



UNIVERSIDAD TÉCNICA DEL NORTE

FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES

**TRABAJO DE GRADO PREVIO A LA OBTENCIÓN DEL TÍTULO DE
INGENIERO EN SISTEMAS COMPUTACIONALES**

TEMA:

**SISTEMA DE GESTIÓN DE INVENTARIOS DEL GOBIERNO
PROVINCIAL DE IMBABURA MEDIANTE LA INTEGRACIÓN DE
TECNOLOGÍA "BUSINESS PROCESS MANAGEMENT (BPM)" Y "RICH
INTERNET APPLICATIONS (RIA)"**

AUTOR: CRISTIAN VICENTE IPIALES FLORES

DIRECTOR: ING. CARPIO PINEDA

IBARRA – ECUADOR

2016



UNIVERSIDAD TÉCNICA DEL NORTE

BIBLIOTECA UNIVERSITARIA

AUTORIZACIÓN DE USO Y PUBLICACIÓN

A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE

1. IDENTIFICACIÓN DE LA OBRA

La UNIVERSIDAD TÉCNICA DEL NORTE dentro del proyecto Repositorio Digital Institucional determina la necesidad de disponer los textos completos de forma digital con la finalidad de apoyar los procesos de investigación, docencia y extensión de la universidad.

Por medio del presente documento dejo sentada mi voluntad de participar en este proyecto, para lo cual pongo a disposición la siguiente información:

| DATOS DE CONTACTO | | | |
|-------------------------|--|-----------------|------------|
| CÉDULA DE IDENTIDAD: | 1002855268 | | |
| APELLIDOS Y NOMBRES: | CRISTIAN VICENTE IPIALES FLORES | | |
| DIRECCIÓN: | Ibarra, San Antonio, Barrio Tanguarín | | |
| EMAIL: | cristianipialesystem@gmail.com | | |
| TELÉFONO FIJO: | 062932765 | TELÉFONO MÓVIL: | 0997521140 |
| DATOS DE LA OBRA | | | |
| TÍTULO: | SISTEMA DE GESTIÓN DE INVENTARIOS DEL GPI MEDIANTE LA INTEGRACIÓN DE TECNOLOGÍA "BUSINESS PROCESS MANAGEMENT (BPM)" Y "RICH INTERNET APPLICATIONS (RIA)" | | |
| AUTOR: | CRISTIAN VICENTE IPIALES FLORES | | |
| FECHA: | 2016-09-26 | | |
| PROGRAMA: | <input type="checkbox"/> PREGRADO <input type="checkbox"/> POSTGRADO | | |
| TÍTULO POR EL QUE OPTA: | INGENIERO EN SISTEMAS COMPUTACIONALES | | |
| DIRECTOR: | ING. CARPIO PINEDA | | |

2. AUTORIZACIÓN DE USO A FAVOR DE LA UNIVERSIDAD

Yo, Cristian Vicente Ipiales Flores, con cédula de identidad Nro. 100285526-8, en calidad de autor (es) y titular (es) de los derechos patrimoniales de la obra o trabajo de grado descrito anteriormente, hago entrega del ejemplar respectivo en formato digital y autorizo a la Universidad Técnica del Norte, la publicación de la obra en el Repositorio Digital Institucional y uso del archivo digital en la Biblioteca de la Universidad con fines académicos, para ampliar la disponibilidad del material y como apoyo a la educación, investigación y extensión; en concordancia con la Ley de Educación Superior Artículo 144.

3. CONSTANCIAS

El autor (es) manifiesta (n) que la obra objeto de la presente autorización es original y se la desarrolló, sin violar derechos de autor de terceros, por lo tanto la obra es original y que es (son) el (los) titular (es) de los derechos patrimoniales, por lo que asume (n) la responsabilidad sobre el contenido de la misma y saldrá (n) en defensa de la Universidad en caso de reclamación por parte de terceros.

Ibarra, Septiembre de 2016

EL AUTOR:

ACEPTACIÓN:

(Firma)..........

Nombre: Cristian Ipiales Flores

C.C.: 1002855268

x(Firma)..........

Nombre: GLADYS ARAUZ

Cargo: JEFE DE BIBLIOTECA

Facultado por resolución de Consejo Universitario



UNIVERSIDAD TÉCNICA DEL NORTE

FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE GRADO A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE

Yo, Cristian Vicente Ipiates Flores, con cédula de identidad Nro. 100285526-8, manifiesto mi voluntad de ceder a la Universidad Técnica del Norte los derechos patrimoniales consagrados en la Ley de Propiedad Intelectual del Ecuador, artículos 4, 5 y 6, en calidad de autor (es) de la obra o trabajo de grado denominado: **“SISTEMA DE GESTIÓN DE INVENTARIOS DEL GPI MEDIANTE LA INTEGRACIÓN DE TECNOLOGÍA BUSINESS PROCESS MANAGEMENT (BPM) Y RICH INTERNET APPLICATIONS (RIA)”**, que ha sido desarrollado para optar por el título de: Ingeniero en Sistemas Computacionales en la Universidad Técnica del Norte, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente. En mi condición de autor me reservo los derechos morales de la obra antes citada. En concordancia suscribo este documento en el momento que hago entrega del trabajo final en formato impreso y digital a la Biblioteca de la Universidad Técnica del Norte.

Firma -----

Nombre: Cristian Ipiates Flores

Cédula: 1002855268

Ibarra, Septiembre del 2016



GOBIERNO PROVINCIAL DE IMBABURA

Ibarra, 01 Junio 2016

Ing. Francisco Arteaga, Director de Tecnologías de la Información del Gobierno Provincial de Imbabura.

CERTIFICA:

Que, siendo auspiciante del proyecto de tesis del egresado CRISTIAN VICENTE IPIALES FLORES con CI: 100285526-8 quien desarrolló su trabajo en el tema "**SISTEMA DE GESTIÓN DE INVENTARIOS DEL GPI MEDIANTE LA INTEGRACIÓN TECNOLÓGICA BUSINESS PROCESS MANAGEMENT Y RICH INTERNET APPLICATIONS (RIA)**", me es grato informar que se ha superado a satisfacción las pruebas técnicas y la revisión del cumplimiento de los requerimientos funcionales, por lo que se recibe el proyecto como culminado por parte del egresado: **CRISTIAN VICENTE IPIALES FLORES**.

El egresado **CRISTIAN VICENTE IPIALES FLORES** puede hacer uso de este documento para los fines pertinentes en la Universidad Técnica Del Norte

Con sentimiento de distinguida consideración.

Atentamente:


Ing. Francisco Arteaga
DIRECTOR GENERAL DE TECNOLOGÍAS DE LA INFORMACIÓN



IBARRA: Bolívar Y Oviedo, esq. Telfs.: (593 6) 295 5225, 295 5832, 295 0939, Fax.: (593 6) 295 5430
email: gpi@imbabura.gob.ec / www.imbabura.gob.ec



UNIVERSIDAD TÉCNICA DEL NORTE

FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

CERTIFICACIÓN DEL ASESOR

Certifico que la tesis previa a la obtención del título de Ingeniero en Sistemas Computacionales con el tema **“SISTEMA DE GESTIÓN DE INVENTARIOS DEL GPI MEDIANTE LA INTEGRACIÓN DE TECNOLOGÍA BUSINESS PROCESS MANAGEMENT (BPM) Y RICH INTERNET APPLICATIONS (RIA)”**, ha sido desarrollada y terminada en su totalidad por el Señor Cristian Vicente IpiALES Flores con CI. 1002855268 bajo mi supervisión para lo cual firmo en constancia.

A handwritten signature in blue ink, appearing to read "Carpio Pineda", is written over a horizontal line.

Ing. Carpio Pineda

DIRECTOR DE TESIS



UNIVERSIDAD TÉCNICA DEL NORTE
FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

DEDICATORIA

A Dios, por haberme dado la vida, y la oportunidad de seguir creciendo mentalmente, y poner siempre a las personas indicadas en el transcurrir de mi vida.

A mis padres Ing. Luis Ipiales y Cecilia Flores, quienes con su ejemplo de lucha y de vida han reflejado en mí sus buenos valores, por haberme guiado por el camino correcto, por la fuerza que me han dado para seguir adelante y sobre todo por darme la oportunidad de ser un profesional.

A mis hermanas Sandra, Anabel, a mis Tíos por el apoyo moral y sentimental que me han dado durante el desarrollo de este proyecto.

A mis amigos/as y compañeros/as, quienes estuvieron en las buenas y en las malas durante mi carrera universitaria compartiendo conocimientos, valores y alegrías.

A mis maestros, quienes me brindaron sus conocimientos a lo largo de mi vida estudiantil.

Cristian...



UNIVERSIDAD TÉCNICA DEL NORTE
FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

AGRADECIMIENTOS

Al ING. CARPIO PINEDA por su orientación y conocimientos otorgados para el desarrollo del presente proyecto.

A todas las personas encargadas del proyecto Sistema de Información Provincial “Gestión GPI” por ser la ayuda indispensable para el desarrollo del presente proyecto.

A mi familia, porque siempre me han apoyado, aconsejado y brindado todo el cariño que ha sido fundamental en mi vida, familia este logro es de todos.

A mis profesores, amigos y compañeros por todos los buenos momentos que vivimos y compartimos durante nuestra vida universitaria.

A todas aquellas personas que fueron parte y pieza fundamental para la culminación de este proyecto.

Cristian...

RESUMEN

El presente proyecto se desarrolló para proveer al Gobierno Provincial de Imbabura de un Sistema Integrado de Información como aporte al eficaz desempeño de las actividades de la institución, por lo cual se desarrolló el Sistema de Gestión de Inventarios mediante la integración de tecnología "Business Process Management (BPM)" y "Rich Internet Applications (RIA)".

Hoy en día las aplicaciones Web se destacan por la facilidad de uso y portabilidad. El hecho que cada día más personas están acostumbradas a la navegación por internet hace que el tiempo de aprendizaje se reduzca considerablemente respecto a aplicaciones tradicionales razón por la cual se decidió implementar el sistema web de Gestión de Inventarios, aplicando herramientas y estándares de código abierto.

Para el desarrollo e implementación del Sistema Web se utilizó PostgreSQL como sistema de gestión de bases de datos relacional orientado a objetos, tecnologías "Rich Internet Applications (RIA)" y "Business Process Management (BPM)" y frameworks de código libre.

La metodología utilizada para el análisis, implementación y documentación es RUP proceso de desarrollo de software el cual utiliza el lenguaje unificado de modelado UML, además de SCRUM proceso en el que se aplican de manera regular un conjunto de buenas prácticas para trabajar en equipo, además de mantener un orden en las actividades y desarrollo del proyecto.

La implementación y desarrollo del presente proyecto se realizó gracias a la ayuda e información pertinente proporcionada por cada uno de los responsables de los diferentes departamentos del GPI en especial la dirección de tecnologías de la información.

SUMMARY

This project was developed to provide the Provincial Government of Imbabura an Integrated Information System as a contribution to the effective performance of the activities of the institution, so the System Inventory Management was developed by integrating technology, "Business Process Management (BPM)" and "Rich Internet Applications (RIA)".

Today's Web applications are characterized by ease of use and portability. The fact that more and more people are accustomed to surfing the internet makes learning time is considerably reduced compared to traditional applications why it was decided to implement the web system Inventory Management, using tools and open source standards.

Development and implementation of Web System was used as PostgreSQL database management system relational database, object-oriented technologies "Rich Internet Applications (RIA)" and "Business Process Management (BPM)" and open source frameworks.

The methodology used for analysis, implementation and documentation is RUP development process software which uses the Unified Modeling UML language besides SCRUM process that apply regularly a set of best practices for teamwork as well to maintain order in activities and project development.

ÍNDICE DE CONTENIDO

| | |
|--|------|
| AUTORIZACIÓN DE USO Y PUBLICACIÓN | II |
| CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE GRADO A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE..... | IV |
| CERTIFICACIÓN DEL ASESOR | VI |
| DEDICATORIA | VII |
| AGRADECIMIENTOS..... | VIII |
| RESUMEN..... | IX |
| SUMMARY | X |
| ÍNDICE DE CONTENIDO | XI |
| ÍNDICE DE FIGURAS..... | XVI |
| ÍNDICE DE TABLAS..... | XIX |
| CAPÍTULO I..... | 1 |
| 1 INTRODUCCIÓN..... | 1 |
| 1.1 ANTECEDENTES..... | 1 |
| 1.1.1 BASE LEGAL DE LA INSTITUCIÓN..... | 1 |
| 1.1.2 OBJETIVOS DEL GOBIERNO PROVINCIAL DE IMBABURA | 2 |
| 1.1.3 MISIÓN INSTITUCIONAL | 2 |
| 1.1.4 VISIÓN INSTITUCIONAL..... | 2 |
| 1.1.5 ESTRUCTURA ORGANIZACIONAL..... | 2 |
| 1.2 SITUACIÓN ACTUAL | 4 |
| 1.3 PROBLEMA..... | 4 |
| 1.4 OBJETIVOS DEL PROYECTO | 5 |
| 1.4.1 OBJETIVO GENERAL | 5 |
| 1.4.2 OBJETIVOS ESPECÍFICOS..... | 5 |
| 1.5 ALCANCE..... | 6 |

| | |
|---|----|
| 1.6 JUSTIFICACIÓN..... | 7 |
| 1.6.1 JUSTIFICACIÓN TECNOLÓGICA | 8 |
| 1.6.2 JUSTIFICACIÓN METODOLÓGICA..... | 9 |
| 1.6.3 ANÁLISIS DE HERRAMIENTAS..... | 10 |
| CAPÍTULO II..... | 11 |
| 2 MARCO TEÓRICO | 11 |
| 2.1 BPM (BUSINESS PROCESS MANAGEMENT) | 11 |
| 2.2 TECNOLOGÍA RIA | 12 |
| 2.3 HERRAMIENTAS Y ESTÁNDARES DE CÓDIGO LIBRE | 13 |
| 2.3.1 INTRODUCCIÓN..... | 13 |
| 2.3.2 SISTEMAS OPERATIVOS (SO) | 13 |
| 2.3.3 SERVIDOR DE APLICACIONES | 14 |
| 2.3.4 BASES DE DATOS. POSTGRESQL | 15 |
| 2.3.5 PHP | 17 |
| 2.3.6 ACTIONSCRIPT | 18 |
| 2.3.7 MXML (MINIMAL XML) | 18 |
| 2.3.8 CSS (CASCADING STYLE SHEETS)..... | 21 |
| 2.4 SEGURIDAD EN UN PORTAL WEB | 22 |
| 2.5 PLATAFORMAS Y FRAMEWORKS | 23 |
| 2.5.1 ADOBE FLASH BUILDER..... | 24 |
| 2.5.2 ADOBE FLEX | 24 |
| 2.5.3 FLASH PLAYER | 25 |
| 2.5.4 ZEND FRAMEWORK..... | 26 |
| 2.5.5 TIBCO BUSINESS STUDIO..... | 27 |
| 2.5.6 TOAD DATA MODELER..... | 28 |
| 2.5.7 PGADMIN III | 29 |
| 2.5.8 NAVICAT | 30 |

| | |
|--|----|
| 2.5.9 PROCESSMAKER..... | 31 |
| CAPÍTULO III..... | 34 |
| 3 MÓDULOS..... | 34 |
| 3.1 FUNCIONAMIENTO Y DESCRIPCIÓN DEL SISTEMA | 34 |
| 3.1.1 VISTA GENERAL | 34 |
| 3.1.2 CARACTERÍSTICAS DEL SISTEMA | 35 |
| 3.1.3 DEPENDENCIAS PARA LA IMPLEMENTACIÓN | 35 |
| 3.1.4 FUNCIONAMIENTO. | 37 |
| 3.1.5 DIAGRAMA ENTIDAD RELACIÓN | 42 |
| 3.2 DESCRIPCIÓN Y FUNCIONAMIENTO DE LOS MÓDULOS..... | 44 |
| 3.2.1 DESCRIPCIÓN DEL MÓDULO DE SISTEMA | 44 |
| 3.2.2 DESCRIPCIÓN DEL MÓDULO DE CATÁLOGO | 47 |
| 3.2.3 FUNCIONAMIENTO DEL MÓDULO DE MOVIMIENTO DE INVENTARIOS..... | 48 |
| CAPÍTULO IV | 50 |
| 4 DISEÑO Y DESARROLLO DEL APLICATIVO..... | 50 |
| 4.1 INTRODUCCIÓN | 50 |
| 4.1.1 METODOLOGÍAS ÁGILES | 50 |
| 4.1.2 MODELADO ÁGIL | 51 |
| 4.1.3 ASPECTOS DE LAS METODOLOGÍAS ÁGILES | 52 |
| 4.2 METODOLOGÍA SCRUM | 53 |
| 4.2.1 INTRODUCCIÓN..... | 53 |
| 4.2.2 CARACTERÍSTICAS | 53 |
| 4.2.3 ROLES DEL SCRUM..... | 55 |
| 4.2.4 IMPLEMENTACIÓN DE SCRUM..... | 55 |
| 4.2.5 VENTAJAS DEL SCRUM | 56 |
| 4.2.6 HISTORIAS DE USUARIO (HERRAMIENTA DE LAS METODOLOGÍAS ÁGILES) | 56 |

| | |
|--|-----|
| 4.2.7 SPRINT..... | 57 |
| 4.2.8 POILA DE PRODUCTOS (BACKLOG) | 58 |
| 4.2.9 DOCUMENTACIÓN GENERADA METODOLOGÍA SCRUM | 59 |
| 4.3 METODOLOGÍA RUP..... | 70 |
| 4.3.1 FASE DE INCEPCIÓN..... | 70 |
| 4.3.1.1 DOCUMENTO DE VISIÓN..... | 70 |
| 4.3.1.2 PLAN DE DESARROLLO DE SOFTWARE | 81 |
| 4.3.2 FASE DE ELABORACIÓN Y CONSTRUCCIÓN..... | 86 |
| 4.3.2.1 DOCUMENTO DE ARQUITECTURA..... | 86 |
| 4.3.2.2 VISTA DE CASOS DE USO..... | 88 |
| 4.3.2.3 VISTA LÓGICA..... | 102 |
| 4.3.2.4 VISTA DE LÓGICA..... | 106 |
| 4.3.2.5 VISTA DE PROCESOS BPM..... | 112 |
| 4.3.2.6 VISTA DE DESPLIEGUE | 118 |
| 4.3.2.7 TAMAÑO Y PERFORMANCE..... | 120 |
| 4.3.2.8 CALIDAD | 120 |
| 4.3.2.9 MANEJO DE ERRORES..... | 121 |
| 4.3.2.10 PLAN DE PRUEBAS..... | 122 |
| 4.3.3 FASE DE TRANSICIÓN..... | 130 |
| 4.3.3.1 MANUAL DE INSTALACIÓN | 130 |
| CAPÍTULO V | 156 |
| 5 CONCLUSIONES Y RECOMENDACIONES | 156 |
| 5.1 CONCLUSIONES | 156 |
| 5.2 RECOMENDACIONES..... | 157 |
| GLOSARIO..... | 159 |
| REFERENCIAS | 162 |
| ANEXOS..... | 167 |

| | |
|---|-----|
| ANEXO A: MANUAL DE USUARIO SE ENCUENTRA ADJUNTO EN EL CD..... | 167 |
| ANEXO B: MANUAL TÉCNICO SE ENCUENTRA ADJUNTO EN EL CD..... | 167 |

ÍNDICE DE FIGURAS

| | |
|---|----|
| FIGURA 1: Estructura Organizacional GPI | 3 |
| FIGURA 2: Módulos GPI_INVENTARIOS | 6 |
| FIGURA 3: Metodología SCRUM | 9 |
| FIGURA 4: Ciclo de Vida BPM | 11 |
| FIGURA 5: Aplicaciones Rich Internet Applications | 13 |
| FIGURA 6: Flujo de Datos entre las Diferentes Herramientas y Frameworks | 13 |
| FIGURA 7: Funcionamiento de Apache..... | 14 |
| FIGURA 8: Ejemplo Código ActionScript 3.0 | 18 |
| FIGURA 9: Estructura de un documento MXML | 19 |
| FIGURA 10: Ejemplo CSS..... | 21 |
| FIGURA 11: Ejemplo de diagrama de proceso | 28 |
| FIGURA 12: Toad Data Modeler..... | 29 |
| FIGURA 13: PGADMIN III | 30 |
| FIGURA 14: Navicat. Programa de mantenimiento y consulta de bases de datos. | 30 |
| FIGURA 15: Generar PM. | 31 |
| FIGURA 16: Ejecutar PM. | 32 |
| FIGURA 17: Reportar PM..... | 32 |
| FIGURA 18: Optimizar PM. | 33 |
| FIGURA 19: Diseñador de los macro-procesos..... | 33 |
| FIGURA 20: Módulos GPIINVENTARIOS..... | 34 |
| FIGURA 21: Funcionamiento General del Sistema..... | 38 |
| FIGURA 22: ARQUITECTURA GPI_GESTION..... | 39 |
| FIGURA 23: Arquitectura MVC+S | 40 |
| FIGURA 24: Arquitectura FLEX y PHP | 40 |
| FIGURA 25: Flujo de Datos entre Flex y PHP | 41 |
| FIGURA 26: Diagrama General de las Historias de Usuario..... | 41 |

| | |
|---|-----|
| FIGURA 27: Diagrama entidad Relación | 43 |
| FIGURA 28: Proceso de trabajo del Scrum | 54 |
| FIGURA 29: Sprint..... | 58 |
| FIGURA 30: Pila de Producto | 59 |
| FIGURA 31: Diagrama de Procesos de Gestión de Inventarios..... | 75 |
| FIGURA 32: Diagrama de flujo de datos del Sistema de Gestión de Inventarios | 76 |
| FIGURA 33: Diagrama de Transición de Estados del Sistema de Gestión de Inventarios | 77 |
| FIGURA 34: Diagrama Hipo Inventarios | 78 |
| FIGURA 35: Diagrama de Flujo y Procesos de Inventario (Azul) y la relación con Compras (Naranja) | 79 |
| FIGURA 36: Diagrama de Flujo del Proceso de Control de Inventarios | 79 |
| FIGURA 37: Diagrama de Flujo para la baja de Bienes por Obsolescencia y/o Daños. | 80 |
| FIGURA 38: Curso, dinámica, funcionamiento e integración con otros módulos. | 80 |
| FIGURA 39: Modelo 4+1 RUP | 87 |
| FIGURA 40: Vista de casos de uso de Gpi_inventarios módulo CU Catálogo Inventario..... | 89 |
| FIGURA 41: Vista de casos de uso de Gpi_inventarios módulo CU Movimientos de Inventario..... | 94 |
| FIGURA 42: Principales Tablas del Sistema..... | 103 |
| FIGURA 43: Diagrama entidad Relación Módulo Catálogo de Inventario | 104 |
| FIGURA 44: Diagrama entidad Relación Módulo Movimientos de Inventario | 105 |
| FIGURA 45: Subdivisión de módulos..... | 106 |
| FIGURA 46: Paquetes de clases de sistema..... | 108 |
| FIGURA 47: Estructura MVC | 109 |
| FIGURA 48: Diseño de Capas..... | 109 |
| FIGURA 49: Flex (MXML, AS3)..... | 110 |
| FIGURA 50: Servicios Flex..... | 111 |

| | |
|---|-----|
| FIGURA 51: Vista Principal TIBCO..... | 113 |
| FIGURA 52: Diagrama BPM Gestión de Inventarios Procesos..... | 114 |
| FIGURA 53: Diagrama BPM Creación Ítem de Inventario Proceso..... | 114 |
| FIGURA 54: Diagrama BPM Generar Registro de Bienes Proceso | 115 |
| FIGURA 55: Diagrama BPM Traslferencia Ítems Proceso..... | 115 |
| FIGURA 56: Diagrama BPM Egreso de Bienes Proceso | 116 |
| FIGURA 57: Diagrama BPM Costeo Inventario Proceso | 116 |
| FIGURA 58: Diagrama BPM Control de Inventarios Proceso | 117 |
| FIGURA 59: Diagrama BPM Toma de Inventario Proceso..... | 117 |
| FIGURA 60: Diagrama BPM Registro Puntos de Reorden Proceso | 118 |
| FIGURA 61: Diagrama BPM Control Puntos de Reorden Proceso | 118 |
| FIGURA 62: Vista de Despliegue | 119 |
| FIGURA 63: Inicio de Instalación Apache..... | 134 |
| FIGURA 64: Configurando PHP | 138 |
| FIGURA 65: Inicio de Instalación PostgreSQL..... | 140 |
| FIGURA 66: Inicio Restaurar BDD..... | 144 |
| FIGURA 67: Página de inicio del sistema. | 153 |
| FIGURA 68: Menú de inicio del sistema. | 153 |
| FIGURA 69: Gestión de Ítems de Inventario..... | 154 |
| FIGURA 70 Clasificación de Productos. | 154 |
| FIGURA 71: Movimientos de Inventario..... | 155 |
| FIGURA 72: Datos Estadísticos de Ítems de Inventario..... | 155 |

ÍNDICE DE TABLAS

| | |
|---|----|
| TABLA 1: Características del Producto GPI_INVENTARIOS | 8 |
| TABLA 2: Límites de PostgreSQL..... | 17 |
| TABLA 3: Características del Sistema..... | 35 |
| TABLA 4: Funcionalidad de Módulo de Administración..... | 45 |
| TABLA 5: Funcionamiento Módulo de Reportes..... | 45 |
| TABLA 6: Funcionamiento Módulo de Catálogos..... | 47 |
| TABLA 7: Funcionamiento del módulo de Movimientos de Inventarios | 49 |
| TABLA 8: H1: Definición del modelo de negocio y plan de proyecto | 59 |
| TABLA 9: H2: Definición del modelo de negocio y plan de proyecto | 60 |
| TABLA 10: H3: Desarrollar el documento de Arquitectura..... | 60 |
| TABLA 11: H4: Implementar y probar los casos de uso. Iteración 1..... | 60 |
| TABLA 12: H5: Implementar y probar los casos de uso. Iteración 2..... | 61 |
| TABLA 13: H6: Implementar y probar los casos de uso. Iteración 3..... | 61 |
| TABLA 14: H7: Ejecutar la Aplicación y hacer prueba de Errores | 61 |
| TABLA 15: Pila de Producto GPI_INVENTARIOS | 62 |
| TABLA 16: Pila de Sprint 00 Requerimientos Iniciales | 63 |
| TABLA 17: Pila de Sprint 01 Instalación y análisis | 64 |
| TABLA 18: Pila de Sprint 02 Estructura | 64 |
| TABLA 19: Pila de Sprint 03 Documentación | 65 |
| TABLA 20: Pila de Sprint 04 Metodología | 65 |
| TABLA 21: Pila de Sprint 05 Funcionalidad..... | 66 |
| TABLA 22: Pila de Sprint 06 Categoría de Productos | 66 |
| TABLA 23: Pila de Sprint 07 Productos..... | 67 |
| TABLA 24: Pila de Sprint 08 Tipos de Productos | 68 |
| TABLA 25: Pila de Sprint 09 Jerarquía de Productos..... | 68 |

| | |
|---|-----|
| TABLA 26: Pila de Sprint 10 Movimientos de Inventarios | 69 |
| TABLA 27: Definición del Problema | 71 |
| TABLA 28: Definición de la Posición del Producto | 72 |
| TABLA 29: Resumen de Stakeholders..... | 73 |
| TABLA 30: Resumen de Usuarios | 73 |
| TABLA 31: Beneficios que obtendrá el cliente | 74 |
| TABLA 32: Requerimientos..... | 81 |
| TABLA 33: Roles y Responsabilidades..... | 85 |
| TABLA 34: CU Catálogo Inventario..... | 89 |
| TABLA 35: CU Movimientos de Inventarios | 94 |
| TABLA 36: Pruebas de Integridad de Datos..... | 124 |
| TABLA 37: Pruebas del Sistema..... | 124 |
| TABLA 38: Pruebas del Ciclo de Negocio | 125 |
| TABLA 39: Pruebas de Interfaz de Usuario..... | 126 |
| TABLA 40: Pruebas de Desempeño | 127 |
| TABLA 41: Pruebas de Seguridad y Control de Acceso..... | 128 |
| TABLA 42: Recursos | 129 |
| TABLA 43: Entregables | 129 |
| TABLA 44: Lista de Riesgo | 129 |

CAPÍTULO I

1 INTRODUCCIÓN

En este capítulo se describen datos importantes del Gobierno Provincial de Imbabura (en adelante GPI), lo que permitirá conocer sus principales actividades y funciones.

1.1 ANTECEDENTES

1.1.1 BASE LEGAL DE LA INSTITUCIÓN.

CONSTITUCIÓN Y DOMICILIO.

“De acuerdo al código orgánico de organización territorial, autonomía y descentralización, los gobiernos autónomos descentralizados provinciales, son personas jurídicas de derecho público, con autonomía política, administrativa y financiera. Integrados por funciones de participación ciudadana, legislación y fiscalización ejecutiva, para el ejercicio de las funciones y competencias que le corresponden.”

FUNCIONES DEL GOBIERNO AUTÓNOMO DESCENTRALIZADO PROVINCIAL

La misión fundamental del Gobierno Provincial de Imbabura es impulsar el desarrollo cultural y material de la provincia de una manera equitativa y solidaria, y colaborar con el Estado y las Municipalidades de la respectiva circunscripción, para la realización armónica de los fines nacionales.

También fortalecer la participación ciudadana, la identidad pluricultural y las potencialidades socioeconómicas de la provincia; el desarrollo educativo y tecnológico que genere competitividad; la infraestructura y la oferta de bienes y servicios de calidad; el manejo sustentable de los recursos naturales; el cumplimiento de los deberes cívicos y el ejercicio de los derechos ciudadanos para alcanzar una alta calidad de vida y una sociedad segura y solidaria.

1.1.2 OBJETIVOS DEL GOBIERNO PROVINCIAL DE IMBABURA

Los objetivos del GPI relevantes para el presente proyecto son los siguientes:

- a) Promover una cultura organizacional de eficiencia y eficacia en la provincia de Imbabura.
- b) Incrementar la eficacia institucional, a través del mejoramiento del desempeño y resultados del talento humano y el adecuado uso de bienes tangibles e intangibles.
- c) Estandarizar los protocolos y mecanismos de información y comunicación institucional e interinstitucional
- d) Optimizar el uso de los recursos técnicos y tecnológicos y de la capacidad operativa del talento humano institucional.

1.1.3 MISIÓN INSTITUCIONAL

El Gobierno Provincial de Imbabura es la Institución encargada de coordinar, planificar, ejecutar y evaluar el Plan de Desarrollo Provincial Participativo; fortaleciendo la productividad, la vialidad, el manejo adecuado de sus recursos naturales y promoviendo la participación ciudadana, a fin de mejorar la calidad de vida de sus habitantes.

1.1.4 VISIÓN INSTITUCIONAL.

El Gobierno Provincial de Imbabura se consolida como una Institución de derecho público autónoma, descentralizada, transparente, eficiente, equitativa, incluyente y solidaria, líder del desarrollo económico, social y ambiental provincial.

1.1.5 ESTRUCTURA ORGANIZACIONAL

La estructura organizacional se derivará de los procesos requeridos para el cumplimiento de actividades y procedimientos, necesarios para el logro de los objetivos institucionales.

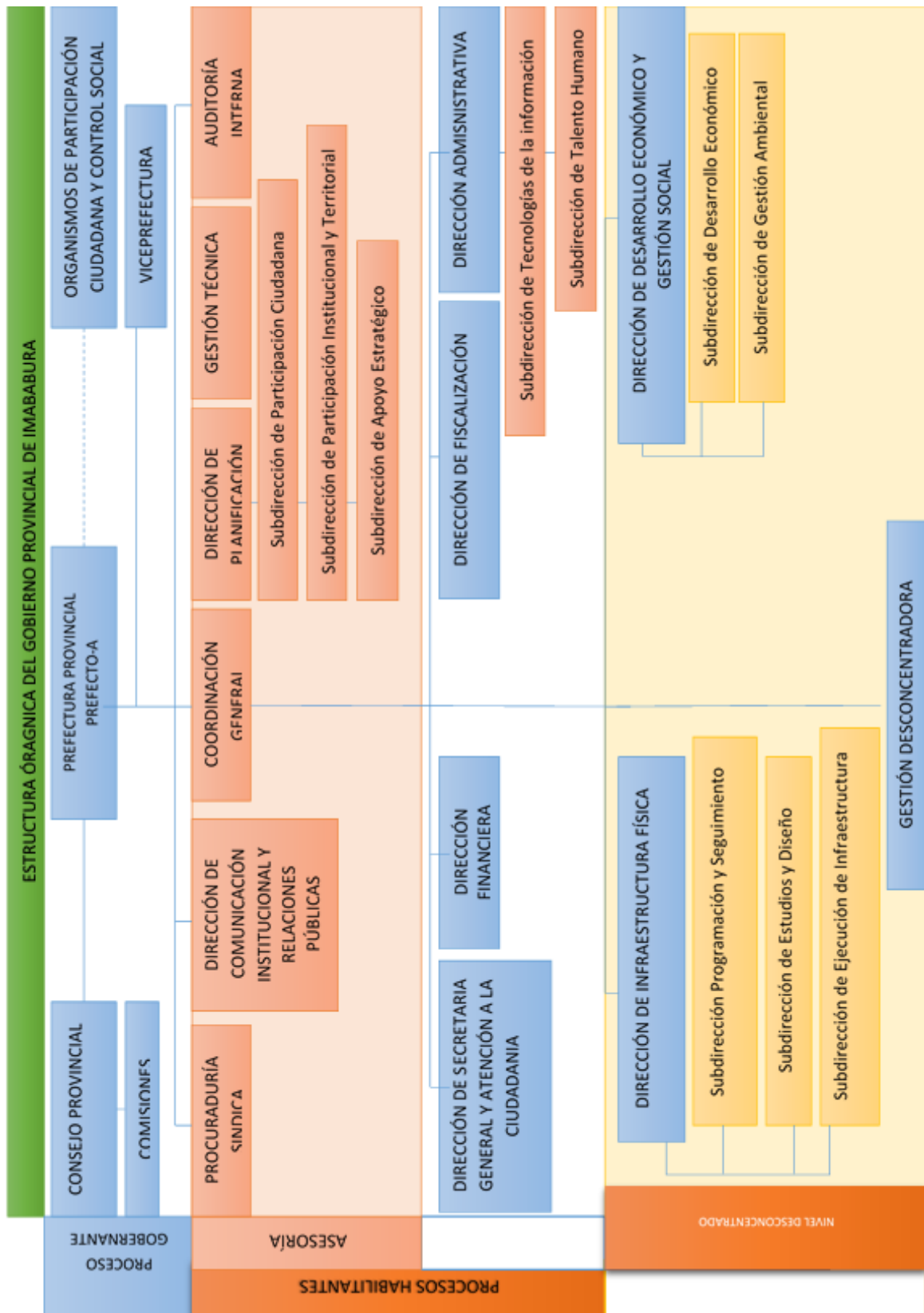


FIGURA 1: Estructura Organizacional GPI
Fuente: (GPI, Plan estratégico institucional, 2014)

1.2 SITUACIÓN ACTUAL

En el Gobierno Provincial de Imbabura se están ejecutando proyectos de mejora para optimizar los procesos, así como también metodologías que definan como se lleva a cabo todas las actividades de la Institución con la ayuda de la Gestión Administrativa.

La Gestión Administrativa consiste en un conjunto de decisiones y acciones que en forma continua se realizan por medio de la planificación, organización y coordinación, para brindar apoyo a los diferentes departamentos del Gobierno Provincial de Imbabura.

El Objetivo de la Gestión Administrativa es administrar los recursos materiales del Gobierno Provincial de Imbabura y dirigir la elaboración de planes, programas y proyectos necesarios para su desarrollo.

La Gestión Administrativa se enmarca en brindar soporte a las Unidades del GPI y asesorar a las autoridades en la toma de decisiones en materia administrativa.

1.3 PROBLEMA

El Gobierno Provincial de Imbabura como en toda institución de gobierno maneja una gran cantidad de suministros, materiales, equipos, maquinaria, etc., que requieren de una correcta administración y control.

Las diferentes reestructuraciones en el ámbito de la gestión pública en nuestro país han provocado que los gobiernos autónomos descentralizados generen una diversidad de proyectos para lograr los grandes objetivos provinciales.

Dichos proyectos requieren, sin duda alguna, de una eficiente gestión en lo que a los insumos, materiales, suministros, maquinaria y equipos se refiere, sin olvidar también los requerimientos internos de la administración general.

Para evitar dificultades en los proyectos y en los procesos internos, El Gobierno Provincial de Imbabura demanda de una adecuada administración del registro, compra y salida del inventario dentro de la institución, así el

Gobierno Provincial de Imbabura reducirá el costo de manejo de materiales y podrá planificar las compras. Se conseguirá realizar despachos “reales” a los departamentos no basados en cantidades supuestas.

Se mantendrá la información al día de los productos que deben ser eliminados de las existencias y que el departamento de contabilidad conozca el valor o capital inmovilizado que posee en sus bodegas. Con esto, la institución ahorrará recursos sin afectar el desempeño del Gobierno Provincial de Imbabura.

1.4 OBJETIVOS DEL PROYECTO

1.4.1 OBJETIVO GENERAL

Desarrollar el Sistema de Gestión de Inventarios, para el Gobierno Provincial de Imbabura utilizando tecnología BPM y RIA.

1.4.2 OBJETIVOS ESPECÍFICOS

- Diagnosticar la situación actual de los procesos de gestión de inventarios que se llevan a cabo en el GPI.
- Emplear las diferentes metodologías planteadas para la construcción del software.
- Definir los módulos del Sistema de Gestión de Inventarios.
- Construir la base de datos donde se alojará toda la información perteneciente al sistema.
- Diseñar formularios que permitan la correcta funcionalidad de la aplicación con los procedimientos señalados.
- Diseñar los diferentes reportes que generará el sistema.
- Probar el funcionamiento correcto del sistema antes de ser implantado.

1.5 ALCANCE

Diseñar, construir e implantar un sistema de información web que automatice la administración y control de inventarios en el GPI. Todo esto con la finalidad de generar acceso rápido a la información e integrar el proceso de control de inventarios a los procesos de planificación presupuestaria y planificación de compras, además se construirá la base de datos donde se alojará toda la información perteneciente al sistema y se desarrollará la interfaz gráfica del sistema enlazándolo a la base de datos para que la información pueda ser almacenada todo esto mediante el uso de tecnología BPM y RIA.

Módulos: Después de definir la BDD previo al análisis y definición de casos de uso, están el módulo de Sistema, módulo de Movimientos de Inventarios, módulo de Catálogo, módulo de Control de Stock y módulo de Reportes.

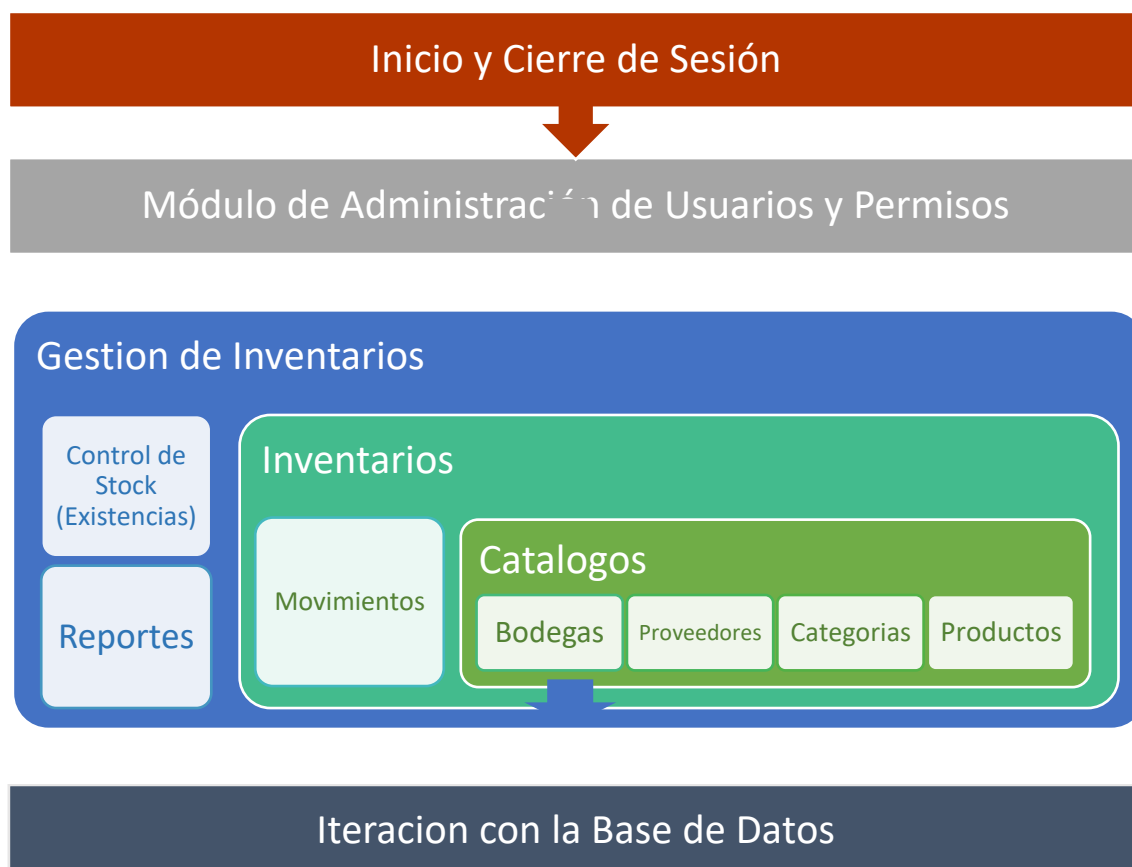


FIGURA 2: Módulos GPI_INVENTARIOS

Fuente: Propia

1.6 JUSTIFICACIÓN

La creciente complejidad de las organizaciones ha aumentado en gran medida la necesidad de tener información más conveniente y oportuna.

