

Revisión Proyecto Interestructurante

Johan Camilo Calderón Álvarez

David Sebastián Blanco Pai

Cristian Camilo Lopez Benavides

Nicolas Alejandro Goyes Muñoz

Ingeniería de Calidad de Calidad

Universidad CESMAG

Dto. Ingeniería de sistemas

2025

Introducción

El propósito del siguiente documento es evaluar mediante técnicas de ingeniería de control de calidad, los requisitos funcionales definidos para la aplicación orientada al control y regulación de tiempo de uso de Facebook.

El análisis se centra en determinar el grado de claridad y viabilidad de cada requisito, que estos cumplan con los estándares de calidad, así mismo que faciliten su posterior implementación validación y mantenimiento.

Objetivo del Proyecto

El objetivo principal es desarrollar una aplicación móvil que permita a los usuarios gestionar y controlar de manera efectiva el tiempo dedicado a la red social Facebook, con el fin de enfrentar la problemática de la adicción y dependencia a las redes sociales. Esta aplicación busca ofrecer herramientas de autogestión como la configuración de límites personalizados de uso, la emisión de alertas preventivas, la activación de bloqueos temporales y el registro histórico del comportamiento digital.

Además, integra un sistema de recompensas y reportes gráficos que motiven el cumplimiento de los objetivos planteados, fomentando así hábitos digitales saludables, fortaleciendo el autocontrol y contribuyendo al bienestar emocional y social de los usuarios.

1- Revisión de Requisitos Funcionales

En la evaluación de estos requisitos se aplicarán criterios como:

- **Claridad**
- **Verificabilidad**
- **Viabilidad**
- **Consistencia**
- **Medibilidad**

RF01: Registro y autenticación segura de usuarios para proteger la información personal.

- **Claridad:** El requisito se entiende, aunque el término “segura” resulta ambiguo.
- **Verificabilidad:** Puede comprobarse mediante pruebas de inicio de sesión, validación de credenciales y cifrado de datos.
- **Viabilidad:** Su implementación es factible utilizando librerías y protocolos de seguridad estándar.
- **Consistencia:** Se alinea con los objetivos de privacidad y protección de datos del sistema.
- **Medibilidad:** No es totalmente medible, ya que no se especifica el estándar de seguridad a aplicar.

RF02: Configuración flexible de límites diarios de uso entre 10 minutos y 5 horas

- **Claridad:** El requisito está expresado de forma clara, con un rango definido.
- **Verificabilidad:** Se puede verificar mediante pruebas configurando distintos tiempos.
- **Viabilidad:** Es técnicamente viable mediante controles de tiempo.
- **Consistencia:** No presenta conflictos con otros requisitos.
- **Medibilidad:** Es medible, aunque convendría precisar si los intervalos son libres o predeterminados.
- **Consistencia:** No presenta conflictos con otros requisitos.
- **Medibilidad:** Es medible porque se puede registrar el número de configuraciones realizadas y el tiempo exacto de uso frente al límite definido.

RF03: Envío de notificaciones automáticas previas para alertar sobre el acercamiento al límite.

- **Claridad:** El requisito se entiende, pero no se detalla la anticipación exacta de la alerta.
- **Verificabilidad:** Puede comprobarse simulando un tiempo de uso cercano al límite y verificando si la notificación se emite.
- **Viabilidad:** Su implementación es factible mediante notificaciones
- **Consistencia:** Se integra adecuadamente con los requisitos de límites (RF02) y bloqueos (RF04).
- **Medibilidad:** No es completamente medible por la falta de definición en el tiempo de anticipación.

RF04: Bloqueo temporal del acceso a Facebook tras superar el límite configurado.

- **Claridad:** El requisito es claro, aunque debería precisar la duración del bloqueo y si es configurable por el usuario.
- **Verificabilidad:** Puede comprobarse excediendo el límite y verificando que el acceso quede bloqueado.
- **Viabilidad:** Su implementación es posible mediante programación interna de restricciones.
- **Consistencia:** Refuerza el cumplimiento del RF02 y complementa al RF03.
- **Medibilidad:** Es medible contando el número de bloqueos aplicados y su duración exacta.

RF05: Permite modificar los límites de uso en cualquier momento desde la configuración.

- **Claridad:** El requisito es entendible y directo.
- **Verificabilidad:** Se puede comprobar modificando los límites y observando el nuevo comportamiento del sistema.
- **Viabilidad:** Es factible y sencillo de implementar.
- **Consistencia:** Es coherente con RF02 y refuerza la flexibilidad del sistema.
- **Medibilidad:** Es medible a través del registro de cambios en los límites y su aplicación.

RF06: Registro histórico de tiempos de uso y bloqueos con visualización para el usuario.

- **Claridad:** La redacción es comprensible y sin ambigüedad.
- **Verificabilidad:** Puede comprobarse accediendo a los reportes históricos dentro de la aplicación.
- **Viabilidad:** Es factible con bases de datos y herramientas gráficas.
- **Consistencia:** Es coherente con RF09, que complementa la generación de reportes gráficos.
- **Medibilidad:** Es medible registrando la cantidad de datos almacenados, los periodos históricos y las visualizaciones generadas.

