

## Diseño de sistema web para gestión y administración empresarial basado en tecnología JEE y Primefaces

Design of a web system for business management and administration based on JEE and Primefaces technology

Projeto de um sistema web para gestão e administração de negócios baseado na tecnologia jee e primefaces

**Jonathan Patricio Cárdenas Ruperti**

jonathan.cardenas.ruperti@utelvt.edu.ec

<https://orcid.org/0000-0003-2379-0917>

Docente Investigador de la Facultad de Ingenierías en la Universidad Técnica Luis Vargas Torres de Esmeraldas, Ecuador

### RESUMEN

Las empresas ecuatorianas requieren gestionar adecuadamente las áreas como Inventario, Nómina y Activos Fijos de forma eficaz, que permitan tener integridad en los datos, sin embargo, no cuentan con una herramienta de administración que les ayude a gestionar todos los recursos de la empresa; por ello la necesidad de automatizar totalmente estas áreas de la organización. Se pretendió investigar y determinar las tecnologías más actuales de desarrollo con RestFull (Representational State Transfer) para poder desarrollar un software de alta jerarquía. Se investigó y analizó los conceptos involucrados para cada una de las áreas antes mencionadas en función a las reglas de negocio dadas. El propósito del presente proyecto se basa en el diseño e implementación de un sistema para la gestión administrativa con JEE (Java Platform Enterprise Edition) y PrimeFaces mediante el uso de la metodología XP (eXtreme Programming). En este proyecto se describe el proceso de desarrollo utilizando la metodología XP, el diseño e implementación del sistema, y los resultados del proyecto. Es así que los módulos principales involucrados dentro de la realización del sistema son: Administración de Activos Fijos, Administración de Nómina y Administración de Inventarios, siendo todos estos módulos los más importantes que requiere la empresa que sean automatizados de acuerdo al flujo de negocio que posee. La distribución de los procesos en varios RestFull para el desarrollo de este proyecto muestra ventajas como: modularidad, interoperabilidad y reusabilidad, que son uno de los principios de la metodología empleada. Durante la construcción del sistema con la metodología XP se elaboraron iteraciones para un avance incremental con mini versiones funcionales de forma inmediata siendo así el alcance del desarrollo ágil y por ende rápido enfocado en la construcción del sistema.

**Palabras clave / Descriptores:** Desarrollo de un sistema. Gestión. Metodologías ágiles. Java. Ecuador.

### ABSTRACT

Ecuadorian companies need to properly manage areas such as Inventory, Payroll and Fixed Assets effectively, which allow data integrity, however, they do not have an administration tool that helps them manage all the company's resources; hence the need to fully automate these areas of the organization. The aim was to investigate and determine the most current development technologies with RestFull (Representational State Transfer) in order to develop high-ranking software. The concepts involved for each of the aforementioned areas were investigated and analyzed based on the given business rules. The purpose of this project is based on the design and implementation of a system for administrative management with JEE (Java Platform Enterprise Edition) and PrimeFaces through the use of XP (eXtreme Programming) methodology. This project describes the development process using the XP methodology, the design and implementation of the system, and the results of the project. Thus, the main modules involved in the realization of the system are: Fixed Asset Management, Payroll Management and Inventory Management, all of these modules being the most important that the company requires to be automated according to the business flow it has. The distribution of the processes in several RestFull for the development of this project shows advantages such as: modularity, interoperability and reusability, which are one of the principles of the methodology used. During the construction of the system with the XP methodology, iterations were elaborated for an incremental advance with mini functional versions immediately, thus being the scope of agile development and therefore fast focused on the construction of the system.

**Keywords:** Development of a system. Management. Agile methodologies. Java. Ecuador.

### RESUMO

As empresas equatorianas precisam gerenciar adequadamente áreas como Estoque, Folha de Pagamento e Ativo Fixo de forma eficaz, que permitem a integridade dos dados, porém, não possuem uma ferramenta de administração que as ajude a gerenciar todos os recursos da empresa; daí a necessidade de automatizar totalmente essas áreas da organização. O objetivo foi investigar e determinar as tecnologias de desenvolvimento mais atuais com RestFull (Representational State Transfer)

para desenvolver software de alto nível. Os conceitos envolvidos para cada uma das áreas mencionadas foram investigados e analisados com base nas regras de negócio dadas. O objetivo deste projeto baseia-se na concepção e implementação de um sistema de gestão administrativa com JEE (Java Platform Enterprise Edition) e PrimeFaces através da utilização da metodologia XP (eXtreme Programming). Este projeto descreve o processo de desenvolvimento utilizando a metodologia XP, o desenho e implementação do sistema e os resultados do projeto. Assim, os principais módulos envolvidos na realização do sistema são: Gestão de Ativo Fixo, Gestão de Folha de Pagamento e Gestão de Estoques, sendo todos esses módulos os mais importantes que a empresa necessita para serem automatizados de acordo com o fluxo de negócios que possui. A distribuição dos processos em diversos RestFul para o desenvolvimento deste projeto apresenta vantagens como: modularidade, interoperabilidade e reusabilidade, que são um dos princípios da metodologia utilizada. Durante a construção do sistema com a metodologia XP, foram elaboradas iterações para um avanço incremental com mini versões funcionais imediatamente, sendo assim o escopo do desenvolvimento ágil e, portanto, rápido focado na construção do sistema.

**Palavras-chave:** Desenvolvimento de um sistema. Gestão. Metodologias ágeis. Java. Equador.

## 1. INTRODUCCIÓN

En la actualidad, los procesos administrativos de las instituciones, organizaciones y empresas son cada vez más exigentes, por lo que se requiere la automatización de los mismos para mejorar la realización de éstos. Es por ello, que los nuevos sistemas de gestión y administración empresarial deben ser completamente estables, fiables y a la vez seguros, para mantener íntegra la información confidencial de las empresas.

La empresa ASESTRATÉGICO CÍA LTDA, presta servicios de asesoría financiera, contable y tributaria con una trayectoria de aproximadamente 20 años de experiencia en el mencionado campo profesional. Esta firma tiene implementado un sistema con muchas falencias en cuanto a estabilidad y seguridad de los datos, por lo cual se lo podría catalogar como un aplicativo ya obsoleto, que procesa incorrectamente la información y sin mucha utilidad para el usuario. La empresa está dispuesta a proveer toda la información necesaria para que se lleve a cabo el presente proyecto y poder solucionar todos las posibles contrariedades y complicaciones que se dan en este momento con el sistema actual. El desarrollo permitirá tener un sistema completamente nuevo y orientado a la Web, para así suplir con todos los problemas de accesibilidad, confiabilidad y disponibilidad que se presentan hoy en día

## 2 FUNDAMENTO TEÓRICO

### Algunos aspectos conceptuales:

#### Administración de Activos fijos

Los activos fijos, son aquellos que no varían durante el tiempo de vida de una empresa, también se consideran que son complicados para su venta en el caso de que la empresa desee conseguir dinero, debido a que el propósito es permanecer durante mucho tiempo en la empresa.

Si bien es cierto que estos activos no son eternos, la contabilidad obliga a que estos se deprecien cada tiempo, debido a que influyen varios factores sobre ellos, como el uso, el desgaste, el paso del tiempo, por obsolescencia, de forma que se refleje su valor más ajustado posible o a amortizar los gastos a largo plazo.

