“La ciudad en el contexto de la Internet de las cosas”

Universitat Oberta de Catalunya
y
Institut Municipal d'Informatica
del Ajuntament de Barcelona

Esteban Roman Castellanos
Master en Software Libre

Universitat Oberta de Catalunya

Ajuntament de Barcelona

Introducción: Estructura

- Introducción
  - Objetivos, estado del arte y propuesta de solución.
  - Política estratégica y requisitos iniciales
  - Planificación y tareas desarrolladas
- Desarrollo
  - Análisis, diseño y desarrollo de las app's.
  - Configuración del entorno de desarrollo
  - Plataforma de recepción de observaciones
  - Extracción de la información y generación de mapas web
- Conclusiones

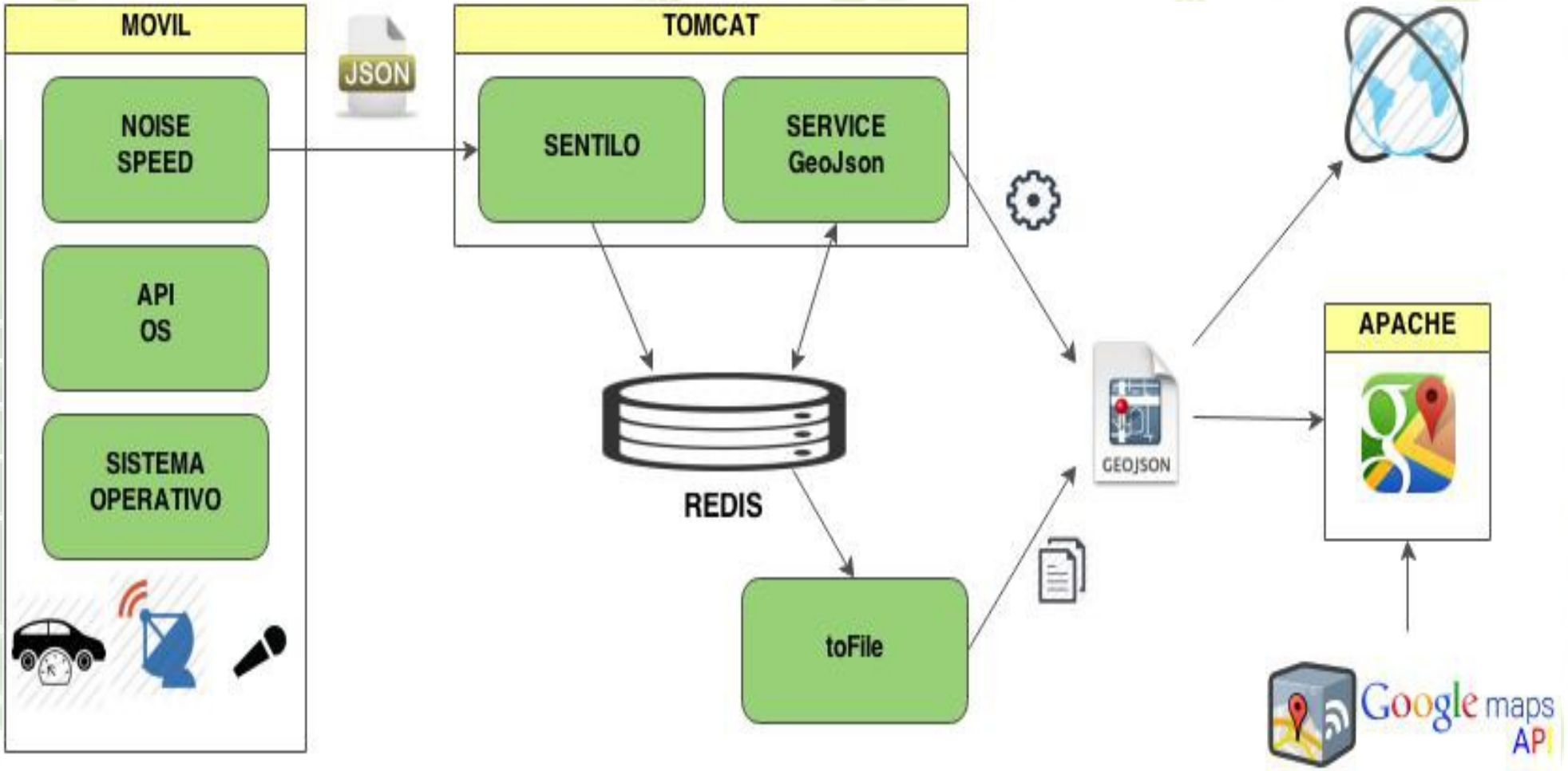
Introducción: Objetivos

- El proyecto se engloba en el desarrollo de la Smart City de la ciudad de Barcelona
- “Internet de las cosas”
- Habitabilidad y sostenibilidad de las ciudades.
- Análisis, diseño e implementación de un sistema de recogida y publicación de observaciones procedentes de dispositivos móviles y fijos.
- Prioridad del sistema de localización geográfica.

Introducción: Estado del arte

- Sistemas operativos para dispositivos móviles
- Técnicas de localización geográfica:
  - GPS y A-GPS: Satélites geoestacionarios.
  - LBS: Cell ID y puntos de acceso
  - Métricas: TTFF, consumo y precisión
- Medición de magnitudes físicas
- Servicios web: SOAP y REST
- Desarrollo de mapas Web de observaciones

Introducción: Solución propuesta



Política estratégica

- Toda la información obtenida debe estar disponible de forma gratuita y universal
- Todo el software debe ser Open Source
- Debe cumplir la LOPD
- La recepción de datos debe ser anónima
- Debe ser extensible, adaptable y de alta disponibilidad.

Requisitos funcionales

- Contendrá todos los elementos necesarios: app, plataforma de recepción, procesos de extracción de datos y representación de mapas web.
- Cada observación contendrá la localización geográfica, una marca de tiempo y la medida.
- Los datos se almacenarán de forma eficiente y segura.
- Se publicarán en forma de texto y sobre un mapa web.