RF07: Sistema de recompensas con puntos e insignias para motivar el cumplimiento.

- **Claridad:** El requisito es entendible, pero no especifica los criterios exactos para otorgar las recompensas.
- **Verificabilidad:** Puede comprobarse configurando límites, cumpliéndolos y verificando si se otorgan puntos o insignias pero resulta difícil de verificar si no se definen reglas cuantificables para las recompensas.
- **Viabilidad:** Es factible mediante técnicas de gamificación sencillas.
- **Consistencia:** Se alinea con el objetivo del proyecto de promover hábitos digitales saludables.
- **Medibilidad:** Carece de parámetros claros para asignación de puntos, lo que dificulta su medición.

RF08: Personalización del idioma (Español/Inglés) y preferencias de notificaciones.

- **Claridad:** La redacción es precisa y no genera dudas.
- **Verificabilidad:** Puede comprobarse cambiando idioma y preferencias de notificaciones en el sistema.
- **Viabilidad:** Es viable porque los dispositivos móviles permiten multilenguaje y notificaciones configurables.
- **Consistencia:** No genera conflictos con otros requisitos.
- **Medibilidad:** Es medible mediante el número de idiomas disponibles, la cantidad de configuraciones aplicadas y la frecuencia de cambios realizados.

RF09: Generación automática de reportes gráficos y estadísticos de uso y progreso.

- **Claridad:** El requisito es comprensible, pero debería detallar los formatos de salida y la periodicidad de los reportes.
- **Verificabilidad:** Puede comprobarse generando un reporte y validando que muestre los datos esperados.
- **Viabilidad:** Es factible mediante librerías de gráficos y módulos de análisis de datos.
- **Consistencia:** Se integra correctamente con el registro histórico.
- **Medibilidad:** No especifica periodicidad, lo que afecta su precisión.

En general, los nueve requisitos funcionales cumplen con los principios de ser claros, factibles y coherentes con los objetivos del proyecto, ya que todos contribuyen directamente al propósito de controlar y regular el uso de Facebook. La mayoría también son verificables, pues es posible validarlos mediante pruebas funcionales dentro del sistema, lo que asegura que se pueda comprobar su implementación de manera práctica.

Sin embargo, durante la evaluación se identificaron algunas debilidades que conviene mejorar para elevar la calidad de los requisitos:

- **Falta de precisión técnica:** En algunos casos se utilizan términos generales como “segura” (RF01), “notificación previa” (RF03), “bloqueo temporal” (RF04) o “recompensas” (RF07), que aunque son entendibles, resultan ambiguos al no definir parámetros concretos como tiempo, duración, criterios o estándares.
- **Problemas de medibilidad:** Varios requisitos (RF01, RF03, RF07, RF09) no establecen métricas específicas, lo que dificulta comprobar con exactitud si se cumplen en el nivel esperado.
- **Necesidad de mayor detalle:** Aunque todos los requisitos son técnicamente viables, algunos carecen de especificaciones claras sobre su alcance, como la periodicidad de los reportes, las reglas para otorgar recompensas o la duración de los bloqueos.

Conclusión

En conclusión, los requisitos evaluados ofrecen una base sólida, pero requieren mayor nivel de especificidad y cuantificación para mejorar su claridad y medibilidad. Con estos ajustes, se facilitará que en las fases siguientes (diseño, desarrollo y pruebas) puedan implementarse de forma más precisa y trazable, asegurando que el producto final cumpla con lo que esperan los usuarios y con los objetivos planteados en el proyecto.

2- Revisión de Requisitos No Funcionales

En la evaluación de estos requisitos se aplicarán criterios como:

- **Claridad**
- **Verificabilidad**
- **Viabilidad**
- **Consistencia**
- **Medibilidad**

RNF 1: Seguridad:

- Cifrado de datos personales y de uso en almacenamiento y transmisión.
- Cumplimiento de normativas de privacidad y autorización para la gestión de datos.

Claridad: Se establece cifrado de datos y cumplimiento de normativas, aunque no se especifica cuáles.

Verificabilidad: Puede probarse mediante auditorías de seguridad y pruebas de cifrado.

Viabilidad: Factible con técnicas estándar de seguridad .

Consistencia: Alineado con RF01 (autenticación).

Medibilidad: Incompleto, ya que no define métricas (ej. nivel de cifrado, normativas específicas como GDPR).

RNF 2: Capacidad

- Almacenamiento de datos de uso, bloqueos y recompensas por al menos un año por usuario.
- Soporte para un mínimo de 500 usuarios concurrentes en la versión inicial.

Claridad: Indica específica y correctamente el almacenamiento por un año y soporte de 500 usuarios concurrentes.

Verificabilidad: Se pueden realizar pruebas de carga y almacenamiento.

Viabilidad: Factible para una versión inicial.

Consistencia: Compatible con RF06 (histórico de datos).

Medibilidad: Sí es medible (1 año de datos, 500 usuarios).

RNF 3: Compatibilidad

- Funcionamiento en principales plataformas móviles y navegadores web modernos.
- Posibilidad de exportación ajustada para gráficos y reportes.

Claridad: Requiere funcionar en plataformas móviles y navegadores modernos.

