

SISTEMA DE GESTION DE ASIGNACION Y GESTION DE
ELEMENTOS TI

DEFINICIÓN Y CONCEPTUALIZACION

LUIS JOSÉ PICHON GOMEZ

STOCKET

Ver. 0.0.0.1

Tabla de contenido

GENERALIDADES	3
OBJETIVOS	3
ANTECEDENTES Y ESTUDIOS PREVIOS:	4
ASPECTOS TECNICOS	5
Arquitectura del sistema:	5
Tecnologías definidas:	5
Requisitos Hardware y Software:	5
Seguridad:	5
Integración con otros sistemas:	5
Rendimiento y escalabilidad:	6
Mantenimiento y actualización:	6
PUBLICO OBJETIVO	6
CALENDARIZACIÓN Y EJECUCIÓN DEL PROYECTO	6
REQUERIMIENTOS	7
FUNCIONALES	7
NO FUNCIONALES:	8
CRITERIOS DE ACEPTACIÓN:	8
CONTROL DE VERSIONES Y NOMENCLATURA	8
ROLES Y PERMISOS Y PERFILES DE USUARIO	10
DISEÑOS PREVIOS	10
Observaciones y Consideraciones Finales	12

GENERALIDADES

En el contexto empresarial actual, la gestión eficiente de los recursos tecnológicos es crucial para garantizar el éxito y la competitividad de una organización. Sin embargo, muchas empresas han experimentado dificultades durante la emergencia sanitaria del COVID-19, evidenciando las consecuencias de la negligencia, la falta de recursos y la falta de planificación en relación con los elementos tecnológicos en las oficinas de sistemas e informática.

Es importante destacar que la dependencia de estas áreas en las entidades es igual o incluso más importante que otras áreas definidas en los organigramas. La oficina de sistemas desempeña un papel fundamental al gestionar, preservar y mantener los recursos de tecnología necesarios para que la entidad pueda llevar a cabo sus procesos misionales, estratégicos, de apoyo y de control.

Contar con elementos tecnológicos confiables y adecuadamente controlados marca la diferencia entre la necesidad de realizar inversiones a gran escala, que a menudo resultan insostenibles debido a los altos costos, y la posibilidad de realizar inversiones planificadas, graduales y puntuales, lo que evita poner en riesgo financiero, operativo y operacional a la entidad.

Además, es importante considerar que la falta de inversiones en tecnología puede ocasionar problemas de competitividad y adaptabilidad a las nuevas exigencias del mercado laboral y comercial. Los recursos obsoletos pueden afectar negativamente las rutas de trabajo internas o externas de las empresas, dificultando sus operaciones y funciones.

OBJETIVOS

Los objetivos de este software de gestión son los siguientes:

- Facilitar y optimizar los procesos internos de la empresa, mejorando la eficiencia operativa y reduciendo los tiempos y costos asociados a las tareas administrativas y de gestión.
- Proporcionar una visión integral y en tiempo real de la empresa, permitiendo una toma de decisiones más informada y estratégica en todos los niveles de la organización.
- Mejorar el control y seguimiento de las actividades y recursos de la empresa, garantizando una asignación eficiente de recursos y un seguimiento efectivo de proyectos y metas.

- Centralizar la información y fomentar la colaboración entre los departamentos y empleados, promoviendo una comunicación fluida y un acceso rápido y seguro a la información relevante.
- Brindar una solución adaptable y escalable, capaz de crecer y adaptarse a medida que la empresa evolucione y se expanda, teniendo en cuenta el incremento de registros y la necesidad de gestionar mayores volúmenes de información.

ANTECEDENTES Y ESTUDIOS PREVIOS:

En el marco de evaluación y observación de las condiciones del mercado empresarial, se ha podido establecer los siguientes precedentes que denotan la importancia de la gestión y organización de activos TI. Para el registro, se añaden dos informes desarrollados a dos entidades que, a pesar de que sus objetivos misionales son diferentes, presentan falencias en la manera que se realiza dicho proceso.

