

# UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS



## Facultad 1

**Título: Servicio para el seguimiento de la utilización de la Plataforma C.U.B.A por parte de los usuarios.**

### **Autor(es):**

Isnel Alejandro Quincoses Pérez

### **Tutor(es):**

MSc. Yadiel Pérez Villazón  
Ing. Lisdania de la Caridad Delgado Olivera

2020

## Introducción

A lo largo de los últimos años las organizaciones y empresas en el mundo han optado por unirse a la red de redes con el objetivo de llegar a un mayor número de usuarios y de esta forma no quedarse obsoletas con respecto a su competencia. Pero con la gran cantidad de datos que se maneja en internet nació la necesidad de procesar estos para el mejoramiento de los sistemas propios creando el concepto de Analítica Web.

La Analítica Web es la ciencia y el arte de mejorar páginas webs para incrementar sus beneficios a través de fidelizar al cliente. Es ciencia porque usa estadísticas, técnicas y procesos metodológicos, y es un arte porque se necesita encontrar la mezcla perfecta que produzca conocimientos prácticos que se apliquen a tu negocio. (Jiménez, 2017).

¿Por qué necesitaría un sitio web aplicar la Analítica Web?, en palabras de Peter Ferdinand Drucker que fue Profesor de negocio, abogado de carrera y considerado uno de los mayores filósofos en administración del siglo XX: “Lo que se puede medir se puede mejorar”. Analizando esta frase pudiéramos decir que la utilización de la Analítica Web para medir métricas que nos ayudan a entender cómo lo estamos haciendo en relación a los objetivos que nos hemos marcado utilizando los KPI (*Indicadores de Rendimiento Clave*) y así un posterior mejoramiento del sitio web.

La mayoría de estas fuentes referentes en el campo de la Analítica Web tienen una relación estrecha tanto en lo profesional como en lo científico con el campo del marketing. A su vez, se encontraron investigaciones las cuales aplican este conocimiento desde un punto técnico y específico.

En estos momentos en nuestro país existe un auge de uso de las tecnologías y aplicación de soluciones informáticas, todas en pos al desarrollo de sistemas dirigidos a automatizar procesos de importancia para la sociedad, para lograr una mejor productividad en los servicios y mayor calidad, tratando de satisfacer las necesidades de los usuarios.

Cuba, a pesar de ser un país subdesarrollado y potentemente bloqueado no se encuentra ajena del acceso a los medios digitales y cada día más cubanos se acercan a esta nueva forma del conocimiento. La mayoría de las instituciones de los diferentes sectores poseen conexión al menos a la intranet cubana y muchos de

ellos acceso a red de redes. Otro dato importante es que a partir del 2014 ETECSA<sup>1</sup>, comenzó a habilitar zonas Wi-fi<sup>2</sup> públicas y a mediados del 2016 habían alcanzado al menos 4 plazas de las 15 principales ciudades cubanas, mientras se preveé que para finales del 2020 el 50 % de los hogares cubanos tengan acceso de banda ancha (Manresa, 2016).

Otro elemento a resaltar es que se ha constatado un crecimiento de los dominios .cu, la **¡Error! No se encuentra el origen de la referencia.** proporcionada por el monitor de dominios cubanos<sup>3</sup>, detalla un aproximado de la cantidad de direcciones web, así como el número de subdominios de los principales dominios de segundo nivel del país (García Torres, 2017):

<b>Direcciones web</b>		<b>2354</b>			
<b>com.cu</b>	<b>36</b>	<b>edu.cu</b>	<b>154</b>	<b>sld.cu</b>	<b>344</b>
<b>cult.cu</b>	<b>201</b>	<b>lcrt.cu</b>	<b>95</b>	<b>bloguea.cu</b>	<b>27</b>
<b>uci.cu</b>	<b>23</b>	<b>gob.cu</b>	<b>82</b>	<b>rimed.cu</b>	<b>30</b>

Tabla 1: Cantidad de sitios por dominios tomados de <https://monitorcuba.redcuba.cu>

La mayoría de los sitios cubanos no están accesibles desde Internet por lo que buscadores como Google<sup>4</sup> o Yahoo<sup>5</sup> no pueden indexar las URLs<sup>6</sup> generadas por estas aplicaciones, además, muchos usuarios que utilizan los medios digitales tienen limitada la navegación solamente a la intranet cubana. Por la necesidad de contar con un sistema que permita centralizar toda la información que se genera a cada instante por los sitios web cubanos y proveer una solución que brinde funcionalidades para la búsqueda en la web cubana, el centro anteriormente conocido como Centro de Ideoinformática (CIDI) en el año 2015 despliega la Plataforma C.U.B.A<sup>7</sup>, una solución informática que permite la recuperación de información de todos los

<sup>1</sup> La **Empresa de Telecomunicaciones de Cuba S.A.**, tiene la misión de "Brindar servicios de telecomunicaciones que satisfagan las necesidades de los clientes y la población, así como respaldar los requerimientos del desarrollo socio-económico del país".

<sup>2</sup> Es una tecnología que permite la interconexión inalámbrica de dispositivos electrónicos.

<sup>3</sup> Sistema para la evaluación de los sitios web cubanos que están prestando servicios.

<sup>4</sup> Es una compañía estadounidense fundada en septiembre de 1998 cuyo producto principal es un motor de búsqueda.

<sup>5</sup> Es una compañía norteamericana focalizada en brindar prestaciones relacionadas con Internet.

<sup>6</sup> Uniform Resource Locator (Localizador Uniforme de Recursos).

<sup>7</sup> Contenidos Unificados de Búsqueda Avanzada.

dominios que se encuentran subordinadas al TLD de primer nivel .cu. Por la complejidad que requiere las tecnologías en la que está soportada esta solución, fue desplegada en una infraestructura de 19 servidores. Entre los principales componentes que presenta la Plataforma C.U.B.A están: los balanceadores webs, servidores dedicados para la interfaz que permite realizar las búsquedas, servidores de indexación y rastreadores.

De manera general C.U.B.A cuenta con una aplicación web (www.redcuba.cu) que brinda las opciones básicas para la recuperación de información, entre las funcionalidades principales se encuentran:

- Búsqueda básica y avanzada en la web.
- Búsqueda de imágenes.
- Búsqueda de documentos de diferentes formatos.
- Directorio (principales sitios por ministerios e instituciones)
- Envío de comentarios.
- Registro y autenticación de cuentas de usuarios.
- Compartir enlaces de resultados de búsquedas.
- Convertir monedas.
- Descargar fondos de pantallas con el estilo de la Plataforma C.U.B.A.

A pesar de que la plataforma ha sido estable durante los últimos tres años de explotación, la misma no cuenta con un mecanismo que permita darle un seguimiento a la utilización por parte de los usuarios de las distintas funcionalidades que brinda. No es posible conocer la utilización real de:

- Cantidad de búsquedas en los distintos módulos (web, imágenes, documentos).
  - Cantidad de búsquedas provenientes de usuarios nacionales o extranjeros.
  - Consultas en tiempo real; términos más buscados, tendencias de búsquedas.
  - Tiempos de respuesta de las consultas y cantidad de resultados de los mismos.
  - Cantidad de clic propiciado por el usuario
  - Cantidad de conversiones de monedas.
  - Cantidad de enlaces compartidos.
-

- Cantidad de usuarios registrados por día, y por mes.
- Cantidad y cuáles son los fondos de escritorio que más se descargan.

Las dificultades antes mencionadas tienen implicaciones en la satisfacción de los usuarios que utilizan este recurso para la búsqueda de información en la red cubana, porque el equipo de desarrollo no puede mejorar los resultados de búsqueda ni enriquecerla con funcionalidades si no tienen información relacionada con las búsquedas. Además, es indispensable conocer el uso real de la plataforma para poder planificar una expansión de los recursos y servicios.

Teniendo en cuenta lo anteriormente descrito se plantea como **problema científico**:

¿Cómo contribuir al registro de los eventos sobre la utilización de la Plataforma C.U.B.A. por parte de los usuarios?

Para dar respuesta al **problema científico** se plantea como **objetivo general**:

Desarrollar un servicio que facilite el registro de eventos sobre la utilización de la Plataforma C.U.B.A por parte de los usuarios.

A partir del **objetivo general** se definen los siguientes **objetivos específicos**:

- Elaborar el marco teórico referencial de la investigación teniendo en cuenta las fuentes existentes para extraer la información relacionada con sistemas de analíticas web.
- Describir el análisis y diseño de la solución para el registro de eventos sobre la utilización de la Plataforma C.U.B.A por parte de los usuarios.
- Desarrollar e implementar la solución para el registro de eventos sobre la utilización de la Plataforma C.U.B.A por parte de los usuarios.
- Validar la solución para el registro de eventos sobre la utilización de la Plataforma C.U.B.A por parte de los usuarios.

Para guiar la investigación se propone como **objeto de estudio**: La investigación de sistemas homólogos basados en la Analítica Web y arquitectura orientada a servicios para una mejor comprensión y análisis que se encamine en una solución viable que cumpla las necesidades del cliente **campo de acción**: Servicios que permita la captura automática eventos del usuario en la Plataforma C.U.B.A.

## Posibles resultados:

1. Microservicio como solución de backend el cual automatizaría la captura de eventos realizados por parte de usuario en su interacción con la plataforma C.U.B.A.

Como **aporte teórico** se plantean un grupo de métricas para el seguimiento del uso de la Plataforma C.U.B.A por parte de los usuarios.

El **aporte práctico** estriba en la solución informática que permite almacenar en el tiempo mediciones asociadas a un grupo de métricas para el seguimiento del uso de la Plataforma C.U.B.A por parte de los usuarios.

Para mostrar el desarrollo de la investigación y los resultados, el trabajo se ha estructurado en tres capítulos, además de las Conclusiones, Recomendaciones, Referencias Bibliográficas y Anexos.

En el primer capítulo se realiza un estudio referente a las tecnologías de Analítica Web que permitan principalmente recopilar, analizar y visualizar en tiempo real métricas asociadas a la creación de eventos con utilización de API'S REST y las tecnologías que se utilizarán para dar solución a la problemática y una descripción de los principales elementos relacionados con la infraestructura de la Plataforma C.U.B.A.

En el segundo capítulo se fundamentan las tecnologías y métricas a utilizar, así como la descripción de la solución tecnológica, requisitos y lógica de funcionamiento usando el escenario 2 de la metodología AUP UCI para poder lograr la creación de un microservicio, que produzca todos los requisitos establecidos por el cliente para así obtener un plan para nuestro sitio un punto de referencia para todos nuestros usuarios

En el tercer capítulo se realiza la validación de la solución tecnológica para el seguimiento de experiencia del usuario con la Analítica Web del sistema de la Plataforma C.U.B.A, a través de la aplicación del criterio de expertos en su variante Delphi para determinar la aplicabilidad y nivel de influencia de la estrategia. Además, contempla el análisis comparativo, con relación al tiempo de atención a las incidencias y a la satisfacción de los usuarios con el servicio prestado.

## **CAPÍTULO 1: Fundamentación teórica de la investigación.**

En el presente capítulo se pretende abordar las principales ideas y conceptos relacionados con la solución para el registro de eventos de la utilización de la Plataforma C.U.B.A por parte de los usuarios. Así como una descripción de la plataforma con las que se trabajarán para la aplicación de la solución y las diferentes tecnologías y herramientas que intervienen en el sistema de registro de eventos. Por otra parte, se analizan las principales soluciones existentes en la actualidad en el campo de la Analítica Web.

### **1.1. La Analítica Web**

La Analítica Web no trata de medir el tráfico de nuestro sitio web, sino de entenderlo y analizarlo para realizar mejoras en el sitio y, finalmente, conseguir los objetivos que nos hemos establecido para nuestro sitio web. (*¿Qué es la Analítica Web?*, 2014).

En la actualidad el desarrollo e impacto que ha tenido el internet en el sistema de vida cotidiana de los cubanos ha posibilitado al país llegar a los usuarios y ofrecerles una variedad de servicios web, algunos como alternativa de servicios web internacional a los cuales no tenemos accesos. Entre las alternativas que se crearon se encuentra la Plataforma de Contenidos Unificados de Búsqueda Avanzada (C.U.B.A.) debido a su estabilidad durante su tiempo en línea desea seguir desarrollándose y mejorando, para esto se desea conocer más sobre la utilización que dan los usuarios a la plataforma, así como conocer que le agrada o disgusta de las funcionalidades que ofrece esta plataforma. Para esto se plantea el estudio en el campo de la Analítica Web al ser esta según la WAA (*Web Analytics Association*) "la Analítica Web es la medición, colección, análisis y reporte de los datos de Internet para el entendimiento y optimización de tu página web" (*¿Qué es la Analítica Web?*, 2014).

#### **Pasos para analizar un sitio web**

- Establecer los objetivos del sitio y los KPI (*Indicadores Clave de Rendimiento*).
- Establecer el periodo de tiempo en el que se van a recoger los datos.
- Elegir la herramienta adecuada para realizar la analítica.
- Analizar la estructura del sitio web, tenemos que ver cómo está organizado, el proceso que siguen los usuarios para llegar al objetivo.

- Etiquetar las diferentes páginas de nuestro sitio web para que se puedan medir los datos. Cada página del sitio debe tener su URL específica.
- Medir y obtener informes.
- Revisar los informes, analizar los datos y realizar cambios que sean necesarios para alcanzar los objetivos establecidos.

