**INFORME DE PRUEBAS**

**Versión 1.0**

**Lima, abril 2023**

**Historial de revisiones**

| **Fecha** | **Revisión** | **Descripción** | **Autor** |
| --- | --- | --- | --- |
| 29/04/2023 | 1.0 | Primera versión del informe de pruebas | Consultora HomeSkill S.A. |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**ÍNDICE**

[**1. Introducción 1**](#_909ss9r16or5)

[**1.1. Propósito del documento 1**](#_c9wlc53vrw4l)

[**1.2. Alcance del informe 1**](#_rhjn2xmifd6a)

[**2. Plan de pruebas 1**](#_rfswte8u3bjm)

[**2.1. Tipos de prueba 1**](#_xf2qh5hqxrs8)

[**2.2. Recursos necesarios 4**](#_1d6dake06ddv)

[**2.3. Estrategias de pruebas 5**](#_jhynj0u3v07i)

[**3. Casos de prueba 6**](#_tyyejnnpem8o)

[**3.1. Escenario de prueba 6**](#_nsl89im0v9fu)

[**3.1.1. Prueba del requerimiento 01 6**](#_mky7j1ntvixi)

[**3.1.2. Prueba del requerimiento 02 7**](#_qcq5d4baunuh)

[**3.1.3. Prueba del requerimiento 03 8**](#_90csfnjylgcx)

[**4. Resultados de prueba 8**](#_9zufgvdgzoc1)

[**5. Conclusiones 9**](#_b1pcvj1zx7gq)

# **1. Introducción**

## **1.1. Propósito del documento**

El propósito principal es proporcionar una descripción detallada de cómo se probó el software, qué casos de prueba se ejecutaron, qué resultados se obtuvieron, qué incidencias se encontraron, y cómo se corrigieron, las áreas que necesitan mejorar y las recomendaciones para futuras pruebas.

## 

## **1.2. Alcance del informe**

El informe de pruebas proporciona una evaluación completa y detallada del software del proyecto GoShop en términos de su calidad y capacidad para satisfacer los requisitos del usuario. El informe incluye información sobre los requisitos funcionales que se han evaluado, y cualquier problema que se haya encontrado durante las pruebas y cómo se han abordado.

# **2. Plan de pruebas**

## **2.1. Tipos de prueba**

**Tabla 1**

*Prueba de Requerimiento 1*

| **Vista de registro de usuario** | |
| --- | --- |
| **Resumen:** Este caso se utiliza para probar el comportamiento de la vista de registro de usuario en una aplicación Django. | |
| **Pasos:**   1. Configurar objetos necesarios para la realización de las pruebas, método setup 2. Primer método de prueba(test\_signup\_view\_with\_get\_request). Solicitud GET 3. Segundo método de prueba(test\_signup\_view\_with\_valid\_post\_request). Solicitud POST 4. Enviar la solicitud POST a la URL de registro de usuario con los datos   (‘self.signup\_url’ y ‘data’   1. Verificar la respuesta HTTP de redirección. Método ‘assertEqual’ 2. Verificar la respuesta redirige a la página deseada. Método ‘assertRedirects’ 3. Verificar la creación del usuario en la base de datos. Método ‘assertTrue’ | |
| **Resultados Esperados:**  **Método ‘test\_signup\_view\_with\_get\_request’**   * Esperamos que la vista de registro de usuario esté disponible y responda con un código de estado HTTP 200 (OK). * Esperamos que la plantilla ‘register.html’ se utilice para mostrar la vista de registro.   **Método ‘test\_signup\_view\_with\_valid\_post\_request’**   * Esperamos que se cree un nuevo usuario en la base de datos con éxito. * Esperamos que el usuario sea redirigido a la página "shop" después de un registro válido. | |

*Nota*. Elaboración Propia.

**Tabla 2**

*Prueba de Requerimiento 2*

| **Vista de búsqueda y visualización de productos** | |
| --- | --- |
| **Resumen:** El caso de prueba usa el cliente de prueba de Django para simular una solicitud a la vista y luego verifica si la respuesta recibida contiene los productos esperados y si el estado de respuesta es 200 (OK). | |
| **Pasos:**   1. Crear una nueva categoría de producto con el nombre "Kitchen". 2. Crear un nuevo producto en la categoría "Kitchen" con el nombre "Kitchen Products", un precio de 10.00 y un stock de 5. 3. Acceder a la vista "products" usando el parámetro de texto "Kitchen". 4. Verificar que se recibe una respuesta HTTP 200 (OK) 5. Verificar que la respuesta de la vista "products" contiene el nombre "Kitchen Products". 6. Crear una nueva categoría de producto con el nombre "test\_category". 7. Crear dos nuevos productos en la categoría "test\_category" con los nombres "Product 1" y "Product 2", con descripciones y precios específicos. 8. Acceder a la vista "products" usando el parámetro de texto "Product". 9. Verificar que se recibe una respuesta HTTP 200 (OK) 10. Verificar que la respuesta de la vista "products" contiene los nombres "Product 1" y "Product 2". | |
| **Resultados Esperados:**  **Método ‘test\_products\_view’**   * Se pueda acceder correctamente a la vista "products" usando el parámetro de texto   **Método ‘test\_products\_search’**   * Se puedan crear nuevos productos y categorías, y que se pueda acceder correctamente a la vista "products" usando el parámetro de texto "Product", y que la respuesta de la vista contenga los nombres de los productos creados en los pasos previos. | |

