

# Tarea 1: Gestión de Inventario para Pequeñas Empresas

INF 331- Pruebas de software

Integrantes:

- Tamara León 201973015-0
- Valentina Lepin 201904113-4

Repositorio GitHub: [https://github.com/vlepin/INF331/tree/Tarea\\_1](https://github.com/vlepin/INF331/tree/Tarea_1)

## Descripción del problema

Un emprendedor ha solicitado una aplicación sencilla para gestionar el inventario de productos en la bodega de su negocio. Es difícil llevar un control manual de los productos disponibles, sus cantidades y su estado, por lo que la solución propuesta debe ser práctica, fácil de usar y enfocada en optimizar la gestión de su inventario.

Para abordar esta problemática, se requiere una aplicación sencilla pero funcional que optimice la gestión del inventario. Esta herramienta debe permitir al usuario registrar productos, actualizar su disponibilidad, realizar búsquedas eficientes y generar reportes que faciliten la toma de decisiones. Además, se solicita que el acceso esté protegido mediante autenticación para garantizar la seguridad de la información.

Si bien se han definido algunos requerimientos clave, estos aún no están completamente especificados. Por lo tanto, será necesario precisar detalles adicionales para evitar ambigüedades en el desarrollo de la solución.

## Análisis y Validación del problema

Para especificar mejor un requerimiento y validarlo, es fundamental mejorar las definiciones, reducir la ambigüedad y establecer metas accesibles, medibles y útiles para el contexto. Para lograrlo, identificamos las secciones con falta de claridad y realizamos supuestos que nos permitan reducir la ambigüedad en los requerimientos.

### CRUD

No se define cuáles son todas las categorías, tampoco queda especificado si serán las categorías predefinidas o el usuario podrá crear nuevas. Para este caso, definimos que las categorías son creadas por el usuario ya que no tenemos conocimiento del inventario del negocio y consideramos más grave dejar fuera una categoría importante que poder llegar a tener variaciones entre una misma categoría con su versión diferente pero igual, esto lo mitigamos en el proceso de filtrado, el que ayuda a reducir las diferencias entre una definición y otra, por ejemplo: elementos con las categorías “ropa”, “Ropa” y “ropas” aparecen en la misma búsqueda.

### Gestión de stock

No está definido si la actualización de stock debe realizarse de forma manual o si puede integrarse con un sistema de ventas (donde varios productos editan su stock). Por lo que nos quedamos sólo con las funciones solicitadas y que se use la función de editar stock.

## Filtrado y búsqueda

No se especifica qué criterios de búsqueda estarán disponibles (por nombre, categoría, precio, stock, etc.). Por lo que nos centramos en coincidencias de nombre, rango de precios y categorías.

No se menciona si la búsqueda debe admitir filtros avanzados o sugerencias automatizadas, así que decidimos no incluirlas bajo el supuesto de que el sistema no lo requiere por su baja complejidad.

## Generación de reportes

No está claro si se requiere una opción para crear un informe o editar uno que visualice los distintos periodos, por lo que se crea un nuevo reporte cada que se solicita que contiene solo con la información actual y no de otros periodos.

## Autenticación

No se especifica si habrá diferentes roles de usuario con permisos diferenciados por ejemplo, administrador y empleado. Por lo que suponemos que el proyecto lo opera una sola persona del negocio y no otros sin credenciales, por ejemplo clientes del negocio.

No se indica si se necesita un sistema de registro o si los usuarios serán creados manualmente por un administrador. Decidimos incluir la función en caso de que ocurran problemas de credenciales o contraten a otro encargado.

No se especifica la gestión de los usuarios o cómo debería ser el método de autenticación, por lo que usamos Fernet para cifrar la contraseña con clave derivada y un registro de usuarios en un txt.