Actualmente las organizaciones, ya sean públicas o privadas, se han abocado a los avances tecnológicos de la época, los cuales tienden a convertirse en una herramienta importante para el desarrollo de las mismas, trayendo a favor un mejor desenvolvimiento del personal y rapidez en los procesos, lo que genera un mejor servicio.

Se desarrollará el Sistema de Gestión de Inventarios como parte del proyecto que está desarrollando el GPI denominado "GPI GESTIÓN", para registrar las actividades que se realizan en los diferentes procesos del departamento de Servicio General como es el registro de entradas y salidas del inventario de almacén, requisiciones de compras y/o servicios, solicitud y entrega de bienes, materiales y suministros.

Permitirá controlar el inventario del almacén, lo cual facilitará saber con qué recursos cuentan la empresa, que materiales se han entregado a los diferentes departamentos, con que material se dispone y que departamento tiene pedidos realizados y faltan por despachar.

Para esto el GPI, ha decidido desarrollar e implantar el Sistema de Gestión de Inventarios mediante el uso de tecnología BPM y RIA; para efecto de optimizar el registro, control y administración de los inventarios generados en el GPI.

El proyecto tendrá efectos favorables para la institución, dentro de los cuales se pueden mencionar

TABLA 1: Características del Producto GPI_INVENTARIOS

| Característica/producto | Descripción |
|---|---|
| Registro en línea de la toma física e identificación de diferencias | Ingreso en línea de cantidades físicas encontradas en el conteo físico. |
| Mejores tiempos de respuestas | Los usuarios podrán mejorar sus procesos al contar con un sistema informático. |
| Alta disponibilidad | Los usuarios tendrán acceso inmediato desde cualquier punto de la intranet del GPI. |
| Facilidades para el acceso a la información. | Brindará reportes de toda la información sobre los inventarios que se generan dentro del GPI. |
| Interfaz de usuario amigable | La Interfaz gráfica del sistema estará enlazado a la base de datos para que la información pueda ser almacenada y será de fácil acceso para el usuario. |
| Evaluación de reportes | Los diferentes reportes que generará el sistema serán entendibles y tendrán características gráficas. |

Fuente: Propia

1.6.1 JUSTIFICACIÓN TECNOLÓGICA

- BPM (Business Process Management)

BPM es una metodología y disciplina de gestión, cuyo objetivo es mejorar el desempeño y la optimización de los procesos que se deben diseñar, modelar, organizar, documentar y optimizar de forma continua.

- Tecnología RIA (Rich Internet Application)

Es un conjunto de aplicaciones web de características similares a las de las aplicaciones de escritorio tradicionales. Estas aplicaciones recurren a navegadores web estandarizados para ejecutarse por medio de complementos. (IBM, 2014)

1.6.2 JUSTIFICACIÓN METODOLÓGICA

▪ Metodología Scrum

Metodología Scrum aplicada en el desarrollo de software.

En Scrum un proyecto se ejecuta en tiempos cortos y fijos. Cada iteración proporciona un resultado completo o parcial que puede ser entregado al cliente cuando lo solicite.

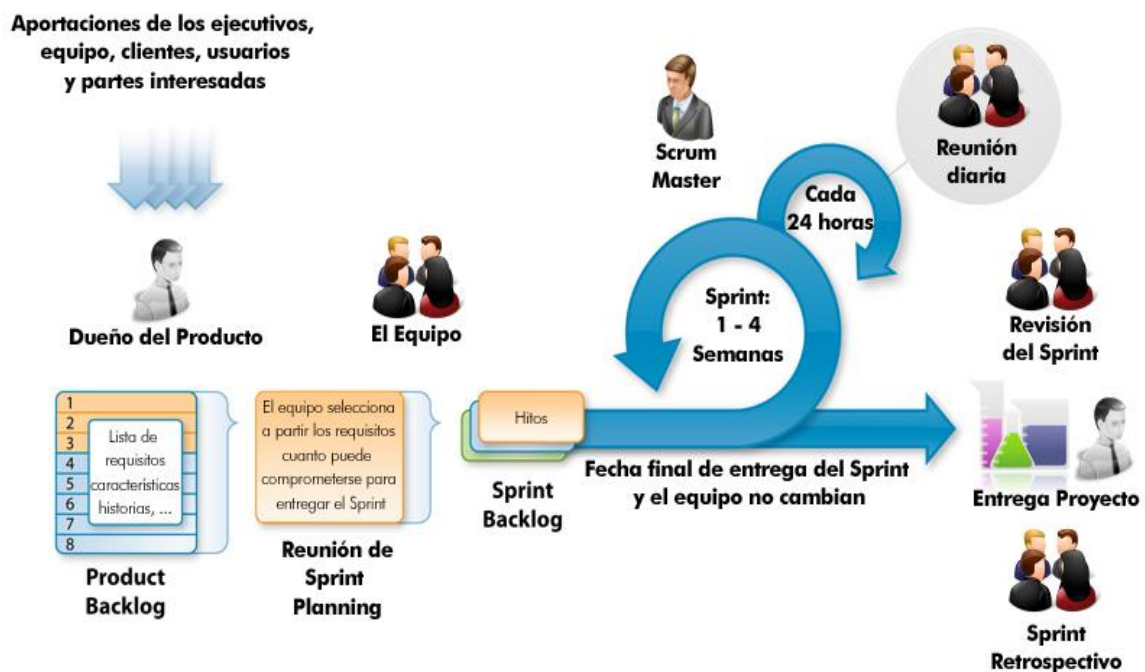


FIGURA 3: Metodología SCRUM

Fuente: Isla Visual (Visual, s.f.)

Beneficios

- Cumplimiento de expectativas
- Flexibilidad a cambios
- Mayor calidad del software
- Mayor productividad
- Predicciones de tiempos
- Reducción de riesgos

1.6.3 ANÁLISIS DE HERRAMIENTAS

Base de datos: PostgreSQL. Es un sistema de gestión de base de datos relacional orientada a objetos.

Lenguaje de Programación: PHP. El lenguaje PHP es un lenguaje de programación con variables, sentencias condicionales, ciclos (bucles), funciones.

Framework: Zend Framework es un framework open source para PHP y orientado a objetos, utilizado en aplicaciones y servicios web.

Diseño multimedia: FLEX es un framework de presentación basado en Flash para construir aplicaciones RIA.

Adobe Flash Builder. Es un entorno de desarrollo integrado (IDE), construido sobre la plataforma Eclipse que acelera el desarrollo de aplicaciones ricas de Internet (RIA) y aplicaciones de escritorio multiplataforma, particularmente para la plataforma de Adobe Flash.

Servidor web: Apache es un servidor web HTTP de código abierto. (ISTICC, 2014)

CAPÍTULO II

2 MARCO TEÓRICO

En este capítulo se describe el funcionamiento, definiciones y características de las herramientas utilizadas para la implementación y desarrollo del presente proyecto.

2.1 BPM (BUSINESS PROCESS MANAGEMENT)

La gestión de procesos de negocio es una manera integral de abordar la mejora de los procesos en cualquier organización. Las actividades de BPM buscan hacer los procesos de negocio más efectivos y eficientes.

Business Process Management (BPM) es una disciplina que involucra cualquier combinación de modelado, automatización, ejecución, control, medición y optimización de los flujos de la actividad institucional, en apoyo a lograr sus objetivos, que abarca sistemas, empleados, clientes y socios dentro y fuera de los límites de la institución.

Es un conjunto de métodos, herramientas y tecnologías utilizados para diseñar, representar, analizar y controlar procesos de negocio operacionales; un enfoque centrado en los procesos para mejorar el rendimiento que combina las tecnologías de la información con metodologías de proceso.



FIGURA 4: Ciclo de Vida BPM

Fuente: Business Process Management

2.2 TECNOLOGÍA RIA

Rich Internet Applications (RIA) es el término según el cual se define a las aplicaciones web que se comportan del mismo modo que las tradicionales aplicaciones de escritorio, pero con la capacidad de ser ejecutadas en cualquier navegador web. A diferencia de las páginas web clásicas, estas aplicaciones están orientadas a realizar tareas complejas que tradicionalmente habían sido exclusivas de los programas de escritorio, aportando una gran flexibilidad y potencia al usuario final. Para el desarrollo de aplicaciones RIA es necesario combinar una serie de tecnologías complementarias a nivel de cliente y servidor. (IBM, 2014)

- **Arquitectura**

- ✓ **Cliente**

Se maneja la interacción entre el usuario y la interfaz de usuario, el usuario invoca comandos, actualiza vistas y carga datos. Aquí se mantiene el estado de la aplicación, se manejan todas las peticiones de datos hacia el servidor y se controla como se presentan los datos. (IBM, 2014)

- ✓ **Servidor**

Aquí se manejan y se procesan todas las peticiones de la aplicación cliente y delega las acciones en el servidor, estas pueden ser, guardar datos en la base de datos, actualizar los archivos del sistema, retornar datos al servidor, o algún tipo de proceso analítico. Determina y le da formato a los datos que son retornados al cliente. (IBM, 2014)

- ✓ **Beneficios**

- No necesitan instalación (solo es necesario mantener actualizado el navegador web).
- Se pueden utilizar desde cualquier ordenador con una conexión a Internet sin depender del sistema operativo que este utilice.
- Más capacidad de respuesta, ya que el usuario interactúa directamente con el servidor, sin necesidad de recargar la página. (org, 2014)

Rich Internet Application

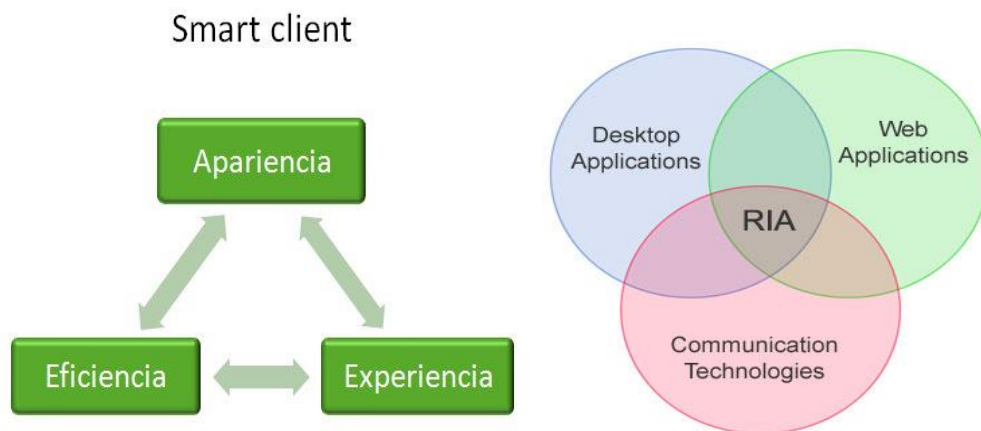


FIGURA 5: Aplicaciones Rich Internet Applications

Fuente: Tecnología RIA (serviciostic, s.f.)

2.3 HERRAMIENTAS Y ESTÁNDARES DE CÓDIGO LIBRE

2.3.1 INTRODUCCIÓN

A continuación se definen las herramientas y estándares usados para el presente proyecto en el diseño de aplicaciones de Internet enriquecidas RIA.

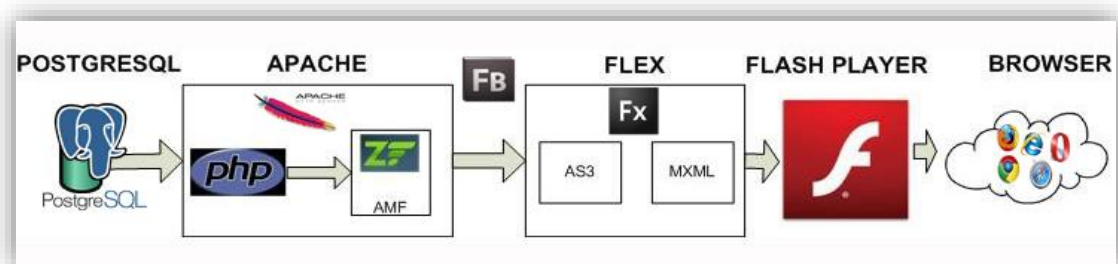


FIGURA 6: Flujo de Datos entre las Diferentes Herramientas y Frameworks

Fuente: GPI (Edición Propia)

2.3.2 SISTEMAS OPERATIVOS (SO)

Es el principal elemento que se necesita para que cualquier aplicación funcione siempre y cuando estas sean compatibles con el Sistema Operativo (en adelante SO),

Pero para la implementación de las aplicaciones web no es una pieza clave ya que el Portal Web funciona a través de un navegador o browser que puede estar instalado en cualquier SO. (SO, 2014)

- **Definición**

Sistema Operativo es un software o programa que controla y administra las funciones y servicios de un computador y además permite la ejecución de otros programas compatibles con éste. El SO es software que permite interactuar con los dispositivos de hardware y software del Computador. (SO, 2014)

2.3.3 SERVIDOR DE APLICACIONES

- **Introducción a Apache Http**

Un servidor de aplicaciones es un programa de servidor en un ordenador en una red distribuida que proporciona la lógica de negocio para un programa de aplicación. El servidor de aplicaciones es frecuentemente visto como parte de una aplicación de tres niveles, que consta de un servidor de interfaz gráfica de usuario (GUI), un servidor de aplicaciones (lógica de negocio), y un servidor de base de datos y transacción.

El servidor HTTP Apache es de código abierto para el desarrollo y mantenimiento del servidor Web Apache.

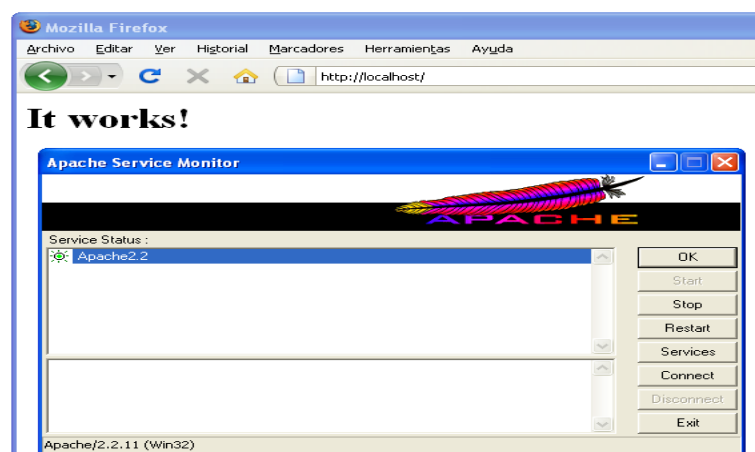


FIGURA 7: Funcionamiento de Apache

Fuente: (Ciber, 2014)

- **Características Apache Http**

- Una interfaz gráfica de usuario de primer nivel, front-end, basado en un navegador Web.
- Una aplicación lógica empresarial de nivel medio o conjunto de aplicaciones, posiblemente en una red de área local o servidor de intranet
- Un tercer nivel, base de datos y servidor de transacciones.

2.3.4 BASES DE DATOS. POSTGRESQL

Una base de datos es una colección de información que se organiza para que fácilmente se puede acceder, administrar y actualizar. Las bases de datos se pueden clasificar de acuerdo a los tipos de contenido: bibliográficas, de texto completo, numéricos e imágenes.

- **POSTGRESQL**

PostgreSQL es un SGBD (Sistema de gestión de bases de datos) relacional orientado a objetos y libre, publicado bajo la licencia BSD (licencia de software libre permisiva).

Los conceptos añadidos a PostgreSQL son:

- Clases
- Herencia
- Tipos
- Funciones

Y otras características que aportan potencia y flexibilidad adicional son las siguientes:

- Restricciones (Constraints)
- Disparadores (triggers)

- Reglas (rules)
- Integridad transaccional

PostgreSQL tiene algunas características que son propias y referidas de las bases de datos orientadas a objetos.

Una base de datos PostgreSQL se pueden migrar lógicamente a todas las bases de datos compatibles. (postgresql, 2014)

- **Características**

- Soporta distintos tipos de datos, también permite la creación de tipos propios.
- Incluye herencia entre tablas, por lo que a este gestor de bases de datos se le incluye entre los gestores objeto-relacionales.
- Copias de seguridad en caliente
- Unicode
- Juegos de caracteres internacionales
- Regionalización por columna
- Multi-Version Concurrency Control (MVCC)
- Múltiples métodos de autenticación
- Acceso encriptado vía SSL
- Completa documentación
- Licencia BSD

PostgreSQL permite que mientras un proceso escribe en una tabla, otros accedan a la misma tabla sin necesidad de bloqueos. Cada usuario obtiene una visión consistente de lo último a lo que se le hizo commit.

Algunas limitaciones que caracterizan a PostgreSQL son las siguientes:

TABLA 2: Límites de PostgreSQL

| Límite | Valor |
|-------------------------------------|---|
| Máximo tamaño base de dato | Ilimitado (Depende de tu sistema de almacenamiento) |
| Máximo tamaño de tabla | 32 TB |
| Máximo tamaño de fila | 1.6 TB |
| Máximo tamaño de campo | 1 GB |
| Máximo número de filas por tabla | Ilimitado |
| Máximo número de columnas por tabla | 250 - 1600 (dependiendo del tipo) |
| Máximo número de índices por tabla | Ilimitado |

Fuente: (Postgresql, s.f.)

2.3.5 PHP

PHP (Hypertext Pre-processor) es un lenguaje de programación de uso general de código del lado del servidor diseñado para el desarrollo web de contenido dinámico. El código es interpretado por un servidor web con un módulo de procesador de PHP que genera la página Web resultante. PHP tiene una interfaz de línea de comandos que puede ser usada en aplicaciones gráficas independientes. (Group T. P., 2014)

PHP puede ser desplegado en la mayoría de los servidores web y en casi todos los sistemas operativos y plataformas sin costo alguno.

✓ Características de PHP

- Orientado al desarrollo de aplicaciones web dinámicas con acceso a información almacenada en una base de datos.

- El código fuente escrito en PHP es invisible al navegador web y al cliente, ya que es el servidor el que se encarga de ejecutar el código y enviar su resultado HTML al navegador. Esto hace que la programación en PHP sea segura y confiable.
- Debido a su flexibilidad ha tenido una gran acogida como lenguaje base para las aplicaciones WEB de manejo de contenido, y es su uso principal. (Group T. P., 2014)

2.3.6 ACTIONSCRIPT

ActionScript 3.0 es un modelo y lenguaje de programación propio de Flash orientada a objetos.

El ActionScript está basado en la especificación ECMA-262, al igual que otros lenguajes como Javascript.

Es una interfaz de programación de aplicaciones (API) ampliada y mejorada, un auténtico modelo objetos, basado en la especificación de eventos.

```
1. var quienes:ContextMenuItem = new ContextMenuItem("QUIENES SOMOS");  
2. var trabajos:ContextMenuItem = new ContextMenuItem("TRABAJOS");
```

FIGURA 1: Ejemplo Código ActionScript 3.0

Fuente: ActionScript 3.0 Básico

2.3.7 MXML (MINIMAL XML)

MXML Lenguaje de programación jerárquica que utiliza Flex conjuntamente con el lenguaje de programación ActionScript. Describe interfaces de usuario, crea modelos de datos y tiene acceso a los recursos del servidor, del tipo RIA (Rich Internet Applications). (Sereni, n.d.)

MXML tiene una mayor estructura en base a etiquetas, similar a HTML, proporciona una gran variedad e inclusive permite extender etiquetas y crear sus propios componentes. (Sereni, n.d.)

- **Modelo de Datos MXML**

En un objeto de documento donde contiene nodos similares a XML, MXML también almacena eventuales errores y la línea donde se ha encontrado un error. El documento también se utiliza para llevar "estilos de formato", solicitudes de entrada o salida. (Sereni, n.d.)

Se establece los siguientes nodos de información:

- a) **Nodos del documento:** Son los nodos del nivel superior, nodos raíz.
- b) **Nodos Tag:** Nodos que almacenan información en el XML
- c) **Nodos de datos:** Datos de un elemento.
- d) **Nodos Comentario:** El bloque de comentario inicia con `<!--` y finaliza con `-->`.

- **Estructura de un documento MXML**

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <s:Group xmlns:fx="http://ns.adobe.com/mxml/2009"
3     xmlns:s="library://ns.adobe.com/flex/spark"
4     xmlns:mx="library://ns.adobe.com/flex/mx" width="400" height="300">
5     <fx:Script>
6         <![CDATA[
7             import gpiinventarios.services.ServiceManager;
8             import gpiinventarios.services.productos servicio.ProductosServicio;
9
10            import mx.controls.Alert;
11            import mx.events.FlexEvent;
12
13            protected var productosServicio:ProductosServicio = ServiceManager.getInstance()
14
15            protected function form_creationCompleteHandler(event:FlexEvent):void
16            {
17                conteoProductosResult.token = productosServicio.conteoProductos();
18            }
19
20        ]]>
21     </fx:Script>
22     <fx:Declarations>
23         <s:CallResponder id="conteoProductosResult"/>
24         <!-- Place non-visual elements (e.g., services, value objects) here -->
25     </fx:Declarations>
26     <s:Form id="form" creationComplete="form_creationCompleteHandler(event)">
27         <s:FormItem label="ConteoProductos">
28             <s:TextInput id="conteoProductosTextInput"
29                 text="{conteoProductosResult.lastResult as int}"/>
30         </s:FormItem>
31     </s:Form>
32 </s:Group>
```

FIGURA 9: Estructura de un documento MXML

Fuente: Propia

Los documentos escritos en MXML están estructurados en:

a) Espacios de Nombre

Los espacios de nombre o namespace son los que nos permite importar las librerías y bibliotecas con la que podemos trabajar en un componente mxml.

Por defecto se crea los siguientes espacios de nombre:

```
xmlns:fx="http://ns.adobe.com/mxml/2009"
```

```
xmlns:s="library://ns.adobe.com/flex/spark"
```

```
xmlns:mx="library://ns.adobe.com/flex/mx"
```

Pero necesitamos incorporar namespaces adicionales para trabajar con datos externos.

Ejemplo:

```
xmlns:entidadesservicio="services.entidadesservicio.*"
```

b) Espacio ActionScript

Esta sección se utiliza para declaraciones y funciones ActionScript si es necesario para acceder a los componentes.

Ejemplo:

```
<fx:Script>  
  
import flexlib.controls.tabBarClasses.SuperTab;  
  
protected function filtroGeneradoEventoHandler(event:Event):void  
{  
  
    ExternalInterface.call("console.log",barraHerramientas);  
  
    productosTiposGridPaginados.buscar("", "");  
  
}  
  
</fx:Script>
```

c) Etiquetas de Componentes

Esta sección es utilizada para declaraciones de etiquetas de componentes y propiedades de los mismos.

Ejemplo:

```
<mx:Label width="150" text="Items Total: " textAlign="right"/>
<mx:TextInput id="itemsTotalTxt" text="0" width="100"/>
```

2.3.8 CSS (CASCADING STYLE SHEETS)

Lenguaje de hojas de estilos creado para controlar la presentación de diferentes documentos, es imprescindible para la creación de páginas web complejas.

Este lenguaje tiene una sintaxis muy sencilla ya que usa unas cuantas palabras claves tomadas del inglés para especificar los nombres de sus selectores, propiedades y atributos como se muestra a continuación.

```
<fx:Style>
  .noArrowsScrollBar
  {
    upArrowSkin      : ClassReference( null );
    downArrowSkin    : ClassReference( null );
    thumb-skin       : ClassReference("PaginatorScrollBar_thumbSkin");
    track-skin       : ClassReference("PaginatorScrollBar_trackSkin");
    thumb-icon       : ClassReference( null );
  }
</fx:Style>
```

FIGURA 2: Ejemplo CSS

Fuente: Propia

Una hoja de estilos se forma en una serie de reglas, cada regla se forma de uno o más selectores y un bloque de estilos con los estilos a aplicar para los elementos del documento que cumplan con el selector que les precede. Cada bloque de estilos se define entre llaves, y está formado por una o varias declaraciones de estilo con el formato propiedad: valor, como se mostró en la imagen anterior. (Bert Bos, 2014)

- **Funcionamiento de CSS**

Para dar formato a un documento MXML y poner en funcionamiento lo que hemos escrito en el documento CSS, se lo realiza de las siguientes formas:

- a) **Incluir CSS en el mismo documento XHTML.-** Ubicándole dentro la cabecera del documento específicamente dentro de la etiqueta `<fx:Style></fx:Style>`. De esta manera se emplea cuando se quiere dar formato o estilos.

Ejemplo:

```
<fx:Style>
    TextInput {
        fontSize : 14;
        color: blue;
    }
</fx:Style>
```

- b) **Definir una hoja de estilo externa.-** Es una hoja con extensión CSS donde se pueden incluir todo tipo de estilos para los elementos de MXML, a los cuales se enlazan mediante la etiqueta `<link>` dentro de la cabera de la página web.

Se pueden crear todos los archivos CSS que sean necesarios y cada documento HTML puede enlazar tantos archivos CSS como necesite.

Ejemplo:

```
<fx:Style source="../assets/SimpleTypeSelector.css"/>
```

2.4 SEGURIDAD EN UN PORTAL WEB

Los portales WEB se ven expuestos a un creciente número de amenazas y vulnerabilidades que pueden afectar la imagen institucional, la disponibilidad de los servicios brindados, la integridad y la confidencialidad de la información que se trasmite a través del mismo, entre otros.

Los problemas de seguridad pueden ser externos a través del internet en busca de información para poder acceder a las aplicaciones.

- **Manejo Adecuado de Errores**

Un portal frecuentemente genera mensajes de error durante su operación normal. Estos errores deben ser manejados de acuerdo a un esquema bien pensado que provea al usuario de un mensaje con sentido, información de diagnóstico para quienes mantienen el sitio, y ninguna información útil para un atacante.

El manejo de errores debe contemplar tanto las entradas generadas por el usuario, así como cualquier error que pueda ser generado por componentes internos.

La amenaza de no realizar un correcto manejo de errores radica en revelar información detallada a través del mensaje de error, como el contenido de variables. (AGESIC)

2.5 PLATAFORMAS Y FRAMEWORKS

El Sistema GpilInventarios será implementado en las siguientes plataformas y Frameworks:

- Entorno de Modelado de Procesos: TIBCO BPM, Process Maker.
- Framework MVC: ZEND
- Administrador de base de datos para PostgreSQL: PG Admin3, Toad Data Modeler, Navicat.
- Servidor web: Apache
- SDK (Kit de desarrollo de Software): Adobe Flex
- Entorno de desarrollo: Adobe Flash Builder

2.5.1 ADOBE FLASH BUILDER

Es una herramienta de desarrollo basada en Eclipse que permite crear en muy poco tiempo aplicaciones móviles, web y de escritorio de gran expresividad haciendo uso de ActionScript y del marco de trabajo de código abierto Flex. (org, 2014)

2.5.2 ADOBE FLEX

Flex es un marco de aplicación de código abierto de alta productividad para la construcción y mantenimiento de aplicaciones web expresivas que se implantan coherentemente en todos los principales navegadores, escritorios y dispositivos. Proporciona un lenguaje moderno basado en estándares y el modelo de programación compatible con los patrones de diseño común adecuado para los desarrolladores de diferentes orígenes.

El desafío de adaptar la metáfora de la animación sobre la plataforma con la cual fue originalmente construido Flash, lo minimiza este problema Flex proveyendo un flujo de trabajo y un modelo de programación que es familiar a los desarrolladores de aplicaciones.

Flex tiene varios componentes y características que aportan funcionalidades tales como Servicios Web, objetos remotos, arrastrar y soltar, columnas ordenables, gráficas, efectos de animación y otras interacciones simples. El lenguaje y la estructura de archivos de Flex buscan el desacoplamiento de la lógica y el diseño.

- **Flex 4**

Los principales objetivos Flex 4 son los siguientes:

- **Diseño en mente:** La arquitectura de personalización (Skinning en inglés) se simplifica.

- Productividad del desarrollador: Se mejoró el desempeño del compilador y se perfeccionó el proceso de enlazamiento de datos (data binding en inglés) de los componentes.
- Evolución del framework: se añadieron nuevos componentes y se modificó el SDK para aprovechar las características del nuevo Flash Player 10.
- Uno de los cambios más notables en Flex 4 es la incorporación de una nueva arquitectura de componentes llamada Spark; ésta reemplaza a algunos de los componentes que existían en el espacio de nombres MX que era usado en las versiones anteriores de Flex.
- **Proceso de desarrollo de una aplicación Flex**
 - Definir un interfaz de aplicación usando un conjunto de componentes pre-definidos (formularios, botones,...)
 - Ordenar estos componentes en el diseño de la interfaz de usuario.
 - Usar estilos y temas para definir el diseño visual.
 - Añadir comportamiento dinámico (una parte de la aplicación interactuando con otra, por ejemplo).
 - Definir y conectar a servicios de datos según sea necesario (servicios HTTP).
 - Compilar el código fuente en un archivo SWF que funcione en el reproductor Flash. (org, 2014)

2.5.3 FLASH PLAYER

Adobe Flash Player es una aplicación en forma de reproductor multimedia. Permite reproducir archivos SWF que pueden ser creados con la herramienta de autoría Adobe Flash, con Adobe Flex o con otras herramientas de Adobe. Estos archivos se reproducen en un entorno determinado.

En un sistema operativo tiene el formato de aplicación del sistema, mientras que si el entorno es un navegador, su formato es el de un Plug-in u objeto ActiveX.

Adobe Flash es el entorno de creación y Adobe Flash Player el reproductor o máquina virtual. Flash Player tiene soporte para el lenguaje de programación interpretado ActionScript(AS) con soporte de programación orientada a objetos, comparable en funciones y uso al lenguaje JavaScript. (Adobe F. P., 2014)

2.5.4 ZEND FRAMEWORK

- **Framework**

Los Frameworks entregan una serie de clases, funciones o librerías, que junto a convenciones comunes, se organizan bajo una estructura sobre la cual desarrollamos nuestros proyectos.

- ***Ventajas de usar Frameworks:***

- ✓ Facilita integrar a otras personas a tus proyectos ya que se comparten convenciones de desarrollo comunes.
- ✓ No nos preocupamos de mantener actualizadas las distintas partes.
- ✓ Aumentan la velocidad de desarrollo.

- ***Desventajas:***

- × Hay que invertir tiempo en aprender a usarlos.
- × En algunos casos una aplicación desarrollada con un framework puede ser más lenta (en cuanto a rendimiento) que una diseñada y desarrollada desde cero. (Kickbill, 2014)

- **Zend**

Zend Framework (ZF) es un framework opensource orientado a objetos para servicios web con PHP 5 y aplicaciones web.

Los componentes de Zend Framework conforman un extensible y potente framework de aplicaciones web al combinarse, basado en MVC de gran rendimiento, una abstracción de base de datos fácil de usar, además de formularios que facilita la prestación de formularios XHTML, filtrado y validación para poder consolidar todas las operaciones usando la interfaz orientada a objetos. Zend_Auth y Zend_Acl, proveen autenticación de usuarios y autorización con gran facilidad.

- **Ventajas**

Si bien tiene varias ventajas, una de sus desventajas es que es un poco más difícil de aprender que otras alternativas.

Elegir Zend Framework por las siguientes razones:

- En general muy buena documentación y una comunidad activa.
- Gran cantidad de componentes, incluidos algunos de Microsoft, Google y Adobe
- Soporta PHP 5. (Kickbill, 2014)

2.5.5 TIBCO BUSINESS STUDIO

TIBCO Business Studio es un entorno empresarial libre, basado en estándares, el proceso de modelado que permite a los expertos de negocios para modelar y simular los procesos de negocio y los datos de apoyo y modelos de organización. (TIBCO, 2014)

Faculta a los expertos del negocio de poseer y gestionar los activos de los procesos de negocio. Modelos de procesos explícitos mejorar la comprensión de los procesos de negocio en toda la organización, la precisión de la ejecución de los procesos, y la capacidad de optimizar los procesos. (TIBCO, 2014)

TIBCO Business Studio es también el entorno de desarrollo unificado para la creación de aplicaciones compuestas a través de BPM de TIBCO y software de SOA. (TIBCO, 2014)

- **Beneficios**

Modelado de procesos basada en estándares usando BPMN¹. Realizar simulaciones para validar los procesos e identificar los costos, tiempos, cuellos de botella, o recursos subutilizados. Modelado de datos de negocios utilizando UML.

Estos son aspectos claves en el comportamiento de los procesos de negocios y cómo la gente realiza trabajo. (TIBCO, 2014)

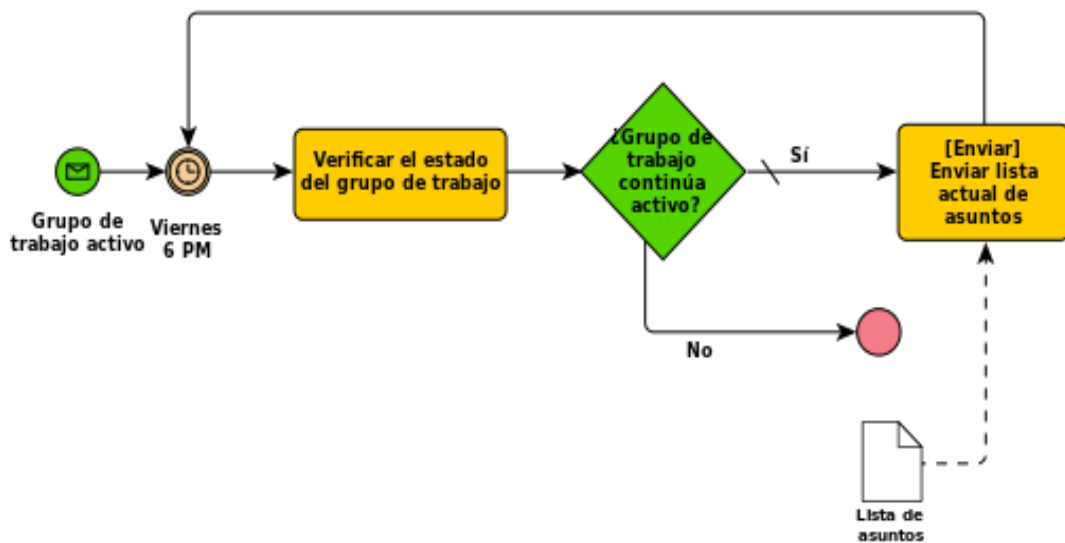


FIGURA 11: Ejemplo de diagrama de proceso

Fuente: BPMN

2.5.6 TOAD DATA MODELER

Toad Data Modeler es una herramienta de modelado de datos y diseño de base de datos el cual permite:

- Crear modelos de datos físicos y lógicos.
- Sincronizar y comparar modelos.

¹ BPMN. Business Process Modeling Notation (Notación para el Modelado de Procesos de Negocio) es una notación gráfica estandarizada que permite el modelado de procesos de negocio.

- Generar DDL/ SQL complejos
- Crear y modificar scripts
- Aplicar ingeniería directa e inversa a bdd y sistemas de almacenamiento de datos. (Software, 2014)

---- NBVB VCCXX<0

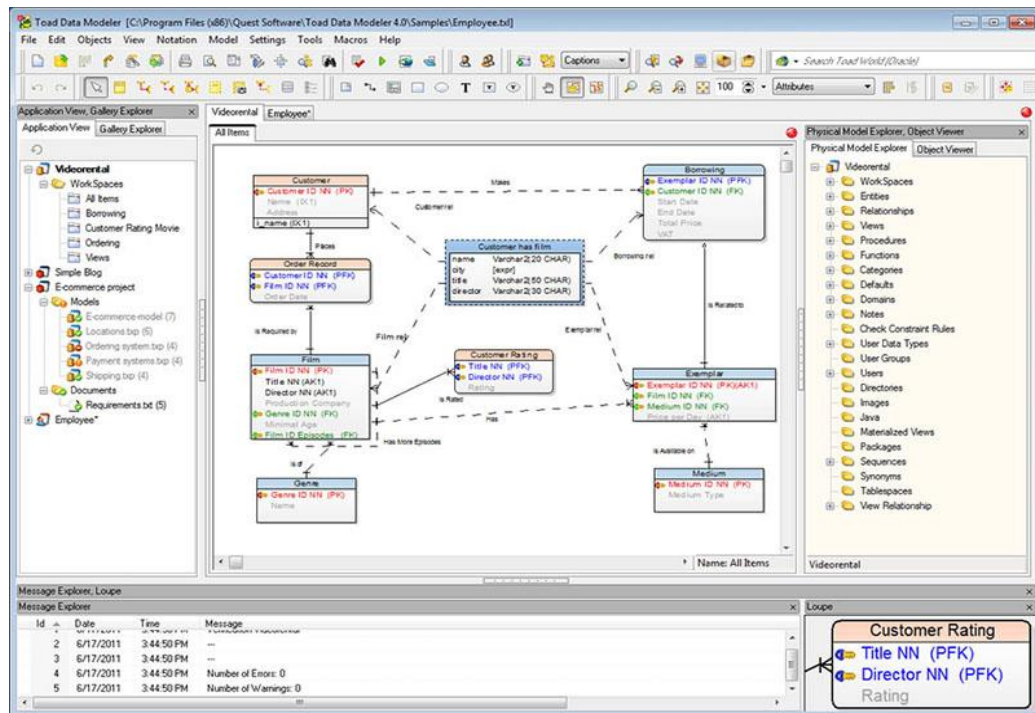


FIGURA 12: Toad Data Modeler

Fuente: (quest, 2014)

2.5.7 PGADMIN III

Es un programa de gestión de bases de datos PostgreSQL. Totalmente libre, que ayuda a los usuarios a escribir consultas SQL.

- **Características principales:**
 - **Gestión de base de datos:** Procesar fácilmente y ordenar datos. Su interfaz sólo contiene las funciones que necesita para que no de problemas a través de menús y pestañas voluminosas.

- **Consultas SQL:** Permite consultas SQL ya sean básicas o complicadas.

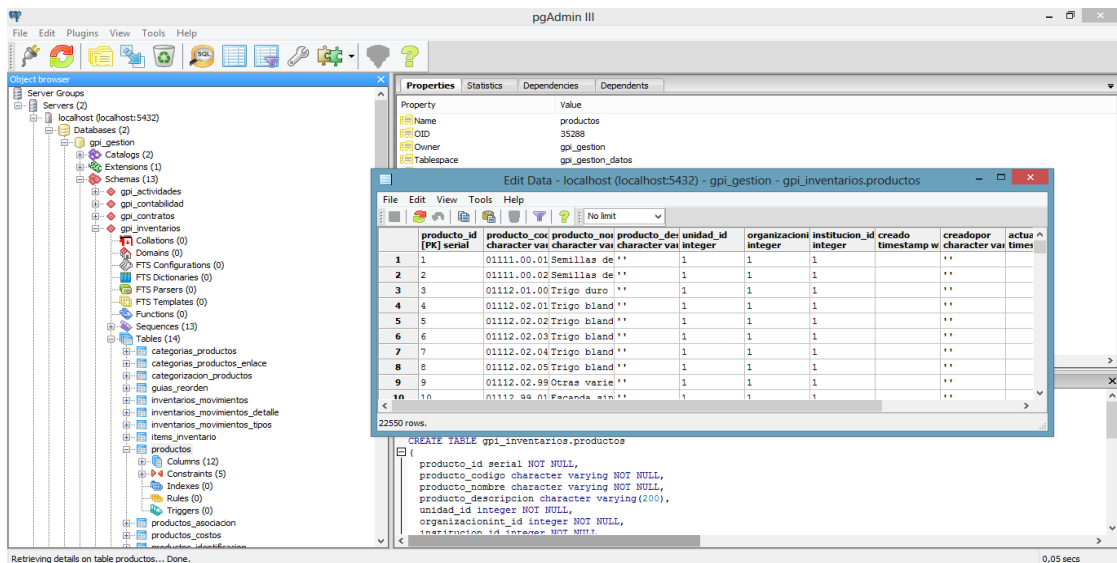


FIGURA 13: PGADMIN III

Fuente: Propia

2.5.8 NAVICAT

Es un conjunto de gestión de base de datos gráfica y desarrollo de software para MySQL, MariaDB, Oracle, SQLite, PostgreSQL y Microsoft SQL Server. Cuenta con una interfaz de usuario similar al Explorador gráfico y admite varias conexiones de base de datos para bases de datos locales y remotos.



FIGURA 14: Navicat. Programa de mantenimiento y consulta de bases de datos.

Fuente: (applesfera, 2014)

2.5.9 PROCESSMAKER

ProcessMaker software utilizado para automatizar flujos de trabajo BPM a gran escala, opensource simple y rentable. Es un Gestor de procesos empresariales.

Características

- **Generar** Una interfaz intuitiva de drag-and-drop permite a los analistas de negocios modelar flujos de trabajo basado en aprobaciones de una forma más efectiva.

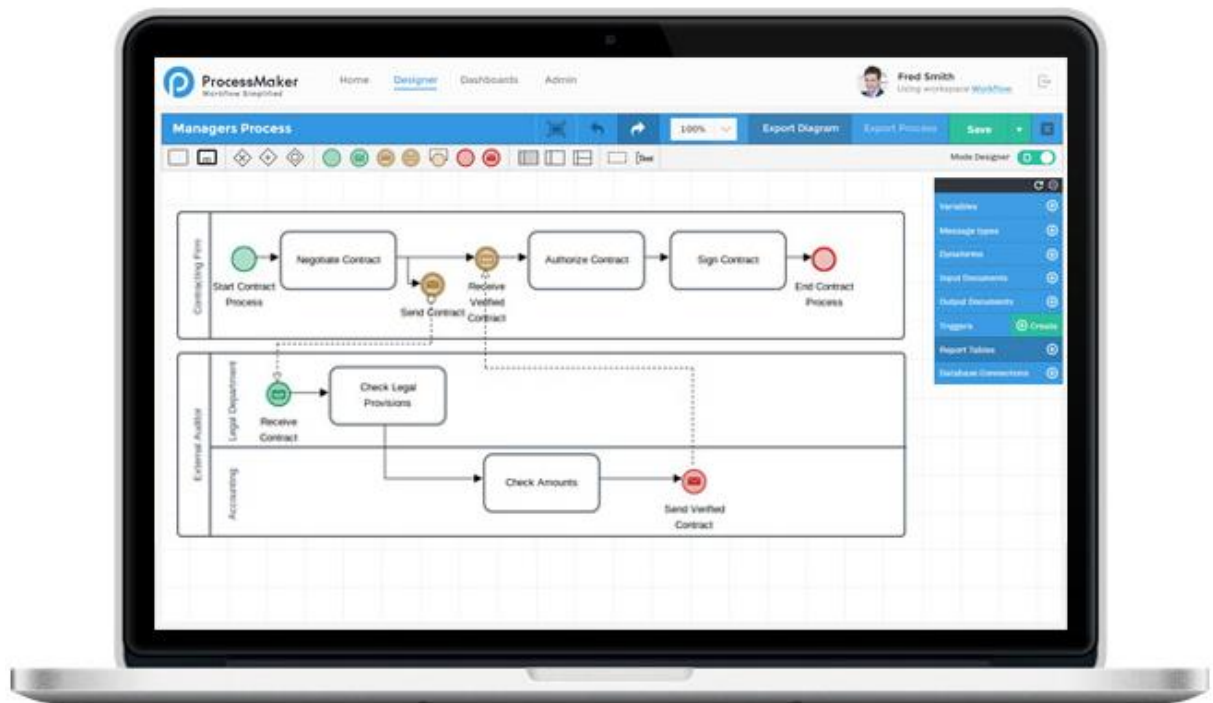


FIGURA 15: Generar PM.
Fuente: (processmaker, 2016)

- **Ejecutar** Los usuarios empresariales pueden completar fácilmente procesos a través de notificaciones automatizadas y una interfaz online u offline en diferentes dispositivos.



FIGURA 3: Ejecutar PM.

Fuente: (processmaker, 2016)

- **Reportar** Los gerentes tienen acceso a las KPIs y métricas para tomar decisiones.

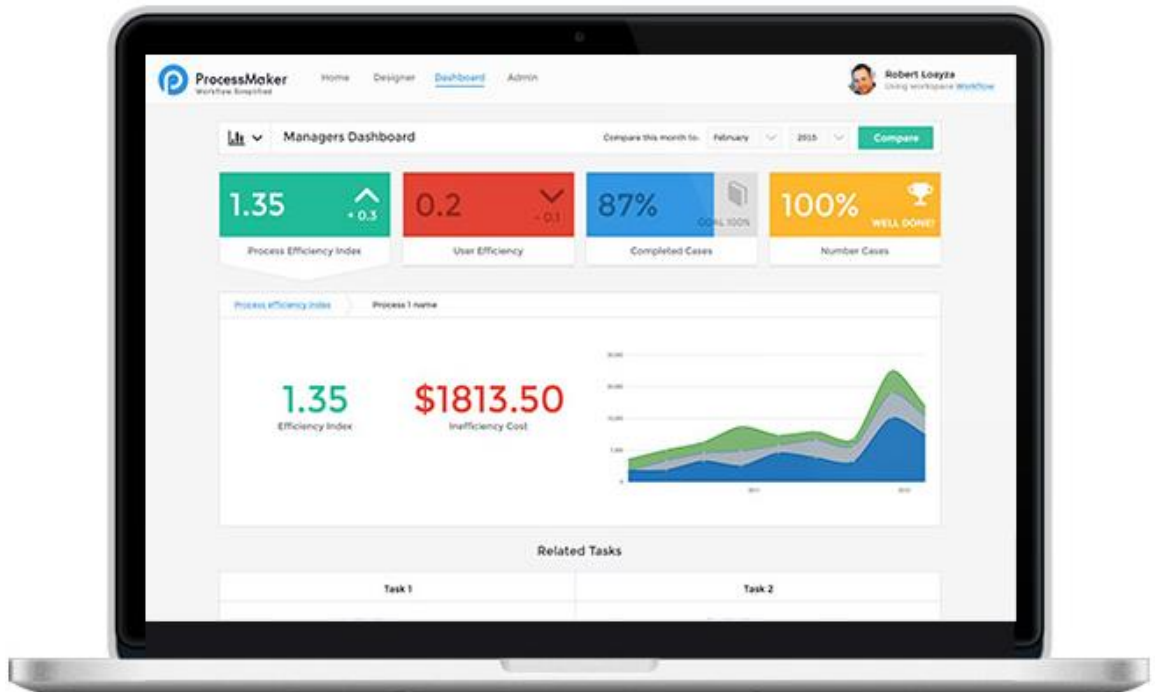


FIGURA 4: Reportar PM.

Fuente: (processmaker, 2016)

- **Optimizar** Los gerentes y analistas pueden mejorar continuamente el rendimiento al descubrir cuellos de botella en procesos e ineficiencias.

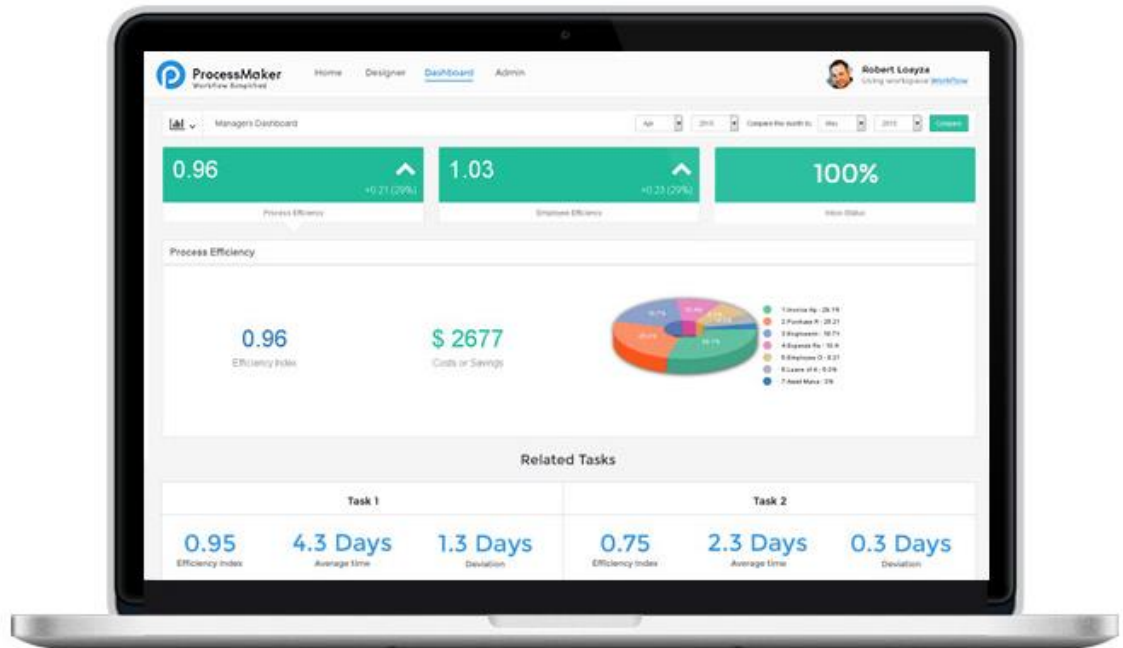


FIGURA 5: Optimizar PM.
Fuente: (processmaker, 2016)

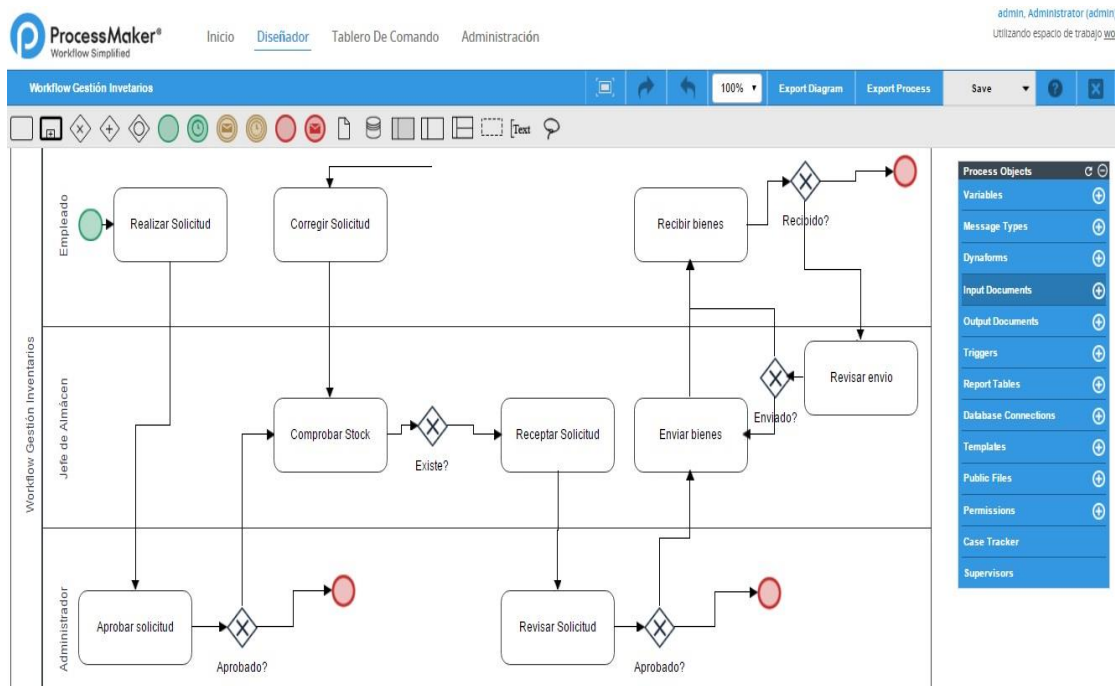


FIGURA 6: Diseñador de los macro-procesos
Fuente: Propia

CAPÍTULO III

3 MÓDULOS.

3.1 FUNCIONAMIENTO Y DESCRIPCIÓN DEL SISTEMA

3.1.1 VISTA GENERAL

En el presente capítulo se definen los Modelos Funcionales realizados como guía para el desarrollo del Sistema de Gestión de Inventarios del GPI. En este documento la Gestión de Inventarios se define como la administración adecuada del registro, entradas y salidas de inventario.

En base a la descripción de las funciones y procesos de gestión de inventarios que actualmente se ejecutan en el GPI y de la información que generan, se ha diseñado el esquema del Sistema de Gestión de Inventarios teniendo como módulos importantes los siguientes:

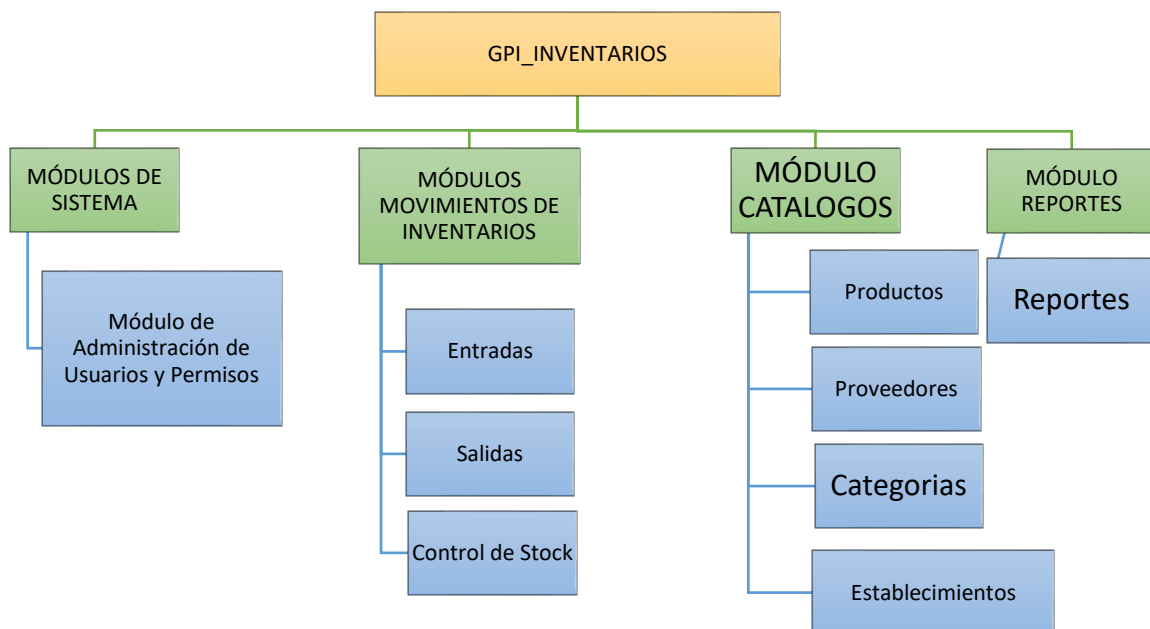


FIGURA 7: Módulos GPIINVENTARIOS

Fuente: Propia

3.1.2 CARACTERÍSTICAS DEL SISTEMA

El sistema tiene las características que se describen a continuación:

TABLA 3: Características del Sistema

| Característica | Descripción |
|-------------------------------------|---|
| Administración asociada | Instalación y actualización del lado del servidor característica principal de una aplicación Web. |
| Usabilidad | Diseño y funcionalidad simples, menú intuitivo, fácil navegación e información consistente. |
| Multiplataforma | Acceso al sistema en diferentes navegadores web. |
| Tiempos de respuestas | Menor tiempo de respuesta en los diferentes procesos. |
| Acceso a la información. | Reportes, opciones de búsqueda y consultas. Brindará reportes de toda la información sobre los inventarios que se generan dentro del GPI. |
| Interfaz de usuario amigable | La Interfaz gráfica del sistema estará enlazado a la base de datos para que la información pueda ser almacenada y será de fácil acceso para el usuario. |
| Evaluación de reportes | Los diferentes reportes que generará el sistema serán entendibles. |

Fuente: Propia

3.1.3 DEPENDENCIAS PARA LA IMPLEMENTACIÓN

Implementar un ambiente de pruebas e identificar posibles errores y conflictos en la información.

- **Costos**

El presente proyecto no aplica costos de licenciamiento ya que se desarrolló a través de Adobe Flex Builder 4, el cual se encuentra disponible en tres ediciones: Standard, Premium y de la Educación. El paquete está disponible de forma gratuita para uso no comercial por los estudiantes y desarrolladores.

Para la implantación del sistema se necesita una computadora personal o de escritorio con las características básicas necesarias para el desarrollo e implementación como un equipo de pruebas.

- **Licenciamiento**

El Sistema de Gestión de Inventarios utiliza las siguientes herramientas establecidas por las cláusulas de GPL que garantiza a los usuarios finales la libertad de usar, estudiar, compartir y modificar el software.

- a. SCRUM: Metodología ágil de Desarrollo.
- b. PostgreSQL: Base de Datos.
- c. Zend Framework: Mapeo de datos.
- d. Adobe Flash Builder: Plataforma de Desarrollo.
- e. Protocolo AMF: Integración de funcionalidades Flex - PHP.
- f. MXML (Minimal XML): Lenguaje descriptivo de código abierto.
- g. PHP: Lenguaje de Programación Licencia Publica General de GNU.

- **Instalación**

La instalación se realizará de forma centralizada en un solo servidor, los clientes accederán al servidor de aplicaciones a través de cualquier navegador web desde la Intranet de la Institución, el navegador deberá contar con el plugin Flash Player gratuito proporcionado por el sitio de Adobe. (<http://www.adobe.com/la/products/flashplayer.html>).

3.1.4 FUNCIONAMIENTO.

El Sistema de Gestión de Inventarios se desarrolla e implementa usando la plataforma Adobe Flash Builder y como patrón de diseño MVC+S (Modelo, Vista, Controlador, Servicios) para aplicaciones interactivas.

La interface gráfica (formularios, menús, etc.) utiliza MXML, teniendo como principal formulario el Login de usuario con la página GpiGestionFrontend.php

El usuario deberá autenticarse con las credenciales correctas, la aplicación verifica la autenticación correcta y le redirecciona al módulo principal de la aplicación, el usuario podrá navegar por las diferentes opciones de la aplicación dependiendo del rol y tipo de usuario.

- **GPI_GESTION**

Se trata de implementar una herramienta ágil y de fácil uso que reduzca las probabilidades de error y los costos de administración, la cual deberá integrar los diferentes módulos de la aplicación.

A continuación se muestra el funcionamiento en General del Sistema GPI_GESTION.

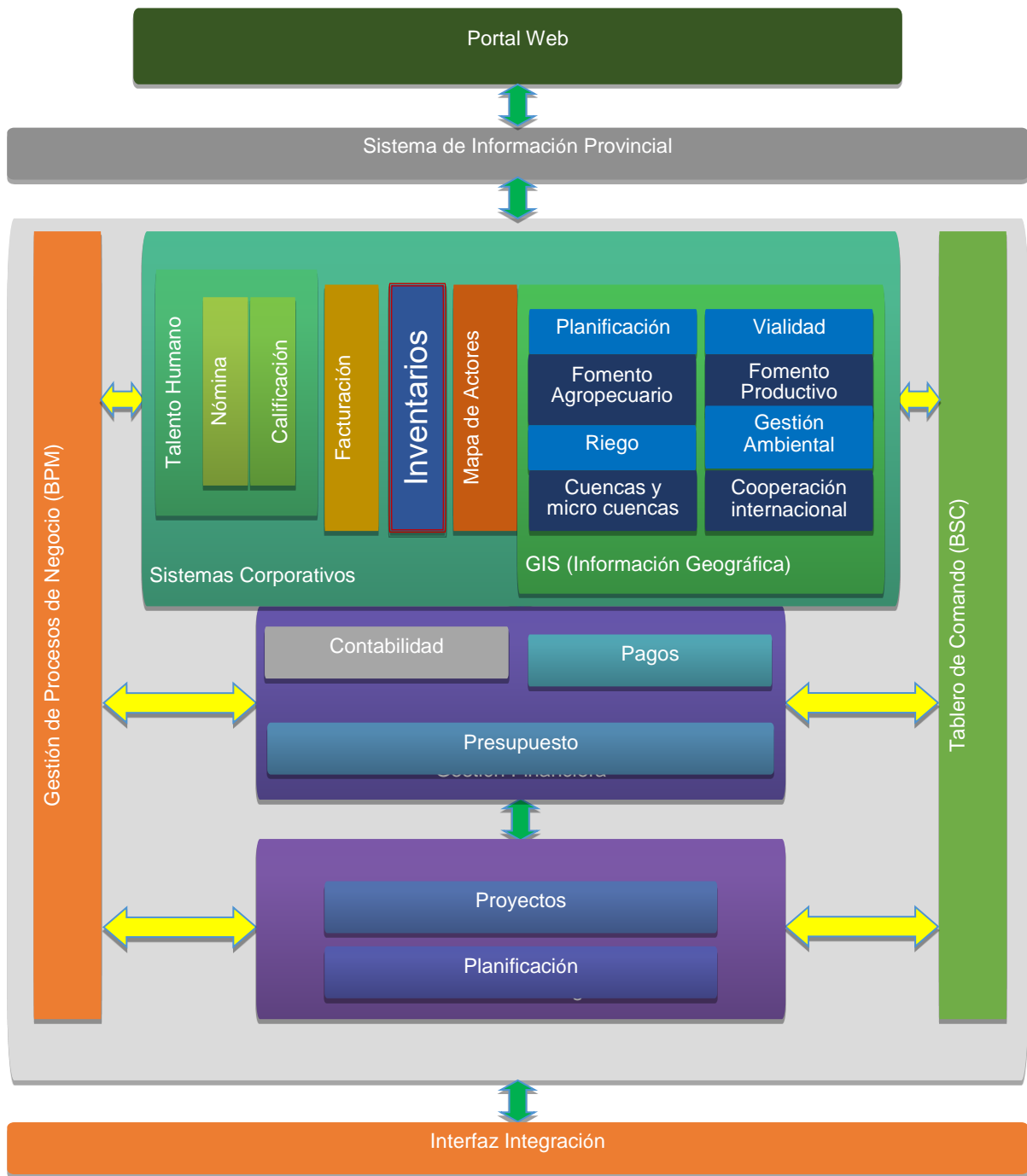


FIGURA 21: Funcionamiento General del Sistema

Fuente: GPI

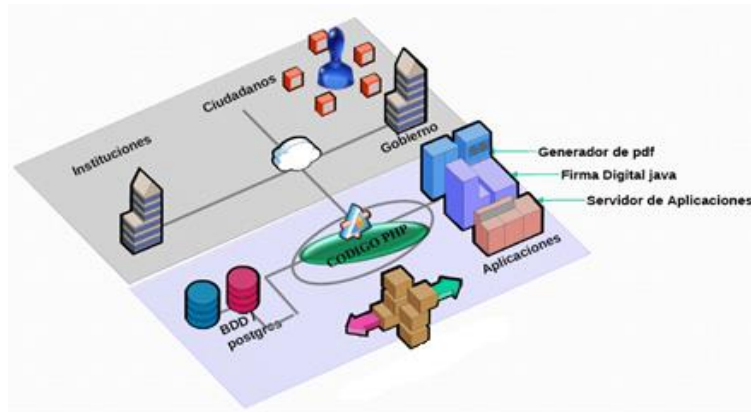


FIGURA 8: ARQUITECTURA GPI_GESTION

Fuente: GPI

- **Arquitectura del sistema**

La arquitectura MVC+S, modelo vista controlador más servicios organiza el diseño y desarrollo de una aplicación separando la presentación de los datos.

Esta aplicación sigue el patrón clásico de meta-diseño conocido como Modelo-Vista-Controlador (MVC), con la adición de un cuarto actor de llamada de servicio.

- a. Modelo. Capa en la cual se realizan las consultas, operaciones y manipulación de datos (CRUD).
- b. Vista. Es la interfaz gráfica de la aplicación, por la cual se presenta los datos al usuario.
- c. Controlador. Acciones de usuarios realizadas en la vista, relaciona los métodos de negocio creados en el modelo y los presenta en la vista seleccionada.
- d. Servicio. Peticiones a los servicios establecidos para la aplicación.

En resumen el modelo se encarga de los datos de la aplicación, la vista de la interfaz gráfica, el controlador del negocio y funcionalidad de la aplicación.

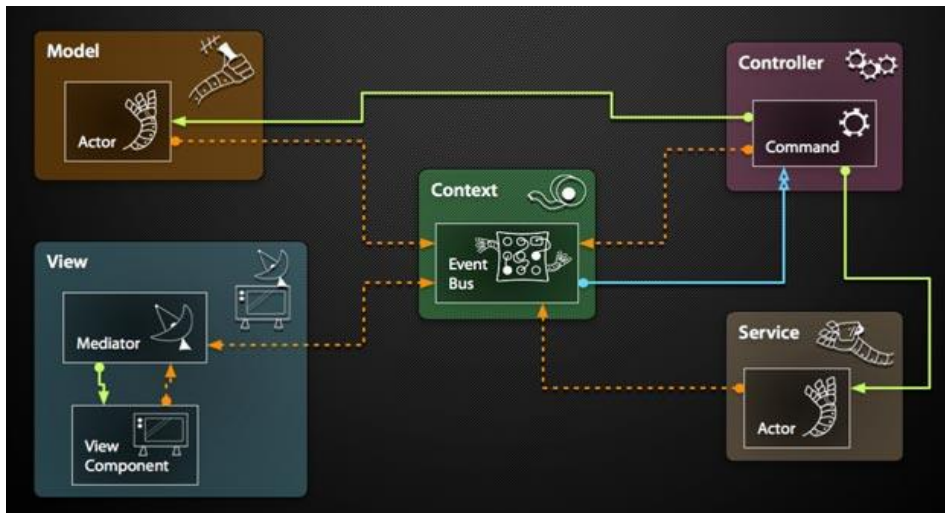


FIGURA 23: Arquitectura MVC+S

Fuente: Robotlegs implementación de referencia (robotlegs-mvcs, 2014)

- **ARQUITECTURA FLEX PHP**

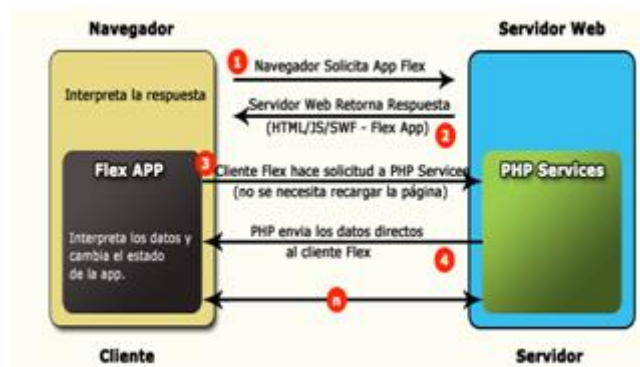


FIGURA 24: Arquitectura FLEX y PHP

Fuente: GPI (adobe.com, 2014)

La arquitectura Flex PHP, funciona:

1. Navegador solicita App Flex, iteración PHP services.
2. Servidor Web retorna respuesta a cliente Flex.
3. Cliente Flex hace solicitud a PHP services, no se necesita recargar la página.
4. PHP envía los datos directos al cliente Flex, pueden ser HTML, JS, SWF-Flex App.

- **FLUJO DE DATOS ENTRE FLEX Y PHP POR MEDIO DE AMF**

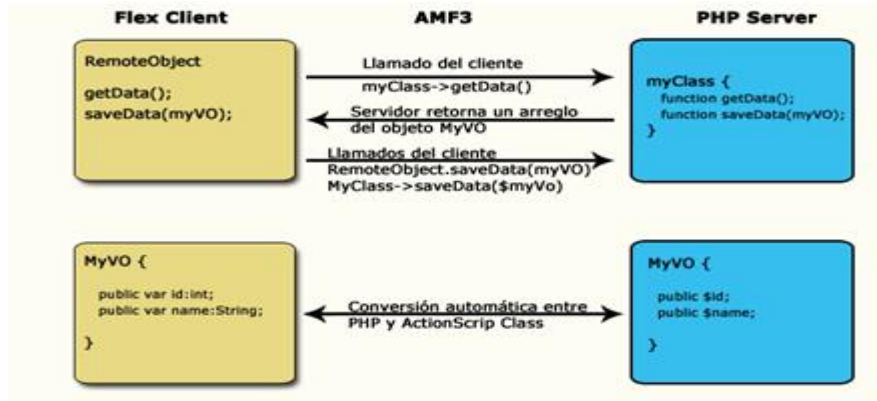


FIGURA 25: Flujo de Datos entre Flex y PHP

Fuente: GPI (adobe.com, 2014)

- ✓ El formato de mensajes es AMF3 (Action Message Format), similar a SOAP/XML/JSON pero más rápido y pequeño.
- ✓ Conversión automática entre PHP y ActionScript.
- ✓ Acceso a métodos públicos de clases PHP del servicio.

- **Diagrama General de las Historias de Usuario**

| Historia de Usuario | |
|--|-----------------------------|
| Número: | Nombre Historia de Usuario: |
| Modificación (o extensión) de Historia de Usuario (Nro. y Nombre): | |
| Usuario: | Iteración Asignada: |
| Prioridad en Negocio: (Alta / Media / Baja) | Puntos Estimados: |
| Riesgo en Desarrollo: (Alto / Medio / Bajo) | Puntos Reales: |
| Descripción: | |
| Observaciones: | |

FIGURA 9: Diagrama General de las Historias de Usuario

Fuente: (extremeprogramming, s.f.)

- **Número:** Número de historia de Usuario
- **Nombre de Historia de Usuario:** Característica Principal del Sistema
- **Usuario:** El Interesado
- **Iteración Asignada:** Cambio Hechos
- **Prioridad de Negocio:** Prioridad de la Característica Alta/Medio/Bajo
- **Riesgo en Desarrollo:** Estimación de Riesgo Alta/Medio/Bajo
- **Puntos Estimados:** Tiempo estimado del Programador
- **Puntos Reales:** Tiempo Estimado del Equipo
- **Descripción:** Descripción global de la Característica Principal del Sistema
- **Observaciones:** Cualquier observación o dudas

3.1.5 DIAGRAMA ENTIDAD RELACIÓN

A continuación se muestra el diagrama Entidad Relación de la Base de Datos del Sistema de Gestión de Inventarios, para el cual se ha utilizado el modelador de datos Toad Data Modeler. BDD de referencia proporcionado por el departamento de informática GPI.

3.2 DESCRIPCIÓN Y FUNCIONAMIENTO DE LOS MÓDULOS

A continuación se muestra la descripción y funcionamiento de los módulos de Administración y seguridad, módulo de Catálogos (Gestión de Activos Fijos/Servicios) y movimiento de Inventarios.

3.2.1 DESCRIPCIÓN DEL MÓDULO DE SISTEMA

El módulo de administración se encarga de todas las funcionalidades de carácter administrativo de la aplicación y la seguridad está basada en permisos otorgados de acuerdo al rol y tipo de usuario.

De acuerdo al tipo de acceso y autorización se muestra el árbol de opciones de los módulos.

- **Funcionamiento del Módulo de Administración**

Se aprobará al usuario el ingreso al sistema si la autenticación es válida, de acuerdo al tipo de acceso y autorización el usuario podrá realizar las operaciones asignadas.

Solo un usuario administrador podrá acceder al módulo de administración y Seguridad donde podrá realizar las acciones que se detallan a continuación en la tabla.

TABLA 4: Funcionalidad de Módulo de Administración

| Paso | Acción |
|------|---|
| 1 | Ingresar como usuario administrador |
| 2 | Menú Administración |
| 3 | Seleccionar las Opciones: 1) Administrar Gestión de Inventarios 2) Reportes |
| 3.1 | Administrar Gestión de Inventarios 1) Catálogo 2) Movimientos 3) Ítems de Inventario 4) Stock |
| 3.2 | Reportes 1) Visualizar Reportes |

Fuente: Propia

La ilustración de los procesos y casos de uso del módulo de administración se muestran en el capítulo 4 en el documento de arquitectura que genera la metodología RUP y las Historias de Usuario que genera la Metodología SCRUM.

- **Descripción del Módulo de Reportes**

Este módulo permitirá configurar las opciones de manera impresa del reporte y despliegue en pantalla.

El módulo de Reportes genera los reportes disponibles para la gestión de inventarios en formato PDF.

TABLA 5: Funcionamiento Módulo de Reportes

| Paso | Acción |
|------|--|
| 1 | Ingresar como usuario Administrador |
| 2 | Seleccionar al módulo de Gestión de Inventarios |
| 3 | Seleccionar una de las siguientes Opciones: 1. Listado de Categorías (CPC Nivel <=5) 2. Listado de Productos Tipos (CPC Nivel 8) |

| | |
|------------|--|
| | <ul style="list-style-type: none"> 3. Listado de Productos (CPC Nivel 9) 4. Listado de Proveedores 5. Listado de Establecimientos 6. Listado de Movimientos de Inventario 7. Listado de Existencias |
| 3.1 | Listado de Categorías Por Nombre, Por Código |
| 3.2 | Listado de Productos Tipos Por Nombre Por Código |
| 3.3 | Lista de Productos Por Nombre Por Código |
| 3.4 | Listado de Proveedores Por Nombre Por Nombre Comercial Por CI/RUC |
| 3.5 | Lista de Establecimientos Por Descripción |
| 3.6 | Listado de Movimientos de Inventario Lista de Entradas Lista de Salidas Lista de Traslados Lista de Traspasos |
| 3.7 | Listado de Existencias Por Nombre, Por Establecimiento |

Fuente: Propia

La ilustración de los procesos, y casos de uso del módulo de Reportes de Inventarios se muestran en el capítulo 4 en el documento de arquitectura que genera la metodología RUP (opcional) y las Historias de Usuario que genera la Metodología SCRUM.

3.2.2 DESCRIPCIÓN DEL MÓDULO DE CATÁLOGO

El módulo catálogo muestra de manera predeterminada todos los productos que tienen el Gobierno Provincial de Imbabura, existencias y control de stock. Además del Catálogo de productos según el Clasificador Central Productos (CPC) para el registro de todos los productos existentes, Tipos de Productos y la Categoría a la que pertenece según el nivel o nodo en el que se encuentre.

- Añade en "Catálogo"
- Personaliza las opciones del catálogo
- Guarda los cambios
- Elimina elementos del Catálogo

TABLA 6: Funcionamiento Módulo de Catálogos

| Paso | Acción |
|------|---|
| 1 | Ingresar como usuario Administrador |
| 2 | Ingresar al módulo de Gestión de Inventarios |
| 3 | Seleccionar una de las siguientes Opciones: 1. Categorías (CPC Nivel <=5) 2. Productos Tipos (CPC Nivel 7) 3. Productos (CPC Nivel 9) 4. Proveedores 5. Establecimientos |
| 3.1 | Categorías Crear, Leer, Actualizar y Borrar |
| 3.2 | Productos Tipos Crear, Leer, Actualizar y Borrar |
| 3.3 | Productos Crear, Leer, Actualizar y Borrar |
| 3.2 | Proveedores Crear, Leer, Actualizar y Borrar |
| 3.4 | Establecimientos (Localizaciones) Crear, Leer, Actualizar y Borrar |

Fuente: Propia

La ilustración de los procesos, y casos de uso del módulo de Catálogo se muestran en el capítulo 4 en el documento de arquitectura que genera la metodología RUP y las Historias de Usuario que genera la Metodología SCRUM.

3.2.3 FUNCIONAMIENTO DEL MÓDULO DE MOVIMIENTO DE INVENTARIOS

Los movimientos de inventario son registrados automáticamente por el sistema y resultan de las operaciones de Entradas y Salidas, además de estos movimientos existen otros tipos de operaciones que implican también movimientos que deben ser asentados manualmente para que los datos arrojados posteriormente sean los reales, estas operaciones pueden ser por ejemplo, productos que se transfieren de una sucursal a otra (Trasposos), productos rotos (Pérdidas), devoluciones y Traslados.

Los movimientos pueden ser de entrada de productos, de salida de productos o de ambos al mismo tiempo.

- **Movimientos de entradas.**

Debe llevar un registro diario de la introducción de los suministros extranjeros, nacionales y nacionalizados de cada establecimiento autorizado.

- **Movimientos de Salidas**

Debe llevar un registro diario de la salida de los suministros.

- **Movimientos de Traslado**

Debe llevar un registro del traslado de los suministros entre las tiendas y bodegas de una misma empresa.

- **Movimientos de Traspaso**

Se deberá contar con el registro del Traspaso de suministros entre empresas.

- **Pérdidas y devoluciones**

El sistema deberá identificar los suministros que se verán afectados por destrucción, robo, traslados y traspasos.

TABLA 7: Funcionamiento del módulo de Movimientos de Inventarios

| Paso | Acción |
|------------|--|
| 1 | Autenticación de usuario |
| 2 | Ingresar al módulo de Gestión de Inventarios |
| 3 | Seleccionar Movimientos de Inventarios |
| 3.1 | Movimientos de Inventario Listado de Todos los Movimientos de Inventario: Por Traslados Por Traspasos Por Entradas Por Salidas Por Pérdidas |
| 3.2 | Control de Stock Existencias Guías Reorden Ítems de Inventario |

Fuente: Propia

La ilustración de los procesos, y casos de uso del módulo de Movimiento de Inventarios se muestran en el capítulo 4 en el documento de arquitectura que genera la metodología RUP (opcional) y las Historias de Usuario que genera la Metodología SCRUM.

CAPÍTULO IV

4 DISEÑO Y DESARROLLO DEL APLICATIVO

4.1 INTRODUCCIÓN

Se utilizó el concepto de Metodologías Ágiles concretamente el uso de la metodología de desarrollo SCRUM combinando conceptos básicos de la Metodología de Desarrollo RUP (Rational Unified Process).

4.1.1 METODOLOGÍAS ÁGILES

Una metodología Ágil es un conjunto de principios y prácticas eficientes para modelar y documentar un proyecto de software.

Expone cuatro valores que son:

a. Ser capaz de responder a los cambios y no obsesionarse sobre el seguimiento de un plan

Es tener la capacidad de adaptación, no decir NO A LOS CAMBIOS, aceptar las sugerencias de los usuarios, sin por eso hacer un lado la planificación. (Proyectos Ágiles, n.d.)

b. Colaboración con el cliente sobre el contrato de negocio

Se trata de colaborar con el cliente el mayor tiempo no de luchar con él sobre un contrato minucioso, esto puede ser difícil ya que los clientes no están acostumbrados, ellos están acostumbrados a trabajar sobre un contrato con el que puedan defenderse si las cosas van mal. (Proyectos Ágiles, n.d.)

c. El software debe cumplir con las necesidades de negocio

Es decir el software debe trabajar con normalidad, la documentación solo sirve para dar soporte y debe existir solo la suficiente.

d. Relaciones sobre las personas y procesos

De las personas depende el éxito o el fracaso de un proyecto, es a las que se les debe motivar.

Estos valores han dado lugar a algunos principios los cuales son:

- a. Introspección, los equipos deben regularmente hacerse una revisión hacia sí mismos y sus procesos para intentar mejorar.
- b. Las mejores arquitecturas, requerimientos, y diseños emergen de equipos auto-organizados.
- c. Simplicidad
- d. Promover el desarrollo sostenible
- e. Atención continua a la excelencia y al buen diseño
- f. Software que trabaje es la mejor medida del progreso
- g. Las comunicaciones cara a cara son las mejores
- h. Gente de negocios y desarrolladores trabajan diariamente en conjunto
- i. Entregar regularmente software que trabaje
- j. Bienvenida a los cambios que puedan ocurrir
- k. La satisfacción del cliente

(SCRUM, 2014)

4.1.2 MODELADO ÁGIL

Se puede aprovechar el modelado de un proyecto RUP, esto es la documentación UML.

• Criterios para un modelado ágil

Un modelado Ágil debe seguir estos criterios:

- a. Deben dar valor positivo, deben servir realmente de ayuda para dar un software funcional.

- b. Deben solo cumplir su propósito y no más, no usar herramientas pesadas, por ejemplo las ofrecidas por Rational Rose, no ser tan estricto.
- c. Debe ser comprensible para su audiencia.
- d. Deben ser lo suficientemente precisos.

Ejemplos de metodologías Ágiles:

- Programación Extrema
- Scrum
- MSF 4.0 Microsoft
- RAD *
- Cristal
- RUP Ágil
- Otras...

Estas metodologías pueden combinarse, por ejemplo Programación Extrema y Scrum, RUP y Scrum, RAD puede integrarse con otras etc. (SCRUM, 2014)

4.1.3 ASPECTOS DE LAS METODOLOGÍAS ÁGILES

Requisitos para poder utilizar Scrum:

Los siguientes puntos son de especial importancia para la implantación de una gestión ágil de proyectos como Scrum:

- a. Cultura de empresa basada en trabajo en equipo, delegación, creatividad y mejora continúa.
