A. PROPUESTA PROYECTO DE INVESTIGACIÓN INTERNO SIN FINANCIAMIENTO O AUTOGESTIONADO

1. TIPO DE INVESTIGACIÓN

Básica		**Aplicada**	**X**

2. DEPARTAMENTO O INSTITUTO

1. Facultad de Ingeniería de Sistemas (FIS)

2. Departamento de Informática y Ciencias de la Computación (DICC)

3. LÍNEA(S) DE INVESTIGACIÓN:

1. DICC-A3-L1. Computación Aplicada a la Ingeniería de Software

2. DICC-A3-L2. Creación y Gestión del Software

4. TÍTULO DEL PROYECTO *(mínimo 10 palabras):*

Modelado de infraestructura para el aprovisionamiento de servicios IaaS en un modelo de distribución de software SaaS.

5. RESUMEN *(máximo 200 palabras)*

DevOps es un paradigma que propone optimizar la entrega de software a través de un conjunto de principios, prácticas y herramientas. La Infraestructura como Código (*Infrastructure as Code*, IaC) es la piedra angular de DevOps y propone la automatización de la infraestructura utilizando prácticas de desarrollo de software. El enfoque IaC define en un *script* las instrucciones para crear y ejecutar recursos de infraestructura. La computación en la nube se ha convertido en el principal modelo de pago por uso utilizado para conseguir servicios en la nube en un corto tiempo. En particular, la Infraestructura como un Servicio (*Infrastructure as a Service*, IaaS) puede aprovechar el enfoque IaC para automatizar el aprovisionamiento de infraestructura en la nube. Sin embargo, la diversidad de lenguajes de scripting de las herramientas IaC junto con la heterogeneidad del tipo de infraestructura que ofrece cada proveedor de servicios IaaS, han ocasionado que utilizar *scripts* sea una actividad lenta y propensa a errores. Para mitigar estos problemas, se propone una herramienta para *abstraer* la complejidad de utilizar lenguajes de scripting, *automatizar* la generación de *scripts* de aprovisionamiento y modelar la infraestructura como un servicio SaaS (*Software as a Service*).

6. PALABRAS CLAVE *(4-6)*

Cloud Computing; Model-Driven Engineering; Infrastructure as Code; Cloud Infrastructure Provisioning; Infrastructure as a Service; Software as a Service.

7. DISCIPLINA CIENTÍFICA (*Marque X, solamente una opción)*

Ciencias Naturales y Exactas;	
Ingeniería y Tecnologías;	X
Ciencias Médicas;	
Ciencias Agrícolas;	
Ciencias Sociales;	
Humanidades	

8. OBJETIVO SOCIOECONÓMICO *(Marque X, solamente una opción)*

Exploración y explotación del medio terrestre;	
Ambiente;	
Exploración y Explotación del espacio;	
Transporte, telecomunicaciones y otras infraestructuras;	
Energía;	
Producción y tecnología industrial;	X
Salud;	
Agricultura;	
Educación;	
Cultura, ocio, religión y medios de comunicación;	
Sistemas políticos y sociales, estructuras y procesos;	
Defensa;	
Avance general del conocimiento: I+D financiada con los Fondos Generales de Universidades (FGU);	
Avance general del conocimiento: I+D financiados con otras fuentes.	

9. OBJETIVOS

9.1. OBJETIVO GENERAL

Rediseñar la herramienta *ad-hoc* ARGON para que funcione bajo el modelo de distribución de software SaaS (*Software as a Service*) y permita un modelado *agnóstico* de la infraestructura para el aprovisionamiento de servicios IaaS (*Infrastructure as a Service*).

9.2. OBJETIVOS ESPECÍFICOS

a. Desarrollar un lenguaje de modelado específico de dominio (*Domain-Specific Language*) *agnóstico* para modelar la infraestructura de los diferentes proveedores de servicios IaaS (*Infrastructure as a Service*).

b. Desarrollar un motor de transformaciones de modelo a texto (*Model-to-Text*) para generar scripts de aprovisionamiento para las herramientas IaC (*Infrastructure as Code*).

c. Validar la solución propuesta mediante un prototipo del artefacto en el problema del contexto.

10. HIPÓTESIS *(opcional)*

a. Un lenguaje de modelado específico de dominio (*Domain-Specific Language*) *agnóstico* modela la infraestructura de los diferentes proveedores de servicio IaaS (*Infrastructure as a Service*).

b. Un motor de transformaciones de modelo a texto (*Model-to-Text*) genera scripts de aprovisionamiento para las herramientas IaC (*Infrastructure as Code*).

c. Un prototipo permite validar la solución propuesta en el problema del contexto.

11. DETALLE DE LOS RESULTADOS ESPERADOS *(con relación a los objetivos)*

a. Un lenguaje de modelado específico de dominio (*Domain-Specific Language*) *agnóstico* para modelar la infraestructura de los diferentes proveedores de servicios IaaS (*Infrastructure as a Service*).

b. Un motor de transformaciones de modelo a texto (*Model-to-Text*) para generar scripts de aprovisionamiento —a partir de un modelo *agnóstico* de infraestructura— para las herramientas IaC (*Infrastructure as Code*).

c. Un articulo científico con los resultados del Proyecto Interno de Investigación (PII) enviado a un congreso indexado en SCOPUS o en una revista científica indexada en SCIMAGO-SCOPUS/WoS/SCIELO/Latindex Catálogo.