1. En Anexo 1, **(Pichon, 2020) Informe de prácticas empresariales en la Contraloría general de la república - Gerencia Colegiada departamental de La Guajira**, se describen procesos de realización de gestión de equipos TI por parte de la entidad estatal, en donde se realizó "... la rectificación de inventario general tecnológico, para verificar por concepto técnico, si es necesaria la solicitud de disposición final de elementos, para su posterior donación a título gratuito, venta, oferta por comodato o destrucción..." (pág.28). La entidad contaba con más de 100 elementos TI, entre los cuales se encuentran periféricos, computadoras, portátiles y elementos de comunicación de red. Este informe cimenta un precedente en el que se estudia el impacto de lo tedioso de una tarea de asignación y disposición de elementos desde el punto de vista de una organización estatal.
2. En Anexo 2, **(Pichon 2021) Estudio de Caso - Unidad médica Radiológica del Carmen (UMRC)**, se describe un proceso de falencia similar al descrito en el informe del numeral anterior, no obstante, la falla es mucho más acentuada en vista de las practicas definidas y descritas en el mismo, donde "... el departamento de sistemas reconoce las fallas que presenta el proceso de inventario tecnológico, el cual siempre queda a disposición del profesional y no bajo lineamientos de la empresa..."

Los informes mencionados anteriormente evidencian la existencia de falencias en la gestión y organización de activos de tecnología de la información en diversas entidades. Tanto en la Contraloría General de la República como en la Unidad Médica Radiológica del Carmen, se identificaron problemas en los procesos de inventario tecnológico y disposición de elementos TI. Estas situaciones demuestran la necesidad de implementar una solución eficiente

que permita una gestión adecuada de los activos, evitando errores, optimizando recursos y garantizando un control efectivo.

Puede observar los informes completos dentro de la carpeta Anexos.

ASPECTOS TECNICOS

A continuación, se listan los aspectos técnicos del sistema:

Arquitectura del sistema:

- El sistema será construido con una arquitectura cliente-servidor, donde el Cliente será la interfaz de manipulación de datos y el servidor la infraestructura en la nube.
- Las comunicaciones serán establecidas por RestFul API

Tecnologías definidas:

- Lenguajes de Programación:
 - Para el cliente, se define JavaScript.
 - Para el servidor, Python
- Frameworks:
 - VueJS para la interfaz
 - Django para la construcción del servidor
- Base de Datos:
 - Se ha establecido PostgreSQL como sistema de almacenamiento de datos

Requisitos Hardware y Software:

Al ser un sistema web, no se requieren especificaciones técnicas de hardware especiales. No obstante, se recomiendan sistemas operativos que soporten navegadores webs basados en Chromium o Mozilla.

Seguridad:

- El acceso al sistema estará protegido mediante autenticación basada en contraseñas y roles de usuario.
- Se implementarán medidas de seguridad para proteger la integridad y confidencialidad de los datos, como el cifrado de comunicaciones y la validación de datos de entrada.

Integración con otros sistemas:

El sistema permitirá integración con otros sistemas como, por ejemplo, sistema de control de versiones locales (git) y en la nube (github), así como también integración con sistemas de contenedores (Docker) y cualquier otro sistema de despliegue que soporte las tecnologías seleccionadas para la construcción del producto.

Rendimiento y escalabilidad:

- Se utilizarán técnicas de optimización de consultas para minimizar los tiempos de respuesta del servidor.
- El sistema considerará rutinas de auto eliminado de datos con el fin de evitar conservar información que ya no sea relevante, así como también prevenir la basura y polvo digital.

Mantenimiento y actualización:

- El desarrollo del proyecto seguirá un enfoque incremental y evolutivo, basado en la entrega de versiones específicas con funcionalidades planificadas. Cada versión se someterá a pruebas y revisión antes de su lanzamiento, y se documentarán los cambios y mejoras realizadas en cada versión.
- Los prototipos serán desechables, para poder preservar la integridad de las versiones estables disponibles, no obstante, las pruebas de los prototipos que entren a fase beta no estarán disponibles para los usuarios finales de forma directa, para evitar manipulación no intencionada de los datos ya existentes.

PUBLICO OBJETIVO

Stocket ha sido definido para empresas que presenten fallas en el manejo del inventario TI, además de aquellas que deseen planificar las inversiones a esta área de forma específica. No obstante, su construcción se encuentra definida para apoyar a las MiPYMES, con el propósito de hacer que la organización de los activos de tecnología sea mucho más planificado y predecible. Hay que recordar que este sector representa más del 80% del mercado laboral del país, por lo que su nivel de conveniencia resulta muy útil a nivel organizacional y operativo, sobre todo si tenemos en cuenta que, como se ha definido en los precedentes anexados en este documento, las entidades no suelen incluir dentro de sus planes de optimización las plantas TI, lo que a la larga puede significar en inversiones significativas de capital tanto humano como económico y tecnológico.