*Nota*. Elaboración Propia.

**Tabla 3**

*Prueba de Requerimiento 3*

| **Vista de descripción de productos** | |
| --- | --- |
| **Resumen:** Se centra en probar la vista que muestra la descripción de un producto en la tienda en línea. Se utiliza el módulo de pruebas de Django TestCase y el objeto Client para hacer solicitudes HTTP simuladas y realizar comprobaciones en la respuesta recibida. | |
| **Pasos:**   1. Importar los módulos necesarios para el caso de prueba. 2. Configurar el objeto Client y crear instancias de ProductCategory y Product para utilizar en las pruebas. 3. Crear un método de prueba para verificar que la vista de descripción de productos funciona correctamente con un ID de producto válido:    1. Hacer una solicitud GET a la vista de descripción de productos con el ID del producto creado anteriormente.    2. Verificar que la respuesta tenga un código de estado 200.    3. Verificar que se utiliza la plantilla correcta ('product\_description.html').    4. Verificar que el contexto de la respuesta incluye el producto creado 4. Crear un segundo método de prueba para verificar que la vista de descripción de productos funciona correctamente con un ID de producto incorrecto    1. Hacer una solicitud GET a la vista de descripción de productos con un ID de producto incorrecto (100 en este caso).    2. Verificar que la respuesta tenga un código de estado 200.    3. Verificar que se utiliza la plantilla correcta ('product\_description.html').    4. Verificar que el contexto de la respuesta no incluye ningún producto (ya que el ID es incorrecto), pero incluye la vista anterior ('Cocina') y la vista actual ('kitchen'). | |
| **Resultados Esperados:**  **Método ‘test\_product\_description\_get’**   * Se espera que la solicitud GET a la vista de descripción de productos con el ID del producto creado anteriormente retorne un código de estado 200 (éxito) * Se espera que la plantilla utilizada sea ‘product\_description.html’   **Método ‘test\_product\_description\_get\_wrong\_product\_id’**   * Se espera que la solicitud GET a la vista de descripción de productos con un ID de producto incorrecto (100 en este caso) retorne un código de estado 200 (éxito). * Se espera que la plantilla utilizada sea ‘product\_description.html’ | |

*Nota*. Elaboración Propia.

## 

## **2.2. Recursos necesarios**

* **Hardware necesario**

Se necesitará un equipo con suficiente capacidad de procesamiento y memoria RAM para ejecutar el servidor de desarrollo de Django, el navegador web y cualquier otra aplicación necesaria para realizar las pruebas.

* **Software necesario**

Se necesitará lo siguiente:

* Python 3.8 y las dependencias necesarias para ejecutar el proyecto de Django.
* Un entorno virtual para el proyecto de Django (Visual Studio).
* Un servidor de desarrollo de Django para ejecutar el proyecto y las pruebas.
* Un navegador web para realizar las pruebas.
* **Base de datos**

Se necesitará una base de datos para el proyecto de Django. Si se usa SQLite para el proyecto, no se necesitará configurar nada adicional, pero si se usa otro motor de base de datos, se necesitará configurarlo y asegurarse de que esté en funcionamiento

.

* **Recursos adicionales**: Para las pruebas específicas del proyecto de Django mencionadas anteriormente, se necesitará lo siguiente:
* Vista de descripción de productos: se necesitará al menos un objeto de categoría de producto y un objeto de producto para realizar las pruebas. Se pueden crear estos objetos en la función setUp del TestCase.
* Vista de registro de usuario: no se necesitan recursos adicionales, ya que se puede utilizar la función de registro de Django para crear nuevos usuarios.
* Vista de búsqueda y visualización de productos: se necesitará al menos un objeto de categoría de producto y varios objetos de producto para realizar las pruebas. Se pueden crear estos objetos en la función setUp del TestCase.