12. PRODUCTOS ESPERADOS

Tipo de Producto:	Marcar con una "X"
a. Disertación a la Comunidad Politécnica (obligatorio);	X
b. Presentación de un artículo en formato de la Revista Politécnica (obligatorio);	X
c. Proyecto de Titulación;	
d. Aplicación tecnológica construida o implementada;	
e. Patente presentada;	
f. Perfil de proyecto de mayor impacto científico, técnico, pedagógico o de innovación.	
g. Publicaciones científicas indexada en SCIMAGO-SCOPUS/WoS/SCIELO/Latindex Catálogo o un artículo en congreso indexado en SCOPUS.	X

13. IMPACTO DE LA INVESTIGACIÓN (*científico, social, económico u otros)*

El impacto del proyecto de investigación tiene relación directa con la educación. En particular, en los campos de la Ingeniería de Software Dirigida por Modelos (*Model-Driven Engineering*, MDE) y de la Computación en la Nube (*Cloud Computing*). Los educadores enfrentan el reto de seleccionar cuál herramienta es la más adecuada para enseñar los fundamentos de la Computación en la Nube y MDE. Por un lado, la herramienta propuesta aprovechará las técnicas dirigidas por modelos de MDE para *abstraer* las capacidades de la nube con el fin de modelar su infraestructura y *automatizar* las transformaciones de modelos para generar scripts de aprovisionamiento. Por otro lado, un lenguaje de modelado específico de dominio (*Domain-Specific Language*, DSL) permitirá a los estudiantes modelar la infraestructura de los servicios IaaS (*Infrastructure as a Service*). El DSL es una alternativa para mejorar la comprensión de los recursos de infraestructura de la nube. En particular, los resultados de la investigación podrían guiar a los educadores a centrarse en los aspectos específicos de la herramienta para ayudar a mejora la comprensión —de los estudiantes— del modelo de servicios IaaS, la computación en la nube, el aprovisionamiento de infraestructura y MDE.

Por otro lado, el proyecto de investigación tendría una implicación práctica y sería relevante para la industria. Los profesionales e investigadores consideran que la Infraestructura como Código (*Infrastructure as Code*, IaC) es el pilar fundamental sobre el cual se implementan las prácticas de DevOps. La herramienta podría ser útil para *modelar* los recursos de infraestructura, *abstraer* la complejidad de utilizar lenguajes de scripting y *generar* scripts de aprovisionamiento para diferentes herramientas IaC. Esto puede reducir el tiempo y el esfuerzo necesarios para escribir scripts de aprovisionamiento para diferentes herramientas IaC y reducir la aparición de defectos en los scripts. Por tanto, los resultados podrían ser de interés para todas aquellas empresas que planean adoptar herramientas IaC para definir recursos de infraestructura en la nube.

14. ESTADO DEL ARTE, E INVESTIGACIONES PREVIAS DEL EQUIPO *(máximo tres carillas)*

El propósito del Proyecto de Investigación Interno (PII) es abstraer la complejidad de definir la infraestructura de los servicios IaaS (*Infrastructure as a Service*) y, en consecuencia, mejorar el aprovisionamiento de infraestructura en la nube utilizando técnica dirigidas por modelos. A continuación, se presenta un análisis de los trabajos de investigación que abordan la Infraestructura como Código (*Infrastructure as Code*, IaC) y la Ingeniería de Software Dirigida por Modelos (*Model-Driven Engineering*, MDE).

La **Tabla 1** presenta las publicaciones que utilizan IaC y MDE. La primera columna representa la indexación de cada publicación como "E#", por ejemplo, el índice "E8" se refiere a la publicación "*Model-Driven Continuous Deployment for Quality DevOps*". La segunda columna muestra los autores de los trabajos relacionado junto con el año de publicación. La tercera columna presenta el título de la publicación.

Tabla 1. Lista de publicaciones que utilizan IaC y MDE

Índice	Autor(es)	Publicación
E1	(Wettinger *et al.*, 2013)	Integrating Configuration Management with Model-Driven Cloud Management Based on TOSCA
E2	(Rossini, 2015)	Cloud Application Modelling and Execution Language (CAMEL) and the PaaSage Workflow
E3	(Gouw *et al.*, 2015)	On the Integration of Automatic Deployment into the ABS Modeling Language
E4	(Marrone & Nardone, 2015)	Automatic Resource Allocation for High Availability Cloud Services
E5	(Glaser, 2016)	Domain Model Optimized Deployment and Execution of Cloud Applications with TOSCA
E6	(Weerasiri *et al.*, 2016)	A Model-Driven Framework for Interoperable Cloud Resources Management
E7	(Chen *et al.*, 2016)	MORE: A Model-driven Operation Service for Cloud-based IT Systems
E8	(Artač *et al.*, 2016)	Model-Driven Continuous Deployment for Quality DevOps
E9	(Nitto *et al.*, 2017)	Model-Driven Development and Operation of Multi-Cloud Applications
E10	(Alipour & Liu, 2018)	Model Driven Deployment of Auto-scaling Services on Multiple Clouds
E11	(Bhattacharjee *et al.*, 2018)	CloudCAMP: Automating the Deployment and Management of Cloud Services
E12	(Artac *et al.*, 2018)	Infrastructure-as-Code for Data-Intensive Architectures: A Model-Driven Development Approach
E13	(Ferry & Rossini, 2018)	CloudMF: Model-Driven Management of Multi-Cloud Applications
E14	(Casale *et al.*, 2019)	RADON: rational decomposition and orchestration for serverless computing
E15	(Brabra *et al.*, 2019)	Model-Driven Orchestration for Cloud Resources

