

Entorno de simulación de redes TCP/IP usando servicios REST basados en la nube computacional

Servicio de Colas

Rafael Cano Parra

29 de octubre de 2012 - Versión 1.2

Resumen

Este documento recoge los requisitos, el análisis y el diseño del Servicio de Colas de la aplicación *Distributed Network Simulation Environment III* (DNSE3). Este servicio ofrece una cola de elementos personalizables para la gestión de tareas asíncronas entre diferentes servicios. Para su definición se ha seguido la metodología ROA/D o Análisis y Diseño Orientado a Recursos, que consta de tres fases: obtención de requisitos, análisis del sistema y diseño del sistema. A partir de los seis requisitos funcionales definidos para la gestión de los elementos de la cola (listar, describir, añadir, aumentar su tiempo de vida, eliminar y obtener el siguiente elemento de la cola) se realizará el análisis y diseño del servicio.

Índice

1. Introducción	3
1.1. Propósito	3
1.2. Alcance	3
1.3. Definiciones, acrónimos y abreviaturas	3
1.4. Referencias	4
1.5. Organización del documento	4
2. Descripción general	4
2.1. Perspectiva del producto	5
2.2. Funciones del producto	5
2.3. Características de los usuarios	5
2.4. Restricciones	5
2.5. Suposiciones y dependencias	5
2.6. Requisitos futuros	5
3. Requisitos específicos	5
3.1. Interfaces externas	6
3.2. Funciones	6
3.3. Requisitos de rendimiento	7
3.4. Restricciones de diseño	7
3.5. Atributos del sistema	7
3.6. Otros requisitos	7
4. Análisis	7
4.1. Escenarios de uso	7
4.2. Modelo de dominio	8
4.3. Secuencias del sistema	9

5. Diseño	12
5.1. Diagrama de recursos	13
5.2. Espacio de URIs	14
5.3. Descripción de métodos y códigos de respuesta y error	14
5.4. Representaciones aceptadas y generadas	16

1. Introducción

En este apartado se presenta el propósito del documento, el alcance del sistema, una sección de definiciones, acrónimos y abreviaturas, las referencias bibliográficas y finalmente la organización del documento.

1.1. Propósito

Este documento de Análisis y Diseño Orientado a Recursos (en inglés, ROA/D) [1], completado con una sección inicial de Especificación de Requisitos Software [2], presenta la obtención de requisitos y el análisis y diseño del Servicio de Colas definido para el Entorno de Simulación de Red Distribuido III (o sus siglas en inglés, DNSE3). Inicialmente se realiza la obtención de los requisitos que ha de cumplir este servicio, de forma que se establece un *contrato* entre el cliente y el analista software, para posteriormente ampliar dichos requisitos realizando un análisis y diseño. Tras la realización de este análisis y diseño, el desarrollador podrá ser capaz de implementar dicho servicio a través de la información recogida en este documento.

1.2. Alcance

El alcance de este servicio abarca la gestión de los elementos de una cola para la comunicación asíncrona entre diferentes servicios de un mismo sistema basado en servicios. La libre disposición del contenido del mensaje de los elementos de la cola permite que el servicio sea generalista, es decir, pueda reutilizarse en otros sistemas. En el sistema DNSE3 se utiliza para comunicación asíncrona entre el Servicio de Orquestación y los diferentes Servicios de Simulación para la ejecución de las numerosas simulaciones del sistema.

Inicialmente, el Servicio de Orquestación incluirá en el mensaje del elemento de la cola, para cada simulación individual, dónde están los diferentes recursos a los que deberá acceder Servicio de Simulación para la consulta de la simulación y deposición de los resultados. Entonces, el servicio añadirá este elemento en la cola para que sea procesado. Por otro lado, un Servicio de Simulación preguntará a la cola cuál es el siguiente elemento pendiente de procesar, que obtendrá de la misma y lo procesará, dejando los resultados en el lugar indicado por el mensaje y marcándolo como procesado. La consulta de elementos pendientes a la cola variará según la disponibilidad de recursos en el Servicio de Simulación.

Otra de las funciones del servicio será la publicación de métricas en el Servicio de Monitorización que se le indique. Estas métricas serán relativas a la ocupación de la cola y ratio de procesamiento de sus elementos por los servicios externos. Su publicación servirá para el escalado de los Servicios de Simulación en el sistema DNSE3.

En cuanto al presente documento, inicialmente se realiza una Especificación de Requisitos Software para obtener todos los requisitos que el sistema debe satisfacer, que posteriormente son aplicados en el análisis y diseño del servicio, donde se establecen los escenarios de uso, el modelo de dominio y las secuencias del sistema en el caso del análisis, y el diagrama de recursos, espacio de URIs, los métodos y sus códigos de respuesta y error, así como las representaciones aceptadas y generadas en el caso del diseño.

Posteriormente, con toda esta definición de requisitos, análisis y diseño, se realizará la implementación del servicio tal y como se establece en esta definición.

1.3. Definiciones, acrónimos y abreviaturas

Las definiciones, acrónimos y abreviaturas utilizadas en este documento son las siguientes:

- **Cola.** Es el listado de elementos que contiene el servicio y que se implementará siguiendo una cola FIFO (*First Input, First Output*, es decir, el primero en entrar en la cola será el primero que salga de ella).
- **DNSE3.** *Distributed Network Simulation Environment III*, en español Entorno de Simulación de Red Distribuido III, es un entorno de simulación de redes TCP/IP distribuido en el que, a través de un paquete de simulaciones, se permiten realizar y obtener resultados de simulaciones individuales o de barrido de parámetros.

