

INDICE

1. [**Alcance de las pruebas** **3**](#_heading=h.gjdgxs)
2. **Descripcion 5**
3. **Objetivos del plan de prueba 5**
4. [**Módulos del software a probar** **6**](#_heading=h.30j0zll)
5. [**Detalle de Ejecución de las pruebas** **8**](#_heading=h.1fob9te)
6. [**Tipos de Prueba a realizar** **10**](#_heading=h.3znysh7)
7. [**Técnicas y estrategias de pruebas a utilizar** **13**](#_heading=h.2et92p0)
8. [**Roles y responsabilidades** **15**](#_heading=h.tyjcwt)
9. [**Entorno y configuración de ambiente de pruebas** **17**](#_heading=h.3dy6vkm)

1.- ALCANCE DE LAS PRUEBAS

**Objetivo**: El alcance de las pruebas para la app MortonShoes incluye pruebas funcionales de aplicaciones básicas y pruebas de interfaz de usuario (UI) para garantizar que el sistema cumpla con los requisitos específicos. Esta área incluye funciones clave del módulo de administración y usuario, así como aspectos de la experiencia de compras y gestión de productos.

**Características incluidas en el área de prueba**

**Autenticación y manejo de usuarios**

**Registro de usuarios**: Garantiza que los usuarios puedan registrarse con información válida y que el sistema almacene los datos correctamente en la base de datos.

**Iniciar sesión**: asegúrese de que los usuarios registrados puedan autenticarse correctamente y que el sistema maneje el estado de la sesión de forma segura y eficiente.

**Gestión de perfiles de usuario**: asegúrese de que los usuarios puedan ver y actualizar la información de su perfil. (nombre, dirección, dirección de correo electrónico, etc.) y estos cambios se reflejan inmediatamente en el sistema.

**Interactuar con el carrito de compras**

**Agregar artículos al carrito de compras**: Asegúrese de que el usuario pueda agregar la cantidad seleccionada de productos al carrito y que el contenido del carrito esté actualizado.

**Mostrar y eliminar del carrito**: asegúrese de que los usuarios puedan ver todos los productos agregados al carrito, ver el precio total y eliminar el producto si es necesario.

**Paga**r: Asegúrese de que el proceso de pago funcione correctamente registrando la compra en la base de datos y creando una nueva entrada en blanco para el siguiente carrito de compras.

**Gestión de productos del administrador**

**Crear, modificar y eliminar productos**: asegúrese de que los administradores puedan agregar nuevos productos, editar información sobre productos existentes y eliminar productos. Estas acciones deben reflejarse inmediatamente en todos los espacios de la tienda.

**Vista de inventario**: garantiza que los administradores tengan acceso a una vista completa del inventario y que todos los cambios se guarden en la base de datos.

**Gestión de compras**: asegúrese de que los administradores puedan ver todas las compras registradas. Incluya detalles sobre los productos y los usuarios que los crearon.

**Filtrar y buscar productos**

**Filtrar por rango de precio y tamaño**: compruebe que los filtros de precio y tamaño en la página del producto funcionan correctamente y muestran solo los productos que coinciden con sus criterios de selección.

**Barra de búsqueda**: garantiza que los usuarios puedan buscar productos por nombre y que los resultados sean precisos y reflejen los términos de búsqueda.

Interfaz de usuario (UI) y experiencia de usuario (UX).

**Navegación**: asegúrese de que todas las páginas se carguen correctamente para que los usuarios puedan moverse entre ellas sin errores.

**Interacción de la interfaz de usuario**: asegúrese de que los botones, menús y formularios respondan correctamente a la entrada del usuario.

**Mensajes de retroalimentación**: asegúrese de que el sistema proporcione mensajes claros y útiles, especialmente para operaciones importantes como autenticación, eliminación y extracción de productos.

**Características fuera del alcance de las pruebas**

**Rendimiento y carga**

**Rendimiento** (tiempo de respuesta) y carga (potencia de procesamiento) probados por múltiples usuarios simultáneamente no serán considerados durante esta fase del proyecto.

**Pruebas de seguridad avanzadas**

Este ciclo de pruebas no incluye pruebas de seguridad avanzadas, como pruebas de penetración o cifrado de datos. Sólo se prueba la seguridad de autenticación básica.

Integración externa

La integración con servicios externos (si se planifica en el futuro) está fuera del alcance de esta fase de prueba. Las pruebas se centran en la funcionalidad interna de la aplicación.

