

Trabajo de grado

**Herramienta de Particionamiento de Código para Sistemas Monolíticos en
Microservicios**

**Nelson David Jiménez Ortiz
Santos David Núñez Villamil
Juan Sebastián Prado Valero
Gustavo Antonio Rivera Delgado**

Entrega SDD

**Presentado a:
Jaime Andrés Pavlich Mariscal**

**Pontificia Universidad Javeriana
Facultad de Ingeniería
Departamento de Ingeniería de Sistemas
2020**

1 Historial de Cambios

Fecha de cambio	Descripción	Sección afectada	Autores
23/04/2021	Se genero el documento		Santos Núñez y Gustavo Rivera
29/04/2021	Se agrego la vista lógica del sistema	7.1	Santos Núñez y Gustavo Rivera
30/04/2021	Se agregaron diagramas de procesos de la herramienta	8	Santos Núñez
06/05/2021	Diagrama de despliegue del monolito y microservicios	7.5	Santos Núñez y Gustavo Rivera
07/05/2021	Se agrego el diagrama de clase	8.1	Santos Núñez y Gustavo Rivera
08/05/2021	Se agrego la sección de persistencia	8.3	Santos Núñez y Gustavo Rivera
08/05/2021	Se agregaron los diagramas de secuencia	8.2	Santos Núñez y Gustavo Rivera
10/05/2021	Se agrego la introducción	6	Gustavo Rivera
10/05/2021	Se agrego el Resumen	2	Santos Nuñez

Tabla 1 Historial de cambios

2 Resumen

En este documento se encontrará una descripción detallada de la manera en que se encuentra organizada la herramienta de particionamiento. Por esta razón encontramos una descripción detallada de partes del sistema que son importantes para cumplir este objetivo como la estructura y arquitectura de este. Se puede comprender de forma detallada la Arquitectura del sistema desde las vistas lógica y de procesos, así como un diseño detallado de su estructura, comportamiento y persistencia de datos. Así se conforma la Descripción de Diseño de Software (SDD) para estudiar y comprender el sistema que compone la herramienta de partición.

3 Tabla de Contenidos

Contenido

1	Historial de Cambios	2
	Se agrego el diagrama de clase.....	2
	Se agrego la sección de persistencia.....	2
	Se agregaron los diagramas de secuencia.....	2
	Se agrego la introducción	2
	Se agrego el Resumen.....	2
2	Resumen	2
3	Tabla de Contenidos	3
4	Lista de Figuras	4
5	Lista de Tablas	5
6	Introducción	1
7	Arquitectura	2
7.1	Vista lógica del sistema	2
7.2	Vista física del sistema.....	3
7.3	Vista de procesos del sistema	3
7.3.1	Proceso general.....	4
7.3.2	Herramienta de partición	4
7.3.3	Lectura del grafo.....	5
7.3.4	Creación de Microservices-register	6
7.3.5	Creación de los N-microservicios.....	6
7.3.6	Creación de Microservices-web	7
7.3.7	Generación de Servicios	7
7.3.8	Generación de controladores	7
7.4	Diagrama de despliegue del monolito y microservicios	9
8	Diseño Detallado	10
8.1	Estructura del sistema	10
8.2	Comportamiento del sistema.....	10
8.3	Persistencia	12
9	Anexos	12

4 Lista de Figuras

Figura 1 Diagrama de paquetes de la herramienta de particionamiento	2
Figura 2 Diagrama de proceso general	4
Figura 3 Diagrama de proceso general de la herramienta de particionamiento	5
Figura 4 Diagrama de proceso de lectura del grafo	5
Figura 5 Diagrama de creación del microservicio register	6
Figura 6 Diagrama de proceso creación de los n microservicios	7
Figura 7 Diagrama de proceso de creación del microservicio web	8
Figura 8 Diagrama de proceso para generar los servicios	6
Figura 9 Diagrama de proceso de generar los controladores	7
Figura 10 Diagrama de despliegue del monolito a particionar	9
Figura 11 Diagrama de despliegue de los microservicios generados	9
Figura 12 Diagrama de clases de la herramienta de particionamiento	10
Figura 13 Diagrama de secuencia creación microservicios	11
Figura 14 Diagrama de secuencia crear el microservicio register	12
Figura 15 Diagrama de secuencia crear microservicio web	12

5 Lista de Tablas

Tabla 1 Historial de cambios	2
------------------------------------	---

6 Introducción

Este documento comprende una descripción detallada del diseño de la herramienta de particionamiento que el equipo de trabajo de grado desarrollo el primer prototipo funcional y el cual hace parte del proyecto marco. Se presentan algunos diagramas que permiten tener un mayor acercamiento del funcionamiento. Se presenta algunos componentes tanto físicos como lógicos para representar tanto el diseño de la herramienta como la transformación del monolito.

