

Pontificia Universidad Javeriana Cali
Facultad de Ingeniería y Ciencias.
Maestría en Ingeniería de Software.
Proyecto de Grado.

Implementación de Infraestructura como código (IaC) con Azure Bicep para generar servicios cloud

Santiago Jaramillo Múnera

Directora: Ph.D. Luisa Rincón

11 de Diciembre de 2024



Ficha Resumen

Trabajo de Grado Maestría en Ingeniería de Software

TÍTULO: Implementación de Infraestructura como código (IaC) con Azure Bicep para generar servicios cloud

1. Énfasis: Ingeniería de Software
2. Área de trabajo: Infraestructura como código
3. Tipo de proyecto (Aplicado, Innovación, Investigación): Aplicado
4. Estudiante: Santiago Jaramillo Munera
5. Correo electrónico: santiagojm84@javerianacali.edu.co
6. Dirección y teléfono: Calle 47N 5an-37 - 3162755394
7. Director: Luisa Fernanda Rincón Pérez
8. Vinculación del director: Planta
9. Correo electrónico del director: lfrincon@javerianacali.edu.co
10. Co-Director (Si aplica): N/A
11. Grupo o empresa que lo avala (Si aplica): N/A
12. Otros grupos o empresas: N/A
13. Palabras clave(al menos 5): IaC, Bicep, Cloud, Azure, Infraestructura como Código
14. ODS que aplica al proyecto (Agenda 2030): 8 - Trabajo decente y crecimiento económico
15. Fecha de inicio: 1 de enero de 2024
16. Resumen: El proyecto tiene como objetivo crear una solución innovadora para automatizar y optimizar el proceso de creación y despliegue de servicios en la nube para los portales corporativos de la empresa TQ. Esto se logrará mediante el uso de Infraestructura como Código (IaC) y plantillas de configuración Bicep, lo que garantizará una gestión eficiente y centralizada que respalde adecuadamente el ciclo de vida de la infraestructura en la nube.

0.1. Resumen

Actualmente, la organización TQ cuenta con portales corporativos para varios de sus productos, los cuales se encuentran desplegados en la nube de Azure. El proceso de aprovisionamiento de cada portal en ambientes de desarrollo, pruebas y producción se hace de forma manual por parte de un empleado de infraestructura, lo cual implica un esfuerzo considerable y está sujeto al error humano. Adicionalmente, se conoce que se realizan copias de seguridad de la configuración de cada servicio, pero no se tiene un gobierno sobre esta información y las mismas no facilitan el proceso de despliegue de cada portal en caso de ser requerido.

El proyecto tiene como objetivo crear una solución innovadora para automatizar y optimizar el proceso de creación y despliegue de servicios en la nube para los portales corporativos de la empresa TQ. Esto se logra mediante el uso de Infraestructura como Código (IaC) y plantillas de configuración Bicep, lo que garantiza una gestión eficiente y centralizada que respalde adecuadamente el ciclo de vida de la infraestructura en la nube.

La solución propuesta integra el proceso de generación de plantillas de configuración utilizando Bicep. Además, se logra la integración mediante API con Azure DevOps para el versionamiento de objetos y el despliegue de los servicios a través de un pipeline. Todo esto se realiza de manera segura mediante una conexión entre DevOps y Azure.

Durante el proceso de evaluación, se incluyeron tres usuarios con diferentes roles en el área de TI, a los cuales se les realizó una presentación inicial donde se expuso la problemática, la propuesta y la solución generada. Luego, se les dio a estos usuarios el acceso a la herramienta y se les asignó la tarea de aprovisionar nuevos servicios. Posteriormente, se les mostró en Azure los servicios generados y se concluyó con una encuesta para evaluar la percepción de utilidad y facilidad de uso de la herramienta por parte de estos usuarios.

Los resultados obtenidos permitieron identificar que la herramienta resulta útil para el gobierno y despliegue de servicios de forma automatizada. La calificación promedio otorgada fue de 5 en una escala del 1 al 5, y se obtuvieron las siguientes recomendaciones respecto a nuevas funcionalidades que podría incorporar en un futuro:

- Ampliar el número de servicios disponibles para configurar y aprovisionar
- Generar copia de la configuración de recursos de un proyecto a otro
- Despliegue de servicios en otras plataformas como AWS y GCP

Palabras clave: IaC, Bicep, Cloud, Azure, Infraestructura como Código.

0.2. Abstract

Currently, the TQ organization has corporate portals for several of its products, which are deployed in the Azure cloud. The process of provisioning each portal in development, test, and production environments is done manually by an infrastructure employee, which involves considerable effort and is subject to human error. Additionally, it is known that backups of the configuration of each service are made, but there is no governance over this information and they do not facilitate the deployment process of each portal if required.

The project aims to create an innovative solution to automate and optimize the process of creating and deploying cloud services for the corporate portals of the company TQ. This is achieved through the use of Infrastructure as Code (IaC) and Bicep configuration templates, ensuring efficient and centralized management that properly supports the cloud infrastructure lifecycle.

The proposed solution integrates the process of generating configuration templates using Bicep. In addition, integration through API with Azure DevOps is achieved for the versioning of objects and the deployment of services through a pipeline. All of this is done securely using a connection between DevOps and Azure.

Three different roles from IT were included in the evaluation process. An initial presentation was made in which the problem, the proposal and the solution generated were presented. Then, they were given access to the management of the tool and were assigned the task of provisioning new services. Subsequently, the services generated were shown in Azure and ended with a survey to evaluate the perception of usefulness and ease of use of the tool by these users.

The results obtained allowed us to identify that the tool is useful for the automated governance and deployment of services. The average rating given was 5 on a scale of 1 to 5, and the following recommendations were made regarding new features that could be incorporated in the future:

- Expand the number of services available for configuration and provisioning
- Generate a copy of the resource configuration from one project to another
- Deploy services on other platforms such as AWS and GCP

Keywords: IaC, Bicep, Cloud, Azure, Infrastructure as Code.

Índice general

0.1. Resumen	5
0.2. Abstract	6
1. Introducción	1
1.1. Definición del problema de investigación	2
1.2. Planteamiento del problema	4
1.3. Objetivos	4
1.3.1. Objetivo General	4
1.3.2. Objetivos específicos	4
1.4. Justificación	5
1.5. Alcance	6
2. Marco de referencia	7
2.1. Bases teóricas	7
2.1.1. Infraestructura como código	7
2.1.2. Azure Bicep	9
2.1.3. Contexto	11
2.1.4. Necesidad	11
2.2. Antecedentes	12
2.3. Resumen del capítulo	15
3. Análisis, Diseño e Implementación	16
3.1. Análisis	16
3.1.1. Proceso de elicitación	16
3.1.2. Portales corporativos	17
3.1.3. Servicios Azure corporativos	21
3.1.4. Parametrización servicios Azure corporativos	23
3.1.5. Requerimientos funcionales (Historias de usuario)	24
3.1.6. Requerimientos no funcionales	27
3.2. Diseño	28
3.2.1. Diseño de la solución	28
3.2.2. Lineamiento de arquitectura empresa TQ	32
3.2.3. Diagramas de diseño de la solución - Módulo Gestión de infraestructura de proyectos	36
3.2.4. Diagrama de diseño de la base de datos	40
3.3. Implementación	42
3.3.1. Stack tecnológico corporativo	42
3.3.2. Flujo del proceso en la aplicación	43

3.3.3. Funcionalidades de la aplicación	48
3.3.4. Estrategias para cumplir atributos de calidad	53
3.4. Pruebas funcionales	54
3.5. Resumen del capítulo	72
4. Evaluación	73
4.1. Diseño de la evaluación	73
4.1.1. Atributos de calidad	73
4.1.2. Diseño de la evaluación	74
4.1.3. Encuesta de Evaluación	76
4.1.4. Observaciones generales del instrumento	77
4.1.5. Limitaciones de la Evaluación	77
4.2. Resultados de la Evaluación	78
4.2.1. Respuesta a las preguntas de evaluación	78
4.3. Resumen del Capítulo	80
5. Conclusiones, lecciones aprendidas y trabajos futuros	81
5.1. Conclusiones	81
5.2. Lecciones Aprendidas	83
5.3. Trabajos Futuros	84
A. Aval de Tecnoquímicas para uso de información	85
B. Cuestionario de evaluación – Herramienta IAC - Tecnoquímicas	86
Bibliografía	87

Índice de figuras

2.1. Ejemplo plantilla de configuración Bicep	10
2.2. Matriz comparativa entre proyectos de IaC	15
3.1. Servicios Azure que Utiliza TQ actualmente (Elaboración propia según datos del área de infraestructura de TQ)	22
3.2. Ejemplo plantilla App Service	22
3.3. Estructura backlog	25
3.4. Ejemplo historia de usuario para adición de parámetros	26
3.5. Estructura Plantilla Bicep	31
3.6. Valor del parámetro con plantilla base para servicio App Service	32
3.7. Resultado plantilla Bicep – App Service	32
3.8. Organigrama Área TI - TQ (Elaboración propia)	33
3.9. Arquitectura Plataforma ERPTQ (Fuente de arquitectura TI de la organización TQ)	35
3.10. Diagrama de contexto	36
3.11. Diagrama de Contenedores	37
3.12. Diagrama de Componentes	38
3.13. Diagrama de la Base de Datos	40
3.14. Diseño general de la solución - Parte 1	44
3.15. Diseño general de la solución - Parte 2	45
3.16. Componente de seguridad - Autenticación y Autorización	46
3.17. Configuración - Parámetros generales	46
3.18. Estructura Rama	47
3.19. Servicios desplegados	47
3.20. Diagrama de Funcionalidades	48
3.21. Funcionalidad - Configuración de Parámetros	49
3.22. Funcionalidad - Configuración de Productos	50
3.23. Funcionalidad - Configuración de Servicios	51
3.24. Funcionalidad - Generador de Servicios	52
3.25. Resultado - Plantilla de Configuración Bicep	52
3.26. Configurador de parámetros - Caso de prueba 1	55
3.27. Configurador de parámetros - Caso de prueba 2	56
3.28. Configurador de parámetros - Caso de prueba 3	57
3.29. Configurador de parámetros - Caso de prueba 4	58
3.30. Configurador de parámetros - Caso de prueba 5	59
3.31. Configurador de productos - Caso de prueba 6	60
3.32. Configurador de productos - Caso de prueba 7	61
3.33. Configurador de productos - Caso de prueba 8	62

3.34. Configurador de productos - Caso de prueba 9	63
3.35. Configurador de productos - Caso de prueba 10	64
3.36. Configurador de servicios - Caso de prueba 11	65
3.37. Configurador de servicios - Caso de prueba 12	66
3.38. Configurador de servicios - Caso de prueba 13	67
3.39. Configurador de servicios - Caso de prueba 14	68
3.40. Configurador de servicios - Caso de prueba 15	69
3.41. Generador de servicios - Caso de prueba 16.1	70
3.42. Generador de servicios - Caso de prueba 16.2	71
A.1. Aval TQ - Uso de Información	85
B.1. Cuestionario Evaluación - Herramienta IaC	86

Introducción

La infraestructura tecnológica comprende el conjunto de elementos necesarios para operar y gestionar entornos de tecnologías de la información (TI) empresariales. Estos elementos incluyen el hardware, el software, los elementos de red, sistemas operativos y el almacenamiento de datos (Rodríguez Quevedo, 2022). Todos estos son utilizados para ofrecer servicios y soluciones de TI que en la actualidad pueden implementarse bajo un modelo OnPremise (Local) o Cloud mediante la virtualización de estos componentes.

En la última década, ha habido un incremento en el número de empresas que gestionan su infraestructura de forma virtualizada mediante plataformas en la nube. Estas plataformas ofrecen los servicios necesarios para operar, como el despliegue de aplicaciones web, bases de datos, almacenamiento y redes. Además, cuentan con una interfaz amigable para la administración y monitoreo de los servicios, y no requieren asumir costos de capital en infraestructura física (Patiño Vanegas, 2019).

Es importante considerar que la adopción del modelo cloud requiere que se realice la administración de la infraestructura virtualizada por parte de personal capacitado, el cual debe invertir tiempo significativo en la creación y parametrización de cada servicio de forma detallada en cada uno de los proyectos y ambientes asociados. Como lo indica IBM (2024), los sistemas requeridos por los clientes van incrementando en complejidad, lo que implica un mayor esfuerzo por parte del personal para crear y configurar la infraestructura requerida para la ejecución de los sistemas en tiempos cada vez menores. Al ser esto un proceso manual, existe un riesgo latente que se presenten errores u omisiones en la parametrización de los servicios, lo que lleva a consecuencias como altos costos de facturación, mal funcionamiento de los sistemas o riesgos de seguridad.

La infraestructura como código (IaC) es un enfoque que permite automatizar la gestión y despliegue de la infraestructura, incluyendo el hardware, los servicios virtuales, las plataformas, los sistemas de contenedores, los servicios y las topologías, mediante el uso de plantillas de configuración reutilizables que se integran con Devops para su ejecución por medio de pipelines. Esto trae beneficios como reducir tiempos en la parametrización y despliegue de infraestructura, disminuir carga de trabajo y errores por una mala parametrización manual (Devops, 2023).

La aplicación de la IaC como práctica que beneficia a las organizaciones puede ser favorecedor en las áreas de TI de empresas grandes. El presente trabajo se enmarca en el desarrollo de una

herramienta dentro de un grupo empresarial colombiano reconocido en la industria farmacéutica Tecnoquímicas (TQ). TQ, es una compañía farmacéutica con 90 años de antigüedad que actualmente cuenta con siete sedes productivas en Colombia y una en Centroamérica que fabrica alrededor de 4000 referencias; produce y comercializa productos de líneas de cuidado de bebé, cuidado personal y aseo del hogar, adhesivos industriales y comerciales, y productos agropecuarios y veterinarios. (Tecnoquímicas, 2024)

Al manejar una amplia gama de productos, la empresa diseña portales corporativos de las diferentes marcas que se encuentran en su portafolio, esto hace que para el área de TI la creación, actualización y manejo de estos portales web se conviertan en el día a día de un grupo de ingenieros. Actualmente, la empresa realiza el desarrollo web de manera manual, lo que puede incrementar la posibilidad de errores humanos en el despliegue de los servicios asociados a sus portales. A esto se suma, la dificultad del gobierno de la información de las copias de seguridad, lo que puede incidir de manera negativa en situaciones donde se deba mitigar o corregir errores de manera rápida.

Al identificar esta situación, el presente trabajo propone el desarrollo de una aplicación web reutilizable, para agilizar el proceso de creación y despliegue de servicios cloud en la plataforma Azure mediante el uso de IaC con Bicep para los portales corporativos.

1.1. Definición del problema de investigación

Morris (2020) explica que la Infraestructura como Código (IaC) es un acercamiento hacia la automatización de la infraestructura basado en prácticas de desarrollo de software, que enfatiza en la implementación de scripts o rutinas consistentes y repetibles que permitan aprovisionar cambios en los sistemas y su parametrización de forma ágil y segura. Entre los principales beneficios de la IaC están:

- Entregar valor rápidamente a los clientes.
- Reducir el esfuerzo y riesgo al hacer cambios en la infraestructura.
- Suministrar a los usuarios de infraestructura los servicios que necesita, cuando los necesita.
- Reutilizar las plantillas de configuración para la implementación de servicios en distintos ambientes
- Contar con sistemas que sean seguros, confiables y con un balance de costo beneficio.

Microsoft (2024) define a Azure como una plataforma de servicios en la nube de Microsoft que incluyen infraestructura como servicio (IaaS), plataforma como servicio (PaaS) y software como servicio (SaaS) para ayudar a las organizaciones a construir, implementar y administrar aplicaciones y servicios en la nube. Una de las funcionalidades con las que cuenta Azure Cloud es Bicep, la cual

mediante el uso de una sintaxis declarativa que se trata como código de aplicación en plantillas de configuración, permite la implementación y despliegue en la nube de servicios de forma consistente y repetible (Bicep, 2024).

Dado que Azure Cloud ofrece más de 200 servicios, según Microsoft (2024), y teniendo en cuenta la afirmación de Cois et al. (2014) que las aplicaciones informáticas en proyectos corporativos suelen ser sistemas distribuidos extensos, compuestos por diversos servicios con parametrizaciones y dependencias específicas, la instalación, parametrización y mantenimiento de la infraestructura demandan un considerable esfuerzo manual por parte del personal responsable, especialmente considerando que cada proyecto generalmente requiere al menos tres entornos (desarrollo, pruebas, producción)

La ejecución, parametrización y mantenimiento de la infraestructura es habitualmente un trabajo repetitivo y manual, por esta razón, está sujeto a problemáticas como la introducción de errores involuntarios y la inversión de tiempo significativo en tareas que, aunque importantes, eran repetitivas y podrían ser automatizadas.

Actualmente, TQ cuenta con varias páginas web corporativas creadas manualmente, cada una con copias de seguridad de sus servicios. Sin embargo, no se tiene un control adecuado sobre esta información, lo que impide agilizar el proceso de despliegue de cada portal cuando es necesario. Como empresa comercial que debe mantenerse a la vanguardia en su presencia en redes, TQ necesita soluciones automáticas que aceleren el despliegue de servicios asociados a sus portales en caso de eventualidades, asegurando una configuración totalmente confiable, sin riesgos de seguridad ni altos costos de uso.

La IaC se convierte en una opción para mitigar el riesgo del error humano, tener gobierno sobre la configuración de cada servicio en cada proyecto, y al integrarse con Devops permite automatizar el despliegue de la infraestructura. Actualmente, existen herramientas que buscan automatizar la IaC como Terraform (2024), Pulumi (2024), Puppet (2024). Sin embargo, estas herramientas requieren personal con competencias en programación para la creación y ajustes de los scripts de cada servicio cloud que se utiliza. (Dominguez-Quintero and Vargas-Lombardo, 2021)

Considerando los beneficios que ofrece la automatización y las posibilidades que tienen las plantillas de parametrización de Azure Bicep, este proyecto de grado está interesado en contestar la siguiente pregunta:

¿Cómo facilitar la selección y parametrización de servicios de Azure Cloud para generar plantillas de Bicep que sirvan para desplegar en la nube los servicios parametrizados?

1.2. Planteamiento del problema

Considerando los beneficios que la automatización puede traer a la gestión de la infraestructura cloud, se establecen las siguientes interrogantes:

- ¿Cuáles servicios de los que ofrece la plataforma Azure son requeridos por la empresa TQ para el aprovisionamiento de los portales corporativos?
- ¿Cuáles son las propiedades y set de valores de parametrización que se deben establecer para cada uno de los servicios identificados?
- ¿Cómo se pueden estructurar las plantillas Bicep para maximizar su reutilización?
- ¿Qué elementos de variabilidad deben ser considerados en el diseño de las plantillas?
- ¿Cómo debe ser la interfaz de usuario para facilitar la selección y parametrización de los servicios de Azure Cloud?

1.3. Objetivos

1.3.1. Objetivo General

Desarrollar una aplicación web que facilite la parametrización y generación de plantillas de configuración Bicep para el despliegue automático de los servicios cloud que incorporan los portales corporativos de la organización TQ.

1.3.2. Objetivos específicos

- Identificar los servicios de infraestructura cloud y las parametrizaciones más utilizadas en los portales de la organización.
- Diseñar la estructura de las plantillas Bicep para considerar la reutilización y la variabilidad que requiere la generación de plantillas de configuración parametrizadas.
- Diseñar e implementar una aplicación web para la selección y parametrización de servicios de Azure Cloud, así como la generación de plantillas de configuración en formato Bicep a partir de los parámetros seleccionados por el usuario.
- Evaluar la percepción de utilidad, la intención de uso y la facilidad de manejo de la solución web para generar y parametrizar las plantillas Bicep.

1.4. Justificación

IBM (2024) menciona que las empresas, cada vez más, acceden a los servicios de TI a través de la nube (cloud) bajo el modelo de pago por uso, lo que les permite disponer de un gran número de servicios informáticos, obteniendo gran capacidad de procesamiento y almacenamiento sin necesidad de instalar máquinas localmente. Esto se traduce en considerables ahorros de tiempo, costos e incluso de consumo energético. Además de estos aspectos, este modelo cloud de pago por uso ofrece ventajas adicionales como la seguridad, ya que los proveedores de servicios de nube suelen implementar medidas de protección avanzadas para salvaguardar los datos y sistemas de sus clientes. La escalabilidad es otra característica destacada, ya que las empresas pueden ajustar rápida y eficientemente sus servicios informáticos según las demandas cambiantes y la actualización tecnológica automática de forma periódica.

A pesar de que existen herramientas para automatizar el proceso de despliegue de servicios en la nube, como IaC, su adopción en las organizaciones es limitada debido a varios motivos, entre los que se destaca la falta de comprensión sobre su utilidad y los beneficios que ofrece. La adopción del modelo cloud y la administración de la infraestructura requiere contar con personal de TI encargado de la parametrización minuciosa y el despliegue de los servicios cloud necesarios. Estos procesos parten de una arquitectura predefinida para cada proyecto y ambiente. No obstante, es importante destacar que esta labor conlleva, en muchas ocasiones, una inversión significativa de tiempo para su implementación debido a su complejidad. Dado que este proceso se realiza manualmente, existe el riesgo inherente de errores humanos que pueden generar consecuencias como altos costos de facturación, servicios no disponibles y riesgos para la seguridad de los sistemas y los datos (Palos Sanchez, 2019).

Por lo anterior, autores como Michalowski (2024) reconocieron la importancia de adoptar la IaC como un mecanismo para automatizar la parametrización y despliegue de la infraestructura y aplicaciones en el modelo cloud. Esto se debía a que se reducía la intervención de los administradores, se agilizaban los tiempos de entrega y se minimizaban los riesgos de una mala parametrización

Es por esto que se propone la implementación de una aplicación web que, a través del uso de Azure Bicep, facilite la configuración y el despliegue automático de servicios y permita el reúso en diferentes proyectos que emplean infraestructura en la nube de Azure.

1.5. Alcance

El alcance de este trabajo de grado fue el siguiente:

- Identificar los servicios utilizados en los portales corporativos de la organización TQ al igual que las propiedades y valores comúnmente utilizados.
- Construir una aplicación siguiendo la arquitectura de referencia de TQ que sea reutilizable por cualquier proyecto que permita parametrizar los servicios Azure, y generar plantillas en formato Bicep para el despliegue de estos en plataforma Azure cloud.
- Evaluar la percepción de utilidad, la intención de uso y la facilidad para utilizar la solución web en la selección y parametrización de las plantillas Bicep por parte de las personas que interactúan con la solución que corresponde a personal de TI de la organización TQ.

Marco de referencia

2.1. Bases teóricas

2.1.1. Infraestructura como código

Kief (2020) menciona que el surgimiento de la arquitectura como código se encuentra relacionada, inicialmente, con acciones relacionadas con el aprovisionamiento de recursos informáticos, estos se realizaban de manera manual y rutinaria, lo que implicaba capital humano. El autor menciona que la aparición de la virtualización dentro del entorno de TI es una revolución, ya que permite gestionar los recursos informáticos a través de software y que, gracias a esto, no es necesario estar presente en un centro de datos.

Esto implica nuevas demandas para los arquitectos y el aprovisionamiento de la infraestructura, iniciando un proceso de liberación de la gestión de entornos virtualizados, lo que permite que los sistemas automaticen sus tareas de aprovisionamiento de recursos. La IaC agrupa los elementos de un centro de datos para gestionarlos mediante código definido a través de scripts, lo que permite automatizar tareas de aprovisionamiento y crear entornos seguros y escalables.

Morris (2020) define la infraestructura como código (IaC) como una práctica que permite automatizar la gestión y despliegue de la infraestructura cloud en las empresas mediante el uso de código, plantillas de configuración y pipelines que ejecutan y realizan el despliegue de los servicios y parametrización establecida.

Entre los beneficios que menciona **Morris (2020)** en su libro, están los siguientes:

- Uso de la infraestructura TI para una entrega rápida de valor al cliente.
- Reducción del esfuerzo y riesgos en cambios de la infraestructura y parametrización de forma manual.
- Creación de sistemas confiables, seguros y de un buen nivel de costo beneficio.

Algunos de los beneficios del uso de IaC que identifica **Carey (2023)** incluyen la reducción de riesgos de errores humanos, mayor agilidad en los despliegues, mayor control y supervisión de los cambios realizados, mejor seguridad, rápidos procesos de replicación de la infraestructura y reducción de costos.

En la actualidad se pueden encontrar herramientas que permiten la utilización de la IaC y que cuentan con clouds públicos como lo son Amazon Web Service (CloudFormation), Microsoft Azure (Azure Resource Manager) y Google Cloud Platform (Cloud Deployment Manager). Algunos de estos soportan lenguajes de programación como Typescript, Java, Golang, Python, C, etc.

Es importante establecer que al implementar la infraestructura como código se deben considerar tanto los beneficios como los retos de trabajar con un lenguaje de programación convencional. La aplicación de estas prácticas puede enfrentar ciertos desafíos, como el incremento de servidores, cambios de configuración para optimizar los servidores, y servidores insustituibles debido a que las aplicaciones heredadas funcionan solo con el servidor desplegado, lo que puede resultar en una infraestructura frágil. Además, el miedo a la automatización presente en algunas personas puede afectar la implementación de esta estrategia (Turim, 2023).

