**PROYECTO FIZZFLEET**

Descripción general:

Una empresa comercializadora y distribuidora de bebidas efervescentes SodaPlus requiere un sistema que posea las siguientes funciones generales:

* control de pedidos.
* seguimiento de locaciones y rutas de transporte.
* administración de inventarios.
* un host para servicio web.
* servidor central que provee a múltiples clientes.
* base de datos

Con este software busca resolver los desafíos de logística y gestión de inventario y pedidos, donde habrá un control lo más cercano al tiempo real donde, puede tenerse mejor seguimiento de las actividades de la compañía. Y esto nos ortigará una mejor planificación de la producción y la distribución, asegurando que los clientes reciban sus pedidos a tiempo y en las cantidades correctas.

El seguimiento de locaciones y rutas de transporte permitirá optimizar las rutas de transporte, reducir costos y mejorar la eficiencia en la entrega de los productos, manteniendo un registro preciso de la cantidad de producto disponible, lo que ayudará a evitar la falta de stock y minimizar su desperdicio.

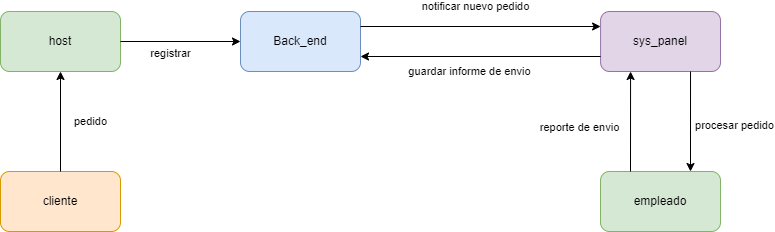
Viabilidad y alcance:

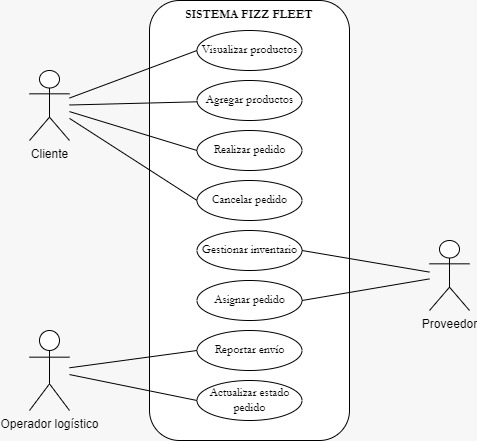
La implementación de un servidor principal que provee a distintas plataformas, nos ayudará a control el acceso y la distribución de la información, asegurando que todas las plataformas tengan su debida respuesta por parte del servidor dependiendo de su nivel de autorización. Con la intención de poder centralizar la administración de la seguridad, lo que es fundamental para garantizar la protección de los datos y sistemas. Esto incluye la gestión de usuarios, la autenticación y el control de acceso a los recursos. Mientras tanto el hosting administra la página web y la aplicación móvil, y gestiona las solicitudes de los usuarios.

Además, el software también brindará las funcionalidades de generación de informes y estadísticas para evaluar el desempeño de la empresa, detectar oportunidades de mejora y tomar decisiones informadas en tiempo y forma. Esta información se actualizará automáticamente en el servidor, lo cual asegura la integridad y la disponibilidad de los datos.

Esquema de procesos:

El sistema ofrece un catálogo de productos al usuario donde puede pedirlos mediante el sitio web o la app móvil, esto se hace con un listado de productos seleccionados (carrito de compra).

Al registrase el pedido del cliente, sé notifica al panel de control del empleado, y este empieza el proceso de envío. Crea un informe con los datos del encargo. Mientras el servidor principal gestiona los registros y cambios en el inventario de productos.