Verificabilidad: Puede comprobarse mediante pruebas multiplataforma.

Viabilidad: Realizable con frameworks multiplataforma.

Consistencia: Alineado con la interfaz de usuario (sección 3).

Medibilidad: Algo genérico, no define navegadores o versiones exactas pero se entiende el objetivo principal.

RNF 4: Confiabilidad

- Registro transparente de eventos y capacidad de recuperación ante fallos.
- Tiempo de actividad esperado mayor al 99% mensual.

Claridad: Define registro de eventos y disponibilidad >99%.

Verificabilidad: Puede comprobarse con monitoreo y métricas de uptime(tiempo que un sitio web permanece operativo, accesible y funcional para los usuarios).

Viabilidad: Requiere infraestructura robusta, pero posible.

Consistencia: Alineado con el objetivo de servicio continuo.

Medibilidad: Es medible, gracias al indicador de 99% mensual.

RNF 5: Escalabilidad

- Arquitectura modular para incorporación sencilla de nuevas funcionalidades o integración con otras redes sociales.
- Soporte para la inclusión futura de módulos adicionales, como recomendaciones y comunidades.

Claridad: Propone arquitectura por módulos y posibilidad de integrar nuevas funcionalidades.

Verificabilidad: Puede comprobarse con pruebas de ampliación de módulos.

Viabilidad: Factible con buenas prácticas de diseño.

Consistencia: Coherente con la visión futura del producto.

Medibilidad: Sería recomendable especificar cuántos módulos podría soportar el sistema para tener métricas más claras.

RNF 6: Mantenibilidad

- Código fuente documentado y pruebas automáticas para cada módulo principal.
- Manual de usuario y sección de preguntas frecuentes para soporte rápido.

Claridad: Menciona documentación y pruebas automáticas.

Verificabilidad: Puede comprobarse revisando el código y la documentación.

Viabilidad: Factible con metodologías ágiles.

Consistencia: Alineado con buenas prácticas de desarrollo.

Medibilidad: No se requieren métricas exactas ya que principalmente se usan medidas cualitativas.

RNF 7: Facilidad de uso

- Interfaz amigable y accesible, con curva de aprendizaje menor a 30 minutos.
- Soporte multilenguaje para ampliar la accesibilidad.

Claridad: Indica interfaz amigable, accesible y curva de aprendizaje <30 min.

Verificabilidad: Puede validarse con pruebas de usabilidad y encuestas a usuarios.

Viabilidad: Realizable con diseño UX(User Experience) adecuado.

Consistencia: Alineado con RF03 y RF08 (interfaz y notificaciones).

Medibilidad: El límite de 30 minutos sí ofrece un parámetro cuantificable.

RNF 8: Otros requisitos no funcionales

- Funcionamiento en modo offline con acceso a datos y bloqueos, y sincronización automática cuando haya conexión a internet.
- Accesibilidad para usuarios con discapacidad visual, con soporte para alto contraste y lectores de pantalla.

Claridad: Se mencionan funcionamiento offline y accesibilidad para personas con discapacidad visual.

Verificabilidad: Puede probarse desconectando internet y utilizando lectores de pantalla.

Viabilidad: Factible con almacenamiento local y soporte de accesibilidad.

Consistencia: Coherente con la finalidad inclusiva del proyecto.

Medibilidad: No define métricas exactas (ej. porcentaje de accesibilidad cumplido según WCAG o Pautas de Accesibilidad al Contenido en la Web).

Conclusión

En general, los ocho requisitos no funcionales analizados aportan una base sólida para garantizar la calidad del sistema, ya que abarcan aspectos clave como seguridad, capacidad, compatibilidad, confiabilidad, escalabilidad, mantenibilidad, facilidad de uso y accesibilidad. Estos requisitos complementan a los funcionales, asegurando no solo que el sistema cumpla lo que debe hacer, sino también **cómo** debe hacerlo, priorizando la experiencia del usuario y la sostenibilidad técnica a largo plazo.

La mayoría de los requisitos presentan un buen nivel de claridad y consistencia, estando alineados con los objetivos del proyecto y con estándares reconocidos de ingeniería de software. Además, varios incluyen parámetros verificables y medibles (ej. almacenamiento por un año, soporte de 500 usuarios, disponibilidad >99%, curva de aprendizaje <30 minutos), lo cual facilita su comprobación práctica en etapas de pruebas.

Sin embargo, al igual que en los requisitos funcionales, se detectaron algunas debilidades:

- Falta de precisión en la medibilidad en algunos casos (RNF01 Seguridad, RNF03 Compatibilidad, RNF05 Escalabilidad, RNF08 Accesibilidad), ya que se formulan de manera general sin métricas concretas.
- Uso de términos amplios o subjetivos, como “arquitectura modular”, “accesible” o “funcionamiento en principales navegadores”, que aunque entendibles, podrían prestarse a interpretaciones distintas si no se delimitan con mayor detalle.

