



**Desarrollo de un sistema web con metodología DevOps utilizando la herramienta Jenkins, para optimizar la gestión de reservas y consultas en línea de los servicios que ofrece SL Laboratorio Clínico y Bacteriológico**

Guangasi Paste, Dario Javier y Sambachi Chilig, Patricia Aracely

Departamento de Ciencias de la Computación

Carrera de Ingeniería en Software

Trabajo de titulación, previo a la obtención del título de Ingeniero en Software

Msc. Corral Diaz, María Alexandra

27 de Julio del 2022

Latacunga



DOCUMENTO DE TESIS\_Guangasi\_Sambachi\_v02.pdf

Scanned on: 14:22 July 28, 2022 UTC



Overall Similarity Score




Results Found



Total Words in Text

Identical Words	327
Words with Minor Changes	305
Paraphrased Words	730
Omitted Words	3208

  
-----  
Msc. Corral Diaz, María Alexandra



**Departamento de Ciencias de la Computación**

**Carrera de Ingeniería en Software**

**Certificación**

Certifico que el trabajo de titulación: **“Desarrollo de un sistema web con metodología DevOps utilizando la herramienta Jenkins, para optimizar la gestión de reservas y consultas en línea de los servicios que ofrece SL Laboratorio Clínico y Bacteriológico”** fue realizado por el señor **Guangasi Paste, Dario Javier** y la señorita **Sambachi Chilig, Patricia Aracely**; el mismo que cumple con los requisitos legales, teóricos, científicos, técnicos y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, además fue revisado y analizado en su totalidad por la herramienta de prevención y/o verificación de similitud de contenidos; razón por la cual me permito acreditar y autorizar para que se lo sustente públicamente.

**Latacunga, 27 de julio de 2022**

**Msc. Corral Diaz, María Alexandra**

C.C.: 0501970487



**Departamento de Ciencias de la Computación**

**Carrera de Ingeniería en Software**

**Responsabilidad de Autoría**

Nosotros, **Guangasi Paste, Dario Javier**, con cédula de ciudadanía No. 1804879961 y **Sambachi Chilig, Patricia Aracely**, con cédula de ciudadanía No. 1725246225, declaramos que el contenido, ideas y criterios del trabajo de titulación: **“Desarrollo de un sistema web con metodología DevOps utilizando la herramienta Jenkins, para optimizar la gestión de reservas y consultas en línea de los servicios que ofrece SL Laboratorio Clínico y Bacteriológico”** es de nuestra autoría y responsabilidad, cumpliendo con los requisitos legales, teóricos, científicos, técnicos, y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, respetando los derechos intelectuales de terceros y referenciando las citas bibliográficas.

**Latacunga, 27 de julio de 2022**

**Guangasi Paste, Dario Javier**

C.C.: 1804879961

**Sambachi Chilig, Patricia Aracely**

C.C.: 1725246225



**Departamento de Ciencias de la Computación**

**Carrera de Ingeniería en Software**

**Autorización de Publicación**

Nosotros, **Guangasi Paste, Dario Javier**, con cédula de ciudadanía No. 1804879961 y **Sambachi Chilig, Patricia Aracely**, con cédula de ciudadanía No. 1725246225, autorizamos a la Universidad de las Fuerzas Armadas ESPE publicar el trabajo de titulación: **“Desarrollo de un sistema web con metodología DevOps utilizando la herramienta Jenkins, para optimizar la gestión de reservas y consultas en línea de los servicios que ofrece SL Laboratorio Clínico y Bacteriológico”** en el Repositorio Institucional, cuyo contenido, ideas y criterios son de mi/nuestra responsabilidad.

**Latacunga, 27 de julio de 2022**

**Guangasi Paste, Darío Javier**

C.C.: 1804879961

**Sambachi Chilig, Patricia Aracely**

C.C.: 1725246225

### **Dedicatoria**

Quiero dedicar el presente trabajo a mis padres Segundo Guangasi y Mariana Paste quienes supieron guiarme, aconsejarme y darme ánimo a lo largo de todo este camino.

A mi novia Lisbeth Quinapanta quien fue un pilar fundamental en todo este proceso de aprendizaje, quien con su alegría, sabiduría y consejos supo guiarme en todo este proceso de formación.

A mis hermanos que han sido un motor fundamental para seguir adelante, que con sus consejos y sabiduría supieron apoyarme y aconsejarme.

Y sobre todo a Dios, quien ha forjado mi camino, me ha guiado por el sendero correcto en todo momento, ayudándome a aprender de mis errores y a no volver a caer en ellos, gracias por ser el conductor de mi vida y guiarme en todo este proceso.

**Dario Javier Guangasi Paste**

### **Dedicatoria**

Este proyecto de titulación se la dedico con todo mi esfuerzo y cariño:

A Dios, por estar siempre guiar mis pasos, por darme salud y fortaleza para poder cumplir una meta más en mi vida.

A mis queridos padres José Sambachi y Ana Chilig por apoyarme en cada momento, por sus consejos, por motivarme a seguir luchando por mis sueños y sobre todo por su infinito amor.

A mi hermano Xavier por el apoyo y consejos que siempre me brindo durante todo el transcurso de mi carrera y vida, quien nunca se ha negado en tenderme la mano en momentos difíciles.

A mi Primas Hermanas Patty y Khrystel por su apoyo y cariño que siempre me han dado en el transcurso de mi vida, quienes siempre me han ayudado y aconsejado en momentos difíciles.

Y finalmente a todas las personas que conocí en el transcurso de mi vida en especial a quienes actualmente siguen conmigo, prometo seguir adelante mejorando día a día para ser una mejor persona tanto personalmente como profesionalmente.

**Patricia Aracely Sambachi Chilig**

## **Agradecimiento**

Quiero agradecer a mi familia por todo el apoyo y la compañía que supieron brindarme durante esta etapa de mi vida.

A mi novia Lisbeth Quinapanta por siempre apoyarme en todas las decisiones tomadas a lo largo de este camino y ser la guía para esta meta.

A mi compañera Patricia Sambachi por todo el esfuerzo, trabajo duro y por su amistad no solo en este presente trabajo sino también a lo largo de la vida universitaria

A nuestra tutora Ing. María Corral porque supo guiarnos en todo el proceso de titulación.

Por último, agradecer a Dios por darme la dicha de compartir día a día con mis seres queridos, mi familia, amigos y por dejarme terminar este ciclo con el mayor de los éxitos.

**Darío Javier Guangasi Paste**



## **Agradecimiento**

En primer lugar, agradezco a Dios por haberme acompañado y guiado cada uno de mis pasos a lo largo de esta etapa universitaria, por brindarme sabiduría y fortaleza en aquellos días de debilidad, pero especialmente por darme una familia maravillosa.

Agradezco a mis padres por todo el cariño, comprensión, esfuerzo y apoyo incondicional que me han dado para culminar una etapa más de mi vida, este logro se lo debo a ustedes.

A mis amigos Erick y Shirley quienes han compartido conmigo esta trayectoria universitaria, por los momentos que compartimos, pero sobre todo por ayudarme en los momentos difíciles de mi vida.

A mis maestros por su enseñanzas y conocimientos durante todo este proceso y darnos aliento a seguir adelante.

A la Msc. María Alexandra Corral por ser una excelente tutora y maestra durante todo este tiempo, estoy muy agradecida por guiarnos y aconsejarnos para poder culminar este proyecto de titulación, por ser muy paciente y brindarnos sus valiosos conocimientos.

A la BQ.CL Sylvia Luján por creer en nosotros y habernos dado la oportunidad de realizar nuestro trabajo de titulación en su Laboratorio Clínico, por todo el apoyo y facilidad que nos brindó para poderlo desarrollar e implementar.

Y finalmente a la Universidad de las Fuerzas Armadas especialmente a la Carrera de Ingeniería en Software y todos sus docentes por abrirme las puertas y haberme brindado tantas oportunidades para desarrollarme como toda una profesional.

**Patricia Aracely Sambachi Chilig**

## ÍNDICE DE CONTENIDOS

Carátula .....	1
Reporte de verificación de contenido.....	2
Certificación .....	3
Responsabilidad de autoría .....	4
Autorización de publicación .....	5
Dedicatoria .....	6
Dedicatoria .....	7
Agradecimiento.....	8
Agradecimiento.....	9
Índice de contenidos .....	10
Índice de tablas .....	15
Índice de figuras .....	16
Resumen.....	19
Abstract .....	20
Capítulo I: Presentación del problema .....	21
Planteamiento y formulación del problema.....	21
Justificación e importancia .....	23
Objetivos.....	24
<i>Objetivo General</i> .....	24
<i>Objetivos Específicos</i> .....	24
Meta.....	24
Hipótesis .....	25

	11
<b>Variables de la investigación.....</b>	<b>25</b>
<i>Variable Dependiente .....</i>	<i>25</i>
<i>Variable Independiente.....</i>	<i>25</i>
<b>Conceptualización de la Variable Independiente. ....</b>	<b>25</b>
<b>Indicadores .....</b>	<b>26</b>
<b>Capítulo II: Marco teórico .....</b>	<b>27</b>
<b>Introducción.....</b>	<b>27</b>
<b>Historia de la web.....</b>	<b>27</b>
<b>Laboratorio clínico y bacteriológico .....</b>	<b>30</b>
<b>Características para una prueba diagnóstica.....</b>	<b>31</b>
<b>Devops .....</b>	<b>32</b>
<b>Ciclo de vida DevOps.....</b>	<b>34</b>
<i>Planeación.....</i>	<i>35</i>
<i>Codificación .....</i>	<i>35</i>
<i>Construcción .....</i>	<i>36</i>
<i>Pruebas .....</i>	<i>36</i>
<i>Lanzamiento.....</i>	<i>37</i>
<i>Despliegue .....</i>	<i>37</i>
<i>Operación.....</i>	<i>38</i>
<i>Monitoreo .....</i>	<i>38</i>
<b>Modelo C4 .....</b>	<b>39</b>
<i>Nivel 1: Diagrama de contexto.....</i>	<i>39</i>
<i>Nivel 2: Diagrama de Contenedores.....</i>	<i>40</i>
<i>Nivel 3: Diagrama de componentes.....</i>	<i>41</i>
<i>Nivel 4: El código.....</i>	<i>42</i>

Jira.....	43
Intellij IDE.....	44
Spring Boot.....	45
Servicios Rest.....	46
Spring Security.....	47
JWT .....	48
Ciclo de vida de un Token JWT .....	49
Bootstrap .....	50
Angular.....	50
<i>Beneficios del desarrollo con Angular.....</i>	<i>51</i>
Java .....	51
Maven.....	52
Github.....	53
Jenkins.....	54
<i>Integración continua con Jenkins .....</i>	<i>55</i>
<i>Flujo de integración continua con Jenkins.....</i>	<i>56</i>
JUnit.....	57
Docker .....	58
DockerFile.....	59
Docker Compose.....	60
Docker Hub .....	61
Liquibase .....	62
Jhipster .....	64

	13
Nginx .....	65
Digital Ocean .....	65
Certbot .....	66
<b>Capítulo III: Desarrollo del sistema.....</b>	<b>67</b>
Introducción del capítulo .....	67
Gestión de requisitos .....	67
<i>Épicas</i> .....	68
Historia de usuario .....	69
Creación del product backlog .....	76
Arquitectura de software .....	77
<i>Diagrama de contexto del sistema</i> .....	77
<i>Diagrama de contenedores del sistema</i> .....	78
Implementación y configuración.....	79
<i>Configuración del repositorio central</i> .....	79
<i>Configuración de Jenkins</i> .....	81
Instalar Jenkins en un servidor en la nube.....	82
Conexión de Jenkins con GitHub.....	83
Configuración de Jenkins con Docker.....	85
<i>Construcción de iteración</i> .....	87
Fase de planificación .....	87
Fase de Codificación.....	89
Fase de Pruebas .....	90
Fase de Construcción .....	92
Fase de Release.....	93
Fase de Despliegue .....	95

Fase de Monitoreo .....	96
<b>Capítulo IV: Análisis de resultados .....</b>	<b>98</b>
Aceptación del sistema.....	98
Recolección de datos.....	100
Análisis de resultados .....	105
Discusión de los resultados .....	107
<b>Capítulo V:Conclusiones y recomendaciones.....</b>	<b>108</b>
Conclusiones.....	108
Recomendaciones.....	109
<b>Bibliografía .....</b>	<b>111</b>
<b>Anexos.....</b>	<b>119</b>

## ÍNDICE DE TABLAS

<b>Tabla 1</b> <i>Criterios de aplicabilidad y de confiabilidad</i> .....	31
<b>Tabla 2</b> <i>Historia de usuario 1</i> .....	69
<b>Tabla 3</b> <i>Historia de usuario 2</i> .....	70
<b>Tabla 4</b> <i>Historia de usuario 3</i> .....	70
<b>Tabla 5</b> <i>Historia de usuario 4</i> .....	71
<b>Tabla 6</b> <i>Historia de usuario 5</i> .....	72
<b>Tabla 7</b> <i>Historia de usuario 6</i> .....	72
<b>Tabla 8</b> <i>Historia de usuario 7</i> .....	73
<b>Tabla 9</b> <i>Historia de usuario 8</i> .....	73
<b>Tabla 10</b> <i>Historia de usuario 9</i> .....	74
<b>Tabla 11</b> <i>Historia de usuario 10</i> .....	75
<b>Tabla 12</b> <i>Historia de usuario 11</i> .....	75
<b>Tabla 13</b> <i>Historia de usuario 12</i> .....	76
<b>Tabla 14</b> <i>Encuesta SUS</i> .....	99
<b>Tabla 15</b> <i>Resultados de test SUS</i> .....	99
<b>Tabla 16</b> <i>Cálculo de la media de la diferencia de tiempos</i> .....	106

## ÍNDICE DE FIGURAS

<b>Figura 1</b> <i>Ciclo de una prueba de laboratorio</i> .....	32
<b>Figura 2</b> <i>Fases del ciclo de vida DevOps</i> .....	34
<b>Figura 3</b> <i>Diagrama de contexto o nivel 1</i> .....	40
<b>Figura 4</b> <i>Diagrama de contenedores o nivel 2</i> .....	41
<b>Figura 5</b> <i>Diagrama de componentes o nivel 3</i> .....	42
<b>Figura 6</b> <i>El Código o nivel 4</i> .....	43
<b>Figura 7</b> <i>Proceso completo del JWT</i> .....	49
<b>Figura 8</b> <i>Proceso de integración continua (CI) utilizando Jenkins</i> .....	56
<b>Figura 9</b> <i>Flujo de trabajo de dockerfile</i> .....	60
<b>Figura 10</b> <i>Ejemplo del sistema de control de versiones de liquibase</i> .....	63
<b>Figura 11</b> <i>Product backlog con tareas por hacer</i> .....	77
<b>Figura 12</b> <i>Diagrama de contexto del sistema</i> .....	78
<b>Figura 13</b> <i>Diagrama de contenedores del sistema</i> .....	79
<b>Figura 14</b> <i>Repositorio backend de código del proyecto</i> .....	80
<b>Figura 15</b> <i>Repositorio frontend de código del proyecto</i> .....	80
<b>Figura 16</b> <i>Flujo de github para la creación de tareas y hotfix</i> .....	81
<b>Figura 17</b> <i>Comandos para la instalación de java en un servidor en la nube</i> .....	82
<b>Figura 18</b> <i>Comandos para la instalación de jenkins en un servidor en la nube</i> .....	82
<b>Figura 19</b> <i>Entorno de ejecución de Jenkins</i> .....	83
<b>Figura 20</b> <i>Conexión de github con Jenkins</i> .....	84
<b>Figura 21</b> <i>Configuración de un job de Jenkins junto con GitHub</i> .....	84



<b>Figura 22</b> <i>Comandos de instalación Docker</i> .....	85
<b>Figura 23</b> <i>Comando para descargar una imagen maven en docker</i> .....	85
<b>Figura 24</b> <i>Ambiente de Jenkins para el proyecto</i> .....	86
<b>Figura 25</b> <i>Job ejecutado con éxito</i> .....	86
<b>Figura 26</b> <i>Falla de un job en un test</i> .....	87
<b>Figura 27</b> <i>Gestión de tareas y sprints en jira</i> .....	87
<b>Figura 28</b> <i>Hoja de ruta en jira</i> .....	88
<b>Figura 29</b> <i>Tablero kanban</i> .....	89
<b>Figura 30</b> <i>Ramas creadas para la gestión de código fuente</i> .....	90
<b>Figura 31</b> <i>Configuración de jenkins para ejecución de pruebas unitarias sl-back-end</i> .....	90
<b>Figura 32</b> <i>Configuración de jenkins para ejecución de pruebas unitarias sl-frontend</i> .....	91
<b>Figura 33</b> <i>Finalización de ejecución de pruebas unitarias en el job de jenkins sl-back-end</i>	91
<b>Figura 34</b> <i>Finalización de ejecución de pruebas unitarias en el job de jenkins sl-frontend</i> ..	92
<b>Figura 35</b> <i>Versiones en jira</i> .....	93
<b>Figura 36</b> <i>Contenido del archivo dockerfile aplicación sl-back-end</i> .....	93
<b>Figura 37</b> <i>Contenido del archivo dockerfile aplicación sl-frontend</i> .....	94
<b>Figura 38</b> <i>Listado de imágenes docker en docker-hub aplicación sl-back-end</i> .....	94
<b>Figura 39</b> <i>Listado de imágenes docker en docker-hub aplicación sl-frontend</i> .....	95
<b>Figura 40</b> <i>Archivo .yml para la descarga y despliegue de la aplicación sl-back-end</i> .....	95
<b>Figura 41</b> <i>Archivo .yml para la descarga y despliegue de la aplicación sl-frontend</i> .....	96
<b>Figura 42</b> <i>Archivo .sh para limpiar los contenedores e imágenes docker</i> .....	96
<b>Figura 43</b> <i>Monitoreo del uso de memoria (digital ocean)</i> .....	97

<b>Figura 44</b> <i>Monitoreo del ancho de banda (digital ocean)</i> .....	97
<b>Figura 45</b> <i>Segunda pregunta de la encuesta</i> .....	101
<b>Figura 46</b> <i>Tercera pregunta de la encuesta</i> .....	102
<b>Figura 47</b> <i>Cuarta pregunta de la encuesta</i> .....	102
<b>Figura 48</b> <i>Quinta pregunta de la encuesta</i> .....	103
<b>Figura 49</b> <i>Segunda pregunta de la encuesta con el aplicativo</i> .....	104
<b>Figura 50</b> <i>Tercera pregunta de la encuesta con el aplicativo</i> .....	104
<b>Figura 51</b> <i>Cuarta pregunta de la encuesta con el aplicativo</i> .....	105
<b>Figura 52</b> <i>Quinta pregunta de la encuesta con el aplicativo</i> .....	105

## Resumen

El presente proyecto de titulación está orientado al desarrollo de un sistema web que permita mejorar la gestión de reservas y consultas en línea de los servicios que ofrece SL Laboratorio Clínico y Bacteriológico, da una solución a las necesidades del personal administrativo, médico y sus pacientes, cuya dificultad principal es la reserva de citas, asesoría en línea de los servicios y consultar de forma online los resultados de los exámenes clínicos, evitando así a sus pacientes y médicos tratantes ir presencialmente al establecimiento para realizar alguna de estas situaciones, con el fin de poder brindar una forma más eficiente de realizarlos, reducir tiempos de espera y mejorar la atención brindada a los pacientes. Este problema se determinó a partir de la correspondiente entrevista e investigación efectuada al personal administrativo, médico y especialmente a los clientes del Laboratorio clínico, observando la manera de ejecutar los procesos antes mencionados y el tiempo que estos toman para llevar a cabo. Además, para el desarrollo de este proyecto se empleó la metodología DevOps que permite que cada módulo sea autónomo, independiente, escalable, mantenible. Se utilizó el Modelo C4 para describir la arquitectura del sistema web en sus diferentes niveles de abstracción, cada uno útil para diferentes grupos de espectadores. Gracias a este sistema web, SL Laboratorio Clínico y Bacteriológico actualmente cuenta con un mejor registro de citas, controles de existencias y exámenes clínicos realizados, tener acceso a la información y la disponibilidad de la misma en el momento que se requiera, alcanzando así una mayor eficiencia en la atención a sus pacientes, médicos y personal administrativo.

*Palabras Clave:* Metodología DevOps, Sistema Web, Modelo C4, Spring Boot, framework Angular

## **Abstract**

This degree project is aimed at developing a web system to improve the management of online booking and consultation of the services offered by SL Laboratorio Clínico y Bacteriológico, providing a solution to the needs of administrative and medical staff and their patients, whose main difficulty is booking appointments, online consultation of services and online consultation of the results of clinical tests, thus avoiding their patients and treating physicians to go in person to the establishment to perform any of these situations, in order to provide a more efficient way to perform them, reduce waiting times and improve the care provided to patients. This problem was determined from the corresponding interview and research carried out with the administrative and medical staff, and especially with the clients of the clinical laboratory, observing the way in which the aforementioned processes are carried out and the time they take to complete them. In addition, the DevOps methodology was used for the development of this project, which allows each module to be autonomous, independent, scalable and maintainable. The C4 Model was used to describe the architecture of the web system in its different levels of abstraction, each one useful for different groups of viewers. Thanks to this web system, SL Clinical and Bacteriological Laboratory currently has a better record of appointments, stock controls and clinical tests performed, access to information and availability of the same at the time it is required, thus achieving greater efficiency in the care of their patients, doctors and administrative staff.

Translated with [www.DeepL.com/Translator](http://www.DeepL.com/Translator) (free version)

*Keywords:* DevOps Methodology, Web System, C4 Model, Spring Boot, Angular framework

## Capítulo I

### Presentación del Problema

#### Planteamiento y Formulación del Problema

En la actualidad el campo de la medicina en el área de laboratorios clínicos exige una mejora continua, actualizando sus técnicas, procedimientos, equipo y sistema que permita ofrecer resultados e información de manera más oportuna y exacta. Los laboratorios de análisis clínicos deben utilizar herramientas, metodologías y normas de calidad que lleven a una optimización de recursos, tanto materiales como tecnológicos, así como también llevar a una mejora continua en la calidad de los servicios que la misma ofrece.

Ahora con la era tecnológica que avanza rápidamente, el desarrollo de sistemas web han alcanzado un gran sitio, en esta época es muy habitual ver diariamente sitios web que ofrecen sus productos, procesos y servicios, sean estos cosméticos, tecnológicos, belleza, alimenticios y dentro de campo de la salud no es la excepción, por lo tanto, implementar un sistema web es de vital importancia, puesto que es una herramienta que ahorra recursos materiales, tiempo y dinero, mejorando la calidad de los productos o servicios y sobre todo forjando un valor extra para la empresa.

Actualmente, un laboratorio clínico es un sector primordial para el área médica, puesto que por medio de sus servicios se puede diagnosticar enfermedades así diagnosticar un tratamiento óptimo al paciente, por lo tanto, se ha convertido en un área que alberga una gran cantidad de pacientes.

SL Laboratorio clínico y bacteriológico es una organización no gubernamental, apolítica, que presta servicios y productos de Salud con énfasis en exámenes médicos para la población ecuatoriana. En esta institución la gestión de exámenes de laboratorio se almacena en archivos físicos, siendo así una combinación compleja de actividades y procesos. A pesar de que la institución cuenta con un sistema informático poco amigable para la interacción con el usuario, se ha visto en la necesidad de implementar un sistema web.

Con el uso de esta tecnología web se van a solventar muchos problemas generados por la recopilación de información de los pacientes de forma manual. Por otra parte, en la actualidad, con la pandemia que estamos sufriendo actualmente por la enfermedad de coronavirus (COVID-19), ha logrado un cambio del comportamiento en todas las personas, como también ha cambiado la percepción e importancia de crear herramientas virtuales en la Salud para ayuda de las consultas médicas, de exámenes médicos, diagnósticos, reserva de citas, y así no exponer a los pacientes a las aglomeraciones y al riesgo de contagio de esta nueva enfermedad.

El sistema integrando en la institución SL Laboratorio Clínico y Bacteriológico gestiona la información y resultados de los pacientes en forma tradicional, es decir que el paciente se presenta de forma presencial al Laboratorio, este presenta la orden del examen a la secretaría, está verifica la orden, revisa si hay turnos, pregunta datos personales y los escribe en la orden para posteriormente ingresar los datos por teclado a un sistema de escritorio.

Con este proceso se ocasiona pérdida de tiempo y un posible contagio de covid-19 por la aglomeración que pueda surgir en el laboratorio o al dirigirse al laboratorio como en el transporte público, por esta razón se ha planteado a solucionar el problema mediante la implementación de un sitio web, este permitirá que la gestión administrativa del laboratorio (Reserva, registro, verificar los resultados en línea de exámenes y consultar información en línea de los servicios que ofrece el laboratorio clínico) esté disponible las 24 horas de manera eficaz y eficiente, como también para el paciente sea más segura, así tener un buen control en el laboratorio y brindar un excelente servicio a sus pacientes.

Para el desarrollo del sistema web se usará la metodología DEVOPS utilizando la herramienta JENKINS. Este es un nuevo movimiento que trata de mejorar la agilidad en la prestación de servicios. Fomenta una mayor colaboración y comunicación entre los equipos de desarrollo y operaciones, evitando que los problemas y las necesidades operacionales se mantengan sub expuestas en el proyecto, afectando así la calidad del software.

Con base en la problemática antes expuesta se formuló la siguiente pregunta:  
¿Cómo optimizar la gestión de reservas y consultas en línea de los servicios que ofrece SL Laboratorio Clínico y Bacteriológico?

### **Justificación e Importancia**

En la actualidad, para que una empresa, organización o institución sea muy competitiva, eficiente y moderna, necesita estar actualizada con las nuevas tecnologías para que así pueda llegar a sus clientes de mejor manera, más eficiente y eficaz con sus servicios y productos.

Por lo que disponer de información y datos en un tiempo real las 24 horas del día hace que el usuario sea muy indispensable para una buena gestión y desempeño del laboratorio clínico. En esta época los sistemas de información Web ayudan a agilizar los distintos procesos de dicha institución y sobre todo la productividad.

