ENTREGA PREVIA SEMANA 7

ETAPAS DEL PROCESO DE CONSTRUCCIÓN DE SOFTWARE

PRESENTADO POR:

JONNATHAN ALEXANDER MORA FRAILE

FACULTAD DE INGENERIA, DISEÑO E INNOVACIÓN

PROCESO DE SOFTWARE PERSONAL – B03 – GRUPO DE TRABAJO

NATALIA MARTINEZ ROJAS

INSTITUCIÓN UNIVERSITARIA POLITÉCNICO GRANCOLOMBIANO

24 DE JUNIO 2025

TABLA DE CONTENIDO

[**INTRODUCCIÓN** 4](#_Toc201696398)

[**OBJETIVOS ESPECÍFICOS** 7](#_Toc201696399)

[**JUSTIFICACIÓN DEL PROBLEMA** 7](#_Toc201696400)

[**PROJECT PLAN SUMMARY (PSP)** 9](#_Toc201696401)

[**FORMATO DE REGISTRO DE PROGRESO PERSONAL DE SOFTWARE** 11](#_Toc201696402)

[**REGISTRO DE DEFECTOS** 12](#_Toc201696403)

[**IDENTIFICAR Y CUANTIFICAR LOS PROXYS** 15](#_Toc201696404)

[**IDENTIFICAR Y CUANTIFICAR LOS PROXYS AÑADIDOS** 18](#_Toc201696405)

[**ESTIMAR EL TAMAÑO DE OTROS ELEMENTOS** 20](#_Toc201696406)

[**ESTIMAR EL TAMAÑO DEL PROGRAMA** 20](#_Toc201696407)

[**ESTIMAR EL TAMAÑO Y ESFUERZO AÑADIDO FORMA A** 20](#_Toc201696408)

[**ESTIMAR ESFUERZO** 21](#_Toc201696409)

[**RESUMEN DEL PLAN DEL PROYECTO DE PSP** 22](#_Toc201696410)

[**NIVELES Y SUBNIVELES PSP** 23](#_Toc201696411)

[**PLAN DE MEJORA PERSONAL** 24](#_Toc201696412)

[**PLAN DE PRUEBAS** 24](#_Toc201696413)

[**BIBLIOGRAFÍA** 27](#_Toc201696414)

CONTENIDO DE TABLAS

**Tabla1. Planeación de las actividades a desarrollar**8

**Tabla1. Planeación de las actividades a desarrollar**9

**Tabla2. Formato De Registro De Progreso Personal De Software HTML**……….10

**Tabla3.Tabla de registro de defectos**12

**Tabla4.Tabla de función inicio de sesión**13

**Tabla 5. Tabla de función registrar nuevo usuario**14

**Tabla 6. Tabla de función mostrar usuario** 14

**Tabla7. Tabla de función agregar producto**14

**Tabla8. Tabla de función búsqueda de producto**15

**Tabla9. Tabla de función Inventario**15

**Tabla10. Tabla de función cerrar sesión**15

**Tabla11.Tabla de función Cargar producto**17

**Tabla12.****Tabla de función guardar producto**17

**Tabla13.Tabla de función eliminar producto**17

**Tabla14.Tabla de función editar producto**17

**Tabla15.Tabla de función cargarSelectSalid producto**18

**Tabla16.Tabla de función mostrarBajoStock producto**18

**Tabla17.Tabla de función generar Grafica Inventario producto**18

**Tabla18.Tabla de Tamaño de otros elementos** 19

**Tabla19.Tabla de resumen plan proyecto psp** 21

**Tabla20.Tabla de plan de mejora personal** 23

**Tabla21.Tabla de****plan de pruebas**24

**Tabla22.Tabla de plan de proyecto a 6 meses** 25

# **INTRODUCCIÓN**

En la compañía ZITRO en la que tengo una relación laboral la cual comercializa y fabrica máquinas de tragamonedas, lo que conlleva una gran cantidad de repuestos que deben estar debidamente controlados. Debido a esto se requiere realizar uno de los procesos clave que es la gestión eficiente del inventario. Debido al crecimiento y expansión de las operaciones, el manejo manual del inventario se ha vuelto cada vez más complejo y propenso a errores ya que todo se maneja con la herramienta Excel que no siempre es lo más práctico por que toca buscar o actualizar información frecuentemente lo que afecta directamente la productividad.

Con el fin de optimizar este proceso y reducir los errores, se ha identificado la necesidad de desarrollar un portal web personalizado para la administración del inventario.

Este portal se realizará con el fin de mejorar el tiempo al ingreso Este portal facilitará una organización más ágil, eficiente y segura, permitiendo acceder a la información en tiempo real desde cualquier dispositivo.

Además, la implementación de este sistema estará alineada con la metodología **Personal Software Process (PSP)**, que proporcionará una estructura meticulosa para el desarrollo del proyecto, asegurando calidad y eficiencia en todas sus fases.

El desarrollo de este portal también será una oportunidad para aplicar mis conocimientos adquiridos en la carrera de Ingeniería de Software y electrónica, integrando soluciones tecnológicas para mejorar la eficiencia operativa en la empresa. A través de este proyecto, se busca proporcionar una herramienta de largo plazo que optimice la gestión del inventario y soporte el crecimiento continuo de ZITRO.

