



**Informe técnico**

**ID: ITIS-2024-CAOSD001**

# Análisis de variabilidad y generación de ficheros de configuración sobre el modelo de características de Kubernetes

Junio 2024

Informe Técnico generado como parte del Proyecto **IRIS: Multi-stage**

**configuration of virtualised services for sustainable adaptation of mobile networks (PID2021-122812OB-I00).** Proyecto de Generación del Conocimiento financiado por Agencia Estatal de Investigación. Ministerio de Ciencia, Innovación y Universidades

Investigadoras principales:

Lidia Fuentes Fernández y Mercedes Amor Pinilla

Realizado por Enrique López Encinas (personal contratado)

## Resumen

Este informe técnico presenta un análisis de la variabilidad de los modelos de características en Kubernetes. Aborda la creación de un modelo de características para Kubernetes, que captura los diferentes componentes y configuraciones posibles dentro de un clúster de Kubernetes, permitiendo la exploración y gestión de diversas combinaciones de características. Este proceso incluye la identificación de características clave, la organización de estas en una estructura jerárquica y la definición de reglas de variabilidad que determinan cómo se pueden combinar las características. Posteriormente, se analiza la variabilidad y las características de Kubernetes, evaluando las configuraciones posibles. Este análisis incluye el cálculo del número total de configuraciones posibles, la distribución de probabilidad de que se incluya, etc., considerando las combinaciones válidas de características. El análisis de la variabilidad en Kubernetes permite a los administradores de sistemas y desarrolladores tomar decisiones informadas sobre la configuración de sus clústeres, optimizando el uso de recursos y mejorando la eficiencia operativa, permitiendo de esta forma que las configuraciones de Kubernetes puedan adaptarse a una amplia gama de necesidades y entornos, desde implementaciones pequeñas hasta grandes infraestructuras distribuidas. Con este análisis, se facilita la toma de decisiones estratégicas en la configuración y gestión de clústeres Kubernetes, maximizando su rendimiento y adaptabilidad. Finalmente, se incluyen los comandos necesarios para ejecutar los scripts desarrollados en el proyecto. Estos comandos permiten reproducir los experimentos y análisis descritos en el informe, facilitando la verificación y aplicación de los resultados obtenidos y se discuten las implicaciones prácticas de los resultados obtenidos y se sugieren posibles direcciones para futuros trabajos en este campo.

## Índice

Resumen.....	3
1. Introducción .....	5
1.1 Kubernetes.....	5
1.2 Líneas de productos software.....	7
1.3 Entorno de trabajo y herramientas usadas .....	8
2. Metodología.....	9
2.1 Generación de los ficheros de configuración.....	9
2.2 Resultados del análisis de variabilidad del FM de Kubernetes .....	10
3. Conclusiones.....	11
Referencias .....	12
Modelo de características de Kubernetes.....	13
Comandos para ejecutar los scripts .....	28



- **Clúster:** Conjunto de nodos (máquinas físicas o virtuales) que funcionan en conjunto para ejecutar aplicaciones en contenedores.
- **Nodo:** Máquina física o virtual que forma parte de un clúster de Kubernetes y es responsable de ejecutar las aplicaciones en contenedores. Como mínimo, el clúster debe tener un nodo máster y dos nodos *worker*.
- **Nodo máster:** Responsable de gestionar el estado del clúster, orquestar la carga de trabajo y realizar tareas administrativas
- **Nodo *worker*:** Ejecutan las aplicaciones en contenedores.
- **Scheduler:** Asigna los pods a los nodos en el clúster.
- **API server:** Proporciona la interfaz RESTful para interactuar con el clúster.
- **Pod:** Es una abstracción que representa una o más instancias de un contenedor (o, en algunos casos, múltiples contenedores que comparten el mismo espacio de red y almacenamiento).
- **Contenedor:** Unidad de software ligera y portátil que encapsula una aplicación y todas sus dependencias (bibliotecas, binarios, archivos de configuración, etc.) en un entorno aislado.
- **Objeto:** Artefactos que se pueden definir en un despliegue de Kubernetes dentro de un clúster.
- **Service:** Permiten la comunicación entre los diferentes componentes de una aplicación, tanto dentro como fuera del clúster de Kubernetes.
- **Deployment:** Abstracción que define un conjunto de pods que realizan la misma función y proporciona una forma de acceder a ellos de manera uniforme a través de una dirección IP y un nombre de servicio.
- **Grafana:** Un panel configurable para visualizar los datos obtenidos por Prometheus.
- **Prometheus:** Herramienta que recoge datos del clúster y los sirve en una API.

## 1.2 Líneas de productos software.

Una línea de productos software (SPL del inglés *Software Product Line*) es una familia de productos software relacionados entre sí que comparten ciertas características comunes (similitudes) pero que también tienen características variables. En un SPL, los sistemas se descomponen en características (*features*) que representan funcionalidades o comportamientos del sistema.

El principal artefacto para modelar la variabilidad en SPL son los modelos de variabilidad. Existen muchos modelos de variabilidad, pero los más extendidos y usados en la práctica son los modelos de características (*feature models*). Un *feature model* (FM) es un modelo para representar la variabilidad de una SPL en base a características comunes y variables (véase Figura 2), especificando qué características se pueden seleccionar en una configuración para generar un producto válido de la SPL.

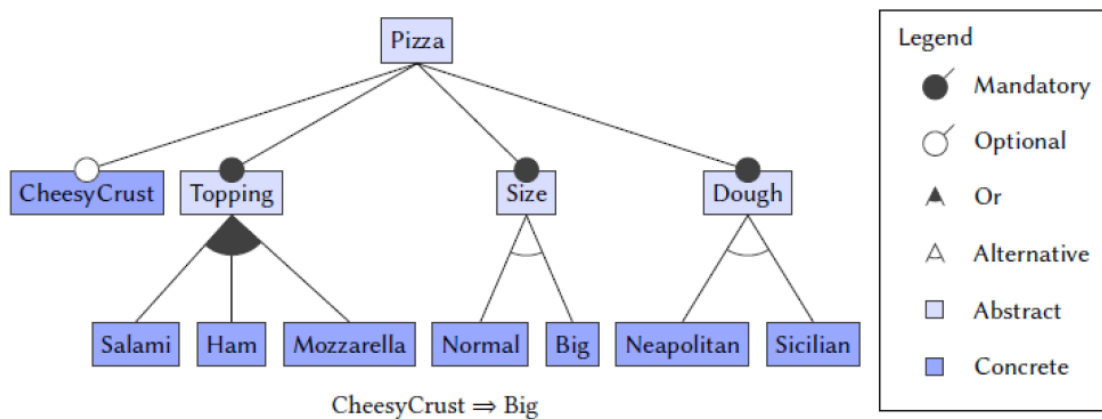


Figura 2. Feature model especificado con la herramienta FeatureIDE (Knüppel et al., 2017)

Un FM contiene dos partes bien diferenciadas: el árbol de características y las restricciones textuales.

