

SISTEMA DE GESTION DE ASIGNACION Y GESTION DE
ELEMENTOS TI

DEFINICIÓN Y CONCEPTUALIZACION

LUIS JOSÉ PICHON GOMEZ

STOCKET

Ver. 2.0.0.0

Tabla de contenido

Generalidades	3
Objetivos.....	3
General.....	3
Específicos	3
ASPECTOS TECNICOS	3
ARQUITECTURA DEL SISTEMA:	3
TECNOLOGIAS DEFINIDAS:	4
<i>OBSERVACIONES</i>	4

Generalidades

El presente documento, es establecido para definir las especificaciones y planes de mejora para el producto denominado “SISTEMA DE GESTION DE ASIGNACION Y GESTION DE ELEMENTOS TI - Stocket” (en adelante, el **producto, software, aplicativo, sistema, stocket o Stocket**)

Las especificaciones y requerimientos seguirán los lineamientos de la versión anterior, no obstante, se plantearán a continuación nuevas especificaciones para la ejecución de las tareas definidas para el mismo.

Objetivos

General

Diseñar y desarrollar una versión más avanzada de Stocket, donde los potenciales clientes puedan obtener más funciones útiles para el desarrollo de sus actividades laborales

Específicos

- Establecer los criterios y requerimientos para la definición del prototipo nuevo, correspondiente a la versión a desarrollar.
- Definir los diseños y prototipos conceptuales (si los hubiere) en los que se pueda establecer cómo y de qué manera deben implementarse los cambios
- Desarrollar el prototipo correspondiente a la versión, siguiendo los lineamientos establecidos para la ejecución del proyecto
- Actualizar manuales y documentación (de ser necesario) para adaptarse a las nuevas características del producto.
- Desplegar el sistema en un entorno web para sus pruebas y consideraciones finales.

ASPECTOS TECNICOS

A continuación, se listan los aspectos técnicos para tener en cuenta para la versión en desarrollo:

ARQUITECTURA DEL SISTEMA:

Se procederá con la arquitectura definida en la versión anterior la cual:

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

TECNOLOGIAS DEFINIDAS:

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

OBSERVACIONES

Se ha tomado la decisión de reemplazar el framework Vue.js por ReactJS. Esta observación tiene como objetivo analizar y justificar los motivos detrás de esta transición.

Motivación Estratégica:

Se observa que la transición se basa en una motivación estratégica, donde se busca alinear la aplicación con las tendencias tecnológicas y las demandas cambiantes de los usuarios.

Tamaño y Vitalidad de la Comunidad:

Se destaca que React.js cuenta con una comunidad significativamente grande y activa, lo que sugiere una ventaja en términos de acceso a recursos y soporte continuo.

Flexibilidad y Escalabilidad:

Se ha observado que React.js ofrece una mayor flexibilidad y escalabilidad, lo que es relevante para la aplicación en consideración, que planea expandir sus características y funcionalidades en el futuro.

Rendimiento:

Se ha constatado que React.js demuestra un rendimiento notable en aplicaciones con un alto número de componentes y actualizaciones en tiempo real, lo que es congruente con los objetivos de la aplicación.

Capacidades de Pruebas:

Se ha señalado la capacidad de React.js para pruebas, lo que sugiere un enfoque en la calidad y la fiabilidad de la aplicación.

Integración con Tecnologías Adicionales:

Se ha notado que React.js se integra de manera efectiva con otras tecnologías, lo que facilitará la incorporación de nuevas características y la adopción de tecnologías emergentes.

Respaldo y Soporte a Largo Plazo:

Se ha subrayado el respaldo sólido de React.js por parte de Facebook y su compromiso con el soporte a largo plazo, lo que brinda confianza en la sostenibilidad de la tecnología.

Tendencias del Mercado:

Se ha observado que React.js goza de una amplia adopción en la industria, lo que garantiza que los desarrolladores adquieran habilidades altamente demandadas en el mercado laboral.

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.
- Se implementará un sistema de control de sesión basado en json web token (JWT), donde las sesiones se podrán finalizar de forma automática asegurando la seguridad del sistema y previendo entradas no autorizadas.

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, como lo es vercel que fue el hosting seleccionado para el despliegue en producción de testing.