#### Tipos de depreciación

Existen dos tipos de depreciación que afectan a los Activos Fijos de manera directa, a continuación, se describe:

**Depreciación Física.** - Relaciona el uso de un activo depreciable a lo largo del tiempo de vida que ha tenido.

**Depreciación Funcional.** - Surge de la obsolescencia o inadecuación del activo con respecto a sus funciones en óptimo estado.

Toda empresa necesita inventariar los activos fijos que posee, donde se procede a identificar las áreas en donde se encuentra estos activos. Las ventajas de automatizar esto, son las siguientes:

- Agilidad para la captura de datos.
- Eliminación de tiempos muertos durante el proceso de inventarios.
- Activos fijos localizables en cualquier momento.
- Permite realizar inventarios con más frecuencia.
- Aumenta la conciencia de usuarios para no realizar movimientos de equipo sin dar aviso al departamento correspondiente.
- Clasificación de sus activos fijos en un máximo de tres niveles según las necesidades que se presenten.
- Le permite planear sus actividades de mantenimiento de activos conociendo los gastos bajo este concepto

El módulo de Activos Fijos permite obtener información actualizada de depreciación y el valor del mismo en cualquier momento que el usuario lo requiera. El usuario puede consultar o imprimir un Estado de Cuenta de cada uno de sus Activo, para tener información sobre el valor de adquisición, las depreciaciones contabilizadas y el Valor en Libros de cada Activo.

## Administración de Nómina

La nómina o rol de pagos es por medio de la cual el trabajador recibe principalmente su remuneración por el trabajo realizado durante cierto tiempo que por lo general es cada mes, dependiendo del beneficio que se le entrega; en otro sentido, la nómina no es más que darle un recibo en donde se informa al trabajador sus ingresos, sus descuentos, bonos, entre otros, estructurando el salario que en su día pactó con la compañía que lo emplea.

El principal beneficio que recibe el trabajador es el sueldo que recibe mes a mes, es por eso que cabe aclarar la diferencia entre el sueldo neto y el sueldo bruto. A grosso modo, el sueldo neto es aquel que realmente recibe cada empleado, es decir, son los valores que le entrega el empleador al empleado, ya sea por transferencia, cheque o en efectivo; por otro lado, el sueldo bruto se refiere al monto total que figura en el contrato del trabajador con todos los complementos salariales, retenciones y aportes al Seguro Social.

La administración de la nómina de una empresa a través de un sistema, constituye un proceso repetitivo en cierta forma, tomando en cuenta datos cambiantes dependiendo de algún evento ocurrido, ya sea en el tiempo que se encuentra, como por ejemplo los décimos, o por algún otro motivo, como por ejemplo, la salida de un empleado o la realización de un préstamo por parte de la empresa hacia el empleado, pero estos eventos son pocos con relación a muchos que siempre se tienen que calcular y que es algo tedioso para la persona encargada de esta actividad.

El módulo de nómina maneja los sueldos de cada empleado, tomando en cuenta todos los datos relacionados con el trabajador, ya sea, de tipo informativo, de descuento, o de incremento sobre el sueldo que recibe el empleado, así como también puede generar datos de tipo estadísticos 11.

A continuación, se nombra alguno de los diferentes rubros que puede constituir la nómina:

Sueldo o Salario. - Indica la remuneración económica que recibe un trabajador por los servicios prestados a la empresa.

Décimo cuarto sueldo. - Es un bono escolar que es equivalente a un salario mínimo vital y se lo paga anualmente dependiendo que región este le empresa, si es costa se lo paga en el mes de febrero y si es sierra es en el mes de julio.

Décimo Tercer Sueldo. - Es considerado como para algunos como bono navideño. Es la suma de todos sus ingresos durante el año como bono, sueldo, horas extras, entre otras; y ese total se divide para los 12 meses del año.

Fondos de reserva. - Son valores que el IESS recauda a los trabajadores mes a mes y que, a partir del año, según la nueva ley el empleador pagará con concepto de fondo de reserva de manera mensualizada y de forma directa a sus trabajadores el 8.33% de la remuneración de aportación.

La versatilidad al cambio de parámetros, es una de las características que poseerá el módulo de nómina, de acuerdo a la situación actual en la que se encuentre la empresa, dependiendo de un evento interno, por ejemplo, el gerente decide que todos los empleados se les incremente el sueldo en un 10%, o un evento externo.

### **Administración de Inventario**

El inventario es el registro de los bienes y demás cosas pertenecientes a una organización de cualquier tipo, con el propósito de evaluar su existencia y fijar el volumen de producción y ventas, esto reflejará la situación actual de la producción de la organización para tomar decisiones importantes que pueden salvar o mantener la producción y cumplir objetivo propuesto. Las razones por las cuales se requiere inventariar los bienes, son las siguientes:

- Reducir costos de pérdida. - Al realizar un pedido voluminoso sin tener la necesidad, influye varios inconvenientes, como la capacidad de almacenamiento, dependiendo del producto, se puede dañar, el recurso económico, el dar seguimiento y si no es la época del producto, se procede a tardar en recuperar la inversión y la ganancia.
- Reducir costos por material faltante. - El no tener el material suficiente para satisfacer la demanda, surgen varios problemas como clientes insatisfechos, adquirir a mayor precio los recursos, entre otros problemas.
- Reducir costos de adquisición. - Cuando existe la oportunidad de que un producto se pueda comprar por mayor a menor precio, puede ser favorable para la organización, debido a que no se invierte mucho capital en la adquisición del producto.

Este módulo maneja proveedores, productos, categorías de productos, entre los más importantes, el cual ayudará a conocer en tiempo real los productos que se encuentran en bodega, que productos tiene órdenes de pedido pendientes, que usuarios han solicitado productos entre otra funcionalidad<sup>13</sup>.

El control se lleva mediante tarjetas kárdex<sup>14</sup>, en donde se registran cada producto, su valor de compra, la fecha de adquisición, el valor de la salida de cada producto y la fecha en que se sale de la bodega. De esta forma se puede conocer el saldo que se encuentra en bodega (la existencia) y el costo de cada producto.

### **3- Herramientas de Desarrollo:**

#### **Netbeans 6.9**

Netbeans es un entorno de desarrollo integrado que principalmente fue hecho para el lenguaje de programación Java. Cabe recalcar que Netbeans es un IDE (Integrated Development Environments) gratuito. La plataforma

Netbeans permite que los proyectos nazcan desde módulos, en donde contienen clases de Java escritas para interactuar con las API (Aplication Programming Interface)<sup>17</sup> de NetBeans y un archivo especial que lo identifica como módulo. Estos módulos son fáciles de extender a lo largo del proyecto, ya sea por el código o por desarrolladores programando en el proyecto.

Esta herramienta, está escrita en java la misma que soporta cualquier tipo de lenguaje de programación, además de soportar el desarrollo de Aplicaciones empresariales con Java EE 5, incluyendo herramientas de desarrollo visuales de SOA (Arquitectura Orientada a Servicios)<sup>18</sup>, herramientas de esquema XML (Extensible Markup Language)<sup>19</sup>, orientación a web servicios, orientación a RESTFUL y modelado UML (Lenguaje Unificado de Modelado).

Existen varios packs, como el NetBeans Enterprise Pack, NetBeans C/C++ Pack, PHP Pack, entre otros, los cuales hacen que este IDE, no solo se pueda desarrollar utilizando el lenguaje Java sino otros tipos de lenguaje.

