Documento Control De Calidad

Trabajo de Grado Ingeniería de Sistemas

BRAINFIT- HERRAMIENTA DE APOYO MÉDICO DE DIAGNÓSTICO DE DEMENCIA

Presentado Por:

LAURA SOFÍA JIMÉNEZ BALLEN
DAVID SANTIAGO QUINTANA ECHAVARRIA
NATALIA GAONA SALAMANCA

Pontificia Universidad Javeriana

Departamento de Ingeniería de Sistemas

Bogotá D.C.

2023

1. Historial de Cambios

Fecha	Descripción	Sección	Responsable
30/10/2023	Se realizó la prueba de seguridad de las apps.	8.1	Natalia Gaona Salamanca
31/10/2023	Se realizó la introducción, la descripción general y la estrategia de las pruebas.	5,6 y 7	Natalia Gaona Salamanca
02/10/2023	Se realizó las pruebas unitarias de la app de escritorio.	8.3	Laura Sofia Jiménez
06/10/2023	Se realizó las pruebas de funcionalidad de la aplicación.	8.4	Laura Sofia Jiménez
28/11/2023	Se realizó las pruebas de carga de la aplicación	8.6	Laura Sofia Jiménez
28/11/2023	Se realizó las pruebas de estres de la aplicación	8.7	David Santiago Quintana Echavarria

Tabla 1. Historial de cambios

2. Tabla de Contenidos

3. Lista de Figuras

4. Lista de Tablas

5. Introducción

Este documento de pruebas tiene como objetivo proporcionar una visión general de la estrategia de pruebas y los procedimientos que se han implementado para asegurar la calidad, eficacia y seguridad de la aplicación BrainFit. BrainFit es una herramienta desarrollada tanto para dispositivos móviles como para computadoras, diseñada específicamente para médicos del hospital San Ignacio. La aplicación permite a los médicos llevar a cabo pruebas de audio y video en pacientes y analizar estos datos para determinar la presencia de signos de demencia.

El proceso de pruebas desempeña un papel crítico en la garantía de que BrainFit funcione de manera óptima, cumpla con los estándares requeridos y ofrezca una experiencia de usuario excepcional. En este documento, detallaremos los objetivos, el alcance y las técnicas de pruebas que se han empleado para alcanzar estos propósitos.

La estrategia de pruebas se basa en un enfoque combinado de pruebas manuales y automáticas para evaluar las diversas funciones y componentes de la aplicación. Nuestra prioridad es identificar y abordar cualquier problema antes de que la aplicación se ponga a disposición del público.

Este proceso de pruebas es esencial para garantizar que BrainFit cumpla con los estándares de calidad y seguridad exigidos en el sector médico y para proporcionar a los médicos una herramienta confiable y efectiva en su trabajo diario. Estamos comprometidos con la mejora continua y la entrega de una aplicación que brinde un alto nivel de confianza a los profesionales de la salud y a los pacientes.

A lo largo de este documento, detallaremos los objetivos específicos, el alcance de las pruebas y las técnicas utilizadas, con el fin de proporcionar una comprensión completa de nuestra estrategia de pruebas para BrainFit.

6. Descripción General BrainFit

6.1. Resumen

BrainFit es una aplicación que ha sido creada tanto para dispositivos móviles como para computadoras, utilizando tecnologías como Flutter y Python. Esta herramienta se ha diseñado específicamente para médicos del hospital San Ignacio. La versión móvil de la aplicación permite a los médicos llevar a cabo pruebas de audio y video en cada paciente, los resultados de las cuales se almacenan en una base de datos en la nube, utilizando Firebase como plataforma. Por otro lado, la versión de escritorio de la aplicación permite la descarga de estos archivos de audio y video, que posteriormente pueden ser analizados utilizando diversas bibliotecas para determinar si el paciente presenta signos de demencia o no.

6.2. Funcionalidades

A continuación, las funcionalidades implementadas fueron:

- Autenticación: Los médicos pueden autenticarse para acceder tanto a la aplicación móvil como a la de escritorio.
- Grabación de multimedia: Los médicos pueden utilizar la aplicación móvil para grabar videos, audios y videos con audios.
- Almacenamiento en la nube: Los médicos tienen la capacidad de subir los videos y audios a Firebase, una plataforma de almacenamiento en la nube.
- Descarga de archivos: Los médicos pueden descargar los videos almacenados en Firebase directamente a sus ordenadores a través de la aplicación de escritorio.
- Procesamiento multimedia: Los médicos cuentan con la capacidad de procesar los videos y audios que han grabado.
- Análisis profundo: Los médicos pueden acceder a gráficas de los videos para realizar un análisis más detallado de la información registrada.
- Resultados de audios: La aplicación les permite a los médicos visualizar los resultados derivados de los archivos de audio.
- Resultado final: Los médicos pueden consultar el resultado final del análisis realizado a través de la aplicación.