Compatibilidad con dispositivos y navegadores.

Este ciclo de pruebas no incluye pruebas exhaustivas de compatibilidad entre dispositivos y navegadores. Las pruebas se llevarán a cabo en un entorno específico con una configuración de equipo estándar.

**2.- DESCRIPCION**

La aplicación **MortonShoes** es una plataforma móvil diseñada para permitir a los usuarios explorar, seleccionar y comprar zapatillas deportivas. La aplicación tiene dos roles de usuarios: **clientes** y **administradores**, cada uno con funcionalidades y permisos distintos que permiten realizar y editar los productos de la tienda

El propósito principal de esta aplicación es facilitar a los clientes la compra de zapatillas desde cualquier lugar, ofreciéndoles una interfaz amigable donde pueden navegar por una variedad de productos, agregar productos a su carrito y finalizar la compra de manera segura. A su vez, proporciona a los administradores herramientas para gestionar el inventario, controlar la disponibilidad de productos y monitorear las compras realizadas.

**3.-OBJETIVOS DEL PLAN DE PRUEBAS**

El propósito de probar esta aplicación es verificar que toda la funcionalidad anterior funcione según lo especificado en los requisitos funcionales. Las pruebas están diseñadas para garantizar que:

Los clientes puedan completar todo el proceso de compra sin interrupciones ni errores.

La gestión de productos es eficiente y permite agregar, cambiar y eliminar productos con precisión.

Las funciones de búsqueda y filtrado son normales y proporcionan a los usuarios información precisa. y resultados correspondientes.

La autenticación y la gestión de perfiles funcionan de forma segura y fluida.

Los datos se almacenan correctamente en la base de datos SQLite y se actualizan en tiempo real dentro de la aplicación.

A través de esta descripción, el equipo de pruebas y otras partes interesadas pueden comprender claramente el propósito de la aplicación, es decir, las principales funciones y objetivos que la prueba intenta lograr en términos de rendimiento y usabilidad.

**4.-MODULOS DEL SOFTWARE A PROBAR**

**Módulo de Autenticación**

* **Descripción**: Este módulo permite el acceso seguro de los usuarios y administradores a la aplicación, proporcionando las funciones de registro, inicio de sesión y cierre de sesión. Incluye validaciones de credenciales y persistencia del estado de sesión mediante localStorage.
* **Funciones Principales**:
  + Registro de nuevos usuarios con validación de datos.
  + Inicio de sesión con autenticación de credenciales.
  + Cierre de sesión y limpieza del estado de autenticación.

**Módulo de Gestión de Perfil de Usuario**

* **Descripción**: Permite a los usuarios ver y actualizar su información personal, incluyendo datos como nombre, correo electrónico y dirección. Las actualizaciones deben reflejarse en tiempo real.
* **Funciones Principales**:
  + Visualización de datos de perfil (nombre, dirección, correo).
  + Actualización de información personal y persistencia de cambios en la base de datos.

**Módulo de Catálogo de Productos**

* **Descripción**: Este módulo muestra el catálogo completo de productos (zapatillas) en un formato visual de tarjetas, permitiendo a los usuarios visualizar detalles específicos de cada producto. Incluye filtros por precio y talla, así como una barra de búsqueda.
* **Funciones Principales**:
  + Listado de productos en tarjetas con imagen, precio, talla y marca.
  + Filtros por rango de precios y tallas.
  + Barra de búsqueda por nombre de producto.

**Módulo de Carrito de Compras**

* **Descripción**: Permite a los usuarios agregar productos al carrito, ver los artículos seleccionados, modificar cantidades y eliminar productos. Este módulo también calcula el precio total de los productos en el carrito y permite al usuario finalizar la compra.
* **Funciones Principales**:
  + Agregar productos al carrito desde el catálogo.
  + Visualización y edición de cantidades de los productos en el carrito.
  + Eliminación de productos y cálculo del total acumulado.
  + Finalización de compra y registro en la base de datos.

**Módulo de Historial de Compras**

* **Descripción**: Este módulo permite a los usuarios revisar sus compras anteriores, mostrando detalles de cada compra, incluyendo fecha, productos adquiridos y el total de la transacción.
* **Funciones Principales**:
  + Listado de compras anteriores por usuario.
  + Visualización de los detalles de cada compra (productos, cantidades, fecha y total).