**DESCRIPCIÓN DEL PROBLEMA**

El manejo de los repuestos en la compañía ZITRO se ha convertido en un proceso cada vez más complejo debido a la variedad y delicadeza de los mismos, muchos de los cuales son costosos y electrónicos. Estos repuestos son importados desde el extranjero, lo que implica un largo proceso de adquisición, con tiempos de espera considerables debido a los trámites en el país de origen y en el ingreso a Colombia. (ZITRO, 2025)

La solicitud de repuestos es especialmente demorada, ya que involucra varios procesos logísticos y aduaneros. En ocasiones, la urgencia de contar con un repuesto no disponible en el inventario obliga a realizar envíos exprés, lo que aumenta significativamente el costo del repuesto debido al envío rápido. (Sánchez, 2020)

El control del inventario se ha vuelto crítico para asegurar que no haya faltantes que puedan interrumpir el funcionamiento de las máquinas tragamonedas de los clientes. Sin embargo, a pesar de organizarme con Excel, el aumento de modelos y repuestos ha generado confusión, especialmente en lo relacionado con la distinción entre repuestos nuevos y reparados. (Pérez, 2022)

La falta de un sistema más robusto y eficiente ha limitado nuestra capacidad de realizar un seguimiento adecuado y rápido de los repuestos. Esta situación impacta negativamente en la capacidad de respuesta del área técnica y, por ende, en la operación general de la empresa, pudiendo ocasionar retrasos o interrupciones en el servicio a los clientes. (Vargas, 2022)

**OBJETIVO GENERAL**

Desarrollar un portal web personalizado para la gestión el manejo y control del inventario de repuestos en la compañía ZITRO. El objetivo es crear una herramienta eficiente que permita registrar, consultar, identificar y actualizar los datos de manera rápida, reduciendo los tiempos de trabajo. Además, el portal servirá como una herramienta práctica tanto para el soporte técnico como para la toma de decisiones dentro de la empresa, mejorando la productividad y la organización en el proceso de gestión de inventarios.

## **OBJETIVOS ESPECÍFICOS**

* Diseñar y desarrollar una interfaz de usuario intuitiva para el portal web, que permita una navegación ágil y un acceso eficiente a los registros de inventario, optimizando el tiempo de búsqueda y actualización de información.
* Desarrollar un sistema que registre datos importantes de cada repuesto, como nombre, referencia, cantidad, estado (nuevo, reparado, dañado), ubicación y fechas de entrada o salida.
* Implementar una base de datos eficiente y escalable, que permita almacenar y gestionar la información de repuestos en tiempo real, minimizando los errores y mejorando la precisión de los registros.
* Pruebas: El último objetivo incluye la fase de pruebas, que es fundamental para validar el funcionamiento del sistema y garantizar su rendimiento óptimo antes de su implementación final.

# **JUSTIFICACIÓN DEL PROBLEMA**

La gestión eficiente de inventarios es un aspecto crucial en cualquier empresa, especialmente cuando se manejan repuestos electrónicos costosos y de difícil acceso, como los productos importados que comercializa ZITRO. Los repuestos de alta demanda, junto con los tiempos largos de espera debido a los trámites administrativos y logísticos tanto en el país de origen como al llegar a Colombia, generan diversos desafíos operativos y financieros. (Chopra, 2022)

señalan que “la gestión de inventarios es fundamental para reducir costos operativos y mejorar la eficiencia en toda la cadena de suministro” (p. 124), lo que hace aún más relevante la optimización de estos procesos en ZITRO.

Uno de los principales problemas es la falta de un sistema adecuado de gestión de inventarios, lo que impide una actualización constante y precisa de la disponibilidad de repuestos. Si un repuesto no está disponible en inventario y se requiere con urgencia, la única opción es realizar una compra con envío rápido, lo que incrementa significativamente los costos. Además, la variedad de modelos y las distintas condiciones de los repuestos (nuevos, reparados, dañados) contribuyen a un aumento en el riesgo de errores y confusión en la toma de decisiones, afectando directamente la productividad del área técnica.

Como afirman (Hammer, 2023)"la automatización de los procesos de gestión empresarial no solo mejora la eficiencia, sino que también reduce el margen de error humano y optimiza el uso de los recursos disponibles" (p. 75). En este contexto, un sistema que permita organizar, actualizar y consultar los datos del inventario de manera centralizada y accesible desde cualquier dispositivo no solo facilitaría la gestión diaria, sino que también contribuiría a la toma de decisiones más precisas y rápidas, especialmente cuando se deben resolver fallas operativas de manera eficiente.

Este proyecto tiene como propósito no solo mejorar los procesos operativos, sino también aplicar los conocimientos adquiridos en mi formación como tecnólogo en instrumental industrial y en ingeniería de software. Como subraya (Pressman, 2024), "la ingeniería de software aplicada adecuadamente permite desarrollar soluciones tecnológicas que optimizan las operaciones empresariales y fomentan la escalabilidad" (p. 98). El desarrollo de este portal web permitirá automatizar tareas, reducir los errores humanos y proporcionar al equipo técnico una herramienta confiable para gestionar los inventarios de manera más efectiva, optimizando los recursos y contribuyendo a la sostenibilidad operativa de ZITRO.

# **PROJECT PLAN SUMMARY (PSP)**

**PLANEACIÓN DE LAS ACTIVIDADES A DESARROLLAR**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Fase** | **Semana** | **Actividades** | **Objetivo** | **Notas** |
| Fase 1 | 1 | Defirnir estructura del portal (páginas necesarias). Crear Bocetos. Definir datos a manejar. Implemnetación de logo | Planeación y diseño | Usar Visual Studio y bocetos a mano. Definir colores, fuentes e iconos |
| Fase 1 | 2 | Crear carpetas /css, /js, /images. Crear páginas principales: index.html, inventario.html, salidas.html, entradas.html. Agregar encabezado y pie. | Estructura en HTML | Estructura básica para navegación. |
| Fase 1 | 3 | Crear style.css y vincularlo. Aplicar estilos (colores, márgenes, botones, fuentes). Diseñar tarjetas visuales en página principal. | Diseño con CSS | Aplicación de identidad visual. |
| Fase 1 | 4 | Crear tabla dinámica de inventario. Permitir agregar filas. Usar localStorage para guardar datos temporalmente. | Tablas funcionales con JavaScript | Sin base de datos aún. Datos Guardados localmente |
| Fase 1 | 5 | Implementar sistema de inicio de sesión con validación de usuarios y control de acceso. | Seguridad y autenticación | Uso de localStorage y sessionStorage. Restricción por rol. |
| Fase 1 | 6 | Diseño gráfico con chart.js. Registrar entradas y salidas con detalle. Agregar tipo de movimiento: Venta y edición, eliminación | Gestión de inventario | Movimientos de productos funcionales y grafico de productos |
| Fase 1 | 7 | Implementar inicio de sesión, Agregar roles de Admin / Lector | Seguridad básica | Formularios de usuarios. |
| Fase 1 | 8 | Pruebas completas, corregir errores, preparar entrega en GitHub. | Cierre de Fase 1 | Verificación visual y funcional. |
| Fase 2 | 1-24 | Implementar base de datos (MySQL o Firebase). Backend con Node.js o PHP. Autenticación. Formularios que guarden datos. Historial y alertas. | Versión avanzada con base de datos, login y control real del inventario. | Duración: 6 meses (24 semanas aproximadamente). |

