

# Guía de Enseñanza para Inicios en DEVOPS y Despliegue Continuo en AWS enfocado en las competencias de la sociedad 5.0

Manuel Alejandro Pastrana Pardo; Hugo Armando Ordoñez Erazo; Carlos Alberto Cobos Lozada

**Abstract**— The speed of technologies evolve makes society need to increase the adaptability of its production processes, increasing its agility and effectiveness. Japan, as a pioneer of this proposal, launches society 5.0, focused on the use of technologies to solve social and economic problems. One of the largest sectors with influence on the economy and that stands out for its constant changes, is the software development industry. This article presents a didactic guide that allows the adoption of good development practices, strengthening soft and technical skills through a process of continuous evaluation in students of systems engineering programs. The guide consists of the definition of an architectural model for the construction of a REST API, the definition of a preventive quality model based on DEVOPS, the configuration of the model and the evaluation of the application of the model in a course. The implementation of this guide improves the competitiveness of students in relation to their professional profile, providing the industry with better trained people with the potential to improve current software development processes.

**Index Terms**— DEVOPS, IC, DC, AWS, Teaching guide, society 5.0.

## I. INTRODUCCIÓN

Actualmente la sociedad, a nivel global, enfrenta la necesidad de aumentar la capacidad de adaptar los procesos productivos e incrementar la agilidad con que estos se llevan a cabo, gracias al potencial que las tecnologías han inyectado a la industria [1], [2]. Esto se debe a que las fábricas autónomas, capitalizan no solo las fortalezas y capacidades emergentes de las máquinas inteligentes, sino también porque empoderan a sus operadores (trabajadores o empleados) con nuevas responsabilidades [3].

Con base en lo anterior, muchos países están buscando los mejores medios para aprovechar la oportunidades científicas y tecnológicas con las que hasta el momento se cuenta. La era de la inteligencia artificial, la robótica y la digitalización masiva crean nuevas oportunidades y, para gobiernos progresistas, la posibilidad de reinversión tecnológica. En la Cumbre del G7 en 2016, Japón anunció la *Sociedad 5.0*, un amplio plan de gobierno para abordar los problemas globales y desafíos económicos y sociales internos mediante la integración de nuevas tecnologías (inteligencia artificial, robótica, internet de las cosas, entre otras), promocionando crear industrias futuras, a través del desarrollo y la promoción

de la innovación.

En este sentido, el concepto de *Sociedad 5.0* no solo aborda la economía y la industria, sino también la población, promoviendo la adquisición de destrezas y habilidades enfocadas a tecnologías convergentes, como Sistemas Ciber-Físicos, Tecnologías de la Información e Inteligencia Artificial [4], con el fin de enfrentarse a la resolución de problemas sociales, que en este momento, vienen dados como consecuencia del ritmo acelerado del desarrollo tecnológico que, en ocasiones, resulta en empresas, comunidades y personas rezagadas que no cuentan con las competencias, habilidades o recursos necesarios para enfrentar las transformaciones que acarrearán estos cambios[5][6].

La sociedad 5.0, se fundamenta como respuesta a un entorno que exige el desarrollo de nuevas capacidades en las organizaciones y en las personas, permitiéndoles afrontar estos retos de la manera más efectiva. Para esto se debe tener la capacidad de unir personas y estimularlas a la acción, combinado el conocimiento y los esfuerzos de todos en la búsqueda de una meta [7]. No se trata de tener un plan escrito, sino de algo que se perfecciona y actualiza con la práctica, la cual requiere diálogo e interacción, para que las personas logren apreciar el sentido de lo que se hace y que comprendan que se trata del bien común, enfocándose en trabajar, vivir y organizarse de manera inteligente.

A nivel global una de las industrias que ha experimentado cambios vertiginosos es la del desarrollo de software. En este sentido, según [8], aproximadamente el 95% de estas empresas están utilizando un enfoque ágil para el desarrollo de software. Los métodos ágiles buscan asegurar un estrecho vínculo entre el cliente y los desarrolladores para ayudar a garantizar que el software cumpla las necesidades del mercado. Los métodos ágiles también se esfuerzan en definir un enfoque de puesta en producción ágil. Sin embargo, aunque el ágilismo pueden lograr una cadencia de desarrollo y despliegue de mayor frecuencia, ha surgido un cuello de botella en el que las funciones de operación (Ops), que coordina la versión real del software en las organizaciones, normalmente no está alineado con las funciones de desarrollo (Dev) [9]. Como consecuencia, esto puede llevar a largos retrasos en la entrega o despliegue de las versiones del software para los clientes. Con el fin de resolver esto, en 2007, Patrick Debois impulso una integración más estrecha entre las funciones Dev y Ops que se denomina DevOps [10].

Manuel Pastrana, GrinTic Institución Universitaria Antonio José Camacho, Cali, Colombia, [mapastrana@admon.uniajc.edu.co](mailto:mapastrana@admon.uniajc.edu.co), <https://orcid.org/0000-0002-6506-0659>

Hugo Ordoñez, Information Technology Research Group (GTI), Universidad del Cauca, Popayán, Colombia, [hugoordonez@unicauca.edu.co](mailto:hugoordonez@unicauca.edu.co), <https://orcid.org/0000-0002-3465-5617>

Carlos Cobos, Information Technology Research Group (GTI), Universidad del Cauca, Popayán, Colombia, [ccobos@unicauca.edu.co](mailto:ccobos@unicauca.edu.co), <https://orcid.org/0000-0002-6263-1911>

La conexión de las funciones de desarrollo y las funciones de operación en las empresas de desarrollo de software ha llevado a la creación de nuevos patrones de colaboración y trabajo inteligente entre roles que incluyen Desarrolladores, Arquitectos, Scrum Masters, Propietarios de productos, ingenieros de versiones, entre otros [11].

En el contexto de las empresas de desarrollo de software, la transición a DevOps facilita y motiva a que la función de ingeniería de software se vuelva inteligente por tres razones principales: adaptabilidad, entrega continua e intercambio de información. En primer lugar, la adaptabilidad se considera un indicador de inteligencia. La agilidad implica poder responder rápida y eficazmente a las actuales demandas internas y externas, y ser proactivo en desarrollar y retener mercados frente a una amplia competencia [12]. Por tanto, la adaptabilidad de las personas en los equipos de desarrollo y operaciones es un requisito para trabajar en un entorno multifuncional inteligente que permite considerar y comprender las necesidades de diferentes compañeros de trabajo.

En el proceso de cambio e innovación planteado en la *Sociedad 5.0*, las universidades deben reconocer su papel en la sociedad, como formadoras, pero también como generadoras de nuevos conocimientos [13]. La articulación Universidad-Empresa es un paso importante para la apropiación de conocimientos que brinden a los egresados, las habilidades o competencias necesarias para enfrentarse a los desafíos que conlleva el ritmo cambiante de la industria [14]. En este sentido, en este trabajo se propone una *Guía de Enseñanza para Inicios en DevOps y despliegue continuo en AWS enfocado en las competencias de la sociedad 5.0*.

La guía propuesta, se enfoca en potencializar en los egresados de un programa en ingeniería de sistemas, dos tipos de habilidades denominadas blandas y duras, a través de prácticas dirigidas. Las Habilidades blandas se refieren a los rasgos de personalidad y actitudes que dirigen el comportamiento de una persona, tales como adaptabilidad, motivación, comunicación eficaz, relación interpersonal, organización, independencia, motivación y creatividad. Por otra parte, las habilidades duras se relacionan con conocimientos técnicos específicos, tales como la planificación de los proyectos, el seguimiento de los mismos y el uso de herramientas específicas como lenguajes de programación web, móvil y de escritorio, frameworks de desarrollo y prueba, así como entornos de analítica, inteligencia artificial, entre otros[15].

II. MODELO PEDAGÓGICO

La metodología de aprendizaje adoptada para la implementación de la guía es el modelo de aprendizaje basado en proyectos, buscando con esto que los estudiantes

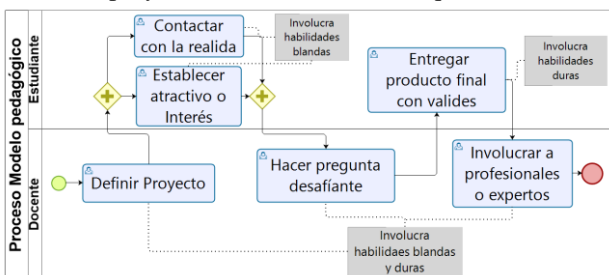


Fig 1. Modelo de pedagogico. Fuente propia

adquieran un rol activo en el desarrollo de proyectos de curso significativos del mundo real, incrementando con ello la motivación académica [16]. El proceso para implementar la metodología se puede ver en la Fig. 1.

Este proceso implementa un conjunto de tareas basadas en la resolución de problemas, que se resuelven a través de un proceso de exploración o creación por parte de los estudiantes, quienes deben trabajar de forma autónoma y con un alto nivel de implicación y cooperación. La resolución del problema identificado en el proyecto culmina con un producto final presentado ante el docente, otros estudiantes del curso e invitados expertos.

Las tareas implementadas en el modelo pedagógico tienen un propósito claramente definido. A continuación, se describe cada una de éstas:

*Definir proyecto*, en esta tarea el docente define o plantea un proyecto autentico centrado en el interés, expectativas, conocimientos y contexto de los estudiantes.

*Conectar con la realidad*, el estudiante debe saber que lo que está aprendiendo es cercano a la realidad y es importante para el desarrollo de sus habilidades. Esta tarea está enfocada en potencializar las habilidades de adaptabilidad, comunicación eficaz, relación interpersonal, organización y creatividad.

*Establecer atractivo o interés*, el estudiante debe entender que el proyecto es una oportunidad de investigar e indagar en soluciones de problemáticas de su propio interés. Esta tarea está enfocada en potencializar las habilidades de adaptabilidad, motivación, organización y creatividad.

