

ESCUELA POLITÉCNICA NACIONAL

**FACULTAD DE INGENIERÍA ELÉCTRICA Y
ELECTRÓNICA**

**ESQUEMA DE AUTOMATIZACIÓN PARA TECNOLOGÍAS DE LA
INFORMACIÓN**

**ANÁLISIS DE LAS PRÁCTICAS DEVOPS EN EL DESARROLLO DE
UN APLICATIVO MÓVIL PARA UNA EMPRESA**

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

ANDRÉS LUIS QUISHPE CASTRO

andres.quishpe@epn.edu.ec

DIRECTOR: XAVIER ALEXANDER CALDERÓN HINOJOSA

xavier.calderon@epn.edu.ec

DMQ, marzo 2023

CERTIFICACIONES

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

ANDRÉS LUIS QUISHPE CASTRO

Certifico que el presente trabajo de integración curricular fue desarrollado por ANDRÉS LUIS QUISHPE CASTRO, bajo mi supervisión.

XAVIER ALEXANDER CALDERÓN HINOJOSA
DIRECTOR

DECLARACIÓN DE AUTORÍA

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

ANDRÉS LUIS QUISHPE CASTRO

XAVIER ALEXANDER CALDERÓN HINOJOSA

DEDICATORIA

Se ha recorrido un gran camino lleno de alegrías y tristezas, personas que ya no están en mi vida y otras que llegaron para quedarse. Una gran alegría llegó a mi vida aportando esa felicidad que me motiva a seguir adelante y luchar cada día por seguir creciendo todo su amor se ve reflejado en una sonrisa, una travesura, un te amo y sobre todo un abrazo después de todo está aprendiendo y observando muchas cosas por primera vez.

Mi Martina

Te amo con todo mi corazón y eres mi fuerza y mi inspiración. Te agradezco todo lo que haces por mí y por nuestra familia. Siempre me has apoyado y me has ayudado a seguir adelante, incluso en los momentos más difíciles. Eres una luz brillante en mi vida y me haces sentir increíblemente afortunado. Gracias por ser la mejor esposa y madre para nuestra hija. Te amo.

Mi Marjury

Agradezco incondicionalmente a mi madre por su amor, apoyo y paciencia incondicional durante mi carrera universitaria. Siempre estuviste ahí para escucharme, incluso cuando las cosas se volvían difíciles. Con tu apoyo me ayudó a superar todas las dificultades que enfrenté para completar mis estudios. Muchas gracias por ser mi pilar en momentos difíciles y por motivarme a seguir adelante. Esta tesis es mi pequeño homenaje.

Mamá Marthita

AGRADECIMIENTO

Quiero agradecerles por todo su apoyo y dedicación durante todos estos años. Sus amables palabras, su ayuda y su compañía han significado mucho para mí. Siempre han estado ahí para escucharme, aconsejarme y apoyarme en todas mis decisiones. Estoy muy agradecida de tenerlas a mi lado a lo largo de mi viaje. ¡Muchas gracias!

ÍNDICE DE CONTENIDO

CERTIFICACIONES	I
DECLARACIÓN DE AUTORÍA	II
DEDICATORIA	III
AGRADECIMIENTO	IV
ÍNDICE DE CONTENIDO	V
ÍNDICE DE FIGURAS	VII
RESUMEN	IX
ABSTRACT	X
1. DESCRIPCIÓN DEL COMPONENTE DESARROLLADO	1
1.1 Objetivo general	2
1.2 Objetivos específicos	2
1.3 Alcance	2
1.4 Marco Teórico	3
Introducción a DevOps	3
1.5 Importancia de DevOps	4
1.6 Método DevOps	5
Integración Continua	5
Entrega Continua	6
Infraestructura como código	6
Microservicios	6
1.7 Herramientas DevOps	7
Git	7
Jenkins	8
Ansible	8
Doker	8
Kubernetes	9
1.8 Ventajas de DevOps	9
1.9 Desventajas de DevOps	10
1.10 Metodología Kanban	11
1.11 DevOps con Kanban	12
1.12 Android Studio	14
Ventajas	15
Desventajas	15

1.13	Lenguaje de Programación Kotlin	15
1.14	Node.js	16
1.15	Herramientas DevOps para el prototipo de Aplicativo Móvil	17
2.	METODOLOGÍA	19
2.1	Descripción Servicio Integrado de Seguridad ECU 911[24]	20
	Misión	21
	Visión	21
	Objetivos de Calidad [24]	21
	Objetivo Estratégico [24]	22
2.2	Organigrama Institucional [24]	22
2.3	Descripción de la Unidad de Tecnologías de Información	23
2.4	DevOps dentro del ECU911	24
2.5	Propuesta de herramientas DevOps para el ECU911	26
2.6	Desarrollo del aplicativo móvil para el ECU911	27
	Prototipo de aplicativo móvil	28
	Repositorio de Documentos	33
	Repositorio de versiones	34
	Nuevas versiones	36
	Contenedores	38
	Prototipo aplicativo móvil Eventos Relevantes	43
3.	RESULTADOS, CONCLUSIONES Y RECOMENDACIONES	47
3.1	Resultados del Uso de Herramientas DevOps	47
3.2	Conclusiones	50
3.3	Recomendaciones	51
4.	REFERENCIAS BIBLIOGRÁFICAS	53
5.	ANEXOS	55
	ANEXO I	56
	ANEXO II	57
	ANEXO III	58
	ANEXO IV	60

ÍNDICE DE FIGURAS

Figura 1.1 Integración Continua [5]	5
Figura 1.2 Infraestructura como código.[7]	6
Figura 1.3 Ejemplo del tablero de Kanban. [18]	12
Figura 1.4 Ejemplo de tablero principal	13
Figura. 1.5 Sintaxis de Kotlin. [22]	16
Figura 1.6 Principales ideas de los temas tratados	18
Figura 2.1 Organización Estructural Propuesta – Administración Nacional [24]	22
Figura 2.2 Organización Estructural – Administración Zonal [24]	23
Figura 2.3 Organización Estructural de Unidad de Tecnologías de Información	23
Figura 2.4. Herramientas DevOps	25
Figura 2.5 Disminuir el uso del aplicativo WhatsApp	28
Figura 2.6 Tablero Kanban	29
Figura 2.7 Tarea dentro del tablero de Kanban	30
Figura 2.8 Progreso de la tarea dentro del tablero Kanban	31
Figura 2.9 Tarea terminada en el tablero Kanban	32
Figura 2.10 Actualización de la tarea principal	32
Figura 2.11 Repositorio de documentos – Google Drive	33
Figura 2.12 Actas almacenadas en Google Drive	33
Figura 2.13 Herramienta GitHub, principales versiones almacenadas.	34
Figura 2.15 Repositorio creado en GitHub utilizado para el prototipo móvil.	35
Figura 2.16 Resumen de la información almacenada en GitHub.	36
Figura 2.17 Registro de versiones	37
Figura 2.18 Cargado de archivos modificados.	37
Figura 2.19 Registro de versiones de proyectos	38
Figura 2.20 Estructura de Docker utilizada en la Institución ECU911.	39
Figura 2.21Imagen de Android Studio almacenada en el repositorio de Docker	40
Figura 2.22 Entorno de Docker Desktop	41
Figura 2.23 Verificando que Docker este ejecutándose correctamente.	41
Figura 2.24 Resultado de ejecutar la línea de código antes comentada.	42
Figura 2.25 Estado de la imagen Android Studio	42

<i>Figura 2.25 Prototipo de aplicativo Móvil, layout de acceso versión 1.</i>	<u>43</u>
<i>Figura 2.26 Versión final del prototipo de aplicativo móvil.</i>	<u>44</u>
<i>Figura 2.27 Diagrama de acceso de usuarios</i>	<u>45</u>
<i>Figura 2.28 Código para el EditText del password.</i>	<u>46</u>
<i>Figura 3.1 Resultados de la primera pregunta.</i>	<u>47</u>
<i>Figura 3.2 Resultados de la segunda pregunta.</i>	<u>48</u>
<i>Figura 3.3 Resultado de la pregunta cuatro.</i>	<u>49</u>
<i>Figura 3.4 Resultados de la pregunta cinco.</i>	<u>49</u>

RESUMEN

Dentro de los ambientes de desarrollo de software se pueden utilizar varias herramientas que ayudan resolver problemas que se presentan durante el desarrollo del proyecto. Para el desarrollo de este proyecto se ha escogido a la Institución ECU911 donde se aplicarán herramientas DevOps durante el desarrollo del prototipo del aplicativo móvil. Durante el desarrollo del proyecto emplearemos la metodología ágil Kanban de esta manera organizaremos nuestro proyecto hasta obtener los resultados deseados.

En el Capítulo 1 encontraremos información referente a las Herramientas DevOps que existen, mostrando un breve resumen, así como también se tomará en cuenta las herramientas que pueden ser aplicadas durante el desarrollo del aplicativo móvil. Además, se habla de la metodología ágil Kanban y su relación con las Herramientas DevOps. También se cuenta con información referente a las ventajas de trabajar con DevOps y algunas desventajas que también pueden presentarse. También se detalla información referente al entorno de desarrollo Android Studio, así como también sobre el lenguaje de programación Kotlin.

El Capítulo 2 se centra en la metodología utilizada para el desarrollo del proyecto. Es decir, contaremos con información referente a la Institución ECU911, muestra la información actual de la Institución con respecto al uso de Herramientas DevOps, también se plantea la propuesta para que la Institución ECU911 haga uso de las Herramientas DevOps. Para el desarrollo del prototipo del aplicativo móvil se encuentran los procesos que se realizó para la obtención final del prototipo.

Para finalizar el proyecto en el Capítulo 3 son detallados los datos obtenidos después de realizar las pruebas sobre el prototipo de aplicativo móvil. Estos datos son analizados para obtener las conclusiones y recomendaciones que son de gran utilidad para continuar con el desarrollo del prototipo de aplicación móvil, llegando a obtener una versión con mejores características ya que se plantean las fallas y posibles mejoras que pueden ser aplicadas.

PALABRAS CLAVE: Kanban, DevOps, ECU911, calidad, Kotlin, Android.

ABSTRACT

Within the software development environment, several tools can be used to help solve problems that arise during the development of the project. For the development of this project we have chosen the ECU911 Institution where DevOps tools will be applied during the development of the prototype of the mobile application. During the development of the project we will use the agile Kanban methodology in this way we will organize our project until we obtain the desired results.

In Chapter 1 we will find information about the DevOps tools that exist, showing a brief summary, as well as the tools that can be applied during the development of the mobile application. In addition, the agile Kanban methodology and its relationship with the DevOps Tools is discussed. There is also information regarding the advantages of working with DevOps and some disadvantages that can also occur. Information regarding the Android Studio development environment is also detailed, as well as the Kotlin programming language.

Chapter 2 focuses on the methodology used for the development of the project. That is to say, we will have information regarding the ECU911 Institution, it shows the current information of the Institution regarding the use of DevOps Tools, also the proposal for the ECU911 Institution to make use of DevOps Tools is presented. For the development of the prototype of the mobile application, the processes that were carried out to obtain the final prototype are shown.

To finalize the project, Chapter 3 details the data obtained after testing the prototype of the mobile application. These data are analyzed to obtain the conclusions and recommendations that are very useful to continue with the development of the prototype of the mobile application, obtaining a version with better characteristics since the failures and possible improvements that can be applied are raised.

KEYWORDS: Kanban, DevOps, ECU911, quality, Kotlin, Android

1. DESCRIPCIÓN DEL COMPONENTE DESARROLLADO

Las prácticas DevOps se centran en la colaboración y la comunicación entre las diferentes partes interesadas en el desarrollo de un prototipo de aplicación móvil. Basándose en la idea de que todos los miembros del equipo deben estar involucrados en todas las etapas del proceso, desde el diseño hasta el lanzamiento y puesta en producción de la aplicación. Durante el desarrollo de software siempre se presentan problemas, para solventar los problemas se utiliza la metodología ágil Kanban, dentro del tablero Kanban los problemas son identificados y permiten que sean resueltos rápidamente, lo que reduce los tiempos muertos y los costos totales del proyecto. Se debe tomar en cuenta que las herramientas DevOps nos facilitan los entornos de producción y la facilidad de colaboración.

Este prototipo de aplicativo móvil contará con una conexión a una Base de Datos (BDD) donde serán registrados los usuarios y también la información de Eventos Relevantes, esta BDD se encuentra en el sistema PostgreSQL, de donde se obtendrá la información para ser mostrada en el prototipo de aplicativo móvil. El aplicativo solamente lo podrán utilizar usuarios previamente registrados de acuerdo con las políticas de la Institución ECU911 en la BDD esto debido al tipo de información que es de alta prioridad. Para validar que el usuario se encuentra autorizado para tener acceso a la información sus datos deben estar previamente registrados en la BDD, para el acceso se utiliza el correo electrónico y una clave asignada por parte del personal del ECU911 de acuerdo con las políticas de seguridad con las que cuente la institución, con los datos antes mencionados se concede el acceso a la información de Eventos Relevante.

Para el desarrollo de este aplicativo se utilizará el entorno de desarrollo Android Studio y el lenguaje de programación Kotlin, se trabaja con el tablero de Kanban donde se detallarán todas las actividades, las cuales serán etiquetadas de acuerdo con el estado en el que se encuentren “por hacer”, “en proceso” y “terminado” lo que permitirá identificar los puntos críticos donde se requiere dar prioridad para avanzar en el desarrollo del aplicativo. Con la finalidad de buscar mayor un desarrollo ágil se trabajará con herramientas DevOps durante todo el proceso de desarrollo.

Las herramientas DevOps cuentan con una gran variedad de opciones para facilitar el desarrollo de software y la comunicación con la parte operativa. Para llevar a cabo el proyecto es necesario realizar un estudio de las Herramientas DevOps y escoger las que mejor se adapten a las necesidades que deben ser solventadas dentro de la Institución ECU911, las herramientas seleccionadas facilitarán el desarrollo del prototipo de aplicativo móvil. Como parte fundamental del trabajo las herramientas DevOps estarán presentes en

todo el desarrollo del aplicativo. Considerando cada etapa en la que se pueda integrar. No se va a trabajar con todas las herramientas DevOps, se opta por las que se ajusten a las necesidades del proyecto.

El aplicativo móvil pretende brindar la seguridad adecuada para el manejo de la información a tratar. Las pruebas a realizar indicarán que tan exitoso resultó el uso de las herramientas DevOps durante el desarrollo del prototipo de aplicativo móvil y su forma de trabajo.

1.1 Objetivo general

Analizar las prácticas DevOps en el desarrollo de un aplicativo móvil para una empresa.

1.2 Objetivos específicos

1. Analizar las prácticas DevOps que pueden ser implementadas con la ayuda de Kanban.
2. Diseñar el prototipo móvil con la ayuda de DevOps.
3. Codificación - Implementación del prototipo.
4. Pruebas del prototipo.

1.3 Alcance

Permitir que las prácticas DevOps sean las adecuadas y se adapten al aplicativo móvil, su tarea será brindar información recopilada de las tareas en campo, permitiendo la lectura y escritura de la información.

Esto se llevará a cabo utilizando la metodología Kanban adecuando tanto al personal de campo como a los desarrolladores para conseguir un trabajo adecuado y funcional.

Considerando el trabajo se ha optado por las siguientes fases:

- a. Recopilación de información.
- b. Diseño del aplicativo móvil.
- c. Implementación del aplicativo móvil.

d. Evaluación y análisis de resultados.

1.4 Marco Teórico

Introducción a DevOps

No se trata de un tema nuevo, es un poco desconocido debido a que son pocos los usuarios que se especializan en esta rama, la palabra DevOps surge del acrónimo de las palabras en inglés Development y Operations, que surge como un medio para facilitar la comunicación entre los desarrolladores de software y el personal de operaciones de TI. [1], [2].

Con la premisa de mejorar la comunicación entre los desarrolladores y el personal de operaciones surgen las Herramientas DevOps. Considerando los avances actuales y la velocidad con que el mundo se encuentra cambiando es necesario adaptar las necesidades de las empresas a los nuevos cambios tecnológicos. Estos cambios implican aprender nuevas estrategias y actualizar nuestros conocimientos para hacer frente a los cambios.

Considerando estos cambios es necesario involucrar los equipos de desarrollo de software y de operaciones en un solo ambiente de tal forma que se cubran las necesidades reales. Es decir, cada departamento de trabajo debe aportar con lo que realmente es importante y no trabajar de forma independiente.

La mejor forma de obtener un buen trabajo es realizar el trabajo en equipo, esto implica que el equipo de desarrollo de software trabaje en conjunto con el equipo de operaciones, bajo este esquema surge DevOps utilizado para describir un conjunto de prácticas con la finalidad de automatizar procesos que han retrasado y no se llega a cumplir con el cronograma, generando problemas han momento de entregar el proyecto.

Los objetivos de DevOps son desarrollar una cultura de colaboración y comunicación entre los equipos de desarrollo de software y operaciones, lo que proporciona la capacidad de ofrecer continuamente valor a los clientes a través de ciclos de desarrollo más cortos.

Implementar las prácticas y herramientas DevOps durante todo el ciclo de desarrollo del software, implicando que el departamento de operaciones debe estar presente con sus observaciones y sugerencias para que sean cubiertas las necesidades más importantes.