1. El árbol de características (feature tree) que descompone las características en una estructura jerárquica en forma de árbol. En el árbol, existen dos tipos de características: características concretas (concrete) y características abstractas (abstract).
2. Las restricciones textuales (cross-tree constraints) son relaciones más complejas entre las características que no pueden ser modeladas en la jerarquía del árbol. Las restricciones textuales suelen especificarse con fórmulas lógicas de diferente complejidad (e.g., fórmulas proposicionales, lógica de primer orden, etc.); pero los tipos de restricciones más simples incluyen restricciones del tipo “*A requires B*” (también escrita como  $A \Rightarrow B$ ).

Los FMs se pueden formalizar a lógica proposicional para analizar y razonar automáticamente sobre sus propiedades. Por ejemplo, para conocer si el modelo es válido, esto es, si representa al menos un producto; para generar todas configuraciones y productos válidos; para extraer aquellas características que están presentes en todas las configuraciones (conocidas como *core features*), o aquellas que no están presentes en ningún producto (conocidas como *dead features*), etc.

Una configuración válida de un modelo es un conjunto de características que cumple con todas las relaciones del árbol y restricciones textuales del modelo. Por ejemplo, una de las configuraciones del FM de ejemplo sería:

[Pizza, Topping, Mozzarella, Ham, Size, Normal, Dough, Sicilian]

La diferencia entre una configuración y un producto viene dada por la diferencia entre las características abstractas y concretas. Un producto es una configuración válida del modelo donde las características abstractas han sido eliminadas. Así, el producto resultante para la configuración del ejemplo anterior sería:

[Mozzarella, Ham, Normal, Sicilian]

Este informe se centra concretamente en los modelos de características, usando un FM de Kubernetes (anexo A) para obtener el número de configuración posibles de un fichero de configuración y para ayudar con la creación de un objeto de Kubernetes.

### 1.3 Entorno de trabajo y herramientas usadas

Todo el trabajo realizado que da lugar a este informe se ha hecho en el entorno de programación Visual Studio Code (a partir de ahora, VS Code), usando el lenguaje Python en gran parte, aunque usando también otros como UVL (*Universal Variability Language*) (Sundermann et al., 2021) para definir los modelos de características (*Feature Models*) o Jinja2 para la creación de las plantillas usadas en la generación de ficheros de configuración de Kubernetes.

Algunos plugin de VS Code han resultado ser muy útiles en el desarrollo. A continuación, se procede a describirlos:

- **UVLS – Universal Variability Language Server** (Loth et al., 2023): Es una herramienta que ayuda durante la creación de los FM detectando errores en su definición mientras se desarrolla. Necesita la instalación de la librería llamada z3 para tener todas sus funcionalidades activas, que son la creación de una configuración del FM respetando las restricciones dadas y generar un archivo usado para visualizar el FM.
- **Graphviz interactive Preview**: Este plugin hace uso del archivo generado por el anterior plugin para mostrar el FM en una nueva ventana del VS Code. Deja de ser tan útil cuando se manejan FM con gran cantidad de características, pues visualizarlos se hace complicado debido a su tamaño.

Otra herramienta usada para el análisis de variabilidad del FM de Kubernetes es Flamapy<sup>1</sup>(A. Galindo et al., 2023). Con ella, se han obtenido datos del modelo tales como el número de características, numero de restricciones, la media de características que se escogen en la configuración del FM, etc.

---

<sup>1</sup> Página oficial de Flamapy: <https://flamapy.github.io/>



## 2. Metodología

### 2.1 Generación de los ficheros de configuración

Uno de los principales objetivos que se busca conseguir es poder generar automáticamente todos los ficheros necesarios para el despliegue de una aplicación en Kubernetes de una forma intuitiva y sin necesidad de conocer todas las dependencias que existen en un proyecto.

Para ello se va a usar un FM que contienen las posibles configuraciones, de forma que el usuario solo tenga que ir especificando aquellas que necesite.

Hasta el momento de escribir este artículo, se ha desarrollado una primera versión del FM de Kubernetes, que permite crear manifiestos YAML para creación de objetos del tipo *Deployment* (que incluyen pods y contenedores) y *Service*, Aunque de forma limitada, pues no incluye aun todas las características que se pueden definir en un manifiesto.

Para poder generar el fichero de configuración, se requieren de 4 elementos:

el *feature model* del que se consigue una configuración válida en formato JSON, una plantilla definida con Jinja2 y un archivo csv que contiene una tabla que relaciona los elementos de la plantilla con los valores de la configuración.

1. **Feature Model:** Aquellas características que se quieran reflejar en la plantilla se indican en la primera columna de la tabla de mapeo, en la columna de *features*.
2. **Configuración válida:** A partir del FM se consigue un archivo JSON que contiene los valores de las características seleccionados para esta configuración.
3. **Plantilla:** Una plantilla hecha con Jinja2 que contiene referencias a los valores almacenados en el JSON de la configuración válida, y que define la estructura final del archivo de configuración.
4. **Archivo de mapeo:** Un archivo csv que contiene una tabla con 3 columnas; *Feature*, *Handler* y *Value*. La primera es el nombre de la *característica* en el archivo de configuración, de la que se consulta su valor. La segunda es el nombre de la variable en la plantilla donde se pondrá el valor referenciado en la primera columna.

Una vez se disponen de estos ficheros, con ayuda de un script en Python basado en el código del GitHub *uvengine* (Jmhorcas. (s. f.)) y cuyo código, con ligeras modificaciones, se puede encontrar en GitHub *fms\_dataset* (CAOSD-group. (s. f.)) se obtiene el fichero de configuración listo para desplegarlo en Kubernetes.