**Módulo de Gestión de Productos (Administrador)**

* **Descripción**: Este módulo está disponible solo para el administrador y permite la gestión completa del inventario de productos en la tienda. Incluye funciones para añadir nuevos productos, editar detalles de productos existentes y eliminar productos del catálogo.
* **Funciones Principales**:
  + Creación de productos con detalles como nombre, marca, talla, precio, cantidad y imagen.
  + Modificación de detalles de productos ya existentes.
  + Eliminación de productos del inventario y actualización en todas las páginas.

**Módulo de Control de Compras (Administrador)**

* **Descripción**: Permite al administrador visualizar todas las compras realizadas en la tienda, incluyendo los detalles de cada transacción y los datos del usuario que realizó la compra.
* **Funciones Principales**:
  + Listado de todas las compras realizadas en la tienda.
  + Visualización de detalles de cada compra, incluyendo usuario, productos comprados y cantidades.

**5.- detalle de ejecución de las pruebas**

**5.1Pruebas Funcionales**

Se validará que todas las funcionalidades principales de la aplicación MortonShoes operen según lo especificado en los requisitos, incluyendo:

* Autenticación y Gestión de Usuarios: Comprobar el registro, inicio de sesión y edición de perfil de los usuarios.
* Catálogo de Productos: Validar la visualización, búsqueda y filtrado de productos por precio y talla.
* Carrito de Compras: Verificar que los usuarios puedan agregar productos, modificar cantidades, y eliminar artículos del carrito.
* Proceso de Compra: Asegurar que los usuarios puedan finalizar compras y que estas se registren correctamente en la base de datos.
* Gestión de Productos (Administrador): Validar las funcionalidades de añadir, editar y eliminar productos por parte del administrador.
* Control de Compras (Administrador): Confirmar que el administrador pueda acceder a un listado completo de compras con detalles de cada transacción.

**5.2. Pruebas de Integración**

Se evaluará la interacción entre los módulos clave de la aplicación para asegurar que funcionen de forma coherente y continua. Las pruebas de integración incluirán:

* Autenticación y Carrito de Compras: Verificar que un usuario autenticado pueda acceder al carrito, agregar productos y realizar la compra sin problemas.
* Catálogo y Carrito de Compras: Validar que los productos seleccionados desde el catálogo se transfieran correctamente al carrito y mantengan los datos de cantidad, precio y talla.
* Carrito de Compras y Proceso de Pago: Asegurar que al finalizar la compra, el proceso de pago se registre correctamente en el módulo de historial y se vacíe el carrito.
* Gestión de Productos e Inventario: Comprobar que los cambios realizados por el administrador en el inventario (añadir, modificar o eliminar productos) se reflejen en tiempo real en el catálogo.

**5.3. Pruebas de Usabilidad**

Se evaluará la experiencia de usuario (UX) y la interfaz de usuario (UI) para asegurar que la aplicación sea intuitiva, fluida y fácil de navegar:

* Navegación: Verificar que los usuarios puedan navegar entre el catálogo, el carrito, y las secciones de perfil sin problemas.
* Facilidad de Uso: Validar que el flujo de compra sea claro y accesible para usuarios sin experiencia técnica.
* Mensajes de Retroalimentación: Comprobar que se muestren mensajes claros y útiles en las interacciones, especialmente en el registro, autenticación, proceso de compra y gestión de productos.
* Compatibilidad en Dispositivos: Asegurar que la aplicación se vea y funcione correctamente en dispositivos de distintas resoluciones y tamaños de pantalla.

**5.4. Pruebas de Rendimiento**

Se evaluará el rendimiento general de la aplicación, incluyendo tiempos de respuesta y estabilidad bajo distintas condiciones de carga:

* Tiempo de Carga: Medir el tiempo de carga inicial de la aplicación y la rapidez de respuesta en cada módulo principal.
* Rendimiento en el Carrito y Proceso de Pago: Asegurar que el sistema pueda manejar múltiples productos en el carrito sin afectar la velocidad de respuesta.
* Estabilidad Bajo Carga: Simular un número elevado de usuarios para observar la estabilidad y respuesta de la aplicación.

**5.5. Documentación de Resultados**

Se realizará un registro detallado de todos los resultados obtenidos durante la ejecución de cada prueba:

* Registro de Resultados: Documentar el rendimiento y las funcionalidades de la aplicación en cada módulo evaluado.
* Captura de Defectos: Registrar, describir y documentar cualquier defecto encontrado, incluyendo pasos para reproducirlo, evidencias (como capturas de pantalla), y el impacto en la funcionalidad de la aplicación.