****Casos de uso:

|  |  |  |
| --- | --- | --- |
| Visualizar producto | | |
| Versión | 1.0 (17/02/2023) | |
| Actor | Usuario | |
| Descripción | Sé describe el siguiente caso donde el servicio web (o móvil) que provee al usuario un catálogo de productos y guarda en lista | |
| Secuencia normal | Paso | Acción |
| 1 | Muestra una lista de productos y sus diferentes presentaciones |
| 2 | El usuario se desplaza atreves de pestañas enumeradas o mediante scroll vertical |
| 3 | Hace click en un producto, muestra una descripción más detallada |

|  |  |  |
| --- | --- | --- |
| Agregar producto | | |
| Versión | 1.0 (17/02/2023) | |
| Actor | Usuario | |
| Dependencias | * Visualizar producto | |
| Descripción | Se añade uno o más producto al carrito de compras del usuario | |
| Pre Condición | El usuario se ha registrado y el servidor cuenta con su información básica y de locación | |
| Secuencia normal | Paso | Acción |
| 1 | Selecciona un producto de los visualizados |
| 2 | Escoge la cantidad del producto |
| 3 | Da click en agregar al “carrito de compra” |
| Excepción | El producto agregado no tiene disponibilidad | |
|  | 4 | Se niega la posibilidad de agregar el actual producto al carrito |
|  |  |  |
| Realizar pedido | | |
| Versión | 1.0 (17/02/2023) | |
| Actor | Usuario | |
| Dependencias | * Agregar producto | |
| Descripción | Tras haber añadidos uno o más productos al carrito, se realiza el proceso de petición de envío | |
| Pre Condición | El usuario se ha registrado y cuenta con productos añadidos en su carrito de compra | |
| Secuencia normal | Paso | Acción |
| 1 | Ingresa los datos de envío |
| 2 | Confirma la compra del producto |
| Post Condición | Se notifica la validación del pedido y se muestra su factura | |
|  | 2 | Se notifica que el elemento actualmente no puede ser cancelado |

|  |  |  |
| --- | --- | --- |
| Cancelar pedido | | |
| Versión | 1.0 (17/02/2023) | |
| Actor | Usuario | |
| Dependencias | * Realizar pedido | |
| Descripción | Sé describe el siguiente caso donde el servicio web (o móvil) que provee al usuario un catálogo de productos y guarda en lista | |
| Pre Condición | El usuario se ha registrado y cuenta con pedidos pendientes | |
| Secuencia normal | Paso | Acción |
| 1 | Visualiza los productos y pedidos registrados en su cuenta |
| 2 | Selecciona un elemento |
| 3 | Elimina elemento seleccionado |
| Excepción 1 | La vista de elementos no ha sido actualizada | |
|  | 4 | Se notifica que el elemento no existe |
| Excepción 2 | El pedido ya está en estado de envío | |

|  |  |  |
| --- | --- | --- |
| Gestionar inventario | | |
| Versión | 1.1 (24/02/2023) | |
| Actor | Proveedor | |
| Descripción | El proveedor modifica en el sistema los productos y cambia sus categorías | |
| Secuencia normal | Paso | Acción |
| 1 | Corrige el stock de productos |
| 3 | Guarda los datos en el sistema |
| Secuencia nuevo producto | 1 | Ingresa los productos |
| 2 | Señala sus categorías |
| 3 | Guarda los datos en el sistema |
| Excepción | Un producto es agregado y coincide con otro ya existente | |
|  | 3 | No guarda los datos |

|  |  |  |
| --- | --- | --- |
| Asignar pedido | | |
| Versión | 1.1 (24/02/2023) | |
| Actor | Proveedor | |
| Dependencias | * Gestionar inventario | |
| Descripción | Sistema al notificar | |
| Pre Condición | Existe stock de productos en el sistema | |
| Secuencia normal | Paso | Acción |
| 1 | Verifica el stock en el inventario |
| 2 | Actualizar el stock en el inventario |
| 3 | Confirma el proceso de envío |
| Post Condición | Notifica al sistema del estado del pedido | |
| Excepción | Stock de productos insuficiente | |
|  | 3 | Cancela el pedido |
|  | 4 | Notifica al sistema de falta de stock |

