VIGO-TECH SOLUTIONS

1.Justificación de la Implantación

Este proyecto surge de la necesidad estratégica de la empresa “Vigo Tech Solutions”. Esta empresa se dedica al ensamblaje y venta de hardware de alto rendimiento. En esta ocasión, la compañía se encuentra en un proceso de transformación digital, cerrando su tienda física para operar como negocio totalmente online.

Esta transición plantea las siguientes problemáticas:

- Gestión de stock de los componentes.

- Fabricación compleja, ensamblaje de equipos personalizados, esto requiere una lista de materiales BoM que descuenta automáticamente los componentes del inventario al fabricar el producto.

- Flujo unificado a la hora de abarcar la necesidad del bien hasta el pago final al proveedor y, desde que el cliente realiza el pedido hasta que se recibe el pago y se contabiliza.

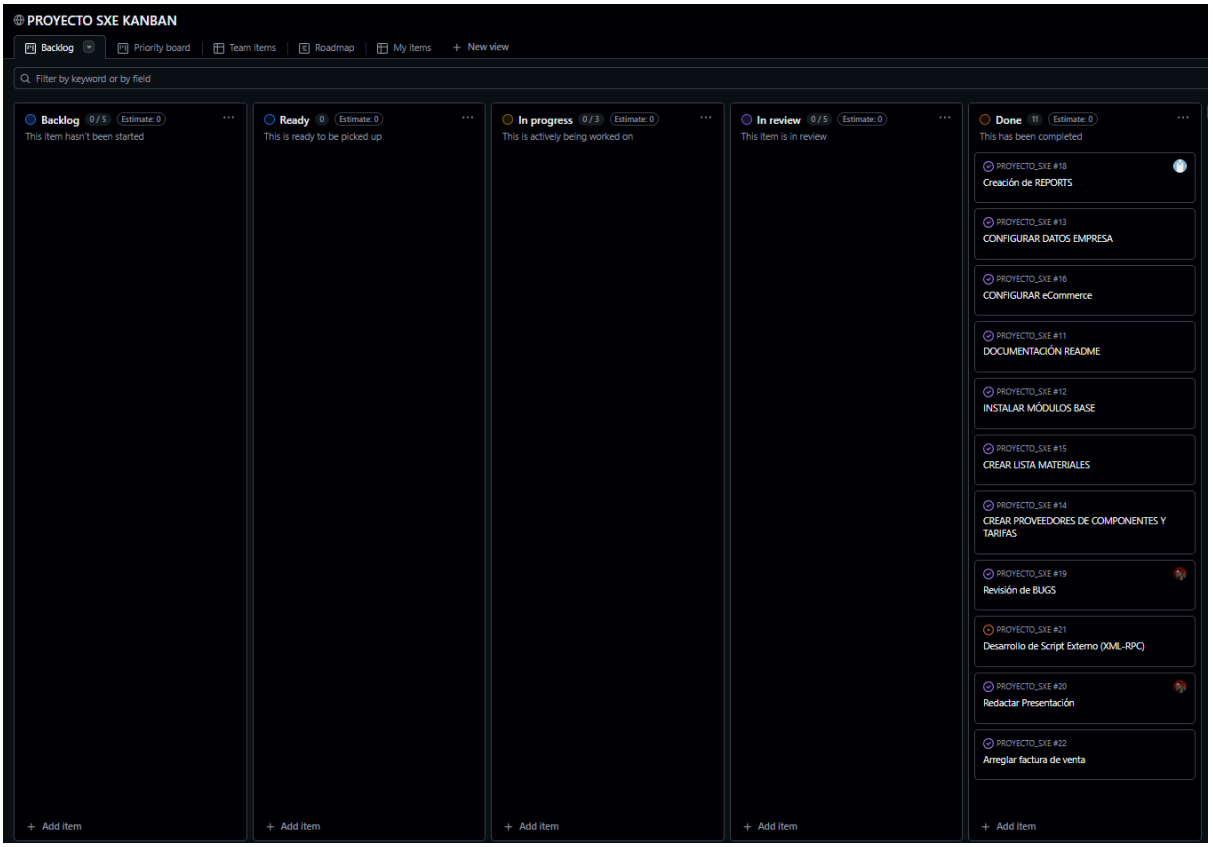
En el desarrollo de este proyecto, se ha optado por una metodología ágil basada en Kanban. La elección de Kanban se basa, sobre todo, en su flexibilidad y su capacidad para permitir el trabajo autónomo remoto y reuniones en diferentes momentos del proyecto. Esto se adapta muy bien a nuestro estilo de trabajo y a la capacidad que tenemos para reunirnos mediante Discord.