*Hacer una pregunta desafiante* el docente debe plantear preguntas para enfocar el proyecto, con el propósito de que los estudiantes encuentren sentido en la resolución del problema. Esta tarea está enfocada en potencializar las habilidades de adaptabilidad, motivación y comunicación eficaz, relación interpersonal, organización, independencia, motivación y creatividad.

*Entregar producto final con validez*, cuando el estudiante termine el desarrollo del proyecto y por ende solucione el problema, debe entregar un producto final valido que contemple la respuesta a las preguntas planteadas. Demostrando así las habilidades adquiridas en el desarrollo del proyecto. Esta tarea está enfocada en potencializar las habilidades de conocimientos técnicos específicos, tales como planificación, tiempo, seguimiento, y sobre el uso de herramientas y lenguajes de programación.

*Involucrar a profesionales o expertos*, el docente proporciona al estudiante una audiencia para presentar el trabajo final del proyecto, lo que incrementa la motivación. Cuando el estudiante presenta su trabajo en público, a sus compañeros o expertos invitados, a menudo se enfrenta a preguntas desafiantes o, a recibir críticas, aspectos que

TABLA I  
MAPEO ENTRE HABILIDADES, TAREAS Y PRÁCTICAS DEVOPS.

Tarea	Habilidades trabajadas	Practicas DevOps
Definir proyecto	Esta parte es definida por el docente.	Se deba integrar todos los conceptos y prácticas de DevOps.
Conectar con la realidad	Blandas: adaptabilidad, comunicación eficaz, relación interpersonal,	Contextualizar al grupo de trabajo sobre el planteamiento de proyecto, versionando los cambios, integrándolos y

	organización y creatividad.	desplegando continuamente, haciendo pruebas unitarias y analizando estáticamente el código para obtener una entrega de alta calidad.
	Duras: habilidades técnicas de programación.	
Establecer atractivo o interés	Blandas: adaptabilidad, motivación y creatividad.  Duras: Versionamiento, IC, DC, pruebas unitarias y análisis estático de código.	Planteamiento del modelo de versionamiento. Planteamiento del correcto uso de IC y DC desde la herramienta de versionamiento. DC orientado a AWS (Amazon Web Services). Exponer impacto del análisis estático de código. Exponer impacto pruebas unitarias en el desarrollo.
Hacer pregunta desafío	Blandas: adaptabilidad, motivación, comunicación eficaz, independencia y creatividad.  Duras: buenas prácticas para desarrollo de software usando las herramientas de DevOps.	¿Cómo obtener la más alta calidad posible durante la ejecución de una iteración de desarrollo utilizando las practicas DevOps relacionadas con versionamiento, integración continua, despliegue continuo, pruebas unitarias y análisis estático de código?
Entregar producto final con valides	Blandas: comunicación eficaz, motivación y creatividad.  Duras: buenas prácticas para desarrollo de software usando las herramientas de DevOps.	Revisión del histórico en la herramienta de versionamiento, IC y DC. Evaluar resultados del análisis estático de código y de la cobertura de pruebas unitarias.
Involucrar a profesionales o expertos	Blandas: comunicación eficaz.  Duras: Mejora en el uso de herramientas y buenas prácticas.	Retroalimentación basada en resultados de la entrega y planteamiento de mejoras.

permiten potencializar sus habilidades de adaptación o de comunicación eficaz. La Tabla I refleja el mapeo entre tareas, habilidades y prácticas DevOps.

### III. MODELO ARQUITECTURAL DE LA GUÍA DE ENSEÑANZA

Al contextualizar al estudiante de las necesidades de la industria y la importancia de la implementación de buenas prácticas para desarrollo de software, el proceso de aprendizaje que se orienta a partir de la guía de enseñanza tiene como objetivo replicar un ambiente de aprendizaje capaz de presentar los mismos retos de los proyectos que día a día afrontan las empresas. Esto genera la necesidad de abordar diferentes conceptos y ponerlos en práctica, a fin resolver las necesidades expuestas en la definición del proyecto, fomentando el aprendizaje autónomo y mejorando tanto habilidades blandas como duras (técnicas).

Así mismo, la guía permite ser un mecanismo de réplica de una configuración de ambiente DevOps para desarrollo de software, que fácilmente puede ser adoptado por profesores y estudiantes a través de diversos cursos orientados al desarrollo de software.

Para la configuración del proyecto del curso se requieren de varios elementos claves para su elaboración, como son el conocimiento previo de un lenguaje de programación orientado a objetos como Java, conocimiento mínimo de diseño de software utilizando UML y capacidad de trabajo en equipo. Estos elementos permitirán al estudiante, garantizar un modelo de desarrollo de software de alta calidad, orientado en un proceso de desarrollo consciente de las buenas prácticas que requiere la industria de desarrollo de software en Colombia y el mundo.

#### A. Arquitectura

El primer paso, consiste en definir el modelo arquitectural del proyecto a construir, identificando claramente a través de un diagrama de paquetes y componentes, los atributos de calidad de la arquitectura y los patrones de diseño que dan solución a las necesidades del proyecto. Para todos los proyectos, se ha modelado una arquitectura multicapa, centrada en un patrón arquitectural llamado backend for frontend. Este modelo, permite que la arquitectura se desacople en dos grandes macro componentes.

El primer macro componente expone las funcionalidades de la lógica de negocios (backend), a través de un API de servicios REST. Este componente posee su propia arquitectura y responde a unos atributos de calidad específicos como son: escalabilidad, mantenibilidad, integralidad, modificabilidad y confiabilidad. Adicionalmente, se expone un elemento clave de autenticación, que hace que los servicios no puedan ser consumidos sin un token de sesión. Este patrón de diseño se conoce como valet key y garantiza la confidencialidad dentro de la arquitectura.

El segundo macro componente, responde a las interfaces graficas de usuario utilizadas por el sistema. Al tener la lógica de negocios expuesta por separado, las pantallas permiten capturar la información que debe ser enviada a los servicios REST, quienes procesan las solicitudes, y devuelven la respuesta que será mostrada en pantalla al usuario. Lo anterior puede verse reflejado en la Fig. 2.

Para efectos de la explicación de la guía, la atención se centrará en el desarrollo del API REST que expone el backend de los proyectos a desarrollar por parte de los estudiantes.

Debido a la versatilidad que posee el lenguaje de programación Java, se selecciona la versión 8 u 11 para el ejercicio. Este lenguaje de programación orientado a objetos posee frameworks de desarrollo bastante aceptados y utilizados en la industria. Un ejemplo de ello es Spring Boot, que será implementado para la construcción de los proyectos a realizar por parte de los estudiantes durante el curso.

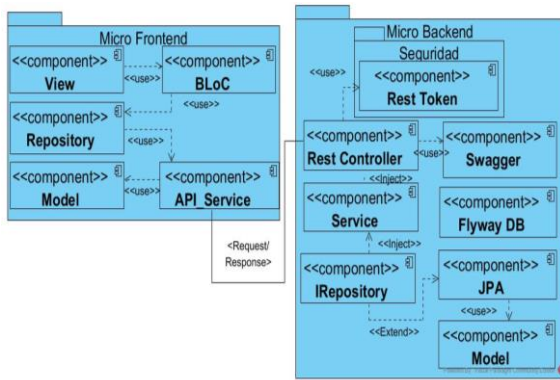


Fig. 2. Arquitectura del proyecto. Fuente propia

### BACKEND - API REST

Bajo la recomendación de la documentación oficial del lenguaje, el patrón arquitectural predominante para la construcción de servicios REST bajo este framework es MVC (Modelo Vista Controlador). Este patrón propone crear una división de las capas que componen el API, dividiendo responsabilidades específicas para cada una de ellas y permitiendo la fácil implementación de otros patrones y frameworks complementarios.

Dentro de este patrón de diseño, las clases denominadas modelos son una representación en memoria de las tablas de la base de datos (persistencia de los objetos).

En la Fig. 3 se encuentra representadas por el componente Model. Las Vistas son la interfaz de comunicación con el exterior, permiten la captura de la información para la transacción y exponen la información de respuesta; se encuentran en la Fig. 3 como el componente RestController y recibe este nombre debido a la etiqueta utilizada por Spring framework para denotar las interfaces de los servicios REST, que son expuestas a través de http y sus métodos. Así mismo, los controladores, que median el paso de la información dentro de la transacción desde la vista al modelo y viceversa se denominan como Service en la Fig. 3, debido a la etiqueta que spring framework indica debe ser utilizada para la clase de lógica.

Adicionalmente, el modelo arquitectural implementa JPA para incrementar la velocidad de desarrollo. Este framework permite que, a partir de una implementación genérica de un DAO, se pueda indicar mediante el modelo de la transacción, que se hereden los métodos clásicos de operaciones de consulta, modificación, inserción y borrado, además de algunas consultas pre parametrizadas como búsqueda por id, entre otras (comúnmente conocidas como operaciones CRUD).

Esta implementación se realiza mediante una clase tipo interfaz que lleva la notación de springboot @repository. En la Fig. 3 se representa en un componente IRepository que extiende de JPA y utiliza al modelo para poder armar las peticiones hacia la base de datos.

Este diseño arquitectural es explicado y entregado a los estudiantes, para el entendimiento de la selección del lenguaje y los frameworks requeridos para la solución. Adicionalmente, se les explica que la variación de la versión del lenguaje no afecta las decisiones de diseño, pero podría ser posible que se requieran algunas adecuaciones a las configuraciones del proyecto.

### B. Buenas Prácticas

Los proyectos incluyen tres buenas prácticas utilizadas en la industria. La primera es el uso del versionador de scripts de base de datos, representado en la Fig. 3 con el componente Flywaydb. Esta práctica permite que los scripts del proyecto que crean la base de datos inserten los datos necesarios para la operatividad inicial e inserten los datos para pruebas, y para que estén administrados dentro del empaquetado que se genera para el despliegue. Así se garantiza que la base de datos sea única, ningún desarrollador la manipula directamente, todos poseen la última versión y al ser desplegada la solución, los cambios realizados a nivel de base de datos serán incluidos durante el despliegue.