2.1.2. Azure Bicep

Bicep es un lenguaje desarrollado por Microsoft, denominado DSL (lenguaje específico de dominio) es decir, que su uso es específicamente para el propósito de definir los servicios y la parametrización requerida de cada uno de los servicios que se requieren aprovisionar. Es de tipo declarativo, es decir, que se establece el resultado esperado sin importar el orden de ejecución (Bicep, 2024).

El uso de Bicep como herramienta de automatización de infraestructura cloud ofrece ventajas como:

- Compatibilidad con todos los tipos de servicios de Azure y las distintas versiones.
- Sintaxis compacta y fácil de entender.
- Detección automática de errores en la definición de las plantillas de configuración mediante el uso de extensiones.
- Resultados repetibles, los archivos de Bicep son idempotentes, lo que significa que pueden implementar el mismo servicio las veces que se requiera.
- La orquestación de los servicios y su orden lo realiza de forma automática el *Resource Manager* de Azure.
- Se pueden definir módulos en donde en cada módulo se puede configurar los servicios relacionados.
- Bicep se integra de forma natural con los servicios de Azure mediante Azure policy.
- Bicep es gratuito y cuenta con documentación técnica y funcional.

La Figura 2.1 muestra un ejemplo de una plantilla de configuración en formato Bicep para la creación de un servidor y una BD SQL server, el cual se explica paso a paso a continuación:

- Las líneas 1-7 definen los parámetros y valores por defecto asignados a las propiedades de los recursos a generar posteriormente, entre los que están el nombre del servidor, el nombre de la BD, la locación del grupo de recursos donde se va a desplegar en Azure. Es de aclarar que al ser parámetros de entrada, al invocar esta platilla desde otra, se pueden enviar los valores de dichos parámetros, en caso de no enviarlos se toman los valores por defecto definidos.
- Las líneas 10-17 establecen la estructura para crear un servidor de base de datos SQL Server, definiendo propiedades obligatorias como el nombre, la ubicación, el usuario y la contraseña del administrador. Este es el tipo de servidor que se creará para el ejemplo.
- Las líneas 20-27 definen la estructura para la creación de una base de datos SQL Server, especificando el nombre, que se asigna como servidor/nombreBD, la ubicación y el tipo de recurso o SKU requerido.

```
1 param serverName string = uniqueString('sql', resourceGroup().id)
2 param sqlDBName string = 'SampleDB'
3 param location string = resourceGroup().location
4 param administratorLogin string
5
6 @secure()
7 param administratorLoginPassword string
8
9
10 resource server 'Microsoft.Sql/servers@2019-06-01-preview' = {
11   name: serverName
12   location: location
13   properties: {
14     administratorLogin: administratorLogin
15     administratorLoginPassword: administratorLoginPassword
16   }
17 }
18
19
20 resource sqlDB 'Microsoft.Sql/servers/databases@2020-08-01-preview' = {
21   name: '${server.name}/${sqlDBName}'
22   location: location
23   sku: {
24     name: 'Standard'
25     tier: 'Standard'
26   }
27 }
```

Figura 2.1: Ejemplo plantilla de configuración Bicep

2.1.3. Contexto

La empresa TQ actualmente cuenta con diversas soluciones tipo portales que implementan los mismos servicios e incluso las mismas parametrizaciones. La administración manual de la infraestructura de cada uno de estos portales toma mucho tiempo y es propensa a errores humanos, lo que puede conducir a configuraciones inconsistentes entre entornos de desarrollo, pruebas y producción, causar errores inesperados y problemas de rendimiento.

En Tecnoquímicas, se cuenta con más de 1100 servicios de Azure, siendo el App Service el más aprovisionado. A continuación, se explicará cómo los portales de la corporación utilizan los siguientes servicios con parametrizaciones comunes.

- **AppServicePlan (2023)**: servicio que define un conjunto de recursos de proceso para que una aplicación web se ejecute.
- **AppService (2023)**: servicio para hospedar aplicaciones web, API REST, y back-ends para dispositivos móviles. Soporta distintas tecnologías de desarrollo como NET, .NET Core, Java, Node.js, PHP y Python.
- **SQLServer (2024)**: servicio que se encarga de la administración de bases de datos relacionales como SQL Database.
- **SQLDatabase (2024)**: servicio que brinda un motor de base de datos de plataforma como servicio (PaaS) totalmente administrado que se encarga de la mayoría de las funciones de administración de bases de datos, como las de actualizar, aplicar revisiones, crear copias de seguridad y supervisar sin intervención del usuario.

2.1.4. Necesidad

Una preocupación de la empresa es que a pesar de que existan copias de seguridad de los servicios por cada portal, no hay un proceso automatizado y ágil que permita desplegar rápidamente la infraestructura de cada portal en caso de ser requerido. La Infraestructura como código (IaC) lograría mitigar el riesgo y se podría acoplar al proceso de CI/CD, brindando mayor gobierno y automatización de la infraestructura.

Considerando la situación de Tecnoquímicas, la implementación de Infraestructura como Código (IaC) para los portales corporativos podría permitir estandarizar, automatizar y gobernar eficazmente la fase de aprovisionamiento de recursos en Azure.

2.2. Antecedentes

A continuación, se presentan diferentes investigaciones relacionadas con el tema del presente proyecto (Gestión de Infraestructura cloud con IaC):

- [Koskelainen \(2021\)](#) creó en su proyecto, los pipelines requeridos para el despliegue de una aplicación web de ejemplo desarrollada en React. Se estableció un pipeline para la creación de la IaC usando la herramienta Pulumi la cual permite codificar las plantillas de configuración en diversos lenguajes de programación como Python, Node JS, Java y los servicios requeridos en Azure para la implementación fueron 2 app server, 1 key vault, 1 servidor de BD y una base de datos SQL Server.

El ejercicio se limitó a 3 servicios, los cuales ofrece la plataforma de Azure (App Service, Key Vault, SQL Server) con una parametrización preestablecida, es decir, no disponible para seleccionar o ajustar por parte de un usuario. En la implementación de este proyecto se concluye que la herramienta utilizada no solo requiere que la persona que gestione la parametrización de los servicios conozca del lenguaje de programación para entender y aplicar cambios sobre los templates de ser requerido, sino que la herramienta si es de uso corporativo, tiene costo.

- El alcance del proyecto de [Moreno Martínez \(2022\)](#) es crear un pipeline con Jenkins y usar Terraform como herramienta de IaC; este pipeline debe ejecutar diversas etapas como son la revisión de calidad de código, pruebas unitarias y despliegue de los servicios en AWS con la parametrización establecida en los archivos de Terraform.

Este proyecto busca estudiar cuáles son las buenas prácticas de desarrollo en IaC y aplicarlo en un ejemplo práctico sobre un pipeline. Es decir, no busca crear una aplicación orientada al reuso y que permita seleccionar los servicios y parametrización de cada uno por parte del usuario

- [Camas Mainato and Riera Taza \(2022\)](#) llevó a cabo la implementación de plantillas de configuración destinadas a una infraestructura ya establecida y configurada inicialmente de manera genérica. Este proceso se ejecutó utilizando la herramienta Terraform y teniendo como destino una nube privada denominada OpenStack. Es importante destacar que las plantillas de configuración de su proyecto fueron construidas de manera manual con una parametrización preestablecida.
- [Ronaldo \(2021\)](#), llevó a cabo la evaluación del alcance e identificó las limitaciones de la herramienta Terraform. Como parte fundamental de esta evaluación, se procedió a la construcción de plantillas de configuración específicas para Terraform, abarcando las plataformas líderes en la nube: Google Cloud Platform, Amazon Web Services y Microsoft Azure. El análisis comparativo posterior se centró en la experiencia de implementación en cada una de estas plataformas, destacando aspectos clave como la eficiencia, la integración con servicios específicos de cada proveedor y la facilidad de mantenimiento. Este enfoque holístico proporciona una visión detallada de la idoneidad de Terraform en diferentes entornos cloud.

- **M. Guerriero and Palomba (2019)** en su artículo, analiza la adopción, el soporte y los desafíos de la infraestructura como código (IaC) desde la perspectiva de la industria. El documento aborda cómo la metodología DevOps está transformando radicalmente el diseño y la gestión del software en la actualidad. La IaC se presenta como la forma de desarrollar código fuente basado en la nube. El objetivo del estudio era identificar las prácticas actuales de desarrollo de infraestructura como código, las herramientas utilizadas y los desafíos enfrentados en este proceso por parte del área de TI de las organizaciones.

Los investigadores realizaron 44 entrevistas semiestructuradas donde se evidencia que es necesario continuar con investigaciones en el campo, puesto que los apoyos que se encuentran son todavía limitados y los desarrolladores sienten la necesidad de técnicas novedosas para testear y mantener el código de la IaC.

- **Hasan MM (2020)** menciona que se puede encontrar que, a pesar de los beneficios nombrados ampliamente de la aplicación de la IaC, también se encuentran ciertos aspectos que pueden ser un desafío, como por ejemplo los scripts de IaC pueden ser susceptibles a defectos. Estos defectos pueden tener consecuencias a gran escala, por lo que es necesario implementar prácticas de testeo de IaC con el fin de mitigar posibles errores. En el estudio se identifican algunas de las prácticas de testeo para la IaC que son: evitar, codificar antipatrones, cobertura de pruebas centradas en el comportamiento, testeo remoto, pruebas de espacio aislado y prueba de cada cambio de la IaC.
- **Patni JC (2020)** plantea que usando templates de Azure Resource Manager (ARM) evidencian los beneficios y la importancia de la IaC en el desarrollo de una aplicación web en Microsoft Azure usando solo código. Con el ejercicio, logran determinar que el uso de templates incrementó la agilidad del proceso de implementación y gestión, las plantillas de ARM proporciona la reutilización de código para ser ejecutado pudiendo utilizar la misma plantilla para implementar varias veces una variedad de infraestructura de requisitos de una organización, se logran definir dependencias para asegurar el orden correcto para implementar las plantillas y usando templates de ARM se puede proporcionar un solo archivo de JSON lo que automatiza el proceso y crea los requisitos necesarios con las configuraciones determinadas.

En la Figura 2.2 se presenta una matriz comparativa con cinco de los proyectos que tienen algunos objetivos comunes con el proyecto actual, como lo son, el uso de plantillas de configuración para el despliegue automático de servicios y la evaluación de los resultados obtenidos en su implementación. Los proyectos no relacionados en la matriz no tienen el mismo alcance de este proyecto y no realizan alguna implementación. La evaluación comparativa se centró en las siguientes categorías que el presente trabajo busca implementar:

- Aplicación web orientada al usuario: se evalúa si en cada trabajo se construyó una aplicación que facilite al usuario la selección de los servicios y parametrización.
- Usuario con conocimiento básico puede usar la aplicación: en caso de que el trabajo implemente alguna aplicación, este criterio evalúa si la aplicación puede ser utilizada por usuarios sin conocimientos técnicos.
- Las plantillas de configuración no son manipuladas manualmente: verifica si las plantillas que se construyen en estos trabajos deben ser manipuladas por parte del usuario manualmente.
- Generación de plantillas a partir de los servicios y configuración seleccionada por el usuario: válida si en estos trabajos se generan plantillas de configuración de forma automática.
- Herramienta reutilizable: evalúa si la herramienta y plantillas generadas pueden reutilizarse para desplegar servicios en cualquier proyecto corporativo.
- Versionamiento de cambios en plantillas de configuración: analiza si las plantillas utilizadas en los proyectos se gestionan y actualizan automáticamente en un sistema de control de versiones.
- Despliegue automático de los servicios seleccionados: válida si a partir de las plantillas de configuración construidas se puede hacer el despliegue automático de servicios cloud.
- Evaluación de reducción de tiempos en la creación y despliegue de infraestructura: verifica si se realiza un caso de prueba real en una empresa y se evalúa el impacto que se tiene en el proceso en reducción de tiempos y errores.

De acuerdo al análisis realizado al estado del arte y como se muestra en la Figura 2.2, los cinco trabajos crean plantillas de configuración con diversas herramientas que permiten el despliegue de servicios cloud, sin embargo, ninguno de los proyectos implementa una aplicación web que evite la creación y manipulación de plantillas de forma manual por lo que utilizar las propuestas ya existentes, requiere tiempo, conocimiento técnico y está sujeto a errores. Las plantillas generadas se pueden ejecutar por la línea de comandos en cualquier proyecto, pero sin variabilidad en los servicios y parametrización requerida.

Es decir, en ninguno de los proyectos evaluados se busca implementar una aplicación reutilizable que permita ser usada en cualquier proyecto para generar las plantillas acordes a los servicios y parametrización requerida, tampoco se evalúan los resultados de su implementación.

Categoría	Trabajo				
	Diseño e implementación de un proveedor de infraestructura en la nube IaaS, mediante segmentación de recursos (VX Camas Mainato, HX Riera Taza)	Creating Azure devops pipelines for web application (Koskelainen, Eetu)	CI/CD en la infraestructura como código (IaC). Caso práctico en Amazon Web Services (Moreno Martínez, Juan)	Terraform como herramienta para automatizar la creación de infraestructura siguiendo el concepto "IaC" (Huerio Quintero, Ronaldo)	Infrastructure as a Code (IaC) to Software Defined Infrastructure using Azure Resource Manager (Patni, Jagdish)
Aplicación web orientada al usuario	●	●	●	●	●
Usuario con conocimiento básico puede usar la aplicación	●	●	●	●	●
Las plantillas de configuración son manipuladas manualmente	●	●	●	●	●
Generación de plantillas de configuración a partir de servicios y configuración seleccionada por el usuario	●	●	●	●	●
Herramienta reutilizable	●	●	●	●	●
Versionamiento de cambios en plantillas de configuración	●	●	●	●	●
Despliegue automático de los servicios seleccionados	●	●	●	●	●
Evaluación de reducción de tiempos en la creación y despliegue de infraestructura	●	●	●	●	●

● Cumple total
 ● Cumple parcial
 ● No cumple

Figura 2.2: Matriz comparativa entre proyectos de IaC

Por estas razones es importante considerar la construcción de una aplicación que esté orientada al reúso a nivel corporativo, que brinde un conjunto de servicios y de configuraciones acorde a las necesidades de cada proyecto.

2.3. Resumen del capítulo

En este capítulo se describieron los siguientes aspectos:

- Las bases teóricas necesarias para la comprensión de conceptos como la IaC y Azure Bicep.
- El contexto de la situación actual en la empresa TQ y la descripción de la necesidad que motivó este proyecto de grado.
- Los antecedentes de proyectos similares en los cuales se implementó la IaC y una matriz comparativa en la que se identifica que factores no incluyen esas otras soluciones.

Análisis, Diseño e Implementación

En este capítulo se brinda información detallada sobre las fases de análisis, diseño e implementación de este proyecto. Comenzamos con un análisis de los portales corporativos de la organización y los servicios de Azure utilizados, incluyendo su parametrización y los requerimientos tanto funcionales como no funcionales. También se detalla el proceso de levantamiento de historias de usuario. En la sección de diseño, se discute el diseño de la solución, identificando a los stakeholders, las decisiones de diseño, la parametrización general, los momentos de resolución y los lineamientos de arquitectura, acompañados de diagramas detallados. Finalmente, en la sección de implementación, se describe el stack tecnológico utilizado y el flujo de proceso de la aplicación, destacando las funcionalidades clave implementadas. Este capítulo está diseñado para proporcionar una comprensión integral y detallada de cada aspecto, asegurando una solución robusta y bien fundamentada.

3.1. Análisis

3.1.1. Proceso de elicitación

Al iniciar el proyecto, fue necesario llevar a cabo una serie de actividades relacionadas con la recopilación de requisitos según las necesidades de la organización. Para ello, se realizaron varias iteraciones con los interesados, con el fin de definir el alcance, los requisitos generales y asegurar la pertinencia de la solución a implementar

Dado que la organización no tenía experiencia previa con este tipo de desarrollo, se seleccionaron unos portales y servicios para realizar una prueba de concepto. Esto permitió evaluar la viabilidad de implementar IaC en las soluciones cloud de Azure de la empresa. El proceso seguido fue el siguiente:

- a) Reunión de contexto con el líder de la subárea de TI que gestiona el ciclo de vida de las aplicaciones (ALM). Aquí se explicó cuál era necesidad, que consistía en la implementación de infraestructura como código para el gobierno, versionamiento y despliegue automático de servicios de infraestructura Azure.
- b) Reunión con propuesta técnica en la cual se llevó un diagrama de contexto que explicaba el alcance y como operaría la solución, cuáles serían las funcionalidades, con que otros sistemas se integraría y bajo qué mecanismos. Se recibe una retroalimentación por parte del líder de ALM y se hacen algunos ajustes al diagrama de la solución.

- c) Socialización formal a la líder de infraestructura, en donde se le explicaba la necesidad, el alcance del proyecto y el diagrama de diseño de la solución. Se recibe una retroalimentación por parte del líder antes de presentar el proyecto a la dirección desde TI.
- d) Reunión de asesoría con la persona encargada de administrar la infraestructura cloud en la empresa, la cual suministró información detallada de los sistemas que la empresa tiene desplegados en la nube Azure y los servicios más comúnmente usados.
- e) Socialización del proyecto a la dirección en el comité de arquitectura, indicando el alcance, el resultado esperado, el tiempo estimado y los recursos necesarios. La dirección da el visto bueno para iniciar fase de análisis y presentar cronograma.
- f) Redacción de historias de usuario tomando como insumo el diseño inicial y comentarios de las partes interesadas en las reuniones realizadas, se solicita al usuario final la revisión y aprobación de las mismas.
- g) Reunión con líder de ALM donde se establece que el alcance del proyecto será la generación de plantillas de configuración Bicep y el despliegue de los servicios asociados a las soluciones tipo portales de la organización, para esto se solicita a la persona de infraestructura que actualmente ha aprovisionado dichos recursos, suministre un Excel con el detalle de los servicios, características y valores que aplica para cada portal.

3.1.2. Portales corporativos

En la fase de análisis del proyecto, se recolectó información sobre los portales corporativos que la organización tiene en Azure Cloud, así como un listado de los servicios más utilizados y sus atributos. Se determinó que, para el año 2024, la organización TQ cuenta con aproximadamente 10 portales corporativos desplegados en la nube de Azure Cloud, los cuales son aprovisionados por el personal de infraestructura. Estos portales corresponden a los productos de la organización que buscan posicionar la marca y mejorar el relacionamiento con el consumidor.

- Bonfiest: el sitio web de Bonfiest Plus está dedicado a proporcionar información sobre su producto, un analgésico y antiácido efervescente. En el sitio, puede encontrar:
 - Descripción del producto: Información detallada sobre los ingredientes activos y sus beneficios.
 - Síntomas: Guía sobre los síntomas que el producto puede aliviar.
 - Preguntas frecuentes: Respuestas a las dudas comunes sobre el uso del producto.
 - Opciones de compra: Enlaces y detalles sobre dónde adquirir Bonfiest Plus.
- Content: el sitio web de Content, parte de Tecnoquímicas, está enfocado en el cuidado del adulto mayor. En el sitio, puede encontrar:

- Productos: Información sobre pañales y otros productos diseñados para la incontinencia y el cuidado de la piel.
 - Consejos y recursos: Guías y artículos para cuidadores sobre cómo manejar la incontinencia y mantener el bienestar integral del adulto mayor.
 - Consultas médicas: Dudas relacionadas con el cuidado del adulto mayor.
 - Comunidad: Oportunidades para registrar y recibir apoyo continuo de la comunidad Content.
- CureBand: el sitio web de Cure Band está dedicado a productos para el cuidado de heridas. En el sitio, puede encontrar:
 - Productos: Información sobre diferentes tipos de curas y apósitos, como las curas premium redondas.
 - Guías de uso: Instrucciones detalladas sobre cómo aplicar los productos correctamente.
 - Consejos de salud: Artículos y recursos sobre el cuidado de heridas y la prevención de infecciones.
 - Opciones de compra: Detalles sobre dónde y cómo adquirir los productos Cure Band.
 - Duraflex: el sitio web de Duraflex Muscular Advance está enfocado en proporcionar información sobre su producto para el alivio del dolor muscular. En el sitio, puede encontrar:
 - Descripción del producto: Información detallada sobre la fórmula y los componentes del producto.
 - Tipos de dolor: Guía sobre los diferentes tipos de dolor muscular que el producto puede aliviar.
 - Dónde comprar: Información sobre los puntos de venta y opciones de compra.
 - Contacto: Formulario y detalles para contactar con el equipo de Duraflex.
 - Gastrofast: El sitio web de Gastrofast está dedicado a proporcionar información sobre su producto para el alivio inmediato del reflujo y la acidez. En el sitio, puede encontrar:
 - Descripción del producto: Detalles sobre cómo funciona Gastrofast y sus ingredientes activos.
 - Síntomas y tratamientos: Información sobre los síntomas que el producto puede aliviar, como el reflujo, la acidez, la gastritis y los gases intestinales.
 - Preguntas frecuentes: Respuestas a las dudas comunes sobre el uso del producto.
 - Opciones de compra: Detalles sobre dónde y cómo adquirir Gastrofast.
 - Ibuflash: El sitio web de Ibuflash está dedicado a productos para el alivio rápido del dolor de cabeza, incluyendo migrañas. En el sitio, puede encontrar:

- Productos: Información sobre las diferentes presentaciones de Ibuflash, como Ibuflash Migran, que combina ibuprofeno, acetaminofén y cafeína para un alivio rápido.
 - Guías de uso: Instrucciones detalladas sobre cómo utilizar los productos correctamente.
 - Consejos de salud: Artículos y recursos sobre el manejo del dolor de cabeza y la prevención de migrañas.
 - Opciones de compra: Detalles sobre dónde y cómo adquirir los productos Ibuflash
- Noraver: El sitio web de Noraver Grippa Fast Total proporciona información sobre este medicamento de venta libre, diseñado para aliviar los síntomas de la gripe. En el sitio, puede encontrar:
 - Descripción del producto: Detalles sobre los ingredientes activos, que incluyen ibuprofeno, fenilefrina clorhidrato, levocetirizina diclorhidrato y dextrometorfano bromhidrato.
 - Indicaciones: Información sobre los síntomas que el producto puede aliviar, como fiebre, congestión nasal, estornudos y tos.
 - Opciones de compra: Enlaces y detalles sobre dónde adquirir Noraver Grippa Fast Total.
- Sal de frutas LUA: El sitio web de Sal de Frutas LUA está dedicado a productos para el alivio de la indigestión, pesadez y llenura. En el sitio, puede encontrar:
 - Productos: Información sobre las diferentes presentaciones, como Sal de Frutas LUA y Sal de Frutas LUA Plus, que también alivia la distensión abdominal y los gases.
 - Guías de uso: Instrucciones detalladas sobre cómo utilizar los productos correctamente.
 - Consejos de salud: Artículos y recursos sobre el manejo de la indigestión y otros males-tares digestivos.
 - Opciones de compra: Detalles sobre dónde y cómo adquirir los productos Sal de Frutas LUA.
- VitaC: El sitio web de Vitac+Zinc MK Mandarina 1gr EFE proporciona información sobre este suplemento vitamínico efervescente. En el sitio, puede encontrar:
 - Descripción del producto: Detalles sobre los ingredientes activos, que incluyen ácido ascórbico (vitamina C) y zinc.
 - Beneficios: Información sobre cómo este suplemento puede ayudar a fortalecer el sistema inmunológico y mejorar la salud general.
 - Opciones de compra: Enlaces y detalles sobre dónde adquirir Vitac+Zinc MK Mandarina.
- Yodora: El sitio web de Yodora está dedicado a productos de higiene personal, especialmente desodorantes. En el sitio, puede encontrar:

- Productos: Información sobre las diferentes presentaciones de desodorantes, como Yodora Clásico, Yodora Derma Control para mujeres, Yodora Total Control para hombres, y productos para el cuidado de los pies.
- Guías de uso: Instrucciones detalladas sobre cómo aplicar los productos correctamente.
- Consejos de salud: Artículos y recursos sobre el cuidado de la piel y la prevención del mal olor.
- Opciones de compra: Detalles sobre dónde y cómo adquirir los productos Yodora.

3.1.3. Servicios Azure corporativos

Para el aprovisionamiento de los portales descritos previamente, se cuenta con aproximadamente 1118 servicios de Azure. Entre los servicios más utilizados están el App Service, y SQL Database como se muestra en la Tabla 3.1.

Es importante considerar que el servicio App Service depende del App Service Plan para obtener los recursos necesarios (CPU, memoria, almacenamiento) y para definir las características de escalabilidad y disponibilidad. Asimismo, el servicio SQL Database depende de SQL Server para la gestión y operación de las bases de datos. SQL Server proporciona la infraestructura y las capacidades necesarias para que SQL Database funcione de manera eficiente y segura. Por esta razón, es requerida la implementación del App Service Plan y SQL Server adicionalmente como servicios base.

