

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA EN SISTEMAS

**DISEÑO DE EXPERIENCIA DE USUARIO DE UN EDITOR GRÁFICO
WEB PARA LA GESTIÓN Y PREPARACIÓN DE INFRAESTRUCTURA**

**DISEÑO DE LA INTERFAZ DE USUARIO DE UN EDITOR GRÁFICO
WEB PARA LA PREPARACIÓN DE INFRAESTRUCTURA. CASO DE
USO: BALANCEO DE CARGA DE APLICACIONES**

**TRABAJO DE INTEGRACIÓN CURRICULAR PRESENTADO COMO
REQUISITO PARA LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN
SOFTWARE**

Diego Alejandro Márquez Coronel

diegomarquez2008@outlook.com

DIRECTOR: Carlos Iñiguez Ph.D.

carlos.iniguez@epn.edu.ec

DMQ, agosto 2022

|

CERTIFICACIONES

Yo, Diego Alejandro Márquez Coronel declaro que el trabajo de integración curricular aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.



DIEGO ALEJANDRO MÁRQUEZ CORONEL

Certifico que el presente trabajo de integración curricular fue desarrollado por Diego Alejandro Márquez, bajo mi supervisión.



CARLOS IÑIGUEZ Ph.D.
DIRECTOR DEL TRABAJO DE INTEGRACIÓN CURRICULAR

Certificamos que revisamos el presente trabajo de integración curricular.

NOMBRE_REVISOR1
REVISOR1 DEL TRABAJO DE INTE-
GRACIÓN CURRICULAR

NOMBRE_REVISOR2
REVISOR2 DEL TRABAJO DE INTE-
GRACIÓN CURRICULAR

DECLARACIÓN DE AUTORÍA

A través de la presente declaración, afirmamos que el trabajo de integración curricular aquí descrito, así como el (los) producto(s) resultante(s) del mismo, son públicos y estarán a disposición de la comunidad a través del repositorio institucional de la Escuela Politécnica Nacional; sin embargo, la titularidad de los derechos patrimoniales nos corresponde a los autores que hemos contribuido en el desarrollo del presente trabajo; observando para el efecto las disposiciones establecidas por el órgano competente en propiedad intelectual, la normativa interna y demás normas.



DIEGO ALEJANDRO MÁRQUEZ CORONEL



CARLOS IÑIGUEZ Ph.D.
DIRECTOR DEL TRABAJO DE INTEGRACIÓN CURRICULAR

AGRADECIMIENTO

Agradezco a mis padres, quienes me han dado su apoyo y aliento en cada paso durante mi carrera y han sido guías y guardianes en mi formación profesional y académica.

A mi hermana, que me ha brindado sus consejos y palabras de ánimo durante mi paso por la universidad, animándome cada vez que lo necesitaba.

A mi tutor Ph. D. Carlos Iñiguez, por la eficaz asesoría, acompañamiento, correcciones y orientación durante todo el proyecto que me permitió desarrollar un mejor proyecto.

A mi novia y compañera Nathy, con su valiosa actitud y sus innumerables palabras de ánimo, esfuerzo, amor y confianza, me incentivo a concluir este proyecto.

ÍNDICE DE CONTENIDO

RESUMEN	IX
ABSTRACT	X
1. INTRODUCCIÓN.....	1
1.1. Definición del problema	1
1.2. Objetivo general.....	2
1.3. Objetivos específicos	2
1.4. Alcance.....	2
1.5. Marco teórico	2
1.5.1. Aprovisionamiento de Infraestructura.....	2
1.5.2. ARGÓN.....	3
1.5.3. Balanceo de Carga	4
1.5.4. Diseño Centrado en el Usuario	7
1.5.5. Experiencia de Usuario (UX).....	8
1.5.6. Diseño de Interfaz de Usuario.....	11
1.5.7. Adobe XD.....	11
2. METODOLOGÍA.....	12
2.1. Análisis de Contexto	12
2.2. Definición de Requisitos	13
2.2.1. Perfil Persona.....	13
2.2.2. Matriz OTI	14
2.2.3. Mapa Navegacional	15
2.3. Diseño.....	16
2.3.1. Bocetos	16
2.3.2. Wireframe.....	16
2.3.3. Diseño Visual	18

2.3.4. Prototipo interactivo	19
2.4. Evaluación	20
2.4.1. Objetivo	20
2.4.2. Variables a evaluar	20
2.4.3. Objetivos y Tareas para Pruebas de Usabilidad	20
2.4.4. Diseño a evaluar	20
2.4.5. Participantes	21
2.4.6. Protocolo de Evaluación	21
2.4.7. Ejecución	21
3. RESULTADOS, CONCLUSIONES Y RECOMENDACIONES	22
3.1. Resultados	22
3.2. Conclusiones	25
3.3. Recomendaciones	26
4. REFERENCIAS BIBLIOGRÁFICAS	26
5. ANEXOS	29
ANEXO I: Enlace de ejecución de evaluaciones de prototipos de alta fidelidad	29
ANEXO II: Encuesta: Modelamiento de Infraestructura en la Nube	29
ANEXO III: Prototipo Interactivo	36

ÍNDICE DE FIGURAS

Figura 1. Interfaz de Usuario de la herramienta ARGON [7].....	4
Figura 2. Fases del Diseño Centrado en el Usuario. Adaptado de [16]	8
Figura 3. Representación de UX. [21]	9
Figura 4. Interfaz de Usuario de Adobe XD	12
Figura 5. Plantilla Persona “Mari”. Fuente: Elaboración propia	13
Figura 6. Plantilla Persona “Edison”. Fuente: Elaboración propia.....	14
Figura 7 Mapa navegacional. Fuente: Elaboración propia.....	15
Figura 8. Boceto de la interfaz de usuario propuesta. Fuente: Elaboración propia	16
Figura 9. Wireframe de la interfaz de usuario propuesta. Fuente: Elaboración propia.....	17
Figura 10. a) Isotipo b) Logotipo. Fuente: Elaboración propia	18
Figura 11. a) Paleta de colores mediante separación complementaria b) Paleta de colores monocromáticos. Fuente: Adobe Color	18
Figura 12. Tipografía Lemon Milk Regular.....	19
Figura 13. Ejemplo del prototipo interactivo elaborado en AdobeXD.	20
Figura 14. Ejecución de la prueba con el usuario a través de Zoom.....	22
Figura 15. Satisfacción (puntaje SUS). Fuente: Elaboración propia	23
Figura 16. Elementos para la implementación de una infraestructura	24
Figura 17. Botones para generar un script y levantar un servidor en la nube	24
Figura 18. Lista desplegable con las diferentes regiones disponibles.....	25

ÍNDICE DE TABLAS

Tabla 1. Elementos de modelamiento. Fuente: Elaboración propia	4
Tabla 2. Sub-características y métricas del Modelo de Usabilidad.....	9
Tabla 3. Matriz OTI. Fuente: Elaboración propia.....	14
Tabla 4. Respuestas del cuestionario SUS. Fuente: Elaboración propia.....	22
Tabla 5. Resultado final SUS. Fuente: Elaboración propia	23
Tabla 6. Promedio final en decimales de cada Perfil Persona. Fuente: Elaboración propia	23

RESUMEN

El aprovisionamiento de infraestructura en la nube es una importante solución tecnológica. Esta conlleva a ciertas tareas complejas, entre las cuales se destaca el balanceo de carga. Mediante la filosofía de IaC (Infraestructura como código), la herramienta ARGON trata de simplificar estas tareas, sin embargo, aún existe dificultad para comprender este tipo de herramienta. Este proyecto propone un diseño de interfaz de usuario interactivo aplicando el enfoque de Diseño Centrado en el Usuario (DCU, Diseño Centrado en el Usuario), con el objetivo de lograr una buena experiencia de usuario (UX), enfocándonos en una de las tres variables de la Usabilidad definidas por la ISO 9241-11:2018: la satisfacción del usuario. El diseño propuesto fue evaluado usando como métrica la Escala de Usabilidad de un Sistema (SUS). Los resultados obtenidos indican que el diseño propuesto logra un alto nivel de satisfacción entre los usuarios.

PALABRAS CLAVE: balanceo de carga, ARGON, Diseño Centrado en el Usuario (DCU), Experiencia del Usuario (UX), satisfacción, Escala de Usabilidad de un Sistema (SUS).

ABSTRACT

Cloud infrastructure provisioning is an essential IT solution. This solution involves certain complex tasks, including load balancing. Through the philosophy of IaC (Infrastructure as Code), the ARGON tool tries to simplify these tasks, however, there is still difficulty in understanding this type of tools. This project proposes an interactive user interface design applying the User-Centered Design (UCD) with the goal of achieving a good User Experience (UX), focusing on one of the three Usability Variables defined by ISO 9241-11:2018: user satisfaction. The proposed design was evaluated using the System Usability Scale (SUS) as a metric. The obtained results indicated that the suggested design achieves a high level of user satisfaction.

KEYWORDS: load balancing, ARGON, User-Centered Design (UCD), User Experience (UX), satisfaction, System Usability Scale (SUS).

1. INTRODUCCIÓN

1.1. Definición del problema

La saturación de información en el servidor es uno de los principales motivos por los cuales un servidor de Internet se tarda en responder [1]. Este problema de lentitud es frecuentemente solucionado ampliando la capacidad del disco duro, memoria o actualizando el procesador. Sin embargo, en la actualidad ampliar las capacidades de hardware de los servidores solo es una solución temporal por el crecimiento acelerado del Internet. Por este motivo una opción más eficaz para evitar la saturación de información es balancear la carga del servidor entre varios servidores, considerando el número de peticiones de los clientes. El balanceo de carga es la forma en la que las peticiones de Internet se distribuyen en una fila de servidores [2].

Amazon y Azure son dos proveedores de infraestructura en la nube actuales y ofrecen diferentes formas de *aprovisionar infraestructura* (modelar, agregar o quitar recursos de infraestructura), incluyendo estrategias para balancear la carga dependiendo de ciertos parámetros. Por ejemplo, tenemos a AWS (Amazon Web Services) que ofrece una infraestructura con tres diferentes tipos de balanceadores de carga: de aplicaciones, red, y clásico [3]. El aprovisionamiento de infraestructura es complejo debido a que implica escribir extensas líneas de código, cuya sintaxis y comandos difiere dependiendo del proveedor de infraestructura. Por ejemplo, las líneas de código para configurar la infraestructura de AWS son diferentes a las líneas de código utilizadas para configurar la misma infraestructura en Azure. Para reducir la complejidad en la gestión, un enfoque utilizado es la *infraestructura como código* (IaC) [4]; que permite automatizar el aprovisionamiento de infraestructura y que está basado en prácticas del desarrollo de software. Basado en el enfoque IaC se desarrolló ARGÓN [4], una herramienta para el modelado de infraestructura. ARGON permite definir los recursos de infraestructura de la nube a través de un lenguaje de modelado específico de dominio (DSL) con lo cual es posible abstraer la complejidad de utilizar lenguajes de scripting para especificar recursos de infraestructura para diferentes proveedores de la nube [5].

A pesar de las amplias funciones que tiene ARGÓN para aprovisionar infraestructura, su interfaz gráfica presenta una carga cognitiva alta para los operadores de infraestructura que son nuevos en el sistema. A finales de la década de 1980 empezó el estudio de la carga cognitiva en medios informáticos y la forma en la que afecta al aprendizaje. La Teoría de la Carga Cognitiva (TCC) llega con la premisa de que un usuario principiante tiene una limitada memoria de trabajo al momento que debe enfrentarse con nueva información [6]. Como, por ejemplo, la gran cantidad de opciones que nos ofrece ARGÓN en su interfaz podría generar al usuario una saturación de información lo cual afecta considerablemente en el aprendizaje y experiencia de uso de esta.