### **Métricas avanzadas (KPI)**

Los **KPI** (Key Performance Indicators) son indicadores claves del rendimiento. Son métricas que nos ayudan a entender cómo lo estamos haciendo en relación a los objetivos que nos hemos marcado. Dado que cada empresa tiene objetivos diferentes, los KPI son propios de cada negocio. (*¿Qué es la analítica web?*, 2014)

**Métrica:** Las métricas son colecciones de entradas de datos numéricos organizadas en grupos de listas ordenadas cronológicamente. Cada entrada de datos consta de un valor de medición registrado, la marca de tiempo en que se realizó la medición y un conjunto de propiedades que lo describen. Cuando las entradas de datos de una métrica se segmentan en intervalos de tiempo fijos y se resumen mediante una transformación matemática de alguna manera significativa, se pueden presentar como series de tiempo e interpretarse en gráficos bidimensionales. (Ligus, 2013).

### **Métricas básicas de analítica web**

Son aquéllas que nos permiten analizar el tráfico en nuestro sitio web, saber qué páginas son las más visitadas, cuánto tiempo pasan los visitantes o si regresan a nuestro sitio.

- Visitantes únicos o exclusivos.
- Porcentaje de visitas nuevas.
- Páginas vistas.
- Páginas/visita.
- Promedio de tiempo en la página y duración media de la visita.
- Porcentaje de rebote.
- Porcentaje de salida.

## **Evento en la Analítica Web**

Luego de haber comprendido que la Analítica Web define la estrategia para recabar datos al utilizar métricas para la optimización de nuestro sitio, debemos definir las acciones que realiza el usuario en el sitio web a lo cual se le conoce como Evento. Nos referimos a Evento a las interacciones que realiza el usuario con los contenidos que se puede medir de forma independiente a partir de una página web. Las descargas, los clics, en los enlaces, los envíos de formularios, etc. Son ejemplos de acciones que se pueden analizar como Eventos. (Jiménez de la Calle, 2017)

## **Seguimiento del usuario**

El objetivo del seguimiento online que deseamos es el conocimiento detallado de los usuarios que interactúan con la plataforma registrando los eventos de las acciones que realizan para conocer su comportamiento y preferencias lo cual servirá para el mejoramiento y optimización de la plataforma C.U.B.A. logrando un uso sencillo y mejorando la experiencia del usuario.(Merelles & de Borja, 2019)

Llevándolo a un punto de vista más técnico el seguimiento del usuario es rastrear la actividad digital que realiza el usuario en el sitio web por el cual navega de los cuales podemos saber:

- La cantidad de búsquedas del usuario en los motores de Búsqueda.
- La frecuencia con el usuario visita una página.
- En que da clic el usuario.
- Descargas que realiza.

## **1.2. Análisis de soluciones similares**

Para el análisis de soluciones similares se investigaron varios sistemas reconocidos de Analítica Web, los cuales a pesar de cumplir algunos de los requisitos para registrar las actividades de los usuarios en la plataforma C.U.B.A. no poseen herramientas o estas que si no son gratis para registrar algunas actividades de servicios específicos de esta plataforma y a su vez ofrecer un entorno para el análisis de la utilización de estos servicios. A pesar de esto la investigación y análisis de estos sistemas permitió entender el funcionamiento del registro de eventos para el seguimiento del usuario necesario para la solución. Los sistemas similares investigados para la solución son los siguientes.

### **1.2.1. Google Analytics**

Google Analytics es una de los muchos sistemas que hay para medir datos obtenidos de una página web. De forma anónima, el programa lleva a cabo un seguimiento de la forma de la cual los usuarios interactúan con el sitio web. Dicho seguimiento incluye la procedencia, acciones realizadas y si han alcanzado alguno de los objetivos de conversión del sitio. Google Analytics además, realiza seguimiento de datos de comercio electrónico combinándolos con información sobre la campaña y conversiones para conseguir así más detalles sobre el rendimiento de sus campañas. Como todas las aplicaciones y herramientas que existen Google Analytics tiene sus ventajas y sus desventajas, que son las que van a hacer que un público elija este programa de Analítica Web para analizar y obtener datos, u otro.(Mateos Fernández & Sparapani Galicia, 2017)

En este mismo documento de(Mateos Fernández & Sparapani Galicia, 2017)se señalan las siguientes ventajas y desventajas:

#### **1.2.1.1. Ventajas de Google Analytics**

- Genera más de 80 informes.
- Paquete de análisis web completo y potente.
- Hay multitud de herramientas que incluyen la API de Google Analytics por defecto y permiten una fácil conexión lo cual simplifica mucho la extracción de datos.
- Todos los datos que extrae se pueden transformar a Excel y PDF, para enviar o almacenar fácilmente.

#### **1.2.1.2 Desventajas de Google Analytics**

- Es gratis solo en su versión básica, esto conlleva a que algunas de las herramientas que posee y cumplen con parte de la solución no estén a nuestra disposición.
- No posee una forma de analizar o registrar algunas de las funcionalidades particulares de la plataforma C.U.B.A., solo logra registrar eventos generales de los usuarios.
- No se puede recuperar información si no se registra, ante cualquier error de código.
- Menor privacidad de todos los datos al alojarse en Google el cual es un servidor externo propiedad de Google.
- Se encuentran variaciones con respecto a otros programas, debido a que Google Analytics procesa una gran cantidad de datos, y esto le lleva a que en situaciones concretas pueda cometer errores.

- Te da información general del usuario, no especifica como otras herramientas de Analítica Web.
- No posee una forma de analizar o registrar algunas de las funcionalidades particulares de la plataforma C.U.B.A., solo logra registrar eventos generales de los usuarios.

### **1.2.2. Matomo:**

Anteriormente conocido como Piwik, Matomo es un sistema para el análisis de tráfico de sitios web, que ofrece informes en tiempo real sobre visitas, páginas y otras métricas imprescindibles en la gestión de páginas web y tiendas online. Matomo es capaz de funcionar en sitios de todo tipo de tráfico y ayudar a sus administradores a obtener los indicadores más importantes de funcionamiento. Desde las visitas, usuarios, objetivos de conversión, palabras clave y mucho más. Es apto para tráfico pequeño y grande, incluso para sitios con millones de visitas por día. Integrado con un software de eCommerce, es capaz de analizar el rendimiento, las ratios de conversión (El ratio de conversión es un métrica que nos indica el porcentaje de usuarios que realizan alguna acción en nuestro blog o web.), y otras informaciones de las tiendas online. El administrador es capaz de acompañar el log de acciones de los usuarios y ver qué es lo que se está visitando, qué caminos siguieron las personas que realizaron una compra, en qué situaciones se abandonaron los carritos de la compra, etc.(Arsys, 2018)

#### **Algunas de sus características y ventajas más destacadas son:**

- El sistema ofrece reportes en tiempo real de manera predeterminada. Sin embargo, para sitios de alto tráfico se puede elegir la frecuencia con la que Matomo procesa los informes.
- Es capaz de adaptarse al nivel de privacidad deseado por el administrador.
- Es completamente Open Source. Cualquier instalación ofrece el software completo sin limitaciones. Instalado en más de 1 millón de sitios web.
- Localización geográfica con mapas en tiempo real.
- Posibilidad de envío de email con informes, seguimiento de objetivos, campañas, cuya frecuencia se puede configurar.
- Hace seguimiento del tráfico que proviene de motores de búsqueda.
- Calcula el tiempo que pasan los usuarios en el sitio de una manera fiel.
- Propiedad de los datos 100%.

#### **Desventajas que posee:**

- Aunque Matomo es una herramienta Open Source, no es gratuita por conllevar un consumo de servidores. Al contrario de Google Analytics en donde los servidores corren a cargo de Google.
- A pesar de contar con una comunidad activa, su dimensión es mucho menor que la que podemos encontrar en Google Analytics.
- No posee una forma de analizar o registrar algunas de las funcionalidades particulares de la plataforma C.U.B.A., solo logra registrar eventos generales de los usuarios.

### **Puntos Comunes de Matomo y Google Analytics**

- Análítica de datos en tiempo real.
- Dimensiones Personalizadas.
- Seguimiento de objetivos.
- Medición a través de páginas vistas y eventos.
- Análisis de búsquedas internas.
- Geolocalización de usuario.
- Acceso vía API.

### **Diferencias de Matomo y Google Analytics.**

Matomo se plantea como alternativa a Google Analytics, capaz de ofrecer un valor similar a sus usuarios, pero con una diferencia fundamental: la pertenencia de nuestra propia información. Mientras que Analytics es un sistema en la Nube, que no requiere ninguna instalación, ya que reside en los servidores de Google, con Matomo es necesario disponer de un servidor donde instalar el software. Sin embargo, la necesidad de poseer nuestra propia infraestructura, lejos de ser una desventaja, es el mayor valor que aporta Matomo.

En resumen, cuando usas Matomo tienes la garantía de que posees tus propios datos, pues estos residen en tus propios servidores. A través del API de analítica de Matomo se mostrarán los datos, de la manera que sea necesaria, pero nunca saldrán de nuestra propia infraestructura.(Arsys, 2018).

### 1.3. Tecnologías y Herramientas

Para la selección de las tecnologías se tuvo en cuenta que el servicio se iba a integrar a la arquitectura de la nueva versión de la Plataforma C.U.B.A, la cual está basada en un desarrollo orientado a servicios, por lo que se requería de un análisis conjunto con el cliente, para conocer las especificaciones técnicas de la misma, por lo que se seleccionan las siguientes tecnologías:

#### **Spring Boot:**

Spring Boot es un proyecto que permite crear aplicaciones web independientes que no tienen una dependencia en un servidor de aplicaciones o en un contenedor externo de servlets (clase en el lenguaje de programación Java, utilizada para ampliar las capacidades de un servidor). Las aplicaciones web creadas con Spring Boot incluyen una versión integrada de Tomcat o Jetty y producen un único archivo java ejecutable (JAR). Spring Boot también proporciona opciones de configuración para generar un archivo de aplicaciones web (WAR) que puede implementarse en un contenedor de servlets liviano o en un servidor de aplicaciones.(Villazón, 2018)

Una característica innovadora clave de Spring Boot es la detección automática de dependencias y configuraciones predeterminadas por convención que minimiza la cantidad de configuración requerida para construir e implementar una aplicación web. Cuando las configuraciones predeterminadas proporcionadas no son adecuadas para su uso, la anulación es posible ya sea proporcionando una implementación de una clase base o interfaz y registrándose con el contexto de la aplicación o mediante la especificación en un archivo de configuración (Lundy, 2015).

#### **Java:**

La principal característica de **Java** es la de ser un lenguaje compilado e interpretado. Todo programa en Java ha de compilarse y el código que se genera bytecodes es interpretado por una máquina virtual. De este modo se consigue la independencia de la máquina, el código compilado se ejecuta en máquinas virtuales que si son dependientes de la plataforma. Java es un lenguaje orientado a objetos de propósito general. Aunque Java comenzará a ser conocido como un lenguaje de programación de applets que se ejecutan en el entorno de un navegador web, se puede utilizar para construir cualquier tipo de proyecto.(Fernández, 2005).

**REST** responde a Transferencia de Estado Representacional(Representational State Transfer), se refiere a un método de transferencia de servicios web que permite a diferentes equipos acceder y manipular representaciones textuales de recursos web mediante un set uniforme y predefinido de operaciones sin estado.(Pryschepa Pryschepa, 2017)

REST es un estilo arquitectónico de servicios web el cual usa la URI para exponer la lógica del negocio. REST a menudo tiene muchos recursos y es una arquitectura orientada a recursos mientras que SOAP es un protocolo el cual usa interfaces de servicios para exponer la lógica del negocio.

### **API REST:**

Una API REST permite la utilización de servicios web disponibles a través de Internet utilizando el protocolo estándar HTTP; en otras palabras, una API REST es un servicio que al igual que el término inicial de una API provee de funciones que dan capacidad de usar un servicio web de terceros dentro de nuestro propio servicio o aplicación propia de un modo seguro.(Gonzalo Soto, 2018)

### **JSON:**

Según la definición que proporciona Microsoft en su sitio web oficial para desarrolladores MSDN, JavaScript Object Notation (JSON) es un formato de codificación eficaz que permite intercambios rápidos de cantidades pequeñas de datos entre los exploradores de cliente, tales como Internet Explorer, Google Chrome y servicios web, por lo cual se le considera un gran método de desarrollo web para la serialización/deserialización en la creación de aplicaciones.(Mora-Castillo, 2016)

Respecto a los protocolos **JSON y REST**, se muestra en (“JSON Based Decentralized SSO Security Architecture in E-Commerce,”)(Jun et al., 2008) como mediante los Servicios Web basados en SSO (Single Song On – Inicio de Sesión Único) se provee a los usuarios un fácil acceso a las aplicaciones y a los recursos de red, donde la parte de seguridad es tratada bajo **JSON** (JavaScript Object Notation) definido por los autores como un formato ligero para el intercambio de datos de fácil lectura y escritura para los seres humanos.(Chanchí et al., 2011).

JSON y REST ofrecen formas más simples de acceder a los recursos de red, dando la capacidad de añadir nuevos servicios web a los que ya ofrece alguna aplicación, de forma que permita un intercambio con un

formato ligero de datos de fácil lectura y escritura. Esto nos permitirá el uso de API REST para añadir los nuevos servicios de captura de eventos con el que se propone dar solución al problema planteado en este trabajo.

## **MySQL**

Es el segundo sistema de gestión de bases de datos relacionales de código abierto (RDBMS) más utilizado. El SQL significa Lenguaje de Consulta Estructurado. Es una opción popular de base de datos para su uso en aplicaciones web, y es un componente central de la pila de *software* de aplicaciones web LAMP de código abierto ampliamente utilizado LAMP significa Linux, Apache, MySQL, Perl / PHP / Python. Se usa a menudo para proyectos de código abierto que requieren un sistema de gestión de bases de datos con todas las funciones (Gawas et al., 2015).

