**Manual Técnico**

1. **Roles / integrantes**

Team leader/ Farfan Villamonte Roger Brandon

Git Master/ Bueno Lara Sebastian Jacob

Dev/ Ocampo Rivera Cinthia

1. **Introducción:**

Este proyecto, parte de la asignatura de Proyecto de Sistemas 3, presenta una pagina web diseñada para satisfacer las necesidades operativas de la sección de almacenes de la fuerza aérea boliviana (FAB), con un enfoque eficiente en la carga de datos la pagina permite registrar nuevas ordenes de compras, gestionar el inventario y también agregar registros de salidas.

1. **Descripción del proyecto:**

La pagina web se centra en la parte de gestión de almacenes, donde muestra una lista de las ordenes de compra y la opción de agregar una nueva orden de compra una vez ingresado los datos de los nuevos componentes, una vez registrado en la sección recepción de material se ingresa a la opción de ingresos pendientes donde ahí se autoriza a que los componentes ingresen al inventario, luego si se desea eliminar el componente del inventario, se realiza un registro de orden de salida, donde detalla el componente, en que almacén se encuentra y se registra como una nueva orden de salida que se puede apreciar como una lista.

1. **Listado de los Requisitos Funcionales del Sistema**

***Sprint 1***

*Crear lista inicial de componentes.*

*Crear lista de partes de un componente.*

*Crear la lista inicial de otros elementos que no son componentes.*

*Crear lista de partes de otros elementos que no son componentes.*

*Crear el detalle final de un componente en una ventana emergente.*

*Crear la lista de detalles de la parte de un componente.*

***Sprint 2***

*Desarrollar método GET para obtener la lista de componentes.*

*Desarrollar el método GET para obtener la lista de elementos que componen una parte del componente.*

*Desarrollar el método GET para obtener el detalle final del elemento que compone la parte del componente.*

***Sprint 3***

*Desarrollar el método GET del componente volátil y no volátil.*

*Desarrollar el método GET del componente volátil según su número de parte para visualizar el número de serie.*

*Desarrollar el método GET para mostrar el detalle final de los componentes volátil y no volátil.*

***Módulo de Herramientas***

***Sprint 2***

*Desarrollar interfaz para el registro de herramientas.*

*Desarrollar interfaz de registro de herramientas especiales.*

*Desarrollar interfaz de salida de herramientas.*

*Desarrollar interfaz - registro de responsable encargado de las herramientas sacadas.*

*Desarrollar interfaz de registro de fechas de calibración de herramientas especiales.*

***Sprint 3***

*Desarrollar el método GET para obtener la lista completa de herramientas.*

*Desarrollar el método POST para agregar una nueva herramienta al inventario.*

*Desarrollar el método POST para agregar una nueva herramienta al inventario de herramientas especiales.*

*Desarrollar el método GET para obtener la lista completa de herramientas especiales.*

***Módulo de Órdenes de Compra y Salida***

***Sprint 1***

*Crear registro de la orden de salida.*

*Crear detalle de la orden de compra.*

*Crear formulario para agregar una orden de compra.*

*Crear un registro para agregar una orden de compra.*

*Crear lista de órdenes de compra.*

*Crear lista de órdenes de salida.*

***Sprint 2***

*Desarrollar el método GET para obtener la lista de la orden de salida.*

***Sprint 3***

*Desarrollar el GET para orden de compra con todos los joins correspondientes.*

*Desarrollar los GETs para el formulario de orden de compra.*

*Desarrollar el POST para el formulario de orden de compra.*

*Desarrollar el método POST para registrar una nueva orden de salida.*

***Módulo de Recepción e Inventario***

***Sprint 1***

*Crear lista de recepción.*

*Crear registro de componentes en el almacén en el apartado de ingresos pendientes.*

*Crear registro de otros elementos que no son componentes en el almacén en el apartado de ingresos pendientes.*