## Verificación

Para asegurar que el programa cumpla con el requerimiento, se deben realizar pruebas ejecutándolo con distintas entradas y enfoques que permitan verificar su correcto funcionamiento. Para ello, desarrollamos historias de usuario que, a partir de las interpretaciones realizadas, descomponen el problema en metas claras, factibles y alineadas con los requerimientos establecidos. Cada historia de usuario incluye criterios de aceptación que facilitan la verificación y garantizan el cumplimiento del requerimiento.

## Historias de Usuario

### [HU01] Registro de productos en el inventario

Como emprendedor

Quiero registrar nuevos productos en mi inventario

Para tener un control de los artículos disponibles en la bodega

Criterios de aceptación:

- Dado un emprendedor que quiere registrar un producto, cuando ingrese el nombre, descripción, categoría, cantidad y precio del producto, entonces el sistema debe registrar el producto correctamente.
- Dado un emprendedor que quiere registrar un producto con una categoría inexistente, cuando intente agregar una nueva categoría, entonces el sistema debe permitir la creación de la categoría.
- Dado un emprendedor que quiere registrar un producto, cuando deje algún campo vacío, entonces el sistema debe impedir el registro y mostrar un mensaje de error.
- Dado un emprendedor que ha registrado un producto correctamente, cuando finalice el registro, entonces el sistema debe mostrar un mensaje de confirmación.

## [HU02] Actualización del stock de productos

Como emprendedor

Quiero modificar la cantidad de productos en el inventario

Para reflejar cambios en el stock disponible

Criterios de aceptación:

- Dado un emprendedor que necesita modificar el stock de un producto, cuando ingrese una nueva cantidad, entonces el sistema debe actualizar el stock correctamente.
- Dado un emprendedor que intenta modificar el stock de un producto, cuando ingrese un valor negativo, entonces el sistema debe impedir la actualización y mostrar un mensaje de error.
- Dado un emprendedor que ha actualizado el stock correctamente, cuando finalice la actualización, entonces el sistema debe mostrar un mensaje de confirmación.

## [HU03] Búsqueda y filtrado de productos

Como emprendedor

Quiero buscar productos en mi inventario por nombre, categoría o rango de precios

Para encontrar rápidamente los artículos que necesito

Criterios de aceptación:

- Dado un emprendedor que desea buscar un producto, cuando ingrese un nombre, categoría o rango de precios, entonces el sistema debe mostrar los productos que coincidan con los criterios de búsqueda.
- Dado un emprendedor que busca productos, cuando ingrese un término de búsqueda con diferencias en mayúsculas y minúsculas, entonces el sistema debe ser flexible y mostrar los resultados correspondientes.
- Dado un emprendedor que busca productos inexistentes, cuando no haya coincidencias, entonces el sistema debe mostrar un mensaje indicando que no se encontraron productos.

## [HU04] Generación de reportes del inventario

Como emprendedor

Quiero generar un reporte del inventario actual

Para analizar la cantidad de productos disponibles y tomar decisiones informadas

Criterios de aceptación:

- Dado un emprendedor que quiere generar un reporte de inventario, cuando solicite la generación del reporte, entonces el sistema debe generar un documento con la lista de productos, sus cantidades y precios.
- Dado un emprendedor que genera un reporte de inventario, cuando se genere el reporte, entonces este debe incluir la fecha y hora de emisión.

## [HU05] Autenticación de usuarios

Como emprendedor

Quiero acceder a la aplicación mediante un usuario y contraseña

Para proteger la información del inventario

Criterios de aceptación:

- Dado un emprendedor que intenta acceder al sistema, cuando ingrese su usuario y contraseña correctos, entonces el sistema debe permitir el acceso.
- Dado un emprendedor que intenta acceder al sistema, cuando ingrese una contraseña incorrecta, entonces el sistema debe denegar el acceso y mostrar un mensaje de error.