- **Elemento.** Un elemento de la cola del Servicio de Colas será aquel objeto que contiene la información necesaria para su procesamiento, el estado de procesamiento en el que se encuentra y el tiempo de vida máximo para su ejecución.
- **Orquestación.** En lo referente al Servicio de Orquestación, es el componente del sistema DNSE3 que controla el flujo de las diferentes funciones, ya sea realizando ciertas funciones atómicas o redirigiendo dichas funciones a uno de los servicios que componen el sistema.
- **REST.** *Representational State Transfer*, en español Transferencia de Estado Representacional, es un conjunto de criterios de diseño para la Arquitectura Orientada a Recursos (*Resource-Oriented Architecture*, ROA).
- **Servicio.** Es un componente del sistema DNSE3 que tiene la capacidad de realizar una funcionalidad específica con cierta independencia dentro del sistema para el reparto de la carga del mismo.
- **Tiempo de vida.** Se establece para un elemento de la cola del Servicio de Colas, y es el tiempo máximo que un elemento puede estar en estado de ejecución antes de que el servicio restablezca su estado a disponible nuevamente, suponiendo que no ha podido finalizarse su ejecución.

1.4. Referencias

- [1] J. Louvel, T. Templier, and T. Boileau, *Restlet in Action*, ch. Designing a RESTful web API. Manning Publications, 13 ed., 2012.
- [2] “IEEE Recommended Practice for Software Requirements Specifications,” *IEEE Std 830-1998*, 1998.
- [3] L. Richardson and S. Ruby, *RESTful web services*. Franham: O’Reilly, 2007.
- [4] J. Louvel, T. Templier, and T. Boileau, *Restlet in Action*. Manning Publications, 13 ed., 2012.

1.5. Organización del documento

La presente introducción de este documento está organizada según las plantillas utilizadas en los artefactos software (por ejemplo, en [2]), en la que se incluye el propósito del documento, el alcance del sistema, las definiciones, acrónimos y abreviaturas utilizadas, las referencias bibliográficas y la organización del documento.

El resto de este documento está organizado, en la parte de requisitos según el estándar IEEE 830-1998 de prácticas recomendadas para la Especificación de Requisitos Software [2], y en la parte de análisis y diseño según la plantilla del artefacto de Análisis y Diseño Orientado a Recursos (en inglés, ROA/D) [1].

De esta manera, en la sección 2 se introduce el contexto de los requisitos del Servicio de Colas del sistema DNSE3 que se pretende desarrollar, mientras que en la sección 3 se describen con detalle dichos requisitos. En la sección 4 se describe qué funciones hace el servicio y qué dominio va a tener, concretando los escenarios de uso, el modelo de dominio y las secuencias del sistema. Finalmente, en la sección 5 se describe cómo está organizado el servicio y cómo se implementarán los escenarios de uso de la sección anterior, definiendo el diagrama de recursos, el espacio de URIs, describiendo los métodos que va a tener, los códigos de respuesta y error, así como las representaciones que va a aceptar y generar.

2. Descripción general

La descripción general del servicio permite al lector obtener una visión general del propósito y funciones que realizará el servicio, así como las características de los usuarios, las restricciones, suposiciones y dependencias que deberá tener, y los posibles requisitos futuros.

2.1. Perspectiva del producto

El servicio interactuará con un único tipo de elementos software, a saber:

- **Cientes:** realizarán la gestión de la cola de elementos, por una parte insertándolos para su procesado, consultando su estado final y eliminándolo, y por otra parte solicitando los pendientes de procesar y modificando aquellos en procesamiento. En el DNSE3, los clientes serán, en primer lugar el Servicio de Orquestación relleno de la cola, y en segundo lugar los Servicios de Simulación procesando la cola.

El servicio ofrecerá una interfaz programática por la cual los clientes podrán acceder para interactuar con el servicio. También ofrecerá una interfaz por línea de comandos por la cual el administrador del DNSE3 incluirá opciones de configuración en el comando empleado para la ejecución del servicio.

2.2. Funciones del producto

Las principales funciones proporcionadas por el servicio son las siguientes:

- Insertar elementos en la cola para que se realice su procesamiento.
- Solicitar el elemento siguiente de la cola pendiente de procesar y modificar el tiempo de vida de un elemento de la cola que está siendo procesado.
- Consultar uno o todos los elementos de la cola y eliminar un elemento de la cola que ya ha sido procesado de forma satisfactoria.

2.3. Características de los usuarios

Se espera que el usuario encargado de la administración del DNSE3 sepa manejar una interfaz por línea de comandos para la ejecución y configuración del servicio.

2.4. Restricciones

El servicio será diseñado e implementado siguiendo el estilo arquitectural REST [3].

2.5. Suposiciones y dependencias

Se supone que el servicio se ejecutará en una instancia con sistema operativo GNU/Linux ejecutada en una nube computacional basada en la versión 2 o superior de Eucalyptus ¹ o en la versión 2012.1 (Essex) o superior de OpenStack ².

2.6. Requisitos futuros

No se definen requisitos futuros para este servicio.

3. Requisitos específicos

En esta sección se describen los diferentes requisitos específicos que tendrá el Servicio de Colas de la aplicación DNSE3. Estos requisitos específicos están clasificados de la siguiente manera: requisitos de interfaces externas, requisitos de funciones, requisitos de rendimiento, restricciones de diseño, atributos del sistema y otros requisitos no clasificados anteriormente.