Es de anotar que el personal de infraestructura realiza copias de seguridad de la configuración de cada servicio en plantillas ARM (archivos en formato JSON (JavaScript Object Notation) que permiten definir, implementar y gestionar la infraestructura de Azure como código, utilizan una sintaxis declarativa, lo que significa que se puede especificar los recursos que se requieren implementar y sus configuraciones), pero no se tiene un gobierno centralizado de esta información y no sirven en la actualidad para agilizar el proceso de despliegue de cada portal en caso de ser requerido.

En la Figura 3.2 se muestra un ejemplo de una plantilla de configuración para un App service indicando las partes descritas en la Tabla 3.1

Servicio Azure	Descripción	Uso	Cuando se usa	Total tipo
App Service	Servicio para alojar aplicaciones web, API y backends móviles.	Proporciona un entorno escalable y gestionado para aplicaciones web.	Quando se requiere desplegar y gestionar aplicaciones web con alta disponibilidad y escalabilidad.	346
SQL Database	Servicio de base de datos relacional en la nube basado en SQL Server.	Ofrece almacenamiento y gestión de datos relacionales con alta disponibilidad y seguridad.	Quando se requiere una base de datos relacional escalable y gestionada en la nube.	125
App Service Plan	Define los recursos (CPU, memoria, almacenamiento) y la capacidad de escalado para las aplicaciones de App Service.	Proporciona los recursos necesarios para ejecutar aplicaciones en App Service.	Quando se requiere configurar los recursos y la escalabilidad para tus aplicaciones web.	109
Storage Account*	Servicio de almacenamiento en la nube para datos no estructurados.	Almacena grandes cantidades de datos no estructurados, como blobs, archivos y colas.	Quando se requiere almacenar y acceder a datos no estructurados de manera escalable y segura.	102
Disk	Almacenamiento persistente para máquinas virtuales.	Proporciona almacenamiento de alto rendimiento y durabilidad para VMs.	Quando se requiere almacenamiento persistente y de alto rendimiento para tus máquinas virtuales.	89
Smart Detector Alert Rule	Herramienta de monitoreo que detecta anomalías en los recursos de Azure.	Proporciona alertas inteligentes basadas en patrones de uso y rendimiento.	Quando se requiere monitorear y recibir alertas sobre el rendimiento y la salud de tus recursos de Azure.	56
SQL Server	Motor de base de datos que gestiona SQL Database.	Proporciona capacidades de procesamiento de transacciones y gestión de datos.	Quando se requiere una infraestructura robusta para gestionar bases de datos SQL.	57
Container Registry Webhook	Notificaciones automáticas para eventos en Azure Container Registry.	Permite la automatización de flujos de trabajo basados en eventos de contenedores.	Quando se requiere desencadenar acciones automáticas en respuesta a eventos en tu registro de contenedores.	51
App Service (Slot)	Entornos de despliegue adicionales para aplicaciones en App Service.	Permite pruebas y despliegues sin afectar la producción.	Quando se requiere realizar pruebas A/B o despliegues en etapas sin interrumpir el entorno de producción.	35
Container Registry	Servicio para almacenar y gestionar imágenes de contenedores.	Proporciona un registro privado para almacenar y gestionar imágenes de contenedores Docker.	Quando se requiere un repositorio seguro y gestionado para tus imágenes de contenedores.	32
Application Insights	Servicio de monitoreo para aplicaciones web.	Proporciona telemetría y análisis de rendimiento para aplicaciones.	Quando se requiere monitorear el rendimiento y el uso de tus aplicaciones web en tiempo real.	27
Network Security Group	Control de acceso a la red para recursos de Azure.	Permite definir reglas de seguridad para controlar el tráfico de red.	Quando se requiere proteger tus recursos de Azure mediante reglas de seguridad de red.	23
SQL Server Database - Azure Arc	Extiende la gestión de SQL Server a entornos híbridos y multicloud.	Proporciona capacidades de gestión unificadas para bases de datos SQL en cualquier entorno.	SQL en entornos híbridos o multicloud con Azure Arc.	22
Public IP Address	Dirección IP pública para recursos de Azure.	Permite la comunicación externa con recursos de Azure.	Quando se requiere que tus recursos de Azure sean accesibles desde Internet.	22
Machine - Azure Arc	Extiende la gestión de máquinas virtuales a entornos híbridos y multicloud.	Proporciona capacidades de gestión unificadas para máquinas virtuales en cualquier entorno.	Quando se requiere gestionar máquinas virtuales en entornos híbridos o multicloud con Azure Arc.	20
Total Servicios				1116

Figura 3.1: Servicios Azure que Utiliza TQ actualmente (Elaboración propia según datos del área de infraestructura de TQ)



```
// Definir el App Service Plan
resource appServicePlan 'Microsoft.Web/serverfarms@2020-06-01' = {
  name: '${webAppName}-plan'  Servicio
  location: location
  sku: {
    name: skuName  Sección propiedades-valores
    capacity: 1
  }
  properties: {
    reserved: true // Indica que es un plan para Linux
  }
}
```

Figura 3.2: Ejemplo plantilla App Service

3.1.4. Parametrización servicios Azure corporativos

Para aprovisionar cada servicio se requiere definir unas propiedades entre las cuales existen unas obligatorias, otras opcionales que se agrupan por secciones y subsecciones de acuerdo a los lineamientos de Azure, estas propiedades se ven reflejadas en las plantillas de configuración de Bicep.

A continuación se detalla en la Tabla 3.2 los servicios seleccionados para este proyecto, sus atributos, obligatoriedad y como se agrupan a nivel de sección y subsección en las plantillas de configuración de Bicep y los valores posibles para cada caso.

Servicio Azure	Propiedad	Sección	Descripción	Tipo	Obligatorio	Valores
App Service Plan	name		Nombre del plan app service	string	X	(Valor libre)
	serverFarmId		Región de la aplicación	string	X	East US - Central US
	reserved	properties	Linux app service plan	string		true - false
	name	sku	Tipo de maquina	string	X	Free F1 - Basic B1
	kind		Sistema Operativo	string	X	Linux - Windows
	tags		Etiquetas nombramiento	object	X	{'Ambiente': '', 'Negocio': '', 'Analista': '', 'solución': '', 'tribu': ''}
App Service	name		Nombre de la aplicación	string	X	(Valor libre)
	location		Región de la aplicación	string	X	East US - Central US
	serverFarmId	properties	id app service plan	string	X	Selección app service plan existente
	linuxFxVersion	siteConfig	Versión de SO Linux	string	X	node 14-lts
	tags		Etiquetas nombramiento	object	X	{'Ambiente': '', 'Negocio': '', 'Analista': '', 'solución': '', 'tribu': ''}
SQL Server	name		Nombre del servidor	string	X	(Valor libre)
	location		Región del servidor	string	X	East US - Central US
	administratorLogin	properties	Usuario administrador BD	string	X	Selección app service plan existente
	administratorLogin	properties	Password administrador BD	string	X	node 14-lts
	tags		Etiquetas nombramiento	object	X	{'Ambiente': '', 'Negocio': '', 'Analista': '', 'solución': '', 'tribu': ''}
SQL Database	name		Nombre de la BD	string	X	(Valor libre)
	location		Región de la BD	string	X	East US - Central US
	name	sku	Nombre Tipo de maquina	string	X	(Valor libre)
	tier	sku	Tipo de maquina	string	X	Basic, Standard, Premium
	tags		Etiquetas nombramiento	object	X	{'Ambiente': '', 'Negocio': '', 'Analista': '', 'solución': '', 'tribu': ''}

Tabla 3.2 - Propiedades Servicios Azure

3.1.5. Requerimientos funcionales (Historias de usuario)

Los artefactos establecidos para el levantamiento de requisitos del proyecto fueron los siguientes:

- Entrevistas: se realizan varias reuniones con los usuarios para recopilar información detallada sobre cómo llevan a cabo los procesos actualmente de aprovisionamiento. Luego, se les presenta una propuesta de solución junto con los mockups de las funcionalidades, explicando qué acciones realizaría en cada opción. Durante estas reuniones, se obtiene retroalimentación de los usuarios sobre sus expectativas y necesidades. Adicionalmente, se realizan reuniones con el equipo de arquitectura para acordar los lineamientos de desarrollo y determinar cuáles son los atributos no funcionales prioritarios para este proyecto. Toda esta información se consolida y se envía por correo a todos los participantes, y es el insumo para generar las historias de usuario.
- Historias de usuario: se crean historias de usuario padre para cada una de las funcionales que integran la solución y las hijas con el detalle de las acciones. En estas se utiliza el método INVEST en el cual se asegura que las historias de usuario sean claras, flexibles, testeables y que entreguen valor al usuario final.
- Prototipos: se crean los mockups y wireframes en la herramienta FIGMA para mostrar a los usuarios en las reuniones, representando cómo sería la interfaz y el flujo del proceso. Estos también se adjuntan a las historias de usuario para que formen parte del alcance y quede registrado que el usuario aprueba tanto las historias como los mockups.

A partir de la información recolectada en fase de análisis en el proceso de elicitación mencionado en el punto 3.1.1, se evaluó y estableció, junto al líder del proyecto, el backlog del mismo. Este backlog se compone del objetivo estratégico de la solución a implementar (Épica), el sistema que soportará la necesidad (Feature), las funcionalidades que contendrá el sistema (Historias de usuario padre) y las acciones o tareas a realizarse en cada una de las funcionalidades (Historias de usuario hijas).

La Figura 3.3 presenta la estructura del backlog acorde a los lineamientos del área de TI de la empresa, detallando las historias de usuario de este proyecto:

- Épica: proceso macro al que corresponde el proyecto
- Feature: funcionalidad asociada al proceso
- Historia de usuario padre: funcionalidad del sistema
- Historia de usuario hija: acciones asociadas a cada funcionalidad

- ▼ 🏰 Infraestructura como código IaC
 - ▼ 🏰 Infraestructura como código
 - ▼ 📄 Inicio
 - > 📄 Menú Principal
 - ▼ 📄 Configuración de Parámetros
 - > 📄 Adición de Parámetros
 - > 📄 Modificación de Parámetros
 - > 📄 Consulta de Parámetros
 - > 📄 Activación/Inactivación de Parámetros
 - ▼ 📄 Configuración de Productos
 - > 📄 Adición de registros
 - > 📄 Consulta de registros
 - > 📄 Activación/Inactivación de registros
 - ▼ 📄 Configuración de Servicios
 - > 📄 Adición Servicio
 - > 📄 Consulta Servicio
 - > 📄 Adición característica a servicio
 - > 📄 Activación/Inactivación feature a servicio
 - ▼ 📄 Generador de Servicios
 - > 📄 Adición tipo de servicio
 - > 📄 Consulta de servicios por proyecto - ambiente
 - > 📄 Generación plantillas de configuración

Figura 3.3: Estructura backlog

Las historias de usuario fueron creadas en el Azure Devops de la corporación, permitiendo establecer la trazabilidad, seguimiento y cumplimiento de los lineamientos de TQ lo que responde a la necesidad del negocio. Se identificaron 5 Historias de usuario padre que corresponden a las funcionalidades del sistema y 15 Historias de usuario hijas que son las acciones que se realizan en cada funcionalidad del sistema. Después de la generación del backlog, en las historias de usuario hijas, se detallan la descripción general y los criterios de aceptación.

Las principales funcionalidades del sistema, que corresponden a las historias de usuario padre, son la configuración de parámetros, configuración de servicios, configuración de productos y el generador de servicios, las cuales se detallan más adelante.

La Figura 3.4 muestra una historia de usuario, la cual se divide en dos secciones, la descripción donde se detalla el usuario, su necesidad funcional y el objetivo a resolver con esta funcionalidad. La sección de criterios de aceptación corresponde a una lista detallada de los puntos que debe cumplir la funcionalidad a desarrollar y que son insumo para los casos de prueba que ejecuta el tester.

La historia hija que se muestra como ejemplo hace parte de la funcionalidad de gestión de parámetros y a la acción de adición de parámetros, en la historia de usuario se indica en la descripción la necesidad general y en los criterios, los puntos que debe cumplir la opción que se construirá.

292260 Adición de Parámetros

Unassigned 0 comments Add tag

State	New	Area	Capacidad TI\Infraestructura
Reason	New	Iteration	Capacidad TI\Infraestructura\19FEB2024

Description

Yo como Ingeniero de Infraestructura de TI, como parte del proyecto ERPTQ - Infraestructura Cloud
Requiero que se permita el registro de parámetros
Con el fin de administrar los parámetros generales del sistema

Acceptance Criteria

- Se debe solicitar de forma obligatoria los campos nombre, valor y estado
- Debe validar que no se permita adicionar un parámetro con mismo nombre, parámetro duplicado
- Si la información registrada es consistente debe almacenarse en la tabla INF_PARAMETER

Figura 3.4: Ejemplo historia de usuario para adición de parámetros

3.1.6. Requerimientos no funcionales

En esta sección se establecen los atributos de calidad que hacen parte de este proyecto y que están enmarcados en los lineamientos de la organización TQ:

- **Funcionalidad:** capacidad de software para realizar las tareas necesarias para que el usuario cumpla su objetivo, ofrecer resultados precisos y el grado en que se cumplen los objetivos que han sido especificados para satisfacer al usuario (Pressman and Maxim, 2022).
- **Seguridad:** capacidad de proteger la información y los datos mientras se permite cierto grado de acceso para que personas y sistemas puedan utilizar esa información según el nivel apropiado de autorización. Tiene varios componentes. La confidencialidad, es decir, que la información solo sea accesible para quienes tengan autorización, la integridad, que hace referencia la capacidad de impedir el acceso no autorizado y la autenticidad, que se mide por el grado en que se puede demostrar la identidad de un sujeto o recurso que reclama acceso (Stallings, 2022).
- **Interoperabilidad:** capacidad de diferentes sistemas, aplicaciones o componentes para intercambiar y utilizar información de manera efectiva y segura. Esto implica que los sistemas pueden comunicarse entre sí y compartir datos sin necesidad de intervención humana significativa (Tanenbaum and Bos, 2022).
- **Mantenibilidad:** facilidad con la que un sistema o componente de software puede ser modificado para corregir fallos, mejorar su rendimiento o adaptarse a cambios en el entorno. Este atributo es crucial porque permite que el software se mantenga funcional y eficiente a lo largo del tiempo (Clements and Kazman, 2022).

En la sección 3.3.4 se explican las estrategias con las cuales se busca cumplir los atributos de calidad establecidos para este proyecto.

3.2. Diseño

3.2.1. Diseño de la solución

En esta sección se describen los puntos referentes al diseño de la solución a implementar.

3.2.1.1. Stakeholders

De acuerdo al panorama actual, la necesidad y el alcance de la solución, se define junto al líder de proyecto que es el líder del equipo de ALM (proceso de gestión del ciclo de vida de una aplicación de software), cuáles son los roles de usuario de la organización que van a interactuar con la solución, su conocimiento, el rol y las funcionalidades a las que tiene acceso. Esto se detalla en la Tabla 3.3. En particular, junto con el líder del proyecto fueron definidos roles para la solución: el rol operador_servidores y el rol lider_infraestructura.

El rol operador_servidores, que corresponde a un ingeniero de infraestructura, puede realizar el aprovisionamiento de servicios mediante la opción del generador de servicios para seleccionar los servicios requeridos, crear las plantillas de configuración a partir de una parametrización preestablecida por el usuario líder y realizar el aprovisionamiento de estos servicios.

El rol lider_infraestructura, que es un jefe de infraestructura, puede hacer la configuración de parámetros para el funcionamiento de la herramienta y para la configuración base de cada servicio, la administración de los productos corporativos y sus ambientes, los servicios, atributos y valores preestablecidos e incluso la generación de servicios al igual que el rol operador.

Rol	Conocimiento	Rol	Permisos
operador_servidores	Aprovisionamiento de servicios en Azure	operador_iac	- Generador de servicios
lider_infraestructura	Configuración general de la aplicación, parametros, servicios, credenciales de acceso	lider_iac	- Configuración de parametros - Configuración de servicios - Configuración de productos y ambientes

Tabla 3.3 - Roles Aplicación

3.2.1.2. Decisiones de diseño

Para el diseño general de la solución, se establecieron las siguientes decisiones que se alinean con las necesidades y el alcance del proyecto.

- Crear una funcionalidad que permita la gestión de datos requeridos para el funcionamiento de la aplicación, integración con Devops y gestionar las estructuras base por cada tipo de servicio.
- Definir niveles para los parámetros, donde uno(1) corresponde a los parámetros clave para el funcionamiento de la solución y 2 a las estructuras de cada tipo de servicio.
- Contar con una opción que permita administrar cada producto corporativo y sus ambientes, que corresponde a los portales que se tienen en la actualidad.
- En la opción de gestión de servicios permitir definir la estructura base por cada tipo de servicio Azure, es decir, sus propiedades, secciones, subsecciones donde se ubican y los posibles valores de selección.
- Las propiedades asociadas a cada servicio pueden ser de 3 tipos, una manual donde el valor lo puede ingresar el usuario, una de tipo parámetro que solo pueda tomar el valor de un parámetro existente y otra librería que solo toma el valor de una variable del library de Devops.
- Contar con una funcionalidad operativa que permita agregar los tipos de servicios requeridos por cada producto y ambiente. Esta funcionalidad debe obtener las propiedades y valores parametrizados, y al ejecutarse, generar una plantilla para cada tipo de servicio, así como una plantilla principal que centralice la ejecución y un archivo YAML que invoque dicha plantilla principal.
- Las plantillas generadas deben versionarse en un repositorio remoto en Devops en la organización, en una estructura de carpetas definido, para esto se utiliza una API existente en la organización de Devops para hacer el versionamiento y generación de PR(Pull Request) a la rama destino.
- El *pull request* generado debe ser aprobado por lineamientos del área, por el líder de infraestructura y el equipo de arquitectura.
- A nivel del library(biblioteca de Devops donde se definen grupos de variables que pueden utilizarse en tiempo de ejecución de un pipeline) para cada producto y ambiente deben definirse variables en el library correspondientes al grupo de recursos donde se deben generar los servicios, el id de la suscripción y las variables que manejan información sensible.

3.2.1.3. Parámetros general

Una de las funciones del rol `lider_infraestructura` es realizar la configuración de los parámetros que permiten el correcto funcionamiento de la herramienta, la integración con otros sistemas y la configuración de la plantilla `bicep base` de cada servicio Azure. Teniendo en cuenta que los parámetros cumplen distintas funciones, se define que se dividan en los siguientes niveles:

- Parámetros de nivel 1: Estos corresponden a los parámetros requeridos para el funcionamiento general de la herramienta e información requerida para integrarse con Azure Devops para el consumo de servicios (rutas, datos de conexión, identificadores).
- Parámetros de nivel 2: En estos parámetros se definen listas de valores y las plantillas base por cada servicio que corresponden a la estructura fija en Bicep requerida para cada uno de ellos.

En el punto 3.3.3.2 se detalla la funcionalidad de configuración de parámetros.

3.2.1.4. Momentos de resolución

Uno de los principales objetivos del proyecto es generar plantillas de configuración de forma dinámica a partir de una configuración preestablecida, las cuales puedan ser reutilizables y se adecuen a las necesidades de cada proyecto. Para esto se definen los siguientes momentos o etapas en el proceso que permitan cumplir el objetivo.

- Configuración base servicios: primer momento donde para cada tipo de servicio Azure a aprovisionar, se requiere definir un parámetro de nivel 2 con la plantilla `bicep base`, esta labor solo la puede realizar un usuario líder.
- Configuración de servicios: segundo momento donde se debe registrar cada uno de los servicios, sus propiedades y valores por defecto, esta labor solo la puede realizar un usuario líder.
- Generación de servicios: tercer momento en el que un usuario líder u operador selecciona el producto, ambiente, servicios a aprovisionar, ajusta los valores de alguna de las propiedades de ser requerido y genera las plantillas de configuración y despliegue de los servicios en Azure. El resultado de este es la generación de las plantillas de configuración y su versionamiento en Azure, el pipeline posteriormente se ejecuta y realiza el despliegue de los servicios.

3.2.1.5. Estructura plantillas de configuración

De acuerdo al análisis realizado se detectó que las plantillas de configuración cuentan con una estructura fija y una estructura variable como se ilustra en la Figura 3.5 donde la estructura fija corresponde al nombre del recurso y la variable a las propiedades y valores requeridos para generar el servicio en Azure.

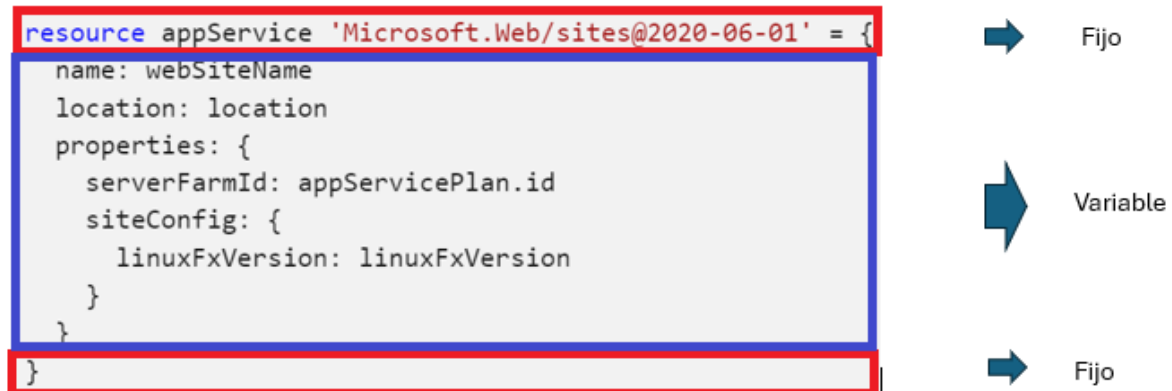


Figura 3.5: Estructura Plantilla Bicep

La estructura fija se define en un primer momento en un parámetro general de nivel 2 por cada tipo de servicio Azure con el nombre **tiposervicio-structure** y se le asigna como valor la plantilla Bicep con la estructura base de la misma.

La estructura variable se realiza en un segundo momento cuando se configura cada servicio Azure con sus atributos y valores preestablecidos.

Por último, al ejecutar la opción de generación de servicios, se obtiene de cada servicio el parámetro asociado, su plantilla base, y la configuración de atributos y valores establecidos.

El valor de la etiqueta **[content]** definido en la plantilla base, se reemplaza de forma dinámica con la configuración de propiedades y valores establecidos para el servicio previamente (parte variable que se muestra en la Figura 3.6). También, se reemplaza el valor de **[tagValues]** con el valor del parámetro general tagValues el cual incorpora las etiquetas que por políticas de la organización deben incorporarse al crear cada servicio en Azure, generando el resultado que se muestra en la imagen 3.7. De esta forma se garantiza la variabilidad en las plantillas de configuración generadas según la necesidad de cada proyecto.

```
as.bicep
1 param tagValues object = [tagValues]
2
3 resource appService 'Microsoft.Web/sites@2020-06-01' = {[content] tags: tagValues}
```

Figura 3.6: Valor del parámetro con plantilla base para servicio App Service

```
param tagValues object = {'Ambiente': 'Pruebas',
                           'Negocio': 'Corporativo',
                           'Analista': 'Alex Rodriguez',
                           'solución': 'IaC',
                           'tribu': 'Infraestructura'}

resource appService 'Microsoft.Web/sites@2020-06-01' = {
  name: 'wapp-prueba1'
  location: 'East US'
  properties: {
    serverFarmId: 'sp-pruebaiacpoc'
    siteConfig: {
      linuxFxVersion: 'node14-lts'
    }
  }
  tags: tagValues
}
```

Figura 3.7: Resultado plantilla Bicep – App Service

3.2.2. Lineamiento de arquitectura empresa TQ

En los siguientes puntos se detalla información relevante de como está constituida el área de TI y cuáles son los lineamientos que deben considerarse en la solución a implementar.

3.2.2.1. Estructura área de TI

En la Figura 3.8 se muestra como está conformado el equipo de TI desarrollo, en donde existen dos grandes equipos: equipo de demanda encargado de la gestión de proyectos y requerimientos y el equipo de soporte que brinda apoyo a los usuarios en las incidencias presentadas sobre las soluciones que se encuentran en ambiente productivo. El equipo de la demanda cuenta con un arquitecto, el cual brinda los lineamientos generales sobre los sistemas a su cargo y junto con los líderes de solución y líderes de célula, participan en el diseño de nuevas soluciones que respondan

a las necesidades de cada proyecto.