Como paso adicional, se pueden usar herramientas de validación como *Kubeconform* (Yannh. (s. f.)) o *Kube-score*<sup>2</sup>(Zegl. (s. f.)) en la ejecución de este script para comprobar que el fichero resultante está bien definido, eliminando la necesidad de tener que desplegarlo en un entorno real para descubrir errores. En el anexo B se da más información acerca de la ejecución del script añadiendo la validación con estas herramientas.

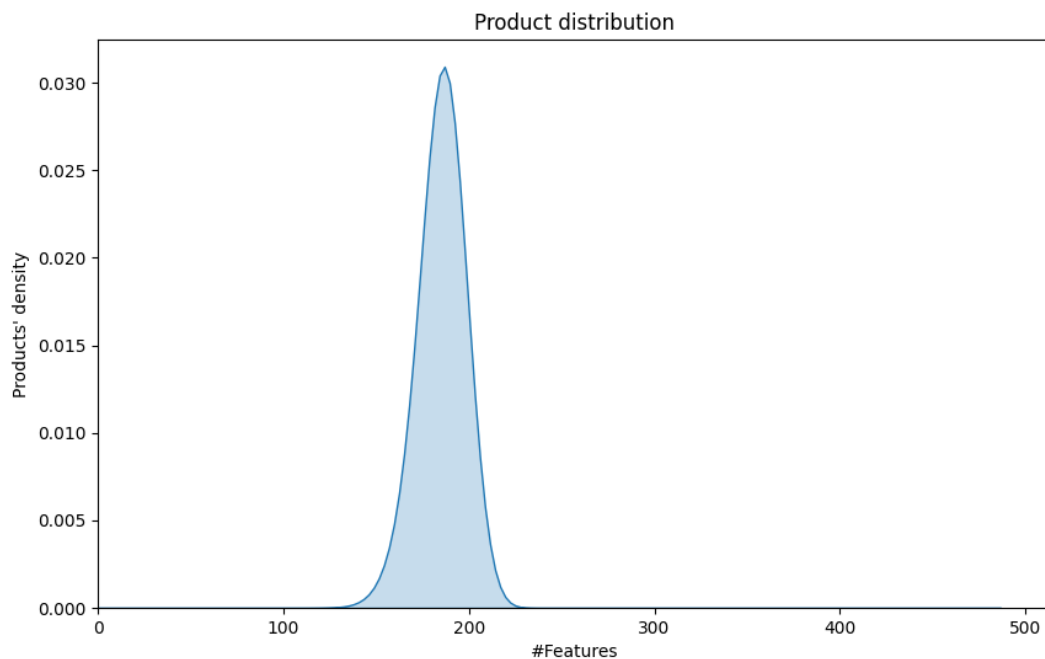
---

<sup>2</sup> Herramienta online de Kube-score: <https://kube-score.com/>

## 2.2 Resultados del análisis de variabilidad del FM de Kubernetes

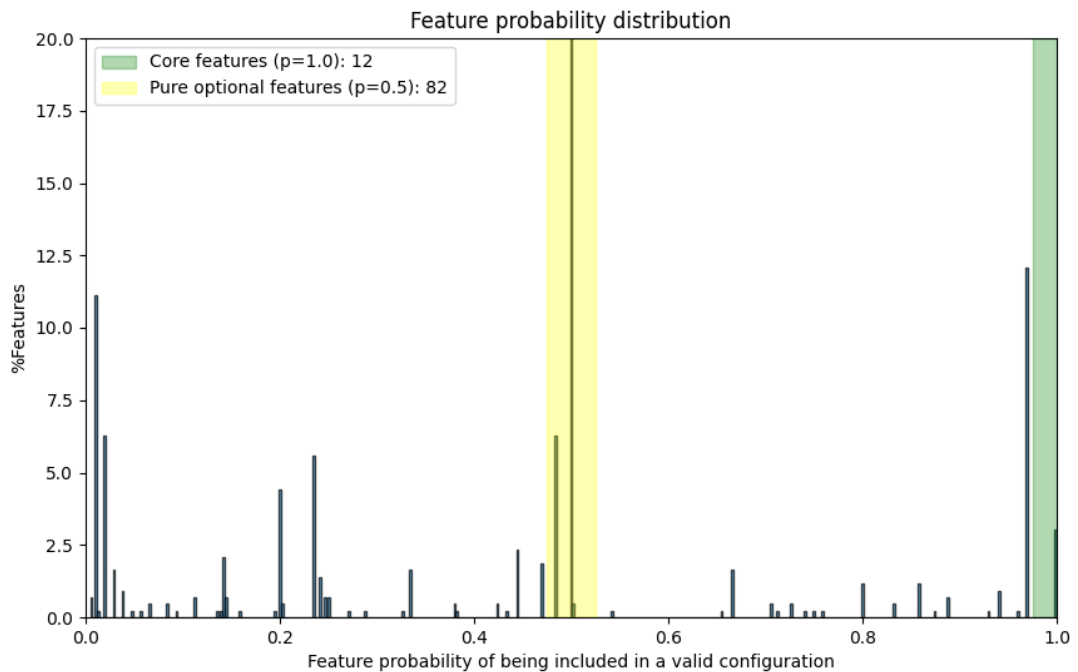
Usando la herramienta Flamapy ya mencionada anteriormente, se ha conseguido analizar la variabilidad del FM de Kubernetes, obteniendo los siguientes datos en esta primera versión:

Con el modelo actual, los datos obtenidos nos dicen que se suelen tener 186 atributos de media en una configuración, siendo la configuración mínima de tan solo 9 características, y la máxima de 282. En la figura 3 se observa mejor la distribución, agrupándose la mayoría de las configuraciones entre 150 y 220 características aproximadamente.



**Figura 3:** Gráfico de distribución de las características del FM de Kubernetes.

La figura 4 proporciona algo más de información sobre el uso de las características, entendiendo que a más a la derecha se encuentre, más probable es que esa característica se incluya en una configuración válida. Por ejemplo, vemos a la izquierda un número importante de ellas que en muy pocas ocasiones se incluyen en una configuración válida, aunque no llegan a ser características a las que nunca se acceden (*dead features*).



**Figura 4:** Distribución de probabilidad de una característica de ser incluida en una configuración válida del FM de Kubernetes.

En definitiva, este FM de Kubernetes tiene 431 características y 43 restricciones, elevando el número de configuraciones posibles a  $6.78e42$  configuraciones válidas aproximadamente.