Gran parte del atractivo de MySQL se origina en su relativa simplicidad y facilidad de uso, que es habilitada por un ecosistema de herramientas de código abierto como phpMyAdmin. En el rango medio, MySQL se puede escalar al implementarlo en un hardware más potente, como un servidor de múltiples procesadores con gigabytes (unidad de almacenamiento de información) de memoria (Katkar, 2015).

Siguiendo la infraestructura de la plataforma y conociendo que los servicios que pretendemos añadir se unirán a los que están, se utilizara MySQL al ser un requerimiento del cliente. MySQL al ser de código abierto y uno de los más utilizados a menudo por su relativa simplicidad y fácil comprensión que permite trabajar con un sistema de gestión de base datos de código abierto.

### **1.3.1. Herramientas Case**

#### **Visual Paradigm**

Visual Paradigm para UML 8.0 es una herramienta profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción y despliegue. Es multiplataforma, utiliza UML como lenguaje de modelado y cuenta con una versión libre para la comunidad, ayudando de una manera rápida a la construcción de aplicaciones de mayor calidad y a un menor costo. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. Esta herramienta también proporciona una mejor interfaz gráfica de usuario y una mayor base de datos de esquema de apoyo, permite generar la documentación del sistema en los formatos PDF,

HTML y el formato de documentos de Microsoft Word y permite importar proyectos de otras herramientas de modelado como Rational Rose, Erwin y Microsoft Visio. Soporta la revisión ortográfica, brindando sugerencias para los idiomas: inglés, español, francés, alemán y portugués (Visual Paradigm, 2013).

Esta herramienta permite una mejor comprensión al poder modelar diagramas y documentación del desarrollo del software, con la fácil comprensión que demuestra la interface gráfica de usuario y una base de datos de esquemas de apoyo, esto demuestra la gran variedad de herramientas que nos ofrece para un análisis y diseño más eficaz de nuestro proyecto, con el añadido de ser una herramienta multiplataforma con UML con lenguaje modelador y contar con una versión gratuita para la comunidad.

#### **1.4. - Despliegue de la Plataforma C.U.B.A**

La Plataforma C.U.B.A se encuentra desplegada en 19 servidores, presentando como elemento significativo la redundancia en cada una de sus capas, entre las principales características que presenta actualmente esta infraestructura se encuentran:

- Balanceo de carga en el acceso de la interfaz del buscador, a la base de datos donde se almacenan las preferencias del sitio y al sistema de indexación, garantizando la alta disponibilidad en este sentido.
- Redundancia en los nodos de la interfaz del buscador, en la base de datos, en el sistema de indexación y en los nodos destinados al rastreo.
- Utilización de Centos<sup>8</sup> en su versión para servidores como sistema operativo de código abierto en cada uno de sus nodos, así como todas las tecnologías presentadas en esta solución.

Dentro de sus funcionalidades se encuentran:

- La búsqueda de imágenes
- Videos.
- Documentos en su mayoría de investigaciones cubanas.
- Noticias del acontecer nacional e internacional.

---

<sup>8</sup> Es un sistema operativo de código abierto, basado en la distribución Red Hat Enterprise Linux, operándose de manera similar, y cuyo objetivo es ofrecer al usuario un software de "clase empresarial" gratuito. Se define como robusto, estable y fácil de instalar y utilizar. Desde la versión 5, cada lanzamiento recibe soporte durante diez años, por lo que la actual versión 7 recibirá actualizaciones de seguridad hasta el 30 de junio de 2024.

- Presenta un microservicio climático el cual muestra el clima actual en la Capital de Cuba.
- Se puede encontrar en su directorio numerosos Dominios asociados a Educación (Primaria, Media, Superior), Ministerios de la República, Grupos Empresariales, El Turismo, Organizaciones políticas y de masas, Electrónicas, entre otras.
- Ofrece servicios de conversor de monedas, servicio para descargar fondos y garabatos de la plataforma C.U.B.A, Servicio de Mapa del País, Servicio de distancia, servicios de resultados en la búsqueda en el mapa, Servicio de Diccionario.

## **1.5. Metodología de desarrollo de software**

El desarrollo de software estuvo guiado por la metodología AUP-UCI la cual es una variante de la metodología Agile Unified Process (AUP) la cual es una versión simplificada del Proceso Unificado de Rational (RUP).

En la documentación Metodología de desarrollo para la Actividad Productiva de la Universidad de las Ciencias Informáticas versión 1.2 Elaborada Tamara Rodríguez Sánchez habla de cómo Agile Unified Process (AUP) describe de una manera simple y fácil de entender la forma de desarrollar aplicaciones de software de negocio usando técnicas ágiles y conceptos que aún se mantienen válidos en RUP.

El AUP-UCI aplica técnicas ágiles incluyendo:

- Desarrollo Dirigido por Pruebas (test driven development -TDD en inglés)
- Modelado ágil
- Gestión de Cambios ágil
- Refactorización de Base de Datos para mejorar la productividad.

Al igual que en RUP, en AUP se establecen cuatro fases que transcurren de manera consecutiva.

1. Inicio: El objetivo de esta fase es obtener una comprensión común cliente-equipo de desarrollo del alcance del nuevo sistema y definir una o varias arquitecturas candidatas para el mismo.

2. Elaboración: El objetivo es que el equipo de desarrollo profundice en la comprensión de los requisitos del sistema y en validar la arquitectura.

3. Construcción: Durante la fase de construcción el sistema es desarrollado y probado al completo en el ambiente de desarrollo.

4. Transición: El sistema se lleva a los entornos de preproducción donde se somete a pruebas de validación y aceptación y finalmente se despliega en los sistemas de producción.

Para la variación orientada AUP-UCI se tuvieron en cuenta de las 4 fases que propone AUP (Inicio, Elaboración, Construcción, Transición) se decide para el ciclo de vida de los proyectos de la UCI mantener la fase de Inicio, pero modificando el objetivo de la misma, se unifican las restantes 3 fases de AUP en una sola, a la que llamaremos Ejecución y se agrega la fase de Cierre.

A partir del Modelado de negocio surgen cuatros escenarios, al analizar cada uno de estos, el escenario que representa con más fidelidad nuestro negocio es:

**El escenario 2:** Aplica a los proyectos que hayan evaluado el negocio a informatizar y como resultado obtengan que no es necesario incluir las responsabilidades de las personas que ejecutan las actividades, al ser nuestro proyecto prácticamente automatizado el cual recibirá y guardara la información de las interacciones del usuario con la plataforma, de esta forma se modelaría exclusivamente los conceptos fundamentales del negocio.

**Escenario 2:** Proyectos que modelen el negocio con modelo conceptual(MC) solo pueden modelar el sistema con Casos de usos del sistema(CUS).

(MC)=(CUS)

## **1.6. Conclusiones del capítulo**

- El análisis realizado de cada uno de los componentes de la Plataforma C.U.B.A, entre los cuales se encuentran su despliegue y funcionalidades permite una mayor comprensión del sistema en el cual trabajaremos.
- El estudio de los conceptos y procesos relacionados con la Analítica Web permitió determinar los componentes principales para la solución de la problemática y los elementos esenciales para la descripción de una solución.
- El estudio de las tecnologías y sistemas homólogos que permitió entender múltiples problemas y soluciones que se pueden presentar con este tipo de tecnologías y sistemas.

## **Capítulo 2: Análisis y diseño de la solución para facilitar el registro de eventos de la plataforma C.U.B.A. por parte de los usuarios.**

En este capítulo se realizará una descripción detallada de los procesos que intervienen en la propuesta de solución, las reglas que rigen el negocio, se demuestra cómo se relacionan los componentes y tecnología que los componen, así como una descripción detallada de las tecnologías, métricas y funcionalidades definidas.

### **2.1 Propuesta de la solución**

Para la solución del servicio de seguimiento del usuario en la plataforma C.U.B.A. tenemos que conocer el funcionamiento actual del negocio definiendo así los procesos que ocurren en él y el flujo que llevan las acciones de los usuarios al interactuar con la plataforma. Con estos datos se conocería cuáles van a ser las métricas para el registro de los eventos y obtener así un microservicio como solución backend el cual se integrará la arquitectura existente en la plataforma C.U.B.A.

### **2.2 Modelo Conceptual**

**Usuario:** Se conecta desde una PC de forma anónima o identifica en la Plataforma C.U.B.A. y solicita algunos o varios servicios que ofrece la plataforma tales como Búsqueda, Enviar Comentarios, ver y descargar Garabatos, realizar Conversión de Moneda, elegir una Publicidad de su interés y realizar Búsquedas de información que necesite.

**Enviar Comentarios:** El usuario escribirá su comentario y enviará y este se guardará en la Base de Datos.

**Garabatos:** Son un Conjunto de imágenes que promociona la plataforma y el usuario puede observar y descargar si así lo desea.

**Publicidad:** La plataforma tiene una sección de Publicidad donde da a conocer informaciones de otras páginas cubanas a las que el usuario puede acceder desde la dirección url que estas poseen.

**Búsqueda:** El usuario puede realizar una búsqueda sobre algún contenido que desee, el buscador obtendrá resultados el cual el seleccionará el que considere lo más cercano a lo que desea y podrá seleccionar el formato que más le guste, ya sea imagen, documento, noticia, o dirección web.

**Conversión de Moneda:** Este servicio cuenta con la posibilidad de que el usuario introduzca la moneda a convertir y selecciona la moneda de cambio.

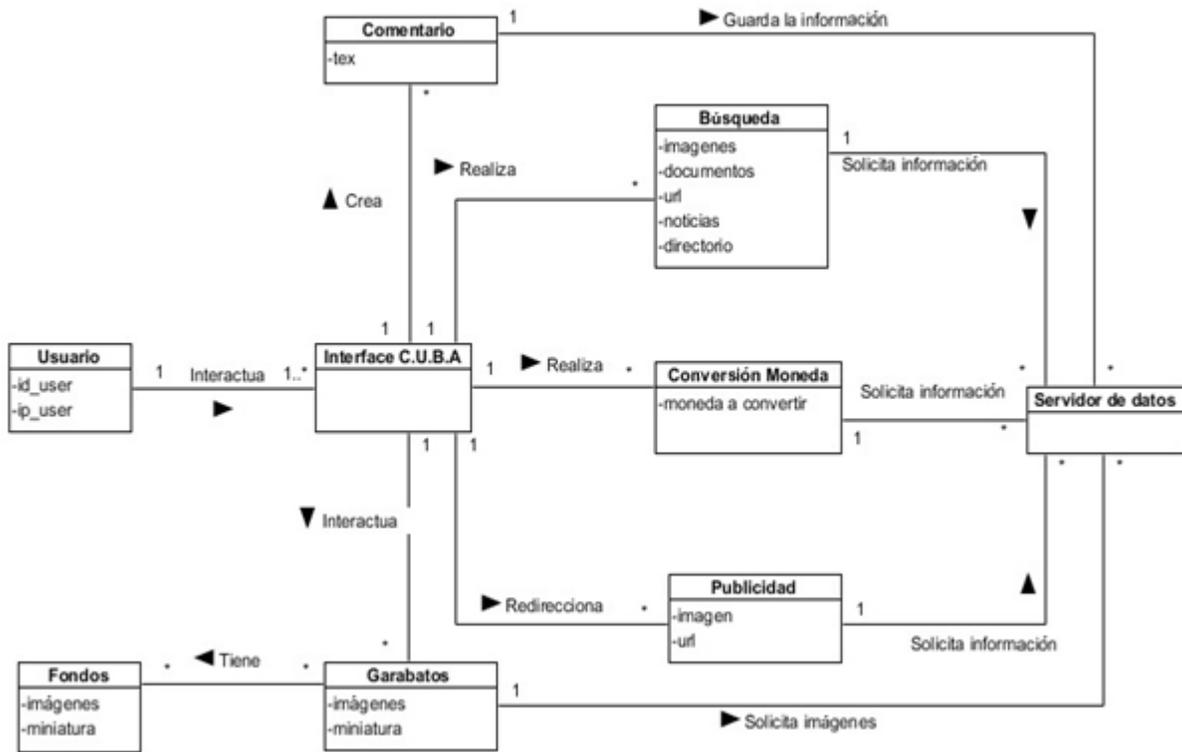


Figura 1. Modelo Conceptual (Interacción del usuario con la interface de la plataforma C.U.B.A). Fuente: elaboración propia.

## 2.3 Requerimientos de Software

Los requerimientos de software definen una condición o capacidad que necesita el usuario para resolver un problema o lograr un objetivo. Existen dos tipos de requerimientos de software, los requerimientos

funcionales, que son las capacidades o condiciones que el sistema debe cumplir, y los requerimientos no funcionales, que son las propiedades o cualidades que el producto de software debe tener. A continuación, se presenta el listado de requerimientos funcionales y no funcionales.