## **2.3. Estrategias de pruebas**

Las estratégias de prueba que hemos usado en estos tres test, cuáles son el registro de usuario, visualización de productos y descripción de productos, son de tipo unitaria y de integración, las prueba unitarias son pruebas a nivel código para verificar su comportamiento y validarlo, mientras que las de integración son para evaluar la interacción entre diferentes requisitos y cómo funcionan en conjunto.

* Estrategia de Prueba del requerimiento 01

Para la Vista de registro de usuario, se utilizaron pruebas de unidad y de integración para asegurarse de que la funcionalidad de registro de usuario funcionara correctamente. Se comprobó que los campos de registro fueran correctos, que la información del usuario se almacena correctamente en la base de datos y que se redirigiera al usuario correctamente después del registro. Todas las pruebas fueron exitosas y no se encontraron problemas en la aplicación.

* Estrategia de Prueba del requerimiento 02

Para la vista de búsqueda y visualización de productos, se utilizaron pruebas funcionales y de integración para asegurarse de que los productos se mostrarán correctamente en la página de búsqueda, que la funcionalidad de búsqueda funcionara correctamente y que la información del producto fuera correcta en la página de detalle del producto. También se probó la funcionalidad de paginación y de ordenamiento de los productos. Todas las pruebas fueron exitosas y no se encontraron problemas en la aplicación.

* Estrategia de Prueba del requerimiento 03

Para la descripción de productos, se utilizaron pruebas funcionales y de regresión para asegurarse de que la página de descripción del producto funcionara correctamente para diferentes productos y en diferentes situaciones. Se comprobó que el producto correcto se mostrará en la página de descripción, que la plantilla y la información del producto fueran correctas y que se mostrarán las vistas de la categoría anterior y actual. Todas las pruebas fueron exitosas y no se encontraron problemas en la aplicación.

En general, se recomienda continuar con las pruebas funcionales y de integración en futuras pruebas para garantizar la calidad y la funcionalidad de la aplicación.

# **3. Casos de prueba**

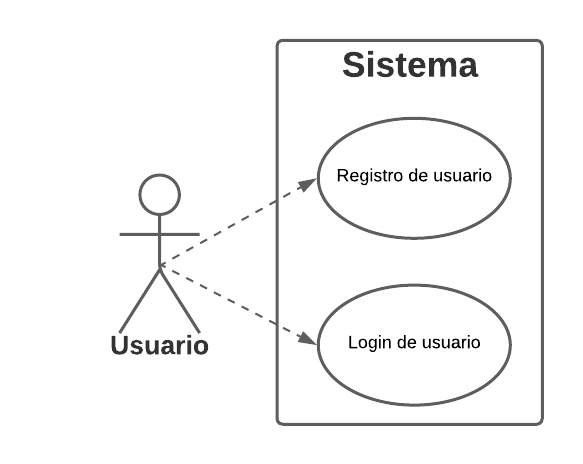
## **3.1. Escenario de prueba**

### ***3.1.1. Prueba del requerimiento 01***

El propósito del caso de prueba es verificar el correcto funcionamiento de la vista de registro de usuario en una aplicación web, cuando se reciben solicitudes GET y POST, se redirecciona al usuario a la página correcta y se crea el usuario en la base de datos sin problemas. Para lograrlo, se configuran los elementos necesarios, se realizan solicitudes GET y POST, se comprueba la respuesta HTTP de la vista, se verifica que la redirección se efectúe adecuadamente y se confirma que el usuario se haya creado sin errores en la base de datos mediante la ejecución de diversas afirmaciones y validaciones.

**Figura 1**

*Diagrama de Interacción 01*

****

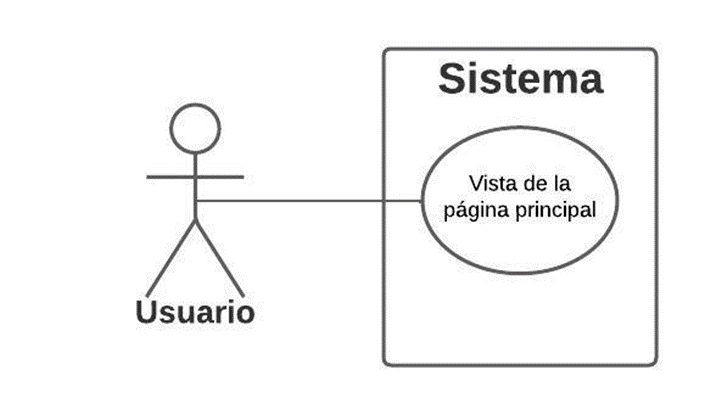
*Nota.* Elaboración propia

### ***3.1.2. Prueba del requerimiento 02***

El propósito de las pruebas fue verificar el correcto funcionamiento de la vista de búsqueda y visualización de productos en una aplicación, para lo cual se emplearon pruebas funcionales e integración. El objetivo era asegurar que los productos se mostrarán de manera adecuada en la página de búsqueda, que la funcionalidad de búsqueda funcionara correctamente y que la información del producto fuera correcta en la página de detalle del producto. Asimismo, se evaluó la funcionalidad de paginación y de ordenamiento de los productos. Tras realizar todas las pruebas, se confirmó que la aplicación funcionaba sin problemas, ya que no se encontraron errores o problemas durante las pruebas.