### 3. Conclusiones

Del desarrollo de este proyecto se concluye que es posible afrontar el problema de la configuración de un despliegue de Kubernetes con un enfoque de líneas de productos software y más concretamente de modelos de características. Se ha conseguido desarrollar una primera versión del FM de Kubernetes para su análisis de variabilidad, dando como resultado un modelo de un tamaño muy elevado y con un gran número de características usadas frecuentemente. Estos resultados podrían aplicarse a otras tecnologías (como Docker) que requieran de grandes configuraciones que no son triviales para el usuario, aportando la idea de una nueva herramienta que agiliza el trabajo de desarrolladores e investigadores.

## Referencias

1. Alexander Knüppel, Thomas Thüm, Stephan Mennicke, Jens Meinicke, and Ina Schaefer. 2017. Is there a mismatch between real-world feature models and product-line research? In Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering (ESEC/FSE 2017). Association for Computing Machinery, New York, NY, USA, 291–302. <https://doi.org/10.1145/3106237.3106252>.
2. Chico Sundermann, Kevin Feichtinger, Dominik Engelhardt, Rick Rabiser, and Thomas Thüm. 2021. Yet another textual variability language? a community effort towards a unified language. In Proceedings of the 25th ACM International Systems and Software Product Line Conference - Volume A (SPLC '21). Association for Computing Machinery, New York, NY, USA, 136–147. <https://doi.org/10.1145/3461001.3471145>.
3. Jacob Loth, Chico Sundermann, Tobias Schroll, Thilo Brugger, Felix Rieg, and Thomas Thüm. 2023. UVLS: A Language Server Protocol For UVL. In Proceedings of the 27th ACM International Systems and Software Product Line Conference - Volume B (SPLC '23), Vol. B. Association for Computing Machinery, New York, NY, USA, 43–46. <https://doi.org/10.1145/3579028.3609014>.
4. José A. Galindo, Jose-Miguel Horcas, Alexander Felferning, David Fernandez-Amoros, and David Benavides. 2023. FLAMA: A collaborative effort to build a new framework for the automated analysis of feature models. In Proceedings of the 27th ACM International Systems and Software Product Line Conference - Volume B (SPLC '23), Vol. B. Association for Computing Machinery, New York, NY, USA, 16–19. <https://doi.org/10.1145/3579028.3609008>.
5. Jmhorcas. (s. f.). GitHub - jmhorcas/uvengine: SPL implementation of the variability using templates. GitHub. <https://github.com/jmhorcas/uvengine>.
6. CAOSD-group. (s. f.). GitHub - CAOSD-group/fms\_dataset: Dataset of feature models of the CAOSD research group. GitHub. [https://github.com/CAOSD-group/fms\\_dataset](https://github.com/CAOSD-group/fms_dataset).
7. Yannh. (s. f.). GitHub - yannh/kubeconform: A FAST Kubernetes manifests validator, with support for Custom Resources! GitHub. <https://github.com/yannh/kubeconform>.
8. Zegl. (s. f.). GitHub - zegl/kube-score: Kubernetes object analysis with recommendations for improved reliability and security. kube-score actively prevents downtime and bugs in your Kubernetes YAML and Charts. Static code analysis for Kubernetes. GitHub. <https://github.com/zegl/kube-score>