- b. Compromiso del cliente. Scrum exige del cliente una alta implicación y una dedicación regular.
- c. Compromiso conjunto y colaboración de los miembros del equipo.

- d. Facilidad para realizar cambios en el proyecto.
- e. Equipo trabajando en un mismo espacio común para maximizar la comunicación.
- f. Estabilidad de los miembros del equipo. (SCRUM, 2014)

4.2 METODOLOGÍA SCRUM

4.2.1 INTRODUCCIÓN

Scrum es una metodología ágil, que puede ser usada para manejar el desarrollo de productos complejos de software, en esta metodología se usan prácticas iterativas e incrementales.

4.2.2 CARACTERÍSTICAS

- Scrum incrementa significativamente la productividad y reduce el tiempo de espera para ver los beneficios así como facilitar la adaptación de los sistemas desarrollados.
- Scrum es un enfoque basado en equipos, incrementa el desarrollo cuando los requerimientos cambian rápidamente.
- Scrum es un proceso que controla el caos entre los conflictos de interés y las necesidades.
- Scrum es un camino para mejorar las comunicaciones y maximizar la cooperación.
- Scrum es un camino para detectar la causa y solucionar cualquier problema en el desarrollo.
- Scrum puede implementarse al principio o a la mitad de un proyecto de desarrollo.

Scrum se basa en el equipo, en reuniones diarias presididas por el Scrum máster para establecer el estado del proyecto, el corazón de Scrum es la iteración y la productividad, que en cada iteración presenta una mejora del funcionamiento del producto final, en cada iteración se evalúa la tecnología y capacidades requeridas, diariamente se pueden modificar el enfoque si se encuentran nuevas dificultades y tratar de remediarlas. (SCRUM, 2014)

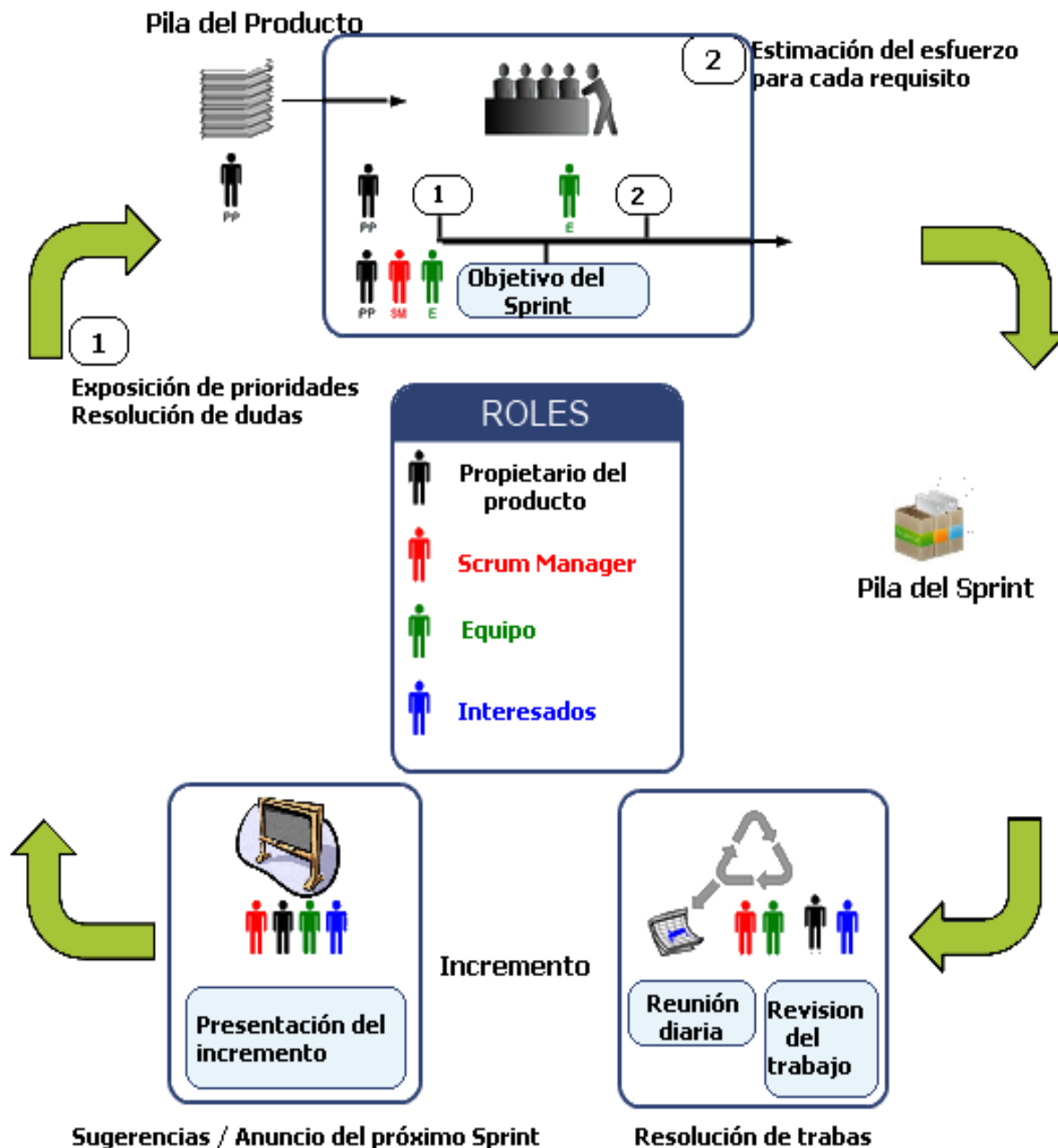


FIGURA 28: Proceso de trabajo del Scrum

Fuente: Scrum Metodologías Ágiles

4.2.3 ROLES DEL SCRUM

Scrum implementa sus procesos a través de tres roles considerados fundamentales, todas las responsabilidades de dirección son divididas en estos roles:

- **El propietario del producto**

Este rol, representa a la persona interesada en el estado del proyecto y el sistema resultante, el Backlog proveído a este rol representa una herramienta poderosa al proyecto, este lo usa para dar a los requerimientos la más alta prioridad, estos mismos son el más alto valor del negocio, este rol conoce cuales son las funcionalidades requeridas para resolver la problemática del negocio.

- **El Scrum máster**

Es el responsable de los procesos del Scrum, de que finalicé exitosamente, También puede ayudar al equipo a decidir cuáles de los elementos (backlog) deben desarrollarse en cada iteración (sprint).

- **El equipo**

Es el responsable del desarrollo, deben ser auto-dirigidos, auto-organizados, son los que sacan las características deseadas (el backlog) en cada iteración. (SCRUM, 2014)

4.2.4 IMPLEMENTACIÓN DE SCRUM

a. Comenzar el proceso de Scrum

Debemos seleccionar al equipo.

Recomendaciones

- No más de 6 – 9 miembros por equipo
- Si hay más miembros, romperlos en grupos

- Cada grupo enfocado en una sola área de trabajo
- Todo el staff trabajará en esta área

b. Nombrar al Scrum Máster

El Scrum máster es la persona que conduce las reuniones diarias, mide empíricamente los progresos, toma decisiones y resuelve los problemas de lentitud.

c. Identificar el acumulado

Acumulado (Backlog) es todo el trabajo pendiente para un área del producto, bien definido en sus términos.

d. Establecer y conducir la reunión diaria del Scrum

Diariamente se hace una reunión para checar el status del trabajo, donde el equipo informa de las actualizaciones, la reunión se enfoca en el trabajo que se está realizando.

4.2.5 VENTAJAS DEL SCRUM

- Se enfoca en equipos de trabajo
- Hay una comunicación diaria
- Ofrece una dirección basada en experiencia y de bajo nivel
- Hace los obstáculos visibles
- Se toman decisiones y se resuelven problemas en tiempo real.

4.2.6 HISTORIAS DE USUARIO (HERRAMIENTA DE LAS METODOLOGÍAS ÁGILES)

Se utilizan para hacer estimaciones de tiempo, evita documentación excesiva, muestran una breve descripción del comportamiento del sistema, emplea terminología del cliente sin lenguaje técnico, se realiza una por cada característica principal del sistema.

Las historias de usuario nacen de la necesidad de una mejor comunicación, de un lenguaje común entre todos los involucrados, desarrolladores, usuarios, etc. (Proyectos Ágiles, n.d.)

- **Ventajas de las historias de usuario sobre la toma tradicional de requerimientos o casos de uso.**
 - Comprensibles para usuarios y desarrolladores.
 - Trabajan para el desarrollo iterativo.
 - Fomentan una mejor comprensión acerca de lo que realmente se necesita.

4.2.7 SPRINT

Se ejecuta en bloques temporales cortos y fijos, cada iteración tiene que proporcionar un resultado completo y potencialmente entregable.

Notifica impedimentos, actualiza el estado de las actividades de la iteración (Sprint Backlog). Se realiza una reunión diaria donde cada miembro menciona el progreso y dificultades de las actividades para poder hacer las modificaciones necesarias.

El **Facilitador (Scrum Master)** actúa como medio de comunicación para expresar y resolver dudas, es el encargado de hacer cumplir los tiempos.

- **Elaboración del Sprint**

En base a las historias de usuario se definen las actividades a realizarse.

| Sprint | Inicio | Fin | Duración | | | | | | | | 20/07/2013 | 21/07/2013 | 22/07/2013 | 23/07/2013 | 24/07/2013 | 27/07/2013 | |
|--------|---|------------|------------|-----------|-------------|----|-------------|---------------|------------|------------|------------|------------|------------|------------|------------|------------|--|
| 1 | 20/07/2012 | 31/07/2013 | 15 días | | | | | | | | | | | | | | |
| Id | Tarea | Tipo | Estimación | Estado | Responsable | % | Impedimento | Observaciones | 20/07/2013 | 21/07/2013 | 22/07/2013 | 23/07/2013 | 24/07/2013 | 27/07/2013 | | | |
| H1 | GESTIONAR PRODUCTOS / ITEMS | | | | | | | | | | | | | | | | |
| | Diseñar el modulo para gestionar productos (CRUD, filtros, Grid Productos) | | 1 | Pendiente | Cristian | 0% | | | | | | | | | | | |
| | Agregar el componente de selección unica de tipo de producto en Grid Productos | | 1 | Pendiente | Cristian | 0% | | | | | | | | | | | |
| | Agregar el componente renderer de tipo de producto en Grid Productos | | 1 | Pendiente | Cristian | 0% | | | | | | | | | | | |
| | Diseñar el componente de selección de productos (filtros, grid, seleccion) | | 1 | Pendiente | Cristian | 0% | | | | | | | | | | | |
| | Diseñar el componente de consulta de productos (filtros, grid) | | 0,5 | Pendiente | Cristian | 0% | | | | | | | | | | | |
| H2 | GESTIONAR CLASIFICACION DE ITEMS | | | | | | | | | | | | | | | | |
| | Diseñar el modulo para gestionar los datos de categorias de productos (CRUD, filtros, Grid) | | 1 | Pendiente | Cristian | 0% | | | | | | | | | | | |
| | Diseñar el modulo para gestionar los datos de tipos de productos (CRUD, filtros, Grid) | | 1 | Pendiente | Cristian | 0% | | | | | | | | | | | |
| | Diseñar el componente de jerarquizacion de categorias de productos (filtros, tree) | | 1 | Pendiente | Cristian | 0% | | | | | | | | | | | |
| | Diseñar el componente de jerarquizacion de productos (CRUD, filtros,ADG) | | 1 | Pendiente | Cristian | 0% | | | | | | | | | | | |
| | Realizar la categorizacion de Productos en un modulo (Maestro - detalle) (Categorias de | | 3 | Pendiente | Cristian | 0% | | | | | | | | | | | |
| H3 | GESTIONAR DATOS DE BODEGAS | | | | | | | | | | | | | | | | |
| | Diseñar y registrar en el formulario de ingreso de los datos de las bodegas | | 2 | Pendiente | Cristian | 0% | | | | | | | | | | | |
| | Diseñar el modulo para gestionar los datos de bodegas (CRUD, filtros, Grid) | | 1 | Pendiente | Cristian | 0% | | | | | | | | | | | |
| | | | | | | 0% | | | | | | | | | | | |

FIGURA 29: Sprint

Fuente: Propia

- **Sprint 0**

Se sugiere hacer un sprint 0, que es donde se hacen los análisis y diseños previos, es un sprint para trazar la ruta del proyecto.

4.2.8 POILA DE PRODUCTOS (BACKLOG)

El Backlog o Pila de Producto es una lista priorizada de requisitos funcionales que pueden ser historias de usuario, cosas que el cliente quiere con su terminología, esta lista puede contener varios campos para cada ítem o producto, estos serían ID el cual es el identificador único del producto, su nombre, la importancia que le da el dueño del producto, la estimación en horas, como se puede probar que la funcionalidad está cubierta y nota, se pueden agregar más campos, categoría de actividad (análisis, diseño etc.), usuario de la actividad, etc.

| Pila de Producto (ejemplo) | | | | | |
|----------------------------|-----------------------------------|------|------|--|--|
| ID | Nombre | Imp. | Est. | Como probarlo | Notas |
| 1 | Depósito | 30 | 5 | Entrar, abrir página de depósito. depositar 10€. ir a página de balance y comprobar que se ha incrementado en 10€ | Necesita un diagrama UML. No preocuparse por encriptación aun |
| 2 | Ver tu historial de transacciones | 10 | 8 | Entrar, ver transacciones. Realizar un depósito de 10€. Ir a transacciones y comprobar que se ha actualizado con el nuevo depósito | Utilizar paginación para no hacer consultas muy grandes a la BB.DD. Diseño similar a la página de usuario. |

FIGURA 11: Pila de Producto
Fuente: Metodologías Ágiles SCRUM

En base a las historias reunidas se seleccionan conforme a varios criterios (alcance, estimación, importancia) las que van a ir en el sprint, en base a estas se definen las actividades que darían cumplimiento a esas historias, se descomponen en sub-actividades. (SCRUM, 2014)

4.2.9 DOCUMENTACIÓN GENERADA METODOLOGÍA SCRUM

- **Objetivos de las Iteraciones (Historias de Usuario)**

TABLA 8: H1: Definición del modelo de negocio y plan de proyecto

| Historia de Usuario | |
|--|--|
| Número: 01 | Nombre: Definición del modelo de negocio y plan de proyecto |
| Usuario: Analista | Iteración Asignada: 01 |
| Prioridad en negocio: Alta | Puntos Estimados: 2 |
| Riesgo en Desarrollo: Alto | Puntos Reales: 1 |
| Descripción: | |
| Clasificar de forma general los requerimientos del Usuario. | |
| Determinar el alcance e impacto del proyecto. | |
| Determinar la factibilidad del proyecto desde el punto de vista del negocio. | |
| Observaciones: | |
| Fase Incepción Actas de Trabajo Documento de Visión | |

Fuente: Propia (Proyectos Ágiles, n.d.)

TABLA 9: H2:.Definición del modelo de negocio y plan de proyecto

| Historia de Usuario | |
|--|--|
| Número: 02 | Nombre: Definición del modelo de negocio y plan de proyecto |
| Usuario: Analista | Iteración Asignada: 01 |
| Prioridad en negocio: Alta | Puntos Estimados: 2 |
| Riesgo en Desarrollo: Alto | Puntos Reales: 1 |
| Descripción: | |
| Clasificar de forma general los requerimientos del Usuario. | |
| Determinar el alcance e impacto del proyecto. | |
| Determinar la factibilidad del proyecto desde el punto de vista del negocio. | |
| Observaciones: | |
| Fase Elaboración Actas de Trabajo Documento de Visión | |

Fuente: Propia (Proyectos Ágiles, n.d.)

TABLA 10: H3. Desarrollar el documento de Arquitectura.

| Historia de Usuario | |
|---|--|
| Número: 03 | Nombre: Desarrollar el documento de Arquitectura. |
| Usuario: Analista | Iteración Asignada: 01 |
| Prioridad en negocio: Alta | Puntos Estimados: 2 |
| Riesgo en Desarrollo: Alto | Puntos Reales: 1 |
| Descripción: | |
| Definir arquitectura y herramientas. | |
| Realizar la lista de riegos técnicos. | |
| Prototipo temprano para revisión de usuarios. | |
| Observaciones: | |
| Fase Elaboración Desarrollar Prototipo | |

Fuente: Propia (Proyectos Ágiles, n.d.)

TABLA 11: H4. Implementar y probar los casos de uso. Iteración 1

| Historia de Usuario | |
|---|--|
| Número: 04 | Nombre: Implementar y probar los casos de uso |
| Usuario: Analista | Iteración Asignada: 01 |
| Prioridad en negocio: Medio | Puntos Estimados: 2 |
| Riesgo en Desarrollo: Alto | Puntos Reales: 1 |
| Descripción: | |
| Implementación de todas las características claves obtenidas de la arquitectura realizada previamente y desde la perspectiva del usuario. | |
| Observaciones: | |
| Fase Construcción | |

Fuente: Propia (Proyectos Ágiles, n.d.)

TABLA 12: H5. Implementar y probar los casos de uso. Iteración 2

| Historia de Usuario | |
|---|--|
| Número: 05 | Nombre: Implementar y probar los casos de uso |
| Usuario: Analista | Iteración Asignada: 02 |
| Prioridad en negocio: Alta | Puntos Estimados: 1 |
| Riesgo en Desarrollo: Alto | Puntos Reales: 1 |
| Descripción: | |
| Implementación y desarrollo de los casos de uso restantes, corregir errores y observaciones. Software revisado por los usuarios involucrados en cada uno de los módulos del sistema. | |
| Observaciones: | |
| Fase Construcción Producto (Software) | |

Fuente: Propia (Proyectos Ágiles, n.d.)

TABLA 13: H6. Implementar y probar los casos de uso. Iteración 3

| Historia de Usuario | |
|--|--|
| Número: 06 | Nombre: Implementar y probar los casos de uso |
| Usuario: Analista | Iteración Asignada: 03 |
| Prioridad en negocio: Alta | Puntos Estimados: 1 |
| Riesgo en Desarrollo: Alto | Puntos Reales: 1 |
| Descripción: | |
| Corregir y mejorar los defectos del reléase inicial. Desarrollo del sistema completo. Toda la funcionalidad del sistema completa para su liberación. | |
| Observaciones: | |
| Fase Construcción Producto (Software) | |

Fuente: Propia (Proyectos Ágiles, n.d.)

TABLA 14: H7. Ejecutar la Aplicación y hacer prueba de Errores

| Historia de Usuario | |
|--|---|
| Número: 07 | Nombre: Ejecutar la Aplicación y hacer prueba de Errores |
| Usuario: Analista | Iteración Asignada: 01 |
| Prioridad en negocio: Medio | Puntos Estimados: 2 |
| Riesgo en Desarrollo: Medio | Puntos Reales: 2 |
| Descripción: | |
| Sistema en fase de pruebas. Una vez terminada la fase de pruebas, implantación del Sistema en su primera versión. | |
| Observaciones: | |
| Fase Transición Software liberado | |

Fuente: Propia (Proyectos Ágiles, n.d.)

- **Requerimientos, Importancia y Estimación (Días) Pila de Producto**

A continuación se muestra las principales actividades del módulo, el artefacto Pila de Producto (Product Backlog), en base a la Metodología SCRUM.

TABLA 15: Pila de Producto GPI_INVENTARIOS

| PILA DE PRODUCTO ESTIMADA | | | | | | | |
|---------------------------|---|-------------|------------|-------|-------|-----------------|--------|
| GPI_INVENTARIOS | | | | | | | |
| ID | Nombre (Requerimiento) | Importancia | Estimación | ¿Cómo | Notas | Módulo | Sprint |
| 1 | FASE DE INCEPCIÓN | | | | | | |
| 1.1 | Analizar la Base de Datos | 100 | 1 | | | | 1 |
| 1.2 | Estudio de Herramientas | 100 | 1 | | | | 1 |
| 1.3 | Entender el funcionamiento de Flex | 100 | 0,5 | | | | 1 |
| 1.4 | Hacer un ejemplo basico de datos con Flex y PHP | 10 | 0,1 | | | | 1 |
| 2 | FASE DE ELABORACIÓN | | | | | | |
| 2.1 | Documento de Vision | 50 | 0,1 | | | | 2 |
| 2.2 | Documento Plan de Desarrollo | 50 | 1,2 | | | | 2 |
| 2.3 | Documento de Arquitectura | 50 | 0,1 | | | | 2 |
| 3 | FASE DE CONSTRUCCIÓN | | | | | | |
| 3.1 | Crear y estructurar el Proyecto | 100 | 1 | | | | 3 |
| 3.2 | Crear los componentes visuales de la Aplicación | 100 | 1 | | | | 3 |
| 3.3 | Implementar la conexión atravez de los servicios de la aplicacion | 100 | 1 | | | | 3 |
| 3.4 | Crear las diferentes clases necesarias para el trabajo | 80 | 1 | | | | 3 |
| 3.5 | Codificar y seguir la metodologia para la aplicación | 50 | 1 | | | | 3 |
| 3.6 | INGRESO AL SISTEMA | | | | | | |
| 3.6.1 | Login | 10 | 0,5 | | | GPI_SISTEMA | 4 |
| 3.6.2 | Logout | 10 | 0,5 | | | GPI_SISTEMA | 4 |
| 3.7 | FORMULARIO PRINCIPAL | | | | | | |
| 3.7.1 | Diseño de la Arquitectura del Sistema | 50 | 0,5 | | | GPI_SISTEMA | 5 |
| 3.7.2 | Diseño del Formulario Principal del Sistema | 40 | 1 | | | GPI_SISTEMA | 5 |
| 3.7.3 | Diseño del Arbol de Opciones del Sistema(Modulos) | 80 | 1 | | | GPI_SISTEMA | 5 |
| 3.8 | CATALOGO DE INVENTARIO | | | | | | |
| 3.8.1 | Gestionar Productos | 100 | 5 | | | GPI_INVENTARIOS | 6 |
| 3.8.2 | Gestionar Clasificacion de Productos (Tablas) | 100 | 1 | | | GPI_INVENTARIOS | 6 |
| 3.8.3 | Gestionar Datos de Bodegas (Establecimientos) | 50 | 0,5 | | | GPI_INVENTARIOS | 6 |
| 3.8.4 | Registrar Equivalencias de Unidades de Medida de Productos | 50 | 1 | | | GPI_INVENTARIOS | 6 |
| 3.8.5 | Consulta de Bodegas (Establecimientos) | 50 | 1 | | | GPI_INVENTARIOS | 6 |
| 3.8.6 | Consulta de Productos Según Categoría (<=Nivel 5 CPC) | 100 | 2 | | | GPI_INVENTARIOS | 6 |
| 3.8.7 | Consulta de Productos Según Producto Tipo (Nivel 7 CPC) | 100 | 2 | | | GPI_INVENTARIOS | 6 |
| 3.8.8 | Consulta de Productos (Nivel 9 CPC) | 100 | 2 | | | GPI_INVENTARIOS | 6 |
| 3.9 | MOVIMIENTOS DE INVENTARIO | | | | | | |
| 3.9.1 | Generar Ingreso de Items (Productos) | 100 | 5 | | | GPI_INVENTARIOS | 7 |
| 3.9.2 | Generar egreso de Items (Productos) | 100 | 5 | | | GPI_INVENTARIOS | 7 |
| 3.9.3 | Generar traslados de Items (Productos) | 80 | 5 | | | GPI_INVENTARIOS | 7 |
| 3.9.4 | Generar traspasos de Items (Productos) | 80 | 5 | | | GPI_INVENTARIOS | 7 |
| 3.9.5 | Registrar Bajas de Activos | 50 | 5 | | | GPI_INVENTARIOS | 7 |
| 3.9.6 | Registrar Adiciones o Mejoras de Activos | 40 | 5 | | | GPI_INVENTARIOS | 8 |
| 3.9.7 | Verificar Stock de Inventario | 100 | 5 | | | GPI_INVENTARIOS | 8 |
| 3.9.8 | Gestionar puntos de Reorden | 50 | 3 | | | GPI_INVENTARIOS | 8 |
| 3.9.9 | Gestionar alertas de Reorden | 50 | 3 | | | GPI_INVENTARIOS | 8 |
| 3.9.10 | Generar Kardex | 100 | 7 | | | GPI_INVENTARIOS | 8 |
| 3.10 | REPORTES | | | | | | |
| 3.10.1 | Lista de todos los Productos (Filtros) | 100 | 3 | | | GPI_INVENTARIOS | 9 |
| 3.10.2 | Lista de Existencias (Filtros) | 100 | 3 | | | GPI_INVENTARIOS | 9 |
| 3.10.3 | Reportes de Catalogo | 50 | 3 | | | GPI_INVENTARIOS | 9 |
| 3.10.4 | Reportes de Movimientos de Inventario (Kardex) | 100 | 3 | | | GPI_INVENTARIOS | 9 |
| 4 | FASE DE TRANSICIÓN | | | | | | |
| 4.1 | Ejecutar la Aplicación y hacer prueba de Errores | 100 | 2 | | | | 10 |
| 4.2 | Mostrar datos | 50 | 1 | | | | 10 |
| | | | 90 | | | | |

Fuente: Propia

- Seguimiento y Control del Proyecto (SPRINTS)

TABLA 16: Pila de Sprint 00 Requerimientos Iniciales

| PILA DE SPRINT | | | | | | | | | | | | | | | |
|----------------|--|------------|------------|------------|--------------|------|-------------|-------------|------------|------------|------------|------------|------------|------------|------------|
| Proyecto | | | | | | | | | | | | | | | |
| Sprint | | | | | | | | | | | | | | | |
| | 1 | 15/12/2012 | 05/01/2013 | Duración | 15 días | | | | | | | | | | |
| ID | Tarea | Tipo | Estimación | Estado | Responsable | % | Impedimento | Observación | 15/12/2012 | 16/12/2012 | 17/12/2012 | 18/12/2012 | 03/01/2013 | 04/01/2013 | 05/01/2013 |
| H1 | Inicio del proyecto / planeación sprint 0 | | | | | | | | | | | | | | |
| | Crear una lista de riesgos de negocio, identificar riesgos iniciales | | 1 | Finalizado | Scrum Master | 100% | | | 100% | - | - | - | - | - | - |
| | Crear el perfil del caso de negocio | | 1 | Finalizado | Scrum Master | 100% | | | 0% | 100% | - | - | - | - | - |
| | Capturar la tolerancia y criterios de aceptación | | 1 | Finalizado | Scrum Master | 100% | | | 0% | 0% | 100% | - | - | - | - |
| | Capturar las mejores prácticas, los estándares y las restricciones del cliente | | 1 | Finalizado | Scrum Master | 100% | | | 0% | 0% | 0% | 100% | - | - | - |
| | Identificar necesidades operacionales | | 1 | Finalizado | Scrum Master | 100% | | | 0% | 0% | 0% | 0% | 50% | 50% | - |
| | Identificar necesidades de mantenimiento | | 1 | Finalizado | Scrum Master | 100% | | | 0% | 0% | 0% | 0% | 50% | 50% | - |
| | Establecimiento del manejo y dirección del plan | | 2 | Finalizado | Scrum Master | 100% | | | 0% | - | - | - | 50% | 50% | - |
| | Establecer criterios de Aseguramiento de Calidad | | 3 | Finalizado | Scrum Master | 100% | | | 0% | - | - | - | 50% | 50% | - |
| | Establecer aspectos de calidad por descubrir | | 4 | Finalizado | Scrum Master | 100% | | | 0% | - | - | - | 50% | 50% | - |
| | Establecer una lista de aspectos de aseguramiento de calidad | | 5 | Finalizado | Scrum Master | 100% | | | 0% | - | - | - | 50% | 50% | - |
| | Plan total del proyecto | | 1 | Finalizado | Scrum Master | 100% | | | 0% | - | - | - | - | - | 100% |
| | Establecer controles del proyecto | | 1 | Finalizado | Scrum Master | 100% | | | 0% | - | - | - | - | - | 100% |
| | Establecer registro de lecciones aprendidas | | 1 | Finalizado | Scrum Master | 100% | | | 0% | - | - | - | - | - | 100% |
| | Junta la documentación de inicio de proyecto | | 1 | Finalizado | Scrum Master | 100% | | | 0% | - | - | - | - | - | 100% |
| H2 | Preparación de productos del backlog inicial | | | | | | | | | | | | | | |
| | Captura y priorización de los requerimientos iniciales del cliente | | 1 | Finalizado | Scrum Master | 100% | | | 0% | - | - | - | - | - | 100% |
| H3 | Revisión para el primer sprint | | | | | | | | | | | | | | |
| | Autorización de etapa (sprint) | | 1 | Finalizado | Scrum Master | 100% | | | 0% | - | - | - | - | - | 100% |

Fuente: Propia

TABLA 17: Pila de Sprint 01 Instalación y análisis

| PILA DE SPRINT | | | | | | | | | | | | | | | |
|----------------|---|------------|------------|------------|-------------|------|--------------|---------------|------------|------------|------------|------------|------------|------------|------------|
| Proyecto | | | | | | | | | | | | | | | |
| Sprint | Inicio | Fin | Duración | | | | | | | | | | | | |
| 1 | 03/01/2013 | 11/01/2013 | 7 días | | | | | | | | | | | | |
| ID | Tarea | Tipo | Estimación | Estado | Responsable | % | Impedimentos | Observaciones | 03/01/2013 | 04/01/2013 | 05/01/2013 | 06/01/2013 | 09/01/2013 | 10/01/2013 | 11/01/2013 |
| H1 | ANALISIS DE HERRAMIENTAS Y FRAMEWORKS | | | | | | | | | | | | | | |
| | Instalacion del Servidor Apache, PHP y Zend Framework | | 1 | Finalizado | Cristian | 100% | | | 100% | - | - | - | - | - | - |
| | Aprendizaje de Zend Framework | | 1 | Finalizado | Cristian | 100% | | | 0% | 100% | - | - | - | - | - |
| | Instalacion de Flash Builder for PHP | | 1 | Finalizado | Cristian | 100% | | | 0% | 0% | 100% | - | - | - | - |
| | Analisis de Frameworks para Flex | | 1 | Finalizado | Cristian | 100% | | | 0% | 0% | 0% | 100% | - | - | - |
| | Instalación de Herramientas | | 1 | Finalizado | Cristian | 100% | | | 0% | 0% | 0% | 0% | 50% | 50% | - |
| H2 | ANALISIS DE LA BASE DE DATOS | | | | | | | | | | | | | | |
| | Analisis del modelo de la base de datos | | 2 | Finalizado | Cristian | 100% | | | 0% | - | - | - | - | - | 100% |
| | Analisis de Información | | 3 | Finalizado | Cristian | 100% | | | 0% | - | - | - | - | - | 100% |
| | Ejecución del proyecto GPI_GESTION | | 4 | Finalizado | Cristian | 100% | | | 0% | - | - | - | - | - | 100% |
| | Inserción de datos | | 5 | Finalizado | Cristian | 100% | | | 0% | - | - | - | - | - | 100% |

Fuente: Propia

TABLA 18: Pila de Sprint 02 Estructura

| PILA DE SPRINT | | | | | | | | | | | |
|----------------|---|------------|------------|------------|-------------|------|--------------|---------------|------------|------------|------------|
| Proyecto | | | | | | | | | | | |
| Sprint | Inicio | Fin | Duración | | | | | | | | |
| 1 | 12/01/2013 | 16/01/2013 | 3 días | | | | | | | | |
| ID | Tarea | Tipo | Estimación | Estado | Responsable | % | Impedimentos | Observaciones | 12/01/2013 | 13/01/2013 | 16/01/2013 |
| H1 | | | | | | | | | | | |
| 1 | Crear y estructurar el Proyecto | | 1 | Finalizado | Cristian | 100% | | | 100% | - | - |
| 2 | Crear los componentes visuales de la Aplicación | | 1 | Finalizado | Cristian | 100% | | | 100% | - | - |
| 3 | Implementar la conexión a través de los servicios de la aplicación | | 1 | Finalizado | Cristian | 100% | | | 100% | - | - |
| 4 | Crear las diferentes clases necesarias para el trabajo con robolegs | | 1 | Finalizado | Cristian | 100% | | | 100% | - | - |
| 5 | Codificar y seguir la metodología para la aplicación con robolegs | | 1 | Finalizado | Cristian | 100% | | | 100% | - | - |
| 6 | Ejecutar la Aplicación y hacer prueba de Errores | | 1 | Finalizado | Cristian | 100% | | | 100% | - | - |
| 7 | Mostrar Datos | | 1 | Finalizado | Cristian | 100% | | | 100% | - | - |

Fuente: Propia

TABLA 19: Pila de Sprint 03 Documentación

| PILA DE SPRINT | | | | | | | | | | | | |
|----------------|---|------------|------------|------------|-------------|------|--------------|---------------|------------|------------|------------|------------|
| Proyecto | | | | | | | | | | | | |
| Sprint | Inicio | Fin | Duración | | | | | | | | 16/01/2013 | 17/01/2013 |
| 1 | 16/01/2013 | 17/01/2013 | 2 días | | | | | | | | | |
| ID | Tarea | Tipo | Estimación | Estado | Responsable | % | Impedimentos | Observaciones | 16/01/2013 | 17/01/2013 | | |
| H1 | | | | | | | | | | | | |
| 1 | Revisión del Anteproyecto de Tesis | | 1 | Finalizado | Cristian | 100% | | | 100% | - | | |
| 2 | Resumen del Contenido | | 1 | Finalizado | Cristian | 100% | | | 100% | - | | |
| 3 | Elaboración de la Presentación en Power Point | | 1 | Finalizado | Cristian | 100% | | | 100% | - | | |
| 4 | Estudio de los Módulos del Sistema | | 1 | Finalizado | Cristian | 100% | | | 100% | - | | |
| 5 | Estudio del Marco teórico del Anteproyecto de Tesis | | 1 | Finalizado | Cristian | 100% | | | 100% | | | |
| 6 | Defensa del Anteproyecto de Tesis | | 1 | Finalizado | Cristian | 100% | | UTN | 100% | | | |
| 7 | Corrección del Anteproyecto de Tesis | | 2 | Finalizado | Cristian | 100% | | | 0% | 100% | | |
| 8 | Diagrama de flujo del sistema GPI. Inventarios | | 2 | Finalizado | Cristian | 100% | | | 0% | 100% | | |
| 9 | Estructuración del contenido y presupuesto del Anteproyecto | | 2 | Finalizado | Cristian | 100% | | | 0% | 100% | | |
| 10 | Revisión del Documento de Guía de ActionScript | | 2 | Finalizado | Cristian | 100% | | | 0% | 100% | | |

Fuente: Propia

TABLA 20: Pila de Sprint 04 Metodología

| PILA DE SPRINT | | | | | | | | | | | | |
|----------------|--|------------|------------|------------|-------------|------|--------------|---------------|------------|------------|------------|------------|
| Proyecto | | | | | | | | | | | | |
| Sprint | Inicio | Fin | Duración | | | | | | | | 18/01/2013 | 19/01/2013 |
| 1 | 18/01/2013 | 19/01/2013 | 2 días | | | | | | | | | |
| ID | Tarea | Tipo | Estimación | Estado | Responsable | % | Impedimentos | Observaciones | 18/01/2013 | 19/01/2013 | | |
| H1 | | | | | | | | | | | | |
| 1 | Revisión de SCRUM | | 1 | Finalizado | Cristian | 100% | | | 100% | - | | |
| 2 | Revisión de Casos de Uso | | 1 | Finalizado | Cristian | 100% | | | 100% | - | | |
| 3 | Revisión del Manual de As3 | | 1 | Finalizado | Cristian | 100% | | | 100% | - | | |
| 4 | Diagramar casos de Uso | | 1 | Finalizado | Cristian | 100% | | | 10% | 90% | | |
| 5 | Hacer Historias de Usuario | | 1 | Pendiente | Cristian | 100% | | | 0% | 100% | | |
| 6 | Realización del Product Backlog | | 2 | Pendiente | Cristian | 100% | | | 0% | 100% | | |
| 7 | Revisión del Documento de Guía de ActionScript | | 2 | Finalizado | Cristian | 100% | | | 10% | 90% | | |

Fuente: Propia

TABLA 21: Pila de Sprint 05 Funcionalidad

| PILA DE SPRINT | | | | | | | | | | | | | | | | | | | | |
|----------------|---|------------|------------|-----------|-------------|------|------------|-------------|--|--|------------|------------|------------|------------|------------|--|--|--|--|--|
| Proyecto | | | | | | | | | | | | | | | | | | | | |
| Sprint | Inicio | Fin | Duración | | | | | | | | | | | | | | | | | |
| 5 | 23/01/2013 | 27/01/2013 | 5 días | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | |
| ID | Tarea | Tipo | Estimación | Estado | Responsable | % | mpedimento | bservacione | | | 23/01/2013 | 24/01/2013 | 25/01/2013 | 26/01/2013 | 27/01/2013 | | | | | |
| H1 | | | | | | | | | | | | | | | | | | | | |
| 1 | Componente de Ingreso Categorías de Productos | | 1 | Pendiente | Cristian | 100% | | | | | 90% | 0% | 0% | 10% | 0% | | | | | |
| 2 | Componente de Ingreso de Productos agrupado en categorías | | 1 | Pendiente | Cristian | 100% | | | | | 90% | 0% | 0% | 10% | 0% | | | | | |
| 3 | Componente de búsqueda de Categorías | | 1 | Pendiente | Cristian | 100% | | | | | 0% | 90% | 0% | 10% | 0% | | | | | |
