



UNIVERSIDAD POLITÉCNICA SALESIANA

SEDE QUITO

CARRERA DE COMPUTACIÓN

IMPLANTACIÓN DE LA HERRAMIENTA COLABORATIVA QUICKLAB PARA
EL CLAUSTRO DOCENTE DE PROGRAMACIÓN DE LA UNIVERSIDAD
POLITÉCNICA SALESIANA

Trabajo de titulación previo a la obtención del

Título de Ingenieros en Ciencias de la Computación

AUTORES: DANIEL ESTEBAN DOMÍNGUEZ RIVADENEIRA

STALIN DANIEL CHIGUANO GUAYNALLA

TUTOR: RODRIGO EFRAÍN TUFÍÑO CÁRDENAS

Quito – Ecuador

2022

CERTIFICADO DE RESPONSABILIDAD Y AUDITORIA DEL TRABAJO DE TITULACIÓN

Nosotros, Stalin Daniel Chiguano Guaynalla con documento de identificación No. 1725049421 y Daniel Esteban Domínguez Rivadeneira con documento de identificación No. 1721529897, manifestamos que:

Somos los autores y responsables del presente trabajo; y, autorizamos a que sin fines de lucro la Universidad Politécnica Salesiana pueda usar, difundir, reproducir o publicar de manera total o parcial el presente trabajo de titulación.

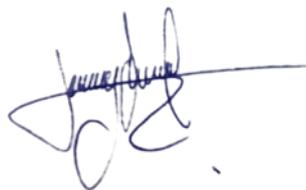
Quito, 19 de septiembre de 2022

Atentamente,



Stalin Daniel Chiguano Guaynalla

1725049421



Daniel Esteban Domínguez Rivadeneira

1721529897

**CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE
TITULACIÓN A LA UNIVERSIDAD POLITÉCNICA SALESIANA**

Nosotros, Stalin Daniel Chiguano Guaynalla con documento de identificación No. 1725049421 y Daniel Esteban Domínguez Rivadeneira con documento de identificación No. 1721529897, expresamos nuestra voluntad y por medio del presente documento cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del proyecto técnico “Implantación de la herramienta colaborativa Quicklab para el claustro docente de programación de la Universidad Politécnica Salesiana”, el cual ha sido desarrollado para optar por el título de: Ingenieros en Ciencias de la Computación, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En concordancia con lo manifestado, suscribimos este documento en el momento que hacemos la entrega del trabajo final en el formato digital a la Biblioteca de la Universidad Politécnica Salesiana.

Quito, 19 de septiembre de 2022

Atentamente,



Stalin Daniel Chiguano Guaynalla

1725049421



Daniel Esteban Domínguez Rivadeneira

1721529897

CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN

Yo, Rodrigo Efraín Tufiño Cárdenas con documento de identificación No 1717646390, docente de la Universidad Politécnica Salesiana, declaro que bajo mi tutoría fue desarrollado el trabajo de titulación: **IMPLANTACIÓN DE LA HERRAMIENTA COLABORATIVA QUICKLAB PARA EL CLAUSTRO DOCENTE DE PROGRAMACIÓN DE LA UNIVERSIDAD POLITÉCNICA SALESIANA**, realizado por Stalin Daniel Chiguano Guaynalla con documento de identificación No 1725049421 y Daniel Esteban Domínguez Rivadeneira y No 1721529897, obteniendo como resultado final el trabajo de titulación bajo la opción Proyecto Técnico que cumple con todos los requisitos determinados por la Universidad Politécnica Salesiana.

Quito, 19 de septiembre de 2022

Atentamente,



Ing. Rodrigo Efraín Tufiño Cárdenas, MsC

1717646390

AGRADECIMIENTO

Este trabajo de titulación está dedicada a mi padre, quien me enseñó que el mejor conocimiento que se puede tener es el que se aprende por sí mismo. También está dedicado a mi madre, quien me enseñó que incluso la tarea más grande se puede lograr si se hace un paso a la vez.

Stalin Daniel Chiguano Guaynalla

El presente trabajo de titulación está dedicado a mi familia, por ser pilar fundamental en mi vida y enseñarme que la dedicación y el trabajo duro cosecha incluso los resultados más difíciles. A mis padres por acompañarme y apoyarme incluso en las decisiones más difíciles, a mis hermanas por el apoyo a lo largo de mi vida y a mi hermana mayor de la que siempre sentí su presencia y apoyo cuando más lo necesitaba.

Daniel Esteban Domínguez Rivadeneira

Contenido

INTRODUCCIÓN.....	1
1.1 Antecedentes.....	1
1.2 Problema de estudio.....	2
1.3 Justificación.....	3
1.4 Objetivos.....	3
1.5 Objetivos específicos.....	4
1.6 Metodología.....	4
CAPÍTULO II.....	6
ACTUALIZACIÓN DE QUICKLAB.....	6
2.1 Plan de mejoras.....	6
2.1.1 Alcance.....	6
2.1.2 Personal involucrado.....	7
2.1.3 Descripción general.....	9
2.1.3.1 Perspectiva del producto.....	9
2.1.3.2 Restricciones.....	9
2.1.4 Requerimientos funcionales.....	9
2.1.4.1 Requerimiento funcional (RF01).....	9
2.1.4.2 Requerimiento funcional (RF02).....	10
2.1.4.3 Requerimiento funcional (RF03).....	11
2.1.4.4 Requerimiento funcional (RF04).....	12

2.1.4.5 Requerimiento funcional (RF05).....	12
2.1.4.6 Requerimiento funcional (RF06).....	13
2.1.4.7 Requerimiento funcional (RF07).....	14
2.1.4.8Requerimiento funcional (RF08).....	15
2.1.4.9 Requerimiento funcional (RF09).....	15
2.1.5 Requerimientos no funcionales.....	16
2.1.5.1 Requerimientos de rendimiento.....	16
2.1.5.2 Requerimientos de seguridad.....	16
2.1.5.3 Requerimientos de Disponibilidad	17
2.1.5.4 Requerimientos de Escalabilidad.....	17
2.1.5.5 Requerimientos de Mantenibilidad.....	17
CAPÍTULO III	18
CONSTRUCCION Y PRUEBAS	18
3.1 Herramientas y tecnologías utilizadas	18
3.1.1 GitHub	19
3.1.2 IntelliJ IDEA	19
3.1.3 PostMan.....	20
3.1.4 Heroku	20
3.1.5 Netlify.....	21
3.1.6 SonarCloud	21
3.1.7 SonarLint	22

3.1.8 React	22
3.1.9 MongoDB	23
2.2.2 Product Backlog.....	23
2.2.3 Sprints	25
2.2.4 Sprint 1 Preparación de entorno.....	25
2.2.4.1 Preparación del entorno	26
2.2.4.2 Herramienta SonarCloud	26
2.2.4.3 Entorno Frontend y Backend.....	27
2.2.5 Sprint 2 Análisis y diseño	27
2.2.6 Sprint 3 Requerimientos RF-01 RF-03 RF-07 RF-08	28
2.2.6.1 Restablecimiento de contraseña.....	29
2.2.6.2 Persistencia de imágenes	32
2.2.6.3 Columna adicional de Docente en cada Ejercicio	33
2.2.6.4 Campo comentario en los ejercicios.....	33
2.2.6.5 Configuración de SonarCloud	35
2.2.7 Sprint 4 Requerimientos RF-02 RF-05	37
2.2.7.1 Filtro por periodo.....	37
2.2.7.2 Corrección al recargar la página.....	39
Capitulo III	40
DESPLIEGUE DE QUICKLAB	40
3.2 Marco de trabajo DevOps	40

3.2.1 Ventajas de DevOps	40
3.3 SonarCloud	41
3.4 DevOps y Quicklab.....	42
3.4.1 Integración continua	42
3.4.2 Despliegue continuo	43
3.5 Despliegue Frontend en entorno de desarrollo	43
3.6 Despliegue Backend en entorno de desarrollo.....	46
3.7 Infraestructura del ambiente de producción.....	46
3.9 Dominios de Quicklab	47
3.9.1 Dominio frontend de Quicklab	47
3.8 Despliegue producción backend	48
3.8.1 Instalación de Node.js y Clonación de proyecto.	48
3.8.2 Nginx	49
3.8.2 Instalación de MongoDB.....	51
3.8.3 Dependencias.....	53
3.9 Despliegue producción Frontend	54
3.9.1 Requisitos previos	54
3.9.2 Dependencias.....	55
3.4 DevOps y SonarCloud	57
3.5.1 Solución de problemas SonarCloud	60
CAPITULO IV	62

OPERATIVIDAD DE LA HERRAMIENTA.....	62
4.1 Creación de usuarios.....	63
4.2 Carga de información.....	64
4.1.1 Carga de ejercicios en MongoDB.....	64
4.1.2 Carga de referencias en MongoDB	65
4.4.2 Carga de temas en MongoDB.....	67
4.3 Generación de guías de laboratorio.....	69
4.4 Manual de usuarios	71
CONCLUSIONES.....	73
RECOMENDACIONES	75
Referencias	76
ANEXOS.....	1

TABLAS

Tabla 1 Desarrollador Frontend y Analista DevOps	7
Tabla 2 Desarrollador Frontend y Backend.....	7
Tabla 3 Coordinador del proyecto	8
Tabla 4 Requerimiento funcional 01	10
Tabla 5 Requerimiento Filtro por Periodos	10
Tabla 6 Requerimiento Persistencia de Imágenes	11
Tabla 7 Requerimiento Información del Dashboard por periodo.....	12
Tabla 8 Requerimiento Corrección de Error al Recargar la Pagina	13
Tabla 9 Requerimiento de Pantalla de Ayuda	13
Tabla 10 Requerimiento de Comentarios en Pantalla Ejercicios.....	15
Tabla 11 Requerimiento de Temas en una Practica.....	15
Tabla 12 Product Backlog	23
Tabla 13 Detalle Sprint 1	25
Tabla 14 Detalle Sprint 2.....	27
Tabla 15 Detalle Sprint 3	28
Tabla 16 Detalle Sprint 4.....	37
Tabla 17 Recursos máquina virtual proporcionado por Data Center Carrera Sistemas UIO	47

ILUSTRACIONES

Ilustración 1 Código de validación correo electrónico de recuperación contraseña.....	29
Ilustración 2 Correo de restablecimiento de contraseña	30
Ilustración 3 Validación de correo electrónico ingresado.....	30
Ilustración 4 Método para enviar pin de recuperación.....	31
Ilustración 5 Código solución a persistencia de imágenes.....	32
Ilustración 6 Campo adicional Docente en tabla Ejercicios	33
Ilustración 7 Código solución a comentario en ejercicios	34
Ilustración 8 Campo comentario en pantalla ejercicios	34
Ilustración 9 Componentes de GitHub Actions	35
Ilustración 10 SonarCloud build YAML	36
Ilustración 11 Columna periodo de ejercicio	38
Ilustración 12 Interfaz gráfica de listado ejercicios	38
Ilustración 13 Validación redirect.....	39
Ilustración 14 Ramas de Desarrollo y Producción.....	43
Ilustración 15 Proyecto en Netlify	44
Ilustración 16 Tarea de ejecución deploy develop.....	44
Ilustración 17 YAML desarrollo Frontend	45
Ilustración 18 YAML desarrollo Backend.....	46
Ilustración 19 Fuente APT con última versión de MongoDB	51

Ilustración 20 Estado del servidor MongoDB.....	52
Ilustración 21 Posible error en librerías	52
Ilustración 22 Levantamiento de Backend en el puerto 1323	53
Ilustración 23 Levantamiento de Servidor PM2	54
Ilustración 24 Build del proyecto	56
Ilustración 25 Funcionamiento de aplicación exitoso.....	57
Ilustración 26 Archivo de Configuración SonarCloud	58
Ilustración 27 Análisis de SonarCloud a la Herramienta Frontend Quicklab v1.0.....	59
Ilustración 28 Análisis de SonarCloud a la Herramienta Backend Quicklab v1.0	59
Ilustración 29 Análisis Final de SonarCloud a la Herramienta Quicklab Frontend.....	60
Ilustración 30 Análisis Final de SonarCloud a la Herramienta Quicklab Backend	60
Ilustración 31 Seguimiento de Resolución de Problemas SonarCloud.....	62
Ilustración 32 Ejercicios cargados en Quicklab.....	65
Ilustración 33 Usuarios de Quicklab	64

RESUMEN

El presente trabajo de titulación presenta el análisis, construcción de mejoras y despliegue de la aplicación colaborativa Quicklab en la infraestructura de la Universidad Politécnica Salesiana, siguiendo un marco de trabajo DevOps que mejore la calidad y mantenibilidad del sistema.

El objetivo del proyecto es mejorar la experiencia de usuario de Quicklab, de tal manera que la usabilidad de la herramienta sea ágil y sencilla para los docentes, implementando la filosofía DevOps que permita el desarrollo en función del tiempo establecido y cumpliendo los estándares de calidad y de funcionamiento en el producto final.

En el desarrollo del proyecto se utilizaron tecnologías basadas en la nube que se acoplen al marco de trabajo DevOps y permitan la implementación de la metodología ágil Scrum y las tecnologías utilizadas en el desarrollo de Quicklab en la versión 1.0.

El desarrollo de mejoras e implementación de técnicas corresponden a los requerimientos funcionales y no funcionales establecidos con el tutor a cargo del proyecto, con la finalidad de que el producto final entregado cubra las principales necesidades del usuario y mejore la mantenibilidad del sistema.

ABSTRACT

This degree work presents the analysis, construction of improvements and deployment of the Quicklab collaborative application in the infrastructure of the Salesian Polytechnic University, following a DevOps framework that improves the quality and maintainability of the system.

The objective of the project is to improve the user experience of Quicklab, in such a way that the usability of the tool is agile and simple for teachers, implementing the DevOps philosophy that allows development based on the established time and meeting quality standards and performance in the final product.

In the development of the project, cloud-based technologies were used that are coupled to the DevOps framework and allow the implementation of the agile Scrum methodology and the technologies used in the development of Quicklab in version 1.0.

The development of improvements and implementation of techniques correspond to the functional and non-functional requirements established with the tutor in charge of the project, so that the final product delivered covers the main needs of the user and improves the maintainability of the system.

INTRODUCCIÓN

Este capítulo presenta la descripción del problema de estudio orientado a la revisión de proyectos relacionados para tener una idea concisa de la problemática y presentar la justificación del proyecto junta mente con sus objetivos.

1.1 Antecedentes

Quicklab es una herramienta colaborativa que automatiza la gestión de guías de laboratorio a los docentes del claustro de programación de la Universidad Politécnica Salesiana, el sistema tiene como objetivo facilitar el proceso de creación de guías en cada periodo académico, las guías son un insumo fundamental de aprendizaje para los estudiantes. La versión 1.0 de Quicklab fue desarrollada por (Cobo et al., 2021) y permitió la automatización del proceso para la creación de guías de laboratorio.

El proceso para crear las guías era manual y repetitivo para cada periodo, por lo cual ocupaba una cantidad considerable de tiempo para el docente, Quicklab se desarrolló para brindar una solución a este proceso, automatizando cada paso que interviene en la creación de la práctica. El alcance del proyecto en su versión 1.0 contemplo el desarrollo y validación del funcionamiento del aplicativo.

Quicklab es el punto de partida del presente proyecto, en el que se llevara a cabo la fase de implantación y validación de su utilidad, así como de mejora en los procesos de desarrollo, actualización y despliegue. Con el principal objetivo de entregar a los usuarios una aplicación que cumpla con la función de gestión de guías de laboratorio y de fácil mantenibilidad a lo largo del tiempo.

A pesar de que la aplicación en su versión inicial cumplió con su objetivo al brindar al docente una alternativa para crear las guías de una manera más sencilla,

debido a que la herramienta fue desplegada en la plataforma “Amazon Web Services”, su funcionamiento se vio afectado por la falta de recursos para mantenerla activa.

1.2 Problema de estudio

En el desarrollo de software es importante considerar la mantenibilidad y escalabilidad del producto final, La mantenibilidad es una de las etapas que más costo representa en el ciclo de vida de un producto de software. Identificar los atributos que afecten la misma en una etapa temprana de desarrollo es crucial para disminuir el costo que representaría corregirlo en una etapa avanzada de desarrollo o mantenimiento. (Francisco Ruiz, 2001)

La escalabilidad es otro factor para tener en cuenta debido a que puede representar el éxito o fracaso de una aplicación, un producto de software puede necesitar distintas clases de recursos, de los mismos pueden identificarse uno o más que desempeñe un valor principal para el correcto funcionamiento de la aplicación, lo que obliga a establecer sobre los mismos un análisis adicional para la planificación de acciones en caso de que los recursos pueden ser modificados.(Adones Farfán & Vega-Zepeda, 2020) Además del desarrollo del producto, se considera una parte importante la correcta documentación del software, los manuales pueden ser enfocados al funcionamiento de la herramienta en base a los diferentes perfiles que existan, y la documentación para levantar y desplegar el sistema en caso de ser necesario.

La disponibilidad de Quicklab se ve afectada por los aspectos mencionados con anterioridad, los costos que representa alojar la aplicación en Amazon Web Services representa un obstáculo por lo que fue importante considerar migrar la aplicación a la propia infraestructura de la Universidad, de esta manera los costos bajarían considerablemente y se mantendría un mejor control de la aplicación.

El deterioro de un sistema web puede verse afectado principalmente al uso de tecnologías obsoletas, versiones sin soporte, malas prácticas utilizadas, fallas de seguridad y la falta de documentación, en el desarrollo dichos problemas pueden no ser identificados con facilidad, y pueden no afectar directamente en esta fase, sin embargo el no tener un estricto control que asegure dichos aspectos representa un coste significativo cuando sea necesario hacer una actualización, para lo cual es importante contar con un análisis automático al código del desarrollador que notifique de cualquier problema que afecte directamente la mantenibilidad de software.

1.3 Justificación

El presente proyecto busca dar una solución ágil a la actual inoperatividad de Quicklab, con ello, se plantea la implementación de mejoras en la funcionalidad del sistema bajo un marco de trabajo DevOps que permita de manera ágil y automatizada cubrir con los principales puntos que pueden afectar el funcionamiento del sistema.

Para cubrir los puntos que presentan problemas en la aplicación y facilitar el uso y mantenibilidad del sistema en el futuro se realizará un análisis completo de la actual herramienta y se levantará la documentación para cada módulo implementado además del manual correspondiente para llevar a cabo el despliegue de la manera más sencilla posible.

1.4 Objetivos

Implantar la herramienta colaborativa Quicklab bajo una arquitectura de microservicios con Docker y un marco de trabajo DevOps sobre la infraestructura de la Carrera de Computación de la Universidad Politécnica Salesiana.

1.5 Objetivos específicos

- Recopilar información bibliográfica de la arquitectura de Docker, marco de trabajo DevOps y el trabajo de titulación de la aplicación actual.
- Implementar mejoras funcionales y actualizar el proyecto a ECMAScript 6.
- Desplegar la aplicación en los servidores de la Universidad Politécnica Salesiana bajo el marco de trabajo DevOps.
- Generar documentación para cada módulo de la aplicación.

