Especialización en Ingeniería de Software

Estudio de caso

(Diseñar una arquitectura de software que permita integrar el proceso de reubicación de mercancía en las bodegas con el software ERP de Continente S.A.S)

Estudiantes:

Alexis Cadrasco García
Elías Ibáñez
Johann Gutiérrez

Barranquilla, Atlántico 09 de diciembre de 2020

Contenido

1. Introducción 4
2. Justificación 5
3. Planteamiento del problema 6
4. Objetivos 7
   4.1 Objetivo general 7
   4.2 Objetivos específicos 7
5. Revisión de la literatura 8
   5.1 Bases teóricas 8
      API REST 8
      Ventajas que ofrece REST para el desarrollo 9
      HTTP 9
      JSON (JavaScript Object Notation - Notación de Objetos de JavaScript) 10
      API (interfaz de programación de aplicaciones) 11
      CLOUD COMPUTING 11
      INFRAESTRUCTURA COMO SERVICIO (IaaS) 11
      PLATAFORMA COMO SERVICIO (PaaS) 11
      SOFTWARE COMO SERVICIO (SaaS) 12
      SERVERLESS (Computación sin servidor) 12
      MICROSERVICIOS 12
6. Metodología 13
7. Desarrollo del proyecto 14
   7.1 Modelado de proceso actual de reubicación de mercancía 14
      7.1.1 Modelado de proceso. Reubicación por IRM (Ingreso y recibo de mercancía, compra) (Actual) 14
      7.1.2 Modelado de proceso. Reubicación por devolución de mercancía NEB (Nota de entrada a bodega) (Actual) 15
   7.2 Requerimientos no funcionales. 16
   7.3 Propuesta solución a implementar 17
      7.3.1 Diseño de la arquitectura EJA 17
      7.3.2 Descripción de los componentes de la arquitectura EJA 18
      7.3.3 Explicar cómo las aplicaciones del lado del cliente se comunican con los servicios expuestos. 19

7.3.4 Describir los microservicios que van a servir para mejorar el proceso de reubicación de mercancía. 20
7.3.5 Sugerir prototipos de las interfaces de usuario, mediante unos mockups para un cliente móvil. 21
7.3.6 Sugerencias de especificaciones técnicas de la terminal móvil 24
7.3.7 Modelado de proceso. Reubicación por IRM (Ingreso y recibo de mercancía, compra) (Con arquitectura propuesta). 25
7.3.8 Modelado de proceso. Reubicación por devolución de mercancía NEB (Nota de entrada a bodega) (Con arquitectura propuesta) 26
8. Resultados obtenidos y discusión 27
9. Conclusiones 28
10. Referencias bibliográficas 29

1. Introducción

"La logística es la encargada de organizar y gestionar los procesos que hacen posible la circulación de los productos desde su estado primario en poder de los proveedores hasta convertirse en el producto final en manos de los clientes". (IMF Business School, 2019, p.3).

"Entre sus principales funciones se encuentran la planificación y el control del flujo de productos incluyendo el abastecimiento, la fabricación, el almacenamiento y la distribución, con una visión integral basada en la efectividad, la eficiencia y la eficacia". (IMF Business School, 2019, p.4)

"Sus objetivos primordiales son el cumplimiento de las expectativas de los clientes y la garantía de una gestión eficaz que permita brindar el servicio en condiciones cada vez mejoradas en cuanto a calidad y costes". (IMF Business School, 2019, p.5)

Continente S.A.S es una empresa transaccional que comercializa y posiciona marcas, es caracterizada por una sólida relación con proveedores y clientes.

Ubicada en el sector comercial, la empresa cuenta con varias bodegas donde almacena sus diferentes líneas o marcas de productos. Para poder continuar y mantener esta fuerte relación con clientes y proveedores, se hace necesario la aplicación de las TI en sus procesos y mucho más aún en aquellos aspectos claves donde se generan reprocesos y se crece mucho más el margen de error, afectando directa e indirectamente la gestión logística.

A continuación, se planteará una propuesta tecnológica que brindara una solución a uno de los procesos claves de la logística en Continente, como lo es la reubicación de mercancía.

2. Justificación

La empresa Continente S.A.S tiene varias bodegas en las cuales se almacena la mercancía, desde estas bodegas se realizan los despachos de los pedidos a los clientes, convirtiendo este proceso en uno de los más importantes de la empresa, ya que, dependiendo de la oportuna entrega, la empresa adquiere eficiencia, eficacia y confianza ante los clientes.

Para que este proceso fluya sin muchos inconvenientes, es importante tener organizada y clasificada la mercancía en la bodega, para esto, la compañía tiene ubicaciones en las estanterías donde reposan los artículos, estas ubicaciones están debidamente registradas en el software ERP, donde se lleva registro de Kardex, indicando en qué ubicación está cada artículo y que cantidad de existencia tiene en dicha ubicación.

Debido a los diversos procesos que se ejecutan en las bodegas, por ejemplo: devoluciones de mercancías, ingresos de mercancía por compras, traslados de mercancía entre bodegas, ajustes de artículos por avería, reubicaciones de artículos, en fin, se vuelve muy complicado tener que reubicar toda esta mercancía impactada por dichos procesos, ya que el software ejecuta estos procesos sin ningún inconveniente, pero alojando la mercancía en ubicaciones transitorias, donde posteriormente el personal logístico deberá clasificar y reubicar la mercancía, tanto físicamente como en el software ERP.

Se desea proponer un diseño de una arquitectura de software que permita integrar el proceso de reubicación de mercancía en las bodegas con el software ERP de Continente S.A.S.

Esta solución traería varios beneficios a la compañía, ya que se reduciría notablemente el tiempo que toma el personal logístico en registrar los movimientos de reubicación en el software ERP, el proceso de despacho fluiría notablemente ya que las inconsistencias al momento de buscar la mercancía en las ubicaciones se reducirían considerablemente.

3. Planteamiento del problema

Continente S.A.S tiene varias bodegas, estas bodegas están compuestas por varias ubicaciones, cada una de las cuales tiene su código de ubicación y está referenciada en el software ERP. Estas indican la existencia como tal qué hay de artículos. Al realizar las reubicaciones de mercancía físicamente en las bodegas (pasar un artículo de una ubicación a otra), se vuelve tedioso tener que anotar manualmente lo que se reubico, para luego ir al software ERP más exactamente al módulo de logística y radicar dicho movimiento (o viceversa). Esto genera un reproceso, y aumenta el margen de error, al momento de confrontar la mercancía ubicada físicamente vs la indicada en el software ERP. Dichos problemas de confrontación se ven impactados fuertemente en dos procesos logísticos como lo son:

- **Toma física de inventario:** Al momento de realizar los inventarios, que son 2 veces al año (semestralmente). Para poder llevar a cabo dicho proceso, el equipo de auditoría requiere de un tiempo considerable para dejar la mercancía ubicada consistentemente con relación al software ERP, haciendo que el inventario lleve mucho más tiempo y requiere de más recursos de personal para que se pueda realizar.
- **Alistamiento de mercancía para su despacho (Picking):** El software ERP genera un picking (Documento de alistamiento de mercancía) que le indica al alistador en que ubicación debe ir a buscar la mercancía para su despacho. Muchas veces el alistador se encuentra con que lo indicado en el documento no concuerda con lo que realmente está en la ubicación, esto obviamente genera una pérdida de tiempo valiosa que afecta los tiempos de despachos.