7. Estrategia de Pruebas

7.1. Objetivos

Alineando la información con la aplicación BrainFit, el objetivo primordial de nuestro proceso de pruebas es asegurar la calidad, eficacia y seguridad de BrainFit. Buscamos identificar y resolver problemas antes de que la aplicación se lance al mercado. Los objetivos específicos para BrainFit incluyen:

Garantizar que todas las funciones de BrainFit trabajen según lo esperado.

Se debe confirmar que BrainFit funcione correctamente en una variedad de dispositivos y versiones de Android.

Validar que BrainFit sea fácil de usar y que ofrezca una experiencia de usuario positiva.

Se debe asegurar que BrainFit mantenga un rendimiento óptimo bajo diferentes condiciones.

Verificar que BrainFit sea seguro y proteja los datos de los pacientes.

Este proceso de pruebas tiene como finalidad mejorar la calidad y la experiencia del usuario de BrainFit, asegurando que la aplicación cumple con los estándares requeridos antes de su lanzamiento al mercado.

7.2. Alcance de las Pruebas

Las pruebas se llevarán a cabo en todos los componentes y funciones de las aplicaciones móviles BrainFit. Este proceso de pruebas abarcará los siguientes tipos de pruebas, con el objetivo de asegurar la calidad, eficacia y seguridad de BrainFit:

- Pruebas unitarias: Para garantizar que cada unidad funcione de acuerdo a lo esperado en BrainFit.
- Pruebas de interfaz de usuario: Con el fin de validar que BrainFit sea fácil de usar y ofrezca una experiencia positiva para los usuarios.
- Pruebas de compatibilidad: Para confirmar que BrainFit funcione correctamente en una variedad de dispositivos y versiones de Android.
- Pruebas de seguridad: Con el objetivo de verificar que BrainFit proteja los datos del usuario y que no existan vulnerabilidades.
- Pruebas de usabilidad: Para evaluar la facilidad de uso y la experiencia del usuario en BrainFit.

Durante este proceso, es posible que algunas pruebas identifiquen vulnerabilidades o aspectos a corregir. En estos casos, el equipo de desarrollo de BrainFit decidirá, según la priorización de riesgo, si es necesario mitigar el riesgo antes del lanzamiento de la aplicación.

7.3. Enfoques y Técnicas de las Pruebas

Nuestra estrategia de pruebas para BrainFit se basará en la combinación de pruebas manuales y automáticas, cada una con un propósito específico:

- Pruebas manuales: Se utilizarán para evaluar la usabilidad y la experiencia del usuario en BrainFit, ya que implican una evaluación subjetiva y perceptiva.
- Pruebas automáticas: Se emplearán para validar las funciones y características de BrainFit de manera exhaustiva y eficiente.

Para cada tipo de prueba, se desarrollarán casos de prueba específicos. Cada caso de prueba estará diseñado para evaluar un aspecto particular de BrainFit. La combinación de estos casos de prueba cubrirá la funcionalidad completa de la aplicación. Las pruebas se llevarán a cabo tanto en el entorno de desarrollo como en el de producción para garantizar que BrainFit funcione adecuadamente en ambos contextos. Es importante destacar que las pruebas detalladas en este documento se refieren a las implementadas en el entorno de producción, asegurando la calidad y confiabilidad de BrainFit antes de su lanzamiento al mercado.

8. Ejecución de las Pruebas

8.1. Prueba de Seguridad

8.1.1. Objetivos

- Identificación de vulnerabilidades y patrones de seguridad débiles mediante la aplicación de técnicas especializadas.
- Evaluación minuciosa de la configuración de seguridad de la aplicación, incluyendo permisos, configuraciones de red y almacenamiento, entre otros aspectos.
- Análisis exhaustivo del código fuente de la aplicación con el objetivo de detectar posibles problemas de seguridad.
- Verificación de la integridad y seguridad de las reglas configuradas en Firebase, garantizando la protección de los datos.
- Realización de pruebas de penetración básicas para evaluar la resistencia de la aplicación ante posibles amenazas y vulnerabilidades.
- Estas medidas se implementarán para fortalecer la seguridad de BrainFit y proteger la información sensible de los usuarios.

8.1.2. Prueba de Seguridad App Móvil

Las pruebas de seguridad tienen como objetivo verificar la capacidad de la aplicación para resistir posibles ataques de usuarios malintencionados, al mismo tiempo que confirman que los desarrolladores siguen prácticas de seguridad en el proceso de programación.(*La Importancia de Pruebas de Seguridad en Apps Móviles*, s. f.)