## MySQL 5.0

Es un sistema de administración de bases de datos para base de datos relacionable, multihilo y multiusuario. Así MySQL no es más que una aplicación que permite gestionar archivos llamados de bases de datos desarrollada como software libre, siendo una de las mejores bases de datos y más populares del mundo.

Aunque MySQL es software libre, MySQL también tiene una distribución comercial, la cual tiene derecho a un soporte técnico pero que no difiera de la versión que no es licenciada. Desarrollado en código ANSI<sup>25</sup>, la mayor parte del código de la base de datos no es open-source, está patrocinado por una empresa privada que posee el copyright del código, esto puede diferenciar de otros proyectos que son de total open-source.

## RESTful

REST (Representational State Transfer) es una arquitectura de software para sistemas hipermedias distribuidos tales como la Web. Son principios para el diseño de arquitecturas en la red que resumen como los recursos son definidos y decepcionados.

Aunque REST no es un estándar, está basado en estándares:

- HTTP<sup>29</sup>
- URL
- Representaciones de los recursos: XML/HTML<sup>30</sup>/GIF/JPEG..
- TIPOS MIME: text/xml, text/HTML.

El estilo de arquitectura subyacente a la Web es el modelo REST. Los objetivos de este estilo de arquitectura se listan a continuación:

- Escalabilidad de la interacción con los componentes. Esto es posible gracias a que la web ha crecido exponencialmente, se refleja en la variedad de clientes que pueden acceder a través de la Web: estaciones de trabajo, sistemas industriales, dispositivos móviles.
- Generalidad de interfaces. A diferencia de SOAP, REST está sobre el protocolo HTTP, esto significa que cualquier cliente puede acceder sin ninguna configuración adicional.
- Funcionamiento independiente. Esta es otra ventaja al estar sobre el protocolo HTTP, debido a que, durante el transcurso de los años, van surgiendo nuevas tecnologías, pues sería un tanto complicado que servidores viejos se entiendan con servidores actuales, gracias al protocolo HTTP, esto puede ser posible.
- Compatibilidad con componentes intermedios. Los componentes intermedios se refieren a: proxys, caches, firewall y Gateway, es por esto que la compatibilidad con estos componentes permite reducir la latencia de interacción, reforzar la seguridad y encapsular otros sistemas.

REST logra satisfacer estos objetivos aplicando cuatro restricciones:

- Identificación de recursos y manipulación de ellos a través de representaciones. Esto se consigue mediante las URL31 que se generan, puesto a que por medio de ellas se manipula los recursos, pero hasta cierto punto, dependiendo de que se pueda realizar con la URL.
- Mensajes auto descriptivos. REST dicta que los mensajes http deberían ser tan detallado como sea posible, para que los componentes intermedios puedan entenderlos y así ejecutar métodos de acuerdo a los mensajes.
- Hipermedia como un mecanismo del estado de la aplicación. El navegador se encarga de saber navegar recurso por recurso y recoge información que le es de necesidad y si es necesario cambia el estado de la aplicación o de ciertos recursos, todo esto se realiza del lado del cliente, del lado del servidor, el servidor es de poco interés seguir la pista a todas las sesiones de clientes conectados a él, pero él sabe en qué estado se encuentran los recursos32.

REST utiliza los métodos HTTP:

Esta es una de las características básicas de REST, puesto que la mayor parte de su funcionalidad lo hace a través de http, por ejemplo, HTTP GET es un método por el que se obtiene datos, el cual ejecuta una consulta esperando que el servidor Web la realice y devuelva un conjunto de recursos. Esto sucede con todos los métodos, convirtiendo una asociación uno-a-uno entre las operaciones de crear, leer, actualizar, borrar y los métodos HTTP. Se detalla a continuación la asociación:

- Se usa POST para crear un recurso en el servidor.
- Se usa GET para obtener un recurso.
- Se usa PUT para cambiar el estado de un recurso o actualizarlo.
- Se usa DELETE para eliminar un recurso.

Cabe recalcar, que el uso indebido de estos recursos, puede perjudicar en cierta forma a la base de datos, por ejemplo, se menciona que el método GET se utiliza para realizar una consulta y que devuelve datos, pues existen algunas APIs que, dentro de este método, incluyen operaciones transaccionales como update, los cuales alteran los datos, esto es muy riesgoso porque si se realiza una consulta, se espera que devuelva recursos y no que se alteren. El desarrollador debe de interpretar bien cada uno de los conceptos. Para ejecutar alguna acción de este tipo, se puede utilizar el método POST, que, por medio de algún lenguaje como el XML, se envía el dato que se quiere que cambie y no enviar datos en el método GET.

#### **4- Construcción del sistema:**

##### **Presencia del cliente**

Según la metodología XP, el cliente debe de estar en el sitio de trabajo para la construcción óptima del sistema y aclarar dudas cara a cara que el desarrollador tenga, pero para este proyecto, fue complicado aplicar esta etapa en la construcción del sistema, ya que el cliente tenía que realizar su trabajo habitual de todos los días y el desarrollador no podía dejar su sitio de trabajo para ir donde el cliente y realizar preguntas, se debería pagar las horas de trabajo al cliente para que esta etapa se pueda cumplir de una forma estricta, cosa que no se pudo realizar esa estrategia; en lugar de aquello, se suplantó la idea de que tanto el programador como el cliente estén cara a cara por colocar un teléfono en el sitio de trabajo del cliente y del programador y así poder comunicarse por teléfono cuando el desarrollador necesitaba aclarar alguna duda que tuviera.

Es significativo aclarar que, es de suma importancia que el cliente se encuentre con el programador, para que pueda aclarar dudas sobre el desarrollo de la aplicación y es esto que esta metodología es fuerte en este aspecto, pero se pudo sustituir con la comunicación vía telefónica y con reuniones pre programadas con el cliente, donde se aprovecharán todas las dudas que tenga y de esta forma no perder mucho tiempo al cliente.

##### **Programación en Pareja**

Según la metodología XP, toda la producción de código debe de ser en parejas, sentadas en frente de un solo monitor para tener un diseño de mejor calidad y un código más organizado, teniendo en cuenta que los problemas que se presentan se pueden solucionar más fácilmente y rápidamente.

Debido a que el desarrollo de este proyecto, es individual, este objetivo fue muy difícil de aplicar; lo que se tuvo que implementar, fue que el cliente en algunos casos, ayudó a colaborar con el programador a realizar la programación en pareja en sus tiempos libres. Esto se pudo llevar a cabo, porque el cliente tenía un poco de conocimiento de programación y sistemas, sin mencionar que su carrera profesional, no tiene nada que ver con eso, pero con la gran experiencia que contaba, se pudo suplir este inconveniente.

En la primera iteración, se realizó el apoyo del cliente como segundo programador en el desarrollo, pero debido a que no contaba con mucho tiempo y con lo que se realizó un balance del tiempo de desarrollo cuando se contaba con esa ayuda extra; se concluyó que no era tan indispensable de su ayuda, pues esto fue sustituido con la ayuda externa de programadores que tienen experiencia sobre la tecnología implementada en este proyecto, citando a cada uno de ellos por medio de sistemas de mensajerías electrónica para dialogar de temas como problemas suscitados y su resolución.

Desde la segunda iteración se vio el cambio en el tiempo de desarrollo, cubriendo con esta estrategia, los principales objetivos, como son, rapidez, fácil resolución de problemas con un diseño de mejor calidad y un código más organizado.