En conclusión, los RNF evaluados cumplen con dar dirección técnica y de calidad al proyecto, pero requieren ajustes en la especificidad y cuantificación para mejorar su verificabilidad y trazabilidad. Incorporar métricas claras (por ejemplo, estándares de cifrado, versiones mínimas de navegadores soportados, nivel de accesibilidad

WCAG alcanzado, cantidad máxima de módulos integrables) permitiría una evaluación más objetiva y reduciría riesgos de ambigüedad en fases posteriores de diseño, desarrollo y pruebas.

PRUEBAS UNITARIAS MÓDULO DE REGISTRO

1. Introducción

La aplicación deberá tener un apartado en el cual los usuarios que ingresen deben registrar los datos necesarios para poder crear su cuenta y así empezar a usar la aplicación

2. Alcance

Únicamente se tendrá en cuenta el módulo de registro, no se tendrá en cuenta el módulo de login ya que si el usuario no se ha registrado, no podrá iniciar sesión para usar la aplicación.

3. Estrategia de Pruebas

- Pruebas de caja negra(se centran en entradas y salidas sin conocer su estructura interna)

4. Casos de Prueba

Elemento	Descripción
ID del Caso de Prueba	TC-001
Título / Nombre	Registrar valores nulos o no válidos
Descripción	En este caso se validará el uso de valores nulos o vacíos o caracteres diferentes a los que se pidan en el registro
Precondiciones	El módulo de registro debe estar funcionando correctamente
Datos de Entrada	Correo: " " Contraseña: " "
Pasos de Prueba	1. Abrir la aplicación 2. Ir al módulo de registro 3. Insertar los datos de entrada
Resultado Esperado	Advertencia de que los valores asignados no son válidos
Resultado Obtenido	Al intentar registrarse sin llenar ninguno de los datos necesarios en el formulario, el sistema muestra una

	alerta la cual describe que los valores están vacíos y no puede realizarse el registro.
Estado	Aprobado
Notas/Observaciones	Sin observaciones

Elemento	Descripción
ID del Caso de Prueba	TC-002
Título / Nombre	Evaluación de usuarios únicos en la aplicación
Descripción	En esta prueba se evaluará que un solo correo le pertenezca a una única cuenta.
Precondiciones	El módulo de registro debe estar funcionando correctamente
Datos de Entrada	Correo: "David@gmail.com" Contraseña: "David123**"
Pasos de Prueba	<ol style="list-style-type: none"> 1. Abrir la aplicación 2. Ir al módulo de registro 3. Insertar los datos de entrada 4. Volver a insertar los datos de entrada
Resultado Esperado	Advertencia de que el correo ya está registrado
Resultado Obtenido	Muestra un mensaje indicando que el correo con el que se intenta registrar ya se encuentra registrado y es necesario registrarse con otro correo
Estado	Aprobado
Notas/Observaciones	Sin observaciones

Elemento	Descripción
ID del Caso de Prueba	TC-003
Título / Nombre	Registro con contraseña demasiado corta
Descripción	Validar que el sistema rechace el registro si la contraseña no cumple con la longitud mínima de seguridad (ej: menos de 6 caracteres).
Precondiciones	El módulo de registro debe estar funcionando correctamente.
Datos de Entrada	Correo: "usuario_nuevo@mail.com" Contraseña: "abc"
Pasos de Prueba	<ol style="list-style-type: none"> 1. Abrir la aplicación 2. Ir al módulo de registro 3. Insertar los datos de entrada 4. Pulsar el botón "Registrarse"
Resultado Esperado	Advertencia de que la contraseña es demasiado corta y debe contener caracteres especiales
Resultado Obtenido	Muestra un mensaje indicando que la contraseña es demasiado corta y es necesario que contenga caracteres especiales.
Estado	Aprobado
Notas/Observaciones	Sin observaciones

ID	Función Probada	Entrada	Salida Esperada	Resultado Obtenido	Estado	Observaciones
UT-001	registrar()	Correo: " " Contraseña: " "	Advertencia: Valores Nulos	El sistema muestra de manera correcta una advertencia	Aprobado	Sin observaciones
UT-002	registrar()	Correo: "david@gmail.com" Contraseña: "David123*"	Advertencia: Correo ya registrado	El sistema indica que el correo ya está registrado.	Aprobado	Sin observaciones
UT-003	registrar()	Correo: "usuario_nuevo@mail.com" Contraseña: "abc"	Advertencia que la contraseña es demasiado corta y debe contener caracteres especiales	Muestra un mensaje indicando que la contraseña es demasiado corta y es necesario que contenga caracteres especiales	Aprobado	Sin observaciones

5. Criterios de Aceptación

- El sistema pasó con un 100% de aprobación

6. Cronograma y Responsables

Fase	Fecha de Inicio	Fecha de Finalización	Responsable
------	-----------------	-----------------------	-------------

Ejecución de Pruebas	08/10/2025	09/10/2025	Equipo QA
----------------------	------------	------------	-----------

7. Reporte de Resultados

- **Número de pruebas ejecutadas: 3**
- **Pruebas aprobadas: 3**
- **Pruebas fallidas: 0**
- **Errores detectados: 0**
- **Correcciones aplicadas: 0**
- **Pasos a seguir:** Seguir en la construcción de los siguientes módulos