## Organización

Para el desarrollo del proyecto, ambos integrantes participaron activamente en el desarrollo, corrección y prueba del código base, adoptando un enfoque iterativo e incremental. Inicialmente, se creó un programa base que sirvió como punto de partida, sobre el cual se realizaron modificaciones y mejoras progresivas para cumplir con los requerimientos del problema.

Para garantizar la calidad del código y su correcto funcionamiento, se llevaron a cabo múltiples ciclos de prueba en distintos niveles. Se realizaron pruebas unitarias para evaluar el comportamiento de funciones individuales y pruebas de integración para verificar la correcta interacción entre los módulos. Además, el código se ejecutó con diversas entradas para evaluar su rendimiento, corrigiendo los casos de prueba con resultados fallidos.

Una vez definidos los requerimientos, se da un enfoque a la gestión de casos de excepción y en el tratamiento de entradas no válidas. Se distinguieron y abordaron de manera específica las diferencias entre fallas (desviaciones del sistema con respecto al comportamiento esperado en ejecución), fallos (deficiencias en la implementación que generaban fallas en tiempo de ejecución), errores (equivocaciones en el código fuente que requerían corrección). Este proceso permitió robustecer el sistema, minimizando posibles problemas en su uso real.

## Pruebas y resultados.

### Ciclo 1:

En esta primera etapa, nuestro objetivo principal fue garantizar que el sistema cumpliera con los requerimientos básicos solicitados. Nos centramos en validar las funciones esenciales del sistema, asegurándonos de que operaran correctamente según lo esperado.

Durante este ciclo, realizamos diversas pruebas enfocadas en las funcionalidades fundamentales, como la generación de reportes, la gestión de productos dentro del inventario y la autenticación de usuarios.

En esta fase, logramos validar el correcto funcionamiento de las funciones esenciales del sistema. Identificamos y corregimos un problema con la ejecución de la función de generación de reportes, lo que permitió su correcta operación en pruebas posteriores.

Todas las pruebas relacionadas con la gestión del inventario y la autenticación de usuarios fueron exitosas, lo que confirma que el sistema cumple con los requerimientos básicos establecidos.

### Ciclo 2:

Una vez que confirmamos el correcto desempeño del sistema en el segundo ciclo, llevamos a cabo una serie de pruebas enfocadas en evaluar su comportamiento ante entradas no válidas.

El objetivo de esta fase fue asegurarnos de que el programa pudiera manejar errores de usuario de manera adecuada, proporcionando mensajes claros y evitando situaciones inesperadas como bloqueos o datos incorrectos en el sistema.

Para ello, se realizaron distintas pruebas simulando errores comunes, como ingresar opciones inválidas en los menús, proporcionar datos incorrectos en formularios y buscar elementos inexistentes.

Las pruebas realizadas confirmaron que el sistema maneja correctamente la mayoría de las entradas no válidas, mostrando mensajes claros y evitando la corrupción de datos.

Sin embargo, se detectaron algunas áreas de mejora que fueron usados para desarrollar un tercer ciclo de pruebas: Corrección necesaria en la validación del stock: Se permite el ingreso de valores negativos, lo cual podría generar inconsistencias en el inventario; Mejor manejo de productos con campos vacíos: Entonces, solo se validaba el stock y el precio, pero no el nombre del producto.

Bajo estas condiciones consideramos el ciclo 2 incluso más provechoso ya que nos dio material de mejora y un nuevo enfoque para futuras pruebas que no se vieron en el primer ciclo.

### Ciclo 3:

Después de haber identificado y corregido varios problemas en el segundo ciclo de pruebas, procedimos a realizar un tercer ciclo, enfocándonos en la robustez del sistema y la prevención de errores críticos que podrían afectar su estabilidad.