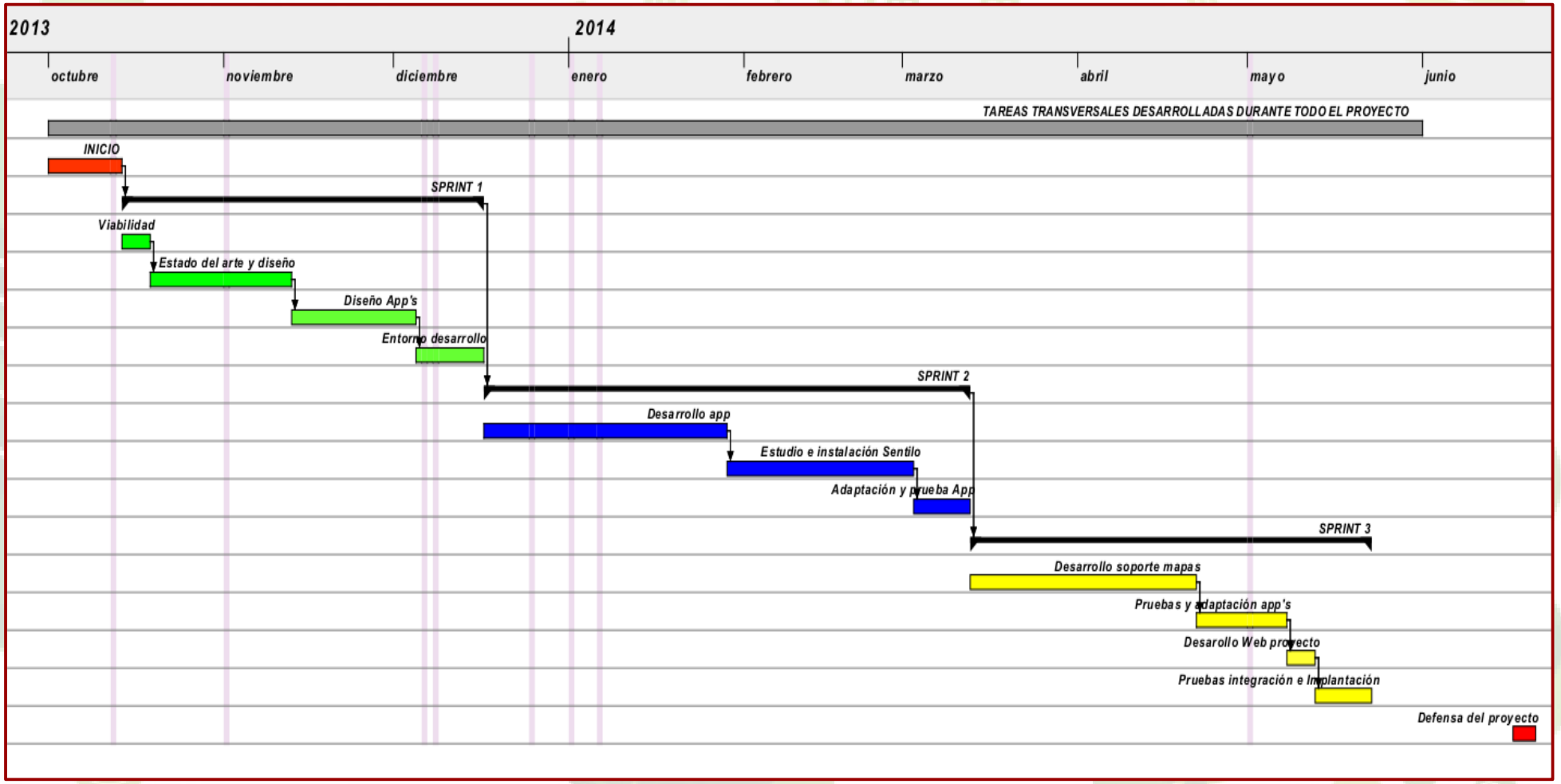
Planificación

- Metodología ágil: Scrum
- Objetivos y recursos no están completamente determinados desde el principio
- Adaptación a los cambios de requisitos
- Product Owner, Scrum Máster y desarrollador
- Tres Sprint con un producto más definido.

Tareas

- Viabilidad y estudio del estado del arte.
- Análisis y diseño las app's: Noise y Speed.
- Preparación del entorno de trabajo.
- Desarrollo de las app's y pruebas unitarias.
- Instalación de la plataforma Sentilo y los servidores necesarios.
- Desarrollo del servicio web de publicación en tiempo real.
- Desarrollo del servicio de publicación de históricos
- Desarrollo de la web del proyecto y los mapas web.
- Pruebas de integración y mejora del producto.