Es necesario la creación de canales de integración y entrega continuas para facilitar la automatización en el proceso, se cuenta con algunas herramientas que ayudan durante el

desarrollo del trabajo. La automatización puede iniciar con las tareas de gestión para las cuales se puede utilizar herramientas como Puppet, Chef o Ansible. Además, es necesario configurar sistemas de monitorización y alertas para detectar los problemas en tiempo real y poder tomar las correcciones necesarias en corto tiempo. En lo posible es importante contar con sistemas de autorrecuperación en los casos que no requieran la intervención del equipo de operaciones o en fallas recurrentes.

Como una alternativa y considerando los cambios que se han desarrollado es posible trabajar con infraestructura en la nube en las que se puede utilizar herramientas como Amazon Web Services (AWS), Microsoft Azure o Google Cloud Platform, cubriendo la mayor parte de las necesidades que los proyectos requieren y finalizar con un trabajo eficiente.[3]

El objetivo de DevOps es permitir que los equipos de desarrolladores y de operaciones trabajen juntos de forma más eficaz y eficiente para acelerar el proceso de desarrollo de software, sin dejar de mantener altos niveles de calidad y seguridad e incluso mejorar los estándares.

DevOps es un conjunto de prácticas que ayuda a las organizaciones a entregar software de forma más rápida y fiable. Al automatizar y estandarizar la forma en que se construye, despliega y mantiene el software, los equipos de DevOps pueden entregar el software con mayor frecuencia y reducir las posibilidades de errores. [4]

1.5 Importancia de DevOps

El objetivo principal de DevOps es disminuir los tiempos de entrega y mejorar la calidad del software, lo que implica que los clientes llegan a obtener un producto que cubre sus necesidades en el tiempo deseado. Para conseguir estos objetivos deseados, los equipos de desarrollo y operaciones cuentan con una serie de herramientas y prácticas DevOps que incluyen procesos como:

- Entrega continua.
- Integración continua.
- Automatización.
- Microservicios.

Las prácticas DevOps son importantes para el entorno laboral ya que ayuda a entender cómo gestionar de forma eficiente y eficaz los proyectos de desarrollo de software.

También ayuda a los desarrolladores a aprender nuevos procesos y a mantener la comunicación con la parte de operaciones para cubrir las necesidades por las cuales se inicia el proyecto. Bajo este nuevo esquema de trabajo existen comunidades que facilitan el aprendizaje de las nuevas herramientas, así como información oficial directamente en las páginas web de cada herramienta DevOps. También se cuenta con herramientas DevOps que tienen un costo y otras son de uso gratuito.

1.6 Método DevOps

Hay muchos métodos diferentes que se pueden utilizar cuando se implementan las prácticas de DevOps. Algunos de los más populares son:

Integración Continua

Este método consiste en integrar continuamente los cambios de código en un repositorio compartido, donde se pueden construir y probar automáticamente. Esto ayuda a garantizar que los errores se descubran y se solucionen rápidamente, y que las nuevas características puedan ser entregadas a los usuarios más rápidamente. En la Figura 1.1 se observa el proceso que se debe seguir para la integración continua. [5]



Figura 1.1 Integración Continua [5]

Entrega Continua

Este enfoque lleva a la IC (Integración Continua) un paso a la vez para desplegar automáticamente los cambios de código, cuyo funcionamiento se ha verificado. Esto permite realizar lanzamientos más rápidos y fiables, ya que no es necesario esperar a la aprobación manual o a las pruebas antes de desplegar el nuevo código. [6]

Infraestructura como código

Esta práctica se refiere al uso de archivos de configuración y otras herramientas de automatización para gestionar la infraestructura tanto de servidores como de equipos de telecomunicaciones, algunas herramientas DevOps también permiten este tipo de configuración, como se puede observar en la Figura 1.2, este tipo de configuraciones son más eficientes en lugar de utilizar los métodos manuales tradicionales. Esto facilita el aprovisionamiento y la gestión de los servidores y reduce las posibilidades de que se produzcan errores humanos al realizar cambios en las configuraciones de los servidores.[7]

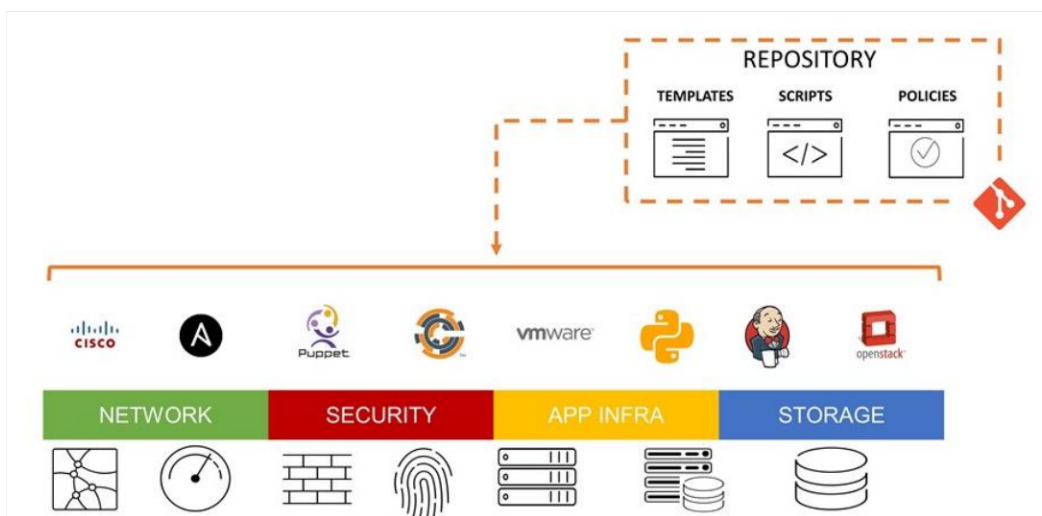


Figura 1.2 Infraestructura como código.[7]

Microservicios

Este estilo de arquitectura desglosa las aplicaciones en pequeños servicios independientes que pueden desplegarse y actualizarse independientemente unos de otros. Este enfoque modular facilita la realización de cambios en los servicios individuales sin afectar a toda la aplicación, y también simplifica el escalado horizontal de las aplicaciones añadiendo más instancias de servicios individuales según sea necesario.[8]

1.7 Herramientas DevOps

DevOps cuenta con un conjunto de herramientas que ayudan durante todo el proceso del desarrollo de software, teniendo herramientas que facilitan automatizar los procesos de desarrollo, prueba y despliegue de software.

El objetivo principal de las herramientas es mejorar la comunicación y la colaboración entre los desarrolladores y los equipos de operaciones, solventando el problema de comunicación entre los equipos de desarrolladores y de operaciones el seguimiento de los cambios en el código del software a desarrollar mejora los resultados y estándares de calidad que se desea obtener.

Algunas de las herramientas DevOps son consideradas más populares por su uso y facilidad de implementación. A continuación, se describen algunas de las herramientas DevOps:

Git

Un sistema de control de versiones distribuido que le ayuda a gestionar su base de código es un sistema de control de versiones distribuido, lo que implica que los usuarios pueden descargar el código fuente completo de un proyecto y tener acceso a todo su historial. Git fue creado por Linus Torvalds, el creador del kernel de Linux, y se ha convertido en la herramienta de control de versiones preferida por muchos desarrolladores debido a su flexibilidad y eficiencia.[9]

GitHub

Una plataforma de trabajo colaborativo que permite alojar proyectos utilizando el sistema de control de versiones. Utilizado en su mayor parte para almacenar código fuente, en los diferentes lenguajes de programación. Permite que el código fuente se encuentre al alcance del público, aunque también permite almacenar de forma privada teniendo algunas restricciones en cuanto a la colaboración de usuarios se refiere. Actualmente es la plataforma más importante de colaboración de código abierto. [10]

Jenkins

Es un servidor de integración y entrega continuas de código libre. Se trata de una herramienta que permite automatizar el proceso de compilación y prueba del código fuente, lo que resulta en un menor tiempo y esfuerzo necesarios para implementar nuevas funcionalidades. Se encuentra disponible en aplicativo web de código abierto, escrito en Java que se ejecuta en servidores remotos, y se puede utilizar tanto para monitorear los procesos de integración y entrega continuos, como para realizarlos. La herramienta está disponible bajo la licencia MIT (Massachusetts Institute of Technology).

Soporta herramientas de control de versiones, incluyendo AccuRev, CVS (Concurrent Versions System), Subversion, Git, Mercurial y ClearCase.[11]

Ansible

Herramienta de automatización de código abierto que se puede usar para implementar, configurar y administrar los sistemas y aplicaciones. Hace el trabajo repetitivo más fácil y le permite concentrarse en las tareas más importantes. Funciona mediante la transferencia de scripts llamados "playbooks" utiliza SSH para conectarse a los servidores remotos y no requiere ninguna instalación adicional en los servidores. Estos playbooks contienen las tareas que se deben realizar en cada nodo, así como las variables y la configuración necesaria para completar las tareas. También es fácil de aprender ya que utiliza un lenguaje de dominio sencillo llamado YAML (YAML Ain't Markup Language). [12]

Doker

Una plataforma de código abierto para crear, desplegar y ejecutar aplicaciones en contenedores. Los contenedores permiten que las aplicaciones se empaquetan junto con todas sus dependencias, lo que hace posible que funcionen de manera consistente en cualquier entorno. Docker proporciona una capa adicional de abstracción y automatización de virtualización en Linux. La plataforma Docker está construida sobre tecnologías de virtualización de nivel inferior como Linux Containers (LXC) y Kernel-based Virtual Machine (KVM), pero difiere de estos enfocándose en la facilidad de uso, la portabilidad y el desarrollador / operador "workflow".[13]

Kubernetes

Conocido por su sistema de código abierto para automatizar la implementación, el aislamiento y el escalado de aplicaciones en entornos virtualizados o en máquinas físicas. Fue diseñado por Google y actualmente es mantenido por la Cloud Native Computing Foundation. Permite que los desarrolladores creen aplicaciones que se ejecutan de forma consistente y segura en todas las plataformas, incluidas las nubes públicas y privadas. También facilita el escalado de aplicaciones para que puedan manejar el tráfico adicional sin problemas. [14]

Nagios

Es una herramienta DevOps de monitoreo, la cual permite realizar el monitoreo de servidores, routers, switches y otros dispositivos que permitan el trabajo con el protocolo SNMP (Simple Network Management Protocol). Permitiendo configurar diferentes tipos de alertas tanto a nivel de software y de hardware. Es una herramienta que genera tantas alertas puedan ser configuradas de los equipos que se encuentren registrados y trabajen dentro de la misma red. Esta herramienta DevOps es ideal para trabajar con software que gestione tickets para solventar los problemas que se presentan en la red. [15]

1.8 Ventajas de DevOps

Dependiendo el tipo de herramienta DevOps que se llegue a utilizar y la actividad que se está desarrollando, las ventajas pueden variar lo que implica, que se debe verificar cual es la herramienta DevOps más adecuada para cada proyecto a realizar. Como se ha observado existen herramientas DevOps que realizan el control de versiones, otras que permiten crear un entorno virtual, que puede ser ejecutado en cualquier equipo y otras que permiten el monitoreo de los equipos sobre los que se trabaja.

Sin embargo, se consideran las siguientes ventajas siendo comunes o similares en la mayor parte de los casos. [16]

- La integración y entrega de aplicaciones se realizan de forma más rápida y eficiente.
- Los problemas de calidad se detectan y solucionan con mayor facilidad aumentando el estándar de calidad.

- Las mejoras en el código se implementan de forma más rápida y sin interrupciones del servicio, permite trabajar en un entorno de pruebas antes de pasar a producción de esta forma se verifica posibles fallas que se puedan presentar.
- Las herramientas DevOps permite un mayor nivel de colaboración entre desarrolladores y operadores, lo que conduce a un mejor entendimiento mutuo de sus respectivos objetivos y disminuir los tiempos muertos durante el desarrollo del proyecto.
- Los equipos de desarrollo y de operaciones, pueden centrarse en la creación de valor para el negocio en lugar de perder el tiempo lidiando con problemas técnicos relacionados con la integración y el despliegue de aplicaciones.
- Al existir varias herramientas DevOps no todas pueden ser implementadas en un mismo proyecto, por lo que es necesario buscar la que más se adapte a las necesidades a cubrir.

Cabe aclarar no siempre se cumplen las ventajas mencionadas, todo depende del tipo de herramienta DevOps que se escoja para solventar el problema. Adicional si el uso de las herramientas DevOps resulta desconocido en lugar de solventar el problema a tratar se puede generar un problema adicional.

1.9 Desventajas de DevOps

No todo es ventajas en el mundo DevOps, al estar en un mundo que cambia constantemente, las nuevas tecnologías deben adaptarse a los cambios y no siempre se lo hace a la velocidad que se espera. A pesar de contar con varias herramientas que ayudan a su funcionamiento, existen puntos que no han sido cubiertos y que deben ser considerados importantes.

A continuación, se detalla algunas desventajas que se consideran importantes al hacer uso de las herramientas DevOps:[17]

- Cubrir las necesidades, habilidades especializadas y conocimientos técnicos. Debido a que DevOps requiere una gran cantidad de automatización y el uso de herramientas y tecnologías específicas, se necesita conocimientos técnicos para implementarlo adecuadamente. La mayor parte de herramientas DevOps no cuentan con un entorno gráfico y es necesario tener conocimiento para trabajar con el terminal.

- Puede ser costoso: Algunas de las herramientas y tecnologías utilizadas en DevOps han sido adoptadas por algunas organizaciones, y lo que hacen es ofrecer paquetes que contienen una serie de herramientas DevOps para adaptarlas a las necesidades del cliente. Estos paquetes o contenedores cuentan con un sistema más fácil de instalar e incluso todo se maneja desde la nube, siendo esto por lo que estarían cobrando algunas organizaciones.
- Complejidad: las herramientas DevOps puede ser complejas en algunos casos, debido a que se involucra gran cantidad de elementos, como diferentes plataformas, sistemas y procesos. Esto puede hacer que sea difícil implementarlo correctamente sin la ayuda de expertos o profesionales capacitados.

Para solventar estas desventajas, aclarando que no son problemas directos de las herramientas DevOps es necesario contar con el personal debidamente capacitado, considerando también el factor económico y buscar cual es la herramienta y procesos más adecuados que se adapten a las necesidades que se requiere cubrir.

1.10 Metodología Kanban

Kanban es una técnica de gestión de proyectos que se enfoca en la mejora del flujo de trabajo. Se basa en la idea de visualizar el flujo de tareas a través de tarjetas, lo que permite a los equipos identificar fácilmente las tareas pendientes y planificar mejor el trabajo. También promueve el trabajo en colaboración y el establecimiento de límites para evitar el exceso de tareas sobre un solo usuario.

En los últimos años ha ganado popularidad entre los desarrolladores de software lo que ha permitido que sea considerada por las entidades empresariales de diferentes ámbitos. Kanban nació para ser aplicada en los procesos de fabricación, con los años se fue adaptando para cubrir procesos similares, obteniendo mayores beneficios durante su aplicación. En la Figura 1.3 se observa un ejemplo del tablero de Kanban.

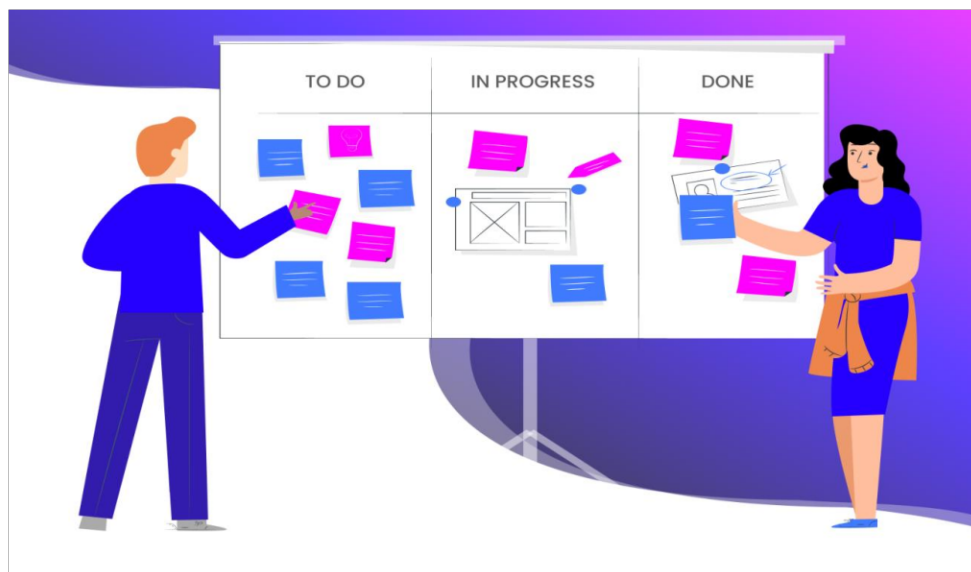


Figura 1.3 Ejemplo del tablero de Kanban. [18]

Kanban es un método de gestión de proyectos que se centra en la mejora del flujo de trabajo a través de la eliminación de cuellos de botella y el establecimiento de límites en el número de tareas en progreso. El objetivo es lograr una mayor eficiencia y productividad al permitir que los equipos se concentren en las tareas actualmente en curso, sin sobrecargarlo con demasiadas tareas a la vez. Kanban también promueve una mejor comunicación y coordinación entre los miembros del equipo, ya que todos pueden ver fácilmente qué tareas están siendo realizadas por quién y cuáles son las siguientes en la cadena.