***Sprint 2***

*Desarrollar backend para ingreso de pendientes.*

*Desarrollar el método GET para obtener la lista de ingresos en el apartado de recepción.*

*Desarrollar el método POST para el registro de una nueva recepción.*

***Sprint 3***

*Desarrollar el ingreso de pendientes de tipo no volátil y volátil.*

***Módulo General del Sistema***

***Sprint 1***

*Crear un menú general que se usará en todo el sistema.*

***Sprint 2***

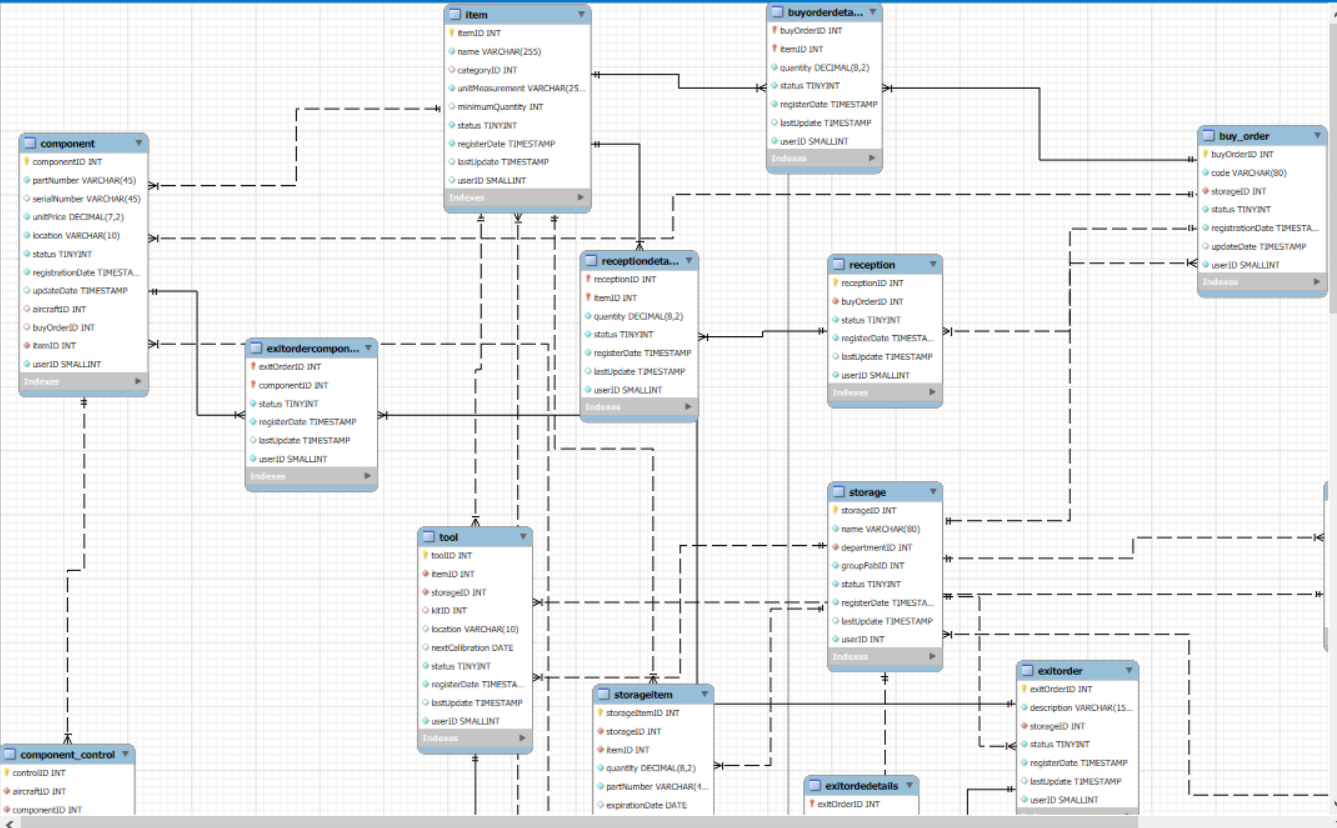
*Desarrollar interfaz de control de inventario - materiales.*