1.6 Metodología

El desarrollo del proyecto se lo realizó siguiendo una metodología SCRUM juntamente con el marco de trabajo DevOps, se decidió usar conjuntamente ambas metodologías debido al ajustado tiempo con el que se cuenta. El uso de SCRUM brindó mayor flexibilidad en el desarrollo de las tareas mediante reuniones continuas con el coordinador del proyecto y mediante DevOps se mejoró el proceso con el que se cumplían los requerimientos, automatizando procesos y con análisis frecuente del código que se implementó.

Definición de roles SCRUM

Según la metodología, se asignaron roles al personal involucrado en el análisis y desarrollo del proyecto.

- Product Owner: El rol estuvo a cargo del tutor del proyecto y coordinador del claustro de docentes de programación Rodrigo Efraín Tufiño Cárdenas, quien conoce los procesos llevados a cabo en la gestión de guías de laboratorio y administración de Quicklab.

- Scrum Master: El tutor del proyecto Rodrigo Efraín Tufiño Cárdenas, fue el encargado de la organización del equipo.
- Equipo de desarrollo: Los estudiantes Stalin Daniel Chiguano Guaynalla y Daniel Esteban Domínguez Rivadeneira, encargados del desarrollo de mejoras de Quicklab

Roles DevOps

Según el marco de trabajo DevOps, se asignaron los siguientes roles que participaron en la gestión del proyecto.

- Release manager: El estudiante Stalin Chiguano, encargado de gestionar los releases del proyecto
- Arquitecto de automatización: El estudiante Daniel Domínguez, encargado de la automatización de procesos de entrega y despliegue continuo

Instrumentos y herramientas

Para el levantamiento de los requerimientos, se realizaron reuniones con el Product Owner, como resultado los requerimientos fueron plasmados en historias, mismas que conformaron el Product Backlog y se distribuyeron a cada Sprint Backlog para cada iteración. El seguimiento de las iteraciones se los realizo mediante reuniones semanales y fueron plasmados en la herramienta de GitHub

CAPÍTULO II

ACTUALIZACIÓN DE QUICKLAB

El presente capítulo define los requerimientos funcionales y no funcionales que se aplicaran a la herramienta colaborativa Quicklab para mejorar los procesos administrativos y académicos a los docentes del claustro de programación

2.1 Plan de mejoras

A continuación, se describe los requerimientos funcionales y no funcionales establecidos en las reuniones con el coordinador del proyecto mediante el análisis de la herramienta colaborativa Quicklab

2.1.1 Alcance

El proyecto tiene como objetivo principal implementar un plan de mejoras a la aplicación, que permita mejorar los procesos con los cuales se gestionan las guías de laboratorio y a su vez maximizar la productividad de los docentes al utilizar la aplicación, por otro lado, se mejorara la administración de la herramienta y la mantenibilidad de la aplicación.

Concretamente, la aplicación se desplegará en la infraestructura de la Universidad Politécnica Salesiana y estará disponible para los docentes, coordinadores y administradores, el plan de mejoras incluye la implementación de funcionalidades y el despliegue continuo en ambientes de prueba y producción.

El aplicativo contempla futuras actualizaciones al sistema para lo cual se ha configurado de tal manera que se pueda realizar cambios en un ambiente de pruebas y el análisis automático de código entrante al ambiente de producción.

2.1.2 Personal involucrado

En el desarrollo del proyecto intervinieron directamente tres personas, la primera es el desarrollador Frontend y analista DevOps. Ver Tabla 1.

Tabla 1 *Desarrollador Frontend y Analista DevOps*

Nombre	Daniel Domínguez
Rol	Programador frontend y analista DevOps
Categoría profesional	Ingeniero en Ciencias de la Computación
Responsabilidad	Diseño de interfaces Implementación DevOps Análisis estático de código
Información de contacto	ddominguezr@est.ups.edu.ec

El segundo involucrado es el desarrollador Frontend – Backend, a cargo de la construcción de APIs utilizadas para cumplir con los requerimientos funcionales del sistema y del desarrollo de interfaces.

Tabla 2 *Desarrollador Frontend y Backend*

Nombre	Stalin Chiguano
---------------	------------------------

Rol	Programador frontend y backend
Categoría profesional	Ingeniero en Ciencias de la Computación
Responsabilidad	Diseño de interfaces Administrador de base de datos Construcción de APIs
Información de contacto	schiguano@est.ups.edu.ec

La tercera persona es el coordinador del proyecto, encargada principalmente del seguimiento y orientación a los desarrolladores para cumplir con los objetivos planteados.

Tabla 3 *Coordinador del proyecto*

Nombre	Rodrigo Tufiño
Rol	Seguimiento
Categoría profesional	Ingeniero en Sistemas
Responsabilidad	Coordinador del proyecto
Información de contacto	rtufino@ups.edu.ec

2.1.3 Descripción general

2.1.3.1 Perspectiva del producto. Quicklab está desarrollado para trabajar en un entorno web, lo que permite rapidez y eficacia en su uso, además no requiere de un sistema experto y funciona de manera independiente.

La aplicación no requiere de mayores recursos, sin embargo, requiere de una base de datos para almacenar la información de la aplicación, además de un servidor donde se realizará el despliegue del aplicativo.

2.1.3.2 Restricciones

- El aplicativo puede ser usado dentro de la UPS
- Lenguajes y tecnologías en uso: HTML, JAVASCRIPT, NODE Y REACT.
- Los servidores utilizados deben garantizar las consultas concurrentes.
- El sistema está diseñado bajo un modelo cliente – servidor
- El sistema deberá contar con la documentación necesaria para garantizar su usabilidad y mantenibilidad.

2.1.4 Requerimientos funcionales

A continuación, se describen los requerimientos funcionales, se les asignó una abreviatura que consta de las letras RF y el número de asignación para identificarlos de mejor manera.

2.1.4.1 Requerimiento funcional (RF01).

El requerimiento describe una funcionalidad adicional que permita recuperar la contraseña de los usuarios del sistema, Ver tabla 4.

Tabla 4 *Requerimiento funcional 01*

Identificación del requerimiento	RF01
Nombre del requerimiento	Restablecimiento de contraseña
Características	Los usuarios podrán reestablecer la contraseña de su cuenta mediante el sistema
Descripción del requerimiento	El sistema podrá ser utilizada para reestablecer contraseña mediante el correo electrónico del usuario y con validación mediante código generado aleatoriamente
Requerimiento no funcional	Requerimientos de seguridad
Prioridad del requerimiento	
Alta	

2.1.4.2 *Requerimiento funcional (RF02).*

El requerimiento describe una funcionalidad adicional en el sistema, se busca una mejora en la visualización de los ejercicios con la posibilidad de filtrar por periodos.

Tabla 5 *Requerimiento Filtro por Periodos*

Identificación del requerimiento	RF02
Nombre del requerimiento	Filtro por periodos
Características	Filtro para visualizar ejercicios dependiendo del periodo académico
Descripción del requerimiento	El sistema permitirá filtrar los ejercicios

dependiendo del periodo académico que el usuario escoja.

Requerimiento no funcional

Prioridad del requerimiento

Alta

2.1.4.3 Requerimiento funcional (RF03).

El requerimiento describe una optimización en el proceso de almacenamiento de imágenes utilizadas para la creación de las guías de laboratorio, en la versión 1.0 las imágenes son almacenadas por tiempo limitado.

Tabla 6 *Requerimiento Persistencia de Imágenes*

Identificación del requerimiento	RF03
Nombre del requerimiento	Persistencia de imágenes
Características	El almacenamiento de imágenes en el sistema debe ser indefinido
Descripción del requerimiento	El sistema debe almacenar permanentemente las imágenes ingresadas a la base de datos
Requerimiento no funcional	Requerimiento de Disponibilidad
Prioridad del requerimiento	
Alta	

2.1.4.4 Requerimiento funcional (RF04)

Contempla la implementación de filtros por periodo en el dashboard principal del sistema, representa una ayuda visual al usuario para reconocer con mayor facilidad los datos estadísticos por cada periodo académico.

Tabla 7 *Requerimiento Información del Dashboard por periodo*

Identificación del requerimiento	RF04
Nombre del requerimiento	Información del Dashboard por periodo
Características	El sistema ofrecerá al usuario información general de los ejercicios y practicas correspondientes al docente por cada periodo seleccionado
Descripción del requerimiento	Muestra la información del dashboard correspondiente al periodo seleccionado
Requerimiento no funcional	Usabilidad y Confiabilidad
Prioridad del requerimiento	
Alta	

2.1.4.5 Requerimiento funcional (RF05)

Hace referencia a la corrección de errores presentes al refrescar páginas de la aplicación.

Tabla 8 *Requerimiento Corrección de Error al Recargar la Pagina*

Identificación del requerimiento	RF05
Nombre del requerimiento	Corrección de error al recargar la pagina
Características	El sistema tiene un error al recargar la página, lo que produce una pantalla en blanco
Descripción del requerimiento	El sistema no provocara errores cuando se recarga una pagina
Requerimiento no funcional	Requerimiento de Seguridad y Mantenibilidad
Prioridad del requerimiento	
Alta	

2.1.4.6 *Requerimiento funcional (RF06)*

Corresponde a la implementación de pantallas de ayuda con información relevante de la funcionalidad para cada perfil de usuario de la herramienta.

Tabla 9 *Requerimiento de Pantalla de Ayuda*

Identificación del requerimiento	RF05
Nombre del requerimiento	Pantalla de ayuda
Características	El sistema no cuenta con un apartado de ayuda para manejar la herramienta

Descripción del requerimiento	El sistema mostrara una pantalla de ayuda con las indicaciones del uso del sistema según el perfil de usuario
Requerimiento no funcional	Mantenibilidad y Usabilidad
Prioridad del requerimiento	
Alta	

2.1.4.7 Requerimiento funcional (RF07)

Se refiere a la implementación de una columna adicional en la información de los ejercicios que corresponda al docente que creo el mismo.

Identificación del requerimiento	RF07
Nombre del requerimiento	Columna adicional del Docente en los ejercicios creados
Características	La pantalla de ejercicios no muestra información del docente que lo creo
Descripción del requerimiento	Se requiere de una columna adicional para identificar al docente que le pertenece cada ejercicio
Requerimiento no funcional	
Prioridad del requerimiento	
Alta	

2.1.4.8 *Requerimiento funcional (RF08)*

Hace referencia a la incorporación de un campo para agregar comentarios a los ejercicios creados

Tabla 10 *Requerimiento de Comentarios en Pantalla Ejercicios*

Identificación del requerimiento	RF08
Nombre del requerimiento	Campo comentario en la pantalla Ejercicios
Características	El sistema permitirá al coordinador asignar comentarios a los ejercicios creados por cada docente
Descripción del requerimiento	Se requiere de una columna adicional con la funcionalidad de agregar comentarios.
Requerimiento no funcional	
Prioridad del requerimiento	
Alta	

2.1.4.9 *Requerimiento funcional (RF09)*

El requerimiento representa el cambio en el proceso de asignación de temas a cada práctica.

Tabla 11 *Requerimiento de Temas en una Practica*

Identificación del requerimiento	RF09
Nombre del requerimiento	Una práctica puede tener más de un tema

Características	El sistema debe permitir que una práctica tenga más de un tema asignado.
Descripción del requerimiento	Una práctica puede tener más de un tema asignado, para lo cual el sistema debe permitir hacer más de una asignación
Requerimiento no funcional	Requerimiento de escalabilidad
Prioridad del requerimiento	
Alta	

2.1.5 Requerimientos no funcionales

2.1.5.1 Requerimientos de rendimiento

- Quicklab tendrá una disponibilidad las 24 horas del día, para lo cual es importante cubrir los requisitos para que el funcionamiento no se vea afectado
- Dado que la aplicación genera guías de laboratorio, se espera un máximo pico de transacciones al comenzar un periodo académico. El sistema debe ser capaz de soportar esta carga y ofrecer un buen rendimiento.

2.1.5.2 Requerimientos de seguridad

- Habrá un solo usuario administrador con el que se puede gestionar usuarios y contraseñas
- La herramienta tiene un límite de tiempo de inactividad para precautelar cualquier vulnerabilidad de sesiones activas
- Las contraseñas almacenadas son encriptadas

2.1.5.3 Requerimientos de Disponibilidad

- La aplicación debe garantizar su disponible dentro y fuera de las instalaciones de la Universidad Politécnica Salesiana

2.1.5.4 Requerimientos de Escalabilidad

- El aplicativo debe tener la capacidad para expandir su funcionamiento a diferentes claustros de docentes, por ende, debe ser altamente escalable en recursos y funcionalidad.

2.1.5.5 Requerimientos de Mantenibilidad

- El desarrollo de la herramienta debe contar con las buenas prácticas de programación
- La herramienta debe ser actualizada a ECMAScript 6
- La herramienta debe contar con un análisis estático de código para identificar posibles futuros problemas.
- Para no interferir con el funcionamiento habitual de la herramienta, se deberá establecer entornos de desarrollo y producción

CAPÍTULO III

CONSTRUCCION Y PRUEBAS

Este capítulo detalla el proceso de desarrollo de requerimientos establecidos en la en el plan de mejoras, mediante el uso de herramientas que permitan seguir el marco de trabajo DevOps, garantizando un flujo de trabajado adecuado y la integración continua del código implementado. Adicionalmente se muestra el código correspondiente a la solución de los requerimientos funcionales con el resultado obtenido en la herramienta.

3.1 Herramientas y tecnologías utilizadas

Para el proceso de desarrollo de actualizaciones e implementación de Quicklab se utilizaron varias herramientas y tecnologías que difieren de las utilizadas en la V1.0. La infraestructura en la cual Quicklab fue inicialmente desplegada es “Amazon Web

Services”, servicio que tiene un costo por alojar la herramienta, por tal motivo la versión V1.1 se desplegara en la propia infraestructura de la Universidad Politécnica Salesiana.

Además de la infraestructura necesaria, se utilizaron diversas tecnologías y servicios gratuitos que se detallan a continuación:

3.1.1 *GitHub*

“GitHub es una plataforma de hospedaje de código para el control de versiones y la colaboración. Esto permite que un equipo de desarrollo trabaje conjuntamente en proyectos de manera independiente” (GitHub, 2016).

En otras palabras, la herramienta permite a sus usuarios, gestionar de manera frecuente los cambios que se realizan en el código fuente, además de contar con diferentes funcionalidades como GitHub Actions, lo cual permitió administrar los cambios del proyecto y a la vez, gestionar de mejor manera el DevOps con pruebas y despliegues automatizados.

3.1.2 *IntelliJ IDEA*

Es el entorno de desarrollo integrado que se utilizó para el desarrollo del proyecto, cuenta con varias funcionalidades y configuraciones que permitieron una mejor conexión con el resto de las herramientas utilizadas y mayor habilidad en el control de versiones, además cuenta con una capa gratuita para estudiantes.

Además, maximiza la productividad del desarrollador con la asistencia inteligente de codificación, IntelliJ IDEA aporta con sugerencias dependiendo el contexto del código, de esta manera se encarga de analizar, completar y asistir al desarrollador en diferentes ambientes. (JetBrains s.r.o., 2022)

3.1.3 PostMan

Postman es una herramienta que permite crear peticiones sobre APIs, cuenta con una versión web y escritorio, ofrece un conjunto de funcionalidades para simplificar cada paso del ciclo de vida de una API. (Postman, 2022).

Postman mejora el rendimiento de desarrollo al facilitar pruebas de métodos implementados en backend, el uso de esta herramienta permite una mayor agilidad en probar las peticiones desarrolladas para el uso del Frontend.

3.1.4 Heroku

“Heroku es una plataforma en la nube que permite a las empresas crear, entregar, monitorear y escalar aplicaciones: somos la forma más rápida de pasar de la idea a la URL, evitando todos esos dolores de cabeza de infraestructura” (Salesforce.es, 2022).

De manera general Heroku es una plataforma de servicios (PaaS) basada en la nube y contenedores. Las plataformas como servicio son un conjunto de servicios basados en la nube que facilitan a los desarrolladores el proceso de creación de aplicaciones, además al ser un servicio en la nube, no existe preocupación alguna por temas relacionados a mantenimientos, actualizaciones, etc.

Heroku es usado por desarrolladores en la implementación y administración de aplicaciones web. La plataforma permite flexibilidad y facilidad de uso, ofreciendo a los desarrolladores el camino más corto para llevar sus aplicaciones al mercado. La herramienta fue utilizada para el despliegue de ambiente de desarrollo de Quicklab,

3.1.5 Netlify

Netlify es una plataforma orientada a la eficiencia en el desarrollo web, permite un camino más corto hacia sitios web eficientes, seguros y escalables. Esta desarrollada para atender con eficiencias el movimiento rápido en el desarrollo, y permite la automatización de despliegue para facilitar en lo máximo posible la implementación de un sitio web. (Netlify, 2022)

Se tomo en cuenta la herramienta debido a que Netlify funciona de mejor manera con React, framework utilizado en el desarrollo Frontend. Otro factor considerado al escoger Netlify es la capacidad de automatización de despliegue que ofrece con ayuda de GitHub Actions.

3.1.6 SonarCloud

SonarCloud es un servicio de análisis de código basado en la nube, diseñado para detectar problemas de calidad de código en más de 20 lenguajes de programación, asegurando una mejor mantenibilidad del software, confiabilidad e incluso seguridad en el código, utiliza técnicas de última generación para encontrar problemas y posibles problemas en el código analizado. (SonarCloud, 2022)

Como parte del marco de trabajo DevOps y al ser una plataforma basada en la nube, SonarCloud permite la ejecución de análisis automático de código utilizado herramientas DevOps como GitHub Actions, Azure DevOps, entre otros

Se utilizo SonarCloud como herramienta para mejorar la mantenibilidad, confiabilidad y seguridad de Quicklab.

3.1.7 SonarLint

SonarLint es uno de los productos de SonarSource, esta herramienta es utilizada como primera línea de defensa para mantener un código limpio. Puede identificar los problemas inmediatamente antes de confirmar un cambio en el IDE utilizado, La herramienta es una extensión IDE disponible de forma gratuita y ejecuta un análisis conforme se escribe líneas de código, así como cuando se confirma un cambio al repositorio remoto utilizado. (SonarSource S.A, 2022).

La herramienta permite minimizar el tiempo que el desarrollador tarda en conocer el análisis arrojado por SonarCloud, identificar los problemas existentes y dar la solución respectiva. SonarLint permite corregir los problemas cuando estos aparecen, sin embargo, los problemas que la herramienta puede analizar no son en totalidad los mismos de SonarCloud, razón por la cual es importante utilizar ambas, de manera independiente.

3.1.8 React

React es una librería basada en JavaScript desarrollada por Facebook y permite la creación de interfaces de usuario robustas, React utiliza una estructura basada en componentes y el uso de estados. React permite la manipulación del DOM con el objetivo de renderizar una pequeña parte de la pantalla y evitar recargar la página en su totalidad.

La interfaz gráfica de la herramienta colaborativa se desarrolló utilizando React Hooks, lo cual optimiza el rendimiento de la aplicación en comparación al React basado en clases.

3.1.9 MongoDB

MongoDB es una base de datos NoSQL la cual almacena la información en colecciones y documentos a diferencia de tablas como las tradicionales bases SQL. Quicklab maneja su base de datos NoSQL para optimizar la velocidad y versatilidad de los campos necesarios en cada una de las funcionalidades del sistema.