El propósito de este documento es brindar una visa general de la arquitectura que fue implementada para el desarrollo de la herramienta de particionamiento. Con lo que se busca detallar aspectos importantes en las estructuras que componen este proyecto desde una vista lógica de los procesos realizados por la misma; podemos encontrar diagrama que representa este esquema, como de paquetes para comprender la arquitectura planteada.

Se cuenta con una descripción de los principales procesos, con el fin de ampliar o dar más detalle a las funciones que realizan dentro de la herramienta y la interacción que tienen entre los mismos.

La importancia de este documento se centra en la comprensión detallada de la estructura del sistema y como está construida para su correcto funcionamiento, centrándose en explicar y comprender los componentes y arquitecturas tanto físicos y lógicos que dan pie a la conformación completa de un sistema funciona. Entender profundamente el rol que desempeña cada componente y como se comunican entre los mismos para satisfacer las principales necesidades.

7 Arquitectura

7.1 Vista lógica del sistema

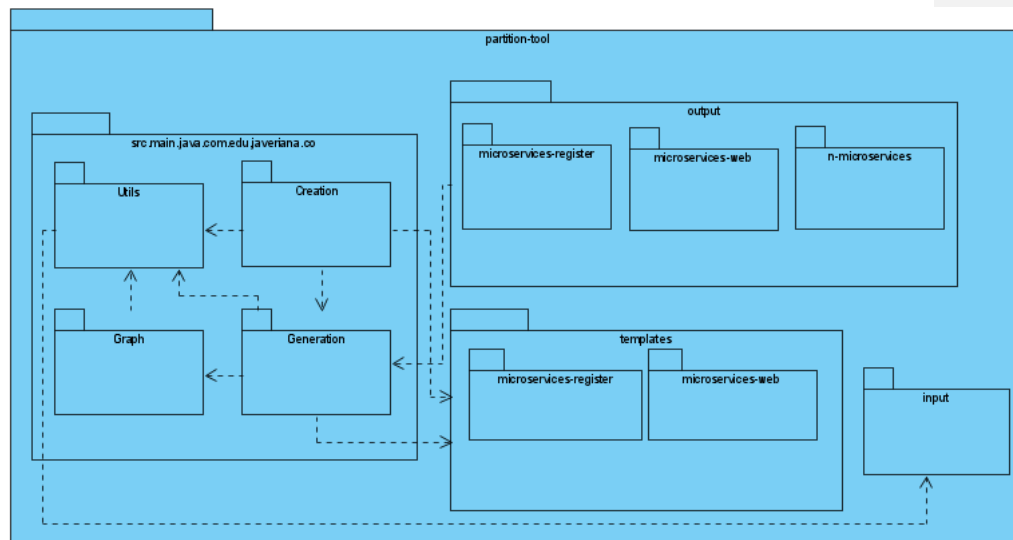


Figura 1 Diagrama de paquetes de la herramienta de particionamiento

1. **Src.main.java.com.edu.javeriana.co:** Paquete donde se encuentran: un paquete de creación que contiene las clases que se encargan de crear los n microservicios y el microservicio web y register. Un segundo Paquete para generación que tienen las clases para generación de código, y algunos archivos como dockerFiles o Pom.xml. Un tercer paquete de Utilidades que contiene las clases que facilitan el manejo de archivos xml, xlsx, y generar llamados por consola. Por último un paquete que contiene las clases que representan el grafo y sus elementos.
2. **Input:** Paquete que contiene las entradas para realizar el particionamiento, en este caso contiene el grafo en un archivo Excel y a este tiene acceso el paquete de utils que toma la información que es empleada por los demás paquetes de la herramienta.
3. **Templates:** Paquete que contiene los elementos que se consideran estándares y/o bases para la mayoría de los microservicios. Se encuentran la estructura genérica para los microservicios web y register, así como plantillas para generación de los archivos RestConfig, RegistrationServer, pom.xml y application.yml. Son accedidos por las clases de los paquetes de creación y generación para construir elementos de código o archivos para el cada microservicio.