La segunda buena práctica, consiste en la inclusión de la documentación del API de servicios REST mediante el framework Swagger, que permite por medio de anotaciones que se colocan dentro de las clases RestController, documentar, los métodos que componen el API, detallando una descripción, los argumentos de entrada y los tipos de respuesta. Dentro de la Fig. 3. Se encuentra como el componente Swagger.

La tercera practica está orientada a incrementar la velocidad de configuración de proyectos realizados con springboot. Para esto, es necesario utilizar la herramienta spring initializr, que se encuentra disponible en el siguiente enlace <https://start.spring.io/>. La Fig. 4 muestra un formulario para configurar la versión de java a utilizar, el gestor de dependencias gradle, las dependencias a utilizar (Swagger, FlywayDB, Spring Boot actuator, para verificar disponibilidad y salud de los servicios, JPA, Spring Boot web para la construcción de servicios REST basado en spring MVC y el driver de la base de datos, que en este caso es PostgreSQL (pero podría ser cualquier base de datos que se requiera). Lo anterior, genera el proyecto bajo el modelo arquitectural y las tecnologías propuestas. Cabe resaltar que tanto las dependencias Swagger, utilizado para generar la documentación del API REST, como JUNIT, utilizado para la realización de pruebas unitarias, deben ser configurados manualmente dentro del proyecto.

Es importante resaltar que el uso de esta herramienta facilita la elaboración del arquetipo siempre y cuando se tenga claro los atributos de calidad de la arquitectura a cubrir mediante la implementación de patrones de diseño que están implementados dentro de los frameworks de desarrollo que componen la solución, por lo que es necesario iniciar siempre por el diseño previo a la generación del arquetipo.



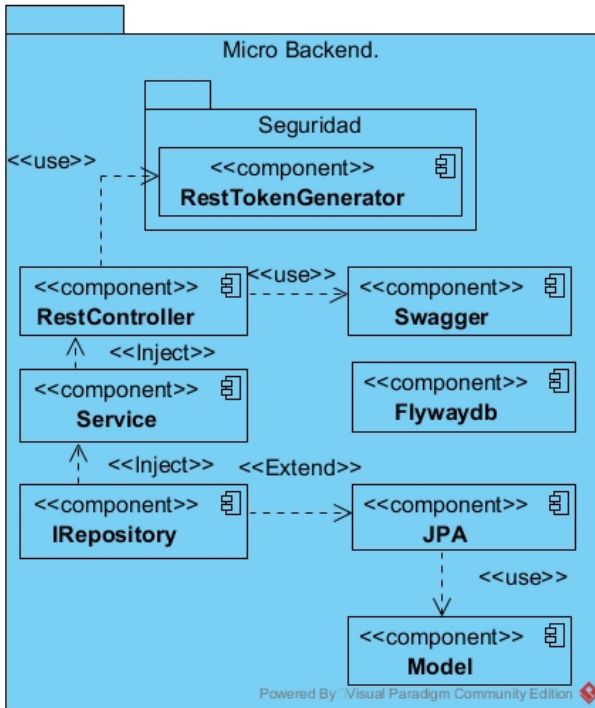


Fig. 3 Arquitectura backend del proyecto. Fuente propia

**Project**

Maven Project  Gradle Project

**Language**

Java  Kotlin  Groovy

**Spring Boot**

2.7.0 (SNAPSHOT)  2.6.3 (SNAPSHOT)  2.6.2  2.5.9 (SNAPSHOT)  2.5.8

**Project Metadata**

Group

Artifact

Name

Description

Package name

Packaging  Jar  War

Java  17  11  8

**Dependencies**

**Spring Web** **WEB**

Build web, including RESTful, applications using Spring MVC. Uses Apache Tomcat as the default embedded container.

**Flyway Migration** **SQL**

Version control for your database so you can migrate from any version (incl. an empty database) to the latest version of the schema.

**PostgreSQL Driver** **SQL**

A JDBC and R2DBC driver that allows Java programs to connect to a PostgreSQL database using standard, database independent Java code.

**Spring Boot Actuator** **OPS**

Supports built in (or custom) endpoints that let you monitor and manage your application - such as application health, metrics, sessions, etc.

Fig. 4. Spring inicializ. Fuente Ejemplo realizado por los autores en <https://start.spring.io/>

### C. Modelo de Calidad Preventiva Basado en DevOps

Una vez generado el arquetipo, se da inicio al entendimiento, configuración y uso de las prácticas sugeridas por DevOps, para lo que es necesario definir el ambiente de calidad preventiva basado en [17].

Aquí, el modelo base implica que, para mantener un proceso continuo de calidad, que permita a los practicantes aprender constantemente de lo que hacen bien y mal dentro del proceso de desarrollo, se debe contar con un versionador de código fuente, como eje principal del proceso, una herramienta de integración y despliegue continuo y un analizador estático de código. De acuerdo con [17], estas herramientas permiten, por el lado del versionamiento, controlar la evolución del producto en todo momento, mantener organizado el código fuente, garantizar que siempre esté disponible para todos los integrantes del equipo, sincronizar los cambios realizados entre todos los colaboradores y recuperarse a fallos en caso de ser necesario. La integración continua (IC), permite verificar si los cambios

que están siendo versionados afectan sintácticamente la unidad desplegable, impidiendo su construcción, o si por el contrario no lo están impidiendo. Así mismo, a este paso se le suma la ejecución de las pruebas unitarias, como mínimo sello de calidad por parte de cada desarrollador. Esto obliga a que el código este probado desde su desarrollo, con una cobertura mínimo del 80%, lo que implica que cada funcionalidad será probada a nivel de los componentes Service y RestController, bajo los distintos caminos posibles de prueba que garanticen la operatividad funcional de la solución, y a su vez el adecuado control de excepciones. Si la unidad desplegable es correctamente construida, posterior a este paso, el archivo generado queda listo para ser desplegado en un servidor de aplicaciones (para ambiente final o de pruebas); a esta práctica se le conoce como despliegue continuo (DC). Por último, este código que pudo ser desplegado, se revisa con un analizador estático de código, que identifica frente a las buenas prácticas del lenguaje de programación, que tan acertado está el producto. Aquí se mide no solo aspectos de vulnerabilidad basado en el top 10 de OWASP (ranking mundial de problemas frecuentes en desarrollo de software que exponen vulnerabilidades de seguridad), también se identifican posibles errores dentro del código o sospechas de que pueden producir errores (Bugs) y también malas prácticas que dificulten el mantenimiento y escalabilidad del proyecto. A esto último se le conoce con el termino de código oloroso (code smell).

#### D. Configuración DevOps

##### 1) Configuración del Versionamiento de la IC y del DC

Para la configuración DevOps propuesta en esta guía, se plantea el versionador Bitbucket. Esta herramienta tiene un servicio en la nube con  $n$  repositorios privados y hasta 5 públicos, limitando el número de colaboradores a 5 por repositorio.

Bitbucket permite realizar el versionamiento del código, permite configurar la integración y el despliegue continuos y la integración con el analizador estático de código sonar, desde su servicio en la nube de sonar cloud. A este concepto se le denomina pipeline o tubería.

Uno de los principales elementos que los estudiantes deben afrontar a la hora de utilizar el modelo de calidad preventiva basado en DevOps, es la forma correcta de utilizar un versionador. Para esta labor, es útil la definición de un esquema como el que se muestra en la Fig. 5, donde el repositorio inicialmente entrega una rama llamada master, dentro de esta rama deberá estar la última versión estable del código fuente con todos los cambios que han pasado por calidad y fueron aprobados. Para ello, se recomienda que la integración de estos cambios no se dé directamente sobre

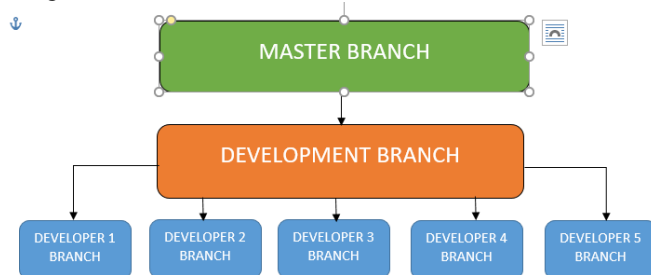


Fig. 5. Modelo de versionamiento utilizado en la guía. Fuente propia  
 master, por el contrario, se debe crear una rama que se desprende de master, donde suceda esto. A esta rama se le

puede nombrar como development o rama de desarrollo. De development se desprenderá una rama por cada desarrollador, de esta manera se puede trabajar con versiones que no alteran el punto de integración de todos los miembros del equipo y solamente cuando el desarrollador está seguro de sus cambios, genera la solicitud de integración de su rama con development. Esto se realiza mediante una solicitud denominada pull request, que es una práctica donde quien solicita los cambios notifica a todo el equipo para que revisen la evolución del entregable, si este es aprobado, los cambios son implementados e integrados dentro de la rama development. Es recomendable estar pendiente de los cambios realizados sobre la rama de integración, para que la rama propia de cada desarrollador este actualizada cuando se ha aprobado un cambio dentro de esta.

Al tener la posibilidad de gestionar los cambios de integración dentro la rama development, el proyecto cuenta con la forma de integrar un script de pasos que permiten disparar de manera automática la IC, el DC y el análisis estático de código al ser aprobado el pull request. Esto se realiza mediante la construcción de un archivo llamado bitbucket-pipelines.yml que se detalla a continuación.