# Anexo A

## Modelo de características de Kubernetes

features

Kubernetes\_manifest

mandatory

Boolean ApiVersion

mandatory

Boolean Group {abstract}

alternative

Boolean GROUP\_admission\_registration

Boolean GROUP\_apiextensions

Boolean GROUP\_apiregistration

Boolean GROUP\_apps

Boolean GROUP\_authentication

Boolean GROUP\_authorization

Boolean GROUP\_autoscaling

Boolean GROUP\_batch

Boolean GROUP\_certificates

Boolean GROUP\_coordination

Boolean GROUP\_core

Boolean GROUP\_discovery

Boolean GROUP\_events

Boolean GROUP\_flowcontrol\_apiserver

Boolean GROUP\_internal\_apiserver

Boolean GROUP\_networking

Boolean GROUP\_node

Boolean GROUP\_policy

Boolean GROUP\_rbac\_authorization

Boolean GROUP\_resource

Boolean GROUP\_scheduling

Boolean GROUP\_storage

Boolean Version {abstract}

alternative

Boolean VERSION\_v1

Boolean VERSION\_v1beta1

Boolean VERSION\_v1beta3

Boolean VERSION\_v1alpha1

Boolean VERSION\_v1alpha2

Boolean VERSION\_v2

Boolean Kind

alternative

Boolean Workloads\_APIs {abstract}

alternative

Boolean KIND\_Container

Boolean KIND\_CronJob

Boolean KIND\_DaemonSet

Boolean KIND\_Deployment

Boolean KIND\_Job

Boolean KIND\_Pod

Boolean KIND\_ReplicaSet

Boolean KIND\_ReplicationController

Boolean KIND\_StatefullSet

Boolean Service\_APIs {abstract}

alternative

Boolean KIND\_Endpoints

Boolean KIND\_EndpointSlice

Boolean KIND\_Ingress

Boolean KIND\_IngressClass

Boolean KIND\_Service

Boolean Config\_and\_storage\_APIs {abstract}

alternative

Boolean KIND\_ConfigMap

Boolean KIND\_CSIDriver

Boolean KIND\_CSINode

Boolean KIND\_Secret

Boolean KIND\_PersistentVolumeClaim

Boolean KIND\_StorageClass

Boolean KIND\_CSISStorageCapacity

Boolean KIND\_Volume

Boolean KIND\_VolumeAttachment

Boolean KIND\_VolumeAttributesClass

Boolean Metadata\_APIs {abstract}

alternative

Boolean KIND\_ClusterTrustBundle

Boolean KIND\_ControllerRevision

Boolean KIND\_CustomerResourceDefinition

Boolean KIND\_Event

Boolean KIND\_LimitRange

Boolean KIND\_HorizontalPodAutoscaler

Boolean KIND\_MutatingWebhookConfiguration

Boolean KIND\_ValidatingWebhookConfiguration

```

Boolean KIND_PodSchedulingContext
Boolean KIND_PodTemplate
Boolean KIND_PodDisruptionBudget
Boolean KIND_PriorityClass
Boolean KIND_ResourceClaim
Boolean KIND_ResourceClaimParameters {added_version 1.30}
Boolean KIND_ResourceSlice {added_version 1.30}
Boolean KIND_ResourceClaimTemplate
Boolean KIND_ResourceClass
Boolean KIND_ValidatingAdmissionPolicy
Boolean KIND_ValidatingAdmissionPolicyBinding
Boolean Cluster_APIs {abstract}
  alternative
    Boolean KIND_APIService
    Boolean KIND_Binding
    Boolean KIND_CertificateSigningRequest
    Boolean KIND_ClusterRole
    Boolean KIND_ClusterRoleBinding
    Boolean KIND_ComponentStatus
    Boolean KIND_FlowSchema
    Boolean KIND_IPAddress
    Boolean KIND_Lease
    Boolean KIND_LocalSubjectAccessReview
    Boolean KIND_Namespace
    Boolean KIND_Node
    Boolean KIND_PersistentVolume
    Boolean KIND_PriorityLevelConfiguration
    Boolean KIND_ResourceQuota
    Boolean KIND_Role
    Boolean KIND_RoleBinding
    Boolean KIND_RuntimeClass
    Boolean KIND_SelfSubjectAccessReview
    Boolean KIND_SelfSubjectReview
    Boolean KIND_SelfSubjectRulesReview
    Boolean KIND_ServiceAccount
    Boolean KIND_ServiceCIDR
    Boolean KIND_StorageVersion
    Boolean KIND_SubjectAccessReview
    Boolean KIND_TokenRequest
    Boolean KIND_TokenReview
    Boolean KIND_NetworkPolicy

```

optional

Boolean Metadata

mandatory

String METADATA\_name

optional

String METADATA\_namespace

Boolean METADATA\_labels cardinality [0..\*]

mandatory

String LABELS\_key

String LABELS\_value

Boolean METADATA\_annotations cardinality [0..\*]

mandatory

String ANNOTATIONS\_key

String ANNOTATIONS\_value

Boolean Spec

optional

Boolean DeploymentSpec {abstract}

optional

Integer DEPLOYMENTSPEC\_minReadySeconds

Boolean DEPLOYMENTSPEC\_paused

Integer DEPLOYMENTSPEC\_progressDeadlineSeconds

Integer DEPLOYMENTSPEC\_Replicas

Integer DEPLOYMENTSPEC\_revisionHistoryLimit

Boolean DEPLOYMENTSPEC\_LabelSelector

or

Boolean LABELSELECTOR\_matchLabels cardinality [0..\*]

mandatory

String MATCHLABELS\_key

String MATCHLABELS\_value

Boolean LABELSELECTOR\_matchExpressions cardinality [0..\*]

mandatory

String MATCHEXPRESSIONS\_key

Boolean MATCHEXPRESSIONS\_values cardinality [0..\*]

mandatory

String MATCHEXPRESSIONS\_value

Boolean MATCHEXPRESSIONS\_operator

alternative

Boolean OPERATOR\_in

Boolean OPERATOR\_notIn

Boolean OPERATOR\_exist

Boolean OPERATOR\_doesNotExist

Boolean OPERATOR\_gt

Boolean OPERATOR\_lt



```

Boolean DEPLOYMENTSPEC_strategy
  mandatory
    Boolean STRATEGY_type
      or
        Boolean Recreate
        Boolean RollingUpdate
          mandatory
            Integer maxUnavailable
          optional
            Integer maxSurge
Boolean DEPLOYMENTSPEC_template
  mandatory
    Boolean DEPLOYMENTSPEC_TEMPLATE_Metadata
      optional
        String TEMPLATE_METADATA_name
        String TEMPLATE_METADATA_namespace
        Boolean TEMPLATE_METADATA_labels cardinality [0..*]
          mandatory
            String TEMPLATE_METADATA_LABELS_key
            String TEMPLATE_METADATA_LABELS_value
        Boolean TEMPLATE_METADATA_annotations cardinality [0..*]
          mandatory
            String TEMPLATE_ANNOTATIONS_key
            String TEMPLATE_ANNOTATIONS_value
        Boolean DEPLOYMENTSPEC_TEMPLATE_spec
Boolean PodSpec {abstract}
  optional
    Integer PODSPEC_activeDeadlineSeconds
    Boolean PODSPEC_affinity
      optional
        Boolean AFFINITY_nodeAffinity
        Boolean AFFINITY_podAffinity
        Boolean AFFINITY_podAntiAffinity
    Boolean PODSPEC_automountServiceAccountToken
    Boolean PODSPEC_containers cardinality [0..*]
      optional
        Boolean CONTAINER_args cardinality [0..*]
          mandatory
            String CONTAINER_ARGS_value
        Boolean CONTAINER_command cardinality [0..*]
          mandatory
            String CONTAINER_COMMAND_value

```

```

Boolean CONTAINER_env cardinality [0..*]
  mandatory
    String CONTAINER_ENV_name
    String CONTAINER_ENV_value
    Boolean CONTAINER_ENV_valueFrom
  alternative
    Boolean configMapKeyRef
      mandatory
        String CONFIGMAPKEYREF_key
        String CONFIGMAPKEYREF_name
        Boolean CONFIGMAPKEYREF_optional
    Boolean fieldRef
      mandatory
        String FIELDREF_apiversion
        String FIELDREF_fieldPath
    Boolean resourceFieldRef
      mandatory
        String RESOURCFIELDREF_containerName
        String RESOURCFIELDREF_divisor
        String RESOURCFIELDREF_resource
    Boolean secretKeyRef
      mandatory
        String SECRETKEYREF_key
        String SECRETKEYREF_name
        Boolean SECRETKEYREF_optional
Boolean CONTAINER_envFrom
  mandatory
    Boolean ENVFROM_configMapRef
      mandatory
        String CONFIGMAPREF_name
        Boolean CONFIGMAPREF_optional
    String ENVFROM_prefix
    Boolean ENVFROM_secretRef
      mandatory
        String SECRETREF_name
        Boolean SECRETREF_optional
String CONTAINER_image
Boolean CONTAINER_imagePullPolicy
  alternative
    Boolean IMAGEPULLPOLICY_always
    Boolean IMAGEPULLPOLICY_never
    Boolean IMAGEPULLPOLICY_ifNotPresent