### **2.3.1 Requisitos Funcionales**

#### **Gestionar eventos de búsquedas:**

**RF-1** Registrar evento de la búsqueda en la web.

**RF-2** Registrar evento de la búsqueda en imágenes.

**RF-3** Registrar evento de la búsqueda en noticias.

**RF-4** Registrar evento de la búsqueda en academia.

**RF-5** Registrar evento de la búsqueda en directorio.

**RF-6** Filtrar los eventos de búsquedas.

**RF-7** Mostrar evento de búsqueda.

**RF-8** Eliminar eventos de búsqueda.

#### **Gestionar eventos de resultado de búsqueda:**

**RF-9** Registrar evento de selección del resultado de búsqueda.

**RF-10** Filtrar los eventos de selección del resultado de búsqueda.

**RF-11** Mostrar evento de selección del resultado de búsqueda.

**RF-12** Eliminar eventos de selección del resultado de búsqueda.

#### **Gestionar eventos de publicidad:**

**RF-13** Registrar evento de acceso a la publicidad.

**RF-14** Filtrar los eventos de acceso a la publicidad.

**RF-15** Mostrar evento de acceso a la publicidad.

**RF-16** Eliminar eventos de acceso a la publicidad.

**Gestionar eventos de garabatos:**

**RF-17** Registrar evento de descarga de garabatos.

**RF-18** Registrar evento de visualización de garabatos.

**RF-19** Filtrar los eventos asociados a garabatos.

**RF-20** Mostrar evento asociado a garabatos.

**RF-21** Eliminar eventos asociados a garabatos.

**Gestionar eventos de fondo de pantalla:**

**RF-22** Registrar evento de descarga de fondo de pantalla.

**RF-23** Registrar evento de visualización de fondo de pantalla.

**RF-24** Filtrar los eventos asociados al fondo de pantalla.

**RF-25** Mostrar evento asociado al fondo de pantalla.

**RF-26** Eliminar eventos asociados al fondo de pantalla.

**Gestión de eventos conversión de monedas:**

**RF-27** Registrar evento de conversión de monedas.

**RF-28** Filtrar los eventos de conversión de monedas.

**RF-29** Mostrar evento de conversión de monedas.

**RF-30** Eliminar eventos de conversión de monedas.

**Gestionar eventos de trazabilidad:**

**RF-31** Registrar evento de trazabilidad.

- Los eventos de trazabilidad se registrarán cuando se realice cualquier tipo de evento.

**RF-32** Filtrar evento de trazabilidad.

### **2.3.3 Requisitos no Funcionales**

#### **RNF-1 Seguridad**

- Seguridad de la base de datos: la base de datos deberá estar fraccionada en esquemas que permitan un mejor uso de la información y la división de forma lógica de las funcionalidades del sistema, trayendo consigo además la protección de la información al ocurrir un incidente sobre una parte de la base de datos.
- Registros sistemáticos de incidencias: el sistema debe ser capaz de registrar el accionar del usuario sobre el mismo, así como permitir auditorias y exámenes de las trazas tanto en tiempo real como en históricos. Se precisa un monitor de incidencia para la visualización y tratamiento de las mismas.
- Alta protección de los datos: al estar trabajando con información sensible, se hace necesario una alta protección de los datos a nivel de aplicación y de tráfico por la red, para tal fin se ha definido además la seguridad en varios niveles dentro de la aplicación (nivel de acceso a datos y nivel de base de datos)
- Autorización de puntos finales: Se debe proporcionar un acceso a los puntos finales separado y exclusivo.

#### **RNF-2 Rendimiento**

- Escalabilidad: el sistema debe ser capaz de mantener un rendimiento y una estabilidad adecuados al gestionar amplios volúmenes de datos, así como de permitir la incorporación de nuevas funcionalidades adaptándose de manera natural a los procesos de gestión que se requieran.

### **RNF-3 Hardware**

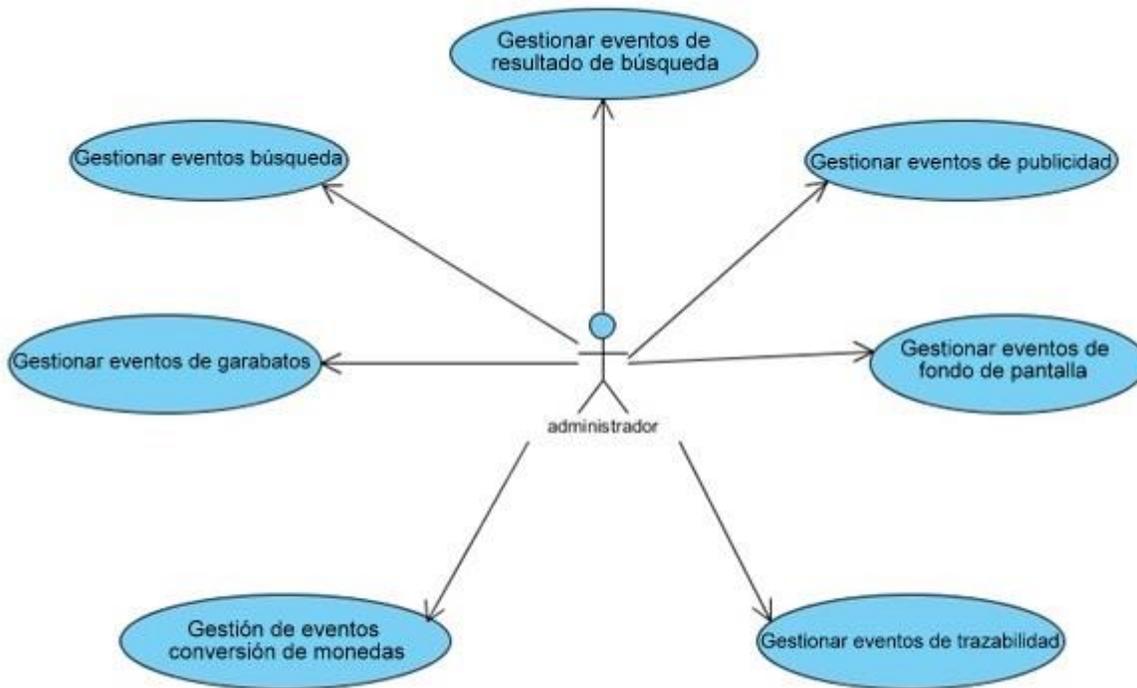
- Para despliegue: CPU Dual Core 2.0 GHZ o superior, memoria RAM de 4 GB (recomendado 6 GB), 250 GB HDD.

### **RNF-4 Usabilidad**

- Debe existir información detallada de los puntos finales con el método HTTP, URI del punto final y descripción de la misma. Además, deben estar correctamente descritos sus entradas y salidas, así como los códigos de errores y descripciones de los mismos.

## **2.4 Descripciones abreviadas de los casos de uso del sistema**

- La descripción de los casos de uso del sistema facilita la comprensión del funcionamiento del sistema desde la perspectiva del usuario. A continuación, se muestra una descripción donde los 32 casos de uso del sistema presentados se recogen abreviados en los gestionar de cada evento de las funcionalidades de la plataforma.



### Gestionar eventos de búsquedas:

#### CUS 1 Registrar evento de la búsqueda en la web

<b>Objetivo</b>	Proveer un punto final que permita registrar un evento cuando se realice una búsqueda en la web en la Plataforma C.U.B.A.
<b>Actores</b>	-
<b>Resumen</b>	Registrar evento de la búsqueda en la web
<b>Complejidad</b>	Alta
<b>Prioridad</b>	¿?
<b>Precondiciones</b>	-
<b>Postcondiciones</b>	Registrar la acción de trazabilidad del evento registrado.

<b>Flujo de eventos</b>		
<b>Flujo básico &lt; Registrar evento de la búsqueda en la web&gt;</b>		
<b>Sistema</b>		
1.	Obtiene los datos a través de una petición POST que se realiza al endpoint <i>/event/create/search/web</i> .	
2.	Se valida la estructura de la entidad del dominio que se obtiene en la petición en caso de ser correcta se envían los datos a la capa de servicios.	
3.	Se validan los datos de la entidad recibida en caso de estar correctos se envían los datos al repositorios.	
4.	Se obtiene la dirección IP de la entidad y a partir de la misma se completa la información de geo localización de la fuente donde se realizó la búsqueda en la web a partir de la base de datos GeoLite.mmdb.	
5.	Una vez completado, se registra el evento, se crea una acción de trazabilidad del evento registrado y se envía un código de estado de satisfacción (201).	
<b>Relaciones</b>	<b>CU Incluidos</b>	Ninguno
	<b>CU Extendidos</b>	Ninguno
<b>Requisitos funcionales</b>	<b>no</b>	Seguridad y Usabilidad
<b>Asuntos pendientes</b>		Ninguno
<b>Flujos alternos</b>		
<b>Nº 2 &lt;Caso de Error&gt;</b>		

<b>Sistema</b>	
2.1	Se valida la estructura de la entidad del dominio que se obtiene en la petición en caso de ser incorrecta
2.2	Se lanza un error 400

### CUS 2 Registrar evento de la búsqueda en imágenes

<b>Objetivo</b>	Proveer un punto final que permita registrar un evento cuando se realice una búsqueda en imágenes en la Plataforma C.U.B.A.
<b>Actores</b>	-
<b>Resumen</b>	Registrar evento de la búsqueda en imágenes
<b>Complejidad</b>	Alta
<b>Prioridad</b>	¿?
<b>Precondiciones</b>	-
<b>Postcondiciones</b>	Registrar la acción de trazabilidad del evento registrado.
<b>Flujo de eventos</b>	
<b>Flujo básico &lt; Registrar evento de la búsqueda en imágenes &gt;</b>	
<b>Sistema</b>	
1.	Obtiene los datos a través de una petición POST que se realiza al endpoint <i>/event/create/search/web</i> .
2.	Se valida la estructura de la entidad del dominio que se obtiene en la petición en caso de ser correcta se envían los datos a la capa de servicios.

3.	Se validan los datos de la entidad recibida en caso de estar correctos se envían los datos al repositorios.	
4.	Se obtiene la dirección IP de la entidad y a partir de la misma se completa la información de geo localización de la fuente donde se realizó la búsqueda en la web a partir de la base de datos GeoLite.mmdb.	
5.	Una vez completado, se registra el evento, se crea una acción de trazabilidad del evento registrado y se envía un código de estado de satisfacción (201).	
<b>Relaciones</b>	<b>CU Incluidos</b>	Ninguno
	<b>CU Extendidos</b>	Ninguno
<b>Requisitos funcionales</b>	<b>no</b>	Seguridad y Usabilidad
<b>Asuntos pendientes</b>	Ninguno	
<b>Flujos alternos</b>		
<b>Nº 2 &lt;Ocurre Error en la validación &gt;</b>		
<b>Sistema</b>		
2.1	Se valida la estructura de la entidad del dominio que se obtiene en la petición en caso de ser incorrecta.	
2.2	Se lanza un error 400.	

### CUS 3 Registrar evento de la búsqueda en noticias.

<b>Objetivo</b>	Proveer un punto final que permita registrar un evento cuando se realice una búsqueda en noticias en la Plataforma C.U.B.A.
<b>Actores</b>	-
<b>Resumen</b>	Registrar evento de la búsqueda en noticias
<b>Complejidad</b>	Alta
<b>Prioridad</b>	Crítica
<b>Precondiciones</b>	-
<b>Postcondiciones</b>	Registrar la acción de trazabilidad del evento registrado.
<b>Flujo de eventos</b>	
<b>Flujo básico &lt; Registrar evento de la búsqueda en noticias &gt;</b>	
	<b>Sistema</b>
1.	Obtiene los datos a través de una petición POST que se realiza al endpoint <i>/event/create/search/web</i> .
2.	Se valida la estructura de la entidad del dominio que se obtiene en la petición en caso de ser correcta se envían los datos a la capa de servicios.
3.	Se validan los datos de la entidad recibida en caso de estar correctos se envían los datos al repositorios.

4.	Se obtiene la dirección IP de la entidad y a partir de la misma se completa la información de geo localización de la fuente donde se realizó la búsqueda en la web a partir de la base de datos GeoLite.mmdb.	
5.	Una vez completado, se registra el evento, se crea una acción de trazabilidad del evento registrado y se envía un código de estado de satisfacción (201).	
<b>Relaciones</b>	<b>CU Incluidos</b>	Ninguno
	<b>CU Extendidos</b>	Ninguno
<b>Requisitos funcionales</b>	<b>no</b>	Seguridad y Usabilidad
<b>Asuntos pendientes</b>		Ninguno
<b>Flujos alternos</b>		
<b>Nº 2 &lt;Ocurre Error en la validación &gt;</b>		
	<b>Sistema</b>	
2.1	Se valida la estructura de la entidad del dominio que se obtiene en la petición en caso de ser incorrecta.	
2.2	Se lanza un error 400.	

**CUS 4** Registrar evento de la búsqueda en academia.

<b>Objetivo</b>	Proveer un punto final que permita registrar un evento cuando se realice una búsqueda en academia en la Plataforma C.U.B.A.
-----------------	---

<b>Actores</b>	-
<b>Resumen</b>	Registrar evento de la búsqueda en academia
<b>Complejidad</b>	Alta
<b>Prioridad</b>	Crítica
<b>Precondiciones</b>	-
<b>Postcondiciones</b>	Registrar la acción de trazabilidad del evento registrado.
<b>Flujo de eventos</b>	
<b>Flujo básico &lt; Registrar evento de la búsqueda en academia &gt;</b>	
	<b>Sistema</b>
1.	Obtiene los datos a través de una petición POST que se realiza al endpoint <i>/event/create/search/web</i> .
2.	Se valida la estructura de la entidad del dominio que se obtiene en la petición en caso de ser correcta se envían los datos a la capa de servicios.
3.	Se validan los datos de la entidad recibida en caso de estar correctos se envían los datos al repositorios.
4.	Se obtiene la dirección IP de la entidad y a partir de la misma se completa la información de geo localización de la fuente donde se realizó la búsqueda en la web a partir de la base de datos GeoLite.mmdb.
5.	Una vez completado, se registra el evento, se crea una acción de trazabilidad del evento registrado y se envía un código de estado de satisfacción (201).
<b>Relaciones</b>	<b>CU Incluidos</b> Ninguno

	<b>CU Extendidos</b>	Ninguno
<b>Requisitos funcionales</b>	<b>no</b>	Seguridad y Usabilidad
<b>Asuntos pendientes</b>		Ninguno
<b>Flujos alternos</b>		
<b>Nº 2 &lt;Ocurre Error en la validación &gt;</b>		
	<b>Sistema</b>	
2.1	Se valida la estructura de la entidad del dominio que se obtiene en la petición en caso de ser incorrecta.	
2.2	Se lanza un error 400.	