Considerando esta teoría, en la investigación de Sandobalin et al. [7] se muestra la interfaz de usuario de ARGON a través de la cual se explica, como un caso de uso, el aprovisionamiento

de un balanceador de carga en la nube.

Este proyecto está orientado en diseñar un prototipo de la interfaz de usuario para el editor gráfico de ARGÓN con el fin de reducir la carga cognitiva de los usuarios al interactuar con la herramienta y por consiguiente lograr un aprendizaje adecuado de la herramienta.

1.2. Objetivo general

El objetivo de este proyecto consiste en diseñar la interfaz de usuario de un editor gráfico Web para la herramienta ARGÓN que facilite a los operadores de infraestructura balancear la carga en el contexto del aprovisionamiento de infraestructura.

1.3. Objetivos específicos

- Identificar usuarios potenciales, sus objetivos y requisitos de usabilidad considerando el caso de uso balanceo de carga.
- Diseñar el prototipo de interfaz de usuario de un editor gráfico Web.
- Evaluar la usabilidad del prototipo de interfaz de usuario del editor gráfico Web diseñado.

1.4. Alcance

El alcance del componente implica diseñar el prototipo de interfaz de usuario de un editor gráfico Web para el “Balanceo de Carga”. El diseño del prototipo estará guiado por la metodología Diseño Centrado en el Usuario (DCU) que consiste en cuatro fases: análisis, diseño, prototipado y evaluación.

1.5. Marco teórico

1.5.1. Aprovisionamiento de Infraestructura

El aprovisionamiento de infraestructura es el manejo de un elemento de la infraestructura tal como un dispositivo de comunicación o un servidor de manera que esté disponible para su uso [8]. Específicamente se trata del proceso en el cual se realiza la creación, actualización, configuración, o eliminación de uno o varios elementos de la infraestructura [9]. Este proceso de aprovisionamiento se refiere al cambio de cualquier elemento de la infraestructura y requiere que el equipo de desarrollo y de operaciones mantenga una constante comunicación y coordinación.

Con el objetivo de orquestar este tipo de procesos aparece DevOps (*Development and Operations*), un enfoque de desarrollo de software que promueve la activa cooperación entre el equipo de desarrollo y el de operaciones con la ayuda de fundamentos, buenas prácticas y herramientas que agilicen este cambio constante a favor del tiempo de entrega de los productos de software [10]. DevOps busca mejorar el proceso de despliegue del software que es con frecuencia una fuente de problemas y atrae demasiada atención de los clientes o gerentes cuando se atrasan las fechas de entrega o un defecto crítico es detectado en fase de producción

[11].

La investigación en el campo de DevOps se ha enfocado en la nube. La comunidad DevOps ha desarrollado varias herramientas para la gestión de la configuración del *aprovisionamiento de infraestructuras* como Ansible, Puppet, Chef y muchas otras [7]. Parte de estas investigaciones proponen automatizar el proceso del aprovisionamiento de infraestructura para acelerar el despliegue en el ciclo de entrega de software mediante un filosofía llamada Infraestructura como Código (IaC) que tiene como objetivo realizar el proceso de creación, actualización y ejecución de una infraestructura; todo mediante scripts y automatizar el aprovisionamiento de la infraestructura mediante prácticas de desarrollo del software enfocadas en el uso de rutinas estables y repetibles [8].

La computación en la nube se ha convertido en la principal forma para obtener una infraestructura en poco tiempo. El uso de la computación en la nube y el IaC genera grandes cambios en la industria de las Tecnologías de Información y promueve a que muchas compañías trasladen la gestión de su información en centros de datos a gestionar su información en la nube. El manejo de infraestructuras mediante hardware se convirtió en un trabajo directo con software, lo cual volvió al equipo de DevOps más eficiente en este trabajo de gestionar la infraestructura [12].

Una herramienta basada en la filosofía IaC tiene su propio lenguaje de script para modelar, desarrollar e implementar una infraestructura. Además, un proveedor de la nube, como Amazon o Microsoft, ofrece la posibilidad de desarrollar infraestructuras de la manera tradicional o en la nube [13]. En consecuencia, la creación de una infraestructura implica escribir extensas líneas de código y una implementación diferente para cada uno de los proveedores de que existe actualmente en el mercado, volviendo a este proceso demandante en tiempo y propenso a errores [4].

1.5.2. ARGÓN

ARGÓN [5] es una herramienta diseñada para el modelamiento de aprovisionamiento de infraestructura en la nube basado en la ingeniería basada en modelos y con soporte del enfoque IaC. La herramienta actualmente está disponible para su uso mediante el IDE de desarrollo Eclipse de modo que es dependiente de los recursos en el computador donde está instalado el IDE. Además, es necesario considerar que el proceso de instalación de la herramienta dependerá de la versión de Eclipse, ya que ARGÓN funciona en Eclipse versión 4.8.

La interfaz de usuario (IU) de ARGÓN está integrada al IDE. La IU (Figura 1) consiste en varias secciones: A) despliegue de diagramas B) paleta, que contiene las opciones para definir colores, barra de opciones con un puntero, opciones de zoom y notas, C) pestaña de propiedades y

problemas.

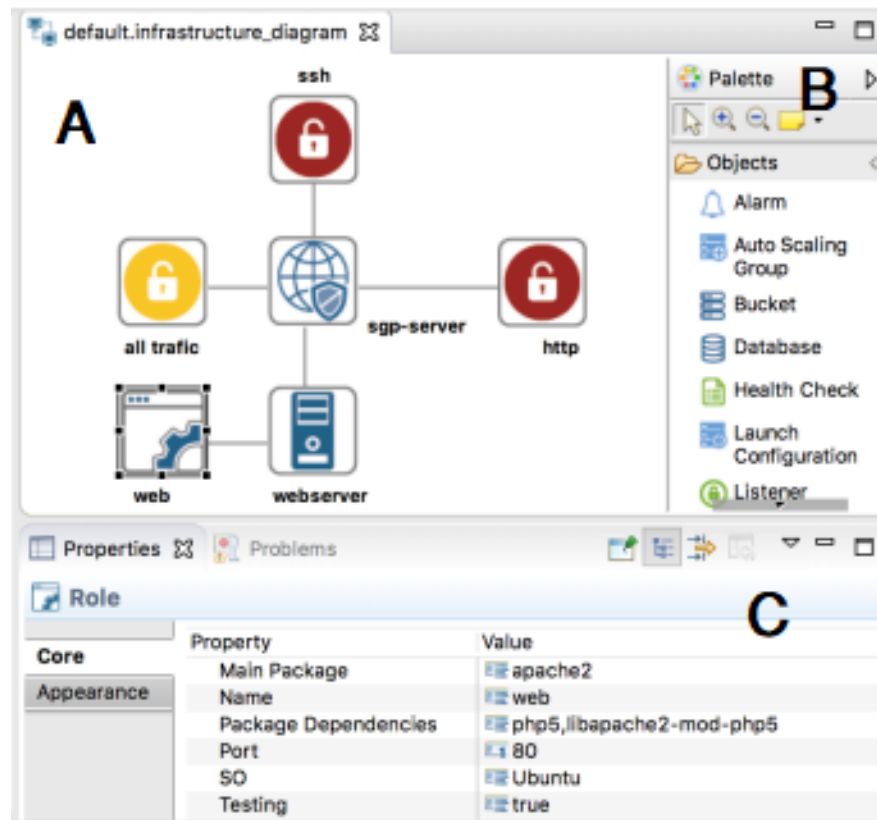


Figura 1. Interfaz de Usuario de la herramienta ARGON [7]

1.5.3. Balanceo de Carga

Balanceo de carga se refiere a la repartición de la cantidad de tareas que un computador debe ejecutar, entre dos o más computadores similares, de tal forma que se realice la mayor cantidad de tareas simultáneamente, de manera que los usuarios obtengan una mayor velocidad de respuesta. El balanceo de carga tiene como objetivo el optimizar el uso de recursos, maximizar el rendimiento del sistema, incrementar el tiempo de respuesta y prevenir la sobrecarga en el procesamiento de un recurso [14]. ARGON, como herramienta de aprovisionamiento en la nube, permite modelar el balanceo de carga, resultando en una distribución de la carga de trabajo entre varias máquinas virtuales [9].

La Tabla 1 muestra cada uno de los elementos que se utilizan en ARGON [4], en específico los requeridos para aprovisionar una infraestructura en el caso de uso del balanceo de carga.

Tabla 1. Elementos de modelamiento. Fuente: Elaboración propia

Elemento	Propiedades	Ejemplo
Modelo de infraestructura	Nombre del archivo	loadBalancer
	Nombre clave	kp-virginia
	Región	us-east-1

Balanceador de carga	Grupos	Security Group sgpLoadBalancer
	Maquinas	Virtual Machine virtualMachine
	Nombre	loadBalancer
Máquina virtual	Conteo	3
	Grupo	Security Group sgpLoadBalancer
	Imagen	ami- 0d7cBdde348d3b09f
	Tipo de instancia	t2-micro
	Nombre	virtualMachine
Listener	Puerto de Instancia	4545
	Puerto del balanceador de carga	80
	Nombre	listener
	Protocolo	TCP
Zona	Nombre	eu-west-1a
HealthCheck	Límite seguro	10 segundos
	Intervalo	30 segundos
	Nombre	healthCheck
	Ruta de ping	/index.html
	Puerto de ping	4545
	Protocolo de ping	TCP
	Tiempo de respuesta	5 segundos
	Limite inseguro	2 segundos
Puerto de entrada	Cidr Ip	0.0.0.0/0
	Del puerto	80
	Nombre del grupo	
	Nombre	Port_80
	Protocolo	TCP

	Hacia el puerto	80
Puerto de salida	Cidr ip	0.0.0.0/0
	Del puerto	0
	Nombre del grupo	
	Nombre	Port_ALL
	Protocolo	ALL
	Hacia el puerto	0

A continuación, se detalla el proceso del balanceo de carga para el aprovisionamiento de una infraestructura mediante un ejemplo ilustrativo utilizando la problemática de una empresa ficticia llamada BalancedEPN (ejemplo adaptado de [4]).

BalancedEPN es una empresa de tecnología especializados en el desarrollo de aplicaciones móviles. El equipo de Devops de BalancedEPN desplegaba sus aplicaciones durante mucho tiempo en Linux Ubuntu 20.04 LTS en Azure Services, pero esto ha generado preocupaciones por motivos de seguridad y problemas con el desempeño de las aplicaciones. Por este motivo se ha decidido en desplegar las aplicaciones en un proveedor de la nube diferente. En este caso, se eligió Amazon Web Service, como plataforma donde se desplegarán las aplicaciones de ahora en adelante. En este sentido, los requerimientos a cumplir son los siguientes:

- Requerimiento 1: La infraestructura debe desplegarse en la región de EE. UU. Este de Amazon Web Services.
- Requerimiento 2: El balanceador de carga deberá distribuir el volumen de trabajo entre tres máquinas virtuales (Instancias EC2). Las maquinas deberán contar con un mínimo de 1 CPU virtual (Intel Xeon) y 1 GB de RAM. Además de venir instalado el sistema operativo Ubuntu 20.04 LTS.
- Requerimiento 3: El balanceador de carga deberá revisar el estado de cada máquina virtual mediante el protocolo TCP y el puerto 4545 utilizando intervalos de 30 segundos, y para notificar en caso de un error deberá esperar hasta 5 segundos. El balanceador de carga deberá recibir hasta dos errores consecutivos para cambiar el estado de una máquina virtual a “inseguro”. En cambio, antes de cambiar el estado de una máquina virtual a un estado “seguro”, se deberá cumplir con 10 verificaciones comprobando su correcto funcionamiento.
- Requerimiento 4: El balanceador de carga utiliza el protocolo TPC para resolver las solicitudes de los clientes a través del puerto 80 y distribuir la carga a través del puerto 4545.
- Requerimiento 5: El balanceador de carga y la máquina virtual deberán trabajar en la

primera zona disponible de la región Este de EE. UU. en Amazon Web Services.