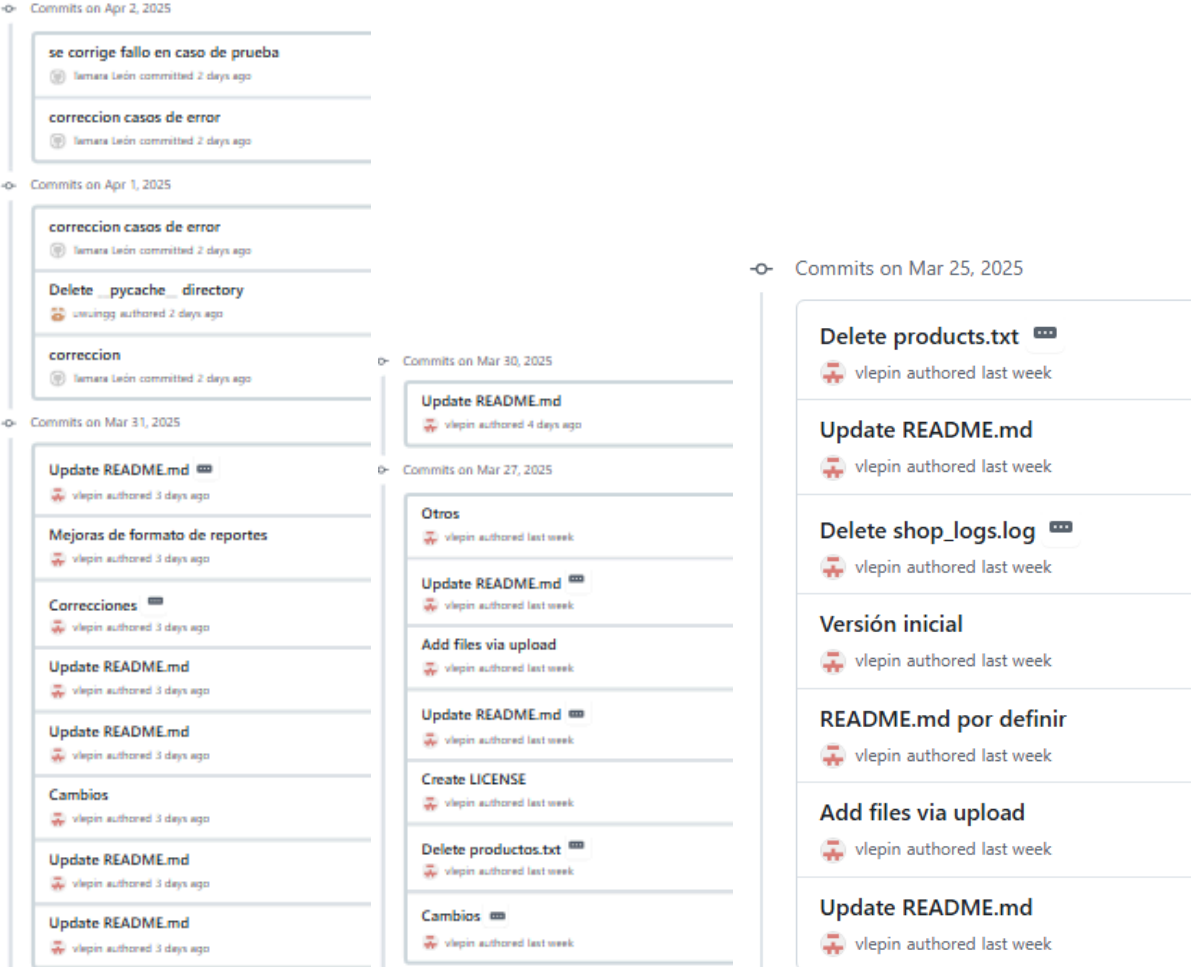
Este ciclo tuvo como objetivo probar y corregir casos en los que el sistema aún presentaba fallos, especialmente en la validación de entradas no válidas, la gestión de errores y el manejo de datos incorrectos. Se realizaron pruebas adicionales para garantizar que el sistema reaccione de forma segura y controlada ante diferentes tipos de entradas.