**CUS 5** Registrar evento de la búsqueda en directorio.

<b>Objetivo</b>	Proveer un punto final que permita registrar un evento cuando se realice una búsqueda en directorio en la Plataforma C.U.B.A.
<b>Actores</b>	-
<b>Resumen</b>	Registrar evento de la búsqueda en directorio
<b>Complejidad</b>	Alta
<b>Prioridad</b>	Crítica
<b>Precondiciones</b>	-

<b>Postcondiciones</b>	Registrar la acción de trazabilidad del evento registrado.	
<b>Flujo de eventos</b>		
<b>Flujo básico &lt; Registrar evento de la búsqueda en directorio &gt;</b>		
	<b>Sistema</b>	
1.	Obtiene los datos a través de una petición POST que se realiza al endpoint <i>/event/create/search/web</i> .	
2.	Se valida la estructura de la entidad del dominio que se obtiene en la petición en caso de ser correcta se envían los datos a la capa de servicios.	
3.	Se validan los datos de la entidad recibida en caso de estar correctos se envían los datos al repositorios.	
4.	Se obtiene la dirección IP de la entidad y a partir de la misma se completa la información de geo localización de la fuente donde se realizó la búsqueda en la web a partir de la base de datos GeoLite.mmdb.	
5.	Una vez completado, se registra el evento, se crea una acción de trazabilidad del evento creado y se envía un código de estado de satisfacción (201).	
<b>Relaciones</b>	<b>CU Incluidos</b>	Ninguno
	<b>CU Extendidos</b>	Ninguno
<b>Requisitos funcionales</b>	<b>no</b>	Seguridad y Usabilidad
<b>Asuntos pendientes</b>		Ninguno
<b>Flujos alternos</b>		

<b>Nº 2 &lt;Ocurre Error en la validación &gt;</b>	
<b>Sistema</b>	
2.1	Se valida la estructura de la entidad del dominio que se obtiene en la petición en caso de ser incorrecta.
2.2	Se lanza un error 400.

**CUS 6** Filtrar los eventos de búsquedas.

<b>Objetivo</b>	Proveer un punto final que permita Filtrar los eventos de búsquedas en la Plataforma C.U.B.A y devolver un listado con los eventos filtrados.
<b>Actores</b>	-
<b>Resumen</b>	Filtrar los eventos de búsquedas
<b>Complejidad</b>	Alta
<b>Prioridad</b>	Secundaria
<b>Precondiciones</b>	-
<b>Postcondiciones</b>	Registrar la acción de trazabilidad del evento filtrado.
<b>Flujo de eventos</b>	
<b>Flujo básico &lt; Filtrar los eventos de búsquedas &gt;</b>	
<b>Sistema</b>	
1.	Se obtiene los datos del filtrado cuando se realiza una petición GET al endpoint <i>/event/filter/search</i> .

2.	Se validan que los datos estén en un formato correcto, en el caso de estar correctos se pasa a la capa del repositorio.	
3.	Una vez completado, se construye el listado con los eventos de búsquedas con coincidieron con los datos de filtrado y se devuelve, se crea una acción de trazabilidad del evento filtrado y se envía un código de estado de satisfacción (200).	
<b>Relaciones</b>	<b>CU Incluidos</b>	Ninguno
	<b>CU Extendidos</b>	Ninguno
<b>Requisitos funcionales</b>	<b>no</b>	Seguridad y Usabilidad
<b>Asuntos pendientes</b>		Ninguno
<b>Flujos alternos</b>		
<b>Nº 2 &lt;Ocurre un error en la validación de los datos&gt;</b>		
	<b>Sistema</b>	
2.1	Si los datos del filtro no se encuentran en los formatos correctos se envía un código de estado de error de petición (400).	

**CUS 7** Mostrar evento de búsqueda.

<b>Objetivo</b>	Proveer un punto final que permita mostrar un evento de una búsqueda en la Plataforma C.U.B.A.
<b>Actores</b>	-

<b>Resumen</b>	Mostrar evento de búsqueda	
<b>Complejidad</b>	Alta	
<b>Prioridad</b>	Secundaria	
<b>Precondiciones</b>	-	
<b>Postcondiciones</b>	Registrar la acción del evento creado.	
<b>Flujo de eventos</b>		
<b>Flujo básico &lt; Mostrar evento de búsqueda &gt;</b>		
	<b>Sistema</b>	
1.	Obtiene el identificador del evento de búsqueda a través de una petición <i>GET</i> que se realiza al endpoint <i>/event/show/search/{id}</i> .	
2.	Se valida que exista un evento con ese identificador en caso de existir se devuelve el evento de búsqueda, se crea la acción de trazabilidad asociado al mostrar evento y se envía un código de estado 200.	
<b>Relaciones</b>	<b>CU Incluidos</b>	Ninguno
	<b>CU Extendidos</b>	Ninguno
<b>Requisitos funcionales</b>	<b>no</b>	Seguridad y Usabilidad
<b>Asuntos pendientes</b>	Ninguno	
<b>Flujos alternos</b>		
<b>Nº 2 &lt;En Caso de no existir el identificador &gt;</b>		

Sistema	
2.1	Si a la hora de validar no existe el identificador se devuelve un código de estado de recurso no encontrado (404)
2.2	

### CUS 8 Eliminar eventos de búsqueda.

<b>Objetivo</b>	Proveer un punto final que permita Eliminar un eventos de búsqueda en la Plataforma C.U.B.A.
<b>Actores</b>	-
<b>Resumen</b>	Eliminar evento de la búsqueda
<b>Complejidad</b>	Alta
<b>Prioridad</b>	Secundaria
<b>Precondiciones</b>	-
<b>Postcondiciones</b>	Registrar la acción de trazabilidad del evento eliminado.
<b>Flujo de eventos</b>	
<b>Flujo básico &lt; Eliminar eventos de búsqueda &gt;</b>	
Sistema	
1.	Se obtiene el o los identificador/es a través de una petición <i>DELETE</i> que se realiza al endpoint  <i>/event/detele/search/{id}</i> .

2.	Se obtienen los eventos de búsquedas asociados a los identificadores y en caso de existir se procede a su eliminación y se devuelve un código de estado sin contenido (204)	
<b>Relaciones</b>	<b>CU Incluidos</b>	Ninguno
	<b>CU Extendidos</b>	Ninguno
<b>Requisitos funcionales</b>	<b>no</b>	Seguridad y Usabilidad
<b>Asuntos pendientes</b>		Ninguno
<b>Flujos alternos</b>		
<b>Nº 2 &lt;Ocurre un error en validación &gt;</b>		
<b>Sistema</b>		

**Gestionar eventos de búsquedas:**

**CUS 9** Registrar evento de selección del resultado de búsqueda.

<b>Objetivo</b>	Proveer un punto final que permita registrar un evento cuando se realice una selección del resultado de búsqueda en la Plataforma C.U.B.A.
<b>Actores</b>	-
<b>Resumen</b>	Registrar evento de selección del resultado de búsqueda
<b>Complejidad</b>	Alta
<b>Prioridad</b>	Crítica
<b>Precondiciones</b>	-

<b>Postcondiciones</b>	Registrar la acción de trazabilidad del evento registrado.	
<b>Flujo de eventos</b>		
<b>Flujo básico &lt; Registrar evento de selección del resultado de búsqueda &gt;</b>		
	<b>Sistema</b>	
1.	Obtiene los datos a través de una petición POST que se realiza al endpoint <i>/event/create/search/web</i> .	
2.	Se valida la estructura de la entidad del dominio que se obtiene en la petición en caso de ser correcta se envían los datos a la capa de servicios.	
3.	Se validan los datos de la entidad recibida en caso de estar correctos se envían los datos al repositorios.	
4.	Se obtiene la dirección IP de la entidad y a partir de la misma se completa la información de geo localización de la fuente donde se realizó la búsqueda en la web a partir de la base de datos GeoLite.mmdb.	
5.	Una vez completado, se registra el evento, se crea una acción de trazabilidad del evento creado y se envía un código de estado de satisfacción (201).	
<b>Relaciones</b>	<b>CU Incluidos</b>	Ninguno
	<b>CU Extendidos</b>	Ninguno
<b>Requisitos funcionales</b>	<b>no</b>	Seguridad y Usabilidad
<b>Asuntos pendientes</b>		Ninguno
<b>Flujos alternos</b>		

<b>Nº 2 &lt;Ocurre Error en la validación &gt;</b>	
<b>Sistema</b>	
2.1	Se valida la estructura de la entidad del dominio que se obtiene en la petición en caso de ser incorrecta.
2.2	Se lanza un error 400.

**CUS 10** Filtrar los eventos de selección del resultado de búsqueda.

<b>Objetivo</b>	Proveer un punto final que permita Filtrar eventos de selección del resultado de búsqueda en la Plataforma C.U.B.A.
<b>Actores</b>	-
<b>Resumen</b>	Filtrar evento de selección del resultado de búsqueda
<b>Complejidad</b>	Alta
<b>Prioridad</b>	Secundaria
<b>Precondiciones</b>	-
<b>Postcondiciones</b>	Registrar la acción de trazabilidad del evento filtrado.
<b>Flujo de eventos</b>	
<b>Flujo básico &lt; Filtrar los eventos de selección del resultado de búsqueda &gt;</b>	
<b>Sistema</b>	
4.	Se obtiene los datos del filtrado cuando se realiza una petición GET al endpoint <i>/event/filter/search</i> .

5.	Se validan que los datos estén en un formato correcto, en el caso de estar correctos se pasa a la capa del repositorio.	
6.	Una vez completado, se construye el listado con los eventos de búsquedas con coincidieron con los datos de filtrado y se devuelve, se crea una acción de trazabilidad del evento filtrado y se envía un código de estado de satisfacción (200).	
<b>Relaciones</b>	<b>CU Incluidos</b>	Ninguno
	<b>CU Extendidos</b>	Ninguno
<b>Requisitos funcionales</b>	<b>no</b>	Seguridad y Usabilidad
<b>Asuntos pendientes</b>		Ninguno
<b>Flujos alternos</b>		
<b>Nº 2 &lt;El Administrador desea eliminar el Evento&gt;</b>		
	<b>Sistema</b>	
2.1		
2.2	Se Muestra Mensaje "Evento Eliminado"	

**CUS 11** Mostrar evento de selección del resultado de búsqueda.

<b>Objetivo</b>	Proveer un punto final que permita registrar un evento cuando se realice una selección del resultado de búsqueda en la Plataforma C.U.B.A.
<b>Actores</b>	-

<b>Resumen</b>	Registrar evento de selección del resultado de búsqueda	
<b>Complejidad</b>	Alta	
<b>Prioridad</b>	Secundaria	
<b>Precondiciones</b>	-	
<b>Postcondiciones</b>	Registrar la acción del evento creado.	
<b>Flujo de eventos</b>		
<b>Flujo básico &lt; Mostrar evento de selección del resultado de búsqueda &gt;</b>		
	<b>Sistema</b>	
3.	Obtiene los datos a través de una petición <i>GET</i> que se realiza al endpoint <i>/event/show/search/{id}</i> .	
4.	Se valida que exista un evento con ese identificador en caso ser correcto se envía a la capa de repositorio directamente.	
5.	Se obtiene una lista con los eventos con ese identificador y devuelve una lista con los eventos con códigos de satisfacción 201	
6.	Una vez completado, se obtiene el evento, se crea una acción de trazabilidad de obtención de evento con ese identificador y se envía un código de estado de satisfacción (201).	
<b>Relaciones</b>	<b>CU Incluidos</b>	Ninguno
	<b>CU Extendidos</b>	Ninguno
<b>Requisitos funcionales</b>	<b>no</b>	Seguridad y Usabilidad

<b>Asuntos pendientes</b>	Ninguno
<b>Flujos alternos</b>	
<b>Nº 2 &lt;Error a la hora de validar&gt;</b>	
	<b>Sistema</b>
2.1	Si no existe un evento con ese identificador cuando se está validando
2.2	Se devuelve un código de estado 404

**CUS 12** Eliminar eventos de selección del resultado de búsqueda.

<b>Objetivo</b>	Proveer un punto final que permita Eliminar un evento de selección del resultado de búsqueda en la Plataforma C.U.B.A.
<b>Actores</b>	-
<b>Resumen</b>	Eliminar evento de selección del resultado de búsqueda
<b>Complejidad</b>	Alta
<b>Prioridad</b>	Secundaria
<b>Precondiciones</b>	-
<b>Postcondiciones</b>	Registrar la acción del evento creado.
<b>Flujo de eventos</b>	
<b>Flujo básico &lt; Eliminar eventos de selección del resultado de búsqueda &gt;</b>	
	<b>Sistema</b>

3.	Se obtiene el o los identificador/es a través de una petición <i>DELETE</i> que se realiza al endpoint  <i>/event/detele/search/{id}</i> .	
4.	Se valida de que exista eventos con ese identificador en caso, en caso de existir se pasa al repositorio.	
5.	Se elimina de forma satisfactoria.	
6.	Una vez completado, se crea un evento de trazabilidad de evento eliminado y no se envía un código de estado. (No contex)	
<b>Relaciones</b>	<b>CU Incluidos</b>	Ninguno
	<b>CU Extendidos</b>	Ninguno
<b>Requisitos funcionales</b>	<b>no</b>	Seguridad y Usabilidad
<b>Asuntos pendientes</b>		Ninguno
<b>Flujos alternos</b>		
<b>Nº 2 &lt;Ocurre un error en la validación &gt;</b>		
	<b>Sistema</b>	
2.1	Si no existe un evento con ese identificador.	
2.2	Se devuelve un código de estado 404	

## Gestionar eventos de publicidad:

**CUS 13** Registrar evento de acceso a la publicidad.

<b>Objetivo</b>	Proveer un punto final que permita registrar un evento cuando se realice un acceso a la publicidad en la Plataforma C.U.B.A.
<b>Actores</b>	-
<b>Resumen</b>	Registrar evento de acceso a la publicidad
<b>Complejidad</b>	Alta
<b>Prioridad</b>	Crítica
<b>Precondiciones</b>	-
<b>Postcondiciones</b>	Registrar la acción del evento creado.
<b>Flujo de eventos</b>	
<b>Flujo básico &lt; Registrar evento de selección del resultado de búsqueda &gt;</b>	
	<b>Sistema</b>
6.	Obtiene los datos a través de una petición POST que se realiza al endpoint <i>/event/create/search/web</i> .
7.	Se valida la estructura de la entidad del dominio que se obtiene en la petición en caso de ser correcta se envían los datos a la capa de servicios.
8.	Se validan los datos de la entidad recibida en caso de estar correctos se envían los datos al repositorios.
9.	Se obtiene la dirección IP de la entidad y a partir de la misma se completa la información de geo localización de la fuente donde se realizó la búsqueda en la web a partir de la base de datos GeoLite.mmdb.