##### **Integración secuencial**

Para aplicar esta etapa de XP, se debe de saber que antes de aplicar nuevo código al proyecto, se debe garantizar que la última versión halla pasado todas las pruebas, solo se debe hacer una integración a la vez y se debe de tener claro cuál es la última versión funcional. Desde el primer momento, se llevó un registro de los respaldos del último código funcional, el cual su nombre estaba compuesto de la fecha y que tipo de servidor maneja para así saber qué es lo que contiene, si es la parte de datos junto a la base de datos o la parte de presentación del sistema.

Estos respaldos eran almacenados en dos partes, un respaldo externo en un host y un respaldo local, haciendo que la información está libre de cualquier eventualidad con respecto al equipo en que se está manejando. De esta forma y cumpliendo con la explicación primera, se garantizó la fluidez en la integración de nuevo código, haciendo que, si existe algún error en la versión que se está trabajando, se pueda volver a la versión anterior sin ningún problema.

## Integraciones frecuentes

Los principios fundamentales de la metodología XP con respecto a este tema, son que se debe hacer integraciones en cada momento como un máximo de tiempo de un día; entre más se tarde en encontrar un problema, resultará más costoso resolverlo y la integración frecuente evita problema como el de trabajar sobre una clase obsoleta. Durante el transcurso de este proyecto, no fue aplicado este tema con frecuencia, ya que el equipo de desarrollo se conformaba por una persona y él fue el único encargado de modificar cada versión y de ser necesario, él podía cambiar otras clases sin que otra parte se dañen en su desarrollo.

## Aplicación de Estándares

Otro de las más importantes etapas es la estandarización del código y en este proyecto se aplicó desde el principio. Esta estandarización del código, fue desarrollada antes de comenzar a codificar, dedicando el tiempo necesario y cubriendo todos los casos posibles que podían suscitarse en el desarrollo de esta aplicación.

Si bien el fin de estandarizar el código es de poder dar un punto colectivo y en el caso de este proyecto no se tiene la parte colectiva, ya que se tiene a un desarrollador; el poder desarrollar el sistema de esta forma, se ven el resultado a futuro, cuando se quiera desarrollar otras funciones y de forma muy acelerada y con más personas, así se realice con la misma persona que desarrolló el proyecto, simplemente es para que se pueda acordar de lo que en algún momento desarrollo, porque tiene un orden preestablecido. Este no es un principio que solo sigue esta metodología, si no muchas otras, las cuales hacen que sea un principio muy global en la vida del desarrollador promedio en nuestro entorno.

### 4.1 Desarrollo de interacciones:

#### Primera iteración (Módulo Sistema).

#### Historia de Usuario 1: Registrar usuarios nuevos.

Tarea 1: Crear interfaz de formulario para el registro

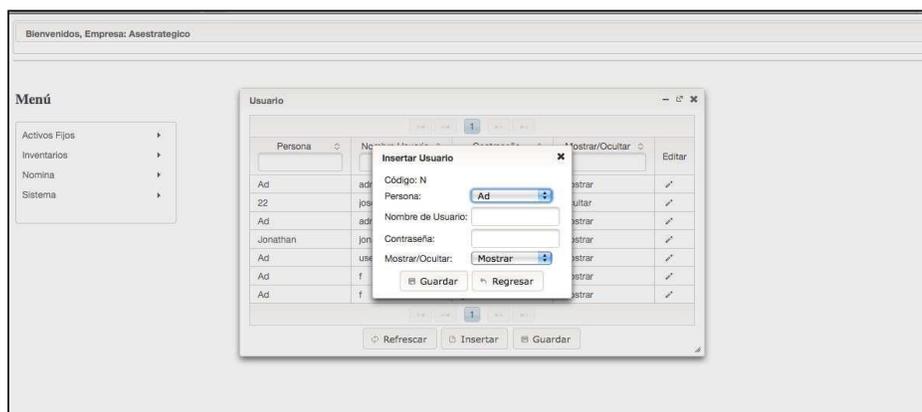


Figura 1: Formulario para el Registro

**Tarea 2: Crear base de datos para usuarios.**

sis_usuario			
<u>susu_codigo</u>	<pi>	Variable characters (12)	<M>
susu_nusuario		Variable characters (30)	
susu_contrasenia		Variable characters (30)	
susu_codigo_ua		Variable characters (12)	
susu_fecha_a		Date	
susu_codigo_um		Variable characters (12)	
susu_fecha_um		Date	
susu_mo_oc		Variable characters (10)	<M>
PK_sis_usuario	<pi>		

Figura 2: Tabla sis\_usuario

**Historia de Usuario 2: Permisos de acceso a usuarios.**

**Tarea 2.1: Crear formulario de mantenimiento de permisos**

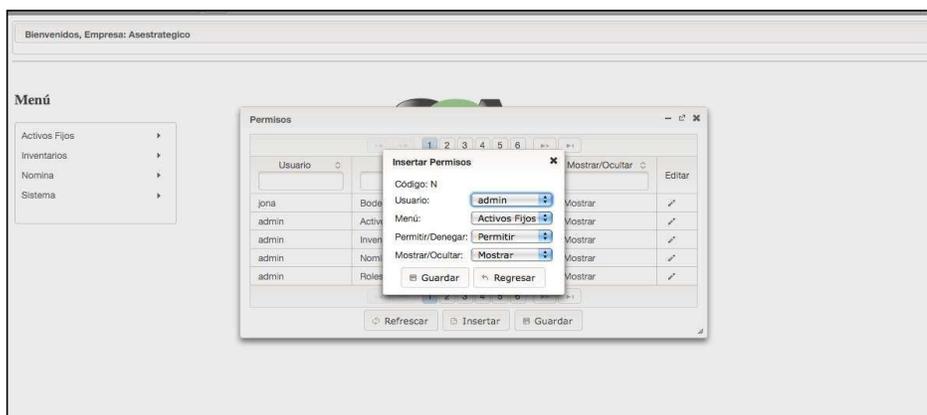


Figura 3: Formulario de mantenimiento de permiso

**Tarea 2.2: Crear base de datos para permisos de usuarios.**

sis_permisos			
<u>sper_cod</u>	<pi>	Variable characters (12)	<M>
sper_permi		Variable characters (15)	<M>
sper_fech_ua		Date	<M>
sper_fech_um		Date	<M>
sper_mo_oc		Variable characters (10)	<M>
PK_sis_permisos	<pi>		