PRUEBAS UNITARIAS MÓDULO DE LOGIN

1. Introducción

El objetivo es verificar el correcto funcionamiento de las funciones de registro y autenticación, validación de credenciales y manejo de errores antes de la integración con otros módulos

2. Alcance

Se evalúa conjunto al módulo de registro ya que únicamente se evalúan los usuarios previamente registrados

3. Estrategia de Pruebas

- Metodología de pruebas: Prueba de caja blanca

4. Casos de Prueba

Elemento	Descripción
ID del Caso de Prueba	TC-001
Título / Nombre	Autenticación exitosa de usuario registrado
Descripción	Verificar que el sistema permita el inicio de sesión cuando el usuario ingresa un correo y contraseña válidos
Precondiciones	El usuario "David@gmail.com" está previamente registrado con la contraseña clave "David123*". Conexión estable y base de datos activa.
Datos de Entrada	Correo: "David@gmail.com" Contraseña: "David123*"
Pasos de Prueba	1. Abrir la aplicación 2. Ingresar correo válido 3. Ingresar contraseña válida 4. Presionar el botón "Iniciar sesión"
Resultado Esperado	El sistema valida las credenciales y muestra el mensaje "Inicio de sesión exitoso"

Resultado Obtenido	El sistema muestra valida los datos ingresados del usuario y permite el inicio de sesión exitosamente
Estado	Aprobado
Notas/Observaciones	Sin observaciones

Elemento	Descripción
ID del Caso de Prueba	TC-002
Título / Nombre	Inicio de sesión fallido por contraseña incorrecta
Descripción	Verificar que el sistema muestre un mensaje de error y no permita el acceso cuando el usuario ingrese una contraseña incorrecta
Precondiciones	El usuario "David@gmail.com" está previamente registrado con la contraseña clave "David-123*". Contraseña de prueba: "Prueba-123*" Conexión estable y base de datos activa
Datos de Entrada	Correo: David@gmail.com Contraseña: Prueba-123*
Pasos de Prueba	<ol style="list-style-type: none"> 1. Abrir la aplicación 2. Ingresar correo válido 3. Ingresar contraseña incorrecta 4. Presionar el botón "Iniciar sesión"
Resultado Esperado	El sistema debe rechazar el inicio de sesión y mostrar el mensaje "Contraseña incorrecta"
Resultado Obtenido	El sistema muestra una alerta la cual describe que la contraseña es invalida.
Estado	Aprobado
Notas/Observaciones	El sistema comprueba correctamente que la contraseña registrada sea correspondiente al correo registrado

Elemento	Descripción
ID del Caso de Prueba	TC-003
Título / Nombre	Inicio de sesión con campos vacíos
Descripción	Verificar que el sistema muestre un mensaje de error y no permita el acceso si se deja el campo de correo o contraseña vacío.
Precondiciones	Conexión estable y base de datos activa
Datos de Entrada	Correo: " ", Contraseña: " "
Pasos de Prueba	<ol style="list-style-type: none"> 1. Abrir la aplicación 2. Ingresar correo válido 3. Ingresar contraseña incorrecta 4. Presionar el botón "Iniciar sesión"
Resultado Esperado	El sistema debe mostrar una alerta indicando que el campo de correo no puede estar vacío.
Resultado Obtenido	El sistema muestra una alerta indicando que el campo de correo y contraseña no puede estar vacío.
Estado	Aprobado
Notas/Observaciones	Sin observaciones

ID	Función Probada	Entrada	Salida Esperada	Resultado Obtenido	Estado	Observaciones
UT-001	validarCorreo	"David@gmail.com"	"Inicio de sesión exitoso"	El sistema valida la dirección de correo electrónico permitiendo el inicio de sesión exitosamente	Aprobado	Sin observaciones
UT-002	validarContraseña	"David123"	"Inicio de sesión exitoso"	El sistema valida la contraseña y permite el inicio de sesión exitosamente	Aprobado	Sin observaciones
UT-003	validarContraseña validarCorreo	" " " "	"Inicio de sesión no exitoso"	El sistema muestra una alerta indicando que el campo de correo y contraseña no puede estar vacío.	Aprobado	Sin observaciones

5. Criterios de Aceptación

- El 100% de las pruebas deben aprobarse sin errores críticos
- No se debe permitir el acceso con credenciales inválidas
- Todas las funciones deben manejar correctamente las excepciones

6. Cronograma y Responsables

Fase	Fecha de Inicio	Fecha de Finalización	Responsable
Ejecución de Pruebas	01/10/2025	09/10/2025	Equipo QA

7. Reporte de Resultados

- **Número de pruebas ejecutadas: 3**
- **Pruebas aprobadas: 3**
- **Pruebas fallidas: 0**
- **Errores detectados: 0**
- **Correcciones aplicadas: 0**

Pasos a seguir: Seguir en la construcción de los módulos siguientes.

PRUEBAS DE INTEGRACIÓN DE SOFTWARE

1. Información General

Nombre del Proyecto: Control de Facebook

Versión del Documento: 1.0

Responsable de Pruebas: David Blanco y Nicolas Goyes

Fecha de Creación: 01/02/2025

Objetivo: Verificar la correcta integración de los módulos del software y la comunicación entre ellos.