En el caso de desarrollo se utilizó la herramienta MongoDB Atlas la cual brinda un plan gratuito y no limita su funcionamiento a una instalación local debido a que la herramienta se encuentra alojada en la nube.

2.2.2 Product Backlog

Con el objetivo de identificar los requerimientos, el equipo de desarrollo juntamente con el Product Owner levanto una lista de actividades en base a los requerimientos plasmados. Ver tabla 12

Tabla 12 *Product Backlog*

ID	Actividad	Duración	Estado
1	Levantamiento y análisis de la herramienta colaborativa Quicklab	20	Completado
2	Configuración de herramientas para el desarrollo	10	Completado
2	Levantamiento de Requerimientos	10	Completado
3	Análisis de requerimientos funcionales y no funcionales	40	Completado

4	Revisión bibliográfica	80	Completado
6	Actualización de librerías y estándares de la herramienta Quicklab	60	Completado
7	Desarrollo de los requerimientos funcionales y no funcionales	170	Completado
8	Implementación de la herramienta Quicklab bajo una arquitectura de Docker	50	Completado
9	Levantamiento de máquina virtual para el despliegue	40	Completado
10	Integración de CI/CD (DevOps) y SonarCloud	50	Completado
11	Despliegue de la aplicación en la infraestructura generada	40	Completado
12	Pruebas de Rendimiento, mantenibilidad y usabilidad	40	Completado
13	Elaboración de manuales para la herramienta colaborativa Quicklab	50	Completado
14	Redactar introducción	30	Completado
15	Redactar metodología	50	Completado
16	Redactar desarrollo y resultados	50	Completado
17	Redactar Conclusiones y resumen	40	Completado
	Total	800	Completado

2.2.3 Sprints

Cada sprint representa una semana de trabajo, se realizó una reunión semanal que permitió llevar un mejor control sobre los avances realizados, y una retroalimentación de parte del Product Owner.

Las actividades que se entregaron semanalmente fueron plasmadas en una tabla donde se refleja la tarea completada, el esfuerzo que requirió, así como los responsables de implementarla.

2.2.4 Sprint 1 Preparación de entorno

En el primer sprint se tuvo como entregable el levantamiento y configuración de la herramienta en el entorno de pruebas, y la instalación de las herramientas necesarias para el desarrollo de los requerimientos.

Tabla 13 *Detalle Sprint 1*

Días						3
Backlog ID	Historia	Tema	Estado	Responsable	Esfuerzo	
1	Levantamiento y análisis de la herramienta colaborativa Quicklab	Análisis previo	Completado	Stalin Chiguano, Daniel Domínguez	20	
2	Configuración de herramientas para el desarrollo	Preparación de entorno	Completado	Stalin Chiguano, Daniel	10	

2.2.4.1 Preparación del entorno

Previo al desarrollo se procedió a levantar la herramienta Quicklab y el ambiente para el desarrollo con las herramientas necesarias, siendo estas: WebStorm, IntelliJ, Postman, MongoDB Atlas y SonarCloud para el análisis estático de código.

2.2.4.2 Herramienta SonarCloud

Para asegurar la calidad del producto entregado se utilizó la herramienta SonarCloud para el análisis estático de código, La herramienta basada en la nube permite conocer el estado del código del proyecto, identificar vulnerabilidades de seguridad, bugs y code smells que afecten directamente a la mantenibilidad de la herramienta. SonarCloud considera un problema al código que puede requerir de mantenimiento o reparación, al realizar el análisis la herramienta minimiza los falsos positivos que puedan presentarse. (SonarCloud, 2022)

Los problemas se agrupan en cuatro tipos:

- **Code Smells:** Los problemas son considerados de este tipo cuando no afectan a la funcionalidad del sistema, pero pueden afectar la mantenibilidad del código, detectar code smells en una etapa temprana de desarrollo minimiza la deuda técnica en un sistema.

- **Bugs:** A diferencia de los code smells, este tipo de problema representa un error en el código que afecte negativamente el funcionamiento de una aplicación y afectar directamente a la confiabilidad del código y del sistema en general.
- **Vulnerabilidades:** Problemas en el código que de ser mal utilizados podrían afectar la seguridad en la aplicación
- **Duplicación de código:** SonarCloud detecta código duplicado e informa el porcentaje y líneas que lo contienen, este estándar es utilizado para mejorar la mantenibilidad de un sistema.

2.2.4.3 Entorno Frontend y Backend

Para levantar la herramienta se requiere de la correcta instalación de las tecnologías usadas en la aplicación, para ello fue necesario la instalación de React, y la instalación de Node.js, ambos lenguajes de programación se lo gestionan en el IDE Webstorm, y la administración de librerías mediante el administrador de paquetes NPM.

2.2.5 Sprint 2 Análisis y diseño

En este sprint se realizó el análisis y diseño de los requerimientos funcionales y no funcionales, mismos que fueron revisados con el coordinador, y tuvo como entregable el documento de análisis y diseño con los requerimientos.

Tabla 14 *Detalle Sprint 2*

Días						4
Backlog	Historia	Tema	Estado	Responsable	Esfuerzo	
ID						
3	Análisis y diseño	Análisis	Completado	Stalin Chiguano,	40	

		y diseño		Daniel Domínguez	
3	Requerimientos	Análisis y diseño	Completado	Stalin Chiguano, Daniel Domínguez	10

2.2.6 Sprint 3 Requerimientos RF-01 RF-03 RF-07 RF-08

El Sprint tuvo como entregable el cumplimiento con los requerimientos RF-01, RF-03 RF-07, RF-08 y la configuración del análisis de SonarCloud mediante GitHub Actions

Tabla 15 *Detalle Sprint 3*

Días						6
Backlog	Historia	Tema	Estado	Responsable	Esfuerzo	
ID						
7	Restablecimiento de contraseña	RF-01	Completado	Stalin Chiguano	6	
7	Persistencia de imágenes	RF-03	Completado	Stalin Chiguano	6	
7	Columna adicional de “Docente” en cada Ejercicio	RF-07	Completado	Stalin Chiguano	6	
7	Campo comentario	RF-08	Completado	Stalin Chiguano	6	

en los ejercicios

7	Configuración SonarCloud mediante GitHub Actions	Análisis estático de código	Completado	Daniel Domínguez	6
---	---	--------------------------------------	------------	------------------	---

2.2.6.1 Restablecimiento de contraseña

El proceso para recuperar la contraseña se realizó mediante la generación de un código aleatorio enviado al correo institucional del usuario, el sistema valida el correo electrónico ingresado antes de generar un pin de recuperación.

Ilustración 1 Código de validación correo electrónico de recuperación contraseña

```
const router = express.Router()
router.post(
  path: '/',
  handlers: [
    check( fields: 'correo', message: 'el correo es obligatorio').not().isEmpty(),
  ],
  recoveryController.recuperarContraseña
)
module.exports = router
```

Tomado de la Ilustración 1 Nota: Fragmento de código donde empieza el proceso de restablecimiento de contraseña elaborado por los autores [Enlace a GitHub](#).

Ilustración 2 Correo de restablecimiento de contraseña



Ilustración 3 Validación de correo electrónico ingresado



Si el correo ingresado es válido y corresponde a un usuario del sistema, la aplicación enviará un correo con el pin de autenticación que solicitara para poder ingresar una nueva contraseña, la solución se muestra en la Ilustración 2 y el resultado en el sistema la Ilustración 3

Ilustración 4 Método para enviar pin de recuperación

```
const sendCode = async (email, code, subj, text) => {
  let newHashedCode = "";
  const hashedCode = await bcryptjs.hash(`${code}`, 10);
  if (hashedCode.includes(value: '/')) {
    newHashedCode = hashedCode.replace(/\/g, 'slash');
  } else {
    newHashedCode = hashedCode;
  }
  const mail = {
    from: process.env.EMAIL_USER,
    to: email,
    subject: `${subj}`,
    html: `${text}${code}`,
  }
  const rta = await sendMail(mail);
  rta.hashedCode = newHashedCode;
  return rta;
}
```

Tomado de la Ilustración 4 Nota: Fragmento de código donde empieza el envío del código de restablecimiento por email elaborado por los autores [Enlace a GitHub](#).

Es importante tomar en cuenta que, aunque se cumplió con el desarrollo del requerimiento y se validó el funcionamiento en entorno de pruebas, debido a las restricciones de seguridad con la que cuenta el servidor de alojamiento de la aplicación, no fue posible validar el funcionamiento en el entorno de producción, sin embargo, el sistema cuenta con una opción adicional que permitirá al usuario administrador la gestión de contraseñas.

2.2.6.2 Persistencia de imágenes

La persistencia de imágenes es de vital importancia para el correcto funcionamiento de Quicklab, la aplicación debe asegurar que las imágenes se mantengan almacenadas incluso después de haber sido utilizadas en una guía, para cumplir con el requerimiento se cambió el controlador con el que se cargan las imágenes y se guardan directamente en la base de datos del sistema como lo muestra la Ilustración 4

Ilustración 5 Código solución a persistencia de imágenes

```
app.post( path: '/upload', handlers: (req :... , res : Response<ResBody, Locals> ) => {
  if (!req.files || Object.keys(req.files).length === 0) {
    res.status( code: 400).send( body: 'No se ha enviado un archivo.')
    return
  }

  const {file} = req.files
  const extension = file.name.split('.')
  const nuevoNombre = `${file.md5}.${extension[extension.length - 1]}`

  const uploadPath = `${__dirname}/uploads/${nuevoNombre}`

  file.mv(uploadPath, callback: (err) => {
    if (err) {
      res.status( code: 500).send(err)
    }
    res.json( body: {
      location: `${req.protocol}://${req.get('host')}/images/${nuevoNombre}`,
    })
  })
})
```

Tomado de la Ilustración 5 Nota: Fragmento de código para la persistencia de imágenes elaborado por los autores [Enlace a GitHub](#) .

2.2.6.3 Columna adicional de Docente en cada Ejercicio

Para mejorar la administración del sistema, este requerimiento tiene como objetivo permitir la visualización de una columna adicional en la pantalla de ejercicios, la columna corresponde al docente que creo el mismo. Para cumplir con el requerimiento se añadió una columna adicional en la que se almacenen los docentes, con ello podremos presentar en la tabla de ejercicios haciendo la misma consulta, pero recibiendo el campo adicional.

Ilustración 6 Campo adicional Docente en tabla Ejercicios

```
{
  title: "Docente",
  dataIndex: "docente",
  key: "docente",
  filterMultiple: false,
  render: (docente) => capitalize(`${docente.nombre.split(' ')[0]} ${docente.apellido[0]}.`),
},
```

Tomado de la Ilustración 6 Nota: Fragmento de código para la nueva columna docente en la tabla ejercicios elaborado por los autores [Enlace a GitHub](#).

2.2.6.4 Campo comentario en los ejercicios

Para cumplir con el requerimiento fue necesario crear la columna adicional en donde se almacenarán los comentarios, debido a que la V1.0 no contemplaba una funcionalidad similar, se desarrolló los métodos para guardar, editar y eliminar el comentario desde cero, con ello se realizaron los cambios correspondientes en la pantalla Ejercicios.

Ilustración 7 Código solución a comentario en ejercicios

```
// Guardar en la base de datos
await comentarioModel.save( options: (err, room) => {
  if (err) {
    res.status(400).send({msg: 'Error al insertar en la base de datos'})
    return
  }
  res
    .status(201)
    .json({msg: 'Comentario ingresado con éxito', id: room.id})
})
} catch (error) {
  res.status(500).json({msg: 'hubo un error en el servidor'})
}
```

Tomado de la Ilustración 7 Nota: Fragmento de código para la inserción a la base de datos de un comentario [Enlace a GitHub](#) .

Mediante el método mostrado en la Ilustración 6, se implementó el nuevo campo en la pantalla de ejercicios como lo muestra la Ilustración 7.

Ilustración 8 Campo comentario en pantalla ejercicios

Título: Deduciendo reglas del algebra de boole.
Descripción:
Dificultad: Medio
Tema: Álgebra de boole
Docente creación: Rodrigo Tufiño

Calificación Mi Calificación

✔ El ejercicio no cuenta con calificaciones

Comentarios Mi Comentario

Comentario:
El ejercicio pertenece al tema Álgebra de boole y no tiene solución

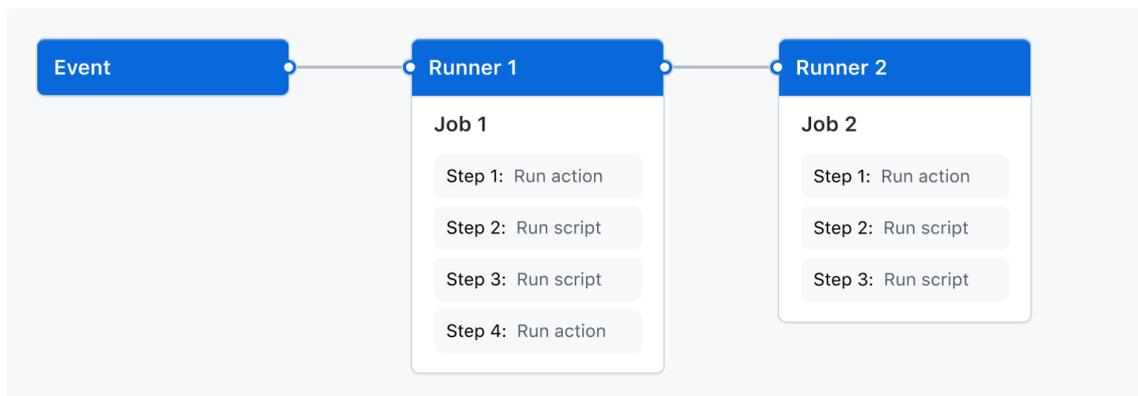
Agrear Comentario

2.2.6.5 Configuración de SonarCloud

Como parte de los requerimientos de Mantenibilidad del sistema y siguiendo el marco de trabajo DevOps, se configuro la herramienta SonarCloud utilizando GitHub Actions.

GitHub Actions en una plataforma DevOps que implementa los conceptos de Integración y Entrega Continua, esto permite la automatización de compilación, análisis, pruebas e implementación de código. A través de esta es posible definir validaciones en cada solicitud de extracción al repositorio, Los componentes que dispone se visualizan en la Ilustración 8.

Ilustración 9 Componentes de GitHub Actions



El flujo de trabajo mostrado es configurado mediante un archivo YAML dentro del repositorio, las condiciones de ejecución pueden controlarse al disparar un evento o de manera automática cada cierto periodo de tiempo, Para la configuración de SonarCloud se utilizará como evento la ejecución de una solicitud de extracción o Pull Request.

En la Ilustración 9 se muestra la implementación de la tarea de compilación que ejecutara el análisis automático siempre que ingrese código a la rama de desarrollo.

Ilustración 10 SonarCloud build YAML

```
name: Build
on:
  push:
    branches:
      - master
  pull_request:
    types: [opened, synchronize, reopened]
jobs:
  sonarcloud:
    name: SonarCloud
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v2
        with:
          fetch-depth: 0 # Shallow clones should be disabled for a better relevancy of analysis
      - name: SonarCloud Scan
        uses: SonarSource/sonarcloud-github-action@master
        env:
          GITHUB_TOKEN: ${ secrets.GITHUB_TOKEN } # Needed to get PR information, if any
          SONAR_TOKEN: ${ secrets.SONAR_TOKEN }
```

Tomado de la Ilustración 10 Nota: Fragmento de código con la configuración de SonarCloud y despliegue continuo con Heroku y Netlify elaborado por los autores [Enlace a GitHub](#).

En la Ilustración anterior se muestra la tarea de compilación de SonarCloud, es importante identificar las tareas que se ejecutan al disparar un evento en la rama de desarrollo:

- Name: El nombre con el que se mostrara en la pestaña de acciones, es un campo opcional.
- On: Representa el tipo de evento que disparara el flujo de trabajo, el valor asignado push representa la ejecución cada vez que el repositorio recibe un cambio en la rama definida en branches.

- **Jobs:** Agrupa todas las acciones que se ejecutaran en el flujo de trabajo, la tarea utilizada para el análisis es SonarCloud, la configuración para esta tarea se muestra en el panel de administración de SonarCloud, es necesario colocar como variable de entorno las credenciales del proyecto.

2.2.7 Sprint 4 Requerimientos RF-02 RF-05

El Sprint tuvo como entregable el cumplimiento con los requerimientos RF-02, RF-05 y la configuración del despliegue continuo en el entorno QA mediante GitHub Actions.

Tabla 16 *Detalle Sprint 4*

Días					5
Backlog ID	Historia	Tema	Estado	Responsable	Esfuerzo
7	Filtro por periodos	RF-02	Completado	Stalin Chiguano, Daniel Domínguez	10
7	Corrección al recargar la pagina	RF-05	Completado	Stalin Chiguano, Daniel Domínguez	10

2.2.7.1 Filtro por periodo

Para el requerimiento, se añadió una columna adicional en la tabla de ejercicios, el campo corresponde al periodo del ejercicio, de esta manera es posible agregar la columna adicional en la interfaz gráfica y colocar el filtro correspondiente mediante la propiedad `sorter`.

Ilustración 11 Columna periodo de ejercicio

```
const columns = [  
  {  
    title: "Periodo",  
    dataIndex: "periodo",  
    defaultSortOrder: "descend",  
    sorter: (a, b) => a.periodo - b.periodo,  
  },  
];
```

Tomado de la Ilustración 11 Nota: Fragmento de código con el nuevo campo en el modelo de ejercicios elaborado por los autores [Enlace a GitHub](#).

El resultado de la implementación se muestra en la Ilustración 11.

Ilustración 12 Interfaz gráfica de listado ejercicios

Listado de ejercicios Crear nuevo ejercicio

	Periodo	Titulo	Tema	Dificultad	Docente	Calificacion	Comentario	Tipo	Archivado
<input type="checkbox"/>	59	Formalización de proposiciones	Álgebra de boole	Medio	Rodrigo T.	☆☆☆ ☆☆		Practica	Sin Archivado
Descripción: Prueba									
<input type="checkbox"/>	60	Prueba	Álgebra de boole	Medio	Rodrigo T.	☆☆☆ ☆☆		Practica	Sin Archivado
Descripción: Prueba									
<input type="checkbox"/>	59	Prueba temas	Algebraico	Medio	Rodrigo T.	☆☆☆ ☆☆		Practica	Sin Archivado

2.2.7.2 Corrección al recargar la página

Para completar el requerimiento, se identificó que el problema al recargar la página era una validación adicional al usar Redirect, para corregir el error se realizó una validación adicional que permite al usuario refrescar la página sin errores.

Ilustración 13 Validación redirect

```
return (  
  <Route  
    {...props}  
    render={({item :...}) =>  
      !autenticado && !cargando ? (  
        <Redirect to="/" />  
      ) : (  
        <Component {...item} />  
      )  
    }  
  />  
)
```

Tomado de la Ilustración 13 Nota: Fragmento de código con la validación de rutas privadas elaborado por los autores [Enlace a GitHub](#).

Capítulo III

DESPLIEGUE DE QUICKLAB

El presente capítulo detalla el proceso de despliegue de Quicklab en el entorno de desarrollo y producción, aplicando las prácticas combinadas de integración y entrega continuas se optimizó el tiempo de desarrollo, de la misma manera con la automatización de análisis de código se mejoró la calidad del código entregado, mantenibilidad y confiabilidad del sistema.