*Tabla1. Planeación de las actividades a desarrollar*

## **FORMATO DE REGISTRO DE PROGRESO PERSONAL DE SOFTWARE**

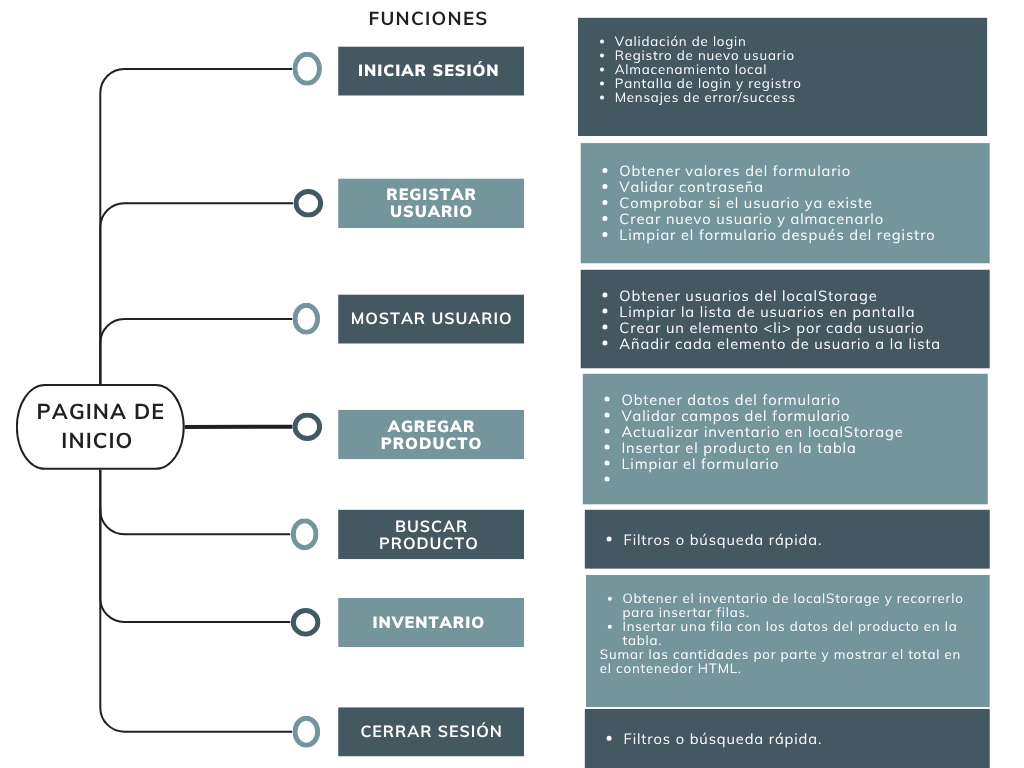
*Tabla2.**Formato De Registro De Progreso Personal De SoftwareHTML*

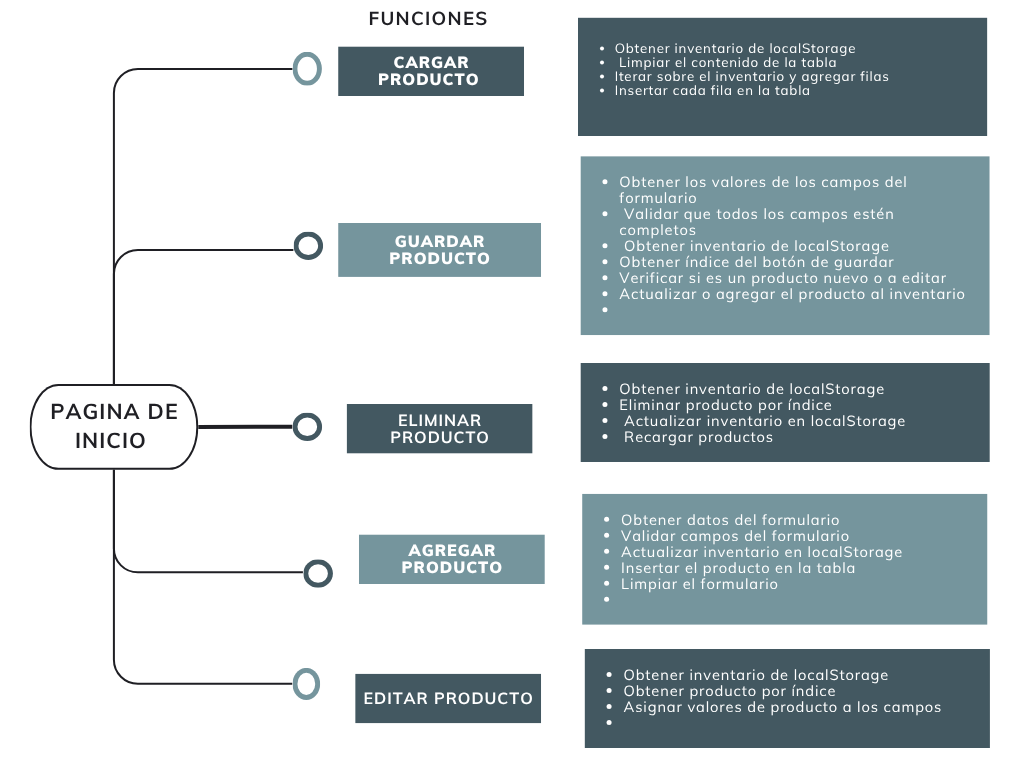
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Fecha | Número / Identificador | Tipo de Defecto | Fase donde aparece | Eliminado durante | Tiempo (min) | Defecto Corregido |
| 20/05/2025 | 1 | 40 (Asignación) | Codificación | Pruebas visuales | 20 | Error de id duplicado en total Productos |
| 22/05/2025 | 2 | 80 (Función) | Codificación | Revisión JS | 50 | No se actualizaba contador de productos |
| 22/05/2025 | 3 | 60 (Chequeo) | Codificación | Revisión lógica | 30 | Filtro de búsqueda no reconocía estados |
| 29/05/2025 | 4 | 100 (Entorno) | Codificación | Diseño/Pruebas | 35 | Error visual por mal cierre de div |
| 6/06/2025 | 5 | 50 (Interfaz) | HTML-JS Integración | Compilación manual | 30 | Falta de enlace entre botones y funciones |
| 13/06/2025 | 6 | 70(Datos) | Inicio - Graficos | Visualización pruebas | 50 | Graficos no actilizaban datos dinamicamente |
| 19/06/2025 | 7 | 90(Sistema) | Login / Seguridad | revisión de Roles | 45 | Login sin validación activa ni control por rol |
| 23/06/2025 | 8 | 30(Contruir, paquetes) | Formularios Y validaciones | Pruebas funcionales | 40 | Campos permitan envío con datos vacíos |