1. **Arquitectura del software: *Módulo de Componentes y Materiales***

* **Gestión de orden de compra**
* **Gestión de recepción de material**
* **Gestión de inventario**
* **Gestión de orden de salida**

1. **Base de datos**
   1. **Diagrama completo y actual**

****

****

(estas son las tablas utilizadas para realizar el proyecto, sin embargo el diagrama del proyecto general es muy mas extenso)

* 1. **En el GIT una carpeta con la base de datos con script de generación e inserción de datos de ejemplo utilizados**
  2. **Script simple (copiado y pegado en este documento)**

****

1. **Requisitos del sistema:**

Al ser un sistema web, solo se necesita un navegador para utilizar el sistema.

1. **Glosario de términos:**

**C#**

Es un lenguaje de programación orientado a objetos desarrollado por Microsoft, que forma parte de la plataforma .NET. Es muy versátil y se utiliza para crear aplicaciones de escritorio, juegos (con Unity), aplicaciones web y servicios.

Es ideal para desarrollar software robusto y escalable, como aplicaciones empresariales, juegos en Unity, APIs y aplicaciones web utilizando frameworks como ASP.NET.

**React**  
Es una biblioteca de JavaScript para construir interfaces de usuario (UIs). Se enfoca en el desarrollo de componentes reutilizables que actualizan de forma eficiente la vista al cambiar los datos.  
Sirve para crear aplicaciones web interactivas y dinámicas.

**Patrón DAO (Data Access Object):**   
Es un patrón de diseño que se utiliza para separar la lógica de acceso a la base de datos del resto de la aplicación. Define una interfaz para interactuar con la base de datos y la implementa para realizar operaciones como insertar, actualizar, eliminar o consultar datos.  
Mejora la organización del código, facilita el mantenimiento y permite cambiar la tecnología de la base de datos sin afectar el resto de la aplicación.

**JavaScript**  
Es un lenguaje de programación utilizado principalmente para crear contenido dinámico en páginas web. Se ejecuta en el navegador y permite interactuar con elementos de una página HTML y CSS.  
Es esencial para agregar interactividad a las páginas web. Por ejemplo, botones que realizan acciones, formularios con validaciones en tiempo real o animaciones.

**Vite.js**  
Es un entorno de desarrollo rápido para proyectos de frontend. Fue creado para reemplazar herramientas más lentas como Webpack, ofreciendo tiempos de inicio y recarga ultrarrápidos gracias a su uso de módulos de ES nativos.  
Sirve para iniciar y desarrollar proyectos web con mayor velocidad. Es ideal para aplicaciones React, Vue, o Svelte, donde necesitas un entorno de desarrollo eficiente y moderno.

1. **Referencias y recursos adicionales:** *Enlaces o referencias a otros recursos útiles, como documentación técnica relacionada, tutoriales o foros de soporte.*

BillWagner. (s/f). *Guía de C#: lenguaje administrado de .NET*. Microsoft.com. Recuperado el 26 de noviembre de 2024, de <https://learn.microsoft.com/es-es/dotnet/csharp/?WT.mc_id=dotnet-35129-website>

Getting started. (2019). *Healthcare Policy | Politiques de Santé*, *1*(1), 1–3. <https://doi.org/10.4337/9781786435767.00007>

*React*. (s/f). React.dev. Recuperado el 26 de noviembre de 2024, de <https://es.react.dev/>

1. **Herramientas de Implementación:**

* Lenguajes de programación:

Backend: C#

Frontend: Javascript

* Frameworks:

Frontend: React

Backend: ASP.NET

* APIs de terceros, etc.