1.11 DevOps con Kanban

La implementación de una metodología DevOps requiere la adopción de una serie de prácticas y técnicas que permitan a las organizaciones mejorar la colaboración y coordinación entre los equipos de desarrollo y operaciones. Algunas de estas prácticas son el uso de herramientas automatizadas para el seguimiento y el control del ciclo de vida del software, así como también la adopción de un enfoque Kanban para visualizar y administrar los flujos de trabajo.

Por su lado Kanban anima a que el trabajo fluya y sea continuo a medida que los equipos se adaptan a los nuevos procesos, se van eliminando los cuellos de botella que implica adaptarse a lo nuevo. Facilitando que nuevas tareas se generen y estas sean resueltas en un tiempo menor, esto facilita el despliegue de herramientas DevOps que anteriormente fueron omitidas por no contar con la coordinación correcta.

Al visualizar todo el flujo de trabajo se pretende garantizar que sea continuo y conocer las tareas primordiales. Es decir, se enfoca en un entorno más amigable y trabaja de forma colaborativa entre las diferentes áreas implicadas en el desarrollo o creación del nuevo proyecto para llegar al objetivo deseado, de esta manera se crea una cultura DevOps que se puede mantener y continuar aplicando en futuros proyectos.

La constante visibilidad de las tareas a desarrollar o procesos que se deben seguir nos permite identificar, los problemas que se deben corregir, estos pueden ser de estructura física, personal técnico o ambiente de trabajo. Considerando que la prioridad es el tiempo de trabajo de cada usuario se pretende optar por el camino que genere el menor retraso posible. En el caso de que una tarea requiera más tiempo de lo estimado, es factible que la tarea sea dividida en varias subtareas que permitan el avance del proyecto.

Dentro del tablero se observa las tareas de alta prioridad, por lo que requieren mayor atención y concentrar el esfuerzo para realizarlas eficiente y eficazmente, garantizando el proceso continuo de las tareas.

Relacionando los tableros Kanban con las herramientas DevOps se crean estructuras de trabajo fáciles de llevar a cabo. Para iniciar el proyecto es necesario crear varios tableros Kanban, donde el tablero principal como se observa en la Figura 1.4 tendrá definido todas las tareas que se deben llevar a cabo. El segundo tablero Kanban manejará tareas que sean asignadas a los diferentes departamentos implicados en el proyecto. El tercer nivel de Tablero Kanban lo maneja cada área con sus tareas asignadas. Puede existir un cuarto nivel de tablero Kanban que lo maneja directamente el usuario asignado a cumplir la tarea.[20]

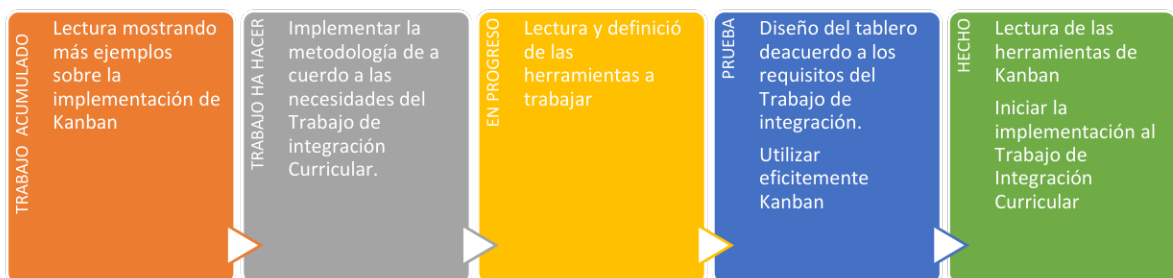


Figura 1.4 Ejemplo de tablero principal

Para el manejo de las herramientas DevOps se puede optar por las siguientes opciones para trabajar con Kanban:

- Tablero de Instalación: contiene las tareas que se deben cumplir para realizar la correcta instalación.
- Tablero de Aplicación: contiene tareas donde se muestra la aplicación de la herramienta DevOps instalada.
- Tablero de Pruebas: este tablero relaciona los diferentes departamentos que harán uso del proyecto o contarán con alguna participación. [19]

Es importante mencionar que no siempre se puede hacer uso de los tableros porque pueden tratarse de tareas pequeñas que no requieren hacer uso. También hay que considerar la experiencia del usuario, un usuario principiante es necesario que siga los procesos hasta obtener el resultado deseado. A diferencia de un usuario experimentado, hará caso omiso de algunas tareas.

1.12 Android Studio

Conocer su utilidad, características y funciones es vital para sacar el máximo provecho de esta herramienta. Android Studio es un entorno de desarrollo integrado (IDE) para la creación de aplicaciones para el sistema operativo Android. El IDE está basado en el intérprete de código fuente JetBrains IntelliJ IDEA y ofrece una serie de características especialmente diseñadas para facilitar el desarrollo de aplicaciones para Android. Las características principales de Android Studio son: [21]

1. Soporte para el desarrollo de aplicaciones (Go edition).
2. Herramientas para el desarrollo de aplicaciones de realidad aumentada ARCore.
3. Editor de código con resaltado de sintaxis y autocompletado de código.
4. Depurador de código para localizar y corregir errores.
5. Gestor de proyectos para organizar y administrar los archivos de código fuente.
6. Control de versión para rastrear y administrar cambios en el código fuente.
7. Generador de paquetes APK para compilar e instalar aplicaciones en dispositivos Android.
8. Emulador de Android para probar aplicaciones en un entorno de simulación.

9. Android Studio profiler para analizar el rendimiento de las aplicaciones.
10. Android Plugin for Gradle para compilar e integrar aplicaciones en el sistema de construcción Gradle.

Ventajas

Es un IDE muy completo y tiene muchas funcionalidades útiles para el desarrollo de aplicaciones Android. La interfaz de usuario es muy intuitiva y fácil de usar. Ofrece un buen soporte para el desarrollo de aplicaciones Android.

Desventajas


Algunas de las funcionalidades avanzadas pueden ser un poco confusas de usar. No es tan ligero y rápido como otros IDEs.

1.13 Lenguaje de Programación Kotlin

Kotlin es un lenguaje de programación de código abierto que se ejecuta en la JVM (Java Virtual Machine) y también puede compilarse a JavaScript. Fue creado por JetBrains, el creador de IntelliJ IDEA, y se lanzó por primera vez en 2011.

A partir de 2019, Kotlin es el lenguaje de programación oficial para Android. Kotlin se diseñó para ser un lenguaje de programación más productivo que Java, tiene una sintaxis más concisa, lo que permite a los desarrolladores escribir código más rápido. En la Figura 1.5 se muestra un ejemplo de la sintaxis básica que se utiliza.

SINTAXIS BÁSICA DE KOTLIN



```

main.kts
fun main(args:Array<String>){
    var dato = 10;
    val num = 10
    val nombre:String = "José"

    var r = num + dato;
    print(r)
    print(nombre)
}

```

Se define una función `main` para poder ejecutar código.


Se recibe un arreglo de `String` en caso de que necesitemos parámetros.

Se usa `var` para definir variables. Y `val` para variables que no van a cambiar.

El `;` es opcional al final de las líneas de las instrucciones.

`print` es una función que ya tiene kotlin definida para imprimir.

Es de tipo dinámico, el intérprete infiere el tipo de dato según su valor.



Prof. José Luján

Kotlin fue nombrado por Google como lenguaje oficial para el desarrollo de aplicaciones Android.

Figura. 1.5 Sintaxis de Kotlin. [22]

También tiene varias características avanzadas que Java no tiene, como el soporte para funciones de orden superior, la inferencia de tipos y el soporte para el patrón de diseño de métodos de fábrica. JetBrains ha estado trabajando en estrecha colaboración con Google para mejorar Kotlin para Android.

En 2018, Google anunció que Kotlin era un lenguaje de programación "first class" para Android, lo que significa que es un lenguaje oficial que se puede usar para el desarrollo de aplicaciones Android. Kotlin también se puede usar para el desarrollo de aplicaciones web con la plataforma de aplicaciones web de Kotlin, Ktor.

Ktor es una plataforma de aplicaciones web de código abierto que se puede usar para crear aplicaciones web escalables y seguras. Está optimizado para Kotlin y se puede usar para crear aplicaciones web de forma rápida y fácil.

1.14 Node.js

Es un entorno de tiempo de ejecución de código abierto para el desarrollo web del lado del servidor. Se basa en el lenguaje de programación JavaScript y permite a los desarrolladores escribir código del lado del servidor de forma rápida y eficiente con una configuración mínima. El lenguaje fue creado por Ryan Dahl en 2009 y, desde entonces, se ha vuelto popular entre los desarrolladores para crear sitios web, aplicaciones y API

(Interfaces de Programación de Aplicaciones). A través de su sistema de biblioteca modular y su marco liviano, brinda un amplio soporte para aplicaciones de alto rendimiento. [23]

1.15 Herramientas DevOps para el prototipo de Aplicativo Móvil

Para iniciar con el proyecto se ha realizado la siguiente selección de herramientas DevOps:

- Repositorio de Documentos: Google Drive.
- Repositorio de Versiones: GitHub.
- Contenedor: Docker.

Estas herramientas son de gran utilidad al momento de desarrollar el prototipo de aplicativo móvil. No son las únicas herramientas que pueden ser aplicadas, pero son las que se adaptan a las necesidades que se requiere solventar.

Dependiendo el tipo de necesidad que se requiera solventar se puede hacer uso de las siguientes herramientas:

- Git.
- Jenkins.
- Nagios.
- Kubernetes.
- Ansible, entre otras.

En la Figura 1.6 se observa un resumen describiendo los puntos más importantes para el desarrollo del prototipo de aplicativo móvil.

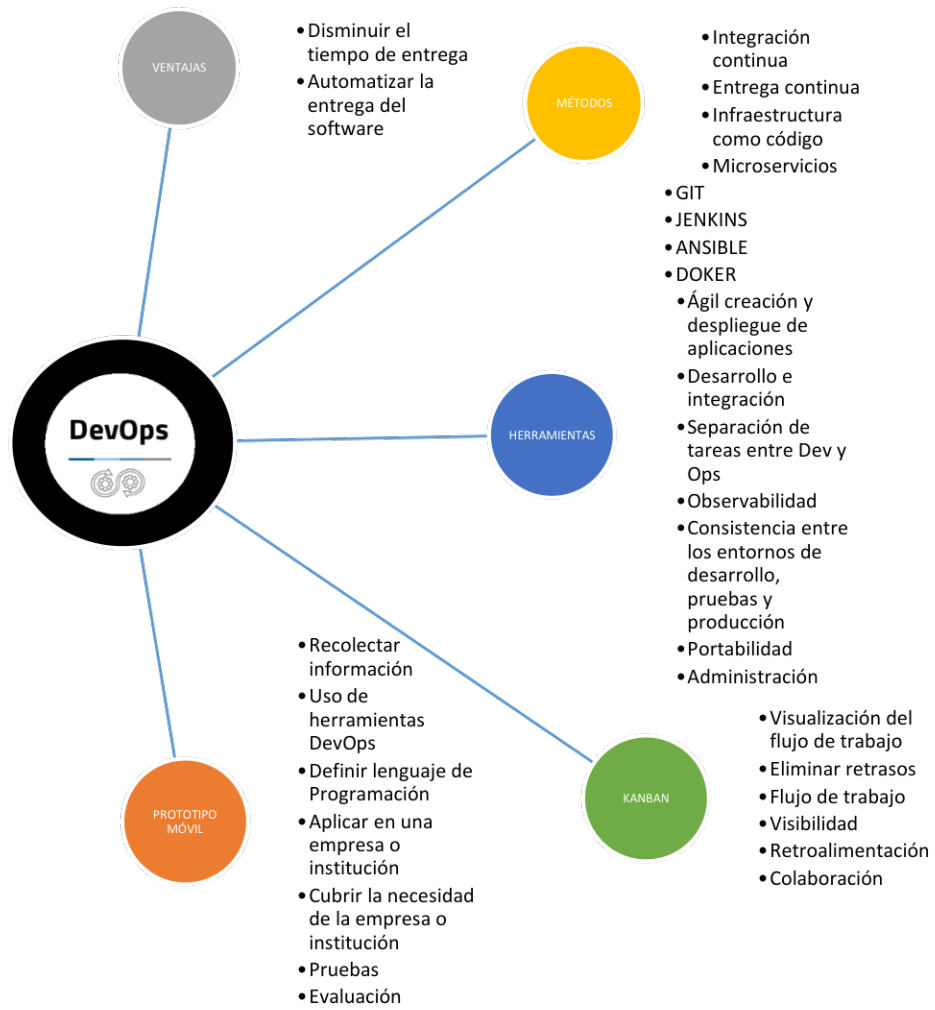


Figura 1.6 Principales ideas de los temas tratados

2. METODOLOGÍA

Dentro de las instituciones públicas en el Ecuador no se encuentra difundido en uso de herramientas que facilitan el desarrollo de software, para enfocarnos en el tema y dar a conocer herramientas vigentes se ha establecido lo siguiente:

- Escoger a la Institución ECU911 para explicar el uso de Herramientas DevOps.
- Conocer al menos un problema donde se pueda aplicar las Herramientas DevOps y este problema pueda ser solventado.
- Proporcionar información sobre las herramientas vigentes que serán de gran ayuda en el desarrollo de software dentro de la institución.
- Poner a consideración una propuesta de uso de las herramientas DevOps para que puedan ser implementadas por los desarrolladores de software.

Uno de los puntos más importantes es hacerles conocer que existen herramientas que les facilitan varias tareas, considerando actualmente que la Institución ECU911 no tiene implementado un proceso que haga uso exclusivamente de Herramientas DevOps.

Para aclarar el tema se plantea solventar uno de los problemas que tiene la Institución ECU911, donde es necesario solventar la siguiente necesidad:

- Disminuir el uso del aplicativo WhatsApp para el envío de información relevante.

Esta consideración es importante puesto que la Institución ECU911 trabaja con información de las emergencias que ocurren a nivel nacional y al hacer uso del aplicativo WhatsApp la información puede ser filtrada desde cualquier usuario.

Con la necesidad planteada se considera desarrollar un prototipo de aplicativo móvil que reemplace al aplicativo WhatsApp de esta manera se obtiene control sobre el aplicativo a desarrollar y es posible controlar a los usuarios que hacen uso del aplicativo.

De ahora en adelante es necesario considerar los siguientes puntos:

- Realizar un estudio de la situación actual de la Institución en cuanto al uso de Herramientas DevOps.
- Explicar de forma breve y concisa el uso de Herramientas DevOps.

- Desarrollar el prototipo de aplicativo móvil e indicar las Herramientas DevOps que pueden ser aplicadas durante el desarrollo de software.
- Realizar pruebas del prototipo de aplicativo móvil.
- Encuesta para conocer los resultados obtenidos.

Para el desarrollo del prototipo de aplicativo móvil se consideran los siguientes puntos:

- Herramienta de desarrollo: Android Studio, para trabajar con dispositivos que ejecuten el Sistema Android.
- Lenguaje de Programación: Kotlin. Debido a los siguientes puntos: [2]
 - Lenguaje de programación de código abierto.
 - Usado fundamentalmente para el desarrollo de aplicaciones Android.
 - Puede desarrollarse sobre la máquina virtual de Java.
 - Permite la interoperabilidad con código Java, permitiendo la migración de forma paulatina.
- Herramientas DevOps:
 - Repositorio de Documentos: Google Drive.
 - Repositorio de Versiones: GitHub.
 - Contenedores: Docker.
- Encuesta de satisfacción.

A continuación, se detallan los puntos previamente planteados.

2.1 Descripción Servicio Integrado de Seguridad ECU 911[24]

Servicio Integrado de Seguridad ECU 911 servicio de respuesta inmediata implementado en Ecuador para brindar atención y respuesta para casos de siniestros, desastres y emergencias movilizando el personal disponible con la finalidad de brindar una atención rápida y oportuna.

Las siguientes instituciones forman parte del ECU 911: Policía Nacional, Fuerzas Armadas, Cuerpo de Bomberos, Comisión Nacional de Tránsito, Ministerio de Salud Pública, Instituto Ecuatoriano de Seguridad Social, Servicio Nacional de Gestión de Riesgos y Emergencias, Cruz Roja Ecuatoriana y otros organismos locales encargados de la atención de emergencias a través de un número único 9-1-1.[24]

Misión

Gestionar en todo el territorio ecuatoriano, la atención de las situaciones de emergencia de la ciudadanía, reportadas a través del número 911, y las que se generen por video vigilancia y monitoreo de alarmas, mediante el despacho de recursos de respuesta especializados pertenecientes a organismos públicos y privados articulados al sistema, con la finalidad de contribuir, de manera permanente, a la consecución y mantenimiento de la seguridad integral ciudadana. [24]

Visión

"Ser una institución nacional líder y modelo en la región para la coordinación de servicios de emergencia utilizando tecnología de punta en sistemas y telecomunicaciones, comprometidos con la calidad, seguridad, salud en el trabajo y el medio ambiente que permitan brindar un servicio único y permanente a la ciudadanía. [24]

Objetivos de Calidad [24]

- "Coordinar y atender emergencias de forma efectiva con un cumplimiento de tiempo de registro de la alerta (TRA) de 2'49". ("Objetivos – Servicio Integrado de Seguridad ECU 911")
- Coordinar y atender emergencias de forma oportuna con un cumplimiento de tiempo de atención de la alerta (TE) de 12".
- Disponibilidad de la plataforma (98%).
- Cumplir el plan de capacitación en atención de emergencias en un 85%.
- Cumplir con el número de personal certificado en atención de emergencias en un 85%.
- Lograr una satisfacción de los usuarios en un 85%.
- Porcentaje de cobertura de territorio ecuatoriano en un 85%.