4. Objetivos

4.1 Objetivo general

Diseñar una arquitectura de software que permita integrar el proceso de reubicación de mercancía en las bodegas con el software ERP de Continente S.A.S.

4.2 Objetivos específicos

- Esquematizar la arquitectura y los componentes que esta va a contener.
- Describir los componentes de la arquitectura que se van a diseñar.
- Explicar cómo las aplicaciones del lado del cliente se comunican con los servicios expuestos.
- Describir los microservicios que van a servir para mejorar el proceso de reubicación de mercancía.
- Sugerir prototipos de las interfaces de usuario, mediante unos mockups para un cliente móvil.

5. Revisión de la literatura

5.1 Bases teóricas

API REST

"A partir del 2000, REST cambió por completo la ingeniería de software. Este nuevo enfoque de desarrollo de proyectos y servicios web fue definido por Roy Fielding, el padre de la especificación HTTP." (BBVA API_Market, 2016, p.1).

"En la actualidad no existe proyecto o aplicación que no disponga de una API REST para la creación de servicios profesionales a partir de ese software" (BBVA API_Market, 2016, p.2).

> REST es el estándar más lógico, eficiente y habitual en la creación de APIs para servicios de Internet; Podemos definir REST, como cualquier interfaz entre sistemas que use HTTP para obtener datos o generar operaciones sobre esos datos en todos los formatos posibles, como XML y JSON. Es una alternativa en auge a otros protocolos estándar de intercambio de datos como SOAP, que disponen de una gran capacidad pero también mucha complejidad. A veces es preferible una solución más sencilla de manipulación de datos como REST. (BBVA API_Market, 2016, p.3)
>
> Dentro de las principales características de REST, encontramos:

- ❏ Protocolo cliente-servidor sin estado: Cada petición HTTP contiene toda la información necesaria para ejecutarla, lo que permite que ni cliente ni servidor necesiten recordar ningún estado previo para satisfacerla. Aunque esto es así, algunas aplicaciones HTTP incorporan memoria caché. Se configura lo que se conoce como protocolo cliente-caché-servidor sin estado: existe la posibilidad de definir algunas respuestas a peticiones HTTP concretas como cacheables, con el objetivo de que el cliente pueda ejecutar en un futuro la misma respuesta para peticiones idénticas. De todas formas, que exista la posibilidad no significa que sea lo más recomendable. (BBVA API_Market, 2016,p.4)
- ❏ Las operaciones más importantes relacionadas con los datos en cualquier sistema REST y la especificación HTTP son cuatro: POST (crear), GET (leer y consultar), PUT (editar) y DELETE (eliminar). (BBVA API_Market, 2016, p.5)
- ❏ Los objetos en REST siempre se manipulan a partir de la URI. Es la URI y ningún otro elemento el identificador único de cada recurso de ese sistema REST. La URI nos facilita acceder a la información para su modificación o borrado, o, por ejemplo, para compartir su ubicación exacta con terceros. (BBVA API_Market, 2016, p.6).
- ❏ Interfaz uniforme: para la transferencia de datos en un sistema REST, este aplica acciones concretas (POST, GET, PUT y DELETE) sobre los recursos, siempre y cuando estén identificados con una URI. Esto facilita la existencia de una interfaz uniforme que sistematiza el proceso con la información. (BBVA API_Market, 2016, p.7).

- Sistema de capas: arquitectura jerárquica entre los componentes. Cada una de estas capas lleva a cabo una funcionalidad dentro del sistema REST. (BBVA API_Market, 2016, p.8).

- Uso de hipermedios: hipermedia es un término acuñado por Ted Nelson en 1965 y que es una extensión del concepto de hipertexto. Ese concepto llevado al desarrollo de páginas web es lo que permite que el usuario pueda navegar por el conjunto de objetos a través de enlaces HTML. En el caso de una API REST, el concepto de hipermedia explica la capacidad de una interfaz de desarrollo de aplicaciones para proporcionar al cliente y al usuario los enlaces adecuados para ejecutar acciones concretas sobre los datos. (BBVA API_Market, 2016, p.9).

  Ventajas que ofrece REST para el desarrollo

  1. Separación entre el cliente y el servidor: el protocolo REST separa totalmente la interfaz de usuario del servidor y el almacenamiento de datos. Eso tiene algunas ventajas cuando se hacen desarrollos. Por ejemplo, mejora la portabilidad de la interfaz a otro tipo de plataformas, aumenta la escalabilidad de los proyectos y permite que los distintos componentes de los desarrollos se puedan evolucionar de forma independiente. (BBVA API_Market, 2016, p.12).

  2. Visibilidad, fiabilidad y escalabilidad. La separación entre cliente y servidor tiene una ventaja evidente y es que cualquier equipo de desarrollo puede escalar el producto sin excesivos problemas. Se puede migrar a otros servidores o realizar todo tipo de cambios en la base de datos, siempre y cuando los datos de cada una de las peticiones se envíen de forma correcta. Esta separación facilita tener en servidores distintos el *front* y el *back* y eso convierte a las aplicaciones en productos más flexibles a la hora de trabajar. (BBVA API_Market, 2016, p.13).

  3. La API REST siempre es independiente del tipo de plataformas o lenguajes: la API REST siempre se adapta al tipo de sintaxis o plataformas con las que se estén trabajando, lo que ofrece una gran libertad a la hora de cambiar o probar nuevos entornos dentro del desarrollo. Con una API REST se pueden tener servidores PHP, Java, Python o Node.js. Lo único que es indispensable es que las respuestas a las peticiones se hagan siempre en el lenguaje de intercambio de información usado, normalmente XML o JSON. (BBVA API_Market, 2016, p.14).

HTTP

HTTP define un conjunto de métodos de petición para indicar la acción que se desea realizar para un recurso determinado. Aunque estos también pueden ser sustantivos, estos métodos de solicitud a veces son llamados *HTTP verbs*. Cada uno de ellos implementa una semántica diferente, pero algunas características similares son compartidas por un grupo de ellos: ej. Un *request method* puede ser safe, idempotent o cacheable. (Mozilla Corporation, 2019, p.1).