Diagrama de Gantt

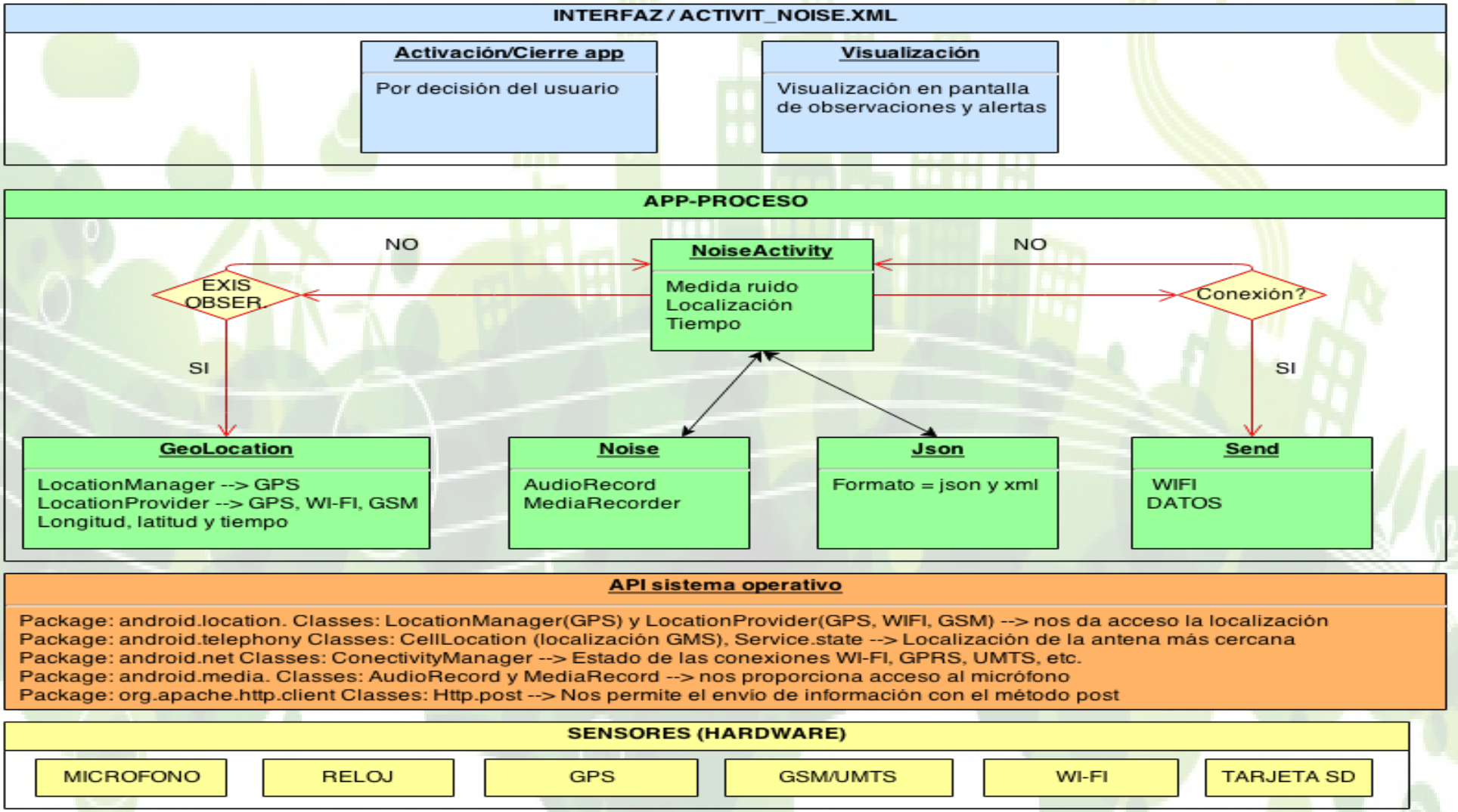


Esteban Roman Castellanos
Master en Software Libre

Análisis y diseño de las app's

- Análisis del sistema operativo Android y su API
- Noise: enviará medidas de ruido y podrá usar los tres proveedores de localización geográfica de forma adaptativa. Uso urbano con prioridad de la localización a la precisión.
- Speed: Enviará la velocidad y solo utiliza el GPS. Uso genérico cuando la precisión es prioritaria.
- Ambas app's se adaptan al proveedor de conexión que esté activo: WIFI o linea de datos del dispositivo.

Arquitectura



Entorno de desarrollo

- SDK de Android y emuladores
- Eclipse y sus distintos plugin
- Dynamic Web Project: servicios web
- Junit: gestión de las pruebas
- JavaDoc: generación de la documentación
- Git: sistemas de control de versiones