10.	Una vez completado, se registra el evento, se crea una acción de trazabilidad del evento creado y se envía un código de estado de satisfacción (201).	
<b>Relaciones</b>	<b>CU Incluidos</b>	Ninguno
	<b>CU Extendidos</b>	Ninguno
<b>Requisitos funcionales</b> <b>no</b>	Seguridad y Usabilidad	
<b>Asuntos pendientes</b>	Ninguno	
<b>Flujos alternos</b>		
<b>Nº 2 &lt;Ocurre Error en la validación &gt;</b>		
	<b>Sistema</b>	
2.1	Se valida la estructura de la entidad del dominio que se obtiene, en la petición en caso de ser incorrecta.	
2.2	Se lanza un error 400.	

**CUS 14** Filtrar los eventos de acceso a la publicidad.

<b>Objetivo</b>	Proveer un punto final que permita Filtrar eventos de acceso a la publicidad en la Plataforma C.U.B.A.
<b>Actores</b>	-
<b>Resumen</b>	Filtrar evento de acceso a la publicidad

<b>Complejidad</b>	Alta	
<b>Prioridad</b>	Secundaria	
<b>Precondiciones</b>	-	
<b>Postcondiciones</b>	Registrar la acción del evento creado.	
<b>Flujo de eventos</b>		
<b>Flujo básico &lt; Filtrar los eventos de acceso a la publicidad &gt;</b>		
	<b>Sistema</b>	
1.	Se obtiene los datos del filtrado cuando se realiza una petición GET al endpoint <i>/event/filter/search</i> .	
2.	Se validan que los datos estén en un formato correcto, en el caso de estar correctos se pasa a la capa del repositorio.	
3.	Una vez completado, se construye el listado con los eventos de búsquedas con coincidieron con los datos de filtrado y se devuelve, se crea una acción de trazabilidad del evento filtrado y se envía un código de estado de satisfacción (200).	
<b>Relaciones</b>	<b>CU Incluidos</b>	Ninguno
	<b>CU Extendidos</b>	Ninguno
<b>Requisitos funcionales</b> no	Seguridad y Usabilidad	
<b>Asuntos pendientes</b>	Ninguno	
<b>Flujos alternos</b>		

Nº 2 <Ocurre un error en la validación >	
Sistema	
2.1	Si existe un Error en la validación
2.2	Se devuelve un código de estado 404

**CUS 15** Mostrar evento de acceso a la publicidad.

<b>Objetivo</b>	Proveer un punto final que permita mostrar un evento de acceso a la publicidad en la Plataforma C.U.B.A.
<b>Actores</b>	-
<b>Resumen</b>	Mostrar evento de acceso a la publicidad.
<b>Complejidad</b>	Alta
<b>Prioridad</b>	Secundaria
<b>Precondiciones</b>	-
<b>Postcondiciones</b>	Registrar la acción del evento creado.
<b>Flujo de eventos</b>	
<b>Flujo básico &lt; Mostrar evento de acceso a la publicidad&gt;</b>	
Sistema	
1.	Obtiene los datos a través de una petición <i>GET</i> que se realiza al endpoint <i>/event/show/search/{id}</i> .
2.	Se valida que exista un evento con ese identificador en caso ser correcto se envía a la capa de repositorio directamente.

3.	Se validan los datos de la entidad recibida en caso de estar correctos se envían los datos al repositorios.	
4.	Una vez completado, se obtiene el evento, se crea una acción de trazabilidad de obtención de evento con ese identificador y se envía un código de estado de satisfacción (201).	
<b>Relaciones</b>	<b>CU Incluidos</b>	Ninguno
	<b>CU Extendidos</b>	Ninguno
<b>Requisitos funcionales</b>	<b>no</b>	Seguridad y Usabilidad
<b>Asuntos pendientes</b>		Ninguno
<b>Flujos alternos</b>		
<b>Nº 2 &lt;Ocurre un error en la validación &gt;</b>		
	<b>Sistema</b>	
2.1	Si no existe un evento con ese identificador en la validación	
2.2	Se devuelve un código de estado 404	

**CUS 16** Eliminar eventos de acceso a la publicidad.

<b>Objetivo</b>	Proveer un punto final que permita Eliminar un evento según un identificador de acceso a la publicidad en la Plataforma C.U.B.A.
<b>Actores</b>	-

<b>Resumen</b>	Eliminar evento de acceso a la publicidad	
<b>Complejidad</b>	Alta	
<b>Prioridad</b>	Secundaria	
<b>Precondiciones</b>	-	
<b>Postcondiciones</b>	Registrar la acción del evento creado.	
<b>Flujo de eventos</b>		
<b>Flujo básico &lt; Eliminar eventos de acceso a la publicidad &gt;</b>		
	<b>Sistema</b>	
7.	Se Obtiene el o los identificador/es a través de una petición <i>DELETE</i> que se realiza al endpoint  <i>/event/detele/search/{id}</i> .	
8.	Se valida de que exista eventos con ese identificador, en caso de existir se pasa al repositorio directamente.	
9.	Se elimina de forma satisfactoria.	
10.	Una vez completado, se crea un evento de trazabilidad de evento eliminado y no se envía un código de estado. (No contex)	
<b>Relaciones</b>	<b>CU Incluidos</b>	Ninguno
	<b>CU Extendidos</b>	Ninguno
<b>Requisitos funcionales</b>	<b>no</b>	Seguridad y Usabilidad

<b>Asuntos pendientes</b>	Ninguno
<b>Flujos alternos</b>	
<b>Nº 2 &lt;Ocurre un error en la validación &gt;</b>	
	<b>Sistema</b>
2.1	Si no existe un evento con ese identificador.
2.2	Se devuelve un código de estado 404

### **Gestionar eventos de garabatos:**

**CUS 17** Registrar evento de descarga de garabatos.

<b>Objetivo</b>	Proveer un punto final que permita registrar un evento cuando se realice una descarga de garabatos en la Plataforma C.U.B.A.
<b>Actores</b>	-
<b>Resumen</b>	Registrar evento de descarga de garabatos.
<b>Complejidad</b>	Alta
<b>Prioridad</b>	Crítica
<b>Precondiciones</b>	-
<b>Postcondiciones</b>	Registrar la acción del evento creado.
<b>Flujo de eventos</b>	
<b>Flujo básico &lt; Registrar evento de descarga de garabatos &gt;</b>	

Sistema		
11.	Obtiene los datos a través de una petición POST que se realiza al endpoint <i>/event/create/search/web</i> .	
12.	Se valida la estructura de la entidad del dominio que se obtiene en la petición en caso de ser correcta se envían los datos a la capa de servicios.	
13.	Se validan los datos de la entidad recibida en caso de estar correctos se envían los datos al repositorios.	
14.	Se obtiene la dirección IP de la entidad y a partir de la misma se completa la información de geo localización de la fuente donde se realizó la búsqueda en la web a partir de la base de datos GeoLite.mmdb.	
15.	Una vez completado, se registra el evento, se crea una acción de trazabilidad del evento creado y se envía un código de estado de satisfacción (201).	
<b>Relaciones</b>	<b>CU Incluidos</b>	Ninguno
	<b>CU Extendidos</b>	Ninguno
<b>Requisitos funcionales</b>	<b>no</b>	Seguridad y Usabilidad
<b>Asuntos pendientes</b>		Ninguno
<b>Flujos alternos</b>		
<b>Nº 2 &lt;Ocurre Error en la validación &gt;</b>		
Sistema		

2.1	Se valida la estructura de la entidad del dominio que se obtiene, en la petición en caso de ser incorrecta.
2.2	Se lanza un error 400.

**CUS 18** Registrar evento de visualización de garabatos.

<b>Objetivo</b>	Proveer un punto final que permita registrar un evento cuando se realice una visualización de garabatos en la Plataforma C.U.B.A.
<b>Actores</b>	-
<b>Resumen</b>	Registrar evento de visualización de garabatos.
<b>Complejidad</b>	Alta
<b>Prioridad</b>	Crítica
<b>Precondiciones</b>	-
<b>Postcondiciones</b>	Registrar la acción del evento creado.
<b>Flujo de eventos</b>	
<b>Flujo básico &lt; Registrar evento de visualización de garabatos &gt;</b>	
	<b>Sistema</b>
16.	Obtiene los datos a través de una petición POST que se realiza al endpoint <i>/event/create/search/web</i> .
17.	Se valida la estructura de la entidad del dominio que se obtiene en la petición en caso de ser correcta se envían los datos a la capa de servicios.
18.	Se validan los datos de la entidad recibida en caso de estar correctos se envían los datos al repositorios.

19.	Se obtiene la dirección IP de la entidad y a partir de la misma se completa la información de geo localización de la fuente donde se realizó la búsqueda en la web a partir de la base de datos GeoLite.mmdb.	
20.	Una vez completado, se registra el evento, se crea una acción de trazabilidad del evento creado y se envía un código de estado de satisfacción (201).	
<b>Relaciones</b>	<b>CU Incluidos</b>	Ninguno
	<b>CU Extendidos</b>	Ninguno
<b>Requisitos funcionales</b>	<b>no</b>	Seguridad y Usabilidad
<b>Asuntos pendientes</b>		Ninguno
<b>Flujos alternos</b>		
<b>Nº 2 &lt;Ocurre Error en la validación &gt;</b>		
	<b>Sistema</b>	
2.1	Se valida la estructura de la entidad del dominio que se obtiene, en la petición en caso de ser incorrecta.	
2.2	Se lanza un error 400.	

**CUS 19** Filtrar los eventos asociados a garabatos.

<b>Objetivo</b>	Proveer un punto final que permita Filtrar eventos asociados a garabatos en la Plataforma C.U.B.A.	
<b>Actores</b>	-	
<b>Resumen</b>	Filtrar eventos asociados a garabatos	
<b>Complejidad</b>	Alta	
<b>Prioridad</b>	Secundaria	
<b>Precondiciones</b>	-	
<b>Postcondiciones</b>	Registrar la acción del evento creado.	
<b>Flujo de eventos</b>		
<b>Flujo básico &lt; Filtrar los eventos asociados a garabatos &gt;</b>		
	<b>Sistema</b>	
4.	Se obtiene los datos del filtrado cuando se realiza una petición GET al endpoint <i>/event/filter/search</i> .	
5.	Se validan que los datos estén en un formato correcto, en el caso de estar correctos se pasa a la capa del repositorio.	
6.	Una vez completado, se construye el listado con los eventos de búsquedas con coincidieron con los datos de filtrado y se devuelve, se crea una acción de trazabilidad del evento filtrado y se envía un código de estado de satisfacción (200).	
<b>Relaciones</b>	<b>CU Incluidos</b>	Ninguno
	<b>CU Extendidos</b>	Ninguno

<b>Requisitos funcionales</b>	<b>no</b>	Seguridad y Usabilidad
<b>Asuntos pendientes</b>		Ninguno
<b>Flujos alternos</b>		
<b>Nº 2 &lt;El Administrador desea eliminar el Evento&gt;</b>		
		<b>Sistema</b>
2.1		
2.2		Se Muestra Mensaje "Evento Eliminado"

**CUS 20** Mostrar evento asociado a garabatos.

<b>Objetivo</b>	Proveer un punto final que permita mostrar un evento asociado a garabatos en la Plataforma C.U.B.A.
<b>Actores</b>	-
<b>Resumen</b>	Mostrar evento asociado a garabatos
<b>Complejidad</b>	Alta
<b>Prioridad</b>	Secundaria
<b>Precondiciones</b>	-
<b>Postcondiciones</b>	Registrar la acción del evento creado.
<b>Flujo de eventos</b>	

<b>Flujo básico &lt; Mostrar evento asociado a garabatos &gt;</b>		
	<b>Sistema</b>	
5.	Obtiene los datos a través de una petición <i>GET</i> que se realiza al endpoint <i>/event/show/search/{id}</i> .	
6.	Se valida que exista un evento con ese identificador en caso ser correcto se envía a la capa de repositorio directamente.	
7.	Se validan los datos de la entidad recibida en caso de estar correctos se envían los datos al repositorios.	
8.	Una vez completado, se obtiene el evento, se crea una acción de trazabilidad de obtención de evento con ese identificador y se envía un código de estado de satisfacción (201).	
<b>Relaciones</b>	<b>CU Incluidos</b>	Ninguno
	<b>CU Extendidos</b>	Ninguno
<b>Requisitos funcionales</b>	<b>no</b>	Seguridad y Usabilidad
<b>Asuntos pendientes</b>		Ninguno
<b>Flujos alternos</b>		
<b>Nº 2 &lt;Error en la validación &gt;</b>		
	<b>Sistema</b>	
2.1	Si no existe un evento con los datos a validar	
2.2	Se devuelve un código de estado 404	