Las publicaciones E2, E3, E9 y E12 pertenecen a proyectos europeos, tales como, PaaSage European Project, ENVISAGE FP7 European Project y DICE H2020 European Project. En consecuencia, los proyectos de investigación relacionados con IaC y MDE tienen relevancia e importancia en la comunidad académica.

Las publicaciones E5, E6 y E15 estudian el aprovisionamiento de infraestructura, mientras que todas las publicaciones (E1-E15) abordan el despliegue de aplicaciones. Adicionalmente, la publicación E9 realiza el monitoreo de aplicaciones desplegadas en la nube. La publicación E5 realiza el aprovisionamiento de máquinas virtuales, mientras que E6 y E15 centran sus esfuerzos en el aprovisionamiento de contendedores Docker. Es importante definir —y tener en cuenta la diferencia— entre despliegue y aprovisionamiento. Por un lado, el *aprovisionamiento* se refiere a escribir y ejecutar código para definir, crear y actualizar la infraestructura. Por otro lado, el *despliegue* se refiere a escribir y ejecutar código para definir e implementar las aplicaciones de software en la infraestructura aprovisionada.

Las publicaciones E1, E2, E3, E5, E7, E8, E11, E12 y E15 afrontan el problema de la heterogeneidad de los lenguajes de scripting para definir, configurar y ejecutar el aprovisionamiento de infraestructura y el despliegue de aplicaciones. Los autores resaltan que utilizar lenguajes de scripting para definir y ejecutar la infraestructura

y aplicaciones es una tareas lenta y propensa a errores. En este caso, el uso de lenguajes de scripting requiere experiencia técnica y conocimientos en el dominio y en muchos casos un tiempo considerable de aprendizaje.

Con respecto a las soluciones propuestas, las publicaciones E2, E3, E6, E7, E8, E9, E10, E11, E13 y E14 proponen un lenguaje de dominio específico (*Domain-Specific Language*, DSL) para modelar el despliegue de aplicaciones en la nube. Los autores en E4 utilizan un perfil de UML (*Unified Modeling Language*) para modelar y analizar sistemas en tiempo real, mientras que los autores en E12 utilizan un perfil de UML para modelar la implementación de arquitecturas intensivas de datos. Por otro lado, en E1 se plantea la integración de la gestión de la configuración y la implementación de los servicios en la nube, mientras que en E5 propone un marco (*Framework*) para escalar las aplicaciones en la nube de acuerdo con las demandas del modelo de dominio de los usuarios. Los autores en E8 y E15 proponen un mapeo y traducción desde los recursos de la nube definidos en TOSCA (OASIS, 2013) hacia herramientas de la comunidad DevOps.

En cuanto a las técnicas MDE, las publicaciones E2, E3, E4, E6, E7, E8, E9, E11, E12, E13 y E14 utilizan modelos para representar en un alto nivel de abstracción la infraestructura, aplicaciones y configuraciones. En E2 y E13 los autores utilizan modelos en tiempo de ejecución (model@run-time) para representar en un modelo el sistema en ejecución. Además, mediante model@runtime una modificación en el modelo se promulga hacia el sistema y un cambio en el sistema se refleja automáticamente en el modelo. Por otro lado, las publicaciones E1, E5, E8 y E15 utilizan transformaciones de modelo a texto (*Model-to-Text*, M2T) para convertir modelos TOSCA en scripts para herramientas IaC, mientras que las publicaciones E7, E12, y E14 utilizan transformaciones M2T para convertir los modelos de infraestructura, aplicaciones y configuraciones en scripts para herramientas IaC. En este contexto, en E9, E10, E13 y E15 los autores utilizan y adaptan la arquitectura dirigida por modelos (*Model-Driven Architecture*, MDA) para definir en un modelo independiente de la nube (*Cloud Independent Model*) los requisitos de la aplicación, luego en un modelo independiente del proveedor de la nube (*Cloud Provider Independent Model*) se definen aspectos generales de la nube y finalmente en un modelo especifico del proveedor de la nube (*Cloud Provider Specific Model*) se definen aspectos específicos de un proveedor de la nube.

