

**Departamento de Ciencias de la
Computación(DCCO)
Carrera de Software**

**Curso de Aseguramiento de la calidad de
software**

Presentado por: Morales Marley, Pozo Steven, Ramírez
Erick, Torres Sebastián (Grupo 2)

Director: Ing. Ruíz Jenny

Ciudad: Quito

Fecha: 27 nov 2024

Índice

1. Introducción	3
2. Planteamiento del trabajo	3
2.1 Formulación del problema	3
2.2 Justificación	4
3. Sistema de Objetivos	4
3.1. Objetivo General	4
3.2. Objetivos Específicos (03)	4
4. Alcance	5
5. Marco Teórico	5
5.1 Metodología (Marco de trabajo 5W+2H)	7
6. Ideas a Defender	8
7. Resultados Esperados	9
8. Viabilidad	10
8.1 Humana	10
8.1.1 Tutor Empresarial	10
8.1.2 Tutor Académico	11
8.1.3 Estudiantes	11
8.2 Tecnológica	11
8.2.1 Hardware	11
8.2.2 Software	11
9. Planificación para el Cronograma:	11
10. Bibliografía	12
11. Anexos	14
12. Anexo I. Crono	14
13. Anexo II. Crono Matriz de identificación de requisitos	14
14. Anexo III. Historia de Usuario	14

1. Introducción

En la actualidad dentro del desarrollo del software, la calidad de un producto es fundamental para satisfacer las necesidades de los usuarios, ya que si los usuarios quedan insatisfechos del producto esto se refleja en la pérdida de confianza y costos adicionales después de su producción. Es por ello que el problema radica en la falta de un plan de pruebas de software que se adapte a las necesidades de cada proyecto

Para lograrlo, es esencial validar su funcionalidad mediante un proceso estructurado de pruebas de software. Estas pruebas no solo verifican el correcto funcionamiento de los módulos del sistema, sino que también aseguran que cada componente cumple con los requisitos establecidos. Mediante el uso de herramientas que automatizan y simplifican estos procesos, se busca optimizar la eficiencia y efectividad de las pruebas.

El objetivo principal de este proceso es documentar los resultados en un Plan de Pruebas, un documento clave que detalla las estrategias, casos de prueba y resultados obtenidos. Este plan servirá como guía para garantizar la calidad de la aplicación, identificando áreas de mejora y asegurando que la experiencia del usuario sea óptima.

2. Planteamiento del trabajo

2.1 Formulación del problema

En el desarrollo de una aplicación móvil para compartir experiencias y recuerdos relacionados con mascotas, garantizar la funcionalidad adecuada del sistema es un desafío clave. Para ello, es necesario implementar un enfoque estructurado de pruebas que abarque aspectos como la planificación, diseño y ejecución de casos de prueba. Sin embargo, la ausencia de un esquema bien definido puede dificultar el análisis y validación de resultados.

Para abordar esta problemática, se propone la creación de un Plan de Pruebas que detalle todas las actividades de validación realizadas. Este documento servirá

como guía para estructurar las pruebas de manera organizada, documentar los resultados obtenidos, y facilitar el análisis de estas, asegurando que la aplicación cumpla con los requisitos funcionales y no funcionales establecidos.

2.2 Justificación

El aplicar correctamente la automatización de pruebas dentro de un proyecto de desarrollo de software asegura que se omitan (en su mayoría) errores en ejecución, y se asegure de entregar un producto funcional de calidad, siguiendo estándares internacionales adecuados que nos proveen de lineamientos para cumplir con un plan de pruebas eficiente.

3. Sistema de Objetivos

3.1. Objetivo General

Desarrollar y documentar un Plan de Pruebas que permita validar la funcionalidad, confiabilidad y robustez de la aplicación móvil, asegurando que cumpla con las expectativas y necesidades del usuario final. Este plan debe garantizar que todas las funcionalidades del sistema operen de manera correcta y eficiente, alineándose con los estándares de calidad establecidos y optimizando la experiencia del usuario.

3.2. Objetivos Específicos (03)

- Diseñar y ejecutar un plan de pruebas estructurado estableciendo un programa detallado que abarque la planificación, diseño y ejecución de pruebas para verificar la funcionalidad del sistema.
- Registrar y analizar las incidencias encontradas documentando de manera precisa las fallas o errores identificados durante las pruebas, proporcionando un informe detallado para su posterior resolución
- Validar la integración y correcto funcionamiento del sistema asegurando que todas las funcionalidades y módulos de la aplicación trabajen de manera conjunta y sin inconvenientes, verificando su correcto acoplamiento y operatividad.