Lo primero a definir es el JDK a utilizar mediante la instrucción `image: openjdk:11`. Luego se indica al script, con que profundidad desea realizar la clonación del repositorio a utilizar para generar el pipeline, mediante la siguiente instrucción:

```
clone:
  depth: full
```

A continuación, se realiza la definición de cache y de pasos a ejecutar, dejando así lista la estructura base del script así:

```
definitions:
  caches:
    sonar: ~/.sonar/cache
  steps:
    - step: &continuous-integration
    - step: &uploadS3
    - step: &continuos-delivery-AWS
```

Por último, se indica que pasos (steps), hará el pipeline cuando el cambio sea subido a la rama que dispara el script:

```
pipelines:
  branches:
    master:
      - step: *uploadS3
      - step: *continuous-delivery-AWS
    development:
      - step: *continuous-integration
      - step: *uploadS3
      - step: *continuos-delivery-AWS
```

Para la definición del paso denominado `&continuous-integration`, se requiere parametrizar los siguientes elementos:

```
steps:
  - step: &continuous-integration
    name: Build (o el nombre que se le quiera dar)
  caches:
    - gradle (selección del gestor de dependencias)
    - sonar (selección herramienta de análisis)
  script:
    - chmod +x gradlew
```

```
- ./gradlew build sonarqube
artifacts:
- build/libs/**
```

La instrucción `chmod +x gradlew` permite otorgar los permisos requeridos sobre el proyecto para ejecutar el gestor de dependencias y poder construir la unidad desplegable, que es el objetivo de la IC. Sin este comando, no es posible realizarlo. El comando: `./gradlew build sonarqube`, ejecuta la revisión del código fuente por parte del analizador estático de código, que en este caso es sonar cloud.

## 2) Configuración del Analizador Estático de Código

En este punto se requiere, por parte de cada equipo de estudiantes, haber creado una cuenta dentro de sonar cloud (<https://sonarcloud.io/>). Una vez creada la cuenta, se ingresa a la plataforma, se crea un proyecto y este otorga una clave (key) para ser configurado dentro del versionador Bitbucket y asociado al repositorio de cada equipo. Esto se realiza en la sección de variables del repositorio (repository variables) como la llave de integración de las dos herramientas.

Realizada la configuración anterior, se procede con la definición del paso que permite generar el proyecto, y subir la unidad desplegable a un servicio de almacenamiento denominado S3 en Amazon, del cual puede ser tomado por un servicio como code-deploy para ser desplegado sin tener que gestionar la infraestructura del servidor. La configuración se detalla a continuación:

```
- step: &uploadS3
  caches:
  - gradle
  name: Upload repository to Bucket
  script:
  (construye el empaquetado)
  - chmod +x gradlew
  - ./gradlew build
  (actualiza el servidor)
  - apt-get update
  (instala herramienta de compresión)
  - apt-get install -y zip
  (comprime el desplegable)
  - zip -j tech.zip build/libs/tech-0.0.1-SNAPSHOT.jar appspec.yml installApp.sh
  (se construye el pipe utilizando AWS code deploy)
  - pipe: atlassian/aws-code-deploy:0.2.10
  variables:
    AWS_DEFAULT_REGION:
    $AWS_DEFAULT_REGION
    AWS_ACCESS_KEY_ID:
    $AWS_ACCESS_KEY_ID
    AWS_SECRET_ACCESS_KEY:
    $AWS_SECRET_ACCESS_KEY
    COMMAND: 'upload'
    APPLICATION_NAME:
    $APPLICATION_NAME
    (se define el archivo a subir)
    ZIP_FILE: 'tech.zip'
    VERSION_LABEL: 'my-app-1.0.0'
    S3_BUCKET: $S3_BUCKET
```

Cuando se está realizando el cargue de los archivos requeridos para el despliegue mediante la instrucción `- zip -j tech.zip build/libs/tech-0.0.1-SNAPSHOT.jar appspec.yml installApp.sh`, se está indicando que hay dos archivos que

poseen la configuración requerida. Por un lado, se encuentra **appspec.yml** que es un archivo de configuración encargado de entregarle a code-deploy las instrucciones para el despliegue. La estructura de este archivo es la siguiente:

```
version:0.0
os: Linux
file:
- source: /
  destination: /app
hooks:
ApplicationStart
- location: ./installApp.sh
timeout:300
```

En la instrucción `file` se le indica al script que el archivo que se ha subido en la ruta raíz debe ser movido a la carpeta `/app`, carpeta que ha sido configurada dentro del EC2 para almacenar las unidades desplegables que serán tomadas por el servidor de aplicaciones. Luego se le indica que ejecute el archivo **installApp.sh**.

Por otra parte, el otro archivo, `installAPP.sh`, es un script que contiene instrucciones para el sistema operativo que permiten realizar el despliegue. Se describe a continuación:

```
sudo killall -9 java (Detiene los procesos de java)
cd /app (abre la ruta donde está el desplegable)
chmod 765 tech-0.0.1-SNAPSHOT.jar (da permisos)
nohup java -jar tech-0.0.1-SNAPSHOT.jar >/dev/null
2>&1 & (instrucción de despliegue, pero no guarda log).
```

Las siguientes variables deben configurarse en la sección de **repository variables** dentro de *Bitbucket*, en el mismo lugar donde se almacenó el token de sonar.

- \$AWS\_DEFAULT\_REGION
- \$AWS\_ACCESS\_KEY\_ID
- \$AWS\_SECRET\_ACCESS\_KEY
- \$APPLICATION\_NAME,

El último paso es más sencillo, que los dos anteriores. A continuación, se detalla:

```
- step: &continuous-delivery-AWS
  name: Deliver to AWS
  deployment: production
  script:
  - pipe: atlassian/aws-code-deploy:0.2.10
  variables:
    AWS_DEFAULT_REGION:
    $AWS_DEFAULT_REGION
    AWS_ACCESS_KEY_ID:
    $AWS_ACCESS_KEY_ID
    AWS_SECRET_ACCESS_KEY:
    $AWS_SECRET_ACCESS_KEY
    APPLICATION_NAME:
    $APPLICATION_NAME
    DEPLOYMENT_GROUP:
    $DEPLOYMENT_GROUP
    S3_BUCKET: $S3_BUCKET
    COMMAND: 'deploy'
    WAIT: 'true'
    VERSION_LABEL: 'my-app-1.0.0'
    IGNORE_APPLICATION_STOP_FAILURES:
    'true'
    FILE_EXISTS_BEHAVIOR: 'OVERWRITE'
```

Este paso utiliza las variables definidas previamente. El punto principal de este paso es la ejecución del comando `deploy`, que dispara la instrucción para ejecutar el servidor de aplicaciones e indica que, si existe un archivo del mismo nombre para el empaquetado dentro del servidor, lo sobrescriba con la última versión que fue subida. Lo anterior se puede realizar gracias a `code-deploy`, que minimiza la cantidad de pasos y configuraciones requeridas para el despliegue.

Gracias al conjunto de pasos mencionados a través del script propuesto, el comportamiento del versionador dependerá de si el cambio de actualización está orientado por un `pull request` a `master` o a `development`. En este caso el ciclo principal que permite ejecutar todas las buenas prácticas de DEVOPS incluidas en este modelo, se disparan al integrar los cambios de los desarrolladores dentro de `development` como indica la Fig. 6.

#### IV. CASO DE ESTUDIO

Para valorar el impacto obtenido al aplicar la guía por parte de los participantes, se recurre a un método de evaluación continua, donde 29 estudiantes del curso de seminario de actualización del programa de ingeniería de sistemas, impartido en la institución Universitaria Antonio José Camacho de Cali, Valle del Cauca, Colombia, son divididos en 5 grupos de 5 integrantes y uno de 4, quienes reciben una revisión y retroalimentación semanal. El periodo de tiempo definido para la recopilación de estos datos fue de 8 semanas en total.

Para iniciar la aplicación de la guía, se exponen ideas de proyectos, acordado entre todos los participantes que se deben cumplir con un conjunto de características comunes dentro de la industria, entre ellos, aspectos de seguridad, mantenibilidad, escalabilidad, confidencialidad y confiabilidad (*Definir proyecto*). Estas características son aterrizadas entre todos los participantes a través de un diagrama UML que detalla las decisiones arquitecturales del aplicativo a construir, permitiendo *contactar con la realidad* el proyecto a través de un proceso organizado de diseño de software. Este punto apoya la organización adecuada de la información que estructura el proyecto, fomenta las relaciones interpersonales y la comunicación eficaz, a modo de que todas las opiniones sean tenidas en cuenta durante la

requiere que sea atractivo para el público objetivo que implementa la guía (*Establecer atractivo o interés*). En este caso se le expone al auditorio que es el versionamiento, su importancia en la industria, un esquema de versionamiento de código fuente y así mismo, se detalla el resto del ambiente DevOps, exponiendo la integración y despliegue continuo, con sus respectivas ventajas y el análisis estático de código, como factor diferenciador en un proceso de autoaprendizaje sobre buenas prácticas para desarrollo de software y mejora continua. Al detallar el ambiente DevOps, se realiza la selección de tecnologías a utilizar. Para este caso se seleccionó el versionador Bitbucket que integra las prácticas de versionamiento IC y DC. Con respecto al ambiente final de despliegue continuo de las soluciones, se hace uso de AWS, debido a la facilidad de configuración de sus servicios en la nube. El versionador seleccionado facilita la integración del análisis estático de código a partir de sonar cloud. Por lo anterior, la configuración del ambiente representa el uso de 3 herramientas técnicas a utilizar dentro del curso: Bitbucket, SonarCloud y AWS. Así mismo, como se mencionó en la sección anterior, la selección de los lenguajes de programación para la construcción se da en el momento del desarrollo de software bajo el lenguaje de programación Java y Flutter.