- **Requerimiento 6:** Un grupo de seguridad deberá habilitarse para el puerto de entrada con las conexiones TCP hacia el balanceador de carga solamente a través del puerto 80. De igual manera, el grupo de seguridad deberá habilitar todas las conexiones de puertos abiertos del balanceador de carga
- **Requerimiento 7:** Un grupo de seguridad deberá habilitar las conexiones de entrada TCP hacia las máquinas virtuales a través del puerto 4545, y el puerto 22. Además, el grupo de seguridad deberá habilitar todas las conexiones de salida de las máquinas virtuales.

La solución a estos requerimientos está reflejada en la Tabla 1. En la columna de ejemplo, cada elemento es configurado en base a los requerimientos solicitados. El modelo de infraestructura tiene un campo de región en el cual se coloca el nombre asignado por Amazon Web Services, en el caso del ejemplo, us-east-1 y de nombre clave kp-virginia. El balanceador de carga se encarga de orquestar y verificar el estado y la cantidad de procesamiento de cada una de las instancias de las máquinas virtuales mediante un HealthCheck donde determinamos el intervalo de tiempo que realiza este chequeo, las condiciones para cambiar el estado de una maquina virtual, protocolos y puertos que se habilitaran. Una vez modelada y verificada la infraestructura es necesario generar scripts que en código representan las configuraciones de cada elemento de infraestructura [4].

1.5.4. Diseño Centrado en el Usuario

El Diseño Centrado en el Usuario (DCU) es una metodología de diseño de productos interactivos en la cual se busca obtener una óptima experiencia del usuario [15]. Con base en DCU, el desarrollo del presente proyecto involucrará la identificación y estudio de los problemas de interacción usuario-interfaz en el uso “Balanceo de Carga” y proponer soluciones de diseño a dichos problemas. La norma ISO 9241-210:2019 [16] define cuatro fases del proceso de diseño en el DCU (Figura 2).

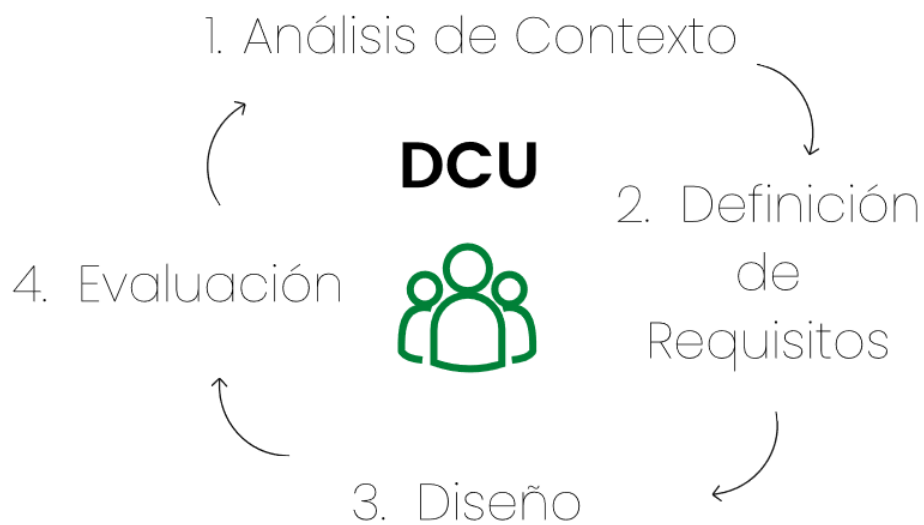


Figura 2. Fases del Diseño Centrado en el Usuario. Adaptado de [16]

Aplicadas al presente proyecto, las cuatro fases se definen de la siguiente manera:

1. Análisis de Contexto: Esta fase plantea comprender el contexto mediante técnicas como el briefing o benchmarking que consiste en la recolección de información relevante acerca de las funcionalidades, competencia y objetivos del producto.
2. Especificación de requisitos: En esta fase se identifica y especifica los requisitos del usuario, con el conocimiento de la fase anterior. Por ejemplo, se puede usar el perfil Persona y los mapas navegacionales para especificar y consolidar la información obtenida.
3. Diseño: En esta fase se busca soluciones de diseño en base a la especificación de requisitos de la fase anterior. Estas soluciones se presentan prototipos de baja y alta fidelidad para satisfacer las necesidades del usuario.
4. Evaluación: En esta fase el usuario interactúa con la propuesta de diseño con el fin de evaluar si la propuesta cumple con los requisitos del usuario.

Este proceso permite comprobar hasta qué punto se ajusta la solución diseñada a las necesidades de los usuarios. Si no se ajusta, se puede refinar la solución en consiguientes iteraciones del proceso [17].

1.5.5. Experiencia de Usuario (UX)

El diseño de la experiencia de usuario (UX) es la primera impresión que tiene un usuario al visitar por primera vez un sitio o página web [18]. La UX es el término que utilizamos para cualquier actividad que provoque una mejor experiencia para el usuario [19]. Su estudio es importante en la creación de diferentes tipos de productos y servicios, y la web es uno de los campos más importantes en los que se aplica el diseño de la experiencia de usuario [20].

Como podemos observar en la Figura 3, la UX abarca varios aspectos con el objetivo de concentrarse en la usabilidad, entre estos aspectos tenemos al propio usuario, factores sociales,

factores culturales, contexto de uso y el producto.

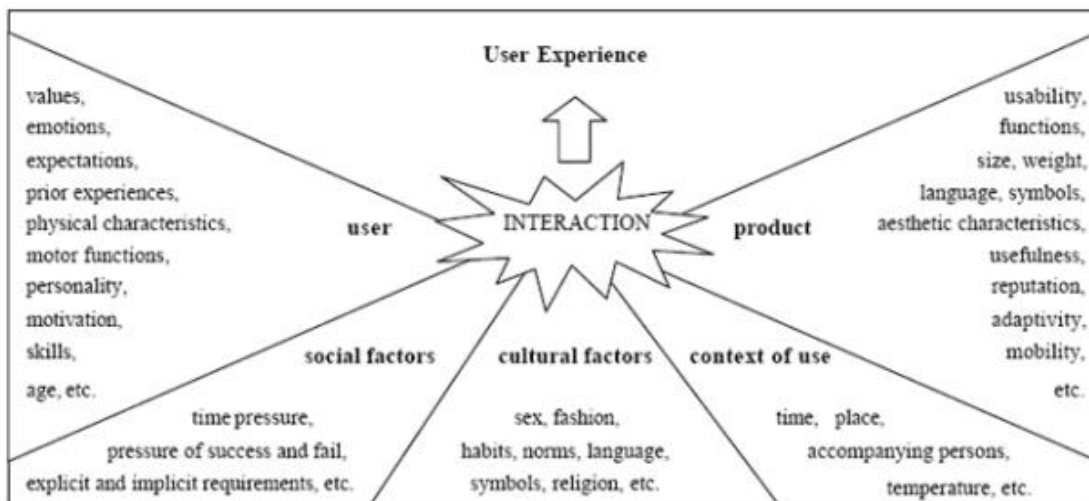


Figura 3. Representación de UX. [21]

La UX abarca los efectos que siente un usuario como resultado de una interacción con un sistema o producto, incluyendo la influencia de la usabilidad, la utilidad, y el impacto emocional durante esta interacción. En este sentido, la UX contiene a la usabilidad.

1.5.5.1. Usabilidad

La usabilidad es un atributo de calidad de la interfaz de usuario, enfocado en la facilidad de aprendizaje del sistema, la eficiencia de uso, y agradable [22]. De acuerdo con la ISO 9241-2010:2019 [16] la usabilidad se refiere al grado en el que un sistema, producto o servicio puede ser utilizado por usuarios específicos con el objetivo de lograr efectividad, eficiencia y satisfacción en un contexto de uso específico.

Según el estándar ISO/IEC 9126-1[23] nos propone el Modelo de Usabilidad que cuenta con diversas características que a su vez se descomponen en atributos en los que se puede aplicar métricas para calcular un valor cuantificable de usabilidad.

El Modelo de Usabilidad está compuesto por las siguientes sub-características y atributos [24]:

Tabla 2. Sub-características y métricas del Modelo de Usabilidad

Sub-características	Métricas
Facilidad de aprendizaje	Capacidad de un producto software para permitir al usuario aprender la funcionalidad del sistema.
Comprensibilidad	Capacidad de un producto software de permitir al usuario comprender si el producto es adecuado y como puede ser utilizado.
Operabilidad	Capacidad de un producto software de permitir al usuario operar con él y

	configurarlo.
Grado de atracción	Aspecto relacionado con la atracción del diseño estético de la interfaz de usuario.

1.5.5.2. Cuestionario SUS

La Escala de Usabilidad de un Sistema (SUS, *System Usability Scale*) es una herramienta metodológica de medición de la usabilidad en objetos, dispositivos o aplicaciones por parte de usuarios los cuales han hecho uso de los mismos [25]. El cuestionario consta de 10 preguntas cuyas respuestas se miden a través de una escala de Likert de cinco niveles [26].

1. Totalmente en desacuerdo.
2. En desacuerdo.
3. Ni de acuerdo ni en desacuerdo.
4. De acuerdo.
5. Totalmente de acuerdo.

El cuestionario SUS se utiliza generalmente después de que el encuestado utiliza el sistema que va a ser evaluado, pero antes de emitir cualquier juicio de valor o discusión. Para mayor efectividad del cuestionario SUS, se debe pedir respuestas rápidas por parte del encuestado en vez de pensar mucho tiempo en una pregunta. En el caso de que un encuestado sienta que no puede responder a una pregunta en particular, este deberá marcar en la casilla del centro de la escala. Todas las preguntas deben ser respondidas para que el cuestionario se realice con éxito.

El formato de las preguntas es el siguiente:

1. Creo que me gustaría utilizar este sistema con frecuencia
2. El sistema me parece innecesariamente complejo
3. Me pareció que el sistema era fácil de utilizar
4. Creo que necesitaría el apoyo de un técnico para ser capaz de utilizar este sistema
5. Me parece que las distintas funciones de este sistema estaban bien integradas
6. Me pareció que había demasiada inconsistencia en este sistema
7. Me imagino que la mayoría de la gente aprendería a utilizar este sistema muy rápidamente
8. El sistema me pareció muy engorroso/cansado de usar
9. Me sentí muy seguro al utilizar el sistema.
10. Tuve que aprender muchas cosas antes de poder ponerme a corriente con este sistema.

La forma en la que se califica una prueba SUS es sumando el puntaje de cada pregunta, considerando que cada pregunta puede ser valorada en el rango de 0 a 4 (escala de Likert). Para las preguntas 1,3,5,7, y 9 el puntaje de la pregunta es la posición en la escala menos 1. Para las preguntas 2,4,6,8 y 10, el puntaje es de 5 menos la posición de la escala. Esta suma

de puntajes se multiplica por el valor de 2,5 para obtener el valor total. El rango de este valor esta entre 0 a 100 [27].

1.5.6. Diseño de Interfaz de Usuario

El principal objetivo de la interfaz de usuario (IU) es el encajar y satisfacer las habilidades y expectativas de los usuarios. Los sistemas siempre son juzgados por el diseño de su IU más que por su funcionalidad. Muchos sistemas de software nunca llegan a ser utilizados debido al carente diseño de su IU, llegando a ser confuso para el usuario y llevándolo en ocasiones a cometer errores innecesarios [28]. Diseñar la IU involucra considerar que existe muchos tipos diferentes de personas que tienen diversos pensamientos y preferencias en cuanto a preferencia visual, de audio, estilos, tipografía, etc. [29] .

Diseñar la IU implica considerar varios principios de diseño a seguir para desarrollar una IU.