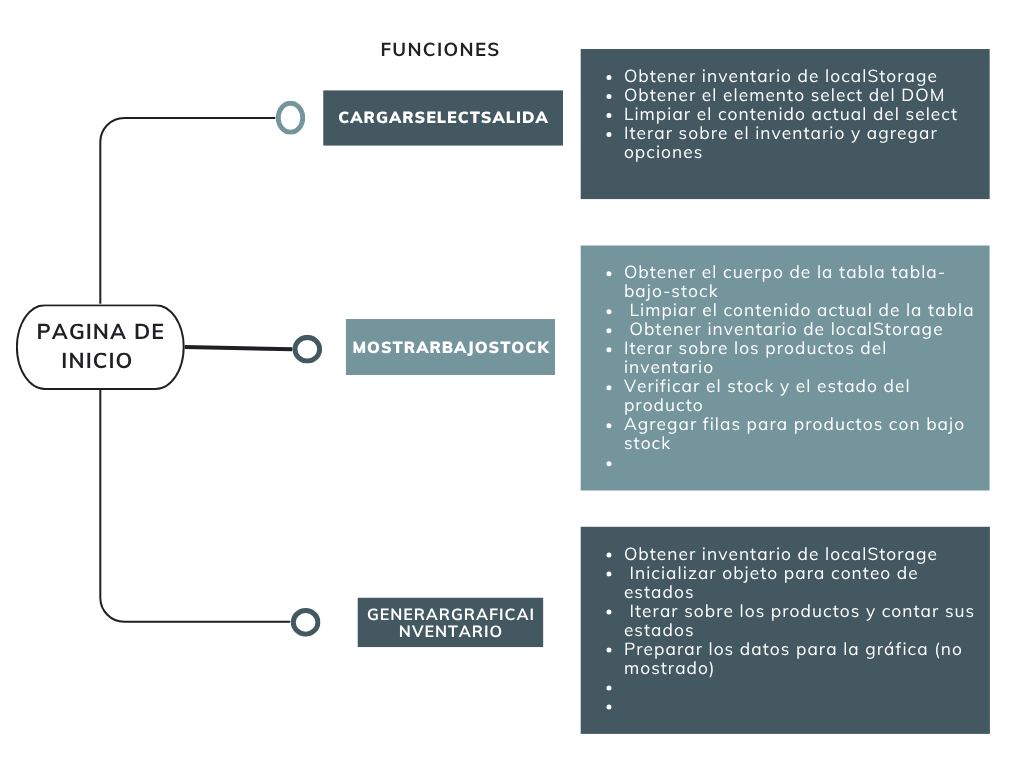
# **REGISTRO DE DEFECTOS**

*Tabla3.Tabla de registro de defectos*

**DISEÑO CONCEPTUAL**







# **IDENTIFICAR Y CUANTIFICAR LOS PROXYS**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Función** | **Inicio de sesión** | | | | |
| **Proxy** | **Subfunciones** | **Tipo** | **Tamaño** | **PA(LOC)** | **TOTAL PA(LOC)** |
| **Validación de login** | Captura de datos (login) | Lógica | Grande | 40 | 160 |
| Comparación de credenciales | Lógica | Pequeño | 50 |
| Manejo de errores (login fallido) | UI | Pequeño | 30 |
| Lectura desde localStorage (login) | Datos | Pequeño | 40 |
| **Registro de nuevo usuario** | Validación de campos del formulario de registro | Lógica | Pequeño | 40 | 220 |
| Verificación de usuario existente | Lógica | Pequeño | 50 |
| Validación de contraseña mínima | Lógica | Pequeño | 30 |
| Almacenamiento en localStorage (registro) | Datos | Pequeño | 40 |
| Conversión JSON.stringify / JSON.parse | Datos | Pequeño | 30 |
| Inicialización de usuarios en memoria | Lógica | Pequeño | 30 |
| **Almacenamiento local (usuarios)** | Almacenamiento en localStorage (registro) | Datos | Pequeño | 40 | 100 |
| Conversión JSON.stringify / JSON.parse | Datos | Pequeño | 30 |
| Inicialización de usuarios en memoria | Datos | Pequeño | 30 |
| **Pantalla de login y registro** | Estructura HTML del formulario login | UI | Mediano | 90 | 240 |
| Estructura HTML del formulario de registro | UI | Mediano | 90 |
| Campo select de rol | UI | Pequeño | 30 |
| Botones y eventos de envío | UI | Pequeño | 30 |
| **Mensajes de error/success** | Mensaje de éxito (registro correcto) | UI | Pequeño | 40 | 40 |