Objetivo Estratégico [24]

- Incrementar la calidad en la prestación del servicio de atención de emergencias a nivel nacional.
- Incrementar la articulación interinstitucional en la prestación de servicios de atención de emergencias.
- Incrementar el posicionamiento del Servicio Integrado de Seguridad ECU 911 a nivel nacional e internacional.

Fortalecer las capacidades institucionales.

2.2 Organigrama Institucional [24]

La Institución se encuentra organizada por Direcciones, encabezado por la Dirección General que se encarga de la administración a nivel nacional, ubicada en la ciudad de Quito. En un nivel inferior se encuentran las Direcciones Zonales las cuales se encuentran ubicadas en las ciudades más importantes del Ecuador.

El territorio nacional se encuentra cubierto debido a que cuentan con 17 centros a escala nacional, distribuido en 16 centros a escala nacional y 1 centro local en las Galápagos. Cuentan con 14 salas operativas en varias provincias y cantones que no cuentan con infraestructura de un centro ECU911.

De esta manera se da cobertura a nivel nacional teniendo presencia en todas las provincias del Ecuador.

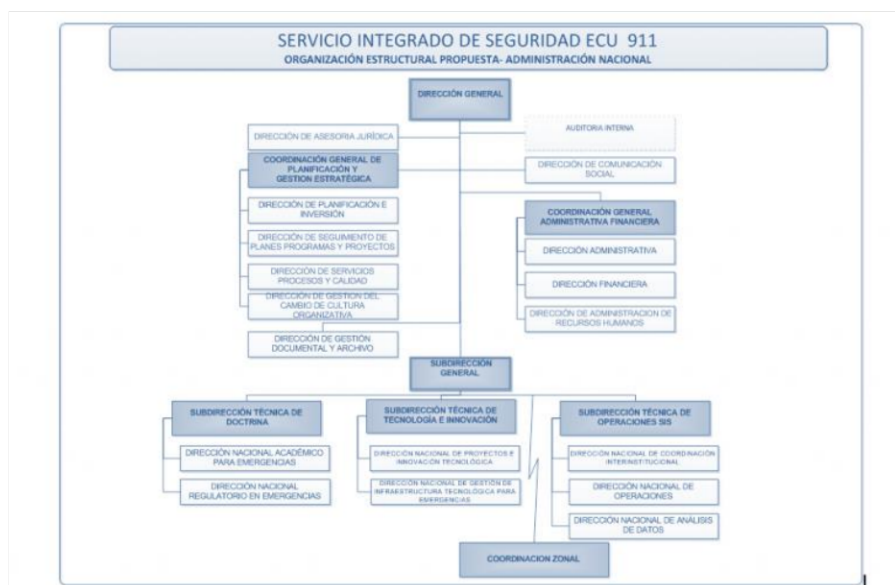


Figura 2.1 Organización Estructural Propuesta – Administración Nacional [24]



Figura 2.2 Organización Estructural – Administración Zonal [24]

2.3 Descripción de la Unidad de Tecnologías de Información

Dentro de la estructura del ECU911 la Unidad de Tecnología de Información se encuentra estructurada de la siguiente manera:



Figura 2.3 Organización Estructural de Unidad de Tecnologías de Información

Esta organización se encuentra establecida en las instalaciones del ECU911 en la ciudad de Quito.

Esta Subdirección es la encargada del desarrollo de software que posteriormente es puesto a producción siguiendo los estándares propios de la Institución. Es en esta Subdirección donde se desarrolló el trabajo de las prácticas DevOps.

2.4 DevOps dentro del ECU911

Las instituciones públicas tienen establecido su sistema de manejo de actividades dentro de los estándares previamente acordados y que mejor se adaptan a las necesidades que desarrollan, dentro del ECU911 cuentan con procesos para la creación de nuevo software. Cabe aclarar que actualmente los procesos para el uso de herramientas DevOps es muy escaso y desconocen las herramientas.

A continuación, se detalla brevemente los pasos que realiza el ECU911 para cumplir con el desarrollo de software:

- Planteamiento de la necesidad a solventar.
- Establecimiento de un cronograma acorde al tiempo necesario para solventar la necesidad.
- Revisión de avances de acuerdo con el cronograma establecido.
- Periodo de pruebas.
- Reservar el espacio en disco y recursos que consume en los servidores donde será alojado el software.
- Producción.

Debido a la confidencialidad de la información no es posible incluir el nombre de las herramientas que utilizan, dentro de esta serie de pasos antes mencionados se observa que son pocas las herramientas DevOps que utilizan desde que inicia el proceso de desarrollo de software hasta que finaliza. A continuación, serán mencionadas las herramientas de uso común:

- Google Drive y SharePoint, para compartir información relevante.
- Almacenamiento local, por cuestiones de seguridad prefieren mantener el software en la máquina que se ha desarrollado. Solamente cuando el software pasa al ambiente de pruebas es trasladado.
- Servidores CentOS.

Para facilitar el trabajo de los desarrolladores haciendo uso de herramientas DevOps se comenta lo siguiente que puede ser incluido dentro del proceso establecido para la generación de software:

- Google Drive es una herramienta para ser usada como Repositorio de Documentos.
- SharePoint al contar con una versión de pago, se cuenta con mayor seguridad lo que permite ser usada como un Repositorio de Documentos y un Repositorio de Versiones con sus limitaciones respectivas.
- Al contar con servidores CentOS se pueden incorporar herramientas DevOps como las que se muestran en la Figura 2.4:

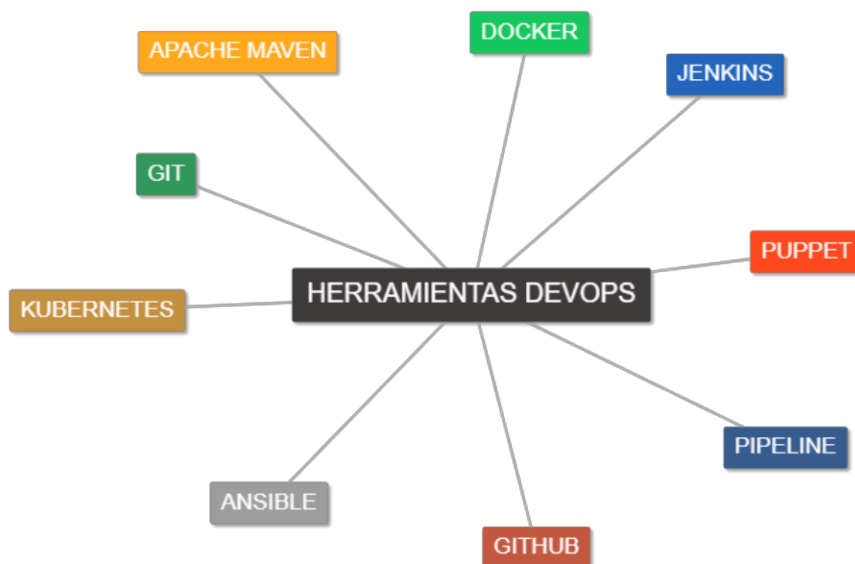


Figura 2.4. Herramientas DevOps

Con una breve explicación de las herramientas DevOps son refrescados los conocimientos y en algunos casos solamente han escuchado hablar de las herramientas, pero no han sido aplicadas en el desarrollo de software.

En muchos de los casos no se puede hacer uso de las Herramientas DevOps esto debido a que en los requisitos de contratación para los Analistas de Proyectos y de Infraestructura no es un requisito y el personal aprende lo que se encuentra establecido.

2.5 Propuesta de herramientas DevOps para el ECU911

Con la finalidad de facilitar las tareas a los desarrolladores de la institución ECU911 se realiza la siguiente propuesta:

1. Considerar como opción el uso de Herramientas DevOps.
2. Incluir en el desarrollo de software herramientas sencillas iniciando con las siguientes:
 - a. Repositorio de Documentos.
 - i. Google Drive.
 - ii. SharePoint.
 - iii. OneDrive.
 - b. Repositorio de Versiones.
 - i. Git.
 - ii. GitHub.
 - c. Herramientas de Monitoreo.
 - i. Nagios (Monitorear la Infraestructura TI).
 - ii. Prometheus.
 - d. Herramientas de Automatización
 - i. Docker.
 - ii. Kubernetes.
3. Levantar un servidor CentOS exclusivo para herramientas DevOps.
4. Contar con un ambiente de pruebas.

Debido a que los procesos ya se encuentran establecidos dentro del desarrollo de software en la Institución ECU911 estas herramientas pueden irse implementando de a poco, con cada nuevo proyecto que ingrese se podría hacer uso de alguna de las herramientas antes mencionadas. Con esto también se ayuda a que el nuevo personal que ingrese se familiarice con el uso de las herramientas DevOps.

Al implementar las herramientas antes mencionadas se obtendrán los siguientes resultados:

- Requisitos claros al finalizar la reunión y todos los miembros del grupo tendrán acceso y libertad de revisarlos.
- Controlar las versiones de software de una manera ordenada y en un solo sitio, sin la necesidad de recurrir a la búsqueda local y en muchos de los casos olvidando cual era la versión final.
- Solventar el problema común “En mi máquina si funciona”.
- Contar con las métricas adecuadas de monitoreo tanto para el software, como para el servidor donde se está ejecutando el software en el caso de que los recursos asignados no sean suficientes.
- Los manuales de instalación siempre estarían en un solo lugar y al alcance de todos, solventando el problema de rotación de personal.

Se ha considerado que se trata de herramientas de gran utilidad y que pueden solventar varios problemas y generar beneficios una vez que sean puestas en práctica y se mantenga una correcta utilización durante el desarrollo de software. Lo importante es iniciar con una herramienta a la vez e ir buscando los beneficios aportados y continuar implementando el resto de las herramientas.

Las herramientas DevOps son de gran utilidad llegando incluso en algunos sectores a solicitar Ingenieros DevOps con la finalidad de mejorar el desarrollo de software y la comunicación con la parte operativa.

2.6 Desarrollo del aplicativo móvil para el ECU911

Para cubrir la necesidad de reemplazar el uso del aplicativo WhatsApp y silenciarlo para la información relevante haciendo referencia a la Figura 2.5, se plantea el desarrollo de un prototipo de aplicativo móvil, el cual puede ser administrado por la Institución ECU911 y controlar la cantidad de usuarios que reciben la información.



Figura 2.5 Disminuir el uso del aplicativo WhatsApp

Prototipo de aplicativo móvil

Para el desarrollo del prototipo de aplicativo móvil se realizaron varias reuniones con el objetivo de definir los requisitos que requiere la institución ECU911. De las cuales se definen los siguientes requisitos:

- El prototipo móvil debe tener interfaz de usuario amigable y fácil de usar.
- Debe tener un layout de acceso, esto debido a que el aplicativo será de uso exclusivo del personal que el ECU911 asignará.
- El usuario debe ingresar su correo electrónico y una clave para poder acceder. Esta información se encuentra almacenada en una base de datos en PostgreSQL en las instalaciones del ECU911.
- Los usuarios son registrados con anterioridad en base a criterios que define el ECU911 en sus políticas internas de seguridad.
- Una vez que el usuario cuente con el acceso autorizado el aplicativo estará en la espera de información. La información por presentar es solamente texto, que previamente es filtrada.
- La información que el usuario recibirá corresponde a Eventos Relevantes, esta información no puede ser modificada por parte del usuario. Existen procesos previos que la Institución ECU911 considera para que la información sea tratada como relevante.

- El prototipo de aplicativo móvil solamente será instalado en los teléfonos celulares autorizados por la institución ECU911, estos dispositivos se encuentran registrados en una base de datos que es de acceso exclusivo para Eventos Relevantes.
- Solamente para el proceso de pruebas se instalará en dispositivos que no han sido autorizados por la Institución ECU911, una vez terminadas las pruebas el aplicativo debe ser removido de los dispositivos o en sus defectos bloqueados desde el aplicativo web que será utilizado para enlazar esta información.
- Al tratarse de un prototipo y de uso exclusivo para personal autorizado de la Institución ECU911, no estará disponible en la AppStore de Google.

Aclarados los requisitos para la Institución ECU911 en cuanto al desarrollo del prototipo de aplicativo móvil es necesario arrancar con las condiciones iniciales para ello es necesario la creación del tablero Kanban.

Para la creación del tablero Kanban se definen dos tareas principales:

- Requisitos ECU911.
- Requisitos DevOps.

Tal como se muestra en la Figura 2.6

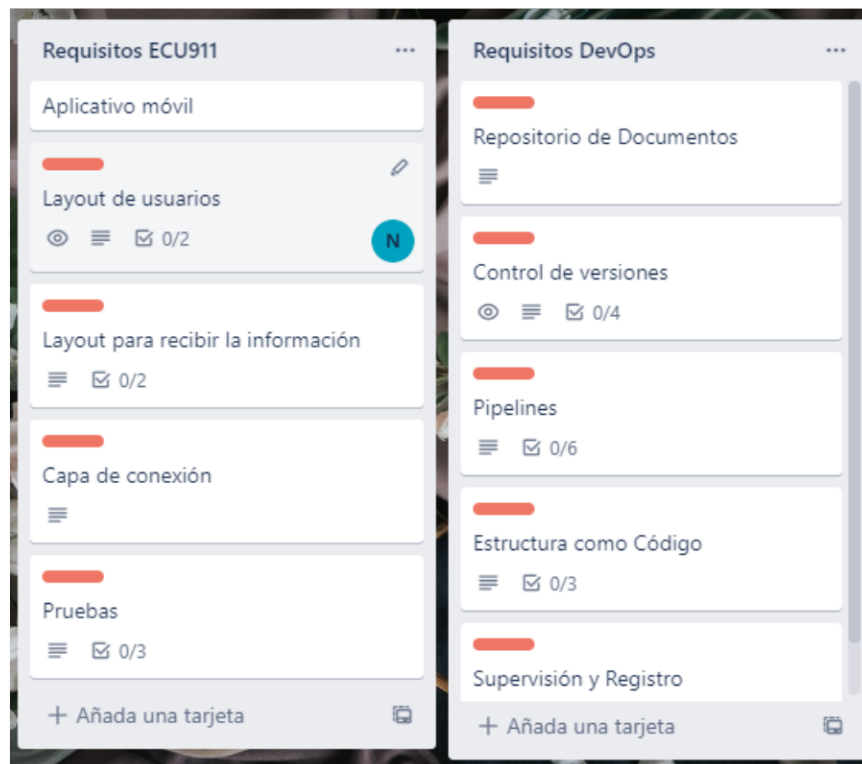


Figura 2.6 Tablero Kanban

Las tareas detallan lo que se debe llevar a cabo hasta obtener el prototipo móvil. Las tareas principales a su vez cuentan con varias subtareas las mismas que registrarán el progreso tanto de la tarea principal como de la subtarea. Todas las subtareas deben pasar con los siguientes procesos:

- Por Hacer.
- En Progreso.
- Terminado.

Las tareas en su estado inicial tienen la etiqueta “Por Hacer” conforme el proyecto avance tomaran diferentes estados dentro del tablero Kanban. Con el objetivo de tener claro las etiquetas se las define con los siguientes colores:

- **“Por Hacer”**
- **“En Progreso”**
- **“Terminado”**

De esta manera se puede identificar fácilmente el estado de cada tarea que tenemos en el tablero como se indica en la Figura 2.7.

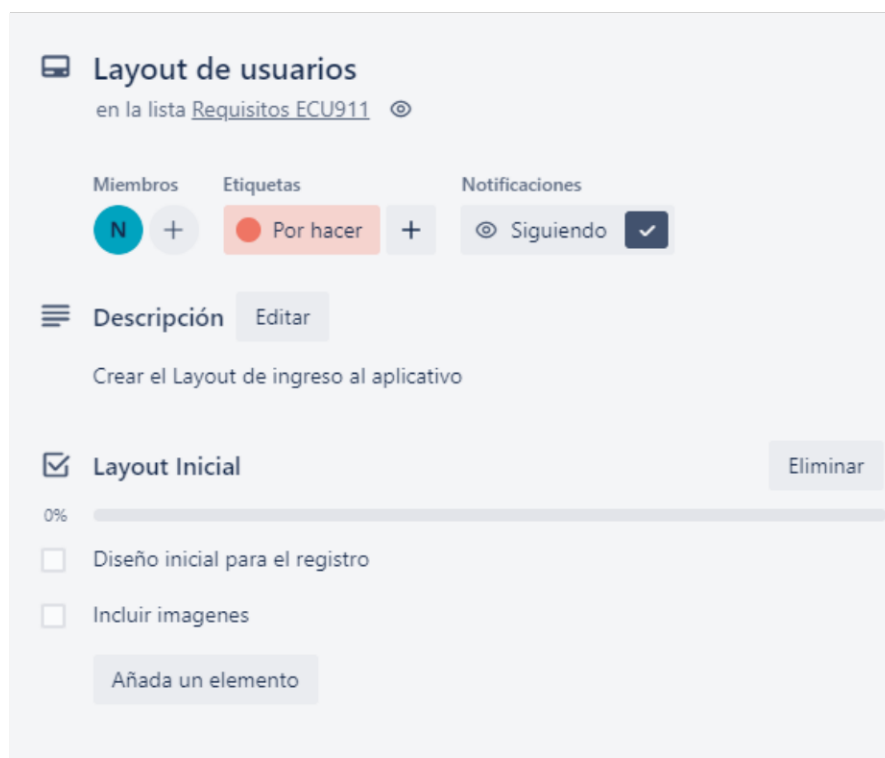


Figura 2.7 Tarea dentro del tablero de Kanban

Como se observa en la Figura 2.7 dentro de cada tarea del tablero Kanban se observa los siguientes datos:

- Tarea principal a la que pertenece.
- Persona a cargo de llevar la tarea.
- Estado actual de la tarea.
- Descripción de la tarea que se está llevando a cabo.
- Incluye las subtareas que se deben cumplir.
- Dentro de las subtareas se encuentra una barra de progreso que identifica el porcentaje de avance como se observa en la Figura 2.8.

