

Pruebas de Integración

23 de diciembre de 2024

Índice

1. Objetivo y Alcance	3
2. Entorno de Pruebas	3
3. Estrategia de Integración	3
4. Pruebas Funcionales de Integración	4
4.1. Pantalla de Login	4
4.1.1. Funcionamiento de los Botones	4
4.1.2. Recordar Contraseña	5
4.1.3. Registrarse	5
4.1.4. Acceder al Sistema	6
4.2. Pantalla del Gestor	6
4.2.1. LOG OUT	7
4.2.2. AÑADIR PRODUCTO	7
4.2.3. ELIMINAR PRODUCTO	8
4.2.4. MOSTRAR PRODUCTOS	8
4.2.5. FIJAR DATOS	8
4.2.6. DESFIJAR DATOS	9
4.2.7. IMPORTAR PRODUCTOS	9
4.2.8. CAMBIAR CONTRASEÑA (Gestor)	9
4.3. Pantalla del Analista	10
4.3.1. LOG OUT (Analista)	10
4.3.2. MODIFICAR AFINIDADES	10
4.3.3. IMPORTAR AFINIDADES	11
4.3.4. CAMBIAR CONTRASEÑA (Analista)	11
4.4. Pantalla del Usuario Estándar	12
4.4.1. LOG OUT (Usuario Estándar)	12
4.4.2. CREAR DISTRIBUCIÓN	12
4.4.3. MOSTRAR DISTRIBUCIÓN	14
4.4.4. CAMBIAR CONTRASEÑA (Usuario Estándar)	14
4.5. Pruebas de Validación de la Lógica de Negocio, la Capa de Modelo y las Excepciones Definidas	15
5. Criterios de Aceptación y Finalización	15

1. Objetivo y Alcance

El presente documento describe el **Plan de Pruebas de Integración** para el sistema en desarrollo. El objetivo principal es:

- Validar que los distintos componentes, pantallas y funcionalidades funcionen correctamente al integrarse entre sí.
- Comprobar la interacción entre las partes del sistema, garantizando el cumplimiento de los requisitos funcionales.
- Detectar defectos o incongruencias en la comunicación o manejo de datos entre componentes.

Alcance:

- Se incluyen pruebas a nivel de integración para el módulo de autenticación, los paneles de gestión, la creación y modificación de datos, los algoritmos de optimización y todos los componentes relacionados.
- Se excluyen pruebas de rendimiento, usabilidad avanzada o pruebas de estrés que no formen parte de la verificación inicial de la integración.

2. Entorno de Pruebas

Para la ejecución de las pruebas de integración, se define el siguiente entorno:

- **Hardware:** Equipo con al menos 8 GB de RAM y procesador Intel Core i5 (o equivalente).
- **Software Base:**
 - Sistema Operativo: Windows 10 / Ubuntu 20.04 LTS.
 - Entorno de Desarrollo: openjdk 21.0.5 2024-10-15
 - Servidor de Base de Datos: SQLite.
 - Herramientas de Pruebas: JUnit.
- **Datos de Prueba:** Conjunto de usuarios, productos y configuraciones de prueba, diseñados para validar la integración.

La configuración descrita es esencial para reproducir los resultados. Cambios en las versiones de software o hardware pueden afectar los comportamientos esperados.

3. Estrategia de Integración

La estrategia de pruebas de integración se basa en un enfoque incremental, en el cual los módulos se integran gradualmente y se validan sus interacciones:

- **Comunicación entre componentes:** Se comprobarán las interacciones entre módulos (APIs, procedimientos, controladores) para garantizar la transmisión y recepción correcta de datos.
- **Manejo de errores:** El sistema debe reaccionar acorde a los requerimientos funcionales ante situaciones excepcionales o entradas inválidas.
- **Consistencia de datos:** Aunque no se realizarán pruebas de rendimiento exhaustivas, se vigilará que la integración no cause conflictos de datos o problemas de latencia significativos.