Figura 4: Tabla sis\_permiso

**Historia de Usuario 3:** Registro del menú del sistema.

**Tarea 3.1:** Crear formulario de mantenimiento de menú.

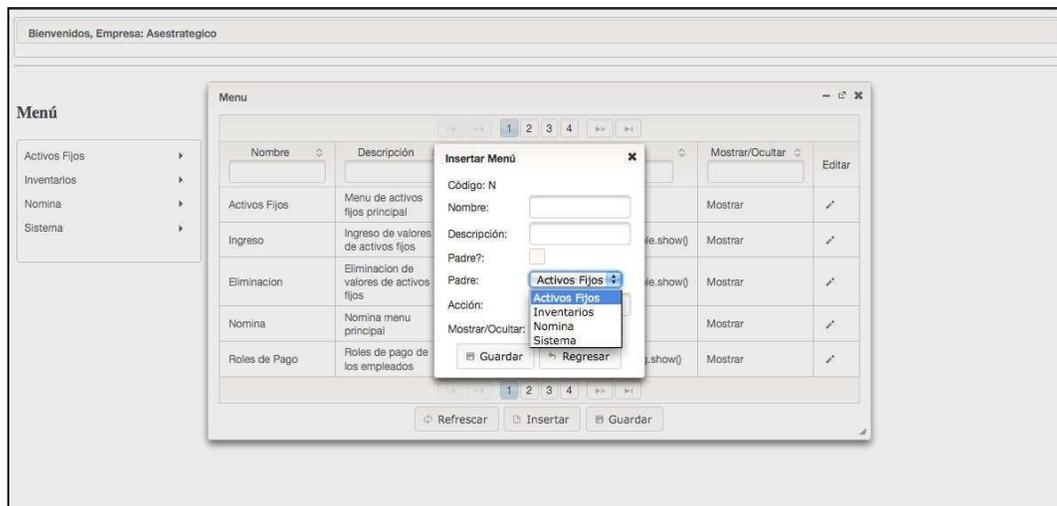


Figura 5: Formulario de mantenimiento de menú

**Tarea 3.2:** Crear base de datos para menú.

sis_menu			
<u>smen_cod</u>	<pi>	Variable characters (12)	<M>
smen_nomb		Variable characters (100)	<M>
smen_desc		Variable characters (200)	<M>
smen_fech_a		Date	<M>
smen_fech_um		Date	<M>
smen_accion		Variable characters (100)	<M>
smen_mo_oc		Variable characters (10)	<M>
PK_sis_menu	<pi>		

Figura 6: Tabla sis\_menu

**Historia de Usuario 4:** Registro de empresas que maneja el sistema.

**Tarea 4.1:** Crear formulario de mantenimiento de empresas.

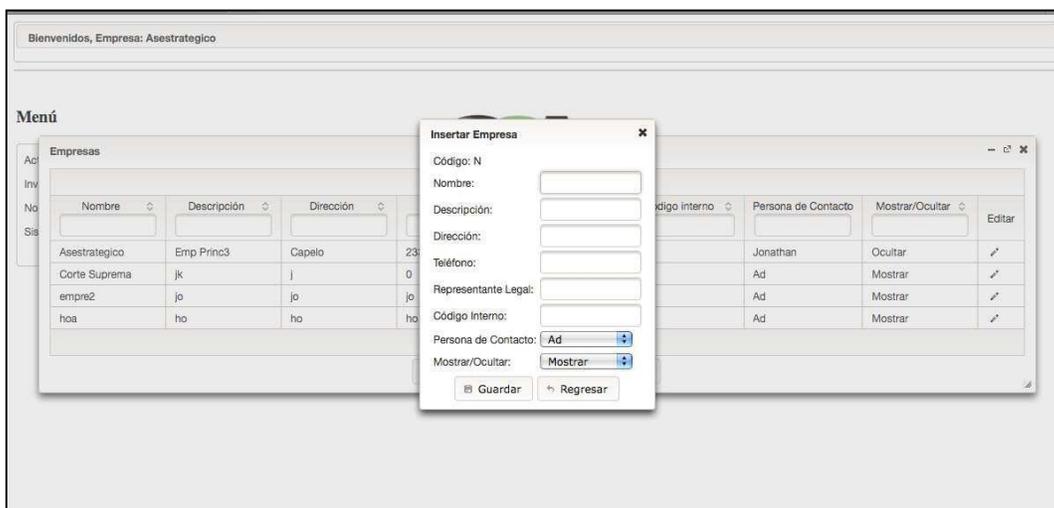


Figura 7: Formulario de mantenimiento de empresas

**Tarea 4.2: Crear formulario de mantenimiento de menú.**

sis_empresa			
<u>sem_cod</u>	<pi>	Variable characters (12)	<M>
sem_nomb		Variable characters (100)	<M>
sem_desc		Variable characters (200)	<M>
sem_direcc		Variable characters (150)	<M>
sem_telf		Variable characters (50)	<M>
sem_represen		Variable characters (100)	<M>
sem_cod_inter		Variable characters (5)	<M>
sem_fech_a		Date	<M>
sem_fech_um		Date	<M>
sem_mo_oc		Variable characters (10)	<M>
PK_sis_empresa	<pi>		

Figura 8: Tabla sis\_empresa

**Segunda iteración (Módulo Nómina).**

**Historia de Usuario 5:** Registro de personas empleadas en cada empresa.

**Tarea 5.1:** Crear formulario de mantenimiento de personas empleadas en las empresas.

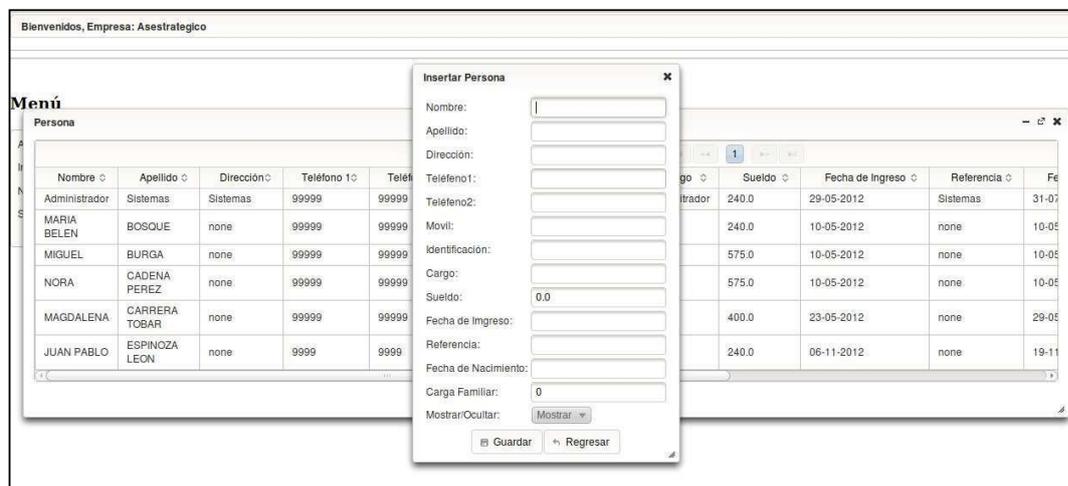


Figura 9: Formulario de mantenimientos de personas empleadas en las empresas

**Tarea 5.2:** Crear base de datos para empleados de cada empresa.

gen_persona			
<u>gper_codigo</u>	<pi>	Variable characters (12)	<M>
gper_nombre		Variable characters (100)	
gper_direccion		Variable characters (300)	
gper_telefono_1		Variable characters (15)	
gper_telefono_2		Variable characters (15)	
gper_movil		Variable characters (15)	
gper_identificacion		Variable characters (15)	
gper_fecha_ua		Date	
gper_fecha_um		Date	
gper_mo_oc		Variable characters (10)	
gper_img_grand		Long binary	
gper_img_peq		Long binary	
gper_img_recar		Variable characters (3)	
gper_apellido		Variable characters (100)	
gper_cargo		Variable characters (100)	
gper_sueldo		Decimal (10,2)	
gper_fech_ingre		Date	
gper_referenci		Variable characters (200)	
gper_fech_nacimi		Date	
gper_carg_famili		Integer	
PK_gen_persona	<pi>		