**5.6. Revisión y Corrección**

Una vez identificados los defectos, se tomarán las siguientes acciones:

* Análisis de Defectos: Revisar cada defecto documentado para determinar su causa raíz y asignar prioridades para la corrección.
* Corrección por el Equipo de Desarrollo: Los defectos identificados se comunicarán al equipo de desarrollo para su resolución, y se hará un seguimiento para monitorear el progreso de las correcciones.
* Re-Pruebas: Ejecutar nuevas pruebas sobre las áreas afectadas por defectos para confirmar que se han corregido adecuadamente y que las correcciones no introducen nuevos problemas.

5.7**. Cierre de Pruebas**

Al finalizar el ciclo de pruebas, se realizarán los siguientes pasos para cerrar el proceso de pruebas:

* Informe de Pruebas: Preparar un informe final que resuma los resultados de las pruebas, los defectos encontrados, y las correcciones realizadas, proporcionando una visión completa del estado de la aplicación.
* Evaluación Final: Realizar una revisión exhaustiva de la aplicación en su estado final para asegurar que cumple con todos los requisitos especificados y que está lista para su lanzamiento.

**6.- tipos de prueba a realizar**

**6.1. Pruebas Funcionales**

* Descripción: Estas pruebas aseguran que cada funcionalidad de la aplicación funcione de acuerdo con los requisitos especificados.
* Objetivo: Verificar que las características principales, como la autenticación, la gestión del carrito de compras, la navegación entre páginas y el proceso de pago, funcionen correctamente.
* Ejemplos de Pruebas:
  + Registro y autenticación de usuarios.
  + Gestión de perfil del usuario.
  + Agregar productos al carrito y finalizar la compra.
  + Filtrado y búsqueda de productos.

**6.2. Pruebas de Integración**

* Descripción: Se evalúan las interacciones entre los distintos módulos de la aplicación para asegurar que se comuniquen de manera correcta.
* Objetivo: Comprobar que los módulos, como la autenticación y el carrito, funcionen bien juntos y que no haya fallos en la integración.
* Ejemplos de Pruebas:
  + Verificar la integración entre el carrito de compras y el proceso de pago.
  + Comprobar que los productos añadidos o eliminados por el administrador se reflejen en el catálogo.
  + Asegurar que los datos del perfil de usuario se sincronicen en todas las secciones.

**6.3. Pruebas de Usabilidad**

* Descripción: Estas pruebas evalúan la experiencia de usuario y la facilidad de uso de la aplicación.
* Objetivo: Garantizar que la aplicación sea fácil de navegar, intuitiva y que los usuarios puedan realizar las tareas sin problemas.
* Ejemplos de Pruebas:
  + Evaluar la navegación entre las distintas páginas de la aplicación.
  + Asegurar que el proceso de compra sea claro y sencillo.
  + Verificar la claridad de los mensajes de retroalimentación (por ejemplo, al agregar productos al carrito o al registrar una cuenta).

**6.4. Pruebas de Rendimiento**

* Descripción: Estas pruebas miden el rendimiento de la aplicación, incluyendo tiempos de carga y la estabilidad bajo carga.
* Objetivo: Asegurar que la aplicación funcione de manera eficiente incluso cuando haya múltiples usuarios o gran cantidad de datos.
* Ejemplos de Pruebas:
  + Medir el tiempo de carga de la aplicación y de los distintos módulos.
  + Evaluar el rendimiento del carrito y el proceso de pago cuando se agregan varios productos.
  + Comprobar la respuesta del sistema bajo condiciones de carga elevada simulando múltiples usuarios.

**6.5. Pruebas de Compatibilidad**

* Descripción: Estas pruebas aseguran que la aplicación funcione correctamente en diferentes dispositivos y plataformas.
* Objetivo: Verificar que la aplicación sea compatible con distintas versiones de dispositivos Android y posibles tamaños de pantalla.
* Ejemplos de Pruebas:
  + Probar la aplicación en diferentes versiones de Android.
  + Verificar que la interfaz se adapte a pantallas de distintos tamaños y resoluciones.