3.2 Marco de trabajo DevOps

El término DevOps resulta de la integración de los términos development (desarrollo) y operations (operaciones), hace referencia a la relación de las personas, procesos y la tecnología con la finalidad de ofrecer un mayor valor al negocio o clientes. (Microsoft, 2022)

En un equipo de desarrollo DevOps permite que los roles que comúnmente trabajan de manera aislada se coordinen y colaboren entre sí para ofrecer un producto de mejor calidad y que brinde confianza al cliente. Al adoptar este marco de trabajo los equipos adquieren una mejor capacidad de respuesta frente a la necesidad de sus clientes, mejora la confiabilidad de los productos entregados y optimiza el tiempo con el que se cumplen los objetivos.

3.2.1 Ventajas de DevOps

Los equipos que adoptan este marco de trabajo, así como sus prácticas y herramientas recomendadas mejoran el rendimiento, calidad y tiempo con el que

entregan un producto, aumentan la cooperación y el entendimiento entre los equipos de la organización.(Microsoft, 2022)

DevOps permite incrementar la producción y despliegue de desarrollo mediante la automatización de procesos mediante sistemas de producción comúnmente conocidos como despliegue e integración continua, de esta manera se logra una mayor flexibilidad y adaptación al entorno, y mayor capacidad de respuesta ante cambios frecuentes.

3.3 SonarCloud

SonarCloud es un servicio basado en la nube para detectar problemas en la calidad de código que puedan afectar la mantenibilidad y seguridad del software, trabaja con 25 lenguajes de programación utilizando técnicas de última generación para el análisis estático de código, lo cual permite encontrar los problemas y posibles futuros problemas en el código desarrollado.

El análisis estático de código se lo realiza sin la ejecución del código, para lo cual SonarCloud permite pruebas automatizadas, revisión manual de problemas y verificación de código. La validación temprana de código permite reducir el riesgo de que problemas lleguen a fases críticas de un proyecto, mejorando en gran medida la calidad de código en producción.

SonarCloud está optimizado para el uso del marco de trabajo DevOps, para su funcionamiento es necesario trabajar con una herramienta DevOps, para el desarrollo del proyecto, la herramienta seleccionada fue GitHub, mediante GitHub Actions podremos ejecutar de manera automática el análisis de código en cada merge o pull request de las ramas previamente establecidas en la herramienta. (SonarCloud, 2022)

3.4 DevOps y Quicklab

El marco de trabajo DevOps fue implementado en el desarrollo de Quicklab v1.1, con el principal objetivo de mejorar la calidad del producto entregado, minimizar el tiempo de desarrollo y mejorar la mantenibilidad del sistema. DevOps fue implementado en tres escenarios

- Integración continua
- Despliegue continuo en ambiente de QA
- Análisis estático de código mediante SonarCloud

3.4.1 Integración continua

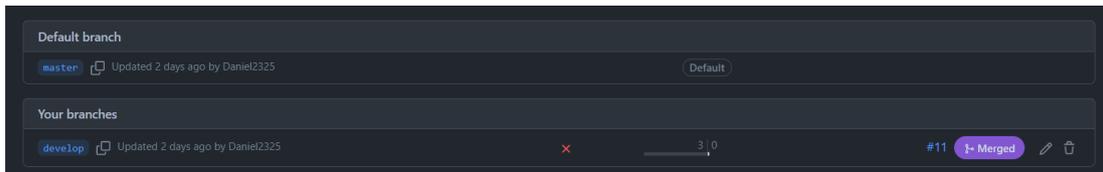
La integración continua es una práctica de desarrollo de software con la que los desarrolladores combinan los cambios en el código en un repositorio central de manera frecuente, con ello es posible un mejor control de versiones y pruebas automáticas al nuevo código.

Quicklab cuenta con un repositorio remoto alojado en la herramienta GitHub, tanto el repositorio Backend y Frontend cuentan con su respectivo repositorio, para mejorar el control de versiones se implementó un flujo de versiones basado en ramas, ambos repositorios cuentan con las siguientes ramas:

- Master: Rama destinada al ambiente de producción, el código que ingrese a esta rama debe ser validado tanto por el administrador del proyecto y el análisis de SonarCloud, el código que ingrese debe estar listo para publicarse y confirmar que no afectara el funcionamiento de la aplicación.
- Develop: Rama destinada al desarrollo, el código que ingresa a esta rama corresponde a nuevas funcionalidades y cambios que deberán ser probados en

un entorno de pruebas que no afecten directamente a la aplicación usada por los usuarios, la rama también debe tener un análisis de código y despliegue continuo para poner a prueba lo implementado.

Ilustración 14 *Ramas de Desarrollo y Producción*



3.4.2 *Despliegue continuo*

La implementación del Despliegue continuo en el proyecto se realizó a través de GitHub Actions, servicio que permite a los desarrolladores preocuparse exclusivamente de la programación. El Despliegue continuo funciona para el Frontend y Backend de maneras similares, en ambos casos se creó un archivo YAML en el que constan las tareas que internamente se realizan después de una solicitud de extracción hasta publicar los cambios en cualquiera de los ambientes.

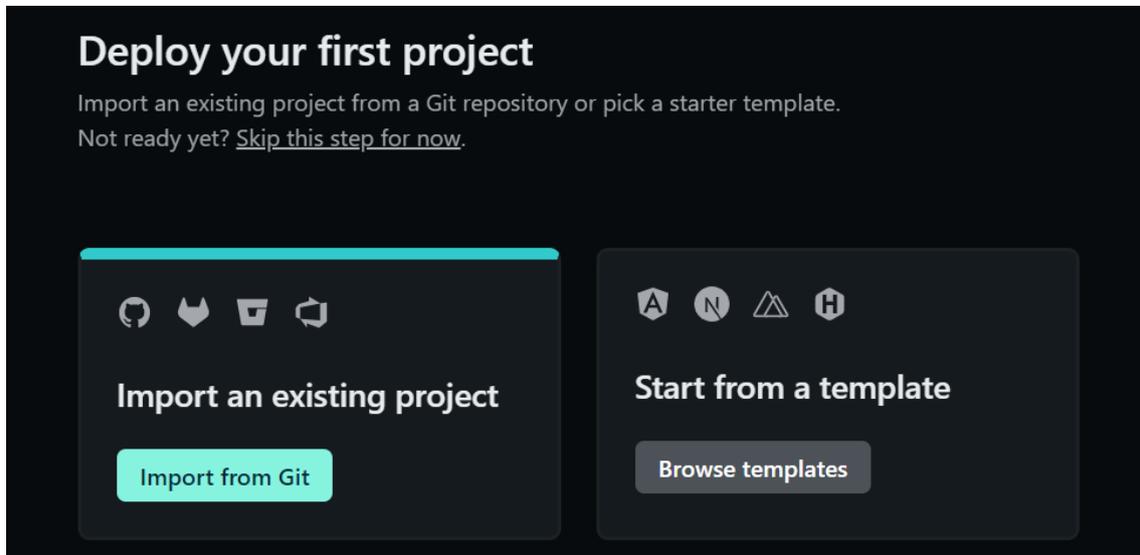
3.5 *Despliegue Frontend en entorno de desarrollo*

El entorno de desarrollo está destinado a pruebas de las desarrolladoras que validen el funcionamiento de funcionalidades nuevas en el sistema, para ello es importante que el proceso de publicación en este ambiente sea en el menor tiempo posible y que no requiera de ninguna acción de parte del desarrollador.

El despliegue del Frontend en este ambiente se lo realizó con la herramienta Netlify, las acciones de despliegue se las implemento en el mismo archivo YAML de SonarCloud, Netlify esta optimizada para realizar despliegues desde un repositorio

remoto, para configurarlo debemos acceder a la página de la herramienta e importar los proyectos de Git como se muestra en la Ilustración 14

Ilustración 15 Proyecto en Netlify



El último paso desde la herramienta es la configuración build que se ejecutara desde GitHub, la configuración se muestra en la Ilustración 15

Ilustración 16 Tarea de ejecución deploy develop

```
build:
  runs-on: ubuntu-latest

  steps:
    - name: trigger netlify build
      run: |
        curl -X POST -d '{}' ${{ secrets.NETLIFY_HOOK }}
```

Tomado de la Ilustración 16 Nota: Fragmento de código con la configuración para el despliegue continuo con Netlify elaborado por los autores [Enlace a GitHub](#).

La tarea de ejecución muestra las tareas que se ejecutan en el despliegue tales que:

- **Runs-on;** Configura el entorno de trabajo a la última versión de Ubuntu, todo el proceso se ejecutará en una máquina virtual de GitHub.
- **Steps:** Agrupación de todos los pasos que se ejecutan en la tarea de compilación
- **Name:** Nombre asignado a la compilación
- **Run:** Comando que generara el despliegue y por ende la nueva versión de la aplicación, es requisito fundamental contar con una clave desde la plataforma de Netlify y un sitio al cual desplegar la aplicación.

La tarea está configurada para que se ejecute siempre que existan cambios de código a la rama develop como se muestra en la Ilustración 16

Ilustración 17 YAML desarrollo Frontend

```
name: Build
on:
  push:
    branches:
      - develop
  pull_request:
    types: [opened, synchronize, reopened]
```

Tomado de la Ilustración 17 Nota: Fragmento de código con la configuración de análisis de código elaborado por los autores [Enlace a GitHub](#).

3.6 Despliegue Backend en entorno de desarrollo

El despliegue del Backend para el ambiente de desarrollo se lo realizo en la plataforma de Heroku, el proceso es similar al explicado en el Frontend, utilizaremos las mismas acciones que el paso anterior, de la misma manera se debe tener disponible un sitio de despliegue y las credenciales que permitan a la tarea de complicación acceder y desplegar la herramienta como se muestra en la Ilustración 17.

Ilustración 18 YAML desarrollo Backend

```
build:
  runs-on: ubuntu-latest
  steps:
    - uses: actions/checkout@v2
    - uses: akhileshns/heroku-deploy@v3.12.12 # This is the action
      with:
        heroku_api_key: ${secrets.HEROKU_API_KEY}
        heroku_app_name: ${secrets.HEROKU_APP_NAME} #Must be unique in Heroku
        heroku_email: ${secrets.HEROKU_EMAIL}
```

Tomado de la Ilustración 18 Nota: Fragmento de código con la configuración de análisis de código elaborado por los autores [Enlace a GitHub](#).

3.7 Infraestructura del ambiente de producción

Quicklab es una herramienta desarrollada en React, framework liviano y que no requiere de mayores recursos para su funcionamiento, del lado del Backend de la misma

manera, para ello se realizó un análisis de recursos necesarios para que la aplicación sea desplegada, de esta manera se estableció la infraestructura con los siguientes detalles

Tabla 17 Recursos máquina virtual proporcionado por Data Center Carrera Sistemas UIO

Sistema operativo	Nombre de maquina	RAM	Disco Duro
Ubuntu 20.04	VM_TESIS_DOMINGUEZ_US	4GB	20GB

Los recursos mostrados en la Ilustración 17 consideran un pico de transacciones al inicio de cada periodo, por lo cual el sistema no deberá verse interrumpido, además el almacenamiento disponible permitirá el almacenamiento de imágenes y datos para los futuros periodos académicos.

3.9 Dominios de Quicklab

Un dominio es un identificador único que tiene un sitio web, se compone de dos partes, la primera parte es el nombre que se le da a la aplicación web, la segunda hace referencia a una ubicación geográfica.(Webempresa, 2022).

En Quicklab se implementaron dominios para facilitar el acceso a la aplicación sin la necesidad de recordar la dirección IP donde está localizada, cuando el usuario accede a un dominio, el DNS es el encargado de interpretar el dominio a la dirección IP correspondiente.

3.9.1 Dominio frontend de Quicklab

Uno de los requisitos no funcionales del proyecto es asegurar la accesibilidad a la herramienta y no restringir su uso en las instalaciones de la Universidad Politécnica

Salesiana. Para lograr el objetivo se añadió un dominio que apunta al servidor donde se encuentra alojada la parte visual de la aplicación. Se puede acceder a la aplicación mediante el siguiente enlace:

- Dominio Quicklab: <https://quicklab.csuioups.org/>

3.9.2 Dominio backend de Quicklab

De la misma manera en la que el dominio apunta a la dirección IP donde se encuentra alojada la aplicación frontend, se lo realiza en el lado backend, para garantizar la conexión cliente servidor. A diferencia del dominio frontend, utilizamos el dominio en el backend para poder establecer conexión desde la pantalla mostrada al usuario, en el siguiente dominio se encuentra localizado el backend de la aplicación.

- Dominio backend: <https://quicklab-api.csuioups.org/>

3.8 Despliegue producción backend

El despliegue de la aplicación se la realizo estableciendo una conexión mediante SSH al servidor, para poder establecer conexión desde una ubicación fuera de las instalaciones de la Universidad Politécnica Salesiana se utilizó un túnel inverso con la herramienta Anydesk.

3.8.1 Instalación de Node.js y Clonación de proyecto.

1. Instalación de Node.js

```
$ sudo curl -sL https://deb.nodesource.com/setup_16.x -o /tmp/nodesource_setup.sh

$ nano /tmp/nodesource_setup.sh

$ sudo bash /tmp/nodesource_setup.sh

$ sudo apt install nodejs
```

2. Clonar repositorio backend

```
$ sudo apt install git-all

$ git clone https://github.com/stalinch98/herramienta-colaborativa-api.git
```

3.8.2 *Nginx*

1. Instalación y configuración Nginx

Nginx funciona como servidor proxy inverso de tal manera que dirige el tráfico web a determinados servidores, para una correcta instalación y configuración es parte importante la actualización de repositorios. Nginx está disponible en los repositorios predeterminados de Ubuntu 20.04. (Aleksandar Kovacević, 2020)

2. Actualización del sistema

```
$ sudo apt-get update

$ sudo apt-get upgrade
```

3. Instalación de Nginx

```
$ sudo apt-get install nginx
$ sudo systemctl enable nginx
```

4. Inicio y status del servidor Nginx

```
$ sudo systemctl start nginx
$ nano /etc/nginx/sites-available/default
```

5. Configuración

```
$ nano /etc/nginx/sites-available/default
$ nano /etc/nginx/sites-available/default
```

Creación de una nueva locación en el archivo de configuración, `proxy_pass` tendrá la IP y el puerto donde se ejecutará la API.

```
location /api {
    proxy_pass https://172.17.42.61:3000;
    proxy_http_version 1.1;
    proxy_set_header Upgrade $http_upgrade;
    proxy_set_header Connection 'upgrade';
    proxy_set_header Host $host;
    proxy_cache_bypass $http_upgrade;
}
```

```
$ systemctl restart nginx
```

3.8.2 Instalación de MongoDB

Los repositorios oficiales de Ubuntu incluyen una versión de MongoDB que no es la última versión LTS disponible. Para obtener una versión más reciente, se debe incluir la fuente APT de MongoDB.org. (Mark Drake, 2020)

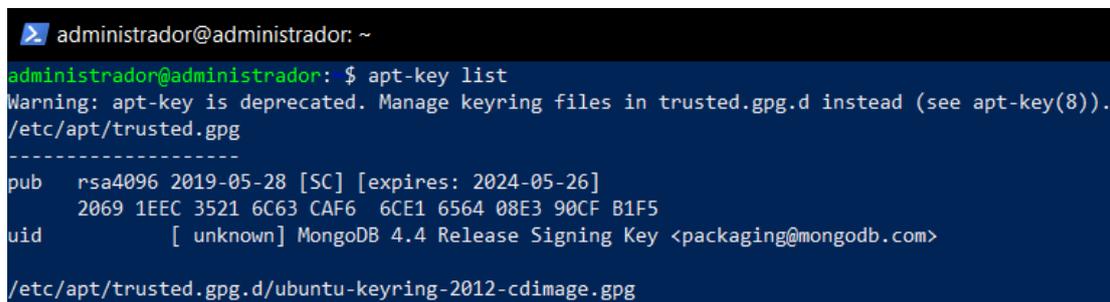
Para esto es necesario ejecutar los siguientes comandos:

1. Agregación de repositorio con última versión de MongoDB

```
$ curl -fsSL https://www.mongodb.org/static/pgp/server-4.4.asc | sudo apt-key add -
```

```
$ apt-key list
```

Ilustración 19 Fuente APT con última versión de MongoDB



```
administrador@administrador: ~
administrador@administrador: $ apt-key list
Warning: apt-key is deprecated. Manage keyring files in trusted.gpg.d instead (see apt-key(8)).
/etc/apt/trusted.gpg
-----
pub   rsa4096 2019-05-28 [SC] [expires: 2024-05-26]
      2069 1EEC 3521 6C63 CAF6 6CE1 6564 08E3 90CF B1F5
uid       [ unknown] MongoDB 4.4 Release Signing Key <packaging@mongodb.com>
/etc/apt/trusted.gpg.d/ubuntu-keyring-2012-cdimage.gpg
```

2. Ruta APT para instalación de MongoDB

El siguiente comando solicitará las credenciales del administrador.

```
$ echo "deb [ arch=amd64,arm64 ]
https://repo.mongodb.org/apt/ubuntu focal/mongodb-org/4.4
multiverse" | sudo tee /etc/apt/sources.list.d/mongodb-org-4.4.list
```

3. Instalación de MongoDB

```
$ sudo apt install mongodb-org
```

4. Iniciar MongoDB automáticamente en el arranque

```
$ sudo systemctl start mongod.service
```

```
$ sudo systemctl status mongod
```

Ilustración 20 Estado del servidor MongoDB

```
administrador@administrador:~$ sudo systemctl start mongod.service
administrador@administrador:~$ sudo systemctl status mongod
● mongod.service - MongoDB Database Server
   Loaded: loaded (/lib/systemd/system/mongod.service; enabled; vendor preset: enabled)
   Active: active (running) since Sun 2022-07-03 18:42:25 UTC; 4s ago
     Docs: https://docs.mongodb.org/manual
   Main PID: 15816 (mongod)
    Memory: 60.2M
      CPU: 568ms
    CGroup: /system.slice/mongod.service
           └─15816 /usr/bin/mongod --config /etc/mongod.conf

Jul 03 18:42:25 administrador systemd[1]: Started MongoDB Database Server.
administrador@administrador:~$ .
```

5. Solución a error en últimas versiones de Ubuntu Server

Ilustración 21 Posible error en librerías

```
requested an impossible situation or if you are using the unstable
distribution that some required packages have not yet been created
or been moved out of Incoming.
The following information may help to resolve the situation:

The following packages have unmet dependencies:
mongodb-org-mongos : Depends: libssl1.1 (>= 1.1.0) but it is not installable
mongodb-org-server : Depends: libssl1.1 (>= 1.1.0) but it is not installable
mongodb-org-shell  : Depends: libssl1.1 (>= 1.1.0) but it is not installable
E: Unable to correct problems, you have held broken packages.
administrador@administrador:~$
```

Los errores que se muestran en la Ilustración 21 ocurren en las últimas versiones de Ubuntu, para corregirlo se deben seguir los siguientes comandos:

```
$ echo "deb http://security.ubuntu.com/ubuntu impish-security
main" | sudo tee
$ /etc/apt/sources.list.d/impish-security.list
$ npm install package.json
```