El principal problema que abordan los trabajos de investigación relacionados con IaC y MDE es la heterogeneidad de los lenguajes de scripting para definir, configurar y ejecutar el aprovisionamiento de infraestructura y el despliegue de aplicaciones. Todos los trabajos centran sus esfuerzos en mejorar el despliegue de aplicaciones en la nube, mientras que solo tres investigan el aprovisionamiento de máquinas virtuales y contenedores. Existen nueve trabajos relacionados que emplean modelos como una solución para definir la infraestructura, aplicaciones y configuraciones. En cambio, cinco trabajos de investigación utilizan y adaptan MDA para generar artefactos IaC mediante transformaciones de modelos y brindan soporte al despliegue de aplicaciones en la nube. Como resultado, los investigadores han centrado sus esfuerzos en mejorar la entrega continua de aplicaciones en la nube, pero existe una brecha que cubrir respecto a la definición y aprovisionamiento de la infraestructura en nube.

En cuanto a los trabajos de investigación realizados previamente por el director y colaborador del proyecto, es importante mencionar su experiencia en MDE, IaC, Computación en la Nube y en la Interacción Humano Computador. Por un lado, se realizó una primera aproximación de una herramienta *ad-hoc* de modelado de infraestructura para el aprovisionamiento de infraestructura llamada ARGON (Sandobalín et al., 2017b) que utiliza IaC y MDE. Se propuso un encadenamiento de herramientas DevOps (Sandobalín *et al.*, 2017a) para la automatización del aprovisionamiento de infraestructura en la nube junto con un enfoque para el modelado y aprovisionamiento de infraestructura en múltiples nubes (Sandobalín *et al.*, 2018). Adicionalmente, se realizó un primer estudio empírico (Sandobalin *et al.*, 2020) sobre la efectividad de la herramientas que soportan IaC desde una comparación de herramientas centradas en el código versus dirigidas por modelos. Por otro lado, se ha realizado estudios para mejorar la usabilidad en interfaces de usuarios (Iñiguez-Jarrín *et al.*, 2020) y se ha definido unos patrones de diseño de interacción (Iñiguez-Jarrín, Panach, & López, 2018). Además, se ha propuesto un diseño centrado en el usuario para interfaces de búsqueda de literatura científica (Iñiguez-Jarrín, Panach, & Pastor, 2018).

15. DESCRIPCIÓN DETALLADA DEL PROYECTO, INCLUIDO METODOLOGÍA *(máximo tres carillas)*

DevOps (*Development & Operations*) es un paradigma que está promoviendo una colaboración continua entre los desarrolladores y el personal de operaciones a través de un conjunto de principios, prácticas y herramientas para optimizar el tiempo de entrega del software (Humble & Farley, 2010). La piedra angular de DevOps es la Infraestructura como Código (*Infrastructure as Code*, IaC). IaC es un enfoque para la automatización de la infraestructura basado en práctica de desarrollo de software (Morris, 2016). El objetivo de IaC es definir en un script todas las instrucciones necesarias para crear, actualizar y ejecutar recursos de infraestructura. Las herramientas IaC utilizan los scripts para orquestar el aprovisionamiento y configuración de la infraestructura. Los principios del enfoque IaC ofrecen muchas ventajas para definir, actualizar y ejecutar recursos de infraestructura, tales como crear, destruir, reemplazar, redimensionar y remover fácilmente la infraestructura, soportando el principio de *reproducibilidad* para construir rápidamente múltiples entornos idénticos (Morris, 2016).

La computación en la nube se ha convertido en el principal modelo de pago por uso que utilizan los profesionales e investigadores para obtener servicios en la nube en poco tiempo. La computación en la nube se compone de servicios basados en hardware, donde la administración del hardware es altamente abstracta y la capacidad de la infraestructura es altamente elástica (Buyya *et al.*, 2011). Los servicios en la nube se dividen de acuerdo con el nivel de abstracción de su modelo de servicio, es decir, Infraestructura como Servicio (*Infrastructure as a Service*, IaaS), Plataforma como Servicios (*Platform as a Service*, PaaS) y Software como Servicio (*Software as a Service*, SaaS). Por un lado, IaaS es un modelo de servicio con recursos virtualizados (computación, almacenamiento y redes) que permite el aprovisionamiento de recursos de infraestructura bajo pedido, junto con varias opciones de configuración de los sistemas operativos y las características de hardware personalizables (CPU, RAM y almacenamiento). Por otro lado, los usuarios finales pueden acceder a los servicios proporcionados por el modelo SaaS a través de portales web.

Existen procesos DevOps que aprovechan los servicios ofrecidos por los proveedores de la nube, tales como computación, redes, almacenamiento y elasticidad. En este contexto, el aprovisionamiento de infraestructura en la nube utiliza el enfoque IaC para definir en un script la creación, actualización y ejecución de recursos de infraestructura en diferentes proveedores de servicios IaaS. La computación en la nube y el enfoque IaC están causando cambios en la industria (Brikman, 2017), tales como:

- En lugar de administrar centros de datos, muchas empresas se están moviendo su infraestructura hacia la nube.
- En lugar de invertir en hardware, los equipos de operaciones pasan todo su tiempo trabajando en software.
- En lugar de configurar servidores y conectar cables de red, muchos administradores de sistemas están escribiendo código.

En la actualidad, las organizaciones que utilizan una cantidad considerable de tecnología están lidiando con la complejidad de cómo lograr que el desarrollo de software y las actividades de operaciones sean compactas, transparentes y totalmente integradas (Mann *et al.*, 2018). En este contexto, IaC proporciona un enfoque de automatización de la infraestructura que acelera las prácticas de integración y despliegue continuos de software. Si bien los investigadores y profesionales han hecho una enorme contribución a DevOps, aún existen brechas y desafíos por resolver relacionados con el aprovisionamiento infraestructura en los diferentes proveedores de servicios IaaS.