Ante la necesidad de cambios que nos ha puesto la pandemia que estamos viviendo actualmente, un laboratorio clínico es una unidad funcional muy importante en la actualidad para el área de medicina, ya que, si no contamos con un área de laboratorio, los médicos no harían diagnósticos eficientes de los diferentes tipos de enfermedades que acomplejan al paciente, y mediante este diagnóstico saber el tipo de tratamiento puede dar al paciente para su mejoría.

Por esa razón, se planteó el desarrollo de un sistema web utilizando la metodología DEVOPS, para optimizar la gestión de las reservas y consulta de resultados en línea de exámenes clínicos del “SL Laboratorio Clínico y Bacteriológico” que permite mejorar la calidad de procesos y servicios que este ofrece.

Con esta implementación mejoramos el proceso tradicional que era acercarse presencialmente al laboratorio clínico coger turno para realizarse el examen clínico, si hay turnos se procede a la toma la muestra y finalmente esperar hasta que nos den los resultados, esto es muy tedioso para el paciente o cliente, ya que muchas veces tenemos

pacientes que necesitan de sus resultados en una fecha establecida y no cuentan con tiempo para esperar y recoger los resultados de sus análisis, es por ello que la falta de un sistema que les facilite la información en el momento que el cliente no pueda estar en el laboratorio presente para coger un turno o recoger sus resultados es de vital importancia tanto para el cliente como para la institución.

## **Objetivos**

### ***Objetivo General***

Desarrollar un sistema web con metodología DEVOPS utilizando la herramienta JENKINS, para mejorar la gestión de reservas y consultas en línea de los servicios que ofrece SL Laboratorio Clínico y Bacteriológico.

### ***Objetivos Específicos***

- Elaborar el marco teórico acerca de la metodología DEVOPS para conocer y aplicar avances metodológicos referidos en la entrega continua del sistema.
- Configurar un entorno de despliegue continuo con la herramienta JENKINS.
- Desarrollar un sistema web basándonos en la metodología DEVOPS.
- Implementar el sistema web en el establecimiento SL laboratorio clínico bacteriológico de la ciudad de Machachi.
- Validar el sistema mediante pruebas de funcionalidad y la valoración del laboratorista encargado.

## **Meta**

Desarrollar un sistema web con metodología DEVOPS utilizando la herramienta JENKINS, para optimizar la gestión de reservas y consultas en línea de los servicios que ofrece SL Laboratorio Clínico y Bacteriológico.



- Se aplicará la metodología de desarrollo DEVOPS, basado en la integración entre desarrolladores y administradores de sistemas, para la construcción del sistema web.
- Se desarrollará el sistema software web con todas las disposiciones en los campos de Laboratorio clínico, bacteriología y microbiología.
- Se desplegará el sistema web en un entorno de producción.
- Se validará la implementación del sistema web, optimizando la gestión y control de entrega de los resultados y reserva de turnos de los exámenes médicos.

### **Hipótesis**

Si se desarrolla un sistema web utilizando DEVOPS junto a la herramienta JENKINS, entonces se mejora la gestión de reservas y consultas en línea de los servicios que ofrece SL Laboratorio Clínico y Bacteriológico.

### **Variables de la Investigación**

#### ***Variable Dependiente***

Optimiza la gestión de las reservas y consulta de resultados en línea de exámenes clínicos del laboratorio clínico y bacteriológico SL.

#### ***Variable Independiente***

Sistema web basado en la metodología DEVOPS.

**Conceptualización de la Variable Independiente.** DEVOPS es una metodología de desarrollo de software en la cual nos permite la automatización de varios procesos como son los procesos de desarrollo y procesos de despliegue, también genera mayor valor empresarial y capacidad de respuesta, mediante la prestación de servicios ágiles de alta calidad. Pero siempre respaldando la satisfacción del cliente.

**Indicadores**

- 90% de efectividad en la reserva de un examen clínico.
- 90% de eficiencia en la entrega de resultados a los usuarios finales.
- Por lo menos un examen clínico enviado a un paciente.

## Capítulo II

### Marco Teórico

#### Introducción

El presente capítulo es una recopilación de información tales como antecedentes históricos, conceptuales, referenciales y contextuales referentes a las reservas y consultas de resultados en línea de exámenes médicos.

Además, se presentan las diferentes herramientas que se utilizaran para el desarrollo del sistema web aplicando la metodología DevOps y JENKINS. Esta investigación previa permite garantizar la factibilidad del desarrollo del proyecto.

#### Historia de la Web

Durante los últimos años, el avance en las tecnologías ha permitido un sin número de mejoras en el campo de la medicina. En lo referente a la comunicación, la creación del internet supuso un enorme cambio en la búsqueda de información, ya que esta herramienta ha facilitado el acceso a la información independientemente al lugar de origen. Fue a partir de 1966 que aparecieron los sistemas Web (WSE) este comenzó a incrementarse sustancialmente hasta la actualidad.

Los aplicativos webs son una agrupación de archivos interconectados por vínculos de hipertextos, disponibles en la red, y comunicados a través de tecnologías digitales. La Web es un “Organismo vivo” que evoluciona con el tiempo. Desde su creación en 1966, con la aparición de la primera red, hasta el surgimiento del internet que conocemos actualmente, no ha parado de transformarse y pulirse, desde la Web 1.0 hasta ahora llegar a tener la Web 4.0 (Ariño, 2018).

La evolución de los sistemas Web (WSE) comenzó con la Web 1.0 apareció en el 1990, esta era una herramienta utilizada para la cognición, es decir, mediante el conocimiento humano (Fuchs, y otros, 2010). Este presentaba información que se lograba

acceder, pero no podía interactuar con ella, solo visualizarla, en si era una comunicación unidireccional.

Esta web primitiva es centralizada, estática y secuencial, solo era de lectura y no era interactiva (García Aretio , 2007). En esta solo podía visualizar la información, y no se actualizaba. Los usuarios eran pasivo que recibían la información o la podían publicar, sin que la posibilidad de tener alguna retroalimentación o interacción con el contenido de la web. Sirve para usar en navegadores, motores de búsqueda, etc (Ariño, 2018).

Con el pasar del tiempo se podía interactuar con la web y mediante estas interacciones se actualizaban con los aportes que le dejaban los usuarios, esta web era la 2.0 (Aghael , Nematbakhsh, & Khosravi Farsani, 2012) .La Web 2.0 fue creada por O'Reilly en 2004 para referirse a la segunda generación de tecnologías Web basadas en comunidades de usuarios y servicios especiales, como redes sociales, blogs, Wikis, chats, foros, fotografía de álbumes y presentaciones web, y muchos más.

Esta nueva generación de web facilitaba el intercambio de información de una forma ágil entre los usuarios (Labra Gayo, 2006). Es interactiva y dinámica, promoviendo el desarrollo de la inteligencia colectiva y favoreciendo el trabajo en equipo o colaborativo. Por esta razón, la web 2.0 es también llamada la web social por el enfoque interactivo y colaborativo social de esta herramienta, que posibilita la conexión de personas con personas mediante redes sociales, wikis, etc., es decir, esta web actúa como punto de encuentro de los usuarios, pues es una comunicación bidireccional.

Con el inicio de la Web 2.0 llegó un fenómeno social que cambio para siempre nuestra interacción con la información y la comunicación, sobre todo porque nos permitió ser parte de ella. Actualmente la web 2.0 está implementada en redes sociales como: Facebook, Twitter, Instagram, foros, blog, páginas de alojamiento de archivos multimedia, etc (Corino López, 2017).

Con el paso de tiempo en un artículo de Zeldman en el año 2006 surgió la web 3.0. Esta nueva web se activó para el público en el año 2010, para ese tiempo esta web era un

paso muy importante para la tecnología, puesto que reflejaban consecuencias muy relevantes para los usuarios de la red (Hernández Asunción, 2013). La Web 3.0 trata de conectar aplicaciones web entre sí, con el objetivo de mejorar la experiencia de los usuarios; con este cambio se agregaron funcionalidades nuevas como: la construcción de la web semántica, la aparición de la web geoespacial y la autonomía respecto del navegador.

La web semántica o más conocida como web 3.0 aprovecha de forma más eficiente los datos: “data web”, puesto que el usuario tiene el control absoluto para modificar y realizar los cambios directamente en la web o base de datos, puesto que esta web nos incluye metadatos ontológicos y semánticos que mediante el uso de un sistema de procesamiento pueden ser rastreados.

La web 3.0 está muy asociada al concepto de personalización porque nos brinda un flujo de contenido e información adaptados al gusto del usuario. Esta web hace que todos disfrutemos del contenido, información y herramientas sin importa el dispositivo que usemos para conéctanos a internet debido que busca una versatilidad y flexibilidad que superen las barreras del formato y la estructura (CORRALES ÁLVAREZ, 2015). En fin, la web 3.0 evoluciono mucho con el transcurso de tiempo, ofreciendo nuevas funcionalidades de personalización y búsqueda que nos ayuda a acceder más rápido a la información que necesitamos, con esto se empezó a construir la nueva versión de la web. (Latorre , 2018).

En el año 2016 se inició con la web 4.0 con esta tenemos un comportamiento más inteligente, más predictivo, de manera que con solo una afirmación o petición ponemos en marcha un conjunto de acciones que nos dan como resultado en poco tiempo aquello que buscamos o pedimos (Rosique, 2016) . Con la evolución de la tecnología, podemos ver que empresas como Microsoft, Amazon, Google, Facebook, entre otras, están desarrollando nuevos sistemas utilizando Deep Learning y Machine Learning, estas herramientas son capaces de procesar información como lo haría un cerebro humano.

## **Laboratorio clínico y bacteriológico**

Los laboratorios clínicos proporcionan información de utilidad clínica a los médicos. Esta información es de gran valor, tanto para la toma de decisiones diagnósticas y/o terapéuticas como para la evaluación del estado de salud de la población (Aguilera Gámiz, y otros, 2004).

Las dos últimas décadas han puesto un cambio drástico en los laboratorios clínicos, debido a una combinación de circunstancias:

- Los avances significativos en la tecnología han permitido realizar una gran cantidad de pruebas sin comprometer la calidad de los resultados.
- La llegada de la informática, ha permitido procesar grandes cantidades de información de forma rápida y segura.
- Desarrollo de varias pruebas diagnósticas nuevas y más eficaces e importante en todas las áreas del conocimiento de laboratorio.
- El hecho de que los resultados de los exámenes se puedan saber sin necesidad de trasladarse presencialmente al laboratorio clínico.

Todo lo anterior ha conducido a que los laboratorios clínicos, que suelen centralizarse en los hospitales, pero dan servicio a todos los niveles del SSPA, ocupen, cada vez más, un puesto clave en el proceso asistencial. Además, se ha ido estableciendo paulatinamente un amplio sistema de transporte de muestras desde los domicilios de los pacientes a los diferentes puntos de extracción y, desde estos últimos, a los laboratorios (Aguilera Gámiz, y otros, 2004).

La actividad asistencial de los laboratorios se agrupa en diferentes áreas de conocimiento, donde desarrollan su labor las distintas especialidades de las ciencias del Laboratorio Clínico: Análisis Clínicos, Bioquímica Clínica, Hematología, Microbiología, Inmunología y Anatomía patológica. Cada una dispone de una gran variedad de pruebas diagnósticas, por lo que el catálogo de servicios de pruebas clínicas incluye en la actualidad miles de pruebas, frente a los cientos disponibles hace apenas unas décadas.

### Características para una prueba diagnóstica

Para que un procedimiento diagnóstico sea verdaderamente útil debe cumplir una serie de requisitos, incluyendo que esté bien indicada, tomada, transportada, realizada, reportada e interpretada (AM, 2015).

Del mismo modo, para que un estudio de laboratorio sea adecuado debe cumplir con criterios de aplicabilidad y de confiabilidad:

**Tabla 1**

*Criterios de aplicabilidad y de confiabilidad*

<b>Aplicabilidad</b>	<b>Confiabilidad</b>
Oportunidad	Precisión
Costo	Exactitud
Seguridad	Sensibilidad
Complejidad	Especificidad

*Nota.* Criterios para una prueba de diagnóstico. Tomado de (AM, 2015).

El criterio de facilidad de uso se basa en la rapidez con la que el médico recibe los resultados, lo que determina en gran medida que la prueba pueda realizarse con urgencia. En cambio, si la prueba es muy compleja o costosa por su rareza, puede ser necesario enviarla a un laboratorio especializado (AM, 2015).

**Figura 1**

*Ciclo de una prueba de laboratorio*



*Nota.* Ciclo típico de una prueba de laboratorio. Tomado de (Campuzano Maya, 2011).

En la figura 1, el círculo exterior representa la relación médico-paciente-laboratorio-médico, el círculo central representa las diferentes etapas de cada prueba de laboratorio: pre-preanálisis, preanálisis, análisis, postanálisis y postanálisis, mientras que el círculo central representa los pasos que llevan cada prueba de laboratorio (Campuzano Maya, 2011).

## DevOps

DevOps fue inventado por Patrick Debois en 2009, que desde entonces se ha convertido en uno de los principales expertos en este campo. El término combina las palabras "desarrollo", "operaciones" y puede utilizarse como punto de partida para entender qué significa DevOps (Jiménez Marco, 2016).



DevOps es una estructura, un ciclo de vida y una metodología de gestión para un conjunto de procesos que integran eficazmente el desarrollo, la entrega y las operaciones, ayudando así a conectar eficazmente los estilos tradicionalmente separados de desarrolladores y operadores. (Ebert, Gallardo, Hernantes, & Serrano, 2016).

El objetivo principal de DevOps es unificar el desarrollo y la operación del software en un proceso único, integrado y continuo. Desde un enfoque DevOps, se espera aprovechar la experiencia y conocimiento de las personas, los procesos y la tecnología para estimular la colaboración y la innovación en todo el proceso de desarrollo y lanzamiento del software, esto de manera rápida, frecuente y confiable, sin sacrificar calidad y el valor entregado a los clientes, características propias del ADN en los enfoques ágiles (Guerrero, Certuche Díaz, Zúñiga, & Pardo Calvache, 2019).

Los principales motivos estratégicos por los que una organización implementa el modelo DevOps son los siguientes (Élez Villamarín, 2019) :

- **Reducción de costes:** al implementar DevOps se mejora los procesos del negocio y se reduce, por tanto, sus costes.
- **Mejora de la calidad:** al generar y probar continuamente versiones, el producto final es de mayor calidad y tiene más valor para el cliente.
- **Mejora de la productividad:** es posible aumentar la productividad y reducir los plazos de entrega final acortando los ciclos de desarrollo e introduciendo las pruebas adecuadas en el entorno de producción.
- **Diferenciación:** la implementación de DevOps reduce la ventaja competitiva sobre los competidores.
- **Innovación:** DevOps permite la innovación cambiando fundamentalmente los procesos empresariales, reduciendo los costes, mejorando la calidad, la eficiencia y el servicio al cliente (menor time to market).

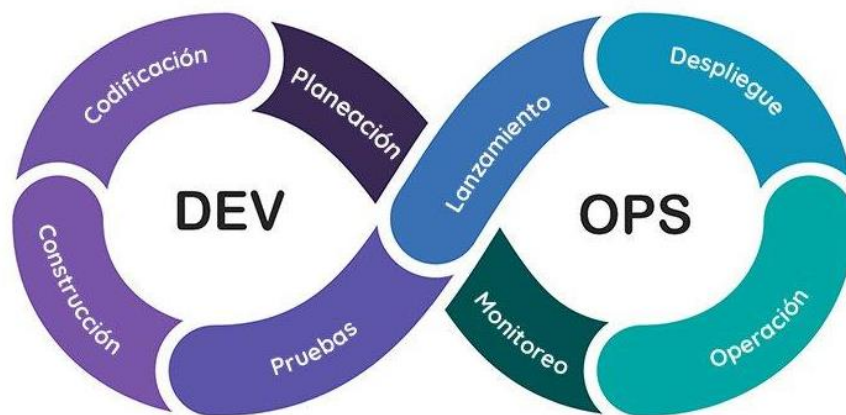
## Ciclo de vida DevOps

DevOps en la actualidad es una metodología muy conocida y aplicada al proceso de la creación de sistemas y aplicaciones tecnológicas tanto para su desarrollo como para su gestión empresarial. Por esa razón, se debe seguir de forma ordenada su ciclo de vida y garantizar su correcto funcionamiento, infraestructura y operaciones, para así proporcionar una entrega continua con alta calidad de software.

El ciclo de vida de DevOps es iterativo, ya que se basa en el ciclo de vida de la metodología ágil. Las fases de los componentes forman un bucle interminable con dos conjuntos de tareas en dos mundos: desarrollo (lado izquierdo del bucle) y operaciones (lado derecho del bucle) (Élez Villamarín, 2019).

### Figura 2

*Fases del ciclo de vida DevOps*



*Nota.* Ciclo de vida DevOps. Tomado de (Sesitdigital, 2020) .

El ciclo de vida de DevOps (a veces denominado flujo de trabajo de entrega continua) es una serie de procesos iterativos automatizados, tareas o fases de desarrollo, como se muestra en la figura 2. Estas fases se ejecutan dentro de un ciclo de vida de desarrollo más amplio, automatizado e iterativo que está optimizado para entregar

rápidamente software de alta calidad. El ciclo de vida de DevOps consta de las siguientes fases:

### ***Planeación***

La primera fase del ciclo de vida DevOps se basa en habilitar los mecanismos necesarios para dar la posibilidad a cualquier usuario de aportar de forma continua sus ideas y que estas puedan ser transformadas en requerimientos u objetivos, siendo estos priorizados e incluidos en próximas iteraciones en forma de historias de usuarios (Iriz Ricote, 2019).

El objetivo de la fase de planificación es maximizar el valor empresarial del producto mediante el desarrollo de una lista de características que, cuando se incluyan en el producto a entregar, producirán el resultado de valor deseado (Crawford, 2019). Es importante centrar parte de la planificación en la respuesta a los cambios a corto plazo para que los equipos de desarrollo y operaciones puedan regular los nuevos cambios de manera ágil (Pérez Hoyos, 2018). Esta fase incluye actividades como:

- Levantamiento de los requerimientos
- Métricas o reglas del negocio
- Actualización y elaboración de actas de métricas de la aplicación
- Calendarización y plan de construcción
- Políticas y requisitos de seguridad
- Construcción y arquitectura de la aplicación.

### ***Codificación***

En la segunda fase de la metodología DevOps se lleva al código todas las funcionalidades antes especificadas en las reglas del negocio (Pérez Hoyos, 2018). En esta fase se comienza a desarrollar el código del software, se avanza en pequeños procesos en el ciclo de desarrollo.

Al mismo tiempo, se empiezan a definir las pruebas que deben realizarse sobre los fragmentos de código para garantizar que se ajustan a las especificaciones funcionales. El director del proyecto se asegura de que el equipo de desarrollo esté familiarizado con las herramientas de seguridad y los ataques a los que puede enfrentarse este nuevo software (Tarisfeño, 2020). Esta fase incluye actividades como:

- Establecer parámetros de seguridad y la automatización de procesos
- Construcción de repositorios de Web Services
- Utilizar un software para la gestión de tareas propias del desarrollo y su seguimiento.

### ***Construcción***

La fase de construcción se refiere a las tareas que incluyen todos los parámetros relacionados con el despliegue de la aplicación (Pérez Hoyos, 2018), los cuales deben estar listos para que no haya problemas cuando se suba la aplicación al entorno de preproducción, por lo que esta etapa también se conoce como preproducción, esta fase incluye tareas como (Landázuri Ortiz, 2019) :

- Preparación del hardware para la aplicación
- Creación de usuarios de aplicación y de base de datos
- Creación de empaquetados y disparadores
- Liberación y puesta en escena en preproducción

### ***Pruebas***

En esta fase se realizan pruebas para garantizar que todas las funciones del proyecto funcionan correctamente. Además de las nuevas pruebas, hay que tener en cuenta las pruebas de conformidad con las especificaciones funcionales existentes. El equipo de control de calidad utiliza herramientas para identificar errores en el nuevo código y los

corrige de forma continua. En el caso improbable de que una prueba no funcione, pueden volver a la fase anterior y arreglarla (Construcción) (Tarisfeño, 2020).

Todos los casos de prueba son necesarios para lograr un producto exitoso en cada compilación (Landázuri Ortiz, 2019). Las principales activadas en esta fase son:

- Pruebas de sistema
- Pruebas de aceptación
- Pruebas de regresión
- Análisis de seguridad y vulnerabilidad

### ***Lanzamiento***

Una vez que la aplicación ha superado todas las pruebas funcionales y de integración, se puede crear una versión de lanzamiento del software. El objetivo es demostrar que ha superado todas las pruebas anteriores. En esta fase, el software se integra con el código existente y se prueba. Hay que tener en cuenta que el desarrollo continuo sólo es posible con la integración y las pruebas continuas. (Tarisfeño, 2020). Las principales activadas en esta fase son:

- Realizar gestión de cambios
- Aprobaciones de versiones
- Automatización de versiones
- Entre otras tareas referentes al versionar el software

### ***Despliegue***

En esta fase se deben establecer los parámetros de control previos a la subida de las aplicaciones en los diferentes entornos (prueba, preproducción y producción), pudiendo ser desplegado de forma automática, pero siempre DevOps promueve la sistematización de los procesos (Landázuri Ortiz, 2019). Las principales activadas en esto son:

- Automatizar el proceso de despliegue
- Automatizar los archivos de configuración
- Integrar de forma segura al sistema

### ***Operación***

Esta fase se centra en la configuración de hardware y parametrización de la aplicación que se ubicará en el sitio de producción, asegurando que los usuarios puedan operar la aplicación y garantizando la escalabilidad, sostenibilidad, disponibilidad, redundancia y seguridad de la misma (Landázuri Ortiz, 2019).

### ***Monitoreo***

En esta fase se definen los parámetros de monitoreo para controlar que la aplicación se encuentra estable tanto en su funcionamiento como en su infraestructura, previo al paso a mantenimiento. Aquí se evalúa el sistema, consultando el histórico de las mediciones durante un periodo de tiempo establecido. Es decir, se analiza toda aquella información que impacte sobre la actividad del plan que incluye cambios, actualizaciones, etc (Landázuri Ortiz, 2019).

También establece los parámetros del sistema que se van a controlar. Se recopila toda la información recogida durante un periodo de tiempo para realizar los ajustes necesarios en la siguiente fase de planificación, también es en esta fase cuando el equipo de operaciones define las medidas de seguimiento y control del estado del sistema y su infraestructura (Tarisfeño, 2020).

Cada una de estas fases que corresponden al ciclo de vida del DevOps forman la metodología, de esta forma se inicia un flujo de tareas continuas, que comienzan en la fase de análisis y finaliza en la aprobación del sistema, repitiendo de nuevo todo el proceso.

## **Modelo C4**

El modelo C4 fue creado por Simón Brown, quien comenzó a enseñar a la gente acerca de la arquitectura de software, mientras trabajaba como desarrollador/arquitecto de software en Londres. Este modelo surge como una gran solución para apaciguar la brecha entre modelo y código, ayudando a comunicar la arquitectura de un sistema en diagrama de contexto, contenedores, componentes y código, proporcionando así diferentes niveles de abstracción, cada uno entendibles para diferente tipo de audiencia.

Está dirigido principalmente a arquitectos y desarrolladores de software, el modelo C4 proporciona una manera para que los equipos de desarrollo de software comuniquen su arquitectura de software de manera eficiente y efectiva, en diferentes niveles de detalle, contando diferentes historias a diferentes tipos de audiencia, al hacer un diseño inicial o retrospectivamente (Brown, 2018).

Está conformado por cuatro niveles que representan al sistema con diferentes niveles de abstracción:

### ***Nivel 1: Diagrama de contexto***

El nivel 1, un diagrama de contexto del sistema como muestra la figura 3, este proporciona un punto de partida, muestra el sistema de software que está construyendo y cómo encaja en el mundo en términos de las personas que lo utilizan y los otros sistemas de software con los que interactúa (Vivanco, 2019).

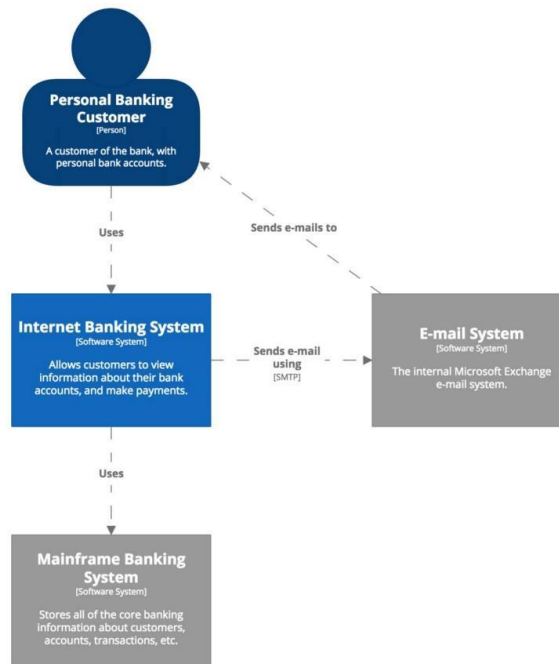
Nos permite conocer a los usuarios finales que usaran las funcionalidades del sistema puesto que nos proporciona una visión general de sistema y su interacción con el mundo exterior (García Peñalvo, García Holgado, & Vázquez Ingelmo, 2020 - 2021).

Los detalles no son importantes aquí, pues esta es una vista alejada que muestra una imagen grande del panorama del sistema, es el tipo de diagrama que podría mostrar a al equipo de desarrollo como al usuario final. Su enfoque está en las personas (actores, roles, personajes, etc.) y los sistemas de software (dependencias externas) que están

directamente conectados al sistema de software en el alcance, en lugar de las tecnologías, los protocolos y otros detalles de bajo nivel (Brown, 2018).