| 4 | Agregar funcionalidad al componente de búsqueda de categorías por filtros | | 1 | Pendiente | Cristian | 100% | | | | | 0% | 0% | 25% | 0% | 75% | | | | | |
| 5 | Componente de búsqueda de productos | | 1 | Pendiente | Cristian | 100% | | | | | 10% | 90% | 0% | 0% | 0% | | | | | |
| 6 | Agregar funcionalidad al componente de búsqueda de productos por filtros | | 2 | Pendiente | Cristian | 100% | | | | | 0% | 0% | 25% | 75% | 0% | | | | | |
| 7 | Agregar funcionalidad para Ingresar Productos y categorías | | 3 | Pendiente | Cristian | 100% | | | | | 0% | 70% | 0% | 20% | 10% | | | | | |
| 8 | Agregar funcionalidad para Actualizar Productos y categorías | | 4 | Pendiente | Cristian | 100% | | | | | 0% | 0% | 70% | 20% | 10% | | | | | |
| 9 | Agregar funcionalidad para Eliminar Productos y categorías | | 5 | Pendiente | Cristian | 100% | | | | | 70% | 0% | 0% | 20% | 10% | | | | | |

Fuente: Propia

TABLA 22: Pila de Sprint 06 Categoría de Productos

| PILA DE SPRINT | | | | | | | | | | | | | | | | | | | | |
|----------------|--|------------|------------|-----------|-------------|------|------------|-------------|--|--|------------|------------|------------|------------|------------|--|--|--|--|--|
| Proyecto | | | | | | | | | | | | | | | | | | | | |
| Sprint | Inicio | Fin | Duración | | | | | | | | | | | | | | | | | |
| 6 | 30/01/2013 | 03/02/2013 | 5 días | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | |
| ID | Tarea | Tipo | Estimación | Estado | Responsable | % | mpedimento | bservacione | | | 30/01/2013 | 31/01/2013 | 01/02/2013 | 02/02/2013 | 03/02/2013 | | | | | |
| H1 | | | | | | | | | | | | | | | | | | | | |
| 1 | Agregar Componente de búsqueda Productos Tipo | | 1 | Pendiente | Cristian | 100% | | | | | 100% | 0% | 0% | 0% | 0% | | | | | |
| 2 | Agregar funcionalidad de búsqueda con filtros | | 1 | Pendiente | Cristian | 100% | | | | | 0% | 100% | 0% | 0% | 0% | | | | | |
| 3 | Agregar componente de listado de categorías de productos | | 1 | Pendiente | Cristian | 100% | | | | | 0% | 0% | 100% | 0% | 0% | | | | | |
| 4 | Agregar componente de categorización de | | 1 | Pendiente | Cristian | 100% | | | | | 0% | 0% | 0% | 100% | 0% | | | | | |
| 5 | Agregar funcionalidades para categorización de productos | | 1 | Pendiente | Cristian | 100% | | | | | 0% | 0% | 0% | 0% | 100% | | | | | |

Fuente: Propia

TABLA 23: Pila de Sprint 07 Productos

| PILA DE SPRINT | | | | | | | | | | | |
|----------------|--|------------|------------|-----------|-------------|------|--------------|---------------|------------|------------|--|
| Proyecto | | | | | | | | | | | |
| Sprint | Inicio | Fin | Duración | | | | | | | | |
| 7 | 04/02/2013 | 04/03/2013 | 30 | | | | | | | | |
| ID | Tarea | Tipo | Estimación | Estado | Responsable | % | Impedimentos | Observaciones | 04/02/2013 | 04/03/2013 | |
| H1 | | | | | | | | | | | |
| 1 | Agregar Componente de Búsqueda de Productos | | 2 | Pendiente | Cristian | 100% | | | 0% | 100% | |
| 2 | Realizar Funcionalidad de Búsqueda de Productos | | 3 | Pendiente | Cristian | 100% | | | 0% | 100% | |
| 3 | Agregar el componente GRID para mostrar e ingresar Productos | | 4 | Pendiente | Cristian | 100% | | | 0% | 100% | |
| 4 | Agregar el componente CRUD | | 4 | Pendiente | Cristian | 100% | | | 0% | 100% | |
| 5 | Agregar la funcionalidad DataManagement Services importante para la Realización del CRUD | | 5 | Pendiente | Cristian | 100% | | | 0% | 100% | |
| 6 | Realizar la funcionalidad de Creación e Inserción de Productos | | 5 | Pendiente | Cristian | 100% | | | 0% | 100% | |
| 7 | Realizar la funcionalidad para obtener de Datos de los Productos | | 6 | Pendiente | Cristian | 100% | | | 0% | 100% | |
| 8 | Realizar la funcionalidad para la inserción y actualización de Productos | | 7 | Pendiente | Cristian | 100% | | | 0% | 100% | |
| 9 | Realizar la funcionalidad para la eliminación de Productos. | | 8 | Pendiente | Cristian | 100% | | | 0% | 100% | |
| 10 | Crear un control personalizado para la visualización de datos provenientes de otras tablas dentro del componente GRID de Productos | | 9 | Pendiente | Cristian | 100% | | | 0% | 100% | |
| 11 | Realizar la funcionalidad para la visualización de datos proveniente de otras tablas para el componente GRID de Productos | | 10 | Pendiente | Cristian | 100% | | | 0% | 100% | |
| 12 | Creación de un control personalizado para selección de Productos | | 9 | Pendiente | Cristian | 100% | | | 0% | 100% | |
| 13 | Realizar la funcionalidad del control personalizado de selección de Productos | | 9 | Pendiente | Cristian | 100% | | | 0% | 100% | |
| 14 | Agregar el componente para la selección de Productos | | 9 | Pendiente | Cristian | 100% | | | 0% | 100% | |
| 15 | Agregar la funcionalidad PopPup para el componente de Selección de Productos | | 10 | Pendiente | Cristian | 100% | | | 0% | 100% | |
| 16 | Agregar el componente de Productos seleccionados | | 11 | Pendiente | Cristian | 100% | | | 0% | 100% | |
| 17 | Creación de un modelo de prueba para el CRUD de los Productos Seleccionados | | 12 | Pendiente | Cristian | 100% | | | 0% | 100% | |
| 18 | Agregar la funcionalidad para la mediación de los datos seleccionados al componente principal de Productos Seleccionados | | 13 | Pendiente | Cristian | 100% | | | 0% | 100% | |
| 19 | Investigar la mediación con RobotLegs en base a la herramienta Flex PopPupManager para el componente de selección de Productos | | 11 | Pendiente | Cristian | 100% | | | 0% | 100% | |
| 20 | Investigar la funcionalidad de un itemrender para la selección de varios Productos a la vez | | 12 | Pendiente | Cristian | 100% | | | 0% | 100% | |

Fuente: Propia

TABLA 24: Pila de Sprint 08 Tipos de Productos

| PILA DE SPRINT | | | | | | | | | | | | | | | | | | | | |
|----------------|---|------------|------------|-----------|-------------|------|------------|-------------|--|--|------------|------------|------------|------------|------------|--|--|--|--|--|
| Proyecto | | | | | | | | | | | | | | | | | | | | |
| Sprint | Inicio | Fin | Duración | | | | | | | | | | | | | | | | | |
| 8 | 05/03/2013 | 09/03/2013 | 5 días | | | | | | | | | | | | | | | | | |
| ID | Tarea | Tipo | Estimación | Estado | Responsable | % | mpedimento | bservacione | | | 05/03/2013 | 06/03/2013 | 07/03/2013 | 08/03/2013 | 09/03/2013 | | | | | |
| H1 | | | | | | | | | | | | | | | | | | | | |
| 1 | Agregar Componente de Búsqueda de Tipos Productos | | 0,5 | Pendiente | Cristian | 100% | | | | | 50% | 50% | 0% | 0% | 0% | | | | | |
| 2 | Funcionalidad de Búsqueda de Tipos Productos | | 0,5 | Pendiente | Cristian | 100% | | | | | 0% | 50% | 50% | 0% | 0% | | | | | |
| 3 | Agregar el componente CRUD | | 0,5 | Pendiente | Cristian | 100% | | | | | 0% | 0% | 100% | 0% | 0% | | | | | |
| 4 | Realizar la categorización de Productos | | 2 | Pendiente | Cristian | 100% | | | | | 0% | 30% | 0% | 50% | 20% | | | | | |

Fuente: propia

TABLA 25: Pila de Sprint 09 Jerarquía de Productos

| PILA DE SPRINT | | | | | | | | | | | | | | | | | | | | |
|----------------|---|------------|------------|-----------|-------------|------|------------|-------------|--|--|------------|------------|------------|------------|------------|--|--|--|--|--|
| Proyecto | | | | | | | | | | | | | | | | | | | | |
| Sprint | Inicio | Fin | Duración | | | | | | | | | | | | | | | | | |
| 9 | 12/03/2013 | 16/03/2013 | 5 días | | | | | | | | | | | | | | | | | |
| ID | Tarea | Tipo | Estimación | Estado | Responsable | % | mpedimento | bservacione | | | 12/03/2013 | 13/03/2013 | 14/03/2013 | 15/03/2013 | 16/03/2013 | | | | | |
| H1 | | | | | | | | | | | | | | | | | | | | |
| 1 | Realizar el renderer para extraer el nombre por el id(Reemplazo de LabelFunction) | | 0,2 | Pendiente | Cristian | 100% | | | | | 100% | 0% | 0% | 0% | 0% | | | | | |
| 2 | Realizar el componente de selección de Categorías Productos (PopPup) | | 0,2 | Pendiente | Cristian | 100% | | | | | 0% | 100% | 0% | 0% | 0% | | | | | |
| 3 | Realizar el componente de selección de Tipos Productos (PopPup) | | 0,4 | Pendiente | Cristian | 100% | | | | | 0% | 0% | 100% | 0% | 0% | | | | | |
| 4 | Agregar el componente de jerarquizacion de Productos | | 3 | Pendiente | Cristian | 100% | | | | | 0% | 0% | 0% | 50% | 50% | | | | | |

Fuente: Propia

TABLA 26: Pila de Sprint 10 Movimientos de Inventarios

| PILA DE SPRINT | | | | | | | | | | | | | | |
|----------------|--|------------|------------|-----------|-------------|------|--------------|---------------|------------|----|----|----|----|------------|
| Proyecto | | | | | | | | | | | | | | |
| Sprint | Inicio | Fin | Duración | | | | | | | | | | | |
| 10 | 01/06/2013 | 01/12/2013 | 90 | | | | | | | | | | | |
| ID | Tarea | Tipo | Estimación | Estado | Responsable | % | Impedimentos | Observaciones | 01/06/2013 | | | | | 01/12/2013 |
| H1 | | | | | | | | | | | | | | |
| 1 | Agregar Componente de Item de Inventarios | | 2 | Pendiente | Cristian | 100% | | | 0% | 0% | 0% | 0% | 0% | 100% |
| 2 | Realizar la funcionalidad del Módulo de Administración del Sistema | | 5 | Pendiente | Cristian | 100% | | | 0% | 0% | 0% | 0% | 0% | 100% |
| 3 | Realizar la funcionalidad del Modulo de Movimientos de Inventarios | | 5 | Pendiente | Cristian | 100% | | | 0% | 0% | 0% | 0% | 0% | 100% |
| 4 | Realizar Funcionalidad de Items de Inventarios | | 3 | Pendiente | Cristian | 100% | | | 0% | 0% | 0% | 0% | 0% | 100% |
| 5 | Agregar el componente GRID para mostrar e ingresar Items de Inventarios | | 4 | Pendiente | Cristian | 100% | | | 0% | 0% | 0% | 0% | 0% | 100% |
| 6 | Agregar el componente CRUD para Items de Inventarios | | 4 | Pendiente | Cristian | 100% | | | 0% | 0% | 0% | 0% | 0% | 100% |
| 7 | Agregar la funcionalidad DataManagement Services importante para la Realización del CRUD | | 5 | Pendiente | Cristian | 100% | | | 0% | 0% | 0% | 0% | 0% | 100% |
| 8 | Realizar la funcionalidad de Creación e Inserción de Movimientos de | | 5 | Pendiente | Cristian | 100% | | | 0% | 0% | 0% | 0% | 0% | 100% |
| 9 | Realizar la funcionalidad para obtener los Movimientos de Inventarios | | 6 | Pendiente | Cristian | 100% | | | 0% | 0% | 0% | 0% | 0% | 100% |
| 10 | Realizar la funcionalidad para la inserción y actualización de Establecimientos | | 7 | Pendiente | Cristian | 100% | | | 0% | 0% | 0% | 0% | 0% | 100% |
| 11 | Realizar la funcionalidad para la eliminación de Items de Inventario | | 8 | Pendiente | Cristian | 100% | | | 0% | 0% | 0% | 0% | 0% | 100% |
| 12 | Crear un control personalizado para la visualización de datos provenientes de otras tablas dentro del componente GRID de Proveedores (Entidades) | | 19 | Pendiente | Cristian | 100% | | | 0% | 0% | 0% | 0% | 0% | 100% |
| 13 | Realizar la funcionalidad para la visualización de datos proveniente de otras tablas para el componente GRID de Proveedores | | 20 | Pendiente | Cristian | 100% | | | 0% | 0% | 0% | 0% | 0% | 100% |
| 14 | Creación de un control personalizado para selección de Items de Inventario | | 9 | Pendiente | Cristian | 100% | | | 0% | 0% | 0% | 0% | 0% | 100% |
| 15 | Realizar la funcionalidad del control personalizado de selección de Tipos de | | 9 | Pendiente | Cristian | 100% | | | 0% | 0% | 0% | 0% | 0% | 100% |
| 16 | Agregar el componente para la selección de Proveedores | | 9 | Pendiente | Cristian | 100% | | | 0% | 0% | 0% | 0% | 0% | 100% |
| 17 | Agregar la funcionalidad PopUp para el componente de Selección de Proveedores | | 20 | Pendiente | Cristian | 100% | | | 0% | 0% | 0% | 0% | 0% | 100% |
| 18 | Agregar el componente de Items de Inventario seleccionados | | 11 | Pendiente | Cristian | 100% | | | 0% | 0% | 0% | 0% | 0% | 100% |
| 19 | Creación de un modelo de prueba para el CRUD de los Localizaciones | | 12 | Pendiente | Cristian | 100% | | | 0% | 0% | 0% | 0% | 0% | 100% |
| 20 | Agregar la funcionalidad para la mediación de los datos seleccionados al componente principal de de Items de Inventarios | | 13 | Pendiente | Cristian | 100% | | | 0% | 0% | 0% | 0% | 0% | 100% |
| 21 | Crear y agregar funcionalidad para componente de Guías de Reorden | | 21 | Pendiente | Cristian | 100% | | | 0% | 0% | 0% | 0% | 0% | 100% |
| 22 | Investigar la funcionalidad de un itemrender o funcionalidades para la extracción de datos de la base de datos por demanda para agilizar la extracción | | 22 | Pendiente | Cristian | 100% | | | 0% | 0% | 0% | 0% | 0% | 100% |
| 23 | Implementar la extracción de datos según demanda de usuario | | 13 | Pendiente | Cristian | 100% | | | 0% | 0% | 0% | 0% | 0% | 100% |
| 24 | Realizar la funcionalidad del Modulo de Reportes | | 10 | Pendiente | Cristian | 100% | | | 0% | 0% | 0% | 0% | 0% | 100% |
| 25 | Fase de Pruebas para solución de errores | | 24 | Pendiente | Cristian | 100% | | | 0% | 0% | 0% | 0% | 0% | 100% |
| 26 | Solución de errores del sistema | | 25 | Pendiente | Cristian | 100% | | | 0% | 0% | 0% | 0% | 0% | 100% |
| 27 | Fase final de pruebas. Presentación del proyecto. | | 26 | Pendiente | Cristian | 100% | | | 0% | 0% | 0% | 0% | 0% | 100% |

Fuente: Propia

4.3 METODOLOGÍA RUP

Es un proceso iterativo de organización de actividades en fases.

- a. **Incepción.-** Se describe la idea central del producto, se desarrolla el documento de visión.
- b. **Elaboración.-** Se detallan los Casos de Uso y se diseña la arquitectura del sistema.
- c. **Construcción.-** Desarrollar y diseñar el producto.
- d. **Transición.-** Requisitos cumplidos, se identifican y corrigen defectos, se capacita o sociabiliza el producto.

4.3.1 FASE DE INCEPCIÓN

En esta fase se describe principalmente el alcance del proyecto a través del documento de Visión. (RUP, An Introduction., 2014)

4.3.1.1 DOCUMENTO DE VISIÓN

- **Introducción.**

El propósito de éste documento es analizar las características del Sistema de Gestión de Inventarios (GPI_INVENTARIOS) del GPI.

La institución debe ser capaz de atender los distintos pedidos que le son realizados en un mínimo de tiempo y que también exista la facilidad de controlar el inventario de la misma.

Un sistema de gestión de inventarios informa al responsable del manejo de existencias cuándo hacer un pedido o hacer una distribución, la cantidad que debe solicitar o distribuir y cómo mantener un nivel de existencias adecuado para todos los productos a fin de evitar desabastecimientos y exceso de existencias.

Los detalles de cómo el sistema cubre los requerimientos se pueden observar en la especificación de los casos de uso que se encuentra en el documento de arquitectura.

- **Alcance**

El sistema permitirá automatizar la administración y control de inventarios en el GPI. Todo esto con la finalidad de generar acceso rápido a la información e integrar el proceso de control de inventarios a los procesos de planificación presupuestaria y planificación de compras, además se construirá la base de datos donde se alojará toda la información perteneciente al sistema y se desarrollará la interfaz gráfica del sistema enlazándolo a la base de datos para que la información pueda ser almacenada, todo esto mediante el uso de tecnología BPM y RIA.

El Sistema de Gestión de inventarios del GPI será desarrollado por Cristian Ipiates Flores.

- **Referencias**

- RUP (Rational Unified Process)
- SCRUM

- **Posicionamiento**

- **Oportunidad**

Este sistema permitirá al GPI manejar los inventarios de manera automatizada facilitando el ingreso de datos, almacenamiento de datos y permitiendo acceder mediante interfaces gráficas sencillas y amigables a dicha información en el momento que sea necesario. También nos permitirá conocer inmediatamente los costos en los que se incurrió al momento de realizar una compra.

- **Definición del Problema**

TABLA 27: Definición del Problema

| Definición del Problema | |
|-------------------------|--|
| Los problemas de | Una correcta administración y control de inventarios (ingreso de datos, almacenamiento) |
| Afecta a | Los responsables de las operaciones de los diferentes departamentos. <ul style="list-style-type: none"> ✓ Jefe de Bodega ✓ Institución ✓ Los proyectos ✓ Los procesos internos |

| | |
|-----------------------------------|---|
| El impacto asociado | <p>Facilitar el ingreso de datos, almacenar toda la información referente a las bodegas.</p> <p>Elaborar el proceso de movimientos de inventario de manera automatizada.</p> <p>Facilitar la obtención de información referente a bodegas y costos de productos.</p> |
| Una solución exitosa sería | <p>Un Sistema de Gestión de Inventarios que permita una adecuada administración del registro, compra y salida del inventario dentro de la institución, así el GPI reducirá el costo de manejo de materiales y podrá planificar las compras. Se conseguirá realizar despachos “reales” a los departamentos no basados en cantidades supuestas. Se mantendrá la información al día de los productos que deben ser eliminados de las existencias y que el departamento de contabilidad conozca el valor o capital inmovilizado que posee en sus bodegas.</p> |

Fuente: Propia

▪ Definición de la Posición del Producto

TABLA 28: Definición de la Posición del Producto

| | |
|----------------------------|---|
| Para | <p>Los responsables de las operaciones de los diferentes departamentos.</p> <ul style="list-style-type: none"> ✓ Jefe de Almacén ✓ Auxiliar Contable |
| Quienes | <p>Necesitan sistemas confiables para la supervisión y control de los procesos y el manejo eficiente de sus recursos.</p> |
| Nombre del producto | GpiInventarios |
| Que | <ul style="list-style-type: none"> ✓ Permitirá controlar las operaciones referentes al control de inventarios. ✓ Permitirá minimizar pérdidas ✓ Permitirá establecer responsabilidades sobre las acciones ejecutadas en los distintos procesos ✓ Facilitar el ingreso de datos, almacenar toda la información referente a las bodegas ✓ Facilitar la obtención de información automatizada referente a bodegas y productos adquiridos (existencias). |
| A diferencia de | Los sistemas manuales. |
| El producto | <ul style="list-style-type: none"> ✓ Será basado en código abierto, escalable, distribuido, extensible. ✓ Permite gestionar las operaciones referentes a la gestión de inventarios. |

Fuente: Propia

- **Descripción de Stakeholders (Participantes en el Proyecto) y Usuarios**

En esta sección se muestra e identifica y se hace el seguimiento de los participantes además de los requisitos y funcionalidades asignadas.

- **Resumen de Stakeholders**

TABLA 29: Resumen de Stakeholders

| Nombre | Descripción | Responsabilidades |
|--------------------------------|--|---|
| Cristian Ipiates Flores | Representante del desarrollo de sistema de gestión de inventarios. | El stakeholder realiza: <ul style="list-style-type: none"> ✓ Encargado de realizar el sistema de gestión de inventarios ✓ Análisis y planificación del proyecto ✓ Seguimiento del desarrollo del proyecto. ✓ Análisis de requisitos y funcionalidades |

Fuente: Propia

- **Resumen de Usuarios**

TABLA 30: Resumen de Usuarios

| Nombre | Descripción | Stakeholder |
|--------------------------|---|----------------|
| Jefe de Bodega | Supervisor del buen funcionamiento de bodega y de gestionar las incidencias de los pedidos de cada departamento, en contacto con el encargado de Logística. | Administración |
| Auxiliar Contable | Es un profesional dedicado a aplicar, realizar la información contable de la empresa, con la finalidad de diseñar e implementar instrumentos y mecanismos de apoyo a las directivas de la organización en el proceso de Toma de decisiones. | Administración |

Fuente: Propia

- **Descripción Global del Producto**
- **Perspectiva del producto**

Diseñar e implementar un sistema que realice el control de inventarios de manera automatizada.

- **Resumen de características**

TABLA 31: Beneficios que obtendrá el cliente

| Beneficios | Características |
|---|---|
| Obtener mayor eficiencia en el registro y generación de informes de toma de inventarios. | El diseño e implementación del módulo del costeo por procesos. |
| Obtener con facilidad, rapidez y eficacia los reportes del estado en bodegas. | El Diseño e implementación del módulo de gestión de inventarios. Con un Diseño de interfaz gráfico amigable acorde a las necesidades de la empresa industrial. |
| Obtener un informe inmediato de movimientos de inventario. | Trabajo conjunto entre el módulo de inventario, compras y presupuesto. |

Fuente: Propia

Con este proyecto se pretende desarrollar un sistema de gestión de Inventarios.

El proceso que se estipula que se logrará terminar es:

✓ **En Inventarios**

Se desea automatizar los ingresos, egresos, traslados y traspasos. También se desea diseñar una base de datos que soporte la información de bodegas que una vez concluida se podrá realizar consultas que proporcionarán información al jefe de bodega.

- **Sistema de Gestión de Inventarios. PROCESOS**

- **Introducción**

El rubro de inventarios es un aspecto medular dentro de la administración de una institución sea pública o privada.

Se hace un análisis sobre la estrecha relación que guarda las variaciones de los productos en inventarios con la determinación de características del producto.

Contiene la aplicación práctica de un sistema de gestión de inventarios, con el propósito de presentar de manera simplificada un detalle de reportes almacenes y de la salida y entrada de productos.

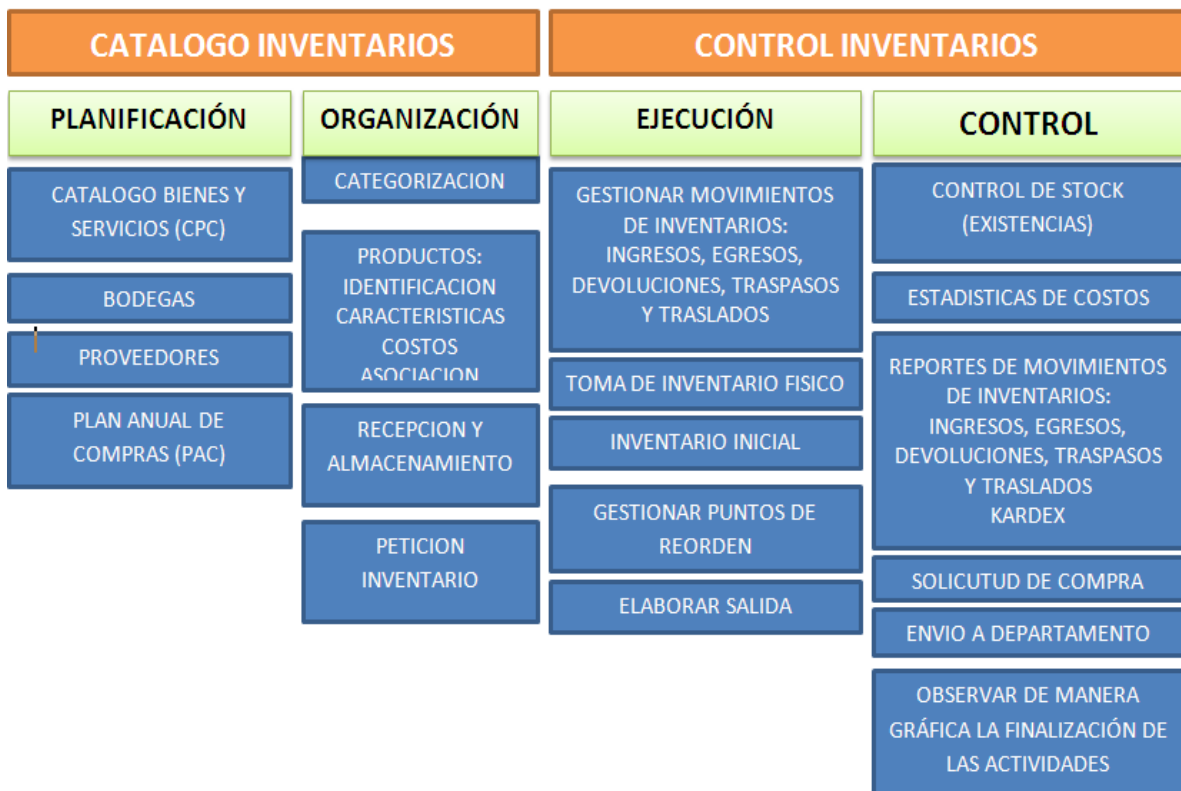


FIGURA 12: Diagrama de Procesos de Gestión de Inventarios

Fuente: Propia

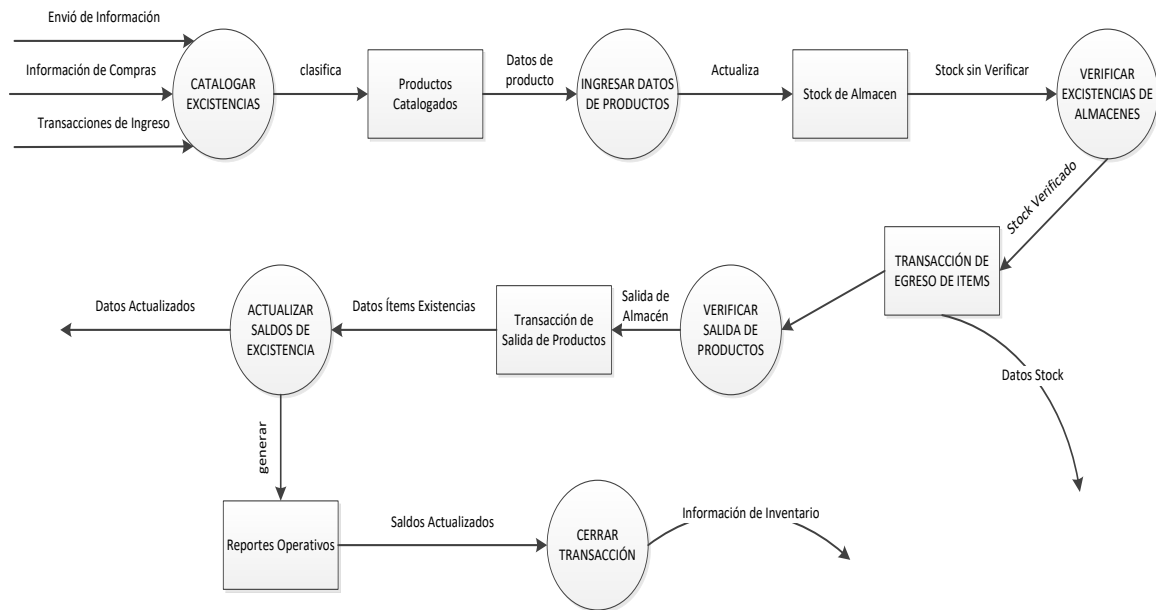


FIGURA 13: Diagrama de flujo de datos del Sistema de Gestión de Inventarios
Fuente: Propia Editada. (LUNA, 2014)

- **Especificación de procesos inventarios**

Definición. Define los procesos o funciones de un sistema.

Especificación de procesos del sistema de gestión de inventarios

I. CATALOGAR EXISTENCIAS

HACER MIENTRAS haya información-productos
LEER siguiente producto
MOSTRAR productos catalogados
FIN HACER

II. INGRESAR DATOS DE PRODUCTOS

SI el dato-producto está catalogado
Actualizar producto a SOCTK-ALMACEN
FIN SI

III. VERIFICAR EXISTENCIAS DE ALMACENES

SI el stock esta sin verificar
Verificar datos-stock en almacenes
OTRO
Stock verificado
Fin SI

IV. VERIFICAR SALIDA DE PRODUCTOS

SI el registro de salida existe el almacén
MOSTRAR Transacción de salidas de productos
Fin SI

V. ACTUALIZAR SALDOS DE EXISTENCIAS

MIENTRAS haya existencias de datos-ítems
MOSTRAR reportes operativos con datos-ítems
Fin MIENTRAS

MOSTRAR datos actualizados

VI. CERRAR TRANSACCIÓN

SI existen saldos actualizados
Cerrar transacción de datos de productos actualizados
Fin si
MOSTRAR información de inventario

- **Diagrama de transición de estados (dte) inventarios**

Definición El Diagrama de Transición de Estado (también conocido como DTE) enfatiza el comportamiento dependiente del tiempo del sistema. (glosario, 2014)



FIGURA 14: Diagrama de Transición de Estados del Sistema de Gestión de Inventarios

Fuente: Propia

Los componentes de un DTE son:

- **Estados:** comportamiento del sistema que es observable en el tiempo. Los sistemas tienen un estado inicial, pero pueden tener múltiples estados finales (mutuamente excluyentes).
- **Cambios de estados:** condiciones y acciones.
- **Diagramas Hipo Inventarios**

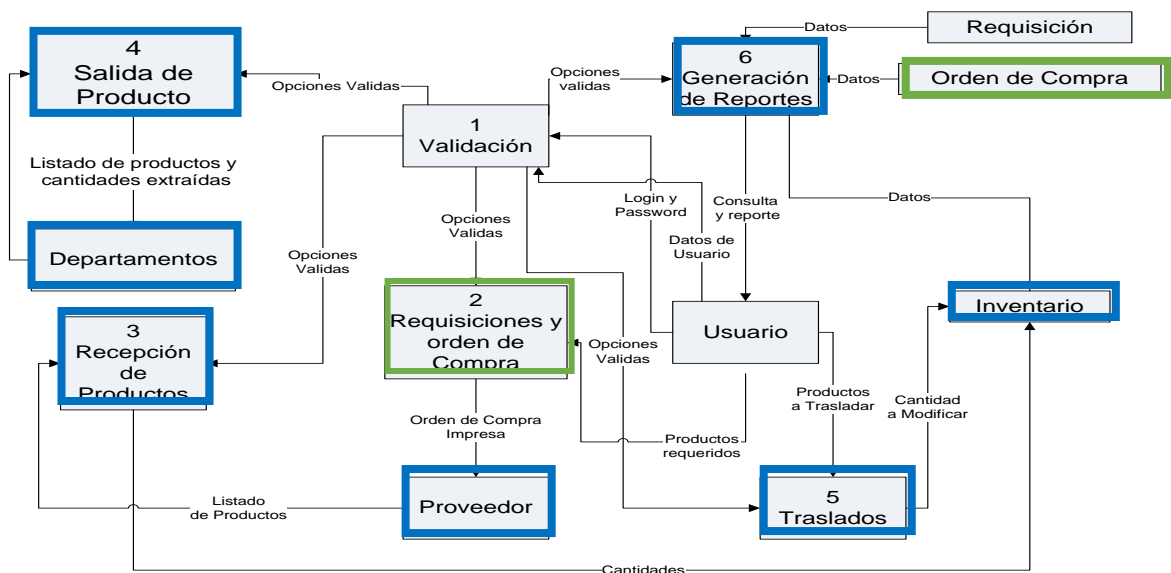
Definición. Diagrama de flujo que define las entradas, elaboración y salidas de un proceso.



FIGURA 15: Diagrama Hipo Inventarios

Fuente: Propia

▪ **Diagramas de Flujo de Inventarios**



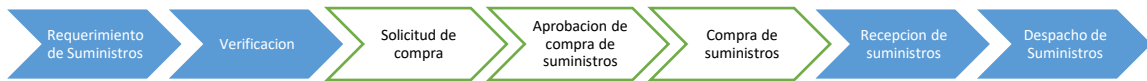


FIGURA 16: Diagrama de Flujo y Procesos de Inventario (Azul) y la relación con Compras (Naranja)

Fuente: Propia

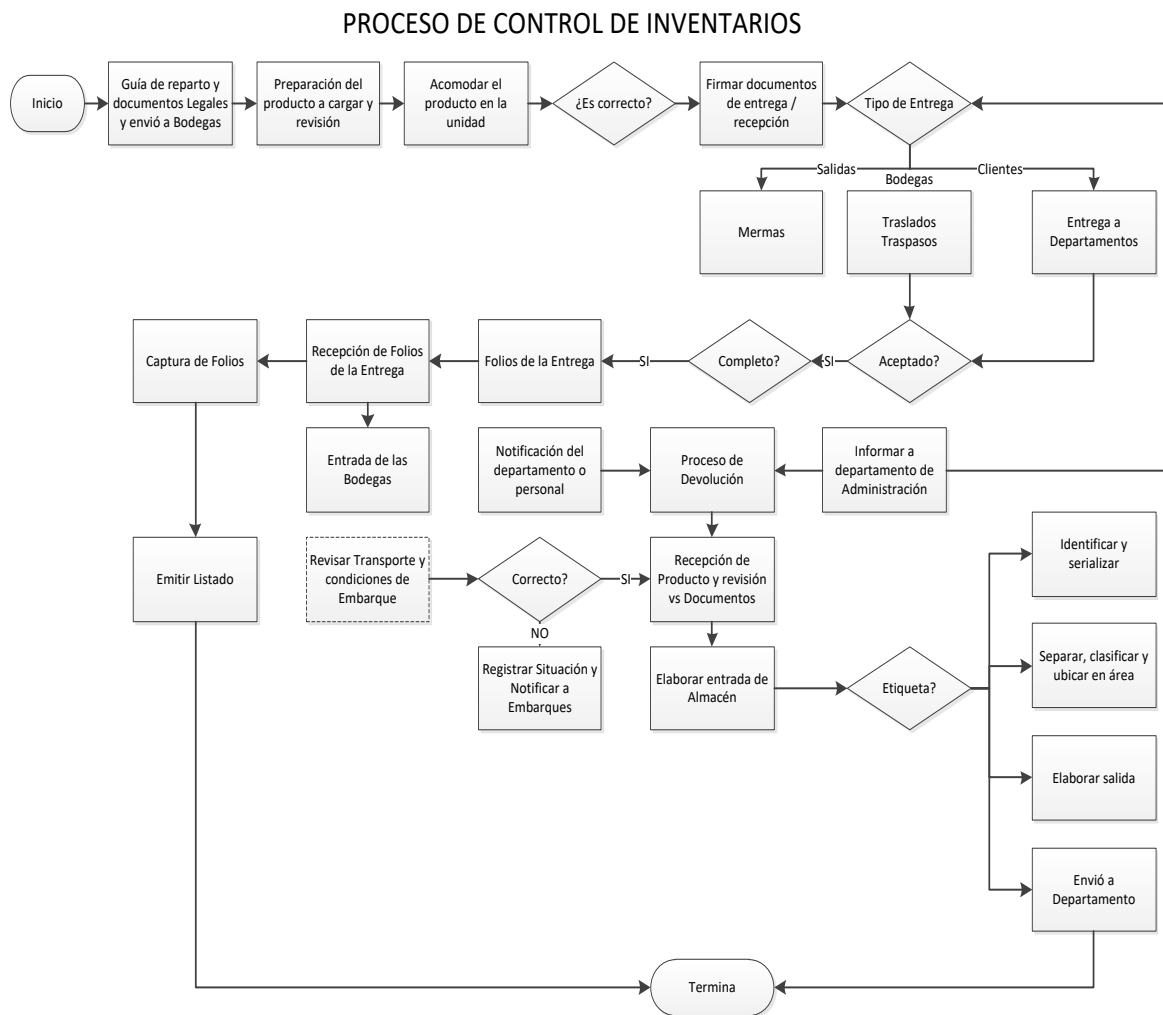


FIGURA 17: Diagrama de Flujo del Proceso de Control de Inventarios

Fuente: Propia

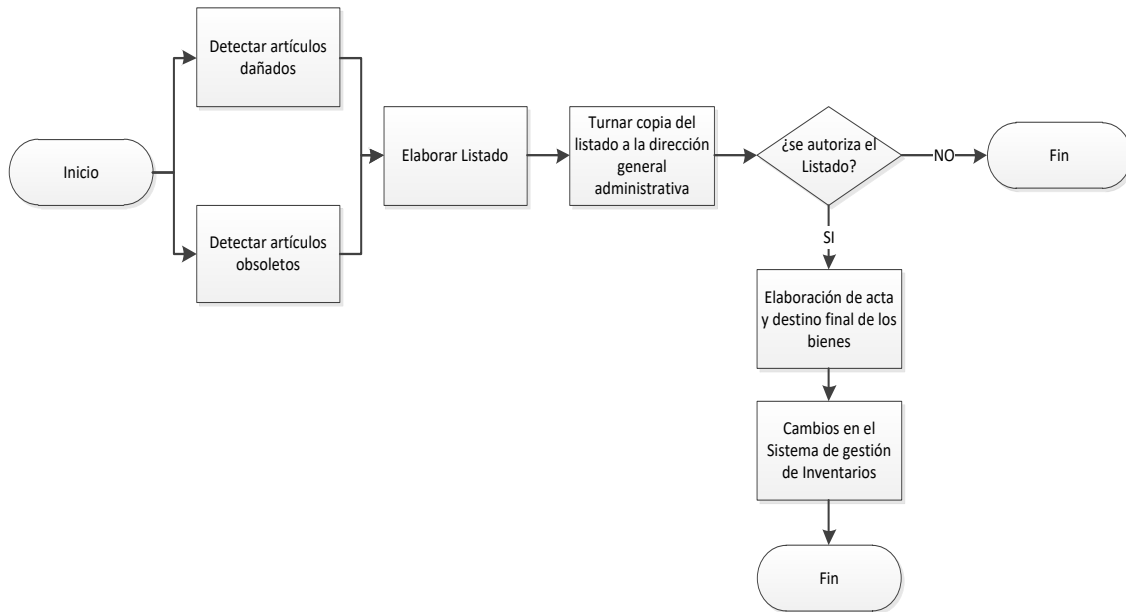


FIGURA 18: Diagrama de Flujo para la baja de Bienes por Obsolescencia y/o Daños
Fuente: Propia

▪ **Curso, dinámica, funcionamiento e integración con otros módulos.**

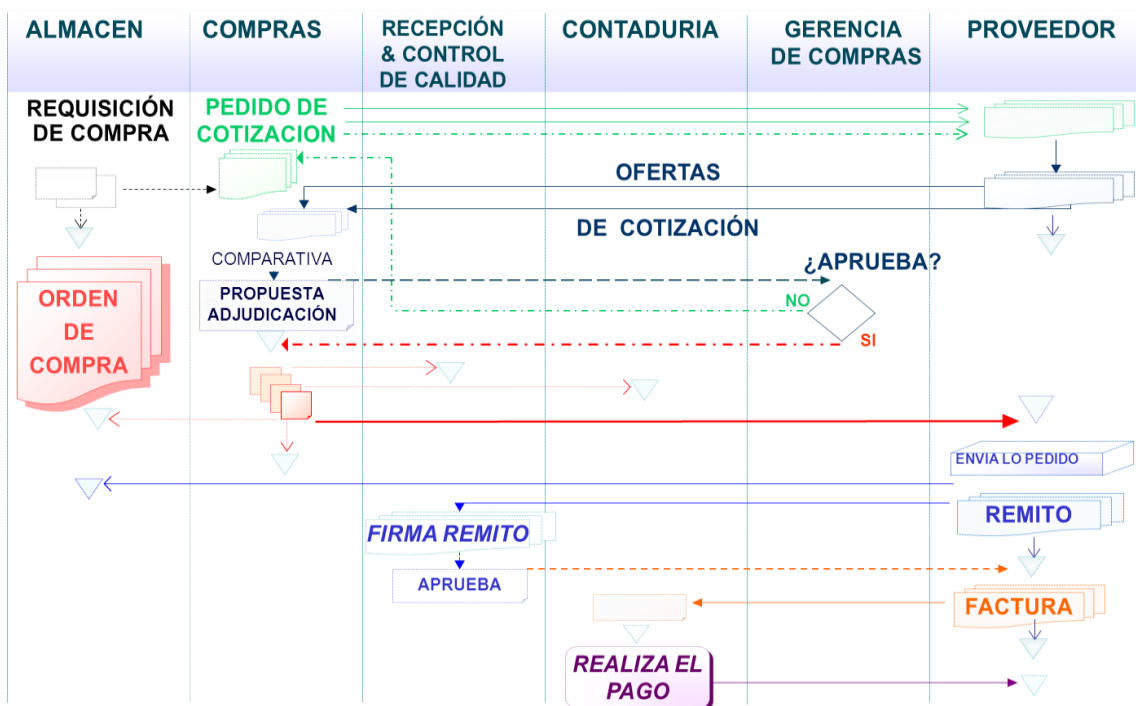


FIGURA 19: Curso, dinámica, funcionamiento e integración con otros módulos.
Fuente: (jotadeveloper.com, 2014)

- **Requerimientos**

TABLA 32: Requerimientos

| Hardware | |
|-----------------|--------------------------|
| N° | Descripción |
| 1 | Servidores |
| 2 | Computador Portátil |
| 3 | Impresora |
| Software | |
| N° | Descripción |
| 1 | Flash Builder (Licencia) |
| 2 | Flex |
| 3 | Zend |
| 4 | Otros |

Fuente: Propia

4.3.1.2 PLAN DE DESARROLLO DE SOFTWARE

- **Introducción**

La propuesta está basada en la metodología SCRUM con la integración de RUP como metodología ágil, en la que únicamente se procederá a cumplir con las tres primeras fases que marca la metodología RUP, constando únicamente en la tercera fase de dos iteraciones.

Es importante destacar esto puesto que se hará referencia a esta terminología RUP en este documento. Se incluirá el detalle para las fases de Inicio y Elaboración y adicionalmente se delinearán las fases de Construcción y Transición.

El enfoque del proceso RUP de acuerdo a las características del proyecto, formará parte de los artefactos (entregables) que serán generados.

- **Propósito**

Proporcionar una herramienta de software que satisfaga las necesidades de Gestión de Inventarios para cada una de las direcciones del GPI. Control de salida y entrada de suministros, traslados, traspasos, mermas, catálogo de productos y reportes de movimientos de inventarios.

Participantes:

Los usuarios del Plan de Desarrollo del Software son:

- El encargado de adquisiciones o compras, lo utilizará con el objetivo de determinar las existencias de materia prima en cada una de las divisiones de la institución, con el fin de realizar el suministro adecuado para las ejecuciones de las diferentes tareas.

- **Alcance**

Describe el desarrollo del Sistema de Gestión de Inventario, una vez comenzado el proyecto y durante la fase de Inicio se generará la primera versión del sistema.

- **Vista General del Proyecto**

- **Propósito, Alcance y Objetivos**

El Gobierno Provincial de Imbabura lleva a cabo la gestión de inventarios en la institución, con el objetivo de reducir los costos que conlleve a mantener altos niveles de inventarios y reducir la probabilidad de incertidumbre de quedarse sin existencias durante los tiempos de abastecimiento. El control de entradas y salidas, el control de existencias, la sistematización de costos de los diversos productos manejados en inventario, la falta de reportes necesarios para una óptima operatividad, son los factores que impulsan a obtener una aplicación de software, nueva, robusta y adecuado a las necesidades de la institución. Por los tipos de gestiones en la que se encuentra inmerso el GPI relacionado al control de inventario.

Por ello, es necesario la implementación del Sistema de Gestión de Inventarios mediante la integración de tecnología BPM y RIA, así como las bases de datos que recogen datos tanto para el monitoreo como seguimientos de los mismos, por tanto se tendría una gestión más rápida, automática y segura de los diferentes movimientos efectuados a nivel de inventario, además de contribuir con el desarrollo y crecimiento del GPI. (GPI, Documentación Institucional, 2014)

- **Suposiciones y Restricciones**

Las suposiciones y restricciones respecto del sistema que provienen después de la entrevista con el Stakeholder del GPI son:

Gestión de flujos de trabajo, seguridad de transacciones e intercambio de información entre cada dependencia hacia el servidor central.

El Sistema de Gestión de Inventarios debe diseñarse como un sistema totalmente integral con los actuales sistemas para ser utilizado posteriormente en cualquier institución, accediendo a la información centralizada.

- **Entregables del proyecto**

De acuerdo al proceso iterativo e incremental, todos los artefactos son modificables en el transcurso del proyecto y al final una versión definitiva y completa de cada uno de ellos.

a. Plan de Desarrollo del Software. El presente documento.

b. Glosario. Recopilación de términos poco conocidos o de difícil interpretación con su respectivo significado.

c. Modelo de Casos de Uso. Mediante diagramas de casos de uso.

d. Visión. Perspectiva del cliente, detalle de las características y necesidades del producto.