2. Estrategia de Pruebas

Tipo de Prueba:

Entorno de Pruebas: QA

Herramientas Utilizadas:

Criterios de Aceptación:

- Todos los módulos deben interactuar correctamente.
- Los datos transferidos deben ser precisos y completos.
- La comunicación con APIs y bases de datos debe ser estable.
- El rendimiento del sistema debe cumplir los estándares definidos.

3. Casos de Prueba de Integración

ID	Módulos Involucrados	Descripción	Precondiciones	Pasos	Entrada	Salida Esperada	Resultado
IN-001	Módulo de registro y módulo de inicio de sesión(login)	Validar que únicamente los usuarios registrados puedan iniciar sesión en la aplicación	Ambos módulos deben estar conectados a la base de datos para poder validar los datos del usuario.	1. Registrar un usuario con datos válidos 2. Iniciar sesión con los datos ya mencionados	Correo: "David@gmail.com" Contraseña: "David123"	Registro e inicio de sesión completado exitosamente	Registro e inicio de sesión completado exitosamente

IN-002	Módulo de registro y base de datos	Validar que el registro almacena el correo en minúsculas y el login lo acepte indistintamente de la capitalización ingresada.	Los módulos deben estar conectados a la base de datos. El usuario "david@gmail.com" está registrado.	1. Intentar iniciar sesión con el correo en mayúsculas/minúsculas.	Correo: "DAVID@GMAIL.COM", Contraseña: "David123"	Inicio de sesión completado exitosamente	Inicio de sesión completado exitosamente
---------------	------------------------------------	---	--	--	---	--	--

4. Gestión de Incidentes

Registro de Errores:

ID de Error	Descripción	Módulo Afectado	Severidad	Estado	Fecha de Detección	Responsable
ERR-001	No se encontraron errores que comprometan el funcionamiento del sistema	Módulo de registro y login	Baja	Cerrado	09/10/2025	Equipo QA

5. Cronograma de Pruebas

Fase	Fecha de Inicio	Fecha de Fin	Responsable
Planificación	01/09/2025	08/09/2025	Equipo QA
Desarrollo de Casos	05/10/2025	08/10/2025	Equipo QA
Ejecución	09/10/2025	09/10/2025	Equipo QA
Reporte de Errores	09/10/2025	09/10/2025	Equipo QA

6. Conclusiones y Recomendaciones

Conclusión:

El sistema comprueba correctamente los datos ingresados tanto de registro como de login, esto muestra que estos dos módulos operan de manera correcta y constante, teniendo una buena conectividad a la base de datos y una buena compatibilidad. Se incluye además un método el cual obliga al usuario a tener una contraseña segura con caracteres especiales. Esto ayuda a que los datos sean seguros y se eviten problemas de ataques cibernéticos en el futuro.

Recomendaciones:

Se recomienda que el desarrollo de los futuros módulos mantenga los altos estándares de efectividad y rendimiento observados en los sistemas de inicio de sesión y registro de usuario. Es crucial replicar la robustez en la validación de credenciales y la calidad de la experiencia de usuario (UX), asegurando que la interfaz se mantenga agradable, intuitiva y funcional a lo largo de toda la aplicación.

7. Aprobaciones

Nombre	Rol	Firma	Fecha
David Blanco	Equipo QA	David Blanco	09/10/2025
Nicolas Goyes	Equipo QA	Nicolas Goyes	09/10/2025

PRUEBAS FUNCIONALES

1. Información General

Proyecto:Control de Facebook

Versión del Software: 1.0

Fecha de Elaboración: 09/10/2025

Responsable: Gabriel Rosero

Equipo de Pruebas: David Blanco - Nicolas Goyes

2. Objetivo de las Pruebas

El principal objetivo es verificar que los módulos de registro e inicio de sesión(login) cumplan lo establecido en el requisito funcional 01.

3. Alcance de las Pruebas

En este caso, el requisito funcional 01 establece:

“Registro y autenticación segura de usuarios para proteger la información personal”.

Por lo que evaluaremos que los módulos de registro e inicio de sesión funcionen correctamente y no haya brechas de seguridad.

4. Criterios de Aceptación y Rechazo

- **Criterios de Aceptación:**
 - Todas las funcionalidades del login deben ejecutarse sin errores críticos.
 - Todos los casos de prueba definidos deben completarse con éxito
 - La autenticación debe ser segura y generar sesión válida
- **Criterios de Rechazo:**
 - Errores que interrumpa el flujo normal de autenticación
 - No validación de credenciales incorrectas o campos vacíos
 - Respuestas del sistemas distintas a las especificadas (Mensajes erróneos o sin control)

5. Estrategia de Pruebas

En este caso al no evaluar código interno, ejecutaremos pruebas de caja negra

- Tipos de Pruebas:
 - Pruebas de caja negra
 - Pruebas de regresión (para confirmar que nuevas funciones no afecten el login)
 - Pruebas de validación de interfaz (Mensajes, botones y navegación)