Comentado [SDNV1]: Se modificaron los textos de la sección para explicar mejor la funcionalidad de herramienta

4. **Microservices-register:** Paquete que contiene la estructura del microservices-register, genérico, donde se encuentran tanto los archivos básicos que permiten la interacción con los otros microservicios; como los archivos necesarios para la configuración de la base de datos. También contiene una plantilla del pom.xml, necesaria para para describir el proyecto a construir, sus dependencias y componentes externos, y el orden de construcción de los elementos.
5. **Microservices-web:** Paquete que contiene la estructura y elementos del Microservices-web, que se considera general para todos, donde se encuentran las principales clases como controladores y servicios. Igualmente contiene un pom.xml, con los componentes y dependencias lo suficientemente genéricos.
6. **Ouput:** Paquete donde se ubican todas las salidas generadas por la herramienta, es decir, cada microservicio de la aplicación junto con microservicios web y register.
7. **Microservices-register(output):** Paquete que contiene el Microservices-register, con las configuraciones en las diferentes clases y pom.xml del proyecto nuevo.
8. **Microservices-web(output):** Paquete que contiene el Microservices-web, con las configuraciones en los controladores y servicios del proyecto nuevo.
9. **n-microservice:** Paquete que representa los n microservicios que serán generados con todos los archivos y elementos des salida de las clases del paquete de generación.
10. **Utils:** contiene clases que facilitan el manejo de archivos xml, xlsx y llamados por consola.
11. **Creation:** contiene las clases para creación de los proyectos para cada n-microservicios, microservice-register y microservice-web. Donde se realiza la configuración para cada uno dadas las especificaciones del grafo.
12. **Graph:** Paquete que contiene las clases de Graph, Vertex y Edge las cuales permiten la manipulación del grafo, de sus nodos y aristas; dependiendo de las necesidades para la creación de los microservicios.
13. **Generation:** Paquete que contiene las clases de para la generación de los DockerFiles y las clases para la generación de cada microservicio con sus diferentes clases como controladores, repositorios y servicios.

7.2 Vista de procesos del sistema

En esta sección se encuentran los procesos principales de la herramienta y se muestran en los diagramas de flujo de datos comenzando por el proceso general y desglosando subprocesos en diferentes diagramas:

7.2.1 Proceso general

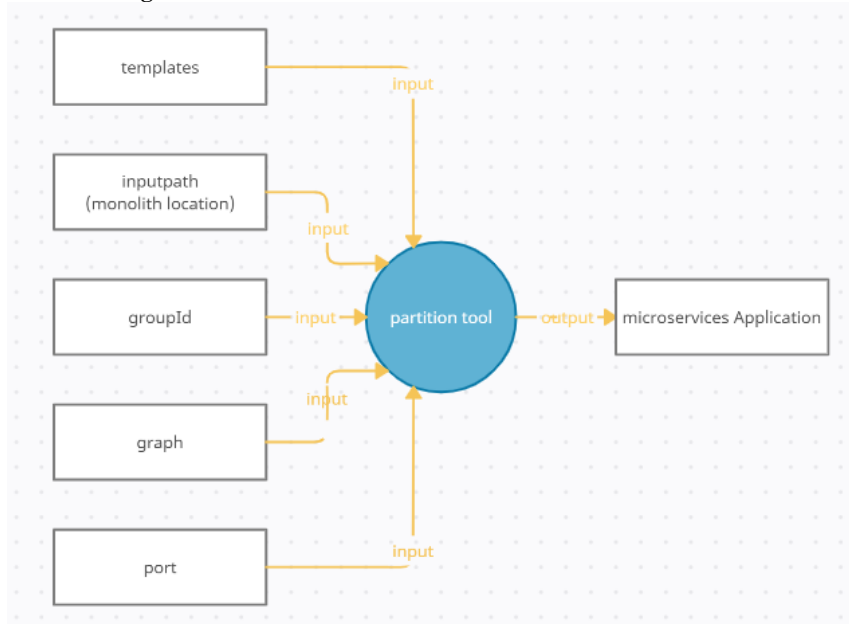


Figura 2 Diagrama de proceso general

Dentro de la herramienta de partición se encuentran los siguientes procesos, comenzando con la lectura de los nodos y conexiones, a partir del grafo que se recibió como entrada. A continuación se crea el Microservices-register a partir de las plantillas que tenemos dentro de la herramienta y se genera el DockerFile para este microservicio. Seguido de esto se realiza la creación de cada microservicio siendo N la cantidad de microservicios del grafo y la generación de sus respectivos DockerFiles. Por último se realiza la construcción de Microservices-web donde al igual que en el Microservices-register se generan archivos a partir de plantillas pero en este caso se generan servicios y controladores.

7.2.2 Herramienta de partición

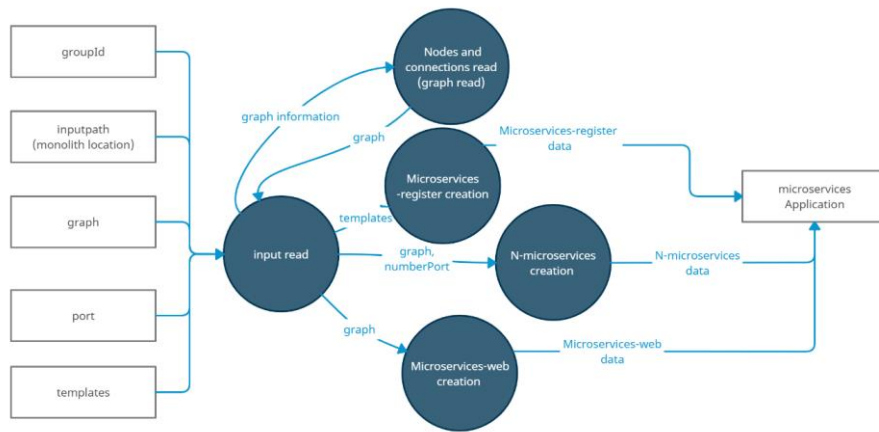


Figura 3 Diagrama de proceso general de la herramienta de particionamiento

El proceso general, a grandes rasgos de la funcionalidad de la herramienta, comienza con la recepción de las entradas al nodo Input Read, para realizar la lectura de los nodos y aristas del grafo en el nodo Nodes and Connections Read. Se retorna el grafo ya estructurado para realizar los diferentes procesos de creación. A continuación, en el nodo de Microservices-register Creation se recibe como entradas los templates y a partir de estos genera el nuevo microservices-register. Después en el nodo N-microservices Creation, se crea cada uno de los n-microservicios especificados en el grafo, por lo que este se manda como parámetro al igual que el número del puerto; como resultado se obtienen todos los microservicios especificados en el grafo. Finalmente, se realiza el proceso del nodo Microservices-web creation, el cual recibe como parámetro el grafo, y como resultado se tiene el nuevo microservice-web con las diferentes configuraciones para el nuevo proyecto.

7.2.3 Lectura del grafo

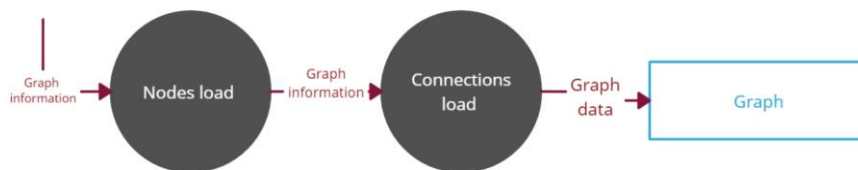


Figura 4 Diagrama de proceso de lectura del grafo

Este proceso recibe la información del grafo, ubicación del archivo Excel de entrada, donde empieza a armar la estructura del grafo, ingresando primero la información de los vértices, proceso que se realiza en el nodo Nodes Load. Siguiendo a esto se realiza la lectura de las aristas, en el nodo Connections load. Para la lectura del archivo Excel cada proceso se apoya de los métodos que provee la clase Excel Utils.