*Tabla4.Tabla de función inicio de sesión*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Función** | **Registrar nuevo usuario** | | | | |
| **Proxy** | **Subfunciones** | **Tipo** | **Tamaño** | **PA(LOC)** | **PA (LOC) TOTAL** |
| registrarUsuario() | Obtener valores del formulario | UI | Grande | 10 | 47 |
| Validar contraseña | Lógica | Mediano | 15 |
| Comprobar si el usuario ya existe | Lógica | Pequeño | 5 |
| Crear nuevo usuario y almacenarlo | Datos | mediano | 10 |
| Limpiar el formulario después del registro | UI | Pequeño | 7 |

*Tabla 5. Tabla de función registrar nuevo usuario*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Función** | **Mostrar usuarios** | | | | |
| **Proxy** | **Subfunciones** | **Tipo** | **Tamaño** | **PA(LOC)** | **PA (LOC) TOTAL** |
| mostrarUsuarios() | Obtener usuarios del localStorage | Datos | Grande | 15 | 50 |
| Limpiar la lista de usuarios en pantalla | UI | Pequeño | 5 |
| Crear un elemento <li> por cada usuario | UI | Mediano | 20 |
| Añadir cada elemento de usuario a la lista | UI | Pequeño | 10 |

*Tabla 6. Tabla de función mostrar usuario*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Función** | **Agregar producto** | | | | |
| **Proxy** | **Subfunciones** | **Tipo** | **Tamaño** | **PA(LOC)** | **PA (LOC) TOTAL** |
| agregarProducto() | Obtener datos del formulario | UI | Pequeño | 10 | 45 |
| Validar campos del formulario | Lógica | Pequeño | 15 |
| Actualizar inventario en localStorage | Datos | Pequeño | 5 |
| Insertar el producto en la tabla | UI | Mediano | 5 |
| Limpiar el formulario | UI | Pequeño | 10 |

*Tabla7. Tabla de función agregar producto*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Función** | **Búsqueda de producto** | | | | |
| **Proxy** | **Subfunciones** | **Tipo** | **Tamaño** | **PA(LOC)** | **PA (LOC) TOTAL** |
| **Filtros o búsqueda rápida** | Captura de input | Lógica | Pequeño | 20 | 50 |
| Filtrado de datos | Lógica | Pequeño | 30 |

*Tabla8. Tabla de función búsqueda de producto*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Función** | **Inventario** | | | | |
| **Proxy** | **Subfunciones** | **Tipo** | **Tamaño** | **PA(LOC)** | **PA(LOC) TOTAL** |
| Cargar inventario | Obtener el inventario de localStorage y recorrerlo para insertar filas. | UI | pequeño | 50 | 80 |
| Insertar fila | Insertar una fila con los datos del producto en la tabla. | UI | pequeño | 20 |
| Calcular total productos | Sumar las cantidades por parte y mostrar el total en el contenedor HTML. | Datos | pequeño | 10 |

*Tabla 9. Tabla de función Inventario*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Función** | **cerrar sesión** | | | | |
| **Proxy** | **Subfunciones** | **Tipo** | **Tamaño** | **PA(LOC)** | **PA (LOC) TOTAL** |
| cerrarSesión() | Eliminación del item 'usuarioActivo' en sessionStorage y redirección a la página de login | UI | pequeño | 50 | 50 |

*Tabla10. Tabla de función cerrar sesión*

# **IDENTIFICAR Y CUANTIFICAR LOS PROXYS AÑADIDOS**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Función** | **Cargar producto** | | | | |
| **Proxy** | **Subfunciones** | **Tipo** | **Tamaño** | **PA(LOC)** | **PA (LOC) TOTAL** |
| Cargar producto | Obtener inventario de localStorage | Datos | Pequeño | 5 | 35 |
| Limpiar el contenido de la tabla | UI | Pequeño | 5 |
| Iterar sobre el inventario y agregar filas | Lógica | Mediano | 15 |
| Insertar cada fila en la tabla | UI | Mediano | 10 |

*Tabla11. Tabla de función Cargar producto*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Función** | **Guardar Producto** | | | | |
| **Proxy** | **Subfunciones** | **Tipo** | **Tamaño** | **PA(LOC)** | **PA (LOC) TOTAL** |
| guardarProducto() | Obtener los valores de los campos del formulario | UI | Pequeño | 5 | 50 |
| Validar que todos los campos estén completos | Lógica | Pequeño | 10 |
| Obtener inventario de localStorage | Datos | Pequeño | 5 |
| Obtener índice del botón de guardar | UI | Pequeño | 5 |
| Verificar si es un producto nuevo o a editar | Lógica | Pequeño | 10 |
| Actualizar o agregar el producto al inventario | Datos | Mediano | 15 |

*Tabla12. Tabla de función guardar producto*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Función** | **Eliminar producto** | | | | |
| **Proxy** | **Subfunciones** | **Tipo** | **Tamaño** | **PA(LOC)** | **PA (LOC) TOTAL** |
| Eliminar producto | Obtener inventario de localStorage | Datos | Pequeño | 5 | 40 |
| Eliminar producto por índice | Lógica | Pequeño | 10 |
| Actualizar inventario en localStorage | Datos | Pequeño | 10 |
| Recargar productos | UI | Mediano | 15 |

*Tabla13. Tabla de función eliminar producto*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Función** | **Editar producto** | | | | |
| **Proxy** | **Subfunciones** | **Tipo** | **Tamaño** | **PA(LOC)** | **PA (LOC) TOTAL** |
| Editar producto | Obtener inventario de localStorage | Datos | Pequeño | 5 | 35 |
| Obtener producto por índice | Lógica | Pequeño | 10 |
| Asignar valores de producto a los campos | UI | Mediano | 20 |