- Diseñar y establecer un plan de auditoría que abarque todos los aspectos críticos del sistema de gestión de calidad.
- Documentar sistemáticamente todas las incidencias de calidad identificadas durante la auditoría, clasificándolas según su severidad e impacto en la calidad del producto.

4. Alcance

- **Planificación de actividades:** Diseñar un diagrama de actividades que permita organizar y estructurar el desarrollo de las pruebas de software, asegurando un flujo de trabajo eficiente y alineado con los objetivos del proyecto.
- **Generación de casos de prueba:** Crear casos de prueba eficientes y detallados, enfocados en validar la funcionalidad de cada módulo del sistema y garantizando que el aplicativo móvil cumpla con los requisitos establecidos.
- **Automatización de pruebas:** Implementar herramientas de automatización que faciliten la ejecución de las pruebas, reduciendo tiempos y minimizando errores humanos, para evaluar de forma eficiente la funcionalidad, usabilidad y desempeño del sistema.
- **Validación de módulos:** Asegurar que cada módulo del sistema, incluyendo el registro de usuarios, la interacción social y el manejo de imágenes, cumpla con los estándares de calidad requeridos para garantizar una experiencia de usuario satisfactoria.
- **Entrega de resultados:** Documentar los hallazgos obtenidos durante las pruebas de software, proporcionando un reporte detallado que incluya observaciones, problemas detectados, soluciones implementadas y recomendaciones para futuras mejoras del sistema.

5. Marco Teórico

A continuación, se explican las herramientas utilizadas para el desarrollo del sistema y las pruebas de este.

IDEs

- **Android Studio:** Es un entorno de desarrollo integrado (IDE) enfocado en el desarrollo de aplicaciones móviles. Soporta el desarrollo de aplicaciones nativas con Kotlin y frameworks como Flutter (*Cómo Descargar Android Studio y App Tools - Android Developers*, s. f.).
- **Visual Studio Code (VSCode):** Es un editor de código fuente desarrollado por Microsoft, ampliamente utilizado para múltiples lenguajes de programación, incluyendo el desarrollo de aplicaciones móviles y web (*Visual Studio Code - Code Editing. Redefined*, 2021).

Herramientas

- **Framework Flutter:** Es un kit de herramientas de interfaz de usuario de código abierto desarrollado por Google, lanzado en 2015. Flutter permite la creación de aplicaciones móviles nativas a partir de un solo código fuente (*Flutter Casual Games Toolkit*, s. f.)
- **Flutter Driver:** Es una herramienta de automatización de pruebas específicamente diseñada para aplicaciones desarrolladas con el framework Flutter. Permite la ejecución de pruebas de integración (*Flutter Casual Games Toolkit*, s. f.)
- **Apache JMeter:** Es una herramienta de software que se utiliza para realizar pruebas de estrés y medir el rendimiento de aplicaciones en diversos escenarios configurables (*Testable*, s. f.).
- **Flutter Test:** Es una herramienta enfocada en la implementación de pruebas unitarias para aplicaciones desarrolladas con Flutter. Facilita la creación de pruebas automatizadas para garantizar la calidad del código (*Flutter Casual Games Toolkit*, s. f.).
- **Chocolatey:** Es una herramienta complementaria que se utiliza junto con Flutter Test para la generación de reportes. Ayuda a gestionar y automatizar la instalación de paquetes en sistemas Windows (*Chocolatey - The Package Manager For Windows*, s. f.)

IEEE 829 y Plan de Pruebas

- **IEEE 829:** Es un estándar para la documentación de pruebas de software que especifica la forma de un conjunto de documentos para su uso en ocho

etapas definidas de prueba de software y sistema. Cada etapa potencialmente produce su propio tipo de documento separado. Los documentos requeridos incluyen el Plan de Prueba Maestro (MTP), el Plan de Prueba de Nivel (LTP), el Diseño de Prueba de Nivel (LTD), el Caso de Prueba de Nivel (LTC), el Procedimiento de Prueba de Nivel (LTPr), el Registro de Prueba de Nivel (LTL), y el Informe de Anomalías (AR). (IEEE SA, s. f.)

- **Plan de Pruebas:** Es un documento detallado que describe la estrategia de prueba, los objetivos, el cronograma, la estimación, los entregables y los recursos necesarios para realizar pruebas de un producto de software. El plan de pruebas ayuda a determinar el esfuerzo necesario para validar la calidad de la aplicación bajo prueba. (*Rational Collaborative Lifecycle Management*, s. f.)