4. Pruebas Funcionales de Integración

En esta sección se describen los escenarios de prueba organizados por pantallas y funcionalidades principales. Además, se especifica la interacción entre los componentes y la forma en que cada módulo se integra con los demás.

4.1. Pantalla de Login

La pantalla de **Login** permite a los usuarios autenticarse y constituye un punto crítico de integración con la base de datos de usuarios y el servicio de notificaciones (para la recuperación de contraseña).

4.1.1. Funcionamiento de los Botones

Estado inicial:

- **Test:** Al abrir la ventana de login, se verifica que los botones no sean funcionales hasta que se cumplan las condiciones iniciales (campos obligatorios no vacíos).
- **Comportamiento esperado:** Ningún botón debe responder si los campos requeridos están vacíos.
- **Resultado:** El sistema cumple con lo esperado y no permite interactuar con los botones hasta que se ingresen datos válidos.

Botón Exit:

- **Test:** Se prueba el cierre de la aplicación con diferentes estados:
 1. Con información en los campos.
 2. Sin información en los campos.
- **Comportamiento esperado:** En ambos casos, el botón debe cerrar la aplicación sin errores.
- **Resultado:** La funcionalidad opera correctamente.

Botón Resetear Super:

- **Test:** Se realiza la prueba bajo dos condiciones:
 1. Con la base de datos vacía.
 2. Con la base de datos no vacía.
- **Comportamiento esperado:** En ambos casos, el botón debe reiniciar correctamente los valores del sistema sin generar errores.
- **Resultado:** El botón opera según lo esperado, reseteando la configuración del sistema.

4.1.2. Recordar Contraseña

Usuario existente:

- **Test:** Se solicita la recuperación de contraseña utilizando un correo válido.
- **Comportamiento esperado:** El sistema muestra en una nueva pantalla la contraseña.
- **Resultado:** Usuario recibe la contraseña.

Usuario no existente:

- **Test:** Se intenta recuperar la contraseña con un correo no registrado.
- **Comportamiento esperado:** El sistema muestra un mensaje de error indicando que el usuario no existe.
- **Resultado:** El mensaje de error aparece correctamente.

Datos incompletos:

- **Test:** Se solicita la recuperación de contraseña dejando campos vacíos o con datos no coincidentes.
- **Comportamiento esperado:** El sistema bloquea la solicitud y notifica la falta de datos.
- **Resultado:** El sistema deniega la acción y muestra un mensaje claro al usuario.

4.1.3. Registrarse

Nuevo registro:

- **Test 1:** Se ingresan datos de usuario y correo inexistentes para un nuevo registro.
- **Comportamiento esperado:** El sistema registra al usuario correctamente y guarda los datos en la base de datos.
- **Resultado:** El registro se realiza exitosamente.

Registro duplicado:

- **Test 2:** Se intentan registrar un usuario o correo ya existentes.
- **Comportamiento esperado:** El sistema bloquea la operación y muestra un mensaje indicando que el usuario ya existe.
- **Resultado:** El sistema deniega correctamente el registro duplicado.

Usuario existente con rol diferente:

- **Test 3:** Se intenta registrar un usuario existente, pero con un rol diferente.
- **Comportamiento esperado:** El sistema detecta que el usuario ya está registrado y bloquea la acción.
- **Resultado:** El registro es bloqueado según lo esperado.

4.1.4. Acceder al Sistema

Usuario válido:

- **Test:** Se inicia sesión con un usuario registrado y contraseña correcta.
- **Comportamiento esperado:** El sistema permite el acceso y redirige al panel correspondiente al rol del usuario.
- **Resultado:** El acceso se realiza correctamente.

Contraseña errónea:

- **Test:** Se intenta acceder con una contraseña incorrecta.
- **Comportamiento esperado:** El sistema deniega el acceso y muestra un mensaje de error.
- **Resultado:** Se bloquea el acceso e informa al usuario.