**6.6. Pruebas de Seguridad**

* Descripción: Estas pruebas verifican que la aplicación proteja adecuadamente la información del usuario y sea resistente a posibles ataques.
* Objetivo: Asegurar que la autenticación, los datos personales y las transacciones estén protegidas.
* Ejemplos de Pruebas:
  + Verificar la protección de contraseñas y datos personales almacenados.
  + Comprobar que las sesiones de usuario se gestionen de forma segura.
  + Validar que no se pueda acceder a secciones restringidas (como la administración) sin permisos adecuados.

**6.7. Pruebas de Regresión**

* Descripción: Estas pruebas aseguran que las funcionalidades anteriores sigan funcionando después de realizar cambios en el código.
* Objetivo: Detectar cualquier problema introducido por las nuevas actualizaciones o correcciones.
* Ejemplos de Pruebas:
  + Re-ejecutar casos de prueba previos después de realizar ajustes en el código.
  + Asegurar que las correcciones a defectos previos no afecten a otras funcionalidades.

**6.8. Pruebas de Interfaz de Usuario (UI)**

* Descripción: Evalúan la apariencia y comportamiento de los elementos visuales de la aplicación.
* Objetivo: Asegurar que la interfaz de usuario cumpla con los estándares visuales y que todos los componentes funcionen correctamente.
* Ejemplos de Pruebas:
  + Verificar que los botones, formularios y enlaces funcionen según lo esperado.
  + Asegurar la consistencia de colores, tipografías y tamaños de fuente en toda la aplicación.
  + Comprobar que los elementos visuales sean accesibles y fáciles de usar.

**6.9. Pruebas de Base de Datos**

* Descripción: Estas pruebas validan la integridad y consistencia de los datos almacenados en la base de datos.
* Objetivo: Asegurar que los datos se almacenen y recuperen correctamente y que la base de datos mantenga la integridad tras operaciones como actualizaciones o eliminaciones.
* Ejemplos de Pruebas:
  + Verificar que el carrito de compras mantenga los datos al agregar o eliminar productos.
  + Comprobar que las actualizaciones de perfil de usuario se reflejen correctamente en la base de datos.
  + Validar que las compras finalizadas se registren con todos los detalles en la base de datos.

**7.- TECNICAS Y ESTRATEGIAS DE LAS PREUBAS A REALIZAR**

**Caja Negra**

* Descripción: Evalúa las funcionalidades del sistema sin considerar su implementación interna, centrando las pruebas en las entradas y salidas.
* Objetivo: Validar que cada funcionalidad de la aplicación cumple con los requisitos y proporciona los resultados esperados.
* Aplicación en el Proyecto:
  + Registro e inicio de sesión de usuarios.
  + Agregar productos al carrito y finalizar la compra.
  + Filtrado y búsqueda de productos en el catálogo.

**Caja Blanca**

* Descripción: Analiza la estructura interna del código para asegurar que se cubran todos los caminos de ejecución importantes.
* Objetivo: Verificar que el código siga las rutas lógicas y ejecute todas las condiciones posibles.
* Aplicación en el Proyecto:
  + Validar la lógica en funciones críticas, como el cálculo del precio total en el carrito y la autenticación de usuarios.
  + Comprobar la implementación de reglas de negocio, como los límites en cantidades de productos y permisos de administrador.

**Partición de Equivalencia**

* Descripción: Divide las entradas en clases de equivalencia y crea casos de prueba representativos para cada clase.
* Objetivo: Reducir el número de casos de prueba al cubrir grupos representativos de entradas válidas e inválidas.
* Aplicación en el Proyecto:
  + Para el registro de usuario, validar entradas de contraseñas válidas e inválidas (longitud, caracteres especiales, etc.).
  + Probar el rango de precios en el filtro del catálogo con valores dentro y fuera del límite especificado.

**Análisis de Valores Límite**

* Descripción: Prueba los valores en los límites de las entradas, donde es más probable que ocurran errores.
* Objetivo: Detectar posibles errores en los bordes de las entradas permitidas.
* Aplicación en el Proyecto:
  + Verificar el filtro de precios usando el valor mínimo, el valor máximo y valores justo fuera del rango.
  + Validar límites de cantidad de productos al agregar al carrito, probando cantidades mínimas (por ejemplo, 1) y máximas posibles.

**Pruebas de Tablas de Decisión**

* Descripción: Se utiliza una tabla de decisiones para mapear todas las combinaciones posibles de condiciones y sus respectivas salidas.
* Objetivo: Asegurar que el sistema reaccione adecuadamente a combinaciones de condiciones específicas.
* Aplicación en el Proyecto:
  + Combinación de rol del usuario y acciones (por ejemplo, que solo el administrador pueda acceder a la edición de productos).
  + Validación de estados en el proceso de compra, como la existencia de productos en el carrito para permitir la compra.