Para el desarrollo del Proyecto de Investigación Interno (PII) se utilizará la metodología de investigación *Design Science* (Wieringa, 2014). En un proyecto *Design Science* el objeto de estudio es un artefacto en contexto, y sus dos actividades principales son el diseño y la investigación del artefacto en contexto. El artefacto debe estar diseñado para interactuar en un problema del contexto, a fin de mejorar algo en dicho contexto. En este caso, el *artefacto* es una herramienta SaaS de modelado de infraestructura para el aprovisionamiento de los servicios IaaS y el *contexto* consiste en los desarrolladores y el personal de operaciones que necesitan definir, actualizar y ejecutar la infraestructura en diferentes proveedores de servicios IaaS. El *problema del contexto* es la diversidad de herramientas IaC junto con la variedad de lenguajes de scripting para definir la infraestructura y la heterogeneidad del tipo de infraestructura que ofrecen los diferentes proveedores de servicios IaaS.

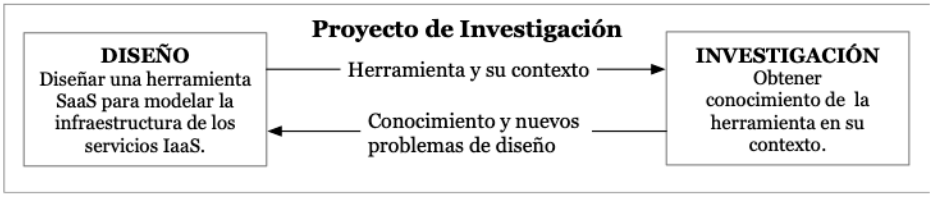

Figura 1. Proyecto *Design Science* de la herramienta SaaS para modelar la infraestructura de servicios IaaS.

La Figura 1 presenta las dos actividades principales de un proyecto *Design Science:* diseño e investigación. Por un lado, la actividad de diseño permite definir problemas de diseño para (re)diseñar un artefacto que contribuye de alguna manera a alcanzar un objetivo y que está relacionado con el diseño de la herramienta. Por otro lado, en la actividad de investigación se plantean preguntas de investigación relacionadas con la obtención de conocimiento acerca de la interacción entre la herramienta y su contexto de aplicación.

En el caso de la *actividad de diseño*, se propone *rediseñar* la herramienta ARGON (Sandobalín *et al.*, 2017b). ARGON es una herramienta *ad-hoc* de modelado de infraestructura para el aprovisionamiento en la nube que implementa el enfoque IaC mediante la Ingeniería de Software Dirigida por Modelos (*Model-Driven Engineering*, MDE). Por un lado, ARGON define diferentes metamodelos para representar la infraestructura de los diferentes proveedores de servicios IaaS. Por otro lado, ARGON utiliza transformaciones de modelo a modelo (*Model-to-Model*, M2M) para instanciar un modelo de infraestructura —que esté conforme a su metamodelo— para un proveedor de servicios IaaS en particular. El *principal problema* de mantener diferentes metamodelos y transformaciones M2M es su mantenimiento y escalabilidad cuando los proveedores de servicios IaaS actualizan o modifican su infraestructura, lo que ocasiona grandes cambios en el código fuente de ARGON. Además, ARGON fue diseñado con una arquitectura dirigida por plugins para funcionar en el entorno de desarrollo integrado Eclipse (Steinberg *et al.*, 2008). En consecuencia, se propone definir un metamodelo *agnóstico* que represente la infraestructura de los diferentes proveedores de servicios IaaS. De igual manera, se propone rediseñar ARGON para que funcione bajo el modelo de distribución de software SaaS. Además, se definirán transformaciones de modelo a texto (*Model-to-Text*, M2T) para generar —a partir de un modelo *agnóstico* de infraestructura— los scripts de aprovisionamiento para las diferentes herramientas IaC. En este caso, se debe *abstraer* las características particulares de cada lenguaje de scripting para generar las plantillas de transformación M2T.

Para realizar la *validación* del artefacto se utilizará la Investigación de Acción Técnica (*Technical Action Research*, TAR). En este caso, TAR utiliza un prototipo del artefacto en un problema del contexto para validar la solución propuesta (Wieringa, 2014). TAR realiza estudios de caso único, porque cada uso individual del artefacto se estudia como un caso único. La diferencia entre TAR y otras formas de investigación-acción es que TAR está impulsada por artefactos.

En el caso de la actividad de *investigación*, se propone como **trabajo futuro** realizar una familia experimentos controlados con estudiantes y/o profesionales de Ingeniería de Software, con el fin de estudiar —mediante la *Ingeniería de Software Empírica*— la interacción de la herramienta y su contexto de aplicación. Es importante mencionar que se requiere de al menos 12 meses para realizar las tareas relacionadas a una familia de experimentos, tales como el diseño de cada experimento, experimentos pilotos, entrenamiento a los participantes, ejecución del experimento, validación de los datos, análisis estadístico de los datos, interpretación de los resultados y redacción de un artículo científico. En consecuencia, el presente PII propone centrar sus esfuerzos en el *rediseño* de ARGON y obtener una herramienta SaaS de modelado de infraestructura para el aprovisionamiento de servicios IaaS.