**CUS 21** Eliminar eventos asociados a garabatos.

<b>Objetivo</b>	Proveer un punto final que permita Eliminar un evento según un identificador del evento asociados a garabatos en la Plataforma C.U.B.A.	
<b>Actores</b>	-	
<b>Resumen</b>	Eliminar evento asociados a garabatos	
<b>Complejidad</b>	Alta	
<b>Prioridad</b>	Secundaria	
<b>Precondiciones</b>	-	
<b>Postcondiciones</b>	Registrar la acción del evento creado.	
<b>Flujo de eventos</b>		
<b>Flujo básico &lt; Eliminar eventos asociados a garabatos &gt;</b>		
	<b>Sistema</b>	
11.	Se Obtiene el o los identificador/es a través de una petición <i>DELETE</i> que se realiza al endpoint <code>/event/detele/search/{id}</code> .	
12.	Se valida de que exista eventos con ese identificador, en caso de existir se pasa a la capa de repositorio.	
13.	Se elimina de forma satisfactoria.	
14.	Una vez completado, se crea un evento de trazabilidad de evento eliminado y no se envía un código de estado. (No contex)	
<b>Relaciones</b>	<b>CU Incluidos</b>	Ninguno

	<b>CU Extendidos</b>	Ninguno
<b>Requisitos funcionales</b>	<b>no</b>	Seguridad y Usabilidad
<b>Asuntos pendientes</b>		Ninguno
<b>Flujos alternos</b>		
<b>Nº 2 &lt;Ocurre un Error en la validación &gt;</b>		
	<b>Sistema</b>	
2.1	Si no existe un evento con ese identificador.	
2.2	Se devuelve un código de estado 404	

**Gestionar eventos de fondo de pantalla:**

**CUS 22** Registrar evento de descarga de fondo de pantalla.

<b>Objetivo</b>	Proveer un punto final que permita registrar un evento cuando se realice una descarga de fondo de pantalla en la Plataforma C.U.B.A.
<b>Actores</b>	-
<b>Resumen</b>	Registrar evento de descarga de fondo de pantalla.
<b>Complejidad</b>	Alta
<b>Prioridad</b>	Crítica

<b>Precondiciones</b>	-	
<b>Postcondiciones</b>	Registrar la acción del evento creado.	
<b>Flujo de eventos</b>		
<b>Flujo básico &lt; Registrar evento de descarga de fondo de pantalla &gt;</b>		
	<b>Sistema</b>	
21.	Obtiene los datos a través de una petición POST que se realiza al endpoint <i>/event/create/search/web</i> .	
22.	Se valida la estructura de la entidad del dominio que se obtiene en la petición en caso de ser correcta se envían los datos a la capa de servicios.	
23.	Se validan los datos de la entidad recibida en caso de estar correctos se envían los datos al repositorios.	
24.	Se obtiene la dirección IP de la entidad y a partir de la misma se completa la información de geo localización de la fuente donde se realizó la búsqueda en la web a partir de la base de datos GeoLite.mmdb.	
25.	Una vez completado, se registra el evento, se crea una acción de trazabilidad del evento creado y se envía un código de estado de satisfacción (201).	
<b>Relaciones</b>	<b>CU Incluidos</b>	Ninguno
	<b>CU Extendidos</b>	Ninguno
<b>Requisitos funcionales</b>	<b>no</b>	Seguridad y Usabilidad
<b>Asuntos pendientes</b>	Ninguno	

<b>Flujos alternos</b>	
<b>Nº 2 &lt;Ocurre Error en la validación &gt;</b>	
	<b>Sistema</b>
2.1	Se valida la estructura de la entidad del dominio que se obtiene, en la petición en caso de ser incorrecta.
2.2	Se lanza un error 400.

**CUS 23** Registrar evento de visualización de fondo de pantalla.

<b>Objetivo</b>	Proveer un punto final que permita registrar un evento cuando se realice una visualización de fondo de pantalla en la Plataforma C.U.B.A.
<b>Actores</b>	-
<b>Resumen</b>	Registrar evento de visualización de fondo de pantalla.
<b>Complejidad</b>	Alta
<b>Prioridad</b>	Critica
<b>Precondiciones</b>	-
<b>Postcondiciones</b>	Registrar la acción del evento creado.
<b>Flujo de eventos</b>	
<b>Flujo básico &lt; Registrar evento de visualización de fondo de pantalla &gt;</b>	
	<b>Sistema</b>

26.	Obtiene los datos a través de una petición POST que se realiza al endpoint <i>/event/create/search/web</i> .	
27.	Se valida la estructura de la entidad del dominio que se obtiene en la petición en caso de ser correcta se envían los datos a la capa de servicios.	
28.	Se validan los datos de la entidad recibida en caso de estar correctos se envían los datos al repositorios.	
29.	Se obtiene la dirección IP de la entidad y a partir de la misma se completa la información de geo localización de la fuente donde se realizó la búsqueda en la web a partir de la base de datos GeoLite.mmdb.	
30.	Una vez completado, se registra el evento, se crea una acción de trazabilidad del evento creado y se envía un código de estado de satisfacción (201).	
<b>Relaciones</b>	<b>CU Incluidos</b>	Ninguno
	<b>CU Extendidos</b>	Ninguno
<b>Requisitos funcionales</b>	<b>no</b>	Seguridad y Usabilidad
<b>Asuntos pendientes</b>		Ninguno
<b>Flujos alternos</b>		
<b>Nº 2 &lt;Ocurre Error en la validación &gt;</b>		
	<b>Sistema</b>	
2.1	Se valida la estructura de la entidad del dominio que se obtiene, en la petición en caso de ser incorrecta.	

2.2	Se lanza un error 400.
-----	------------------------

**CUS 24** Filtrar los eventos asociados al fondo de pantalla.

<b>Objetivo</b>	Proveer un punto final que permita Filtrar eventos asociados al fondo de pantalla en la Plataforma C.U.B.A.
<b>Actores</b>	-
<b>Resumen</b>	Filtrar eventos asociados al fondo de pantalla
<b>Complejidad</b>	Alta
<b>Prioridad</b>	Secundaria
<b>Precondiciones</b>	-
<b>Postcondiciones</b>	Registrar la acción del evento creado.
<b>Flujo de eventos</b>	
<b>Flujo básico &lt; Filtrar los eventos asociados al fondo de pantalla &gt;</b>	
	<b>Sistema</b>
7.	Se obtiene los datos del filtrado cuando se realiza una petición GET al endpoint <i>/event/filter/search</i> .
8.	Se validan que los datos estén en un formato correcto, en el caso de estar correctos se pasa a la capa del repositorio.
9.	Una vez completado, se construye el listado con los eventos de búsquedas con coincidieron con los datos de filtrado y se devuelve, se crea una acción de trazabilidad del evento filtrado y se envía un código de estado de satisfacción (200).

<b>Relaciones</b>	<b>CU Incluidos</b>	Ninguno
	<b>CU Extendidos</b>	Ninguno
<b>Requisitos funcionales</b>	<b>no</b>	Seguridad y Usabilidad
<b>Asuntos pendientes</b>		Ninguno
<b>Flujos alternos</b>		
<b>Nº 2 &lt;Ocurre un error en la validación &gt;</b>		
	<b>Sistema</b>	
2.1	Si no existe un evento con los datos en la validación	
2.2	Se devuelve un código de estado 404	

**CUS 25** Mostrar evento asociado al fondo de pantalla.

<b>Objetivo</b>	Proveer un punto final que permita mostrar un evento asociado a fondo de pantalla en la Plataforma C.U.B.A.
<b>Actores</b>	-
<b>Resumen</b>	Mostrar evento asociado a fondo de pantalla
<b>Complejidad</b>	Alta
<b>Prioridad</b>	Secundaria

<b>Precondiciones</b>	-	
<b>Postcondiciones</b>	Registrar la acción del evento creado.	
<b>Flujo de eventos</b>		
<b>Flujo básico &lt; Mostrar evento asociado a fondo de pantalla &gt;</b>		
	<b>Sistema</b>	
9.	Obtiene los datos a través de una petición <i>GET</i> que se realiza al endpoint <i>/event/show/search/{id}</i> .	
10.	Se valida que exista un evento con ese identificador en caso ser correcto se envía a la capa de repositorio directamente.	
11.	Se validan los datos de la entidad recibida en caso de estar correctos se envían los datos al repositorios.	
12.	Una vez completado, se obtiene el evento, se crea una acción de trazabilidad de obtención de evento con ese identificador y se envía un código de estado de satisfacción (201).	
<b>Relaciones</b>	<b>CU Incluidos</b>	Ninguno
	<b>CU Extendidos</b>	Ninguno
<b>Requisitos funcionales</b>	<b>no</b>	Seguridad y Usabilidad
<b>Asuntos pendientes</b>	Ninguno	
<b>Flujos alternos</b>		
<b>Nº 2 &lt;Error en la validación &gt;</b>		

Sistema	
2.1	Si no existe un evento con ese identificador
2.2	Se devuelve un código de estado 404

**CUS 26** Eliminar eventos asociados al fondo de pantalla.

<b>Objetivo</b>	Proveer un punto final que permita Eliminar según un identificador asociados un evento asociados al fondo de pantalla en la Plataforma C.U.B.A.
<b>Actores</b>	-
<b>Resumen</b>	Eliminar evento asociados al fondo de pantalla
<b>Complejidad</b>	Alta
<b>Prioridad</b>	Secundaria
<b>Precondiciones</b>	-
<b>Postcondiciones</b>	Registrar la acción del evento creado.
<b>Flujo de eventos</b>	
<b>Flujo básico &lt; Eliminar eventos de asociados al fondo de pantalla &gt;</b>	
Sistema	
15.	Se Obtiene el o los identificador/es a través de una petición <i>DELETE</i> que se realiza al endpoint  <i>/event/detele/search/{id}</i> .
16.	Se valida de que exista eventos con ese identificador en caso, en caso de existir se pasa al repositorio.

17.	Se elimina de forma satisfactoria.	
18.	Una vez completado, se crea un evento de trazabilidad de evento eliminado y no se envía un código de estado. (No contex)	
<b>Relaciones</b>	<b>CU Incluidos</b>	Ninguno
	<b>CU Extendidos</b>	Ninguno
<b>Requisitos funcionales</b>	<b>no</b>	Seguridad y Usabilidad
<b>Asuntos pendientes</b>		Ninguno
<b>Flujos alternos</b>		
<b>Nº 2 &lt;Ocurre un error&gt;</b>		
	<b>Sistema</b>	
2.1	Si no existe un evento con ese identificador.	
2.2	Se devuelve un código de estado 404	

**Gestión de eventos conversión de monedas:**

**CUS 27** Registrar evento de conversión de monedas.

<b>Objetivo</b>	Proveer un punto final que permita registrar un evento cuando se realice una conversión de monedas en la Plataforma C.U.B.A.
<b>Actores</b>	-

<b>Resumen</b>	Registrar evento de conversión de monedas	
<b>Complejidad</b>	Alta	
<b>Prioridad</b>	¿?	
<b>Precondiciones</b>	-	
<b>Postcondiciones</b>	Registrar la acción del evento creado.	
<b>Flujo de eventos</b>		
<b>Flujo básico &lt; Registrar evento de conversión de monedas &gt;</b>		
	<b>Sistema</b>	
1.	Obtiene los datos a través de una petición POST que se realiza al endpoint <i>/event/create/search/web</i> .	
2.	Se valida la estructura de la entidad del dominio que se obtiene en la petición en caso de ser correcta se envían los datos a la capa de servicios.	
3.	Se validan los datos de la entidad recibida en caso de estar correctos se envían los datos al repositorios.	
4.	Se obtiene la dirección IP de la entidad y a partir de la misma se completa la información de geo localización de la fuente donde se realizó la búsqueda en la web a partir de la base de datos GeoLite.mmdb.	
5.	Una vez completado, se registra el evento, se crea una acción de trazabilidad del evento creado y se envía un código de estado de satisfacción (201).	
<b>Relaciones</b>	<b>CU Incluidos</b>	Ninguno
	<b>CU Extendidos</b>	Ninguno

<b>Requisitos funcionales</b>	<b>no</b>	Seguridad y Usabilidad
<b>Asuntos pendientes</b>		Ninguno
<b>Flujos alternos</b>		
<b>Nº 2 &lt;Ocurre Error en la validación &gt;</b>		
	<b>Sistema</b>	
2.1	Se valida la estructura de la entidad del dominio que se obtiene, en la petición en caso de ser incorrecta.	
2.2	Se lanza un error 400.	

**CUS 28** Filtrar los eventos de conversión de monedas.

<b>Objetivo</b>	Proveer un punto final que permita Filtrar eventos de conversión de monedas en la Plataforma C.U.B.A.
<b>Actores</b>	-
<b>Resumen</b>	Filtrar eventos de conversión de monedas
<b>Complejidad</b>	Alta
<b>Prioridad</b>	Secundaria
<b>Precondiciones</b>	-
<b>Postcondiciones</b>	Registrar la acción del evento creado.