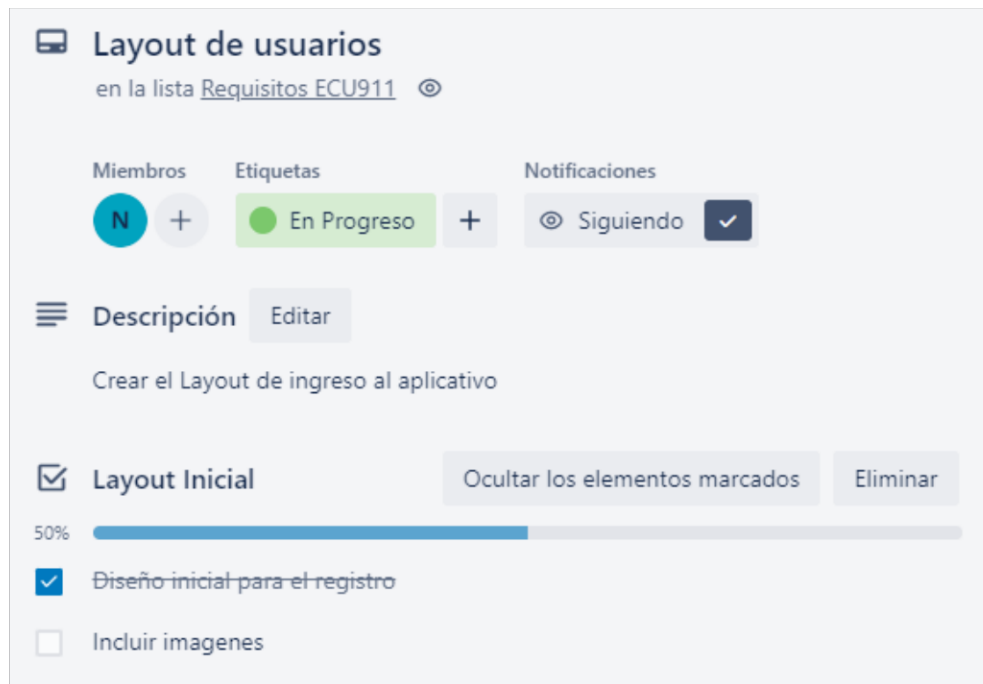


Figura 2.8 Progreso de la tarea dentro del tablero Kanban

En la Figura 2.8 podemos ver el progreso de la tarea, de etiqueta ha cambiado a **“En Progreso”**, de igual manera se observa el progreso de las subtareas.

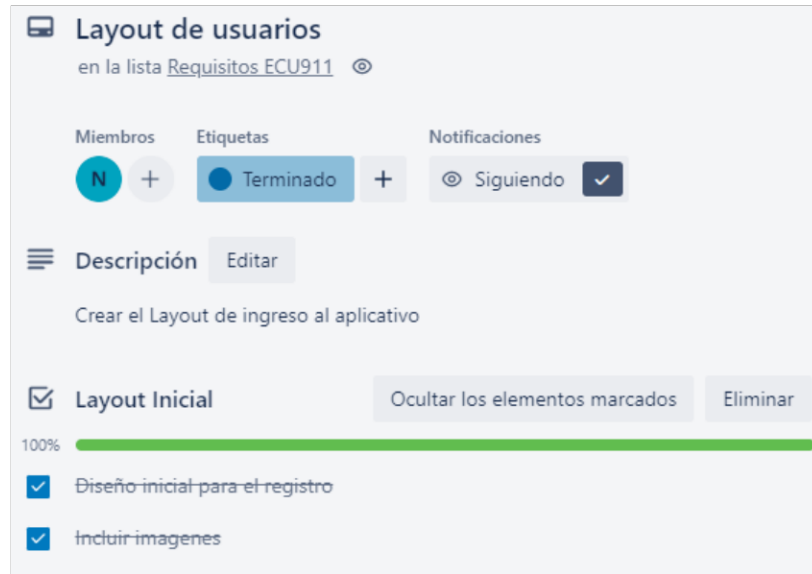


Figura 2.9 Tarea terminada en el tablero Kanban

Como se observa en la Figura 2.9 se tiene la tarea completa, de igual manera se observa que la etiqueta ha cambiado a **“Terminado”**.

En la Figura 2.10 se observa el progreso de la tarea principal una vez que las subtareas han sido completadas. Se observa que tres subtareas están en el estado de “Terminado ” y una subtarea en el estado de “En Progreso”. Como observamos la tarea principal muestra al usuario que se encuentra a cargo y el progreso individual de cada subtarea.

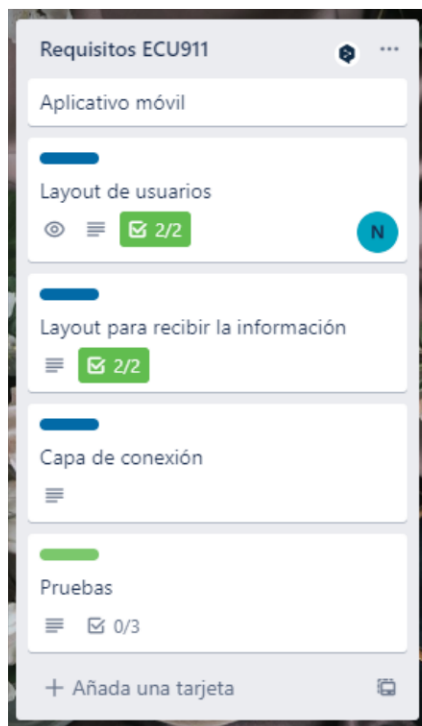


Figura 2.10 Actualización de la tarea principal

Repositorio de Documentos

Durante el desarrollo del proyecto para almacenar los documentos se ha utilizado el repositorio en línea Google Drive, donde serán almacenados los documentos referentes al proyecto. En la Figura 2.11. se observa los documentos que son almacenados durante la elaboración del proyecto.



Figura 2.11 Repositorio de documentos – Google Drive

Durante el proyecto las actas que se encuentran almacenadas en Google Drive son:

- Acta de primera reunión donde se explica los detalles iniciales.
- Acta de requisitos definidos por la Institución ECU911.
- Acta de seguimiento, como se observa en la Figura 2.12.

Nombre	Propietario	Última modificación	Tamaño de archivo
Reunión Avance	yo	4:09 yo	2 kB
Segunda Reunión	yo	3:50 yo	2 kB
Primera Reunión	yo	22:10 yo	2 kB

Figura 2.12 Actas almacenadas en Google Drive

Los documentos que se encuentran almacenados son referentes a las reuniones para definir el proyecto y los requisitos solicitados.

Repositorio de versiones

Para llevar un control de las versiones se utiliza la herramienta GitHub la cual nos ayuda a mantener todas las versiones que se han utilizado durante el desarrollo del proyecto, algunas versiones cuentan con cambios importantes mientras que otras han sido pequeños cambios los que se han ido implementando, estas versiones han sido organizadas de acuerdo con los requisitos que solicita la institución que requiere que cumpla el aplicativo móvil. Algunas versiones de prueba no han sido consideradas, por no tener cambios que realmente aporten al aplicativo.

El trabajo se ha realizado tomando en cuenta los requisitos que se han establecido en las reuniones no todos los cambios han sido aceptados y otros han sido descartados debido a que comprometen la seguridad del usuario, por lo que en un principio se consideró como aceptado en el transcurso del desarrollo del prototipo aplicativo móvil fue descartado y se ha optado por otras sugerencias. En la Figura 2.13 se muestran las principales versiones que se encuentran almacenadas en GitHub, cada una fue de ayuda hasta obtener la versión final.



Figura 2.13 Herramienta GitHub, principales versiones almacenadas.

Un ejemplo de ello y el más notorio fue el de obtener el número telefónico directamente desde la SIM CARD del usuario que tiene instalado el prototipo de aplicativo móvil, se hicieron algunas pruebas tratando de obtener esta información, pero no se llegó a obtener

el resultado adecuado. Llevando la consulta a foros indican que esta opción ya no se permite, en años anteriores se podía obtener dicha información, pero resulto ser una falla de seguridad hacia la información del usuario por lo que este tipo de información había sido deshabilitada al público en general y solamente ciertas compañías podían tener acceso a esta información a través de una API que implicaba un costo monetario para hacer uso de esta información.

Después de informar este tema al personal del ECU911 se toma la decisión de descartar esta opción de verificación de usuario, por lo que se mantiene el formato de autenticación a través del uso del correo electrónico y una contraseña.

A pesar de descartar la verificación del número telefónico directamente desde la Sim Card fueron creadas versiones de esta opción de autenticación. Las cuales también fueron almacenadas en el repositorio GitHub, en la Figura 2.15 se muestra el repositorio creado donde se almacena el proyecto y las diferentes versiones.

Para ello dentro de la cuenta de GitHub se crea la carpeta “TrabajoFinal” haciendo referencia al proyecto de TIC que se lleva a cabo. Dentro de esta carpeta se almacena los diferentes proyectos que se realizaron con la finalidad de obtener el prototipo móvil final, de igual manera se encuentran almacenadas las versiones de los diferentes proyectos

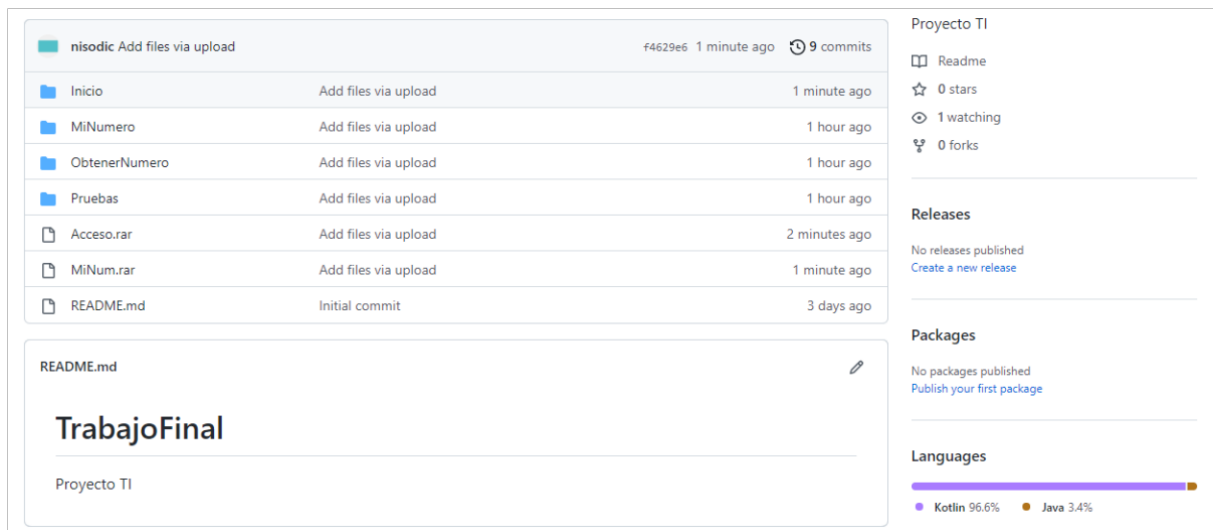


Figura 2.15 Repositorio creado en GitHub utilizado para el prototipo móvil.

De los proyectos almacenados GitHub nos muestra un resumen de la información más relevante, como los Lanzamientos, los Paquetes y el Idioma que se usa en los proyectos. Esta información se muestra en la Figura 2.15, al iniciar solamente nos va a mostrar la información de los Lenguajes de Programación que han sido utilizados.



Figura 2.16 Resumen de la información almacenada en GitHub.

Al iniciar los registros esta información muestra que no se han realizado nuevas versiones por lo que no puede mostrar información, de igual manera no se han generado paquetes, al iniciar se muestra la información del lenguaje de programación que se está utilizando, como se muestra en la Figura 2.16 registra 2 tipos de lenguajes, Kotlin en un 97,2% y Java en un 2,8%. Esta información es coherente dado que el proyecto inicial se utilizó Java. Se decidió el uso de Kotlin debido a que es mayormente soportado por parte de Google. De igual manera Java cuenta con un gran soporte por parte de comunidades y foros en sitios de Internet.

Nuevas versiones

Para realizar el registro de las versiones de los proyectos realizamos los siguientes pasos:

- Hacer clic en “Crear una nueva versión”. Nos abre un formulario donde vamos a registrar los datos que nos solicitan.
- Asignar el número de versión sobre el que se ha trabajado, se puede utilizar una etiqueta predeterminada o escribir una personalizada.
- Colocar un nombre de la nueva versión, este proceso lo podemos observar en la Figura 2.17.

- Hay que describir brevemente que cambios se han realizado.
- Se deben adjuntar los archivos donde se realizaron los cambios.
- Finalmente se hace clic en “Publicar comunicado” tal como se observa en la Figura 2.18.



Figura 2.17 Registro de versiones



Figura 2.18 Cargado de archivos modificados.

Terminado estos pasos notamos que en la sección de código ya contamos con una versión beta, a medida que se realicen nuevos cambios, podemos registrar las nuevas versiones y llevar el registro adecuado de los cambios que hemos realizados. Al momento se trabaja

con versiones beta, una vez que tengamos la versión estable nos permite establecer como una versión de prelanzamiento. El número de versiones seguirá incrementando de acuerdo a los cambios que sean registrados, en la Figura 2.19 se observa la primera versión.



Figura 2.19 Registro de versiones de proyectos

Contenedores

Para la creación de contenedores se ha optado por utilizar Docker, esto debido a que los proyectos realizados por pasantes dentro de la institución ECU911 son realizados en las máquinas personales de cada pasante, cuando el proyecto es terminado y cambia de lugar de ejecución presenta problemas y en la mayor parte de los casos son complicados de solventar.

Para el desarrollo del prototipo de aplicativo móvil el desarrollo lo hemos llevado a cabo sobre una máquina que tiene las siguientes características:

- Disco duro: 1TB.
- Procesador: Intel Core i7 de 7th Generación.
- Memoria RAM de 8GB.

El equipo con las características antes mencionadas fue proporcionado por parte de la Institución ECU911, el cual es utilizado para desarrollo de aplicaciones. Cabe aclarar que

este equipo tiene instalado muchas librerías, las cuales permiten ejecutar las aplicaciones sin problemas.

Al realizar una copia del proyecto para continuar el trabajo en casa, se presenta el problema al momento de intentar correr el proyecto, en el nuevo equipo el proyecto genera errores de librerías que anteriormente no presentaba. Al tratar de corregir este detalle llevo bastante tiempo. Entre la investigación previa que se realizó se optó por trabajar con un contenedor, de esta manera fue más sencillo llevar a cabo el trabajo.

El siguiente problema que se presenta es que el equipo prestado por la Institución tenía el disco duro a más del 80% por lo que no facilitaba la ejecución de una máquina virtual. Para llevar a cabo se optó por utilizar Docker sobre Windows a través de la virtualización WLS (Windows Subsystem for Linux), lo que permite compartir el kernel de Linux con Windows.[25]

WLS permite correr un sistema operativo Linux, en este caso se ha utilizado Ubuntu, es posible que bajo este esquema se presente un poco de lentitud, pero para el ambiente de desarrollo trabaja muy bien, no es recomendable utilizarlo en un ambiente de producción. Esto debido a que los recursos que se consumen son mayores a los que se tiene en un ambiente en producción.

En la Figura 2.20 se observan los pasos para poder crear el contenedor que fue utilizado durante el desarrollo del aplicativo móvil.

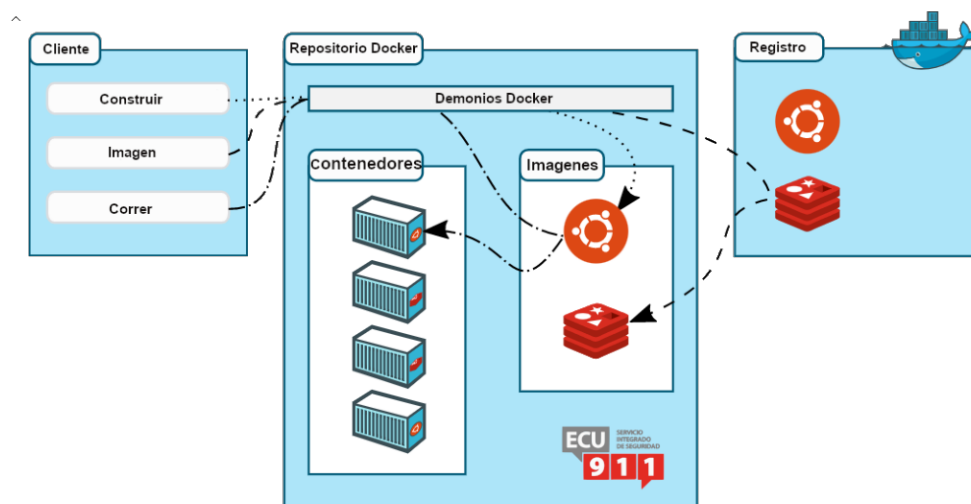


Figura 2.20 Estructura de Docker utilizada en la Institución ECU911.