Respecto a la herramienta de gestión usada, se ha hecho uso Github projects para visualizar el flujo de trabajo, dividiendo tareas en columnas clasificadas en categorías del estilo, pendiente, en curso, pruebas, terminado...

Para repartir las diferentes tareas se ha recurrido a una reunión Kanban, donde hemos puesto en común nuestras habilidades y preferencias de cara al trabajo, siguiendo los objetivos pautados en el documento. Junto con esto, desarrollamos una serie de roles para centrarnos cada integrante en un aspecto concreto y poder trabajar más dinámicamente. Los roles escogidos fueron:

- Analista Funcional
- DevOps / Arquitecto
- Backend Developer
- QA Tester

Tras esta puesta en común empleamos la herramienta GitHub Projects, ya mencionada, para tener un control en tiempo real de los diferentes aspectos a completar. Esta distribución del trabajo nos ha permitido tener una idea general de la evolución del trabajo, dejándonos enfocarnos en los aspectos que requieran mayor prioridad o se hayan dejado de lado.



2. Justificación y Metodología

Estre trabajo tiene como finalidad principal, llevar a VigoTech de una gestión de negocio obsoleta a un entorno digital donde poder gestionar su empresa de forma efectiva y sencilla.

Debido a su carente digitalización encontramos ciertas problemáticas a las que debemos prestar especial atención.

- El stock de la web estaba totalmente desconectado del almacén real, lo que generaba problemas con pedidos además de una pérdida de tiempo del personal en gestionar estas incidencias.

- Los productos con ensamblado personalizado, como su PC GAMING BESTIA, no poseían una lista de materiales que se sincronice con su inventario. Esta desconexión requería a los técnicos revisar manualmente el inventario para evitar problemas.

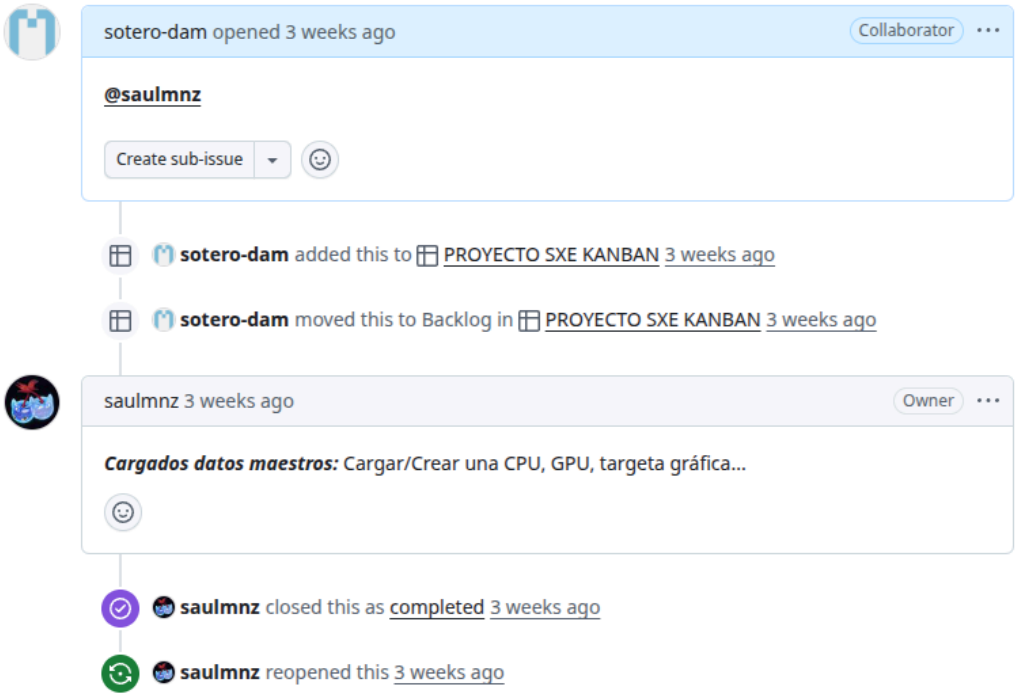
- No contaban con un sistema de flujo unificado *Procure-to-Pay* y *Order-to-Cash* en una única plataforma; haciendo extremadamente tediosos estos procesos.

Viendo la situación a la que nos enfrentamos, decidimos que implantar el ERP Odoo 18 sería la mejor solución. Ya que nos permitiría resolver los problemas centralizando la base de datos, automatizando el cálculo de costes y permitiendo gran escalabilidad gracias a Docker.

2.2. Metodología de trabajo

Como mencionamos anteriormente, Kanban ha sido la herramienta elegida, ya que flexibiliza enormemente el flujo de trabajo.