Campos vacíos:

- **Test:** Se intenta acceder sin completar los campos obligatorios.
- **Comportamiento esperado:** El sistema bloquea el inicio de sesión y solicita llenar los campos.
- **Resultado:** El acceso es denegado y el mensaje de advertencia aparece correctamente.

4.2. Pantalla del Gestor

La **Pantalla del Gestor** permite la administración de productos y configuraciones del sistema. Implica la integración con la base de datos para operaciones *CRUD* (Crear, Leer, Actualizar, Eliminar) y validaciones de seguridad para restringir ciertas funcionalidades.

4.2.1. LOG OUT

- **Test:**

1. Se selecciona el botón de salida: aparece una ventana emergente preguntando si se desea salir.
2. Opción **No**:
 - **Comportamiento esperado:** Regresa a la pantalla del gestor sin cerrar la sesión.
 - **Resultado:** Se cumple el comportamiento esperado.
3. Opción **Sí**:
 - **Comportamiento esperado:** Cierra la sesión y redirige a la pantalla de login.
 - **Resultado:** La funcionalidad opera correctamente.
4. Se cierra la ventana emergente sin responder:
 - **Comportamiento esperado:** Mantener la sesión activa y la pantalla del gestor abierta.
 - **Resultado:** El sistema mantiene el estado esperado.

4.2.2. AÑADIR PRODUCTO

- **Test 1:** Se llenan todos los campos con datos válidos y se pulsa el botón de agregar.
 - **Comportamiento esperado:** El producto se registra en la base de datos sin errores.
 - **Resultado:** El producto se guarda correctamente.
- **Test 2:** Se intenta agregar un producto con datos incompletos.
 - **Comportamiento esperado:** El sistema muestra un mensaje de error indicando los campos faltantes.
 - **Resultado:** El sistema bloquea la operación y muestra el error esperado.
- **Test 3:** Se ingresan datos inválidos en campos (por ejemplo, texto en un campo numérico).
 - **Comportamiento esperado:** El sistema muestra un mensaje de error especificando el tipo de dato requerido.
 - **Resultado:** La validación funciona como se espera.
- **Test 4:** Se añade un producto ya existente.
 - **Comportamiento esperado:** El sistema alerta que el producto ya está registrado.
 - **Resultado:** La funcionalidad opera correctamente.
- **Test 5:** Uso del botón **Limpiar Campos** tras rellenar el formulario.
 - **Comportamiento esperado:** Todos los campos vuelven a estar vacíos.
 - **Resultado:** El formulario se limpia correctamente.

4.2.3. ELIMINAR PRODUCTO

- **Test 1:** Se introduce un ID inexistente y se intenta eliminar el producto.
 - **Comportamiento esperado:** El sistema muestra un mensaje de error indicando que el ID no se encuentra.
 - **Resultado:** El comportamiento es correcto.
- **Test 2:** Se introduce un ID existente y se pulsa el botón de eliminar.
 - **Comportamiento esperado:** El producto desaparece de la base de datos.
 - **Resultado:** El producto es eliminado correctamente.

4.2.4. MOSTRAR PRODUCTOS

- **Test 1:** Se prueba con la base de datos vacía.
 - **Comportamiento esperado:** No se muestra ningún producto en la lista.
 - **Resultado:** La pantalla refleja el estado vacío correctamente.
- **Test 2:** Se eliminan productos y se actualiza la lista.
 - **Comportamiento esperado:** La lista no debe mostrar los productos eliminados.
 - **Resultado:** La lista responde adecuadamente.
- **Test 3:** Se crean productos con nombres y descripciones largas.
 - **Comportamiento esperado:** Los productos se visualizan de manera legible sin romper el diseño.
 - **Resultado:** La interfaz maneja adecuadamente los textos extensos.