CALENDARIZACIÓN Y EJECUCIÓN DEL PROYECTO

El proyecto será ejecutado en seis meses como mínimo para la definición de un prototipo que cumpla con las siguientes características:

1. Que cumpla con las especificaciones mínimas que lo conviertan en un MVP (Producto mínimo viable)
2. Que la documentación incluya:
 - a. Manuales de despliegue
 - b. Manuales de Uso

- c. FAQ (Preguntas comunes)
- d. Diagramas
- e. Documentación de desarrollo:
 - i. Endpoints de consumo, con argumentos y respuestas
 - ii. Reportes de mejoras y/o refactorización (si los hubiere)
 - iii. Repositorios de código (locales o en línea, independientemente del proveedor)
- 3. Que se hayan establecido los criterios para despliegue y pruebas en entornos reales
- 4. Que se encuentren disponibles reportes de pruebas automatizados y/o reportes de terceros.
- 5. Que la versión no se encuentre en fase beta, y si lo estuviere, que no exista consideración de descartarla.

Las tareas se encuentran distribuidas de la siguiente manera:

MES	ÍTEMES	DÍAS
MES 1	Identificación de las necesidades	Por establecer
	Definición de las especificaciones técnicas y tecnológicas	Por establecer
	Definición de requerimientos Funcionales y no funcionales	Por establecer
MES 2	Definición de prototipo base	Por establecer
	Diseño de Diagramas y Modelos de datos y operaciones	Por establecer
MES 3 Y 4	Desarrollo de Prototipo	Por establecer
	Desarrollo de Pruebas	Por establecer
MES 5	Definición de Diagramas de despliegue, nodos y mantenimiento	Por establecer
	Definición de manuales	Por establecer
MES 6	Definición de Reportes de sistema	Por establecer
TOTAL, DÍAS ESTIMADOS		Por establecer

Dado el tamaño del proyecto, su ejecución se estima en 130 días aproximadamente.

REQUERIMIENTOS

FUNCIONALES

El sistema debe:

1. Permitir la creación de usuarios nuevos
2. Gestionar los permisos y roles de cada usuario registrado

3. Gestionar la información de la empresa que lo ejecute con las siguientes características:
 - Datos de la empresa
 - Datos de las sedes adscritas a la empresa
 - Datos de las dependencias de la empresa
 - Datos de las dependencias anidadas en cada sede
4. Gestionar la información de los funcionarios de la entidad
5. Gestionar la información de los elementos TI de la entidad
6. Asignar los elementos TI a los funcionarios de la entidad
7. Mostrar y gestionar información sobre el mantenimiento de los elementos TI de la entidad
8. Gestionar los elementos declarados “de baja” de la entidad
9. Generar reportes PDF de las operaciones de asignación, mantenimiento, bajas y certificaciones paz y salvo.

NO FUNCIONALES:

El sistema de forma opcional puede:

- I. Enviar reportes vía correo electrónico
- II. Permitir Uso de fuentes externas (Como aplicaciones móviles, chatbots, etc.)
- III. Habilitar soporte para de accesibilidad y modo oscuro para la interfaz
- IV. Exportar datos en formato .xlsx

CRITERIOS DE ACEPTACIÓN:

- El sistema será declarado como PMV (Producto mínimo viable) si se cumple por lo menos con los requerimientos funcionales de los numerales 1 al 8 de la pila representada en esta tipificación.
- El sistema será considerado estable si, dentro de su pila de desarrollo, se incluye la documentación exigida para estos fines.

OBSERVACION: Futuros requerimientos que sean anexados deberán considerarse requerimientos nuevos que, incluirán sus respectivos diagramas, modelos, calendarización de desarrollo, pruebas y documentación, y demás ítems conexos como los presentes en este documento. Entiéndase que, por la naturaleza del producto en desarrollo, al ser incremental y evolutivo, el anexo de dichas especificaciones debe contar con su previa justificación técnica y/o tecnológica de ser necesario.