6. Casos de Prueba

ID Caso	Funcionalidad	Descripción	Datos de Entrada	Resultado Esperado	Estado
CP-001	Inicio de Sesión	Verificar autenticación de usuario	Correo: "David@gmail.com" Contraseña: "David123*"	Acceso concedido.	Aprobado
CP-002	Registro de Usuario	Validar registro con datos ya registrados	Correo: "David@gmail.com" Contraseña: "David123*"	Mensaje de registro exitoso.	Aprobado
CP-003	Acceso con credenciales incorrectas	Intentar ingresar con una contraseña incorrecta y un usuario ya registrado	Correo: "David@gmail.com" Contraseña: "123*David"	Mensaje de acceso denegado.	Aprobado

7. Recursos Necesarios

- **Servidor activo en Android Studio**
- **Conexión a base de datos funcional**
- **Datos de prueba:** Usuario registrado "David@gmail.com" , "David123*" , "123*David"

8. Plan de Ejecución

Actividad	Responsable	Fecha de Inicio	Fecha de Finalización
Definición de casos de prueba	Equipo QA	08/10/2025	08/10/2025
Configuración del entorno	Equipo QA	09/10/2025	09/10/2025
Ejecución de pruebas	Equipo QA	09/10/2025	09/10/2025

Reporte de defectos	Equipo QA	09/10/2025	09/10/2025
---------------------	-----------	------------	------------

9. Reporte de Resultados

- **Casos de Prueba Ejecutados: 3**
- **Casos de Prueba Exitosos: 3**
- **Casos de Prueba Fallados: 0**
- **Defectos Reportados: 0**

10. Conclusiones y Recomendaciones

Se evaluó el requisito funcional 01, el cual indica “Registro y autenticación segura de usuarios para proteger la información personal.”

Los módulos de Login y Registro cumplen correctamente con lo establecido en el requisito funcional, ya que permite registrar exitosamente un usuario y a su vez le permite al usuario ingresar con sus credenciales correctas además las credenciales de cada usuario son únicas, por lo cual no permite ingresar al sistema usando una contraseña diferente a la ya establecida.

Aprobado por:

David Blanco

Nicolas Goyes

09/10/2025

PRUEBAS NO FUNCIONALES

Información General

Proyecto: Aplicación de Control de Uso de Facebook

Versión del Documento: 1.0

Responsable de Pruebas: Equipo QA

Fecha: [09/10/2025]

Objetivo: Validar que el sistema cumpla con los requisitos no funcionales establecidos: rendimiento, seguridad, usabilidad, compatibilidad y mantenibilidad.

2. Alcance

Las pruebas no funcionales cubrirán los siguientes aspectos del sistema:

- Rendimiento de la app durante (tiempos de respuesta y carga).
- Seguridad en el manejo de datos de usuario y cifrado de contraseñas.
- Usabilidad y comprensión de mensajes e interfaz (facilidad de uso e interacción).
- Compatibilidad entre diferentes versiones de Android e IOS
- Mantenibilidad (facilidad de actualización y corrección).

Excluye:

- Validaciones funcionales como registro o login.
- Pruebas de integración entre módulos.

3. Tipos de Pruebas No Funcionales

Tipo de Prueba	Objetivo	Herramientas / Métodos
Rendimiento	Medir el tiempo de respuesta del sistema en operaciones críticas	Cronómetro manual del dispositivo, observación directa de la latencia.
Seguridad	Validar la robustez de los campos de entrada y la seguridad de las contraseñas	Ingreso manual de los datos maliciosos o no válidos. Observación de mensajes de error.
Usabilidad	Evaluar la facilidad de uso de la interfaz y comprensión de mensajes.	Pruebas guiadas con usuarios (<i>walkthroughs</i>), encuestas de percepción y observación directa.

4. Casos de Prueba No Funcionales

ID Caso	Tipo de Prueba	Descripción	Métrica / Criterio	Resultado Esperado	Estado
NF-01	Rendimiento	Medir tiempo de carga del login	< 2 segundos	Cumple el umbral	Aprobado
NF-02	Seguridad	Los datos de los	Base de datos sin	Cumple validación	Aprobado

		usuarios deben ser íntegros y confidenciales.	caídas / respuesta estable		
NF-03	Usabilidad	Evaluar que tan intuitiva es la aplicación al momento de registrarse e iniciar sesión.	90% de usuarios puede navegar en ella	Cumple	Aprobado

5. Criterios de Aceptación

- El sistema debe responder en menos de 2 segundos en acciones críticas.
- No deben existir vulnerabilidades de seguridad detectables.
- Al menos el 90% de los usuarios deben calificar la app como fácil de usar.

6. Recursos y Entorno

Herramientas (Manuales/Soporte): Cronómetro de dispositivo, bases de datos de prueba para inspección, emulador de android.

Ambiente de pruebas: Entorno QA con base de datos de prueba y red controlada.

Datos de prueba: Usuarios simulados y credenciales genéricas.