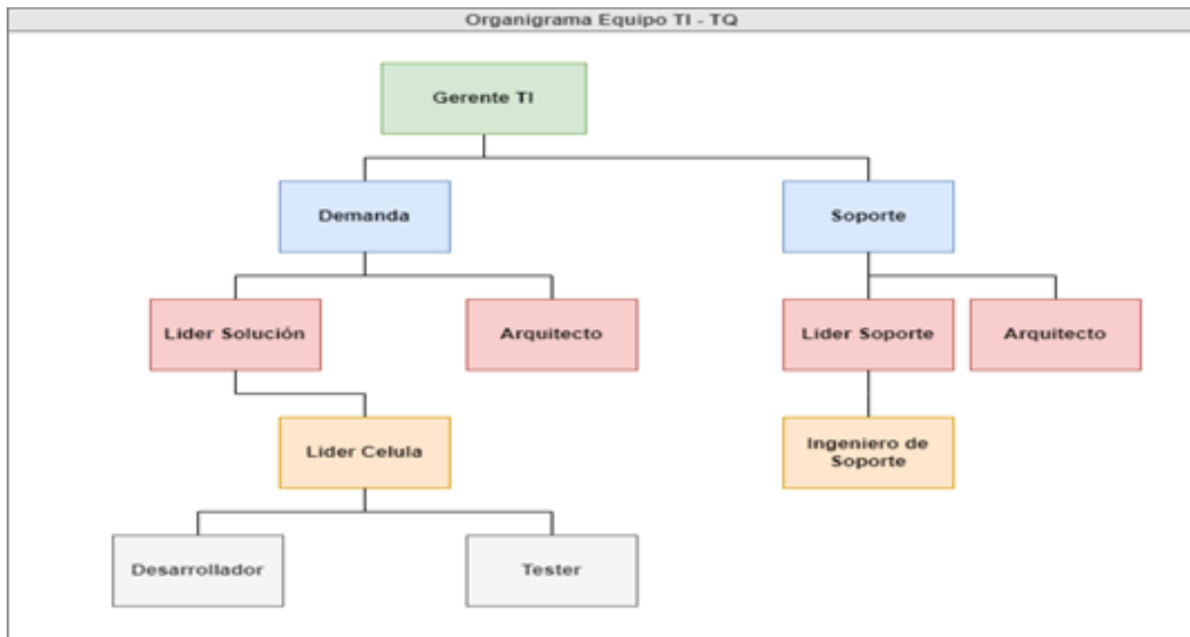


Figura 3.8: Organigrama Área TI - TQ (Elaboración propia)

Por su parte, el equipo de soporte cuenta con un arquitecto que brinda labores de apoyo y con líderes ingenieros de soporte que tienen a su cargo los sistemas sobre los cuales brindan asistencia a los usuarios finales.

3.2.2.2. ERPTQ - Arquitectura de referencia

La Figura 3.9 representa la arquitectura de referencia donde existe una plataforma global llamada ERPTQ que es una solución distribuida que permite integrar y administrar el acceso a las diversas soluciones corporativas y cuenta con unas capacidades generales que pueden ser implementadas desde cualquier módulo que se integre a la misma. La plataforma ERPTQ conformada por diversos módulos transversales como lo son el módulo general, errores, auditorías y trazas; cuenta con un servicio de seguridad y logueo que es transversal, y con un conjunto de utilitarios genéricos que permiten realizar acciones comunes como lo son el manejo de archivos, el envío de correos, entre otros.

Cada módulo que se integre a la plataforma debe contar con su aplicación web o capa de presentación (Microfrontend), una capa media que gestione las API's, la lógica de negocio y el acceso al modelo de datos (Microservicio) y una tercera capa de persistencia.

La plataforma ERPTQ cuenta con una API Gateway que es una capa media por medio de la cual se debe realizar el registro y consumo de las API's que expone cada uno de los módulos que incorporan la plataforma.

Adicionalmente, esta plataforma cuenta con unos artefactos utilitarios con funciones genéricas y transversales que van orientadas al reúso de las mismas por medio de los sistemas que incorporan la plataforma. Estos se utilizan en los módulos por medio del gestor de paquetes NuGet como librerías.

De acuerdo con los lineamientos corporativos, toda solución construida en la organización se incorporará como un módulo de ERPTQ y se debe construir siguiendo los principios de arquitectura limpia con Microservicios.

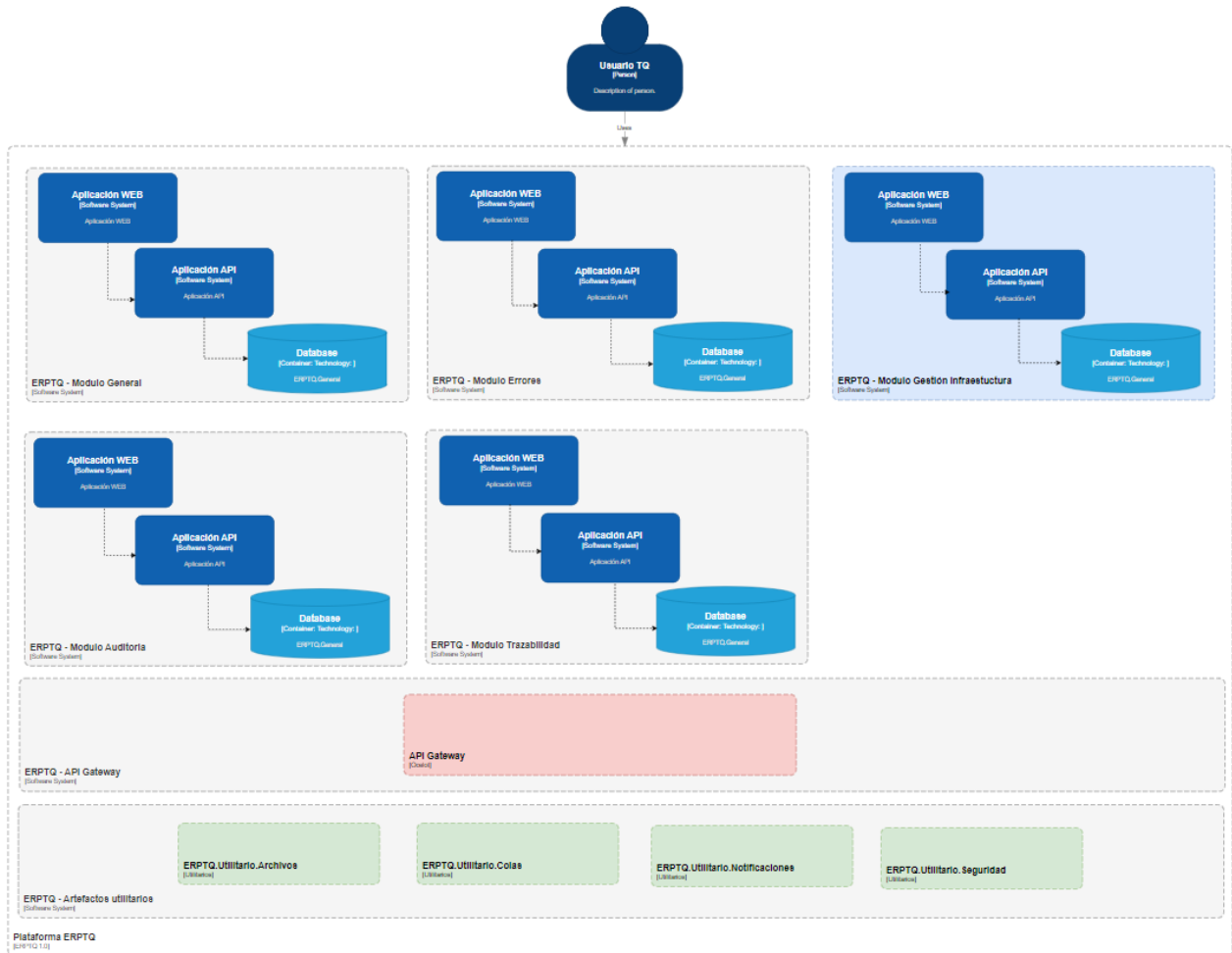


Figura 3.9: Arquitectura Plataforma ERPTQ (Fuente de arquitectura TI de la organización TQ)

3.2.3. Diagramas de diseño de la solución - Módulo Gestión de infraestructura de proyectos

A continuación se representa mediante diagramas de arquitectura C4 como se conforma la nueva solución, con qué sistemas interactúa y quienes son los actores que intervienen en el proceso.

3.2.3.1. Diagrama de Contexto

En la Figura 3.10 se representa la solución que es un módulo llamado *Gestión de Infraestructura de Proyectos* que hace parte de una gran plataforma llamada ERPTQ. Este módulo se integra con el componente de seguridad corporativo para gestionar la autorización y autenticación de los usuarios, y con Azure Devops para realizar el versionamiento de las plantillas de configuración generadas y el uso de pipelines para el despliegue automático de los servicios en Azure cloud.

A nivel de usuarios, se identifican dos tipos que corresponden al administrador, el cual se encarga de la parametrización global de la plataforma, y el ingeniero de servidores, que gestiona la configuración de los servicios y la generación de las plantillas de configuración.

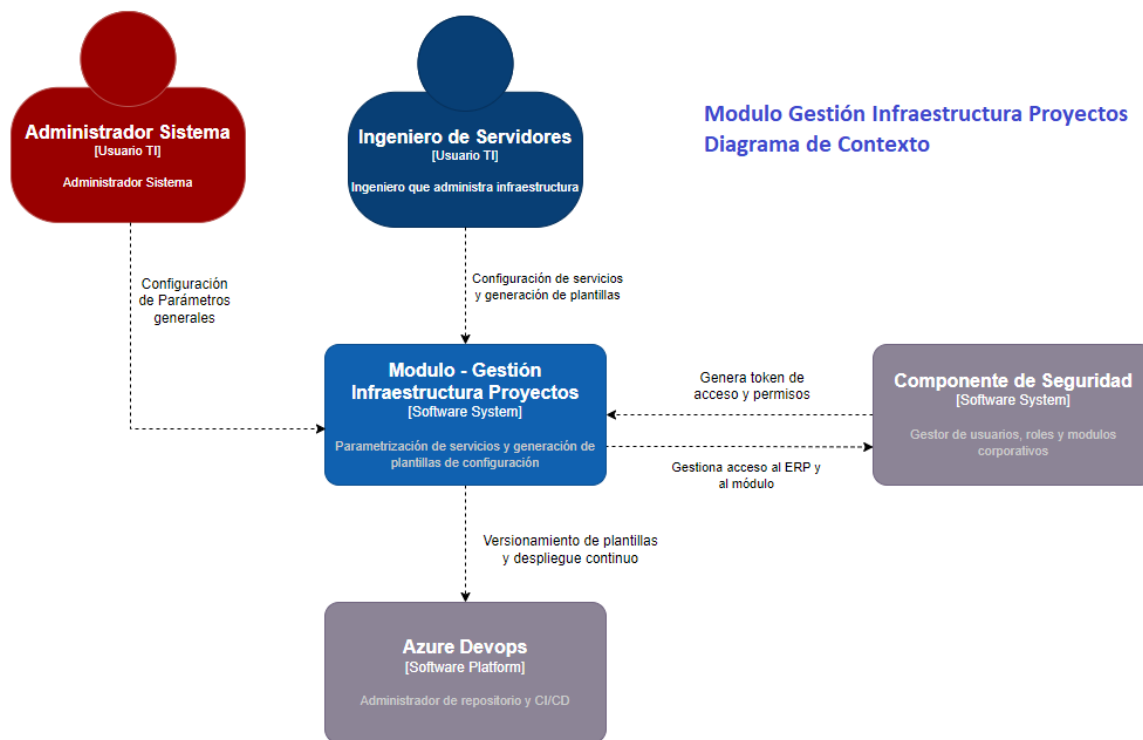


Figura 3.10: Diagrama de contexto

3.2.3.2. Diagrama de Contenedores

La Figura 3.11 detalla los contenedores que integran la aplicación (SPA, API, BD), en este caso la aplicación se compone de la siguiente manera:

- La capa front corresponde a una SPA desarrollada con Angular (HTML, CSS y Typescript), la aplicación se ejecuta en el navegador web del cliente.
- La capa back corresponde a un conjunto de API's desarrollados con framework .Net y en lenguaje C#.
- La base de datos se encuentra en un motor SQL Server.

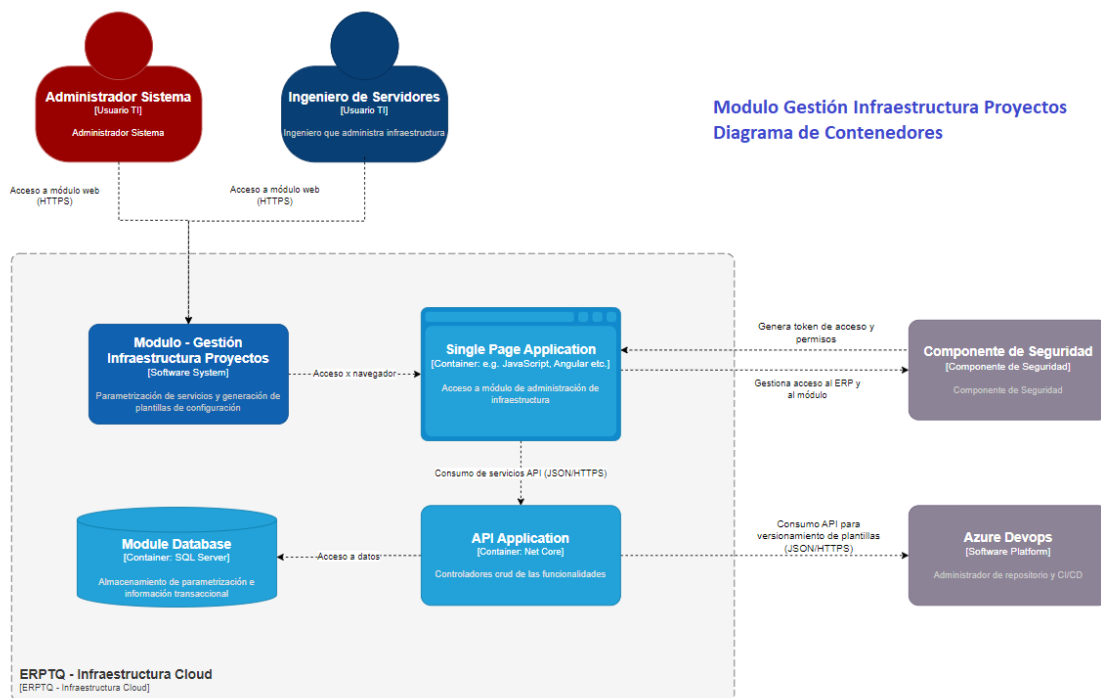


Figura 3.11: Diagrama de Contenedores

3.2.3.3. Diagrama de Componentes

En la Figura 3.12 se detallan los componentes que hacen parte de la solución, a nivel de front se maneja un SPA y en el back para la generación de los APIS, se genera un conjunto de controladores, cada uno asociado a una funcionalidad del sistema:

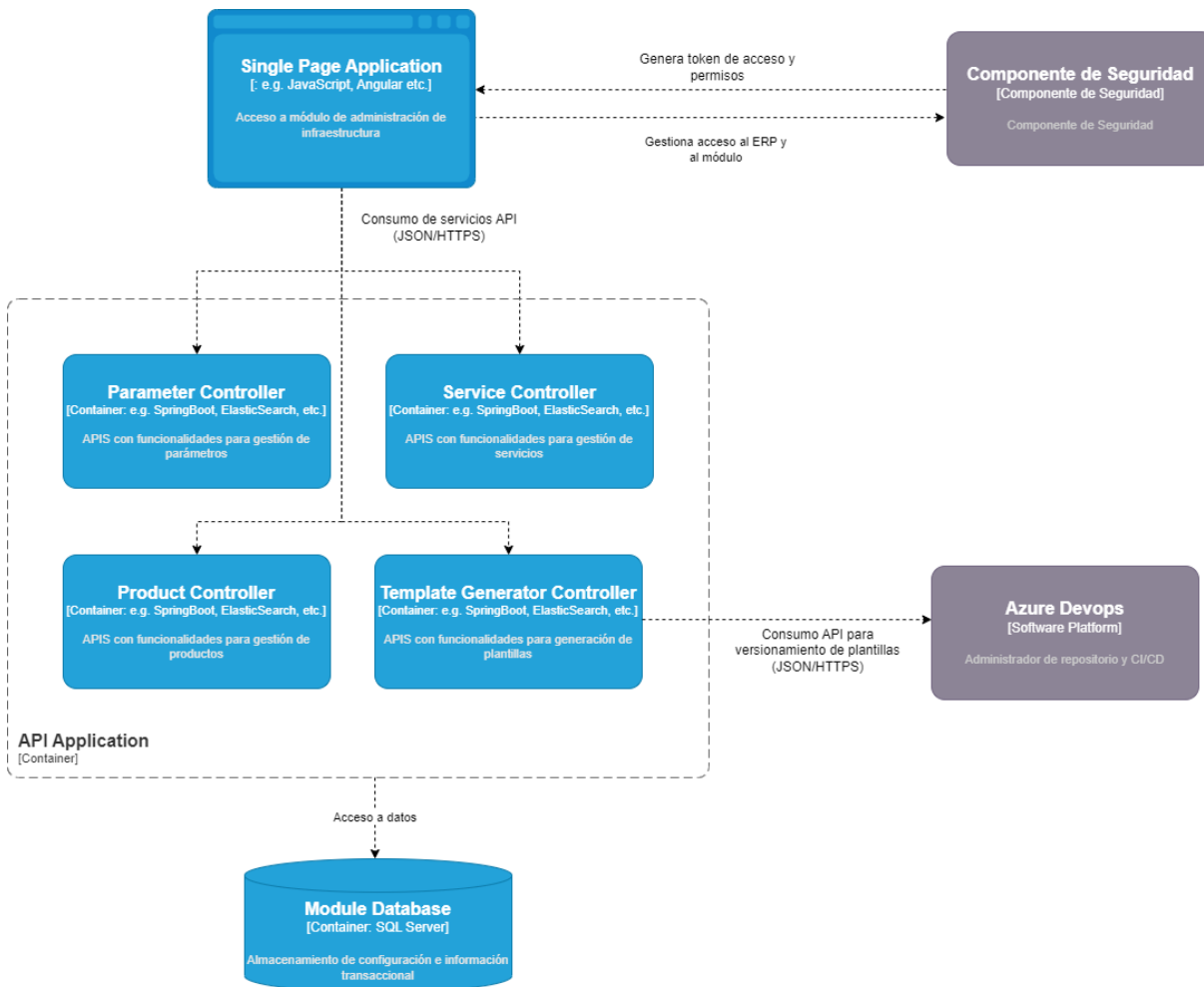


Figura 3.12: Diagrama de Componentes

-
- `ParameterController`: controlador que gestiona las operaciones CRUD asociadas al manejo de los parámetros de nivel 1 y 2 de la plataforma.
 - `ProductController`: controlador que administra los productos corporativos y sus respectivos ambientes.
 - `ServiceController`: controlador que gestiona las operaciones CRUD sobre los tipos de servicios, propiedades y valores, que definen la estructura por servicio de las plantillas de configuración.
 - `TemplateGeneratorController`: controlador operativo que genera, a partir de la configuración de cada tipo de servicio y los parámetros generales, las plantillas de configuración y el versionamiento de las mismas, integrándose con Azure Devops.

3.2.4. Diagrama de diseño de la base de datos

En la Figura 3.13 se presenta el Modelo entidad-relación del módulo **Gestión de Infraestructura de Proyectos**. Este modelo ilustra cómo se almacena la información relacionada con cada uno de los servicios Azure que se deben aprovisionar, incluyendo sus propiedades y valores predefinidos. Además, se detalla la información de los productos corporativos, sus servicios asociados y la parametrización registrada al realizar la generación de las plantillas de configuración. Adicionalmente, se incluye una tabla maestra de parámetros que registra la configuración general para el funcionamiento del módulo, así como las estructuras o plantillas base de cada servicio a aprovisionar.

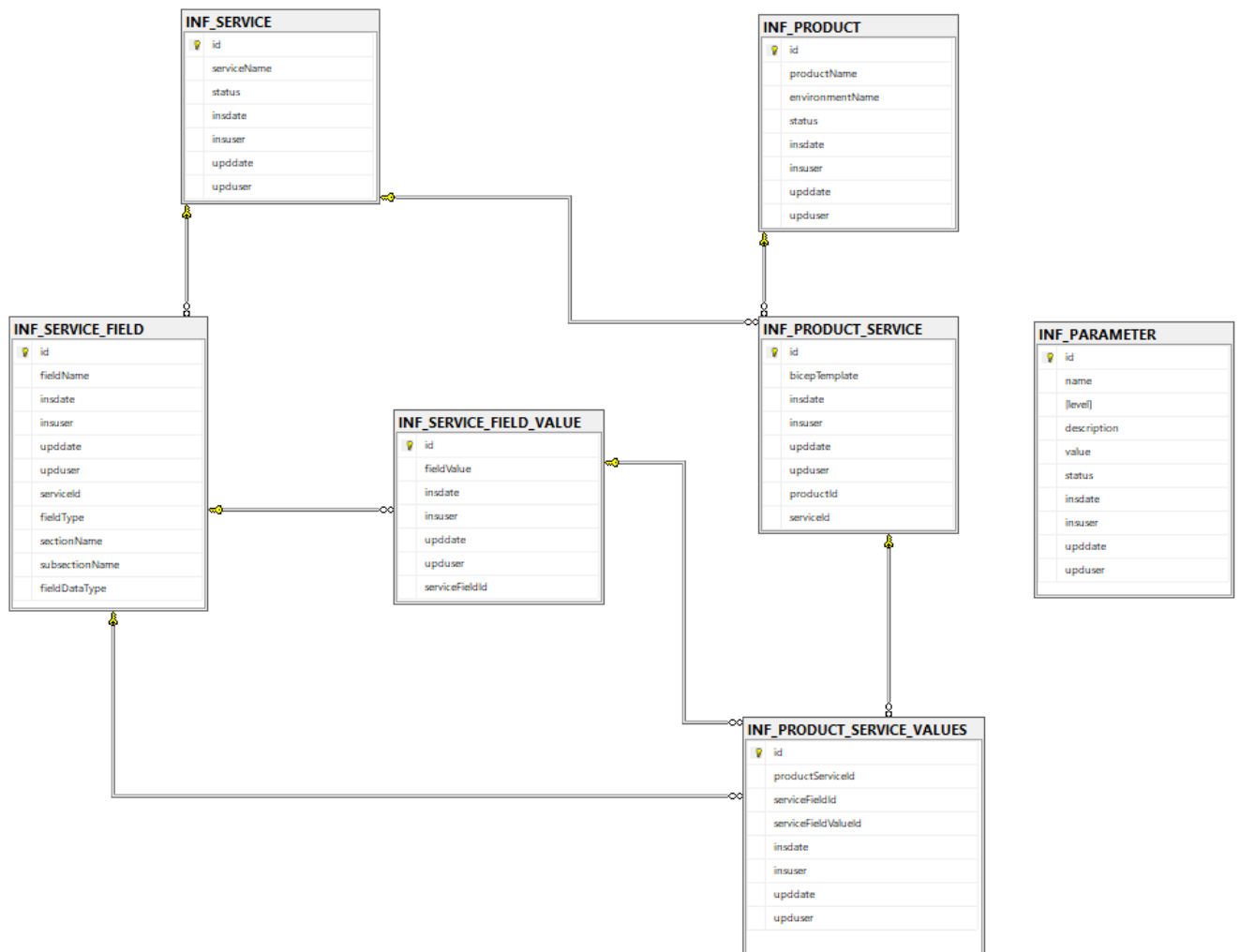


Figura 3.13: Diagrama de la Base de Datos

A continuación se indica la información que se registra en cada tabla

- **INF_SERVICE:** servicios azure.
- **INF_SERVICE_FIELD:** propiedades definidas para cada servicio.
- **INF_SERVICE_FIELD_VALUE:** valor definido para cada propiedad de los servicios.
- **INF_PRODUCT:** productos corporativos.
- **INF_PRODUCT_SERVICE:** servicios generados para cada producto corporativo.
- **INF_PRODUCT_SERVICE_VALUES:** valor indicado para cada propiedad de los servicios corporativos generados.
- **INF_PARAMETER:** parámetros generales del módulo y plantillas base de cada servicio azure.

3.3. Implementación

3.3.1. Stack tecnológico corporativo

A continuación, se detallan los frameworks definidos por el área de arquitectura de TQ para los proyectos web corporativos, así como los lineamientos y buenas prácticas definidas en la Wiki organizacional del área de TI en TQ.

- Las versiones de las diferentes herramientas usadas para la construcción de una aplicación deben tener Soporte de Largo Plazo, LTS, por ejemplo: lenguajes de programación, Frameworks, CMS, librerías, etc., deben de ser LTS en la última versión soportada al momento del desarrollo.
- Utilizar solo las librerías autorizadas por arquitectura, en las versiones aprobadas (Tener en cuenta la versión LTS).
- Toda aplicación que requiera de una capa de presentación (FRONTEND) debe ser construida con lenguajes o frameworks basados en JavaScript, HTML5 y CSS3. Los frameworks definidos son Angular y MVC-Razor como sintaxis de marcado para las soluciones nuevas y actuales.
- Toda aplicación que requiera de una capa de reglas de negocio y validaciones (BACKEND) debe ser construida con Net 6 para las nuevas soluciones.
- Las herramientas de desarrollo permitidas para desarrollo son: Visual Studio Professional (licencia), los proyectos en caso de requerir las licencias deben solicitar a infraestructura las mismas. Una vez finalizado el proyecto se retorna la licencia y se instala Visual studio code (Free).