### Figura 3

#### Diagrama de contexto o Nivel 1



**System Context diagram for Internet Banking System**

The system context diagram for the Internet Banking System.  
Last modified: Wednesday 02 May 2018 13:46 UTC

*Nota.* Ejemplo de un diagrama de contexto. Tomado de (Vivanco, 2019)

#### Nivel 2: Diagrama de Contenedores

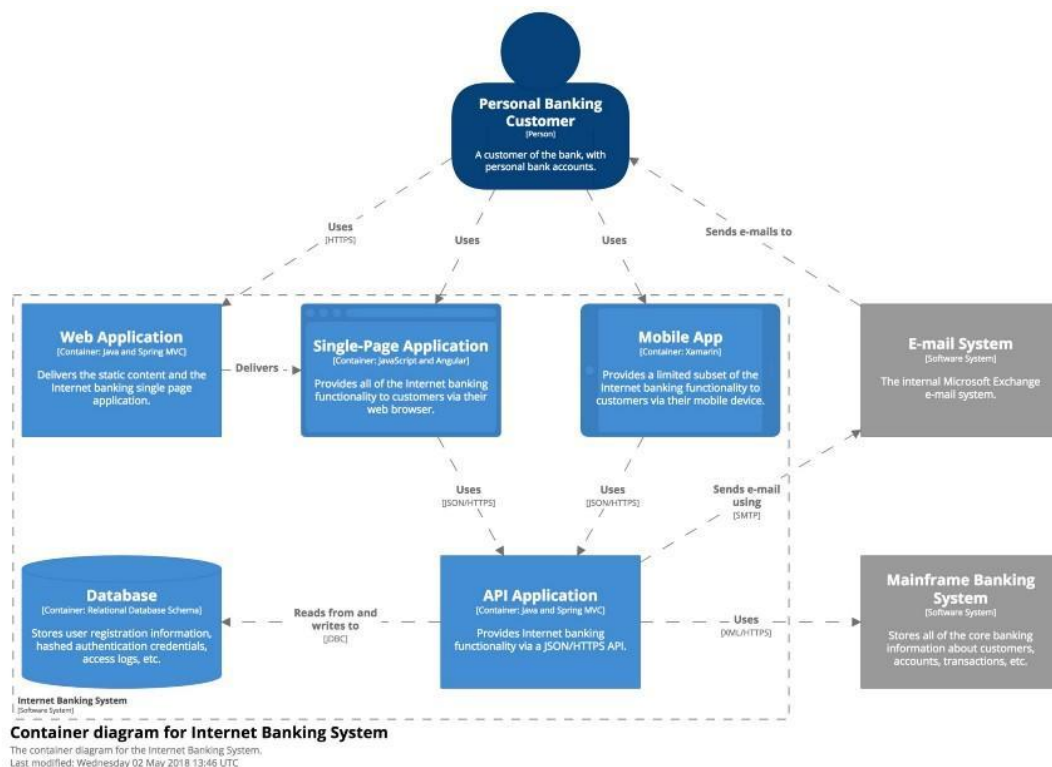
El nivel 2 o diagrama de contenedor como se muestra la figura 4, indica el alcance del sistema de software y representando los componentes de construcción en un alto nivel, como son los contenedores que son forman parte del sistema (Vivanco, 2019). Es el tipo de diagrama que podría mostrar a personas técnicas dentro y fuera del equipo de desarrollo de software; incluyendo arquitectos de software, desarrolladores y personal de operaciones/sopORTE.



Un contenedor puede considerarse como una unidad ejecutable o desplegable y hace referencia a toda entidad que ejecuta código o almacena datos. (García Peñalvo, García Holgado, & Vázquez Ingelmo, 2020 - 2021).

#### Figura 4

##### Diagrama de contenedores o Nivel 2



*Nota.* Ejemplo de un diagrama de contenedores. Tomado de (Vivanco, 2019)

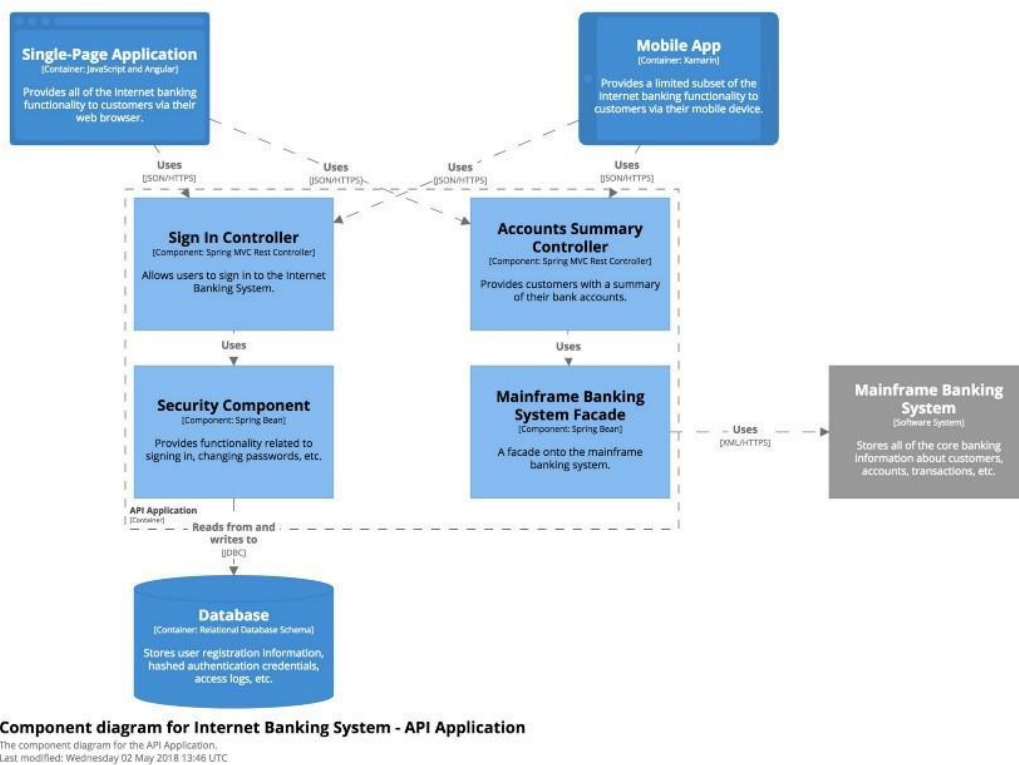
#### Nivel 3: Diagrama de componentes

El nivel 3 o un diagrama de componentes como se muestra la figura 5, amplía un contenedor individual exponiendo cada uno de los componentes que este tiene, estos deben colocarse en función al código mediante abstracciones reales. Es el tipo de diagrama que podría mostrar a personas técnicas como arquitectos o desarrolladores de software.

En cada contenedor podemos localizar muchos componentes, estos representan un conjunto de funcionalidades que pueden estar relacionados entre sí y entre los usuarios finales. Con este diagrama podemos mostrar los detalles de implementación, así como, la responsabilidad de cada componente (García Peñalvo, García Holgado, & Vázquez Ingelmo, 2020 - 2021).

**Figura 5**

*Diagrama de componentes o Nivel 3*



*Nota.* Ejemplo de un diagrama de componentes. Tomado de (Vivanco, 2019)

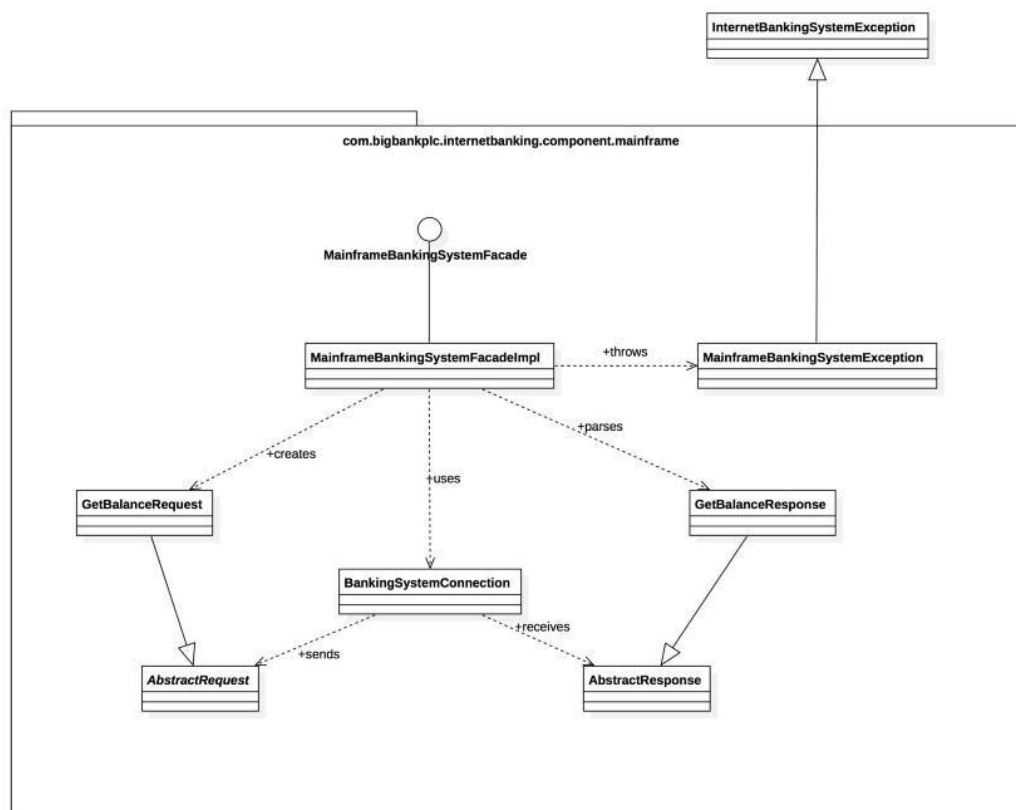
#### **Nivel 4: El código**

En este nivel se exponen detalles de la implementación de cada componente, se puede usar un diagrama de código para ampliar un componente individual y mostrar cómo se implementa ese componente. Este tipo de diagrama podría ser mostrado a Arquitectos y desarrolladores de software. Idealmente, este diagrama se generaría automáticamente

usando herramientas de modelado como IDE o UML y debe exponer aquellos atributos y método que permitan contar una historia. Este nivel se recomienda realizar solo para los componentes más complejos o importantes (Brown, 2018).

**Figura 6**

*El código o nivel 4*



*Nota.* Ejemplo de un diagrama de clases UML – nivel 4. Tomado de (Vivanco, 2019)

## Jira

La clave de un buen proyecto está en la planificación. Es por eso que con Jira Software puedes crear historias de usuario, planificar Sprints y distribuir las tareas entre tu equipo de software.

Jira es una herramienta de gestión de proyectos desarrollada por Atlassian, sus inicios fueron en el año 2002 teniendo como objetivo hacer seguimiento a los

inconvenientes que fueran surgiendo en el desarrollo de un software (Muradas, 2020). Este forma parte de una gama de productos diseñados para ayudar a equipos de todo tipo a gestionar el trabajo. En principio, Jira se diseñó como un gestor de incidencias y errores. Sin embargo, se ha convertido en una potente herramienta de gestión de trabajo para todo tipo de casos de uso, desde la gestión de requisitos y casos de prueba hasta el desarrollo de software ágil (ATLASSIAN, 2020).

## **IntelliJ IDE**

Es un IDE para desarrollar software. Está desarrollado por JetBrains y está disponible como una edición comunitaria con licencia de Apache 2 y en una edición comercial pagada. Ambos se pueden utilizar para el desarrollo comercial. Soporta lenguajes basados en la máquina virtual de Java (JVM), como Groovy, Scala, Kotlin y Android. Cada aspecto de IntelliJ IDEA se ha diseñado para maximizar la productividad del desarrollador. En conjunto, la asistencia de codificación inteligente y el diseño ergonómico hacen que el desarrollo no solo sea productivo sino también agradable (Jetbrains, 2021).

Además, IntelliJ IDEA, con sus potentes herramientas integradas, su compatibilidad con JavaScript y las tecnologías relacionadas, y su soporte avanzado para marcos de trabajo populares como Spring, Spring Boot, Jakarta EE, Micronaut, Quarkus y Helidon puede ayudar a desarrollar aplicaciones web full-stack. También, puede ampliar IntelliJ IDEA con complementos gratuitos desarrollados por JetBrains, para trabajar con otros lenguajes de programación como Go, Python, SQL, Ruby y PHP (Jetbrains, 2021).

IntelliJ IDEA completa el código de manera automática. Además, mientras vas escribiendo este va mostrándote código que puedan coincidir con lo que estás tratando de hacer, si IntelliJ IDEA detecta algún código mal escrito lo resalta y te dirá que está mal escrito y si presionas ALT + ENTER te ofrecerá opciones para solucionar el problema, es muy fácil corregir el código a medida que se va escribiendo (PI, 2020).

## Spring Boot

Es un framework Java basado en el modelo Controlador-Vista, con componentes y bibliotecas que facilitan el desarrollo y la implementación de servicios REST, con énfasis en la eliminación de la necesidad de configurar las aplicaciones a través de archivos XML (Toledano López, López del Castillo, & Vasquez Sánchez , 2018).

Spring Boot, lanzado en 2012, es una solución para el marco de trabajo de Spring Java que sigue el principio de "convención por encima de la configuración" y reduce la complejidad del desarrollo de nuevos proyectos basados en Spring. Para ello, Spring Boot adapta el camino para el desarrollo temprano proporcionando una estructura de proyecto básica, que incluye orientación sobre cómo utilizar el marco y todas las bibliotecas de terceros necesarias para la aplicación (Digital Guide IONOS, 2019).

Como las principales características de Spring Boot tenemos que nos permite crear aplicaciones de Spring independientes, al poseer servidores de aplicación embebidos como Tomcat, Jetty o Undertow, no es necesario generar un WAR para su ejecución, también permite configurar automáticamente bibliotecas de Spring y de terceros, no es necesario realizar una configuración en archivos XML, por lo que es más fácil y rápido la integración con otros proyectos de Spring (Haro, Guarda, Ninahualpa Quiña, & Zambrano Peñaherrera, 2018).

Spring Boot centra su éxito en las siguientes particularidades que lo hacen extremadamente fácil de utilizar (Blancarte, 2018):

- **Configuración:** Spring Boot cuenta con sofisticados módulos que configuran automáticamente todos los aspectos de la aplicación y poder ejecutarse sin necesidad de definir nada.
- **Resolución de dependencias:** Con Spring Boot, sólo tienes que decidir qué tipo de proyecto quieres y Spring Boot se encarga de todas las bibliotecas/dependencias y hace que la aplicación funcione.

- **Despliegue:** Spring Boot puede ejecutarse como una aplicación independiente, pero también es posible ejecutar aplicaciones web, ya que es posible desplegar las aplicaciones a través de un servidor web integrado, como Tomcat, Jetty o Undertow.
- **Métricas:** Por defecto, Spring Boot dispone de servicios que permiten consultar el estado de salud de la aplicación, permitiéndote saber si la aplicación está encendida o apagada, la memoria utilizada y disponible, el número y detalles de los beans creados por la aplicación, los controles de activación y desactivación, etc.
- **Extensible:** Spring Boot permite la creación de complementos que ayudan a la comunidad del software libre a crear nuevos módulos que facilitan aún más el desarrollo.

## Servicios Rest

Los servicios web basados en la arquitectura REST (Representational State Transfer) son una alternativa más sencilla a SOAP y WSDL, donde los datos se transfieren a través del protocolo estandarizado HTTP. Varias empresas como Google, Facebook y Yahoo! han convertido con éxito sus servicios a esta tecnología. REST fue publicado en 2000 por Roy Fielding, de la Universidad de California, en una conferencia en la que presentó los principios de la arquitectura de software para utilizar Internet como plataforma de procesamiento distribuido (De Seta, 2008).

El cliente tiene acceso a los recursos utilizando una única URI y una representación del recurso es devuelto con un estado de la transferencia de la información (Pai & Asnika, 2014). REST no está estrechamente relacionado con el protocolo HTTP, pero es con el que más comúnmente se asocia. Los servicios web basados en REST cumplen los siguientes principios (Haro, Guarda, Ninahualpa Quiña, & Zambrano Peñaherrera, 2018):

- **Utiliza métodos de HTTP de manera explícita:** Cada método tiene una función específica: GET se utiliza para obtener un recurso del servidor, POST se utiliza para crear un nuevo recurso en el servidor, PUT actualiza o cambia el estado de un recurso, DELETE elimina un recurso.
- **REST no mantiene estado:** Es necesario escalar para cubrir la demanda constante y creciente mediante balanceadores, clusters y servidores de alta disponibilidad para distribuir las peticiones entre los equipos y reducir los tiempos de respuesta.
- **URI en formato de directorios:** La estructura de los URI se crea en forma de lista, lo que es fácil de entender y permite un uso intuitivo.
- **Representaciones:** Los servicios web REST proporcionan una respuesta en formato JSON o XML, que puede ser utilizada por cualquier aplicación desarrollada en cualquier lenguaje.

## Spring Security

Spring Security es un marco que proporciona autenticación, autorización y protección contra ataques comunes. Con soporte de primera clase para asegurar aplicaciones imperativas y reactivas, es el estándar de facto para asegurar aplicaciones basadas en Spring. Al igual que todos los proyectos de Spring, el verdadero poder de Spring Security se encuentra en la facilidad con la que se puede ampliar para cumplir con los requisitos personalizados (Spring, 2021).

Spring Security proporciona servicios de seguridad para aplicaciones de software empresariales basados en J2EE, enfocado particularmente sobre proyectos construidos empleando SpringFramework. En el nivel de "autenticación" Spring Security soporta muchos modelos de autenticación, muchos de estos modelos de autenticación son proporcionados por terceros o son desarrollados por estándares importantes como el IETF(Internet

Engineering tTask Force), adicionalmente, Spring Security proporciona su propio conjunto de características de autenticación (Cacique, 2010).

## JWT

JSON Web Token o JWT, como se le llama más comúnmente, es un estándar abierto de Internet (RFC 7519) para transmitir de forma segura información confiable entre las partes de una manera compacta. Los tokens contienen notificaciones que se codifican como un objeto JSON y se firman digitalmente mediante un secreto privado o un par de clave pública/clave privada. Son autónomos y verificables, ya que están firmados digitalmente. Los JWT pueden ser firmados y/o encriptados. Los tokens firmados verifican la integridad de las notificaciones contenidas en el token, mientras que los cifrados ocultan las notificaciones de otras partes (tutorialpoint, 2020).

Los JWT también se pueden usar para el intercambio de información, aunque se usan más comúnmente para la autorización, ya que ofrecen muchas ventajas sobre la administración de sesiones utilizando tokens aleatorios en memoria. Si un usuario intenta acceder a la API protegida, solo permitirá el acceso si una solicitud tiene un JWT válido. La validación se realizará en el filtro registrado en la cadena de filtros Spring Security (Gordadze, 2021).

JSON Web Token tienen varias ventajas sobre los métodos tradicionales de autenticación y autorización basados en cookies y se utilizan en las siguientes situaciones (Digital Guide - IONOS, 2020):

- **Aplicaciones REST:** En las aplicaciones REST, JWT garantiza que no haya estática al enviar los datos de autenticación directamente con la solicitud.
- **Intercambio de recursos de origen cruzado:** JSON Web Token transmiten la información mediante el llamado cross-origin resource sharing, lo cual tiene importantes ventajas sobre las cookies, que normalmente no se transmiten de esta manera.



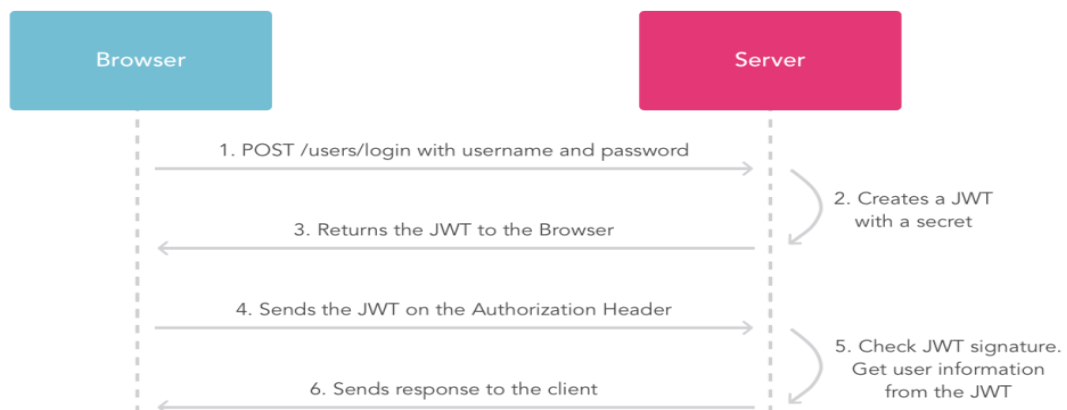
- **Uso de varios frameworks:** JSON Web Token está estandarizados y pueden utilizarse una y otra vez. Cuando se utilizan varios frameworks, es más fácil compartir los datos de autenticación.

### Ciclo de vida de un Token JWT

Ahora que sabemos que es JWT, veamos cómo funciona todo el proceso con un diagrama de uso:

#### Figura 7

*Proceso completo del JWT*



*Nota.* Diagrama explicando de proceso del JWT. Tomado de (Macias, 2018)

El proceso completo del JWT, como se muestra en la figura 7 consta de estos pasos:

1. El usuario de una aplicación web/móvil/desktop se conecta con sus credenciales en el servidor donde está publicada la API.
2. El usuario se valida en el servidor y se crea un nuevo Token JWT (usando el "secret-key") que se entrega al usuario.

3. El servidor revuelve el JWT firmado que contiene los datos/claims relativos al usuario y caducidad del Token.
4. El cliente/browser almacena el JWT para su uso y lo envía en cada solicitud a través "Authorization: Bearer".
5. El servidor verifica la firma del Token, su caducidad y comprueba si el usuario tiene permisos al recurso leyendo los datos del payload.
6. El servidor responde a la solicitud del cliente una vez que ha confirmado el Token y los permisos del usuario son correctos.

## **Bootstrap**

Bootstrap se lanzó en 2011. Originalmente fue desarrollado por Twitter, aunque luego se lanzó bajo una licencia MIT. En la actualidad sigue siendo desarrollado y creciendo este se encuentra en un repositorio de GitHub. Es un framework que está basado en librerías CSS y JQuery que nos permite desarrollar un sitio web receptivo. En las librerías CSS podemos encontrar una gran cantidad de elementos listos para ser utilizados como tipografías, menús, botones, tablas, entre otros (AcensTecnologies, 2018).

Usando este framework mejora formidablemente la UX del usuario, permitiendo que el sitio web se ajuste a cualquier dispositivo sin perder la navegabilidad de las pantallas de escritorio, bajando la tasa de rebote y mejorando el CTR desde el punto de vista SEO (am Desing, 2021). Reduciendo el tiempo de desarrollo del sistema web y permitiendo de forma efectiva aplicar una metodología ágil.

## **Angular**

Angular es un marco de front-end de código abierto diseñado para sitios web potentes y modernas. Es la única plataforma que garantiza el funcionamiento perfecto para MVC (Handa, 2021). Es un framework open source basado en JavaScript y sirve para crear aplicaciones del tipo SPA. Una SPA (Single Page Application) es una aplicación web que se

carga completamente en la página y modifica el contenido que expone según sea necesario sin necesidad de refrescar la página(campusMVP, 2021).

### ***Beneficios del desarrollo con Angular***

- Es un framework de código abierto.
- Tiene una estructura MVC.
- Ofrece elementos de diseño que ya están prediseñados.
- Se complementan fácilmente a plataformas Android e iOS.
- Proporciona la inyección de dependencias y el enlace de datos estos ayudan al programador a prevenir la escritura de código.
- Fácil de usar e instalar, proporcionando documentación amplia, comandos sencillos de ejecutar y herramientas de prueba innovadoras.

### **Java**

Java es un lenguaje de programación desarrollado por Sun Microsystems. Java se introdujo en la segunda mitad de 1995 y desde entonces se ha convertido en un lenguaje de programación muy popular. Java es un lenguaje muy popular porque los programas pueden ejecutarse en una gran variedad de sistemas operativos como Windows, Mac OS, Linux y Solaris. (Martinez Ladrón de Guevara, 2015). Se constituye como un lenguaje orientado a objetos, su intención es permitir que los desarrolladores de aplicaciones escriban el programa una sola vez y lo ejecuten en cualquier dispositivo.

Java es la tecnología utilizada para desarrollar aplicaciones que hacen la Web más interesante y útil. Java es la base de prácticamente todos los tipos de aplicaciones en red y el estándar mundial para el desarrollo y la distribución de aplicaciones móviles e integradas, juegos, contenidos web y software empresarial. Con más de 9 millones de desarrolladores en todo el mundo, Java permite el desarrollo, la implantación y el uso eficientes de aplicaciones y servicios interesantes (Java, 2021).

Java ha sido probado, perfeccionado, ampliado y comprobado por toda una comunidad de desarrolladores, arquitectos de aplicaciones y entusiastas de Java. Java está diseñado para permitir el desarrollo de aplicaciones portátiles y de alto rendimiento para la mayor variedad posible de plataformas informáticas. Al poner las aplicaciones a disposición de todo el mundo en entornos heterogéneos, las empresas pueden ofrecer más servicios y mejorar la productividad, la comunicación y la colaboración del usuario final, así como reducir drásticamente el coste de propiedad de las aplicaciones tanto para los usuarios como para las empresas (Java, 2021). Java se ha convertido en un activo inestimable para los desarrolladores, ya que les permite:

- Escribir software en una plataforma y ejecutar virtualmente en otra.
- Crear software que pueda ejecutarse en un navegador y acceder a los servicios web disponibles.
- Desarrollar aplicaciones de servidor para foros en línea, almacenes, encuestas, procesamiento de formularios HTML, etc.
- Combinar aplicaciones o servicios que utilizan el lenguaje Java para crear aplicaciones o servicios altamente personalizables
- Escribir aplicaciones potentes y eficaces para teléfonos móviles, procesadores remotos, microcontroladores, módulos inalámbricos, sensores, pasarelas, productos de consumo y prácticamente cualquier otro dispositivo electrónico.