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 tras cada despliegue.
- 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 y preservar la estabilidad de las funciones ya desplegadas. En este orden de ideas, es que se declara descartar el cliente de Vuejs y se selecciona React.js, tal y como se define en las OBSERVACIONES listadas de las tecnologías definidas.

CALENDARIZACIÓN Y EJECUCIÓN DEL PROYECTO

El proyecto será ejecutado en seis meses como mínimo para la definición del prototipo correspondiente a la versión 2 del sistema, cumpliendo con las siguientes características:

1. Que los cambios realizados no alteren a la versión previa (Versión 1) definida como el MVP (Producto mínimo viable) desplegado.
2. Que la documentación incluya:
 - a. Manuales de despliegue (si los hubiere)
 - b. Manuales de Uso (si los hubiere)
 - c. FAQ (si las hubiere)
 - d. Diagramas y prototipos conceptuales
 - 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:

Meses	Ítems	días
1	Identificación de las necesidades	Por establecer
	Definición de las especificaciones técnicas y tecnológicas	Por establecer
	Definición de requerimientos Funcionales y no funcionales nuevos	Por establecer
2	Definición de prototipo base	Por establecer
	Diseño de Diagramas y Modelos de datos y operaciones	Por establecer
3 ,4 y 5	Refactorización de procedimientos del servidor	Por establecer
	Desarrollo de Prototipo	Por establecer
	Desarrollo de Pruebas	Por establecer
6	Definición de Diagramas de despliegue, nodos y mantenimiento	Por establecer
	Definición de manuales	Por establecer
7	Definición de Reportes de sistema	Por establecer
<i>Total, días estimados</i>		Por establecer

Dado el tamaño del proyecto, su ejecución se estima en 200 días aproximadamente. Considerar estos valores como referencia de fecha de entrega del nuevo MVP.

REQUERIMIENTOS

FUNCIONALES

El sistema debe:

- 1- Generar códigos QR para fácil lectura de registros sin acceder a la aplicación.

Dichos códigos deben permitir al usuario redirigirse a la plataforma y ver datos del elemento.

- 2- Integración con bots de terceros (ya sea de Telegram, signal o WhatsApp)

NO FUNCIONALES:

El sistema de forma opcional puede:

- I. Enviar notificaciones vía correo electrónico a los funcionarios cuando:
 - a. Se le realice alguna asignación o traspaso.
 - b. Se le genere un reporte de paz y salvo en el sistema recibiendo una copia.
- 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

FLUJO DE TRABAJO IDENTIFICADO

El flujo de desarrollo de las tareas para el producto se ha identificado de la siguiente manera:

- I. Refactorización de procedimientos del servidor y creación y mejora de vistas de cliente.
- II. Creación de códigos QR para su consulta
- III. Lectura/escritura de archivos xlsx o csv para su procesamiento externo

CRITERIOS DE ACEPTACIÓN:

- El sistema será declarado como PMV (Producto mínimo viable) si se cumple por lo menos con el requerimiento funcional que corresponde al numeral 1 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:
 - Alpha, si la característica, modulo o función se encuentra en desarrollo.
 - Beta, si la característica, modulo o función se encuentra en versión de demostración para revisión, comentarios y corrección de errores
 - 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 respetará la identificación de roles y permisos establecidos en el documento de Definición y conceptualización correspondiente a la versión 1 del sistema.

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. 2.7.0.0 - Alpha
 - a. Configuración y refactorización de servidores Django y ReactJS.
 - b. Configuración de base de datos (si fuere necesario)
 - c. Configuración de CORS y API (si fuere necesario)
 - 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. 2.8.0.0 -Alpha
 - a. Generación de códigos QR:
 - i. Generación de códigos con su respectiva nomenclatura e información de la referencia del dispositivo
 - b. Reporte de pruebas y configuración
 - c. Manuales de uso
3. Ver. 2.9.0.0 - 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. Actualización de FAQ y manuales
- e. Configuraciones para despliegue en producción adicionales

4. Ver. 2.9.0.0 -stable

Despliegue en entorno de producción

5. Ver 2.10.0.0 - 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 2.9 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 2.9.0.0 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.