- La IU debe lucir familiar para el usuario, es decir conceptos tales como documentos, carpetas, identificadores de archivos deben ser evidentes y estar basados en términos comunes en vez de términos computacionales.
- Mantener una consistencia en cuanto a la forma de los menús, utilizar un mismo formato y puntuación
- El usuario debe poder utilizar un comando u operación sin preocuparse que se produzca un efecto inesperado al estar manipulando el sistema y tener la posibilidad de que exista la opción de volver atrás si aparece algún error durante el proceso de manipulación de la interfaz.
- Para esto, la IU debe incluir un manual para guiar al usuario en cada paso y abarcar a la mayor cantidad posible de diferentes usuarios volviendo accesible para todos a la interfaz [30].

1.5.7. Adobe XD

Adobe XD, o conocida también como *Adobe Experience Design CC*, es una herramienta de software de diseño y prototipado basado en vectores creado por la empresa de software estadounidense Adobe Inc., el cual abarca con todo el proceso de trabajo que se requiere para crear un wireframe hasta la implementación de un prototipo interactivo [31]. Para la elaboración de este trabajo de integración se utilizó la versión 51.0.12.6 de Adobe XD. Es una las dos herramientas de prototipado más utilizada en el mercado [32] y nos ofrece una versión demo para utilizarla en sistemas operativos Windows y Mac OS.

En la Figura 4 podemos observar la interfaz de usuario que nos ofrece Adobe XD en su modo de diseño. Tendremos tres modos disponibles, el de diseño que podemos observar en la figura, el de prototipo, y compartir. En el lado izquierdo tenemos la barra de herramientas con opciones de creación de figuras y agregar texto. En el lado derecho en cambio tendremos la opción de modificar las propiedades del entorno y de los elementos agregados a este [33].

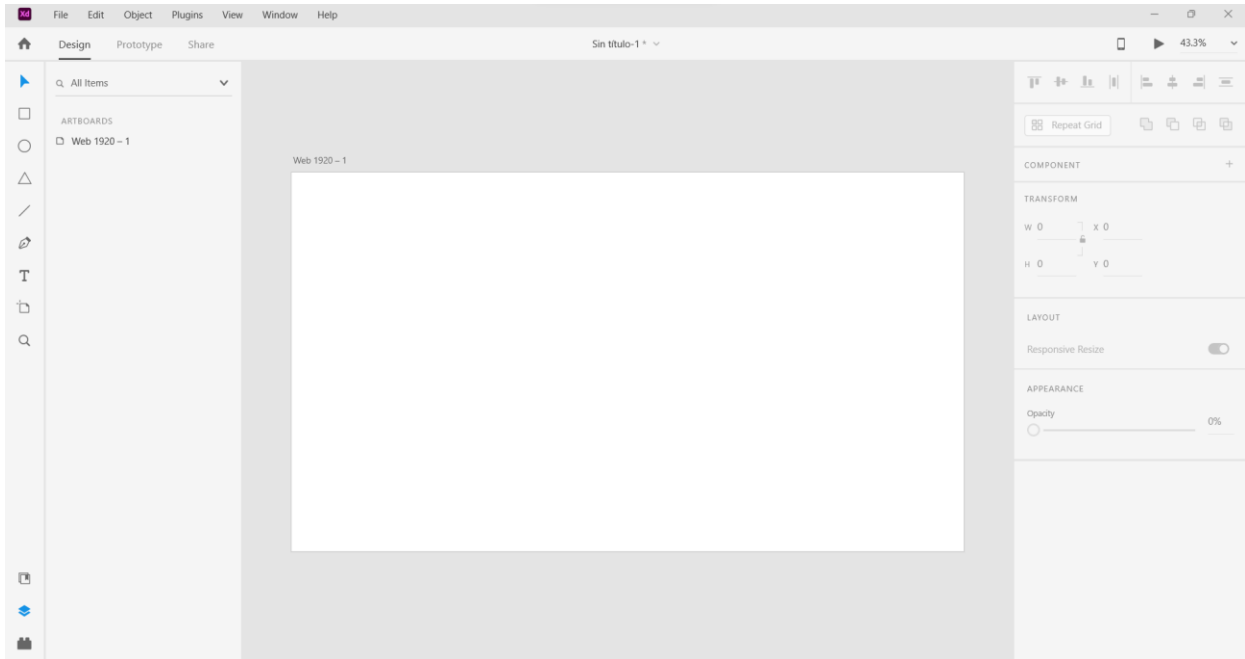


Figura 4. Interfaz de Usuario de Adobe XD

El modo de prototipado nos permite editar las interacciones y el modo compartir nos permitirá compartir el prototipo creado [34].

2. METODOLOGÍA

La metodología propuesta para el desarrollo de este trabajo de integración curricular es el Diseño Centrado en el Usuario (DCU) definida en la sección 1.4.4. A continuación, detallaremos las fases que se han seguido al seguir esta metodología.

2.1. Análisis de Contexto

Para el análisis de contexto se encuestó a los estudiantes de la Facultad de Sistemas de la Escuela Politécnica Nacional, específicamente a estudiantes que cursan el pregrado de las carreras de Computación y Software, desde sexto semestre hasta noveno semestre. Las encuestas fueron contestadas por 54 estudiantes, que equivale aproximadamente al 10% del total de estudiantes cursando estudios en la Facultad. Las encuestas fueron realizadas mediante Microsoft Forms y consistió en 19 preguntas, divididas en cuatro secciones (Anexo 1) :

1. La primera sección, "Información Personal", consta de 5 preguntas para conocer los datos personales además de los intereses de los estudiantes encuestados.
2. La segunda sección, "Infraestructura de TI", consta de 5 preguntas con el propósito de conocer el nivel de conocimiento acerca del diseño y modelado de Infraestructura TI de los estudiantes.
3. La tercera sección, "Infraestructura como Servicio", consta de 5 preguntas con el propósito de conocer el nivel de conocimiento acerca de proveedores de infraestructura y aprovisionamiento de infraestructura de los estudiantes.
4. Finalmente, la cuarta sección, "Editores e Interfaz de Usuario", consta de 4 preguntas en

las cuales se quiere conocer el conocimiento y el uso que ha realizado el estudiante en interfaces de usuario que sirven para la gestión de infraestructura de TI en la nube. A breves rasgos, los resultados de la encuesta arrojaron que los estudiantes de la Facultad se encuentran en un rango de 21 a 23 años y cursando los niveles de entre sexto hasta noveno semestre. Más detalles son sintetizados en la fase de “Definición de Requisitos”.

2.2. Definición de Requisitos

2.2.1. Perfil Persona

En la fase de la definición de requisitos se utilizó la técnica “Persona” [35] para sintetizar la información recabada en la fase anterior. A partir de la aplicación de la técnica, se obtuvo el perfil de la potencial audiencia. Mediante esta técnica se definieron dos Personas:

- 1) “Mari” como estudiante de la carrera de Software con poca experiencia en el área del cloud computing (Figura 5).
- 2) “Edison” como estudiante de la carrera de Computación con conocimiento acerca del balanceo de carga (Figura 6).

Estas plantillas están basadas en las recopilaciones de los rasgos más comunes que presentaron los estudiantes que hemos encuestado.



Mari
24 años, Ecuador

Soy estudiante de la carrera de software de la EPN. No tengo mucha experiencia en cloud computing pero es necesario para mi trabajo de desarrollo de FrontEnd y BackEnd

OBJETIVOS

- Realizar el levantamiento de servidores en la nube para para desarrollo de FrontEnd y BackEnd
- Implementar servidores de diferentes proveedores para aplicaciones web

USO DE TECNOLOGÍAS

- Uso de proveedores en la nube (Algunas veces)
- Levantar servidores para FrontEnd (Bueno)
- Internet (Bueno)

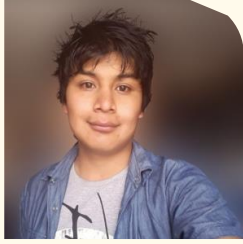
MOTIVACIONES

- Agilizar su trabajo en la nube
- Obtener facilidad para levantar servicios

FRUSTRACIONES

- No tener conocimiento de los diferentes proveedores de infraestructura
- Gran cantidad de pasos a seguir para levantar servicios en la nube

Figura 5. Plantilla Persona “Mari”. Fuente: Elaboración propia



Edison

24 años, Ecuador

Soy estudiante de la carrera de computación de la EPN. Me dedico al trabajo con infraestructura tecnológica, Tengo conocimiento sobre balanceo de carga.

OBJETIVOS

- Identificar elementos dentro de la infraestructura en la nube
- Realizar balanceo de carga para infraestructura en la nube

USO DE TECNOLOGÍAS

- Servicios de AWS (bueno)
- Uso de editores gráficos para diagramar (Algunas veces)

MOTIVACIONES

- Modelar infraestructura en la nube de una forma ms sencilla mediante in diagramador

FRUSTRACIONES


- Dependecia con el proveedor para realizar el modelado.
- Modelado de infraestructura por linea de comandos.

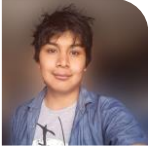
Figura 6. Plantilla Persona “Edison”. Fuente: Elaboración propia

2.2.2. Matriz OTI

Siguiendo con la fase de la definición de los requisitos, realizamos la matriz OTI (Objetivos, Tareas e Interfaces). La Matriz OTI es una técnica que facilita la identificación de las interfaces de usuario a diseñar a partir de los objetivos de los perfiles Persona que hemos definido en la sección anterior. La Tabla 3 muestra la matriz OTI realizada para este trabajo de integración curricular, que consta de los objetivos, tareas e interfaces.

Tabla 3. Matriz OTI. Fuente: Elaboración propia

	Objetivos	Tareas	Interfaces
 Mari	O.1.- Implementar los servicios de balanceo de carga para la infraestructura en la nube.	Crear reglas de balanceo de carga.	Diseño
		Configurar las reglas de balanceo de carga	Configuración del balanceo de carga
		Seleccionar imagen a replicar	Configuración del balanceo de carga
	O.2.- Configurar servicios de balanceo	Crear grupo de trabajo	Diseño

	de carga en infraestructura en la nube.	Modificar reglas de balanceo de carga	Configuración de balanceo de carga
		Generar script de infraestructura	Configuración de ejecución
 <p>Edison</p>	O.3.- Realizar el levantamiento de servidores en la nube para desarrollo de FrontEnd y BackEnd.	Crear un diagrama	Inicio
		Crear una instancia	Configuración de instancia
	O.4.- Implementar servidores de diferentes proveedores para aplicaciones web.	Seleccionar tipo de instancia	Configuración de instancia
		Configurar de la instancia	Configuración de instancia