En un proyecto *Design Science* un ciclo de ingeniería es un proceso racional de resolución de problemas (Wieringa, 2014). El resultado de un ciclo de ingeniería es un ciclo de diseño de un *tratamiento* validado —a través de un ciclo empírico— que se transfiere al mundo real, se utiliza y se evalúa. En este contexto, un *tratamiento* es la interacción entre el artefacto y el contexto del problema.

El proyecto de investigación centrará sus esfuerzos en el ciclo de diseño —dejando el ciclo empírico como un trabajo futuro— de acuerdo con las siguientes tareas fundamentales:

- **T1. Investigación del Problema.** Se investiga los fenómenos que hay que mejorar en el problema del contexto.
- **T2. Diseño del tratamiento.** Se realiza el diseño del artefacto para tratar el problema del contexto.
- **T3. Validación del tratamiento.** Se realiza la validación del tratamiento para evaluar si el diseño mejora el problema del contexto.

En el contexto del proyecto *Design Science*, en la **T1** se define el artefacto y el contexto de la investigación, así como el planteamiento del problema que hay que mejorar. Además, se plantean los objetivos y las preguntas de investigación, así como el marco conceptual que define los conceptos —llamados *constructos*— para definir las estructuras en el artefacto y su contexto. Finalmente, una revisión de la literatura científica presenta el estado actual de los trabajos de investigación relacionados con el proyecto de investigación.

En la **T2** se especifican los requisitos del artefacto y de la interacción entre el artefacto y su contexto. Además, se estudia como los requisitos contribuyen a alcanzar los objetivos de la investigación. Finalmente, se realiza el (re)diseño y la construcción de la herramienta SaaS de modelado de la infraestructura para el aprovisionamiento de servicios IaaS.

En la **T3** se realiza la validación mediante la Investigación de Acción Técnica (*Technical Action Research*, TAR). En este caso, se utiliza un prototipo del artefacto en un problema del contexto para validar la solución propuesta.

Bibliografía

Alipour, H., & Liu, Y. (2018). Model Driven Deployment of Auto-Scaling Services on Multiple Clouds. *2018 IEEE International Conference on Software Architecture Companion (ICSA-C)*, 93–96. https://doi.org/10.1109/ICSA-C.2018.00033

Artac, M., Borovsak, T., Di Nitto, E., Guerriero, M., Perez-Palacin, D., & Tamburri, D. A. (2018). Infrastructure-as-Code for Data-Intensive Architectures: A Model-Driven Development Approach. *2018 IEEE International Conference on Software Architecture (ICSA)*, 156–165. https://doi.org/10.1109/ICSA.2018.00025

Artač, M., Borovšak, T., Di Nitto, E., Guerriero, M., & Tamburri, D. A. (2016). Model-Driven continuous deployment for quality devops. *2016 International Workshop on Quality-Aware DevOps (QUDOS)*, 40–41. https://doi.org/10.1145/2945408.2945417

Bhattacharjee, A., Barve, Y., Gokhale, A., & Kuroda, T. (2018). CloudCAMP: Automating the deployment and management of cloud services. *2018 IEEE International Conference on Services Computing (SCC)*, 237–240. https://doi.org/10.1109/SCC.2018.00038

Brabra, H., Mtibaa, A., Gaaloul, W., Benatallah, B., & Gargouri, F. (2019). Model-Driven Orchestration for Cloud Resources. *2019 IEEE International Conference on Cloud Computing (CLOUD)*, 422–429. https://doi.org/10.1109/cloud.2019.00074

Brikman, Y. (2017). *Terraform: Up and Running* (First Edit). O'Reilly Media.

Buyya, R., Broberg, J., & Gościński, A. (2011). *Cloud computing : principles and paradigms*. Wiley.

Casale, G., Artač, M., van den Heuvel, W.-J., van Hoorn, A., Jakovits, P., Leymann, F., Long, M., Papanikolaou, V., Presenza, D., Russo, A., Srirama, S. N., Tamburri, D. A., Wurster, M., & Zhu, L. (2019). RADON: rational decomposition and orchestration for serverless computing. In *SICS Software-Intensive Cyber-Physical Systems* (pp. 1–11). https://doi.org/10.1007/s00450-019-00413-w

Chen, W., Liang, C., Wan, Y., Gao, C., Wu, G., Wei, J., & Huang, T. (2016). MORE: A model-driven operation service for cloud-based IT systems. *2016 IEEE International Conference on Services Computing (SCC)*, 633–640. https://doi.org/10.1109/SCC.2016.88

Ferry, N., & Rossini, A. (2018). CloudMF: Model-Driven Management of Multi-Cloud Applications. *ACM Transactions on Internet Technology (TOIT)*, *18*(2), 16–24. https://doi.org/10.1145/3125621

Glaser, F. (2016). Domain Model Optimized Deployment and Execution of Cloud Applications with