HTTP utiliza métodos de petición para indicar la acción que se desea realizar para un recurso determinado, dentro de estos métodos se encuentran:

“GET, este método solicita una representación de un recurso específico. Las peticiones que usan este método sólo deben recuperar datos.” (Mozilla Corporation, 2019, p.2)

“HEAD, este método pide una respuesta similar a la de una petición GET, pero sin el cuerpo de la respuesta.” (Mozilla Corporation, 2019, p.3)

“POST, este método se utiliza para enviar una entidad a un recurso en específico, causando a menudo un cambio en el estado o efectos secundarios en el servidor.” (Mozilla Corporation, 2019, p.4)

“PUT, este método reemplaza todas las representaciones actuales del recurso de destino con la carga útil de la petición.” (Mozilla Corporation, 2019, p.5)

“DELETE, este método borra un recurso en específico.” (Mozilla Corporation, 2019, p.6)

“CONNECT, este método establece un túnel hacia el servidor identificado por el recurso.” (Mozilla Corporation, 2019, p.7)

“OPTIONS, este método es utilizado para describir las opciones de comunicación para el recurso de destino.” (Mozilla Corporation, 2019, p.8)

“TRACE, este método realiza una prueba de bucle de retorno de mensaje a lo largo de la ruta al recurso de destino.” (Mozilla Corporation, 2019, p.9)

“PATCH, este método es utilizado para aplicar modificaciones parciales a un recurso.” (Mozilla Corporation, 2019, p.10)

.

JSON (JavaScript Object Notation - Notación de Objetos de JavaScript)

> JSON es un formato basado en texto estándar para representar datos estructurados en la sintaxis de objetos de JavaScript. Es comúnmente utilizado para transmitir datos en aplicaciones web (por ejemplo: enviar algunos datos desde el servidor al cliente, así estos datos pueden ser mostrados en páginas web, o viceversa). Se enfrentará a menudo con él, así que este artículo le entrega todo lo que necesita saber para trabajar con JSON utilizando JavaScript, incluyendo el análisis JSON para acceder a los datos en su interior, y cómo crear JSON.(Mozilla Corporation, 2020, p.1).
>
> JSON es un formato de datos basado en texto que sigue la sintaxis del objeto de JavaScript, popularizado por Douglas Crockford. Aunque es muy parecido a la sintaxis de objeto literal de JavaScript, puede ser utilizado independientemente de JavaScript, y muchos entornos de programación poseen la capacidad de leer (convertir; *parsear*) y generar JSON. (Mozilla Corporation, 2020, p.4).

Los JSON son cadenas - útiles cuando se quiere transmitir datos a través de una red. Debe ser convertido a un objeto nativo de JavaScript cuando se requiera acceder a sus datos. Esto no es un problema, dado que JavaScript posee un objeto global JSON que tiene los métodos disponibles para convertir entre ellos. (Mozilla Corporation, 2020, p.5)

API (interfaz de programación de aplicaciones)

Una API es un conjunto de definiciones y protocolos que se utiliza para desarrollar e integrar el software de las aplicaciones. API significa interfaz de programación de aplicaciones. Las API permiten que sus productos y servicios se comuniquen con otros, sin necesidad de saber cómo están implementados. Esto simplifica el desarrollo de las aplicaciones y permite ahorrar tiempo y dinero. Las API le otorgan flexibilidad; simplifican el diseño, la administración y el uso de las aplicaciones, y proporcionan oportunidades de innovación, lo cual es ideal al momento de diseñar herramientas y productos nuevos (o de gestionar los actuales). (RED HAT,p. 1).

CLOUD COMPUTING

La definición de cloud computing es ofrecer servicios a través de la conectividad y gran escala de Internet. La computación en la nube democratiza el acceso a recursos de software de nivel internacional, pues es una aplicación de software que atiende a diversos clientes. La multiplicación es lo que diferencia la computación en la nube de la simple tercerización y de modelos de proveedores de servicios de aplicaciones más antiguos. Ahora, las pequeñas empresas tienen la capacidad de dominar el poder de la tecnología avanzada de manera escalable. (SALESFORCE, p.2).

La computación en la nube ofrece a los individuos y a las empresas de todos los tamaños la capacidad de un pool de recursos de computación con buen mantenimiento, seguro, de fácil acceso y bajo demanda, como servidores, almacenamiento de datos y solución de aplicaciones. Eso proporciona a las empresas mayor flexibilidad en relación a sus datos e informaciones, que se pueden acceder en cualquier lugar y hora, siendo esencial para empresas con sedes alrededor del mundo o en distintos ambientes de trabajo. Con un mínimo de gestión, todos los elementos de software de la computación en la nube pueden ser dimensionados bajo demanda, usted solo necesita conexión a Internet. (SALESFORCE, p.3).

INFRAESTRUCTURA COMO SERVICIO (IaaS)

"Ofrece a los usuarios los recursos de la nube, como la informática, las redes y el almacenamiento en la nube, a través de una conexión de red". (RED HAT, p. 11)

PLATAFORMA COMO SERVICIO (PaaS)

"Ofrece a los usuarios una plataforma de software de aplicaciones, además de toda la infraestructura de TI que se necesita para ejecutarla, a través de una conexión de red. Normalmente así se ofrecen las plataformas de nube". (RED HAT, p. 12)

SOFTWARE COMO SERVICIO (SaaS)

"Ofrece a los usuarios una aplicación en completo funcionamiento, así como la plataforma en la que se ejecuta y la su infraestructura de TI que necesita, a través de una conexión de red". (RED HAT, p. 13).

SERVERLESS (Computación sin servidor)

El serverless computing o la arquitectura serverless es un modelo en la nube que permite a los usuarios crear y ejecutar aplicaciones y procesos sin entrar en contacto con el servidor subyacente. Por lo tanto, a pesar de su denominación, estos entornos en la nube también cuentan con servidores, con la diferencia de que es el proveedor el que se encarga de suministrarlo, gestionarlo y escalarlo. De esta forma, el enfoque serverless va un paso más allá del modelo Platform as a Service (PaaS), en el que los desarrolladores o clientes se ven obligados a dirigirse al servidor o los servidores mediante una API y a añadir o liberar recursos por su cuenta. (IONOS España S.L.U., 2020, p.2).

Como en los proyectos con una infraestructura de serverless computing no es necesario realizar estas tareas, este enfoque permite centrar toda la atención en el desarrollo y la ejecución del software correspondiente. No obstante, para que esto sea posible, al margen del propio código de programa, también hay que crear funciones sin estado que contienen instrucciones acerca de cómo debe reaccionar un programa a determinados eventos. Entonces, el correspondiente código de programa, junto con las funciones deseadas (que gracias a su ausencia de estado no cuentan con ningún tipo de dependencia de la infraestructura), se puede cargar en el entorno en la nube en cualquier momento para ejecutarse allí. Debido precisamente al papel esencial que juegan las funciones, hay proveedores que ofrecen sus servicios serverless bajo el nombre "Function as a Service (FaaS)". (IONOS España S.L.U., 2020,p.3)

MICROSERVICIOS

El software se puede desarrollar de diversas maneras. En lugar de realizar un proyecto como un todo global, en ocasiones puede ser más pertinente distribuir las tareas en pequeños componentes, como los llamados microservicios o microservices. Este término no solo tiene que ver con la programación de una aplicación informática, sino que también desempeña un papel significativo en la planificación con el fin de agilizar la gestión del proyecto. (IONOS España S.L.U., 2020, p.1).