4.2.5. FIJAR DATOS

- **Test 1:** Se fija la base de datos cuando ya está fijada.
 - **Comportamiento esperado:** Aparece un mensaje indicando que los datos ya están fijados.
 - **Resultado:** El sistema bloquea la operación como se espera.
- **Test 2:** Se intenta añadir productos con los datos fijados.
 - **Comportamiento esperado:** El sistema deniega la acción e informa al usuario.
 - **Resultado:** La validación funciona correctamente.
- **Test 3:** Se fija la base de datos, se cierra sesión y se regresa al programa.
 - **Comportamiento esperado:** Los datos permanecen fijados tras el reinicio.
 - **Resultado:** La integridad de la configuración se mantiene.

4.2.6. DESFIJAR DATOS

- **Test 1:** Se intenta desfijar datos sin que hayan sido fijados previamente.
 - **Comportamiento esperado:** Aparece un mensaje de error.
 - **Resultado:** El mensaje se muestra correctamente.
- **Test 2:** Tras fijar datos, se desfijan y se realizan operaciones de agregar o eliminar productos.
 - **Comportamiento esperado:** El sistema permite las operaciones sin errores.
 - **Resultado:** El comportamiento cumple con lo esperado.

4.2.7. IMPORTAR PRODUCTOS

- **Test 1:** Se intenta importar un archivo con los datos fijados.
 - **Comportamiento esperado:** El sistema bloquea la acción y muestra una advertencia.
 - **Resultado:** La respuesta es correcta, manteniendo la coherencia de la base de datos.
- **Test 2:** Se importa un archivo válido con datos desfijados y la base de datos ya contiene productos.
 - **Comportamiento esperado:** Los productos nuevos se añaden sin eliminar los existentes.
 - **Resultado:** Los datos se combinan correctamente.
- **Test 3:** Se importa un archivo en una base de datos vacía.
 - **Comportamiento esperado:** Los productos del archivo se añaden sin problemas.
 - **Resultado:** La funcionalidad opera correctamente.

4.2.8. CAMBIAR CONTRASEÑA (Gestor)

- **Test 1:** Se introduce una contraseña actual incorrecta.
 - **Comportamiento esperado:** El sistema deniega el cambio y muestra un mensaje de error.
 - **Resultado:** La validación funciona según lo esperado.
- **Test 2:** Se introducen contraseñas nuevas que no coinciden.
 - **Comportamiento esperado:** El sistema muestra un mensaje indicando la falta de coincidencia.
 - **Resultado:** La operación es bloqueada correctamente.
- **Test 3:** Se cambia la contraseña con datos correctos.
 - **Comportamiento esperado:** La contraseña se actualiza exitosamente.
 - **Resultado:** El cambio se realiza sin errores.

4.3. Pantalla del Analista

Esta pantalla está diseñada para que los analistas gestionen afinidades entre elementos y realicen configuraciones específicas del sistema. A continuación, se detallan las pruebas realizadas para cada funcionalidad disponible:

4.3.1. LOG OUT (Analista)

■ **Test:**

1. Se selecciona el botón y aparece una ventana emergente preguntando si se desea salir.
2. Opción **No**:
 - **Comportamiento esperado:** El sistema regresa a la pantalla del analista sin cerrar la sesión.
 - **Resultado:** El sistema opera correctamente.
3. Opción **Sí**:
 - **Comportamiento esperado:** El sistema cierra la sesión y redirige a la pantalla de login.
 - **Resultado:** La funcionalidad responde adecuadamente.
4. Se cierra la ventana emergente sin seleccionar ninguna opción:
 - **Comportamiento esperado:** La sesión permanece activa y el sistema regresa a la pantalla del analista.
 - **Resultado:** El comportamiento es el esperado.