```

```

Boolean CONTAINER_lifecycle
  optional
    Boolean LIFECYCLE_postStart
      alternative
        Boolean POSTSTART_exec
          mandatory
            String POSTSTART_EXEC_command
        Boolean POSTSTART_httpGet
          mandatory
            String POSTSTART_HTTPGET_host
            Boolean POSTSTART_HTTPGET_httpHeaders
              mandatory
                String POSTSTART_HTTPHEADERS_name
                String POSTSTART_HTTPHEADERS_value
            String POSTSTART_HTTPGET_path
            String POSTSTART_HTTPGET_port
            String POSTSTART_HTTPGET_scheme
        Boolean POSTSTART_sleep
          mandatory
            Integer POSTSTART_SLEEP_seconds
    Boolean LIFECYCLE_preStop
      alternative
        Boolean PRESTOP_exec
          mandatory
            String PRESTOP_EXEC_command
        Boolean PRESTOP_httpGet
          mandatory
            String PRESTOP_HTTPGET_host
            Boolean PRESTOP_HTTPGET_httpHeaders
              mandatory
                String PRESTOP_HTTPHEADERS_name
                String PRESTOP_HTTPHEADERS_value
            String PRESTOP_HTTPGET_path
            String PRESTOP_HTTPGET_port
            String PRESTOP_HTTPGET_scheme
        Boolean PRESTOP_sleep
          mandatory
            Integer PRESTOP_SLEEP_seconds
    Boolean CONTAINER_livenessProbe
      mandatory
        Boolean LIVENESSPROBE_exec
          mandatory

```

```

        String LIVENESSPROBE_EXEC_command
Integer LIVENESSPROBE_failureThreshold
Boolean LIVENESSPROBE_grpc
    mandatory
        Integer LIVENESSPROBE_GRPC_port
        String LIVENESSPROBE_GRPC_service
Boolean LIVENESSPROBE_httpGet
    optional
        String LIVENESSPROBE_HTTPGET_host
        Boolean LIVENESSPROBE_HTTPGET_httpHeaders
            mandatory
                String LIVENESSPROBE_HTTPHEADERS_name
                String LIVENESSPROBE_HTTPHEADERS_value
        String LIVENESSPROBE_HTTPGET_path
        Integer LIVENESSPROBE_HTTPGET_port
        String LIVENESSPROBE_HTTPGET_scheme
Integer LIVENESSPROBE_initialDelaySeconds
Integer LIVENESSPROBE_periodSeconds
Integer LIVENESSPROBE_successThreshold
Boolean LIVENESSPROBE_tcpSocket
    mandatory
        Boolean LIVENESSPROBE_TCPSOCKET_host
        Boolean LIVENESSPROBE_TCPSOCKET_port
Integer LIVENESSPROBE_terminationGracePeriodSeconds
Integer LIVENESSPROBE_timeoutSeconds
String CONTAINER_name
Boolean CONTAINER_ports cardinality [0..*]
    mandatory
        Integer CONTAINER_PORTS_containerPort
    optional
        String CONTAINER_PORTS_hostIP
        Integer CONTAINER_PORTS_hostPort
        String CONTAINER_PORTS_name
        Boolean CONTAINER_PORTS_protocol
            alternative
                Boolean PROTOCOL_UDP
                Boolean PROTOCOL_TCP
                Boolean PROTOCOL_SCTP
Boolean CONTAINER_readinessProbe
    mandatory
        Boolean READINESSPROBE_exec
            mandatory

```

```

        String READINESSPROBE_EXEC_command
Integer READINESSPROBE_failureThreshold
Boolean READINESSPROBE_grpc
    mandatory
        Integer READINESSPROBE_GRPC_port
        String READINESSPROBE_GRPC_service
Boolean READINESSPROBE_httpGet
    optional
        String READINESSPROBE_HTTPGET_host
        Boolean READINESSPROBE_HTTPGET_httpHeaders
            mandatory
                String READINESSPROBE_HTTPHEADERS_name
                String READINESSPROBE_HTTPHEADERS_value
        String READINESSPROBE_HTTPGET_path
        Integer READINESSPROBE_HTTPGET_port
        String READINESSPROBE_HTTPGET_scheme
Integer READINESSPROBE_initialDelaySeconds
Integer READINESSPROBE_periodSeconds
Integer READINESSPROBE_successThreshold
Boolean READINESSPROBE_tcpSocket
    mandatory
        Boolean READINESSPROBE_TCPSOCKET_host
        Boolean READINESSPROBE_TCPSOCKET_port
Integer READINESSPROBE_terminationGracePeriodSeconds
Integer READINESSPROBE_timeoutSeconds
Boolean CONTAINER_resizePolicy cardinality [0..*]
    mandatory
        Boolean RESIZEPOLICY_resourceName
            alternative
                Boolean CONTAINER_RESIZEPOLICY_RESOURCENAME_cpu
                Boolean CONTAINER_RESIZEPOLICY_RESOURCENAME_memory
        String RESIZEPOLICY_restartPolicy
Boolean CONTAINER_resources
    or
        Boolean RESOURCES_request
            mandatory
                String REQUEST_cpu
                String REQUEST_memory
        Boolean RESOURCES_limits
            mandatory
                String LIMITS_cpu
                String LIMITS_memory

```

```

        Boolean RESOURCES_claims
            mandatory
                String RESOURCES_CLAIMS_name
Boolean CONTAINER_securityContext
Boolean CONTAINER_startupProbe
    mandatory
        Boolean STARTUPPROBE_exec
            mandatory
                String STARTUPPROBE_EXEC_command
        Integer STARTUPPROBE_failureThreshold
        Boolean STARTUPPROBE_grpc
            mandatory
                Integer STARTUPPROBE_GRPC_port
                String STARTUPPROBE_GRPC_service
        Boolean STARTUPPROBE_httpGet
            optional
                String STARTUPPROBE_HTTPGET_host
                Boolean STARTUPPROBE_HTTPGET_httpHeaders
                    mandatory
                        String STARTUPPROBE_HTTPHEADERS_name
                        String STARTUPPROBE_HTTPHEADERS_value
                String STARTUPPROBE_HTTPGET_path
                Integer STARTUPPROBE_HTTPGET_port
                String STARTUPPROBE_HTTPGET_scheme
        Integer STARTUPPROBE_initialDelaySeconds
        Integer STARTUPPROBE_periodSeconds
        Integer STARTUPPROBE_successThreshold
        Boolean STARTUPPROBE_tcpSocket
            mandatory
                Boolean STARTUPPROBE_TCPSOCKET_host
                Boolean RSTARTUPPROBE_TCPSOCKET_port
        Integer STARTUPPROBE_terminationGracePeriodSeconds
        Integer STARTUPPROBE_timeoutSeconds
Boolean CONTAINER_stdin
Boolean CONTAINER_stdinOnce
String CONTAINER_terminationMessagePath
String CONTAINER_terminationMessagePolicy
Boolean CONTAINER_tty
Boolean CONTAINER_volumeDevices cardinality [0..*]
    mandatory
        String VOLUMEDEVICES_devicePath
        String VOLUMEDEVICES_name