No está acotado claramente cuándo se puede hablar de una arquitectura basada en microservicios y cuándo de otra que no lo está. A esta confusión se añade que los microservicios no hacen sólo referencia a la tecnología de desarrollo de software, sino también, y en gran medida, a una forma de trabajar de los propios programadores. (IONOS España S.L.U., 2020, p.2).

6. Metodología

Para llegar a la meta de los objetivos planteados en el proyecto, se establecieron fases en el mismo. Un punto muy fuerte que nos ayudó es la experiencia que tiene laborando en la empresa a la cual se le va a proponer la arquitectura uno de los integrantes del proyecto, el cual conoce bien el proceso al cual se le va a dar una solución tecnológica, como lo es la reubicación de mercancía al ERP

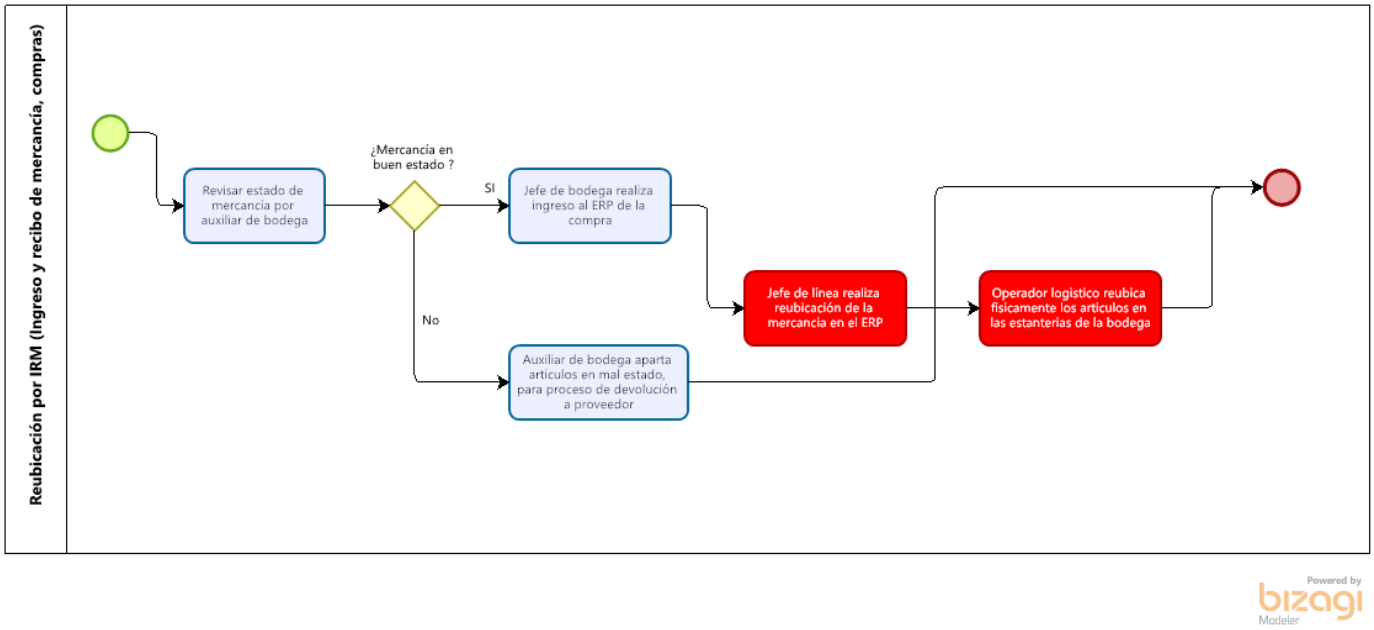
Fase	Actividad	Descripción
1. Iniciación	Identificar y tener claro el procedimiento de reubicación de mercancía en las bodegas de Continente S.A.S. Así como los actores implicados en cada parte del proceso	En esta fase, se solicitaron copias de los procedimientos escritos de reubicación de mercancía, se realizaron entrevistas con los jefes de bodegas y gerente de logística
2. Diseño de la arquitectura	Se identificaron los componentes idóneos, para poder brindar una solución segura, íntegra y con disponibilidad	Elaboración de la arquitectura mediante herramientas gráficas, creación mockups del aplicativo front end sugerido
3. Finalización	Documentación del diseño de la arquitectura	Se documentó cada componente arquitectónico, siendo detallados en cada uno de los mismos, con el fin de facilitar e identificar las herramientas de desarrollo más idóneas para soportar la arquitectura.

7. Desarrollo del proyecto

7.1 Modelado de proceso actual de reubicación de mercancía

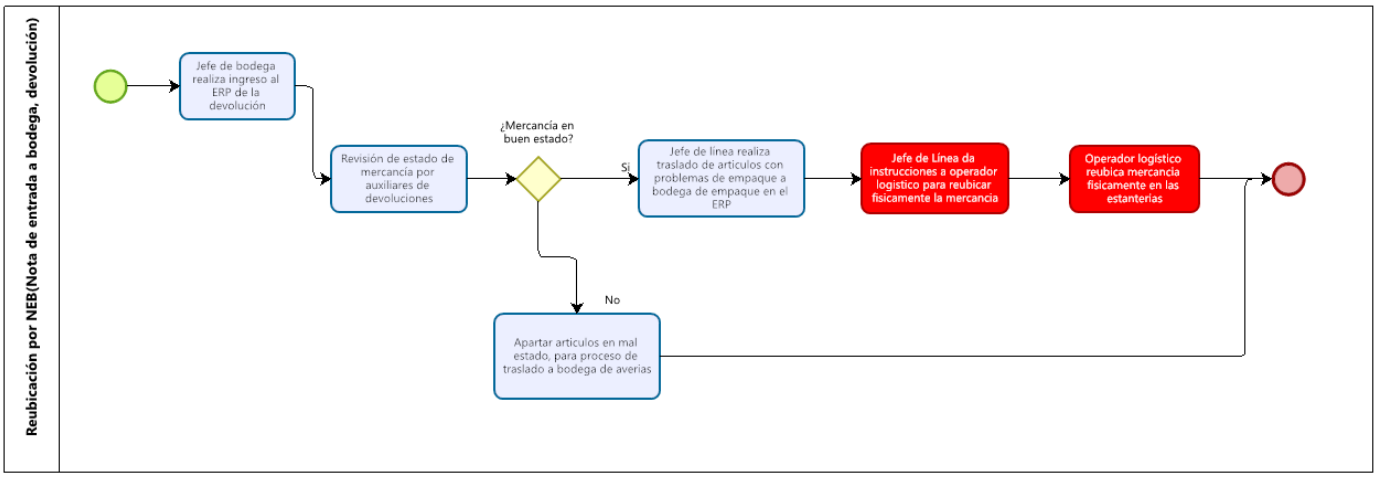
7.1.1 Modelado de proceso. Reubicación por IRM (Ingreso y recibo de mercancía, compra) (Actual)

La reubicación de mercancía en Continente S.A.S, se dispara con 2 procesos principales. El primero es la reubicación por IRM (Ingreso y recibo de mercancía por una compra). A continuación, se ilustra un diagrama de flujo que explica el proceso, los recuadros en rojo ilustran los puntos críticos en el proceso, donde se incrementa el margen de error entre lo ubicado en el ERP contra lo físico realizado por el operador logístico.