## **Maven**

Maven es una herramienta de software para la gestión y construcción de proyectos, basado en el concepto de un modelo de objetos de proyecto (POM), Maven puede administrar la construcción, los informes y la documentación de un proyecto desde una pieza central de información (Maven Apache, 2022). Usando Maven podemos construir y administrar cualquier proyecto basado en Java.

El objetivo principal de Maven es permitir que un desarrollador comprenda el estado completo de un esfuerzo de desarrollo en el menor tiempo posible. Para lograr este objetivo, Maven se ocupa de varias áreas de preocupación (Maven, 2022):

- Facilitando el proceso de construcción
- Proporcionar un sistema de construcción uniforme
- Proporcionar información de proyectos de calidad.
- Fomentar mejores prácticas de desarrollo

Maven ofrece un marco de trabajo para construir las fases del ciclo de vida. El grupo de desarrollo puede automatizar la infraestructura para poder crear el proyecto en corto tiempo.

## **GitHub**

GitHub es una compañía sin fines de lucro que ofrece un servicio de hosting de repositorios almacenados en la nube. GitHub es una plataforma de desarrollo colaborativo de software para alojar proyectos que usa el sistema de control de versiones Git, uno de los SCV más empleados en la actualidad tanto de forma particular como profesionalmente (GitHub, GitHub: The world's leading software development platform, 2021).

GitHub emerge como una herramienta para el aprendizaje que te proporciona, por un lado, un repositorio para el código y herramientas muy útiles para el trabajo en equipo y por otro como una herramienta docente que facilita la monitorización de un proyecto de software (Lopez-Pellicer, Béjar, Latre, Noguerras-Iso, & Zarazaga-Soria, 2015). Entre las principales características que nos ofrece GitHub podemos destacar (AcensTechnologies, 2020):

- Una Wiki que opera con Git para el mantenimiento de las distintas versiones de las páginas.
- Sistema de seguimiento de problemas. Se trata de un sistema muy parecido al tradicional ticket, donde cualquier miembro del equipo o persona (si

nuestro repositorio es público) puede abrir una consulta o sugerencia que se quiera hacer.

- Herramienta de revisión de código. Permite añadir anotaciones en cualquier punto de un fichero.
- Visor de ramas que permite comparar los progresos realizados en las distintas ramas de nuestro repositorio.

GitHub te ofrece dos enormes beneficios, que lo hacen ideal tanto para trabajar solo cómo en equipo (GitHub, Tutoriales en PDF, 2021):

- **Desarrollo colaborativo:** Con GitHub pueden trabajar simultáneamente en el mismo proyecto grandes grupos de desarrollo o empresas sin miedo a interferir en el trabajo de los otros, haciendo de su desarrollo algo fácilmente escalable.
- **Control de versiones:** Puedes estar tranquilo a cada paso que das en distintas partes y archivos de tu desarrollo, porque GitHub mantiene un historial de capturas completas de tu trabajo. Incluso puedes ir creando distintos “branch” o ramas paralelas de versiones de tu código, cómo la principal o “master” y otra de desarrollo, donde puedes trabajar sin comprometer la integridad de la rama principal.

## Jenkins

Jenkins, el servidor de automatización de código abierto líder, proporciona cientos de complementos para apoyar la creación, implementación y automatización de cualquier proyecto (Jenkins, 2021). Proyecto escindido de Hudson en disputa con Oracle. Proporciona servicios de integración continua para el desarrollo de software; es un sistema basado en un servidor que se ejecuta en contenedores de servlets como Apache Tomcat; es compatible con AccuRev, CVS, Subversion, Git, Mercurial, Perforce, el Soporta herramientas SCM como AccuRev, CV, Subversion, Git, Mercurial, Perforce, Clearcase y RTC, y puede

ejecutar proyectos basados en Apache Ant y Apache Maven, así como scripts de shell arbitrarios y comandos por lotes de Windows. Publicado bajo la licencia MIT, Jenkins es software libre (Gautam , y otros, 2018).

Con Jenkins, las organizaciones pueden acelerar los procesos de desarrollo y entrega de software mediante la automatización. A través de cientos de plugins, se puede desplegar en diferentes etapas del ciclo de vida del desarrollo, incluyendo la compilación, la documentación, las pruebas y el despliegue (Sentrio, 2021).

Con Jenkins, se pueden automatizar muchas tareas para acelerar el tiempo de comercialización de los productos digitales y sus nuevos lanzamientos (Sentrio, 2021).

Concretamente, con esta herramienta podemos:

- Automatizar la compilación y pruebas de software.
- Notificar a los equipos sobre la detección de errores.
- Desplegar los cambios en el código que hayan sido validados.
- Seguimiento de la calidad del código y de la cobertura de las pruebas.
- Generar la documentación de un proyecto.

### ***Integración continua con Jenkins***

La integración continua o Continuous Integration (CI) es una práctica común en el desarrollo de software que implica un proceso mediante el cual se irá compilando y ejecutando de manera continua un proyecto o código, para detectar errores o fallas cada cierto tiempo, la incorporación periódica tiene como objetivo mejorar el código del proyecto.

Todos los cambios realizados por el desarrollador (corrección de errores, creación de nuevas funciones, etc.) se prueban compilando el código fuente y obteniendo un archivo ejecutable (llamado build). Si es validado, será integrado al código fuente y desplegado (Sentrio, 2021). Cuando se habla de integración se refiere a compilación y ejecución. Mientras que un proceso continuo, es un ciclo de implementación, ejecución, compilación y actualización de un proyecto (Partner, 2021).

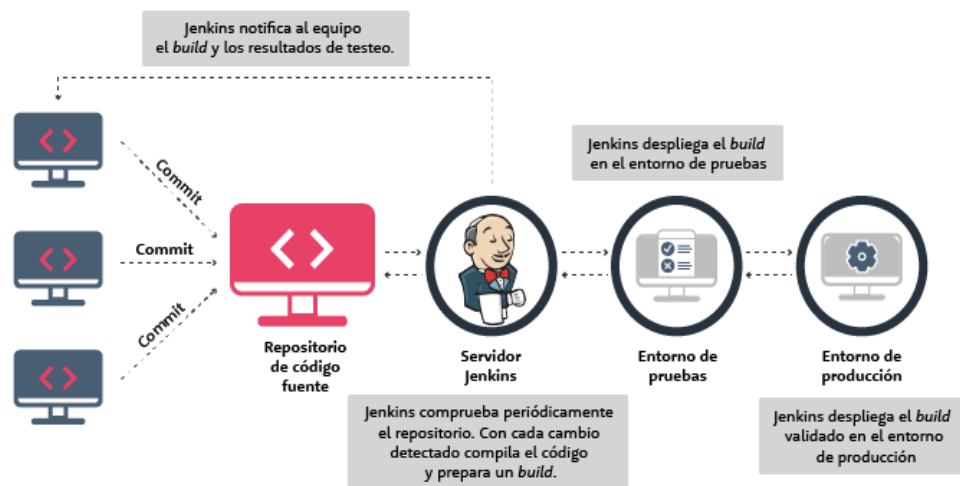
Además, la herramienta Jenkins le permite conocer el estado de su software en cualquier momento, supervisar la calidad del código, la cobertura de las pruebas y reducir la deuda técnica y los costes. La integración continua es una parte integral de DevOps y se basa en el objetivo de entregar valor a los usuarios lo más rápidamente posible. Suele estar vinculada a las metodologías de programación extrema (XP) y ágil (Sentrio, 2021).

### **Flujo de integración continua con Jenkins**

Jenkins es una herramienta esencial para automatizar la implementación de metodologías de integración y despliegue continuo. Utilizando esta herramienta debemos seguir el siguiente flujo de integración continua como se muestra la figura 8:

**Figura 8**

*Proceso de integración continua (CI) utilizando Jenkins*



*Nota.* Ejemplo de proceso de integración continua (CI). Tomado de (Sentrio, 2021)

- El desarrollador envía el código al repositorio de código fuente.



- El servidor de Jenkins comprueba regularmente el repositorio en busca de cambios.
- Inmediatamente después del commit, Jenkins detecta cualquier cambio que se haya producido en el código fuente. El código se compila y se prepara para la compilación. Si la compilación falla, se envía una notificación al equipo correspondiente. Si tiene éxito, se despliega en el servidor de pruebas.
- Después de la prueba, Jenkins genera un feedback y notifica al equipo el build y los resultados del testeo.
- Jenkins sigue comprobando el repositorio con frecuencia y todo el proceso se repite (Sentrío, 2021).

## JUnit

JUnit es un framework, creado por Erich Gamma y Kent Beck, utilizado para automatizar las pruebas unitarias en el lenguaje Java. JUnit es un marco simple para escribir pruebas repetibles para el lenguaje de programación Java. Es una instancia de la arquitectura xUnit para marcos de prueba de unidades (contributors, 2020). JUnit es un paquete de Java utilizado para automatizar los procesos de prueba. Al crear pruebas, JUnit ejecutará una prueba sobre el código especificado por el usuario (Gasteiz, 2014).

El concepto básico de estas herramientas son los casos de prueba (test-case), y los conjuntos de pruebas (test suite). Los casos de prueba son clases o módulos que disponen de métodos para probar los métodos de una clase o módulo concreta/o.

Existen varias razones para utilizar JUnit a la hora de hacer pruebas de código:

- La herramienta es gratuita y puede descargarse directamente del sitio web oficial.
- Es una herramienta muy utilizada, por lo que no será difícil buscar documentación.

- Existen varios plugins para poder usar con diferentes Entornos de Desarrollo Integrado (IDE).
- Con JUnit, ejecutar pruebas es tan fácil como compilar el código. El compilador "prueba" la sintaxis del código y las pruebas "validan" la integridad del código.
- Los resultados son comprobados por la propia aplicación y darán los resultados inmediatamente.
- Las pruebas de JUnit pueden organizarse en suites, que contendrán copias de las pruebas, e incluso pueden contener otras suites.
- Utilizando pruebas programadas en JUnit, la estabilidad de nuestras aplicaciones mejorará sustancialmente.
- Las pruebas realizadas se pueden presentar, junto con el código, para validar el trabajo realizado.

## **Docker**

Docker es una herramienta de código abierto, este permite crear, probar e implementar aplicaciones, sus uso y sintaxis es sencilla de aplicar. Docker se encarga de empaquetar software en contenedores, estos incluyen bibliotecas, herramientas de sistema, código, todo lo necesite para ejecutar el software. Con Docker, puede ajustar e implementar las aplicaciones rápidamente (AWS, 2022).

Las aplicaciones se empaquetan en forma de imágenes que normalmente contienen múltiples capas, cada capa está en un sistema de archivos completo con los paquetes de aplicaciones. Docker ayuda a los desarrolladores a dar vida a sus ideas, conquistando la complejidad del desarrollo de aplicaciones. Simplificamos y aceleramos los flujos de trabajo de desarrollo con un pipeline de desarrollo integrado y mediante la consolidación de los componentes de la aplicación. Utilizado activamente por millones de desarrolladores de

todo el mundo, Docker Desktop y Docker Hub proporcionan simplicidad, agilidad y elección sin igual (AWS, 2022).

Docker permite entregar código con mayor rapidez, transferir código con facilidad y ahorrar dinero al optimizar el uso de recursos. Con Docker, se obtiene un solo objeto, este puede ejecutarse de manera rápida y fiable. La amplia adopción significa que existe un gran ecosistema de herramientas y aplicaciones listas para su utilización que puede usarse con Docker (AWS, 2022). Existen varias razones para utilizar Docker las cuales son:

- **Envía más rápido:** la frecuencia de envío de software es siete veces más rápida que sus competidores.
- **Estandariza las operaciones:** facilita la implementación e identificación de problemas y puedes retornar a una fase anterior.
- **Transfiere más sencilla:** transfiere a la perfección desde equipos locales de desarrollo a implementaciones de producción en AWS.
- **Ahorra dinero:** mejorando su uso facilita la ejecución de código en cada servidor.

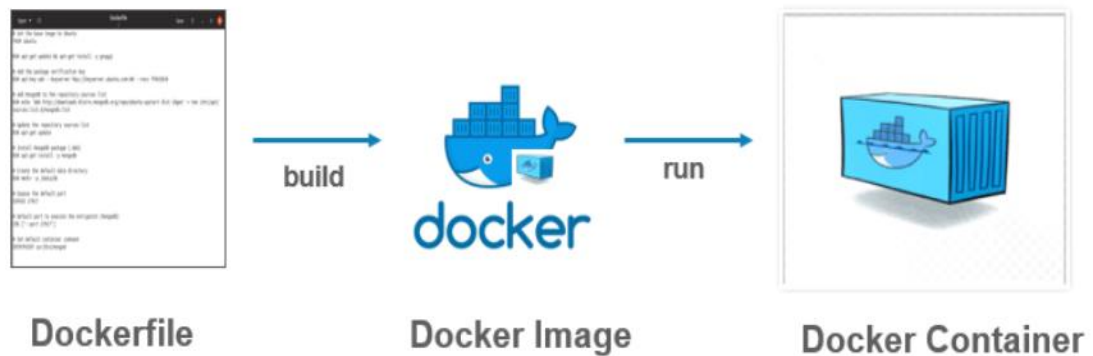
## DockerFile

Es un archivo de texto simple con un conjunto de comandos o instrucciones. Estos comandos / instrucciones se ejecutan sucesivamente para realizar acciones en la imagen base para crear una nueva imagen de la ventana acoplable. A continuación, se muestra en la figura 9 cómo se verá su flujo de trabajo (Avi, 2019):

- Cree un Dockerfile y mencione las instrucciones para crear su imagen de Docker
- Ejecute el comando `docker build` que creará una imagen de Docker
- Ahora que la imagen de la ventana acoplable está lista para usarse, use el comando de ejecución de la ventana acoplable para crear contenedores

**Figura 9**

*Flujo de trabajo de DockerFile*



*Nota.* Diagrama de flujo de trabajo de DockerFile. Tomado de (Avi, 2019)

## Docker Compose

Docker Compose es otro proyecto open source que permite definir aplicaciones multicontenedor de una manera sencilla y declarativa. Es una herramienta ideal para gestionar entornos de desarrollo, pero también para configurar procesos de integración continua (Martinez, 2019).

Docker Compose es una herramienta que facilita el uso de Docker y por tanto proporciona un conjunto de scripts que facilitan la construcción de nuestros servicios. La principal ventaja es la posibilidad de crear diferentes contenedores y diferentes servicios para cada contenedor. En pocas palabras, dockerfile define cómo generar imágenes de programa, mientras que docker compose permite encadenar y configurar estos contenedores para construir múltiples servicios (Rodríguez, 2019).

Además, Docker Compose nos permite trabajar con diferentes entornos como Production, Staging, Development y Testing, lo que nos permite crear un entorno de integración continua.

La configuración de Docker Compose se realiza en archivos con formato YAML con la extensión yml. Esto significa que todos los archivos de configuración se denominan docker-compose.yml. yml se creó con la convicción de que todos los datos pueden representarse adecuadamente como una combinación de listas, hashes y datos escalares. La sintaxis es relativamente simple y muy fácil de leer, y algunas de las características del archivo se enumeran a continuación (Yáñez , 2021):

- Los contenidos en YAML se describen utilizando el conjunto de caracteres imprimibles de Unicode, bien en UTF-8 o UTF-16.
- La estructura del documento se denota intentando utilizar espacios en blanco y no se permiten caracteres de tabulación para la sangría.
- Los miembros de la lista se denotan encabezados por un guión (-) con un miembro por línea.

## Docker Hub

Docker Hub es un registro de repositorios de software en la nube y un tipo de biblioteca para imágenes Docker. Es un servicio en línea que consta de repositorios públicos y privados (Digital Guide IONOS, 2019). Los repositorios públicos son accesibles para todos, pero el privado es accesible para el propietario interesado de los repositorios; También hay un costo asociado si almacenamos más de un cierto número de repositorios como privado.

Docker Hub es un repositorio público en la nube, similar a GitHub, para distribuir los contenidos. Está mantenido por la propia Docker y hay multitud de imágenes, de carácter gratuito, que se pueden descargar y así no tener que hacer el trabajo desde cero al poder aprovechar “plantillas” (Villacampa, 2021).

Docker Hub presenta las siguientes características o elementos principales (DesdeLinux, 2020) :

- **Repositorios:** Para la gestión push y pull de imágenes de contenedores.

- **Equipos y organizaciones:** Para gestionar el acceso a repositorios privados de imágenes de contenedores.
- **Imágenes oficiales:** Son imágenes de alta calidad de los contenedores proporcionadas por Docker.
- **Imágenes de editores:** Son imágenes de alta calidad de contenedores proporcionadas por proveedores externos. Estas imágenes certificadas, también incluyen soporte y garantía de compatibilidad con Docker Enterprise.
- **Creaciones:** Imágenes de contenedores propios creadas desde GitHub y Bitbucket, que luego son cargadas a Docker Hub.
- **Webhooks:** Acciones programadas sobre un repositorio cargado para integrar Docker Hub con otros servicios.

## Liquibase

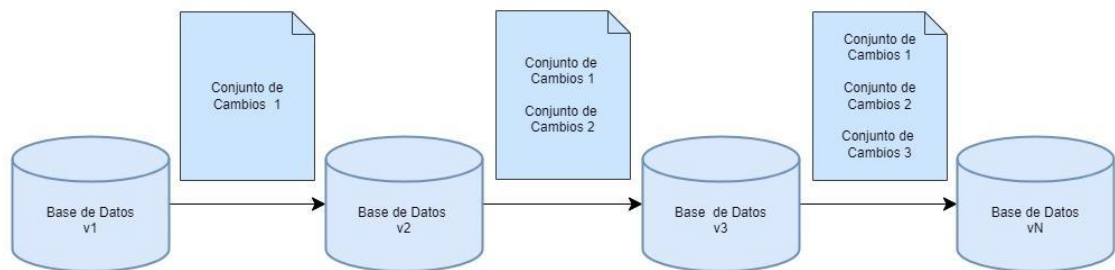
Liquibase es una librería de código abierto (licencia Apache 2.0) para la monitorización, gestión y distribución de los cambios del modelo de base de datos desde la línea de comandos y se integra fácilmente con Maven y Spring (Liquibase, 2022). Fue lanzada en 2006 para facilitar la supervisión de los cambios en las bases de datos, especialmente en entornos de desarrollo de software ágiles (Vivas, 2017).

La integración de Liquibase Spring Boot garantiza que la base de datos de la aplicación se actualice junto con el código de la aplicación, como se muestra en la figura 10 mediante el uso de funciones y configuración automática de Spring Boot.

Hoy en día, en cualquier desarrollo profesional de software no puede faltar un sistema de control de versiones.

## Figura 10

*Ejemplo del sistema de control de versiones de liquibase*



*Nota.* Representa el diagrama de sistema de control de versiones usando Liquibase.

Tomado de (Vivas, 2017)

Uno de los mayores logros de Liquibase es su integración con herramientas CI/CD como Jenkins y Travis CI, y herramientas de gestión de proyectos como Rake y Grunt. Solo se requiere Java 1.6 para poder ejecutarse. Soporta bases de datos como: MySQL, PostgreSQL, Oracle, etc.

Liquibase elimina las tareas manuales y sin sentido que impiden que su equipo innove y reduce la fricción para los equipos que dependen unos de otros para cumplir (Liquibase, 2022).

Con el uso de Liquibase podrás realizar los siguientes cambios de manera más fácil:

- Cambios en el esquema de la base de datos controlados por versiones
- Ordena automáticamente los scripts para su implementación
- Ramificación y fusión para equipos
- Se integra en su producto o crea herramientas, como Jenkins
- Revertir cambios fácilmente

## Jhipster

Jhipster es una plataforma de desarrollo para construir, desarrollar y desplegar rápidamente aplicaciones web modernas y arquitecturas de microservicios (Jhipster, 2022).

JHipster es un generador de Yeoman, que usa Spring Boot y Angular para crear aplicaciones web (Sandrigo, 2017). JHipster nos ofrece un conjunto de herramientas para desarrollar y diseñar los elementos correspondientes al frontend y también al backend de un proyecto de desarrollo.

JHipster, una plataforma de desarrollo full-stack que puede eliminar el 70% de su esfuerzo en escribir código repetitivo y configurar librerías. JHipster le ayuda a centrarse en la lógica empresarial en lugar de dedicar tiempo a la integración de tecnologías (Lima, ACERVO LIMA, 2021). Es compatible con una amplia gama de tecnologías modernas de aplicaciones web.

Por ejemplo, el framework Spring Boot es la base más adecuada para obtener un robusto stack de Java del lado del servidor que nos permita conectarnos a varias bases de datos, motores de virtualización y herramientas de seguimiento. Para conectarse al frontend, utiliza una interfaz REST. En cuanto al desarrollo de frontend, JHipster se basa principalmente en el framework JavaScript Angular.js y en la biblioteca JavaScript React. Estos dos componentes pueden combinarse con Bootstrap.

El objetivo de JHipster es generar una Aplicación Web completa y moderna utilizando:

- Un stack robusto en server-side con Java y Spring Boot.
- Un Front-end moderno y compatible con dispositivos móviles usando Angularjs y Bootstrap
- Un sistema de build avanzado con Yeoman, Webpack o Gulp y Maven o Gradle.



## **NGINX**

NGINX, pronunciado "engine-ex", es un popular software de servidor web de código abierto. En su versión original, funcionaba en servidores web HTTP. Sin embargo, ahora también sirve como proxy inverso, equilibrador de carga HTTP y proxy de correo electrónico para IMAP, POP3 y SMTP (B., 2022).

Nginx fue creado originalmente por Igor Sysoev y su primera versión pública se lanzó en octubre de 2004. Igor diseñó por primera vez el software en respuesta al problema C10K. C10k es la dificultad de gestionar diez mil conexiones al mismo tiempo. Hoy en día, los servidores web tienen que gestionar aún más conexiones. Por ello, NGINX ofrece una arquitectura asíncrona y basada en eventos, una característica que hace de NGINX uno de los servidores más fiables en términos de velocidad y escalabilidad.

Muchos sitios web de alto tráfico utilizan el servicio NGINX debido a su capacidad superior para manejar muchas conexiones y su velocidad entre las empresas que lo utilizan se encuentran Google, Netflix, Adobe, Cloudflare, WordPress.com y muchos otros.

## **Digital Ocean**

Digital Ocean es una popular solución de servicios en la nube, su sede se encuentra en estados Unidos, New York, está equipada con una infraestructura sólida y que brinda múltiples servicios. Comenzó como un negocio de alojamiento administrado llamado Server Stack bajo el liderazgo de Ben y Moisey Uretsky en 2003. Ellos querían crear un producto mediante la combinación de servidores virtuales y alojamiento web. Analizaron la industria del alojamiento en la nube para observar una tendencia de que había menos opciones de alojamiento para las empresas de desarrollo de software, hasta que, en 2011, ofreciendo servicios de aprovisionamiento de servidores y alojamiento en la nube para desarrolladores.

Se utiliza principalmente para alojar sitios web y casos de uso de aplicaciones, y los usuarios lo prefieren por su facilidad de uso. Los centros de datos oceánicos digitales proporcionan un alto nivel de seguridad para las aplicaciones. Es popular entre los

desarrolladores y compite mucho especialmente con los servicios de Google Compute Engine, Amazon Web Services (AWS) y Azure (LATAM, 2020).

Para implantarla como un ambiente de IaaS (Infraestructura como servicio), los desarrolladores lanzan una instancia privada de máquinas virtual (VM), que llama "Droplet". Los desarrolladores pueden elegir el tamaño del "Droplet", la región geográfica y data center (LATAM, 2020).

Digital Ocean ofrece las siguientes ventajas que sus competidores no ofrecen (Román Martín, 2017):

- **Precio:** para proyectos pequeños, tiene un precio asequible. El servicio mínimo (servidor VPS) cuesta alrededor unos 5\$/mes.
- **Experiencia de usuario:** la interfaz y los paneles son extremadamente amigables, agradables y sencillos de utilizar debido a que tiene un UX muy cuidado y tardas pocos segundos en aprender a usarla.
- **Base de conocimiento:** tiene una extensa base de datos y documentación donde ofrecen guías e información para personalizar, configurar o hacer prácticamente cualquier cosa en tu servidor.

## **Certbot**

Certbot es una herramienta de software gratuita y de código abierto para usar automáticamente certificados Let's Encrypt en sitios web administrados manualmente para habilitar HTTPS. Está construido principalmente sobre Python por Electronic Frontier Foundation (EFF) (Lima, ACERVO LIMA, 2020).

Este puede automatizar la emisión e instalación de certificados con cero tiempos de inactividad. También tiene modos de expertos para personas que no quieren autoconfiguración. Es fácil de utilizar, funciona en muchos sistemas operativos, y tiene documentación genial.

## **Capítulo III**

### **Desarrollo del Sistema**

#### **Introducción del Capítulo**

El presente capítulo describe el desarrollo de la propuesta “Desarrollo de un sistema web con metodología DevOps utilizando la herramienta Jenkins, para optimizar la gestión de reservas y consultas en línea de los servicios que ofrece SL Laboratorio Clínico y Bacteriológico “.

Se comienza con la identificación del usuario externo e interno del sistema web y el análisis de sus necesidades identificando los requisitos de software. Dentro del proceso de análisis se incluye la identificación y descripción de los procesos principales del sistema web. Una vez identificadas las necesidades se procede a definir la arquitectura del software mediante el uso del modelo C4.

Continuamos con el marco metodológico, se selecciona y aplica la metodología de desarrollo de software DevOps con la definición de las herramientas de automatización y el proceso de desarrollo en cada una de sus fases. Finalmente, se realizan los diferentes tipos de pruebas con las interfaces de usuarios para así corregir cualquier tipo de error.

#### **Gestión de requisitos**

Para esta etapa se hicieron reuniones con los usuarios expertos de las diferentes áreas del laboratorio clínico. Uno de los requisitos principales es: obtener un sistema web que sea capaz de optimizar la gestión de reservas y consultas en línea de los servicios que ofrece SL laboratorio clínico y bacteriológico.