Para estas pruebas se utilizó MobSF. MobSF es una solución integral de evaluación de seguridad de aplicaciones móviles que facilita la realización de análisis tanto estáticos como dinámicos de binarios de aplicaciones móviles (como APK, IPA y APPX) y del código fuente comprimido.(Alexynior, 2020).

Gracias al empleo de la herramienta MobSF, hemos llevado a cabo un análisis exhaustivo y detallado de cada componente de la aplicación móvil. Este análisis abarca cada aspecto de la aplicación, desde su estructura general hasta sus funciones específicas, con el propósito de evaluar y mejorar su seguridad. Los resultados y observaciones se encuentran documentados en el archivo adjunto denominado "SeguridadAppMovil.pdf", el cual ofrece un desglose completo de las áreas evaluadas y las medidas recomendadas para fortalecer la seguridad de la aplicación móvil. Este informe proporciona información valiosa que ayudará a garantizar la integridad y la confidencialidad de los datos, así como la protección de la aplicación en su conjunto.

8.1.3. Prueba de Seguridad App Desktop

Bandit es una utilidad desarrollada con la finalidad de detectar las vulnerabilidades habituales de seguridad presentes en el código Python. Para lograrlo, Bandit examina cada archivo, construye un Árbol de Sintaxis Abstracta (AST) a partir de su contenido y aplica los plugins pertinentes a los nodos del AST. Una vez que Bandit ha concluido el escaneo de todos los archivos, produce un informe con los resultados obtenidos.(*Bienvenido a Bandit — Documentación de Bandit*, s. f.)

En el informe report.json se detalla los resultados del análisis estático de seguridad realizado en el código fuente del archivo "start.py" de una aplicación Python. Se complace en informar que el análisis no ha revelado problemas de seguridad significativos.

Durante el análisis estático, no se encontraron problemas críticos ni vulnerabilidades importantes en el código fuente del archivo "start.py", que es donde se realiza la debida autenticación. Esto indica que, en general, la seguridad del código es satisfactoria y que no hay riesgos significativos para la aplicación en este aspecto.

En el análisis no se identificó problemas de seguridad críticos, no se requieren acciones inmediatas de corrección. No obstante, se debe mantener una práctica continua de análisis de seguridad y buenas prácticas de codificación.

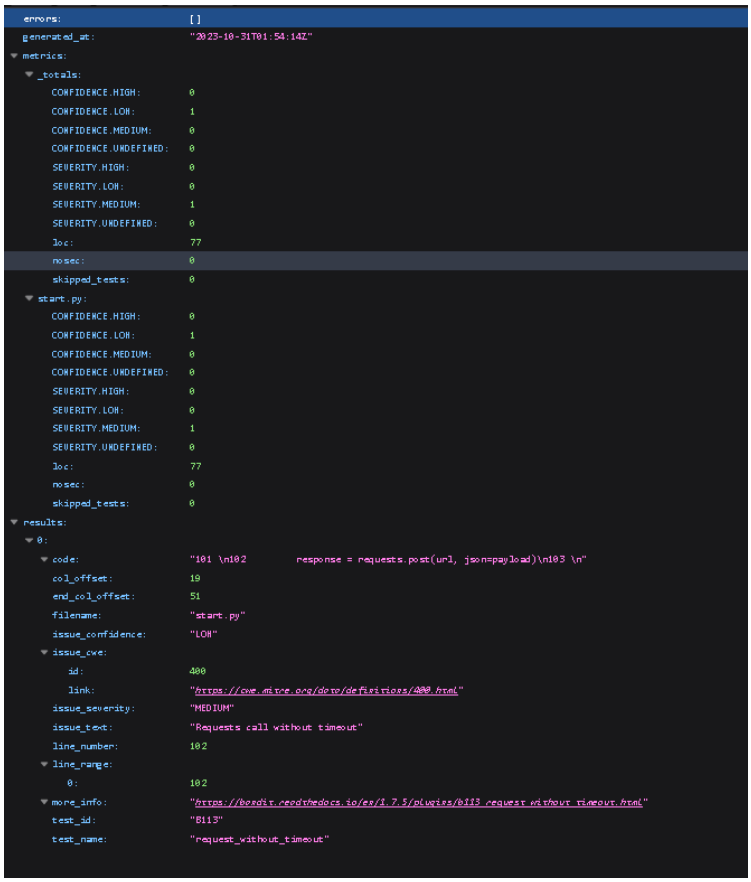

Ilustración 1. Prueba de seguridad App Desktop

8.1.4. Firebase

Se diseñaron reglas específicas con el propósito de garantizar que solo los usuarios que han completado el proceso de autenticación tengan la capacidad de escribir en el Firestore Database. Para una comprensión más detallada de estas restricciones, le invitamos a consultar la siguiente imagen, donde se pueden observar y apreciar en su totalidad las reglas establecidas:

```
rules_version = '2';
service cloud.firestore {
  match /databases/{database}/documents {
    match /pacientes/{id} {
      allow read: if true;
      allow write: if request.auth != null;
    }
  }
}
```