2.2.3. Mapa Navegacional

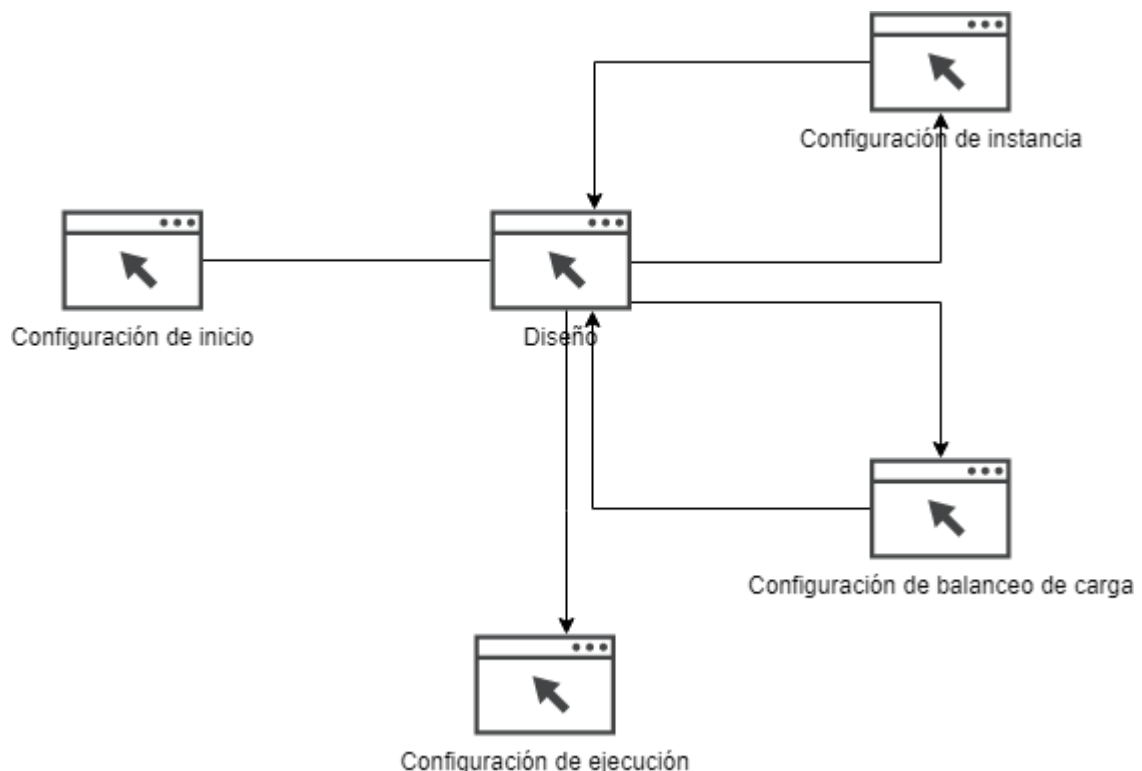


Figura 7 Mapa navegacional. Fuente: Elaboración propia

De la columna “Interfaces” de la matriz OTI obtenemos cinco interfaces de usuario que permitirían al usuario realizar las tareas para lograr sus objetivos: “Configuración de inicio”, “Diseño”, “Configuración de instancia”, “Configuración de balanceo de carga”, y “Configuración de ejecución”. Con estas interfaces, se ha elaborado el mapa navegacional que se muestra en

la Figura 8.

2.3. Diseño

Considerando el Mapa Navegacional de la Figura 7, empezamos la fase de “Diseño” del DCU. En esta fase utilizaremos cuatro herramientas de diseño para idear la solución de diseño: “Bocetos”, “Wireframes”, “Diseño Visual”, y “Prototipo Interactivo”.

2.3.1. Bocetos

El boceto es un diseño de baja fidelidad de la interfaz de usuario. La Figura 8 presenta el boceto correspondiente a las pantallas definidas en el mapa navegacional (Figura 7) con cada uno de sus componentes debidamente rotulados y definidos en el mapa navegacional. En base al conocimiento que hemos adquirido y recopilado en el marco teórico acerca del balanceo de carga y la herramienta ARGON se ha decidido utilizar el patrón concentrador [36], que consiste en un patrón de diseño que agrupa todas las funcionalidades en una sola interfaz con el objetivo de brindar al usuario todas las herramientas visibles al momento de ingresar a la interfaz. Como por ejemplo la interfaz de Adobe XD (Figura 4).

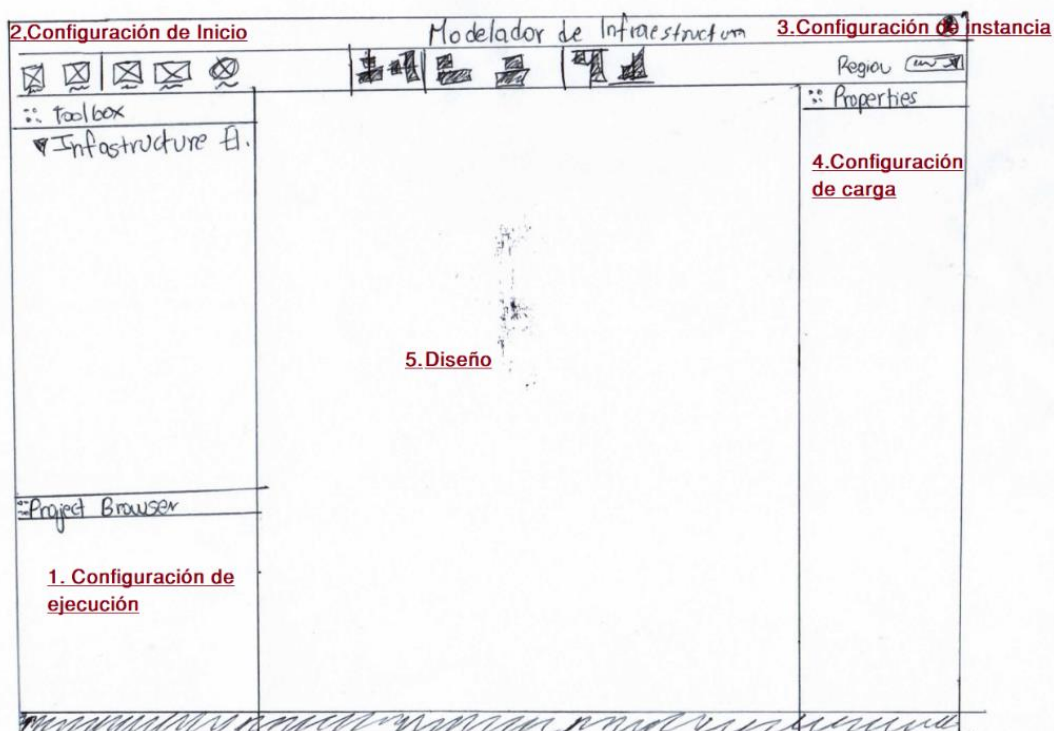


Figura 8. Boceto de la interfaz de usuario propuesta. Fuente: Elaboración propia

2.3.2. Wireframe

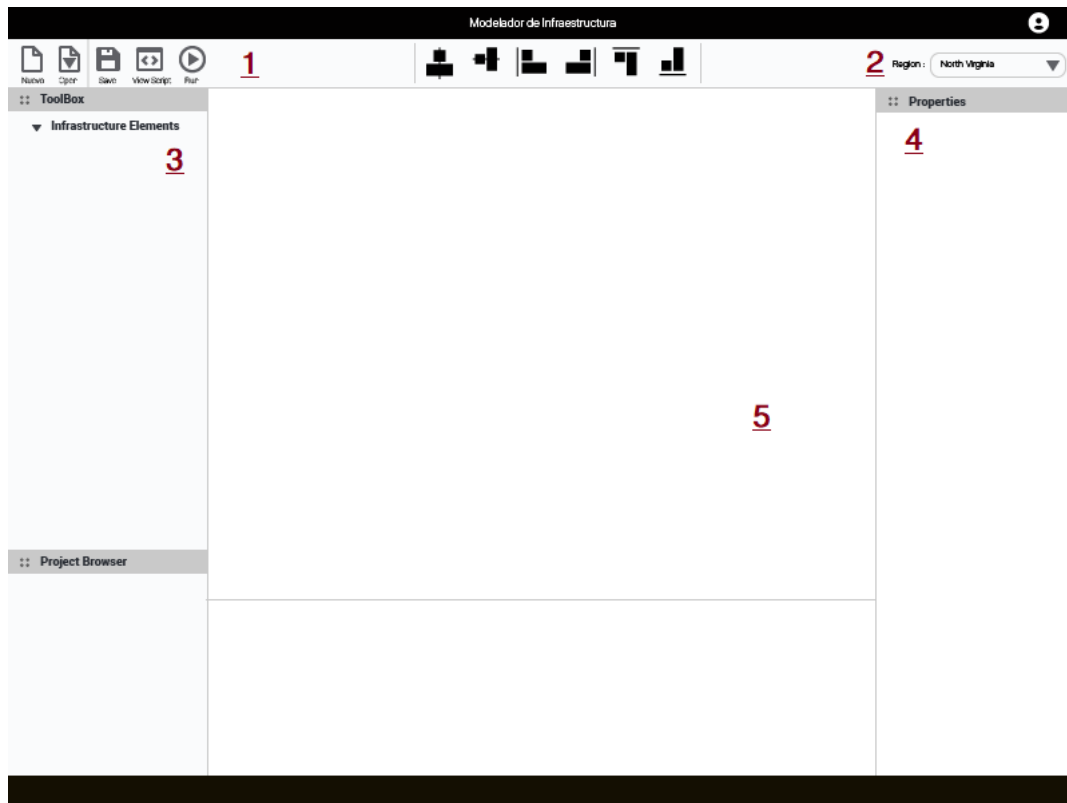


Figura 9. Wireframe de la interfaz de usuario propuesta. Fuente: Elaboración propia

A partir del boceto, generamos un wireframe. Un wireframe es un prototipo de baja fidelidad enfocado en representar la distribución estructurada de los elementos de la interfaz. La Figura 9 indica el wireframe principal con 5 componentes distribuidos en él, los cuales fueron definidos previamente mediante el mapa navegacional:

- 1) **Configuración de inicio**, ubicada en la parte superior izquierda del wireframe, cuenta con una cinta de opciones en los cuales se podrá crear, abrir, guardar, revisar el script, y ejecutar el script que en código expresa el modelo gráfico diseñado.
- 2) **Configuración de instancia**, ubicada en la parte superior derecha del wireframe, cuenta con una lista desplegable en la cual encontraremos cada una de las regiones para la instancia.
- 3) **Configuración de ejecución**, abarca las pestañas de la caja de herramientas del proyecto y el listado jerárquico de los elementos que han sido colocados en el diagrama del modelo.
- 4) **Configuración de balanceo de carga**, está en la sección derecha del wireframe. Presentará las opciones y propiedades de los elementos de la infraestructura que hayan sido colocados en el diagrama del modelo.
- 5) **Diseño**, es el diagrama del modelo y se ubica en la parte central de la interfaz. Mediante *Drag & Drop*, una técnica que facilita la interacción del usuario de forma intuitiva con el fin de llevar un elemento de la interfaz de un lugar a otro,

arrastraremos los elementos desde la caja de herramientas y posteriormente configurar estos elementos en el componente de configuración de balanceo de carga.

2.3.3. Diseño Visual

En el diseño visual definimos un logo, paleta de colores y tipografía.

2.3.3.1. Logo

ARGON no cuenta con un logo como tal, por lo cual se decidió diseñar un isotipo y logotipo Figura 10 para el diseño visual propuesto en este trabajo. Un logotipo es un identificador visual o gráfico formado por letras, cifras o palabras en cambio un isotipo está compuesto exclusivamente por imágenes [37]. Considerando que en muchos sitios web el color negro, blanco y azul se utilizan con frecuencia para representar tecnología [38], se decidió que el logotipo e isotipo tengan una apariencia formal dado que el enfoque del prototipo está dirigido a un ambiente profesional y educativo en el manejo de infraestructura en el Cloud Computing.



Figura 10. a) Isotipo b) Logotipo. Fuente: Elaboración propia

2.3.3.2. Color y Tipografía

A partir de los colores del logotipo diseñado, se obtuvo dos paletas de colores, cada color con su respectivo código hexadecimal para facilitar su uso en la web. Considerando el predominio de la tonalidad de azul y negro, la primera paleta se obtuvo mediante separación complementaria, y la otra paleta se obtuvo utilizando los monocromáticos del azul elegido Figura 11. La separación complementaria es una variación de los colores complementarios, formada por dos colores adyacentes con su complementario [39], mientras que los colores monocromáticos son todos los colores de un solo matiz o tono [40].



Figura 11. a) Paleta de colores mediante separación complementaria b) Paleta de colores monocromáticos del azul elegido

monocromáticos. Fuente: Adobe Color

En relación con la tipografía, se eligió Lemon Milk (Figura 12), la misma en la que se basó el logotipo. Esta tipografía está disponible bajo licencia libre y fue desarrollada por Muhammad Ariq Syauql conocido como MARSNEV [41], un artista tipográfico indonesio.