5.1 Metodología (Marco de trabajo 5W+2H)

- **WHO:**
 - Marley Morales
 - Steven Pozo
 - Erick Ramírez
 - Sebastián Torres
- **WHAT:**
 - Elaborar un Plan de Pruebas para validar el correcto funcionamiento de la aplicación móvil desarrollada en Flutter. Este plan incluirá la planificación, diseño y ejecución de pruebas automatizadas y manuales para asegurar que la aplicación cumpla con los requisitos funcionales y no funcionales, proporcionando una experiencia de usuario confiable y de alta calidad.
- **WHEN:**
 - Este proyecto está contemplado desde el 28 de octubre del 2024 hasta el 22 de febrero del 2025.
- **WHERE:**
 - Mediante reuniones virtuales

- **WHY:**
 - Asegurar que la aplicación cumpla con los estándares de calidad establecidos, identificando y corrigiendo errores antes de su despliegue. Un plan de pruebas bien definido es esencial para garantizar la funcionalidad, confiabilidad y robustez del sistema, así como para cumplir con las expectativas de los usuarios finales.
- **HOW:**

El Plan de Pruebas abarcará las siguientes actividades:

 - **Definición de estrategias de prueba:** Identificar los tipos de pruebas necesarias para cubrir todos los módulos del sistema.
 - **Diseño de casos de prueba:** Crear escenarios específicos que permitan evaluar tanto funcionalidades individuales como integradas.
 - **Ejecución de pruebas:** Aplicar pruebas unitarias, de integración, de rendimiento, de estrés y de aceptación.
 - **Automatización:** Utilizar herramientas para automatizar casos de prueba recurrentes, asegurando eficiencia y consistencia.
 - **Análisis y documentación:** Registrar los resultados obtenidos, documentar incidencias y generar reportes de calidad.
- **HOW MUCH:**
 - El plan y ejecución de pruebas no representa Ningún costo.

6. Ideas a Defender

Las ideas clave de este proyecto son fundamentales para garantizar que las pruebas realizadas al aplicativo cumplan con los estándares de calidad y funcionalidad establecidos. Estas ideas se basan en principios sólidos de ingeniería de software y pruebas de calidad, orientadas al desarrollo de un Plan de Pruebas efectivo. A continuación, se detallan:

- **Importancia del Plan de Pruebas como garantía de calidad:** El Plan de Pruebas es un documento esencial que permite estructurar y ejecutar las pruebas

de manera eficiente, asegurando que la aplicación cumpla con los requisitos establecidos y brindando confianza en su calidad final.

- **Identificación temprana y gestión efectiva de errores:** Las pruebas no solo evalúan el funcionamiento correcto del sistema en escenarios definidos, sino que también detectan fallos en funcionalidades específicas. Este enfoque facilita la corrección temprana y mejora la estabilidad general del aplicativo.
- **Cumplimiento de estándares internacionales de pruebas de software:** La adopción del estándar IEEE 829 asegura la uniformidad y calidad del proceso de pruebas, proporcionando una base sólida para documentar y analizar los resultados obtenidos de manera profesional y detallada.
- **Impacto de la usabilidad en la experiencia del usuario:** Las pruebas de aceptación validadas por el plan aseguran que el aplicativo sea intuitivo, fácil de usar y satisfaga las expectativas de los usuarios finales, garantizando una experiencia positiva.
- **Eficiencia y mantenimiento a largo plazo:** Las buenas prácticas en el desarrollo y las pruebas aseguran que el sistema sea eficiente, seguro y fácilmente mantenible, optimizando su desempeño en diversas condiciones operativas.

7. Resultados Esperados

Se espera obtener los siguientes resultados:

- **Aplicativo funcional y de calidad:** La aplicación debe cumplir con todos los requisitos funcionales y no funcionales, integrando correctamente los módulos y proporcionando un rendimiento óptimo bajo diferentes cargas de trabajo.
- **Documentación completa del proceso de pruebas:** Basado en el estándar IEEE 829, se generará un Plan de Pruebas que incluirá la planificación, diseño, ejecución, y resultados de las pruebas. Este documento servirá como evidencia del trabajo realizado y como base para futuros mantenimientos o actualizaciones del sistema.