¹<http://www.eucalyptus.com/>

²<http://www.openstack.org/>

3.1. Interfaces externas

La interfaz programática del servicio manejará como entrada o salida los datos que se relacionan a continuación:

- **RIE001.** Identificador de un elemento existente en la cola del servicio.
- **RIE002.** Lista de los identificadores de los elementos existentes en la cola del servicio.
- **RIE003.** Mensaje del elemento de la cola con el contenido necesario para su procesado.
- **RIE004.** Tiempo de vida estimado de un elemento de la cola para realizar su procesado en el que deberá finalizar, o pasado ese tiempo volverá a estar pendiente de procesar.
- **RIE005.** Estado de procesado en el que se encuentra un elemento de la cola (pendiente, en ejecución o terminado).
- **RIE006.** Confirmación de la acción realizada sobre un elemento de la cola del servicio.

A su vez, la interfaz de comandos del servicio manejará como entrada o salida los datos que se relacionan a continuación:

- **RIE007.** Número del puerto usado por el servicio.
- **RIE008.** URI del Servicio de Monitorización con el que interactuará este servicio.

3.2. Funciones

Las funciones que realizará el servicio a través de la interfaz programática son las siguientes:

- **REF001.** Listar los elementos existentes en la cola del servicio.
 - **Entrada:** ninguna.
 - **Salida:** RIE002.
 - **Prioridad:** alta.
- **REF002.** Añadir un elemento a la cola del servicio.
 - **Entrada:** RIE003, RIE004.
 - **Salida:** RIE001.
 - **Prioridad:** alta.
 - **Observaciones:** el nuevo elemento se añadirá al final de la cola definida en el servicio.
- **REF003.** Ver la descripción de un elemento de la cola del servicio.
 - **Entrada:** RIE001.
 - **Salida:** RIE001, RIE003, RIE004, RIE005.
 - **Prioridad:** alta.
- **REF004.** Aumentar el tiempo de vida de un elemento de la cola del servicio.
 - **Entrada:** RIE001, RIE004.
 - **Salida:** RIE006.
 - **Prioridad:** alta.
- **REF005.** Eliminar un elemento de la cola del servicio.
 - **Entrada:** RIE001.
 - **Salida:** RIE006.
 - **Prioridad:** alta.
- **REF006.** Solicitar un elemento de la cola del servicio.

- **Entrada:** ninguna.
- **Salida:** RIE001.
- **Prioridad:** alta.
- **Observaciones:** se ofrecerá el siguiente elemento del inicio de la cola del servicio.

A su vez, las funciones que realizará el servicio a través de la interfaz de comandos son las siguientes:

- **REF007.** Indicar el puerto en el que escuchará el servicio y el Servicio de Monitorización con el que interactuará.
 - **Entrada:** RIE007, RIE008.
 - **Salida:** ninguna.
 - **Prioridad:** alta.

3.3. Requisitos de rendimiento

No se definen requisitos de rendimiento.

3.4. Restricciones de diseño

Las restricciones de diseño que cumplirá el servicio son las siguientes:

- **RDD001.** El servicio sera diseñado de acuerdo con el estilo arquitectural REST [3].

3.5. Atributos del sistema

Los atributos que tendrá el sistema son los siguientes:

- **ADS001.** La implementación se llevará a cabo empleando la versión más reciente del marco Restlet [4] para desarrollo de servicios web REST basados en Java.

3.6. Otros requisitos

No se definen otros requisitos.

4. Análisis

El siguiente paso de la metodología ROA/D es realizar el análisis de los requisitos recogidos en las secciones anteriores para entender el problema del dominio y presentarlos con artefactos más prácticos. El objetivo de esta sección es investigar y describir cómo se usará el sistema y cuáles son las principales funciones, así como la posibilidad de identificar nuevos requisitos.

En esta sección, primero se describen los principales escenarios de uso junto con un diagrama de casos de uso. A su vez, se representa el modelo de dominio junto con el diagrama de clases correspondiente. Finalmente se describen las principales secuencias del sistema, correspondientes a los escenarios de uso, junto con sus diagramas de secuencia.

4.1. Escenarios de uso

Una buena ayuda para explicar qué se espera que haga el sistema es la representación del mismo como una caja negra en el que se describe cómo se interacciona con él. Para ello, se definen los principales escenarios de uso de forma textual y se complementa con un diagrama de casos de uso en el que se muestra visualmente la interacción del cliente con el Servicio de Colas (llamado *DNSE3Queue* de forma abreviada).

La interacción con el servicio *DNSE3Queue* va a estar únicamente hecha por un actor: el cliente. Este cliente puede referirse a otros servicios del sistema, servicios externos, o el sistema operativo de la instancia de la nube computacional.