3.8.3 Dependencias

1. Instalación de dependencias

```
$ cd herramienta-colaborativa-api/
$ npm install package.json
```

```
$ sudo apt-get update
$ sudo apt-get install libssl1.1
```

2. Eliminar repositorio por seguridad

```
$ Rm /etc/apt/sources.list.d/impish-security.list
$ sudo apt install mongodb-org
$ systemctl restart nginx
```

Ilustración 22 Levantamiento de Backend en el puerto 1323

```
administrador@administrador:~/herramienta-colaborativa-api$ node server/server.js
escuchando en 1323
DB Conectada
```

3. Instalación y ejecución de PM2

```
$ sudo npm install pm2@latest -g
$ pm2 start herramienta-colaborativa-api/server/server.js
$ pm2 startup systemd
```

En el siguiente comando la ruta administradora hace referencia al usuario del servidor.

```
sudo env PATH=$PATH:/usr/bin /usr/lib/node_modules/pm2/bin/pm2
startup systemd -u administrador --hp /home/administrador
```

4. Guardar la lista de procesos

```
$ pm2 save
$ systemctl restart pm2-administrador
```

Ilustración 23 *Levantamiento de Servidor PM2*

```
administrador@administrador:~$ systemctl start pm2-administrador
==== AUTHENTICATING FOR org.freedesktop.systemd1.manage-units ===
Authentication is required to start 'pm2-administrador.service'.
Authenticating as: administrador
Password:
==== AUTHENTICATION COMPLETE ===
Job for pm2-administrador.service failed because the service did not
See "systemctl status pm2-administrador.service" and "journalctl -xe"
```

3.9 Despliegue producción Frontend

3.9.1 Requisitos previos

1. Instalación de Node.js y Yarn

```
$ sudo apt-get install nodejs  
  
$ node -v  
  
$ npm install yarn
```

2. Clonación de proyecto

```
Enlace del proyecto  
$ Git clone https://github.com/stalinch98/herramienta-colaborativa
```

3.9.2 Dependencias

1. En la terminal del Ide ejecutaremos el comando.

```
$ yarn install
```

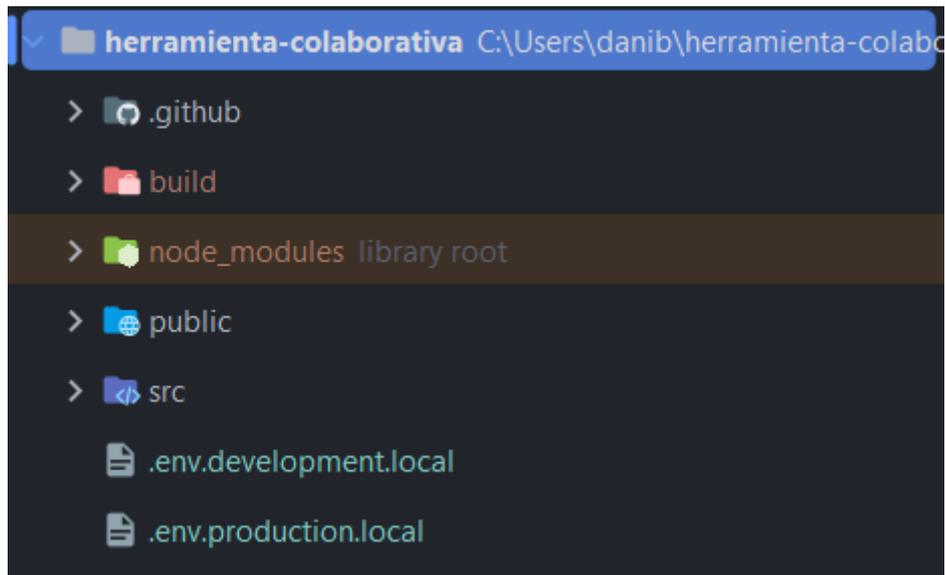
2. Cambiar la url del backend en la variable de entorno

```
Ejemplo:  
  
REACT_APP_BACKEND_URL=http://quicklab-api.csuioups.org/  
REACT_APP_RICHTEXT_KEY=  
cvjovu3ub4qdjg1mfknc11m3yb2nw4enc1yfey9jteokn3b7
```

3. Compilar el proyecto

```
$ yarn run build
```

Ilustración 24 Build del proyecto



Copiar la carpeta compilada por scp al servidor

```
$ scp -r .\build\ usuario@ip:path
```

Eliminar el contenido por default de Nginx

```
$ sudo mv path/build/* /var/www/html/
```

Ejemplo:

```
$ sudo mv /home/administrador/build/* /var/www/html/
```

4. Reiniciar y verificar el estado el servidor Nginx

```
$ sudo systemctl restart nginx
```

```
$ sudo systemctl status nginx
```

Como resultado final la aplicación estará funcionando en el entorno desplegado.

Ilustración 25 Funcionamiento de aplicación exitoso



UNIVERSIDAD POLITÉCNICA
SALESIANA
ECUADOR

Correo electrónico
rtufino@ups.edu.ec

Contraseña
.....

Iniciar

[Restablecer contraseña](#)

3.4 DevOps y SonarCloud

La integración de SonarCloud para el análisis estático de código se lo hizo mediante la herramienta GitHub Actions, esto nos permitió automatizar el análisis en cada merge o pull request a las ramas de desarrollo y producción en cada uno de los repositorios del proyecto. La herramienta permite importar los repositorios y a su vez automatizar el análisis mediante un archivo YAML.

La configuración del archivo YAML consta de la configuración inicial donde se especifica la rama que analizara, además se especifica que condición ejecutara el análisis, en este caso cualquier cambio a la rama de desarrollo ejecutara el análisis de manera automática, por último, se especifica las tareas que se especifican en el panel de configuración del repositorio en SonarCloud como se muestra en la Ilustración 3.

Ilustración 26 Archivo de Configuración SonarCloud

```
1  name: Build
2  on:
3    push:
4      branches:
5        - develop
6    pull_request:
7      types: [opened, synchronize, reopened]
8  jobs:
9    sonarcloud:
10     name: SonarCloud
11     runs-on: ubuntu-latest
12     steps:
13       - uses: actions/checkout@v2
14         with:
15           fetch-depth: 0 # Shallow clones should be disabled for a better relevancy of analysis
16       - name: SonarCloud Scan
17         uses: SonarSource/sonarcloud-github-action@master
18         env:
19           GITHUB_TOKEN: ${{ secrets.GITHUB_TOKEN }} # Needed to get PR information, if any
20           SONAR_TOKEN: ${{ secrets.SONAR_TOKEN }}
21
```

Tomado de la Ilustración 26 Nota: Fragmento de código con la configuración para el análisis de código con SonarCloud elaborado por los autores [Enlace a GitHub](#).

Al ejecutar el análisis por primera ocasión se pudo constatar el estado inicial del proyecto, los resultados del análisis para Backend y Frontend se muestran en la Ilustración 4 y 5 respectivamente.

Ilustración 27 Análisis de SonarCloud a la Herramienta Frontend Quicklab v1.0

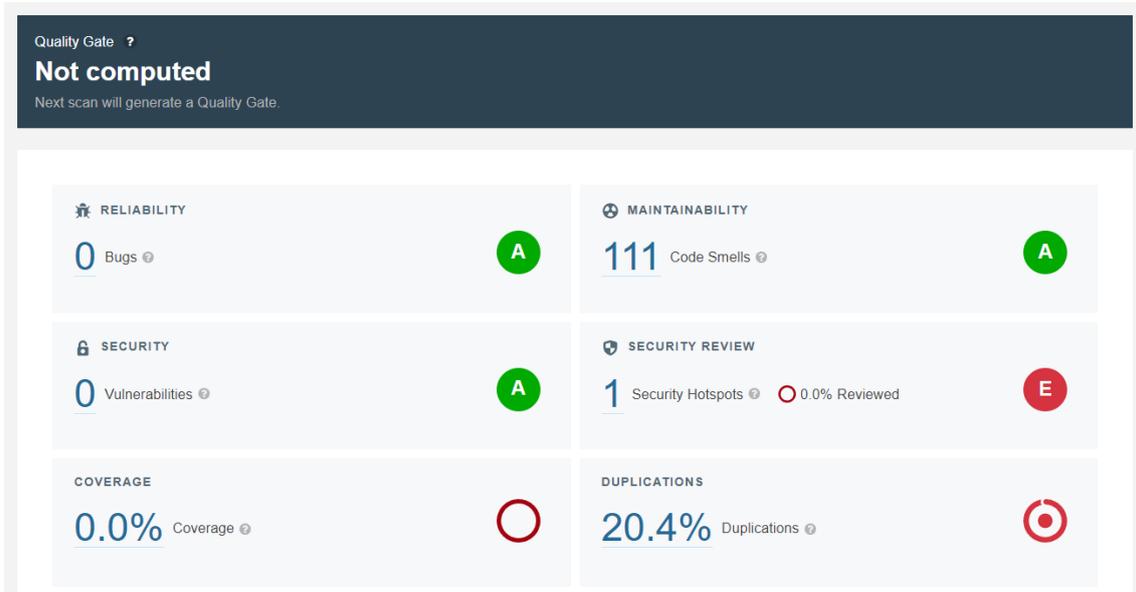
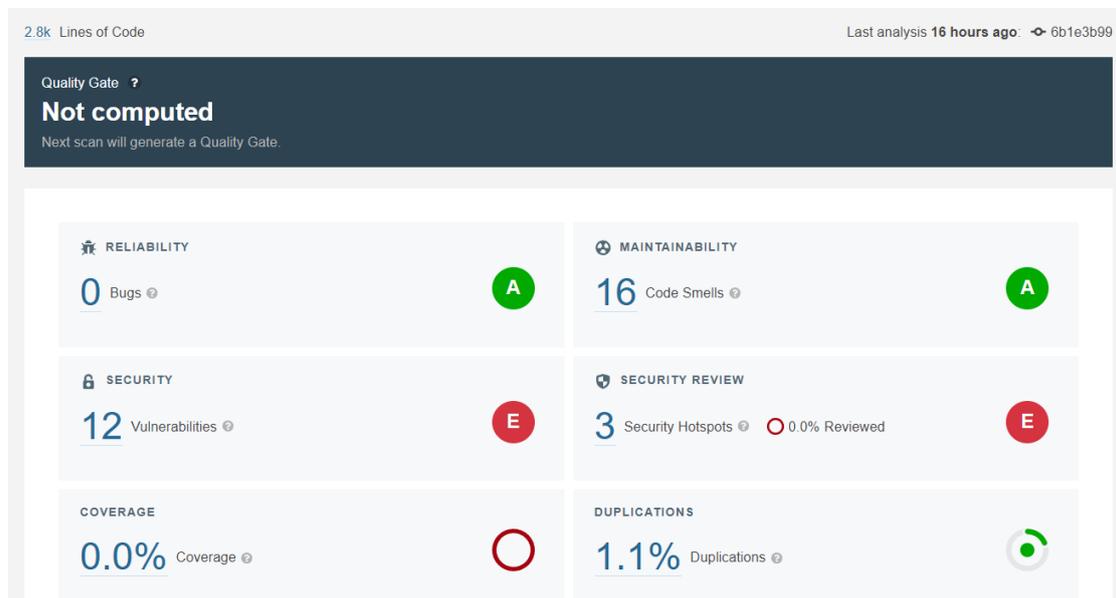


Ilustración 28 Análisis de SonarCloud a la Herramienta Backend Quicklab v1.0



3.5.1 Solución de problemas SonarCloud

Para cumplir con el objetivo de mejorar la mantenibilidad de software se implementó el análisis automático de código, que busca informar al desarrollador sobre los posibles problemas encontrados en el código. Además, se realizó cambios en el código inicial para dar solución a la mayor cantidad de problemas existentes en el código de la v1.0, los resultados al terminar el proyecto se los puede visualizar en la Ilustración 6 y 7

Ilustración 29 *Análisis Final de SonarCloud a la Herramienta Quicklab Frontend*

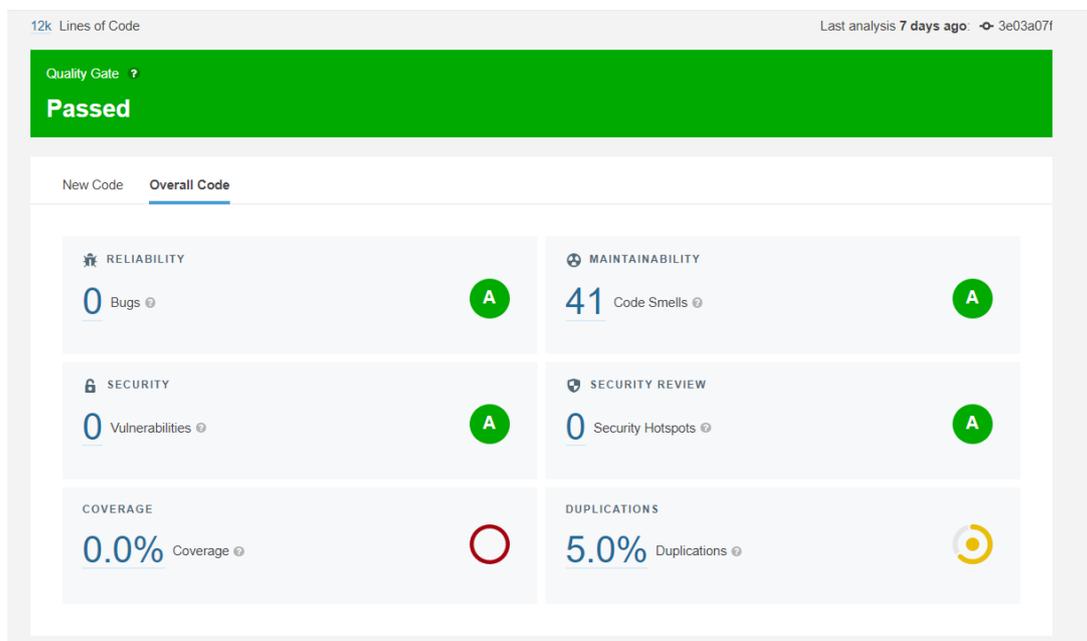
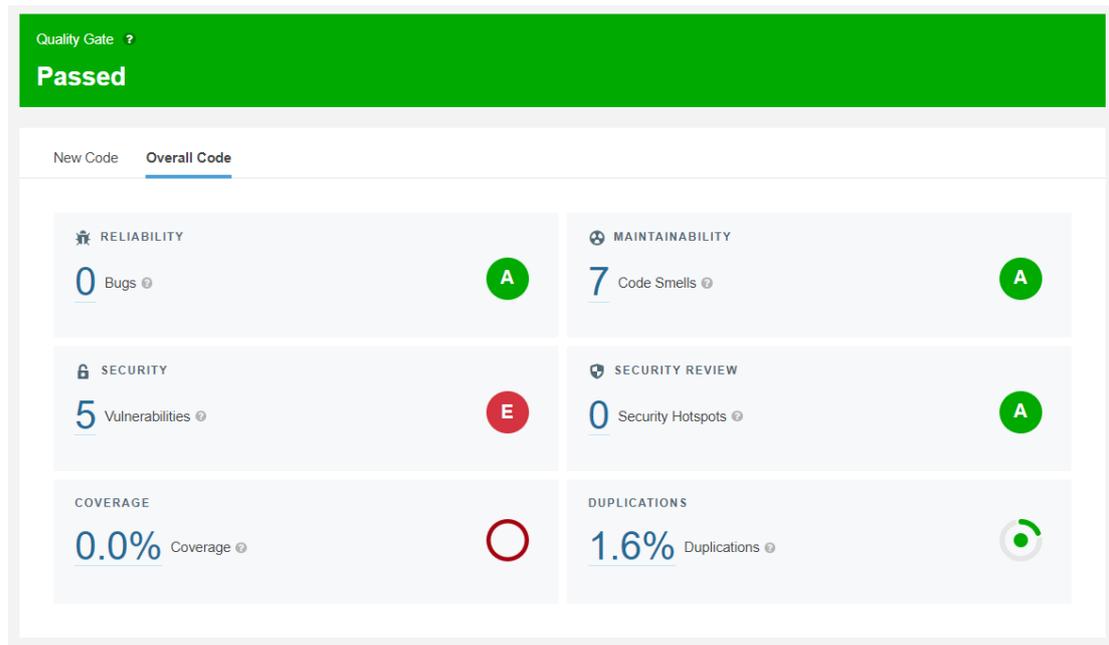
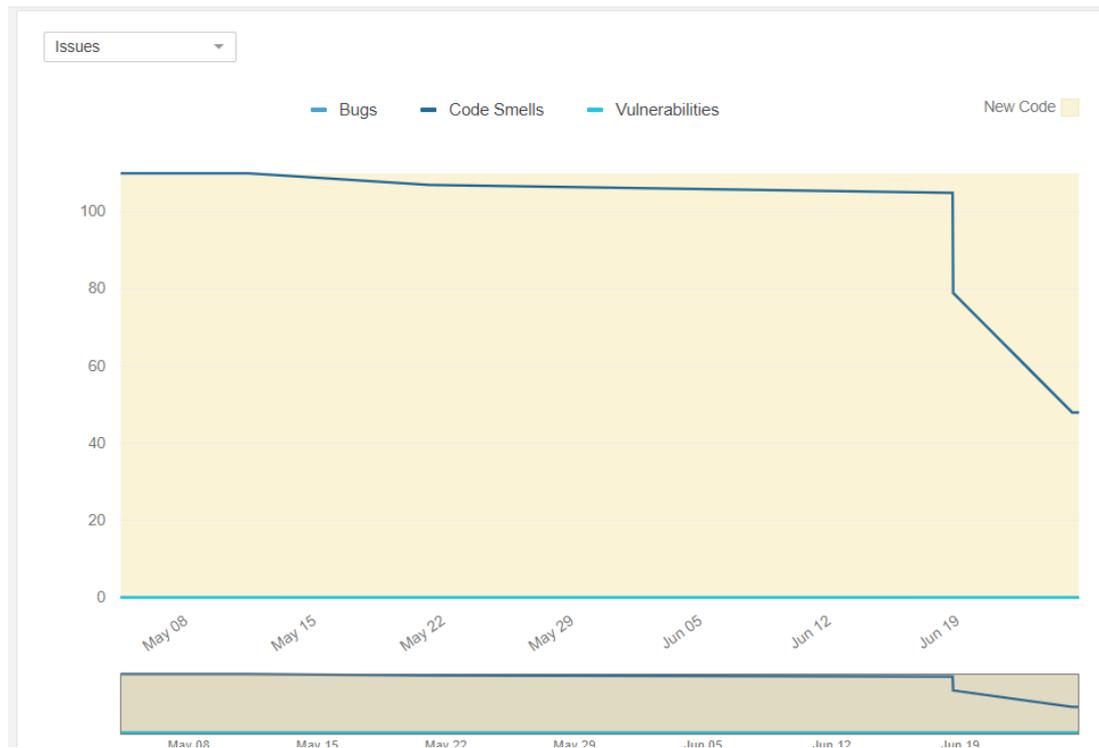


Ilustración 30 *Análisis Final de SonarCloud a la Herramienta Quicklab Backend*



En la Ilustración 8 se puede visualizar el seguimiento a los problemas encontrados en el código, al garantizar una considerable disminución de los problemas encontrados podemos garantizar que la mantenibilidad del sistema será óptima para futuros desarrollos

Ilustración 31 Seguimiento de Resolución de Problemas SonarCloud



CAPITULO IV

OPERATIVIDAD DE LA HERRAMIENTA

Este capítulo detalla el proceso llevado a cabo para brindar operatividad a la herramienta, para que la herramienta pueda ser utilizada se debe contar con información en la base de datos y manual de usuario para cada perfil que ingresa al sistema, para ello se implementó una pantalla adicional de ayuda para todos los perfiles de usuario, esta pantalla se encuentra disponible para cada perfil y muestra el manual de usuario correspondiente al perfil que ingreso al sistema. Para que la herramienta inicie su operación también se cargaron las guías de periodos recientes para cada docente según la documentación entregada por el coordinador del proyecto.