Con las motivaciones técnicas expuestas y la identificación, por parte del grupo, de los factores diferenciadores técnicos que adquieren con el aprendizaje de estas tecnologías, se procede a iniciar el desarrollo del proyecto y semana a semana *se hace una pregunta desafío* que cada uno de los participantes debe responder durante la *entrega del producto final con validez*. Para la semana 1, la pregunta objetivo es demostrar *¿cómo realizó la configuración del ambiente DevOps y el primer ejemplo de servicio REST?* Para la semana 2, la pregunta objetivo es *¿Cuál fue su implementación de pruebas unitarias para este proyecto?* La semana 3 corresponde a la pregunta *¿Cómo ha sido el proceso de desarrollo de software y la comunicación entre los integrantes del equipo gracias a las herramientas implementadas?* La semana 4 evalúa la pregunta *¿Cuál ha sido la evolución de la calidad del proyecto desde el inicio hasta este punto?* La semana 5 responde a la pregunta *¿Las herramientas le han alertado a este punto sobre posibles malos despliegues para que estos sean corregidos a tiempo, y mostrar las alertas que les dieron las diversas herramientas?* La semana 6 responde a identificar *¿cuál es la mala práctica más frecuente que tienen a nivel de codificación según el analizador estático de código?* La semana 7 propende por *¿identificar cuáles han sido las mejoras que los estudiantes han obtenido sobre las métricas de calidad a partir de subsanar las malas prácticas que han podido cometer en el pasado y que gracias a las herramientas ya no cometen?* La semana 8 evalúa *¿cuál es el resultado de la calidad de la entrega final?*

Durante cada entrega ha estado presente como mínimo un experto del área, quién bajo su óptica da aportes y retroalimentaciones que permitan a los estudiantes alcanzar los objetivos propuestos y refinar aún más sus habilidades (*involucrar un experto del área*).

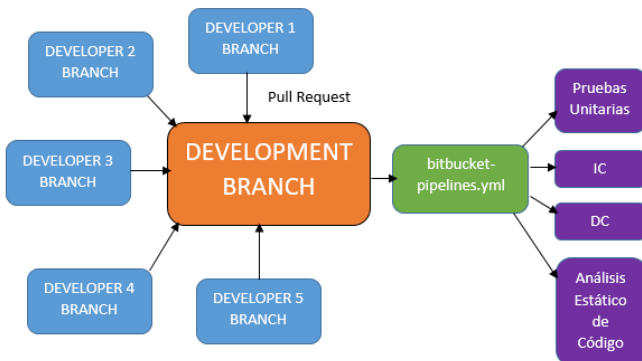


Fig. 6. Modelo DevOps implementado utilizando Bitbucket pipeline, junit java framework, EC2, S3 y code deploy de AWS. Fuente propia

elaboración del diseño de la solución. Así mismo, los participantes deben hacer uso de toda su creatividad al momento de determinar que patrones de diseño son los que mejor se acoplan a la solución.

Una vez el proyecto ha sido contactado con la realidad, se



## V. RESULTADOS

Para evaluar el impacto de la guía sobre el proceso de formación de los estudiantes, se diseñaron dos encuestas que permiten valorar los grupos de habilidades blandas y duras (técnicas) relacionados con la implementación de la guía desde una perspectiva reflexiva y personal de cada uno de los participantes en el ejercicio.

A partir de lo expuesto, en la Tabla II se presentan los resultados obtenidos en la encuesta sobre habilidades blandas con una participación de 19 estudiantes. Se puede observar con respecto a la motivación, que inicialmente el 84,2% de los encuestados, se encuentra altamente motivado con el ejercicio propuesto (porcentaje que corresponde a los valores de 5 y 4 con respecto a la escala de evaluación de la pregunta). Esta motivación se mantiene posterior a la primera evaluación, donde se obtiene el mismo porcentaje y persiste hasta la 4 revisión. Para la segunda propuesta de implementación de la guía los resultados con respecto a la motivación decaen a un 78,9%, debido a que ya conocen el tema, lo dominan y lo ponen en práctica fácilmente. Con respecto a la creatividad, un total de 9 personas consideran que su creatividad es fomentada por el reto, aunque se podría incrementar más, dando una calificación de 4, seguido por 7 personas, que contestan que su creatividad es altamente estimulada por el planteamiento del reto. Por otra parte, 3 personas que ya conocían algunas de las tecnologías perciben que el reto no les potencia la creatividad. Luego de la evolución de las primeras 4 retroalimentaciones, los equipos son nuevamente expuesto a otro reto. Los resultados de volver a preguntar si se estimuló su creatividad indican que 15 personas contestan si y 4 personas indica que Tal vez. No hay percepciones negativas que demuestren que los encuestados no se sientan estimulados a hacer uso de su creatividad para desarrollar el reto.

Con respecto a la comunicación efectiva frente al ejercicio, 16 personas contestan 5, es decir que fue un factor clave para llevar con éxito la implementación de la guía y responder adecuadamente al ejercicio propuesto. Por otra parte, 3 personas responden 4. No hay respuestas con valor 1, 2 o 3. Lo que permite identificar que el ejercicio fomenta esta habilidad blanda. Adicionalmente con respecto a las relaciones interpersonales, en una evaluación inicial el 84,2% de los encuestados (16 personas) responden que el ejercicio fomento esta habilidad, mientras que el restante 15,8% de los encuestados (3 personas) indica que no fue así.

Reflejando que de alguna manera las relaciones interpersonales son un factor que requiere mayor atención y fortalecimiento.

TABLA II  
ENCUESTA DE AUTOEVALUACIÓN SOBRE HABILIDADES BLANDAS.

Pregunta	Resultado
1) Posterior a la presentación de los contenidos y la forma de trabajo, por favor evaluar su motivación con respecto a los presentados, en una escala del 1 al 5, donde 1 es el valor mínimo y 5 el máximo	El 47,4% de los encuestados (9 personas) contestan 5. El 36,8% (7 personas) responde 4 y el restante 15,8% (3 personas) indican 3. No hay respuestas con valor 1 o 2.
2) Posterior a la primera evaluación de las 4 recibidas durante el 1er corte, por favor indique su motivación con respecto a los contenidos, en una	El 47,4% de los encuestados (9 personas) contestan 5. El 36,8% (7 personas) responden 4, el 10,5% (3 personas) indican 3 y el restante

escala del 1 al 5, donde 1 es el valor mínimo y 5 el máximo	5,3% (1 persona) indica 2. No hay respuestas con valor 1.
3) En el primer corte, después de obtener la retroalimentación de la 4ta evaluación, por favor indique su motivación con respecto a los temas vistos, en una escala del 1 al 5, donde 1 es el valor mínimo y 5 el máximo	El 42,1% de los encuestados (8 personas) contestan 5, El 42,1% de los encuestados (8 personas) contestan 4, el 10,5% (2 personas) indican 3 y el restante 5,3% (1 persona) indica 2. No hay respuestas con valor 1.
4) En una escala del 1 al 5, donde 1 es el valor mínimo y 5 el máximo, por favor indique ¿Cuánto cree que su creatividad fue potenciada con el reto del primer curso?	El 36,8% de los encuestados (7 personas) contestan 5, El 47,4% de los encuestados (9 personas) contestan 4, el 10,5% (2 personas) indican 3 y el restante 5,3% (1 persona) indica 2. No hay respuestas con valor 1.
5) ¿Considera Ud. que la comunicación efectiva fue necesaria para lograr el reto? Por favor, evaluar en una escala del 1 al 5 el grado de importancia.	El 84,2% de los encuestados (16 personas) contestan 5, El 15,8% de los encuestados (3 personas) responde 4. No hay respuestas con valor 1, 2 o 3.
6) ¿Considera Ud. que las relaciones interpersonales fueron fomentadas con el ejercicio propuesto durante el primer corte?	El 84,2% de los encuestados (16 personas) contestan si, El 15,8% de los encuestados (3 personas) indica que no.
7) Una vez finalizado el primer corte, se plantea un nuevo proyecto, donde se utiliza el mismo modelo de calidad orientado a DevOps y buenas prácticas. En una escala del 1 al 5, ¿Cómo calificaría su motivación para el segundo corte?	El 42,1% de los encuestados (8 personas) contestan 5, El 36,8% de los encuestados (7 personas) contestan 4, el 15,8% (3 personas) indican 3 y el restante 5,3% (1 persona) indica 2. No hay respuestas con valor 1.
8) Posterior a la primera y segunda evaluación del corte dos, ¿considera Ud. que sus relaciones interpersonales han mejorado significativamente	El 52,6% de los encuestados (10 personas) contestan si, El 10,5% de los encuestados (2 personas) indica que no, el 36% de los encuestados (7 personas) indican Tal vez.
9) ¿Considera Ud. que el segundo ejercicio propuesto fomenta la creatividad?	El 78,9% de los encuestados (15 personas) contestan si, El 21,1% de los encuestados (4 personas) indica que Tal vez.
10) ¿Cómo evalúa su adaptabilidad al curso y sus contenidos durante el primer corte? Utilizar una escala del 1 al 5, donde 1 es el valor mínimo y 5 el máximo.	El 15,8% de los encuestados (3 personas) contestan 5, El 47,4% de los encuestados (9 personas) contestan 4, el 21,1% (4 personas) indican 3 y el restante 15,8% (1 persona) indica 2. No hay respuestas con valor 1.
11) ¿Cómo evalúa su adaptabilidad al curso y sus contenidos durante el segundo corte? Utilizar una escala del 1 al 5, donde 1 es el valor mínimo y 5 el máximo.	El 26,3% de los encuestados (5 personas) contestan 5, El 42,1% de los encuestados (8 personas) contestan 4, el 15,8% (3 personas) indican 3, 10,5% (2 personas) indican 2. El restante 5,3% (1 persona) indica 1.
12) En una escala del 1 al 5, donde 1 es el valor mínimo y 5 el máximo, por favor indicar ¿Qué tanto cree Ud. que los ejercicios propuestos ayudan a la adaptabilidad en la industria?	El 68,4% de los encuestados (13 personas) contestan 5, El 26,3% de los encuestados (5 personas) contestan 4, el 5,3% (1 personas) indica 3, no hay respuestas de valor 2 o 1.

Posterior a esto, con una segunda evaluación se identifica que 10 personas consideran que sus habilidades de relación interpersonal mejoraron, mientras que 2 personas indica que no. Existe un 36% de los encuestados que equivale a 7 personas que indican que tal vez hay una mejoría, pero no es muy fuerte. Corroborando que la guía debe mejorar este aspecto.