La descripción de los principales escenarios de uso del servicio son los siguientes:

- **ESC001. Listar los elementos de la cola:** el cliente solicitará el listado de todos los elementos de los que está compuesta la cola del servicio en ese momento, ya estén pendientes de procesar, en procesamiento o procesados.
- **ESC002. Añadir un elemento a la cola:** el cliente solicitará la inserción de un elemento en la cola del servicio, que será incluido al final de la cola y con un estado pendiente de procesar. Se deberá adjuntar el contenido del mensaje a procesar y el tiempo de vida que se estima que tardará su procesamiento.
- **ESC003. Ver la descripción de un elemento:** el cliente solicitará la descripción de un elemento de la cola del servicio, independientemente del estado en el que se encuentre, y del que podrá conocer el contenido del mensaje, el tiempo de vida actual y el estado en el que se encuentra en ese momento.
- **ESC004. Aumentar el tiempo de vida de un elemento:** el cliente solicitará la actualización del tiempo de vida de un elemento de la cola que esté procesándose en ese momento para que el servicio no suponga que dicho procesamiento no ha concluido con éxito y vuelva a asignarlo como pendiente de procesar nuevamente.
- **ESC005. Eliminar un elemento de la cola:** el cliente solicitará la eliminación de un elemento de la cola, independientemente de su estado.
- **ESC006. Solicitar un elemento de la cola:** el cliente solicitará el siguiente elemento de la cola que esté pendiente de ser procesado y que corresponda según el algoritmo de gestión de la cola que se implemente.
- **ESC007. Configurar el servicio:** el cliente solicitará, desde la interfaz de comandos, la configuración del servicio indicando el puerto de escucha al inicio de su ejecución y el Servicio de Monitorización con el que interactuará el servicio.

El diagrama de casos de uso que recoge todos los escenarios de uso anteriores y su interacción con los actores del sistema está recogido en la siguiente Figura 1:

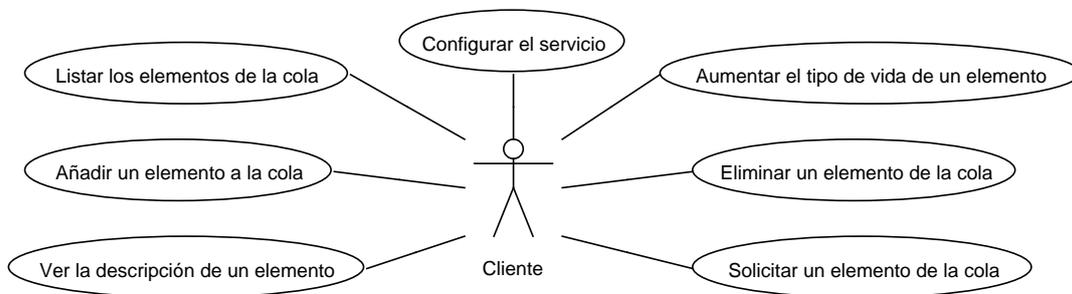


Figura 1: Diagrama de casos de uso de los escenarios de uso del servicio.

4.2. Modelo de dominio

El modelo de dominio es uno de los artefactos más importantes en el análisis, puesto que define las bases de los modelos más concretos que se definirán en la sección de diseño. Con este modelo se identifican las principales entidades del dominio y sus relaciones entre sí. Las entidades y sus propiedades se identifican a partir de los requisitos textuales definidos en la sección 3.

Por tanto, las entidades, propiedades y relaciones entre sí que se han identificado para este servicio están representadas en la Figura 2 y son las siguientes:

- **DNSE3Queue:** es la entidad que define el servicio, donde se define la configuración del puerto en el que escuchará y el Servicio de Monitorización donde se registrará y publicará sus métricas, y desde la que partirán el resto de recursos.

- **Elements:** es la entidad que representa a la cola del servicio y que tiene un listado con todos los identificadores de los elementos que hay actualmente en la cola, independientemente de su estado de procesamiento.
- **Element:** es la entidad que representa a un elemento de la cola, el cual estará compuesto por un identificador único, un mensaje con el contenido a procesar, un tiempo de vida estimado para su procesamiento (o actualizado durante su procesado) y un estado actual que puede ser pendiente, en ejecución o finalizado.

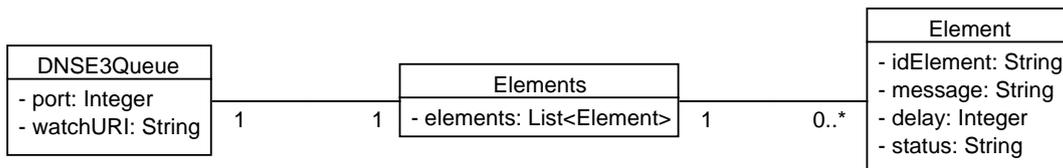


Figura 2: Diagrama de modelo de dominio del servicio.

4.3. Secuencias del sistema

Las secuencias del sistema complementan lo descrito en el diagrama de casos de uso, puesto que éstos identifican los actores y la lista de los principales usos de la aplicación, pero no describen de forma precisa los eventos de entrada/salida y las secuencias de las acciones.

En los siguientes puntos se describen las secuencias del sistema para cada uno de los escenarios de uso anteriores y se complementan con su correspondiente diagrama de secuencia:

- **ESC001. Listar los elementos de la cola:** el cliente solicitará al servicio DNSE3Queue el listado de todos los elementos que hay actualmente en la cola, independientemente de su estado. El servicio responderá con el listado de los identificadores de todos los elementos de la cola. Este comportamiento está descrito en el diagrama de secuencia de la Figura 3.

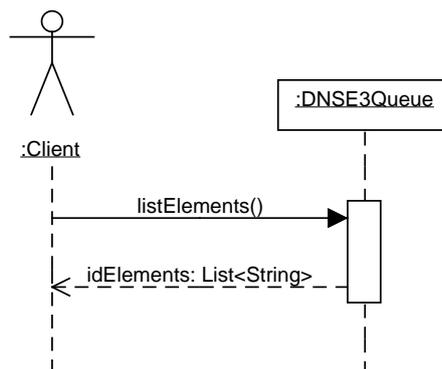


Figura 3: Diagrama de secuencia del escenario de uso ESC001.