7.2.4 Creación de Microservices-register

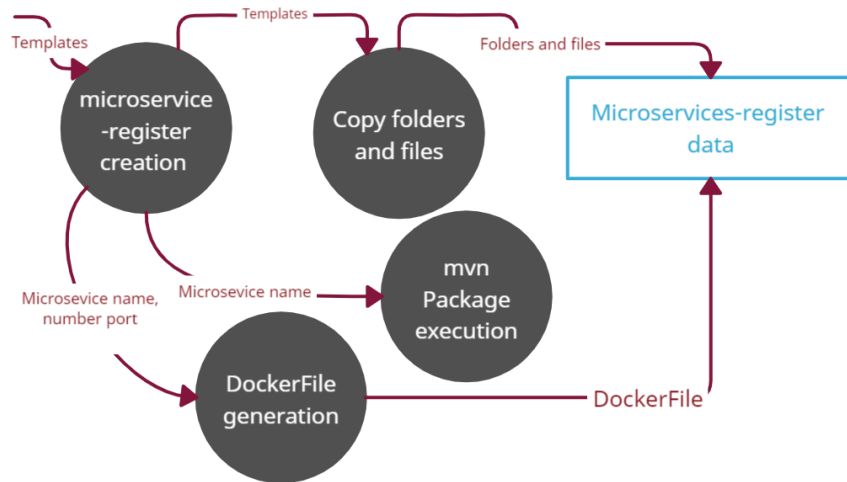


Figura 5 Diagrama de creación del microservicio register

Este proceso, que se realiza en el nodo microservice-register creation, recibe como entradas los templates, una estructura genérica del microservice-register; la cual se emplea en el siguiente nodo, Copy folders and files, donde se copian carpetas y archivos necesarios. Posteriormente, en el nodo DockerFile generation se realiza la creación de su respectivo dockerFile, proceso que recibe el nombre del microservicio y el puerto designado para este. Por último, se realiza la compilación del proyecto en el nodo mvn Package Execution.

7.2.5 Creación de los N-microservicios

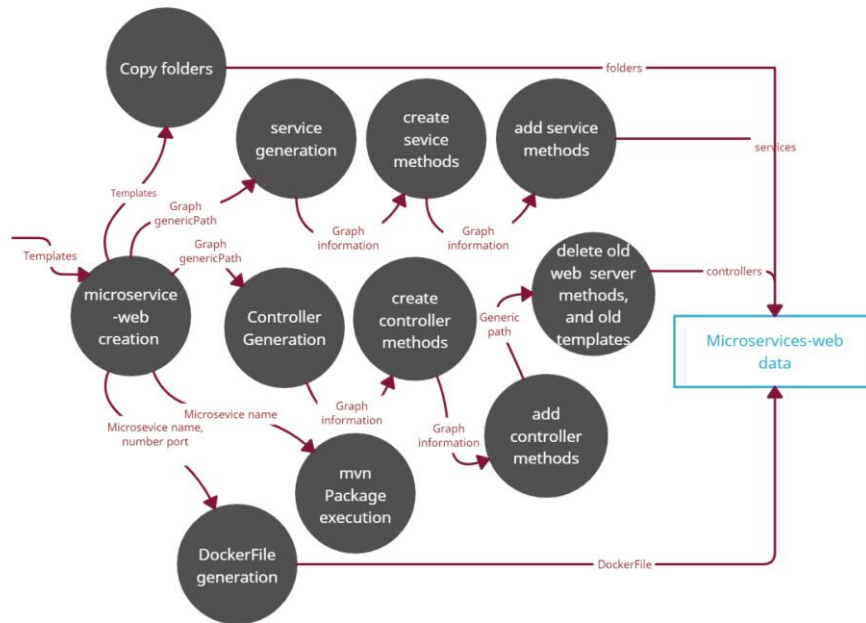


Figura 7 Diagrama de proceso de creación del microservicio web

El siguiente proceso que se muestra en el diagrama, recibe el template con la estructura del microservices-web en el nodo Microservices-web Creation y la envía a los demás nodos que la necesiten junto con el grafo. Primero, se recibe a las plantillas para copiarlas, en el nodo Copy Files. A continuación, se envía información del grafo al nodo Service Generation, donde se van generan los principales servicios; para ello se envía información del grafo al nodo Create Service Methods, en donde se crean los métodos que corresponden a cada servicio; para posteriormente continuar el flujo en el nodo Add Service Methods, donde estos se agregan al servicio. Después de la creación de los servicios se envía información del grafo junto a un path genérico para realizar la creación de los controladores en el nodo Controller Generation; por lo cual se envía esta información al Create Controller Methods, que se encarga de crear los controladores; para después agregarlos en el nodo Add Methods Controller, donde estos métodos son adicionados a la clase controlador; para posteriormente terminar este flujo con la eliminación de aquellos métodos que no son empleados dentro del microservicio.