7.1.2 Modelado de proceso. Reubicación por devolución de mercancía NEB (Nota de entrada a bodega) (Actual)

El segundo proceso es la reubicación por NEB (Nota de entrada a bodega, devolución). A continuación, se ilustra un diagrama de flujo que explica el proceso, los recuadros en rojo ilustran los puntos críticos en el proceso, donde se incrementa el margen de error entre lo ubicado en el ERP contra lo físico realizado por el operador logístico.



7.2 Requerimientos no funcionales.

Continente S.A.S viendo la necesidad de escalar su ERP, más exactamente en su módulo de reubicación de mercancía, se ha visto en la necesidad de proyectarse hacia una arquitectura en la nube, la cual permita que el proceso de reubicación de mercancía sea mucho más eficaz y eficiente, reduciendo así el margen de error en los procesos como picking y toma física de inventario, los cuales dependen de una reubicación consistente con respecto al ERP Por lo tanto se han definido los siguientes requerimientos no funcionales.

Requerimiento no funcional	Descripción del requerimiento
Escalabilidad	1. Proporcionar crecimiento vertical y horizontal de forma fácil y auto escalado.
Disponibilidad	1. El módulo de reubicación de bodega deberá estar disponible todos los días del año. 2. El período de inactividad previsto del módulo de reubicación de bodega no debe superar las 40 horas al año. 3. El módulo de reubicación de mercancía debe tolerar una alta disponibilidad, se debe disponer de componentes que puedan brindar la posibilidad de asegurar la continuidad en la prestación del servicio. Se espera que el porcentaje de disponibilidad sea del 99%.
Flexibilidad	1. El tiempo de inactividad no prevista del módulo de reubicación de bodega, no debe superar las 10 horas al trimestre. 2. Se debe garantizar una independencia con la topología de red que se vaya a utilizar, por lo tanto, el ERP debería poder funcionar con varios tipos de comunicación.
Fiabilidad	1. La arquitectura propuesta debe responder de manera óptima a los fallos por lo cual deberá conservar el nivel descrito en el rendimiento. 2. La arquitectura debe poder brindar las garantías de recuperación de datos.
Interfaz de comunicación	1. La solución de arquitectura debe proponer la utilización de servicios web tipo REST, los cuales van a servir para enviar los recursos que se van a consumir por parte de los clientes.

7.3 Propuesta solución a implementar

Diseñar una arquitectura de software que sirva de interfaz y registre en tiempo real la reubicación de mercancía al módulo de logística del software ERP.

7.3.1 Diseño de la arquitectura EJA

Tomando los diagramas y requerimientos no funcionales descritos por parte de la empresa Continente S.A.S, hemos propuesto un diseño de arquitectura que funciona en la nube utilizando Software como servicio (SaaS) y función como servicio (FaaS), este diseño dará cumplimiento a esas necesidades identificadas en el proceso de reubicación de mercancía en las bodegas y tomará como base los requerimientos no funcionales relacionados, para que los atributos de calidad del ERP, no se vean afectados por esta solución.

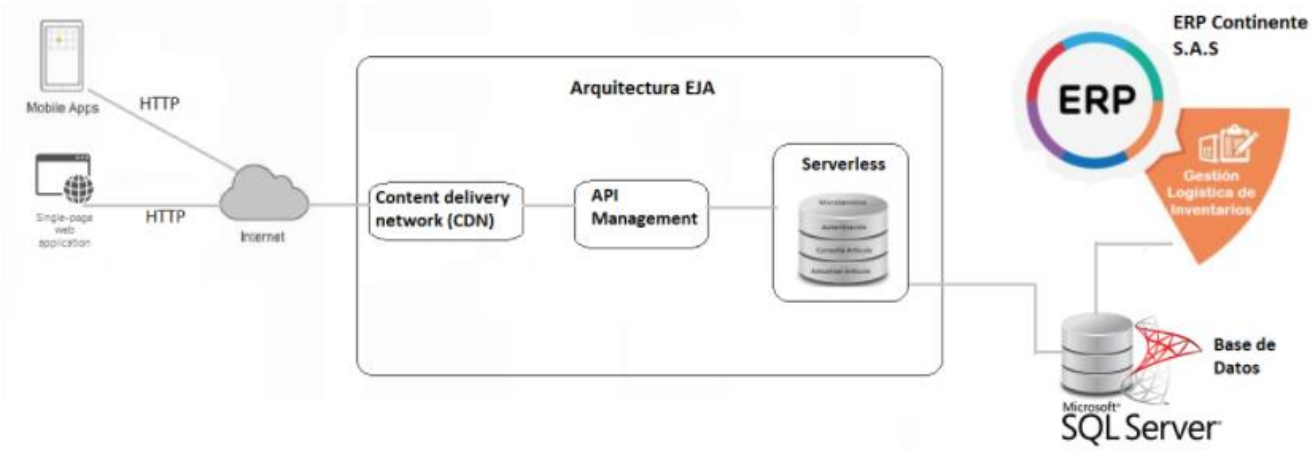
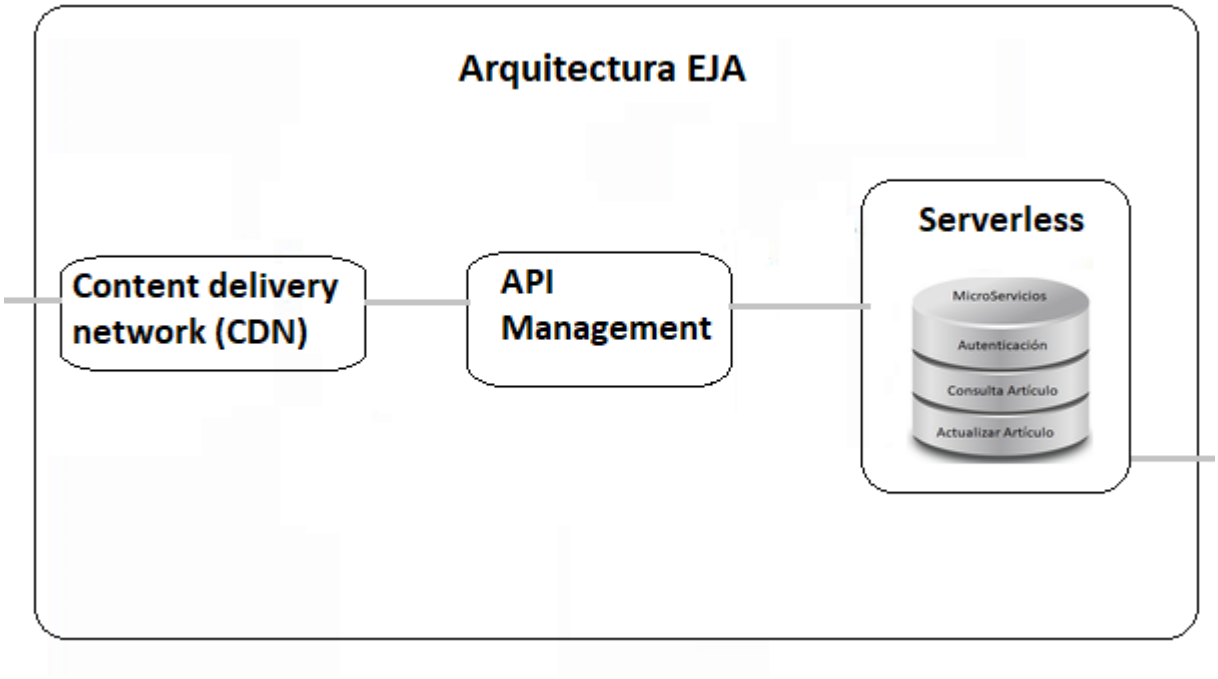