3.3.2. Flujo del proceso en la aplicación

La Figura 3.14 y 3.15 muestra de forma global como opera el módulo **Gestión de Infraestructura de Proyectos**, cuáles son sus funcionalidades y como se integra con otras herramientas para cumplir las tareas propuestas. A continuación se van a describir cada una de las funcionalidades y como intervienen en el proceso.

El flujo de proceso se divide en 4 partes que se explican a continuación y se señalan en la siguiente imagen:

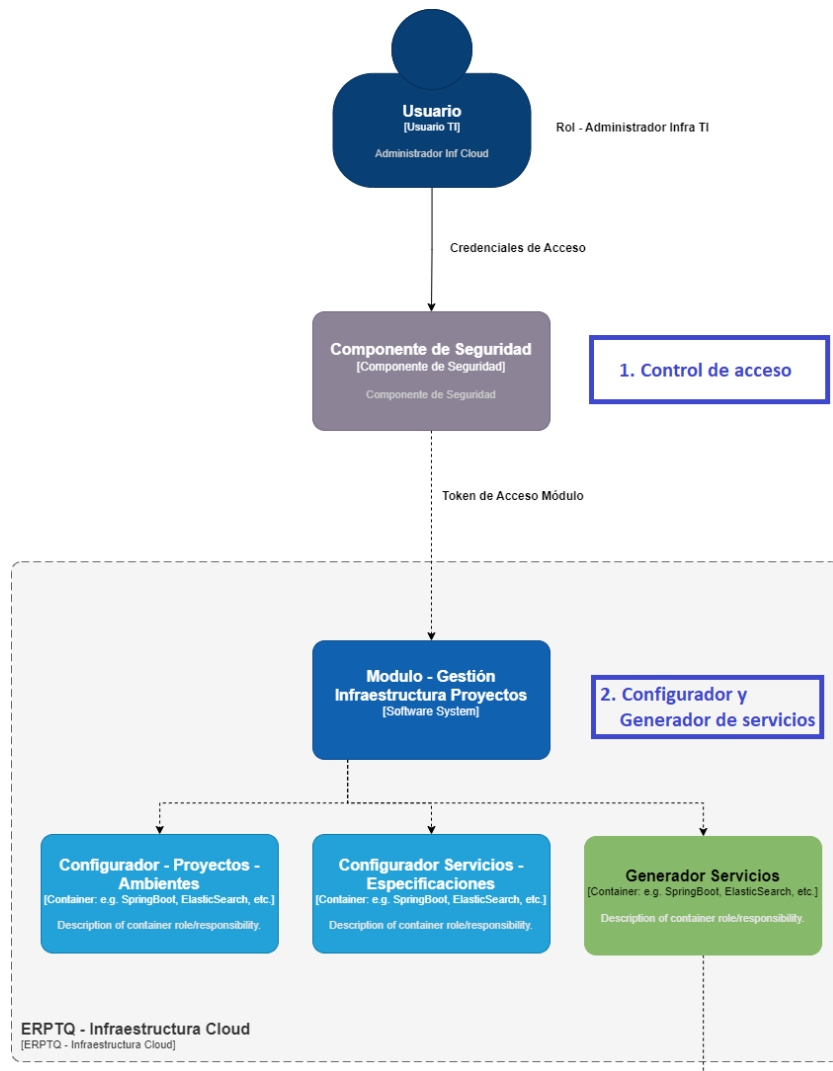


Figura 3.14: Diseño general de la solución - Parte 1

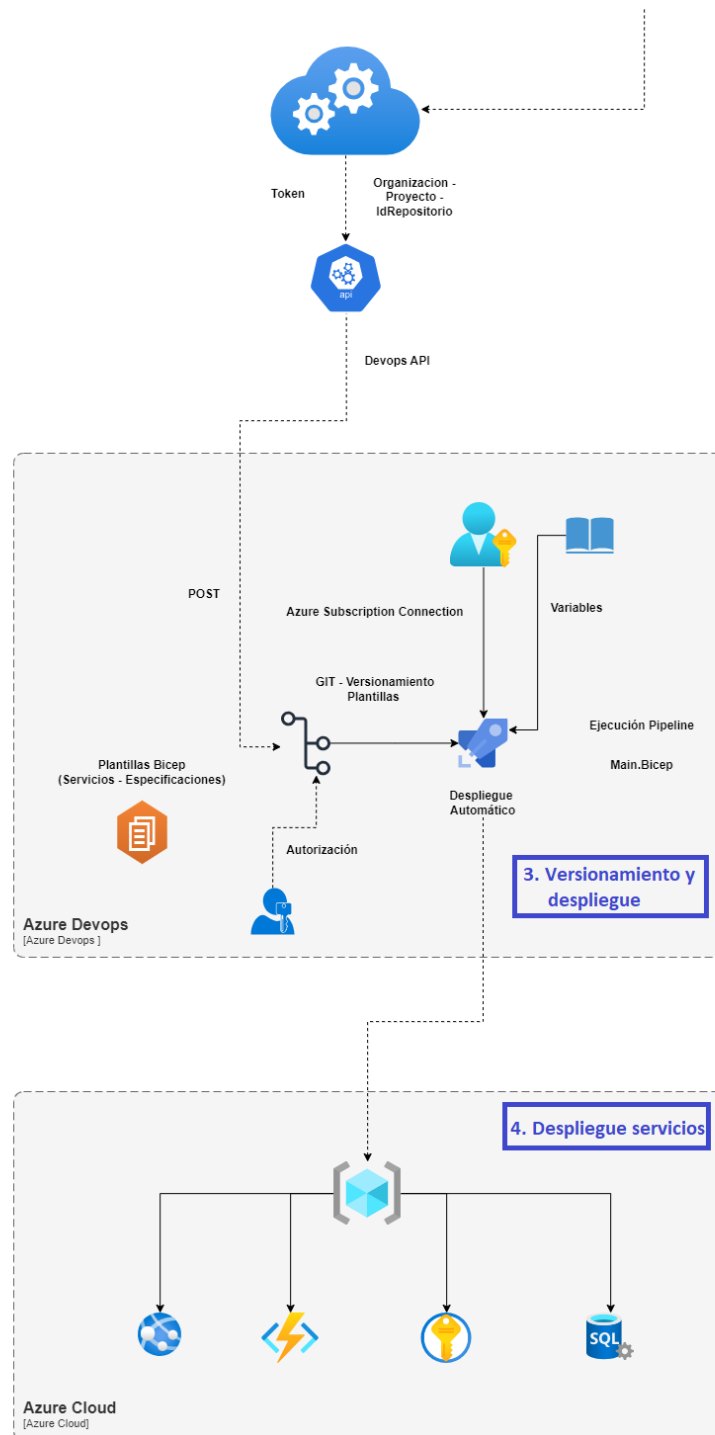


Figura 3.15: Diseño general de la solución - Parte 2

1. El módulo de Gestión de Infraestructura propuesto se integra con el componente de seguridad para el acceso seguro, como se muestra en la Figura 3.16.

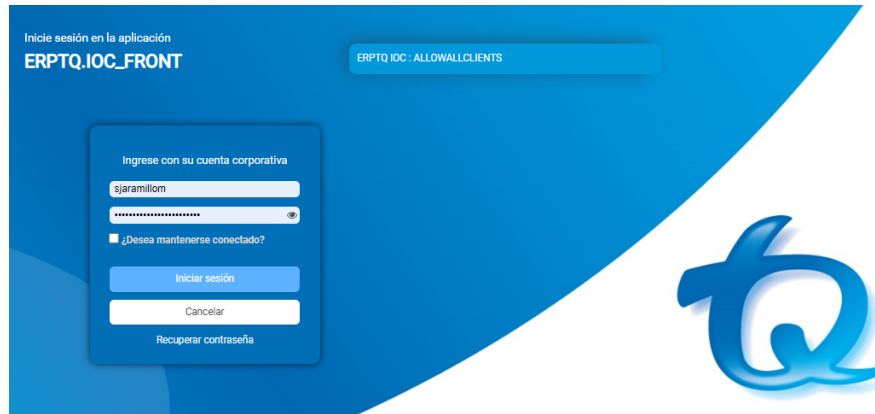


Figura 3.16: Componente de seguridad - Autenticación y Autorización

2. El módulo cuenta con funcionalidades que permiten la configuración de parámetros generales, ambientes por producto y la estructura general de propiedades y valores por cada tipo de servicio, como se muestra en la Figura 3.17.

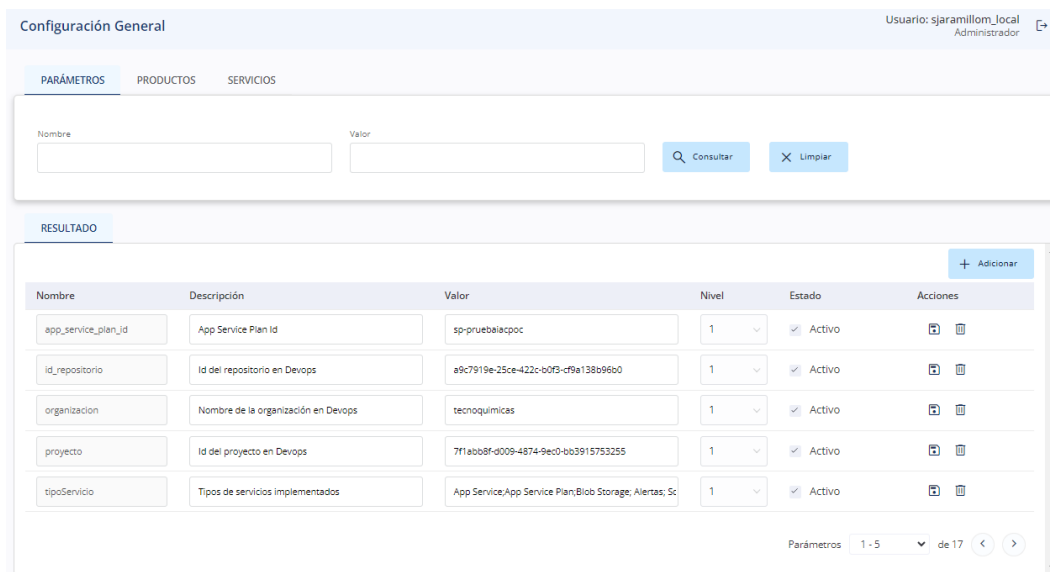


Figura 3.17: Configuración - Parámetros generales

3. El proceso de generación de servicios ejecuta un conjunto de pasos que va desde la creación de plantillas de configuración y el consumo de servicios API de DevOps que permiten versionar los cambios en una rama con la estructura “producto-ambiente-fecha”.

En la Figura 3.18, se muestra una rama generada por el sistema. La carpeta “CI” contiene el archivo YAML utilizado por el pipeline, y las plantillas de configuración se almacenan en una estructura de carpetas a nivel de producto y ambiente.

La configuración de los pipelines para cada ambiente se realizó en DevOps, incorporando la conexión segura hacia Azure y la ejecución inicial del archivo YAML generado desde la aplicación. Este archivo YAML ejecuta el archivo main.bicep, el cual integra todos los servicios a aprovisionar.

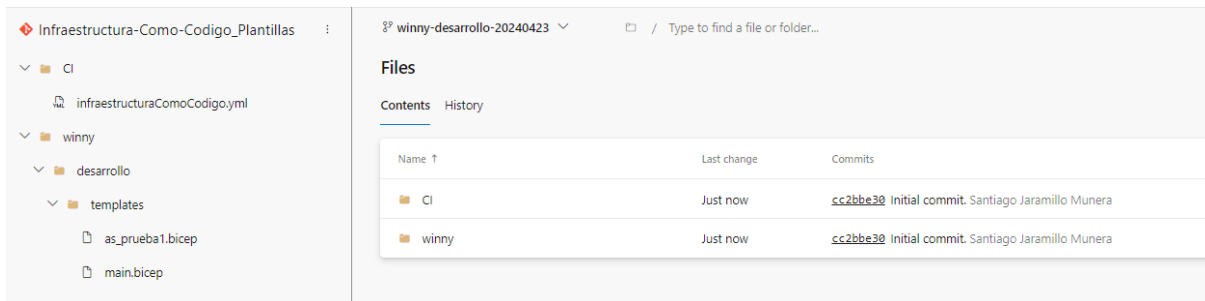


Figura 3.18: Estructura Rama

- Al completar el pull request, el pipeline ejecuta la plantilla main.bicep y por medio de una conexión segura se integra con Azure cloud para aprovisionar los servicios seleccionados en el grupo de recursos parametrizado en las variables del pipeline como se muestra en la Figura 3.19.

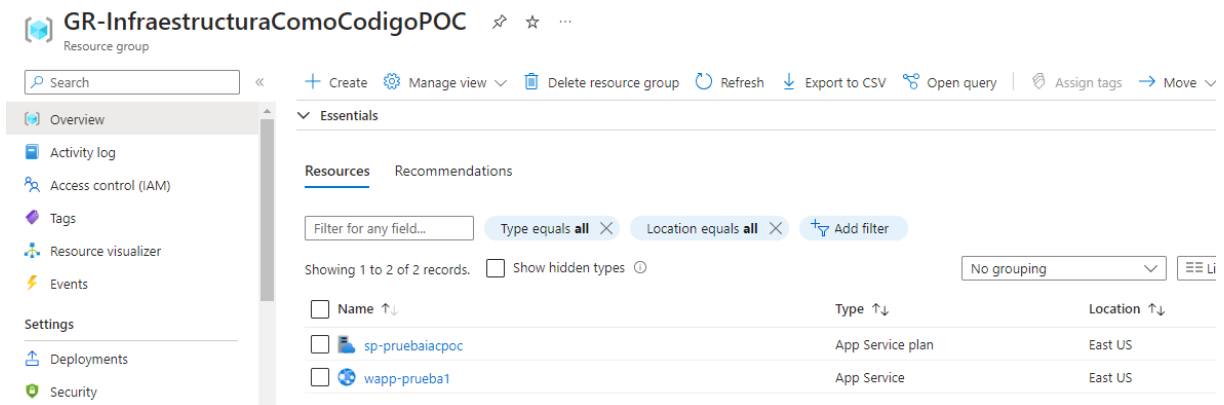


Figura 3.19: Servicios desplegados

Las siguientes secciones de este capítulo presentan los detalles de como se implementaron las funcionalidades que soporte el módulo **Gestión de Infraestructura de Proyectos**

3.3.3. Funcionalidades de la aplicación

3.3.3.1. Diagrama de Funcionalidades

En la Figura 3.20 se presentan las cuatro funcionalidades macro que componen el módulo, las cuales se detallan a continuación.

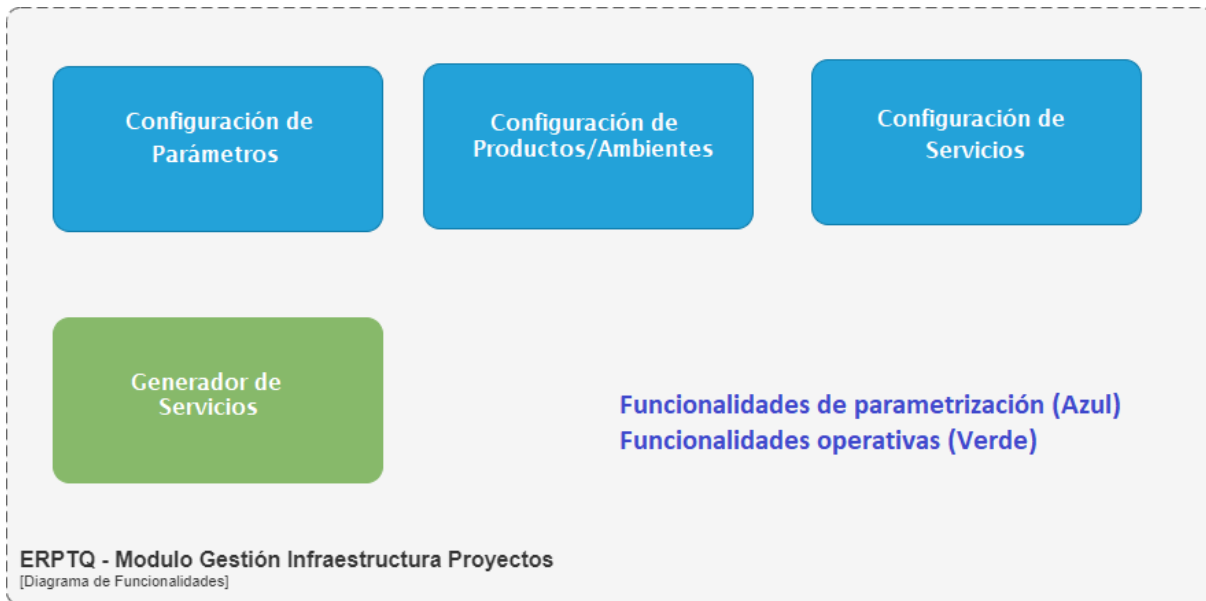


Figura 3.20: Diagrama de Funcionalidades

3.3.3.2. Configuración de Parámetros

En la Figura 3.21 se presenta la funcionalidad de configuración de parámetros generales requeridos para el correcto funcionamiento del módulo **Gestión de Infraestructura de Proyectos**, como se mencionó en la sección 3.2.1.3 - Parametrización general los parámetros se encuentran segmentados en niveles donde los parámetros de nivel 1 son los requeridos para el funcionamiento general del módulo y para la integración con Devops, los parámetros de nivel 2 son donde se realiza la parametrización a nivel de cada tipo de servicio a administrar y aprovisionar.

Cabe recordar, que como se menciona en el punto 3.2.1.3, los parámetros de nivel 1 solo pueden ser modificados por un usuario con rol de administrador y los de nivel 2 por el administrador o el usuario operador.

The screenshot displays the 'Configuración General' interface. At the top right, the user is identified as 'sjaramillom_local Administrador'. Below the title bar, there are three tabs: 'PARÁMETROS', 'PRODUCTOS', and 'SERVICIOS'. A search bar is present with 'Nombre' and 'Valor' input fields, a 'Consultar' button, and a 'Limpiar' button. The main content area is titled 'RESULTADO' and contains a table with the following data:

Nombre	Descripción	Valor	Nivel	Estado	Acciones
app_service_plan_id	App Service Plan Id	1	1	Activo	[Iconos]
id_repositorio	Id del repositorio en Devops	a9c7919e-25ce-422c-b0f3-cf9a138b96b0	1	Activo	[Iconos]
organizacion	Nombre de la organización en Devops	tecnoquimicas	1	Activo	[Iconos]
proyecto	Id del proyecto en Devops	4cfd6fed-7a25-48af-adb5-7a886b022cf5	1	Activo	[Iconos]
tipoServicio	Tipos de servicios implementados	App Service:Blob Storage:Alertas	1	Activo	[Iconos]

At the bottom right of the table, there is a pagination control showing 'Parámetros 1 - 5 de 12' with navigation arrows.

Figura 3.21: Funcionalidad - Configuración de Parámetros

3.3.3.3. Configuración de Productos

En la Figura 3.22 se presenta la funcionalidad de configuración de productos con sus respectivos ambientes, sobre los cuales se generan las plantillas de configuración de los servicios a aprovisionar desde la herramienta.

Esta funcionalidad da como resultado la configuración de cada uno de los productos que tienen un portal corporativo, los ambientes se requiere que opere (desarrollo, pruebas, producción) y si esta configuración se encuentra activa, teniendo en cuenta que si no está activa no se visualizará posteriormente en la opción del Generador de Servicios.

Esta funcionalidad puede ser utilizada por usuarios con rol de administrador o de operador.

Configuración General Usuario: sjaramillom_local
Administrador

PARÁMETROS **PRODUCTOS** SERVICIOS

Producto Ambiente Consultar Limpiar

RESULTADO + Adicionar

Nombre Producto	Nombre Ambiente	Estado	Acciones
<input type="text" value="winny"/>	<input type="text" value="desarrollo"/>	Activo	<input type="checkbox"/> <input type="checkbox"/>
<input type="text" value="winny"/>	<input type="text" value="pruebas"/>	Activo	<input type="checkbox"/> <input type="checkbox"/>
<input type="text" value="winny"/>	<input type="text" value="produccion"/>	Activo	<input type="checkbox"/> <input type="checkbox"/>
<input type="text" value="noraver"/>	<input type="text" value="desarrollo"/>	Activo	<input type="checkbox"/> <input type="checkbox"/>
<input type="text" value="noraver"/>	<input type="text" value="pruebas"/>	Activo	<input type="checkbox"/> <input type="checkbox"/>

Productos 1 - 5 de 6

Figura 3.22: Funcionalidad - Configuración de Productos

3.3.3.4. Configuración de Servicios

En la Figura 3.23 se presenta la funcionalidad que permite la configuración de cada tipo de servicio, sus propiedades y valores, los cuales definen la estructura base para la generación de las plantillas de configuración. En esta imagen se muestra a manera de ejemplo la configuración de un servicio App Service.

Esta funcionalidad puede ser utilizada por usuarios con rol de administrador o de operador.

The screenshot displays the 'Configuración General' interface for 'SERVICIOS'. At the top, there are tabs for 'PARÁMETROS', 'PRODUCTOS', and 'SERVICIOS'. Below the tabs, there is a search bar for 'Servicio' with a 'Consultar' button and a 'Limpiar' button. The main section is titled 'SERVICIOS' and contains a table with columns: 'Registro', 'Servicio', 'Estado', and 'Acciones'. The first row is selected and shows 'App Service' with an 'Activo' status. Below this, there is a 'CARACTERÍSTICAS' section with a table with columns: 'Registro', 'Característica', 'Sección', 'Subsección', 'Tipo', and 'Acciones'. The first row is selected and shows 'name' with a 'Manual' type. The second row shows 'token' with a 'Manual' type. The third row shows 'servFamilo' with a 'Parametro' type. The fourth row shows 'inurlArteson' with a 'Manual' type. At the bottom, there is a 'VALORES' section with a table with columns: 'Valor' and 'Acciones'. The first row shows 'is_prueba'.

Registro	Servicio	Estado	Acciones
<input checked="" type="checkbox"/>	App Service	Activo	[Iconos]

Registro	Característica	Sección	Subsección	Tipo	Acciones
<input checked="" type="checkbox"/>	name			Manual	[Iconos]
<input type="checkbox"/>	token			Manual	[Iconos]
<input type="checkbox"/>	servFamilo	propiedades		Parametro	[Iconos]
<input type="checkbox"/>	inurlArteson	propiedades	eneConfig	Manual	[Iconos]

Valor	Acciones
is_prueba	[Iconos]

Figura 3.23: Funcionalidad - Configuración de Servicios

3.3.3.5. Generador de Servicios

En la Figura 3.24 se presenta la funcionalidad de generación de servicios, que permite para un proyecto y ambiente, seleccionar los tipos de servicio, ajustar valores en las propiedades y generar las plantillas de configuración en formato Bicep por cada tipo de servicio, la plantilla main en formato Bicep que ejecuta cada módulo, y el archivo yaml(archivo de configuración de arranque del pipeline en Devops) que será utilizado posteriormente por el pipeline.

Adicionalmente, el módulo **Gestión de Infraestructura de Proyectos** se integra con Azure Devops para la generación de una rama y el versionamiento de las plantillas de configuración en el repositorio, también genera un pull request hacia la rama destino.

Figura 3.24: Funcionalidad - Generador de Servicios

La plantilla de configuración Bicep generada a partir de la parametrización realizada se muestra en la Figura 3.25

```

1 param tagValues object = {'Ambiente': 'Pruebas', 'Negocio': 'Corporativo', 'Analista': 'Alexander Rodriguez Ospina', 'solución': 'IaC', 'tribu': 'Infraestructura'}
2
3 -- resource appService 'Microsoft.Web/sites@2020-06-01' = {
4 -- name: 'wapp-pruebal'
5 -- location: 'East US'
6 -- properties: {
7 -- serverFarmId: 'sp-pruebaiacpoc'
8 -- siteConfig: {
9 -- linuxFxVersion: 'node14-lts'
10 -- }
11 -- }
12 tags: tagValues}

```

Figura 3.25: Resultado - Plantilla de Configuración Bicep

3.3.4. Estrategias para cumplir atributos de calidad

A continuación se detallan las estrategias mediante las cuales se busca cumplir con los atributos de calidad establecidos para este proyecto:

Funcionalidad

- Revisión y aprobación de historias por parte del usuario, validando que el alcance corresponda a lo requerido
- Pruebas en conjunto con el usuario, validando el funcionamiento de cada funcionalidad y cruzando contra los criterios de aceptación de cada historia, calificando la completitud funcional
- Implementación de pruebas unitarias en el back

Seguridad

- Integración al componente de seguridad corporativo para la autenticación y acceso de los usuarios
- Definición de roles y configuración de permisos en el componente de seguridad que son validados cuando se ingresa al sistema.