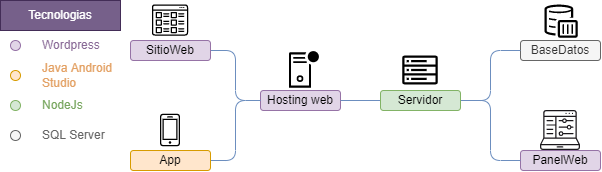
|  |  |  |
| --- | --- | --- |
| Reportar envío | | |
| Versión | 1.1 (24/02/2023) | |
| Actor | Operador logístico | |
| Descripción | Sé notifica al panel web del operador logístico y comienza el informe con los datos de transporte del envío | |
| Pre Condición | El proveedor ha confirmado el proceso de envío | |
| Secuencia normal | Paso | Acción |
| 1 | Inicializa el proceso para exportar el pedido |
| 2 | Asigna el vehículo y confirma su capacidad para el encargo |
| 3 | Indica a el panel web que él envío ha salido de la zona de transporte |
| Excepción | El vehículo no cumple el espacio requerido para el envío | |
|  | 2 | Deja en estado de espera el envío |
| Post Condición | Nuevos vehículos están disponibles (Procede con los pasos 2 y 3 de la secuencia normal) | |

|  |  |  |
| --- | --- | --- |
| Actualizar estado pedido | | |
| Versión | 1.1 (24/02/2023) | |
| Actor | Operador logístico | |
| Dependencias | * Reportar envío | |
| Descripción | Sé modifica el informe al inicio del proceso de envío con los nuevos datos | |
| Pre Condición | El vehículo ha entregado el paquete y regresa a la zona de transporte | |
| Secuencia normal | Paso | Acción |
| 1 | Ingresa los datos en el panel web |
| 2 | Guarda los cambios |
|  | 3 | Notifica al proveedor del estado final del envío |
| Excepción | El paquete no ha sido entregado | |
|  | 3 | Notifica al proveedor del estado final del envío |

Arquitectura:

Sé optó por una estructura apropiada para la línea de procesos planteada y los casos de uso, que sería la Arquitectura cliente-servidor, esta arquitectura se basa en la separación del sistema en dos componentes principales: el cliente y el servidor. El cliente es responsable de presentar la información al usuario y de manejar la interacción con éste, mientras que el servidor se encarga del procesamiento y almacenamiento de la información. En este caso, el cliente (el sitio web o la app móvil) sería la interfaz de usuario que utilizan los clientes de la empresa para hacer sus pedidos, mientras que el servidor incluiría la base de datos, gestión de pedidos y la web local para empleados.