- **Identificación y resolución de incidencias:** Un registro detallado de errores encontrados durante las pruebas, junto con su análisis y resolución, garantizará que el sistema funcione correctamente en diversos escenarios de uso.
- **Satisfacción del usuario final:** Se validará que el aplicativo sea fácil de usar, con una interfaz intuitiva y controles claros, proporcionando una experiencia positiva a los usuarios.
- **Optimización del rendimiento:** Asegurar que la aplicación sea estable y responda eficientemente bajo diversas condiciones operativas, fortaleciendo la confianza en el producto final.

Al cumplir con estos resultados, el proyecto no solo demostrará la eficacia del Plan de Pruebas, sino que también reforzará la calidad y confiabilidad del aplicativo.

8. Viabilidad

Cantidad	Descripción	Valor Unitario (USD)	Valor Total (USD)
	EQUIPO DE OFICINA		
1	Laptop	1000	1000
1	Laptop	1200	1200
1	Laptop Gamer	825	825
1	Laptop	1050	1050
	SOFTWARE		
1	Microsoft 365	0	0
1	VSCode	0	0
1	Android Studio	0	0
		TOTAL	4075

Tabla 1 Presupuesto del proyecto

8.1 Humana

8.1.1 Tutor Empresarial

Ing. Diego Gamboa

8.1.2 Tutor Académico

Ing. Jenny Ruiz

8.1.3 Estudiantes

- Jeimy Morales
- Steven Pozo
- Erick Ramírez
- Sebastián Torres

8.2 Tecnológica

8.2.1 Hardware

	Requisitos mínimos	Disponibilidad
Memoria RAM	8 GB de RAM	Alta
Almacenamiento	2 G de espacio de almacenamiento	Alta

8.2.2 Software

	Requisitos mínimos	Disponibilidad
Sistema Operativo	Windows 10 Pro Android	Alta
Herramientas	Visual Studio Code Android Studio Framework Flutter Flutter Driver Apache JMeter Flutter Test Chocolatey	Alta

9. Planificación para el Cronograma:

En la tabla se observa la planificación de las actividades a desarrollarse

Tabla: Planificación de actividades

Nº	Actividad	Tiempo de duración
1	Busca de entandares de automatización de pruebas de software	2
2	Recopilación y análisis de los estándares de pruebas encontrados	3
3	Seleccionar el estándar que se aplica a las pruebas	1
4	Creación de los lineamientos de evaluación en base al estándar	2
5	Planificación de la Estrategia de Automatización de Pruebas	3
6	Seleccionar herramientas de automatización	2
7	¿Las herramientas seleccionadas son adecuadas?	1
8	Preparar el entorno de pruebas	2
9	Diseño de las pruebas	5
10	¿Cumplen con los lineamientos definidos?	1
11	Ejecutar las pruebas	10
12	Realizar informe con los resultados obtenidos	2
13	Enviar los resultados a los desarrolladores	1

10. Bibliografía

Flutter Casual Games Toolkit. (s. f.).
https://flutter.dev/games?gad_source=1&gclid=Cj0KCQiAgJa6BhCOARIsAMiL7V98HP-KbLfaiTujlw6E7UrVf_DrUh7Ra8m8cEVyXIxK8wyEDPdErMaAr9VEALw_wcB&gclid=aw.ds

Chocolatey - The package manager for Windows. (s. f.). Chocolatey Software.
<https://chocolatey.org/>

Testable. (s. f.).
https://testable.io/jmeter?gad_source=1&gclid=Cj0KCQiAgJa6BhCOARIsAMiL7V_5AuRAkGooOQnbtjx2SK6rLR43RkXjd_fyQL0S5x_NlMhVaVorV8aAgMaEALw_wcB

Cómo descargar Android Studio y App Tools - Android Developers. (s. f.). Android Developers.

https://developer.android.com/studio?gad_source=1&gclid=Cj0KCQiAgJa6BhCOARIsAMiL7V_mEvaDIRN-bniGm-s6omOsrVR2kL-QJrI-3ekzhn3i01qysgbGBMIaAuixEALw_wcB&gclsrc=aw.ds&hl=es-419

Visual Studio Code - Code editing. Redefined. (2021, 3 noviembre).
<https://code.visualstudio.com/>

IEEE SA. (s. f.). *IEEE Standards Association.* IEEE Standards Association.
<https://standards.ieee.org/ieee/829/3787/>

Rational Collaborative Lifecycle management. (s. f.).
<https://www.ibm.com/docs/es/engineering-lifecycle-management-suite/lifecycle-management/6.0.6.1?topic=tutorials-develop-test-plan>

11. Anexos.

12. Anexo I. Crono

13. Anexo II. Crono Matriz de identificación de requisitos

14. Anexo III. Historia de Usuario