El sistema web cuenta con los servicios de: Gestión de la información personal del usuario y paciente, reservación de citas, consulta de resultados de exámenes médicos, información de requisitos, gestión administrativa, facturación electrónica, reportes y estadísticas correspondientes a la reserva de citas.

Los usuarios registrados en el sistema web recibirán diferentes visualizaciones dentro del menú de navegación con base en su perfil profesional o rol asignado.

Basándonos en el análisis previo de estos procesos, a continuación, se detallan las épicas del proyecto:

### ***Épicas***

- Como usuario del sistema Web SL Laboratorio Clínico quiero iniciar sesión en el sistema web (correo - contraseña) para poder acceder a las funcionalidades de cada usuario.
- Como Usuario del sistema Web SL Laboratorio Clínico quiero recuperar mi contraseña con un envío de correo electrónico para así no perder información que se tiene almacenada en la cuenta registrada.
- Como usuario del sistema web SL Laboratorio Clínico quiero recuperar la contraseña de mi cuenta para poder colocar una nueva contraseña y poder ingresar de nuevo a mi cuenta.
- Como Usuario Administrador del sistema Web SL Laboratorio Clínico quiero gestionar usuarios para controlar la accesibilidad del usuario del sistema.
- Como laboratorista quiero visualizar las reservas de citas realizadas por los pacientes mediante la web para conocer y verificar los exámenes que se van a realizar en dicha fecha los pacientes.
- Como usuario del sistema web SL Laboratorio Clínico quiero que tenga una página de inicio para visualizar fotos, promociones, servicios que ofrece el laboratorio clínico.
- Como usuario del sistema web SL Laboratorio Clínico quiero que tenga su página de contactos donde irán la información necesaria para que el cliente pueda contactarnos y ayudarles con información más específica de los servicios que ofrece el laboratorio clínico.

- Como paciente quiero ejecutar la reserva de exámenes de laboratorio por medio del aplicativo web para ahorrar tiempo y no ir presencialmente al laboratorio a ejecutar dicha reserva.
- Como paciente quiero consultar los requisitos que tiene cada examen de laboratorio para conocer cómo se debe presentar para hacer dicho examen.
- Como paciente quiero consultar mis resultados en línea para poder verlos, compartirlos o imprimirlos sin necesidad de ir a retirarlos presencialmente en el laboratorio clínico.

### Historia de usuario

A continuación, se presentan las tablas que contienen la descripción de las historias de usuarios junto a los criterios de aceptación, estimación y prioridad.

Para establecer el valor de la prioridad se utilizó una escala entre 1 a 3, donde el 1 representa el valor de mayor importancia para el cliente y 3 representa el valor de menos importancia para aquellas historias de funcionamiento del sistema web.

**Tabla 2**

*Historia de usuario 1*

<b>HU-0001</b>	
<b>Como</b>	Usuario del sistema Web SL Laboratorio Clínico
<b>Quiero</b>	Iniciar sesión en el sistema web (correo - contraseña)
<b>Para</b>	Garantizar la seguridad e información de los datos
<b>Validación</b>	<ul style="list-style-type: none"> <li>• Dada el usuario y contraseña se puede acceder a la página principal del sistema web.</li> <li>• Dado un usuario que no se identifica en el sistema, cuando intente acceder al mismo no se le permite el acceso y muestra un mensaje de error.</li> </ul>

	<ul style="list-style-type: none"> <li>• Con el token obtenido cuando inicio sesión en el sistema web, permite el acceso o deniegue acceso a las pantallas.</li> </ul>		
<b>Prioridad</b>	1	<b>Estimación</b>	2d

**Tabla 3***Historia de usuario 2*

<b>HU-0002</b>			
<b>Como</b>	Usuario del sistema Web SL Laboratorio Clínico		
<b>Quiero</b>	Recuperar mi contraseña con un envío de correo electrónico		
<b>Para</b>	No perder información que se tiene almacenada en la cuenta registrada previamente.		
<b>Validación</b>	<ul style="list-style-type: none"> <li>• Dado una recuperación de contraseña e ingresado el correo electrónico con el cual se creó la cuenta, se procede a enviar un correo de recuperación de contraseña.</li> <li>• Enviado el correo electrónico debe ingresar al enlace que tendrá en el correo para poder redireccionar a la pantalla de cambio de contraseña.</li> </ul>		
<b>Prioridad</b>	1	<b>Estimación</b>	2d

**Tabla 4***Historia de usuario 3*

<b>HU-0003</b>	
<b>Como</b>	Usuario Administrador del sistema Web SL Laboratorio Clínico
<b>Quiero</b>	Gestionar los usuarios
<b>Para</b>	Controlar la accesibilidad de los usuarios del sistema.
<b>Validación</b>	<ul style="list-style-type: none"> <li>• Dado que el usuario administrador requiera visualizar información de todos los usuarios habilitados y deshabilitados, podrá visualizar la información de los usuarios.</li> </ul>

---

	<ul style="list-style-type: none"> <li>• Dado que el usuario administrador requiera modificar la información de un usuario cuando se encuentre algún error de digitación, entonces podrá realizar la modificación adecuadamente.</li> <li>• Dado que cualquier usuario registrado el sistema no cuenta con el rol administrador cuando intente acceder a la administración del sistema, se le muestra un mensaje de error.</li> </ul>		
<b>Prioridad</b>	1	<b>Estimación</b>	2d

---

**Tabla 5***Historia de usuario 4*


---

<b><i>HU-0004</i></b>	
<b>Como</b>	Laboratorista
<b>Quiero</b>	Gestionar las reservas de citas realizadas por los pacientes mediante la aplicación web
<b>Para</b>	Conocer y verificar los exámenes que se va a realizar dicha fecha los pacientes
<b>Validación</b>	<ul style="list-style-type: none"> <li>• Dado que el usuario laboratorista aún no tenga registros de reservas, se mostrará un mensaje de "Aún no se encuentran reservas."</li> <li>• Dado que el usuario laboratorista requiera cambiar de estado a la reserva, se muestra un modal para confirmar la acción que desea ejecutar.</li> <li>• Dado que el usuario laboratorista requiere eliminar una reserva, se muestra un modal para confirmar la acción que desea ejecutar.</li> <li>• Dado que el usuario laboratorista requiere agregar una nueva cita cuando lo considere necesario lo pueda realizar de manera fácil y segura.</li> <li>• Dado que el usuario laboratorista requiera modificar una reserva cuando exista algún error de digitación o cuando lo requiera necesario lo pueda ejecutar de manera fácil.</li> </ul>

---

<b>Prioridad</b>	2	<b>Estimación</b>	2d
------------------	---	-------------------	----

**Tabla 6***Historia de usuario 5*

<b>HU-0005</b>			
<b>Como</b>	Laboratorista		
<b>Quiero</b>	Gestionar las reservas de citas realizadas por los pacientes mediante la aplicación web		
<b>Para</b>	Conocer y verificar los exámenes que se va a realizar dicha fecha los pacientes		
<b>Validación</b>	<ul style="list-style-type: none"> <li>● Dado que el usuario laboratorista aún no tenga registros de reservas, se mostrará un mensaje de “Aún no se encuentran reservas.”</li> <li>● Dado que el usuario laboratorista requiera cambiar de estado a la reserva, se muestra un modal para confirmar la acción que desea ejecutar.</li> <li>● Dado que el usuario laboratorista requiere eliminar una reserva, se muestra un modal para confirmar la acción que desea ejecutar.</li> <li>● Dado que el usuario laboratorista requiere agregar una nueva cita cuando lo considere necesario lo pueda ejecutar de manera fácil y segura.</li> <li>● Dado que el usuario laboratorista requiera modificar una reserva cuando exista algún error de digitación o cuando lo requiera necesario lo pueda ejecutar de manera fácil.</li> </ul>		
<b>Prioridad</b>	2	<b>Estimación</b>	2d

**Tabla 7***Historia de usuario 6*

<b>HU-0006</b>	
<b>Como</b>	Usuario de la aplicación SL Laboratorio Clínico



<b>Quiero</b>	Que tenga una Landing Page		
<b>Para</b>	Visualizar en la página de inicio fotos, promociones y los distintos servicios que ofrece el Laboratorio clínico y bacteriológico		
<b>Validación</b>	<ul style="list-style-type: none"> <li>• Dado que el usuario de la aplicación landing page puede visualizar imágenes relevantes del laboratorio clínico.</li> <li>• Dado que el usuario de la aplicación landing page puede visualizar los distintos servicios que ofrece el laboratorio clínico y bacteriológico.</li> <li>• Dado que el usuario de la aplicación landing page puede visualizar las promociones que existen en el laboratorio clínico y bacteriológico.</li> </ul>		
<b>Prioridad</b>	3	<b>Estimación</b>	3d

**Tabla 8***Historia de usuario 7*

<b>HU-0007</b>			
<b>Como</b>	Usuario de la aplicación SL Laboratorio Clínico		
<b>Quiero</b>	Visualizar una pestaña de contactos		
<b>Para</b>	poder contactar con el laboratorio clínico y ayudarlos con información más específica sobre los servicios que se ofrece.		
<b>Validación</b>	<ul style="list-style-type: none"> <li>• Dado que el usuario de la aplicación landing page puede visualizar una pestaña de contactarnos con información de contacto del laboratorio (imágenes del laboratorio, número celular, correo electrónico).</li> </ul>		
<b>Prioridad</b>	3	<b>Estimación</b>	1d

**Tabla 9***Historia de usuario 8*

<b>HU-0008</b>	
<b>Como</b>	Usuario de la aplicación SL Laboratorio Clínico

<b>Quiero</b>	Realizar la reserva de examen de laboratorio
<b>Para</b>	Ahorrar tiempo y no ir presencialmente al laboratorio a realizar la reserva.
<b>Validación</b>	<ul style="list-style-type: none"> <li>• Dado que el usuario de la aplicación landing page puedes crear una reserva del examen de laboratorio cuando lo requiera.</li> <li>• Dado el usuario que genere una reserva, se debe enviar un correo electrónico al usuario que está en ese momento generando la reserva con un código de validación para poder concluir con la creación del examen.</li> <li>• Dado que el usuario finaliza la creación de la reserva del examen, se debe notificar mediante un correo electrónico al usuario laboratorista para que proceda a la cotización del examen.</li> </ul>
<b>Prioridad</b>	3
<b>Estimación</b>	2d

**Tabla 10**

*Historia de usuario 9*

<b>HU-0009</b>	
<b>Como</b>	Usuario de la aplicación SL Laboratorio Clínico
<b>Quiero</b>	Editar la reserva de examen de laboratorio
<b>Para</b>	Poder corregir los errores de digitación o especificar un nuevo examen de laboratorio.
<b>Validación</b>	<ul style="list-style-type: none"> <li>• Dado que el usuario de la aplicación landing page puede buscar su reserva de acuerdo al número de identificación, se desplegará todas las reservas que tiene, en caso de no existir ninguna reserva se procede a mostrar un mensaje “No existen reservas para el número de identificación antes mencionado.”</li> <li>• Dado que el usuario requiera realizar la modificación de una reserva, se procederá al envío de un correo con una clave personal para poder culminar con la modificación.</li> </ul>

<b>Prioridad</b>	3	<b>Estimación</b>	2d
------------------	---	-------------------	----

**Tabla 11***Historia de usuario 10*

<b>HU-0010</b>			
<b>Como</b>	Usuario de la aplicación SL Laboratorio Clínico		
<b>Quiero</b>	Eliminar la reserva de examen de laboratorio		
<b>Para</b>	No asistir el día del examen.		
<b>Validación</b>	<ul style="list-style-type: none"> <li>• Dado que el usuario de la aplicación landing page puede buscar su reserva de acuerdo al número de identificación, se desplegará todas las reservas que tiene, en caso de no existir ninguna reserva se procede a mostrar un mensaje “No existen reservas para el número de identificación antes mencionado.”</li> <li>• Dado que el usuario requiera proceder a la eliminación de una reserva, se procederá al envío de un correo con una clave personal para poder eliminar la reserva, se mostrará un modal para que ingrese la clave y procederá a la confinación de la misma.</li> </ul>		
<b>Prioridad</b>	3	<b>Estimación</b>	1d

**Tabla 12***Historia de usuario 11*

<b>HU-0011</b>			
<b>Como</b>	Usuario de la aplicación SL Laboratorio Clínico		
<b>Quiero</b>	Consultar los requisitos que tiene cada examen de laboratorio		
<b>Para</b>	Conocer cómo se debe presentar para realizar el examen.		
<b>Validación</b>	<ul style="list-style-type: none"> <li>• Dado que el usuario de la aplicación landing page pueda visualizar información más detallada sobre el examen y su forma de presentarse a la misma, caso que no se encuentre información de cómo debe presentarse el día del examen, sé</li> </ul>		

mostrar un mensaje “Estamos trabajando en la información gracias por su comprensión”.			
<b>Prioridad</b>	3	<b>Estimación</b>	1d

**Tabla 13***Historia de usuario 12*

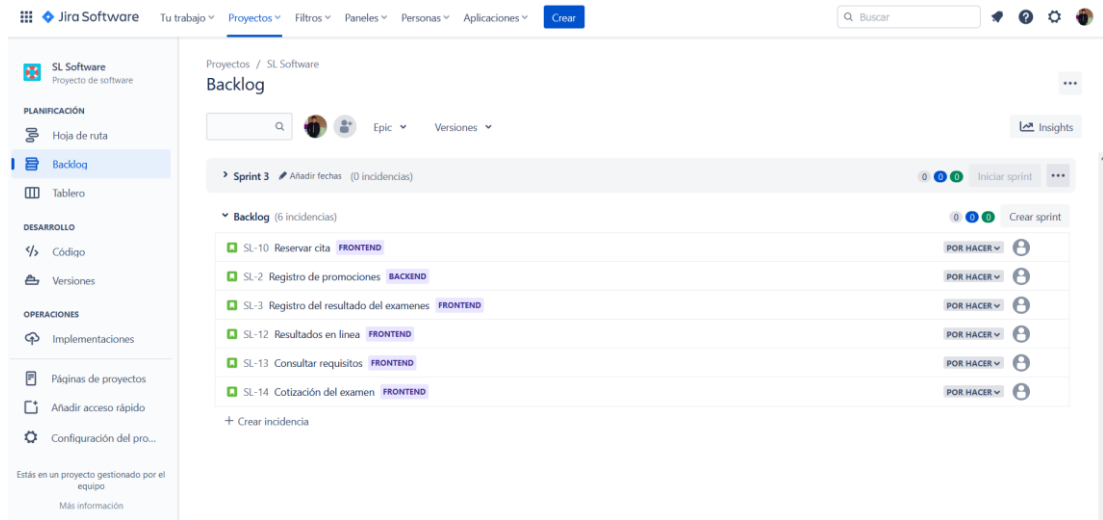
<b>HU-0012</b>	
<b>Como</b>	Usuario de la aplicación SL Laboratorio Clínico
<b>Quiero</b>	Consultar mis resultados en línea
<b>Para</b>	Poder verlos, compartirlos o imprimirlos sin necesidad de ir a retirarlos presencialmente en el laboratorio clínico.
<b>Validación</b>	<ul style="list-style-type: none"> <li>• Dado que el usuario de la aplicación landing page pueda visualizar sus resultados más detallados sobre el examen efectuado, caso que no se encuentre información se mostrará un mensaje “Estamos trabajando en la información gracias por su comprensión”.</li> </ul>
<b>Prioridad</b>	3
<b>Estimación</b>	2d

**Creación del Product Backlog**

Para la correcta gestión de las historias de usuario se procede a utilizar la herramienta Jira que nos proporciona un tablero Kanban que nos ayuda a mejorar la gestión de las tareas a realizarse en cada Sprint como se representa en las figuras 11.

## Figura 11

### *Product Backlog con tareas por hacer*

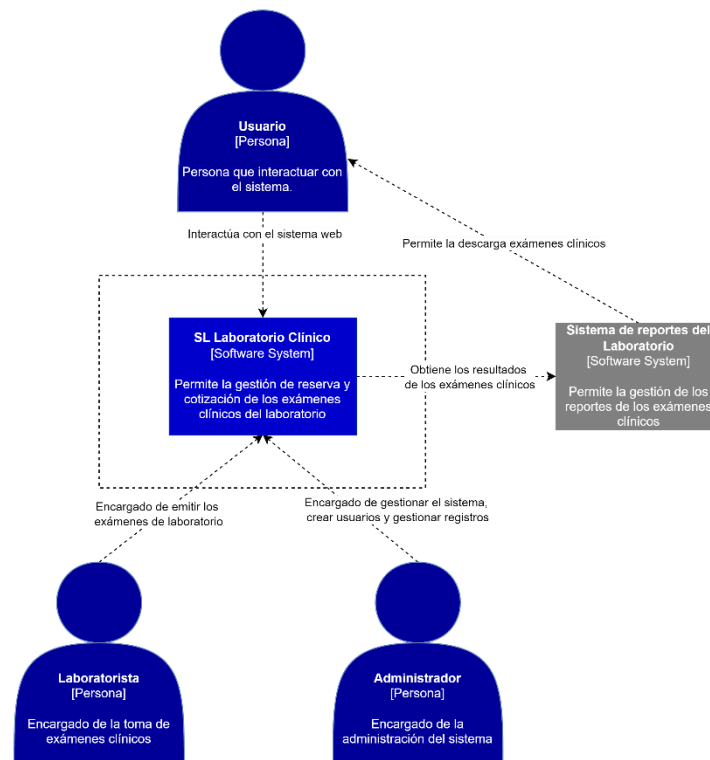


## Arquitectura de software

Según lo mencionado por Simón Brown (Brown, 2018) existe una forma sensacional de comunicar cómo se plantea la construcción de un sistema software mediante los diagramas de arquitectura. La propuesta del autor fue un nuevo modelo de arquitectura llamado “modelo-c4” la cual es un diagrama de cuatro niveles, los mismos que serán planteados para la especificación de la arquitectura de un sistema software.

### ***Diagrama de contexto del sistema***

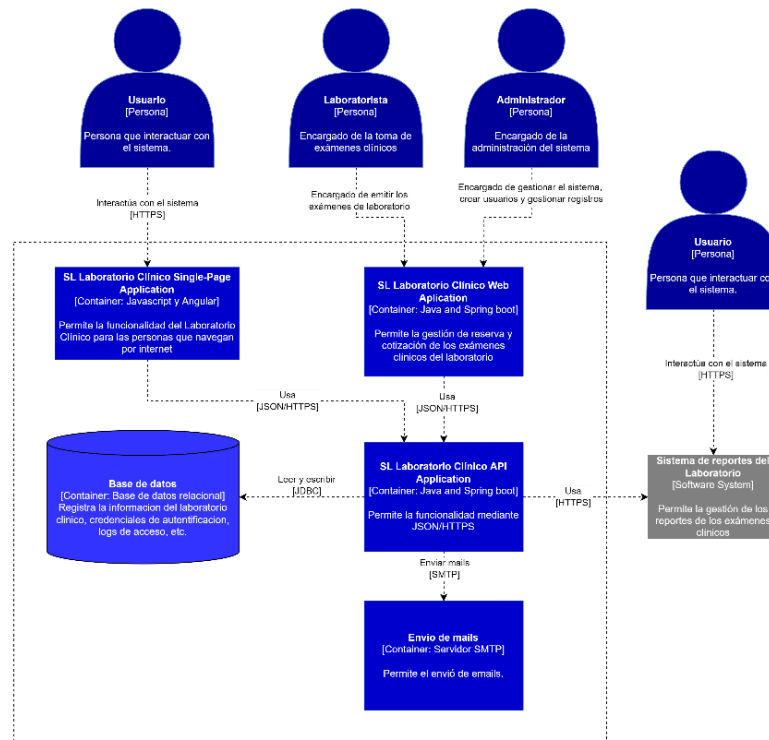
En la figura 12 se presenta el diagrama de contexto para el desarrollo del sistema web. En la parte central del diagrama se encuentra el sistema denominado SL Laboratorio Clínico y a los costados se puede evidenciar los usuarios que interactúan con el sistema. Además, se puede observar en color gris el sistema externo que interactúa con el sistema.

**Figura 12***Diagrama de contexto del sistema**Diagrama de contenedores del sistema*

En el siguiente nivel del modelo c4 permite la visualización de los contenedores del sistema, la relación con el usuario y la relación con el sistema externo. Como se observa en la figura 13, un usuario accede al sistema en el cual puede gestionar las funcionalidades que ofrece el sistema SL Laboratorio clínico Single-Page Application, mediante la aplicación SL Laboratorio Clínico Web Application el laboratorista y el personal administrativo encargado puede gestionar las distintas funcionalidades de sistema. De igual manera, las dos aplicaciones antes mencionadas interactúan con la API “SL Laboratorio Clínico API Application” en la cual realiza las transacciones necesarias para guardar, extraer, actualizar y borrar información en una base de datos Postgresql, de igual manera la API “SL Laboratorio Clínico API Application” se comunica con un servicio SMTP el cual nos ayuda al envío de correos electrónicos.

Figura 13

## Diagrama de contenedores del sistema



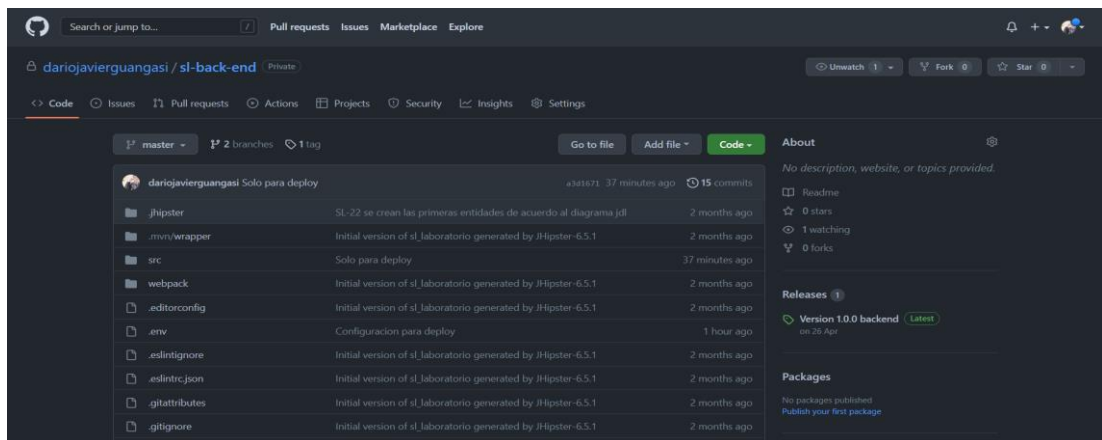
## Implementación y configuración

**Configuración del repositorio central**

Para la configuración del repositorio central del proyecto se procede a la configuración de un repositorio de código fuente. De aquí en adelante, la descripción de cada una de las fases del ciclo de vida de DEVOPS se basarán en el repositorio de código “sl-back-end” y “sl-frontend” tal como se muestra en las figuras 14 y 15.

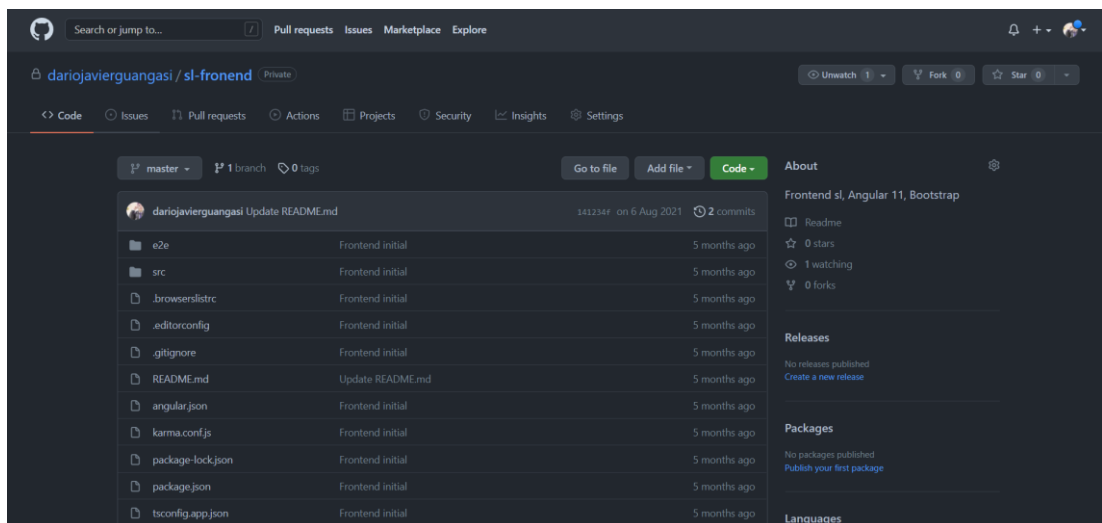
**Figura 14**

*Repositorio backend de código del proyecto*



**Figura 15**

*Repositorio frontend de código del proyecto*



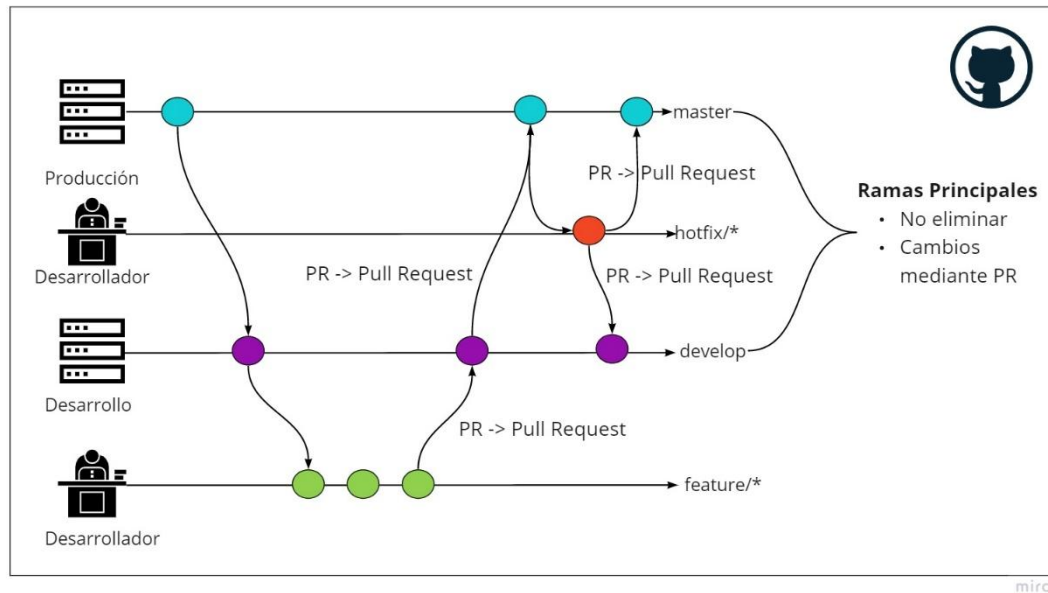
Cabe recalcar que los dos proyectos van a tener una rama principal que va a ser la rama MASTER o MAIN de la cual partimos y tendremos una rama secundaria DEVELOP para la codificación y de la cual saldrán varias ramas para cada una de las tareas según corresponda para luego unir las y verificar que no den ningún conflicto para posteriormente unirla a la rama MASTER y poner el sistema en producción.