**Figura 2**

*Diagrama de interacción 02*

******

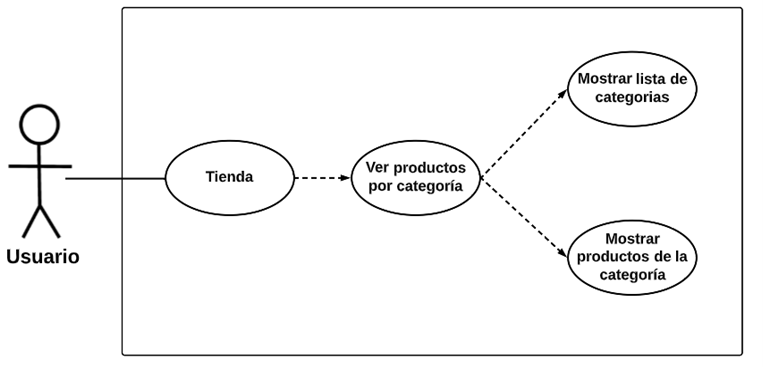
*Nota.* Elaboración propia

### ***3.1.3. Prueba del requerimiento 03***

El propósito de las pruebas fue verificar el correcto funcionamiento de la página de descripción de productos en una aplicación, mediante la realización de pruebas funcionales y de regresión. El objetivo era asegurarse de que la página de descripción de cada producto se mostrará de manera adecuada, tanto para diferentes productos como en diferentes situaciones. Se evaluó que el producto correcto se mostrará en la página de descripción, que la plantilla y la información del producto fueran correctas, y que se visualizarán las vistas de la categoría anterior y actual de manera adecuada. Tras realizar todas las pruebas, se confirmó que la aplicación funcionaba sin problemas, ya que no se encontraron errores o problemas durante las pruebas.

**Figura 3**

*Diagrama de interacción 03*

****

*Nota.* Elaboración propia

# **4. Resultados de prueba**

Se efectuaron 3 pruebas utilizando el comando "python3 manage.py test -v 1", en donde con ayuda del TestCase de python se identificaron las secciones a testear, en donde cada archivo de test contenía 2 funcionalidades a revisar

Los resultados obtenidos indican que todas las pruebas fueron exitosas, lo que sugiere que las funcionalidades Vista de registro de usuario, Vista de búsqueda y visualización de productos, y Vista de descripción de productos funcionan correctamente. Se creó una base de datos de prueba para ejecutar las pruebas, y no se identificaron problemas en el sistema durante la ejecución de las mismas.

Es importante tener en cuenta que cada prueba incluyó dos métodos diferentes para verificar si la funcionalidad funciona correctamente, lo que asegura una mayor precisión en la validación de los resultados. El tiempo total de ejecución de las pruebas fue de 0.583s.

# **5. Conclusiones**

* De las tres pruebas presentadas, se puede concluir que las pruebas son una parte fundamental del proceso de desarrollo de software, ya que permiten verificar que el software se comporte como se espera y que funcione correctamente. En los tres casos, se utilizaron diferentes tipos de pruebas para verificar el correcto funcionamiento de distintas partes de la aplicación, como la vista de registro de usuario, la vista de búsqueda y visualización de productos, y la página de descripción de productos.
* Además, se puede observar que en todos los casos se realizó una evaluación exhaustiva de las diferentes funcionalidades, a través de la configuración de elementos, la realización de solicitudes GET y POST, la comprobación de respuestas HTTP, la verificación de redirecciones y la ejecución de diversas afirmaciones y validaciones. En todos los casos, se confirmó que la aplicación funcionaba sin problemas y que no se encontraron errores o problemas durante las pruebas.
* En resumen, las pruebas son una parte fundamental del proceso de desarrollo de software, ya que permiten verificar el correcto funcionamiento de la aplicación y garantizar que se cumplan los requisitos y expectativas del usuario. La realización de pruebas adecuadas puede ayudar a detectar errores y fallos en el software antes de que sean detectados por los usuarios, lo que puede ahorrar tiempo y costos en la solución de problemas en una etapa posterior.