4.3.2. MODIFICAR AFINIDADES

- **Test 1:** Se intenta modificar afinidades cuando los datos están fijados.
- **Comportamiento esperado:** El sistema bloquea la acción y muestra un mensaje indicando que los datos están fijados.
 - **Resultado:** La funcionalidad opera correctamente.
- **Test 2:** Se modifica una afinidad existente con datos válidos.
- **Comportamiento esperado:** El cambio se realiza con éxito y los datos se actualizan en la base de datos.
 - **Resultado:** La modificación es exitosa.
- **Test 3:** Se intentan modificar elementos de la diagonal (afinidad consigo mismo).
- **Comportamiento esperado:** El sistema no permite la modificación e informa al usuario.
 - **Resultado:** El sistema bloquea correctamente la acción.
- **Test 4:** Se prueba la funcionalidad con afinidades inexistentes (nuevas entradas).

- **Comportamiento esperado:** El sistema permite añadir afinidades con datos válidos.
- **Resultado:** La operación se completa correctamente.
- **Test 5:** Se realiza un cambio, se cierra la sesión y se vuelve a ingresar al sistema.
 - **Comportamiento esperado:** Los cambios realizados permanecen guardados.
 - **Resultado:** Los datos se mantienen consistentes.

4.3.3. IMPORTAR AFINIDADES

- **Test 1:** Se intenta importar afinidades con los datos fijados.
 - **Comportamiento esperado:** El sistema bloquea la acción y muestra un mensaje de error.
 - **Resultado:** La funcionalidad responde correctamente.
- **Test 2:** Se importa un archivo válido en un sistema con afinidades ya establecidas.
 - **Comportamiento esperado:** Los nuevos datos se combinan con los existentes sin sobrescribirlos.
 - **Resultado:** La operación se realiza correctamente.
- **Test 3:** Se prueba la importación en un sistema sin afinidades previas.
 - **Comportamiento esperado:** Las afinidades del archivo se añaden sin problemas.
 - **Resultado:** El sistema carga los datos correctamente.
- **Test 4:** Se importa un archivo con errores en el formato (datos incompletos o inválidos).
 - **Comportamiento esperado:** El sistema muestra un mensaje de error indicando el problema.
 - **Resultado:** La validación funciona como se espera.

4.3.4. CAMBIAR CONTRASEÑA (Analista)

- **Test 1:** Se introduce una contraseña actual incorrecta.
 - **Comportamiento esperado:** El sistema deniega el cambio y muestra un mensaje de error claro.
 - **Resultado:** La funcionalidad opera correctamente.
- **Test 2:** Se ingresan contraseñas nuevas que no coinciden.
 - **Comportamiento esperado:** El sistema bloquea la acción y muestra un mensaje de falta de coincidencia.
 - **Resultado:** La validación funciona adecuadamente.

- **Test 3:** Se realiza el cambio de contraseña con datos correctos y coincidentes.
 - **Comportamiento esperado:** La contraseña se actualiza correctamente y el usuario recibe confirmación.
 - **Resultado:** La operación es exitosa.
- **Test 4:** Se prueba ingresar al sistema con la nueva contraseña tras el cambio.
 - **Comportamiento esperado:** El acceso debe ser exitoso.
 - **Resultado:** El sistema permite el inicio de sesión sin errores.
- **Test 5:** Se intenta ingresar con la contraseña antigua tras el cambio.
 - **Comportamiento esperado:** El sistema debe denegar el acceso.
 - **Resultado:** El comportamiento es el esperado.

4.4. Pantalla del Usuario Estándar

Esta pantalla permite a los usuarios estándar realizar operaciones relacionadas con la creación, visualización y modificación de distribuciones de productos.

4.4.1. LOG OUT (Usuario Estándar)

Test: El botón *LOG OUT* y su implementación son idénticos a los de otras pantallas, por lo que se aplican las mismas pruebas documentadas anteriormente. No se describen de nuevo para evitar redundancia.

4.4.2. CREAR DISTRIBUCIÓN

Permite a los usuarios crear una nueva distribución de productos basándose en configuraciones y algoritmos de optimización.

