GENERALIDADES DE LA CENTRAL DE PROCESAMIENTO DE INFRACCIONES DE TRÁNSITO (CPIT) Versión 2

La Central de Procesamiento de Infracciones de Tránsito (CPIT) es la plataforma que integra y recibe todas las diferentes fuentes de evidencias de las presuntas infracciones, procesa y dispone a la autoridad de tránsito la información para la imposición del comparendo.

La CPIT es un sistema integral diseñado para la gestión, control y seguimiento de infracciones de tránsito. Está construido sobre una arquitectura robusta y modular basada en microservicios, lo que permite un manejo eficiente de la información, flexibilidad en su integración y actualización constante con tecnologías modernas.

1. Construcción de la CPIT

La CPIT está construida con una arquitectura basada en microservicios y tecnologías modernas. A continuación, se detallan los principales componentes y tecnologías utilizadas.

1.1 Lenguaje de Programación y Frameworks:

- La mayoría de los microservicios están desarrollados en **Java 17**.
- Utilizan el framework **Spring Boot 3.0.5** para el desarrollo backend.
- La gestión de dependencias se realiza con **Gradle**.

1.2 Microservicios Principales:

- **Backend:** Contiene las funciones principales del sistema, como parametrización, validación de evidencias, consultas al RUNT, manejo de pico y placa, generación de evidencias manuales y tareas programadas (schedules).
- **Streaming:** Utiliza **FFMPEG** para el manejo de contenido multimedia y transmisión de video mediante el protocolo **HLS (HTTP Live Streaming)**.
- **Consultas y Auditoría:** Realiza consultas a la base de datos y generación de reportes, principalmente en formato **PDF** usando **JasperReports**.
- **Reportes:** Genera informes en formato **XLSX**.
- **Frontend:** Está desarrollado con **Angular 14**, usando **Angular Material 14**, **Bootstrap 5.3.2** y gestión de estados con **NGXS**.

2. Protocolo para la Conexión de la Cámara a la CPIT:

2.1 Protocolo para la Conexión de Cámaras a la CPIT

La conexión de la CPIT con una cámara se realiza a través del servicio de Streaming de la plataforma, utilizando el protocolo HLS (HTTP Live Streaming). Este protocolo garantiza la transmisión continua de video, compatible con múltiples dispositivos y navegadores. El servicio de streaming emplea FFMPEG para recibir el flujo de video desde la cámara, fragmentarlo y crear una lista de reproducción accesible desde la plataforma.

2.2. Flujo de Conexión

- Configuración de la Cámara: Las cámaras están configuradas para enviar el flujo de video directamente al servicio de streaming de la CPIT.
- Identificación: Cada cámara tiene un identificador único (event_radar_id) para organizar las evidencias.
- Visualización: La conexión se activa cuando hay usuarios visualizando el stream y se desactiva automáticamente al no haber actividad.

3. Protocolo de Conexión:

- La CPIT utiliza el protocolo **HLS (HTTP Live Streaming)** para conectar y transmitir el contenido desde las cámaras.
- Este protocolo es ampliamente compatible con navegadores y dispositivos debido a su origen en Apple.

4. Herramienta de Streaming:

- El servicio de streaming utiliza **FFMPEG**, que es un software especializado en la decodificación, codificación, transcodificación y transmisión de contenido multimedia.
- FFMPEG se encarga de recibir el flujo de video desde la cámara y generar pequeños fragmentos de video (segmentos) que se agrupan en una lista de reproducción.

5. Flujo de Conexión:

1. **Configuración de la Cámara:**
   - Las cámaras están configuradas para enviar el flujo de video directamente al servicio de streaming de la CPIT.
   - Cada cámara tiene un **identificador único (event_radar_id)** que permite organizar las evidencias capturadas.
2. **Conexión a la Plataforma:**
   - Cuando un usuario necesita visualizar el video en tiempo real, el servicio de streaming crea una conexión con la cámara específica.
   - Si no hay usuarios visualizando, la conexión se desactiva automáticamente para ahorrar recursos.
3. **Transmisión en la Plataforma:**
   - El servicio de streaming utiliza **HLS** para dividir el video en fragmentos y crear una lista de reproducción que el navegador pueda consumir.
   - El video se muestra en el frontend a través de un visor compatible con HLS.