Fig. 1 Diseño de la arquitectura

7.3.2 Descripción de los componentes de la arquitectura EJA

Este es el Sistema de información principal de la compañía continente S.A.S este tiene automatizado todos los procesos de la empresa, pero presenta una limitación y es que este no está diseñado para desplegarse en la web, el módulo de gestión logística de inventarios será el que se afectará por esta solución.

La base de datos sobre la que está edificado el ERP es SQl Server, esta administra todas las transacciones que se llevan a cabo diariamente en los procesos que el ERP maneja, también será accedida desde los microservicios que harán posible que el proceso de reubicación de mercancía se haga desde las bodegas que tiene la empresa continente S.A.S.



Content delivery network (CDN) nos permitirá entregar de manera rápida los servicios que se van a consumir por parte de las aplicaciones front end.
API Management va a servir como puerta de entrada para que las aplicaciones accedan a los servicios que se van a prestar, estos estarán representados mediante un API Rest.
Serverless contendrá los microservicios que tendrán la lógica del negocio para mejorar el proceso de reubicación de la mercancía en las bodegas.

Los servicios serán consumidos por aplicaciones móviles y/o aplicaciones web que pueden ser desarrolladas desde cualquier lenguaje de programación front end, las cuales mediante la uri del end point van acceder al servicio, todo esto soportado por API Rest y el protocolo http.

7.3.3 Explicar cómo las aplicaciones del lado del cliente se comunican con los servicios expuestos.

las aplicaciones se comunicarán con los microservicios mediante el API Gateway de AWS el cual expondrá una API REST que permitirá mediante uri's comunicarse con el componente de AWS Lambda el cual tiene los servicios que serán accedidos, estos servicios afectan de manera directa a la base de datos que soporta al ERP.

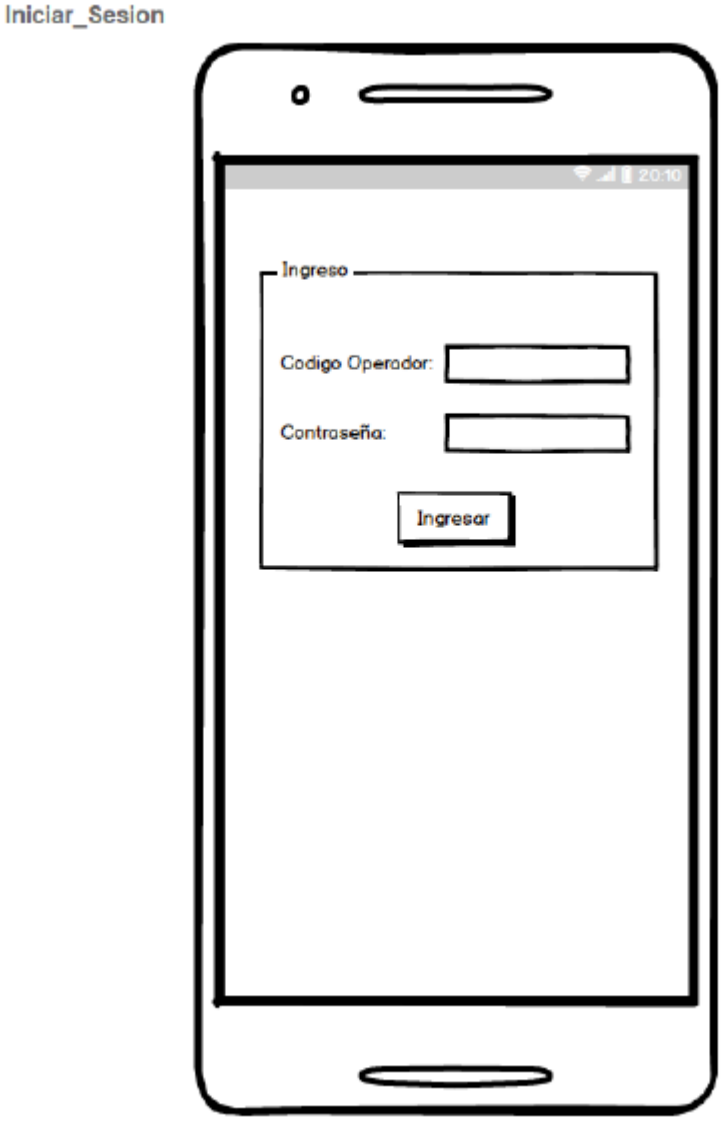
7.3.4 Describir los microservicios que van a servir para mejorar el proceso de reubicación de mercancía.

- Autenticar al operador logístico desde un dispositivo móvil y/o aplicación web con las credenciales ya existentes en el ERP Continente S.A.S. El operador logístico deberá poder autenticarse en la terminal móvil, con las credenciales ya existentes, permitiendo así tener un control de que usuario realiza los movimientos de reubicación de mercancía en el sistema ERP
- Consultar la cantidad de existencia en el inventario de un artículo en específico, ya sea indicando el código interno, el código de barras o la referencia de proveedor. El operador logístico al indicar ya sea el código interno, la referencia de proveedor o el código de barras, se debe poder mostrar la cantidad de existencias de este artículo en las diferentes bodegas.
- Actualizar la existencia de un artículo en una ubicación de una bodega puntual. El operador logístico al indicar: un artículo, una bodega, una ubicación origen y una ubicación destino, podrá trasladar la cantidad de existencia de una ubicación a otra, lo que se conoce como reubicación de mercancía.