7. Cronograma y Responsables

Actividad	Responsable	Fecha Inicio	Fecha Fin
Diseño de casos no funcionales	Equipo QA	07/10/2025	08/10/2025
Configuración del entorno	Equipo QA	08/10/2025	09/10/2025
Ejecución de pruebas	Equipo QA	09/10/2025	10/10/2025
Análisis de resultados	Equipo QA	10/10/2025	11/10/2025

8. Reporte de Resultados

- **Casos de prueba ejecutados: 3**
- **Casos aprobados: 3**
- **Casos fallados: 0**
- **Principales hallazgos: 0**
- **Recomendaciones:**

Lista de Chequeo para Verificación de Interfaces Gráficas de Usuario (GUI)

Formato basado en las recomendaciones de las normas ISO/IEC 9126, ISO 9241-110 y las heurísticas de Nielsen. Esta lista permite evaluar el cumplimiento visual, funcional y de usabilidad de una interfaz gráfica. El evaluador debe marcar cada criterio con ✓ (cumple), ✗ (no cumple) o N/A y anotar observaciones.

1. Estructura y diseño visual

Criterio	Cumple (✓ / ✗/N/A)	Observaciones
Los elementos están alineados correctamente.	✓	El diseño es pulcro y profesional.
Los márgenes y espaciados son consistentes.	✓	El espaciado uniforme contribuye a la estética agradable.
Los botones, etiquetas y campos tienen el mismo estilo visual.	✓	Hay una clara identidad visual que hace que la interfaz se vea "bonita".
Los íconos son comprensibles y coherentes con su función.	✓	Los íconos son modernos y su significado es claro.
Se usa una paleta de colores uniforme y legible.	✓	La paleta es atractiva y asegura la legibilidad del texto.

Se mantiene consistencia en el uso de tipografía (tamaño, estilo).	✓	Tipografía limpia y consistente en todo el sistema.
--	---	---

2. Navegación

Criterio	Cumple (✓ / ✗/N/A)	Observaciones
La navegación entre pantallas es intuitiva.	✓	El flujo es lógico y fácil de seguir.
Los menús están organizados lógicamente.	✓	La estructura de la información es clara.
Existe una forma clara de regresar o cancelar una acción.	✓	Botones "Atrás" y "Cancelar" visibles y funcionales.
El usuario siempre sabe en qué parte del sistema se encuentra.	✓	Se usa resaltado de menú para indicar la ubicación actual.
Los enlaces o botones son fácilmente identificables.	✓	Son visualmente prominentes y sugieren acción.

3. Funcionalidad básica

Criterio	Cumple (✓ / ✗/N/A)	Observaciones
Todos los botones y enlaces responden correctamente al clic.	✓	No se detectan fallos funcionales en la interacción.
Los campos obligatorios están claramente marcados.	✓	Se usa el asterisco (*) o etiquetas claras.
Las validaciones de datos funcionan correctamente (campos vacíos, formatos, etc.).	✓	Las validaciones son robustas, evitando errores en el sistema.
Los mensajes de error o confirmación aparecen de forma oportuna.	✓	La retroalimentación de las acciones es inmediata.
El sistema evita acciones no válidas.	✓	Previene el envío múltiple al deshabilitar el botón.

4. Accesibilidad y legibilidad

Criterio	Cumple (✓ / ✗/N/A)	Observaciones
El contraste entre texto y fondo es adecuado.	✓	El contraste es óptimo para la lectura.
El tamaño de letra es legible en diferentes resoluciones.	✓	Se adapta bien, sin necesidad de zoom excesivo.
El sistema es usable con teclado (tabulación, accesos rápidos).	✓	No se verificó el uso exclusivo de teclado en esta revisión.
No se depende solo del color para transmitir información.	✓	Se usan íconos o texto complementario a los indicadores de color.

5. Retroalimentación e interacción

Criterio	Cumple (✓ / ✗/N/A)	Observaciones
El sistema muestra mensajes claros tras realizar acciones.	✓	Los mensajes de éxito son concisos y agradables.
Los mensajes de error son específicos y ayudan al usuario a corregir.	✓	Los errores son amigables y orientan al usuario.

El tiempo de respuesta es razonable (<2 segundos para acciones simples).	✓	La percepción de velocidad es óptima.
Los cambios en el sistema son visibles inmediatamente.	✓	No hay latencia en la actualización de la interfaz.

6. Coherencia y usabilidad general

Criterio	Cumple (✓ / ✗/N/A)	Observaciones
El lenguaje usado es claro y sin tecnicismos.	✓	El vocabulario es apropiado para el público objetivo.
Los íconos, botones y menús mantienen su posición en todas las pantallas.	✓	Consistencia de posición y función.
Los usuarios nuevos pueden entender el sistema sin ayuda externa.	✓	Interfaz intuitiva que minimiza la curva de aprendizaje.
Las acciones frecuentes están accesibles sin pasos innecesarios.	✓	Diseño eficiente para las tareas principales.

7. Pruebas específicas de GUI

Criterio	Cumple (✓ / ✗/N/A)	Observaciones
Las ventanas se ajustan correctamente al tamaño de pantalla.	✓	El diseño es adaptativo y se ve bien en diferentes dispositivos.
No hay elementos cortados o superpuestos.	✓	Diseño visualmente limpio y sin defectos de maquetación.
Las etiquetas no se salen de su contenedor.	✓	El espacio de las etiquetas es suficiente.
Los campos de texto aceptan la longitud adecuada.	✓	Los límites de entrada son funcionales.
No hay errores visuales al cambiar el idioma o el tema.	✓	Sin observaciones