- **ESC002. Añadir un elemento a la cola:** el cliente solicitará al servicio DNSE3Queue añadir un nuevo elemento a la cola del servicio, indicando para ello el contenido del mensaje a procesar y el tiempo de vida que se estima para su procesado. El servicio responderá con el identificador único que ha asignado a ese nuevo elemento de la cola. Este comportamiento está descrito en el diagrama de secuencia de la Figura 4.

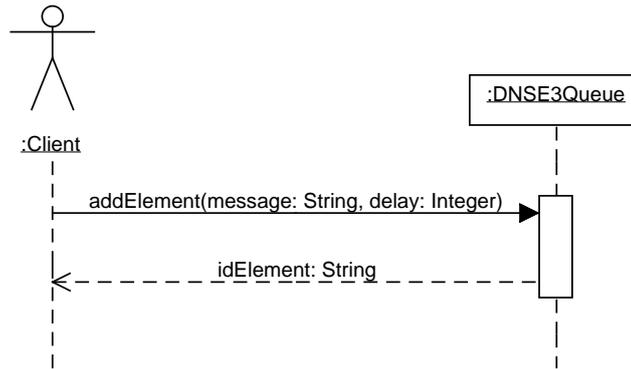


Figura 4: Diagrama de secuencia del escenario de uso ESC002.

- **ESC003. Ver la descripción de un elemento:** el cliente solicitará al servicio DNSE3Queue la descripción completa de uno de los elementos de la cola del servicio, indicando para ello su identificador único. El servicio responderá con el identificador único del elemento, el contenido del mensaje a procesar, el tiempo de vida actual de procesado que está estimado y el estado en el que se encuentra (pendiente, en ejecución o finalizado). Este comportamiento está descrito en el diagrama de secuencia de la Figura 5.

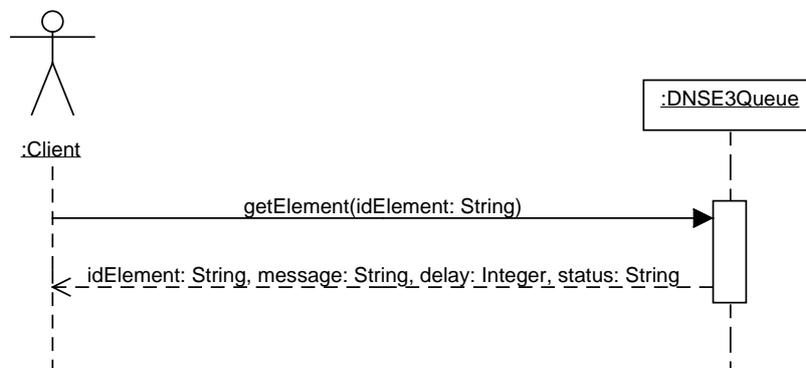


Figura 5: Diagrama de secuencia del escenario de uso ESC003.

- **ESC004. Aumentar el tiempo de vida de un elemento:** el cliente solicitará al servicio DNSE3Queue la actualización del tiempo de vida de un elemento de la cola del servicio que esté en ese instante en ejecución, para aumentarlo y que no se suponga un elemento *huérfano* de procesamiento. Para ello se indicará el identificador único del elemento y el nuevo tiempo de vida estimado. El servicio responderá con la confirmación de la actualización del elemento. Este comportamiento está descrito en el diagrama de secuencia de la Figura 6.

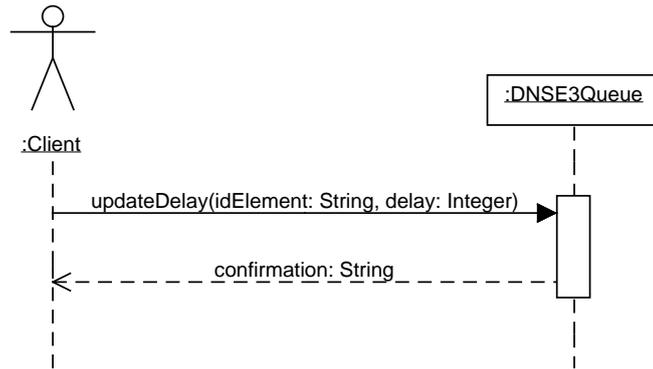


Figura 6: Diagrama de secuencia del escenario de uso ESC004.

- **ESC005. Eliminar un elemento de la cola:** el cliente solicitará al servicio DNSE3Queue la eliminación de un elemento de la cola del servicio independientemente del estado en el que se encuentre, para lo cual indicará el identificador único del elemento a eliminar. El servicio responderá con la confirmación de la eliminación del elemento. Este comportamiento está descrito en el diagrama de secuencia de la Figura 7.

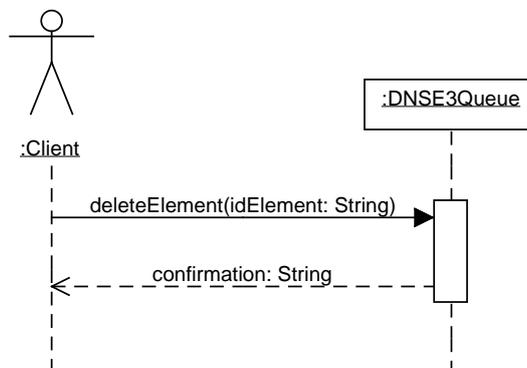


Figura 7: Diagrama de secuencia del escenario de uso ESC005.