TOSCA. In *System Analysis and Modeling. Technology-Specific Aspects of Models. SAM 2016. Lecture Notes in Computer Science* (Vol. 9959). Springer International Publishing. https://doi.org/10.1007/978-3-319-46613-2

Gouw, S. de, Lienhardt, M., Mauro, J., Nobakht, B., & Zavattaro, G. (2015). On the Integration of Automatic Deployment into the ABS Modeling Language. *Service Oriented and Cloud Computing. ESOCC 2015. Lecture Notes in Computer Science*, *9306*. https://doi.org/10.1007/978-3-319-24072-5_4

Humble, J., & Farley, D. (2010). Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation. In *Continuous Delivery* (First Edit). Addison-Wesley Professional. https://doi.org/10.1007/s13398-014-0173-7.2

Iñiguez-Jarrín, C., Panach, J. I., & López, O. P. (2018). Defining interaction design patterns to extract knowledge from big data. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. https://doi.org/10.1007/978-3-319-91563-0_30

Iñiguez-Jarrín, C., Panach, J. I., & López, O. P. (2020). Improvement of usability in user interfaces for massive data analysis: an empirical study. *Multimedia Tools and Applications*. https://doi.org/10.1007/s11042-019-08456-6

Iñiguez-Jarrín, C., Panach, J. I., & Pastor, O. (2018). User-Centered Design for Biomedical Literature Search User Interfaces. *27th International Conference on Information Systems Development*.

Mann, A., Brown, A., Stahnke, M., & Kersten, N. (2018). 2018 State of DevOps Report. In *Puppetlabs*. https://doi.org/10.1016/S0022-3913(12)00047-9

Marrone, S., & Nardone, R. (2015). Automatic resource allocation for high availability cloud services. *Procedia Computer Science*, *52*(1), 980–987. https://doi.org/10.1016/j.procs.2015.05.176

Morris, K. (2016). *Infrastructure As Code: Managing Servers in the Cloud* (First Edit). O'Reilly Media.

Nitto, E. Di, Matthews, P., Petcu, D., & Solberg, A. (2017). Model-Driven Development and Operation of Multi-Cloud Applications. In E. Di Nitto, P. Matthews, D. Petcu, & A. Solberg (Eds.), *Springer International Publishing*. Springer International Publishing. https://doi.org/10.1007/978-3-319-46031-4

OASIS. (2013). *Topology and Orchestration Specification for Cloud Applications (TOSCA)*. https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=tosca

Rossini, A. (2015). Cloud Application Modelling and Execution Language (CAMEL) and the PaaSage Workflow. *2015 European Conference on Service-Oriented and Cloud Computing (ESOCC)*, *567*, 437–439. https://doi.org/10.1007/978-3-319-33313-7

Sandobalin, J., Insfran, E., & Abrahao, S. (2020). On the effectiveness of tools to support infrastructure as code: Model-driven versus code-centric. *IEEE Access*. https://doi.org/10.1109/ACCESS.2020.2966597

Sandobalín, J., Insfrán, E., & Abrahão, S. (2017a). End-to-End Automation in Cloud Infrastructure Provisioning. *Proceedings - 26th International Conference on Information Systems Development, ISD*.

Sandobalín, J., Insfrán, E., & Abrahão, S. (2018). An Infrastructure Modeling Approach for Multi-Cloud Provisioning. *Proceedings - 27th International Conference on Information Systems Development, ISD 2018*.

Sandobalín, J., Insfrán, E., & Abrahão, S. (2017b). An Infrastructure Modelling Tool for Cloud Provisioning. *Proceedings - IEEE 14th International Conference on Services Computing, SCC*, 354–361. https://doi.org/10.1109/SCC.2017.52

Steinberg, D., Budinsky, F., Merks, E., & Paternostro, M. (2008). EMF: eclipse modeling framework. In *Engineering*. http://portal.acm.org/citation.cfm?id=1197540

Weerasiri, D., Barukh, M., Benatallah, B., & Cao, J. (2016). A Model-Driven Framework for Interoperable Cloud Resources Management. In *Service-Oriented Computing. ICSOC 2016. Lecture Notes in Computer Science* (Vol. 9936). https://doi.org/10.1007/978-3-319-46295-0

Wettinger, J., Behrendt, M., Binz, T., Breitenbücher, U., Breiter, G., Leymann, F., Moser, S., Schwertle, I., & Spatzier, T. (2013). Integrating configuration management with model-driven cloud management based

on TOSCA. *2013 International Conference on Cloud Computing and Services Science (CLOSER)*, 437–446. http://www.scopus.com/inward/record.url?eid=2-s2.0-84884491041&partnerID=40&md5=6cb394970ee2bb4f317324851be8fae8

Wieringa, R. (2014). Design Science Methodology for Information Systems and Software Engineering. In *Springer Berlin Heidelberg*. https://doi.org/10.1145/1810295.1810446

16. INFRAESTRUCTURA Y EQUIPOS

Infraestructura	**Equipos**	
Laboratorio/Recurso	**Nombre del Equipo**	**Ubicación del Equipo**
Ordenador portátil	Lenovo Thinkpad L490	Ordenador portátil de la EPN entregado a Julio Sandobalín.
Ordenador portátil	Lenovo Thinkpad L490	Ordenador portátil de la EPN entregado a Carlos Íñiguez.
Servicios IaaS de Amazon Web Services	No aplica	Al crear una cuenta en Amazon Web Services AWS, el proveedor entrega una capa gratuita de acceso por 12 meses a los servicios EC2, S3 y DynamoDB. Por lo tanto, se utilizará la capa gratuita que ofrece AWS.