Figura 10: Tabla gen\_persona

**Historia de Usuario 6:** Registro de tipo de rubro.

**Tarea 6.1:** Crear formulario de mantenimiento de tipo de rubro.

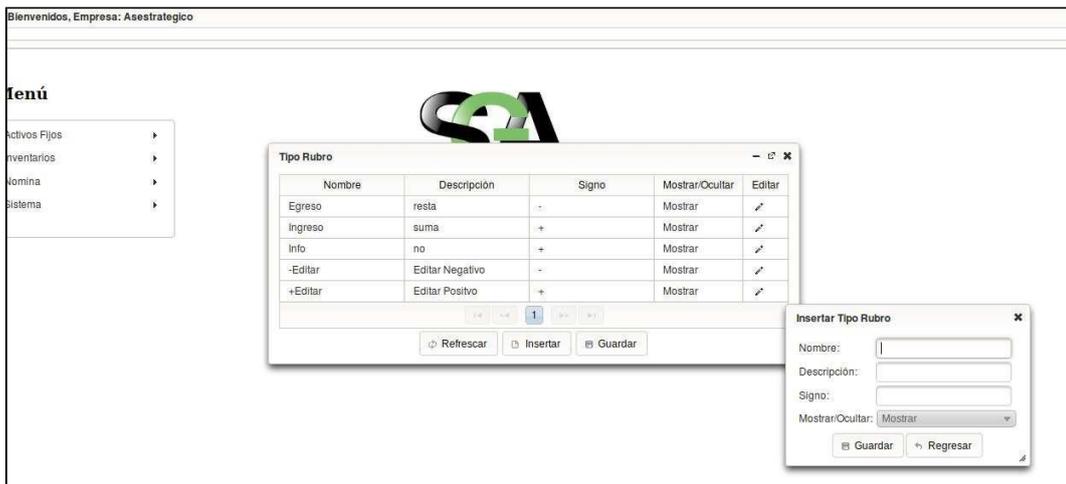


Figura 11: Formulario de mantenimiento de tipo de rubro

**Tarea 6.2: Crear base de datos de tipo de rubro.**

nom_tip_ru			
<u>ntir_cod</u>	<pi>	Variable characters (12)	<M>
ntir_nomb		Variable characters (100)	<M>
ntir_desc		Variable characters (200)	<M>
ntir_fech_ua		Date	<M>
ntir_fech_um		Date	<M>
ntir_mo_oc		Variable characters (50)	<M>
ntir_signo		Variable characters (5)	<M>
PK_nom_tip_ru	<pi>		

Figura 12: Tabla nom\_tip\_ru

**Historia de Usuario 7: Registro de rubros.**

**Tarea 7.1: Crear formulario de mantenimiento de rubros.**

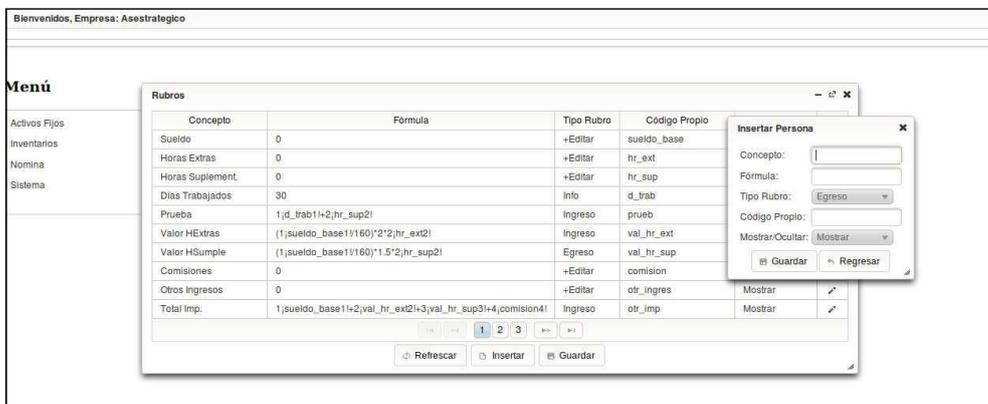


Figura 13: Formulario de mantenimiento de rubro

**Tarea 7.2: Crear base de datos para rubros.**

nom_rubros			
<u>nrub_cod</u>	<pi>	Variable characters (12)	<M>
nrub_concep		Variable characters (100)	<M>
nrub_formu		Variable characters (300)	<M>
nrub_cod_pro		Variable characters (100)	<M>
nrub_fech_ua		Date	<M>
nrub_fech_um		Date	<M>
nrub_mo_oc		Variable characters (50)	<M>
PK_nom_rubros	<pi>		

Figura 14: Tabla nom\_rubros

**Historia de Usuario 8: Administración de Roles de Pago.**

**Tarea 8.1: Crear formulario para Administración Roles de Pago.**

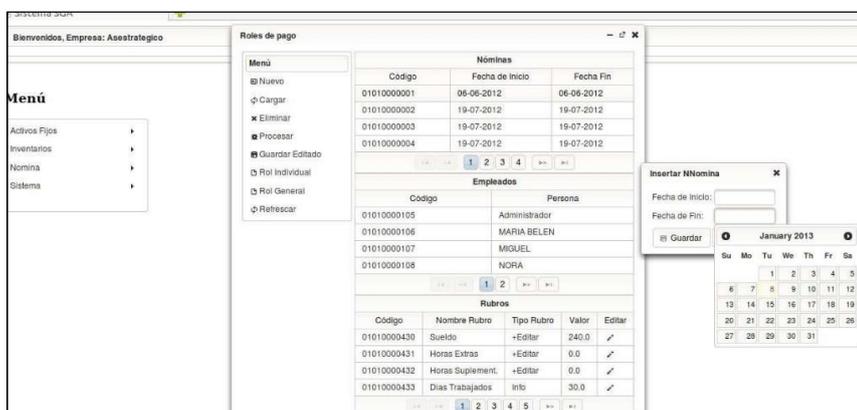


Figura 15: Formulario de administración de roles de pago

**Tarea 8.2: Crear base de datos para Administración Roles**

nom_n_rubro			
<u>nrru_cod</u>	<pi>	Variable characters (12)	<M>
nrru_valor		Decimal (20,4)	<M>
nrru_fech_ua		Date	<M>
nrru_fech_um		Date	<M>
nrru_mo_oc		Variable characters (15)	<M>
PK_nom_n_rubro	<pi>		

Figura 16: Tabla nom\_n\_rubro

nom_n_nomina	
+ nnnno_cod	: java.lang.String
+ nnnno_fech_ini	: java.util.Date
+ nnnno_fech_fin	: java.util.Date
+ nnnno_fech_ua	: java.util.Date
+ nnnno_fech_um	: java.util.Date
+ nnnno_mo_oc	: java.lang.String

Figura 17: Tabla nom\_n\_nomina

nom_n_employado	
+ nnem_cod	: java.lang.String
+ nnem_fech_ua	: java.util.Date
+ nnem_fech_um	: java.util.Date
+ nnem_mo_oc	: java.lang.String

Figura 18: Tabla nom\_n\_employado

### Tarea 8.3: Crear Reporte Rol Individual

**SA**  
**Sistema Gestor Administrativo**

## Rol de Pago

Individual

Rol de pagos individual por cada uno de los empleados de la empresa. Se detalla todos los rubros calculados por el sistema