Ilustración 2. Reglas Firebase Database

Las siguientes reglas de Firebase Storage permiten que los usuarios autenticados puedan leer y escribir en cualquier ruta dentro del depósito de almacenamiento. Si un usuario no está autenticado, se le deniega el acceso. Estas reglas aseguran que solo los usuarios autenticados puedan acceder a los recursos de almacenamiento, lo que es una práctica común para proteger los datos y archivos en Firebase Storage.

```
service firebase.storage {
  match /b/{bucket}/o {
    match /{allPaths=**} {
      allow read, write: if request.auth != null;
    }
  }
}
```

Ilustración 3. Reglas Firebase Storage

8.2. Prueba de Compatibilidad

El propósito de las pruebas de compatibilidad es verificar que la aplicación se desempeña adecuadamente en versiones particulares de la plataforma. Esto resulta fundamental debido a que las actualizaciones de Flutter y Python, en cada una de sus nuevas versiones, podrían influir negativamente en el funcionamiento de la aplicación. Por lo tanto, es esencial comprobar que todos los elementos de la aplicación sean compatibles con las diversas versiones de Flutter y Python, y en caso de que no lo sean, es necesario examinar las razones detrás de la incompatibilidad.

8.2.1. Objetivos

- Evaluar si la aplicación móvil es compatible con diferentes versiones de Android.
- Evaluar si la aplicación de escritorio es compatible con diferentes versiones de windows.

8.2.2. Casos de Prueba

A continuación, se mostrarán los escenarios en los que se realizaron la prueba de compatibilidad para la app móvil:

Dispositivo	Versión Android	Resultado de Prueba
Vivo	Android 13	Aprobado
Xiaomi 11	Android 13	Aprobado
Pixel 5 API 28	Android 9	Aprobado
Pixel 6 API 24	Android 7	Aprobado
Samsung Galaxy J2	Android 7	Aprobado
Nexus 5X, LG	Android 7	Aprobado
Nexus 7	Android 7	Aprobado

Tabla 2. Caso de prueba celulares

A continuación, se mostrarán los escenarios en los que se realizaron la prueba de compatibilidad para la app de escritorio:

Versión Windows	Resultado de Prueba
Windows 11	Aprobado
Windows 10	Aprobado

Tabla 3. Caso de prueba windows

Se puede notar que ambas aplicaciones fueron diseñadas para operar en entornos distintos y ejecutarse de manera adecuada.

8.3. Prueba de Rendimiento

Las pruebas de rendimiento tienen como objetivo determinar el rendimiento del sistema bajo unas condiciones determinadas, en este caso esta prueba solo se realizó en la app móvil. Con el propósito de mejorar la experiencia del usuario al utilizar el software y prevenir situaciones como respuestas lentas al ejecutar acciones específicas, bloqueo de animaciones, un consumo excesivo de batería sin necesidad, y bloqueos inesperados.

Para esto, se quiso hacer el control de lo que es la memoria RAM y la CPU dos factores claves a la hora de ejecutar la app móvil.

Para realizar estas pruebas se utilizaron los siguientes dispositivos o emuladores:

Dispositivo	**Versión Android**	**RAM**	**Almacenamiento**	**Emulador o Dispositivo físico**
Vivo Y53s	Android 13	69.39 MB	81.92 MB	Dispositivo físico
Pixel 6 API 24	Android 7	201 MB	201 MB	Emulador
Pixel 5 API 28	Android 9	201 MB	201 MB	Emulador
Nexus 5X, LG	Android 7	150 MB	150 MB	Emulador

Tabla 4. Prueba de rendimiento celulares

8.3.1. Objetivos

- Medir el consumo de recursos del sistema al usar la aplicación BrainFit en dispositivos móviles Android.
- Medir el uso de memoria en la aplicación para que su consumo sea razonable.

8.3.2. Casos de Prueba

Dispositivo	Característica	Descripción
Vivo Y53s	RAM	El Vivo Y53s cuenta con 69.39 MB de RAM, lo que indica que tiene una cantidad relativamente baja de memoria RAM. Esta cantidad puede ser adecuada para tareas básicas como navegación web, redes sociales y aplicaciones de mensajería, pero puede ser insuficiente para ejecutar aplicaciones más exigentes o juegos con fluidez. Usuarios que buscan un rendimiento más ágil y multitarea avanzada podrían encontrar