Para la instalación de Docker fue necesario el uso exclusivo de herramientas para Windows, ya que el equipo proporcionado solo contaba con dicho sistema operativo y poner a correr una máquina virtual empeoraba las condiciones del equipo.

Durante el proyecto se utilizan las siguientes herramientas:

- WSL.
- Docker Desktop para Windows.
- Ubuntu (Máquina virtual que se ejecuta con WLS).

Que nos permite levantar el servicio de Docker, para crear el contenedor se debe contar con los siguientes requisitos:

- Dockerfile.
- Imagen.
- Contenedor.

Para evitar el conflicto con las librerías que puede faltar durante el desarrollo del prototipo de aplicativo móvil se utiliza con una imagen de Docker, que se encontraba almacenada en el repositorio, lo que ayuda también a la hora de ahorrar recursos del sistema. En la Figura 2.21 observamos la imagen que se puso en funcionamiento.

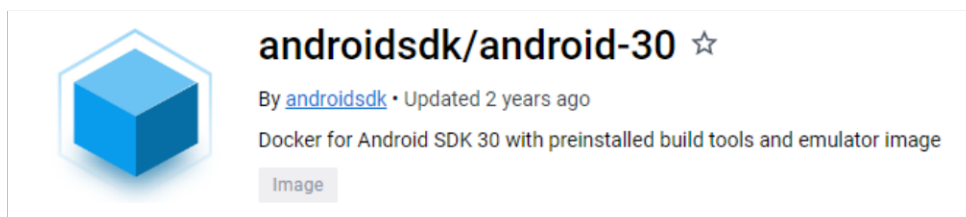


Figura 2.21 Imagen de Android Studio almacenada en el repositorio de Docker

Se puede crear una imagen personalizada del entorno que se quiere trabajar, como también permite hacer uso de imágenes existentes lo que es recomendable esto debido a que usuarios de Docker ya corrigieron los problemas previos lo que facilita su uso.

Para hacer uso de la imagen lo primero que se debe realizar es ejecutar Docker Desktop, esta aplicación es utilizada para tener Docker en Windows. Al abrir la aplicación, revisará las imágenes que tiene descargadas y ejecuta los contenedores. Cuenta con un entorno bastante simple y preciso en mostrar lo necesario.

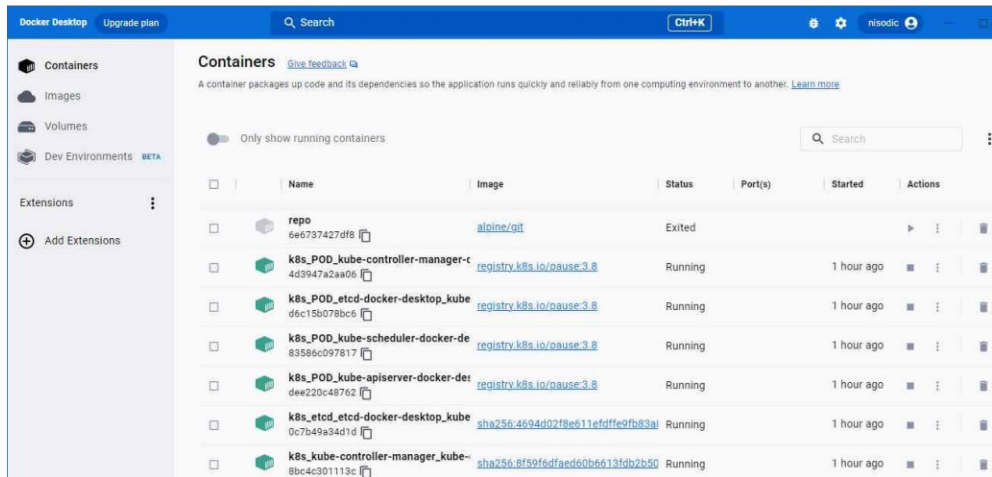


Figura 2.22 Entorno de Docker Desktop

La parte gráfica siempre es más amigable y muestra los contenedores que se encuentran corriendo de acuerdo con lo mostrado en la Figura 2.22, para hacer uso de la imagen de Android Studio verificar que Docker se encuentre correctamente ejecutándose, esto lo validamos en la máquina virtual Ubuntu. Esta máquina virtual se encuentra trabajando dentro del entorno de Windows, permite trabajar como si se tratase de una máquina virtual ejecutándose en un entorno virtual. En la Figura 2.23 se observa los comandos que son aceptados por el sistema operativo Linux.

```

nisodic@aragon:~$ docker run postgres
Unable to find image 'postgres:latest' locally
docker: Error response from daemon: Get "https://registry-1.docker.io/v2/": dialing registry-1.docker.io:443 with direct connection: connecting to 34.205.13.154:443: dial tcp 34.205.13.154:443: i/o timeout.
See 'docker run --help'.
nisodic@aragon:~$ docker

Usage: docker [OPTIONS] COMMAND

A self-sufficient runtime for containers

Options:
  --config string      Location of client config files (default "/home/nisodic/.docker")
  -c, --context string  Name of the context to use to connect to the daemon (overrides DOCKER_HOST env var and default context set with "docker context use")
  -D, --debug           Enable debug mode
  -H, --host list       Daemon socket(s) to connect to
  -l, --log-level string Set the logging level ("debug"|"info"|"warn"|"error"|"fatal") (default "info")
  --tls                Use TLS; implied by --tlsverify
  --tlscacert string   Trust certs signed only by this CA (default "/home/nisodic/.docker/ca.pem")
  --tlscert string     Path to TLS certificate file (default "/home/nisodic/.docker/cert.pem")
  --tlskey string      Path to TLS key file (default "/home/nisodic/.docker/key.pem")
  --tlsverify          Use TLS and verify the remote
  -v, --version        Print version information and quit

Management Commands:
  builder      Manage builds
  buildx*     Docker Buildx (Docker Inc., v0.10.0)
  compose*    Docker Compose (Docker Inc., v2.15.1)
  config      Manage Docker configs
  container   Manage containers
  context     Manage contexts
  dev*       Docker Dev Environments (Docker Inc., v0.8.5)
  extension*  Manages Docker extensions (Docker Inc., v0.2.17)
  image       Manage images
  manifest    Manage Docker image manifests and manifest lists
  network     Manage networks
  node        Manage Swarm nodes
  plugin      Manage plugins
  sbom*      View the packaged-based Software Bill Of Materials (SBOM) for an image (Anchore Inc., 0.6.0)
  scan*      Docker Scan (Docker Inc., v0.23.0)
  secret     Manage Docker secrets
  service     Manage services
  stack      Manage Docker stacks

```

Figura 2.23 Verificando que Docker este ejecutándose correctamente.

Para descargar la imagen se ejecuta la siguiente línea de comandos:

docker pull androidsdk/android-30:latest

En la línea de comando se indica lo siguiente:

- docker – busca en el repositorio.
- pull – descarga la imagen almacenada en el repositorio.
- androidsdk/android-30:latest – implica la imagen y la versión.

Terminado el proceso se tiene el resultado que se muestra en la imagen del contenedor, muestra el mensaje de completado tal como se observa en la Figura 2.24.

```
nisodic@aragonr:~$ docker pull androidsdk/android-30:latest
latest: Pulling from androidsdk/android-30
25fa05cd42bd: Pull complete
c18a9fab2c3d: Pull complete
b979512aa75a: Pull complete
b711bf444941: Pull complete
d936739390bf: Pull complete
f98256d063b4: Pull complete
537b3d41d622: Pull complete
ae6004d245fe: Pull complete
220687841428: Pull complete
efe0705869f0: Pull complete
e157c3789c08: Pull complete
6ed9f924bbea: Pull complete
Digest: sha256:77596f360975e139d146ee0f6f0b2760bbbbaa0b698bde5e446194a9c348e6c19
Status: Downloaded newer image for androidsdk/android-30:latest
docker.io/androidsdk/android-30:latest
```

Figura 2.24 Resultado de ejecutar la línea de código antes comentada.

Para ejecutar la imagen descargada se ejecuta la siguiente línea de comando:

docker run 719db0146c62

Se debe ejecutar la imagen previamente descargada, una vez que se ejecuta podemos notar que la imagen cambia de estado de “inactivo” a “In use” como se puede observar en la Figura 2.25.

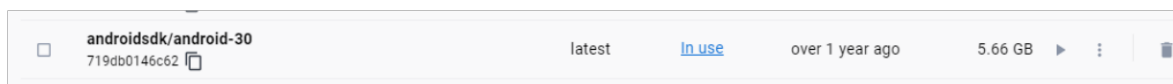


Figura 2.25 Estado de la imagen Android Studio

Una vez en ejecución podemos hacer uso de la imagen y trabajar, cabe aclarar que no se puede guardar información sobre la imagen descargada, cualquier información extra debe ser almacenada de forma local o en el repositorio que se tenga asignado.

Prototipo aplicativo móvil Eventos Relevantes

Una vez explicado en funcionamiento de las herramientas DevOps que más se adaptaron al entorno de trabajo se hablará sobre el desarrollado del prototipo del aplicativo móvil. Iniciando con las siguientes tareas:

- Creación del entorno gráfico, primera pantalla de inicio, en un inicio se consideró tres factores para la autenticación de usuarios, siendo estos los siguientes:
 - Correo electrónico.
 - Password.
 - Obtención del número de teléfono desde la Sim Card.
- Debido a problemas con el código y los permisos que el usuario debe aceptar se descartó esta opción dejando las siguientes opciones para la autenticación:
 - Correo electrónico.
 - Password.
- Continuando con el proceso del desarrollo del aplicativo móvil se continuo con la configuración del layout de acceso. En la Figura 2.25 se observa el diseño básico (sketchs) que utiliza la aplicación. Para no dejar todo el trabajo a la capa de conexión los EditText cuentan con validación de formato para el correo y un mensaje de información que el password debe ser mayor a 5 caracteres. Las figuras de esta información se encuentran detalladas en el Anexo II

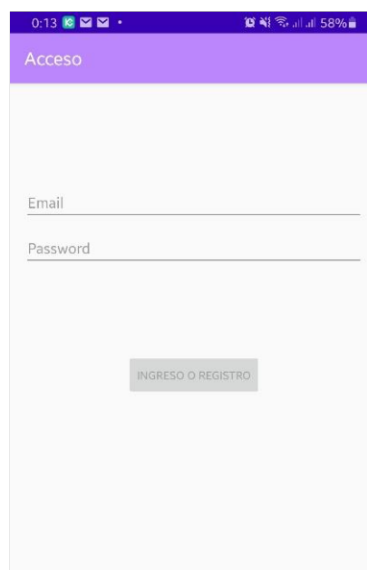


Figura 2.25 Prototipo de aplicativo Móvil, layout de acceso versión 1.

Para la versión final del prototipo de aplicativo móvil se han incorporado nuevos elementos, que mejoran la visualización como se puede observar en la Figura 2.26, se han cambiado los colores de acuerdo a la personalización de la Institución ECU911.

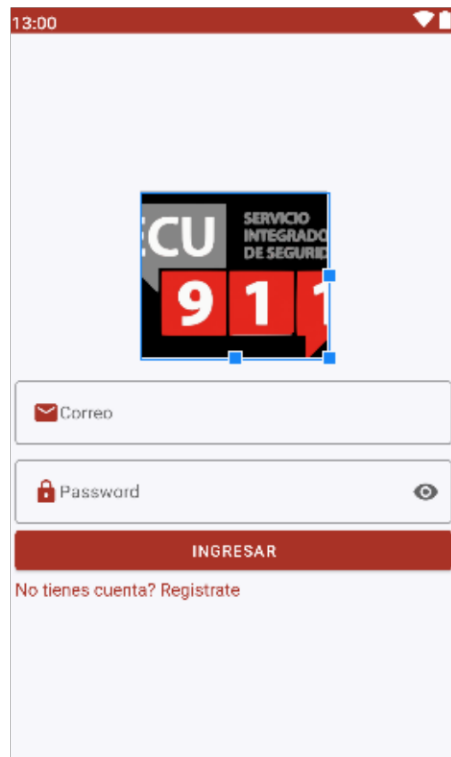


Figura 2.26 Versión final del prototipo de aplicativo móvil.

Adicional se incluye un formulario de registro, después del cambio de director de Proyecto. Este requerimiento no forma parte de lo indicado en los requisitos iniciales.

Para cumplir con los requisitos planteados por parte de la Institución ECU911 el prototipo de aplicativo móvil debe mantener una conexión con el Aplicativo Web para poder acceder a los eventos relevante de donde se obtendrá la información que debe ser presentada. Para poder realizar las pruebas se implementa un API en Node.js que permite interactuar con la base de datos que se encuentra en el gestor de PostgreSQL.

Este entorno se realizó para poder hacer las pruebas debido a que el entorno de producción no es factible interrumpir el servicio.

El aplicativo Web se encargará de enviar la información hacia el prototipo de aplicativo móvil, cabe aclarar que el prototipo solo va a estar disponible para usuarios que la Institución ECU911 ha decidido permitir el acceso.

En la Figura 2.27 se observa el proceso que debe seguir el usuario para tener acceso a la información de eventos relevante. En el ANEXO III se encuentran más diagramas de funcionalidad.

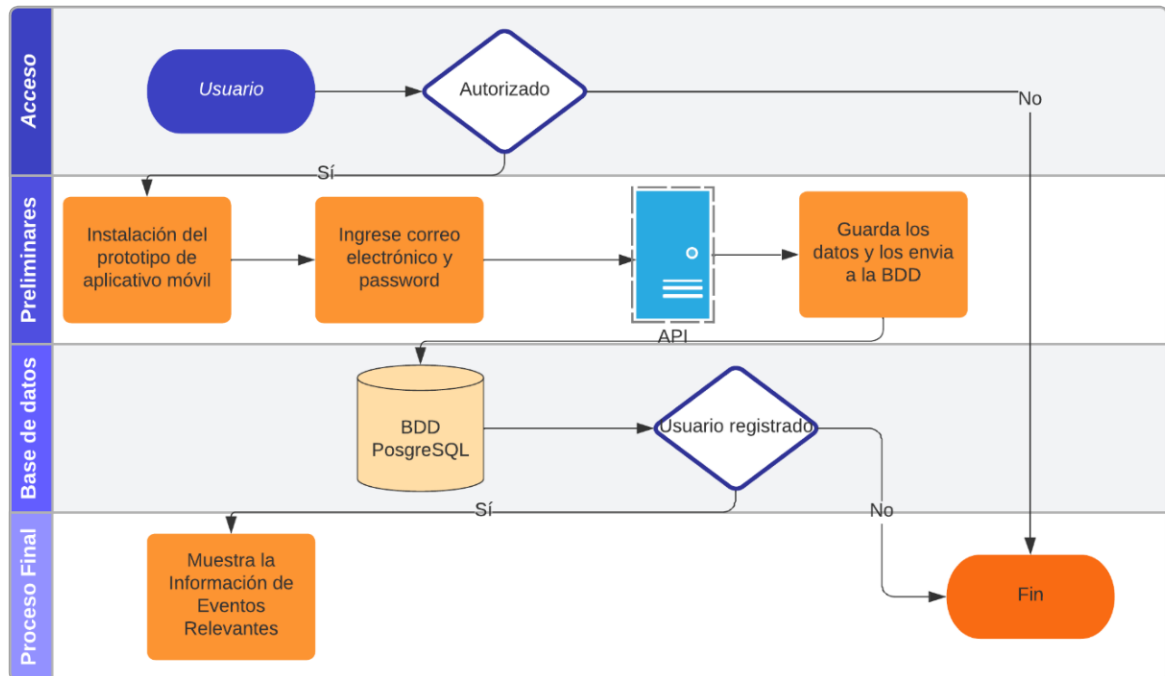


Figura 2.27 Diagrama de acceso de usuarios

Como se mencionó anteriormente el prototipo de aplicativo móvil está desarrollado en lenguaje Kotlin como de muestra en la Figura 2.28, el fragmento de código se refiere al EditText que se utiliza para el ingreso del password, este EditText cuenta con una configuración diferente al resto de EditText porque debe ocultar la información que se ingresa. Pero si el usuario requiere ver lo que escribe también está la opción de mostrar texto que puede ser observada en la Figura 2.26.


```
<com.google.android.material.textfield.TextInputLayout
    android:id="@+id/filledTextFieldPassword"
    style="@style/Widget.MaterialComponents.TextInputLayout.OutlinedBox"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Password"
    android:paddingTop="8dp"
    app:endIconMode="password_toggle">

    <com.google.android.material.textfield.TextInputEditText
        android:id="@+id/edit_password"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:drawableTint="@color/back_color_button"
        android:drawableStart="@drawable/baseline_lock"
        android:imeOptions="actionDone"
        android:inputType="textPassword" />

</com.google.android.material.textfield.TextInputLayout>
```

Figura 2.28 Código para el EditText del password.

Para la revisión del código se encuentra detallada la información en el ANEXO II.