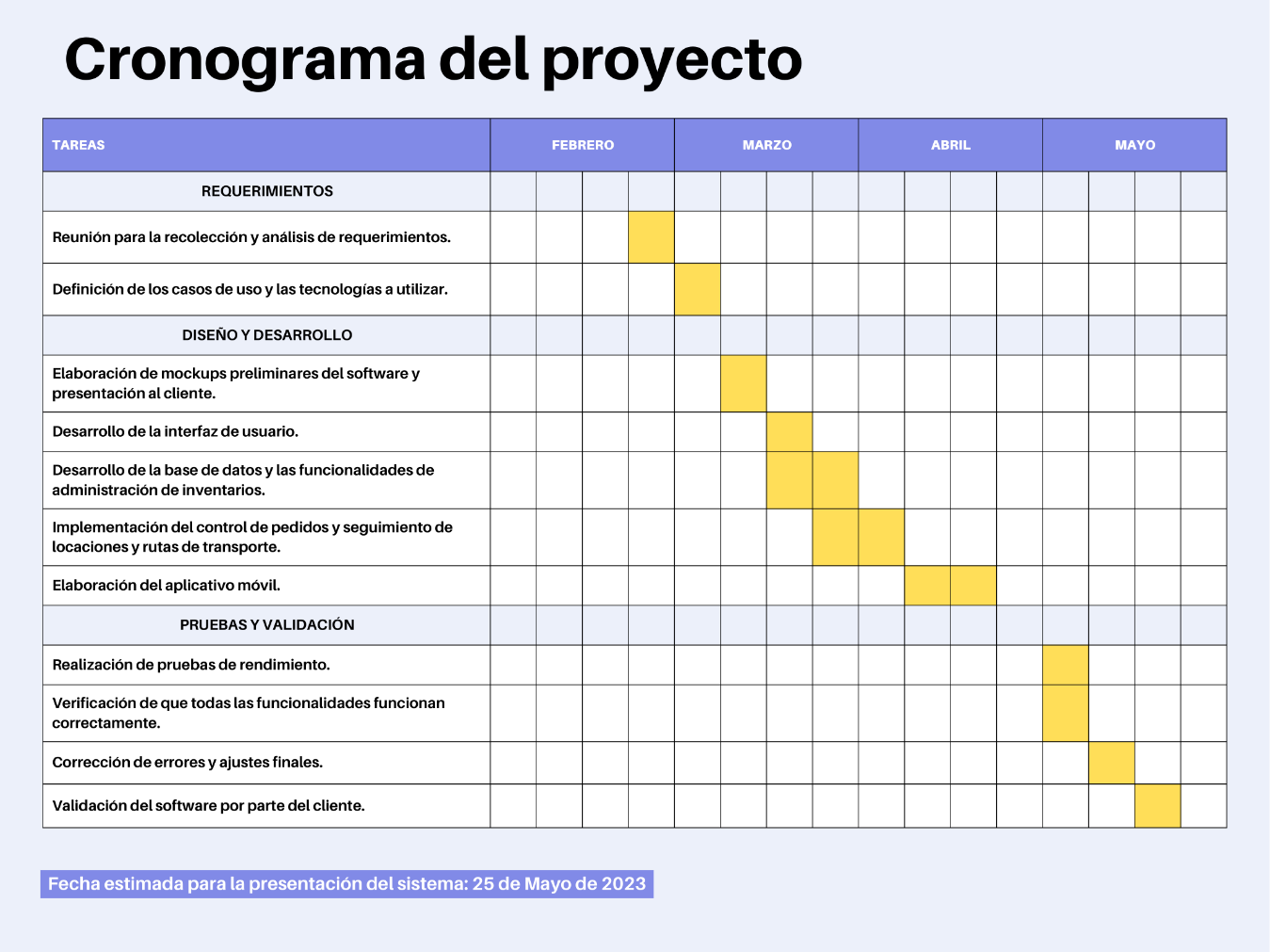
Los clientes se comunicarían con el servidor mediante datos en formato JSON y respuestas HTTP.



Mientras cada cliente web utilizaría la Arquitectura basada en micro-servicios que está enfocada en dividir el sistema en servicios independientes (que en este contexto serian funcionabilidades), que se comunican entre sí a través de una interfaz bien definida el cual se trabaja mediante la API de wordpress.

Metodología de desarrollo:

En este caso, se podría utilizar la metodología Ágil de Desarrollo de Software. Esta metodología se enfoca en la entrega temprana y continua de software funcional y en la adaptación a los cambios de los requisitos del proyecto. Se basa en ciclos iterativos de desarrollo, llamados “sprints”, en los que el equipo trabaja en pequeñas partes del proyecto y realiza entregas al final de cada sprint. Esto permite un enfoque iterativo e incremental para el desarrollo de cada componente del sistema, lo que facilitaría la integración de los mismos. Además, el enfoque en la entrega continua da la oportunidad testing de las funcionalidades del software, lo que aumentaría la satisfacción del cliente y reduciría los riesgos de los cambios en los requisitos a medida que se avanza en el proyecto.



En la sección de desarrollo, cada campo marcado en amarillo representa un “sprint” de cada componente y funcionabilidad.