7.3.5 Sugerir prototipos de las interfaces de usuario, mediante unos mockups para un cliente móvil.

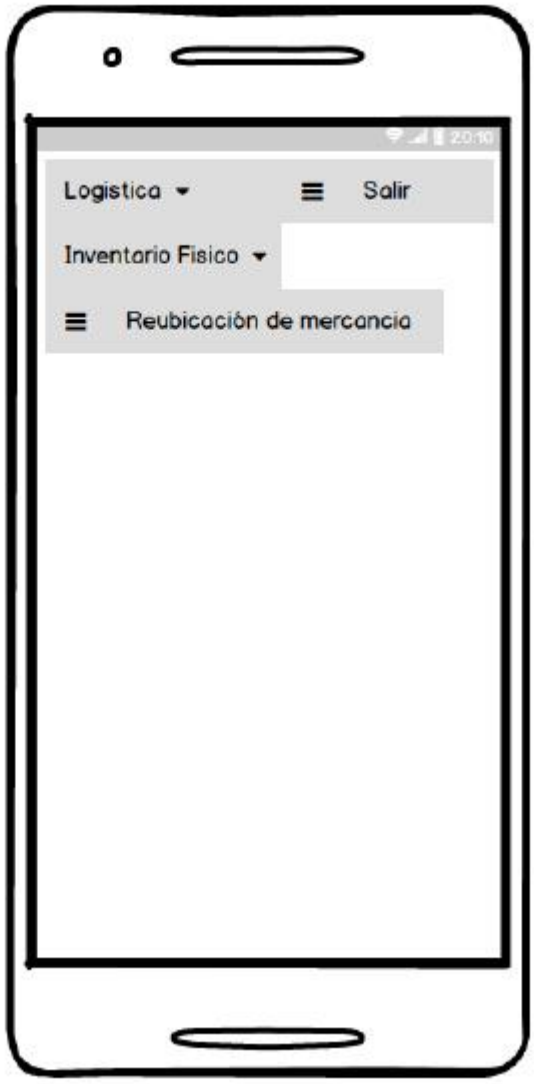
A continuación, presentamos unos prototipos, de la interfaz cliente de una terminal móvil.

Prototipo de inicio de sesión desde una terminal móvil



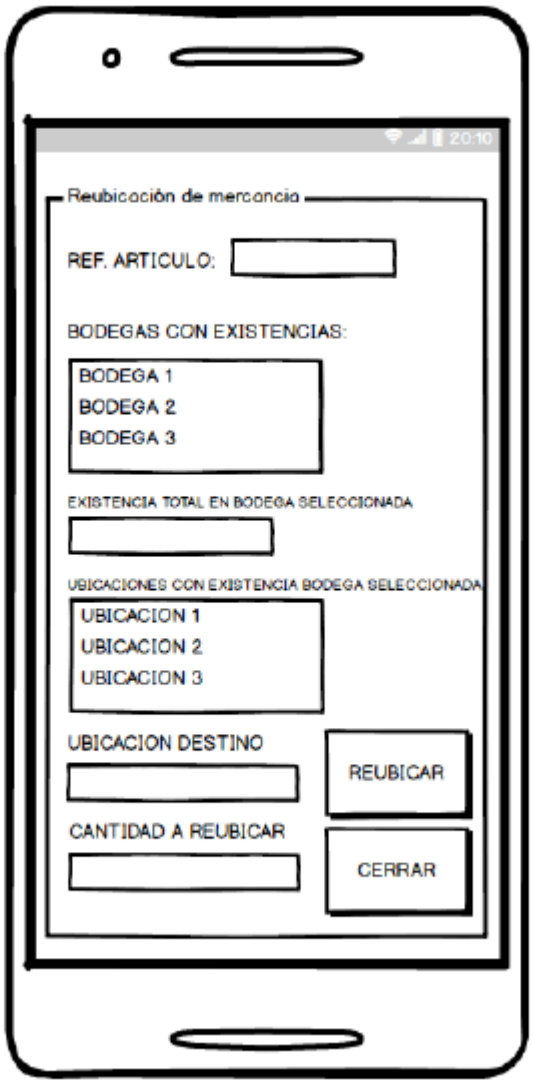
Prototipo del menú principal

Menu



Prototipo opción de reubicación de mercancía

Reubicacion_Mercancia



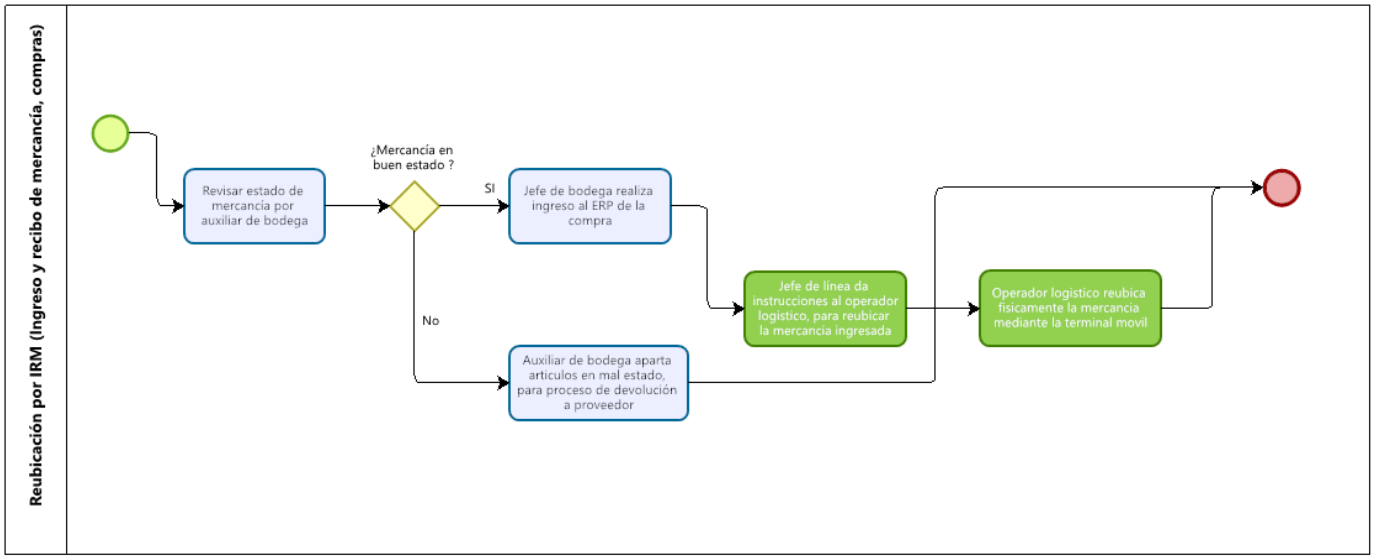
7.3.6 Sugerencias de especificaciones técnicas de la terminal móvil

- Ordenador móvil, se sugiere el siguiente: ZEBRA MC9200 (La cantidad que la empresa considere necesario para su operación logística)