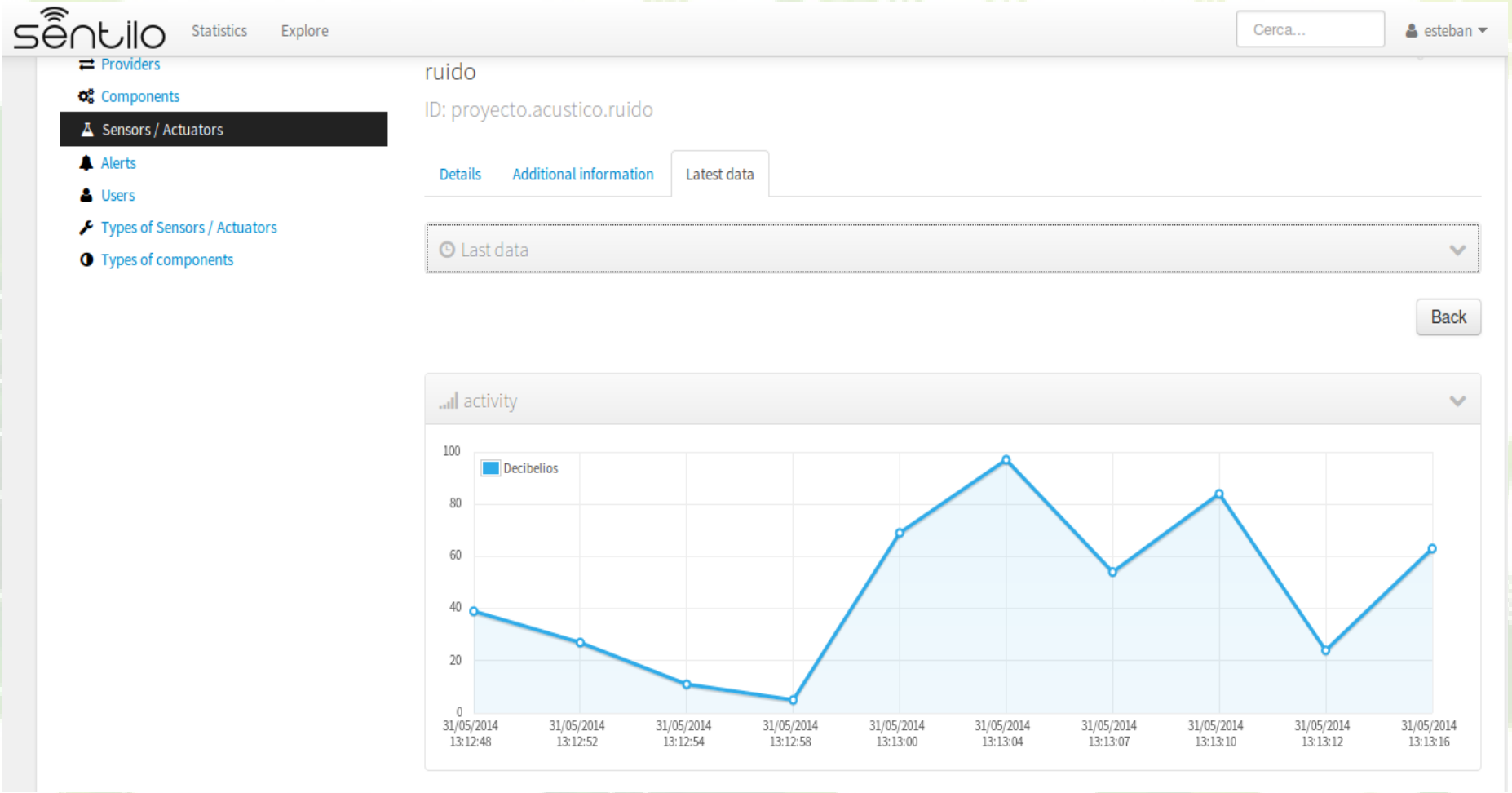
Desarrollo y uso de las app's

- API Level: Minimum Required SDK y Build SDK
- AndroidManifest y permisos sobre el hardware.
- Pruebas sobre emuladores y dispositivos físicos
- Consumo del 15% al 30%
- Precisión de medidas va de 1 Km a 10 m

Plataforma de recepción: Sentilo

- Compilación y configuración del código fuente (Maven)
- Instalación y configuración de los servidores
  - Redis, MongoDB y Mysql
  - Tomcat
- Puesta en marcha de la plataforma
  - Despliegue de la aplicación web: Catalogo
  - Arranque de los servicios: API 8081
- Representación de dispositivos fijos