*Tabla14. Tabla de función editar producto*

*Tabla15. Tabla de función cargarSelectSalid producto*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Función** | **MostrarBajoStock(alarma)producto** | | | | |
| **Proxy** | **Subfunciones** | **Tipo** | **Tamaño** | **PA(LOC)** | **PA (LOC) TOTAL** |
| mostrarBajoStock() | Obtener el cuerpo de la tabla tabla-bajo-stock | UI | Pequeño | 5 | 70 |
| Limpiar el contenido actual de la tabla | UI | Pequeño | 5 |
| Obtener inventario de localStorage | Datos | Pequeño | 10 |
| Iterar sobre los productos del inventario | Lógica | Mediano | 20 |
| Verificar el stock y el estado del producto | Lógica | Mediano | 15 |
| Agregar filas para productos con bajo stock | UI | Mediano | 15 |
|  |  |  |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Función** | **CargarSelectSalida producto** | | | | |
| **Proxy** | **Subfunciones** | **Tipo** | **Tamaño** | **PA(LOC)** | **PA (LOC) TOTAL** |
| cargarSelectSalida() | Obtener inventario de localStorage | Datos | Pequeño | 5 | 50 |
| Obtener el elemento select del DOM | UI | Pequeño | 10 |
| Limpiar el contenido actual del select | UI | Mediano | 20 |
| Iterar sobre el inventario y agregar opciones | Lógica | Mediano | 15 |

*Tabla16. Tabla de función mostrarBajoStock producto*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Función** | **GenerarGraficaInventario() producto** | | | | |
| **Proxy** | **Subfunciones** | **Tipo** | **Tamaño** | **PA(LOC)** | **PA (LOC) TOTAL** |
| generarGraficaInventario() | Obtener inventario de localStorage | Datos | Pequeño | 5 | 40 |
| Inicializar objeto para conteo de estados | Lógica | Pequeño | 5 |
| Iterar sobre los productos y contar sus estados | Lógica | Mediano | 10 |
| Preparar los datos para la gráfica (no mostrado) | Lógica | Mediano | 20 |

*Tabla17. Tabla de función generar Grafica Inventario producto*

|  |  |  |  |
| --- | --- | --- | --- |
| **Componente** | **PA(LOC)** | **BA (LOC)** | **M (LOC)** |
| Inicio de sesión | 760 | 0 | 0 |
| Registrar nuevo usuario | 47 | 0 | 0 |
| Mostrar usuarios | 50 | 0 | 0 |
| Agregar producto | 45 | 0 | 0 |
| Búsqueda de producto | 45 | 0 | 0 |
| Inventario | 80 | 0 | 0 |
| cerrar sesión | 50 | 0 | 0 |
| Cargar producto | 35 | 0 | 0 |
| Guardar producto | 50 | 0 | 0 |
| Eliminar producto | 40 | 0 | 0 |
| Editar producto | 35 | 0 | 0 |
| Cargar Select Salida | 50 | 0 | 0 |
| Mostrar BajoStock | 70 | 0 | 0 |
| Generar Grafica Inventario | 40 | 0 | 0 |

# **ESTIMAR EL TAMAÑO DE OTROS ELEMENTOS**

*Tabla18.Tamaño de otros elementos*

# **ESTIMAR EL TAMAÑO DEL PROGRAMA**

E= 1397

# **ESTIMAR EL TAMAÑO Y ESFUERZO AÑADIDO FORMA A**

Tamaño del programa LOC: 1.077

Tamaño añadido LOC (PA) = 320

Tamaño Total LOC (PA) = 1.077 + 320 = 1.397

# **ESTIMAR ESFUERZO**

Datos históricos: 2 horas por 100 líneas de código

Cálculo de esfuerzo LOC añadidos: = 6.4 horas

Esfuerzo total: 28 horas

Esfuerzo = a + b x PA

a = constante de ajuste = 0%

b = coeficiente basado en datos históricos

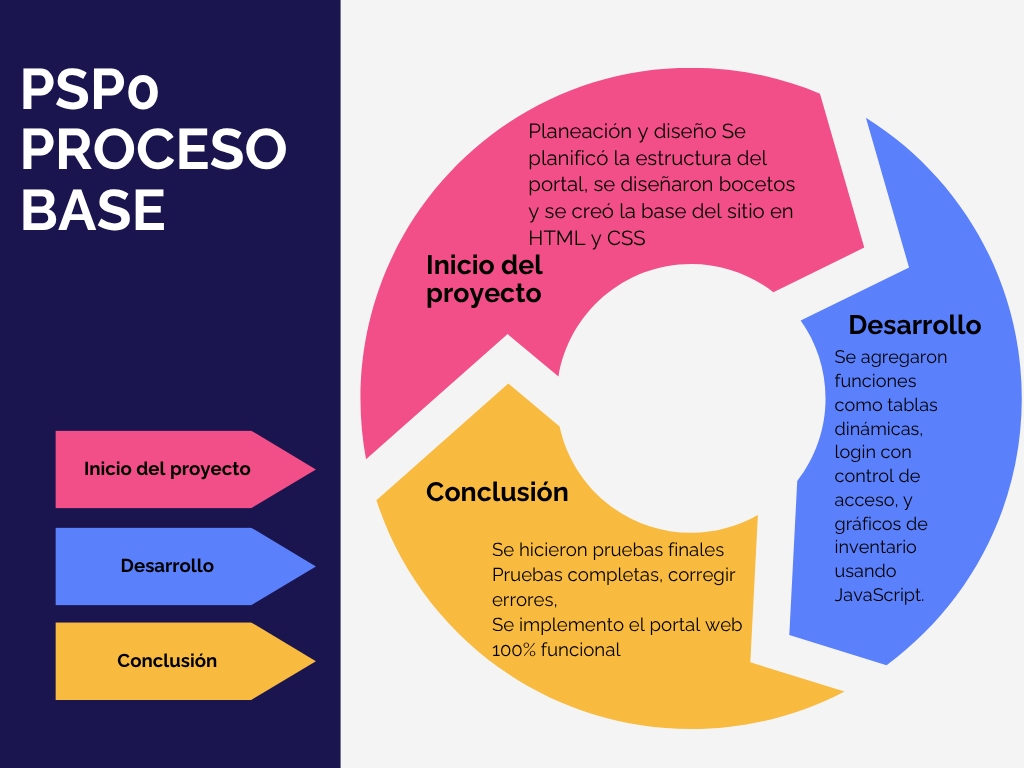
PA = tamaño total (en líneas de código, LOC) = 1.397