  - Características de rendimiento mínimas
    - CPU Procesador OMAP 4 de doble núcleo de 1 GHz
    - Sistema operativo Android con Mobility Extensions (Mx)
    - Memoria mínima: 512 MB RAM / 2 GB de flash.
    - Ampliación: Ranura para tarjetas SDHC de 32 GB accesible por el usuario
    - Desarrollo de aplicaciones: PSDK y EMDK disponibles a través del sitio web de Zebra Support Central
    - Opción de captura de datos: SE965:1,4 motor de escaneado 1D de alcance estándar
  - Comunicaciones de datos inalámbricas
    - WILAN: Android: IEEE 802.11 a/b/g/n/d/h/i/k/r
    - Antena interna
    - Frecuencia de datos: 802.11a: hasta 54 Mbps; 802.11b: hasta 11 Mbps;
    - Seguridad de WLAN: WPA2 Enterprise, 802.1x; EAP-TLS; TTLS (CHAP, MS-CHAP, MS-CHAPv2, PAP o MD5); PEAP (TLS, MSCHAPv2, EAP-GTC); LEAP, EAP-FAST (TLS, MS-CHAPv2, EAP-GTC), WPA2/AES, CCX v4, FIPS 140-2 y IPv6.
- Routers en la bodega, para dar cobertura de red inalámbrica a las terminales móviles.
- Repetidores de señal que soporte **Wifi 802.11 n**

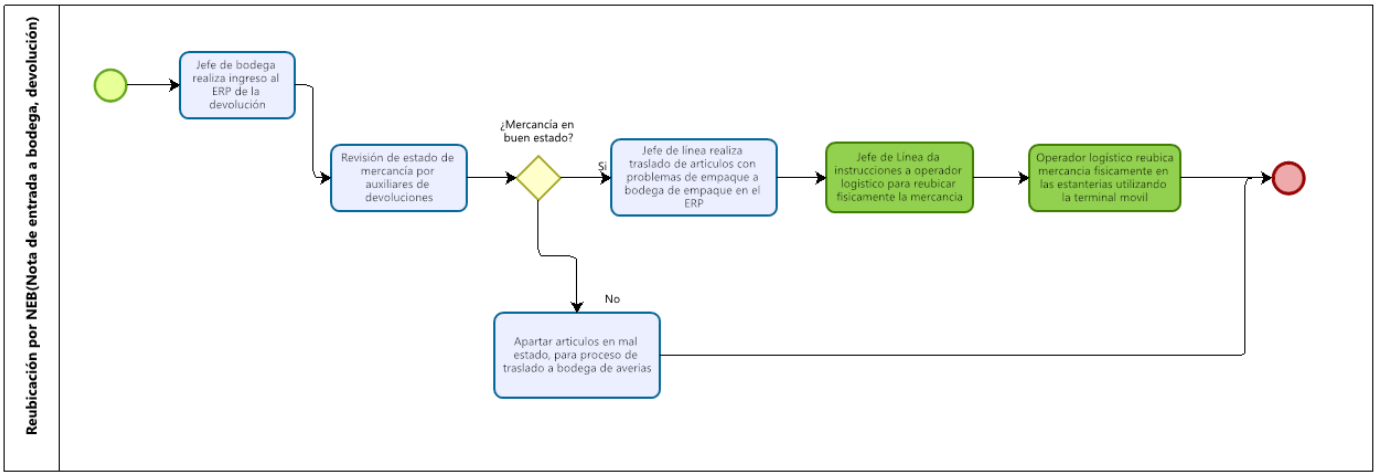
7.3.7 Modelado de proceso. Reubicación por IRM (Ingreso y recibo de mercancía, compra) (Con arquitectura propuesta).

Los procesos de color verde, indican la mejora ante la arquitectura propuesta, se logra evidenciar que el operador logístico mediante la terminal móvil afectara la reubicación directamente al ERP y no el jefe de línea.



7.3.8 Modelado de proceso. Reubicación por devolución de mercancía NEB (Nota de entrada a bodega) (Con arquitectura propuesta)

Los procesos de color verde indican la mejora ante la arquitectura propuesta, se logra evidenciar que el operador logístico mediante la terminal móvil afectara la reubicación directamente al ERP y no el jefe de línea.



Powered by bizagi Modeler

8. Resultados obtenidos y discusión

Los resultados obtenidos en este proyecto, muestran la importancia de la Cloud Computing como fuente primordial para llevar a cabo los siguientes atributos de calidad como la escalabilidad e interoperabilidad del Sistema ERP de la empresa Continente S.A.S, este diseño agrupa diferentes componentes que interactúan entre ellos para llevar una integración del proceso de reubicación de mercancía en las bodegas, podemos decir que al utilizar la arquitectura EJA el módulo de gestión de logística del ERP será más eficiente al manejar los procedimientos de picking y toma física de inventario.

9. Conclusiones

En este trabajo, se diseñó una arquitectura de software la cual va a permitir integrar el proceso de reubicación de mercancía en las bodegas de Continente S.A.S, con el software ERP de esta misma empresa. Lo más importante del diseño de esta arquitectura es la interoperabilidad y escalabilidad que ofrece, ya que de esta manera será posible integrar los movimientos de reubicación de mercancía directamente al módulo de logística del ERP. Lo que más ayudó al diseño de esta arquitectura fueron las diferentes tecnologías de computación en la nube que existen hoy en día en el mercado, descritas en la arquitectura de este caso de estudio. Lo más complejo del diseño de esta arquitectura, fue establecer los componentes idóneos y seguros, que faciliten la integración al ERP, ya que, debido al desarrollo y diseño de este, lo caracterizan como un ERP de limitado acceso.

10. Referencias bibliográficas

Mozilla Corporation. (23 de Marzo de 2019). *Métodos de petición HTTP*. https://developer.mozilla.org/es/docs/Web/HTTP/Methods

BBVA API_Market. (23 de Marzo de 2016). *API REST: qué es y cuáles son sus ventajas en el desarrollo de proyectos*.https://www.bbvaapimarket.com/es/mundo-api/api-rest-que-es-y-cuales-son-sus-ventajas-en-el-desarrollo-de-proyectos/

IMF Business School. (21 de Noviembre de 2019). *Relación entre la logística y la cadena de suministros*. Qué es la logística: https://blogs.imf-formacion.com/blog/logistica/logistica/relacion-logistica-cadena-de-suministros/

IONOS España S.L.U. (02 de Marzo de 2020). *Microservicios: más que la suma de sus partes*.https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/los-microservicios-en-el-desarrollo-de-aplicaciones/

IONOS España S.L.U. (30 de Enero de 2020). *Modelo Serverless: así funciona el moderno modelo en la nube*. https://www.ionos.es/digitalguide/servidores/know-how/serverless-computing/

Mozilla Corporation. (04 de Mayo de 2020). *Trabajando con JSON*. https://developer.mozilla.org/es/docs/Learn/JavaScript/Objects/JSON

RED HAT. (s.f.). *¿Qué es una API?* Obtenido de Qué son las API y para qué sirven: https://www.redhat.com/es/topics/api/what-are-application-programming-interfaces

RED HAT. (s.f.). *COMPUTACIÓN EN LA NUBE*. Servicios de la nube: https://www.redhat.com/es/topics/cloud

SALESFORCE. (s.f.). *Cloud Computing - Aplicaciones en un solo tacto*. ¿Qué es Cloud Computing?: https://www.salesforce.com/mx/cloud-computing/