de la APK. Aquí te explico los pasos y cómo puedes asegurarte de que las claves no sean accesibles en el APK:

Claves API en el código fuente y APK

Las claves API que se incluyen directamente en el código fuente o en archivos como `.env` pueden acabar dentro del APK final si no se maneja correctamente. Si alguien descompila el APK, podría potencialmente acceder a esas claves.

Opciones para evitar exponer las claves en el APK:

1. No incluir claves directamente en el código o el APK:

No almacenes las claves API directamente en archivos como `.env` o en el código fuente. En lugar de eso, usa servicios remotos para proporcionar las claves de forma segura cuando sea necesario.

2. Uso de servicios backend:

- Una buena práctica es manejar las claves API en un backend controlado por ti. El frontend (la aplicación móvil) realiza llamadas al backend y este es el encargado de hacer las solicitudes a los servicios externos que necesitan las claves API.
- Esto evita que las claves API se incluyan en el APK y expone únicamente el endpoint del backend, que tú controlas y puedes proteger con autenticación y otras medidas de seguridad.

3. Ofuscación de código:

- Si las claves deben incluirse en el APK por alguna razón, asegúrate de usar ofuscación de código en el APK para hacer más difícil que alguien pueda descompilar la APK y extraer información como las claves API.
- Flutter/Dart usa ProGuard para ofuscación en Android. Para habilitarlo, puedes agregar la configuración adecuada a tu proyecto. Esto no garantiza una seguridad total, pero añade una capa de dificultad.

4. Variables de entorno en GitHub Actions:

- Durante el proceso de generación del APK en GitHub Actions, usa **secrets** y variables de entorno de la misma forma que en los tests para inyectar las claves API temporalmente solo en el entorno de compilación, sin persistirlas en el código fuente.
- Una opción es usar GitHub Secrets y pasar las claves API como argumentos de compilación. Así, las claves no quedan incrustadas en el código, sino que son usadas solo durante la ejecución en el entorno CI/CD.

5. Build flavors y configuraciones separadas:

Puedes utilizar diferentes configuraciones para tu proyecto, como "build flavors" en Android, donde las versiones de producción y desarrollo usan distintas claves API. En producción, podrías usar un backend o servicios gestionados para no incluir las claves API directamente en la APK.

Ejemplo de configuración en GitHub Actions para compilar el APK:

Aquí tienes un ejemplo de cómo podrías generar una APK usando GitHub Actions, inyectando las claves API temporalmente durante el proceso de compilación, pero sin incluirlas en la APK final:

yaml

```
name: Build Flutter APK
on:
  push:
    branches:
      - main

jobs:
  build:
    runs-on: ubuntu-latest
    container:
      image: ghcr.io/cirruslabs/flutter:stable

    steps:
      - uses: actions/checkout@v4

      - name: Set up Flutter
        run: flutter pub get

      # Crear el archivo .env temporal para uso durante la compilación
      - name: Create .env file
        run: |
          echo "GOOGLE_API_KEY=${{ secrets.GOOGLE_API_KEY }}" >> .env
          echo "GEMINI_API_KEY=${{ secrets.GEMINI_API_KEY }}" >> .env
          echo "GOOGLE DIRECTIONS_API_KEY=${{
secrets.GOOGLE DIRECTIONS_API_KEY }}" >> .env
          echo "GOOGLE PLACES_API_KEY=${{ secrets.GOOGLE PLACES_API_KEY
}}}" >> .env

      # Compilar la APK
      - name: Build APK
        run: flutter build apk --release

      # Opcional: Subir la APK generada como artefacto o a un servicio
de despliegue
      - name: Upload APK
        uses: actions/upload-artifact@v2
        with:
          name: app-release.apk
          path: build/app/outputs/flutter-apk/app-release.apk
```

Seguridad de las claves en la APK:

1. **Al inyectar las claves temporalmente en GitHub Actions** durante el proceso de compilación, estas claves no deberían aparecer en el APK final si sigues las mejores prácticas, como no almacenar las claves directamente en el código fuente.
2. **Si usas servicios backend**, el APK no necesitará almacenar claves API en absoluto, lo que elimina completamente el riesgo de que se extraigan del APK.
3. **Ofuscación y otras medidas** pueden mitigar el riesgo si las claves deben estar presentes en el código, aunque no es la solución más segura.

Resumen:

- **No**, las claves API **no deben estar expuestas** durante el proceso de compilación si sigues estos pasos.
- La clave está en no incluirlas directamente en el APK final. Maneja las claves en un backend o úsalas temporalmente en el entorno de CI/CD sin que formen parte del código compilado.

Si necesitas más detalles sobre alguno de estos puntos o cómo configurarlo, házmelo saber.