- **ESC006. Solicitar un elemento de la cola:** el cliente solicitará al servicio DNSE3Queue cuál es el siguiente elemento de la cola de todos aquellos que están pendientes de procesar. El servicio responderá con el identificador único del elemento que, según el algoritmo de gestión de cola, estime oportuno. Este comportamiento está descrito en el diagrama de secuencia de la Figura 8.

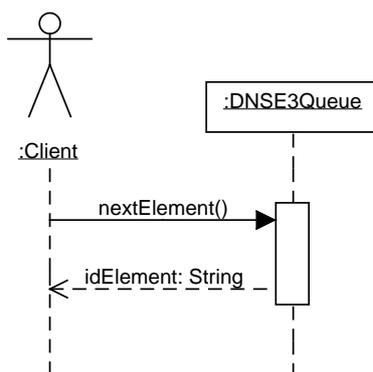


Figura 8: Diagrama de secuencia del escenario de uso ESC006.

- **ESC007. Configurar el servicio:** el cliente, por línea de comandos, indicará al servicio DNSE3Queue en el inicio el puerto por el que escuchará las peticiones Web a través del protocolo HTTP y el Servicio de Monitorización con el que interactuará para registrarse y publicar sus métricas del estado de la cola. El servicio no responderá con nada a la indicación. Este comportamiento está descrito en el diagrama de secuencia de la Figura 9.

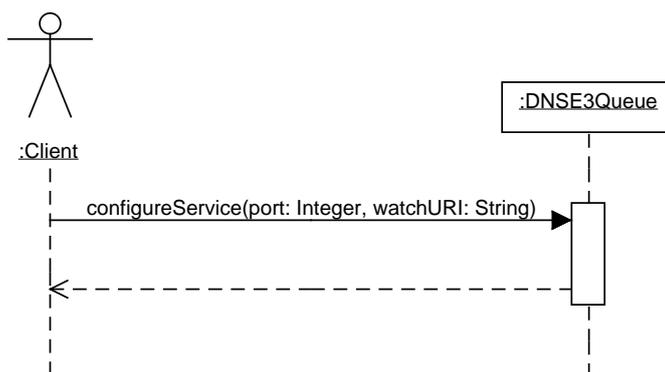


Figura 9: Diagrama de secuencia del escenario de uso ESC007.

5. Diseño

Tras la especificación del sistema en las secciones de requisitos y análisis, definiendo las necesidades y entendiendo y describiendo lo que hace, la metodología ROA/D continua con la fase del diseño de la solución. En esta fase se han de tener muy en cuenta las especificaciones del paradigma de la arquitectura orientada a recursos (en inglés, ROA), debiendo tener en mente una solución e implementación que siga dicho paradigma. En esta sección se presenta el diseño de la solución según la metodología ROA/D, que se usará como una guía para la implementación del servicio.

La metodología ROA/D define una arquitectura para los servicios reutilizable para la mayoría de los servicios que se definen a través de ella, como es el caso. En esta definición de la arquitectura se utilizan dos representaciones: la arquitectura lógica y la arquitectura topológica.

La arquitectura lógica que sigue la metodología ROA/D se basa en cuatro capas:

- La capa más superior es la de *interfaz de usuario* que respete los principios de REST. Será la capa con la que interactúen los usuarios y que, en el caso específico de este servicio, será una interfaz Web que usarán los usuarios y que interactuará con este servicio o lo realizará a través del Servicio de Orquestación actuando como un *proxy* de la comunicación.

- La siguiente capa es la de *recursos* sobre la que se realizarán las transferencias de representaciones invocando los métodos HTTP sobre los recursos. Esta capa se implementará con el marco Restlet [4] siguiendo los principios de REST. Las responsabilidades de esta capa son la identificación de recursos a través de URIs, el formateo y parseo de las representaciones de los recursos y el enrutamiento de las llamadas para un refuerzo de la seguridad. A través de esta capa se ofrece una interfaz programática para la interacción con otros clientes.
- La siguiente capa es la de *objetos*, previa a la capa de datos, que define el modelo de datos del sistema y en el que se usa una programación orientada a objetos (POO), por ejemplo, con Java. En esta capa se definen los objetos y la lógica asociada a estos objetos. También permite la flexibilización y extensión del modelo de dominio.
- La capa más inferior es la de *datos* que es la que se encarga de la persistencia de los objetos de la capa de objetos e, indirectamente, de la capa de recursos. La opción más frecuente para la persistencia (y que se utilizará en este servicio) es el uso de bases de datos y sistemas de ficheros.

Una de las ventajas de la división en capas de la arquitectura del servicio es la baja cohesión que tienen entre sí dichas capas, de forma que se puede modificar una de ellas sin que afecte al resto y que se pueden reutilizar para otros servicios. Estas capas son más específicas cuanto más altas son y más genéricas (facilitando la reutilización) cuanto más bajas son.

En una arquitectura topológica del servicio, la capa de interfaz de usuario estará localizada en la parte del cliente, mientras que el resto de capas estarán localizadas en la parte del servidor.