3. RESULTADOS, CONCLUSIONES Y RECOMENDACIONES

3.1 Resultados del Uso de Herramientas DevOps

Es un tema bastante interesante, a pesar de ser herramientas que ya tienen tiempo en producción no todos los usuarios conocen del tema, y en muchos de los casos no han experimentado el uso de las mismas. Algunos de los temas fueron tratados durante la época de estudiantes, pero solamente como una introducción, no se ha llegado a trabajar con las herramientas.

Para tener una mejor apreciación se ha encuestado a 10 personas consultándoles sobre el uso de las Herramientas DevOps y sobre el diseño del prototipo del aplicativo móvil que se desarrolló para ser usado en la Institución ECU911.

Se aclara que las personas encuestadas pertenecen al departamento de Tecnología de Información y conocen o han escuchado sobre el tema a tratar.

En el ANEXO I se observa la información proveniente de la base de datos y lo que es mostrado en el prototipo de aplicativo móvil.

Como primera observación que se tiene es que dentro del estándar que maneja el departamento de Tecnología de Información no tienen claro el concepto de las herramientas DevOps, a continuación, se detallan las preguntas y los resultados que se obtuvieron:

1) Conoce sobre las Herramientas DevOps

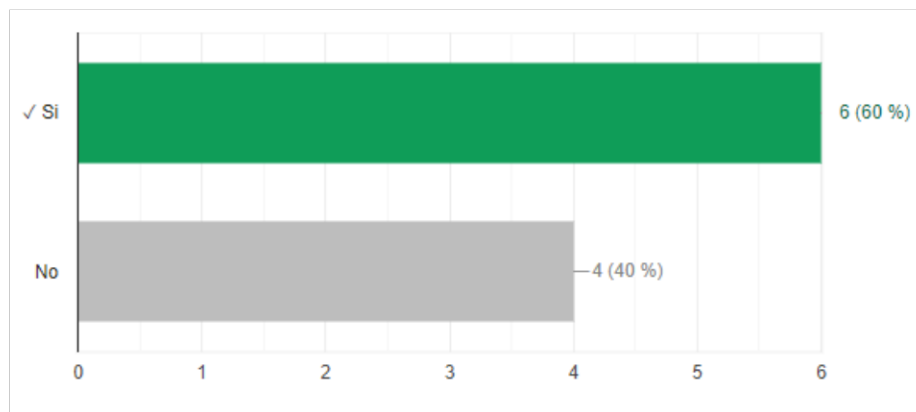


Figura 3.1 Resultados de la primera pregunta.

La mayoría conoce sobre las herramientas DevOps, en comentarios saben usar algunas, pero no las han llegado a implementar dentro de la Institución.

2) Si la respuesta a la primera pregunta fue “Sí”, marque las casillas de las herramientas que ha escuchado o trabajado.

De las 6 personas que respondieron afirmativamente se les planteó la opción de marcar las herramientas que conocen, obteniendo el siguiente resultado:

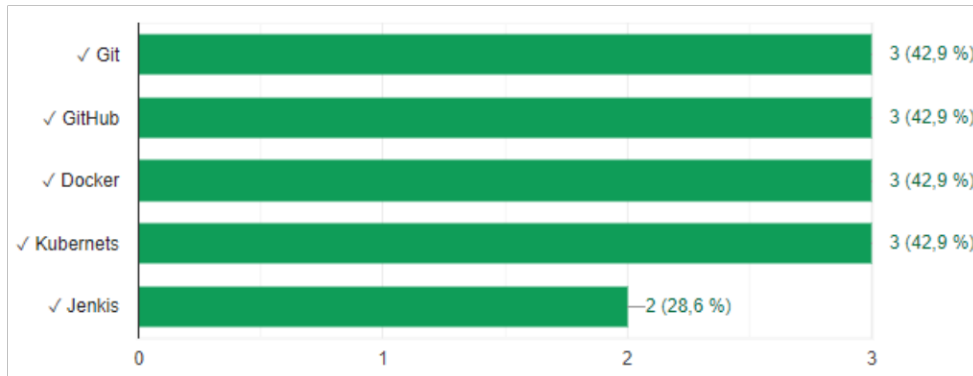


Figura 3.2 Resultados de la segunda pregunta.

3) Si la respuesta a la primera pregunta fue “No”, puede escribir qué herramientas maneja para guardar los repositorios durante el desarrollo de software.

Al ser una pregunta abierta se tuvo las siguientes respuestas:

- Google Drive.
- SharePoint.
- Almacenamiento local.

Dentro de las respuestas se observa que trabajan con almacenamiento en la nube, lo que también les permite llevar un control de versiones, no siendo el más adecuado ya que este tipo de herramientas es apto para trabajar con documentos.

4) ¿Aceptaría asistir a una conferencia para conocer el uso de las herramientas DevOps?

La mayor parte del personal está dispuesto a aprender sobre el tema planteado, ya que este puede servir para futuros proyectos de desarrollo de software, lo que se observa es un poco de resistencia al uso del sistema operativo Linux, que son donde corren la mayor parte de las herramientas DevOps.

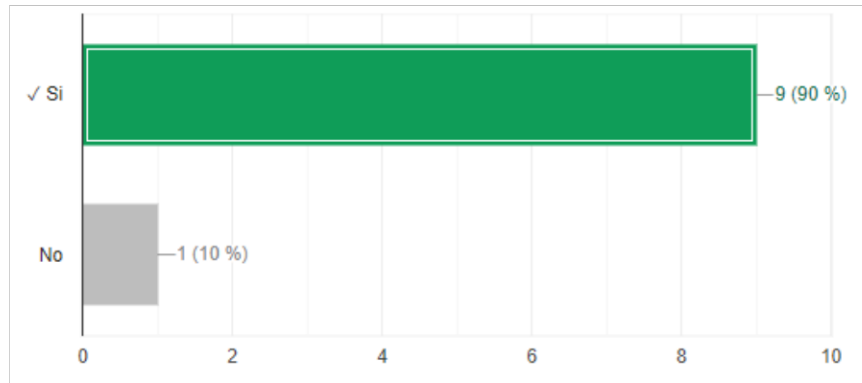


Figura 3.3 Resultado de la pregunta cuatro.

5) ¿Qué fallas considera que tiene el prototipo de aplicativo móvil realizado para el ECU911? Marque las que considere

Considerando los requisitos se planteó la pregunta y después de probar el prototipo del aplicativo móvil se obtuvieron estos resultados:

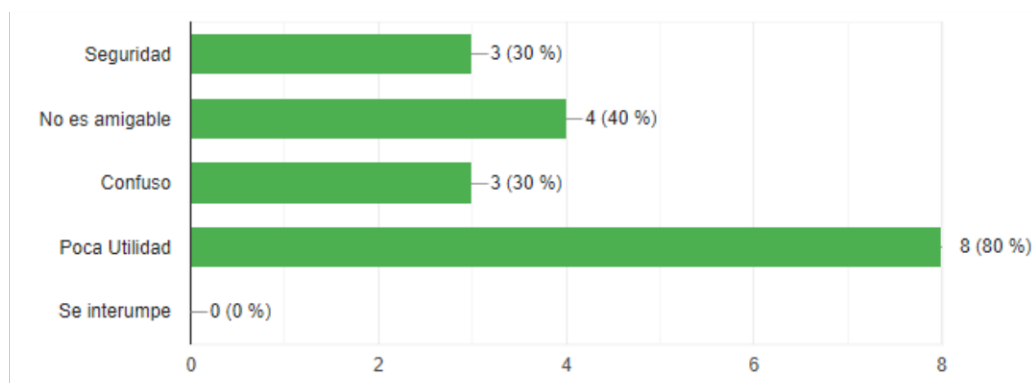


Figura 3.4 Resultados de la pregunta cinco.

Considerando que las personas encuestadas no consideraron los requisitos iniciales pues desconocían del alcance, adicional con el cambio de director de proyecto no se entregó la información correcta, por tal motivo se observa que los resultados en la opción que marca “Poca Utilidad”, se elevó y no cumple con las expectativas que consideraban para este proyecto.

6) Recomendaciones para mejorar el prototipo de aplicativo móvil.

Considerando que los usuarios encuestados y que probaron el prototipo de aplicativo móvil no conocían los requerimientos iniciales; estas fueron las recomendaciones:

- Mejorar la apariencia.

- Implementar la recepción de imágenes.
- Aumentar la cantidad de usuarios que la deben usar.
- Modificar los mensajes recibidos.
- No debe recibir solo texto.

3.2 Conclusiones

Se establece una nueva forma de trabajo dentro de la institución ECU911, esto debido a que anteriormente no han trabajado con herramientas DevOps, cuentan con un procedimiento diferente y sus normas establecidas con anterioridad. Para llevar a cabo este prototipo de aplicativo móvil se contó con el apoyo del personal, brindando la información necesaria para llegar al objetivo deseado. Dicho esto, se obtienen las siguientes conclusiones:

- Es importante que toda reunión que se lleve a cabo debe estar respaldada por un acta de la reunión que se llevó a cabo, en donde se detalle los puntos principales a cumplir durante el desarrollo del proyecto. Esto evita tener complicaciones al momento de realizar la entrega y aceptación por parte del personal que recibe la información. Además, que se cuenta con un respaldo en caso de que el personal sea cambiado o no conozca mayores detalles del proyecto.
- Las herramientas DevOps son de gran utilidad durante el desarrollo del proyecto permitiendo crear repositorios tanto de documentos, código y paquetes.
- Al trabajar con Kanban para gestionar el proyecto se pudo manejar de mejor manera los cuellos de botella, ya que se podía trabajar sobre otras tareas lo que permitía el avance del proyecto, disminuyendo de esta manera los tiempos muertos.
- La herramienta Docker se ajusta a las necesidades que requiere cumplir la institución ECU911 para solventar el problema de portabilidad, esto debido a que puede existir cambio de computadores para continuar con el desarrollo del software.
- En cuanto al manejo del repositorio de versiones es mejor mantenerlo localmente esto porque se trabaja con información que no debe ser de acceso público. Para poner controlar las versiones de software es necesario llevar actas de las reuniones con los cambios solicitados, y en lo posible trabajar en una nube privada.

- Para futuros proyectos de desarrollo de software y sobre todo en la etapa inicial se puede trabajar con un repositorio de versiones de acceso público, de esta manera se promueve la colaboración entre el personal implicado en el desarrollo del proyecto.
- Aplicar herramientas DevOps en un entorno donde ya tienen establecido su procedimiento es complicado que el personal lo empiece a usar, esto debido a dos factores:
 - No pueden alterar su procedimiento debido a la burocracia que ya se encuentra establecida en el entorno.
 - El personal no siempre la puede aplicar debido a la rotación que existe, por lo que mantienen sus normas iniciales.
- Para la Institución ECU911 se deja una propuesta de uso para herramientas DevOps que los pueden llegar a usar en futuros proyectos, donde pueden dar la apertura para el uso de nuevos procesos durante el desarrollo de software. Esta propuesta se encuentra detallada en el Capítulo 2.5.
- Finalmente se propone que las herramientas DevOps que son de acceso gratuito tengan mayor apertura para facilitar los procesos y disminuir la burocracia existente.

3.3 Recomendaciones

Dentro de los entornos de trabajo se debe manejar ambientes que ayuden con la ejecución de los proyectos, como lo son un ambiente de trabajo que permita desarrollar el proyecto, así como con un espacio físico que permita desarrollar las ideas de trabajo.

Contar con el espacio y tiempo para optar por nuevas tecnologías pueden ayudar y mejorar el rendimiento del entorno de trabajo. Para lo cual se establece las siguientes recomendaciones:

- Trabajar en un entorno cómodo con un mobiliario ergonómico y adecuado para cuidar la salud del personal.
- Para cualquier tipo de proyecto es necesario contar con respaldo de lo que se está solicitando al iniciar el proyecto. Con esta documentación se puede considerar si se llega a cumplir con todas las tareas planteadas.

- Tomar el tiempo para conocer nuevas herramientas que se desarrollan en la actualidad, se puede optar por cursos gratuitos o videos que se encuentran en la plataforma de YouTube para aprender más sobre las herramientas DevOps.
- Durante el desarrollo del prototipo de aplicativo móvil se trabajó con Kubernetes como una opción de herramienta DevOps, esta herramienta fue descartada y no es de utilidad cuando se trabaja con solo un proyecto. Por lo que su aplicación no es recomendable.
- Para la contratación del nuevo personal que ingrese a trabajar en la Institución ECU911 en el área de desarrollo, se debe considerar que cuente con el conocimiento para el manejo de servidores Linux, considerando que la mayor parte de herramientas DevOps es ejecutan en dicho sistema operativo.
- Para el desarrollo de software se debe contar con al menos dos personas que estén al tanto sobre lo que se está trabajando, en caso de ausencia se pueda continuar con el proyecto y no generar tiempos muertos.
- No es buena opción trabajar en una nube pública cuando se trabaja con información de alta prioridad, en ese caso es mejor hacer uso de una nube privada, manteniendo la información con la seguridad que amerita.
- Se debe llevar a cabo reuniones semanales, no muy extensas para conocer el avance de los proyectos, no solamente cuando se trate de desarrollo de software, este tipo de reuniones aplica para cualquier proyecto.
- Dentro de las instituciones públicas los procesos deberían ser precisos y evitar la burocracia, ya que esta causa retrasos en los procesos. Estos retrasos son los causantes que los proyectos no sean finalizados en el tiempo propuesto.
- Aprender el uso de una buena herramienta DevOps ayuda a agilizar los procesos sin importar lo difícil que estos parezcan. Al igual que implementar herramientas de forma paulatina sin provocar cambios bruscos dentro de los procesos establecidos.
- Es importante conocer que dentro de las Instituciones Públicas el personal puede ser removido de su cargo lo que dificulta la aprobación de los proyectos en curso, por lo que es necesario contar con la documentación necesaria para que el próximo funcionario revise y apruebe el proyecto.

4. REFERENCIAS BIBLIOGRÁFICAS

- [1] K.K. Ambily, *Azure DevOps for Web Developers*, Barkeley, CA: Apress: Imprint Apress, 2020
- [2] "Tribalyte Techniligies", Internet: <https://tech.tribalyte.eu/blog-introduccion-devops>.
- [3] A. Mishra, Z. Otaiwi, "Devops and Software Quality: A systematic mapping", *Computer Sience Review*, pp. 1-14, oct.2020
- [4] LE. Lwakatare, T Karvonen, T Sauvola, "Devops in practice: A multiple case study of five company", *Information and Software Technology*, vol. 114, pp 217-230, oct. 2019
- [5] "Smartnodus", Internet: <https://www.smartnodus.cl/integracion-continua/>
- [6] Palermo, Jeffrey, *.Net DevOps for Azure*, Berkeley, CA: Apress: Imprint Apress, 2019
- [7] "F5 Glossary", Internet: https://www.f5.com/es_es/services/resources/glossary/what-is-infrastructure-as-code
- [8] WP Luz, G Pinto, R Bonifacio, "Adopting Devops in the real world: A theory, a model, and a case study", *Journal of System and Software*, vol. 157, nov. 2019
- [9] "Atlassian Bitbivket", Internet: <https://www.atlassian.com/es/git/tutorials/what-is-version-control>
- [10] GitHub, Inc. "GitHub Docs". Internet: <https://docs.github.com/en/get-started/quickstart/hello-world>, 08-2022
- [11] Jenkins, "Jenkins Handbook". Internet: <https://www.jenkins.io/doc/book/>
- [12] Red Hat "Ansible: conceptos básicos". Internet: <https://www.redhat.com/es/topics/automation/learning-ansible-tutorial>, 21-06-2022
- [13] "Docker Docs", Internet: <https://docs.docker.com/>, 2023
- [14] "Kubernetes", Internet: <https://kubernetes.io/es/docs/home/>, April 29, 2022
- [15] "Nagios", Internet: <https://www.nagios.org/documentation/>
- [16] M. Senapathi, J. Buchan, H. Osman, "DevOps Capabilities, and Challenges: Insights from a case study", *EASE18: Proceedings of the 22nd International Conference on Evaluation and Assessment in Software Engeneering 2018*, pp57-67, jun. 2018
- [17] A. Wiedeman, M. Wiesche, "Are you ready for Devops? Requiered skill set for Devops teams", *AIS Electronic Library (AISeL)*, 2018
- [18] "Chisel blog", Internet: <https://chisellabs.com/blog/what-is-a-kanban-board-elements-how-to-scrum/>
- [19] MO Ahmad, D Dennehy, K Conboy, M Oivo, "Kanban in software engineering: A systematic mapping study", *The Journal of System and Software*, vol. 137, pp 96-113, mar 2018

[19] W Zayat, O Senvar, "Framework study for agile software development via scrum and kanban", *International Journal of Innovation and Technology Management*, vol. 17, may. 2020

[20] ZM Rodríguez, LDP Rodríguez, JCG Suarez, "Arquitectura basada en Microservicios y DevOps para una ingeniería de software continua. Internet: <https://www.redalyc.org/journal/816/81665362014/html/>", jan. 13, 2020[agu. 10, 2020]

[21] "Android Studio", Internet: <https://developer.android.com/studio>,

[22] EDteam, Internet: <https://ed.team/cursos/kotlin>

[23] "Node.js", Internet: <https://nodejs.org/api/>

[24] "Servicio Integrado de Seguridad ECU 911" Internet: <https://www.ecu911.gob.ec/>

[25] Docker Inc, "Docker Docs".Internet: <https://docs.docker.com/desktop/windows/wsl/>

5. ANEXOS

ANEXO I. Ejemplo de ejecución del prototipo de aplicativo móvil

ANEXO II. Código de la aplicación.