```

```

Boolean CONTAINER_volumeMounts cardinality [0..*]
    mandatory
        String VOLUMEMOUNTS_mountPath
        String VOLUMEMOUNTS_name
    Boolean CONTAINER_workingDir
Boolean PODSPEC_dnsPolicy
    alternative
        Boolean DNSPOLICY_ClusterFirstWithHostNet
        Boolean DNSPOLICY_ClusterFirst
        Boolean DNSPOLICY_Default
        Boolean DNSPOLICY_None
Boolean PODSPEC_enableServiceLinks
Boolean PODSPEC_hostIPC
Boolean PODSPEC_hostNetwork
Boolean PODSPEC_hostPID
Boolean PODSPEC_hostUsers
String PODSPEC_hostname
String PODSPEC_nodeName
Boolean PODSPEC_Volumes cardinality [0..*]
    mandatory
        String VOLUMES_name
        Boolean VOLUMES_type {abstract}
    or
        Boolean emptyDir
        Boolean hostPath
        Boolean persistentVolumeClaim
        Boolean configMap
        Boolean secret
        Boolean nfs
        Boolean awsElasticBlockStore
        Boolean gcePersistentDisk
        Boolean azureDisk
        Boolean csi
        Boolean azureFile
        Boolean cephfs
        Boolean cinder
        Boolean downwardAPI
        Boolean ephemeral
        Boolean fc
        Boolean flexVolume
        Boolean flocker
        Boolean glusterfs

```

```

        Boolean iscsi
        Boolean photonPersistentDisk
        Boolean portworxVolume
        Boolean projected
        Boolean quobyte
        Boolean rbd
        Boolean scaleIO
        Boolean storageos
        Boolean vsphereVolume
Boolean ServiceSpec {abstract}
    optional
        Boolean SERVICESPEC_allocateLoadBalancerNodePorts
        Boolean SERVICESPEC_clusterIP {abstract}
            alternative
                Boolean CLUSTERIP_none
                Boolean CLUSTERIP_emptyString
                String CLUSTERIP_IP
        Boolean SERVICESPEC_clusterIPs {abstract}
            alternative
                Boolean CLUSTERIPS_none
                Boolean CLUSTERIPS_emptyString
                String CLUSTERIPS_IPs
        String SERVICESPEC_externalIPs
        String SERVICESPEC_externalName
        Boolean SERVICESPEC_externalTrafficPolicy
            alternative
                Boolean EXTERNALTRAFFICPOLICY_Local
                Boolean EXTERNALTRAFFICPOLICY_Cluster
        Integer SERVICESPEC_healthCheckNodePort
        Boolean SERVICESPEC_internalTrafficPolicy
            alternative
                Boolean INTERNALTRAFFICPOLICY_Local
                Boolean INTERNALTRAFFICPOLICY_Cluster
        Boolean SERVICESPEC_ipFamilies
            optional
                Boolean IPFAMILIES_IPv4
                Boolean IPFAMILIES_IPv6
        Boolean SERVICESPEC_ipFamilyPolicy
            alternative
                Boolean SingleStack
                Boolean PreferDualStack
                Boolean RequireDualStack

```



```

String SERVICESPEC_loadBalancerClass
Boolean SERVICESPEC_Ports cardinality [0..*]
    mandatory
        Integer SERVICESPEC_PORTS_port
    optional
        String SERVICESPEC_PORTS_name
        Boolean SERVICESPEC_PORTS_protocol {abstract}
            alternative
                Boolean PORTS_PROTOCOL_tcp
                Boolean PORTS_PROTOCOL_udp
                Boolean PORTS_PROTOCOL_sctp
        Integer SERVICESPEC_PORTS_nodePort
        String SERVICESPEC_PORTS_targetPort
Boolean publishNotReadyAddresses
Boolean SPEC_Selector cardinality [0..*]
    mandatory
        String SELECTOR_key
        String SELECTOR_value
Boolean SERVICESPEC_sessionAffinity
    alternative
        Boolean SESSIONAFFINITY_ClientIP
        Boolean SESSIONAFFINITY_None
Boolean SERVICESPEC_sessionAffinityConfig
    mandatory
        Boolean SESSIONAFFINITYCONFIG_clientIP
            mandatory
                Integer CLIENTIP_timeoutSeconds
String SERVICESPEC_trafficDistribution
Boolean SERVICESPEC_Type
    alternative
        Boolean TYPE_ExternalName
        Boolean TYPE_ClusterIP
        Boolean TYPE_NodePort
        Boolean TYPE_LoadBalancer
Boolean ConfigMapSpec {abstract}
Boolean SecretSpec {abstract}

constraints
    // Restricciones de Group
    (GROUP_apiextensions | GROUP_apiregistration | GROUP_apps | GROUP_authorization | GROUP_batch |
GROUP_coordination | GROUP_core | GROUP_discovery | GROUP_events | GROUP_node | GROUP_policy |
GROUP_rbac_authorization | GROUP_scheduling) => VERSION_v1
    (GROUP_admission_registration | GROUP_authentication) => VERSION_v1 | VERSION_v1beta1 | VERSION_v1alpha1