		limitaciones con esta cantidad de RAM.
	Almacenamiento	En cuanto al almacenamiento, el Vivo Y53s ofrece 81.92 MB, lo que también es una cantidad bastante limitada. Esto significa que tendrás espacio limitado para almacenar aplicaciones, fotos, videos y otros archivos en el dispositivo. Los usuarios que deseen almacenar una cantidad significativa de datos en su dispositivo pueden sentirse restringidos por este espacio de almacenamiento.
Pixel 6 API 24	RAM	El Pixel 6 API 24 tiene 201 MB de RAM, lo que es una cantidad mayor en comparación con el Vivo Y53s. Esta cantidad de RAM es más adecuada para ejecutar aplicaciones y tareas más exigentes de manera fluida. Los usuarios podrán disfrutar de un rendimiento más suave y una mejor multitarea en este dispositivo.
	Almacenamiento	El Pixel 6 API 24 también cuenta con 201 MB de almacenamiento, lo cual, aunque es igual en tamaño al almacenamiento de RAM, probablemente sea un error o dato inusual, ya que los dispositivos suelen tener una cantidad mucho mayor de almacenamiento que de RAM. Es importante tener en cuenta que, en la mayoría de los

		casos, se esperaría que el almacenamiento sea en gigabytes (GB) en lugar de megabytes (MB).
Pixel 5 API 28	RAM	El Pixel 5 API 28 dispone de 201 MB de RAM, que es una cantidad bastante baja en comparación con la mayoría de los dispositivos modernos. Esta cantidad limitada de RAM puede resultar insuficiente para realizar tareas avanzadas o ejecutar aplicaciones y juegos exigentes con fluidez. Los usuarios pueden experimentar retrasos y limitaciones en la multitarea debido a esta cantidad de RAM
	Almacenamiento	Al igual que la RAM, el Pixel 5 API 28 también tiene 201 MB de almacenamiento. Al igual que mencioné antes, esta cantidad de almacenamiento parece inusualmente baja. Los dispositivos típicos suelen tener varias decenas o incluso cientos de gigabytes (GB) de almacenamiento, en lugar de megabytes (MB). Dado que esta cifra es muy baja, es importante verificar si se trata de datos correctos o de un valor atípico.
Nexus 5X, LG	RAM	El Nexus 5X tiene 150 MB de RAM, lo que indica una cantidad extremadamente limitada de memoria RAM. Esta cantidad es inusualmente baja y, en la práctica, resultaría en un rendimiento muy

		limitado para la mayoría de las tareas y aplicaciones modernas. Los dispositivos actuales suelen tener varios gigabytes de RAM para ofrecer un rendimiento óptimo en una amplia gama de aplicaciones y tareas.
	Almacenamiento	Al igual que la RAM, el almacenamiento del Nexus 5X también es de 150 MB, lo cual es inusualmente bajo para un dispositivo moderno. Los dispositivos típicos tienen varios gigabytes de almacenamiento para permitir a los usuarios almacenar aplicaciones, fotos, videos y otros datos de manera efectiva. Con solo 150 MB de almacenamiento, el espacio disponible sería insuficiente para la mayoría de las necesidades de almacenamiento.

Tabla 5. Caso de prueba rendimiento

8.4. Prueba de Funcionalidad

El propósito de las pruebas de funcionalidad es evaluar si un sistema o una aplicación cumple con sus requisitos funcionales y opera de acuerdo con las especificaciones y expectativas definidas. Las pruebas funcionales se centran en verificar que el software realiza las funciones y tareas para las cuales fue diseñado de manera adecuada

8.2.1. Objetivos

- Evaluar si la aplicación de escritorio descarga de manera correcta los videos del firebase.
- Evaluar si la aplicación de escritorio realiza correctamente el login y reconoce cuando un usuario no es correcto.
- Evaluar si la aplicación de escritorio se conecta correctamente con el firebase.

8.2.2. Casos de Prueba

A continuación, se mostrarán los escenarios en los que se realizaron la prueba de funcionalidad para la aplicación de escritorio:

Caso 1:

Para este caso se llamó la función download_test y se le envió una ruta, un ide de algún paciente existente y el bucket de python y la prueba salió exitosa, esto significa que la función si descarga los videos de manera correcta.

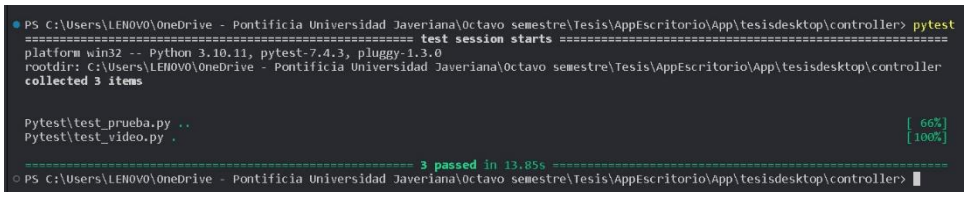

Ilustración 4. Prueba unitaria download_test

Caso 2:

En este caso de prueba se quería comprobar que la aplicación realizara de manera correcta el login y la conexión con el firebase:

```python
# Fixture para configurar el objeto mock para simular una respuesta de error
@pytest.fixture
def failed_login_mock():
    with patch('requests.post') as mock_post:
        mock_post.return_value.status_code = 400
        mock_post.return_value.json.return_value = {"error": "Invalid password"}
        yield mock_post

def test_login_successful(successful_login_mock):
    email = "tatis.gaona1942@gmail.com"
    password = "1234567"
    response = login(email, password)

    assert response.status_code == 200
    assert response.json() == {"idToken": "your_token"}

def test_login_failure(failed_login_mock):
    email = "test@example"
    password = "invalid_password"
    response = login(email, password)

    assert response.status_code == 200
    assert response.json() == {"error": "Invalid password"}
```

Ilustración 5. Prueba unitaria login

En este caso realizamos dos pruebas en la función “login_successfull” se ingresa una credencial correcta y en login_failure al pasarle un usuario incorrecto la prueba falla ya que este usuario no existe.

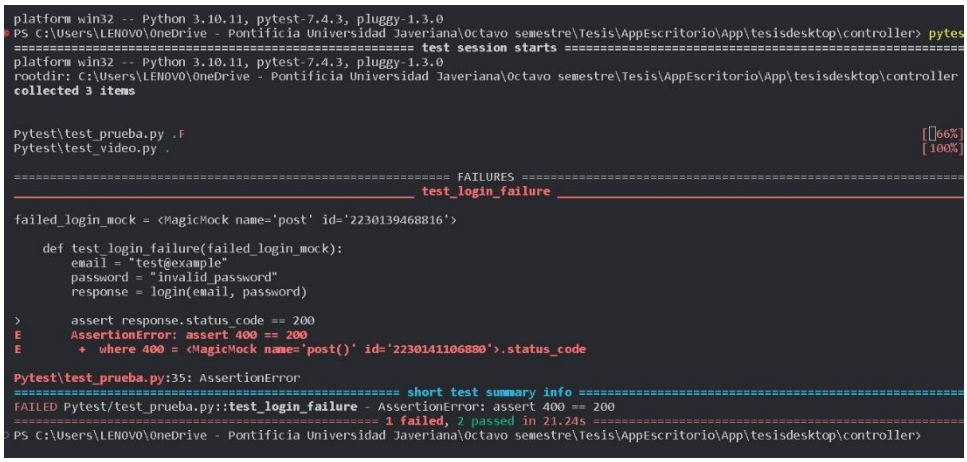

Ilustración 6. Prueba unitaria login_successfull

8.5. Prueba de usabilidad

El propósito de las pruebas de funcionalidad es evaluar la usabilidad de un sitio web, una aplicación móvil u otro producto interactivo. Estas pruebas implican la recopilación de datos y retroalimentación de los usuarios a través de formularios diseñados específicamente con preguntas relacionadas con la experiencia del usuario.

8.2.1. Objetivos

- Evaluar si la aplicación de escritorio y móvil es fácil de utilizar.
- Evaluar si la aplicación de escritorio y móvil cuenta con una interfaz intuitiva.
- Evaluar si la aplicación de escritorio y móvil es fácil de navegar.
- Evaluar si la aplicación de escritorio y móvil realizan las tareas específicas.
- Evaluar si la aplicación de escritorio y móvil cuenta con una interfaz agradable.
- Evaluar si la aplicación de escritorio y móvil cuentan con todas las funcionalidades esperadas.

8.2.2. Casos de Prueba

A continuación, se mostrarán los escenarios en los que se realizaron la prueba de funcionalidad para la app de escritorio y móvil:

1. ¿Con qué facilidad aprendiste a usar la aplicación?

App móvil:

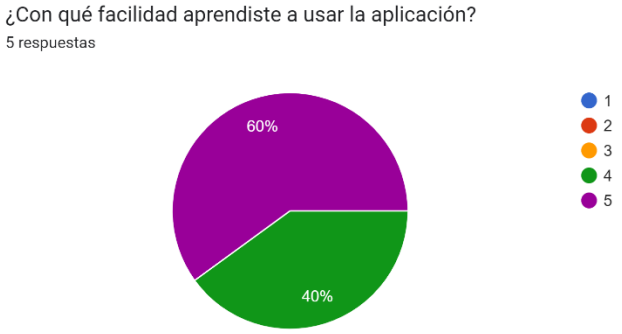

Ilustración 7. Prueba usabilidad app Móvil 1

App de escritorio:

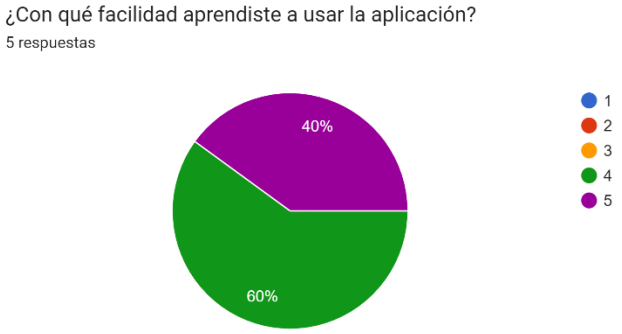

Ilustración 8. Prueba usabilidad app de Escritorio 1

2. ¿Encuentras la interfaz de la aplicación intuitiva?