- e. **Especificaciones de Casos de Uso.** Explicación detallada de los casos de uso: flujo de eventos, alternativo de eventos, precondiciones y pos condiciones.
- f. **Prototipos de Interfaces de Usuario.** Vistazos iniciales del sistema.
- g. **Modelo de Datos.** Representación lógica de la estructura de los datos, relación de tablas y restricciones de integridad.
- h. **Casos de Prueba.** Condiciones de ejecución y resultados esperados.
- i. **Lista de Riesgos.** Lista de los riesgos distinguidos en el proyecto, organizados de acuerdo a la prioridad y soluciones.
- j. **Manual de Instalación.** Pasos necesarios para la correcta instalación del sistema.
- k. **Producto.** El sistema GPI_GESTION y el Módulo de Gestión de Inventarios GPI_INVENTARIOS se entregará en un CD con las instrucciones necesarias para su instalación.

- **Organización del Proyecto**

Se muestra como está estructurado el proyecto indicando participantes, interfaces y roles.

- **Participantes en el Proyecto**

Analistas.- Involucrados en el entorno de desarrollo del proyecto.

Programadores.- Amplios conocimientos en el lenguaje de programación PHP, la herramienta Flex, administración de la base de datos PostgreSQL, diseño UML, metodología RUP y SCRUM.

- **Roles y Responsabilidades**

A continuación se muestran las principales responsabilidades de cada uno de los miembros del equipo de desarrollo.

TABLA 33: Roles y Responsabilidades

| Puesto | Responsabilidad |
|---------------|---|
| Analista | Captura, especificación y validación de requisitos, interactuando con los interesados mediante entrevistas. Colaboración en la elaboración de las pruebas funcionales y el modelo de datos. |
| Programador | Son los responsables del desarrollo del proyecto, desarrollando los módulos del proyecto, creando versiones de los módulos hasta llegar a la versión final. |

Fuente: Metodología RUP

- **Gestión del Proceso**
 - **Seguimiento y Control del Proyecto**
 - **Gestión de Requisitos**

Los cambios en los requisitos serán evaluados para asegurar la integridad del sistema.

- **Control de Plazos**

Los tiempos del proyecto se definirán una vez iniciado el desarrollo del Sistema GPI_INVENTARIOS según las Metodologías SCRUM y RUP.

- **Control de Calidad**

Los defectos detectados en las revisiones y formalizados también vía electrónica, escrita o personal tendrán un seguimiento para asegurar la conformidad respecto de la solución de dichas deficiencias, este control se realizará en base a la Metodología SCRUM. (Proyectos Ágiles, n.d.)

- **Control de Plazos**

El seguimiento de actividades del proyecto será semanal, según la metodología SCRUM a través del Artefacto SPRINT de Producto y guiado por el jefe de proyecto.

- **Control de Calidad**

Los defectos detectados tendrán un seguimiento y solución para asegurar la funcionalidad del proyecto.

- **Referencias**

Documentación de Rational Unified Process, manuales de ayuda, tutoriales, etc.

4.3.2 FASE DE ELABORACIÓN Y CONSTRUCCIÓN

En estas fases se realiza el diseño básico de la arquitectura y modelos de datos de los módulos a implementar y se realiza los casos de uso con su respectiva descripción de cada uno de ellos.

4.3.2.1 DOCUMENTO DE ARQUITECTURA

- **Introducción**

El presente documento muestra el enfoque de la arquitectura del Sistema GpilInventarios haciendo uso de diferentes vistas arquitectónicas para así poder ilustrar las características más importantes del sistema. Siendo dicho artefacto de gran importancia ya que proporciona, entre otras cosas, una vista arquitectónica del sistema a todos los implicados en el mismo. (GPI, Documentación Institucional, 2014)

- **Propósito**

El Documento de Arquitectura de Software presenta la arquitectura del Sistema GpilInventarios a través de diferentes vistas, cada una de las cuales ilustra un aspecto en particular del software desarrollado. Se pretende de esta forma que el documento brinde al lector una visión global y comprensible del diseño general. (GPI, Documentación Institucional, 2014)

- **Alcance**

Este documento muestra la arquitectura en la cual está basado el Sistema GpilInventarios.

- **Visión General**

Arquitectura, presentación de 4+1 visto, desempeño del software y calidad del sistema.

- **Representación Arquitectónica**

A partir del estudio de las características particulares del sistema a desarrollar, siendo el mismo un módulo de GPI_GESTION; como parte de las decisiones arquitectónicas para definir la propuesta de arquitectura se decidió utilizar RUP para el Sistema de Gestión de Inventarios GpilInventarios. El porqué de esta decisión viene dado por:

- Un Sistema de Gestión de Inventarios está formado por un conjunto de procesos que se ejecutan y que son reutilizables para otros módulos como el de contabilidad, logística, entre otros.

El modelo presentado por RUP para mostrar la arquitectura del sistema utiliza el siguiente conjunto de vistas:

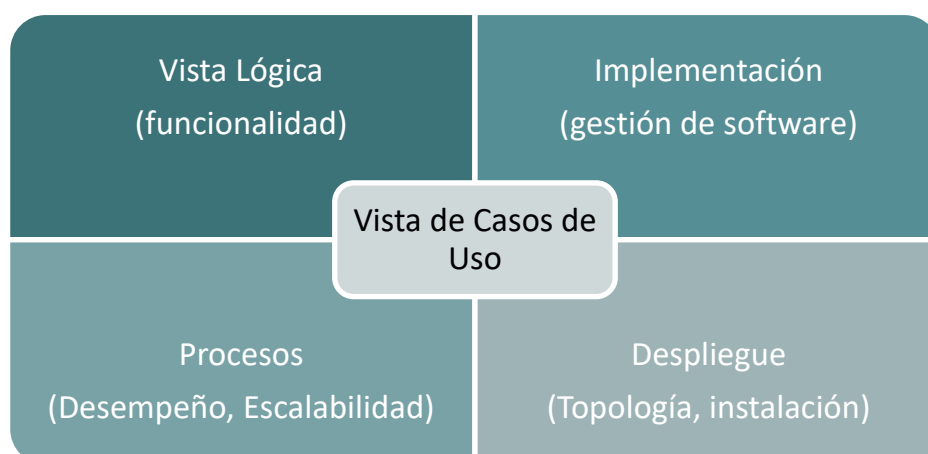


FIGURA 39: Modelo 4+1 RUP

Fuente: RUP

- **Vista de casos de uso:** casos de uso.
- **Vista lógica:** clases.
- **Vista de implementación:** capas de presentación, lógica y de representación de datos.
- **Vista de procesos:** interacción de las clases activas.
- **Vista de despliegue:** nodos, componentes y asociaciones.
- **Objetivos Arquitectónicos y Restricciones**
- **Objetivos Arquitectónicos**

El objetivo principal del Sistema GpilInventarios es diseñar e implementar el sistema de Gestión de Inventarios en el Gobierno Provincial de Imbabura integrándolo al módulo transaccional de administración y control de inventarios usando tecnología BPM y RIA, todo esto a efecto de optimizar el uso de los recursos en el Gobierno Provincial de Imbabura.

El Sistema GpilInventarios será implementado en las siguientes plataformas:

- Apache: Servidor web
- PHP: Lenguaje de Programación
- PostgreSQL: Base de datos
- ZEND: Framework MVC
- FLEX: Framework de Presentación

4.3.2.2 VISTA DE CASOS DE USO

Permite describir los escenarios o casos de uso que tienen un alto grado de importancia para la arquitectura puesto a que encapsulan gran parte de la funcionalidad central del sistema.

- **Casos de Uso Catálogo Inventario**

A continuación se presenta la vista de casos de uso de gpi_inventarios módulo CU Catálogo Inventario:

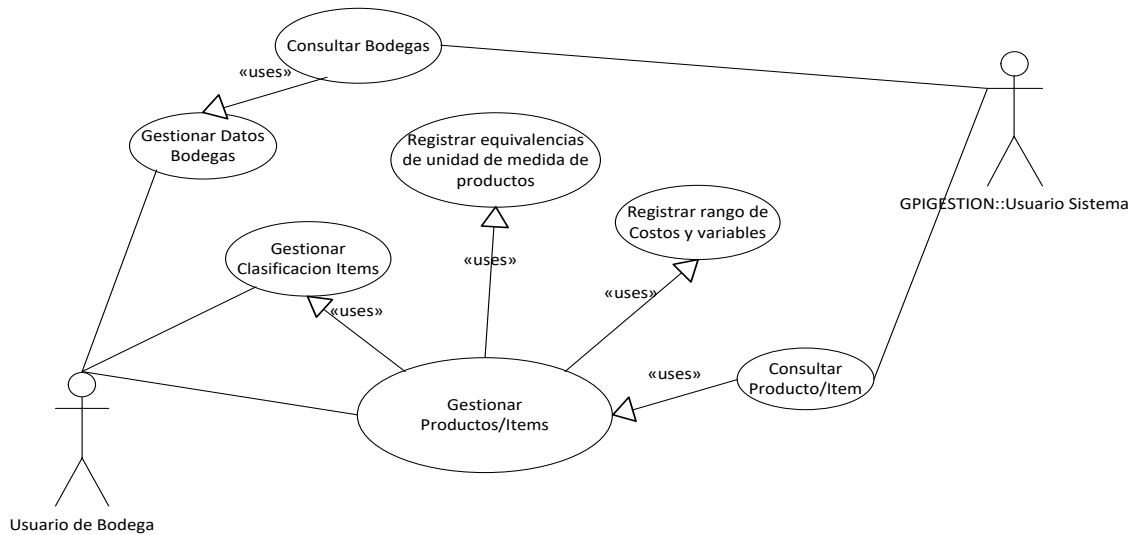


FIGURA 20: Vista de casos de uso de Gpi_inventarios módulo CU Catálogo Inventario

Fuente: Propia

Los siguientes casos de uso son de suma importancia para el funcionamiento del sistema puesto que constituye la base conceptual sobre la cual se realizarán las distintas operaciones de ingreso.

TABLA 34: CU Catálogo Inventario

| Referencia | Caso de Uso | Prioridad |
|----------------------|---|-----------|
| GPIINV-CU-001 | <p>Gestionar Datos Bodegas</p> <p>1.1 Breve Descripción</p> <p>Permite al jefe de bodega tener una gestión de todos los datos de las bodegas que existen en una institución (capacidad, requerimientos) para luego poder administrarlos en el funcionamiento de nuestro sistema, el jefe de almacén podrá crear, editar y eliminar datos de bodega(s) según sea el criterio de la institución.</p> <p>Flujo de Eventos</p> <p>1.2 Flujo Básico</p> <p>Muestra la interfaz “Gestionar Bodegas”.</p> <p>1.2.1 Crear Bodega</p> <p>1. Inserta registro Bodegas.</p> | 8 |

| | | |
|-----------------------------|---|----------|
| | <p>1.2.2 Editar Bodega</p> <ol style="list-style-type: none"> 1. El JB busca registros en la pantalla Bodegas. 2. El JB modifica el registro <p>1.2.3 Eliminar Bodega</p> <ol style="list-style-type: none"> 1. El JB busca registros en la pantalla Bodegas. 2. El JB selecciona los registros a eliminar. 3. El JB elimina registros seleccionados. <p>1.3 Flujos Alternativos</p> <p>1.3.1 < Error datos obligatorios > Validación de campos</p> <p>1.3.3 <Error selección bodega> Mensaje de alerta</p> <p>Requerimientos Especiales Ningún requerimiento en especial</p> <p>Precondiciones Autorizaciones Acceso GPI_ORGANIZACION</p> <p>Poscondiciones Lista de bodegas actualizada</p> <p>Puntos de extensión Módulo GPI_ORGANIZACION Establecimientos.</p> | |
| <p>GPIINV-CU-002</p> | <p>Gestionar Clasificación Ítems</p> <p>2.1 Breve Descripción</p> <p>Permite al jefe de bodega tener una gestión de Tipos de productos según Categorías de productos para luego poder administrarlos en el funcionamiento de nuestro sistema, el jefe de bodega podrá crear, editar y eliminar categorías y tipos de producto(s).</p> <p>Flujo de Eventos</p> <p>2.2 Flujo Básico</p> <p>Se visualiza la pantalla “Gestionar Clasificación Ítems” con los campos: código, nombre categorías o tipos.</p> <p>2.2.1 Crear Categorías Producto / Tipos</p> <ol style="list-style-type: none"> 1. Se crea el registro nuevo de Categorías Productos/Tipos con datos ingresados. <p>2.2.2 Editar Categorías Producto / Tipos</p> <ol style="list-style-type: none"> 1. El JB busca de Categorías Productos/Tipos en la interfaz “Gestionar Clasificación Ítems”. 2. El jefe de bodega presiona “editar” en las Categorías Productos/Tipos que desea modificar. 3. El sistema guarda los datos modificados y genera un mensaje “Actualización Completada” | <p>1</p> |

| | | |
|-----------------------------|--|-----------|
| | <p>2.2.3 Eliminar Categorías Producto / Tipos</p> <ol style="list-style-type: none"> 1. El JB busca de las Categorías Productos/Tipos en la interfaz. 2. El Jefe de bodega selecciona las Categorías Productos/Tipos que desea eliminar presionando la fila correspondiente. <p>2.3 Flujos Alternativos</p> <p>2.3.1 < Error datos obligatorios > Validación de campos</p> <p>2.3.3 <Error selección producto> Mensaje de alerta</p> <p>Requerimientos Especiales Ningún requerimiento en especial</p> <p>Precondiciones Autorizaciones Accesos Que existan Categorías Productos/Tipos ingresados a la base de datos</p> <p>Poscondiciones Categorías Productos/Tipos actualizados</p> <p>Puntos de extensión El sistema llama al caso de uso consultar producto</p> | |
| <p>GPIINV-CU-003</p> | <p>Consultar Bodega</p> <p>3.1 Breve Descripción Este caso de uso describe el procedimiento necesario para que un Jefe de Bodega y el Usuario de bodega puedan generar un reporte con un resumen del inventario de la institución.</p> <p>Flujo de Eventos</p> <p>3.2 Flujo Básico</p> <ol style="list-style-type: none"> 1. Este caso de uso es iniciado por un Jefe de Bodega y el Usuario de bodega. 2. El Jefe de Bodega y el Usuario de bodega hace la petición al sistema de generar el inventario. 3. El sistema revisa la base de datos. 4. El sistema retorna el reporte del inventario. <p>3.3 Flujos Alternativos</p> <p>3.3.1 <Información insuficiente> El sistema no encuentra alguna información para generar el reporte El sistema genera el reporte con los datos que posee.</p> <p>3.3.2 <Error en reporte></p> | <p>10</p> |

| | | |
|-----------------------------|--|----------|
| | <p>El sistema no puede generar el reporte. El sistema informa al Jefe de Bodega y al Usuario de bodega que no se pudo realizar la consulta.</p> <p>Precondiciones. Roles</p> <p>Poscondiciones. Se debe generar un reporte en PDF con los datos solicitados.</p> <p>Puntos de extensión CU consultar producto</p> | |
| <p>GPIINV-CU-004</p> | <p>Gestionar Productos Ítems</p> <p>4.1 Breve Descripción El JB gestiona del catálogo de productos ítems existentes en la institución</p> <p>Flujo de Eventos</p> <p>4.2 Flujo Básico Se visualiza la pantalla “Gestionar Producto” con los campos: código, tipo, descripción, nombre producto, categoría.</p> <p>4.2.1 Crear Producto 1. Se ingresa nuevo registro de productos.</p> <p>4.2.2 Editar Producto 1. El JB busca el producto en la pantalla Gestión de Productos. 2. El JB edita el producto. 3. El JB modifica el producto.</p> <p>4.2.3 Eliminar Producto 1. El JB busca el producto en la pantalla de Gestión de productos. 2. El JB selecciona fila para eliminar el producto(s). 3. El JB elimina productos seleccionados</p> <p>4.3 Flujos Alternativos</p> <p>4.3.1 < Error datos obligatorios > Validación de campos</p> <p>4.3.3 <Error selección producto> Mensaje de alerta</p> <p>Requerimientos Especiales Ningún requerimiento en especial</p> <p>Precondiciones Que existan productos ingresados a la base de datos</p> <p>Poscondiciones Lista de productos actualizada</p> <p>Puntos de extensión CU consultar productos</p> | <p>1</p> |

| | | |
|----------------------|---|----|
| GPIINV-CU-005 | Registrar Equivalencias de unidades de medida de productos 5.1 Breve Descripción Permite registrar las equivalencias entre las unidades de medidas en que pueden encontrarse un producto Ejemplo: Conversión de Litros a galones, De Kilogramos a Gramos, etc. | 5 |
| GPIINV-CU-006 | Registrar rangos de costos de variables 6.1 Breve Descripción Permite registrar, modificar, actualizar y eliminar costos de productos. 6.2 Flujo básico <ol style="list-style-type: none"> 1. El Jefe de Bodega o el Usuario de Bodega selecciona visualizar los productos y sus costos. 2. El sistema muestra los datos de costos 3. El Jefe de Bodega o el Usuario puede ingresar los costos de cada producto según requiera. 4. El sistema muestra los productos ingresados y sus costos. 6.3 Flujos Alternativos Si no conoce el código de producto él (JB) ;(UB) presiona el botón “Buscar” y se carga la interfaz “Buscar Producto” Luego de realizar el criterio de búsqueda, se seleccionará el producto que se desea ingresar y regresará a la interfaz ingresar productos y costos. 6.5 Precondiciones Roles Existen productos registrados con sus respectivos costos. 6.6 Poscondiciones Se visualiza la pantalla de costos y productos. 6.7 Puntos de extensión Se extiende al caso de uso Consultar Producto. | 5 |
| GPIINV-CU-007 | Consultar Producto/Ítem Ver Casos de Uso Movimientos de Inventarios Caso de uso <u>GPIINV-CU-007</u> | 10 |

Fuente: Propia

- **Casos de Uso Movimientos de Inventarios**

A continuación se presenta la vista de casos de uso de Gpilinventarios módulo CU Movimientos Inventario:

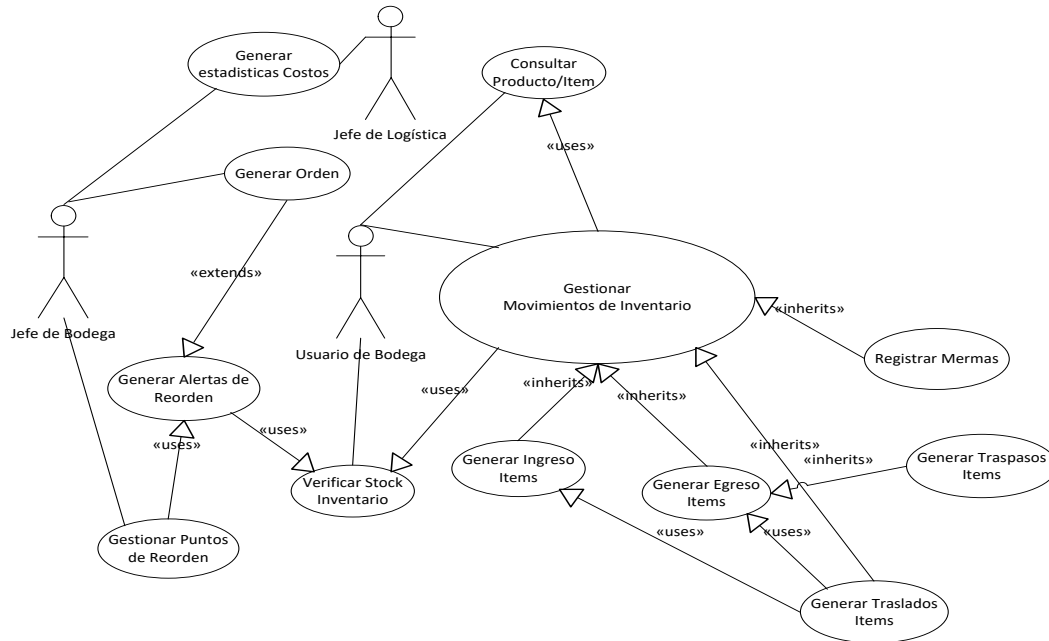


FIGURA 21: Vista de casos de uso de Gpi_inventarios módulo CU Movimientos de Inventario

Fuente: Propia

TABLA 35: CU Movimientos de Inventarios

| Referencia | Caso de Uso | Prioridad |
|----------------------|---|-----------|
| GPIINV-CU-001 | <p>Generar estadísticas Costos</p> <p>1.1 Breve Descripción Permite visualizar estadísticas de costos de productos/ítems según demanda de productos (A mayor cantidad de productos /ítems pedidos menor costo) y proveedores.</p> <p>1.2 Flujo básico</p> <ol style="list-style-type: none"> 1. El sistema realiza las consultas y las cuentas necesarias para mostrar las estadísticas. 2. El sistema muestra los datos de costos, demanda y proveedores más óptimos. 3. El Administrador puede cambiar la vista de los resultados, eligiendo visualizarlos en modo numérico, en modo gráfico de torta o en modo histograma. <p>1.3 Flujos Alternativos Ninguno</p> | 6 |

| | | |
|-----------------------------|--|----------|
| | <p>1.4 Requerimientos Especiales No existen</p> <p>1.5 Precondiciones Roles Existen productos registrados con sus respectivos costos.</p> <p>1.6 Poscondiciones Se visualiza la pantalla de estadísticas de costos.</p> <p>1.7 Puntos de extensión CU Consultar Producto.</p> | |
| <p>GPIINV-CU-002</p> | <p>Generar Orden</p> <p>2.1 Breve Descripción Permite al Jefe de Bodega (JB) generar una orden de pedido para la reposición de productos que tengan bajo stock. Esta orden de pedido será enviada al Jefe de Logística, para su trámite correspondiente.</p> <p>Flujo de Eventos</p> <p>2.2 Flujo Básico</p> <ol style="list-style-type: none"> 1. El caso de uso se inicia cuando el JB, selecciona la opción generar orden de pedido en menú de Gestionar Órdenes de Pedido. 2. Se visualiza la pantalla orden de pedido con los siguientes campos: El código de pedido (autogenerada), fecha (autogenerada), código de producto, cantidad, el empleado que lo registro (no editable). Además de las opciones agregar, generar, anular y salir. 3. El JB ingresa datos. 4. El JB selecciona la opción generar. <p>2.3 Flujos Alternativos</p> <p>2.3.1 <Código de producto desconocido> Si desconoce el código de producto el (JB); (UB) presiona el botón “Buscar” y se carga la interfaz “Buscar Producto” Luego de realizar el criterio de búsqueda, se seleccionará el producto que se desea ingresar.</p> <p>2.3.2 < Código de producto inexistente > Mensaje de alerta</p> <p>2.3.3 < Cantidad negativa > Mensaje de alerta</p> <p>2.4 Requerimientos Especiales Clasificador Central de productos</p> <p>2.5 Precondiciones Accesos</p> | <p>3</p> |

| | | |
|-----------------------------|--|----------|
| | <p>Autorizaciones Productos existentes 2.6 Poscondiciones Registrar Orden de Pedido. 2.7 Puntos de extensión CU Consultar Producto.</p> | |
| <p>GPIINV-CU-004</p> | <p>Generar alertas de reorden 4.1 Breve Descripción Permite al Sistema generar una alerta de orden de pedido para la reposición de productos que tengan bajo stock. Estas alertas serán visualizadas por el usuario de bodega, el jefe de bodega y demás involucrados, para posteriormente generar una orden de pedido que será enviada al Jefe de Logística, para su trámite correspondiente. Reorden (Referirse al caso de uso Gestionar Puntos de Reorden) Flujo de Eventos 4.2 Flujo Básico</p> <ol style="list-style-type: none"> 1. El caso de uso se inicia cuando el sistema genera alertas de reorden según como se haya ingresado la información en Gestionar Puntos de Reorden. 2. Se visualiza la pantalla de alertas de reorden (señal de bajo stock) de ítems y sus campos. Id de alerta (autogenerada), fecha inicio (autogenerada), fecha fin, código de producto, cantidad, nivel. 3. El JB ingresa código de producto. 4. El sistema muestra los datos en una tabla con los datos de todos los productos con nivel mínimo de existencias (código, nombre, tipo, descripción, cantidad). 5. Por cada producto se tiene una opción de seleccionar. 6. El JB selecciona la opción generar Pedido. 7. Se visualiza la pantalla del caso de uso de Generar Orden. <p>4.3 Flujos Alternativos 4.3.1 <Código de producto desconocido> Buscar por nombre de producto 4.3.2 < Producto inexistente > Mensaje de alerta 4.4 Requerimientos Especiales Clasificador central de productos 4.5 Precondiciones Accesos</p> | <p>3</p> |

| | | |
|-----------------------------|---|----------|
| | <p>Autorizaciones Productos registrados 4.6 Poscondiciones Orden de Pedido registrado. 4.7 Puntos de extensión CU Consultar Producto.</p> | |
| <p>GPIINV-CU-005</p> | <p>Gestionar puntos de reorden 5.1 Breve Descripción El punto de reorden consiste en la existencia de una señal al departamento encargado de colocar pedidos, indicando que la existencia de determinado material ha llegado a cierto nivel y que debe hacerse un nuevo pedido. El punto debe ser aquel que le permita seguir en funcionamiento normal de funciones mientras llega el otro pedido. Flujo de Eventos 5.2 Flujo Básico</p> <ol style="list-style-type: none"> 1. El caso de uso se inicia cuando el JB Fija el punto de reorden tomando en cuenta el consumo estimado, el tiempo para colocar la orden de compra, el tiempo de entrega del proveedor y el margen de seguridad establecido. 2. Se visualiza la pantalla de Gestionar Puntos de reorden de ítems con sus campos: Id (autogenerada), fecha inicio (autogenerada), fecha fin, código de producto, cantidad, nivel. Además de la opción agregar, editar, grabar y salir. 3. El JB ingresa código de producto. 4. Por cada producto se tiene una opción de seleccionar. <p>5.3 Flujos Alternativos 5.3.1 <Código de producto desconocido> Si se desconoce el código de producto, el JB busca el producto de acuerdo al nombre. 5.3.2 < Producto inexistente > Ingresar puntos de reorden</p> <p>5.4 Requerimientos Especiales Mensaje de alerta</p> <p>5.5 Precondiciones Accesos Autorizaciones</p> <p>5.6 Poscondiciones Puntos de reorden registrados.</p> | <p>2</p> |

| | | |
|----------------------|--|-----------|
| GPIINV-CU-006 | <p>Verificar Stock Inventario</p> <p>6.1. Breve Descripción Dado un producto o ítem, el sistema comprueba la cantidad en stock y el estado de todos los ítems.</p> <p>Flujo de Eventos</p> <p>6.2 Flujo Básico Proporciona el listado completo de:</p> <ol style="list-style-type: none"> 1. todos los ítems disponibles en la institución 2. los componentes no disponibles en ese establecimiento, pero si disponibles en otro establecimiento 3. los ítems no disponibles en ningún establecimiento. <p>6.3 Flujos Alternativos</p> <p>6.3.1 <Código de producto desconocido> Buscar el producto por nombre</p> <p>Requerimientos Especiales Ítems de inventario existentes en bdd.</p> <p>Precondiciones Accesos Autorizaciones</p> <p>Poscondiciones Se visualiza la pantalla de productos en existencia en un establecimiento.</p> <p>Puntos de extensión CU Consultar Producto/Ítem.</p> | <p>2</p> |
| GPIINV-CU-007 | <p>Consultar Producto/Ítem</p> <p>6.1. Breve Descripción Dado un producto o ítem, el sistema realiza la consulta del producto o ítem según el criterio de búsqueda ingresado por el Jefe de Bodega, Usuario de bodega o usuario de sistema.</p> <p>Flujo de Eventos</p> <p>6.2 Flujo Básico Proporciona el listado completo de:</p> <ol style="list-style-type: none"> 1. todos los ítems disponibles en la institución 2. los componentes no disponibles en ese establecimiento, pero si disponibles en otro establecimiento <p>6.3 Flujos Alternativos</p> <p>6.3.1 <No se conoce código de producto> Si no conoce el código de producto él (JB) ;(UB) presiona el botón "Buscar" Luego de realizar el criterio de búsqueda, se seleccionará</p> | <p>10</p> |

| | | |
|-----------------------------|--|----------|
| | <p>el producto que se desea verificar existencias.</p> <p>Requerimientos Especiales Debe de existir en la base de datos productos ingresados.</p> <p>Precondiciones Roles</p> <p>Poscondiciones Se visualiza la pantalla de productos</p> <p>Puntos de extensión Ninguno</p> | |
| <p>GPIINV-CU-008</p> | <p>Gestionar Movimientos de Inventario</p> <p>8.1 Breve descripción El caso de uso permite al Usuario (Usuario de Bodega o Jefe de Bodega) realizar un informe detallado del stock de almacén, las actividades y las entradas y salidas de productos.</p> <p>Flujo de Eventos</p> <p>8.2 Flujo Básico</p> <ol style="list-style-type: none"> 1. El caso de uso comienza cuando el Usuario (Usuario de Bodega o Jefe de Bodega) solicita generar reporte al menú principal. 2. Se visualiza la pantalla “Generar Reporte” 3. El sistema genera número de reporte y guarda reporte detallando la fecha y hora de creación, así como datos del usuario. 4. El sistema imprime Reporte. <p>8.3 Sub Flujo Cancelar</p> <ol style="list-style-type: none"> 1. El Usuario elige la operación Cancelar 2. El sistema cierra la interfaz “Generar Reporte” y el caso de uso finaliza. <p>8.4. Flujos Alternativos</p> <p>Pre Condiciones Roles Disponible la lista de inventariado</p> <p>Post Condiciones En el sistema queda registrado el reporte con su detalle. El sistema imprime el reporte.</p> <p>Puntos de extensión Ninguno</p> <p>Requerimientos Especiales El tiempo de respuesta del sistema debe ser menor a 3 segundos. La interfaz debe ser amigable. Formato especial para los reportes</p> | <p>1</p> |

| | | |
|-----------------------------|--|----------|
| <p>GPIINV-CU-009</p> | <p>Generar Ingreso de Ítems</p> <p>9.1. Breve Descripción</p> <p>Permite al Usuario de Bodega registrar el ingreso del Producto(s) al inventario de almacén tanto el Jefe de Bodega (JB) y el Usuario de Bodega (UB).</p> <p>Flujo de Eventos</p> <p>9.2 Flujo Básico</p> <ol style="list-style-type: none"> 1. El (JB); (UB) ingresa código del producto y presiona el botón “Buscar” 2. El sistema busca el código del producto 3. El sistema carga los campos descripción, categoría, producto y tipo según el código de producto. 4. El (JB); (UB) ingresa la cantidad y precio unitario del producto. 5. Se registra el ingreso de productos (categoría, tipo y descripción), luego por cada producto ingresado registra el movimiento de Producto (código movimiento, código de ingreso, código de producto ingresado, cantidad ingresada, precio unitario). 6. El sistema actualiza stock de producto <p>9.3 Flujos Alternativos</p> <p>9.3.1 <Código de producto desconocido></p> <p>Luego de realizar la búsqueda, se seleccionará el producto que se desea ingresar.</p> <p>9.3.2 <Error campos vacíos></p> <p>Solución sistema</p> <p>Requerimientos Especiales</p> <p>Ninguno</p> <p>Precondiciones</p> <p>Roles</p> <p>Poscondiciones</p> <p>El sistema almacena el producto ingresado</p> <p>Puntos de extensión</p> <p>Se extiende al caso de uso Consultar Producto/Ítem.</p> <p>Se extiende del caso de uso Gestionar movimientos de inventario.</p> | <p>1</p> |
| <p>GPIINV-CU-010</p> | <p>Generar Egreso de Ítems</p> <p>10.1. Breve Descripción</p> <p>Permite al Usuario de Bodega (UB); jefe de bodega registrar el pase del producto(s) mediante un tipo de movimiento al área a donde tendrá destino final (salida de Ítems).</p> <p>Flujo de Eventos</p> | <p>1</p> |

| | | |
|-----------------------------|---|----------|
| | <p>10.2. Flujo Básico</p> <ol style="list-style-type: none"> 1. Se visualiza la pantalla Registrar Salida de Productos del almacén con fecha autogenerada y los campos: Motivo de Salida (salida por Traspaso o Traslado), Código Único de Inventario del Producto, además de las opciones: agregar al listado, buscar, eliminar seleccionados del listado, grabar. 2. El (UB) selecciona Motivo de salida. 3. El Sistema muestra el código único de inventario en el campo respectivo. 4. Por cada producto se tiene una opción de seleccionar. Al final se habilita el botón de “eliminar”. 5. El sistema guarda el registro de salida con número de código de salida autogenerado, ingresa el movimiento de salida de productos. <p>10.3 Flujos Alternativos</p> <p>10.3.1 <Error no encuentra ítem></p> <p>Si el UB digita el Código de producto y no lo encuentra registrar producto (extiende al caso de uso de Generar Ingreso de Ítems)</p> <p>Requerimientos Especiales</p> <p>Ninguno</p> <p>Precondiciones</p> <p>El código único de inventario y los ítems deben existir en la base de datos.</p> <p>Poscondiciones</p> <p>La salida del producto queda registrada</p> | |
| <p>GPIINV-CU-011</p> | <p>Generar Traspaso Ítems</p> <p>10.1. Breve Descripción</p> <p>Tipo especial de Egreso que se realiza a otra institución</p> <p>Ejemplo:</p> <p>Traspaso de un producto de una bodega de una institución a otra bodega de otra institución.</p> <p>Puntos de extensión</p> <p>Se extiende al caso de uso Generar Egresos de Ítems</p> | <p>4</p> |
| <p>GPIINV-CU-012</p> | <p>Generar Traslados Ítems</p> <p>10.1. Breve Descripción</p> <p>Traslado de Ítem entre bodegas pertenecientes a la misma institución</p> <p>Puntos de extensión</p> <p>Se extiende al caso de uso Generar Egresos de Ítems</p> <p>Se extiende al caso de uso Generar Ingresos de Ítems</p> | <p>4</p> |

| | | |
|----------------------|--|---|
| GPIINV-CU-013 | Registrar Mermas 10.1. Breve Descripción Permite registrar la pérdida o reducción de un cierto número de ítems o de la actualización de un stock que provoca una fluctuación, es decir, la diferencia entre el contenido de los libros de inventario y la cantidad real de productos dentro de un establecimiento, conlleva a una pérdida monetaria. | 4 |
|----------------------|--|---|

Fuente: Metodología RUP

4.3.2.3 VISTA LÓGICA

Para el Sistema de Gestión de Inventarios se utiliza Zend Framework conjuntamente con el gestor de Base de Datos PostgreSQL.

Las principales tablas del sistema son las siguientes:

- **Diagrama Entidad Relación**

Principales Tablas del Sistema:

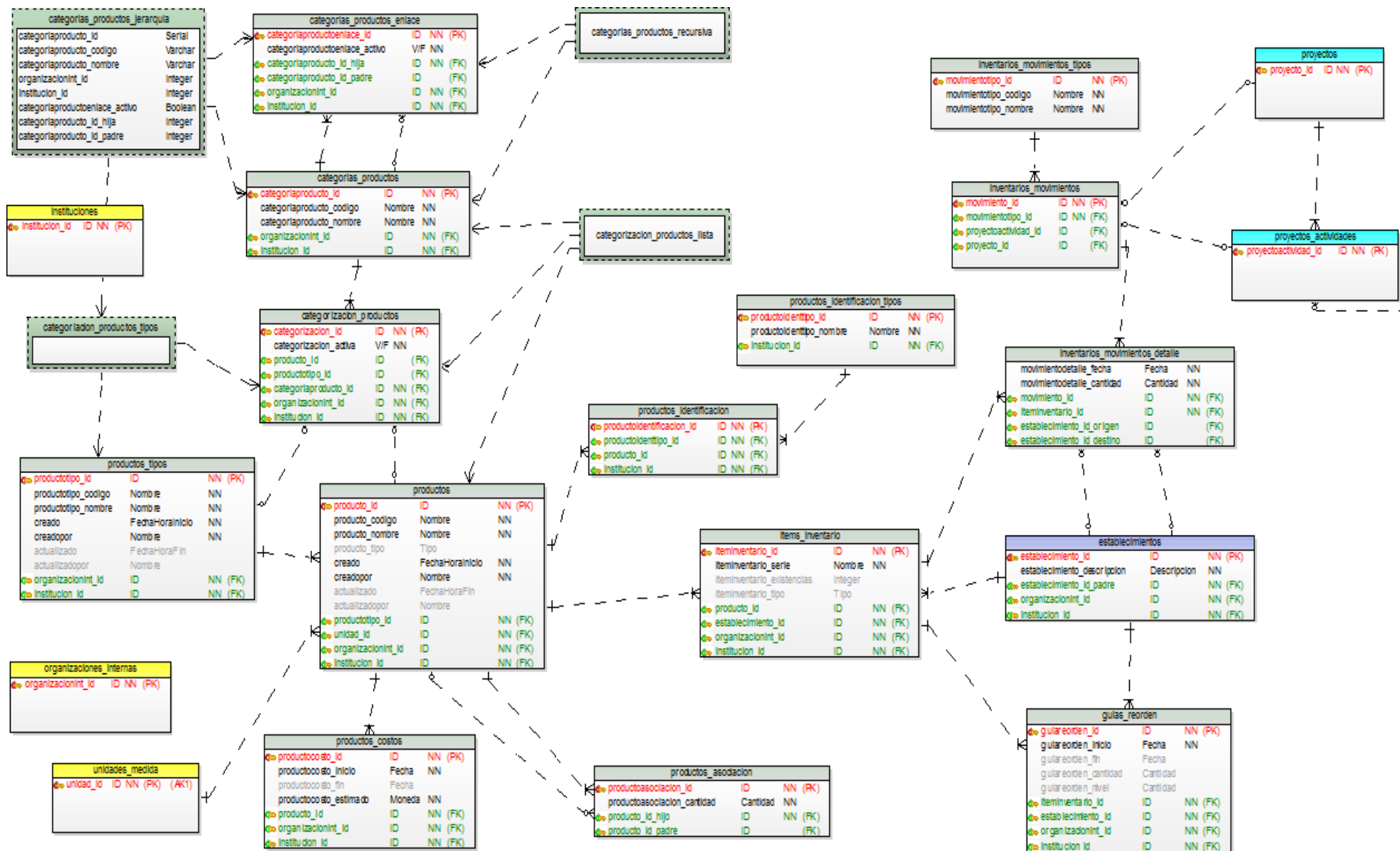


FIGURA 22: Principales Tablas del Sistema

Fuente: GPI

- **Dentidad Relacion Módulo Catálogo de Inventario**

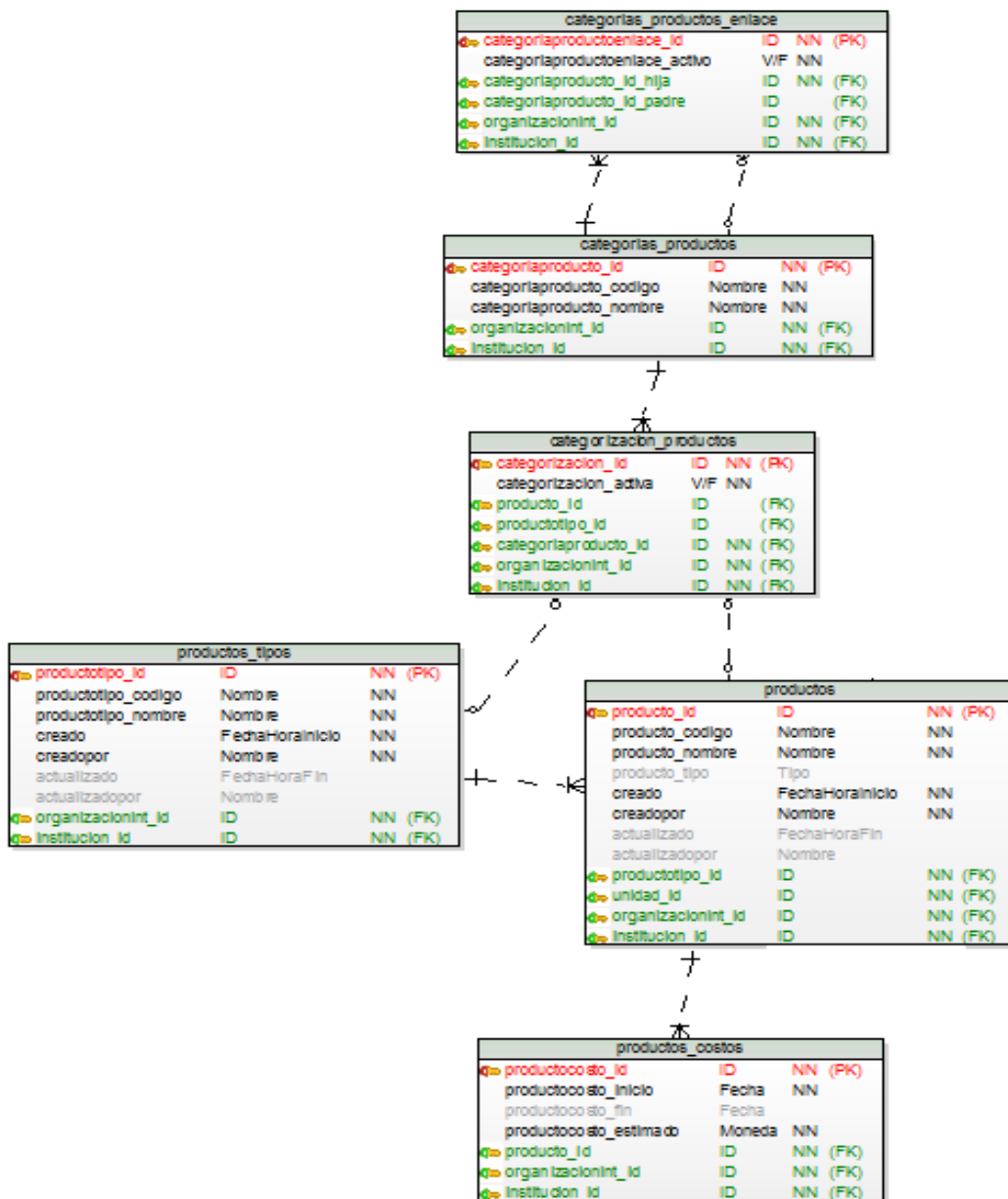


FIGURA 23: Diagrama entidad Relación Módulo Catálogo de Inventario

Fuente: GPI

Aquí se describen las tablas más importantes del catálogo de inventario:

Tabla Productos: En esta tabla se registra los productos (CPC Productos Nivel 8) de acuerdo a su tipo proveniente de la tabla productos tipos y con sus campos (producto_id, producto_código, producto_nombre, descripción, unidad_id)

Tabla Productos Tipos: En esta tabla se registra los tipos de productos (CPC Productos Nivel 8) enlazados a sus diferentes categorías y contiene campos como (productotipo_id, productotipo_codigo, productotipo_nombre, categoriaproducto_id)