Para la evaluación de la habilidad adaptabilidad, se mide en un punto inicial la capacidad que tuvo cada estudiante para adaptarse a los contenidos del curso. Para la primera revisión se encuentra que 3 personas contestan que su adaptabilidad se encuentra muy alta durante las 4 primeras revisiones de avance. 9 encuestados contestan que se logran adaptar

adecuadamente, 4 personas indican que logran adaptarse presentando algún grado de dificultad debido a los nuevos conocimientos que deben adquirir y el restante indica que presentó dificultades para adaptarse. No hay respuestas con valor 1 que denoten que algún participante no pudo adaptarse. Como complemento, se buscó dar una visión de adaptabilidad a la industria, buscando identificar si la implementación de la guía permite preparar mejor a los futuros profesionales de cara a su ejercicio laboral.

TABLA III  
ENCUESTA DE AUTOEVALUACIÓN SOBRE HABILIDADES TÉCNICAS.

Pregunta	Resultado
1) ¿Conocía Ud. la práctica de versionamiento?	El 90% de los encuestados (18 personas) contestan que sí. El 10% (2 personas) que no.
2) ¿Considera Ud. que el uso del versionamiento de código bajo el esquema propuesto en el curso, fomenta sus competencias laborales?	El 100% de los encuestados indica que sí. No hay respuestas con valor no.
3) ¿Conocía Ud. la práctica de integración continua?	El 50% de los encuestados (10 personas) contestan que no, el 45% (9 personas) que sí y el 5% (1 persona) indica que tal vez.
4) ¿Conocía Ud. la práctica de despliegue continuo?	El 65% de los encuestados (13 personas) contestan que no, el 35% (7 personas) que sí. No hay respuestas que indica tal vez.
5) ¿Considera Ud. que estas prácticas de integración y despliegue continuo fomentan factores de competitividad en su perfil profesional?	El 100% de los encuestados (20 personas) contestan que sí.
6) ¿Conocía Ud. la práctica de análisis estático de código?	El 55% de los encuestados (11 personas) contestan si, El 40% de los encuestados (8 personas) indica que no. El restante 5% (1 persona) indica que tal vez.
7) ¿Considera Ud. que estas prácticas le permiten aprender constantemente como mejorar sus habilidades de desarrollo?	El 100% de los encuestados (20 personas) contestan que sí.
8) En una escala del 1 al 5, donde uno es el mínimo valor y cinco el máximo, ¿Qué tan difícil fue la configuración del ambiente DevOps utilizado para el primer corte del curso?	El 25% de los encuestados (5 personas) contestan 5, El 45% de los encuestados (9 personas) indican 4, el 30% de los encuestados (6 personas) indican 3. No hay valores para las calificaciones 2 o 1.
9) En una escala del 1 al 5, donde uno es el mínimo valor y cinco el máximo, ¿Qué tan difícil fue la configuración del ambiente DevOps utilizado para el segundo corte del curso?	El 20% de los encuestados (4 personas) contestan 5, El 20% de los encuestados (4 personas) indican 4, el 15% de los encuestados (3 personas) indican 3. El 20% de los encuestados (4 personas) contestan 2 y El 25% de los encuestados (5 personas) contestan 1.

El 68,4% de los encuestados (13 personas), contestan que el ejercicio fomenta la posibilidad de adaptación al mundo laboral, seguido por un 26,3% (5 personas) quienes contestan que, aunque fomenta esta posibilidad, hay una oportunidad de mejora dentro de la guía que posibilita incrementar estos resultados mediante la inclusión de un ejercicio orientado al despliegue de aplicaciones móviles y no solo web. El 5,3% restante (1 persona) indica que es posible mejorar la guía por la misma razón anterior. No hay respuestas que indiquen que la guía no fomenta en alguna medida esta habilidad.

A continuación, en la Tabla III se presentan los resultados obtenidos por parte de la encuesta sobre habilidades técnicas y que obtuvo una participación de 20 estudiantes, se puede observar con respecto a la práctica de versionamiento que 18

de los participantes ya conocían esta práctica, solamente 2 no la conocían. Un aspecto interesante de este punto es que, si bien conocían algunas herramientas, no conocían modelos de versionamiento diferente a subir su código fuente a la rama master, razón por la que todos coinciden en indicar en la encuesta que el uso del versionamiento de código bajo el esquema propuesto por la guía, fortalece sus competencias laborales y les permite trabajar más organizadamente.

Con respecto a la práctica de integración continua El 50% de los encuestados, es decir 10 personas, contestan que no conocían ni la práctica, ni herramientas que lo permitieran implementar. Solamente el 45% (9 personas) manifiestan que sí conocen esta práctica y al menos una herramienta. El 5% (1 persona) indica que tal vez conocía alguna herramienta, pero no sabía cómo utilizarla. Así mismo, con respecto al despliegue continuo, El 65% de los encuestados (13 personas) contestan que no conocen esta práctica ni como llevarla a cabo. Sin embargo, un 35%, equivalente a 7 personas manifiesta que sí conoce o han realizado algún ejercicio previo sobre este punto. No hay respuestas que indiquen que tal vez se conoce algo del tema. En complemento con este punto, el total de los encuestados manifiesta que estas prácticas fortalecen las habilidades laborales que requieren para ser más competitivos, generando un diferenciador en su perfil profesional.

Con respecto al análisis estático de código, el 55% de los encuestados, es decir 11 personas, contestan que, si conocían esta práctica, mientras que el 40% de los encuestados (8 personas), indica que no la conocían y hay un caso de una persona que corresponde al restante 5% que indica que la conocía, pero no sabía que se llamaba así respondiendo a la encuesta con un tal vez. En este punto se identifica que, si bien hay un buen porcentaje de personas que conocen la práctica, no conocían como interpretar los diversos resultados producto de las mediciones y como estas mediciones fomentan la mejora continua, a partir de la reducción de la deuda técnica dentro de los productos de software. Por lo que todos los encuestados coinciden con que este punto fortalece mucho el perfil profesional.

Adicionalmente se mide la dificultad de la implementación del ambiente DevOps en los dos proyectos. En etapa inicial ningún encuestado considera fácil la implementación (valores de 1 y 2 en la escala utilizada para evaluar este punto). Aquí el 25% de los encuestados, correspondiente a 5 personas contestan el máximo valor posible que es 5, denotando una complejidad significativa. El 45% de los encuestados, es decir 9 personas, indican una complejidad considerable dándole un valor de 4, el 30% de los encuestados restantes, 6 personas, indican una complejidad moderada, calificando con un valor de 3. Posterior a esta primera implementación, nuevamente se utiliza la guía para la implementación de un segundo proyecto, donde se obtiene que El 20% de los encuestados, que corresponde a 4 personas, contestan que existe una complejidad alta en la implementación dando una calificación de 5. Obteniendo una reducción de 5% en esta medida con respecto a la medición anterior. Así mismo, el 20% de los encuestados, 4 personas, indicaron una complejidad considerable dando una calificación de 4. Aquí se obtiene una reducción del 25% con respecto a la primera medición. El 15% de los encuestados (3 personas) indican una complejidad moderada con una calificación de 3,

teniendo una reducción del 15% sobre la medida en comparación con el primer informe. El 20% de los encuestados (4 personas) contestan que existe una complejidad baja 2 y El 25% de los encuestados (5 personas) contestan 1 que existe una complejidad significativamente baja, lo que demuestra un incremento en la facilidad de réplica de la guía con varias ejecuciones.

## VI. DISCUSIÓN DE RESULTADOS

La definición del proyecto de manera colaborativa entre todos los participantes del ejercicio, como ha sido definida en la guía, genera un punto inicial de fortalecimiento de las habilidades blandas. Aquí la comunicación efectiva, el trabajo en equipo, el respeto por las opiniones de los demás y la capacidad de negociación entran en acción de manera contundente. Adicionalmente, para lograr que esta idea sea concretada a través del proyecto, se requiere de contactar el proyecto con la realidad, entrando en colaboración con las habilidades blandas mencionadas, la capacidad exploratoria y autodidacta, la motivación y la capacidad de innovar por medio de la solución.

Así mismo, la motivación por el ejercicio juega un papel clave, fomentando la competitividad y el deseo de aprendizaje del auditorio objetivo. Esto se evidencia en los resultados expuestos en la sección anterior, donde solamente quienes ya conocen algunas de las tecnologías involucradas (un porcentaje relativamente bajo), no sienten demasiada motivación en algunos puntos específicos del ejercicio. Pero al tener variedad en las prácticas DevOps incluidas como el versionamiento, IC, DC y el análisis estático de código, la motivación general no decae, sino que por el contrario es fomentada a través de la pregunta reto en la entrega del producto final con validez, en la cual la retroalimentación de expertos ayuda a incrementar tanto las habilidades blandas como las duras (técnicas).

Adicionalmente, los retos propuestos van de la mano con las necesidades de la industria y acordes a la sociedad 5.0. Por este motivo, la participación de expertos, que están actualmente laborando en la industria, es clave para fomentar el desarrollo del perfil profesional y disminuir la brecha entre academia e industria.

## VII. TRABAJOS RELACIONADOS

Al consultar en Scopus por DevOps y Teaching entre 2016 y 2021 se encontraron un total de 40 trabajos que muestran un interés creciente por el tema, no sólo en el ámbito universitario (formación profesional) sino también en el ámbito empresarial o corporativo.

La Tabla IV lista los 20 trabajos previos más relacionados con la presente investigación. Estos trabajos afirman que muy a pesar de que DevOps es una de las competencias más requeridas en la industria, las instituciones académicas se han demorado en integrar este concepto en los planes de estudio de ingeniería de software, ingeniería informática o carreras afines. Resaltan además que DevOps es difícil de entender para los estudiantes cuando ellos no han tenido experiencia en la industria o en proyectos de tamaño y complejidad significativos. También señalan que configurar las herramientas y flujos de trabajo incrementa el esfuerzo de los educadores y que por eso se necesitan herramientas que ayuden a realizar la configuración de los entornos de práctica

y el uso de evaluaciones automáticas.