App móvil:

Ilustración 9. Prueba usabilidad app Móvil intuitiva

App de escritorio:

Ilustración 10. Prueba usabilidad app de escritorio intuitiva

3. ¿Te resulta fácil navegar por la aplicación?

App móvil:

Ilustración 11. Prueba usabilidad app Móvil navegabilidad

App de escritorio:

Ilustración 12. Prueba usabilidad app de escritorio intuitiva

4. ¿La aplicación facilita la realización de tareas específicas?

App móvil:

Ilustración 13. Prueba usabilidad app Móvil tareas específicas

App de escritorio:

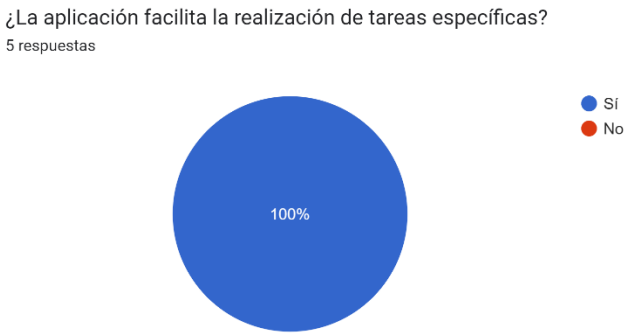

Ilustración 14. Prueba usabilidad app de Escritorio tareas específicas

5. ¿Te gusta el diseño general de la aplicación?

App móvil:

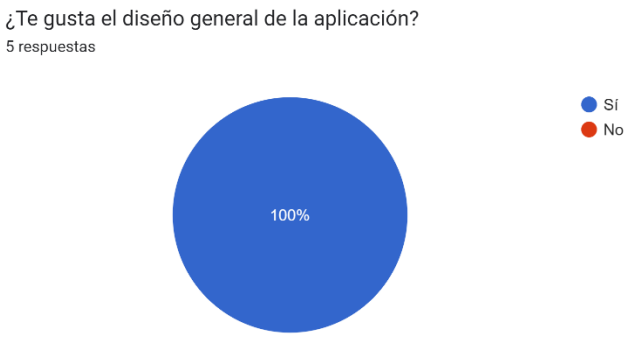

Ilustración 15. Prueba usabilidad app Móvil diseño general

App de escritorio:

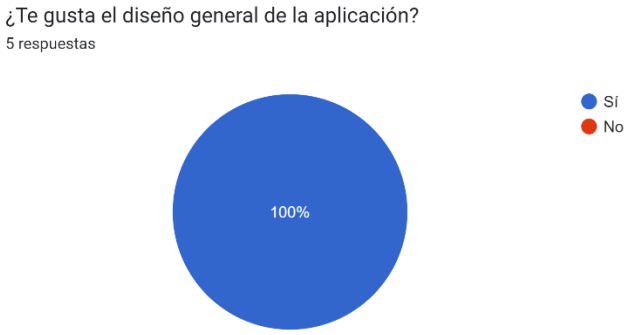

Ilustración 16. Prueba usabilidad app de Escritorio diseño general

6. ¿La aplicación ofrece todas las funciones que esperabas?

App móvil:

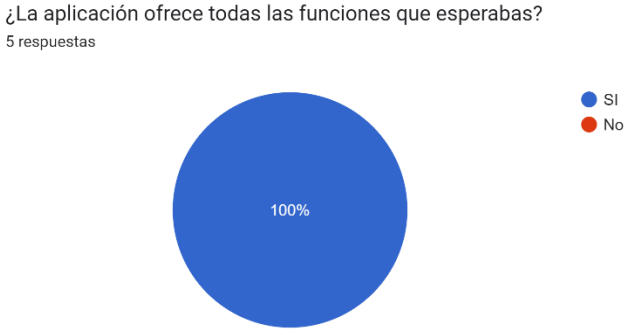

Ilustración 17. Prueba usabilidad app Móvil funciones

App de escritorio:

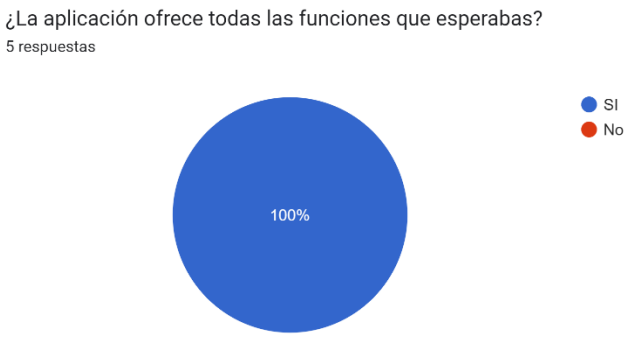

Ilustración 18. Prueba usabilidad app de Escritorio funciones

En ambas aplicaciones, se observa de manera consistente que las respuestas a la primera pregunta se sitúan en un rango que va de 4 a 5, donde 5 representa el puntaje más alto y denota una facilidad de uso excepcional. Además, en el caso de las demás preguntas, se destaca que el 100% de la muestra de usuarios coinciden al otorgar respuestas positivas en todas las cuestiones planteadas. Estos patrones de evaluación confirman de manera concluyente que ambas aplicaciones son altamente intuitivas y fáciles de utilizar desde la perspectiva de los usuarios.

Este descubrimiento es de gran importancia, ya que indica que ambas aplicaciones han alcanzado con éxito uno de los objetivos primordiales del diseño de interfaces de usuario: garantizar la accesibilidad y la facilidad en la interacción. La coherencia en las calificaciones sugiere que los usuarios experimentan una experiencia de usuario uniforme y exenta de obstáculos al interactuar con estas aplicaciones. Esta uniformidad es esencial para retener a los usuarios y aumentar su satisfacción, ya que una interfaz de fácil uso reduce la resistencia y mejora la eficiencia en la realización de tareas, lo que, a su vez, optimiza la percepción global del producto.

Estos resultados también reflejan un diseño de interfaz bien concebido que incorpora con éxito los principios de usabilidad y la retroalimentación de los usuarios durante todo el proceso de desarrollo. La simplicidad en la navegación, la claridad en la presentación de información y la eficacia en la realización de tareas son elementos esenciales que contribuyen a la obtención de altas calificaciones en términos de facilidad de uso.

8.6. Prueba de carga

Las pruebas de carga son un tipo de prueba de rendimiento que se realiza en sistemas, aplicaciones o infraestructuras para evaluar cómo se comportan bajo condiciones de carga específicas. Estas pruebas son esenciales para garantizar que un sistema pueda manejar la cantidad de usuarios, transacciones o datos previstos sin comprometer su rendimiento o su capacidad para funcionar de manera eficiente.

8.2.1. Objetivos

- Realizar una evaluación para determinar la capacidad de la capa gratuita de Firebase para almacenar y procesar el número máximo de pacientes por día que puede gestionar al subir datos a la nube.

8.2.2. Caso de Prueba

- Se llevaron a cabo múltiples pruebas, cada una consistiendo en la carga de 4 videos y 2 grabaciones de audio por paciente, con el propósito de evaluar la capacidad de la capa gratuita de Firebase. Los resultados indicaron que fue posible subir exitosamente información correspondiente a 14 pacientes en un solo día, cumpliendo con la restricción diaria establecida por Firebase para la carga en su servicio de almacenamiento en la nube de forma gratuita.

8.7. Prueba de estrés

Como complemento de las pruebas de rendimiento y de carga se desarrolló pruebas de estrés desarrollada por medio de Firebase Test Lab ya que la aplicación no cuenta con solicitudes tipo Rest. Por medio de esta herramienta se simulo la aplicación en distintos disipativos virtuales dados por la herramienta, uno de estos fue en un Pixel 5. La prueba busca encontrar la respuesta que tiene el sistema en casos donde se ponga en condiciones de estrés, como son condiciones no ideales y pueden perjudicar el sistema, encontrar características de rendimiento de la aplicación en dichas circunstancias.

8.2.1. Objetivos

- Realizar la evaluación de características de rendimiento sobre la aplicación a partir de condiciones específicas que ponen el sistema en estrés.

8.2.2. Caso de Prueba

- Se llevaron a cabo a partir de Firebase Test Lab, donde por medio de dispositivos virtuales que corrían la aplicación, en este caso un Pixel 5, obtiene características de rendimiento en una condición de estrés, en este caso la condición es intento de acceso denegado ingresando credenciales falsas. Se abordo el caso de intento de ingreso a la aplicación sin autorización, por medio de 119 acciones automáticas en la actividad y dio como resultado aprobado ya que no se logró caer o vulnerar el sistema, adicional a esto se llegó a que tardo 446ms para la carga total de la interfaz, además con una latencia de entrada alta del 9% y un subproceso de la IU lento del 13%.