Plataforma de recepción: Sentilo

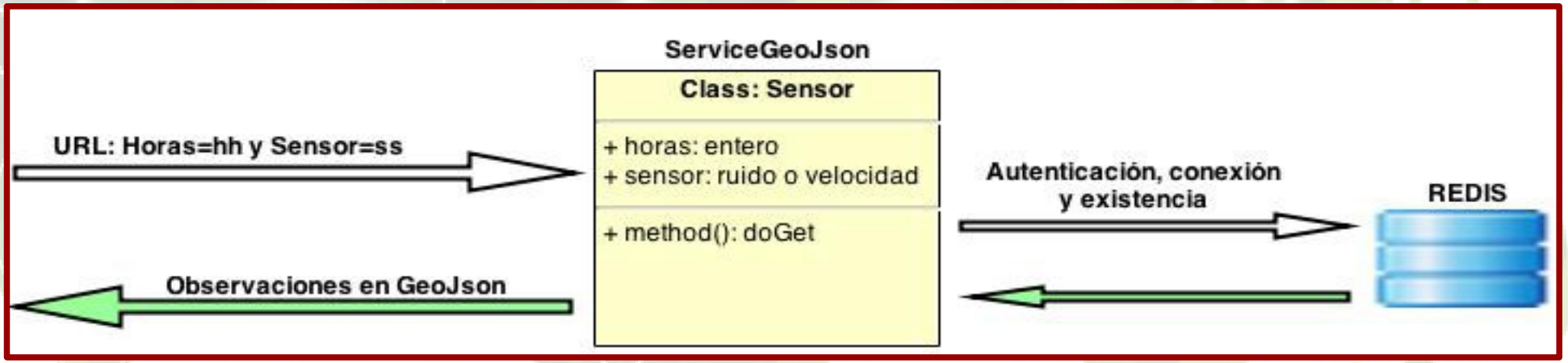


Exportación en formato texto

- Los datos almacenados en la plataforma son accesibles de forma anónima de dos formas:
  - Mediante el servicio web ServiceGeoJson (servlet)
  - Mediante ficheros históricos
- Ambos servicios utilizan el cliente Jedis para acceder a los datos almacenados en Redis.
- La estructura interna de datos está organizada en dos tipos de objetos Redis: Sort Set y Hash.

ServiceGeoJson

- Desplegado sobre Tomcat ofrece las observaciones correspondientes a dos parámetros contenidos en la URL.
- Envía los datos en formato GeoJson o códigos de error html: 200, 202, 400, 401, 404 o 503



Ficheros de históricos

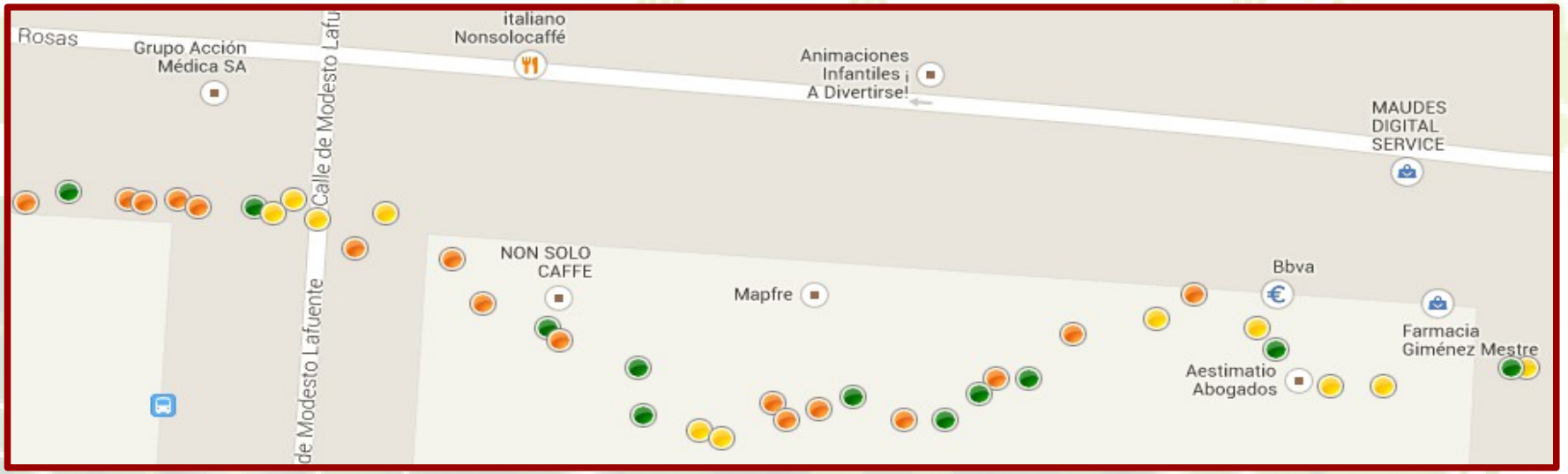
- Ofrecen el total de las observaciones de cada uno de los sensores
- Son generados automáticamente cada intervalo de tiempo determinado
- El acceso al fichero correspondiente al sensor Speed se realiza mediante la siguiente URL:

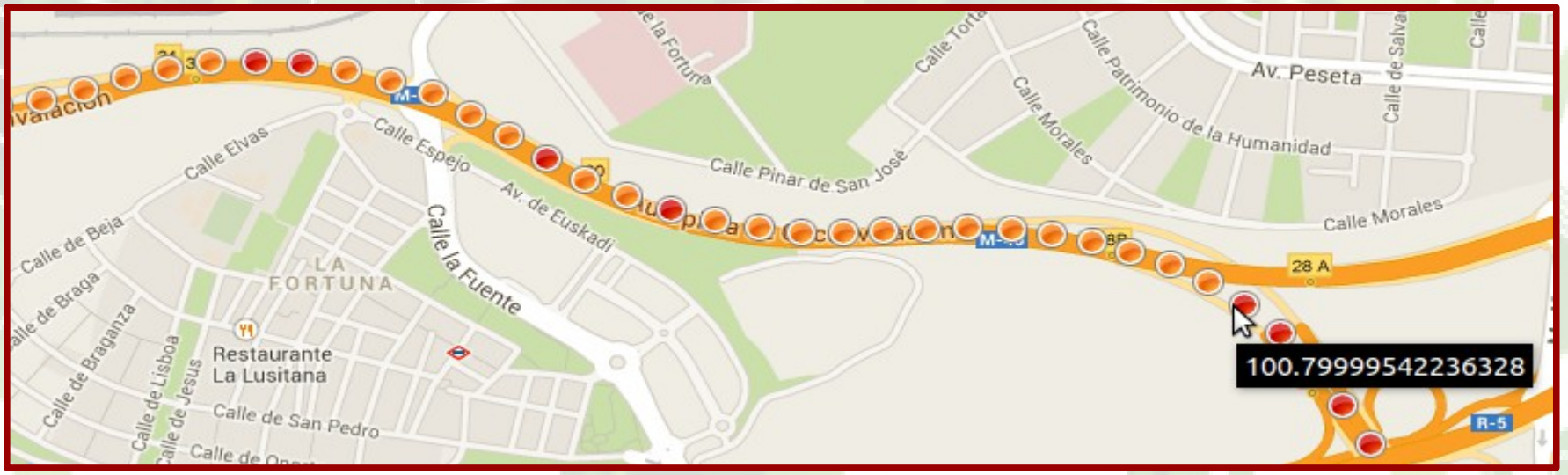
  http://www.smartcitything.es/speed.json

Generación de mapas web

- Versión 3 del API de JavaScript de Google Maps
- Superposición de los datos procedentes del servicio web y ficheros históricos.
- Formato GeoJson y permisos explícitos en la cabecera de respuesta
- Tipos de representación de puntos
- Precisión de observaciones

GPS vs Antenas de telefonía





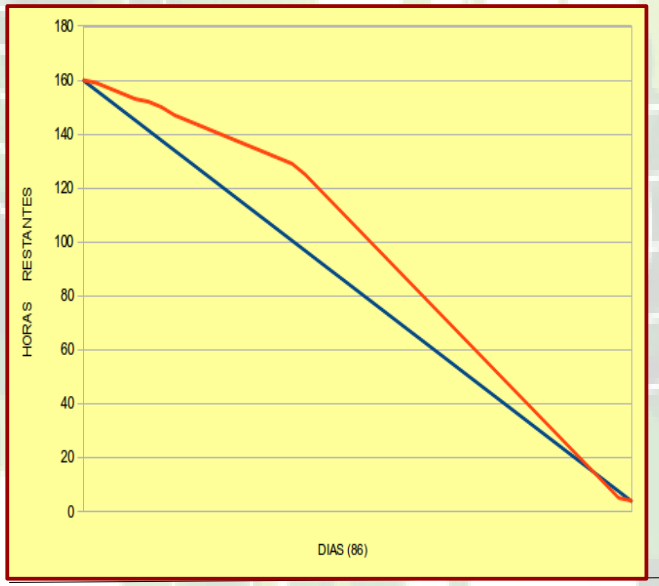
Esteban Roman Castellanos
Master en Software Libre