El flujo que se va a manejar en el transcurso del desarrollo del sistema se visualiza en la figura 16.

**Figura 16**

*Flujo de GitHub para la creación de tareas y hotfix*



- Master: Rama principal del proyecto donde comenzamos el proyecto.
- Develop: Rama que utilizamos para el desarrollo del sistema.
- Feature: Rama usada para agregar nuevas funcionalidades al sistema, una vez agregada las nuevas funcionalidades se iguala a la rama develop.
- Hotfix: Rama empleada para corregir errores que pueden surgir en producción, se basa en la rama master y una vez realizada la corrección se iguala a la rama develop y se iguala a la rama master.

### **Configuración de Jenkins**

Para iniciar con el proceso de CI/CD con la herramienta de Jenkins utilizaremos un entorno basado en Linux en su distribución de Ubuntu en la versión 20.04

Los requisitos que necesitaremos para la óptima configuración del servidor de DevOps son:

- Servidor Linux
- GitHub
- Jenkins
- Docker
- DockerHub

Para continuar con la configuración del servidor de devops procedemos a la instalación de Jenkins.

**Instalar Jenkins en un servidor en la nube.** Para la correcta instalación de Jenkins necesitamos:

1. Instalar la versión de Java 11 ejecutando los comandos mostrados en la figura 17.

### Figura 17

*Comandos para la instalación de Java en un servidor en la nube.*

```
sudo apt-get update
sudo apt-get install openjdk-11-jdk (Mandatory for GTC)
java -version
```

2. Instalar Jenkins ejecutando los comandos mostrados en la figura 18.

### Figura 18

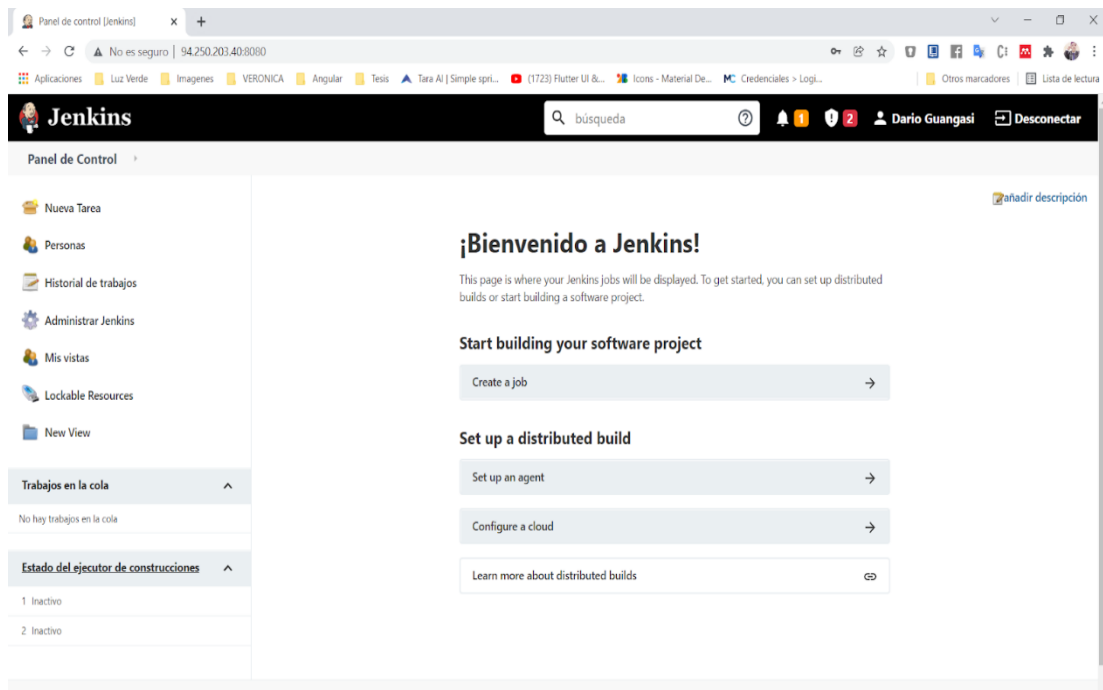
*Comandos para la instalación de Jenkins en un servidor en la nube.*

```
wget -q -O - https://pkg.jenkins.io/debian-stable/jenkins.io.key | sudo apt-
key add -
sudo apt-add-repository "deb https://pkg.jenkins.io/debian-stable binary/"
sudo apt-get update
sudo apt-get install jenkins
sudo timedatectl set-timezone America/Lima
```

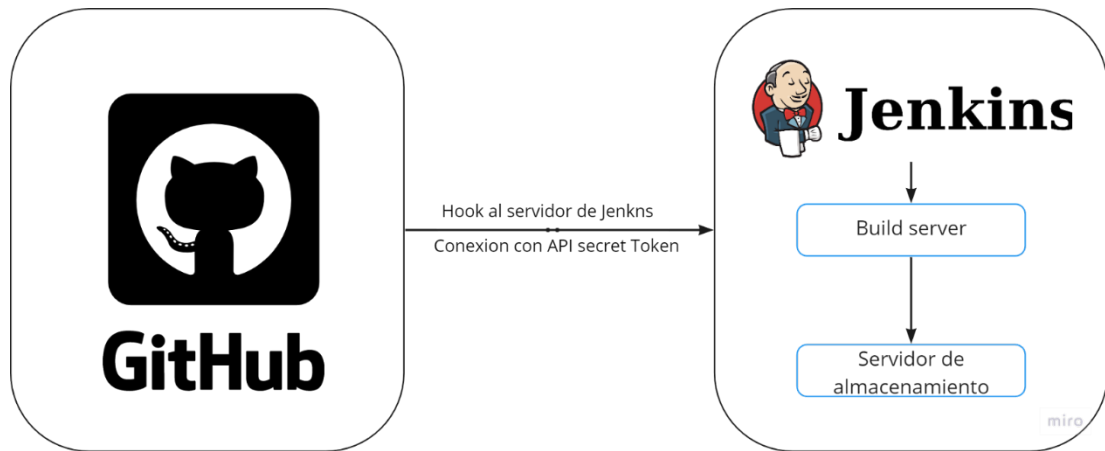
Una vez ejecutado terminado la instalación de Jenkins ingresamos a la url del servidor en la nube con el puerto 8080, debemos realizar una configuración inicial y agregar un usuario como se muestra en la figura 19.

## Figura 19

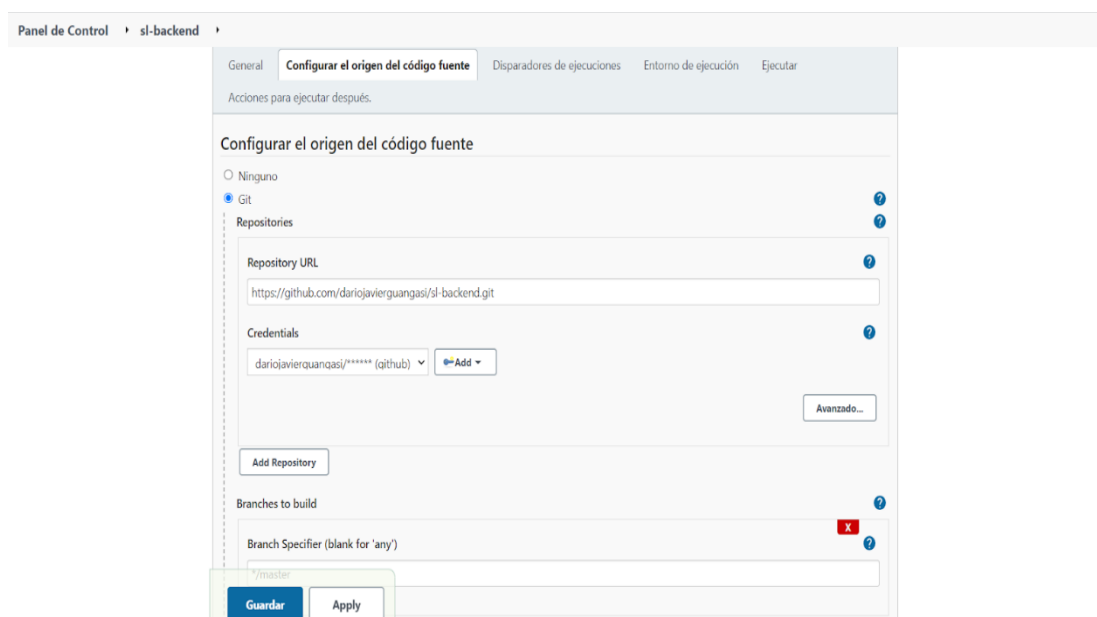
### *Entorno de ejecución de Jenkins*



**Conexión de Jenkins con GitHub.** En este apartado se procederá a la configuración de Jenkins para poder hacer la conexión a GitHub para obtener el código fuente de nuestra aplicación, la manera que se conecta Jenkins con GitHub se representa en la figura 20.

**Figura 20***Conexión de GitHub con Jenkins*

Como se aprecia en la figura 21, Jenkins se conectará con GitHub, para poder construir el entregable del proyecto, ejecutando a la vez los test unitarios que comprueben el correcto funcionamiento del paquete antes de proceder a generar y subirlo a un ambiente de producción.

**Figura 21***Configuración de un Job de Jenkins junto con GitHub*

**Configuración de Jenkins con Docker.** Para configurar un entorno de Jenkins junto con Docker, es necesario instalar docker en el mismo servidor que está Jenkins.

- Instalación de docker en el servidor en la nube

Para la instalación de Docker ejecutamos en una shell los siguientes comandos

figura 22:

### Figura 22

*Comandos de instalación Docker*

```
sudo apt-get update
sudo apt-get install curl apt-transport-https ca-certificates software-properties-common
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable"
sudo apt-get update
sudo apt install docker-ce
sudo usermod -aG docker $(whoami)
```

- Jenkins con Docker

Continuando con la instalación de docker en el servidor descargamos una imagen de Maven con el comando de la figura 23.

### Figura 23

*Comando para descargar una imagen Maven en Docker*

```
docker pull maven:3.8.4-openjdk-8
```

Creamos un contenedor del proyecto compartiendo un volumen para poder almacenar el proyecto compilado para poder subirlo a Docker-hub.

Una vez terminada la respectiva configuración de Jenkins se puede observar el ambiente como quedó en la figura 24, también podemos realizar pruebas de cómo se ejecutan los Jobs de trabajo figura 25 y 26.

Figura 24

Ambiente de Jenkins para el proyecto

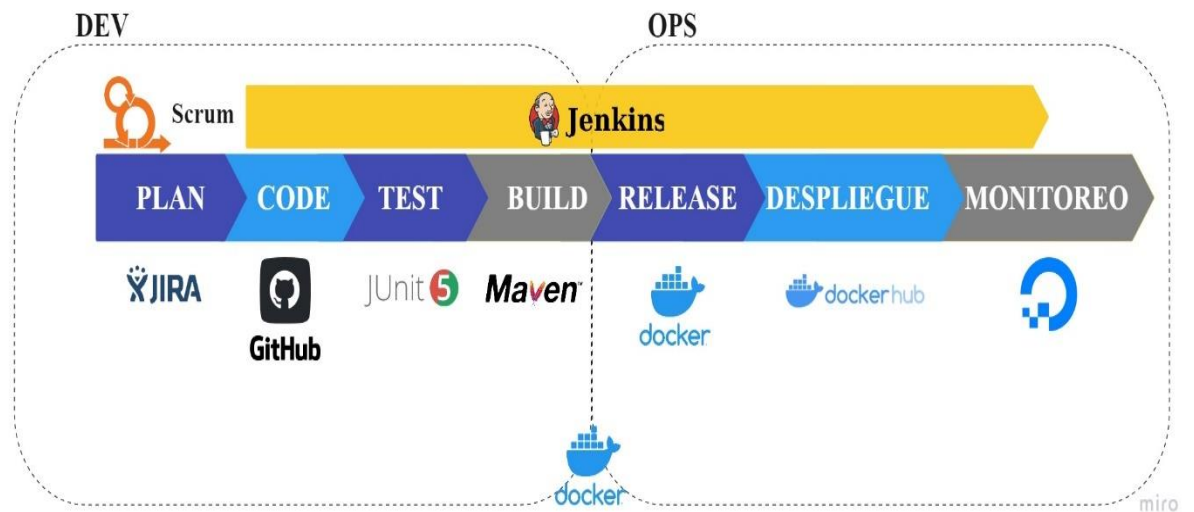


Figura 25

Job ejecutado con éxito

```

Panel de Control > sl-backend > #2
[INFO] --- maven-compiler-plugin:3.8.1:compile (default-compile) @ sl-backend ---
[INFO] Changes detected - recompiling the module!
[INFO] Compiling 1 source file to /var/lib/jenkins/workspace/sl-backend/target/classes
[INFO] --- maven-resources-plugin:3.2.0:testResources (default-testResources) @ sl-backend ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] Using 'UTF-8' encoding to copy filtered properties files.
[INFO] skip non existing resourceDirectory /var/lib/jenkins/workspace/sl-backend/src/test/resources
[INFO] --- maven-compiler-plugin:3.8.1:testCompile (default-testCompile) @ sl-backend ---
[INFO] Changes detected - recompiling the module!
[INFO] Compiling 1 source file to /var/lib/jenkins/workspace/sl-backend/target/test-classes
[INFO] --- maven-surefire-plugin:2.22.2:test (default-test) @ sl-backend ---
[INFO] Tests are skipped.
[INFO] --- maven-jar-plugin:3.2.0:jar (default-jar) @ sl-backend ---
[INFO] Building jar: /var/lib/jenkins/workspace/sl-backend/target/sl-backend-0.0.1-SNAPSHOT.jar
[INFO] --- spring-boot-maven-plugin:2.5.3:repackage (repackage) @ sl-backend ---
[INFO] Replacing main artifact with repackaged archive
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 25.837 s
[INFO] Finished at: 2022-01-14T10:56:56-05:00
[INFO] -----
Archiving artifacts
Finished: SUCCESS

```

Figura 26

Falla de un Job en un test

```

Panel de Control > sl-backend > #18
[INFO] Results:
[INFO]
[ERROR] Errors:
[ERROR]   S1BackendApplicationTests.contextLoads > IllegalStateException Failed to load Applicati...
[INFO]
[ERROR] Tests run: 1, Failures: 0, Errors: 1, Skipped: 0
[INFO]
[INFO] -----
[INFO] BUILD FAILURE
[INFO] -----
[INFO] Total time: 46.771 s
[INFO] Finished at: 2022-03-16T10:03:50-05:00
[INFO] -----
[ERROR] Failed to execute goal org.apache.maven.plugins:maven-surefire-plugin:2.22.2:test (default-test) on project sl-backend: There are test failures.
[ERROR]
[ERROR] Please refer to /var/lib/jenkins/workspace/sl-backend/target/surefire-reports for the individual test results.
[ERROR] Please refer to dump files (if any exist) [data].dump, [data]-jvmRun[N].dump and [data].dumpstream.
[ERROR] -> [Help 1]
[ERROR]
[ERROR] To see the full stack trace of the errors, re-run Maven with the -e switch.
[ERROR] Re-run Maven using the -X switch to enable full debug logging.
[ERROR]
[ERROR] For more information about the errors and possible solutions, please read the following articles:
[ERROR] [Help 1] http://cwiki.apache.org/confluence/display/MAVEN/MojoFailureException
Build step 'Invoke top-level Maven targets' marked build as failure
Archiving artifacts
[NS-CLEANUP] Deleting project workspace...
[NS-CLEANUP] Deferred wipeout is used...
[NS-CLEANUP] done
Finished: FAILURE

```

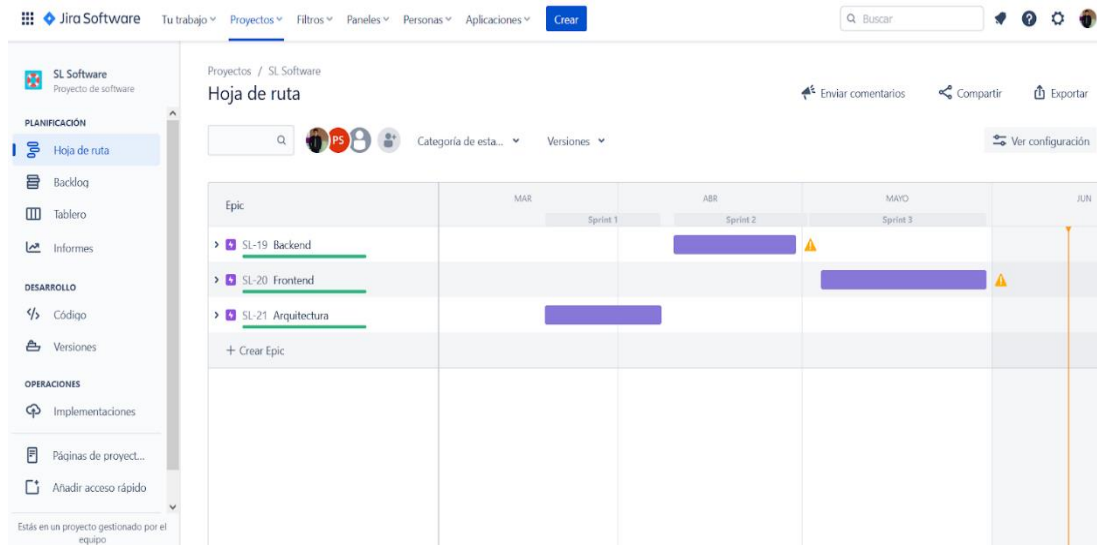
## Construcción de iteración

**Fase de planificación.** Para la gestión de las tareas utilizamos la herramienta Jira que nos permite gestionar las tareas y poder crear los sprints con las mismas tareas, como se puede observar en la figura 27, dentro de la herramienta de Jira también se puede visualizar una hoja de ruta en la que se visualiza las tareas a finalizadas y los sprints que ya se finalizaron figura 28.

Figura 27

Gestión de tareas y sprints en Jira

The screenshot shows the Jira Software interface for a project named 'SL Software'. The main view is the 'Backlog', which is organized into an 'Epic' and a 'Sprint'. The 'Epic' section shows a list of tasks, including 'Backend', 'Frontend', and 'Arquitectura'. The 'Sprint' section shows a list of tasks, including 'Crear proyectos e inicializarlos en GITHUB', 'Instalar Jenkins en servidor en la nube', 'Crear las entidades, dtos, mapper, servicios en el backend', and 'Inicio de sesión'. The tasks are color-coded by status: 'LISTO' (green) and 'POR HACER' (grey). The interface also includes a sidebar with navigation options like 'Hoja de ruta', 'Backlog', 'Tablero', 'Código', and 'Implementaciones'.

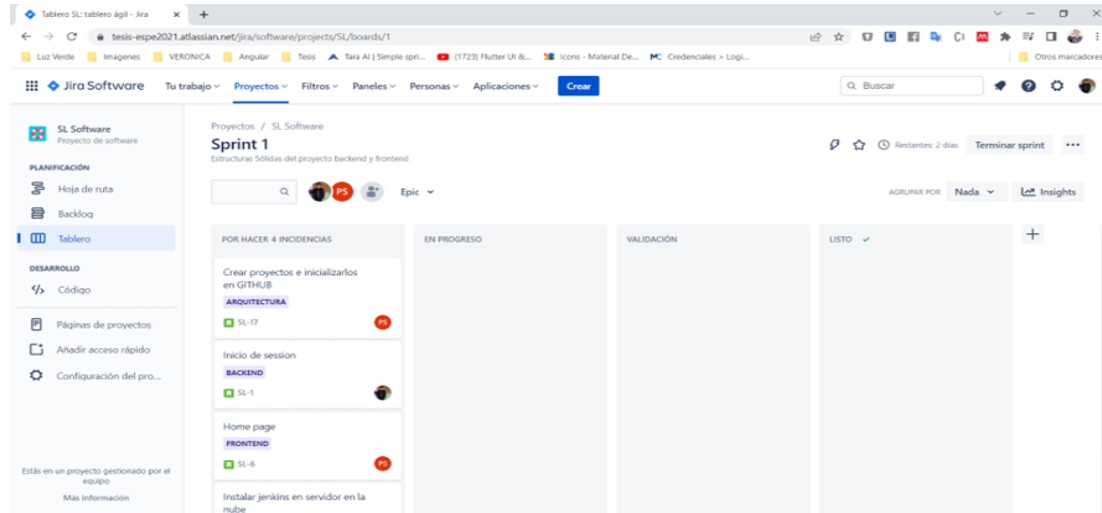
**Figura 28***Hoja de ruta en Jira*

Otra forma de gestionar las tareas es mediante los tableros kanban donde se visualizan todas las tareas dependiendo de su estado. Las historias de usuarios, las futuras implementaciones o correcciones son detalladas como una tarjeta dentro del backlog, posterior a la creación de la tarjeta se designa al desarrollador que va a realizar la tarea y también el sprint a que pertenece, un ejemplo del tablero kanban se puede observar en la figura 29.



Figura 29

## Tablero kanban

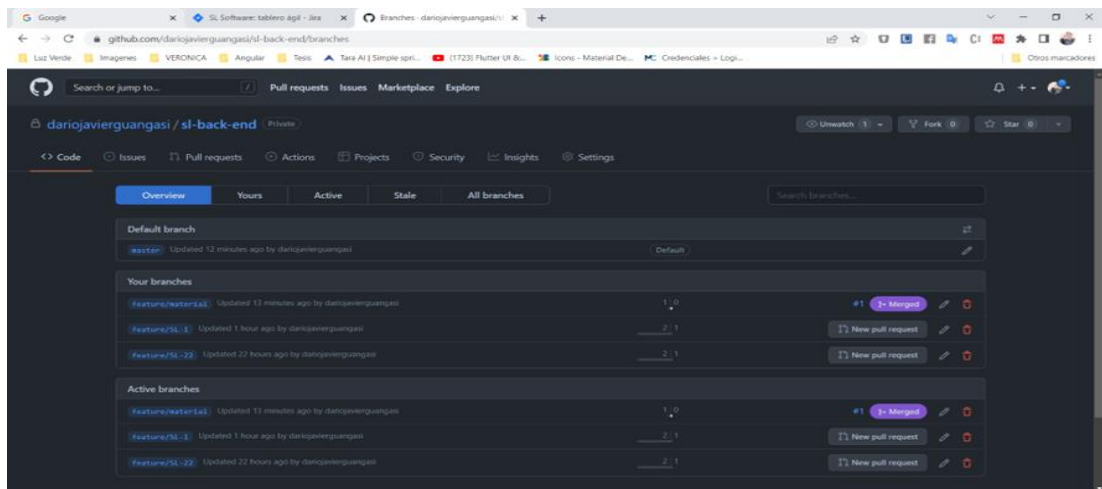


**Fase de Codificación.** En esta fase se incluyen las herramientas para el desarrollo de la aplicación, visualización y gestión de código fuente. En la figura 16 se presenta todo el flujo para la creación de las ramas para gestionar el código fuente y así poder optar a diferentes flujos de trabajo.

En la figura 30 y 31 se puede observar las diferentes ramas que se fueron creando a partir de la rama secundaria que es develop, es la rama que se utiliza para realizar las nuevas implementaciones de código o nuevas tareas.

**Figura 30**

*Ramas creadas para la gestión de código fuente*



**Fase de Pruebas.** En esta fase se ejecutarán las pruebas unitarias escritas por el desarrollador, para lo cual en nuestro servidor de Jenkins procederemos a configurar los comandos para ejecutar las pruebas unitarias del proyecto, ver figura 31 y 32, el resultado de las pruebas se visualiza en la ejecución del Job como se indica en la figura 33 y 34.