The image shows the text 'LEMON MILK' in a bold, sans-serif font. Above the word 'LEMON' is a small, light blue text 'LEMONMILK-Regular.otf'. The letters are black and have a clean, modern appearance.

Figura 12. Tipografía Lemon Milk Regular

2.3.4. Prototipo interactivo

En esta sección se presentan la solución en forma de un prototipo de alta fidelidad interactivo (el usuario puede interactuar con la interfaz) y su archivo descargable se encuentre en el ANEXO III. El prototipo se basa en el boceto y wireframe definidos anteriormente y siguiendo la trazabilidad desde la definición del perfil persona hasta el producto final que se presenta en esta sección. El prototipo cuenta con 21 pantallas en Adobe XD, mediante las cuales se realizará la respectiva evaluación con el usuario. La evaluación se describe más adelante, en la sección 2.4.

Los iconos utilizados en el prototipo provienen de material de Material Design [42]. Los iconos fueron seleccionados en función de dos consideraciones: 1) los iconos deben ser minimalistas, garantizar la legibilidad y claridad en tamaños grandes y pequeños, ser compatibles con varios navegadores como Google Chrome, Mozilla Firefox y Apple Safari y tener la posibilidad de ser optimizados para la web, 2) los iconos deben ser de libre uso. En efecto, Google nos presenta iconos bajo la licencia Apache versión 2.0 [43] por lo cual son de libre uso y totalmente gratuitos. La Figura 13 presenta el ejemplo inicial al usuario de una infraestructura completa. Este es un posible resultado final el cual pueda tomar como referencia para sus futuros trabajos.

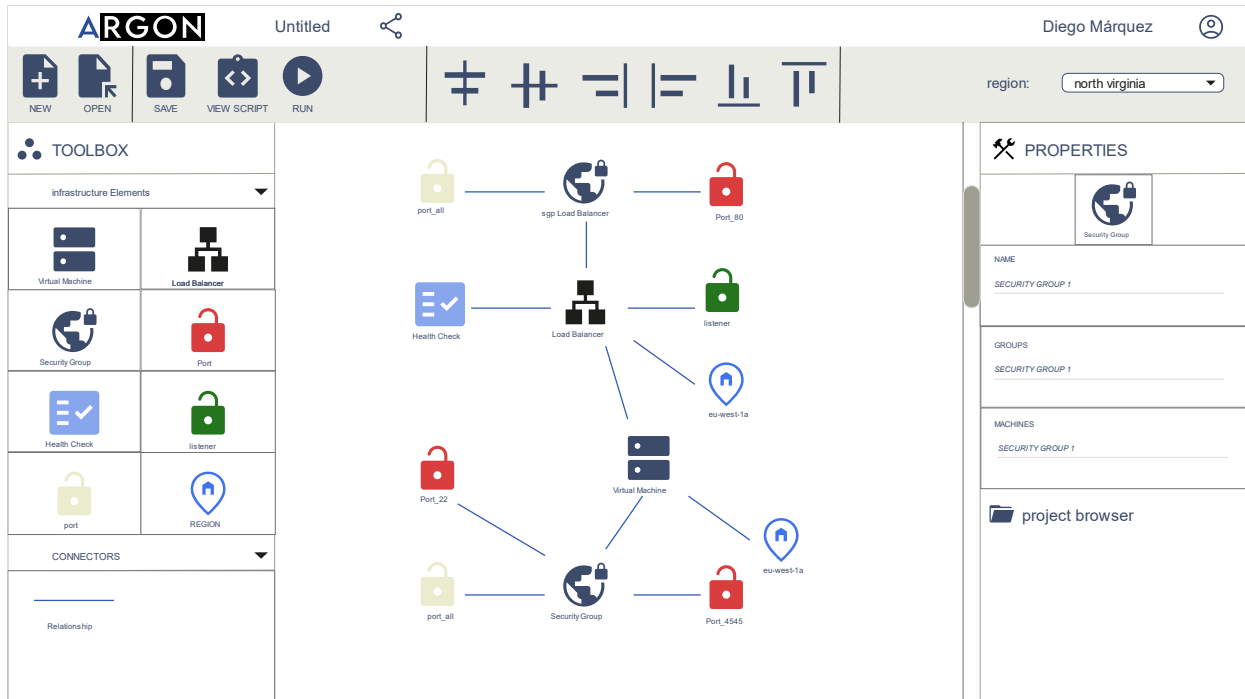


Figura 13. Ejemplo del prototipo interactivo elaborado en AdobeXD.

2.4. Evaluación

2.4.1. Objetivo

Para evaluar el prototipo de la sección anterior, se realiza pruebas con usuarios [17]. Este tipo de prueba permite evaluar la usabilidad de la interfaz de una manera sencilla y fiable, permitiéndonos ver de primera mano dónde los usuarios encuentran mayor dificultad.

2.4.2. Variables a evaluar

Según la ISO 9241-11:2018 [44], la Usabilidad se define en función de tres variables: la satisfacción, la eficacia y la eficiencia. Los estudios de Beltré Ferreras [45] mencionan que la satisfacción del usuario es una de las variables que determina el grado de aceptación de un producto y por tanto su usabilidad. Por consiguiente, la variable que evaluaremos en este trabajo es la satisfacción, definida como la comodidad y la aceptabilidad de uso [46]. Para medir la satisfacción se utiliza el cuestionario SUS, definido en la sección 1.5.5.2.

2.4.3. Objetivos y Tareas para Pruebas de Usabilidad

Los objetivos y tareas para realizar las pruebas de usabilidad son los mismos que hemos definido en la matriz OTI (Tabla 3).

2.4.4. Diseño a evaluar

El diseño a evaluar es el que se muestra en la Figura 13, el cual ha sido diseñado como un prototipo interactivo que permite a los usuarios realizar las tareas para modelar una infraestructura para el caso del balanceo de carga. Las pruebas de usabilidad generalmente

consisten en comparar dos diseños, uno existente con un propuesto. En el caso de este estudio, el uso de la interfaz de usuario ya existente de ARGON suponía una serie de limitaciones, como la búsqueda de la versión deprecada de Eclipse en la cual se levanta esta, permisos de uso, y posibles errores de ejecución durante la evaluación. Sin embargo, como trabajo futuro se planteará realizar la comparación entre diferentes versiones del diseño propuesto, además en diferentes contextos y casos de uso.

2.4.5. Participantes

La evaluación contó con 5 participantes: 3 estudiantes de la carrera de Ingeniería en Computación y 2 estudiantes de la carrera de Ingeniería en Software, ambos grupos de la Escuela Politécnica Nacional. Los participantes han sido seleccionados considerando el cumplimiento de las características de los perfiles de usuarios definidos en la sección 2.2.1.

2.4.6. Protocolo de Evaluación

La ejecución de la evaluación sigue un procedimiento de tres pasos:

1. El evaluador explica al participante el contexto de la prueba y las tareas a realizar.
2. El evaluador solicita al participante realizar diversas acciones o tareas, a la vez que responde las dudas del participante.
3. A lo largo de la prueba, el evaluador irá anotando gestos, acciones, y comentarios del participante.
4. El participante evalúa la satisfacción percibida al usar el prototipo utilizando el cuestionario SUS.

Una acotación importante en el protocolo, principalmente en el paso 2, es que se decidió utilizar el método de evaluación de *Think Aloud*, en el cual se pide al usuario que durante la prueba indique en voz alta lo que piensa con el objetivo de recopilar la mayor cantidad de información a lo largo de la prueba [17]. En el contexto de la pandemia actual y dado que al momento de realizar las evaluaciones se suspendió la presencialidad en la Escuela Politécnica Nacional, se tomó la decisión de realizar la prueba de manera remota con la ayuda del programa de videoconferencia Zoom.

2.4.7. Ejecución

La Figura 14 muestra una captura de pantalla de una de las pruebas con el usuario a través de una sesión de Zoom. En esta sesión se guía al usuario a través de una serie de preguntas para el manejo del prototipo interactivo para el modelamiento de infraestructura.

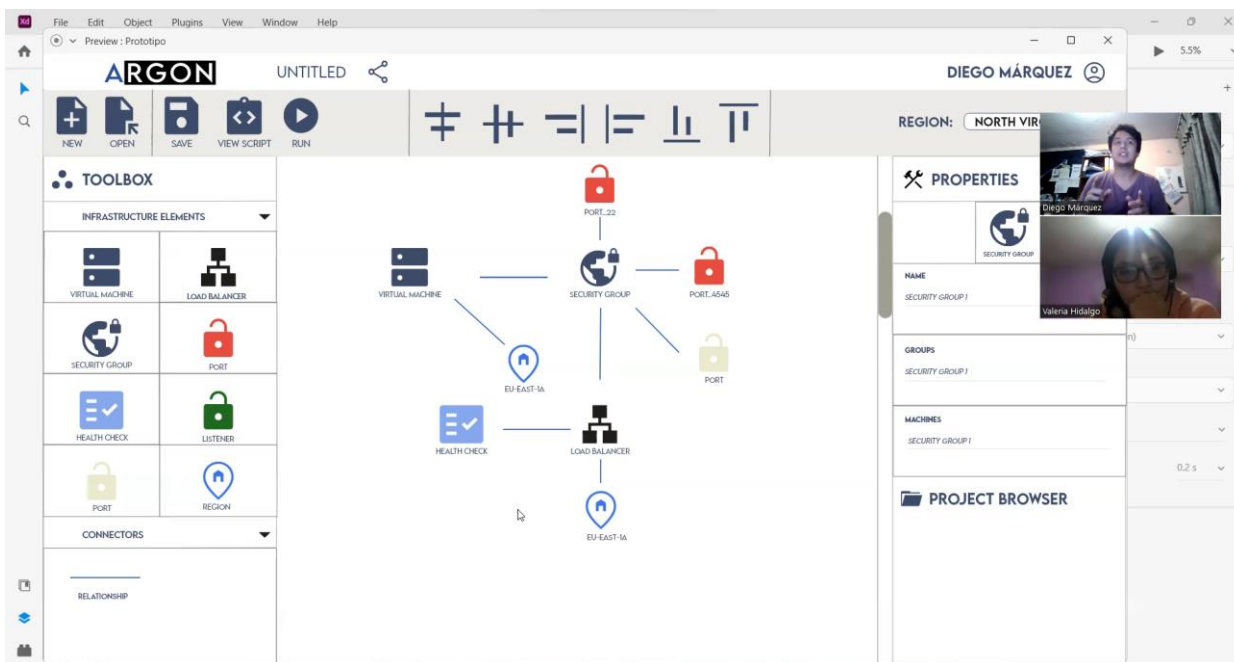


Figura 14. Ejecución de la prueba con el usuario a través de Zoom

El lector puede reproducir las grabaciones de otras pruebas similares a la que se muestra en la Figura 14, accediendo a los enlaces indicados en el ANEXO I.

3. RESULTADOS, CONCLUSIONES Y RECOMENDACIONES

3.1. Resultados

La Tabla 4 presenta los resultados del cuestionario SUS realizado por los participantes de la evaluación. Los participantes 1,3,5 corresponden a estudiantes de la carrera de Computación y los participantes 2,4 corresponden a estudiantes de la carrera de Software; ambos grupos de usuarios corresponden con el perfil Persona “Mari” y “Edison” respectivamente.

Tabla 4. Respuestas del cuestionario SUS. Fuente: Elaboración propia

Usuario	Pregunta 1	Pregunta 2	Pregunta 3	Pregunta 4	Pregunta 5	Pregunta 6	Pregunta 7	Pregunta 8	Pregunta 9	Pregunta 10
1	5	1	5	1	5	1	5	1	5	1
2	5	1	5	3	5	1	5	1	5	2
3	3	3	3	2	2	3	3	3	3	3
4	4	2	4	2	4	3	4	1	4	4
5	5	2	5	1	5	1	5	1	4	1

Determinamos los resultados siguiendo los lineamientos de cálculo descritos en la sección 1.5.5.2, es decir, para las preguntas impares (1,3,5,7, y 9) restamos el valor asignado por el participante menos uno y en cambio las preguntas pares (2,4,6,8,10) se restará cinco menos el valor asignado por el participante. El resultado parcial es multiplicado por 2,5 y de esta manera se pondera el valor de cada pregunta (Cuestionario SUS). La Tabla 5, muestra el resultado final ponderado.