5.1. Diagrama de recursos

Teniendo clara la arquitectura del servicio, el siguiente paso es la definición del diagrama de recursos. Este diagrama se obtiene a partir del modelo de dominio de la fase de análisis, en el que se identifican aquellos objetos representados en el modelo de dominio que pasarán a formar parte como recursos del servicio en un proceso llamado *Resource-Object Mapping* (ROM). Tras este proceso se obtendrá el diseño de la API web de RESTful del servicio en el que se identifican sus recursos, su enrutamiento, los métodos para su interacción y la clasificación entre ellos.

Por tanto, basándose en el modelo de dominio de la subsección 4.2 se pueden identificar las siguientes clases de recursos para el servicio:

- **DNSE3Queue:** es el recurso principal y de partida del Servicio de Colas.
- **Elements:** representa a la cola del servicio que contiene todos los elementos de la misma y dependerá del recurso principal *DNSE3Queue*. Este recurso permitirá obtener la lista de todos los elementos dependientes de él y crear un nuevo elemento que dependa de él.
- **Element:** representa a un elemento de la cola del servicio y siempre dependerá del recurso *Elements* anterior. Este recurso permitirá obtener su descripción, modificar su tiempo de vida o ser eliminado.
- **NextElement:** representa al siguiente elemento de la cola del servicio que está pendiente de ser procesado y dependerá del recurso *Elements*. Este recurso permitirá obtener el identificador único del siguiente elemento pendiente de procesar.

En el diagrama de recursos de la Figura 10 están representados de forma visual todos los recursos anteriores, el esquema de identificación único de cada uno de ellos, los métodos para su interacción y la interrelación entre sí en una estructura jerárquica de recursos.

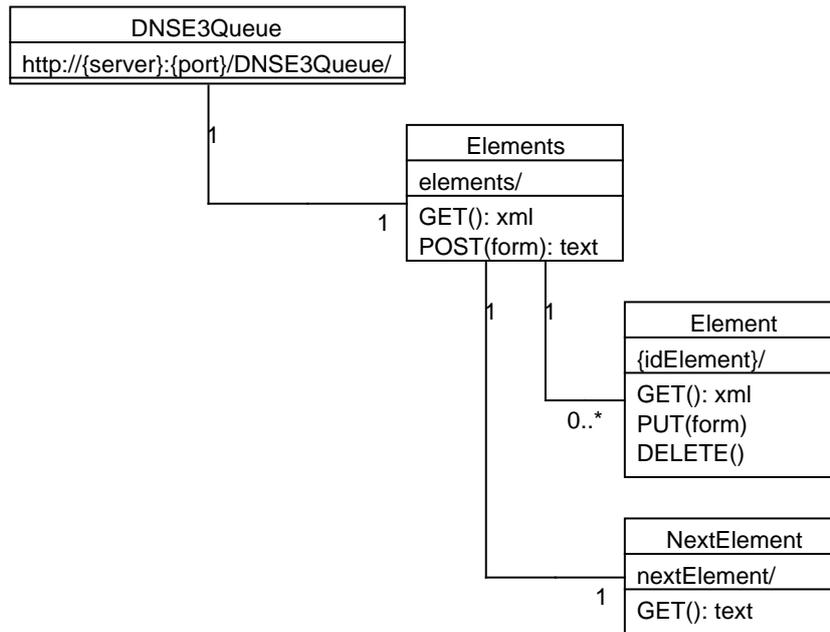


Figura 10: Diagrama de recursos del servicio.

5.2. Espacio de URIs

Ya visto en el diagrama de recursos, todos los recursos deberán ser accesibles de forma pública a través de una URI única disponible por HTTP. Esta URI tendrá la información suficiente para identificar la estructura de la red para encontrar el servidor que aloja el recurso e identificar el recurso en él.

Por tanto, las URIs de acceso a los recursos del servicio son las siguientes:

- `http://{server}:{port}/DNSE3Queue`
Representa a la raíz del Servicio de Colas.
- `http://{server}:{port}/DNSE3Queue/elements`
Representa a la cola del servicio y contendrá la totalidad de los elementos de dicha cola, independientemente del estado en el que se encuentren.
- `http://{server}:{port}/DNSE3Queue/elements/{idElement}`
Representa a un elemento de la cola del servicio y contendrá toda la información de dicho elemento.
- `http://{server}:{port}/DNSE3Queue/elements/nextElement`
Representa al siguiente elemento de la cola del servicio que está pendiente de procesar y contendrá el identificador único de dicho elemento.

5.3. Descripción de métodos y códigos de respuesta y error

El protocolo HTTP define un conjunto de operaciones predefinidas o métodos que permiten la manipulación de los recursos. Los métodos más comunes (y utilizados en este servicio) son cuatro: GET para obtener un recurso, POST para crear un nuevo recurso, PUT para actualizar un recurso y DELETE para eliminar un recurso.

Estos métodos siempre van a generar un código de estado tras su ejecución que será una información muy importante para la comunicación entre el cliente y el servidor del servicio. Estos códigos de estados están codificados con tres números y representan a un mensaje con el estado del recurso, en el que indican al cliente que dicho recurso se ha procesado de forma satisfactoria o ha ocurrido un error que se deberá subsanar.