= 27.94 horas

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **RESUMEN DEL PLAN DEL PROYECTO DE PSP** | | | | | |
| **Nombre** | Jonnathan Alexander Mora fraile | | **Fecha** | 24-jun-25 | |
| **Programa** | Proceso de Software Personal | | **Programa #** | B03 | |
| **Profesor** | Natalia Marinez Rojas | | **Lenguaje** | HTML Java scrpt CSS | |
| **Tiempo por fase (Minutos)** | **Plan** | **Real** | **Ejecutado** | **%To Date** | **Estado** |
| Análisis de Requerimientos | 60 | 60 | 60 | 100% | Terminado |
| Diseño (Diagrama de Clases) | 180 | 250 | 250 | 1.38 % | Terminado |
| Codificación y pruebas | 120 | 130 | 130 | 1.08% | Terminado |
| Documentación | 100 | 100 | 100 | 100% | Terminado |
| **Total** | 460 | 540 | 540 | 100% | Terminado |
| **Tamaño del programa (LCD)** | **Plan** | **Real** | **Ejecutado** | **%To Date** | **Estado** |
| Total líneas nuevas + Cambiadas | 1500 | 1397 | 1397 | 93% | Terminado |
| Tamaño máximo | 1500 | 1397 | 1397 | 93% | Terminado |
| Tamaño mínimo | 1300 | 1397 | 1397 | 100% | Terminado |
| Tamaño Documentación | 1300 | 1397 | 1397 | 100% | Terminado |
| **Total** | 5600 | 5588 | 5588 | 98% | Terminado |
| **Bugs Encontrados** | **Plan** | **Real** | **Ejecutado** | **%To Date** | **Estado** |
| Análisis de Requerimientos | 3 | 3 | 3 | 100% | Terminado |
| Diseño(Diagrama de clases) | 0 | 0 | 0 | 0 | 0 |
| Codificación y pruebas | 4 | 4 | 4 | 100% | Terminado |
| Documentación | 0 | 0 | 0 | 0 | 0 |
| **Total** | 7 | 7 | 7 | 100% | Terminado |
| **Bugs Retirados** | **Plan** | **Real** | **Ejecutado** | **%To Date** | **Estado** |
| Análisis de Requerimientos | 0 | 0 | 0 | 0 | 0 |
| Diseño(Diagrama de clases) | 0 | 0 | 0 | 0 | 0 |
| Codificación y pruebas | 3 | 3 | 3 | 100% | Terminado |
| Documentación | 0 | 0 | 0 | 0 | 0 |
| Total retirados en el desarrollo | 3 | 3 | 3 | 100% | Terminado |
| Total retirados después del desarrollo | 6 | 4 | 4 | 100% | Terminado |
| **Resumen** | **Plan** | **Real** | **Ejecutado** | **%To Date** | **Estado** |
| Minutos / líneas de códigos | 1500 | 1379 | 1379 | 98% | Terminado |
| Líneas de código por hora | 50 | 50 | 50 | 100% | Terminado |
| Defectos / 1000 líneas de código | 60 | 40 | 40 | 66% | Terminado |

*Tabla19. Tabla de resumen plan proyecto psp*

# **NIVELES Y SUBNIVELES PSP**

****

# **PLAN DE MEJORA PERSONAL**

|  |  |  |  |
| --- | --- | --- | --- |
| **Área de Mejora** | **Situación Observada** | **Acción de Mejora Propuesta** | **Beneficio Esperado** |
| **Gestión del tiempo e interrupciones** | Varias fases presentan altos tiempos de interrupción (hasta 70 min), afectando el tiempo real de ejecución | Establecer bloques de trabajo con pausas programadas y reducir interrupciones externas con técnicas como Pomodoro | Mayor concentración y cumplimiento de tiempos previstos |
| **Estimación de tiempos de desarrollo** | Se observaron desfases negativos de hasta -90 minutos (ej. codificación inicial) | Usar métricas históricas propias para ajustar estimaciones futuras y descomponer tareas grandes en subtareas | Estimaciones más realistas y menor diferencia entre lo planificado y real |
| **Planificación de soporte externo** | Múltiples fases se vieron afectadas por soporte a cliente o consultas a compañeros | Dedicar tiempos específicos para consultas o soporte y documentar soluciones para futuras referencias | Menor impacto de imprevistos en el desarrollo principal |
| **Mejora en el aprendizaje de herramientas** | Hubo interrupciones por falta de dominio en Visual Studio, estilos y funciones | Capacitación previa en herramientas clave antes de iniciar cada fase del proyecto | Menor curva de aprendizaje durante ejecución y mayor fluidez técnica |
| **Priorización de fases críticas** | Fases como diseño gráfico y control de acceso presentan desfases altos por tareas no previstas | Repriorizar tareas críticas en fases tempranas y realizar pruebas parciales antes de avanzar | Menor acumulación de errores y ajustes en fases finales |
| **Documentación de errores y soluciones** | Interrupciones causadas por búsqueda de errores o estilo | Crear bitácora de errores frecuentes y sus soluciones, y usarla como referencia interna | Resolución más rápida de problemas recurrentes |