La revisión presentada en [19] muestra las estrategias educativas y herramientas más usadas para soportar el proceso de enseñanza aprendizaje de DevOps, enunciando entre ellos el uso (primero las más usadas y de últimas las menos usadas) de ambientes de soporte educacional, lecturas, laboratorios o prácticas, el aprendizaje basado en proyectos, los procesos ágiles, el aprendizaje basado en problemas, el aprendizaje colaborativo, las aulas invertidas (aprender haciendo y no memorizando), el método de contar historias, los tutoriales, el aprendizaje basado en estudio, el aprendizaje experimental y combinaciones de las anteriores.

Por otro lado, la mayoría de los trabajos proponen un curso específico donde buscan que los estudiantes aprendan los conceptos y las habilidades de DevOps, pero sólo en algunos se propone la integración del concepto durante diversos cursos del programa completo de formación para que los estudiantes usen estos conceptos de forma repetitiva. Finalmente, la mayoría de los trabajos incluyen el concepto de la integración continua pero no el de la entrega continua.

El presente trabajo se asemeja a los trabajos previos en que usa un enfoque de un sólo curso y no de varios cursos entretreídos en el plan curricular del programa que buscan que el estudiante repita el uso de los conceptos relacionados con DevOps. En esta propuesta se usa un enfoque basado en proyectos y en resolución de problemas que motiven y desafíen a los estudiantes, buscando además la integración de la Universidad con la Empresa desde varios puntos de vista, el desarrollo de un proyecto conectado con la realidad y la participación de expertos de las empresas como evaluadores de lo desarrollado.

Cuando se realiza un curso basado en la guía propuesta se busca que el estudiante desarrolle habilidades blandas y duras, pero, además, incluyendo el concepto de entrega continua y el uso de buenas prácticas, que son día a día más necesarias para las organizaciones.

## VIII. CONCLUSIONES

Producto de la libre competencia que existe en una industria tan globalizada como la actual, se hace necesario el fomento de la competitividad a partir del uso de las tecnologías de la información, aplicadas en el incremento de la eficiencia y calidad con la que las empresas elaboran sus proyectos. Lo anterior, requiere que la sinergia entre industria y academia forme profesionales con altos conocimientos en nuevas tecnologías, modelos de trabajo, diferentes enfoques de aproximación a la calidad y con la capacidad de resolver los retos que se le presenten. La inclusión de DevOps dentro de los modelos de enseñanza en las carreras de ingeniería de sistemas y afines, permite incrementar estas habilidades necesarias para fomentar la sociedad 5.0. A partir de desarrollos innovadores de alta

TABLA IV  
TRABAJOS RELACIONADOS.

Título	Año	Referencia
DevOps Research-based Teaching Using Qualitative Research and Inter-Coder Agreement	2021	[18]
Analyzing DevOps Teaching Strategies: An Initial Study	2021	[19]
Qualifying Software Engineers Undergraduates in DevOps -	2021	[20]

Challenges of Introducing Technical and Non-technical Concepts in a Project-oriented Course		
Threading DevOps Practices through a University Software Engineering Programme	2020	[21]
Challenges and Recommendations in DevOps Education: A Systematic Literature Review	2020	[22]
Teaching container-based devops practices	2020	[23]
Teaching DevOps in Academia and Industry: Reflections and Vision	2020	[24]
Industry-academy collaboration in teaching devops and continuous delivery to software engineering students: Towards improved industrial relevance in higher education	2019	[25]
An agile-devops reference architecture for teaching enterprise agile	2019	[26]
Teaching devops in corporate environments: An experience report	2019	[27]
Design of a (yet another?) devops course	2019	[28]
Teaching devops at the graduate level: A report from polytech nice sophia	2019	[29]
Devops'18 education panel: Teaching feedback and challenges	2019	[30]
Student participation in HFOSS: Challenges and opportunities	2018	[31]
Quality and testing - new teaching approaches for software engineers	2018	[32]
Enable cloud DevOps approach for industry and higher education	2017	[33]
DevOpsEnvy: An Education Support System for DevOps	2017	[34]
Software engineer education support system ALECSS utilizing devOps tools	2016	[35]
Teaching devops and cloud computing using a cognitive apprenticeship and story- Telling approach	2016	[36]
DevOpsUse for rapid training of agile practices within undergraduate and startup communities	2016	[37]

calidad, se forman profesionales capaces de explorar, dentro de los conceptos aplicados en la guía, el impacto significativo que tiene estas prácticas dentro de su forma de trabajo, permitiendo aprender constantemente sobre cómo elaborar soluciones cada vez más mantenibles y escalables.

Así mismo, el uso de la guía propuesta posibilita una forma simple y efectiva de crear un ambiente de aprendizaje que replica situaciones reales de la industria, donde se pone a prueba diversos elementos de la formación profesional del estudiante para trabajar en conjunto en pro de la solución. Esta guía, no solo replica la construcción de un ambiente DevOps que soporte la evolución del proyecto, también, permite al estudiante apropiarse de diversos conceptos en torno a las buenas prácticas de desarrollo, desde una visión reflexiva y analítica, soportada por las diversas alertas que puede entregar este ambiente y la retroalimentación constante del modelo de evaluación incremental.

Por otra parte, en los resultados se evidencia que las habilidades blandas que son clave en la sociedad 5.0 se

fortalecen, en especial la motivación, la adaptabilidad y la creatividad, pero se requiere una mejora dentro del planteamiento para fortalecer las relaciones interpersonales, probablemente incluyendo un marco de trabajo ágil como scrum, que guíe la interacción de los equipos de trabajo. Cabe resaltar que los encuestados, identifican que la comunicación eficaz es clave como factor de éxito en la ejecución de los retos. Adicionalmente, con respecto a las habilidades técnicas el total de los encuestados manifiestan que el fortalecimiento de sus capacidades técnicas, de alta relevancia para la sociedad 5.0 y que es entregado por la guía, es un factor clave para mejorar su perfil profesional, fomentar la competitividad e incrementar las oportunidades de ingreso a la industria, generando un impacto significativo en su formación profesional.

Para trabajos futuros es necesario explorar este mismo modelo para el desarrollo de soluciones desde el frontend, tanto de aplicaciones web como de aplicación móviles.

#### AGRADECIMIENTOS

Los autores expresan su agradecimiento en primer lugar a la revista IEEE RITA y a su editor en jefe, por el apoyo recibido para la organización de la sección especial sobre desarrollo de competencias de egreso en los ingenieros en las áreas de computación para la sociedad 5.0. Así mismo, agradecer a la Universidad del Cauca, en asociación con la Institución Universitaria Antonio José Camacho. También nuestro agradecimiento a los autores por haber contribuido con sus trabajos a esta sección especial y, por último, a los revisores por ser los garantes de la calidad de los trabajos aceptados.

#### REFERENCIAS

- [1] M. Bhavin, S. Tanwar, N. Sharma, S. Tyagi, and N. Kumar, "Blockchain and quantum blind signature-based hybrid scheme for healthcare 5.0 applications," *J. Inf. Secur. Appl.*, vol. 56, p. 102673, 2021, doi: <https://doi.org/10.1016/j.jisa.2020.102673>.
- [2] T. H. Tran, H. L. Pham, and Y. Nakashima, "A High-Performance Multimem SHA-256 Accelerator for Society 5.0," *IEEE Access*, vol. 9, 2021, doi: [10.1109/ACCESS.2021.3063485](https://doi.org/10.1109/ACCESS.2021.3063485).
- [3] K. Fukuda, "Science, technology and innovation ecosystem transformation toward society 5.0," *Int. J. Prod. Econ.*, 2020, doi: [10.1016/j.ijpe.2019.07.033](https://doi.org/10.1016/j.ijpe.2019.07.033).
- [4] A. Deguchi *et al.*, "What is society 5.0?," in *Society 5.0: A People-centric Super-smart Society*, 2020.
- [5] V. Potočan, M. Mulej, and Z. Nedelko, "Society 5.0: balancing of Industry 4.0, economic advancement and social problems," *Kybernetes*, 2020, doi: [10.1108/K-12-2019-0858](https://doi.org/10.1108/K-12-2019-0858).
- [6] "Industry 4.0 and Society 5.0: Opportunities and Threats," *Int. J. Recent Technol. Eng.*, 2020, doi: [10.35940/ijrte.d8764.018520](https://doi.org/10.35940/ijrte.d8764.018520).
- [7] L. Ellitan, "Competing in the Era of Industrial Revolution 4.0 and Society 5.0," *J. Maksipreneur Manajemen, Koperasi, dan Entrep.*, 2020, doi: [10.30588/jmp.v10i1.657](https://doi.org/10.30588/jmp.v10i1.657).
- [8] Digital.ai, "Version One. (2021). 14th annual state of agile report. (April 2018) (p. 8). Alpharetta, GA, USA. <http://explore.versionone.com/state-of-agile/> versionone-14th-annual-state-of-agile-report," 2021. .
- [9] A. Mishra and Z. Otaiwi, "DevOps and software quality: A systematic mapping," *Computer Science Review*. 2020, doi: [10.1016/j.cosrev.2020.100308](https://doi.org/10.1016/j.cosrev.2020.100308).
- [10] K. Maroukian and S. R. Gulliver, "Leading devops practice and principle adoption," *arXiv*. 2020, doi: [10.5121/csit.2020.100504](https://doi.org/10.5121/csit.2020.100504).
- [11] T. Laukkarinen, K. Kuusinen, and T. Mikkonen, "Regulated software meets DevOps," *Inf. Softw. Technol.*, vol. 97, no. December 2017, pp. 176–178, 2018, doi: [10.1016/j.infsof.2018.01.011](https://doi.org/10.1016/j.infsof.2018.01.011).
- [12] W. P. Luz, G. Pinto, and R. Bonifácio, "Adopting DevOps in the real world: A theory, a model, and a case study," *J. Syst. Softw.*, vol. 157, p. 110384, 2019, doi: [10.1016/j.jss.2019.07.083](https://doi.org/10.1016/j.jss.2019.07.083).