```

```

(GROUP_certificates | GROUP_internal_apiserver | GROUP_networking | GROUP_storage) => VERSION_v1 |
VERSION_v1alpha1

GROUP_autoscaling => VERSION_v2 | VERSION_v1

GROUP_flowcontrol_apiserver => VERSION_v1beta3 | VERSION_v1

GROUP_resource => VERSION_v1alpha2 | VERSION_v1

// Restricciones de Kind

(KIND_DaemonSet | KIND_Deployment | KIND_ReplicaSet | KIND_StatefulSet | KIND_ControllerRevision) =>
GROUP_apps

(KIND_Container | KIND_Pod | KIND_ReplicationController | KIND_Service | KIND_Endpoints | KIND_ConfigMap |
KIND_Secret | KIND_PersistentVolumeClaim | KIND_Volume | KIND_LimitRange | KIND_PodTemplate | KIND_Binding |
KIND_ComponentStatus | KIND_Namespace | KIND_Node | KIND_PersistentVolume | KIND_ResourceQuota |
KIND_ServiceAccount) => GROUP_core

(KIND_CronJob | KIND_Job) => GROUP_batch

KIND_EndpointSlice => GROUP_discovery

(KIND_Ingress | KIND_IngressClass | KIND_IPAddress | KIND_ServiceCIDR | KIND_NetworkPolicy) =>
GROUP_networking

(KIND_CSIDriver | KIND_CSINode | KIND_StorageClass | KIND_CSISStorageCapacity | KIND_VolumeAttachment |
KIND_VolumeAttributesClass) => GROUP_storage

(KIND_ClusterTrustBundle | KIND_CertificateSigningRequest) => GROUP_certificates

KIND_CustomerResourceDefinition => GROUP_apiextensions

KIND_Event => GROUP_events

KIND_HorizontalPodAutoscaler => GROUP_autoscaling

(KIND_MutatingWebhookConfiguration | KIND_ValidatingWebhookConfiguration | KIND_ValidatingAdmissionPolicy
| KIND_ValidatingAdmissionPolicyBinding) => GROUP_admission_registration

(KIND_PodSchedulingContext | KIND_ResourceClaim | KIND_ResourceClaimTemplate | KIND_ResourceClass |
KIND_ResourceClaimParameters | KIND_ResourceSlice) => GROUP_resource

KIND_PodDisruptionBudget => GROUP_policy

KIND_PriorityClass => GROUP_scheduling

KIND_APIService => GROUP_apiregistration

(KIND_LocalSubjectAccessReview | KIND_SelfSubjectAccessReview | KIND_SelfSubjectRulesReview |
KIND_SubjectAccessReview) => GROUP_authorization

(KIND_FlowSchema | KIND_PriorityLevelConfiguration) => GROUP_flowcontrol_apiserver

KIND_Lease => GROUP_coordination

KIND_RuntimeClass => GROUP_node

(KIND_SelfSubjectReview | KIND_TokenRequest | KIND_TokenReview) => GROUP_authentication

KIND_StorageVersion => GROUP_internal_apiserver

(KIND_ClusterRole | KIND_ClusterRoleBinding | KIND_Role | KIND_RoleBinding) => GROUP_rbac_authorization

// Resto de restricciones

(TYPE_NodePort | TYPE_LoadBalancer) => (SERVICESPEC_PORTS_nodePort > 0)

KIND_Pod => PodSpec

KIND_Deployment => DeploymentSpec & PodSpec

KIND_Service => ServiceSpec

KIND_ConfigMap => ConfigMapSpec

KIND_Secret => SecretSpec

(maxUnavailable == 0) => (maxSurge > 0)

DeploymentSpec => PodSpec

```

```

KIND_Service => ServiceSpec
SERVICESPEC_clusterIP => !SERVICESPEC_clusterIPs
SERVICESPEC_clusterIPs => !SERVICESPEC_clusterIP
SERVICESPEC_clusterIP => !TYPE_ExternalName
(SERVICESPEC_externalName == '') => TYPE_ExternalName
SERVICESPEC_ipFamilyPolicy => !TYPE_ExternalName
SERVICESPEC_ipFamilies => TYPE_ClusterIP | TYPE_NodePort | TYPE_LoadBalancer
(SERVICESPEC_healthCheckNodePort > 0) => TYPE_LoadBalancer & EXTERNALTRAFFICPOLICY_Local
(SERVICESPEC_loadBalancerClass == '') => TYPE_LoadBalancer
SESSIONAFFINITY_ClientIP => (CLIENTIP_timeoutSeconds > 0) & (CLIENTIP_timeoutSeconds < 86401)
PODSPEC_activeDeadlineSeconds > 0
LIVENESSPROBE_GRPC_port > 0 & LIVENESSPROBE_GRPC_port < 65535
LIVENESSPROBE_HTTPGET_port > 0 & LIVENESSPROBE_HTTPGET_port < 65535
LIVENESSPROBE_failureThreshold > 0
LIVENESSPROBE_periodSeconds > 0
LIVENESSPROBE_successThreshold > 0
LIVENESSPROBE_terminationGracePeriodSeconds > 0
LIVENESSPROBE_timeoutSeconds > 0
CONTAINER_PORTS_hostPort > 0 & CONTAINER_PORTS_hostPort < 65535
READINESSPROBE_GRPC_port > 0 & READINESSPROBE_GRPC_port < 65535
READINESSPROBE_HTTPGET_port > 0 & READINESSPROBE_HTTPGET_port < 65535
READINESSPROBE_failureThreshold > 0
READINESSPROBE_periodSeconds > 0
READINESSPROBE_successThreshold > 0
READINESSPROBE_terminationGracePeriodSeconds > 0
READINESSPROBE_timeoutSeconds > 0
STARTUPPROBE_GRPC_port > 0 & STARTUPPROBE_GRPC_port < 65535
STARTUPPROBE_HTTPGET_port > 0 & STARTUPPROBE_HTTPGET_port < 65535
STARTUPPROBE_failureThreshold > 0
STARTUPPROBE_periodSeconds > 0
STARTUPPROBE_successThreshold > 0
STARTUPPROBE_terminationGracePeriodSeconds > 0
STARTUPPROBE_timeoutSeconds > 0

```

# Anexo B

## Comandos para ejecutar los scripts

El script de generación de ficheros para Kubernetes acepta las siguientes opciones:

- `--config` ó `-c`: Ruta al archivo de configuración del FM (obligatorio).
- `--map` ó `-m`: Ruta al archivo del mapeo (obligatorio).
- `--template` ó `-t`: Ruta al archivo plantilla (obligatorio).
- `--kubernetes` ó `-k`: Opción de bandera para indicar si se debe comprobar la validez del resultado para Kubernetes (opcional).
- `--details`: Opción de bandera para dar detalles de optimización del archivo YAML (opcional).
- `--dockerfile` ó `-d`: Opción de bandera para indicar si se debe comprobar la validez del resultado para Dockerfile (opcional).

Un ejemplo de la ejecución de este script es el siguiente:

```
python main_resolve_variability.py -c config.yaml -m mapping.yaml -t template.yaml
```