4.1 Creación de usuarios.

Como parte importante del proceso de puesta en producción de la aplicación Quicklab v1.1 se llevó a cabo el proceso de creación de usuarios conforme a los docentes que pertenecen al claustro de programación, de esta manera la operatividad de la herramienta está garantizada.

Los usuarios cargados a la herramienta pertenecen al perfil de:

- Administrador
- Coordinador
- Docentes

El perfil de administrador deberá estar atado a un correo diferente a los demás usuarios, la aplicación no permite crear un usuario que tenga el perfil de administrador y docente a la vez, lo cual se considera como una funcionalidad a implementar en futuras versiones del sistema.

Ilustración 32 *Usuarios de Quicklab*

Nombre	Correo	rol	
Daniel Dominguez	ddominguezr@est.ups.edu.ec	Administrador	✎ ✖
Alberto Rusbel Duchi Bastidas	aduchi@ups.edu.ec	Docente	✎ ✖
Silvana Lizette Vargas Jacome	svargas@ups.edu.ec	Docente	✎ ✖
Yadira Paola Jerez Narvaez	yjerez@ups.edu.ec	Docente	✎ ✖
Flavio Vinicio Changoluisa Panchi	fchangoluisa@ups.edu.ec	Docente	✎ ✖
Rodrigo Efrain Tufiño Cardenas	rtufino@ups.edu.ec	Docente	✎ ✖

4.2 Carga de información.

Para que la aplicación sea funcional y su operatividad esté garantizada, se cargó la información correspondiente a los periodos 57, 58 y 60 según la información disponible de cada docente, esta información se importó en la ambiente producción para que los usuarios puedan observar todas las referencias, temas, ejercicios y practicas creadas en los últimos periodos.

4.1.1 Carga de ejercicios en MongoDB

Tomando en cuenta la documentación de los ejercicios disponibles de los periodos requeridos, se realizó la carga de 53 ejercicios., el proceso con el que se realizó el ingreso de los datos fue mediante la herramienta, de esta manera se determinó que el proceso es funcional y disminuyó de manera considerable el tiempo con el que el usuario puede crear un ejercicio.

Ilustración 33 Ejercicios cargados en Quicklab

Dashboard / Programación / Ejercicios

Listado de ejercicios [Crear nuevo ejercicio](#)

	Periodo	Título	Tema	Dificultad	Docente	Calificación	Tipo	Archivado	Acciones
+	57	Ataque de orcos	Variables y tipo de datos	Medio	Rodrigo T.	☆☆☆ ☆☆☆	Practica	Sin Archivar	👁️ 📄 🗑️
+	57	Considere el siguiente lenguaje proposicional donde:	Proposiciones	Medio	Rodrigo T.	☆☆☆ ☆☆☆	Practica	Sin Archivar	👁️ 📄 🗑️
+	57	Construya la tabla de verdad para el circuito dado y derive las expresiones booleanas de salida para cada uno	Compuertas lógicas	Medio	Flavio C.	☆☆☆ ☆☆☆	Practica	Sin Archivar	👁️ 📄 🗑️
+	57	Dado el siguiente circuito lógico, realizar las actividades siguientes.	Compuertas lógicas	Medio	Rodrigo T.	☆☆☆ ☆☆☆	Practica	Sin Archivar	👁️ 📄 🗑️

4.1.2 Carga de referencias en MongoDB

El proceso para crear referencias puede ser realizado por el coordinador o docente, el sistema cuenta con un formulario para ingresar los datos de la referencia, el sistema se encargará de generar el formato adecuado de la referencia y lo mostrará en una tabla como lo muestra la Ilustración 35.

Ilustración 34 Formulario de creación referencias

Agregar nueva referencia

Tipo: Libro Sitio Web

* Título

+ Agregar Autor

Año de publicación

Edición

Editorial

Agregar Referencia

Ilustración 35 Referencias creadas en la aplicación

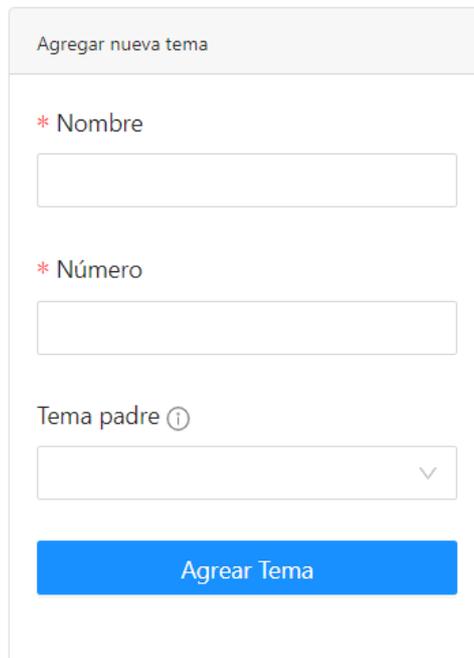
Alberto Rusbel Duchí Bastidas Cerrar sesión					
Referencias					
	Título	Autores	Publicación	Tipo	Acciones
+	Sistemas digitales: principios, análisis y diseño.	Héctor Arturo Flórez Fernández.		Libro	✎ ✖
+	Fundamentos de sistemas digitales.	Thomas L. Floyd.	2016	Libro	✎ ✖
+	Lógica y pensamiento formal.	Andrade, E., Cubides, P., Márquez, C., Vargas, E.,	2008	Libro	✎ ✖
+	Programación y Lógica Proposicional.	Arias, A.,	2014	Libro	✎ ✖
+	Introducción Práctica a la Lógica.	Costas, A.,	2016	Libro	✎ ✖
+	Instrucciones para un computador	Pérez-Sanagustin M.		Libro	✎ ✖
+	Lesson 1: Programming: Graph Paper Programming	Code.org.		Libro	✎ ✖

4.4.2 Carga de temas en MongoDB

La carga de temas en la base de datos se la realizo con la cuenta del usuario coordinador de la asignatura, debido a las restricciones en la funcionalidad se lo llevo a cabo utilizando el usuario rtufino@ups.edu.ec.

El formulario que solicita la aplicación es bastante intuitivo y solicita tres campos como lo muestra la Ilustración 36

Ilustración 36 Formulario de creación de temas



El formulario, titulado "Agregar nueva tema", contiene tres campos de entrada obligatorios: "Nombre", "Número" y "Tema padre". El campo "Tema padre" incluye un ícono de información. Debajo de los campos se encuentra un botón azul con el texto "Agregar Tema".

El sistema muestra en una tabla el detalle de los temas creados, los temas estarán disponible para el docente en el momento en que se lo cree, si una guía requiere de un tema diferente, se debe acudir al coordinador para la creación de este.

Ilustración 37 Temas creados en la aplicación

Número	Nombre	Padre	
1	Álgebra de boole		 
2	Compuertas lógicas		 
3	Proposiciones		 
4	Representación de algoritmos simples		 
5	Resolución de problemas utilizando algoritmos		 
6	Variables y tipo de datos		 

< 1 >

4.4.3 Carga de plantillas

De la misma manera que con los temas de una guía, las plantillas son administradas por el coordinador del proyecto, en Quicklab se crearon las plantillas correspondientes a los periodos proporcionados, la generación de una guía requiere de una plantilla para su generación. El proceso para generar una plantilla es bastante eficaz y sencillo de realizar, los campos que solicita el formulario son claros y no muestran dificultad como lo muestra la Ilustración 38

Ilustración 38 Formulario para crear una plantilla

Nueva practica

* Título

* Formato * Número Practica * Tema

Seleccionar un formato Seleccionar tema

Objetivos

+ Agregar Objetivos

Pre-Requisitos

+ Agregar Requisito

Instrucciones

+ Agregar Instrucción

Resultados obtenidos

+ Agregar Resultado

4.3 Generación de guías de laboratorio

Para contar con las guías de laboratorio generadas en los periodos 57, 58 y 60 se realizó la carga manual de referencias, plantillas, temas para poder generar cada guía. Dicho proceso fue llevado a cabo con la documentación del claustro de docentes de programación. La cual proporciona información acerca de cada docente y el ejercicio creado.

Los pasos realizados con anterioridad en la carga de datos permitieron crear las guías correspondientes a los periodos citados, las guías son visibles para el coordinar y docente de la materia y pueden ser visualizadas, confirmadas o eliminadas según sea la necesidad.

Ilustración 39 Tabla de Practicas en Quicklab

Dashboard / Programación / Practicas

Practicas generadas

Nueva practica

Periodo	Plantilla	Creado	Lista de ejercicios	Acciones
58	Sentencias de repetición	13 de julio de 2022	> 5 Ejercicios	Descargar practica Confirmar practica con soluciones Eliminar practica
58	Proposiciones	13 de julio de 2022	> 8 Ejercicios	Descargar practica Confirmar practica con soluciones Eliminar practica
58	Compuertas lógicas	13 de julio de 2022	> 4 Ejercicios	Descargar practica Confirmar practica con soluciones Eliminar practica
57	Algebra booleana	13 de julio de 2022	> 5 Ejercicios	Descargar practica Confirmar practica con soluciones Eliminar practica
57	Resolucion de problemas utilizando funciones	13 de julio de 2022	> 5 Ejercicios	Descargar practica Confirmar practica con soluciones Eliminar practica

En la Ilustración 40 se muestra el formato final de la practica con cada componente que previamente le fue asignado, al dar clic en descargar, el sistema mostrará la práctica y al final del documento permitirá dar clic en descargar

Ilustración 40 Formato referencial de una guía generada en Quicklab



VICERRECTORADO DOCENTE Código: GUIA PRL 001

CONSEJO ACADEMICO Aprobación: 2016/04/06

Formato: Practica de laboratorio

FORMATO DE PRACTICA DE LABORATORIO PARA DOCENTES

CARRERA: Computación

ASIGNATURA: Programación

PRACTICA: 8

TÍTULO: Sentencias de repetición

OBJETIVO:

- Aplicar bucles de repetición para la solucionar problemas con programación

PRE-REQUISITOS:

- a. Computador con sistema operativo Microsoft Windows, MacOS X o GNU/Linux
- b. Programa IDE para desarrollo
- c. Conexión a Internet

INSTRUCCIONES:

1. Lea detenidamente cada uno de los enunciados propuestos.
2. Plantee una solución a cada uno de los ejercicios.
3. Desarrolle la solución planteada.
4. Elabore un informe con la solución de los ejercicios.

ACTIVIDADES A DESARROLLAR:

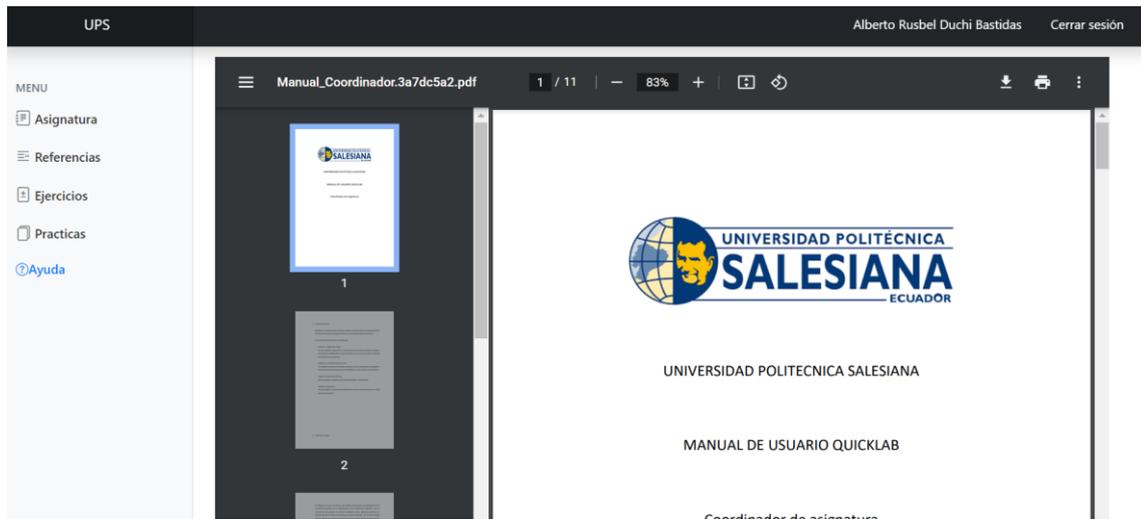
1. Tabla de multiplicar

Escriba un programa denominado TablaMultiplicar que solicite al usuario el tamaño (un entero positivo); e imprima la tabla de multiplicar como se muestra en el ejemplo

4.4 Manual de usuarios

Para facilitar el uso de la herramienta, se implementó una pantalla adicional en donde el usuario puede visualizar un manual de uso sobre la funcionalidad asignada al perfil del docente, este manual detalla cada funcionalidad del sistema, además estará disponible para descarga en caso de ser necesario

Ilustración 41 Pantalla de ayuda al usuario



CONCLUSIONES

1. El objetivo principal de la herramienta colaborativa es brindar una manera eficaz y sencilla para generación de guías de laboratorio, Quicklab está diseñada para disminuir el tiempo en este proceso, al cargar las guías de periodos anteriores se constató que el proceso es eficaz, la generación de una guía con todos sus componentes tarda treinta minutos en el caso más complejo, a diferencia del proceso manual en el que el tiempo llegó a ser de una hora y quince minutos.
2. En el desarrollo de software, la correcta documentación de la construcción del sistema, resulta ser una pieza para asegurar la mantenibilidad en el futuro. Es importante considerar que el software puede estar sujeto a cambios y nuevas implementaciones por lo cual asegurar el correcto manejo del mantenimiento representa de cierta manera el éxito o fracaso en la construcción.
3. El uso de DevOps como marco de trabajo y la integración de SonarCloud permitió mejorar la mantenibilidad del sistema y la calidad del código en la construcción, SonarCloud permitió reducir el tiempo que costaría invertir en solucionar problemas en el futuro, lo cual brinda mayor confiabilidad a la aplicación, por su lado el uso de DevOps mejoro la habilidad en el desarrollo y pruebas, el tiempo en el que el despliegue y análisis del entorno de pruebas tomaría hacerlo de manera manual llegaría a ser mayor a las tres horas, en Quicklab el tiempo es de cinco minutos y no requiere de manipulación del desarrollador.
4. El despliegue de la herramienta en un ambiente de producción puede verse afectado a restricciones dentro de la infraestructura disponible, el análisis de recursos y permisos pueden ayudar a identificar de mejor manera posibles problemas relacionados, de igual manera es importante contar con acciones de

contingencia en caso de tener problemas en la implementación de la herramienta.

5. Los proyectos desarrollados bajo la plataforma Node.js pueden verse afectados principalmente por sus dependencias, el crecimiento conforme avanza un proyecto y con la demanda de nuevas librerías puede perjudicar directamente al almacenamiento disponible para realizar el despliegue del aplicativo, si el espacio de almacenamiento disponible es mínimo, el uso de Docker permite distribuir el proyecto en varios contenedores, optimizando de esta manera el espacio que ocupa y sin afectar su funcionalidad.

RECOMENDACIONES

- Se recomienda implementar una funcionalidad que permita exportar ejercicios individualmente o en conjunto para su posterior importación en la plataforma AVAC de la Universidad Politécnica Salesiana.
- La funcionalidad para crear una cita bibliográfica actualmente es de forma manual, se recomienda consumir algún servicio como Mendeley e implantarlo en la herramienta para hacer este procedimiento de forma automática.
- Debido a las restricciones de la infraestructura de la Universidad Politécnica Salesiana no hubo una manera de integrar el despliegue continuo con el ambiente de producción, por lo tanto, se recomienda la creación de un script que se ejecute y despliegue la aplicación cuando existan diferencias entre el repositorio de GitHub y el ambiente de producción.
- Para la recuperación de la contraseña se utilizó NodeMailer el cual utiliza un puerto específico para el envío del correo, pero debido a las seguridades del servidor este puerto se encuentra restringido, se recomienda utilizar algún otro mecanismo de restablecimiento de la contraseña.
- Se recomienda la implementación de una funcionalidad que permite la creación de una guía de laboratorio en un formato de seguimiento de pasos además de la creación de ejercicios como se lo maneja en la versión actual. De esta manera se puede brindar al estudiante un acompañamiento en cada paso realizado durante la guía de laboratorio.
- Una funcionalidad que ayudaría al control de usuarios es permitir crear un usuario administrador que también tenga un perfil de coordinador o docente, existe la posibilidad de que así se requiera por lo cual es importante tomarlo en cuenta en futuras implementaciones y versiones.

REFERENCIAS

- Adones Farfán, J., & Vega-Zepeda, V. (2020). Mantenibilidad del Software. Consideraciones para su especificación y validación. *Ingeniare. Revista Chilena de Ingeniería*, 28(4), 654–667. <https://doi.org/10.4067/s0718-33052020000400654>
- Aleksandar Kovacević. (2020). *How to Install and Configure Nginx on Ubuntu 20.04*. <https://phoenixnap.com/kb/how-to-install-nginx-on-ubuntu-20-04>
- Cobo, Oña, & Tufiño. (2021). *Creación de una herramienta colaborativa para prácticas de laboratorio del claustro docente de programación de la Universidad Politécnica Salesiana, sede Quito*. <http://dspace.ups.edu.ec/handle/123456789/20658>
- Francisco Ruiz, M. P. (2001). Mantenimiento del Software. *Crisis*, 1–31. <https://alarcos.esi.uclm.es/per/fruiz/curs/mso/trans/S3.pdf>
- GitHub, I. (2016). *What is GitHub?* <https://docs.github.com/es/get-started/quickstart/hello-world>
- JetBrains s.r.o. (2022). *Por qué IntelliJ IDEA*. <https://www.jetbrains.com/es-es/idea/>
- Mark Drake. (2020). *How To Install MongoDB on Ubuntu 20.04*. <https://www.digitalocean.com/community/tutorials/how-to-install-mongodb-on-ubuntu-20-04>
- Netlify. (2022). *About Netlify*. <https://www.netlify.com/about/>
- Postman, I. (2022). *About Postman*. <https://www.postman.com/company/about-postman/>
- Salesforce.es. (2022). *What is Heroku*. <https://www.heroku.com/what#a-focus-on-apps>

SonarCloud. (2022). *What is SonarCloud?* <https://docs.sonarcloud.io/>

SonarSource S.A. (2022). *Clean Code starts in your IDE.* <https://www.sonarlint.org/>

Webempresa. (2022). *Qué es un dominio y cómo funciona.*

<https://www.webempresa.com/hosting/que-es->

[dominio.html?gclid=CjwKCAjw2rmWBhB4EiwAiJ0mtXMjzE9rZa_yWFEwTM](https://www.webempresa.com/hosting/que-es-dominio.html?gclid=CjwKCAjw2rmWBhB4EiwAiJ0mtXMjzE9rZa_yWFEwTM)

[CfdSJqAS5tMAB8XfFlgegwr5YmKyMIFj45RoCqVwQAvD_BwE](https://www.webempresa.com/hosting/que-es-dominio.html?gclid=CjwKCAjw2rmWBhB4EiwAiJ0mtXMjzE9rZa_yWFEwTM)

ANEXOS

ANEXO 1: Manual de usuario del administrador



UNIVERSIDAD POLITECNICA SALESIANA

MANUAL DE USUARIO QUICKLAB

Administrador del Sistema

Contenido

1. Uso del documento.....	4
2. Ámbito del trabajo	4
3. Funcionamiento de Quicklab	5
3.1 Navegación por la web.....	5
3.2 Manual de uso de la web.....	7
3.2.1 Gestión de usuarios	7
3.2.2 Administración de carreras	10

Ilustraciones

Ilustración 1 Formulario de inicio de sesión	6
Ilustración 2 Pantalla principal del administrador	6
Ilustración 3 Menú de funcionalidades.....	7
Ilustración 4 Pantalla Usuarios Quicklab	8
Ilustración 5 Formulario de registro de un usuario	9
Ilustración 6 Captura de la administración de usuarios creados.....	9
Ilustración 7 Captura de la pantalla Carreras.....	10
Ilustración 8 Captura del formulario de creación de asignaturas	11
Ilustración 9 Tabla de carreras creadas.....	11
Ilustración 10 Captura del formulario de periodos.....	12
Ilustración 11 Captura de la tabla Periodos	12

1. Uso del documento

Quicklab es un sistema web creado para optimizar la gestión de guías de laboratorio del claustro docentes de programación de la Universidad Politécnica Salesiana.