- [13] C. Oonk, J. T. M. Gulikers, P. J. den Brok, R. Wesselink, P. J. Beers, and M. Mulder, "Teachers as brokers: adding a university-society perspective to higher education teacher competence profiles," *High. Educ.*, 2020, doi: 10.1007/s10734-020-00510-9.
- [14] A. O. Moscardini, R. Strachan, and T. Vlasova, "The role of universities in modern society," *Stud. High. Educ.*, 2020, doi: 10.1080/03075079.2020.1807493.
- [15] J. E. Montandon, C. Politowski, L. L. Silva, M. T. Valente, F. Petrillo, and Y. G. Guéhéneuc, "What skills do IT companies look for in new developers? A study with Stack Overflow jobs," *Inf. Softw. Technol.*, vol. 129, p. 106429, Jan. 2021, doi: 10.1016/j.infsof.2020.106429.
- [16] P. Guo, N. Saab, L. S. Post, and W. Admiraal, "A review of project-based learning in higher education: Student outcomes and measures," *Int. J. Educ. Res.*, 2020, doi: 10.1016/j.ijer.2020.101586.
- [17] M. Pastrana, H. Ordoñez, A. Rojas, and A. Ordoñez, "Ensuring Compliance with Sprint Requirements in SCRUM: Preventive Quality Assurance in SCRUM," in *Advances in Intelligent Systems and Computing*, 2019, vol. 924, pp. 33–45, doi: 10.1007/978-981-13-6861-5\_3.
- [18] J. Perez, A. Gonzalez-Prieto, J. Diaz, D. Lopez-Fernandez, J. Garcia-Martin, and A. Yague, "DevOps Research-based Teaching Using Qualitative Research and Inter-Coder Agreement," *IEEE Trans. Softw. Eng.*, 2021, doi: 10.1109/TSE.2021.3092705.
- [19] S. Ferino, M. Fernandes, A. Fernandes, U. Kulesza, E. Aranha, and C. Treude, "Analyzing DevOps Teaching Strategies: An Initial Study," 2021, pp. 180–185, doi: 10.1145/3474624.3477071.
- [20] I. Alves and C. Rocha, "Qualifying Software Engineers Undergraduates in DevOps - Challenges of Introducing Technical and Non-technical Concepts in a Project-oriented Course," 2021, pp. 144–153, doi: 10.1109/ICSE-SEET52601.2021.00024.
- [21] R. Chatley and I. Procaccini, "Threading DevOps Practices through a University Software Engineering Programme," 2020, pp. 90–94, doi: 10.1109/CSEET49119.2020.9206211.
- [22] M. Fernandes, S. Ferino, U. Kulesza, and E. Aranha, "Challenges and Recommendations in DevOps Education: A Systematic Literature Review," 2020, pp. 648–657, doi: 10.1145/3422392.3422496.
- [23] J. Kousa, P. Ihanola, A. Hellas, and M. Luukkainen, "Teaching container-based devops practices," vol. 12128 LNCS. Springer, University of Helsinki, Helsinki, Finland, pp. 494–502, 2020, doi: 10.1007/978-3-030-50578-3\_34.
- [24] E. Bobrov, A. Bucchiarone, A. Capozucca, N. Guelfi, M. Mazzara, and S. Masayagin, "Teaching DevOps in Academia and Industry: Reflections and Vision," vol. 12055 LNCS. Springer, Innopolis University, Innopolis, Russian Federation, pp. 1–14, 2020, doi: 10.1007/978-3-030-39306-9\_1.
- [25] K. Kuusinen and S. Albertsen, "Industry-academy collaboration in teaching devops and continuous delivery to software engineering students: Towards improved industrial relevance in higher education," 2019, pp. 23–27, doi: 10.1109/ICSE-SEET.2019.00011.
- [26] G. B. Ghantous and A. Q. Gill, "An agile-devops reference architecture for teaching enterprise agile," *Int. J. Learn. Teach. Educ. Res.*, vol. 18, no. 7, pp. 128–144, 2019, doi: 10.26803/ijlter.18.7.9.
- [27] M. Mazzara, A. Naumchev, L. Safina, A. Sillitti, and K. Urysov, "Teaching devops in corporate environments: An experience report," vol. 11350 LNCS. Springer Verlag, Innopolis University, Innopolis, Russian Federation, pp. 100–111, 2019, doi: 10.1007/978-3-030-06019-0\_8.
- [28] A. Capozucca, N. Guelfi, and B. Ries, "Design of a (yet another?) devops course," vol. 11350 LNCS. Springer Verlag, Faculty of Science, Technology and Communication, University of Luxembourg, Maison du Nombre, 6, Avenue de la Fonte, Esch-sur-Alzette, 4364, Luxembourg, pp. 1–18, 2019, doi: 10.1007/978-3-030-06019-0\_1.
- [29] B. Benni, P. Collet, G. Molines, S. Mosser, and A.-M. Pinna-Déry, "Teaching devops at the graduate level: A report from polytech nice sophia," vol. 11350 LNCS. Springer Verlag, Université Côte d'Azur, CNRS, I3S, Sophia Antipolis, France, pp. 60–72, 2019, doi: 10.1007/978-3-030-06019-0\_5.
- [30] J.-M. Bruel and M. Jiménez, "Devops'18 education panel: Teaching feedback and challenges," vol. 11350 LNCS. Springer Verlag, University of Toulouse, IRIT, Toulouse, France, pp. 221–226, 2019, doi: 10.1007/978-3-030-06019-0\_17.
- [31] H. J. C. Ellis and G. W. Hislop, "Student participation in HFOSS: Challenges and opportunities," 2018, p. 150, doi: 10.1145/3241815.3241874.
- [32] D. Almog, H. Chasidim, and S. Mark, "Quality and testing - new teaching approaches for software engineers," *World Trans. Eng. Technol. Educ.*, vol. 16, no. 2, pp. 140–145, 2018.
- [33] M. Airaj, "Enable cloud DevOps approach for industry and higher education," *Concurr. Comput. Pract. Exp.*, vol. 29, no. 5, 2017, doi: 10.1002/cpe.3937.
- [34] G. Rong, S. Gu, H. Zhang, and D. Shao, "DevOpsEnvy: An Education Support System for DevOps," 2017, vol. 2017-January, pp. 37–46, doi: 10.1109/CSEET.2017.17.
- [35] M. Ohtsuki, K. Ohta, and T. Kakeshita, "Software engineer education support system ALECSS utilizing devOps tools," 2016, pp. 209–213, doi: 10.1145/3011141.3011200.
- [36] H. B. Christensen, "Teaching devops and cloud computing using a cognitive apprenticeship and story- Telling approach," 2016, vol. 11-13-July-2016, pp. 174–179, doi: 10.1145/2899415.2899426.
- [37] P. de Lange, P. Nicolaescu, R. Klamma, and I. Koren, "DevOpsUse for rapid training of agile practices within undergraduate and startup communities," vol. 9891 LNCS. Springer Verlag, Advanced Community Information Systems (ACIS) Group, RWTH Aachen University, Ahornstr. 55, Aachen, 52056, Germany, pp. 570–574, 2016, doi: 10.1007/978-3-319-45153-4\_65.

**Manuel Alejandro Pastrana Pardo.** Ingeniero de sistemas de la universidad Santiago de Cali (2013), especialista en procesos para desarrollo de software y magister en ingeniería de software por la Universidad San Buenaventura de Cali, Colombia (2016 y 2017 respectivamente). Es profesor ocasional tiempo completo en pregrado o e investigador de la Institución Universitaria Antonio José Camacho. Es coautor de un libro y varios artículos internacionales publicados en revistas indexadas. Ha sido codirector de un graduado de maestría. Sus intereses de investigación incluyen ingeniería de software, smart campus, sistemas para el apoyo a la toma de decisiones, y modelado de procesos de negocios. Actualmente es investigador Asociado de acuerdo con la última clasificación del Ministerio de Ciencias (e-mail: [mapastrana@admon.uniajc.edu.co](mailto:mapastrana@admon.uniajc.edu.co))

**Hugo Armando Ordoñez Erazo.** Ingeniero de sistemas de la Fundación Universitaria San Martín (2004), especialista en gerencia informática de la corporación universitaria Remington (2006), magister en computación de (2011) y doctor en ingeniería telemática (2015) por la Universidad del Cauca. Fue profesor Titular tanto en pregrado como en posgrado e Investigador del Departamento de Ciencias de la Computación de la Universidad de San Buenaventura. Actualmente, se desempeña como profesor de planta de la Universidad del Cauca. Es coautor de tres libros, cinco artículos de JCR y más de treinta artículos de SJR y Scielo. Ha sido tutor de once graduados de maestría. Sus intereses de investigación incluyen Minería de datos, Análisis de datos, Aprendizaje automático, Recuperación de información e Ingeniería de software. Sus principales colaboraciones internacionales son con España, México, Cuba, Ecuador y Brasil. Está clasificado como Investigador Senior de acuerdo con la clasificación del Ministerio de Ciencias. (e-mail: [hugoordonez@unicauca.edu.co](mailto:hugoordonez@unicauca.edu.co))

**Carlos Alberto Cobos Lozada.** Ingeniero de sistemas (1995) y magister en informática (2004) de la Universidad Industrial de Santander y doctor en ingeniería de sistemas y computación de la Universidad Nacional de Colombia sede Bogotá (2014). Ha impartido diversos cursos en pregrado y posgrado, participado en proyectos de investigación financiados por el Ministerio de Ciencias de Colombia y entidades internacionales, dirigido 2 tesis de doctorado, 17 tesis de maestría y más de 60 trabajos de pregrado. Ha sido coautor de 3 libros, 19 artículos JCR y 59 artículos SJR y Scielo. Actualmente es investigador Senior de acuerdo con la última clasificación del Ministerio de Ciencias (e-mail: [ccobos@unicauca.edu.co](mailto:ccobos@unicauca.edu.co)).