B. DATOS INFORMATIVOS

1. INFORMACIÓN DEL DIRECTOR, COLABORADORES Y COLABORADORES TÉCNICOS

Apellidos y nombres	**No. de Cédula**	**HSS***	**Departamento****	**Rol**	**Título de mayor nivel y mención.**
Sandobalín Guamán Julio César	171546037-2	8	Departamento de Informática y Ciencias de la Computación	Director	Ph.D. en Informática
Íñiguez Jarrín Carlos Efraín	171590069-0	6	Departamento de Informática y Ciencias de la Computación	Colaborador	Ph.D. en Informática

*HSS =Horas Semana Semestre: Es el número de horas que se dedica por semana a la investigación.
** En el caso de que los colaboradores sean de otro departamento se debe adjuntar el aval de las horas de dedicación del Jefe de Departamento.

C. DECLARACIÓN FINAL
DECLARACIÓN DEL DIRECTOR DEL PROYECTO

El equipo de investigadores, representado por el Director del Proyecto declara lo siguiente:

- Que el presente proyecto es una creación original de mi autoría y del equipo de investigadores, y por tanto asumimos la completa responsabilidad legal en caso de que un tercero alegue la titularidad de los derechos intelectuales del proyecto, exonerando a la EPN de cualquier acción legal que se derive por esta causa.

- Que el presente proyecto no ha sido presentado en ninguna convocatoria de otra institución pública o privada. El incumplimiento será causal para que el proyecto no sea tomado en consideración.

- Que si el proyecto genera algún producto o procedimiento susceptible de obtener derechos de propiedad intelectual, de los cuales se deriven beneficios, aceptamos que éstos serán compartidos entre los investigadores y la institución o las instituciones participantes en el proyecto, conforme a lo establecido en el COESC.

- Que el equipo de investigadores y/o instituciones participantes se comprometen a mantener la confidencialidad de la información si ésta podría ser susceptible de protección por patentes, y solicitar la valoración de propiedad intelectual respectiva previa a cualquier publicación o difusión.

- Que para el caso de derechos de autor otorgamos una licencia de uso exclusivo con fines académicos para la o las instituciones participantes en el proyecto.

- Que aceptamos conocer y cumplir con la normativa vigente para la gestión de proyectos.

Firma del Director del Proyecto
Nombre: Julio Sandobalín, Ph.D.
C.I.: 1715460372

ESCUELA POLITÉCNICA NACIONAL

VICERRECTORADO DE INVESTIGACIÓN, INNOVACIÓN Y VINCULACIÓN

Proyecto de Investigación Interno

CRONOGRAMA DE ACTIVIDADES DEL PROYECTO

Título del Proyecto: **Modelado de infraestructura para el aprovisionamiento de servicios IaaS en un modelo de distribución de software SaaS**

AÑO 1

Nº	Actividad	Mes 1				Mes 2				Mes 3				Mes 4				Mes 5				Mes 6				Mes 7				Mes 8				Mes 9				Mes 10				Mes 11				Mes 12			
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1	**Investigación del Problema**	o	o	o	o	o	o	o	o																																								
1.1	Definición del artefacto, contexto y problema del contexto	x	x																																														
1.2	Objetivos y preguntas de investigación			x																																													
1.3	Marco conceptual y constructos.				x																																												
1.4	Revisión de la literatura científica					x	x	x	x																																								
2	**Diseño del tratamiento**									o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o												
2.1	Especificación de los requisitos del artefacto y su contexto									x	x	x	x																																				
2.2	Análisis de cómo los requisitos apoyan a los objetivos											x	x																																				
2.3	Diseño y construcción de un lenguaje de modelado específico de dominio agnóstico para modelar la infraestructura de los diferentes proveedores de servicios IaaS													x	x	x	x	x	x	x	x																												
2.4	Diseño y construcción de un motor de transformaciones de modelo a texto para generar scripts de aprovisionamiento para las herramientas IaC																					x	x	x	x	x	x	x	x																				
2.5	Diseño y construcción de una herramienta SaaS de modelado de infraestructura para el aprovisionamiento de servicios IaaS																													x	x	x	x	x	x	x	x												
3	**Validación del tratamiento**																																					o	o	o	o								
3.1	Validación de los conceptos abstraídos en el metamodelo agnóstico de infraestructura																																					x	x										
3.2	Validación del modelado de infraestructura como un servicio SaaS																																						x	x									
3.3.	Validación de la generación de scripts de aprovisionamiento para herramientas IaC																																								x								
4	**Escritura de un artículo científico**																																									o	o	o	o	o	o	o	o
4.1	Escritura del artículo de investigación																																									x	x	x	x	x	x	x	
4.2	Revisión del idioma inglés del artículo																																												x	x	x	x	
4.3	Envío del artículo a un congreso o revista científica																																																x