La estructura del documento es el siguiente:

- **Capítulo 2. Ámbito del trabajo**

En este capítulo se presenta una visión general del contexto asociado al trabajo y encuadrado al objetivo de la web presentada en este manual en base al perfil del administrador del sistema.

- **Capítulo 3. Funcionamiento de la web**

Este capítulo muestra los tutoriales asociados al rol del administrador, incluyendo descripciones de las funcionalidades en cada sección y sus limitaciones.

2. Ámbito del trabajo

El trabajo en el que se enmarca este sistema web propone la implantación de la herramienta Quicklab en la infraestructura de la Universidad Politécnica. Para la creación de una práctica es necesario establecer temas, objetivos, ejercicios, etc. Quicklab permite al usuario hacerlo de una manera dinámica y en el menor tiempo posible. Además de plantear un sistema capaz de generar de forma automática las practicas, se destaca el uso de herramientas y recursos utilizados, aprovechando la información existente, para extraer, organizar, seleccionar y publicar las practicas clasificadas por tema y por cada periodo existente dentro del sistema.

Para administrar el sistema se ha creado el usuario “**Administrador**” con el que se gestionará la funcionalidad del sistema, el usuario será capaz de crear periodos académicos, creación de usuarios, cambio de contraseña, creación de carreras, asignaturas y perfiles de usuario.

Las funciones principales que tiene el administrador son:

- Creación de usuarios y asignación de perfiles
- Creación de periodos
- Cambio de contraseñas
- Creación de carreras
- Creación de asignaturas

3. Funcionamiento de Quicklab

Esta sección se divide en dos partes. En primer lugar, se muestra la navegación a lo largo de la web, con las distintas secciones que ofrece para cada tipo de usuario. En segundo lugar, se muestran los tutoriales para cada funcionalidad de las descritas en la sección anterior.

3.1 Navegación por la web

Para acceder a la página, se accede mediante la siguiente url:

<https://quicklab.csuioups.org/>

Para acceder al sistema se lo puede hacer mediante el botón “iniciar sesión” con las credenciales establecidas del administrador como lo muestra la siguiente imagen:

UNIVERSIDAD POLITÉCNICA
SALESIANA
ECUADOR

Correo electronico
schiguanog@est.ups.edu.ec

Contraseña
.....

Iniciar

[Restablecer contraseña](#)

Ilustración 42 Formulario de inicio de sesión

Una vez registrado el usuario, se accede a la página principal como se muestra en la Ilustración 2:

Stalin Chiguanano Cerrar sesión

Usuarios Gestionar usuarios

Agregar nuevo usuario

* Nombre

* Apellido

* Correo Electronico
schiguanog@est.ups.edu.ec

* Contraseña

Rol
Docente Administrador

Crear Usuario

Nombre	Correo	rol	
Stalin Chiguanano	schiguanog@est.ups.edu.ec	Administrador	✎ ✖
Alberto Rusbel Duchi Bastidas	aduchi@ups.edu.ec	Docente	✎ ✖
Rodrigo Efraim Tufiño Cardenas	rtufino@ups.edu.ec	Docente	✎ ✖
Silvana Lizette Vargas Jacome	svargas@ups.edu.ec	Docente	✎ ✖
Yadira Paola Jerez Narvaez	yjerez@ups.edu.ec	Docente	✎ ✖
Flavio Vinicio Changoluisa Panchi	fchangoluisa@ups.edu.ec	Docente	✎ ✖

Ilustración 43 Pantalla principal del administrador

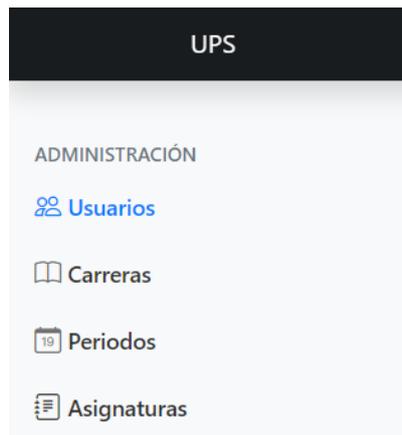


Ilustración 44 Menú de funcionalidades

Las opciones que se encuentran en las ilustraciones anteriores corresponden a la funcionalidad del sistema para el administrador.

3.2 Manual de uso de la web

En este punto se mostrará el tutorial relativo a cada funcionalidad

3.2.1 Gestión de usuarios

La página mostrada al entrar en la opción Usuarios es la mostrada en la Ilustración 4:

Usuarios Gestionar usuarios

Agregar nuevo usuario

* Nombre

* Apellido

* Correo Electronico

* Contraseña

Rol

Docente Administrador

< 1 >

Nombre	Correo	rol	
Stalin Chiguano	schiguanog@est.ups.edu.ec	Administrador	↗ ✖
Alberto Rusbel Duchi Bastidas	aduchi@ups.edu.ec	Docente	↗ ✖
Rodrigo Efrain Tufiño Cardenas	rtufino@ups.edu.ec	Docente	↗ ✖
Silvana Lizette Vargas Jacome	svargas@ups.edu.ec	Docente	↗ ✖
Yadira Paola Jerez Narvaez	yjerez@ups.edu.ec	Docente	↗ ✖
Flavio Vinicio Changoluisa Panchi	fchangoluisa@ups.edu.ec	Docente	↗ ✖

Ilustración 45 Pantalla Usuarios Quicklab

La pantalla muestra un formulario para crear un nuevo usuario, todos los campos son requeridos, el rol Docente estará limitado a la creación, calificación y comentario de ejercicios, el rol de administrador tendrá los mismos permisos que el usuario ingresado, es importante tener en cuenta que el rol de coordinador se lo asigna a un docente para cada asignatura.

No existe un rol de coordinador al crear un usuario, se lo debe asignar a una asignatura para lograr establecer dicho rol.

Agregar nuevo usuario

* Nombre

* Apellido

* Correo Electronico

* Contraseña

Rol
 Docente Administrador

[Crear Usuario](#)

Ilustración 46 Formulario de registro de un usuario

En el lado derecho del formulario se podrá visualizar los usuarios creados, nombre, correo y rol, además de contar con una columna adicional con las funciones de editar y eliminar un usuario.

Nombre	Correo	rol	
Stalin Chiguano	schiguanog@est.ups.edu.ec	Administrador	✎ ✖
Alberto Rusbel Duchi Bastidas	aduchi@ups.edu.ec	Docente	✎ ✖
Rodrigo Efrain Tufiño Cardenas	rtufino@ups.edu.ec	Docente	✎ ✖
Silvana Lizette Vargas Jacome	svargas@ups.edu.ec	Docente	✎ ✖
Yadira Paola Jerez Narvaez	yjerez@ups.edu.ec	Docente	✎ ✖
Flavio Vinicio Changoluisa Panchi	fchangoluisa@ups.edu.ec	Docente	✎ ✖

Ilustración 47 Captura de la administración de usuarios creados

3.2.2 Administración de carreras

En la sección de carreras se podrá administrar las carreras, el coordinador y docentes que pertenecen a la misma, para crear una carrera se le asigna un nombre y se crea una tarjeta con la opción de editar, eliminar y ver asignaturas.



Ilustración 48 Captura de la pantalla Carreras

La opción “ver Asignaturas” nos llevara a la sección de asignaturas de la herramienta, la misma permite administrar la carrera mediante la creación de asignaturas y asignación de coordinador y docentes, cada carrera debe tener al menos una asignatura, un coordinador y un docente para poder crear prácticas de laboratorio, la pantalla es similar a la creación de usuarios, se desplegará un formulario con la información requerida.

Agregar nueva asignatura

* Código

* Nombre

* Carrera

Seleccionar una carrera
▼

Coordinador

Seleccionar un coordinador
▼

Docentes

Seleccionar docentes
▼

Agregar Asignatura

Ilustración 49 Captura del formulario de creación de asignaturas

En la pantalla también se encontrará la información creada con anterioridad, y las opciones para editar o eliminar la información ingresada

1

	Codigo	Nombre	Carrera	Coordinador	
+	asd123	Programacion	Computación	Rodrigo Efrain Tufiño Cardenas	✎ ✖

Ilustración 50 Tabla de carreras creadas

La última sección en el menú corresponde a la administración de periodos académicos, en esta funcionalidad se podrá crear periodos por fechas de inicio y fin, además del estado, el estado activo será utilizado en la creación de prácticas.

Agregar nuevo periodo

* Periodo:

* Fecha: →

Ilustración 51 Captura del formulario de periodos

Similar a las pantallas anteriores, se muestra los periodos creados y el estado actual, con la respectiva funcionalidad para editar y eliminar según sea el caso.

1

Periodo	Inicio periodo	Fin periodo	Estado
60	27/03/2022	31/07/2022	ACTIVAR ✎ ✖
59	12/09/2021	28/02/2022	ACTIVAR ✎ ✖
58	28/03/2021	31/07/2021	ACTIVO

Ilustración 52 Captura de la tabla Periodos

Anexo 2: Manual de usuario del coordinador



UNIVERSIDAD POLITECNICA SALESIANA

MANUAL DE USUARIO QUICKLAB

Coordinador de asignatura

Contenido

1. Uso del documento	16
2. Ámbito del trabajo	16
3. Funcionamiento de Quicklab	17
3.1 Navegación por la web	17
3.2 Manual de uso de la web	19
3.2.1 Asignatura.....	20
3.2.2 Temas	22
3.2.3 Plantillas	23
3.2.4 Ejercicios	25
4.2.5 Gestión de practicas.....	26

Ilustraciones

Ilustración 1 Captura del formulario de inicio de sesión.....	18
Ilustración 2 Formulario para restablecer contraseña.....	18
Ilustración 3 Dashboard del coordinador de asignatura	19
Ilustración 4 Menú de funcionalidades.....	19
Ilustración 5 Dashboard de la asignatura programación	20
Ilustración 6 Grafica de temas usados por ejercicio	20
Ilustración 7 Grafica de calificación de ejercicios.....	21
Ilustración 8 Docentes de la asignatura de programación	21
Ilustración 9 Formulario para la creación de docentes	22
Ilustración 10 Formulario para la creación de temas.....	22
Ilustración 11 Tabla de temas.....	23
Ilustración 12 Tabla informática de las plantillas.....	23
Ilustración 13 Formulario de las plantillas	24
Ilustración 14 Formulario de practicas	24
Ilustración 15 Listado de ejercicios	26
Ilustración 16 Captura del formulario de creación de ejercicios	26
Ilustración 17 Formulario para crear una practica.....	27
Ilustración 18 Listado de ejercicios	27

1. Uso del documento

Quicklab es un sistema web creado para optimizar la gestión de prácticas de laboratorio del claustro docentes de programación de la Universidad Politécnica Salesiana.

La estructura del documento es el siguiente:

Capítulo 2. Ámbito del trabajo

En este capítulo se presenta una visión general del contexto asociado al trabajo y encuadrado al objetivo de la web presentada en este manual en base al perfil de coordinador de la asignatura.

Capítulo 3. Funcionamiento de la web

Este capítulo muestra los tutoriales asociados al rol de coordinador de asignatura, incluyendo descripciones de las funcionalidades en cada sección y sus limitaciones.

2. Ámbito del trabajo

El trabajo en el que se enmarca este sistema web propone la implantación de la herramienta Quicklab en la infraestructura de la Universidad Politécnica. Para la creación de una práctica es necesario establecer temas, objetivos ejercicios, etc. Quicklab permite al usuario hacerlo de una manera dinámica y en el menor tiempo posible. Además de plantear un sistema capaz de generar de forma automática las practicas, se destaca el uso de herramientas y recursos utilizados, aprovechando la

información existente, para extraer, organizar, seleccionar y publicar las practicas clasificadas por tema y por cada periodo existente dentro del sistema.

Para la gestión de carreras dentro del sistema se ha creado el usuario “Coordinador” con el que se gestionará la funcionalidad del sistema.

Las funciones principales que tiene el administrador son:

- Creación de temas
- Creación de plantillas
- Administración de referencias
- Creación y administración de ejercicios
- Creación de practicas

3.Funcionamiento de Quicklab

Esta sección se divide en dos partes. En primer lugar, se muestra la navegación a lo largo de la web, con las distintas secciones que ofrece para cada tipo de usuario. En segundo lugar, se muestran los tutoriales para cada funcionalidad de las descritas en la sección anterior.

3.1 Navegación por la web

Para acceder a la página, se accede mediante la siguiente url:

<https://quicklab.csuioups.org/>

Para acceder al sistema se lo puede hacer mediante el botón “iniciar sesión” con las credenciales establecidas por el administrador como lo muestra la siguiente imagen:



The image shows a login form for the Universidad Politécnica Salesiana Ecuador. At the top, there is a logo featuring a globe and a portrait of a man, with the text "UNIVERSIDAD POLITÉCNICA SALESIANA ECUADOR". Below the logo, there are two input fields: "Correo electronico" with the value "rtufino@ups.edu.ec" and "Contraseña" with masked characters. A blue button labeled "Iniciar" is positioned below the password field. At the bottom left, there is a link that says "Restablecer contraseña".

Ilustración 53 Captura del formulario de inicio de sesión

Quicklab cuenta con la funcionalidad de recuperar la contraseña en caso de olvidarla, para poder recuperar la contraseña es necesario el correo electrónico con el que se inicia sesión, el sistema enviará un pin mediante el cual se podrá cambiar la contraseña



The image shows a password recovery form for the Universidad Politécnica Salesiana Ecuador. At the top, there is a logo featuring a globe and a portrait of a man, with the text "UNIVERSIDAD POLITÉCNICA SALESIANA ECUADOR". Below the logo, there is a heading "Recuperar contraseña". Underneath, there is an input field labeled "Correo electronico" with the placeholder text "Correo". A blue button labeled "Restablecer" is positioned below the input field.

Ilustración 54 Formulario para restablecer contraseña

Una vez registrado el usuario, se accede a la página principal como se muestra en la Ilustración 3:

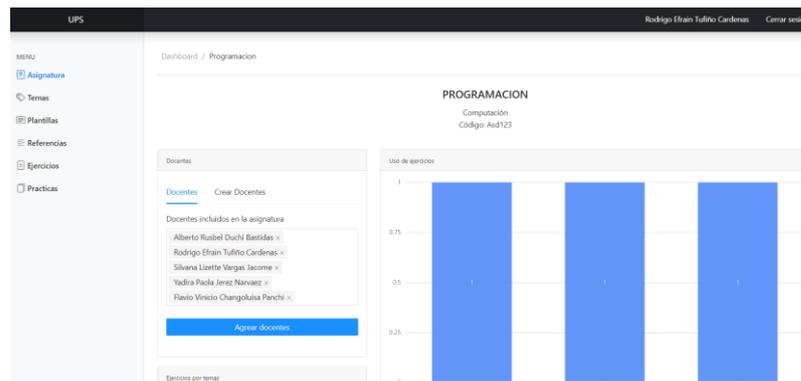


Ilustración 55 Dashboard del coordinador de asignatura

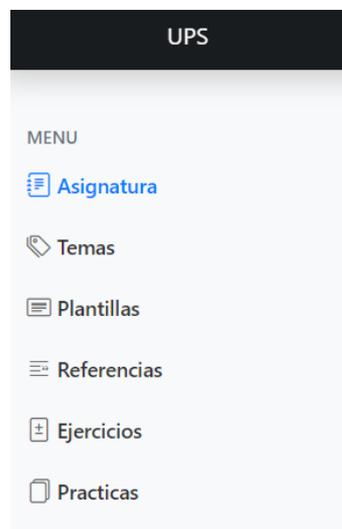


Ilustración 56 Menú de funcionalidades

Las opciones que se encuentran en las ilustraciones anteriores corresponden a la funcionalidad del sistema para el administrador.