<b>Flujo de eventos</b>		
<b>Flujo básico &lt; Filtrar los eventos conversión de monedas &gt;</b>		
	<b>Sistema</b>	
1.	Se obtiene los datos del filtrado cuando se realiza una petición GET al endpoint <i>/event/filter/search</i> .	
2.	Se validan que los datos estén en un formato correcto, en el caso de estar correctos se pasa a la capa del repositorio.	
3.	Una vez completado, se construye el listado con los eventos de búsquedas con coincidieron con los datos de filtrado y se devuelve, se crea una acción de trazabilidad del evento filtrado y se envía un código de estado de satisfacción (200).	
<b>Relaciones</b>	<b>CU Incluidos</b>	Ninguno
	<b>CU Extendidos</b>	Ninguno
<b>Requisitos funcionales</b>	<b>no</b>	Seguridad y Usabilidad
<b>Asuntos pendientes</b>		Ninguno
<b>Flujos alternos</b>		
<b>Nº 2 &lt;Error en la Validación &gt;</b>		
	<b>Sistema</b>	
2.1	No existe ningún evento con ese identificador	
2.2	Se devuelve un código de estado 404	

**CUS 29** Mostrar evento de conversión de monedas.

<b>Objetivo</b>	Proveer un punto final que permita mostrar un evento de conversión de monedas en la Plataforma C.U.B.A.	
<b>Actores</b>	-	
<b>Resumen</b>	Mostrar evento de conversión de monedas.	
<b>Complejidad</b>	Alta	
<b>Prioridad</b>	Secundaria	
<b>Precondiciones</b>	-	
<b>Postcondiciones</b>	Registrar la acción del evento creado.	
<b>Flujo de eventos</b>		
<b>Flujo básico &lt; Mostrar evento de conversión de monedas.&gt;</b>		
	<b>Sistema</b>	
1.	Obtiene los datos a través de una petición <i>GET</i> que se realiza al endpoint <i>/event/show/search/{id}</i> .	
2.	Se valida que exista un evento con ese identificador en caso ser correcto se envía a la capa de repositorio directamente.	
3.	Se validan los datos de la entidad recibida en caso de estar correctos se envían los datos al repositorios.	
4.	Una vez completado, se obtiene el evento, se crea una acción de trazabilidad de obtención de evento con ese identificador y se envía un código de estado de satisfacción (201).	
<b>Relaciones</b>	<b>CU Incluidos</b>	Ninguno

	<b>CU Extendidos</b>	Ninguno
<b>Requisitos funcionales</b>	<b>no</b>	Seguridad y Usabilidad
<b>Asuntos pendientes</b>		Ninguno
<b>Flujos alternos</b>		
<b>Nº 2 &lt;No encontrar datos&gt;</b>		
	<b>Sistema</b>	
2.1	Se devuelve un código de estado 404	

**CUS 30** Eliminar eventos de conversión de moneda.

<b>Objetivo</b>	Proveer un punto final que permita Eliminar según un identificador un evento de conversión de moneda en la Plataforma C.U.B.A.
<b>Actores</b>	-
<b>Resumen</b>	Eliminar evento de conversión de moneda
<b>Complejidad</b>	Alta
<b>Prioridad</b>	Secundaria
<b>Precondiciones</b>	-
<b>Postcondiciones</b>	Registrar la acción del evento creado.

<b>Flujo de eventos</b>		
<b>Flujo básico &lt; Eliminar eventos de conversión de moneda &gt;</b>		
<b>Sistema</b>		
1.	Se obtiene el o los identificador/es a través de una petición <i>DELETE</i> que se realiza al endpoint <i>/event/detele/search/{id}</i> .	
2.	Se valida de que exista eventos con ese identificador, en caso de existir se pasa a la capa de repositorio.	
3.	Se elimina de forma satisfactoria.	
4.	Una vez completado, se crea un evento de trazabilidad de evento eliminado y no se envía un código de estado. (No contex)	
<b>Relaciones</b>	<b>CU Incluidos</b>	Ninguno
	<b>CU Extendidos</b>	Ninguno
<b>Requisitos funcionales</b>	<b>no</b>	Seguridad y Usabilidad
<b>Asuntos pendientes</b>		Ninguno
<b>Flujos alternos</b>		
<b>Nº 2 &lt;Ocurre un error en la validación &gt;</b>		
<b>Sistema</b>		
2.1	Si no existe un evento con ese identificador.	
2.2	Se devuelve un código de estado 404	

## Gestionar eventos trazabilidad:

**CUS 31** Filtrar los eventos de trazabilidad.

<b>Objetivo</b>	Proveer un punto final que permita Filtrar los eventos de trazabilidad en la Plataforma C.U.B.A y devolver un listado con los eventos filtrados.	
<b>Actores</b>	-	
<b>Resumen</b>	Filtrar los eventos de trazabilidad	
<b>Complejidad</b>	Alta	
<b>Prioridad</b>	Secundaria	
<b>Precondiciones</b>	-	
<b>Postcondiciones</b>	Registrar la acción de trazabilidad del evento filtrado.	
<b>Flujo de eventos</b>		
<b>Flujo básico &lt; Filtrar los eventos de trazabilidad &gt;</b>		
	<b>Sistema</b>	
7.	Se obtiene los datos del filtrado cuando se realiza una petición GET al endpoint <i>/event/filter/search</i> .	
8.	Se validan que los datos estén en un formato correcto, en el caso de estar correctos se pasa a la capa del repositorio.	
9.	Una vez completado, se construye el listado con los eventos de trazabilidad que coincidieron con los datos de filtrado y se devuelve, se crea una acción de trazabilidad del evento filtrado y se envía un código de estado de satisfacción (200).	
<b>Relaciones</b>	<b>CU Incluidos</b>	Ninguno

	<b>CU Extendidos</b>	Ninguno
<b>Requisitos funcionales</b>	<b>no</b>	Seguridad y Usabilidad
<b>Asuntos pendientes</b>		Ninguno
<b>Flujos alternos</b>		
<b>Nº 2 &lt;Ocurre un error en la validación de los datos&gt;</b>		
	<b>Sistema</b>	
2.1	Si los datos del filtro no se encuentran en los formatos correctos se envía un código de estado de error de petición (400).	

## 2.5. Arquitectura del software.

Para estructurar un proyecto es necesario conocer los distintos estilos de arquitectura que existen y escoger cuál se adecua a las características y necesidades del proyecto. Una **Arquitectura de Software** es un conjunto de patrones que establecen la estructura, el funcionamiento y la interacción entre las partes del software, proporcionando así un marco de referencia necesario para guiar la construcción de un software a los desarrolladores del mismo. (Jiménez, 2017).

### **Arquitectura N-Capas Orientación al Dominio:**

La arquitectura que se analiza es la desplegada en la plataforma, por lo cual se tuvo en cuenta al ser este un requisito del cliente. Esta arquitectura permitirá la incorporación de la solución propuesta sin causar daños ni cambios a los servicios que ya ofrece la plataforma.

La **capa de servicio web o de presentación** es la que presenta al usuario los conceptos de negocio mediante una interfaz de usuario.

La **capa de servicios o de aplicación** realiza las llamadas a servicios de la capa inferior y tiene la responsabilidad de adaptar la información que le llega, a los requerimientos de los servicios de dominio.

La **capa de dominio** es responsable de las validaciones, define las interfaces de persistencia a datos (contratos de repositorio) pero no los implementa y está compuesta por entidades del dominio que representan objetos del dominio y están definidas fundamentalmente por su identidad, servicios de dominio que contienen la lógica que trata a las entidades como un todo y los contratos de repositorios que son interfaces que especifican las operaciones que deben implementar los repositorios.

La **capa de Repositorio o de persistencia** es responsable de contener el código necesario para persistir los datos, contiene como componente los repositorios que son clases que implementan los contratos de repositorios definidos en la capa de dominio.

Finalmente, la **capa Infraestructura transversal** es responsable de promover la reutilización de código, ella albergará las operaciones de seguridad, *logging*, monitoreo del sistema, mecanismos de persistencia reutilizables, validadores genéricos y todas aquellas operaciones que se puedan utilizar desde otras capas. (Figura 2. Vista de Arquitectura de software de la plataforma C.U.B.A.).

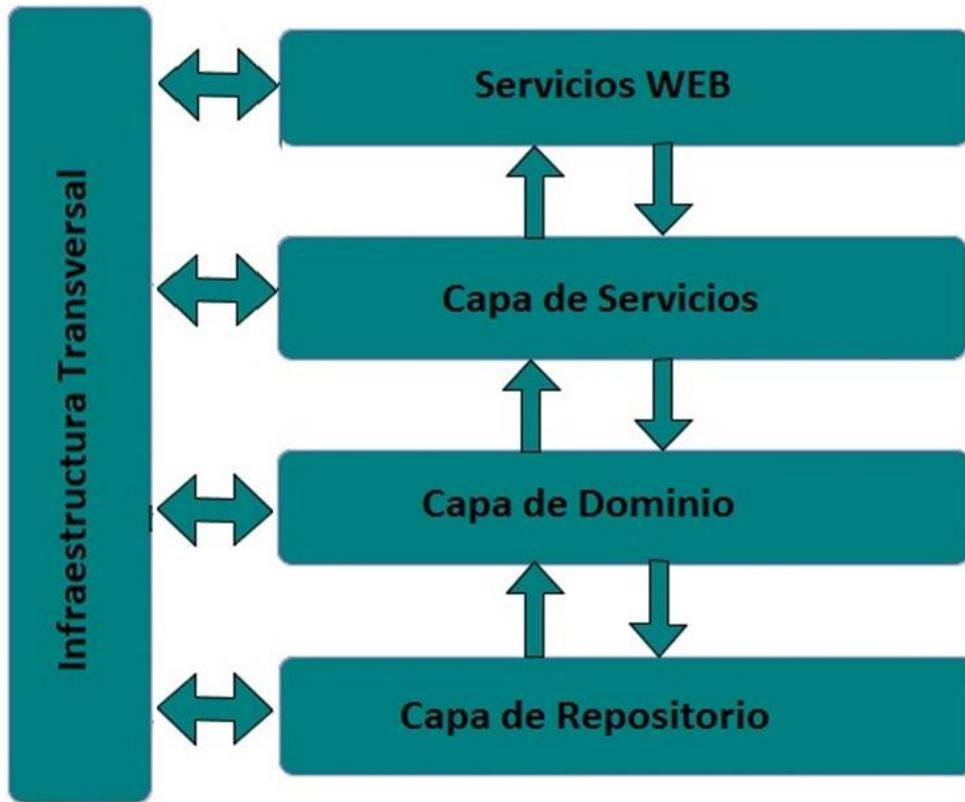


Figura 2. Vista de Arquitectura de software de la plataforma C.U.B.A.

## 2.6. Conclusiones del capítulo

- Se propone como solución un microservicio como solución backend el cual registrará los eventos que se produzcan en la plataforma cuando el usuario interactúa con esta, esta solución se integra a la arquitectura ya existente en la plataforma.
- Para una mejor comprensión del negocio se hace un modelado conceptual, el cual permite realizar un análisis del flujo de interacción del usuario con la plataforma que origina los eventos a registrar para nuestra solución.
- Para los requisitos del software se plantean los requisitos funcionales con los cuales debe cumplir el sistema agrupados para facilitar su análisis en grupos de eventos relacionados, y sus requisitos no funcionales con las propiedades que este necesita para llegar a la solución expuesta.
- En el Diagrama de casos de uso del sistema se analiza por grupos de eventos los casos de uso del sistema(CUS), que se obtienen del análisis de los requerimientos para el software y su posterior descripción de cómo funciona el flujo de registro de evento que comprende la solución propuesta.
- Como propuesta del cliente se le añadirá a la arquitectura que utiliza la plataforma los servicios que contribuirán al registro de eventos de la utilización de la plataforma por los usuarios, al permitir este tipo de arquitectura que se añada nuevos servicios sin dañar o cambiar ya los existentes.

## Bibliografía

- Arsys. (2018, mayo 30). Administra tu propio servidor de estadísticas con Matomo. *Blog de arsys.es*.  
<https://www.arsys.es/blog/soluciones/cloud/cloudbuilder-next/estadisticas-matomo/>
- Chanchí, G. E., Campo, W. Y., Amaya, J. P., & Arciniegas, J. L. (2011). Esquema de servicios para Televisión Digital Interactiva, basados en el protocolo REST-JSON. *Cadernos de Informática*, 6(1), 233-240.
- Fernández, O. B. (2005). Introducción al lenguaje de programación Java. *Una guía básica*, 9.
- Gonzalo Soto, N. (2018). *Desarrollo de una API para datos abiertos*.
- Jiménez de la Calle, D. (2017). *Analítica web en ecommerce y herramientas para ecommerce*.
- Jun, Y., Zhishu, L., & Yanyan, M. (2008). Json based decentralized sso security architecture in e-commerce. *2008 International Symposium on Electronic Commerce and Security*, 471-475.
- Mateos Fernández, A. M., & Sparapani Galicia, L. (2017). *Reporte analítica de proyectos web con Google Analytics*. <http://uvadoc.uva.es/handle/10324/25513>
- Merelles, G., & de Borja, F. (2019). *Analítica web en medios digitales: Uso de datos de audiencia y métricas en los grandes grupos de comunicación españoles*. Universidad Complutense de Madrid.
- Mora-Castillo, J. A. (2016). Serialización/deserialización de objetos y transmisión de datos con JSON: una revisión de la literatura. *Revista Tecnología en Marcha*, 29(1), ág. 118-125.
- Pryshepa Pryshepa, M. (2017). *API REST LoopBack (Node. JS) vs. Rust*. Universitat Politècnica de Catalunya.
- ¿Qué es la analítica web? ¿Para qué sirve la analítica web?* (2014, julio 11). Identio.  
<https://www.idento.es/blog/analitica-web/que-es-la-analitica-web-para-que-sirve/>

Villazón, Y. P. (2018). *Solución tecnológica para el monitoreo del sistema de recuperación de información de la Plataforma C.U.B.A.* <https://repositorio.uci.cu/jspui/handle/123456789/7911>