Rubros	Valor
<b>NORA</b>	
Sueldo	575.0
Horas Extras	0.0
Horas Suplement.	0.0
Días Trabajados	30.0
Prueba	30.0
Valor HExtras	0.0
Valor HSumple	0.0
Comisiones	0.0
Otros Ingresos	0.0
Total Imp.	575.0
Fondo Reserva	47.92
Total Ingresos	622.92
Aporte Personal	53.76
Anticipo	0.0
Prestamo IESS	0.0
Salud y Uniforme	0.0
Prestamo Banco	0.0
Total Descuento	53.76

Monday 07 January 2013 Page 1 of 1

Figura 19: Reporte Rol Individual

### Tarea 8.4: Crear Reporte Rol General



Figura 22: Tabla inv\_bodega

**Historia de Usuario 10: Registro de producto.**

**Tarea 10.1: Crear formulario de mantenimiento de producto.**

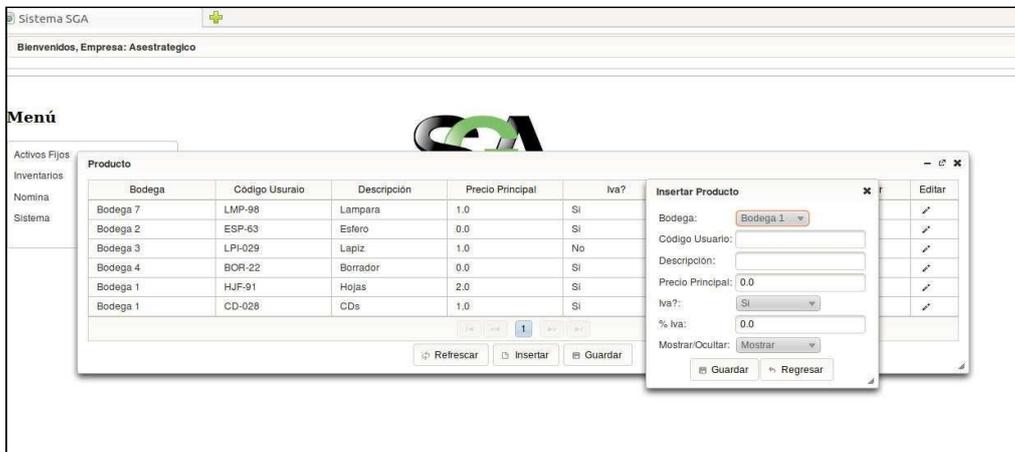


Figura4.23: Formulario de mantenimiento de producto

**Tarea 10.2: Crear base de datos para producto.**

inv_producto			
ipro_cod	<pi>	Variable characters (12)	<M>
ipro_cod_us		Variable characters (30)	<M>
ipro_desc		Variable characters (100)	<M>
ipro_p_prim		Long float	<M>
ipro_u_com		Long float	<M>
ipro_iva		Variable characters (5)	<M>
ipro_p_iva		Long float	<M>
ipro_mo_oc		Variable characters (10)	<M>
ipro_fech_a		Date	<M>
ipro_fech_m		Date	<M>
PK_inv_producto	<pi>		

Figura 4.24: Tabla inv\_producto

**Historia de Usuario 11: Registro de Movimientos de productos**

**Tarea 11.1: Crear formulario de movimientos de productos**

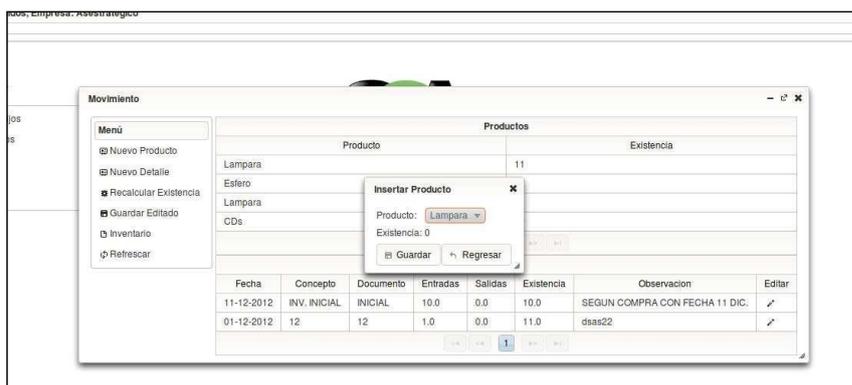


Figura.25: Formulario de movimiento de producto

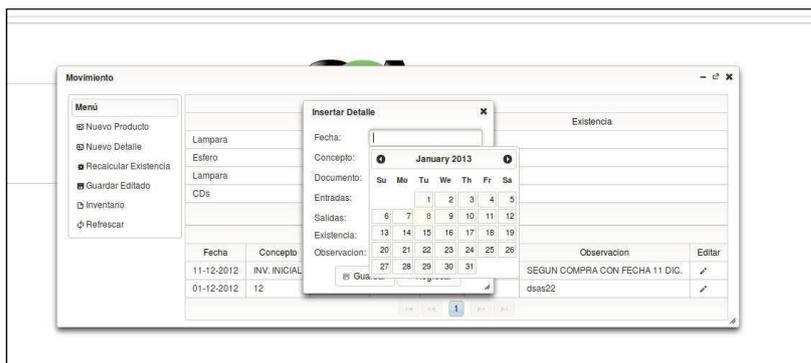


Figura 26: Formulario de detalle del producto

Tarea 11.2: Crear base de datos para registro de movimiento

inv_movi_det	
+ imod_cod	: java.lang.String
+ imod_fech_act	: java.util.Date
+ imod_concep	: java.lang.String
+ imod_docu	: java.lang.String
+ imod_entrad_c	: double
+ imod_salida_c	: double
+ imod_exist	: double
+ imod_observ	: java.lang.String
+ imod_fech_a	: java.util.Date
+ imod_fech_m	: java.util.Date

Figura 27: Tabla inv\_movi\_det

inv_movi_cab	
+ imoc_cod	: java.lang.String
+ imoc_exist	: int
+ imoc_fech_a	: java.util.Date
+ imoc_fech_m	: java.util.Date

Figura 28: Tabla inv\_movi\_det

Tarea 11.3: Crear Reporte Inventario de Productos

SA		Inventario		de Productos	
Sistema Gestor Administrativo		Información sobre la existencia de cada producto			
Nombre Producto	Código Producto	Existencia			
Lampara	LMP-98	11			
Esfero	ESP-83	12			
Lampara	LMP-98	1			
CDs	CD-028	10			

Monday 07 January 2013 Page 1 of 1

Figura 29: Reporte Inventario de Productos

### Cuarta iteración (Módulo Activos Fijos).

#### Historia de Usuario 12: Registro de Tipos de Activos Fijos

#### Tarea 12.1: Crear formulario de mantenimiento de activos fijos



Figura 30: Formulario de Mantenimiento de Activos Fijos

#### Tarea 12.2: Crear base de datos para registro de movimiento

acfi_tipo	
+ afti_cod	: java.lang.String
+ afti_desc	: java.lang.String
+ afti_porcent	: double
+ afti_fech_a	: java.util.Date
+ afti_fech_m	: java.util.Date