Para visualizar este flujo con la herramienta GitHub Projects, hemos empleado la vista Kanban que nos permite separar las diferentes tareas en Pendiente, En curso, Pruebas y Terminado. También hicimos uso de los issues de GitHub para poder cerrar tareas, asignarles etiquetas o añadir comentarios sobre ellas.



Complementamos esto con la gestión de versiones git, para asegurarnos de mantener en su integridad el trabajo y pudiendo retroceder nuestros cambios en cualquier momento.

En relación a los roles, estos no se han mantenido estáticos durante todo el trabajo. Cada cierto tiempo alternamos nuestra posición para asegurar que el trabajo se cumpliera debidamente además de aportar nuevas perspectivas al desarrollo individual. Las funciones que se desempeñan durante todo el proyecto fueron:

- Analista Funcional: Define los flujos de compra, venta y fabricación según los requisitos del cliente.

- DevOps / Arquitecto: Configura la infraestructura mediante Docker y Docker Compose. Backend Developer: Desarrolla el módulo personalizado en Python para modelos y XML para vistas e informes Web.

- QA Tester: Ejecuta los scripts de auditoría vía XML-RPC y valida los flujos del sistema.

3. Plan de Trabajo

3.1. Etapas del proyecto

Dividimos el proyecto en cinco fases:

ANÁLISIS

Nos enfocamos en entender las necesidades de Vigo-Tech y preparar un entorno de desarrollo adecuado. Esto incluye, diseño del modelo de datos E-R y configuración del archivo docker-compose.yml para orquestar los contenedores de Odoo y PostgreSQL. Con esto obtuvimos un entorno local desplegado accesible desde el host 8069.

CONFIGURACIÓN BASE

En este momento, lo fundamental era habilitar los módulos nativos que necesitáramos. Estos incluyen Inventario, Compras, Ventas, Facturación y Fabricación. También creamos categorías de productos, configuramos la moneda y datos de la empresa. De esta forma obtuvimos un ERP operativo

DESARROLLO DEL MÓDULO PERSONALIZADO

El objetivo principal fue adaptar Odoo a las especificidades de la venta de hardware. Para ello, realizamos el scaffolding del módulo y aplicamos herencia sobre el modelo product.template, añadiendo campos técnicos como garantía y tipo de componente. También modificamos las vistas XML para que estos campos fueran visibles en los formularios. Como resultado, logramos un módulo instalado con fichas de producto personalizadas.

IMPLEMENTACIÓN DE INFORMES Y SCRIPTS

En esta etapa buscamos cumplir con los requisitos avanzados de documentación y auditoría. Diseñamos un informe Web PDF personalizado denominado Ficha Técnica para sustituir al estándar y desarrollamos un script en Python externo. Este script utiliza la librería xmlrpc para consultar el stock crítico de forma remota. El resultado fue la capacidad de generar documentación técnica y realizar auditorías externas automatizadas.

PRUEBAS Y DOCUMENTACIÓN

Finalmente, validamos que todo el sistema funcionara correctamente. Realizamos simulaciones de ciclos completos de negocio, abarcando desde la compra y fabricación hasta la venta final. Además, redactamos el manual técnico en el archivo README para asegurar que el proyecto sea comprensible y mantenible.

3.2. Dificultades previsibles

Identificamos y mitigamos varios riesgos técnicos durante el proceso.:
Al usar Docker, evitamos conflictos en el puerto 8069 mediante el mapeo explícito en el archivo de orquestación.

También superamos errores de identificadores externos en XML revisando la nomenclatura del módulo para asegurar el correcto renderizado de vistas. Para los informes PDF, utilizamos imágenes oficiales de Odoo que ya incluyen las dependencias de wkhtmltopdf, evitando fallos de librerías.

Gestionamos la dificultad de recopilar datos técnicos de componentes unificando criterios de entrada ante la falta de estándares entre proveedores.

4. Calendario de Acciones

4.1. Diagrama de Gantt

A continuación, se presenta la distribución temporal de las tareas a lo largo de las 4 semanas de duración del proyecto:

5. Evaluación de las Acciones

5.1. Indicadores de Calidad (KPIs)

Para considerar el proyecto como exitoso, se definieron los siguientes criterios de aceptación:

- Integridad del Stock: Al fabricar un "PC Gaming Bestia", el stock de los componentes (CPU, GPU, Torre) debe disminuir automáticamente en las cantidades exactas definidas en la Lista de Materiales (BoM).
- Personalización Efectiva: Los campos personalizados (Garantía, Specs) deben persistir en la base de datos y ser visibles en los informes impresos.
- Conectividad Externa: El sistema debe permitir consultas desde scripts externos mediante la API XML-RPC, devolviendo datos en tiempo real.