Seguidamente, en el nodo DockerFile generation se realiza la creación del dockerFile, que recibe el nombre del microservicio y el puerto. Por último, se realiza la compilación del proyecto en el nodo mvn Package Execution.

7.3 Diagrama de despliegue del monolito y microservicios

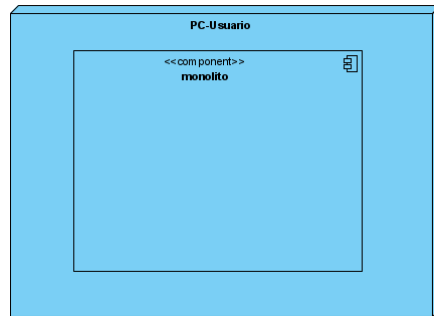


Figura 8 Diagrama de despliegue del monolito a particionar



Transformación mediante la herramienta

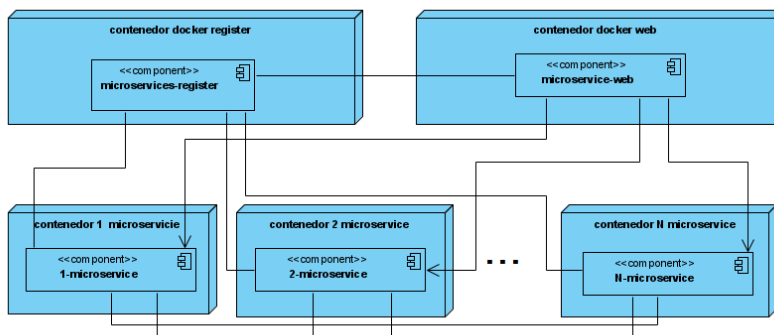


Figura 9 Diagrama de despliegue de los microservicios generados

- ❑ **PC-Usuario:** En este elemento de despliegue se encuentra el monolito, sin particionar. Debido a que es un único componente se despliega en una sola unidad.
- ❑ **Contenedor Docker register:** En esta unidad de despliegue se encuentra el componente del Microservices-register puede funcionar sin necesidad de estar en la misma unidad.
- ❑ **Contenedor Docker web:** En esta unidad de despliegue se encuentra el componente del Microservices-register puede funcionar sin necesidad de estar en la misma unidad.

- 2 **Contenedor n microservice:** En esta unidad se encuentra el componente del microservicio, independiente de la cantidad de estos se despliegan en unidades diferentes.

8 Diseño Detallado

8.1 Estructura del sistema

En esta parte se muestra la estructura de los paquetes modelados en la sección “vista lógica del sistema”. El diagrama de clases, para dar más detalle, se encuentra en el archivo, que se encuentra en anexos, llamado “Estructura del sistema”.