Tabla 5. Resultado final SUS. Fuente: Elaboración propia

Usuario	Resultado parcial SUS	Resultado final SUS
1	40	100
2	37	92.5
3	20	50
4	28	70
5	38	95
Promedio:	32.6	81.5

En base a los resultados de la Tabla 5, se realiza un promedio considerando que el usuario 1,3,5 coinciden con el perfil Persona de “Mari” y el 2 y 4 con el perfil Persona de “Edison” (Tabla 6). Representando así en un gráfico los resultados para cada plantilla Persona (Figura 15) en función a la satisfacción.

Tabla 6. Promedio final en decimales de cada Perfil Persona. Fuente: Elaboración propia

Mari	81.67
Edison	81.25

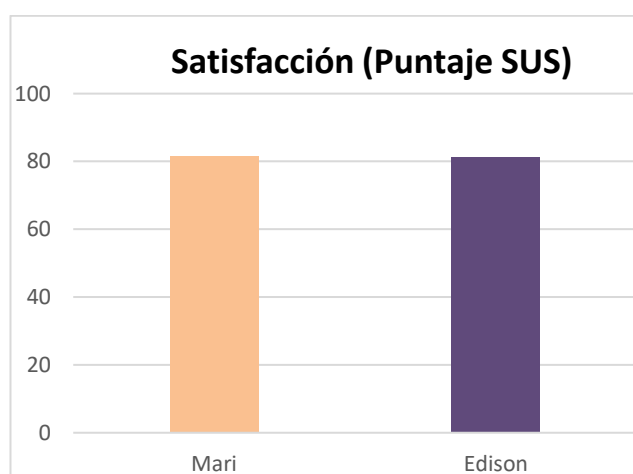


Figura 15. Satisfacción (puntaje SUS). Fuente: Elaboración propia

A continuación, se explica a detalle los resultados obtenidos en la Figura 15 en función de los objetivos planteados en la matriz OTI (Tabla 3).

- **Objetivo 1 (Mari): Implementar servicios de balanceo de carga para la infraestructura en la nube:** La percepción del diseño creado tuvo buena acogida por el participante, de tal forma que no presentó problemas para cumplir con los objetivos. El participante destacó la facilidad con la cual se puede agregar elementos para implementar los servicios de balanceo de carga además de la identificación y tamaño de los elementos, como se puede apreciar en la Figura 16.

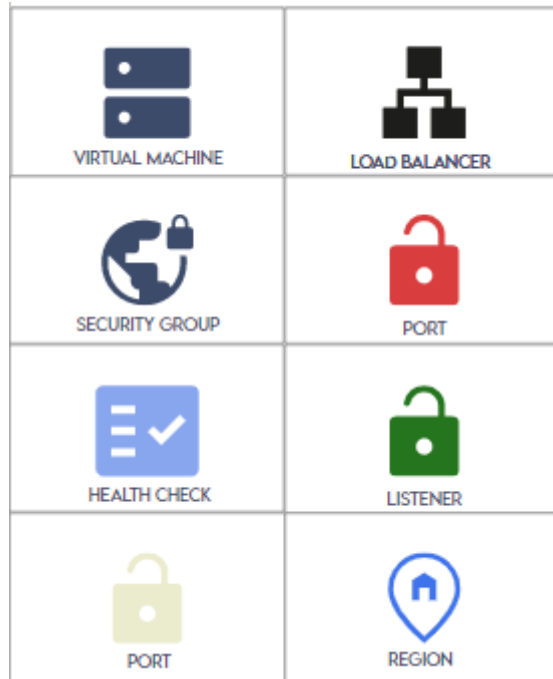


Figura 16. Elementos para la implementación de una infraestructura

- **Objetivo 2 (Mari): Configurar servicios de balanceo de carga en infraestructura en la nube:** La percepción del diseño creado tuvo buena acogida por el participante, de tal forma que no presentó problemas para cumplir con los objetivos. El participante destacó la facilidad de encontrar todas las opciones concentradas en una sola interfaz.
- **Objetivo 3 (Edison): Realizar el levantamiento de servidores en la nube para desarrollo de FrontEnd y BackEnd:** La percepción del diseño creado tuvo buena acogida por el participante, de tal forma que no presentó problemas para cumplir con los objetivos. El participante menciono agradecerle la idea de ejecutar el modelo antes de generar el script para levantar un servicio en la nube para desarrollo de FrontEnd y Backend mediante un botón. Como podemos apreciar en la Figura 17.



Figura 17. Botones para generar un script y levantar un servidor en la nube

- **Objetivo 4 (Edison): Implementar servidores de diferentes proveedores para aplicaciones web:** La percepción del diseño creado tuvo buena acogida por el participante. El participante no presentó problemas para cumplir con los objetivos. El participante destacó la posibilidad de cambiar de regiones sin cambiar de interfaces. Como lo muestra la Figura 18.



Figura 18. Lista desplegable con las diferentes regiones disponibles

3.2. Conclusiones

- Respecto al Objetivo 1: “Identificar usuarios potenciales, sus objetivos y requisitos de usabilidad considerando el caso de uso balanceo de carga”, se definió el contexto de uso mediante encuestas a los estudiantes de la Escuela Politécnica Nacional de la facultad de Sistemas de las carreras de Computación y Software, de las cuales se destaca que i) la mayoría de los estudiantes consideran tener bases sólidas acerca del aprovisionamiento de infraestructuras, ii) la mayoría de estudiantes estaría dispuesto a aprender a utilizar scripts de código para gestionar la infraestructura de TI y iii) la mayoría de los estudiantes consideran que un editor visual gráfico como aplicación web les resultaría cómodo para gestionar la infraestructura de TI en la nube. Mediante la técnica Persona se especificó dos perfiles Personas (Mari y Edison) que recogen datos compartidos de un conjunto de individuos que presentan rasgos, objetivos, motivaciones y frustraciones semejantes. Los perfiles Persona han resultado de gran utilidad para orientar las decisiones de diseño realizadas a lo largo del proceso. En función a estos perfiles se decidió utilizar el patrón concentrador en la fase del diseño dado que en las motivaciones de los perfiles Persona “Mari” y “Edison” enfatizan la necesidad de modelar una infraestructura de una forma simple, por lo que con este patrón podemos simplificar la mayor cantidad de funciones y características en una sola interfaz.
- Respecto al Objetivo 2: “Diseñar el prototipo de interfaz de usuario de un editor gráfico Web”, se ideó un interfaz de usuario interactiva según la metodología DCU, partiendo de bocetos a lápiz, wireframes y prototipos de alta fidelidad, atendiendo a las características del Perfil de las Personas.
- Respecto al Objetivo 3: “Evaluar la usabilidad del prototipo de interfaz de usuario del editor gráfico Web diseñado”, se evaluó las interfaces de usuario utilizando el cuestionario SUS, llegando a constatar la satisfacción de los usuarios en el desempeño de los objetivos de las Personas (Mari y Edison) definidas dentro del marco del estudio inicial del trayecto del DCU. Se alcanzó una puntuación de 81,6/100 para "Mari", y "Edison" obtuvo una puntuación de 81,2/100.
- Respecto al Objetivo General: “El objetivo de este proyecto consiste en diseñar la interfaz de usuario de un editor gráfico Web para la herramienta ARGÓN que facilite a los operadores de infraestructura balancear la carga en el contexto del aprovisionamiento de infraestructura”, se diseñó la interfaz de usuario de un editor gráfico Web para la herramienta ARGÓN mediante la metodología DCU, siguiendo cada una de las fases

determinadas en esta, evaluando este diseño mediante el cuestionario SUS enfocándonos en la métrica de la satisfacción, y de esta manera concluyendo que el resultado obtenido es mayor a la media en base a las investigaciones de Jeff Sauro [47] por lo tanto se interpreta como un buen resultado para este diseño.

3.3. Recomendaciones

- Es recomendable que el software para creación de prototipos interactivos (de alta fidelidad) soporte las interacciones requeridas en el proyecto. El software AdobeXD utilizado en este trabajo para crear el prototipo interactivo tuvo limitaciones al momento de recrear la interacción de “Drag and Drop” de los elementos de infraestructura al diagrama de modelado.
- Con el fin de evitar inconvenientes o confusiones a los participantes durante la prueba, es recomendable que las pruebas con el usuario sean realizadas en un ambiente sin distracciones tanto para el evaluador como para los participantes. Si se realiza la prueba en un ambiente remoto, es recomendable verificar previamente la conexión a Internet y sí es necesario esperar un periodo de tiempo hasta que la conexión mejore.

4. REFERENCIAS BIBLIOGRÁFICAS

- [1] H. Wackner, “¿Qué es el balanceo de carga?,” *Computerworld*, 2000.
<https://www.computerworld.es/tendencias/que-es-el-balanceo-de-carga> (accessed Jun. 05, 2022).
- [2] NGINX, “What Is Load Balancing? How Load Balancers Work.” 2018. Accessed: Jun. 05, 2022. [Online]. Available: <https://www.nginx.com/resources/glossary/load-balancing/>
- [3] A. W. Services, “Amazon Elastic Container Service Developer Guide,” AWS, 2014, Accessed: Jun. 05, 2022. [Online]. Available: <https://aws.amazon.com/de/eks/>
- [4] J. Sandobalin, E. Insfran, and S. Abrahao, “ARGON: A model-driven infrastructure provisioning tool,” *Proc. - 2019 ACM/IEEE 22nd Int. Conf. Model Driven Eng. Lang. Syst. Companion, Model. 2019*, pp. 738–742, 2019, doi: 10.1109/MODELS-C.2019.00114.
- [5] J. Sandobalin, E. Insfran, and S. Abrahao, “On the effectiveness of tools to support infrastructure as code: Model-driven versus code-centric,” *IEEE Access*, vol. 8, pp. 17734–17761, 2020, doi: 10.1109/ACCESS.2020.2966597.
- [6] H. Asma and S. Dallel, “Cognitive Load Theory and its Relation to Instructional Design: Perspectives of Some Algerian University Teachers of English,” *Arab World English J.*, vol. 11, no. 4, pp. 110–127, 2020, doi: 10.24093/awej/vol11no4.8.
- [7] J. Sandobalin, E. Insfran, and S. Abrahao, “End-to-end automation in cloud infrastructure provisioning,” *Inf. Syst. Dev. Adv. Methods, Tools Manag. - Proc. 26th Int. Conf. Inf. Syst. Dev. ISD 2017*, 2017.
- [8] K. Morris, *Infrastructure as Code Dynamic Systems for the Cloud Age*. 2020. Accessed: Jun. 13, 2022. [Online]. Available: <https://infrastructure-as->