Ejemplo:

Si el Producto Tipo es un artículo, el producto es Computador Portátil Dell o computador portátil HP, etc.

Tabla Categorías Productos: En esta tabla se registra los datos de categorías de productos según el Clasificador central de Productos (CPC Niveles <= 7) y contiene campos como (categoriaproducto_id, categoriaproducto_codigo, categoriaproducto_nombre).

- **Diagrama entidad Relación Módulo Movimientos de Inventario**



FIGURA 24: Diagrama entidad Relación Módulo Movimientos de Inventario

Fuente: GPI

Aquí se describen las tablas más importantes de Control de Inventarios:

Tabla Ítems de Inventario: En esta tabla se registran los ítems de inventario, contiene campos como (id, existencias, tipo).

Tabla Inventarios_movimientos: En esta tabla se registran los movimientos de inventarios, entradas y salidas.

Tabla guias_reorden: En esta tabla se registran los puntos de reorden que establecen el nivel mínimo de existencias de inventario.

4.2.3.4 VISTA DE LÓGICA.

- **Visión general de la arquitectura**

En la descripción de la arquitectura, la vista lógica describe las clases más importantes que formarán parte del ciclo de desarrollo atendiendo a los niveles de abstracción. Aquí se describen los paquetes utilizados, y las relaciones que entre ellos existen ya sea de dependencia o de uso. (Lógica Negocio)

A continuación se muestra la subdivisión de módulos del Sistema de Gestión de Inventarios para en base al negocio de cada submódulo poder identificar fácilmente sus características.

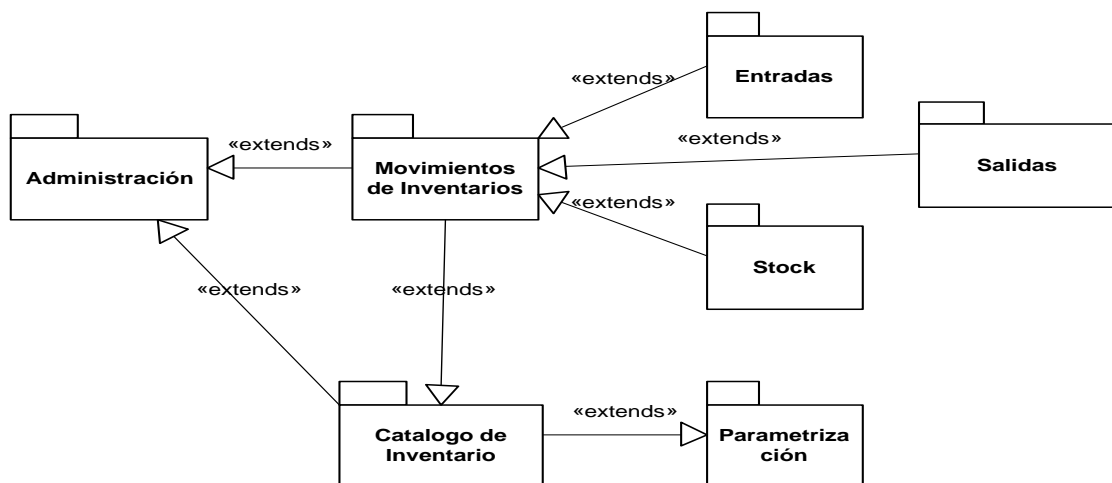


FIGURA 25: Subdivisión de módulos

Fuente: Propia

a. **Administración:** Gestión de Usuarios, Accesos, Autorizaciones, Configuraciones iniciales del sistema.

b. **Catálogo de Inventarios:** Gestión de Productos/Ítems.

- **Parametrización:** Ingreso de los códigos, características, costos y otros.

c. **Movimientos de Inventario:** Entrada y salida de productos, solicitar productos, rebajar productos por movimientos y otras.

- **Stock:** Gestión de Stock de ítems en el inventario de la institución.

- **Entradas:** Entradas de productos, traspasos, traslados.

- **Salidas:** Registro de salidas de productos, traslados, traspasos, mermas.

El sistema contiene paquetes de código divididos entre controladores, servicios, modelo, eventos y de acceso a base de datos.

- **Clases Controladores**

Clases como sección intermedia de código

- **Clases de Servicios (services)**

Interfaz de usuario y consulta a base de datos.

- **Clases de Modelo**

Entidades de modelo que contienen los datos del sistema.

- **Clases Eventos**

Contienen eventos personalizados de Flex.

- **Clases de Acceso a Base de Datos (ValueObjects)**

Proporcionan la comunicación con la bdd del Sistema.

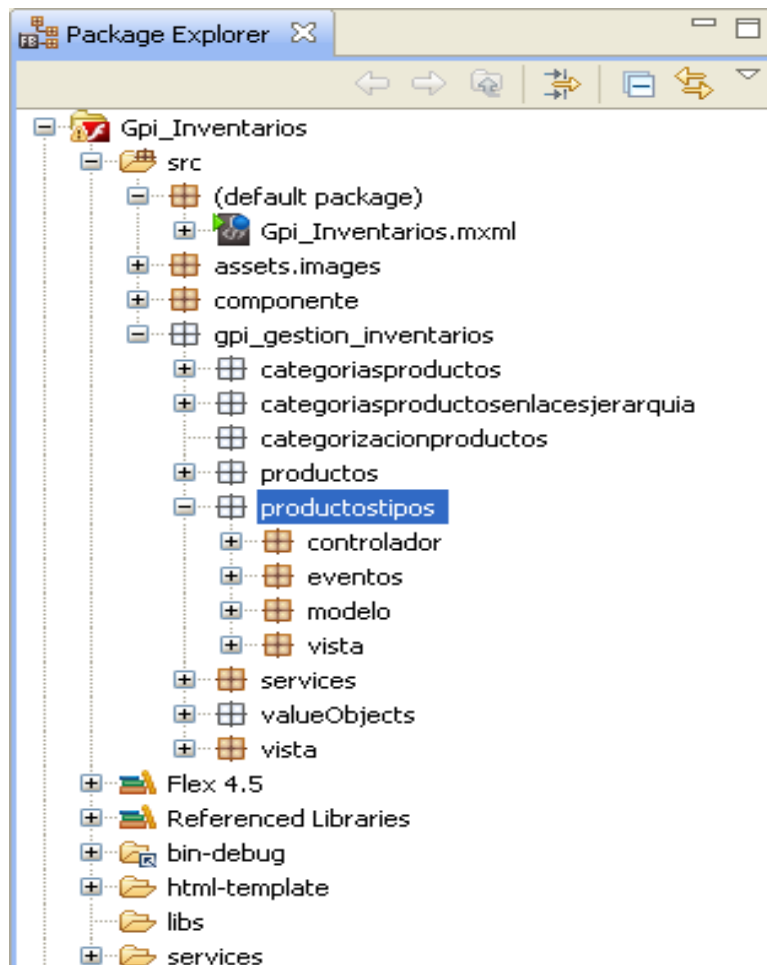


FIGURA 26: Paquetes de clases de sistema

Fuente: Propia

- **Descripción**

Se define el patrón Model View Controller más Servicios (MVC+S) y divide el software en capas y subsistemas.

- **Capa Vista (View)**

Contiene las clases que definen la interacción del usuario con el sistema.

- **Capa Modelo (Model)**

Son las clases encargadas de la persistencia de la información.

- **Capa Controlador (Controller)**

Permite operar con la lógica del negocio.

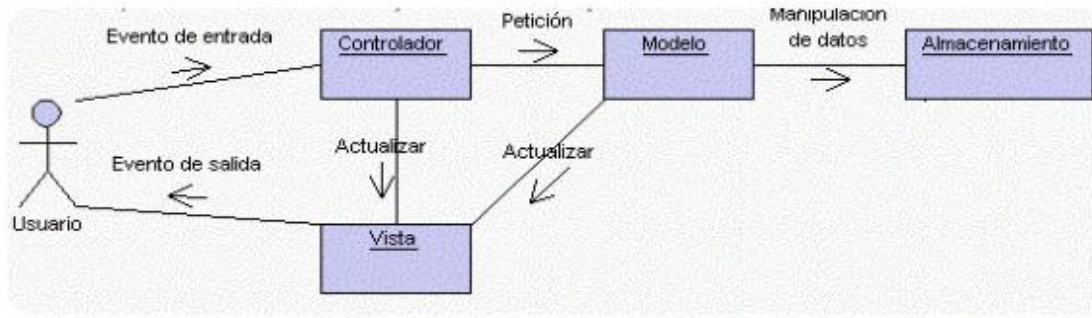


FIGURA 27: Estructura MVC

Fuente: (GPI, 2014)

La vista lógica está compuesta de:

Objetos de Acceso de Datos. Clases para manejar la persistencia de los objetos con la Base de Datos.

Entidades del Negocio. Clases del sistema.

Lógica del Negocio. Clases de servicios del negocio.

Interfaz de usuario. Clases que permiten la visualización de la aplicación.

- **Diseño en Capas**

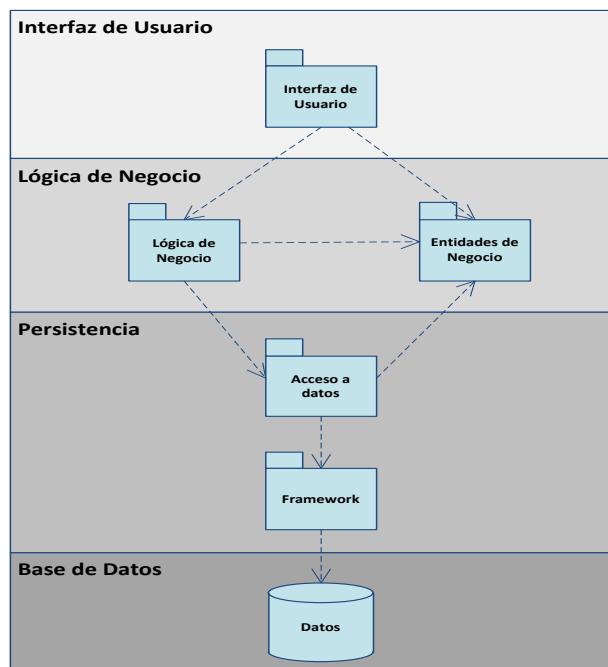


FIGURA 28: Diseño de Capas

Fuente: Lógica de Negocio. Capas

- **Capa Interfaz de Usuario**

Las interfaces del usuario están diseñadas en Flex 4, que maneja componentes mxml y el lenguaje ActionScript 3, AS3

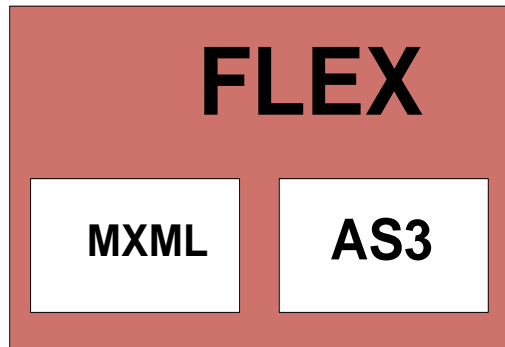


FIGURA 29: Flex (MXML, AS3)

Fuente: GPI

Los componentes serán diseñados en componentes mx y spark.

En Flex se definen dos tipos de componentes: Spark y MX. Los componentes Spark son nuevos para Flex 4 y se definen en la spark.

Spark y MX definen componentes en diferentes paquetes. MX define un control de botón en el paquete mx.controls. Cuando un componente está disponible en la Spark y MX, Adobe recomienda que utilice el componente de la Spark.

Spark y MX también definen los componentes que son únicos. MX define los componentes de visualización de datos, tales como la cuadrícula de datos y los controles AdvancedDataGrid, no incluidos en el Spark. Las aplicaciones a menudo contiene una mezcla de componentes Spark y MX. (GPI, 2014)

- **Capa Lógica del Negocio**

La lógica del negocio está dada por los servicios php y ValueObjects VO, servicios Flex DataServices DS

Los servicios php contienen los métodos públicos que dan la funcionalidad CRUD.

Existen diferentes tipos de servicios los cuales se visualiza en la siguiente Figura.

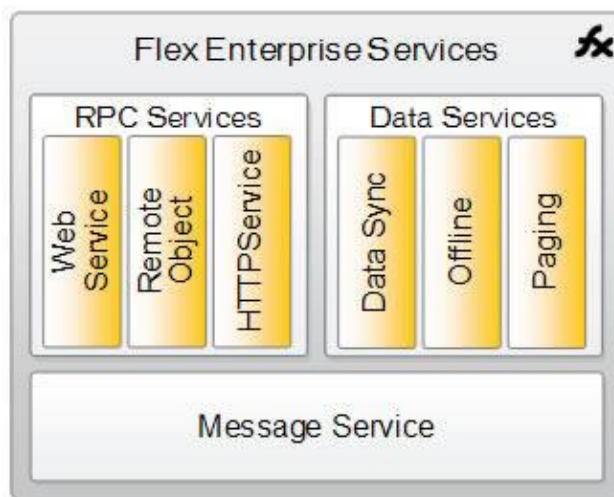


FIGURA 30: Servicios Flex

Fuente: GPI. Flex Enterprise Services

Web Service. Acceso a Servicios Web mediante SOAP usado principalmente para la integración con BPM processmaker, llamada de servicios.

RemoteObject. Acceso a métodos de objetos java en el servidor mediante AMF, protocolo binario y ligero basado en SOAP.

HTTPService.

Servicio usado para hacer conexiones HTTP (o HTTPS).

▪ **Capa Persistencia**

Esta capa contiene el paquete de Objetos de Acceso de Datos, que brinda una interfaz transparente para la interacción con el Framework el cual enviará al Driver el conjunto de sentencias para interactuar con la Base de Datos. (GPI, Documentación Institucional, 2014)

a. Base de datos: PostgreSQL

Es un sistema de gestión de base de datos relacional orientada a objetos.

Características:

- Altamente extensible.
- Soporte SQL comprensivo.
- Integridad referencial.
- API flexible.
- Cliente/Servidor (Ecured, 2014)

b. Framework: ZEND

Zend Framework es un framework open source para PHP desarrollado por Zend, empresa encargada de la mayor parte de las mejoras hechas a PHP, por lo que se podría decir que es el framework “oficial”. (Souto, 2014)

4.3.2.5 VISTA DE PROCESOS BPM

Procesos GpilInventarios BPM con el uso de la Herramienta TIBCO de modelado de Procesos de Negocios BPMN (Notación para el Modelado de Procesos de Negocio). (GPI, Documentación Institucional, 2014)

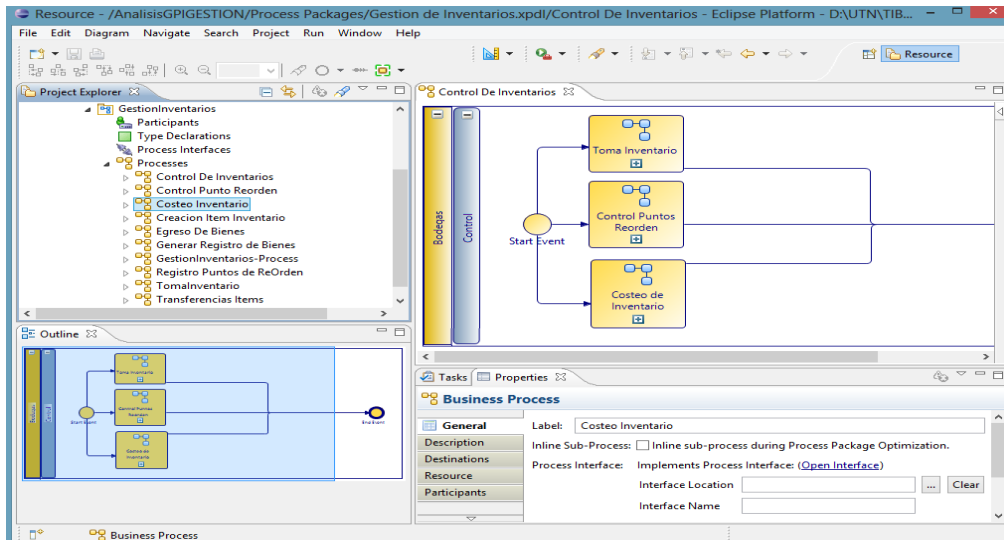


FIGURA 31: Vista Principal TIBCO

Fuente: Propia

Un modelo de procesos de negocio estándar y notación (BPMN) podrá ofrecer a las empresas la capacidad de comprensión de sus procesos de negocio internos en una notación gráfica y darán a las organizaciones la capacidad de comunicar estos procedimientos de una manera estándar. Por otra parte, la notación gráfica facilitará la comprensión de las colaboraciones de rendimiento y las transacciones comerciales entre las organizaciones. (Group O. M., 2014)

A continuación se muestra mediante diagramas el modelado de procesos de Negocio:

- **Diagramas BPM. Procesos**

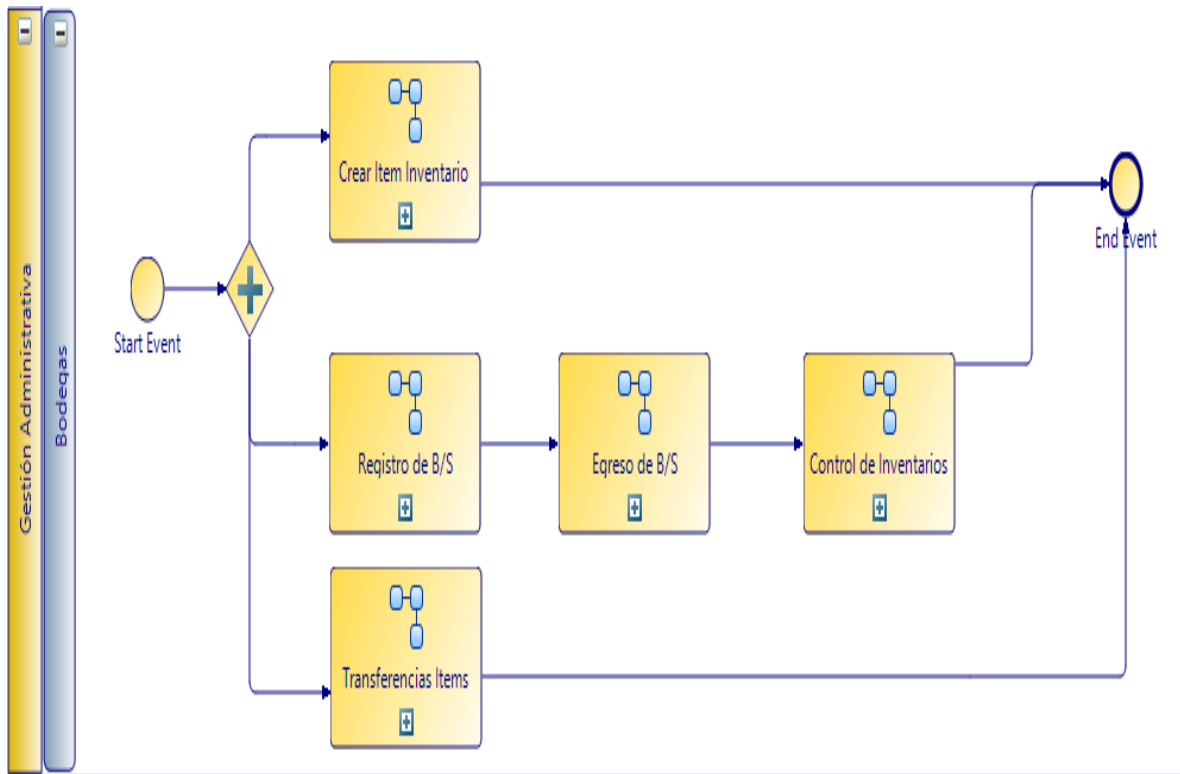


FIGURA 32: Diagrama BPM Gestión de Inventarios Procesos

Fuente: Propia

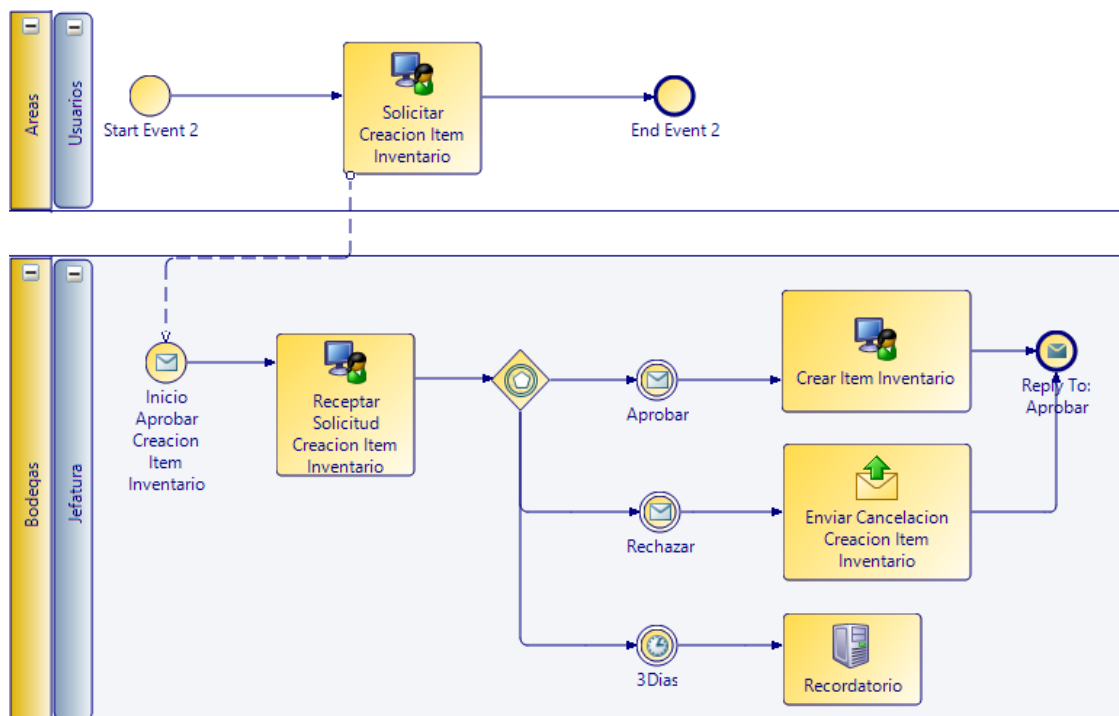


FIGURA 33: Diagrama BPM Creación Ítem de Inventario Proceso

Fuente: Propia

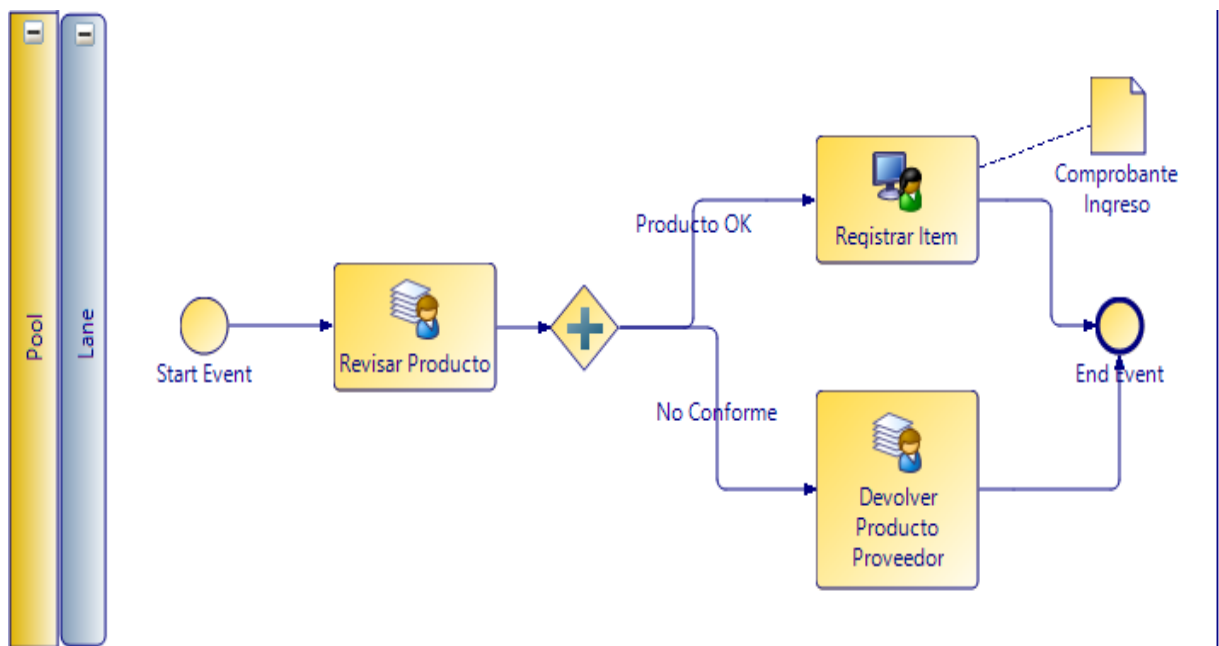


FIGURA 34: Diagrama BPM Generar Registro de Bienes Proceso

Fuente: Propia

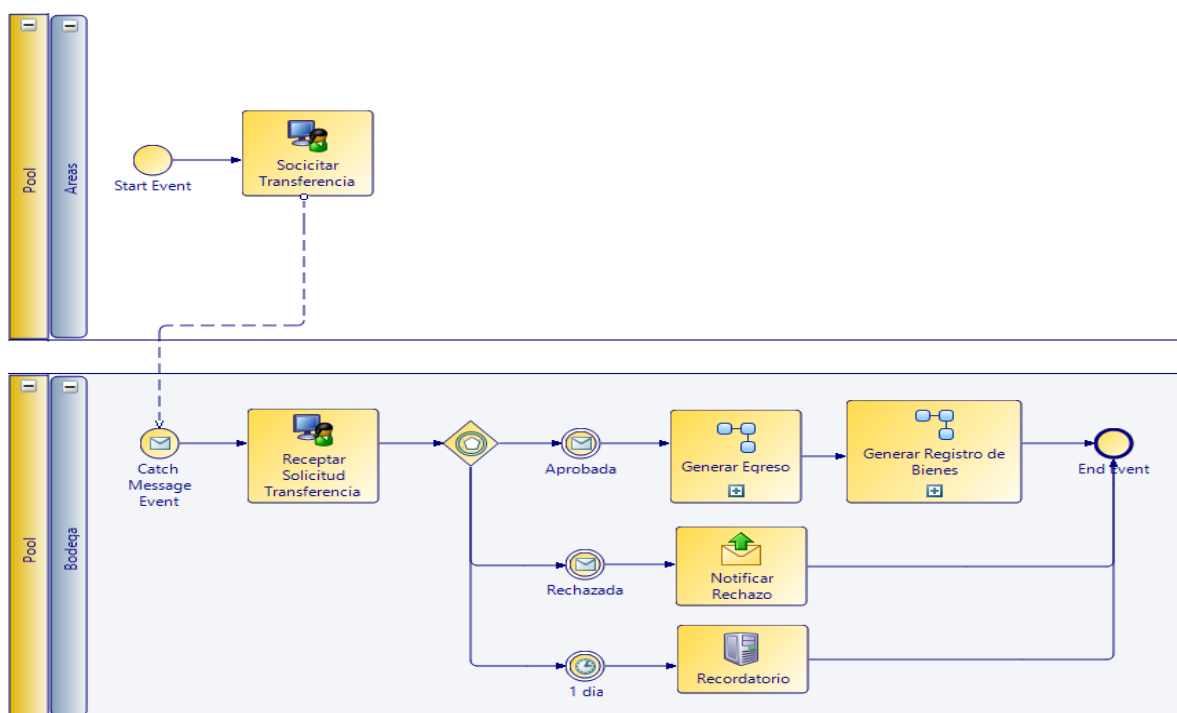


FIGURA 35: Diagrama BPM Trasferencia Ítems Proceso

Fuente: Propia

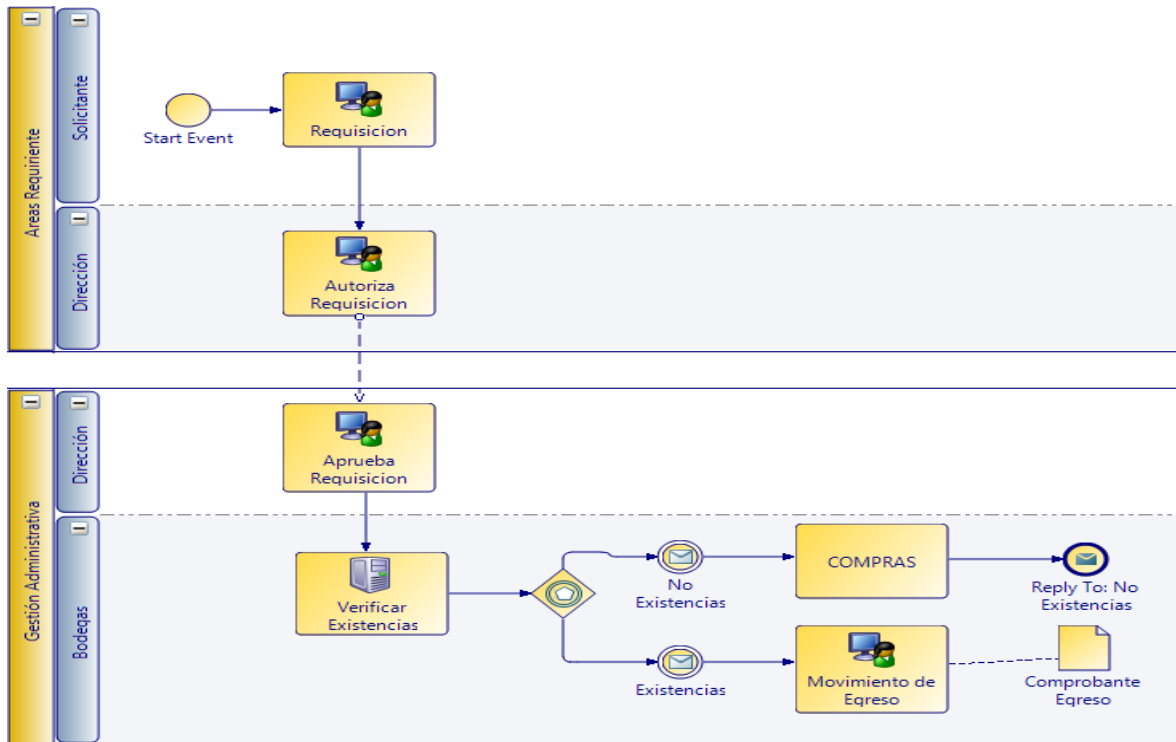


FIGURA 36: Diagrama BPM Egreso de Bienes Proceso

Fuente: Propia

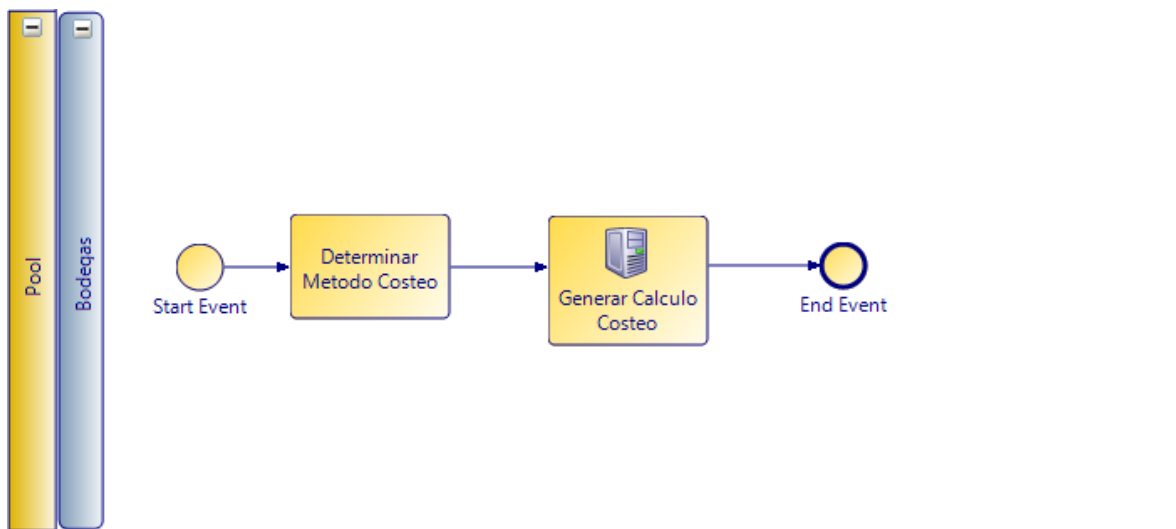


FIGURA 37: Diagrama BPM Costeo Inventario Proceso

Fuente: Propia

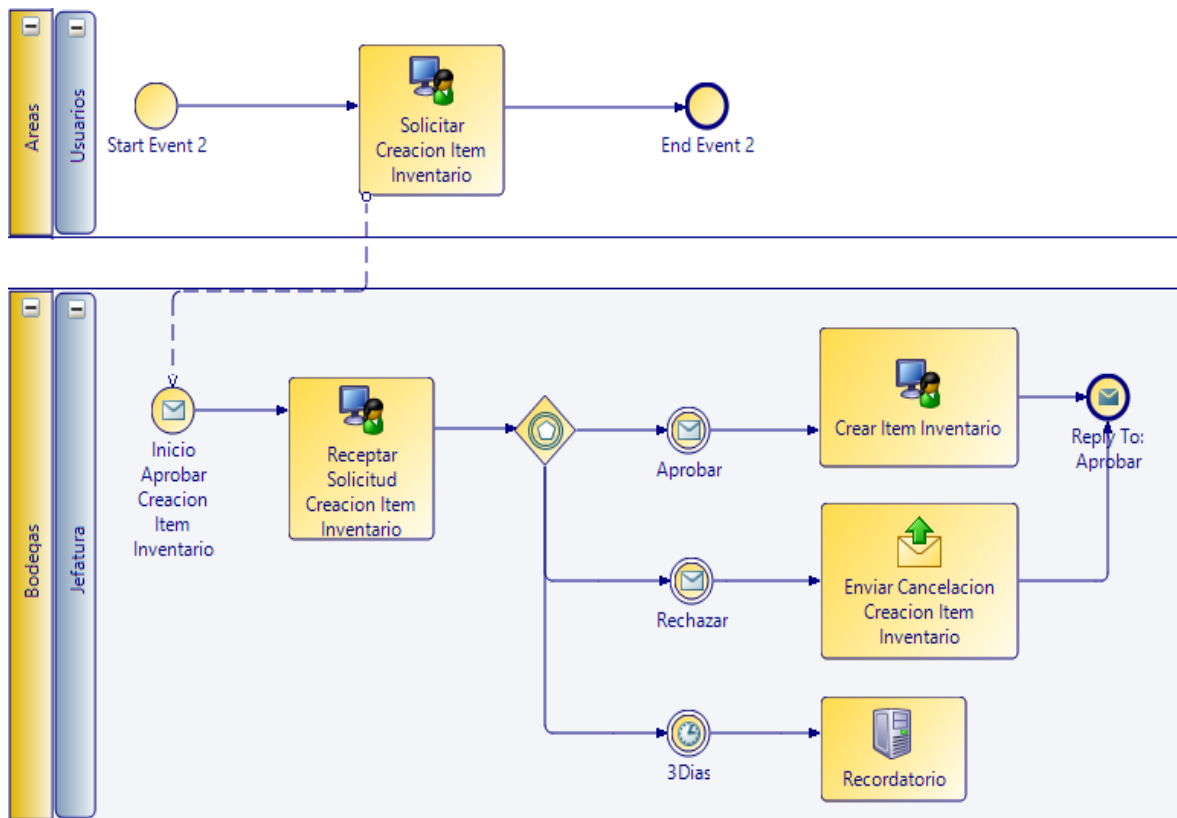


FIGURA 38: Diagrama BPM Control de Inventarios Proceso

Fuente: Propia

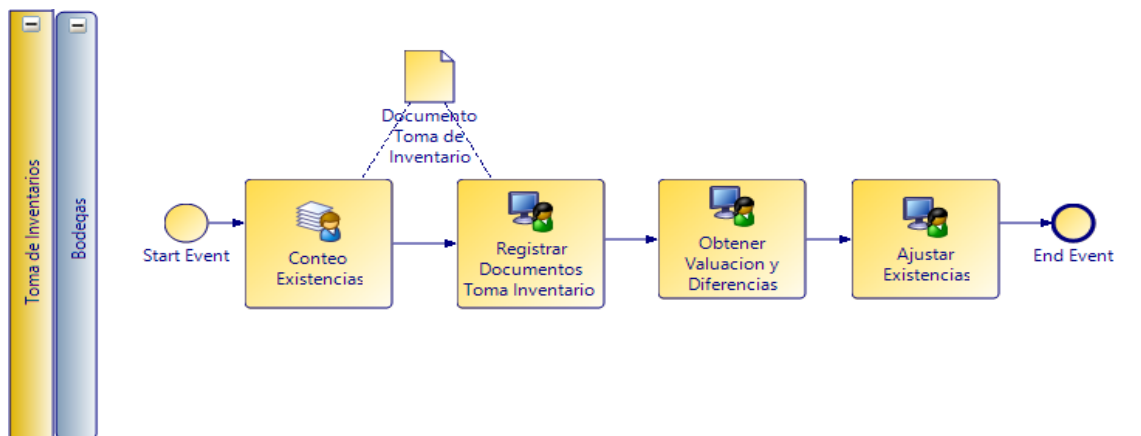


FIGURA 39: Diagrama BPM Toma de Inventario Proceso

Fuente: Propia

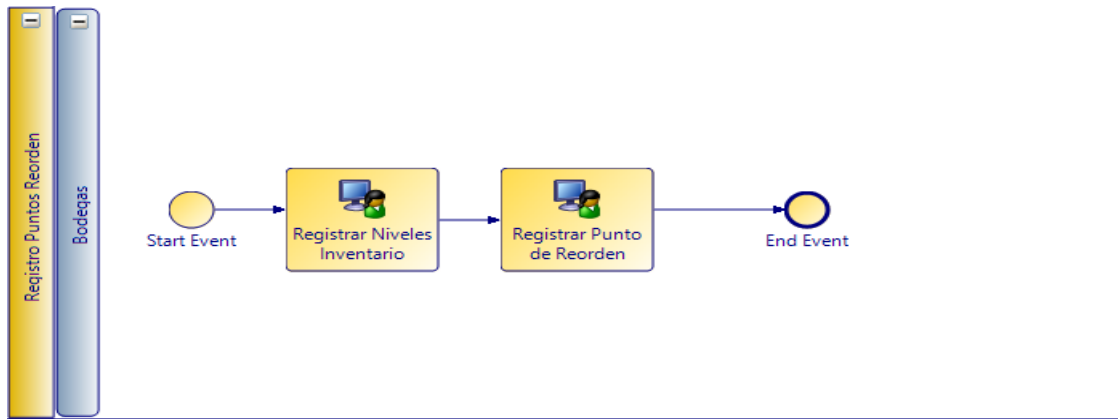


FIGURA 40: Diagrama BPM Registro Puntos de Reorden Proceso

Fuente: Propia

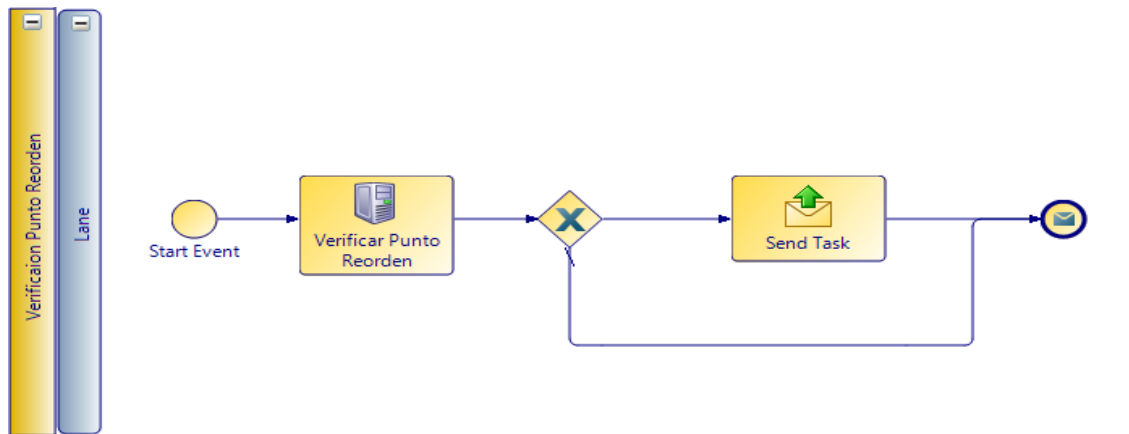


FIGURA 41: Diagrama BPM Control Puntos de Reorden Proceso

Fuente: Propia

4.3.2.6 VISTA DE DESPLIEGUE

Esta vista da una medida de la tecnología necesaria para el correcto funcionamiento del sistema a desarrollar, propone la distribución física de los elementos que lo conforman ya que representa cómo estarán distribuidos y cómo se satisfacen los requerimientos no funcionales de hardware y software. (IS, 2014)

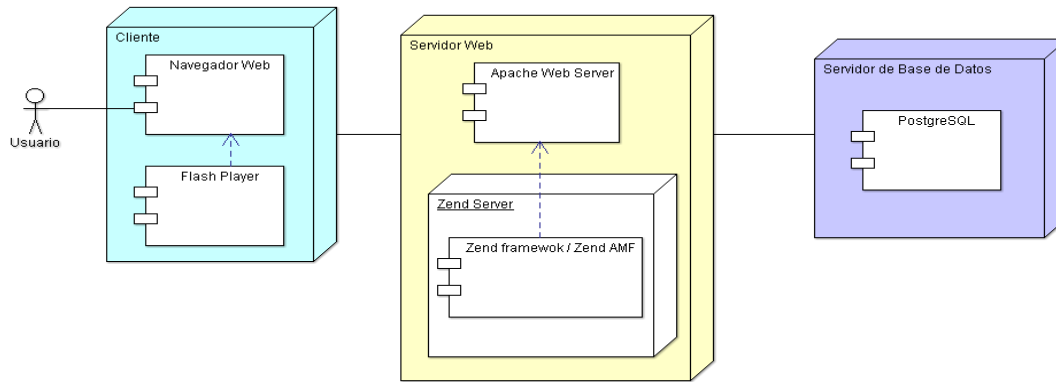


FIGURA 42: Vista de Despliegue

Fuente: GPI

- **PC Cliente:** Su función es acceder al sistema e interactuar con el mismo según sus necesidades. Además de interactuar con los dispositivos de escaneo e impresión. Al estar la aplicación desarrollada sobre la web la máquina cliente necesita disponer de muy pocas prestaciones puesto a que sólo necesita un navegador web para poder acceder al sistema y realizar las operaciones necesarias. (IS, 2014)
- **Aplicación Web:** En este nodo es donde descansa la capa de presentación del sistema, la cual es accedida por las máquinas clientes a través de un navegador web. Para poder responder a las peticiones interactúa con la UDDI para que le provea la dirección donde se encuentran los servicios necesarios para poder atenderlas. (IS, 2014)
- **Servidor Apache:** Este servidor es el encargado de atender las solicitudes, y ante una petición, asignar al servidor que le corresponde dicha petición, realizando de esta forma un balance de carga entre los servidores que contienen los servicios web permitiendo de esta manera adaptarse al creciente número de usuarios y sobrellevar las cargas de trabajo. (IS, 2014)
- **Servidor de Base de Datos:** Es el encargado de almacenar toda la información generada del sistema y replicarla al servidor de respaldo previendo la disponibilidad y la tolerancia a fallos del sistema ante cualquier falla posible. (IS, 2014)

- **BD de Respaldo:** Su función es la de almacenar la réplica de información proveniente del Servidor de Base de Datos y ante alguna falla de este prestar servicios por él hasta que se restablezca. (IS, 2014)

4.3.2.7 TAMAÑO Y PERFORMANCE

- El tiempo de latencia de acceso a la base de datos no deberá exceder el minuto para el registro de maestros o ejecución de alguna transacción. (IS, 2014)
- Los componentes han sido diseñados para asegurar que los requerimientos de memoria y disco sean los mínimos para los servidores. (IS, 2014)
- Las transacciones y/o consultas de los clientes evitarán tiempos muertos y colas de información donde se tenga que esperar un tiempo excesivo para poder realizar una transacción. (IS, 2014)
- El servidor Web y de Aplicaciones requerirá al menos 5GB de disco duro y 2GB de RAM. (IS, 2014)

4.3.2.8 CALIDAD

- El cliente puede ejecutarse en cualquier sistema operativo, debido a que es un sistema Web. Esto se puede comprobar accediendo a la aplicación a través de Internet Explorer en una computadora con sistema operativo Windows y también a través de Mozilla Firefox en una computadora con sistema operativo Linux. (IS, 2014)
- La interfaz de usuario del sistema será de fácil entendimiento por parte de los usuarios. Esto se comprueba visualizando la ubicación de los componentes, de los sencillos formularios y de la navegabilidad intuitiva. (IS, 2014)
- El sistema permitirá que las transacciones que se realicen sean seguras. Para este fin se definirán perfiles de acceso. (IS, 2014)

4.3.2.9 MANEJO DE ERRORES

En el desarrollo de un sistema es importante crear una buena política de manejo de errores de tal manera que se cubran la mayoría de los posibles casos de mal funcionamiento durante el uso de todas las funciones que ofrece el sistema. Además, la información que puede ofrecer una buena política de manejo de excepciones es muy valiosa a la hora de depurar el sistema y por lo tanto, el mantenimiento puede simplificarse a gran escala. (IS, 2014)

- **Buenas Prácticas**

Para el manejo de errores se seguirán las siguientes prácticas:

- **Encapsulamiento:** en el caso de que una excepción viaje a través de las capas del sistema y se cree alguna otra a partir de la primera generada, se guarda toda la información del primer error a través del encapsulamiento en el momento de la creación de la nueva excepción. De esta forma, no se pierde información debido a la propagación de la excepción o de la sustitución a través de las capas del sistema. (IS, 2014)
- **Lanzar temprano:** las excepciones son lanzadas en el preciso instante en el que ocurre el error. Así se tiene un mejor control a la hora de depurar y de buscar el origen de la falla. (IS, 2014)
- **Atrapar tarde:** las excepciones se atraparán cuando se tenga la mayor cantidad de información posible y cuando se esté en la capa correspondiente del sistema que tenga las herramientas necesarias para manejar correctamente el error. (IS, 2014)
- **Obtener y desplegar la mayor cantidad de información:** a la hora de generar un error, la mayor cantidad de información debe ser desplegada de tal manera que se pueda ubicar fácilmente el origen y por lo tanto, se depure rápidamente el sistema. (IS, 2014)

- **Política del Manejo de Excepciones**

La política de manejo de errores en el sistema GpilInventarios se definirá por capas. Para cada capa aplica una política distinta que permitirá que las excepciones que permitan manejar la mayor cantidad de errores del sistema en general. (IS, 2014)

- **Capa de Datos y Utilidades**

En esta capa se utilizará una política de Propagación y de encapsulado de excepción. Esto tendrá como consecuencia que la información del error ocurrido no se pierda y se transmita lo más posible a través de su viaje por las capas del sistema. (IS, 2014)

- **Capa Lógica**

Para la capa lógica se aplicará una política de sustitución y encapsulamiento. Se sustituye la excepción proveniente de la capa inferior de tal manera que se cree alguna otra más específica a la causa del error y que, por consiguiente, brinde información más específica a la capa superior. (IS, 2014)

- **Capa de Presentación e Interfaz**

En la capa de interfaz se aplicará la política de manejo de errores. En esta capa del sistema se realizará el manejo de todas las excepciones provenientes de las capas inferiores y se desplegará al usuario la información necesaria al error ocurrido. (IS, 2014)

4.3.2.10 PLAN DE PRUEBAS

- **Introducción**

Plan de pruebas del sistema de Gestión de inventarios antes de la implementación del sistema.

- **Propósito**

Consiste en la manejo de pruebas del sistema GPI_INVENTARIOS con el fin de encontrar posibles errores o fallas.

- **Alcance**

Se aplicará el plan de pruebas a cada uno de los submódulos del sistema de gestión de inventarios.

- **Estrategia de Pruebas**

Se definen las consideraciones y estrategias que se aplicarán en el plan de pruebas para poder identificar cuando las pruebas se consideren completas.

- **Tipos de Pruebas**

- **Pruebas de Integridad de Datos**

TABLA 36: Pruebas de Integridad de Datos

| | |
|-----------------------------------|--|
| Objetivo: | Asegurar la integridad de datos |
| Técnica: | <ul style="list-style-type: none"> ✓ Registrar datos con tipos válidos. ✓ Registrar datos en entidad que tengan relación con otras. ✓ Revisar el esquema de base de datos para asegurarse que los datos se han guardado satisfactoriamente y de acuerdo a los estándares definidos. |
| Criterio de completitud | Todos los métodos de acceso y procesos de la Base de datos funcionan como fueron diseñados. |
| Consideraciones especiales | <ul style="list-style-type: none"> ✓ Se debe utilizar un conjunto pequeño de datos para incrementar la visibilidad de cualquier evento anormal o inesperado. ✓ Los datos de pruebas deberían ser reales y de uso común. |

Fuente: RUP (RUP, An Introduction., 2014)

▪ **Pruebas del Sistema**

TABLA 37: Pruebas del Sistema

| | |
|-----------------------------------|--|
| Objetivo: | Asegurar la apropiada navegación dentro del sistema, ingreso de datos, procesamiento y recuperación. |
| Técnica: | <p>Ejecute cada caso de uso, flujo básico o función utilizando datos válidos e inválidos, para verificar que:</p> <ul style="list-style-type: none"> ✓ Los resultados esperados ocurren cuando se utiliza un dato válido. ✓ Los mensajes de error o de advertencia aparecen en el momento adecuado, cuando se utiliza un dato inválido. ✓ Cada regla de negocios es aplicada adecuadamente. |
| Criterio de completitud | <ul style="list-style-type: none"> ✓ Todas las pruebas planeadas han sido ejecutadas. ✓ Todos los defectos que se identificaron han sido tenidos en cuenta. |
| Consideraciones especiales | Considerar aspectos que impactan la implementación y ejecución de las pruebas del Sistema |

Fuente: Metodología RUP (RUP, An Introduction., 2014)

▪ **Pruebas del Ciclo del negocio**

TABLA 38: Pruebas del Ciclo de Negocio

| | |
|-----------------------------------|--|
| Objetivo: | Asegurar que el sistema funciona de acuerdo con el modelo de negocios emulando todos los eventos en el tiempo y en función del tiempo. |
| Descripción de la prueba: | Las pruebas del ciclo de negocio deberían emular las actividades ejecutadas en él a través del tiempo. Debería identificarse un periodo, como por ejemplo un año, y las transacciones y actividades que podrían ocurrir durante un periodo de un año deberían ejecutarse. Incluyendo todos los ciclos y eventos diarios, semanales y mensuales. |
| Técnicas: | <p>Ejecute cada caso de uso, flujo básico o función utilizando datos válidos e inválidos, para verificar que:</p> <ul style="list-style-type: none"> ✓ Incremente el número de veces en que una función es ejecutada para simular diferentes usuarios sobre un periodo especificado ✓ Todas las fechas o funciones que involucren tiempos serán probadas con datos válidos e inválidos de fechas o periodos de tiempo. |
| Criterio de completitud | <ul style="list-style-type: none"> ✓ Todas las pruebas planeadas han sido ejecutadas. ✓ Todos los defectos que se identificaron han sido tenidos en cuenta. |
| Consideraciones especiales | <ul style="list-style-type: none"> ✓ Las fechas y eventos del sistema pueden requerir actividades especiales de soporte. ✓ Se requiere un modelo de negocios para identificar requisitos y procedimientos de prueba apropiados. |

Fuente: Metodología RUP (RUP, An Introduction., 2014)

▪ **Pruebas de Interfaz de Usuario**

TABLA 39: Pruebas de Interfaz de Usuario

| | |
|-----------------------------------|--|
| Objetivo: | <p>Verificar lo siguiente:</p> <ul style="list-style-type: none"> ✓ La navegación a través de los objetos de la prueba refleja la funcionalidad del proyecto; Se realiza una navegación de todos los menús y los formularios de cada submenú. ✓ Los elementos de los formularios tales como botones, íconos, alertas, etc. Deben mantener un solo formato. |
| Descripción de la prueba: | <p>Las pruebas de interfaz de usuario verifican la adecuada interacción del usuario con el software. El objetivo es verificar que cada interfaz corresponda con la acción que realiza y que la interfaz tenga una adecuada navegación.</p> |
| Técnicas: | <ul style="list-style-type: none"> ✓ Con la ayuda de los usuarios que usarán el sistema, se les pide que usen el sistema realizando las actividades y procesos cotidianos en el sistema. ✓ Los usuarios del sistema son reales y trabajan en su área de trabajo normal. ✓ Los desarrolladores no están presentes. ✓ Los usuarios son advertidos que el sistema puede fallar. |
| Criterio de completitud | <p>Se establece un periodo de pruebas, en el que los errores detectados no sean clasificados como críticos para el sistema.</p> |
| Consideraciones especiales | <p>Se debe establecer el mecanismo de comunicación entre los usuarios y los desarrolladores para que los errores que se detecten puedan ser solucionados.</p> |

Fuente: Metodología RUP (RUP, An Introduction., 2014)

▪ **Pruebas de Desempeño**

TABLA 40: Pruebas de Desempeño

| | |
|-----------------------------------|--|
| Objetivo: | <p>Validar y verificar los requisitos de desempeño especificados para el sistema.</p> <p>Validar el tiempo de respuesta para las transacciones o procesos bajo las siguientes condiciones:</p> <ul style="list-style-type: none"> ✓ Volumen normal anticipado. ✓ Volumen máximo anticipado. |
| Descripción de la prueba: | Las pruebas de desempeño miden los tiempos de respuesta que tiene el sistema y otros aspectos sensibles al tiempo. |
| Técnicas: | <ul style="list-style-type: none"> ✓ Con la ayuda de los usuarios que usarán el sistema, se les pide que usen el sistema realizando las actividades y procesos cotidianos en el sistema. ✓ Los usuarios del sistema son reales y trabajan en su área de trabajo normal. ✓ Se pide a los usuarios que todos accedan a una determinada acción o proceso al mismo tiempo. ✓ Los desarrolladores no están presentes. ✓ Los usuarios son advertidos que el sistema puede fallar. |
| Criterio de completitud | Se establece un periodo de pruebas, en el que los errores detectados no sean clasificados como críticos. |
| Consideraciones especiales | Se debe establecer el mecanismo de comunicación entre los usuarios y los desarrolladores para que los errores que se detecten puedan ser solucionados. |

Fuente: Metodología RUP (RUP, An Introduction., 2014)

▪ **Pruebas de Seguridad y Control de Acceso**

TABLA 41: Pruebas de Seguridad y Control de Acceso

| | |
|-----------------------------------|--|
| Objetivo: | <p>Seguridad en el Funcionamiento y Datos, verificar que los usuarios puedan acceder solo aquellas funciones y datos para los cuales se le ha otorgado permisos al momento de crear su perfil de usuario.</p> <p>Seguridad en Administración del Sistema, comprobar que solo aquellos usuarios con permisos privilegiados puedan acceder a las funciones parametrizables del sistema y opciones del sistema.</p> |
| Descripción de la prueba: | <p>Las pruebas se realizará enfocándonos en dos aspectos principales:</p> <ul style="list-style-type: none"> ✓ Seguridad en la aplicación controlando el acceso a determinada información y funciones del negocio, y ✓ Seguridad del sistema realizando registro de accesos de usuarios al sistema. |
| Técnicas: | <ul style="list-style-type: none"> ✓ Seguridad de Datos y Funciones, identificar los tipos de usuarios y asignar funciones a las que tiene acceso el tipo de usuarios para asignarle al usuario. ✓ Realizar pruebas para cada tipo de usuario y verificar los permisos creado transacciones para cada tipo de usuario. |
| Criterio de completitud | <p>Para cada tipo de dato se puede asignar las funciones y datos apropiados para su desempeño.</p> |
| Consideraciones especiales | <p>El acceso al sistema debe ser revisado con el administrador de la red y de la base de datos.</p> <p>Con esto podremos visualizar cualquier anomalía.</p> |

Fuente: Metodología RUP (RUP, An Introduction., 2014)

- **Recursos**

TABLA 42: Recursos

| Rol | Recurso Requerido | Responsabilidad Específica |
|--------------------------|--|--|
| Administrador de Pruebas | Encargado del Proyecto Cristian Ipiales | Proveer las directrices de las pruebas. Adquirir los recursos Necesarios. |
| Diseñador de Pruebas | Encargado del Proyecto Cristian Ipiales | Identificar y priorizar las pruebas. Generar Plan de Pruebas |
| Gestión de Proyecto | Cristian Ipiales | Responsables de ejecutar las pruebas, registro de resultados. |

Fuente: Metodología RUP (RUP, An Introduction., 2014)

- **Entregables**

TABLA 43: Entregables

| Entregable | Propietario | Revisión/Distribución |
|-------------------|--------------------|------------------------------|
| Plan de Pruebas | Cristian Ipiales | Coordinadores del proyecto |

Fuente: Metodología RUP (RUP, An Introduction., 2014)

- **Lista de Riesgos**

TABLA 44: Lista de Riesgo

| Ranking | Descripción del Riesgo e Impacto | Estrategia de reducción del riesgo |
|----------------|--|--|
| 7 | La liberación del sistema GPI_INVENTARIOS podría no estar lista para el mes de enero del 2013, mes en que se inicia un nuevo periodo de Administración. | Aumentar el esfuerzo en el desarrollo del sistema. |
| 5 | Que las características del servidor donde será alojado el sistema, no cumpla con los requerimientos necesarios para un correcto funcionamiento del sistema. | Antes de su uso especificar los debidos requerimientos de hardware y software. |
| 3 | Abandono del proyecto por parte de los desarrolladores. | Aumentar el compromiso de terminar todo el sistema. |
| 3 | Incompatibilidad del navegador de internet de los usuarios. | Instalar el navegador compatible con el sistema con Flash Player. |
| 2 | Presencia de virus en las computadores de los usuarios finales. | Instalar un antivirus y actualizar constantemente la base de datos del anti virus. |

Fuente: Metodología RUP (RUP, An Introduction., 2014)

4.3.3 FASE DE TRANSICIÓN

El Objetivo de esta etapa de RUP es lograr que el usuario quede satisfecho con el trabajo realizado y lograr un producto final tan rápidamente y costo efectivo como sea posible. (RUP, An Introduction., 2014)

En esta fase se realiza la entrega del Sistema, se realiza capacitaciones y se entrega el manual de instalación. (RUP, An Introduction., 2014)

4.3.3.1 MANUAL DE INSTALACIÓN

- **Introducción**

A continuación se detalla como instalar cada una de las herramientas que son necesarias para el presente proyecto.

- **Instalación rápida**

En el CD entregable se encuentra una carpeta con lo necesario, por este método de implantación del software del sistema es mucho más fácil ya que los paquetes incluidos en el .rar “appserv.rar” como PHP, Zend Framework y Apache ya están configurados previamente para su uso rápido y sin mayor complicación, solo se debe seguir los siguientes pasos:

- 1. Crear Carpeta “appserv” en el disco local D:**

- a. Instalar Apache Http en el directorio D:\appserv\Apache 2.2

- b. Parar el servicio de Apache

- c. Extraer en el escritorio appserv.rar y copiar las carpetas contenidas

- ✓ PHP

- ✓ Zend Framework

- ✓ Apache 2.2

- ✓ www

- d. Copiar dentro de la carpeta “appserv” y Reemplazar las carpetas existentes

2. Crear Carpeta “GPI” en el disco local D:

- a. Extraer en el escritorio GPI.rar
- b. Copiar las carpetas contenidas en la carpeta GPI en D:\GPI

3. Instalar la Base de Datos en Postgresql

Más adelante se explica el método de Instalación de Postgresql y la base de datos. En el Apartado 4.5.1.3.3 Instalación de Postgres.

4. Establecer Variables de entorno en el Sistema:

- a. Seleccionar Equipo en el menú Inicio
- b. Seleccionar Propiedades del sistema en el menú contextual
- c. Hacer clic en Configuración avanzada del sistema > ficha Opciones avanzadas
- d. Hacer clic en Variables de entorno, en Variables del sistema, busque **PATH** y haga clic en él.
- e. Modificar **PATH** agregando la ubicación de la clase al valor de **PATH**. Si no dispone del elemento **PATH**, puede optar por agregar una nueva variable y agregar **PATH** como el nombre y la ubicación de la clase como valor.
- f. Ingresar lo siguiente como valor:

D:\appserv\php5\;D:\appserv\ZendFramework-1.11.6\bin;

5. Ingresar la dirección proporcionada por el Administrador del Sistema.

Para efectos de pruebas la dirección es la siguiente:

<http://localhost/GpiGestionBackend/public/bin-debug/GpiGestionFrontend.php>

- **Instalación de todos los paquetes de software.**

En el CD se encuentra todo lo necesario para la instalación del software, copiar todos esos archivos y pegarlos en D: //appserv.

- **Instalación de Zend Framework**

Configurar Apache para admitir archivos .htaccess. Esto se hace generalmente cambiando la configuración de:

! AllowOverride None a

! AllowOverride All

1. Buscar el archivo "[httpd.conf](#)" en la carpeta "conf" dentro de la carpeta de instalación de Apache.
2. Encontrar la siguiente línea "[#LoadModule rewrite_module modules/mod_rewrite.so](#)" en el archivo "[httpd.conf](#)".
3. Quitar el "#" que representa un comentario.

Extraer el archivo

D:\appserv\ZendFramework

Agregar al PATH

D:\appserv\ZendFramework\bin

- **Crear el proyecto (Para probar la instalación de Zend)**