Pantalla de selección de productos:

1. Se selecciona el botón **Eliminar Producto** con el cajón vacío.
 - **Comportamiento esperado:** El sistema bloquea la acción y muestra un mensaje indicando que no hay productos para eliminar.
 - **Resultado:** La funcionalidad opera correctamente.
2. Se selecciona un producto y se elimina.
 - **Comportamiento esperado:** El producto se elimina del cajón y no aparece en la lista.
 - **Resultado:** El producto es eliminado correctamente.
3. Se pulsa el botón **Aceptar** sin haber añadido productos.
 - **Comportamiento esperado:** El sistema muestra un mensaje de error indicando que no hay productos seleccionados.

- **Resultado:** El mensaje de error aparece correctamente.
4. Se pulsa **Aceptar** con productos añadidos, pero sin realizar cálculos.
 - **Comportamiento esperado:** El sistema bloquea la acción y solicita realizar un cálculo antes de continuar.
 - **Resultado:** La funcionalidad cumple con lo esperado.
 5. Se cierra la pestaña emergente de forma abrupta.
 - **Comportamiento esperado:** Los datos del cajón permanecen guardados hasta reabrir la pestaña.
 - **Resultado:** El sistema mantiene los datos sin pérdida de información.
 6. Se prueba **Calcular Distribución** sin productos en el cajón.
 - **Comportamiento esperado:** El sistema bloquea la acción y muestra un mensaje indicando que no hay productos para calcular.
 - **Resultado:** El sistema responde correctamente.
 7. Se calcula una distribución con productos válidos.
 - **Comportamiento esperado:** El sistema realiza el cálculo sin errores y muestra una configuración válida.
 - **Resultado:** El cálculo se realiza correctamente y la configuración es válida.

Validación de número de baldas:

8. Se ingresa un número de baldas que no cumple los requisitos.
 - **Comportamiento esperado:** El sistema bloquea la acción e informa que el número de baldas es inválido.
 - **Resultado:** El sistema valida correctamente los datos.
9. Se ingresa un número de baldas válido.
 - **Comportamiento esperado:** El sistema acepta la configuración y permite continuar.
 - **Resultado:** Los datos son aceptados correctamente.
10. Se prueba la configuración generada con diferentes algoritmos de optimización:
 - **Obtener Distribución Lateral:** Algoritmos: *Bruteforce*, *Held-Karp*, *MST 2-Approximation*, *Christofides*.
 - **Comportamiento esperado:** Cada algoritmo debe generar una distribución válida según su lógica.
 - **Resultado:** Todas las distribuciones se generan correctamente.
 - **Obtener Distribución Vecinos:** Algoritmos: *Bruteforce*, *Hill Climbing*, *Simulated Annealing*.

- **Comportamiento esperado:** Las distribuciones generadas son coherentes con los datos ingresados.
- **Resultado:** La funcionalidad cumple con lo esperado.

11. Se prueban los botones **Atrás** y **Aceptar** tras realizar los cálculos.

- **Comportamiento esperado:** Ambos botones operan correctamente, permitiendo volver a la pantalla previa o aceptar la configuración generada.
- **Resultado:** El comportamiento es el esperado.

4.4.3. MOSTRAR DISTRIBUCIÓN

Permite a los usuarios visualizar y modificar distribuciones previamente calculadas.

- **Test 1:** Se selecciona una distribución del panel izquierdo tras realizar cálculos previos.
 - **Comportamiento esperado:** La distribución seleccionada se muestra correctamente en el panel principal.
 - **Resultado:** La funcionalidad opera según lo esperado.
- **Test 2:** Se prueban distribuciones de diferentes tamaños.
 - **Comportamiento esperado:** El sistema debe adaptarse y mostrar correctamente distribuciones grandes y pequeñas.
 - **Resultado:** Todas las distribuciones se representan de forma ordenada.
- **Test 3:** Se modifica una distribución haciendo **swap** entre dos productos.
 - **Comportamiento esperado:** La modificación se refleja en la distribución y se guarda correctamente en la base de datos.
 - **Resultado:** El sistema actualiza y guarda la distribución sin errores.
- **Test 4:** Se selecciona una distribución modificada y se vuelve a mostrar.
 - **Comportamiento esperado:** La distribución reflejará los cambios realizados y se visualizará correctamente.
 - **Resultado:** La funcionalidad cumple con lo esperado.