CONTROL DE VERSIONES Y NOMENCLATURA

El sistema ha adquirido por nombre principal “Sistema de gestión y asignación de elementos TI”, no obstante, se ha seleccionado el nombre clave “Stocket” para referirse a él, siendo una breve modificación a la palabra stock, que en ingles significa inventario.

Las versiones para el sistema se han decidido manejar bajo semántica de versionado, siguiendo las convenciones W.X.Y.Z - Branch donde:

- W (Versión principal): Este dígito se incrementa cuando se realizan cambios mayores en el software que implican modificaciones significativas en la funcionalidad o la arquitectura del sistema. Un incremento en este número puede indicar cambios que podrían afectar la compatibilidad con versiones anteriores o cambio en las especificaciones que lo rigen.
- X (Versión secundaria): Este dígito se incrementa cuando se realizan mejoras significativas en el software, como la adición de nuevas características o funcionalidades. Un aumento en este número generalmente indica cambios que no rompen la compatibilidad con versiones anteriores o las especificaciones en las que el sistema se encuentra en desarrollo.
- Y (Versión de parche): Este dígito se incrementa cuando se realizan correcciones de errores o se solucionan problemas menores en el software. Estas correcciones no deben introducir cambios significativos en la funcionalidad existente y no deben afectar la compatibilidad con versiones anteriores.
- Z (Versión de revisión o de compilación): Este dígito se utiliza para indicar revisiones o compilaciones adicionales del software. Suele ser utilizado para pequeñas correcciones de errores o mejoras de rendimiento que no afectan la funcionalidad principal del software.
- Branch (rama): esta porción de la versión especifica el estado del desarrollo del producto bajo las siguientes convenciones:
 - o Alpha, si la característica, modulo o función se encuentra en desarrollo.
 - o Beta, si la característica, modulo o función se encuentra en versión de demostración para revisión, comentarios y corrección de errores
 - o Estable, si la característica, modulo, función y/o producto se encuentran listos para ser llevados a producción

En ese orden de ideas, por ejemplo

Ver. 1(W).2(X).5(Y).4(Z) - Beta (Branch)

- 1 o W significa que es la versión principal No. 1, es decir que este es el código del producto estable
- 2 o X significa, que posee la segunda característica, modulo o funciones definidas para el mismo implementada
- 5 o Y significa que el producto lleva 5 correcciones internas, que pueden ser bugs o problemas de procedimientos
- 4 o Z significa que el producto lleva 5 correcciones internas de rendimiento, es decir, se ha modificado el comportamiento de la aplicación para que

ahora los tiempos de respuesta sean más cortos, entre otros tipos de correcciones

- Beta o Branch significa, que el producto se encuentra disponible para sus pruebas en un entorno de producción simulada, en esta fase se están recibiendo recomendaciones de mejoras y detectando fallos adicionales que previamente no se habían detectado.

ROLES Y PERMISOS Y PERFILES DE USUARIO

Se han identificado los siguientes roles previos, tras las observaciones hechas en los casos de estudio:

- Encargado: se define como persona que fue designada por la oficina de sistemas, este puede:
 - o Gestionar funcionarios
 - o Generar reportes Paz y Salvo
 - o Crear asignaciones y traspasos
 - o Crear historias de mantenimiento
- Administrador: Se define como persona que tiene permisos sobre todas las operaciones del sistema
- Super Usuario: Se define como persona que puede gestionar todas las operaciones de sistemas, que incluye modificación de usuarios y contraseñas de perfiles creados, permisos y roles a nivel de base de datos, entre otros.

DISEÑOS PREVIOS

Anexo a este documento, encontrará los diagramas definidos para la elaboración del producto. Entre los cuales encontrará:

1. Diagrama de Caso de uso
2. Diagrama ER
3. Diagrama de Clases
4. Diagrama de Actividades
5. Diagrama de secuencias

No obstante, a continuación, se establecerán los prototipos, con sus funciones esperadas y su potencial versión:

1. Ver. 0.1.0.0 - alpha
 - a. Configuración de servidores django y vuejs.
 - b. Configuración de base de datos
 - c. Configuración de CORS y API
 - d. Dependencias de paquetes
 - e. Configuración de entorno de pruebas para producción:
Este apartado incluye:
Configuración de contenedores Docker