Interoperabilidad

- Integración con el componente de seguridad mediante las API y librerías dispuestas por el lineamiento de arquitectura.
- Consumo de servicios de terceros como Azure para el versionamiento de objetos y despliegue de servicios.

Mantenibilidad

- Desarrollo de funcionalidades modulares, donde cada una cumple una responsabilidad específica requerida por el proyecto.
- Seguir los lineamientos de arquitectura definidos por la organización y seguir las prácticas de arquitectura limpia

3.4. Pruebas funcionales

En esta sección, se muestran las evidencias de las pruebas funcionales realizadas en este proyecto para evaluar el funcionamiento del módulo **Gestión de Infraestructura de Proyectos**. Las pruebas funcionales realizadas buscaban garantizar que cada componente del software operara conforme a las especificaciones y requisitos establecidos. Este proceso permite asegurar la calidad del producto final y cumplir con el alcance establecido.

A continuación se muestran las evidencias de las siguientes funcionalidades testeadas:

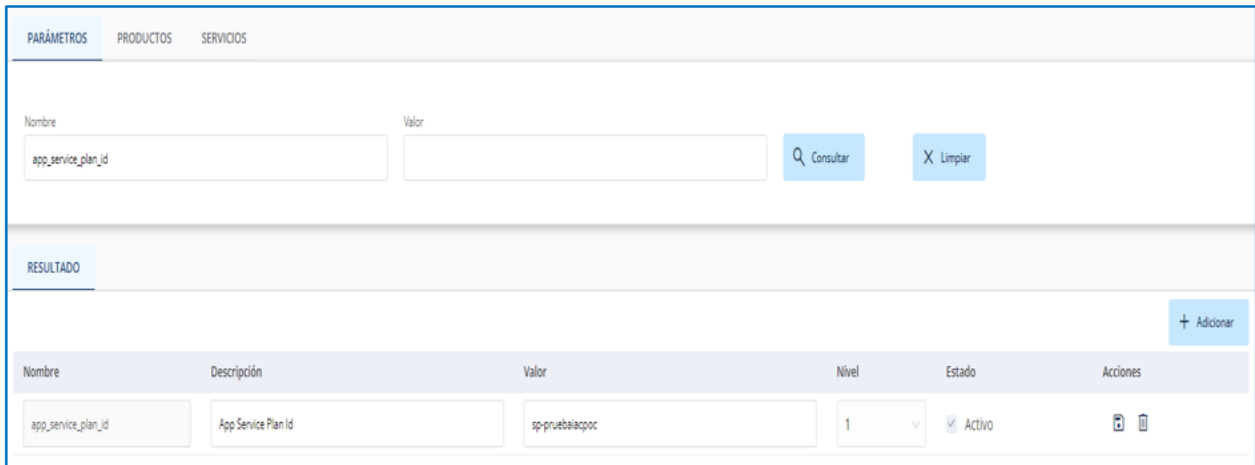
- Configurador de parámetros
- Configurador de productos
- Configurador de servicios
- Generador de servicios

ID del Caso de Prueba: TC001
Funcionalidad: Configurador de parámetros
Nombre del Caso de Prueba: Consulta de parámetros
Descripción: Este caso de prueba verifica que se realice la consulta por nombre y/o valor.
Precondiciones: El usuario debe estar registrado y tener credenciales válidas.

Pasos para Ejecutar la Prueba:

1. Abrir la aplicación.
2. Acceda a la opción Configurador Servicios
3. Ingrese en el campo Nombre del panel superior el parámetro a consultar
4. Presionar clic en el botón Consultar

Resultado Esperado: Debe mostrarse en la grilla inferior los registros que coincidan con el filtro de búsqueda.
Estado: Aprobado
Evidencia:





Nombre	Descripción	Valor	Nivel	Estado	Acciones
app_service_plan_id	App Service Plan Id	sp-pruebaiaqpc	1	Activo	 

Figura 3.26: Configurador de parámetros - Caso de prueba 1

ID del Caso de Prueba: TC002

Funcionalidad: Configurador de parámetros

Nombre del Caso de Prueba: Limpiar filtro de parámetros

Descripción: Este caso de prueba verifica que se reestablezcan los filtros de la consulta y la grilla de resultados.

Precondiciones: El usuario debe estar registrado y tener credenciales válidas.

Pasos para Ejecutar la Prueba:

1. Abrir la aplicación.
2. Acceda a la opción Configurador Servicios
3. Ingrese en el campo Nombre del panel superior el parámetro a consultar
4. Presionar clic en el botón Consultar
5. Presione el botón Limpiar

Resultado Esperado: Debe mostrarse en la grilla inferior todos los registros y limpiar todos los campos de filtro.

Estado: Aprobado

Evidencia:

Configuración General Usuario: spramirio@
Administrador

PARÁMETROS PRODUCTOS SERVICIOS

Nombre: Valor: Consultar Limpiar

RESULTADO + Adicionar

Nombre	Descripción	Valor	Nivel	Estado	Acciones
app_service_plan_id	App Service Plan Id	app-pruebasapoc	1	Activo	

Parámetros 1 - 5 de 1

Figura 3.27: Configurador de parámetros - Caso de prueba 2

ID del Caso de Prueba: TC003


Funcionalidad: Configurador de parámetros

Nombre del Caso de Prueba: Adición de parámetros

Descripción: Este caso de prueba verifica que se permita registrar parámetros.

Precondiciones: El usuario debe estar registrado y tener credenciales válidas.











Pasos para Ejecutar la Prueba:

1. Abrir la aplicación.
2. Acceda a la opción Configurador Servicios
3. Ubicado en el tab Parámetros, presione clic en el botón Adicionar
4. Ingresar el valor para los campos Nombre, Descripción, Valor, Nivel y Estado
5. Presione clic en el botón  (Guardar) en el panel de acciones

Resultado Esperado: Debe mostrarse una alerta notificando que el registro se creó satisfactoriamente y debe verse el nuevo registro en la grilla de resultados

Estado: Aprobado

Evidencia:

Nombre	Descripción	Valor	Nivel	Estado	Acciones
app_service_plan_id	App Service Plan Id	sp-pruebaapoc	1	Activo	 
id_repositorio	Id del repositorio en Devops	a9c7919e-25ce-422c-60f9-ef9a138b96b0	1	Activo	 
organizacion	Nombre de la organización en Devops	tecnouimicas	1	Activo	 
proyecto	Id del proyecto en Devops	7f1aabb8f-d009-4874-9e0d-bb3915753255	1	Activo	 
tipoServicio	Tipos de servicios implementados	App Service; App Service Plan; Sql Database; Sql Server	1	Activo	 

Parámetros 1 - 5 de 18


 Se creó el registro satisfactoriamente

Figura 3.28: Configurador de parámetros - Caso de prueba 3

ID del Caso de Prueba: TC004


Funcionalidad: Configurador de parámetros

Nombre del Caso de Prueba: Modificación de parámetros

Descripción: Este caso de prueba verifica que se permita modificar valores en los parámetros existentes.

Precondiciones: El usuario debe estar registrado y tener credenciales válidas.

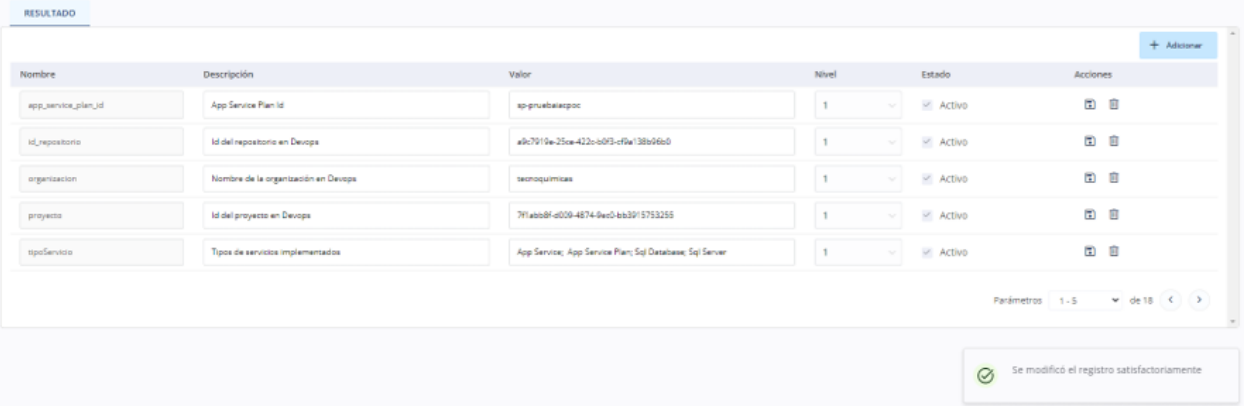
Pasos para Ejecutar la Prueba:

1. Abrir la aplicación.
2. Acceda a la opción Configurador Servicios
3. Ubicado en el tab parámetros, selección en la grilla de resultados el registro a modificar
4. Modifique el valor del campo descripción y valor
5. Presione clic en el botón  (Guardar) en el panel de acciones











Resultado Esperado: Debe mostrarse una alerta notificando que el registro se modificó satisfactoriamente y debe verse el nuevo registro en la grilla de resultados

Estado: Aprobado

Evidencia:



The screenshot shows a web interface for parameter configuration. At the top, there is a tab labeled 'RESULTADO' and a '+ Adicionar' button. Below is a table with the following columns: Nombre, Descripción, Valor, Nivel, Estado, and Acciones. The table contains five rows of parameters. At the bottom right, there is a success alert: 'Se modificó el registro satisfactoriamente' with a green checkmark icon.

Nombre	Descripción	Valor	Nivel	Estado	Acciones
app_service_plan_id	App Service Plan Id	sp-pruebasazpoc	1	ACTIVO	 
id_repositorio	Id del repositorio en Devops	a3c7910e-25ca-412c-b093-ef6e138b06b0	1	ACTIVO	 
organizacion	Nombre de la organización en Devops	tecnosquimicas	1	ACTIVO	 
proyecto	Id del proyecto en Devops	7f1aabb8f-c009-4874-8ec0-bb30f5753255	1	ACTIVO	 
tipoServicio	Tipos de servicios implementados	App Service; App Service Plan; Sql Database; Sql Server	1	ACTIVO	 

Parámetros 1 - 5 de 18

Se modificó el registro satisfactoriamente

Figura 3.29: Configurador de parámetros - Caso de prueba 4

ID del Caso de Prueba: TC005


Funcionalidad: Configurador de parámetros

Nombre del Caso de Prueba: Eliminación de parámetros

Descripción: Este caso de prueba verifica que se permita borrar parámetros existentes.

Precondiciones: El usuario debe estar registrado y tener credenciales válidas.











Pasos para Ejecutar la Prueba:

1. Abrir la aplicación.
2. Acceda a la opción Configurador Servicios
3. Ubicado en el tab parámetros, selección en la grilla de resultados el registro a eliminar
4. Presione clic en el botón  (Eliminar) en el panel de acciones

Resultado Esperado: Debe mostrarse una alerta notificando que el registro se eliminó satisfactoriamente y no debe verse el registro en la grilla de resultados

Estado: Aprobado

Evidencia:

Nombre	Descripción	Valor	Nivel	Estado	Acciones
id_repositorio	Id del repositorio en Devops	a9c7919e-25ce-422c-b0f3-cf9a138b9960	1	Activo	 
organizacion	Nombre de la organización en Devops	tecniquimicas	1	Activo	 
proyecto	Id del proyecto en Devops	7f1ab08f-e009-4874-9ec0-b63915753255	1	Activo	 
tipoServicio	Tipos de servicios implementados	App Service; App Service Plans; Sql Database; Sql Server	1	Activo	 
app_service_structure	Estructura tipoapp de servicio app service	param sag/values object = [sag/values] resource appService Microsoft.Wi	2	Activo	 

Parámetros 1 - 5 de 16


 Se eliminó el registro satisfactoriamente

Figura 3.30: Configurador de parámetros - Caso de prueba 5

ID del Caso de Prueba: TC006

Funcionalidad: Configurador de productos

Nombre del Caso de Prueba: Consulta de productos

Descripción: Este caso de prueba verifica que se realice la consulta por producto y/o ambiente.

Precondiciones: El usuario debe estar registrado y tener credenciales válidas.

Pasos para Ejecutar la Prueba:

1. Abrir la aplicación.
2. Acceda a la opción Configurador Servicios, y seleccione la pestaña PRODUCTOS
3. Ingrese en nombre del producto y ambiente a consultar
4. Presionar clic en el botón Consultar

Resultado Esperado: Debe mostrarse en la grilla inferior los registros que coincidan con el filtro de búsqueda.

Estado: Aprobado

Evidencia:

Configuración General Usuario: sjaramillom
Administrador

PARÁMETROS **PRODUCTOS** SERVICIOS

Producto: winry Ambiente: pruebas Consultar Limpiar

RESULTADO + Adicionar

Nombre Producto	Nombre Ambiente	Estado	Acciones
winry	pruebas	Activo	

Productos 1 - 5 de 1

Figura 3.31: Configurador de productos - Caso de prueba 6

ID del Caso de Prueba: TC007

Funcionalidad: Configurador de productos

Nombre del Caso de Prueba: Limpiar filtro de consulta

Descripción: Este caso de prueba verifica que se reestablezcan los filtros de la consulta y la grilla de resultados.

Precondiciones: El usuario debe estar registrado y tener credenciales válidas.

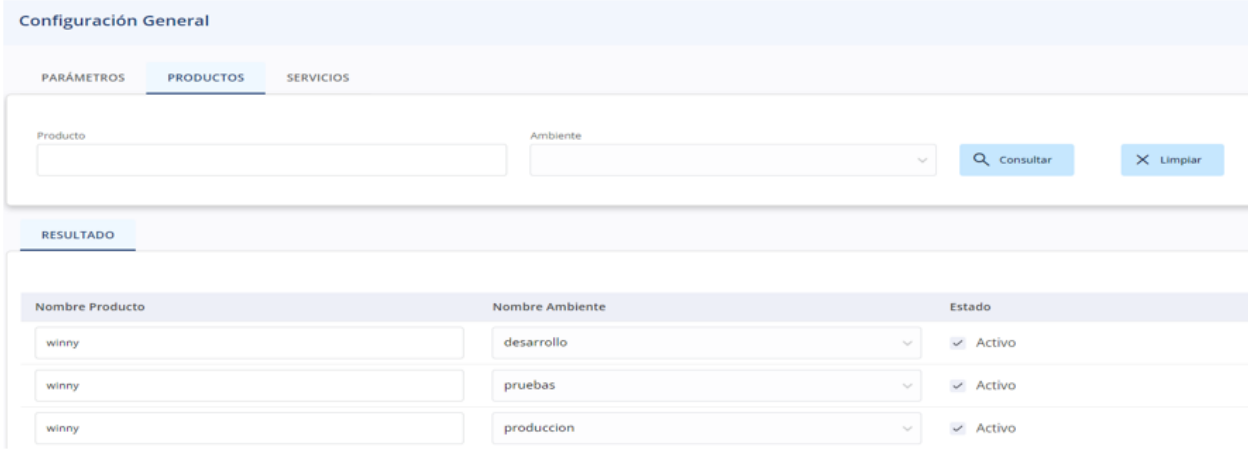
Pasos para Ejecutar la Prueba:

1. Abrir la aplicación.
2. Acceda a la opción Configurador Servicios, y seleccione la pestaña PRODUCTOS
3. Ingrese en nombre del producto y ambiente a consultar
4. Presionar clic en el botón Consultar
5. Presione el botón Limpiar

Resultado Esperado: Debe mostrarse en la grilla inferior todos los registros y limpiar todos los campos de filtro.

Estado: Aprobado

Evidencia:



The screenshot shows a web interface titled 'Configuración General' with three tabs: 'PARÁMETROS', 'PRODUCTOS', and 'SERVICIOS'. The 'PRODUCTOS' tab is active. It contains two input fields: 'Producto' and 'Ambiente'. Below these fields are two buttons: 'Consultar' (with a magnifying glass icon) and 'Limpiar' (with an 'X' icon). Below the search area is a 'RESULTADO' section containing a table with three columns: 'Nombre Producto', 'Nombre Ambiente', and 'Estado'. The table lists three entries, all with 'winny' in the product name column and 'Activo' in the status column.

Nombre Producto	Nombre Ambiente	Estado
winny	desarrollo	✓ Activo
winny	pruebas	✓ Activo
winny	produccion	✓ Activo

Figura 3.32: Configurador de productos - Caso de prueba 7

ID del Caso de Prueba: TC008


Funcionalidad: Configurador de productos

Nombre del Caso de Prueba: Adición de productos

Descripción: Este caso de prueba verifica que se permita registrar productos.

Precondiciones: El usuario debe estar registrado y tener credenciales válidas.

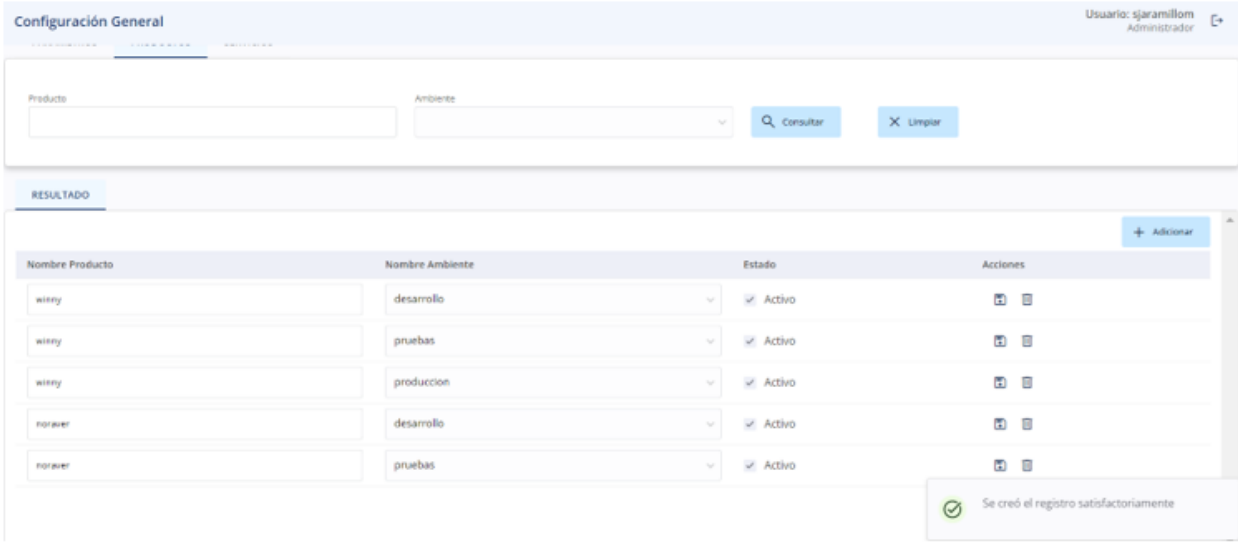
Pasos para Ejecutar la Prueba:

1. Abrir la aplicación.
2. Acceda a la opción Configurador Servicios, y seleccione la pestaña PRODUCTOS
3. Presione clic en el botón Adicionar
4. Ingresar el valor para los campos Nombre Producto, Nombre Ambiente y Estado
5. Presione clic en el botón  (Guardar) en el panel de acciones











Resultado Esperado: Debe mostrarse una alerta notificando que el registro se creó satisfactoriamente y debe verse el nuevo registro en la grilla de resultados

Estado: Aprobado

Evidencia:



The screenshot shows a web application interface for product configuration. At the top, it says 'Configuración General' and 'Usuario: sjaramillom Administrador'. Below this, there are input fields for 'Producto' and 'Ambiente', along with 'Consultar' and 'Limpiar' buttons. A 'RESULTADO' section contains a table with the following data:

Nombre Producto	Nombre Ambiente	Estado	Acciones
winey	desarrollo	✓ Activo	 
winey	pruebas	✓ Activo	 
winey	produccion	✓ Activo	 
noraver	desarrollo	✓ Activo	 
noraver	pruebas	✓ Activo	 

At the bottom right, a green checkmark icon and the message 'Se creó el registro satisfactoriamente' are displayed.

Figura 3.33: Configurador de productos - Caso de prueba 8

ID del Caso de Prueba: TC009


Funcionalidad: Configurador de productos

Nombre del Caso de Prueba: Modificación de productos

Descripción: Este caso de prueba verifica que se permita modificar valores en los productos existentes.

Precondiciones: El usuario debe estar registrado y tener credenciales válidas.

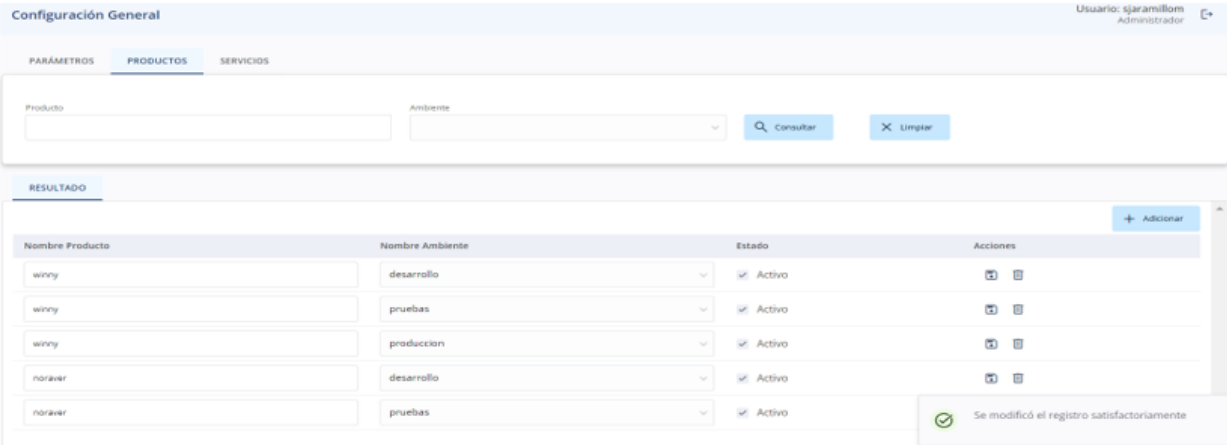
Pasos para Ejecutar la Prueba:











1. Abrir la aplicación.
2. Acceda a la opción Configurador Servicios, y seleccione la pestaña PRODUCTOS
3. Selección en la grilla de resultados el registro a modificar
4. Modifique el valor del campo descripción y valor
5. Presione clic en el botón  (Guardar) en el panel de acciones

Resultado Esperado: Debe mostrarse una alerta notificando que el registro se modificó satisfactoriamente y debe verse el nuevo registro en la grilla de resultados

Estado: Aprobado

Evidencia:



Nombre Producto	Nombre Ambiente	Estado	Acciones
winny	desarrollo	✓ Activo	 
winny	pruebas	✓ Activo	 
winny	produccion	✓ Activo	 
noraver	desarrollo	✓ Activo	 
noraver	pruebas	✓ Activo	 

Se modificó el registro satisfactoriamente

Figura 3.34: Configurador de productos - Caso de prueba 9

ID del Caso de Prueba: TC0010


Funcionalidad: Configurador de productos

Nombre del Caso de Prueba: Eliminación de productos

Descripción: Este caso de prueba verifica que se permita borrar productos existentes.

Precondiciones: El usuario debe estar registrado y tener credenciales válidas.