3.2 Manual de uso de la web

En este punto se mostrará el tutorial relativo a cada funcionalidad

3.2.1 Asignatura

Al ingresar a la opción de asignatura, el sistema nos mostrara el Dashboard con información relevante de la asignatura a la que el coordinado pertenece, en el dashboard se puede encontrar la información relevante a los docentes que pertenecen a la misma, graficas de los ejercicios y temas utilizados y la calificación de los ejercicios previamente cargados:

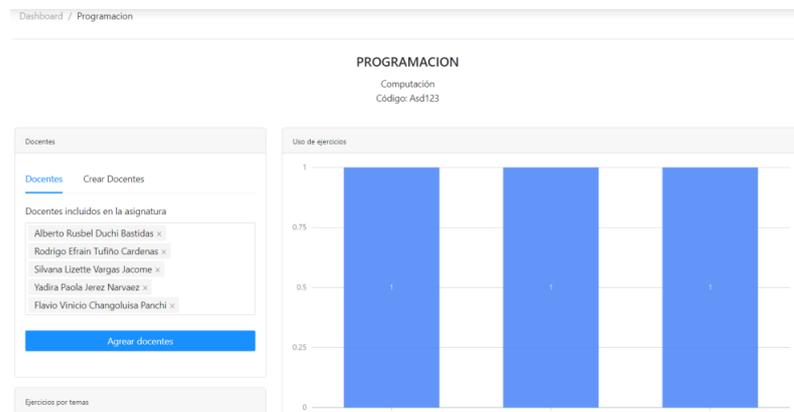


Ilustración 57 Dashboard de la asignatura programación

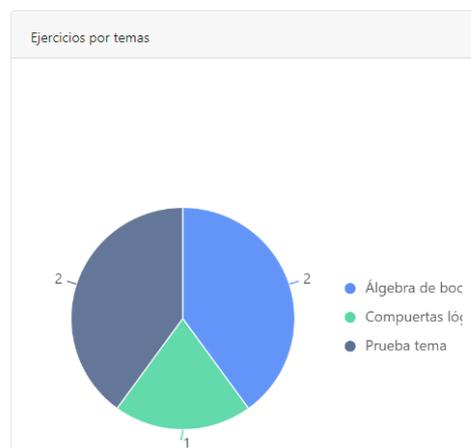


Ilustración 58 Grafica de temas usados por ejercicio

Calificación ejercicios	
Costo de pasaje	★ ★ ★ ★ ★
Formalización de proposiciones	★ ★ ★ ★ ★
Llamadas telefónicas	★ ★ ★ ★ ★
Prueba	★ ★ ★ ★ ★

Ilustración 59 Grafica de calificación de ejercicios

En la sección de docentes, la herramienta permite al coordinador visualizar los docentes de la asignatura, además le permite añadir nuevos docentes según sea el caso

Docentes

Docentes [Crear Docentes](#)

Docentes incluidos en la asignatura

Alberto Rusbel Duchi Bastidas ×

Rodrigo Efrain Tufiño Cardenas ×

Silvana Lizette Vargas Jacome ×

Yadira Paola Jerez Narvaez ×

Flavio Vinicio Changoluisa Panchi ×

Agregar docentes

Ilustración 60 Docentes de la asignatura de programación

Docentes

Docentes [Crear Docentes](#)

* Nombre

* Apellido

* Correo Electronico

* Contraseña

Crear Usuario

Ilustración 61 Formulario para la creación de docentes

3.2.2 Temas

En esta pantalla el coordinador podrá crear temas que se utilizaran en la respectiva asignatura, mismas a las que se le asignaran los ejercicios creados por docentes. Un tema puede tener un tema padre al que pertenece en el formulario de creación se podrá escoger dicha información.

Agregar nueva tema

* Nombre

* Número

Tema padre ⓘ

Agregar Tema

Ilustración 62 Formulario para la creación de temas

La pantalla mostrará una tabla con los temas creados y permitirá editar o eliminar de ser necesario

Temas

Número	Nombre	Padre	
1	Álgebra de boole		 
1	Compuertas lógicas		 
1	Prueba tema		 

< 1 >

Ilustración 63 Tabla de temas

3.2.3 Plantillas

La sección de plantillas permite al coordinador visualizar, editar, eliminar y crear plantillas. La pantalla principal mostrara el listado de plantillas creadas como se muestra en la Ilustración

Listado de plantilla		Crear plantilla	
Título	Objetivo principal	Temas	
Practica prueba	Aplicar sentencias condicionales para solucionar problemas con programación	—	 
Representación de algoritmos	Prueba	—	 

< 1 >

Ilustración 64 Tabla informática de las plantillas

La función editar mostrara un formulario con los campos obligatorios y opcionales que tiene una practica de laboratorio, y el coordinador será el único usuario que podrá editar la plantilla.

Editar plantilla

• Título

• Formato **• Número Practica** **• Tema**

Objetivos
 Objetivos

 + Agregar Objetivos

Pre-Requisitos
 Requisitos

 + Agregar Requisito

Instrucciones
 Instrucciones

Ilustración 65 Formulario de las plantillas

Para crear una plantilla el formulario es el mismo mostrado en la opción editar, como se lo muestra en la Ilustración

Nueva practica

• Título

• Formato **• Número Practica** **• Tema**

Objetivos
 + Agregar Objetivos

Pre-Requisitos
 + Agregar Requisito

Instrucciones
 + Agregar Instrucción

Resultados obtenidos
 + Agregar Resultado

Ilustración 66 Formulario de practicas

3.2.4 Referencias

Esta funcionalidad permite la administración de referencias de la guía de laboratorio, el sistema solicitará la información necesaria para generar una referencia basando en la norma APA séptima edición, la referencia puede ser de sitio web o libro según sea el caso, la información requerida es la mostrada en la Ilustración 15

Dashboard / Programacion / Referencias

Agregar nueva referencia

Tipo: Libro Sitio Web

* Titulo

+ Agregar Autor

Año de publicacion

Edición

Editorial

Agregar Referencia

Ilustración 67 Formulario de referencias

En el lado derecho de la pantalla se mostrará el detalle de referencias creadas, además de las funcionalidades para editar y eliminar una referencia

Referencias

	Titulo	Autores	Publicación	Tipo	Acciones
-	Fundamentos de sistemas digitales	Floyd, Thomas L.		Libro	✎ ✖
Thomas, F. (s.f). Fundamentos de sistemas digitales					

< 1 >

Ilustración 68 Listado de referencias

3.2.5 Ejercicios

Esta pantalla mostrara al coordinador el listado de ejercicios creados por los docentes de la asignatura, y bajo el mismo esquema de las plantillas podrá editar los ejercicios según sea el caso.

una plantilla y ejercicios de esta, el coordinador podrá crear, eliminar y confirmar la práctica.

Dashboard / Programación / Prácticas

[Prácticas generadas](#)

Nueva practica				
Periodo	Plantilla	Creado	Lista de ejercicios	Acciones
59	Practica prueba	15 de mayo de 2022	> 2 Ejercicios	Confirmar practica Confirmar practica con soluciones Eliminar practica
59	Representación de algoritmos	15 de mayo de 2022	> 1 Ejercicio	Confirmar practica Confirmar practica con soluciones Eliminar practica

El formulario para crear una práctica solicitará al coordinador la selección de una plantilla y ejercicios para la práctica.

Dashboard / Programación / Prácticas / Nueva Practica

Plantilla

Seleccione una plantilla

- Practica prueba
- Representación de algoritmos

Tema

Dificultad

No Data

Ilustración 71 Formulario para crear una practica

<input type="checkbox"/>	Título	Descripción	Tema	Dificultad	Utilizados	
<input type="checkbox"/>	Llamadas telefónicas	Enlace cada proposición con su formalización	Álgebra de boole	Medio	5	•
<input type="checkbox"/>	Formalización de proposiciones	Enlace cada proposición con su formalización	Álgebra de boole	Medio	1	•
<input type="checkbox"/>	Prueba	Practica	Compuertas lógicas	Medio	1	•

Ilustración 72 Listado de ejercicios

Anexo 3: Manual de usuario del docente



UNIVERSIDAD POLITECNICA SALESIANA

MANUAL DE USUARIO QUICKLAB

Docente

Contenido

1. Uso del documento	31
2. Ámbito del trabajo	32
3. Funcionamiento de Quicklab	32
3.1 Navegación por la web	33
3.2 Manual de uso de la web	35
3.2.1 Asignatura.....	35
3.2.2 Referencias	37
3.2.3 Ejercicios	38
3.2.5 Gestión de practicas.....	39

Ilustraciones

Ilustración 1 Captura del formulario de inicio de sesión.....	33
Ilustración 2 Formulario para restablecer contraseña.....	34
Ilustración 3 Dashboard del coordinador de asignatura	34
Ilustración 4 Menú de funcionalidades.....	35
Ilustración 5 Dashboard de la asignatura programación	36
Ilustración 6 Grafica de temas usados por ejercicio	36
Ilustración 7 Grafica de calificación de ejercicios.....	36
Ilustración 8 Docentes de la asignatura de programación	37
Ilustración 9 Formulario de Referencias	37
Ilustración 10 Listado de referencias.....	38
Ilustración 11 Practicas generadas.....	39

1. Uso del documento

Quicklab es un sistema web creado para optimizar la gestión de prácticas de laboratorio del claustro docentes de programación de la Universidad Politécnica Salesiana.

La estructura del documento es el siguiente:

Capítulo 2. Ámbito del trabajo

En este capítulo se presenta una visión general del contexto asociado al trabajo y encuadrado al objetivo de la web presentada en este manual en base al perfil de coordinador de la asignatura.

Capítulo 3. Funcionamiento de la web

Este capítulo muestra los tutoriales asociados al rol de coordinador de asignatura, incluyendo descripciones de las funcionalidades en cada sección y sus limitaciones.

2. Ámbito del trabajo

El trabajo en el que se enmarca este sistema web propone la implantación de la herramienta Quicklab en la infraestructura de la Universidad Politécnica. Para la creación de una práctica es necesario establecer temas, objetivos ejercicios, etc. Quicklab permite al usuario hacerlo de una manera dinámica y en el menor tiempo posible. Además de plantear un sistema capaz de generar de forma automática las practicas, se destaca el uso de herramientas y recursos utilizados, aprovechando la información existente, para extraer, organizar, seleccionar y publicar las practicas clasificadas por tema y por cada periodo existente dentro del sistema.

Para la gestión de carreras dentro del sistema se ha creado el usuario “Docente” con el que se gestionará la funcionalidad del sistema.

Las funciones principales que tiene el administrador son:

- Administración de referencias
- Creación y administración de ejercicios
- Descarga de practicas

3. Funcionamiento de Quicklab

Esta sección se divide en dos partes. En primer lugar, se muestra la navegación a lo largo de la web, con las distintas secciones que ofrece para cada tipo de usuario. En

segundo lugar, se muestran los tutoriales para cada funcionalidad de las descritas en la sección anterior.

3.1 Navegación por la web

Para acceder a la página, se accede mediante la siguiente url:

<https://quicklab.csuioups.org/>

Para acceder al sistema se lo puede hacer mediante el botón “iniciar sesión” con las credenciales establecidas por el administrador como lo muestra la siguiente imagen:



The image shows a login interface for the Universidad Politécnica Salesiana Ecuador. At the top left is the university's logo, which features a globe and a portrait of a man. To the right of the logo, the text reads 'UNIVERSIDAD POLITÉCNICA SALESIANA ECUADOR'. Below the logo and text, there are two input fields: 'Correo electrónico' with the value 'rtufino@ups.edu.ec' and 'Contraseña' with masked characters. A blue 'Iniciar' button is positioned below the password field. At the bottom left, there is a link that says 'Restablecer contraseña'.

Ilustración 73 Captura del formulario de inicio de sesión

Quicklab cuenta con la funcionalidad de recuperar la contraseña en caso de olvidarla, para poder recuperar la contraseña es necesario el correo electrónico con el que se inicia sesión, el sistema enviará un pin mediante el cual se podrá cambiar la contraseña

Recuperar contraseña



**UNIVERSIDAD POLITÉCNICA
SALESIANA
ECUADOR**

Correo electrónico

Restablecer

Ilustración 74 Formulario para restablecer contraseña

Una vez registrado el usuario, se accede a la página principal como se muestra en la Ilustración 3:

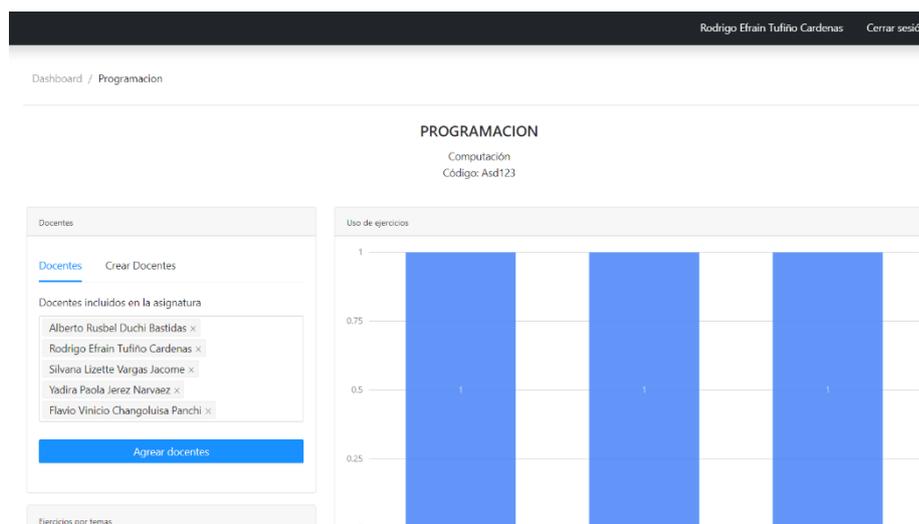


Ilustración 75 Dashboard del coordinador de asignatura

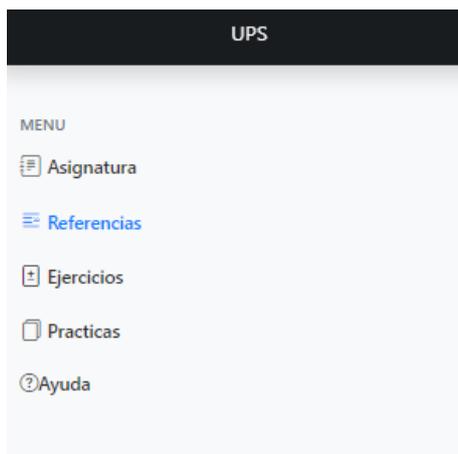


Ilustración 76 Menú de funcionalidades

Las opciones que se encuentran en las ilustraciones anteriores corresponden a la funcionalidad del sistema para el administrador.

3.2 Manual de uso de la web

En este punto se mostrará el tutorial relativo a cada funcionalidad del sistema

3.2.1 Asignatura

Al ingresar a la opción de asignatura, el sistema nos mostrara el Dashboard con información relevante de la asignatura a la que el docente pertenece, además se muestra la información relevante a los docentes que pertenecen a la misma, graficas de los ejercicios y temas utilizados y la calificación de los ejercicios previamente cargados:

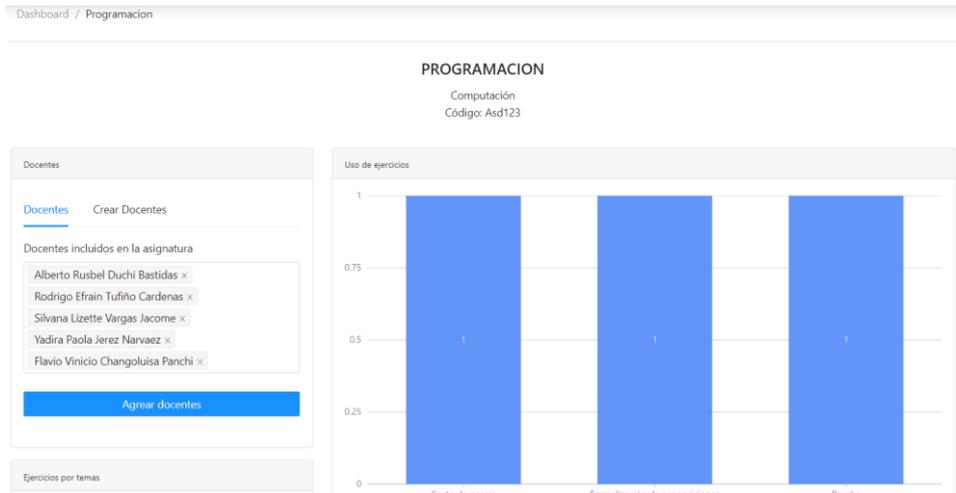


Ilustración 77 Dashboard de la asignatura programación

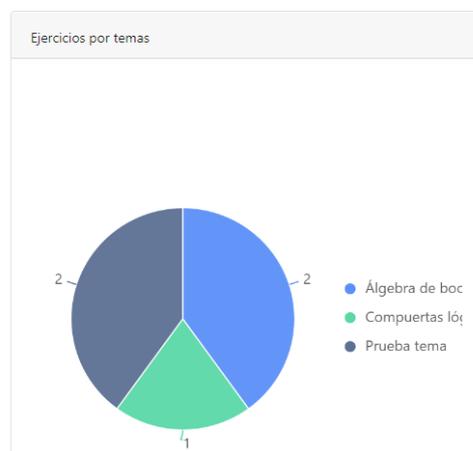


Ilustración 78 Grafica de temas usados por ejercicio

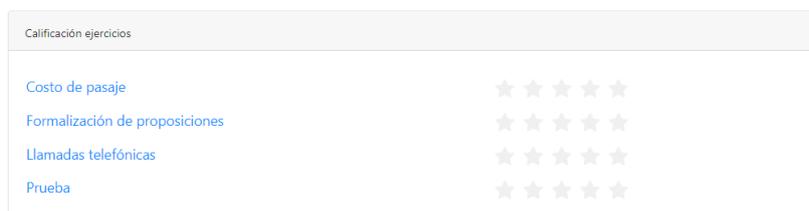


Ilustración 79 Grafica de calificación de ejercicios

En la sección de docentes, la herramienta muestra los docentes que pertenecen a la asignatura.

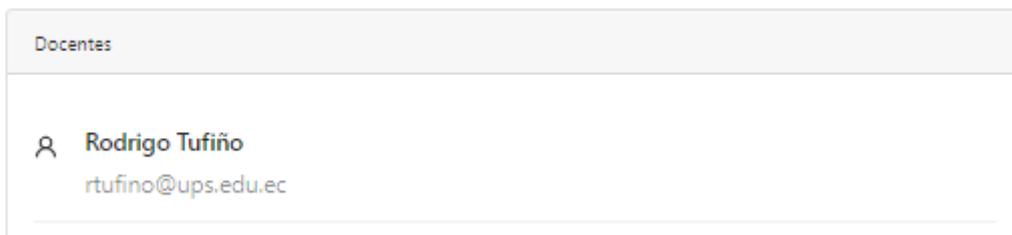


Ilustración 80 Docentes de la asignatura de programación

3.2.2 Referencias

Esta funcionalidad permite la administración de referencias de la guía de laboratorio, el sistema solicitará la información necesaria para generar una referencia basando en la norma APA séptima edición, la referencia puede ser de sitio web o libro según sea el caso, la información requerida es la mostrada en la Ilustración 9

Ilustración 81 Formulario de Referencias

En el lado derecho de la pantalla se mostrará el detalle de referencias creadas, además de las funcionalidades para editar y eliminar una referencia

Referencias

Título	Autores	Publicación	Tipo	Acciones
- Fundamentos de sistemas digitales	Floyd, Thomas L.		Libro	 
Thomas, F. (s.f.). Fundamentos de sistemas digitales				

< 1 >

Ilustración 82 Listado de referencias

3.2.3 Ejercicios

Esta pantalla mostrara el listado de ejercicios creados por el docente, y bajo el mismo esquema de las Referencias podrá editar los ejercicios según sea el caso.

Dashboard / Programacion / Ejercicios

Listado de ejercicios [Crear nuevo ejercicio](#)

	Periodo	Título	Tema	Dificultad	Docente	Calificacion	Tipo	Archivado	Acciones
+	59	Costo de pasaje	Prueba tema	Medio	Rodrigo T.	☆☆☆☆☆	Practica	Sin Archivar	 
+	59	Formalización de proposiciones	Álgebra de boole	Medio	Rodrigo T.	☆☆☆☆☆	Practica	Sin Archivar	 
+	60	Llamadas telefónicas	Álgebra de boole	Medio	Rodrigo T.	☆☆☆☆☆	Practica	Sin Archivar	 
+	59	Prueba	Compuertas lógicas	Medio	Rodrigo T.	☆☆☆☆☆	Practica	Sin Archivar	 

< 1 >

Ilustración 10 Listado de ejercicios

Para crear o editar un ejercicio se mostrará un formulario donde se deben ingresar obligatoriamente el título, dificultad, tipo y tema del ejercicio, la herramienta permite ingresar de manera opcional el ejercicio, solución y ejemplo de este.