- Configuración de paquetes y dependencias extras.
- f. Reporte de configuración.
- 2. Ver. 0.2.0.0 -alpha
 - a. Módulo de Usuarios
 - i. Crear y registrar usuario nuevo
 - ii. Gestionar Perfil de Usuario
 - iii. Roles y permisos de usuario
 - b. Seguridad de API
 - i. Configuración y ciclo de vida de tokens
 - c. Pruebas unitarias de módulo de Usuarios
 - d. Reportes de pruebas y configuraciones.
 - e. Mocks de prueba y manuales de desarrollo y uso
- 3. Ver. 0.3.0.0 - alpha
 - a. Módulo de Personas:
 - i. Gestión de funcionarios
 - ii. Gestión de Proveedores
 - b. Gestión de Empresa
 - i. Gestión de datos de empresa
 - ii. gestión de datos de sedes
 - iii. gestión de datos de dependencias
 - c. Pruebas unitarias de módulos de usuarios
 - d. Reportes de pruebas y configuraciones
 - e. Mocks de pruebas y manuales de desarrollo y uso
- 4. Ver. 0.4.0.0 - alpha
 - a. Módulo de Inventario
 - i. Gestión de referencias
 - ii. Gestión de tipos
 - iii. gestión de Elementos
 - b. Pruebas unitarias de módulos de inventario
 - c. Reportes de pruebas y configuraciones
 - d. Mocks de pruebas y manuales de desarrollo y uso
- 5. Ver. 0.5.0.0 - alpha
 - a. Módulo de Mantenimiento
 - i. gestión de mantenimientos
 - b. Módulo de Bajas
 - i. gestión de baja.
 - ii. Rutinas de borrado automático de registros
 - c. Módulo de Asignaciones
 - i. gestión de asignaciones
 - ii. gestión de traspasos
 - d. Módulo de certificaciones

- i. Reporte de certificación de Paz y salvo de sistemas
- e. Pruebas unitarias de módulos de Mantenimiento, Bajas y Asignaciones
- f. Reportes de Pruebas y configuraciones
- g. Mocks de Pruebas y manuales de desarrollo
- 6. Ver. 0.6.0.0 - beta
 - a. Cambio a Fase Beta
 - i. Reportes de errores generales
 - ii. Reporte de correcciones y características adicionales
 - iii. Pruebas en entorno de producción
 - b. Revisión de documentación
 - c. Creación de diagramas de colaboración, despliegue y nodos
 - d. Escritura de FAQ y manuales
 - e. Propuesta de selección de tecnologías para despliegue
 - f. Configuraciones para despliegue en producción adicionales

7. Ver. 1.0.0.0 -stable

Despliegue en entorno de producción

8. Ver 1.1.0.0 - alpha

- a. Reporte de análisis de nuevas funcionalidades
- b. Diseños de adaptación de nuevas funciones
- c. Calendarización de funciones nuevas
- d. Configuración de entorno para desarrollo de versión nueva

OBSERVACIONES Y CONSIDERACIONES FINALES

Tras la elaboración de este documento, se han establecido los lineamientos y el flujo de trabajo para el desarrollo de la versión 1 de Stocket. A continuación, se detallan las acciones a seguir:

- Esquemas de base de datos en Django: Se procederá a escribir los esquemas de base de datos utilizando Django como marco de trabajo.
- Pruebas unitarias: Antes de iniciar el proceso de desarrollo del API, se llevarán a cabo pruebas unitarias para garantizar la calidad del código.
- Liberación del API: Una vez que las pruebas resulten satisfactorias, el API estará listo para su consumo y se procederá a su liberación.
- Consumo del API: Se seguirán los lineamientos establecidos en el backend en cuanto a los argumentos y respuestas para consumir el API de manera correcta.

- Documentación: Se elaborará un manual de uso que incluirá información sobre los endpoints y los tipos de datos utilizados en el sistema.
- Control de desarrollo: Se utilizará el enfoque de "conventional commits" para llevar un mejor control del desarrollo y mantener una comunicación clara en el equipo.

Una vez revisados y aprobados los esquemas y condiciones de desarrollo, se iniciará la codificación del producto de manera inmediata. Es importante tener en cuenta que el entregable principal se enfocará en la versión 1.0.0.0 del software, ya sea que se mantenga en fase beta o se ponga en producción.

Con esta planificación y enfoque, todo se encuentra preparado para avanzar en el desarrollo del proyecto, cumpliendo con los estándares establecidos y entregando una versión inicial de Stocket.