Figura 31: Tabla acfi\_tipo

#### Historia de Usuario 13: Registro de Activos Fijos

#### Tarea 13.1: Crear formulario de mantenimiento de activos fijos

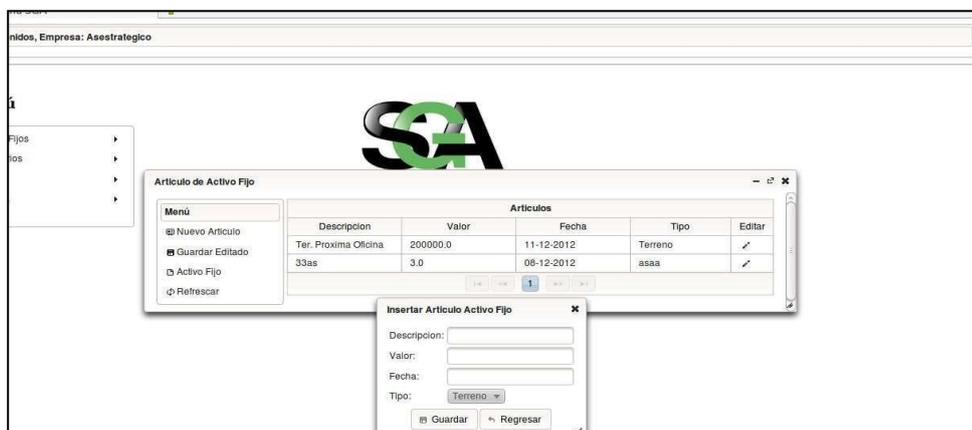


Figura 32: Formula de mantenimientos de Activos Fijos

### Tarea 13.2: Crear base de datos para registro de activos fijos



```
acti_articulo
+ afar_cod      : java.lang.String
+ afar_desc     : java.lang.String
+ afar_valor    : double
+ afar_fecha_ing : java.util.Date
+ afar_fech_a   : java.util.Date
+ afar_fech_m   : java.util.Date
```

Figura 33: Tabla acti\_articular

### Tarea 13.3: Crear Reporte de Activos Fijos



SA Sistema Gestor Administrativo		Activos Fijos Depreciación para los artículos de Activos Fijos	
Articulo	Valor		
Tar. Proxima Oficina	194000.0		
33as	2.67		

Thursday 03 January 2013 Page 1 of 1

Figura 34: Reporte de Activos Fijos

## PRINCIPALES CONCLUSIONES

A través de la presente propuesta de estudio de caso profesional se ha logrado automatizar los procesos de gestión administrativa de Activos Fijos, Inventarios y Nómina con la generación de sus respectivos reportes; ahorrando recursos como: tiempo, esfuerzo y dinero dentro de un software confiable, estable y de alto rendimiento con la tecnología que en la actualidad se encuentra al alcance.

De acuerdo con el análisis de las diferentes historias de usuario, se pueden planificar y realizar las mínimas inversiones del sistema según la metodología XP, sin embargo, este proyecto fue de un corto alcance, lo que genera una gran limitante al momento de realizar nuevas funciones sin mencionar también que las herramientas de desarrollo permiten esto. Restful da un aporte muy esencial y significativo en la construcción del aplicativo, convirtiéndose en el eje central de todo el desarrollo. Si se analiza la arquitectura del sistema, se tiene el mapeo de objetos de acuerdo a la base de datos representado en los POJOs siguiendo lo especificado por EJB3; y por otro lado se tiene la capa de usuario donde se construyó con la librería de componentes Primefaces. Ambos lados se comunican a través de XML.

En la construcción del sistema se tienen ejes fundamentales y atractivos por ambas partes; la primera parte está enfocada en el desarrollo, mencionando que la arquitectura que se empleó, permite realizar cambios de una forma en la que el desarrollador conoce donde va cada función, método o algoritmo y así poder reutilizar el código ya escrito; la segunda parte es la del usuario, gracias a la implementación de la librería de componentes Primefaces en la interface, muestra una solución muy visible, interactiva y fácil de manejar.

## RECOMENDACIONES

Utilizar herramientas de desarrollo reconocidas con versiones estables, que poseen suficiente documentación de apoyo, ya que si se utilizan versiones actuales suelen carecer de documentación y de estabilidad al momento de utilizarlas en el desarrollo del sistema.

Realizar interfaces de usuario amigables, siguiendo un estándar de diseño y utilizando la experiencia de usuario hace que este no sea muy difícil de manejar y de comprender para el usuario final.

Utilizar herramientas de software libre para evitar los altos costos de licenciamiento, además se tiene las mismas prestaciones que el software propietario y en algunos casos resulta más conveniente.

Implementar la integración de este sistema con otros aplicativos de la empresa que cuentan con módulos complementarios y que se encuentran de igual forma con los requerimientos que necesita el usuario dentro de un software estable y actual.

Implementar una interfaz móvil (ya que se cuenta con las capas de negocio donde se interactúa a través de RESTful y XML con la capa de presentación) donde estén funciones básicas como reportes y una que otra funcionalidad necesaria para el usuario, la cual permita disponibilidad de la aplicación en cualquier momento a través del internet.

## REFERENCIAS

- Canós, J. H., Letelier, P., & Penadés, M. C. (2003). Metodologías ágiles en el desarrollo de software. *Universidad Politécnica de Valencia, Valencia*, 1-8.
- Colina, Y. B. (2011). Aplicaciones de programación lineal, entera y mixta. *Ingeniería Industrial. Actualidad y Nuevas Tendencias*, 2(7), 85-104.
- Eberhart, A. (2002, June). Automatic generation of java/sql based inference engines from rdf schema and ruleml. In *International Semantic Web Conference* (pp. 102-116). Springer, Berlin, Heidelberg.
- Fu, Y., Ong, K. W., Papakonstantinou, Y., & Petropoulos, M. (2011, January). The SQL-based all-declarative FORWARD web application development framework. In *CIDR* (pp. 69-78).
- Gómez, O. T., López, P. P. R., & Bacalla, J. S. (2010). Criterios de selección de metodologías de desarrollo de software. *Industrial data*, 13(2), 70-74.
- Guerrero Posadas, M., Silva y Rodríguez García, B. E., Valdez Martínez, L. M., & Martínez Rodríguez, M. A. (2022). Desarrollo de un sistema de información web para evaluar proveedores de servicios de mantenimiento en una industria del ramo automotriz. *Sapienza: International Journal of Interdisciplinary Studies*, 3(1), 298-311. <https://doi.org/10.51798/sijis.v3i1.200>
- Jacobson, I., Booch, G., & Rumbaugh, J. (2000). *El proceso unificado de desarrollo de software* (Vol. 7). Madrid: Addison Wesley.
- Kieyzun, A., Guo, P. J., Jayaraman, K., & Ernst, M. D. (2009, May). Automatic creation of SQL injection and cross-site scripting attacks. In *2009 IEEE 31st international conference on software engineering* (pp. 199-209). IEEE.
- McAffer, J., VanderLei, P., & Archer, S. (2010). *OSGi and Equinox: Creating highly modular Java systems*. Addison-Wesley Professional.
- Ong, K. W. (2010). *Web application creation made easy: A SQL-driven rapid development framework and a do-it-yourself platform*. University of California, San Diego.
- Potter, K. J., & Sanvido, V. (1994). Design/build prequalification system. *Journal of Management in Engineering*, 10(2), 48-56.
- Richards, M., Monson-Haefel, R., & Chappell, D. A. (2009). *Java message service: creating distributed enterprise applications*. " O'Reilly Media, Inc."
- Winston, W. L., & Goldberg, J. B. (2005). *Investigación de operaciones: aplicaciones y algoritmos* (Vol. 4). Mexico, Mexico: Thomson.