ANEXO III. Diagramas

ANEXO IV. Acta de Reuniones

ANEXO I

Para el funcionamiento del prototipo de aplicativo móvil se crea la base de datos, la información a ser mostrada debe llegar desde un aplicativo web que se encuentra en producción en el ECU911, por lo que la información que se muestra en la pantalla del layout eventos debe ser ingresada de forma manual en la base de datos en la tabla evento, como se observa en la Figura I. El prototipo de aplicativo móvil al actualizar la información muestra lo que es ingresado en la base de datos.

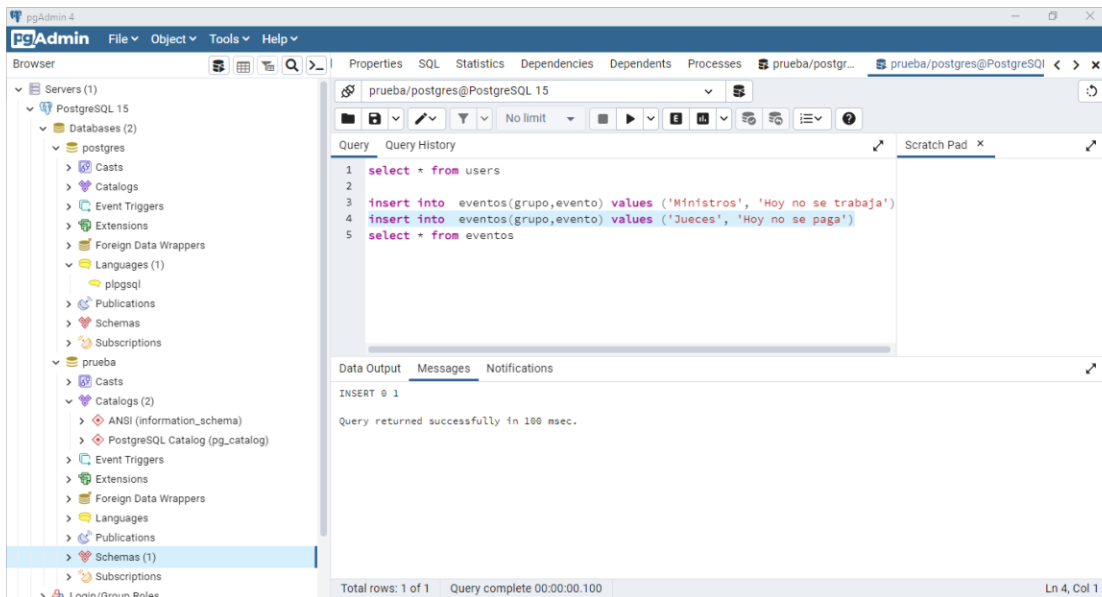


Figura I. Ingreso de información en la tabla evento

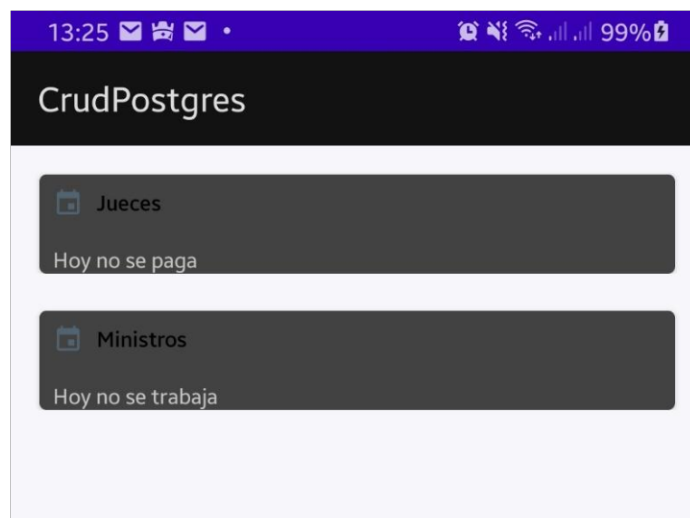


Figura II. Layout item_events muestra la información proveniente de la base de datos

ANEXO II

Para tener a al código de la aplicación se ha creado una carpeta en OneDrive donde está almacenada toda la información referente a:

- Prototipo de aplicativo Móvil.
- Api en Node.js.
- Scripts PostgreSQL.

Link de acceso a la carpeta compartida:

[II](#)

ANEXO III

En el diagrama observado en la Figura III muestran las etapas por las cuales debe pasar el prototipo de aplicativo móvil para obtener una versión final.

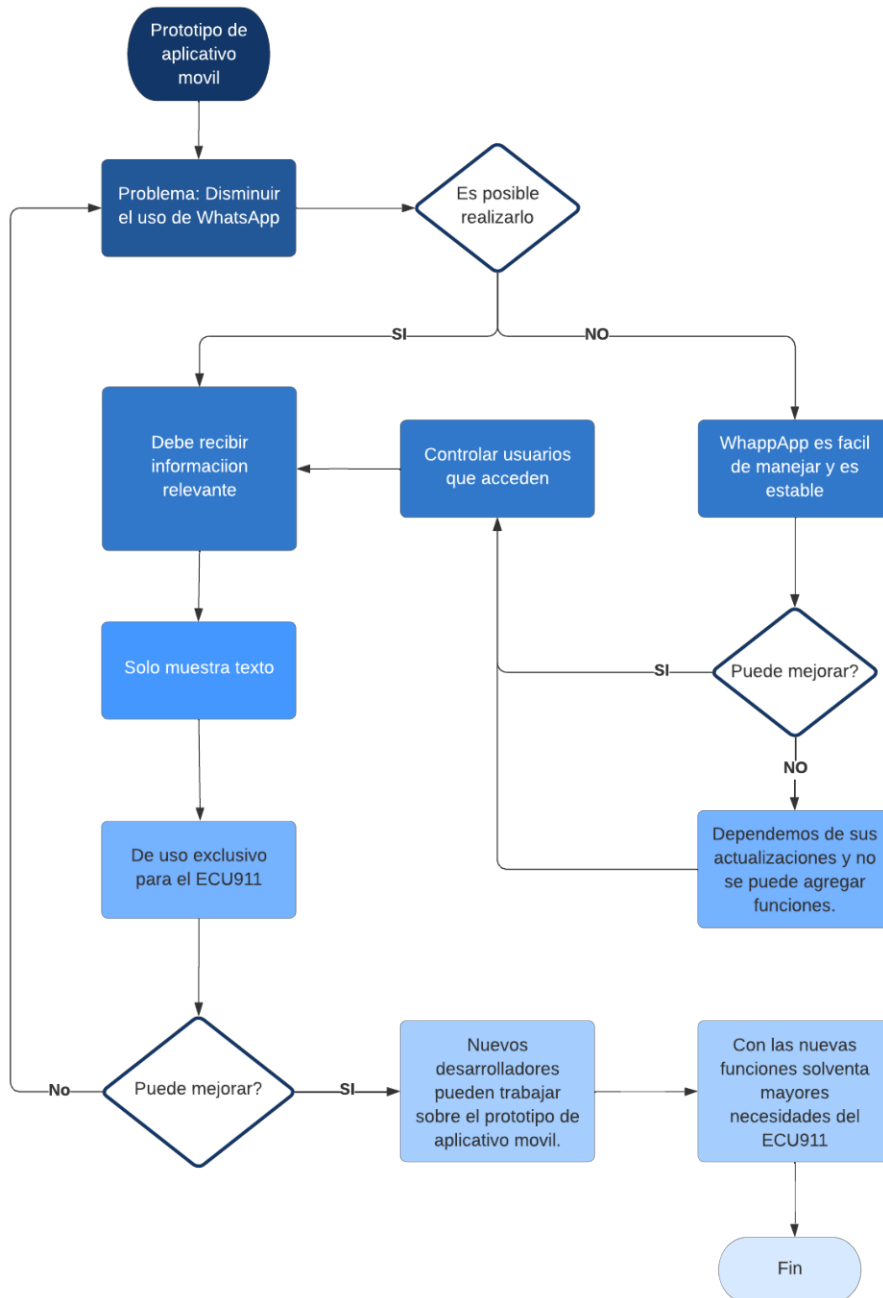


Figura III. Diagrama de funcionalidad del prototipo de aplicativo móvil

En la Figura IV, se tiene la verificación de la longitud de caracteres del password, que debe cumplir con una longitud mayor a 5 caracteres.

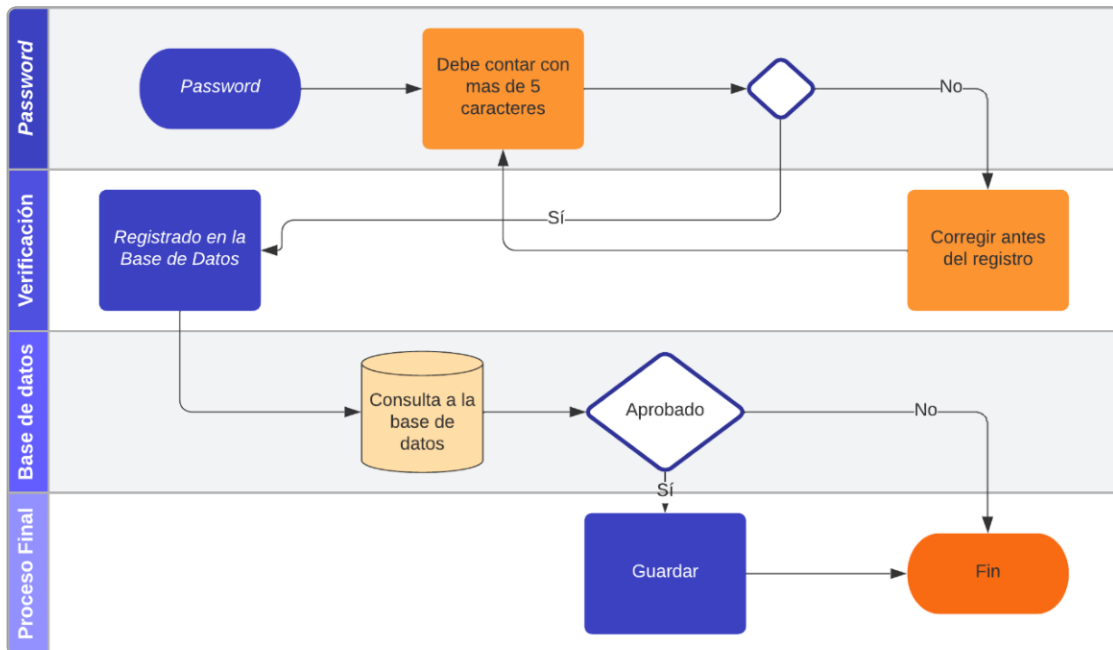


Figura IV. Diagrama de verificación de la longitud de caracteres.

En la Figura V. se observa el flujo que debe seguir la información para registrar un nuevo usuario.

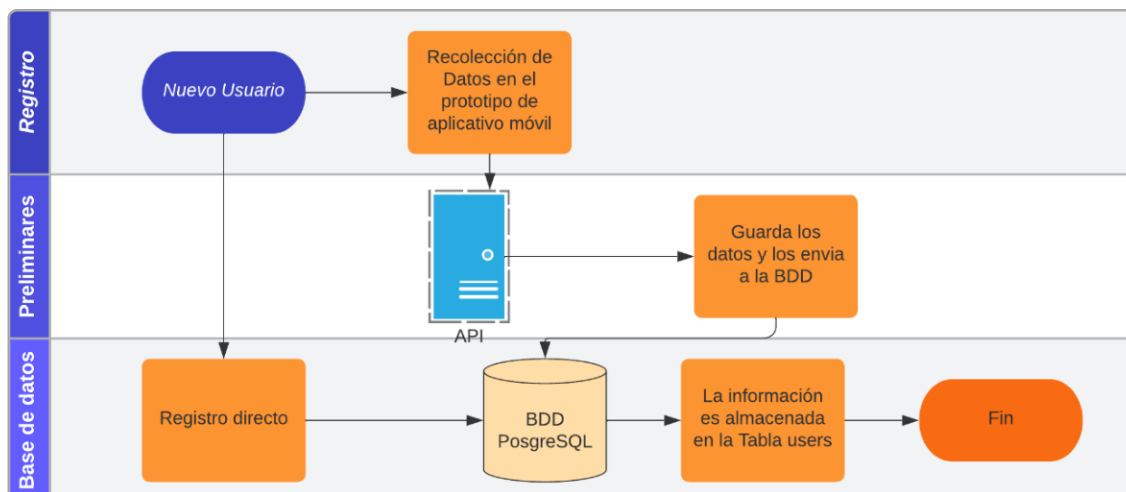


Figura V. Registro de nuevo usuario

ANEXO IV

PRIMERA REUNION

Quito 20 de octubre de 2022

Para llevar a cabo el Proyecto se pacta la reunión en las instalaciones del ECU911 en Quito el día 20 de octubre de 2022 a las 10:00, las personas que asisten a la reunión son:

- Diego Ureña - Representante de ECU911
- Xavier Calderón - Docente de la EPN
- Andrés Quishpe - Estudiante

Durante la reunión se definen las necesidades que requiere cubrir la institución este tema es explicado por parte de Diego Ureña, donde explica lo siguiente:

- Requiere un sistema para gestionar información de “Eventos Relevantes” donde se cree grupos y se muestre información en un aplicativo móvil.
- El acceso a esta información solamente lo tendrán los usuarios registrados en una Base de Datos bajo consideraciones del ECU911.
- Información es de alta confidencialidad.
- La nueva herramienta debe sustituir a los grupos de WhatsApp que se manejan.
- Desde el servicio web se debe administrar la información.
- El aplicativo móvil mostrará la información de Eventos Relevantes solamente el texto, mismo que actualmente es enviado desde la aplicación en producción. Desde donde se envía a los grupos de WhatsApp.
- La información recibida en el aplicativo móvil no debe ser modificada.
- Para establecer la conexión entre la aplicación existente y la aplicación móvil se requiere un aplicativo web para establecer la comunicación.

Definidos los requisitos preliminares, se acepta la propuesta para iniciar a trabajar sobre el proyecto.

La propuesta es aceptada por Andrés Quishpe, quien estará a cargo del diseño del aplicativo móvil, el servicio Web está a cargo de Rommel Correa.

SEGUNDA REUNION

Quito 14 de noviembre de 2022

Se establece una nueva reunión los asistentes son las siguientes personas:

- Mónica Nicolalde - Personal del ECU911
- Diego Ureña - Personal del ECU911
- Andrés Quishpe - Estudiante de la EPN

En esta nueva reunión Diego Ureña establece los requisitos que debe cumplir el aplicativo móvil y el servicio web.

Aplicativo móvil:

- Acceso solamente a usuarios autorizados.
- Debe existir un control de acceso para evitar la fuga de información.
- La aplicación debe estar disponible tanto para Android como para IOS.
- Debe mostrar la información de Eventos Relevantes.
- Esta información no debe ser modificada por los usuarios de la aplicación.
- Se desea que la aplicación reemplace a los grupos de WhatsApp que actualmente utilizan para compartir la información.

Después de escuchar los requisitos por parte de Diego Ureña, se define lo que es factible llevar a cabo en el aplicativo móvil.

- Acceso solamente a usuarios autorizados
 - Diego Ureña define los usuarios que tienen acceso a la información.
- Control de acceso
 - Solamente el personal que cumpla con los requisitos podrá visualizar la información
- Disponible para Android y IOS
 - Se establece trabajar con un lenguaje híbrido, esto debido a que se presentan dos obstáculos
 - El ECU911 no cuenta con un entorno MAC para desarrollar el aplicativo
 - Andrés Quishpe no cuenta con el conocimiento para desarrollar el aplicativo móvil para IOS.
- Mostrar la información de Eventos Relevantes, texto plano

- Se realizarán pruebas para verificar si es posible reemplazar a WhatsApp.

Terminada la reunión se procede a iniciar con el trabajo, considerando los requisitos antes mencionados.

Como una medida de seguridad adicional se sugiere que el control de acceso también se lleve a cabo desde el terminal móvil donde se encuentre instalado la aplicación. Es decir, acceder a la información del dispositivo móvil y obtener el número telefónico y realizar una verificación directamente a la BDD donde se encuentra almacenada la información.

TERCERA REUNIÓN

Quito, 4 de enero de 2023

Para esta reunión asisten las siguientes personas:

- Diego Ureña - Personal ECU911
- Mónica Nicolalde - Personal ECU911
- Rommel Correa - Estudiante EPN
- Andrés Quishpe - Estudiante EPN

Durante la reunión se trata los siguientes puntos

Aplicativo Móvil

- Se descarta el tratar de obtener el número de teléfono desde la SIM del usuario. Esto debido a que presenta los siguientes problemas:
 - Invasión a la privacidad del usuario.
 - Requiere consentimiento del usuario para acceder a esta información.
 - A nivel de programación la obtención de esta información se encuentra restringida.
- El aplicativo móvil únicamente será presentado en Android, esto debido a que la aplicación híbrida requiere mayor tiempo de aprendizaje para obtener la compatibilidad en las dos plataformas.
- Los requerimientos anteriores se mantienen

Aceptados los requerimientos se finaliza la reunión.