**Pruebas Basadas en Escenarios**

* Descripción: Simula casos de uso completos, siguiendo el flujo típico del usuario para cumplir un objetivo.
* Objetivo: Evaluar el comportamiento del sistema en condiciones reales y asegurar una experiencia de usuario fluida.
* Aplicación en el Proyecto:
  + Realizar un flujo completo de compra desde la selección de productos hasta la finalización de la compra.
  + Simular el flujo de administración de productos, donde el administrador añade, edita y elimina productos.

**8.- ROLES Y RESPONSABILIDADES**

Nicolás Marín – Líder de Pruebas y Analista de Pruebas Funcionales

* Responsabilidades:
  + Planificación de Pruebas: Definir el plan de pruebas, estrategias y técnicas a utilizar, incluyendo los casos de prueba detallados para cada funcionalidad.
  + Ejecución de Pruebas Funcionales: Encargado de realizar pruebas funcionales para validar que cada módulo (autenticación, gestión de carrito, administración de productos) cumpla con los requisitos.
  + Documentación de Resultados: Registrar los resultados de cada prueba, capturando evidencia de los resultados obtenidos (capturas de pantalla, notas de observación, etc.).
  + Detección y Registro de Defectos: Identificar cualquier defecto en las funcionalidades de la aplicación y documentar los pasos para reproducirlos, asignando un nivel de prioridad.
  + Comunicación con el Equipo de Desarrollo: Informar a los desarrolladores sobre defectos críticos y colaborar en su resolución, asegurando claridad en los informes de errores y revisando correcciones aplicadas.

José Oporto – Analista de Pruebas de Integración y Rendimiento

* Responsabilidades:
  + Diseño de Pruebas de Integración: Definir y ejecutar pruebas que validen la interacción entre los distintos módulos (por ejemplo, integración entre carrito de compras y proceso de pago).
  + Pruebas de Rendimiento y Carga: Medir y analizar el rendimiento de la aplicación en condiciones de carga, verificando tiempos de respuesta y estabilidad bajo escenarios de múltiples usuarios.
  + Pruebas de Usabilidad y UX: Realizar pruebas de usabilidad, evaluando la experiencia del usuario en los flujos de navegación y compra, y recolectar feedback de posibles mejoras en la interfaz.
  + Revisión de Defectos y Re-Pruebas: Coordinar las re-pruebas necesarias para verificar que los defectos reportados se hayan solucionado sin generar problemas adicionales en el sistema.
  + Consolidación del Informe Final: Colaborar en la elaboración del informe de pruebas, resumiendo los resultados, defectos encontrados y mejoras sugeridas para la entrega final.

9.- ENTORNO Y CONFIGURACION DEL AMBIENTE DE PRUEBA

9.1. Entorno de Hardware

* Dispositivo Móvil Real: Teléfono Android (versión 10 o superior).
* Emulador de Android: Configurado en Android Studio con diferentes tamaños de pantalla para pruebas de compatibilidad.

9.2. Entorno de Software

* Sistema Operativo: Windows 10 o superior.
* Herramientas de Desarrollo:
  + Ionic CLI 7.x, Angular 18.x, Capacitor 4.x para ejecutar y depurar la aplicación.
* Base de Datos: SQLite configurada localmente para almacenar datos de usuarios, productos, carrito y compras.

9.3. Configuración del Ambiente de Prueba

* Instalación de Dependencias: Ejecutar npm install para instalar todas las dependencias.
* Sincronización de Capacitor: Ejecutar ionic cap sync para sincronizar el proyecto con el entorno nativo.
* Datos de Prueba: Configuración de datos iniciales en SQLite para simular productos y usuarios en el sistema.

9.4. Herramientas de Prueba

* Android Studio: Para ejecutar la aplicación en emulador y depurar.
* Herramientas de Captura: Para documentar resultados y defectos encontrados.

9.5. Procedimientos de Prueba

* Ejecución de la Aplicación: ionic sirve para pruebas en navegador o ionic cap run android para pruebas en emulador o dispositivo físico.
* Pruebas de Red: Conexión a internet para sincronización y pruebas en modo offline para funcionalidades locales.