Las pruebas del tercer ciclo revelaron mejoras en algunas validaciones, pero aún persisten fallos críticos que corregimos para la implementación final del código, algunos fallos fueron: El sistema se cierra inesperadamente en algunos casos de error, por lo que se mejoró la gestión de excepciones para evitar que el programa termine su ejecución en escenarios de entrada no válida; Las categorías de productos aceptaban valores numéricos. Las validaciones de opciones en los menús han mejorado.

Ahora el sistema detecta correctamente opciones fuera del rango permitido en los casos estudiados para evitar un colapso del sistema.

## Evidencia de flujo de trabajo (Imágenes de pantalla).

### Flujo de trabajo visto desde Github




The screenshot displays a GitHub commit history for a repository, organized by date. The workflow is as follows:

- Commits on Apr 2, 2025:**
  - se corrige fallo en caso de prueba (lamana León committed 2 days ago)
  - correccion casos de error (lamana León committed 2 days ago)
- Commits on Apr 1, 2025:**
  - correccion casos de error (lamana León committed 2 days ago)
  - Delete `__pycache__` directory (viveingg authored 2 days ago)
  - correccion (lamana León committed 2 days ago)
- Commits on Mar 31, 2025:**
  - Update README.md (vlepin authored 3 days ago)
  - Mejoras de formato de reportes (vlepin authored 3 days ago)
  - Correcciones (vlepin authored 3 days ago)
  - Update README.md (vlepin authored 3 days ago)
  - Update README.md (vlepin authored 3 days ago)
  - Cambios (vlepin authored 3 days ago)
  - Update README.md (vlepin authored 3 days ago)
  - Update README.md (vlepin authored 3 days ago)
- Commits on Mar 30, 2025:**
  - Update README.md (vlepin authored 4 days ago)
- Commits on Mar 27, 2025:**
  - Otros (vlepin authored last week)
  - Update README.md (vlepin authored last week)
  - Add files via upload (vlepin authored last week)
  - Update README.md (vlepin authored last week)
  - Create LICENSE (vlepin authored last week)
  - Delete productos.txt (vlepin authored last week)
  - Cambios (vlepin authored last week)
- Commits on Mar 25, 2025:**
  - Delete products.txt (vlepin authored last week)
  - Update README.md (vlepin authored last week)
  - Delete shop\_logs.log (vlepin authored last week)
  - Versión inicial (vlepin authored last week)
  - README.md por definir (vlepin authored last week)
  - Add files via upload (vlepin authored last week)
  - Update README.md (vlepin authored last week)



## Integración Github - Slack



**GitHub** APP 13:01

Lunes, 31 de marzo ▾

This channel will receive notifications from **vlepin/INF331** for:
 

issues, pulls, commits: 'tarea\_1', releases, deployments


---


**GitHub** APP 13:02

1 new commit pushed to **Tarea\_1** by **vlepin**

8a44ffb0

 - Update README.md


 vlepin/INF331

---


1 new commit pushed to **Tarea\_1** by **vlepin**

45cb9848

 - Correcciones

 vlepin/INF331

---

 **GitHub** APP 13:34

Issue created by vlepin

#4 [HUD4] Generación de reportes del inventario


Yo como emprendedor  
Quiero generar un reporte del inventario actual  
Para analizar la cantidad de productos disponibles, su valor conjunto y tomar decisiones

Información:

Criterios de aceptación:


- El sistema debe generar un reporte con la lista de productos, sus cantidades y precios.
- El reporte debe incluir la fecha y hora en que fue generado.

Asignee @uvulnrgg      Labela documentation

 vlepin/INF331 · 37 min

Comment Edit Close

---

 **GitHub** APP 13:34

Issue created by vlepin


#5 [HUD5] Autenticación de usuarios

Yo como emprendedor  
Quiero acceder a la aplicación mediante un usuario y contraseña  
Para proteger la información del inventario

Criterios de aceptación:


- El sistema debe solicitar usuario y contraseña para acceder.
- Solo los usuarios registrados pueden acceder al sistema.