*Tabla20. Tabla de plan de mejora personal*

# **PLAN DE PRUEBAS**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Función/Módulo** | **Veces de prueba** | **Tipo de Prueba** | **Entrada Esperada** | **Resultado Esperado** | **Criterio de Éxito** |
| P1 | Inicio de sesión | 3 | Prueba funcional | Usuario y contraseña válidos | Acceso al portal con sesión iniciada | Redirección correcta y datos visibles |
| P2 | Inicio de sesión | 4 | Prueba negativa | Usuario o contraseña incorrectos | Mensaje de error y bloqueo de acceso | Error visible sin redireccionamiento |
| P3 | Registro de nuevo usuario | 3 | Prueba funcional | Datos completos y contraseña válida | Usuario registrado y guardado en localStorage | Registro exitoso y mensaje de confirmación |
| P4 | Registro de nuevo usuario | 5 | Prueba de validación | Campos vacíos o contraseña inválida | Error de validación, sin guardar usuario | Mensajes de error apropiados |
| P5 | Agregar producto | 4 | Prueba funcional | Datos completos del producto | Producto agregado a inventario y visible en tabla | Registro correcto y persistencia en sesión |
| P6 | Agregar producto | 5 | Prueba de validación | Campos incompletos o datos inválidos | Producto no registrado, mostrar mensajes de error | Validación impide guardar datos erróneos |
| P7 | Buscar producto | 4 | Prueba funcional | Término de búsqueda existente | Filtrado correcto en la tabla de productos | Solo productos coincidentes se muestran |
| P8 | Buscar producto | 3 | Prueba límite | Término inexistente | Mensaje de "sin resultados" o tabla vacía | Resultado coherente con búsqueda |
| P9 | Cerrar sesión | 4 | Prueba funcional | Click en botón "Cerrar sesión" | Redirección al login y eliminación de sessionStorage | Sesión terminada correctamente |
| P10 | Mostrar bajo stock | 5 | Prueba lógica y visual | Productos con cantidades mínimas (stock < límite definido) | Solo productos críticos se muestran en la tabla de alertas | Detección precisa y visual de inventario crítico |
| P11 | Generar gráfica de inventario | 2 | Prueba visual y de datos | Datos de inventario con distintos estados | Renderización de gráfico con categorías y colores correspondientes | Visualización clara y datos precisos |
| P12 | Editar producto | 3 | Prueba funcional | Selección de producto + datos editados | Datos actualizados y reflejados en tabla | Cambios guardados correctamente |
| P13 | Eliminar producto | 4 | Prueba funcional | Click en botón de eliminar | Producto eliminado del inventario y actualizado en pantalla | Producto desaparece y datos se actualizan |

*Tabla21. Tabla de plan de pruebas*

**PLAN DE PROYECTO A 6 MESES**

|  |  |  |  |
| --- | --- | --- | --- |
| **Mes** | **Actividad Principal** | **Subactividades** | **Resultado Esperado** |
| 1 | *Planificación técnica y arquitectura del sistema* | - Elegir tipo de base de datos (MySQL/Firebase)- Definir estructura de tablas- Especificar campos sensibles sujetos a Habeas Data | Documento de especificación y base de datos estructurada |
| 2 | *Desarrollo de backend y conexión con base de datos* | - Implementar API con Node.js o PHP- Conectar frontend con BD- Configurar operaciones CRUD- Probar conexión | Backend funcional conectado a BD con operaciones básicas |
| 3 | *Control de usuarios y roles con seguridad* | - Autenticación con hashing- Login con tokens/session- Roles Admin/Lector- Registro con consentimiento de tratamiento de datos | Sistema de acceso seguro y conforme con Habeas Data |
| 4 | *Protección de datos personales (Habeas Data)* | - Alerta de privacidad- Checkbox obligatorio de consentimiento- Políticas visibles- Registro de auditoría (log de cambios por usuario) | Sistema conforme con normativas de protección de datos |
| 5 | *Funciones avanzadas y reportes* | - Inventario en tiempo real- Reportes por fecha, estado, movimiento- Exportación a Excel o PDF- Alertas automatizadas | Funcionalidades completas para gestión y análisis |
| 6 | *Pruebas finales, documentación y despliegue* | - Pruebas de usuario- Manual técnico y de usuario- Despliegue en servidor o hosting- Evaluación post mortem del proyecto | Proyecto desplegado, funcional y documentado |

*Tabla22. Tabla de plan de proyecto a 6 meses*

**CONCLUSION**

El desarrollo del portal web para la gestión de inventario en la empresa ZITRO ha permitido evidenciar la importancia de aplicar metodologías estructuradas como el Proceso de Software Personal (PSP) para alcanzar un producto funcional, eficiente y alineado con las necesidades reales de la organización. A lo largo del proceso se identificaron los problemas clave del manejo manual del inventario, como los errores, la duplicación de datos y la falta de visibilidad, proponiendo una solución tecnológica que integra HTML, CSS, JavaScript y almacenamiento en localStorage, con proyecciones futuras hacia una implementación con base de datos. El registro sistemático de tiempos, defectos y líneas de código permitió controlar el avance del proyecto, mejorando su calidad y facilitando la toma de decisiones. En suma, este trabajo demuestra cómo la combinación de conocimientos técnicos y disciplina en el desarrollo puede transformar procesos críticos, optimizando la operación y fortaleciendo la capacidad de respuesta de la empresa.

# **BIBLIOGRAFÍA**

Chopra, S. &. (2022). *Supply Chain Management: Strategy, Planning, and Operation .* Pearson Education.

Hammer, M. (2023). *The Process Audit. Harvard Business Review.*

Pérez, M. &. (2022). *Gestión de inventarios en sectores tecnológicos: Herramientas y metodologías.* Editorial Tecnológica.

Pressman, R. S. (2024). *Software Engineering: A Practitioner's Approach .*

Sánchez, L. &. (2020). *Logística y costos: Retos en la cadena de suministro moderna.* Ediciones Logísticas.

Vargas, A. &. (2022). *Sistemas integrales para la gestión de repuestos y piezas especializadas.* Editorial Innovación Empresarial.

ZITRO. (2025). Zitrogames. https://zitrogames.com/es/zitro-se-establece-en-colombia-y-peru/.

*afor digital [@afor\_lives]. (n.d.). CURSO de FIGMA 2025 DESDE CERO | curso de introducción. Youtube. Retrieved May 28, 2025, from* <https://www.youtube.com/watch?v=VdS9ZGHHXWQ>

*Coper, P. (2022, January 18). Home. Zitrogames.*

<https://zitrogames.com/es/>