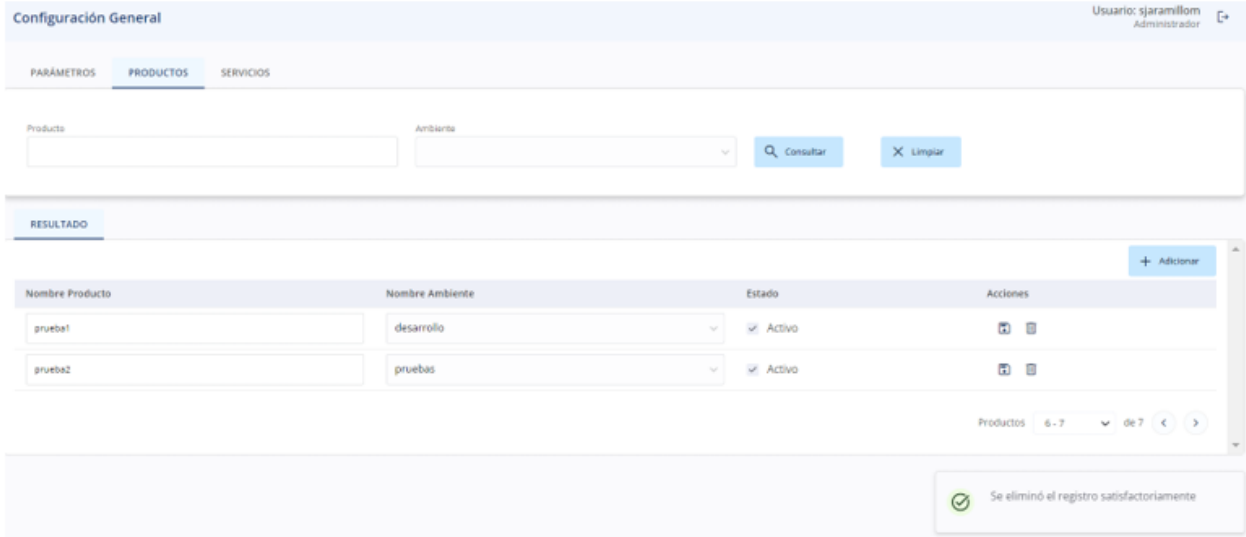
Pasos para Ejecutar la Prueba:

1. Abrir la aplicación.
2. Acceda a la opción Configurador Servicios, y seleccione la pestaña PRODUCTOS
3. Seleccione en la grilla de resultados el registro a eliminar
4. Presione clic en el botón  (Eliminar) en el panel de acciones

Resultado Esperado: Debe mostrarse una alerta notificando que el registro se eliminó satisfactoriamente y no debe verse el registro en la grilla de resultados

Estado: Aprobado

Evidencia:







Configuración General Usuario: sjaramillom
Administrador

PARÁMETROS **PRODUCTOS** SERVICIOS

Producto Ambiente

RESULTADO

Nombre Producto	Nombre Ambiente	Estado	Acciones
prueba1	desarrollo	Activo	 
prueba2	pruebas	Activo	 

Productos 6 - 7 de 7


 Se eliminó el registro satisfactoriamente

Figura 3.35: Configurador de productos - Caso de prueba 10

ID del Caso de Prueba: TC011

Funcionalidad: Configurador de servicios

Nombre del Caso de Prueba: Consulta de servicios

Descripción: Este caso de prueba verifica que se realice la consulta por servicio.

Precondiciones: El usuario debe estar registrado y tener credenciales válidas.

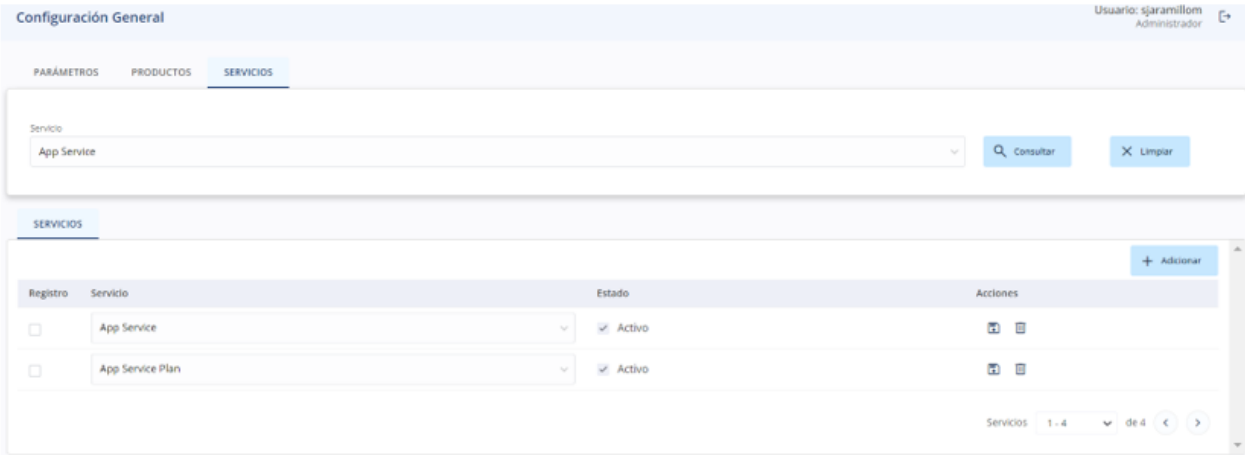
Pasos para Ejecutar la Prueba:

1. Abrir la aplicación.
2. Acceda a la opción Configurador Servicios, y seleccione la pestaña SERVICIOS
3. Seleccione el servicio requerido
4. Presionar clic en el botón Consultar

Resultado Esperado: Debe mostrarse en la grilla inferior los registros que coincidan con el filtro de búsqueda.

Estado: Aprobado

Evidencia:







Registro	Servicio	Estado	Acciones
<input type="checkbox"/>	App Service	Activo	 
<input type="checkbox"/>	App Service Plan	Activo	 

Figura 3.36: Configurador de servicios - Caso de prueba 11

ID del Caso de Prueba: TC012

Funcionalidad: Configurador de servicios

Nombre del Caso de Prueba: Limpiar filtro de consulta

Descripción: Este caso de prueba verifica que se reestablezcan los filtros de la consulta y la grilla de resultados.

Precondiciones: El usuario debe estar registrado y tener credenciales válidas.

Pasos para Ejecutar la Prueba:

1. Abrir la aplicación.
2. Acceda a la opción Configurador Servicios, y seleccione la pestaña SERVICIOS
3. Seleccione el servicio requerido
4. Presionar clic en el botón Consultar
5. Presione el botón Limpiar

Resultado Esperado: Debe mostrarse en la grilla inferior todos los registros y limpiar todos los campos de filtro.

Estado: Aprobado

Evidencia:

Configuración General Usuario: sjaramillom
Administrador

PARÁMETROS PRODUCTOS **SERVICIOS**

Servicio

SERVICIOS

Registro	Servicio	Estado	Acciones
<input type="checkbox"/>	App Service	✓ Activo	<input type="button" value="Editar"/> <input type="button" value="Eliminar"/>
<input type="checkbox"/>	Sql Server	✓ Activo	<input type="button" value="Editar"/> <input type="button" value="Eliminar"/>
<input type="checkbox"/>	Sql Database	✓ Activo	<input type="button" value="Editar"/> <input type="button" value="Eliminar"/>
<input type="checkbox"/>	App Service Plan	✓ Activo	<input type="button" value="Editar"/> <input type="button" value="Eliminar"/>

Figura 3.37: Configurador de servicios - Caso de prueba 12

ID del Caso de Prueba: TC013


Funcionalidad: Configurador de servicios

Nombre del Caso de Prueba: Adición de servicios

Descripción: Este caso de prueba verifica que se permita registrar productos.

Precondiciones: El usuario debe estar registrado y tener credenciales válidas.

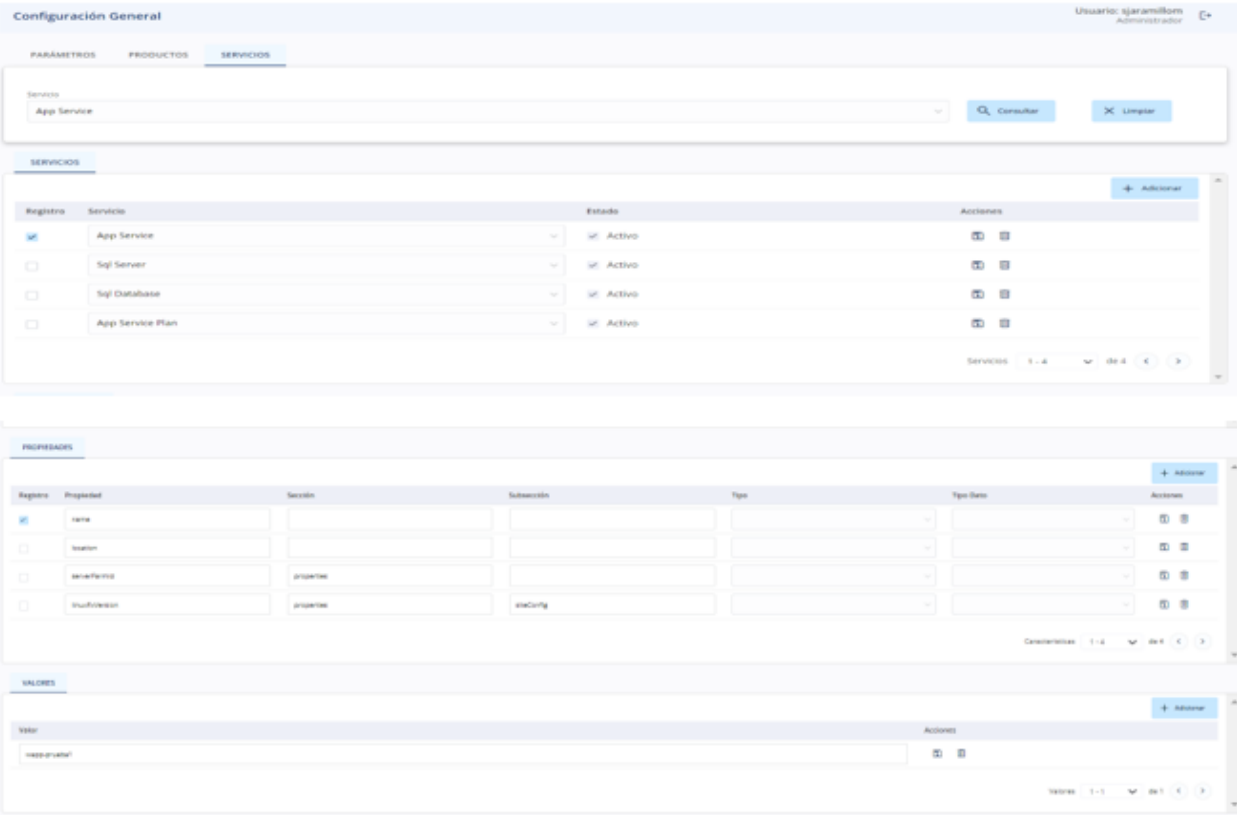
Pasos para Ejecutar la Prueba:

1. Abrir la aplicación.
2. Acceda a la opción Configurador Servicios, y seleccione la pestaña SERVICIOS
3. Presione clic en el botón Adicionar
4. Ingresar en el panel inferior el servicio requerido y el estado
5. Presione clic en el botón  (Guardar) en el panel de acciones

Resultado Esperado: Debe mostrarse una alerta notificando que el registro se creó satisfactoriamente y debe verse el nuevo registro en la grilla de resultados

Estado: Aprobado

Evidencia:



The screenshot displays the 'Configuración General' application interface. At the top right, the user is identified as 'Usuario: sjaramilob Administrator'. The main navigation tabs are 'PARAMETROS', 'PRODUCTOS', and 'SERVICIOS', with 'SERVICIOS' currently selected. Below the tabs, there is a search bar for services with the text 'App Service' entered. To the right of the search bar are buttons for 'Consultar' and 'Limpiar'. Below the search bar is a table of services. The table has columns for 'Registro', 'Servicio', 'Estado', and 'Acciones'. The first row is selected, showing 'App Service' with an 'Activo' status. Below the table is a 'PROPIEDADES' section with a table for adding new services. This table has columns for 'Registro', 'Propiedad', 'Servicio', 'Subservicio', 'Tipo', 'Tipo Base', and 'Acciones'. The first row is selected, showing 'name' as the property. Below the 'PROPIEDADES' section is a 'VALORES' section with a table for adding new values. This table has columns for 'Valor' and 'Acciones'. The first row is selected, showing 'propiedad' as the value.

Figura 3.38: Configurador de servicios - Caso de prueba 13

ID del Caso de Prueba: TC014


Funcionalidad: Configurador de servicios

Nombre del Caso de Prueba: Modificación de servicios

Descripción: Este caso de prueba verifica que se permita modificar valores en los servicios existentes.

Precondiciones: El usuario debe estar registrado y tener credenciales válidas.

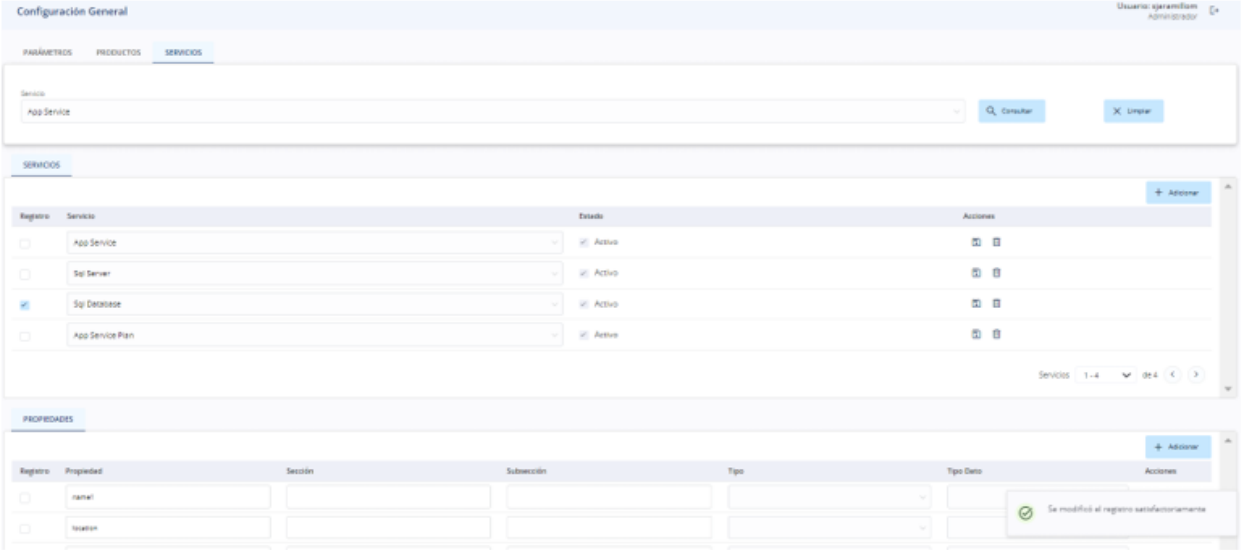
Pasos para Ejecutar la Prueba:

1. Abrir la aplicación.
2. Acceda a la opción Configurador Servicios, y seleccione la pestaña SERVICIOS
3. Selección en la grilla de resultados el registro a modificar
4. Modifique el valor del campo descripción y valor
5. Presione clic en el botón  (Guardar) en el panel de acciones

Resultado Esperado: Debe mostrarse una alerta notificando que el registro se modificó satisfactoriamente y debe verse el nuevo registro en la grilla de resultados

Estado: Aprobado

Evidencia:



Configuración General Usuario: ggeramifam Administrador

PARAMETROS PRODUCTOS **SERVICIOS**

Servicio: App Service Consultar Limpiar

Registro	Servicio	Estado	Acciones
<input type="checkbox"/>	App Service	<input checked="" type="checkbox"/> Activo	
<input type="checkbox"/>	Sip Server	<input checked="" type="checkbox"/> Activo	
<input checked="" type="checkbox"/>	Sip Database	<input checked="" type="checkbox"/> Activo	
<input type="checkbox"/>	App Service Plan	<input checked="" type="checkbox"/> Activo	

SERVICIOS 1-4 024

PROPIEDADES

Registro	Propiedad	Sección	Subsección	Tipo	Tipo Dato	Acciones
<input type="checkbox"/>	name					
<input type="checkbox"/>	location					

Se modificó el registro satisfactoriamente

Figura 3.39: Configurador de servicios - Caso de prueba 14

ID del Caso de Prueba: TC015


Funcionalidad: Configurador de servicios

Nombre del Caso de Prueba: Eliminación de servicios

Descripción: Este caso de prueba verifica que se permita borrar servicios existentes.

Precondiciones: El usuario debe estar registrado y tener credenciales válidas.

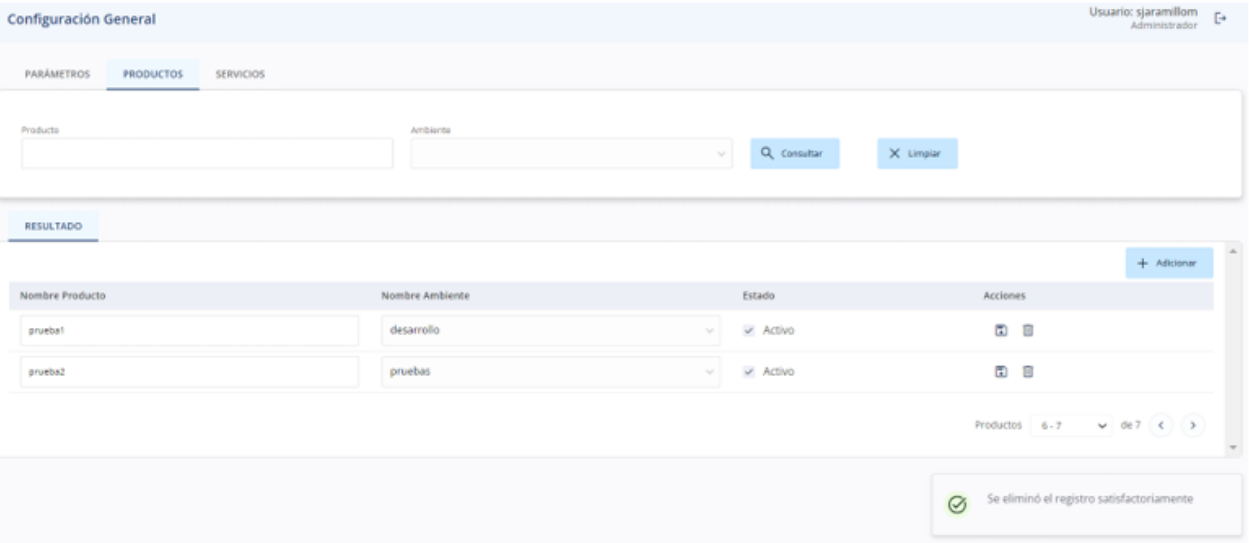
Pasos para Ejecutar la Prueba:

1. Abrir la aplicación.
2. Acceda a la opción Configurador Servicios, y seleccione la pestaña SERVICIOS
3. Seleccione en la grilla de resultados el registro a eliminar
4. Presione clic en el botón  (Eliminar) en el panel de acciones

Resultado Esperado: Debe mostrarse una alerta notificando que el registro se eliminó satisfactoriamente y no debe verse el registro en la grilla de resultados

Estado: Aprobado

Evidencia:







Configuración General Usuario: sjaramillom
Administrador

PARÁMETROS **PRODUCTOS** SERVICIOS

Producto Ambiente

RESULTADO

Nombre Producto	Nombre Ambiente	Estado	Acciones
prueba1	desarrollo	✓ Activo	 
prueba2	pruebas	✓ Activo	 

Productos 6 - 7 de 7


 Se eliminó el registro satisfactoriamente

Figura 3.40: Configurador de servicios - Caso de prueba 15

ID del Caso de Prueba: TC016

Funcionalidad: Generador de servicios

Nombre del Caso de Prueba: Generar [app](#) service con valores parametrizados

Descripción: Este caso de prueba verifica que se realice la generación de la rama con el servicio seleccionado, el PR hacia desarrollo y mediante la ejecución del pipeline, el despliegue del servicio en Azure

Precondiciones: El usuario debe estar registrado y tener credenciales válidas.

Pasos para Ejecutar la Prueba:

1. Abrir la aplicación.
2. Acceda a la opción Generador de Servicios
3. Seleccione el producto y ambiente requerido
4. Seleccione en la pestaña Servicios el tipo de servicio a generar y su nombre
5. En el panel DETALLE DE SERVICIOS se observa la parametrización preestablecida para el servicio
6. Presionar clic en el botón Ejecutar del panel superior

Resultado Esperado: Debe mostrarse una alerta indicando en nombre de la rama generado, crear la rama en el repositorio y el PR hacia desarrollo, una vez ejecutado el pipeline debe desplegar el servicio en Azure en el grupo de servicios preestablecido.

Estado: Aprobado

Figura 3.41: Generador de servicios - Caso de prueba 16.1

Evidencia:

Generador de Servicios Usuario: sjaramilom
Administrador

GENERADOR

Producto: Ambiente:

SERVICIOS

Tipo Servicio: Nombre Servicio:

DETALLE SERVICIOS

as1

Característica	Valor
name	wapp-prueba1
location	EastUS
servicefqn	sp-pruebaipoc
troubleshoot	node14ts

Branch	Co...	Author	Author...	Behind Ahead	Sta...	Pub...
desarrollo	58a73r	Santiago J...	jueves	0 92	🟢	:
main	b5112r	Alexander ...	29 de ...			★
master	0e482r	Santiago J...	5 de ...	1 1		
pruebas	b5112r	Alexander ...	29 de ...	0 0		
winy-desarrollo-20241014-134348	909ec	Santiago J...	8m ago	1 1		📦 27f

+ Create Manage view Delete resource group Refresh Export to CSV Open query Assign tags Move Delete

Essentials

Subscription (move): [Sandbox](#) Deployments: [13 Succeeded](#)

Subscription ID: f1bcfd7e-d732-46f4-bccd-2dd2492c7cdd Location: East US 2

Tags (edit): Ambiente: Pruebas analista: Alexander Rodriguez Ospina tribu: Infraestructura solución: IaC Negocio: Corporativo

Resources Recommendations

Filter for any field... Type equals all Location equals all Add filter

Showing 1 to 2 of 2 records. Show hidden types

Name	Type
sp-pruebaipoc	App Service plan
wapp-prueba1	App Service

Figura 3.42: Generador de servicios - Caso de prueba 16.2

3.5. Resumen del capítulo

En este capítulo se describieron los siguientes aspectos:

- Información de la fase de análisis: paso a paso del proceso de elicitación del proyecto, el detalle de la información recolectada referente a los portales corporativos, servicios utilizados, parametrización y los requerimientos funcionales y no funcionales establecidos.
- Diagramas de diseño de la solución, modelo de la base de datos y los lineamientos definidos por arquitectura.
- Detalles de la implementación: stack tecnológico, el flujo del proceso, de la solución, explicación de cada funcionalidad y del flujo del proceso, de cómo funciona, y las evidencias de las pruebas funcionales realizadas.

Evaluación

En este capítulo se explica el proceso realizado para evaluar la herramienta de IaC desarrollada con el fin de automatizar el aprovisionamiento de los servicios cloud que se utilizan para los portales corporativos de la organización TQ, se incluye el protocolo definido y los resultados obtenidos mediante su aplicación en la empresa TQ.

En este caso, la evaluación correspondió a una de tipo formativa, dado que buscaba evaluar que tan bien se adaptaba la herramienta a las necesidades de la organización, considerando aspectos como la utilidad, reusabilidad y facilidad de uso (Davidson, 2021).

Adicionalmente, se deseaba identificar las oportunidades de mejora que puede tener la herramienta a futuro para potenciar su uso.

4.1. Diseño de la evaluación

4.1.1. Atributos de calidad

Se identificaron tres atributos sobre los cuales era relevante conocer la percepción que los usuarios del instrumento generaban: (i) utilidad, (ii) reusabilidad, (iii) facilidad de uso. Se seleccionaron estos atributos porque se considera son los más relevantes de acuerdo al objetivo primario de la solución propuesta. Estos atributos se describen a continuación:

- Percepción de Utilidad: Grado en el que una persona estima que utilizando un sistema particular incrementa su resultado del trabajo o reduce sus tiempos en ciertos procesos. (Davis, 1989)
- Reusabilidad: Capacidad de la herramienta de ser implementada para otros proyectos incluso por fuera de la organización, requiriendo un mínimo esfuerzo en configuración.
- Percepción de Facilidad de Uso: Grado en el que una persona considera es sencillo utilizar una herramienta particular para cumplir sus funciones. (Davis, 1989)

4.1.2. Diseño de la evaluación

Para realizar la evaluación de la herramienta desarrollada por parte de los interesados, se realizó un proceso en el que se estableció inicialmente qué propósito iba a tener la evaluación, que criterios específicamente iban a evaluarse, cuáles serían los métodos de recolección de la información y qué usuarios debían ser parte de la muestra, también se determinaron las métricas para evitar la subjetividad y ponderar los resultados obtenidos.

El diseño de la evaluación se consignó en un protocolo de evaluación que sirvió de insumo durante la ejecución y análisis de las evaluaciones.

4.1.2.1. Propósito de la evaluación

El propósito de la evaluación es identificar la percepción de utilidad, intensidad de uso y facilidad de uso de la herramienta por parte del usuario.

4.1.2.2. Preguntas a resolver con la evaluación

Las preguntas de investigación se formularon para evaluar el grado de logro de algo, en lugar de generar preguntas de sí o no. Las fuentes de información con las que se obtuvieron los datos por parte del investigador fueron la observación, entrevista y el pensamiento en voz alta por parte de los usuarios sobre el manejo de la herramienta diseñada.

- Pregunta 1: En qué medida considera usted que la herramienta le va a permitir hacer su trabajo de forma más rápida y por qué?
- Pregunta 2: Considera que la herramienta le resulta fácil de utilizar?
- Pregunta 3: Que tan probable es, que usted utilice la herramienta para aprovisionar nuevos servicios y para otros proyectos y por qué?
- Pregunta 4: Que mejoras considera se podrían hacer a la herramienta para versiones futuras?

4.1.2.3. Selección de la muestra

La selección de personas para evaluar la herramienta se realizó tomando como base las siguientes características:

- Tener relación directa o indirecta con el uso de la herramienta y la gestión de servicios cloud corporativos.
- Ser empleado del área de TI de la organización TQ.
- Contar con aproximadamente una hora de tiempo para participar en la evaluación.