En el siguiente listado están clasificados, por cada recurso, cada uno de los métodos ya identificados en la Figura 10 del diagrama de recursos, y se ofrece una descripción del comportamiento de dicho método, así como los códigos de respuesta y error que se generarán tras su procesamiento:

- Métodos para el recurso “*Elements*”:
 - **GET(): xml**
 - **Descripción:** se listan todos los identificadores de los elementos que están actualmente en la cola.
 - **Códigos de respuesta:**
 - ◇ SUCCESSFUL - OK (200), se ha generado el XML de la petición.
 - **Códigos de error:**
 - ◇ SERVER ERROR - INTERNAL SERVER ERROR (500), error interno del servidor al procesar la petición.
 - **POST(form): text**
 - **Descripción:** se añade un nuevo elemento a la cola.
 - **Códigos de respuesta:**
 - ◇ SUCCESSFUL - ACCEPTED (202), se ha aceptado el nuevo elemento.
 - **Códigos de error:**
 - ◇ CLIENT ERROR - BAD REQUEST (400), los nombres o valores de los parámetros enviados no son correctos.
 - ◇ SERVER ERROR - INTERNAL SERVER ERROR (500), error interno del servidor al procesar la petición.
- Métodos para el recurso “*Element*”:
 - **GET(): xml**
 - **Descripción:** se obtiene la descripción detallada de un elemento de la cola: identificador, estado, mensaje y tiempo de vida.
 - **Códigos de respuesta:**
 - ◇ SUCCESSFUL - OK (200), se ha generado el XML de la petición.
 - **Códigos de error:**
 - ◇ CLIENT ERROR - NOT FOUND (404), no se ha encontrado ningún elemento con ese identificador.
 - ◇ SERVER ERROR - INTERNAL SERVER ERROR (500), error interno del servidor al procesar la petición.
 - **PUT(form)**
 - **Descripción:** se actualiza el tiempo de vida de un elemento de la cola.
 - **Códigos de respuesta:**
 - ◇ SUCCESSFUL - OK (200), se ha actualizado el elemento.
 - **Códigos de error:**
 - ◇ CLIENT ERROR - BAD REQUEST (400), los nombres o valores de los parámetros enviados no son correctos.
 - ◇ CLIENT ERROR - NOT FOUND (404), no se ha encontrado ningún elemento con ese identificador.
 - ◇ SERVER ERROR - INTERNAL SERVER ERROR (500), error interno del servidor al procesar la petición.
 - **DELETE()**
 - **Descripción:** se elimina un elemento de la cola.
 - **Códigos de respuesta:**
 - ◇ SUCCESSFUL - OK (200), se ha eliminado el elemento.
 - **Códigos de error:**
 - ◇ CLIENT ERROR - NOT FOUND (404), no se ha encontrado ningún elemento con ese identificador.

- ◊ SERVER ERROR - INTERNAL SERVER ERROR (500), error interno del servidor al procesar la petición.
- Métodos para el recurso “*NextElement*”:
 - **GET(): text**
 - **Descripción:** se obtiene la URI del siguiente elemento en estado pendiente de ejecución de la cola.
 - **Códigos de respuesta:**
 - ◊ SUCCESSFUL - OK (200), se ha devuelto la URI del elemento.
 - ◊ SUCCESSFUL - NO CONTENT (204), no hay más elementos pendientes en la cola.
 - **Códigos de error:**
 - ◊ SERVER ERROR - INTERNAL SERVER ERROR (500), error interno del servidor al procesar la petición.

5.4. Representaciones aceptadas y generadas

Algunas de las peticiones o respuestas HTTP de los métodos anteriores de los recursos contienen una representación de dichos recursos. Una representación se define como una serie de datos que describen el estado actual o deseable de un recurso. Por ello, es importante definir, diseñar y documentar un formato común para cada una de estas representaciones de cada método.

Por tanto, en el siguiente listado, clasificado por recursos, se definen, diseñan y documentan las representaciones aceptadas y generadas por cada método de los recursos del servicio:

- Métodos para el recurso “*Elements*”:
 - **GET(): xml**
 - **Representación aceptada:** ninguna.
 - **Representación generada:** un documento en formato XML con la siguiente estructura:


```

<DNSE3Queue>
  <elements>
    <element>
      <identifier>{idElement}</identifier>
    </element>
    ...
  </elements>
</DNSE3Queue>
```
 - **POST(form): text**
 - **Representación aceptada:** un formulario con los siguientes campos:
 - ◊ *message*: mensaje de texto (texto plano, documento XML, etc.) con el contenido a procesar.
 - ◊ *delay*: tiempo de vida, esto es, tiempo previsto de procesado del elemento.
 - **Representación generada:** un texto con la URI del nuevo elemento en la que se incluye su identificador único.

- Métodos para el recurso “*Element*”:
 - **GET(): xml**
 - **Representación aceptada:** ninguna.
 - **Representación generada:** un documento XML con la siguiente estructura:


```

<DNSE3Queue>
  <elements>
    <element>
      <identifier>{idElement}</identifier>
```

```
<message>{message}</message>
<delay>{delay}</delay>
<status>{status}</status>
</element>
</elements>
</DNSE3Queue>
```

- **PUT(form)**
 - **Representación aceptada:** un formulario con el siguiente campo:
 - ◊ *delay*: tiempo de vida, esto es, tiempo previsto de procesado del elemento.
 - **Representación generada:** ninguna.
- **DELETE()**
 - **Representación aceptada:** ninguna.
 - **Representación generada:** ninguna.
- Métodos para el recurso “*NextElement*”:
 - **GET(): xml**
 - **Representación aceptada:** ninguna.
 - **Representación generada:** un texto con la URI del siguiente elemento de la cola pendiente de procesar.