4.4.4. CAMBIAR CONTRASEÑA (Usuario Estándar)

Test: El botón y su implementación son análogos a los de otras pantallas, por lo que las mismas pruebas documentadas se aplican aquí.

4.5. Pruebas de Validación de la Lógica de Negocio, la Capa de Modelo y las Excepciones Definidas

En esta sección se introducen las pruebas orientadas a validar la interacción entre diferentes pantallas y funcionalidades del sistema, con un enfoque específico en los aspectos críticos de la lógica de negocio, la capa de modelo y las excepciones definidas.

El objetivo principal de estas pruebas es garantizar la coherencia y robustez del sistema al combinar distintas acciones. Además, se busca asegurar que cada pantalla valide correctamente los estados de la base de datos, incluyendo la detección y el manejo apropiado de errores, así como el cumplimiento estricto de las reglas de negocio.

Estas validaciones se han implementado utilizando **JUnit** y el **driverIntegracion**, una herramienta diseñada específicamente para probar casos críticos y verificar excepciones que no pueden ser evaluadas completamente mediante las pruebas funcionales de integración descritas anteriormente. El código correspondiente a estas pruebas se encuentra documentado y estructurado en el proyecto.

El uso del **driverIntegracion** permite abordar:

- Condiciones límite y escenarios extremos.
- Errores deliberados para evaluar el manejo de excepciones.
- Casos críticos en los que la lógica de negocio puede entrar en conflicto con los datos o las reglas definidas.

Gracias a este enfoque, se asegura un comportamiento consistente y predecible en la capa de modelo y la lógica de negocio, fortaleciendo la estabilidad y fiabilidad general del sistema.

5. Criterios de Aceptación y Finalización

Criterios de Aceptación:

- Se debe completar el 100% de las pruebas de integración sin defectos críticos o bloqueantes.
- Los defectos de severidad media o baja deben ser evaluados y/o corregidos antes de la entrega a la siguiente fase (pruebas de sistema o aceptación).

Criterios de Finalización:

- Todos los módulos y pantallas mencionados han sido verificados con resultados aceptables. Algunos errores identificados durante las pruebas realizadas con el **driverIntegración** se atribuyen a problemas relacionados con la latencia en la actualización del estado de la base de datos entre cada test. Tras realizar un análisis detallado, se confirmó que las funciones individuales funcionan correctamente al ejecutarse por separado, lo que indica que los errores no están en la lógica del sistema, sino en la configuración del entorno de pruebas.
- Se han documentado todos los hallazgos en el sistema de seguimiento de defectos para su evaluación y resolución.

6. Conclusiones

El presente *Plan de Pruebas de Integración* cubre los escenarios más relevantes para validar la interacción entre los módulos del sistema:

- La **Pantalla de Login** verifica la comunicación con el módulo de autenticación y la base de datos de usuarios.
- La **Pantalla del Gestor** confirma la adecuada integración con el módulo de productos y la configuración del sistema, cubriendo aspectos *CRUD*, restricciones de seguridad y fijación/desfijación de datos.
- La **Pantalla del Analista** permite comprobar la gestión de afinidades y la importación de datos, validando las reglas aplicables a los analistas.
- La **Pantalla del Usuario Estándar** permite crear y modificar distribuciones de productos, probando los algoritmos y la interacción con la base de datos.

En caso de detectar errores o comportamientos inesperados, se registrarán de acuerdo con el *Plan de Gestión de Incidencias*. Una vez superados los criterios de aceptación, el proyecto podrá avanzar a las fases de prueba de sistema y de aceptación final.