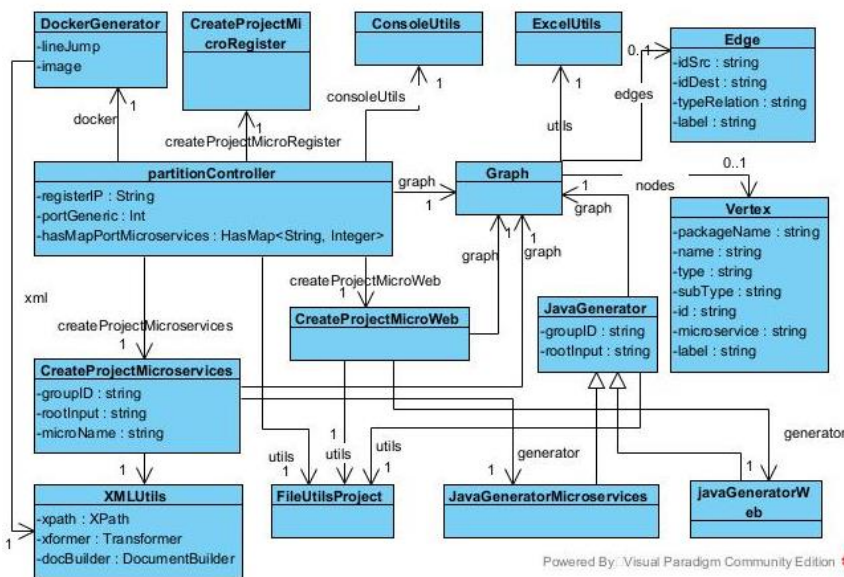


Figura 10 Diagrama de clases de la herramienta de particionamiento

8.2 Comportamiento del sistema

En esta sección se encuentran los diagramas de secuencia más complejos e importantes para el sistema.

Los procesos importantes:

- Crear un microservicio
- Crear el microservice web
- Crear el microservice Register

A continuación, se muestran los diagramas de secuencia de las funcionalidades mencionadas:

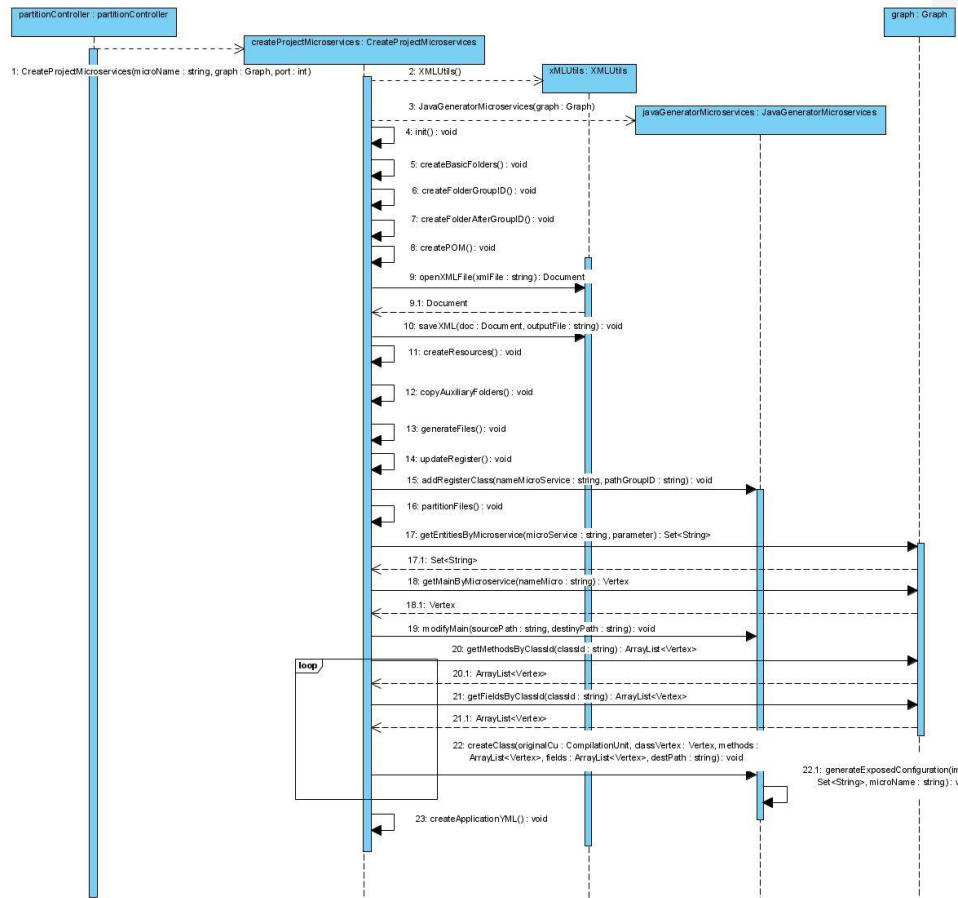


Figura 11 Diagrama de secuencia creación microservicios

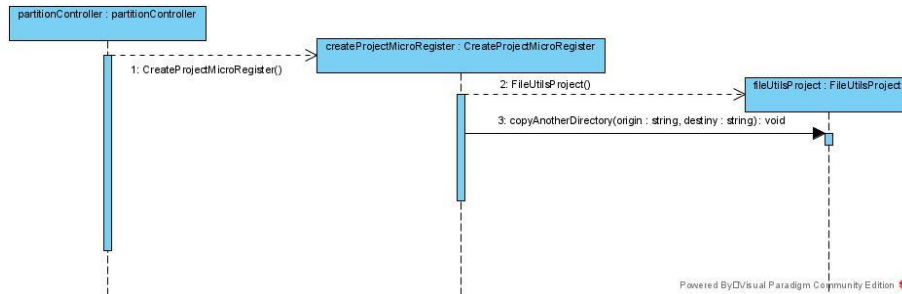


Figura 12 Diagrama de secuencia crear el microservicio register

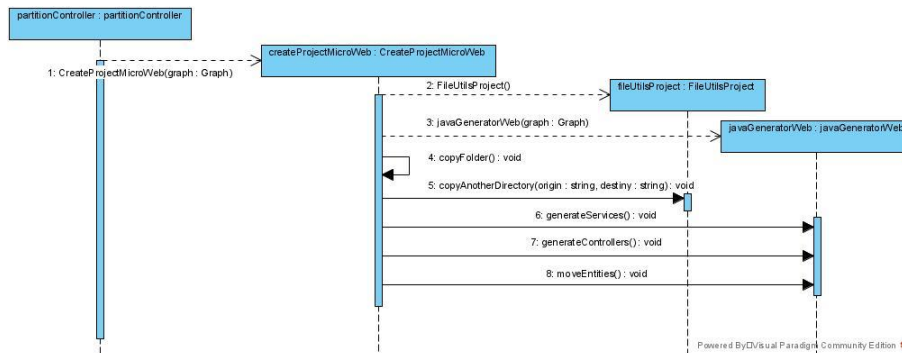


Figura 13 Diagrama de secuencia crear microservicio web

8.3 Persistencia

La herramienta cuenta con varias unidades de persistencia donde se almacena la información necesaria para realizar el particionamiento, las salidas de dicho proceso y las plantillas necesarias para generar los microservicios.

En las entradas se encuentran tanto el grafo, con las especificaciones para la partición, los archivos fuente del monolito y las plantillas, que contiene una estructura genérica para crear los microservicios. En las salidas se encuentran algunos los archivos fuente de los microservicios generados.

Las entradas se ven de la siguiente forma:

- Graph.xls

Este grafo es un Excel compuesto por dos hojas, la primera de los nodos del grafo y la segunda sus aristas.

Nodos		
Nombre del atributo	Descripción	Ejemplo:
ID	Identificador del nodo	micro.org.springframework.samples.model.BaseEntity

Comentado [JM2]: Indicar que las tablas son hojas dentro del excel

Comentado [SDNV3R2]: Se aclaro que representa cada hoja del archivo del grafo.

Package	Nombre del paquete al que pertenece la clase o método a la que hace referencia el nodo	org.springframework.samples.m ode
Names	Nombre de la clase o método	BaseEntity
Label	Nombre para poner impresiones en el código	label
Type	Nombre para diferenciar a las clases y métodos	Class
SubType	Nombre para diferenciar a los tipos de clases, ya sea entity, repository o controller	entity
Microservice	Nombre del microservicio al que pertenece el nodo	micro

Edges		
Nombre del atributo	Descripción	Ejemplo:
Src	ID del nodo rigen de la arista	pet1.org.springframework.samples.petclinic.owner.PetController.update(int, int, Pet)
Dst	ID del nodo destino de la arista	owner.org.springframework.samples.petclinic.owner.OwnerRepository.findById(Integer)
Type	Tipo de relación de la arista, ya sea calls, uses class, has method, has field y has annotation	Calls
Label	Nombre para poner impresiones en el código	Label

- Archivo-fuente-monolito
- templates

Las salidas que se generan son:

- Microservice-register
- Microservice-web
- Microservice-1
- Microservice-2
- Microservice-N

Siendo n el número de microservicios a generar.

Tanto como las entradas como salidas tienen más componentes para más especificación revisar la documentación.

9 Anexos

- Anexo 1: Diagrama de clases. *Ubicado en la carpeta “anexos/Estructura del sistema”*