5.2. Tabla de Fases de Prueba

Se han realizado las siguientes pruebas unitarias y de integración para validar el sistema:

- **T01:** Ciclo de Aprovisionamiento Creación de PO (10 unidades de cada una) + Recepción de Albarán ➔ Incremento de stock (+10 de cada una) ➔ Generación de factura en borrador.
- **T02:** Ciclo de Fabricación (MRP) Orden de Producción (PC Bestia) ➔ Consumo automático de componentes (CPU/GPU/Torre) ➔ Entrada de producto terminado en almacén.
- **T03:** Ciclo de Venta Online Confirmación de Pedido (eCommerce) ➔ Emisión del Albarán de salida ➔ Facturación y conciliación de pago.
- **T04:** Verificación de Desarrollos (QWeb) Ejecución de reporte "Ficha Técnica" ➔ Renderizado de PDF corporativo ➔ Visualización de campo personalizado "Garantía".
- **T05:** Auditoría Externa (XML-RPC) Ejecución de script consulta_stock.py ➔ Conexión remota vía UID ➔ Vemos los productos con stock crítico

**Pruebas gráficas en el Readme.md*

6. Conclusiones

6.1. Cumplimiento de necesidades

La digitalización de Vigo-Tech Solutions mediante Odoo 18 y Docker ha cumplido todos los puntos establecidos:

- Centralización: Se ha eliminado la dispersión de datos. Ventas, Almacén y Fabricación trabajan sobre una base de datos única, evitando errores de stock.
- Control del MRP: La gestión de las Listas de Materiales (BoM) permite fabricar equipos con un control exacto del coste de cada componente.
- Escalabilidad: El desarrollo del módulo propio Vigo-tech y la prueba de conexión XML-RPC demuestran que el sistema puede crecer y conectarse con apps externas sin limitaciones.

6.2. Mejoras futuras

Para una segunda fase de desarrollo, se proponen las siguientes mejoras:

- Cierre del ciclo de cobro: Integración con Stripe o PayPal para automatizar el cobro en el eCommerce.
- CRM avanzado: Reforzar el seguimiento de clientes para fidelizar a los compradores más habituales y los esporádicos.
- Backups automáticos: Garantizar la seguridad de los datos del contenedor Docker en la nube por si el servidor falla.

7. Referencias

Para la realización de este proyecto se han consultado las siguientes fuentes de información técnica y documentación oficial:

Desarrollo Técnico y Lógica de Negocio:

- **Odoo ORM Framework:** Consulta de la API para la definición de modelos, tipos de campos y decoradores de Python en el módulo vigo_tech. **https://www.odoo.com/documentation/18.0/developer/reference/backend/orm.html**
- **Vistas y Reportes QWeb:** Documentación sobre herencia XML para la personalización de formularios y el diseño de informes PDF.
- **Flujos MRP y BoM:** Guías sobre Listas de Materiales y procesos de fabricación para el ensamblaje de hardware.

Conectividad y API:

- **Protocolo XML-RPC:** Especificación del estándar para la comunicación externa y ejecución de métodos remotos en Odoo. https://www.odoo.com/documentation/18.0/developer/reference/external_api.html
- **Librería xmlrpc.client:** Documentación técnica de Python para la autenticación y consulta de inventario desde scripts externos.

Metodología y Estándares:

- **Git Flow Workflow:** Estándar utilizado para la gestión de ramas y control de versiones del equipo. https://www.atlassian.com/git/tutorials/comparing-workflows/gitflow-workflow

Uso de Inteligencia Artificial: Se ha utilizado el asistente de IA Google Gemini como herramienta de apoyo para la depuración de código y consulta de errores específicos.

Ejemplos de Prompts utilizados para resolución de errores:

- **Prompt 1: (Error de Vistas/XML):**

  "Estoy intentando sacar el PDF de la ficha técnica en Odoo 18 pero me salta este error: ValueError: External ID not found. El ID que he puesto en el XML es 'scaffold.report' y el módulo se llama 'vigo_tech'. ¿Qué me falta en el manifest o en la cabecera del XML para que lo reconozca?"

- **Prompt 2: (Error de Conexión Docker/DB):**

  "Tengo la factura del 'PC Gaming Bestia' validada, pero al darle a 'Registrar pago' para darla por cobrada me da error con el diario de banco. No me deja seleccionarlo o me dice que no está configurado. ¿Cómo asocio el pago del eCommerce para que la factura pase a estado Pagado' automáticamente?"