4.1.2.4. Unidades de Análisis

Las unidades de análisis seleccionadas para evaluar la herramienta corresponden a tres usuarios con roles distintos del área de TI de la empresa TQ, que tienen relación directa con la gestión de servicios cloud para el despliegue de soluciones corporativas:

- **Arquitectura:** Rol encargado de definir los lineamientos de arquitectura y de diseñar, en conjunto con los líderes de solución y arquitectura, las nuevas soluciones requeridas por la organización.
- **Infraestructura:** Rol encargado de la gestión de la infraestructura de red tanto física como virtual que posee la organización a nivel nacional. Cumple funciones como el aprovisionamiento de infraestructura requerida por cada proyecto, monitoreo y mantenimientos periódicos.
- **Líder Célula:** Rol encargado de la planeación, diseño e implementación de nuevos proyectos donde se requieren nuevas soluciones de software que busquen apoyar algún objetivo de negocio.

4.1.2.5. Métricas Definidas

Las métricas son las unidades de observación que sirven para recolectar información con el fin de contestar las preguntas a resolver con la evaluación. A continuación, se presentan las métricas definidas:

- **Cantidad de aclaraciones requeridas:** Número de aclaraciones solicitadas por los usuarios mientras usan la herramienta de aprovisionamiento de servicios cloud.
- **Porcentaje de tareas realizadas sin ayuda:** Estimado porcentual de tareas asignadas al usuario para realizar en la herramienta, que pudo hacer sin requerir asistencia o aclarar alguna inquietud por parte del investigador.
- **Cantidad de observaciones sobre elementos faltantes:** cantidad de observaciones de los usuarios sobre elementos que consideran hacen falta para facilitar el manejo de la herramienta.

4.1.2.6. Método de Recolección de Datos

Para el proceso de evaluación de la herramienta por parte de los usuarios, se establecieron las siguientes etapas:

1. Se realiza una primera reunión con los usuarios en la que se hace una presentación sobre los objetivos que se buscaban suplir con el desarrollo de la herramienta, el alcance y se muestra la herramienta en funcionamiento explicando el alcance de cada funcionalidad.
2. Se agenda un espacio con cada uno de los usuarios para que usen la herramienta, la evalúen por medio del protocolo y expresen sus observaciones.

3. En la reunión con los usuarios se les explica la dinámica, que consiste en utilizar la herramienta para realizar unas tareas puntuales, expresar sus observaciones en voz alta durante su uso, continuar con el diligenciamiento de la encuesta y expresar sus observaciones. El investigador estará atento a tomar apuntes para evaluar las métricas establecidas.
4. Por último se finaliza la reunión y se agradece a los usuarios por su participación.

4.1.2.7. Resultados Esperados

Los resultados esperados al realizar la evaluación fueron:

- Resultados de la participación de los usuarios al realizar el ejercicio de evaluación de la herramienta.
- Información consolidada del protocolo de evaluación aplicado con los usuarios.
- Recomendaciones a considerar en versiones posteriores de la herramienta.

4.1.3. Encuesta de Evaluación

Se definieron 4 preguntas para conformar el cuestionario que permite evaluar los atributos de calidad definidos en el diseño de la evaluación.

Todas las preguntas se pueden responder tomando como base la siguiente escala de Likert:

- 6. No aplica
- 5. Completamente de acuerdo
- 4. De acuerdo
- 3. Ni en acuerdo ni en desacuerdo
- 2. En desacuerdo
- 1. Totalmente en desacuerdo

En el Anexo B. se encuentra el cuestionario de evaluación de la herramienta IaC aplicado a cada uno de los roles mencionados.

4.1.4. Observaciones generales del instrumento

4.1.4.1. Tipo de Preguntas

Las diferentes opciones de la Escala Likert facilitan a los usuarios la respuesta para que se acomode a su opinión sin requerir asistencia del orientador.

4.1.4.2. Tiempo Invertido para el Ejercicio

El instrumento está integrado por tres componentes que son exposición de la herramienta a los encuestados, el manejo de la herramienta con la asignación de una tarea y la encuesta con observaciones de su uso. El ejercicio tomó aproximadamente 30 minutos.

4.1.5. Limitaciones de la Evaluación

Es importante considerar que la organización TQ establece un acuerdo de confidencialidad, el cual restringe el uso externo y la divulgación de cualquier tipo de contenido creado para la empresa durante el ejercicio de las funciones de sus empleados. Este acuerdo de confidencialidad se mantiene tanto durante la permanencia en la empresa como después de finalizado el vínculo laboral, asegurando así la protección de la información sensible.

Debido a esta política, la principal limitación en el proceso de evaluación es que la herramienta solo puede ser socializada y evaluada por el personal de TQ. Esto genera un sesgo, ya que la evaluación se realiza exclusivamente para verificar que la solución se adapte a las necesidades específicas de la organización.

4.2. Resultados de la Evaluación

En esta sección se describen los resultados obtenidos en el proceso de evaluación de la herramienta y el cruce de las diversas fuentes de información que se tuvieron en cuenta. Se organizó la información para dar respuesta a las preguntas de investigación asociadas con los atributos de calidad definidos y se presentan en una sección independiente las observaciones generales del instrumento.

Se consideraron las fuentes de evaluación cualitativa y cuantitativa como se describe a continuación:

- **Evaluación cualitativa:** De acuerdo con los comentarios generados por parte de los usuarios durante y después del uso del instrumento bajo evaluación, se clasificaron las observaciones generadas en i) evidencia positiva y ii) evidencia negativa. La evidencia positiva corresponde a comentarios a favor del instrumento y aspectos considerados beneficiosos del uso de este. De igual manera, los comentarios que hacían referencia a oportunidades de mejora, aspectos en contra del instrumento o considerados poco beneficiosos se clasificaron como evidencia negativa.
- **Evaluación cuantitativa:** Después de utilizar el instrumento con cada empleado se aplicó una encuesta con las preguntas establecidas para cada atributo de calidad expuesto en la sección 4.1.1 y se promediaron los resultados por cada atributo. No se recibieron comentarios negativos u observaciones respecto al funcionamiento del módulo **Gestión de Infraestructura de Proyectos**, aunque los usuarios sí sugirieron nuevas funcionalidades que se pueden incluir a futuro.

4.2.1. Respuesta a las preguntas de evaluación

- **Pregunta 1:** ¿Considera usted que la herramienta le va a permitir hacer su trabajo de forma más rápida y por qué?

La calificación promedio en esta pregunta fue de 5 en una escala de 1 a 5 siendo 5 la calificación máxima posible, lo cual indica que los encuestados consideran que la herramienta va a permitir agilizar el proceso de despliegue de servicios para los portales corporativos relacionándose específicamente al atributo de utilidad de la herramienta.

Adicionalmente, en las observaciones, los encuestados indican que la interfaz de usuario es sencilla, permite operar de forma rápida y que usando esta herramienta se mejora la calidad de los servicios. Ninguno de los encuestados requiere aclaraciones ni manifestó observaciones consideradas negativas para este atributo de calidad.

- **Pregunta 2:** ¿Considera que la herramienta es fácil de utilizar?

La valoración cuantitativa asociada con este ítem fue de 5 en una escala de 1 a 5, siendo 5 la calificación máxima posible, lo cual indica que la interfaz es intuitiva en su manejo. Los encuestados lograron resolver la tarea asignada (generación y despliegue de servicios, configurar un nuevo producto y ambiente) sin solicitar asistencia, respondiendo esto al criterio de calidad de facilidad de uso.

Adicionalmente, los encuestados anotan que se genera valor al proceso de infraestructura y es fácil identificar donde realizar cada acción.

- Pregunta 3: ¿Es probable que usted utilice la herramienta para aprovisionar nuevos servicios y para otros proyectos y por qué?

La calificación fue de 5 en una escala de 1 a 5, lo cual confirma que en el atributo de calidad de reusabilidad los encuestados considerarían el uso de la herramienta en futuros proyectos para agilizar el proceso de despliegue.

Los encuestados mencionan que la herramienta minimiza el riesgo de errores y que *puede servir para gobernar el proceso de aprovisionamiento de infraestructura y poder tener control de este proceso desde el área de TI.*

- Pregunta 4: ¿Qué mejoras considera podrían hacerse a la herramienta para versiones futuras?

De acuerdo con el ejercicio llevado a cabo por los encuestados, estos consideran importante la implementación de un módulo de informes que permitan ver el detalle de servicios y configuración para cada proyecto y ambiente. También sugieren, incluir una opción que permita importar plantillas ARM (las plantillas de Azure Resource Manager son archivos en formato JSON que permiten definir y desplegar la infraestructura y configuración de proyectos en Azure de manera declarativa) y a partir de estas generar la configuración de un servicio.

4.3. Resumen del Capítulo

En este capítulo se indicó como fue elaborado el instrumento de evaluación de la herramienta y el resultado de su aplicación con los usuarios establecidos en la empresa TQ.

En la primera parte, que comprende el diseño del instrumento de evaluación, se define el propósito, los atributos de calidad, las preguntas a resolver por los encuestados, los mecanismos de evaluación y la muestra que corresponde a los roles de TI que participan del ejercicio.

En el segundo apartado, posterior a la reunión con cada usuario, se consolida la información de las encuestas y observaciones realizadas por los usuarios, producto del ejercicio práctico usando la herramienta. En este se evidencia que el promedio de calificación de los apartados fue de 5 en una escala de 1 a 5 considerando entonces que se tiene una percepción positiva de la herramienta desarrollada y sus posibilidades de aplicación dentro de la organización.

Conclusiones, lecciones aprendidas y trabajos futuros

El presente trabajo buscaba aportar a la organización TQ en el proceso de aprovisionamiento de servicios en la nube mediante el desarrollo de una herramienta que automatizara el proceso, reduciendo el trabajo manual, tiempos de ejecución y generando respaldo de la configuración de cada uno de los servicios para cada proyecto en cada ambiente.

Para lograr este cometido se hizo el análisis del proceso manual que realizan actualmente los usuarios, el levantamiento de requerimientos con el líder del proyecto y el diseño de una solución que pudiera ser reutilizada.

A continuación se describen las conclusiones obtenidas en este proceso, las lecciones aprendidas (Sección 5.2) y los trabajos que a futuro se pueden aplicar en la herramienta para brindar un mayor beneficio a la organización (Sección 5.3).

5.1. Conclusiones

La práctica de la Infraestructura como Código en las organizaciones permite automatizar la gestión y el despliegue de la infraestructura cloud en las empresas, ?. Esto puede resultar beneficioso, especialmente en organizaciones como TQ, donde se manejan diversos ambientes de desarrollo relacionados con páginas web que son la base para el ámbito comercial de las empresas. La parametrización y el despliegue de servicios debe apoyar a la minimización del error humano y la eficacia en costo beneficio.

Para implementar esta práctica es necesario identificar los servicios de infraestructura cloud y las parametrizaciones más utilizadas en los portales de la organización, lo que se convierte en el primer objetivo específico. En este apartado se logra establecer el uso de Azure Bicep con más de 1100 servicios, siendo el de mayor aprovisionamiento el App Service.

El segundo objetivo plantea diseñar la estructura de las plantillas Bicep para considerar la reutilización y la variabilidad requerida en la generación de plantillas de configuración parametrizadas. Para realizar el diseño fue necesario aplicar un proceso de elicitación que permite visualizar las necesidades de la organización con la finalidad de establecer el alcance, los requisitos generales y

las funciones de la herramienta. La estrategia para generar plantillas de configuración de forma variable, fue configurar la estructura base en un parámetro general y el contenido, es decir, sus propiedades y valores, con la configuración realizada de cada servicio en el módulo de **Gestión de Infraestructura de Proyectos**

Para desarrollar el tercer objetivo que consistía en diseñar e implementar una aplicación web para la selección y parametrización de servicios de Azure Cloud, así como la generación de plantillas de configuración en formato Bicep a partir de los parámetros seleccionados por el usuario, se toma la información obtenida de los lineamientos de arquitectura. Este paso se convierte en una fuente fundamental para el desarrollo de este proyecto, puesto que se identifica una línea base de frameworks y lenguajes definidos para la creación del módulo de **Gestión de Infraestructura de Proyectos**.

Adicionalmente, el desarrollo de la solución se realiza en un módulo que se integra con un componente de seguridad corporativo que gestiona y autentica los usuarios, puesto que se maneja información confidencial; esto evidencia la necesidad de la gobernabilidad de la información en grandes organizaciones como TQ. Con este ejercicio se evidencia que la IaC incide en la mitigación, el riesgo y podría apoyar al proceso de CI/CD, brindando mayor gobierno y automatización de la infraestructura, evitando errores y fallas de código.

El cuarto objetivo consiste en evaluar que las plantillas generadas funcionen correctamente, así como la facilidad de uso, la percepción de utilidad e intención de uso de la aplicación web. Para esto, se estableció un cuestionario de evaluación que consideró los atributos de calidad enmarcado en un protocolo. Inicialmente, se expone la herramienta al encuestado, posteriormente se asigna una tarea y finalmente se responden unas preguntas de evaluación. La percepción de utilidad, y facilidad de uso de la herramienta obtuvieron resultados positivos con oportunidades de mejora detallados en el capítulo 4.

Esta investigación permite evidenciar la importancia de la aplicabilidad de la IaC en grandes organizaciones, puesto que demuestra el valor agregado que se puede tener en las áreas de arquitectura de TI, minimizando el desafío de la generación manual de ambientes web. Este trabajo logra los objetivos propuestos al demostrar que el desarrollo de este tipo de herramientas enmarcadas en IaC se puede convertir en una práctica organizacional que crea sistemas confiables, seguros y de costo beneficio.

5.2. Lecciones Aprendidas

La puesta en marcha de este proyecto permitió al investigador experimentar una serie de aprendizajes significativos, asociados a la implementación de prácticas nuevas en las áreas de TI.

Abordar este proceso de manera sistemática, identificando las necesidades de la organización, estableciendo los lineamientos de arquitectura y al mismo tiempo determinar cómo se realizaría el desarrollo de la herramienta ejemplifica de manera real como sería la implementación de proyectos y su proceso de acople.

La IaC puede hacer evidentes desafíos internos de las organizaciones como el gobierno de la información, la curva de aprendizaje de las empresas, ya que se requiere adquirir nuevas habilidades en lenguajes de scripting y repositorios, cambios culturales y técnicos además del manejo de seguridad (acceso restringido de la información) generando evaluaciones propias para determinar el funcionamiento y la gestión de operaciones del área de TI.

La creación de una herramienta basada en IaC en la organización TQ, a pesar de todas las bondades que esto ofrece, no ha sido tarea fácil y ha requerido una gran cantidad de reuniones con diversos roles como arquitectura, infraestructura y líderes de solución para socializar el alcance del proyecto y que se brinde el apoyo y los permisos requeridos para poder hacer las pruebas en un ambiente establecido.

También se identificó durante la evaluación, la intención de los encuestados del uso del instrumento, ya que estos pudieron experimentar de primera mano las funcionalidades de la herramienta calificándola como: “interfaz sencilla permitiendo la ejecución de una tarea rápida”, lo que muestra al investigador los beneficios inmediatos de la aplicación de prácticas como estas.

Finalmente, la aplicación de este proyecto impacta al investigador no solamente desde el ámbito académico sino también desde lo laboral, pues al realizarse el ejercicio en un ambiente real permite ampliar la perspectiva de la aplicación de nuevas prácticas en grandes empresas.



5.3. Trabajos Futuros

Para futuros estudios relacionados, se recomienda, con base en los hallazgos obtenidos durante el desarrollo del proyecto, fortalecer los siguientes aspectos:

- Ampliar el conjunto de servicios disponibles para aprovisionar.
- Implementar una funcionalidad que permita generar informes detallados sobre los servicios disponibles por producto corporativo y ambiente.
- Establecer cuáles parámetros de los servicios, debido a que manejan información sensible, como por ejemplo credenciales de acceso, tomen su valor de variables del library en DevOps durante la ejecución del pipeline. De esta manera, estos valores no serán visibles para los usuarios que accedan al repositorio y visualicen el contenido de las plantillas de configuración.
- Crear configuraciones de servicios y parámetros para nuevos productos y ambientes a partir de configuraciones existentes.
- Incluir un asistente que ayude al usuario en la configuración de los servicios, indicando y restringiendo las propiedades disponibles por cada servicio, indicando para qué se usa cada propiedad y los posibles valores que se pueden indicar.

Aval de Tecnoquímicas para uso de información

RE: Correo aprobación tesis

 Byron German Botina Silva
Para  Santiago Jaramillo Munera

Señores:
UNIVERSIDAD JAVERIANA CALI

Cordial Saludo

Por medio del presente correo se autoriza al colaborador **Santiago Jaramillo Munera** que se desempeña como **Ingeniero de Sistemas de Información**, para incluir la siguiente información de la organización **correspondiente, única y exclusivamente** al proyecto de **Implementación de Infraestructura como Código con Azure Bicep**, en su documento de tesis final.

La información que se podrá incluir en el documento final es la siguiente:

- Backlog e Historias de usuario del proyecto
- Lineamientos de la plataforma ERPTQ y del módulo
- Diagramas de diseño de la solución
- Mockups y pantallas de la aplicación desarrollada
- Encuestas de evaluación realizadas a los empleados

Cualquier duda con gusto será atendida.

Byron Botina Silva
Gerente de Soluciones de Información
Área de TI
Tecnoquímicas S.A.
(57) 602 882 5555 EXT 1456
Cali - Colombia
bbotina@tecnoquimicas.com
www.tecnoquimicas.com

 Responder  Responder a todos  Reenviar  

martes 28/05/2024 5:46 p. m.

Figura A.1: Aval TQ - Uso de Información

Cuestionario de evaluación – Herramienta IAC - Tecnoquímicas

Cuestionario de Evaluación – Herramienta IAC – Tecnoquímicas

Nombre del colaborador: _____
Cargo: _____
Área: _____

1. Tarea asignada: _____
2. ¿Pudo resolver la tarea sin asistencia?

Escalas de evaluación

1. Totalmente en desacuerdo
2. En desacuerdo
3. Ni en acuerdo ni en desacuerdo
4. De acuerdo
5. Completamente de acuerdo
6. No aplica

Evaluación

3. ¿Considera usted que la herramienta le va a permitir hacer su trabajo de forma más rápida, lenta, ¿es igual y por qué?

4. ¿Considera que la herramienta le resulta fácil de utilizar?

5. ¿Es probable es que usted utilice la herramienta para aprovisionar nuevos servicios y para otros proyectos y por qué?

6. Que mejoras considera se podrían hacer a la herramienta para versiones futuras?

Observaciones adicionales

Figura B.1: Cuestionario Evaluación - Herramienta IAC

Bibliografía

- AppService (2023). Información general de app service. <https://learn.microsoft.com/es-es/azure/app-service/overview>.
- AppServicePlan (2023). Información general de app service plan. <https://learn.microsoft.com/en-us/azure/app-service/overview-hosting-plans>.
- Bicep (2024). Bicep language for deploying Azure resources - Azure Resource Manager — learn.microsoft.com. <https://learn.microsoft.com/es-es/azure/azure-resource-manager/bicep/overview?tabs=bicep>.
- Camas Mainato, W. X. and Riera Taza, H. X. (2022). Diseño e implementación de un proveedor de infraestructura en la nube inp, mediante segmentación de recursos. *Diseño e implementación de un proveedor de infraestructura en la nube INP, mediante segmentación de recursos*.
- Carey (2023). What is infrastructure as code? automating your infrastructure builds. <https://www.infoworld.com/article/2259359/what-is-infrastructure-as-code-automating-your-infrastructure-builds.htm>.
- Clements, L. B. P. and Kazman, R. (2022). *Software Architecture in Practice*. Addison-Wesley, 4th edition.
- Cois, C. A., Yankel, J., and Connell, A. (2014). Modern devops: Optimizing software development through effective system interactions. In *2014 IEEE international professional communication conference (IPCC)*, pages 1–7. IEEE.
- Davidson, E. (2021). Evaluation methodology basics: The nuts and bolts of sound evaluation. *Evaluation Methodology Basics: The Nuts and Bolts of Sound Evaluation*.
- Davis, F. D. (1989). Perceived usefulness, perceived ease of use, and user acceptance of information technology. -.
- Devops, M. (2023). Qué es la infraestructura como código (iac). <https://learn.microsoft.com/es-es/devops/deliver/what-is-infrastructure-as-code>.
- Dominguez-Quintero, L. and Vargas-Lombardo, M. (2021). Herramientas de infraestructura como código: ansible, terraform, chef, puppet. *I+ D Tecnológico*, 17(2):25–29.
- Hasan MM, B. F. (2020). Testing practices for infrastructure as code. proceedings of the 1st acm sigsoft international workshop on languages and tools for next-generation testing. -.
- IBM (2024). Public cloud use cases. <https://www.ibm.com/mx-es/think/topics/public-cloud-use-case>.

- Kief (2020). Infrastructure as code. <https://www.oreilly.com/library/view/infrastructure-as-code/9781098114664/>.
- Koskelainen, E. (2021). Creating azure devops pipelines for web application. *Creating azure devops pipelines for web application*.
- M. Guerriero, M. Garriga, D. A. T. and Palomba, F. (2019). Adoption, support, and challenges of infrastructure-as-code: Insights from industry,”2019 IEEE International Conference on Software Maintenance and Evolution (ICSME), Cleveland, OH, USA, 2019, pp. 580-589, doi: 10.1109/icsme.2019.00092”. -.
- Michalowski (2024). Benefits and best practices for infrastructure as code. <https://devops.com/benefits-and-best-practices-for-infrastructure-as-code>.
- Microsoft (2024). Qué es Azure: Servicios en la nube de Microsoft — Microsoft Azure — azure.microsoft.com. <https://learn.microsoft.com/en-us/azure/azure-portal/>.
- Moreno Martínez, J. (2022). Ci/cd en infraestructura como código (iac). caso práctico en Amazon Web Services (AWS). *CI/CD en Infraestructura como código (IaC). Caso práctico en Amazon Web Services (AWS)*.
- Morris, K. (2020). *Infrastructure as code*. O’Reilly Media.
- Palos Sanchez, P. (2019). Modelos de adopción de tecnologías de la información y cloud computing en las organizaciones. información tecnológica. <https://dx.doi.org/10.4067/S0718-07642019000300003>.
- Patiño Vanegas, J. (2019). Modelo para la adopción de cloud computing en las pequeñas y medianas empresas del sector servicios en Medellín. <https://dx.doi.org/10.4067/S0718-07642019000600157>.
- Patni JC, B. S. (2020). Infrastructure as a code (iac) to software defined infrastructure using azure resource manager (arm). -.
- Pressman, R. S. and Maxim, B. R. (2022). *Software Engineering: A Practitioner’s Approach*. McGraw-Hill Education, 9th edition.
- Pulumi (2024). What is Pulumi? Key Concepts and Features Overview — spacelift.io. https://www.pulumi.com/why-pulumi/?utm_source=google&utm_medium=cpc&utm_campaign=Competitive-IaC&utm_term=pulumi&utm_medium=cpc&utm_source=google&hsa_grp=164192522628&hsa_cam=20775470768&hsa_mt=e&hsa_net=adwords&hsa_ver=3&hsa_acc=1926559913&hsa_ad=680872926398&hsa_src=g&hsa_tgt=kwd-1177947170384&hsa_kw=pulumi&gad_source=1&gclid=CjwKCAjwvIWzBhAlEiwAHHWgvQzj7aGdzJDJXiK82h4rc8jCj011ZL1H2A7D-YRXWowjh60o3V60KxoCpTQQAvD_BwE.

- Puppet (2024). What is puppet. <https://www.puppet.com/blog/what-is-infrastructure-as-code>.
- Rodriguez Quevedo, J. (2022). Infraestructura tecnológica para la transformación digital en la universidad de la habana. <https://revistas.uh.cu/rces/article/view/189>.
- Ronaldo, H. Q. J. (2021). Terraform como herramienta para automatizar la creación de infraestructuras siguiendo el concepto “infraestructura como código”. *Terraform Como Herramienta Para Automatizar La Creación De Infraestructuras Siguiendo El Concepto “Infraestructura Como Código”*.
- SQLDatabase (2024). Información general de sql database. <https://learn.microsoft.com/es-es/azure/azure-sql/database/sql-database-paas-overview?view=azuresql>.
- SQLServer (2024). Información general de sql server. <https://learn.microsoft.com/es-es/sql/sql-server/what-is-sql-server?view=sql-server-ver16>.
- Stallings, W. (2022). *Computer Security: Principles and Practice*. Pearson, 5th edition.
- Tanenbaum, A. S. and Bos, H. (2022). *Modern Operating Systems*. Pearson, 4th edition.
- Tecnoquimicas (2024). Quienes somos.
- Terraform (2024). What is Terraform — Terraform — HashiCorp Developer — developer.hashicorp.com. <https://developer.hashicorp.com/terraform/intro>.
- Turim (2023). The pros and cons of iac - what you need to know. <https://thenewstack.io/the-pros-and-cons-of-iac-what-you-need-to-know>.