6. Almacenamiento de Evidencias:

- Los videos y las imágenes capturadas por las cámaras se almacenan en **buckets de GCP**.

Secretaría Distrital de Movilidad
Calle 13 # 37 - 35
Teléfono: (1) 364 9400
www.movilidadbogota.gov.co
Información: Línea 195

- Cada bucket se organiza en una estructura de directorios según el mes, año, día y el identificador único del evento.
- Los archivos multimedia pueden estar en formatos como **MP4** o imágenes en **JPG/PNG o AVI**dependiendo de la configuración de la cámara.

7. Formatos de Archivos para Imágenes, Videos y Evidencias:

Los archivos generados y almacenados por la CPIT siguen los siguientes formatos:

- **Videos:** Principalmente en formato **MP4** o en fragmentos de video compatibles con el protocolo HLS.
- **Imágenes:** Generalmente en formato **JPG** o **PNG**.
- **Documentos Generados (Evidencias, Comparendos):** Formato **PDF**.
- **Informes y Reportes:** Formato **XLSX** para facilitar el análisis de datos.

8. Metadatos Asociados a las Evidencias en la CPIT

8.1 . Metadatos Generales de las Evidencias:

Las evidencias capturadas por las cámaras deben contener metadatos fundamentales para garantizar su validez y trazabilidad en el sistema CPIT. Los principales metadatos que deben acompañar una evidencia son:

- **Fecha y Hora de Captura:**
  - Registro preciso del momento en que se tomó la imagen o se grabó el video.
  - Este dato es fundamental para correlacionar la infracción con el contexto temporal.
- **Ubicación Geográfica:**
  - Identificación del punto donde la cámara está instalada.
  - Puede incluir coordenadas geográficas o una referencia al punto de fotodetección registrado.
- **Identificador Único del Evento (event_radar_id):**
  - Un ID único generado automáticamente por la cámara en el momento de la captura.
  - Este identificador permite vincular el archivo multimedia con la infracción correspondiente.
- **Identificación de la Cámara:**
  - Número o código único de la cámara que capturó la evidencia.
  - Esto asegura la trazabilidad en caso de revisión técnica o auditoría.
- **Código de la Infracción:**
  - El tipo de infracción detectada (por ejemplo, exceso de velocidad, cruce de semáforo en rojo).
  - Está parametrizado en la plataforma y vinculado con el evento.
- **Formato de la Evidencia:**
  - Se especifica el tipo de archivo (MP4 para videos, JPG/PNG para imágenes).

- Esta información permite una correcta clasificación y almacenamiento en los buckets.

8.2. Metadatos Específicos del Sistema:

- **Consecutivo de Comparendo:**
  - Un valor numérico que corresponde al consecutivo dentro del rango de comparendos asignados.
- **Estado de Validación:**
  - Indica si la evidencia ya ha sido revisada y validada por un agente.
- **Firma Digital o Certificación:**
  - En algunos casos, la evidencia incluye una certificación digital para garantizar su autenticidad.

8.3. Estructura de Almacenamiento en Buckets:

- Los metadatos asociados a cada archivo permiten su almacenamiento organizado en **buckets de GCP** bajo una estructura que sigue el esquema:
  - **[MES]*[AÑO]/ [DIA]/[EVENT_RADAR_ID]/[ID_EVIDENCIA]*[CONSECUTIVO]*[CODIGO_ DE_INFRACCION]*[TIPO].pdf**
- Esta estructura asegura que los archivos puedan ser fácilmente identificados y recuperados para su uso en procedimientos contravencionales o legales.