Asignee @uvulnrgg      Labela documentation

 vlepin/INF331 · 37 min

Comment Edit Close


---

 **GitHub** APP 13:34

1 new commit pushed to **Tarea\_1** by **vlepin**

ab8e5a1a


 - Mejoras de formato de reportes

 vlepin/INF331


1 new commit pushed to **Tarea\_1** by **vlepin**

12abaf6e

 - Update README.md

 vlepin/INF331


---

 **GitHub** APP 23:41

1 new commit pushed to **Tarea\_1** by **uvulnrgg**

43f21a6b


 - correcion

 vlepin/INF331


1 new commit pushed to **Tarea\_1** by **uvulnrgg**

ba44980a

 - Delete \_\_pycache\_\_ directory

 vlepin/INF331

---

 **GitHub** APP 00:00


2 new commits pushed to **Tarea\_1** by **uvulnrgg**

a18e789e


 - correcion casos de error

5a0a001e

 - correcion casos de error

 vlepin/INF331


---

 **GitHub** APP 01:26






1 new commit pushed to **Tarea\_1** by **uvulnrgg**

ea0e181a

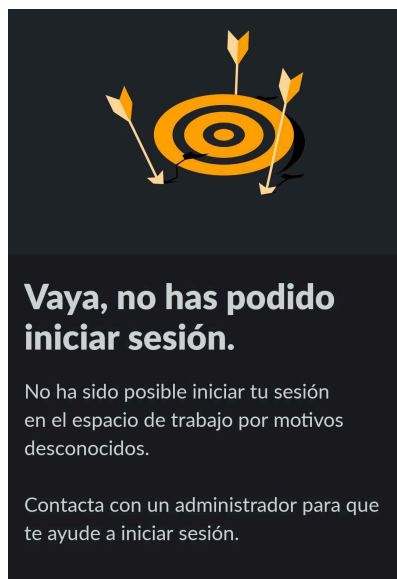
 - se corrige fallo en caso de prueba

 vlepin/INF331

## Asignación de historias de usuario

<input type="checkbox"/>	<input checked="" type="checkbox"/>	[HU05] Autenticación de usuarios <a href="#">documentation</a>	
		#5 · vlepin opened 13 minutes ago	
<input type="checkbox"/>	<input checked="" type="checkbox"/>	[HU04] Generación de reportes del inventario <a href="#">documentation</a>	
		#4 · vlepin opened 15 minutes ago	
<input type="checkbox"/>	<input checked="" type="checkbox"/>	[HU03] Búsqueda y filtrado de productos <a href="#">documentation</a>	
		#3 · vlepin opened 21 minutes ago	
<input type="checkbox"/>	<input checked="" type="checkbox"/>	[HU02] Actualización del stock de productos <a href="#">documentation</a>	
		#2 · vlepin opened 24 minutes ago	
<input type="checkbox"/>	<input checked="" type="checkbox"/>	[HU01] Registro de productos en el inventario <a href="#">documentation</a>	
		#1 · vlepin opened 27 minutes ago	

## Problemas encontrados y cómo se solucionaron.



Se presentaron problemas recurrentes de conexión con Slack, impidiendo el reconocimiento de los correos electrónicos y la recepción de los mensajes de validación necesarios para el acceso. Esto afectó directamente la integración con GitHub, ya que no se pudo ingresar a la página. Como resultado, esta parte del proyecto tuvo que ser postergada en varias ocasiones debido a fallos en la plataforma.

Dado que los problemas estaban fuera de nuestro control, la estrategia adoptada fue realizar pruebas diarias para verificar la funcionalidad de la plataforma. Finalmente, tras varios intentos, se logró completar la integración con GitHub, lo que permitió continuar con el desarrollo del proyecto y el registro de la evidencia.