**Figura 31**

*Configuración de Jenkins para ejecución de pruebas unitarias si-back-end.*

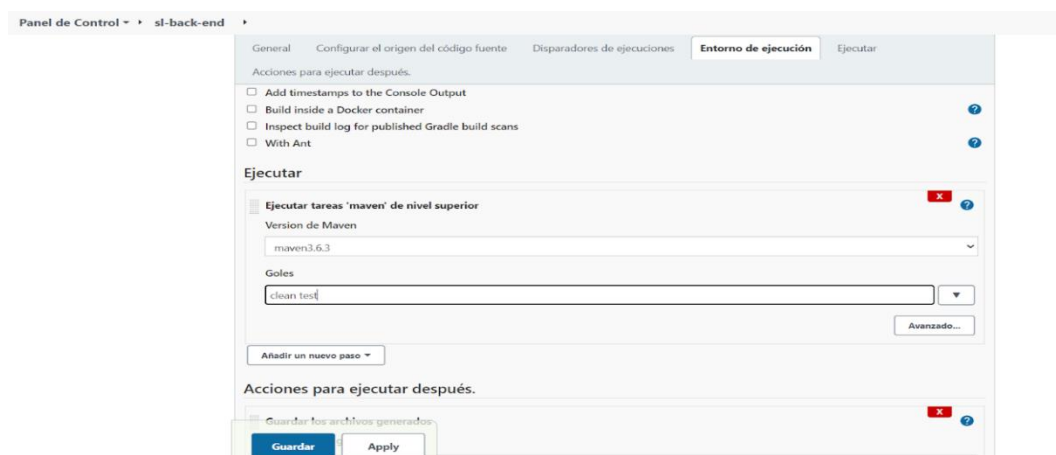


Figura 32

Configuración de Jenkins para ejecución de pruebas unitarias sl-frontend

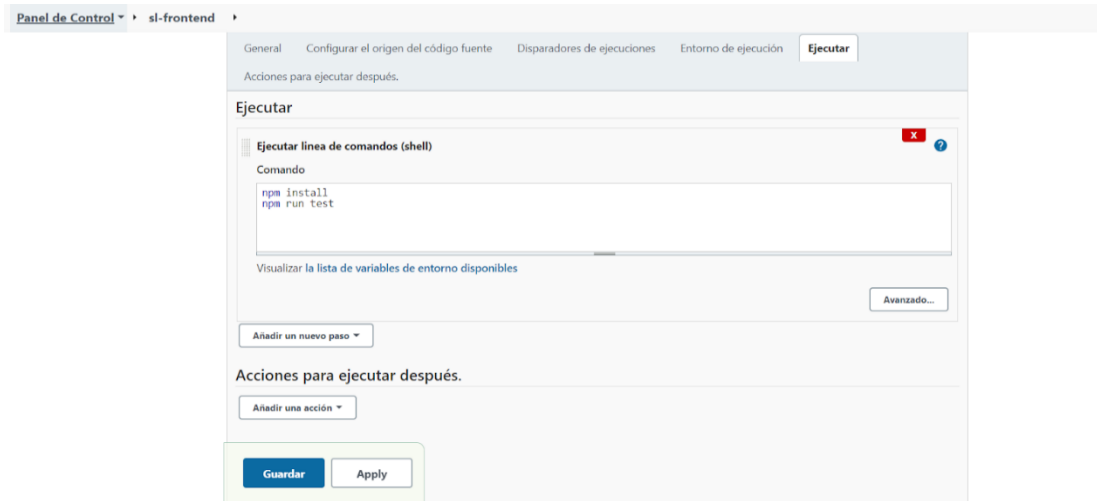


Figura 33

Finalización de ejecución de pruebas unitarias en el job de Jenkins sl-back-end.



## Figura 34

*Finalización de ejecución de pruebas unitarias en el job de Jenkins sl-frontend.*

Panel de Control ▸ sl-frontend ▸ #3						
paciente-detail.component.html	100	100	100	100		
paciente-detail.component.ts	88.89	100	75	85.71	22	
paciente-update.component.html	100	100	100	100		
paciente-update.component.ts	83.87	66.67	66.67	84	35,36,37,97	
paciente.component.html	100	100	100	100		
paciente.component.ts	91.49	70	80	93.18	99,107,108	
paciente.service.ts	100	66.67	100	100	61,62,68,70	
app/shared/alert	97.73	92.31	83.33	97.56		
alert-error.component.ts	97.73	92.31	83.33	97.56	76	
app/shared/constants	100	100	100	100		
error.constants.ts	100	100	100	100		
input.constants.ts	100	100	100	100		
pagination.constants.ts	100	100	100	100		
app/shared/login	90	85.71	66.67	91.89		
login.component.html	100	100	100	100		
login.component.ts	89.74	85.71	66.67	91.67	35,63,79	
app/shared/model	100	100	100	100		
cita.model.ts	100	100	100	100		
examen.model.ts	100	100	100	100		
paciente.model.ts	100	100	100	100		
app/shared/util	100	83.33	100	100		
request-util.ts	100	83.33	100	100	5	
----- ----- ----- ----- ----- ----- -----						
Test Suites: 39 passed, 39 total						
Tests: 141 passed, 141 total						
Snapshots: 0 total						
Time: 41.063s						
Ran all test suites.						
Finished: SUCCESS						

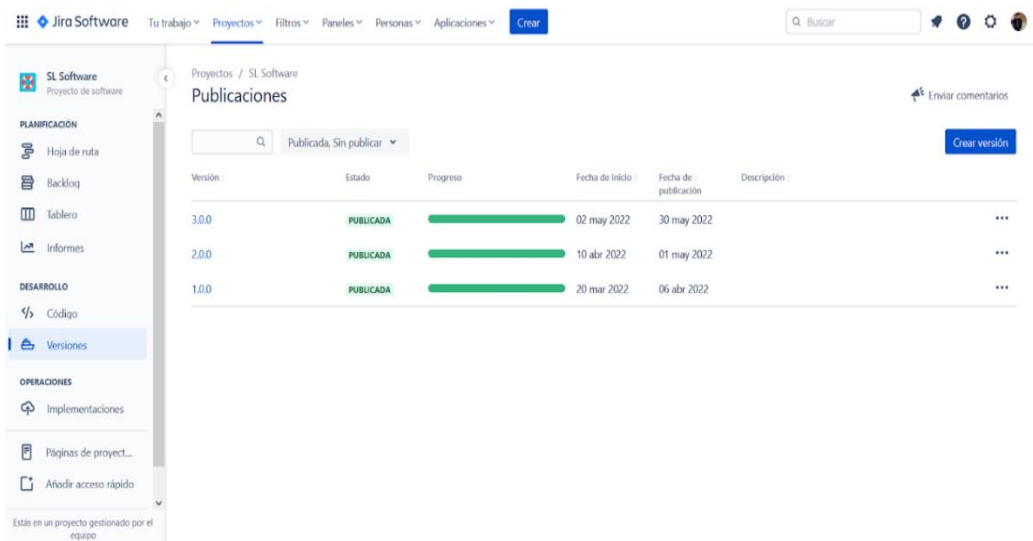
*Nota.* Si las pruebas ejecutadas no concluyen de una forma satisfactoria, no podemos avanzar a la siguiente fase.

**Fase de Construcción.** En esta fase de construcción se inician los procesos de integración y entrega continua conocidos como CI y CD. Para esta fase nos apoyaremos con la herramienta de Jira para poder construir nuestros artefactos posteriormente en la fase de releases.

Jira nos proporciona un apartado de versiones, la cual nos ayuda a verificar cuál es la versión que se publica y las tareas que están dentro de la versión como se puede observar en la figura 35.

## Figura 35

### Versiones en Jira



**Fase de Release.** Concluida la fase de pruebas se procede a la construcción de los artefactos que serán empaquetados mediante el uso de imágenes docker, para la construcción de las imágenes docker es necesario crear un archivo "Dockerfile" dentro de la carpeta del proyecto como se muestra en la figura 36 y 37, posterior a la construcción de la imagen docker es necesario subirlo a un repositorio de artefactos como docker-hub evidenciados en la figura 38 y 39.

## Figura 36

*Contenido del archivo Dockerfile aplicacion sl-back-end*

```
FROM openjdk:8-alpine
ARG JAR_FILE=target/*.jar
COPY ${JAR_FILE} app.jar
EXPOSE 8080:8080
ENTRYPOINT ["java", "-jar", "/app.jar"]
```

## Figura 37

Contenido del archivo Dockerfile aplicacion sl-frontend

```
FROM nginx:1.14.2-alpine
RUN rm -rf /usr/share/nginx/html/*
COPY nginx.conf /etc/nginx/nginx.conf
COPY dist /usr/share/nginx/html
EXPOSE 80
ENTRYPOINT ["nginx", "-g", "daemon off;"]
EXPOSE 80
```

## Figura 38

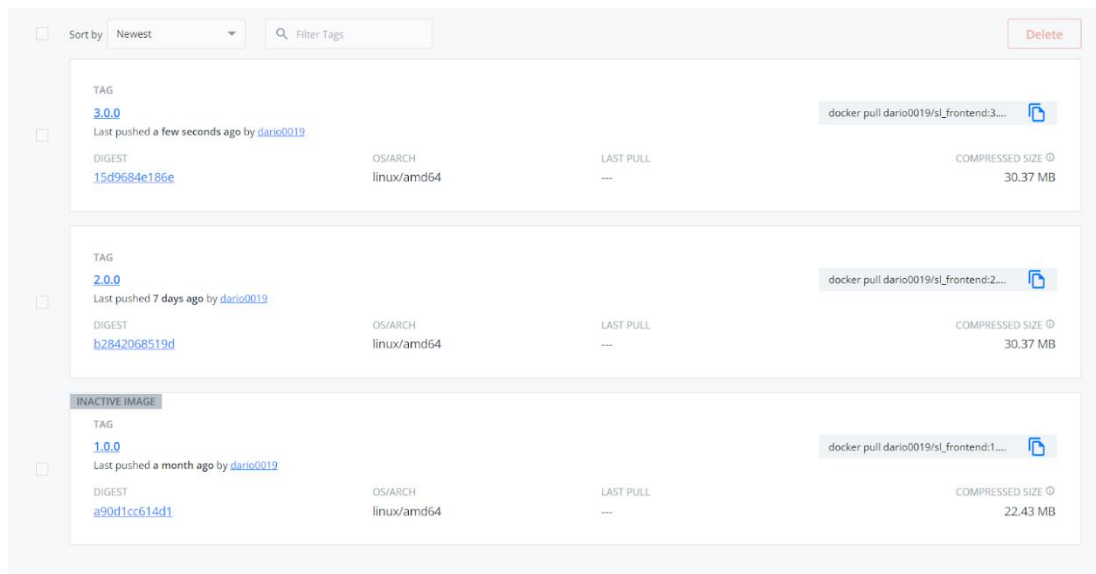
Listado de imágenes docker en docker-hub aplicacion sl-back-end

The screenshot shows the 'Advanced Image Management' interface on Docker Hub. At the top, there is a header with an information icon, the text 'Advanced Image Management', and a 'View preview' link. Below the header, there is a 'Sort by' dropdown menu set to 'Newest' and a 'Filter Tags' search box. A 'Delete' button is located on the right side. The main content area displays a list of Docker images. The first image is tagged '2.0.0', pushed 4 minutes ago by 'dario0019', with a digest of 'ff28f3edb8df', OS/ARCH of 'linux/amd64', and a compressed size of 70.09 MB. The second image is tagged '1.0.0', pushed 2 months ago by 'dario0019', with a digest of '71ee22e09b30', OS/ARCH of 'linux/amd64', and a compressed size of 124.36 MB. Both images are marked as 'INACTIVE IMAGE'.

TAG	DIGEST	OS/ARCH	LAST PULL	COMPRESSED SIZE
<a href="#">2.0.0</a> Last pushed 4 minutes ago by <a href="#">dario0019</a>	<a href="#">ff28f3edb8df</a>	linux/amd64	---	70.09 MB
<b>INACTIVE IMAGE</b>				
<a href="#">1.0.0</a> Last pushed 2 months ago by <a href="#">dario0019</a>	<a href="#">71ee22e09b30</a>	linux/amd64	---	124.36 MB

**Figura 39**

Listado de imágenes docker en docker-hub aplicacion sl-frontend



**Fase de Despliegue.** En la fase de despliegue se da inicio al proceso de integración continua donde se utilizan las imágenes previamente construidas. En esta fase se realizará el despliegue de las aplicaciones con ayuda de Jenkins junto con archivos yml para poder descargar y desplegar las imágenes en contenedores docker.

Para el despliegue, Jenkins se conectará al servidor remoto vía SSH para y copiará los archivos yml que se encuentran dentro del proyecto para poder hacer el despliegue en el servidor de producción o desarrollo.

**Figura 40**

Archivo .yaml para la descarga y despliegue de la aplicación sl-back-end

```
version: '3'
services:
  sl_backend:
    container_name: sl_backend
    image: usuarioDocker/sl_backend:${VERSION}
    ports:
      - "8082:8080"
      - "8088:15432"
      - "587:587"
```

## Figura 41

*Archivo .yml para la descarga y despliegue de la aplicación sl-frontend*

```
version: '3'
services:
  sl_frontend:
    container_name: sl_frontend
    image: usuarioDocker/sl_frontend:${VERSION}
    ports:
      - "8080:80"
      - "8084:8082"
```

*Nota.* Antes de ejecutar los archivos .yml de la figura 40 y 41 debemos limpiar los contenedores y las imágenes docker con los comandos de la figura 42.

## Figura 42

*Archivo .sh para limpiar los contenedores e imágenes docker*

```
docker rm sl_backend sl_frontend
docker rmi usuarioDocker/sl_backend usuarioDocker/sl_frontend
```

*Nota.* En el caso de que no existan todavía los archivos no sucederá ningún error y si podremos continuar con la fase de despliegue.

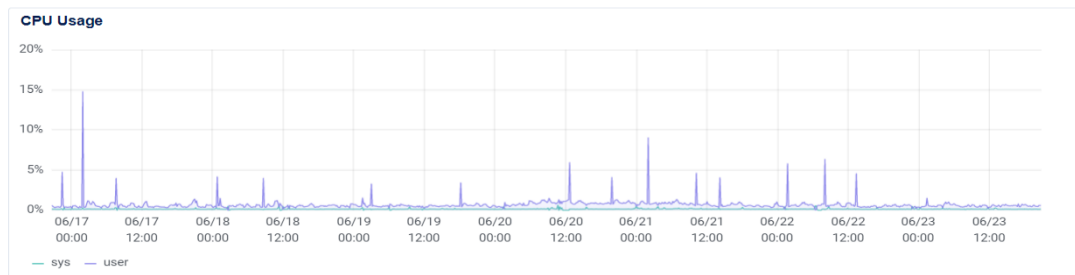
**Fase de Monitoreo.** La última fase del ciclo de vida de DevOps es aplicada durante todo el ciclo de vida y consiste en el monitoreo de los recursos ocupados por los servidores desplegados, ya sea el uso de memoria, uso de almacenamiento y el uso de la red.

Para esta etapa existen múltiples herramientas para monitorear los servicios, para el proyecto utilizaremos la herramienta que nos provee el propia digital ocean. En las figuras 44 y 45 se mostrarán a más detalle los consumos de memoria y conexiones que se realizan a los servidores.

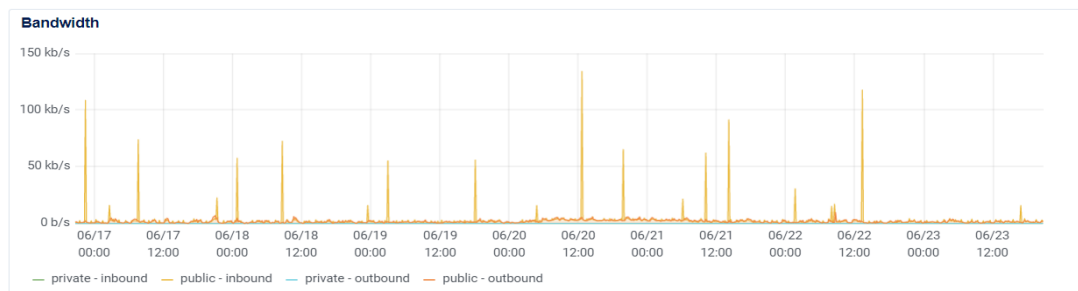


**Figura 43**

*Monitoreo del uso de memoria (Digital Ocean)*

**Figura 44**

*Monitoreo del de ancho de banda (Digital Ocean)*



## Capítulo IV

### Análisis de Resultados

#### Aceptación del Sistema

La aceptación del sistema se puede medir en diferentes niveles de satisfacción del usuario, como por ejemplo al usar la aplicación web, es decir, como se siente utilizando el sistema, la velocidad a la que aprende interactuar con el mismo y si necesitara alguna capacitación o soporte técnico.

Con lo mencionado anteriormente para evaluar la usabilidad del presente proyecto se utilizará el test Sistema de Escalas de Usabilidad o SUS, por sus siglas en inglés (System Usability Scale). El test tiene definido la escala, así como las preguntas que se evalúan, las cuales se pueden observar en la tabla 14.

Los criterios de evaluación de la escalada SUS son los siguientes

1. Totalmente de desacuerdo
2. En desacuerdo
3. Neutro
4. De acuerdo
5. Totalmente de acuerdo

Los resultados de la ejecución del test SUS son calculados mediante los argumentos:

- Preguntas impares se les resta 1.
- El resultado de las preguntas pares será el resultado de restar 5 menos el valor de la respuesta.
- Sumar el valor obtenido de cada respuesta y multiplicar por 2.5
- La nota máxima a obtener es de 100.

**Tabla 14***Encuesta SUS*

<b>No</b>	<b>Pregunta</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>
1	Cree usted que le gustaría utilizar este sistema con frecuencia.					
2	El sistema me pareció innecesariamente complejo.					
3	Me pareció que el sistema era fácil de usar.					
4	Creó que necesitaría el apoyo de un técnico para poder utilizar este sistema.					
5	Me pareció que las distintas funciones de este sistema estaban bien integradas.					
6	Pensó que había demasiada inconsistencia en este sistema.					
7	Imaginó que la mayoría de la gente aprendería a utilizar este sistema muy rápidamente.					
8	Encontró el sistema muy complicado de usar.					
9	Se siente muy seguro usando el sistema.					
10	Necesitó aprender muchas cosas antes de poder empezar a trabajar con este sistema.					

El presente instrumento de evaluación fue aplicado a 10 pacientes del SL laboratorio clínico y bacteriológico, obteniendo los siguientes resultados representados en la tabla 15.

**Tabla 15***Resultados del test SUS*

<b>Número de encuestas</b>	<b>Resultado</b>
1	80
2	72.5
3	65
4	57.5
5	97.5
6	67.5
7	77.5
8	80
9	65
10	82.5
<b>Promedio</b>	<b>74.5</b>

Como se observa en la tabla 15, el promedio de los resultados al aplicar el test es de 74.5, teniendo en cuenta que los lineamientos del test SUS el valor máximo es 100, afirmamos que la usabilidad del proyecto es aceptable.

### **Recolección de datos**

Los datos que se plantean para la validación de la hipótesis fueron obtenidos de dos encuestas realizadas a 25 pacientes del laboratorio clínico y bacteriológico SL, para lo cual la encuesta se aplicó cuando aún no tenían conocimiento alguno del aplicativo y posterior al conocimiento del aplicativo.

Las preguntas de la encuesta fueron las siguientes:

1. ¿Qué tiempo se demora en realizar los exámenes “Desde que sale de su hogar hacia el laboratorio” (poner respuesta en minutos)?
2. ¿Tiene un historial de sus exámenes online?
  - Si
  - No
3. ¿Cuál es la forma en la que retira sus exámenes clínicos?
  - Online
  - Presencial
4. ¿Cuál es la forma en que realiza la reserva de un examen clínico?
  - Online
  - Telefónica
  - Presencial
5. ¿Qué tan satisfecho está con el servicio de agendamiento de citas?
  - Excelente
  - Muy bueno
  - Bueno

- Regular
- Malo

Los resultados de la encuesta antes del aplicativo son:

### Pregunta 1

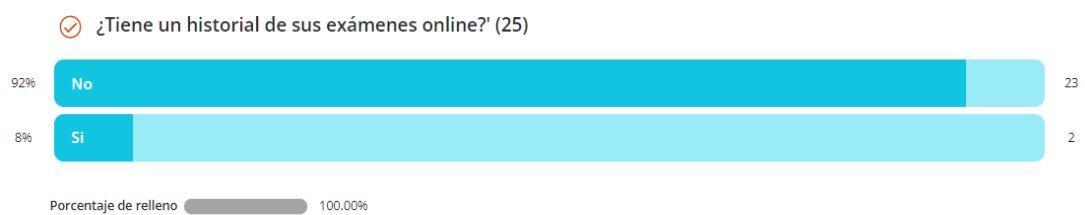
Cuando los pacientes no tenían un conocimiento previo del sistema, existe un promedio de 102 minutos con 20 segundos que se demora el paciente en realizar una reserva del examen.

### Pregunta 2

En la figura 45, un 92% de los encuestados no tienen un historial de sus exámenes online, mientras que un 8% si tienen historial de sus exámenes online (“En gmail o drive”).

### Figura 45

*Segunda pregunta de la encuesta*

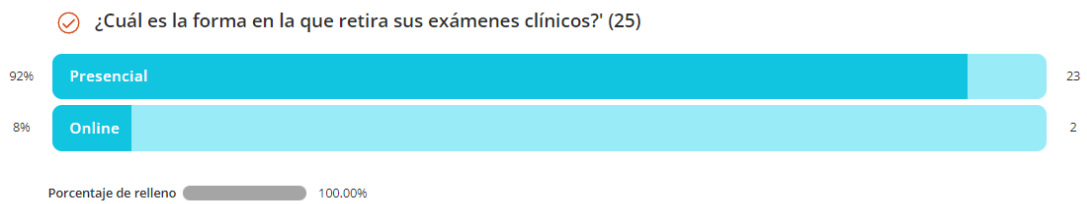


### Pregunta 3

La figura 46 muestra que un 92% de los encuestados retiran sus exámenes de manera presencial, mientras que un 8% retiran los exámenes de manera online (“Mediante Gmail u otros canales de envío de documentos”).

## Figura 46

### Tercera pregunta de la encuesta

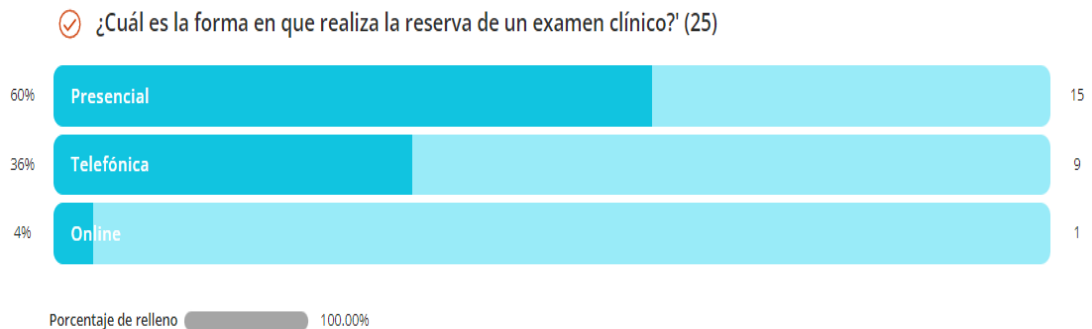


## Pregunta 4

En la figura 47, un 60% de los encuestados realizan la reserva del examen de manera presencial, mientras que el 36% lo realiza de manera telefónica y un 4% online (“Mediante redes sociales”).

## Figura 47

### Cuarta pregunta de la encuesta

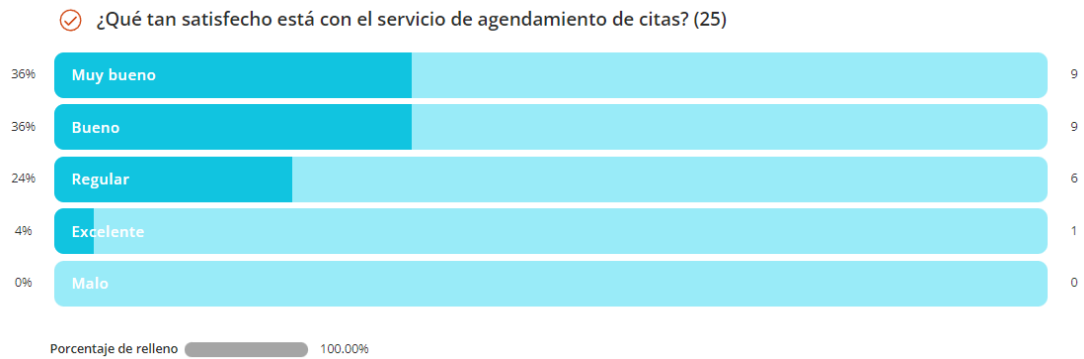


## Pregunta 5

La figura 48 muestra que un 36% de los encuestados comentan servicio de agendamiento de citas es muy bueno, el 36% de los encuestados comentan que el servicio es bueno, el 24 % de los encuestados comentan que el servicio es regular y el 4% nos comenta que es excelente.

## Figura 48

### Quinta pregunta de la encuesta



Los resultados de la encuesta después del aplicativo son:

### Pregunta 1

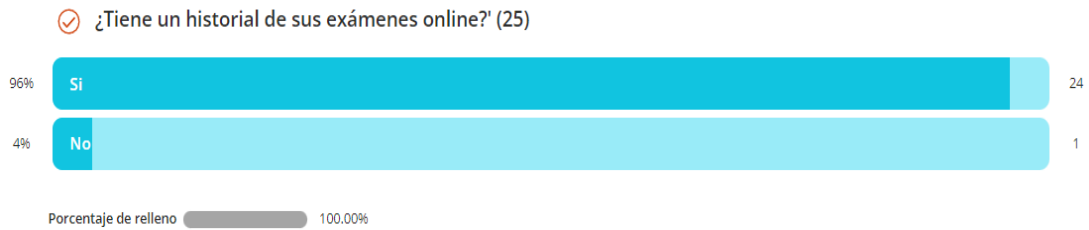
Cuando los pacientes no tenían un conocimiento previo del sistema, existe un promedio de 9 minutos con 16 segundos que se demora el paciente en realizar una reserva del examen.

### Pregunta 2

En la figura 49, un 95% de los encuestados si tienen un historial de sus exámenes online, mientras que un 5% si tienen historial de sus exámenes online (“aplicativo web”).