8.4 Metadata Asegurada con HLS:

- **Fecha y Hora de Captura:** Registro exacto del momento de captura.
- **Ubicación Geográfica:** Punto de instalación de la cámara.
- **Identificador Único del Evento:** Código generado automáticamente que identifica la evidencia.
- **Código de la Infracción:** Tipo específico de infracción capturada (ej., exceso de velocidad).
- **Identificación de la Cámara:** Número único asociado a la cámara que capturó el evento.

9. Interoperabilidad en la CPIT

La interoperabilidad en la CPIT se logra gracias a su arquitectura basada en microservicios y el uso de estándares abiertos para la transmisión y gestión de datos. A continuación, detallo los elementos que permiten la interoperabilidad del sistema:

10. Arquitectura Basada en Microservicios:

- La CPIT está construida con una arquitectura modular que permite que cada componente funcione de manera independiente, pero se comunique eficazmente con otros servicios.
- Los microservicios están desarrollados principalmente en **Java 17** usando el framework **Spring Boot 3.0.5**.
- Esta estructura permite integrar fácilmente nuevos módulos o servicios externos sin afectar la operación del sistema principal.

11. Protocolos de Comunicación:

- Los microservicios se comunican mediante protocolos HTTP/HTTPS, lo que garantiza una conexión segura y estandarizada.
- La autenticación y gestión de usuarios se realiza a través de **Keycloak** usando el protocolo **LDAPS (LDAP sobre SSL)** para garantizar seguridad en el manejo de identidades.
- Para la transmisión de contenido multimedia desde las cámaras, se utiliza el protocolo **HLS (HTTP Live Streaming)**, que permite el streaming continuo y la compatibilidad con navegadores modernos.

12. Manejo de Datos:

- Los datos generados por los sistemas de detección (como videos e imágenes) se almacenan en **buckets de GCP** con estructuras de datos jerárquicas y estandarizadas:
    - Carpeta organizada por **mes, año, día y el identificador único del evento (event_radar_id)**.
- Los formatos de archivo estándar (como **MP4**, **JPG**, **PNG**, **PDF**, **XLSX**) garantizan que otros sistemas puedan acceder a los datos sin problemas de compatibilidad.

13. Arquitectura y Tecnologías Utilizadas

13.1 Lenguaje de Programación y Frameworks

- **Lenguaje principal:** Java 17, seleccionado por su estabilidad, soporte a largo plazo y compatibilidad con microservicios.
- **Framework backend:** Spring Boot 3.0.5, utilizado para el desarrollo rápido y seguro de servicios web escalables.
- **Gestión de dependencias:** Gradle, empleado para administrar eficientemente librerías y versiones de componentes.
- **Frontend:** Desarrollado con Angular 14, empleando Angular Material 14 y Bootstrap 5.3.2 para interfaces modernas y responsive.

Secretaría Distrital de Movilidad
Calle 13 # 37 - 35
Teléfono: (1) 364 9400
www.movilidadbogota.gov.co
Información: Línea 195

- **Gestión de estados:** NGXS, implementado para manejar el flujo de datos de forma estructurada en el frontend.

13.2 Microservicios Principales

- **Backend:** Comprende la lógica principal del sistema, incluyendo la parametrización de infracciones, validación de evidencias, integración con el RUNT para la verificación de datos vehiculares, gestión de restricciones como el pico y placa, generación de evidencias manuales y programación de tareas automáticas (schedules).
- **Streaming:** Utiliza FFMPEG para la codificación, transcodificación y transmisión de video en tiempo real mediante el protocolo HLS. Esto garantiza una experiencia de visualización continua desde los dispositivos de monitoreo.
- **Consultas y Auditoría:** Implementa servicios para la consulta de información histórica y la generación de auditorías detalladas. Los reportes se generan en formato PDF utilizando JasperReports, facilitando la verificación documental.
- **Reportes:** Genera informes periódicos y bajo demanda en formato XLSX, facilitando el análisis cuantitativo de infracciones y desempeño del sistema.