- code.com/book/%0Ahttps://www.oreilly.com/library/view/infrastructure-as-code/9781098114664/
- [9] J. C. Sandobalín Guamán, “MoCIP: Un enfoque dirigido por modelos para el aprovisionamiento de infraestructura en la nube,” 2020, [Online]. Available: <https://riunet.upv.es/handle/10251/139077>
- [10] J. Humble and D. Farley, *Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation*. 2010.
- [11] S. Raskin, “the 2018 State of Crowdsourced Innovation,” *2018-11-01*, p. 25, 2018, Accessed: Jun. 05, 2022. [Online]. Available: <https://dzone.com/articles/the-2018-state-of-devops-reports>
- [12] Y. Brikman, *Terraform: Up & Running!* 2019.
- [13] IBM, “¿Qué es infraestructura de TI? - México | IBM,” 2021. <https://www.ibm.com/es-es/topics/infrastructure> (accessed Jun. 13, 2022).
- [14] M. Kushwaha and S. Gupta, “Various Schemes of Load Balancing in Distributed Systems-A Review,” *Int. J. Sci. Res. Eng. Technol.*, vol. 4, no. 7, pp. 741–748, 2015, [Online]. Available: www.ijret.org
- [15] Interaction Design Foundation, “User Centered Design What is User Centered Design ? UCD is an Iterative Process,” *Interact. Des. Found.*, pp. 1–14, 2021, [Online]. Available: <https://www.interaction-design.org/literature/topics/user-centered-design>
- [16] ISO 9241-210, “Ergonomics of human-system interaction-Human-centred design for interactive systems COPYRIGHT PROTECTED DOCUMENT,” 2019. Accessed: Jun. 05, 2022. [Online]. Available: <https://www.iso.org/obp/ui/#iso:std:iso:9241:-210:ed-2:v1:en>
- [17] A. Moreno Muñoz, “Diseño centrado en el usuario (I). Introducción,” *El Prof. la Inf.*, vol. 12, no. 1, pp. 52–54, 2003, doi: 10.1076/epri.12.1.52.19713.
- [18] M. Conde, “Designing the User Experience for Different User Needs for B2B E-Commerce”, Accessed: Jun. 15, 2022. [Online]. Available: <http://www.teknat.uu.se/student>
- [19] R. Hartson and P. S. Pyla, *The UX Book: Process and Guidelines for Ensuring a Quality User Experience*. 2012. doi: 10.1016/C2010-0-66326-7.
- [20] M. Orlova, “Mariia Orlova USER EXPERIENCE DESIGN (UX DESIGN) IN A WEBSITE DEVELOPMENT Website redesign Mariia Orlova,” no. December, 2016, [Online]. Available: <http://www.theseus.fi/handle/10024/120948>
- [21] L. Arhippainen and M. Tähti, “Empirical Evaluation of User Experience in Two Adaptive Mobile Application Prototypes,” 2003.
- [22] J. Nielsen, “User interface directions for the web,” *Commun. ACM*, vol. 42, no. 1, pp. 65–72, 1999, doi: 10.1145/291469.291470.
- [23] I.-I. 9126-1, “Ingeniería del software. Calidad del producto software. Modelo de calidad (ISO/IEC 9126-1:2001, IDT),” 2001, Accessed: Jul. 19, 2022. [Online]. Available:

- https://www.normalizacion.gob.ec/buzon/normas/n-te_inen_iso_iec_9126-1.pdf
- [24] S. España, I. Pederiva, J. I. Panach, S. Abrahão, and Ó. Pastor, "Evaluación de la usabilidad en un entorno de arquitecturas orientadas a modelos1," *Actas IDEAS 2006 - 9th Work. Iberoam. Ing. Requisitos y Ambient. Softw.*, no. May 2014, pp. 331–344, 2006.
- [25] F. Devin, "Sistema de escalas de usabilidad: ¿Qué es y para qué sirve?," *Uxpanol*, 2017. <https://uxpanol.com/teoria/sistema-de-escalas-de-usabilidad-que-es-y-para-que-sirve/> (accessed Aug. 12, 2022).
- [26] J. Brooke, "SUS: A 'Quick and Dirty' Usability Scale," *Usability Eval. Ind.*, no. November 1995, pp. 207–212, 2020, doi: 10.1201/9781498710411-35.
- [27] M. González Arencibia, D. Martínez Cardero, M. González Arencibia, and D. Martínez Cardero, "Dilemas éticos en el escenario de la inteligencia artificial," *Econ. y Soc.*, vol. 25, no. 57, pp. 93–109, May 2020, doi: 10.15359/EYS.25-57.5.
- [28] R. Guntupalli, "User interface design: methods and qualities of a good user interface design.," *Softw. Eng. 2008*, p. 37, 2008, [Online]. Available: www.hv.se
- [29] M. Grissinger, "Understanding human over-reliance on technology," *P and T*, vol. 44, no. 6, pp. 320–321, 2019. Accessed: Jun. 06, 2022. [Online]. Available: <https://www.interaction-design.org/literature/topics/ux-design>
- [30] J. Nielsen and R. Molich, "Heuristic evaluation of user interfaces," *Conf. Hum. Factors Comput. Syst. - Proc.*, no. April, pp. 249–256, 1990, doi: 10.1145/97243.97281.
- [31] Adobe Inc., "Herramienta de diseño y colaboración de experiencias e interfaces de usuario | Adobe XD," 2016. <https://www.adobe.com/la/products/xd.html> (accessed Jun. 05, 2022).
- [32] J. Palmer, Taylor & Bowman, "2021 Design Tools Survey," 2021. <https://uxtools.co/survey-2021/> (accessed Jul. 19, 2022).
- [33] F. Trojahn, "Hochschule für Telekommunikation Leipzig (FH) Institut für Telekommunikationsinformatik Master of Engineering Improving the Usability of UI Prototypes by," no. December, 2020, doi: 10.13140/RG.2.2.26202.77767.
- [34] P. Navigation, "Welcome to the Notability User Guide Table of Contents :", Accessed: Jul. 19, 2022. [Online]. Available: <https://helpx.adobe.com/xd/user-guide.html>
- [35] J. W. Castro and S. T. Acuña, "Integrando la Técnica P la Actividad de Análisis," no. November, 2015.
- [36] IBM, "Categoría de reglas Patrones estructurales - Documentación de IBM," 2021. <https://www.ibm.com/docs/es/rsar/9.5?topic=analysis-structural-patterns-rule-category> (accessed Aug. 12, 2022).
- [37] E. Harada, "Logotipos, isotipos, imagotipos e isologos: una aclaración terminológica," *Rev. Mix.*, vol. 2, no. 33, pp. 36–47, 2014.
- [38] 99desing, "¿Cómo elegir el color de un logo de tecnología? | 99designs," 2022. <https://99designs.es/logo-design/business-of-color/technology> (accessed Aug. 12, 2022).

- [39] C. McIntyre., “Let’s Make Mud – Understanding & Mixing Complementary Colors | Munsell Color System; Color Matching from Munsell Color Company,” 2015.
<https://munsell.com/color-blog/mud-understanding-mixing-complementary-colors/>
 (accessed Aug. 20, 2022).
- [40] Google, “Color monocromático — Google Arts & Culture,” 2022.
<https://artsandculture.google.com/entity/m0cxml?hl=es> (accessed Aug. 20, 2022).
- [41] MARSNEV, “Lemon_Milk.” 2020. Accessed: Jul. 30, 2022. [Online]. Available:
<https://www.dafont.com/lemon-milk.font>
- [42] Google, “Guía de íconos de material | Google Fonts | Google Developers,” 2022.
https://developers.google.com/fonts/docs/material_icons (accessed Jul. 31, 2022).
- [43] Apache.org, “Apache License, Version 2.0,” *Apache.Org*, no. January, pp. 2–4, 2004,
 Accessed: Jul. 31, 2022. [Online]. Available: <https://www.apache.org/licenses/LICENSE-2.0>
- [44] I. O. for Standardization, “ISO 9241-11,” in *International Organization*, 1998. Accessed:
 Aug. 12, 2022. [Online]. Available: <https://www.sis.se/std-611299>
- [45] U. DE Politecnica Madrid Tesis De Máster, H. Jacquelín Beltré Ferreras Asesor, and X. Ferré Grau, “FACULTAD DE INFORMATICA APLICACIÓN DE LA USABILIDAD AL PROCESO DE DESARROLLO DE PÁGINAS WEB,” 2008.
- [46] L. Perurena Cancio and M. Moráquez Bergues, “Usabilidad de los sitios Web, los métodos y las técnicas para la evaluación,” *Rev. Cuba. Inf. en Ciencias la Salud*, vol. 24, no. 2, pp. 176–194, Jan. 2013, Accessed: Aug. 20, 2022. [Online]. Available:
<http://www.rcics.sld.cu/index.php/acimed/article/view/405/306>
- [47] J. Sauro, “Measuring Usability with the System Usability Scale (SUS),” 2011.
<https://measuringu.com/sus/>

5. ANEXOS

ANEXO I: Enlace de ejecución de evaluaciones de prototipos de alta fidelidad

[Evaluaciones - ARGON](#)

ANEXO II: Encuesta: Modelamiento de Infraestructura en la Nube

Modelamiento de Infraestructura en la Nube

Como estudiante de la carrera de Software/Computación solicitamos de su ayuda para el levantamiento e identificación de usuarios objetivos, en el uso de cierta herramienta enfocada al modelamiento de Infraestructura en la nube.

Sección 1

...

Información Personal

1

¿Cuál es su género? *

- Masculino
- Femenino

2

Indique su rango de edad *

- menor a 21 años
- 21 - 23 años
- 23 - 25 años
- mayor a 25 años

3

Indique la carrera que actualmente está cursando *

- Software
- Computación

4

¿Cuál es el semestre que está cursando actualmente? *

- Cuarto
- Quinto
- Sexto
- Séptimo
- Octavo

- Noveno

5

¿Qué área(s) de la informática le interesa más? *

- Desarrollo FrontEnd
- Desarrollo BackEnd
- Infraestructura Tecnológica
- Seguridad
- Diseño UX
- Otras

Infraestructura de (Tecnologías de la Informa- ción) TI



6

¿Cree usted que modelar la Infraestructura TI de una empresa es complejo? *

- Sí
- No

7

Describa en un párrafo corto, ¿por qué cree que es complejo modelar la Infraestructura TI? *

Escriba su respuesta

8

¿Ha diseñado/modelado una Infraestructura TI? (incluye proyectos pequeños o medianos hechos en clases de Redes) *

- Sí
- No

9

¿Qué tipo de casos ha modelado? *

- Servidores
- Balanceo de Carga
- Escalamiento
- Otras

10

¿Con qué aspecto cree usted que tiene o tendría problemas al diseñar e implementar una Infraestructura TI? *

- Identificar elementos de infraestructura
- Esquematización lógica
- Configuración
- Levantamiento de Servicios
- Otras

Sección 3

...

Infraestructura como Servicio



11

¿Conoce el concepto de "Infraestructura como Servicio" (IaaS)? *

- Si
- No

12

¿Conoce o ha escuchado acerca del término "aprovisionamiento de infraestructura" en el contexto de la informática? *

- Si
- No

13

¿De qué Proveedor de Infraestructura tiene mayor conocimiento? *

- AWS
- Azure
- Google Cloud Platform
- Otro
- Ninguno

...

14

¿Qué problemas ha encontrado al utilizar los servicios de los Proveedores de Infraestructura? *

- Dependencia del proveedor
- Diferentes formas para realizar la misma actividad en cada proveedor
- Necesidad de acceso online
- Otras

15

¿Estaría dispuesto a aprender a gestionar la Infraestructura de TI utilizando scripts de código? *

- Si
- No

Sección 4

Editores e Interfaz de Usuario



16

¿Qué editores gráficos ha utilizado para diagramar y/o entender visualmente una infraestructura TI? *

- Lucid Chart
- Visio
- Draw.io
- Otras

17

Describa la utilidad que le aporta usar el o los editores gráficos seleccionados *

Escriba su respuesta

18

¿Qué tipo de interfaz de usuario le haría sentir más cómodo para gestionar la infraestructura de TI en la nube? *

- Consola de Línea de Comandos (CLI)
- Editor visual gráfico como aplicación de escritorio
- Editor visual gráfico como aplicación Web
- Otras

⋮

19

¿Qué dispositivo(s) electrónico(s) usted utilizaría para diseñar e implementar una infraestructura TI? *

- Laptop
- Teléfono Móvil
- Tableta

ANEXO III: Prototipo Interactivo

[Prototipo Interactivo en XD](#)