## Figura 49

*Segunda pregunta de la encuesta con el aplicativo*

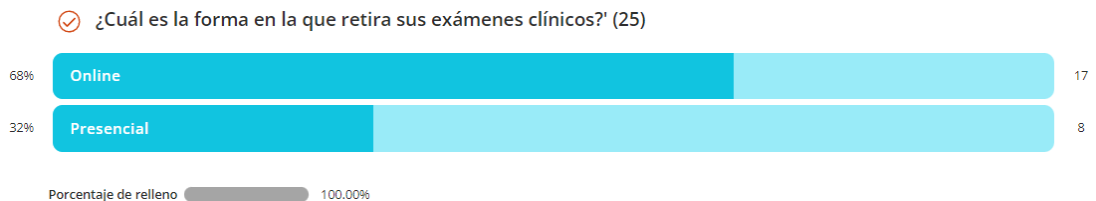


### Pregunta 3

La figura 50 muestra que un 32% de los encuestados retiran sus exámenes de manera presencial, mientras que un 68% retiran los exámenes de manera online (“aplicativo web”).

## Figura 50

*Tercera pregunta de la encuesta con el aplicativo*



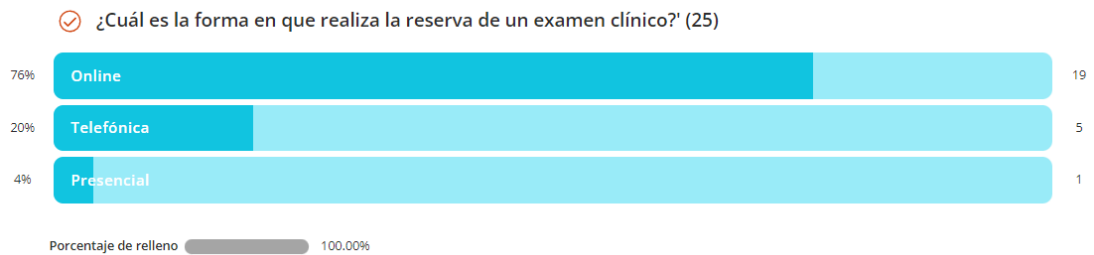
### Pregunta 4

En la figura 51, un 4% realizan la reserva de un examen de manera presencial, mientras que el 20% realizan la reserva de un examen de manera telefónica y el 70% lo realizan online (aplicativo web).



## Figura 51

*Cuarta pregunta de la encuesta con el aplicativo*

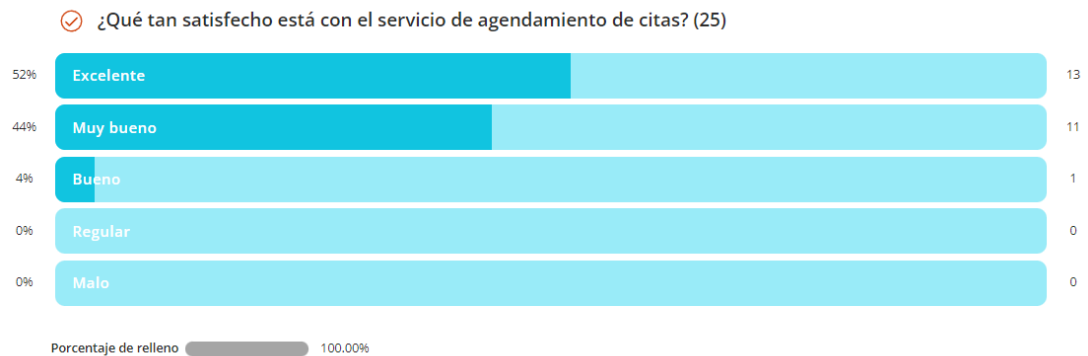


## Pregunta 5

En la figura 52, un 44% dicen que el servicio de agendamiento de citas es muy bueno, el 4% dicen que el servicio de agenda

## Figura 52

*Quinta pregunta de la encuesta con el aplicativo*



## Análisis de resultados

La hipótesis del presente proyecto es: ¿Si se desarrolla un sistema web utilizando DEVOPS junto a la herramienta JENKINS, entonces se mejora la gestión de reservas y consultas en línea de los servicios que ofrece SL Laboratorio Clínico y Bacteriológico?

La validación de la hipótesis se realizó mediante la distribución t de Student, ya que nos permite evaluar el tamaño de la población menor a 30.

Es necesario definir la hipótesis nula y la alternativa:

Hipótesis nula (Ho): ¿Si se desarrolla un sistema web utilizando DEVOPS junto a la herramienta JENKINS, entonces no se mejora la gestión de reservas y consultas en línea de los servicios que ofrece SL Laboratorio Clínico y Bacteriológico??

Hipótesis alternativa (H1): ¿Si se desarrolla un sistema web utilizando DEVOPS junto a la herramienta JENKINS, entonces se mejora la gestión de reservas y consultas en línea de los servicios que ofrece SL Laboratorio Clínico y Bacteriológico??

En la tabla 16 se observa el cálculo de las medias de los tiempos que se tardan los pacientes en realizar una cotización con el aplicativo y son el mismo.

**Tabla 16**

*Cálculo de la media de la diferencia de tiempos*

	<i>Sin en aplicativo</i>	<i>Con el aplicativo</i>	<i>Diferencia entre medias</i>
<b><i>Reserva del examen (Promedio)</i></b>	102.20	9.16	93.04

Para el cálculo de los grados de libertad se realiza con la fórmula  $n - 1$  obteniendo así el valor de 24, para el nivel de confianza es de 5%, entonces el valor crítico de t es 1.71088, el cual representa el límite si se rechaza o se acepta cualquiera de las hipótesis:

Si  $t \leq$  entonces no se rechaza Ho

Si  $t >$  entonces se rechaza Ho

Se procede a la aplicación de la ecuación

$$t = \frac{d}{S_d/\sqrt{n}}$$

Donde:

t: Estadístico t calculado

$\bar{d}$  : Promedio de las diferencias

$S_d$  : Desviación estándar de las diferencias

n: Tamaño de la muestra

$$t = \frac{93.04}{43.0324296/\sqrt{25}}$$

$$t = 10,81045$$

### **Discusión de los resultados**

El valor obtenido mediante el análisis de los datos es 10.81045 el cual es mayor a 1.71088, por lo tanto, la hipótesis alternativa es aceptada y podemos afirmar que gracias al desarrollo del aplicativo web se mejora la gestión de reservas y consultas en línea de los servicios que ofrece SL Laboratorio Clínico y Bacteriológico.

## Capítulo V

### Conclusiones y Recomendaciones

#### Conclusiones

- Se cumplió con el objetivo de desarrollar un sistema web con metodología DEVOPS utilizando la herramienta JENKINS, para mejorar la gestión de reservas y consultas en línea de los servicios que ofrece SL Laboratorio Clínico y Bacteriológico.
- El desarrollo del marco teórico permitió recopilar información y adquirir conocimientos sobre la importancia de un laboratorio clínico, el proceso de creación de reservas y resultados de los exámenes médicos y de las técnicas, enfoques y herramientas recomendadas para el desarrollo de un sistema web.
- El uso de la metodología DevOps permitió tener un desarrollo seguro y eficiente del sistema web, cumpliendo su ciclo de vida y llevando a cabo una correcta planificación y documentación, junto con el uso de herramientas de seguimiento y versionamiento, indicando así al usuario final el proceso del desarrollo del sistema web.
- El uso de la modelación C4 resultó de gran importancia para describir la arquitectura del sistema web, este consiste en un conjunto de diagramas jerárquicos que ayudó a la comprensión, construcción y funcionamiento en los diferentes niveles de abstracción, cada uno útil para el entendimiento de cada audiencia.
- El desarrollo e implementación del sistema web resolvió todas las necesidades que tenían los usuarios en lo que corresponde las reservas y consultas en línea de los servicios que ofrece SL Laboratorio Clínico y Bacteriológico.
- Se determinó que el uso del sistema web permite al personal del laboratorio tener una administración más eficiente, rápida y confiable, además de optimizar el proceso tradicional que estaba anteriormente de preguntar, reservar o recoger los resultados de los exámenes clínicos de forma presencial en el laboratorio, siendo la usabilidad

el factor más importante, especialmente para mejorar el tiempo de atención y entrega de resultados a los pacientes o clientes.

- Con la implementación de sistema web se mejoró la atención a los pacientes, puesto que estos pueden consultar, reservar o ver sus resultados en el momento que deseen, además pueden descargarlos en caso de que requiera imprimirlos para mostrarlos en forma física a su doctor de cabecera.
- El sistema web se desplegó con éxito cumpliendo con el ciclo de vida DevOps, logrando obtener una excelente acogida por los usuarios externos e internos del Laboratorio clínico tanto en su funcionalidad, apariencia y seguridad.
- Gracias a las encuestas elaboradas en la etapa inicial y final, se logró comprobar la gran aceptación y acogida por parte del personal y clientes de SL Laboratorio Clínico & Bacteriológico.

### **Recomendaciones**

- Se recomienda el uso de la metodología DevOps debido a que este aporta la eficiencia y la flexibilidad, además acaba con las barreras entre los profesionales del desarrollo y de las operaciones para así tener un producto de software de mayor calidad y fiabilidad.
- Se recomienda el uso de SpringBoot debido a que proporciona una estructura adecuada para el desarrollo de backend permitiendo ahorrar tiempo, costo y genera código de alta calidad.
- Se recomienda el uso de angular, especialmente para el desarrollo de frontend, debido a su facilidad de uso, estabilidad, compatibilidad y extensa documentación.
- Se recomienda la utilización de un repositorio en la nube para el tema de versionamiento de código, ya que ayudara a integrar de una manera más fácil y óptima todo el flujo de integración continua y entrega continua.

- Se recomienda que, al utilizar Jenkins tener en cuenta que ya existen muchos plugins que podemos instalarlos y utilizarlos, así como la extensa documentación que existe, siempre consultar en la documentación oficial para no instalar plugins innecesarios o que a su vez consuma espacio en el servidor de Jenkins.
- Se recomienda que, al conectar Jenkins con un servidor exterior, realizarlo vía una conexión ssh, ya que es más segura y nos genera un archivo .pem que solo un cliente puede tener y así no compromete la conexión del cliente (Jenkins) con el servidor de producción o desarrollo.

## Bibliografía

- AcensTechnologies*. (2020). Obtenido de AcensTechnologies: <https://www.acens.com/wp-content/images/2015/07/git-github-wp-acens.pdf>
- AcensTecnologies. (2018). *AcensTecnologies*. Obtenido de AcensTecnologies: <https://www.acens.com/wp-content/images/2016/10/bootstrap-framework-acens-wp.pdf>
- Aghael , S., Nematbakhsh, M., & Khosravi Farsani, H. (2012). EVOLUTION OF THE WORLD WIDE WEB : FROM WEB 1.0 TO WEB 4.0. *International Journal of Web & Semantic Technology*. doi:10.5121/IJWEST.2012.3101
- Aguilera Gámiz, C., Aznar Martín, J., Durán Serantes, M., Gascón Luna, F., de Haro Muñoz, T., Hortas Nieto, L., . . . Ruiz Ruda, E. (2004). *Proceso de Soporte de Laboratorios Clínicos*. Sevilla: Consejería de Salud.
- am Desing*. (04 de 2021). Obtenido de am Desing: <https://www.am-design.es/que-es-bootstrap-y-para-que-sirve#:~:text=Qu%C3%A9%20es%20Bootstrap%20y%20para%20qu%C3%A9%20sirve%20Bootstrap,nos%20permiten%20desarrollar%20interfaces%20HTML%20atractivas%20y%20responsive.>
- AM, T. S. (2015). *CRITERIOS PARA UNA PRUEBA DIAGNÓSTICA*. Obtenido de <https://www.medigraphic.com/anuncios/pdfs/terres/Cap2.pdf>
- Ariño, M. L. (2018). *HISTORIA DE LAS WEB*. Universidad Marcelino Champagnat.
- ATLASSIAN. (2020). *ATLASSIAN*. Obtenido de ATLASSIAN: <https://www.atlassian.com/es/software/jira/guides/use-cases/what-is-jira-used-for#Jira-for-requirements-&-test-case-management>
- Avi. (30 de 07 de 2019). *GEEKFLARE*. Obtenido de GEEKFLARE: <https://geekflare.com/es/dockerfile-tutorial/>
- AWS. (2022). Obtenido de AWS: <https://aws.amazon.com/es/docker/>

B., G. (23 de 02 de 2022). *HOSTINGER*. Obtenido de

<https://www.hostinger.es/tutoriales/que-es-nginx>

Blancarte, O. (17 de 07 de 2018). *Software Architect*. Obtenido de Software Architect:

<https://www.oscarblancarteblog.com/2018/07/17/spring-boot-relacion-los-microservicios/>

Brown, S. (11 de 2018). *c4model*. (C. C. License., Editor) Obtenido de c4model:

<https://c4model.com/>

Cacique, C. (2010). *¿Qué es Spring Security?* España.

campusMVP. (07 de 07 de 2021). *campusMVP*. Obtenido de campusMVP:

<https://www.campusmvp.es/recursos/post/que-es-angular-para-que-sirve-y-por-que-aprenderlo.aspx#:~:text=1%20Qu%C3%A9%20es%20Angular.%20Angular%20es%20un%20framework,5%20La%20mejor%20forma%20de%20aprender%20Angular.%20>

Campuzano Maya, G. (2011). *Valores críticos en el laboratorio clínico: de la teoría a la práctica*. Medellín: Edimeco.

contributors, S. O. (2020). *Aprendizaje Junit*. eBook. Obtenido de

<https://riptutorial.com/Download/junit-es.pdf>

Corino López, C. (2017). *Evolución de la web 2.0 a la 3.0 y su impacto en la empresa*. España.

CORRALES ÁLVAREZ, M. (2015). Concepto asociados. En C. Á. Mario, *Selección, elaboración, adaptación y utilización de materiales, medios y recursos didácticos en formación profesional para el empleo* (pág. 153). España: Paraninfo,S.A.

Crawford, A. (02 de 10 de 2019). *IBM*. Obtenido de IBM: <https://www.ibm.com/cloud/learn/devops-a-complete-guide>

De Seta, L. (13 de 11 de 2008). *Introducción a los servicios web RESTful*. Obtenido de <https://dosideas.com/noticias/java/314-introduccion-a-los-servicios-web-restful>



*DesdeLinux*. (12 de 2020). Obtenido de DesdeLinux: <https://blog.desdelinux.net/docker-hub-aprendiendo-tecnologia-docker/>

*Digital Guide - IONOS*. (05 de 10 de 2020). Obtenido de Digital Guide - IONOS: <https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/json-web-token-jwt/>

*Digital Guide IONOS*. (04 de 09 de 2019). Obtenido de Digital Guide IONOS: <https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/spring-boot-tutorial/>

Ebert, C., Gallardo, G., Hernantes, J., & Serrano, N. (Junio de 2016). *DevOps*. Obtenido de IEEEExplore: <https://ieeexplore.ieee.org/document/7458761>

Élez Villamarín, A. (2019). *Introducción a DevOps para la mejora de los procesos de desarrollo con herramientas Open Source*. España: 3.0 España de Creative Commons.

Fuchs, C., Hofkirchner, W., Schafranek, M., Raffl, C., Sandoval, M., & Bichler, R. (2010). *Theoretical Foundations of the Web: Cognition, Communication, and Co-Operation. Towards an Understanding of Web 1.0, 2.0, 3.0. Future Internet*.

García , M., & González, J. (2019). *Apache Maven*. Ibarra: Universidad Tecnológica Nacional. Obtenido de <http://www.profmatisgarcia.com.ar/uploads/tutoriales/Anexo1-Maven.pdf>

García Aretio , L. (10 de 2007). *¿Web 2.0 vs Web 1.0?* México : BENED.

García Peñalvo, F. J., García Holgado, A., & Vázquez Ingelmo, A. (2020 - 2021). *MODELO C4*. Universidad de Salamanca, Informática y Automática. Salamanca: GRIAL. Obtenido de <https://repositorio.grial.eu/bitstream/grial/2482/1/C4%20model.pdf>

Gasteiz, V. (2014). *Manual de usuario JUnit*. EJIE S.A. doi:Posta-kutxatila / Apartado: 809

Gautam , J., Girish, K., Katu, Pablo, Priyanshu, S., Peixoto, R., & Venkat, S. (2018). *APRENDIZAJE jenkins*. Stack OverFlow contributors. Obtenido de <https://riptutorial.com/Download/jenkins-es.pdf>

GitHub. (2021). *GitHub: The world's leading software development platform*. Recuperado el 03 de 2021, de <https://github.com>

GitHub. (12 de 2021). *Tutoriales en PDF*. Obtenido de Tutoriales en PDF:

<https://tutorialesenpdf.com/github/>

Gordadze, I. (2021). *Seguridad de Spring con JWT para LA API REST*. Developers.

Obtenido de <https://www.toptal.com/spring/spring-security-tutorial>

Guerrero, J., Certuche Díaz, S., Zúñiga, K., & Pardo Calvache, C. (2019). What is there about DevOps? Preliminary Findings from a Systematic Mapping Study. *XIV Jornadas Iberoamericanas de Ingeniería de Software e Ingeniería del Conocimiento (JIISIC'2019)*, (pág. 55.70). Costa Rica.

Handa, U. (2021). *7 razones para usar Angular para sus aplicaciones web en 2022*.

Obtenido de <https://cynoteck.com/es/blog-post/reasons-to-use-angular-for-your-web-app/>

Haro, E., Guarda, T., Ninahualpa Quiña, G., & Zambrano Peñaherrera, A. O. (2018).

Desarrollo backend para aplicaciones web, Servicios Web Restful: Node.js vs Spring Boot. Santa Elena - Ecuador: risti. Obtenido de <https://www.proquest.com/openview/a78cfaa62708fd24f38ac8d1025050eb/1?pq-origsite=gscholar&cbl=1006393>

Hernández Asunción, K. I. (2013). *De la Web 2.0 a la Web 3.0: antecedentes y consecuencias de la actitud e intención de uso de las redes sociales en la web semántica*. Madrid, España: Portal Universia S.A. ISSN: 1698-5117 .

Iriz Ricote, S. (04 de 06 de 2019). *Seguridad en el ciclo de vida de desarrollo del Software*.

Universidad Oberta de Catalunya, Seguridad Empresarial, España. Obtenido de UOC:

<http://openaccess.uoc.edu/webapps/o2/bitstream/10609/96129/6/sirizTFM0619memoria.pdf>

Java. (12 de 2021). *Java*. Obtenido de Java: <https://www.java.com/es/about/>

*Jenkins*. (2021). Obtenido de Jenkins: <https://www.jenkins.io/>

*Jetbrains*. (2021). Obtenido de JetBrains: <https://www.jetbrains.com/es-es/idea/>

- Jetbrains*. (2021). Obtenido de Jetbrains: <https://www.jetbrains.com/es-es/idea/features/#:~:text=Adem%C3%A1s%2C%20IntelliJ%20IDEA%20Ultimate%20puede,Jakarta%20EE%2C%20Micronaut%2C%20Quarkus%20y>
- Jhipster*. (2022). Obtenido de Jhipster: <https://www.jhipster.tech/>
- Jiménez Marco, G. (2016). *DevOps, la nueva tendencia en el desarrollo de sistemas TI, un caso práctico en el análisis de incidencias de software*. catalán: Everis.
- Labra Gayo, J. E. (2006). *Introducción a la Web 2.0*.
- Landázuri Ortiz, C. (2019). *Diseño de un modelo de gobernabilidad y gestión de TI para el área de desarrollo de proyectos de software de corporación favorita, basado en la metodología DevOps*. Universidad Internacional SEK , Quito. Obtenido de <https://repositorio.uisek.edu.ec/bitstream/123456789/3625/1/Tesis%20maestr%C3%ADa%20-%20Carlos%20Land%C3%A1zuri.pdf>
- LATAM, T. (27 de 02 de 2020). *TIVIT* . Obtenido de TIVIT : <https://latam.tivit.com/blog/ques-digital-ocean-serie-de-proveedores-de-cloud>
- Latorre , M. (2018). *HISTORIA DE LAS WEB, 1.0, 2.0, 3.0 y 4.0*. Perú: Universidad Marcelino Champagnat.
- Lima, A. (2020). *ACERVO LIMA*. Obtenido de [https://es.acervolima.com/uso-de-certbot-manualmente-para-certificados-ssl/#:~:text=Certbot%20es%20una%20herramienta%20de,Electronic%20Frontier%20Foundation\(EFF\)](https://es.acervolima.com/uso-de-certbot-manualmente-para-certificados-ssl/#:~:text=Certbot%20es%20una%20herramienta%20de,Electronic%20Frontier%20Foundation(EFF)).
- Lima, A. (2021). *ACERVO LIMA*. Obtenido de <https://es.acervolima.com/jhipster-java-hipster-una-plataforma-de-desarrollo-web-de-pila-completa-para-el-desarrollador-moderno/>
- Liquibase. (01 de 2022). *Liquibase*. Obtenido de <https://www.liquibase.com/>
- Lopez-Pellicer, F., Béjar, R., Latre, M., Nogueras-Iso, J., & Zarazaga-Soria, F. J. (2015). *GitHub como herramienta docente*. Universidad de Zaragoza, Depto. de Informática e Ing. de Sistemas. España: Universitat Oberta La Salle. doi:978-99920-70-10-9

- Macias, S. (21 de 03 de 2018). *enmilocalfunciona*. Obtenido de enmilocalfunciona:  
<https://enmilocalfunciona.io/construyendo-una-web-api-rest-segura-con-json-web-token-en-net-parte-i/>
- Martinez Ladrón de Guevara, J. (2015). *Fundamentos de programación en Java*. EME.  
doi:ISBN 978-84-96285-36-2
- Martinez, M. (12 de 2019). *Docker Compose*. Obtenido de SCRIBD:  
<https://es.scribd.com/document/474837000/DOCKER-COMPOSE-docx>
- Maven*. (26 de 07 de 2022). Obtenido de Maven: <https://maven.apache.org/what-is-maven.html>
- Maven Apache*. (26 de 07 de 2022). Obtenido de Maven Apache: <https://maven.apache.org/>
- Muradas, Y. (06 de 08 de 2020). *OpenWebinars*. Obtenido de OpenWebinars:  
<https://openwebinars.net/blog/que-es-jira/>
- Pai, M. V., & Asnika, S. (2014). RESTful Web Services. . *International Journal of Advanced Information Science and Technology (IJAIST)*, (págs. 46–50.).
- Partner, M. (2021). *Vsphost*. Obtenido de Vsphost: <https://vsphost.com/blog/que-es-jenkins/>
- Pérez Hoyos, L. (2018). *DevOps: IT Development in the Era of Digitalization*.  
UNIVERSIDAD DE VALLADOLID, INGENIERIAS INDUSTRIALES, España.  
Obtenido de <https://uvadoc.uva.es/bitstream/handle/10324/31469/TFG-I-994.pdf?sequence=1&isAllowed=y>
- PI, N. (10 de 01 de 2020). *APINFOX*. Obtenido de APINFOX: <https://apinfox.com/java/que-es-intellij-idea/>
- Rodríguez, X. (24 de 07 de 2019). *OpenWebinars*. Obtenido de OpenWebinars:  
<https://openwebinars.net/blog/que-es-dockerfile/>
- Román Martín, J. (30 de 01 de 2017). *EMEZETA*. Obtenido de  
<https://www.emezeta.com/articulos/digitalocean-guia-para-alojar-tu-web-en-un-servidor-vps>
- Rosique, R. (2016). *Un Asomo a la Educación y Web 1.0, 2.0 y 3.0*. Chile.

Sandrigo, C. (27 de 03 de 2017). *FOLDER IT*. Obtenido de FOLDER IT:

<https://folderit.net/es/blog/jhipster-spring-boot-angular-minutos-es/>

Sentrio. (16 de 09 de 2021). *Sentrio*. Obtenido de Sentrio: <https://sentrio.io/blog/que-es-jenkins/>

*Sesitdigital*. (21 de 11 de 2020). Obtenido de Sesitdigital:

<https://sesitdigital.com/arquitectura-de-software-y-cultura-devops/>

*Spring*. (2021). Obtenido de Spring: <https://spring.io/projects/spring-security#overview>

Tarisfeño, V. (26 de 10 de 2020). *Kranio*. Obtenido de Kranio:

<https://www.kranio.io/blog/introduccion-al-ciclo-de-vida-de-devops>

Toledano López, O., López del Castillo, D., & Vasquez Sánchez, A. (2018). Services Layer for the platform of processing of massive educational data of the Faculty 4 of the University of the Informatics Sciences. *Informatica 2018*. La Habana. Obtenido de [https://www.researchgate.net/publication/323813065\\_Capa\\_de\\_servicios\\_para\\_la\\_plataforma\\_de\\_procesamiento\\_de\\_datos\\_educativos\\_masivos\\_de\\_la\\_Facultad\\_4\\_de\\_la\\_Universidad\\_de\\_las\\_Ciencias\\_InformaticasServices\\_Layer\\_for\\_the\\_platform\\_of\\_processing\\_of\\_massiv](https://www.researchgate.net/publication/323813065_Capa_de_servicios_para_la_plataforma_de_procesamiento_de_datos_educativos_masivos_de_la_Facultad_4_de_la_Universidad_de_las_Ciencias_InformaticasServices_Layer_for_the_platform_of_processing_of_massiv)

tutorialspoint. (2020). *Seguridad de resorte - JWT*. Recuperado el 03 de 2022, de tutorialspoint:

[https://www.tutorialspoint.com/spring\\_security/spring\\_security\\_with\\_jwt.htm](https://www.tutorialspoint.com/spring_security/spring_security_with_jwt.htm)

Villacampa, Ó. (29 de 01 de 2021). *ondho*. Obtenido de ondho: <https://www.ondho.com/que-es-docker-para-que-sirve/>

Vivanco, J. (05 de 06 de 2019). *Medium*. Obtenido de Medium:

<https://medium.com/@javiervivanco/el-modelo-c4-de-documentaci%C3%B3n-para-la-arquitectura-de-software-424704528390>

Vivas, J. (20 de 12 de 2017). *The Ninja CTO*. Obtenido de The Ninja CTO:

<https://www.theninjactoxyz/Liquibase-Control-Versiones-DataBase-Refactoring/>

Yáñez , N. J. (02 de 11 de 2021). *CÓDIGO XULES*. Obtenido de CÓDIGO XULES:

<https://codigoxules.org/que-es-docker-compose/>

**Anexos**