13.3. Protocolos y Conexiones con Cámaras

- **Protocolo HLS (HTTP Live Streaming):** Garantiza la transmisión continua y compatible con múltiples dispositivos y navegadores.
- **Servicio de Streaming:** FFMPEG recibe el flujo de video desde la cámara, lo convierte en segmentos y crea una lista de reproducción dinámica para el frontend.
- **Identificación de Cámaras:** Cada cámara está registrada en la plataforma con un identificador único (event_radar_id), permitiendo un rastreo preciso de la evidencia capturada.
- **Desactivación Automática:** El sistema optimiza recursos apagando la conexión cuando no hay usuarios visualizando el video en tiempo real.

13.4. Almacenamiento y Gestión de Evidencias

- **Organización en Buckets de GCP:** Los videos y documentos se almacenan en estructuras jerárquicas basadas en fecha (mes, año, día) y el identificador del evento (event_radar_id), garantizando una recuperación eficiente.
- **Formatos de Video:** MP4 y fragmentos HLS para garantizar compatibilidad y calidad.
- **Formatos de Imágenes:** JPG y PNG, seleccionados por su alta resolución y baja pérdida de calidad.
- **Documentos Generados:** PDF para comparendos y otros documentos oficiales.

- **Informes:** Archivos XLSX que permiten el análisis y comparación de datos históricos.

13.5. Metadatos Asociados a las Evidencias

13.5.1 Metadatos Generales

- **Fecha y Hora de Captura:** Registro preciso, esencial para la correlación de la infracción.
- **Ubicación Geográfica:** Coordenadas exactas o referencia al punto de fotodetección.
- **Identificador del Evento (event_radar_id):** Garantiza la unicidad de la evidencia en el sistema.
- **Identificación de la Cámara:** Número o código único asignado a cada dispositivo de captura.
- **Código de la Infracción:** Clasificación según el tipo de falta cometida.
- **Formato de la Evidencia:** Definido como MP4 para videos o JPG/PNG para imágenes.

13.5.2 Metadatos Específicos del Sistema

- **Consecutivo de Comparendo:** Número progresivo asignado dentro del rango oficial.
- **Estado de Validación:** Indica si la evidencia ha sido verificada por un agente autorizado.
- **Firma Digital:** Certificación para garantizar la integridad y autenticidad del documento.

13.6. Interoperabilidad del Sistema

- **Arquitectura Modular:** Facilita la incorporación de nuevos componentes sin afectar los servicios existentes.
- **Protocolos de Seguridad:** HTTP/HTTPS para transmisión de datos y LDAPS mediante Keycloak para autenticación segura.
- **Compatibilidad Multimedia:** HLS garantiza la visualización en navegadores modernos sin necesidad de plugins adicionales.

13.7. Integración con Sistemas Externos

- **Sistemas de Gestión de Infracciones:** SICON, SIMUR y FENIX para la consolidación de datos de comparendos.
- **Validación de Datos Vehiculares:** Conexión directa con el RUNT para la verificación de información de vehículos y conductores implicados en infracciones.

Este documento proporciona una descripción exhaustiva de los componentes, tecnologías y protocolos empleados en la CPIT, resaltando su eficiencia operativa y capacidad de integración.

14. Integración con Sistemas Externos:

- La CPIT se conecta de forma interoperable con varios sistemas externos de gestión de infracciones:
    - **SICON, SIMUR y FENIX:** Para el envío de comparendos validados, lo que permite que los registros se mantengan actualizados en múltiples plataformas gubernamentales.
    - **RUNT (Registro Único Nacional de Tránsito):** Para verificar y validar la información de vehículos y conductores involucrados en las infracciones.

Secretaría Distrital de Movilidad
Calle 13 # 37 - 35
Teléfono: (1) 364 9400
www.movilidadbogota.gov.co
Información: Línea 195