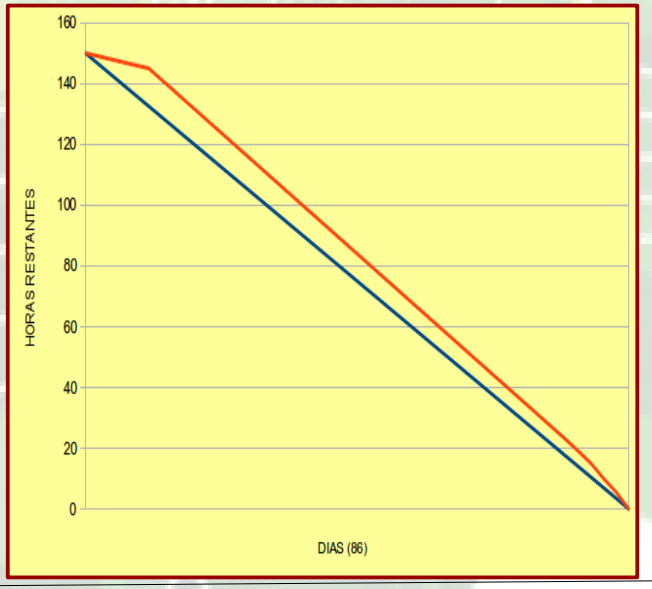
Web del proyecto

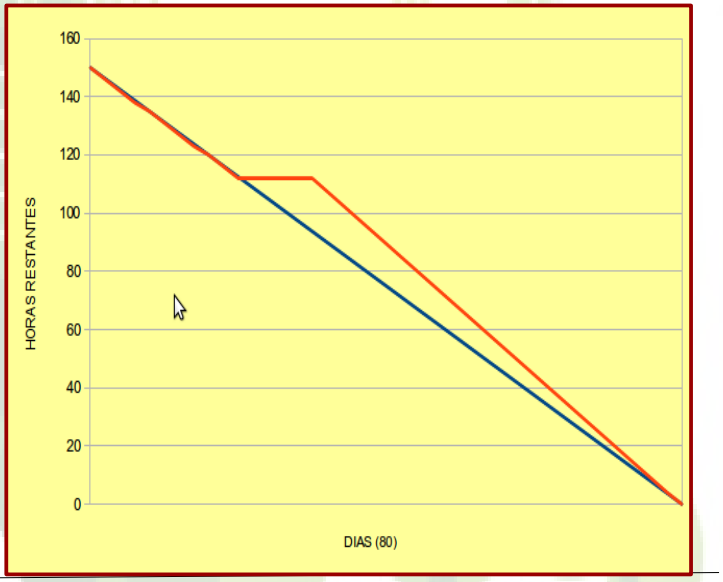
- HTML5, JavaScript y CSS sobre Apache2
- Descarga e instalación de las app's
- Acceso a todas las secciones del proyecto.
  - Documentación de JavaDoc
  - Código fuente con GitWeb
  - Mapas en tiempo real e históricos
  - Plataforma de recepción
  - Dispositivos fijos para una futura ampliación
- www.smartcitything.es

Burn Down Chart y retrospectiva

- Adaptación a las circunstancias cambiantes
- Aceptación por parte del cliente







Conclusiones

- Software utilizado: Debian 7, Tomcat7, Apache2, Maven2, Sentilo, Redis, MongoDB, MySQL, Eclipse, Junit, JavaDoc, Git, GitWeb, SSH, SSHFS, Filezilla, LibreOffice, etc.
- Elementos desarrollados
  - Aplicaciones para móviles: Noise y Speed
  - Servicio web: ServiceGeoJson
  - Procesos de generación de ficheros: speed.json y noise.json
  - Web del proyecto
- Envío de observaciones desde dispositivos fijos utilizando placas Arduino.
- Implementación de las medidas de contaminación acústica.