```
$:>zf create project zf-tutorial
```

Editar application/configs/application.ini y agregar:

```
phpSettings.date.timezone= "America/Guayaquil"
```

despues en phpSettings insertar valores en la sección [production]

- **Crear acciones (Para probar la instalación de Zend)**

```
zf create action add Index
```

```
zf create action edit Index
```

```
zf create action delete Index
```

Ingresar lo siguiente al navegador

<http://localhost/zf-tutorial/public/index/add>.

- **Configuración de base de datos**

Para utilizar Zend_Db_Table, nombrar base de datos y utilizar junto con un nombre de usuario y una contraseña. El Zend_Application componente se envía con un recurso de configuración de base de datos, así todo lo que se necesita es configurar la información adecuada en el configs/Application.ini archivo y se hará el resto. (GPI, Documentación Institucional, 2014)

Ingresa a Application/configs/Application.ini y agrega lo siguiente al final de la sección [production] (es decir, por encima de la sección [staging]):

```
resources.db.adapter = "PDO_PGSQL"  
resources.db.params.username = "postgres"  
resources.db.params.password = "pgadmin"  
resources.db.params.dbname = "postgres"  
resources.db.isDefaultTableAdapter = "true"
```

- **Para configurar acceso de base de datos**

zf configure dbadapter

```
"adapter=PDO_PGSQL&username=postgres&password=pgadmin&dbname=postgres"
```

- **Instalar y configurar PHP 5 y Apache 2.2 en Windows**

- **Instalación Apache 2.2**

Pasos.

1. Instalar Apache 2.2

- Hacer doble clic en el archivo de instalación (apache_2.2-win32-x86-no_ssl).



- Hacer clic en siguiente.
- Seleccionar “Acepto los términos del contrato de licencia”, hacer clic en siguiente y otra vez hacer clic en siguiente.

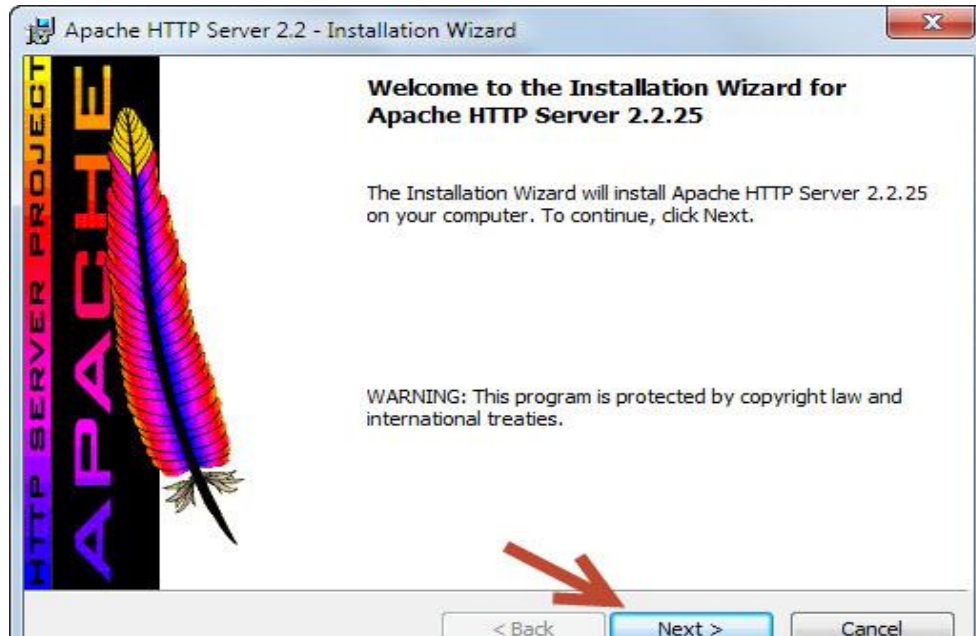
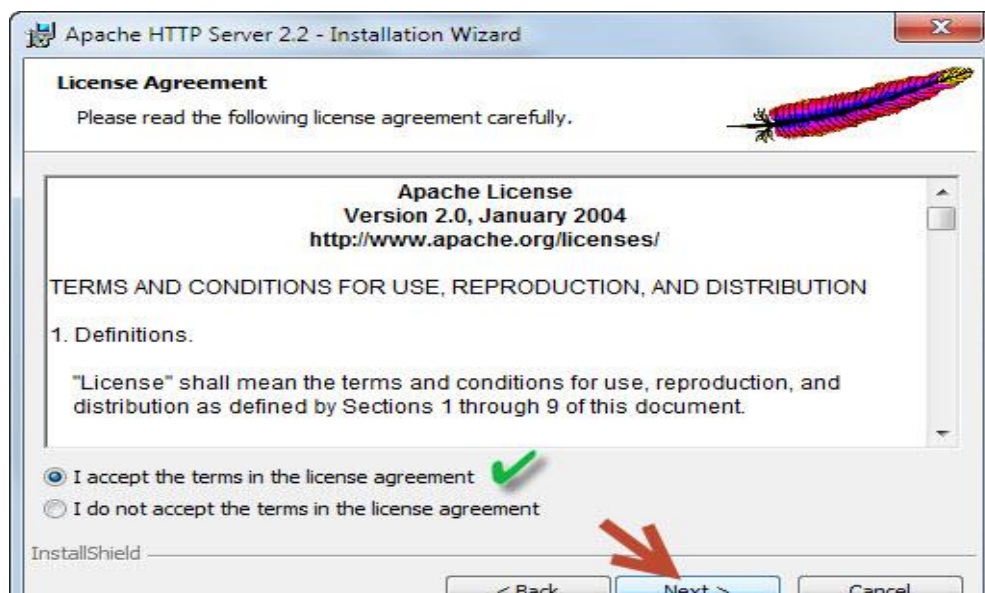
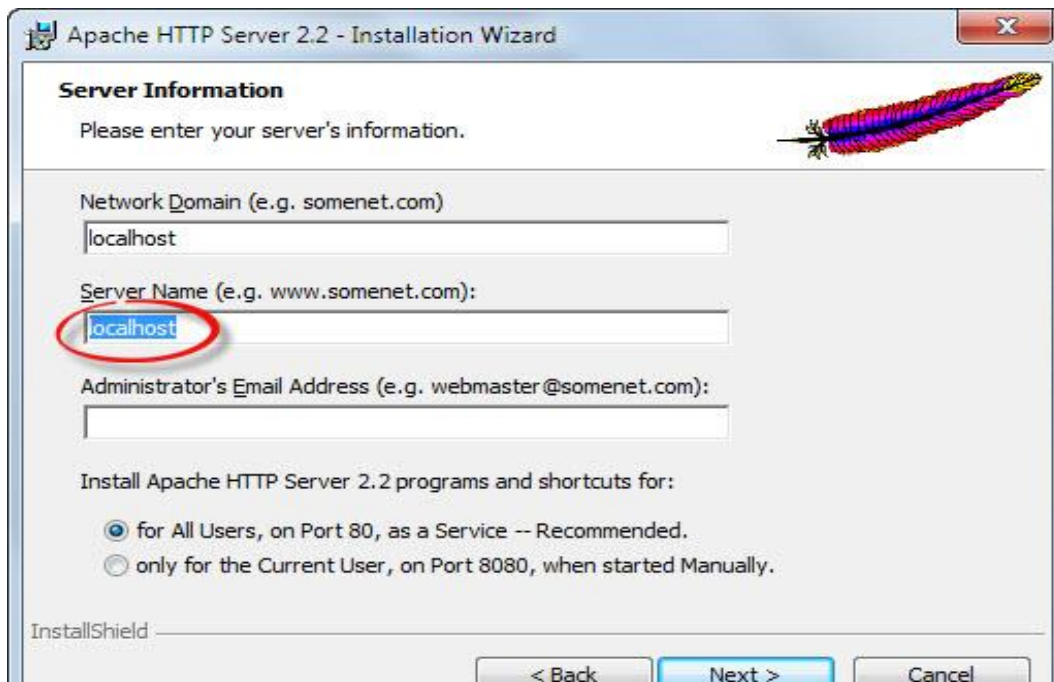


FIGURA 63: Inicio de Instalación Apache

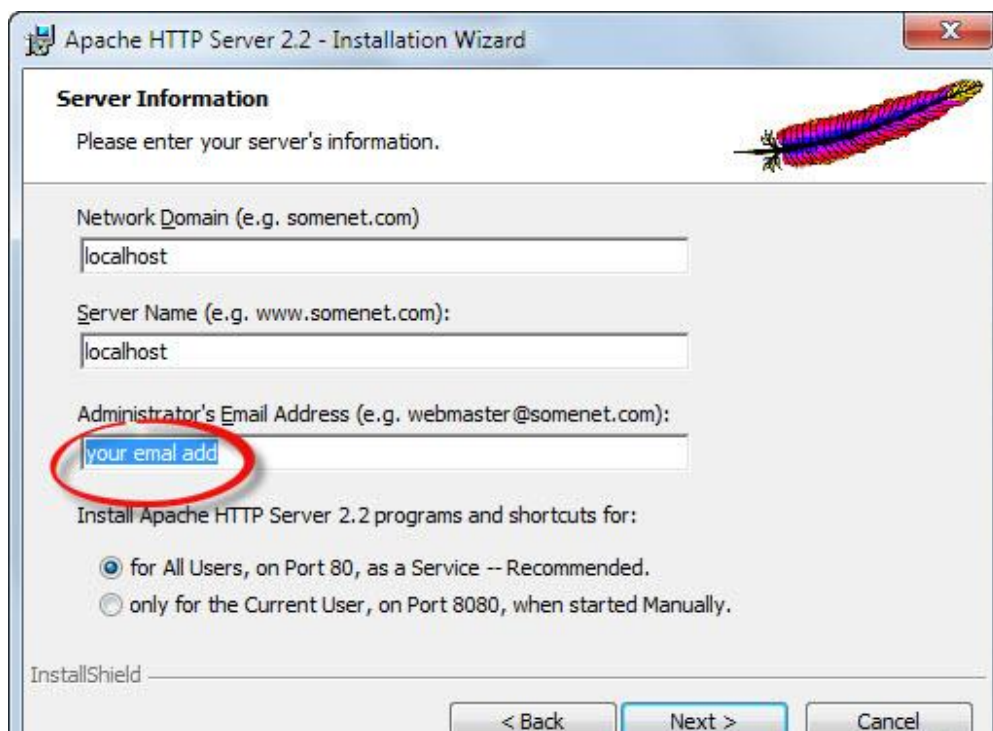
Fuente: Propia



- Escribir “localhost” en el dominio del servidor y en Nombre del servidor.

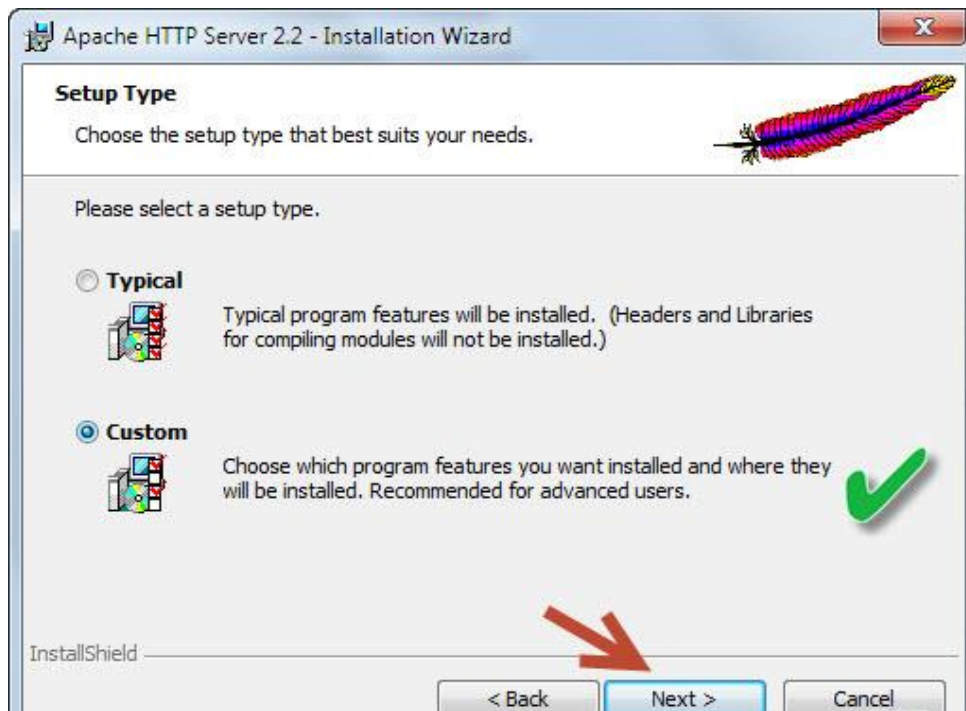


- Escribir una dirección de correo electrónico en dirección de correo electrónico del administrador.

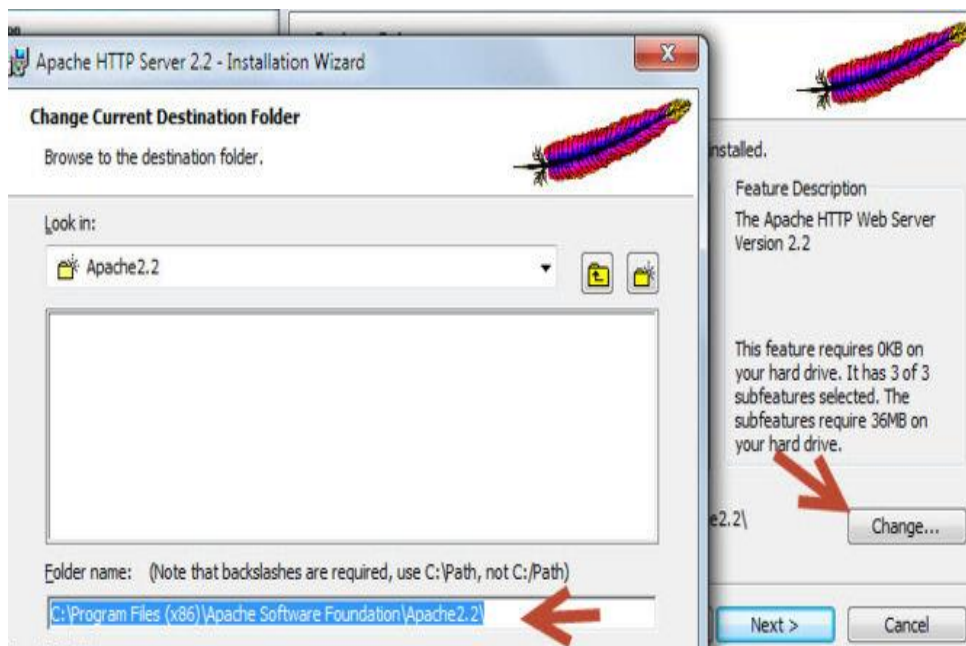


- Seleccionar “para todos los usuarios, en el Puerto 80, como un Servicio – Recomendado” y hacer clic en siguiente.

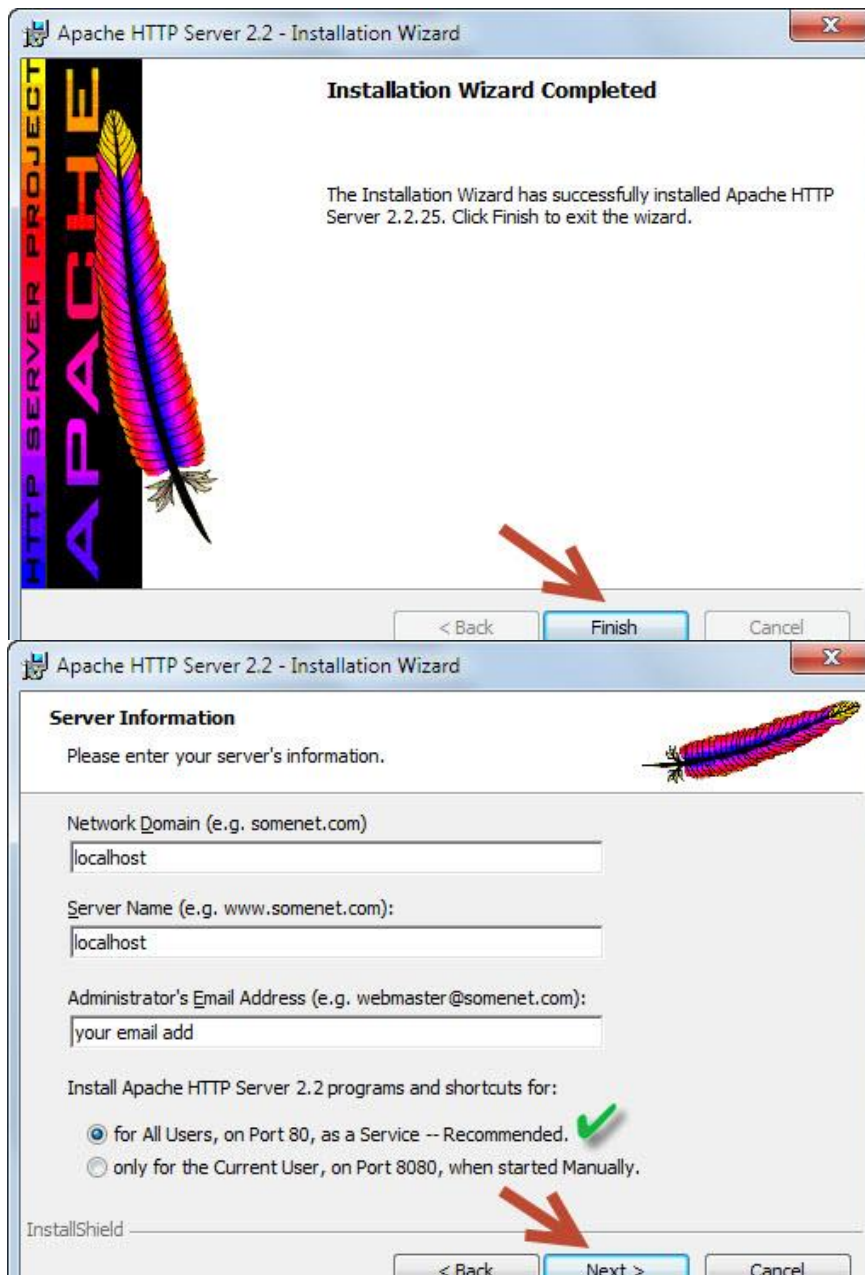
- Seleccionar tipo de organización personalizado y hacer clic en siguiente.



- Cambiar la carpeta del destinatario “D:\appserv \Apache 2.2” y hacer clic en Aceptar, después clic en siguiente y por último clic en instalar.



- Hacer clic en Finalizar.



- Probar si Apache se instaló correctamente ingresando en el navegador "localhost".



It works!

▪ Configurar PHP 5

1. Ingresar a la carpeta D:\appserv\php5 en el archivo php.ini.

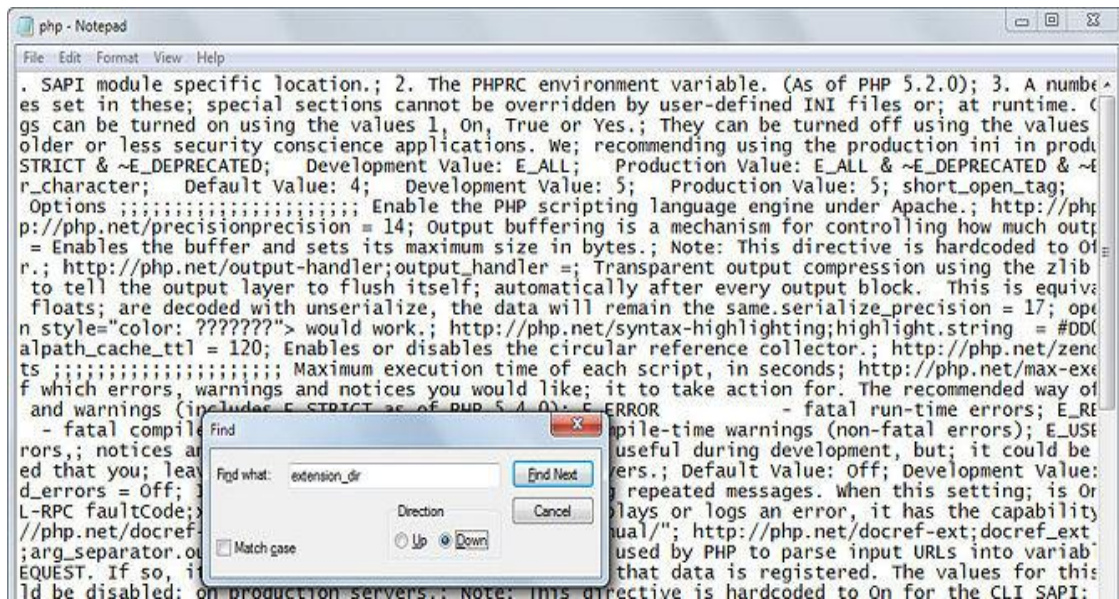
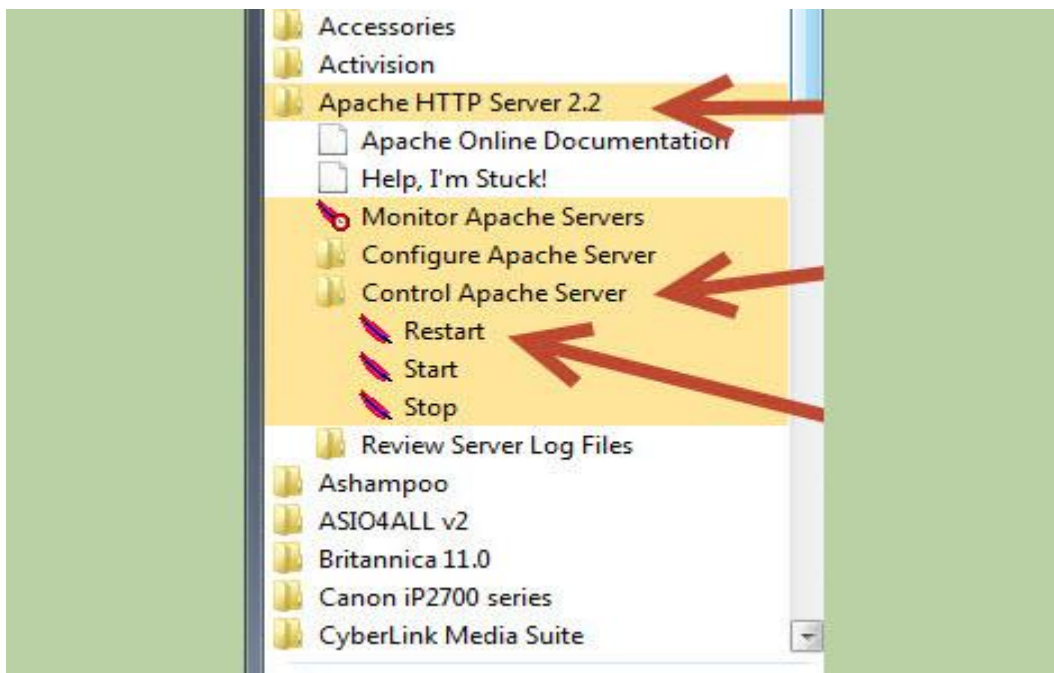


FIGURA 64: Configurando PHP

Fuente: Propia

doc_root = C:\Users\[mi nombre]\Documentos\[Carpeta Mis sitios web]



- Sustituir por: `extension_dir = "D:\appserv\php\ext"`

2. Guardar e ingresar al menú Inicio -> Todos los Programas -> Apache HTTP Server 2.2 -> Control de servicio Apache ->> Reiniciar.

3. Probar Apache y PHP.

Para comprobar si apache y php están configurados, crear un nuevo documento de texto (en el Bloc de notas, por ejemplo) en la carpeta www en D://appserv/www escribir "`<?php phpinfo(); ?>`" y después en Guardar como (Nombre del archivo) "phpinfo.php", escribir "Todos los archivos". Ingresar al buscador de internet y escribir "localhost/phpinfo.php".

- Se muestra la siguiente página al haber configurado correctamente php 5 y apache 2.2.



| | |
|---|--|
| System | Windows NT HPENVY 6.2 build 9200 (Windows 7 Home Premium Edition) i586 |
| Build Date | Mar 17 2011 10:34:15 |
| Compiler | MSVC9 (Visual C++ 2008) |
| Architecture | x86 |
| Configure Command | <code>cscrip /nologo configure.js "--enable-snapshot-build" "--disable-isapi" "--enable-debug-pack" "--disable-isapi" "--without-mssql" "--without-pdo-mssql" "--without-pi3web" "--with-pdo-oci=D:\php-sdk\oracle\instantclient10\sdk,shared" "--with-oci8=D:\php-sdk\oracle\instantclient10\sdk,shared" "--with-oci8-11g=D:\php-sdk\oracle\instantclient11\sdk,shared" "--enable-object-out-dir=../obj/" "--enable-com-dotnet" "--with-mcrypt=static"</code> |
| Server API | Apache 2.0 Handler |
| Virtual Directory Support | enabled |
| Configuration File (php.ini) Path | C:\Windows |
| Loaded Configuration File | D:\appserv\php5\php.ini |
| Scan this dir for additional .ini files | (none) |
| Additional .ini files parsed | (none) |
| PHP API | 20090626 |
| PHP Extension | 20090626 |
| Zend Extension | 220090626 |

- **Instalación de PostgreSQL**

1. Hacer doble clic sobre el icono y empezará la instalación.

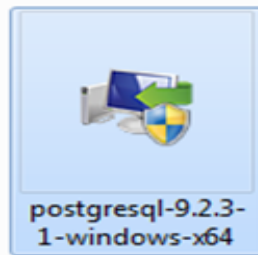


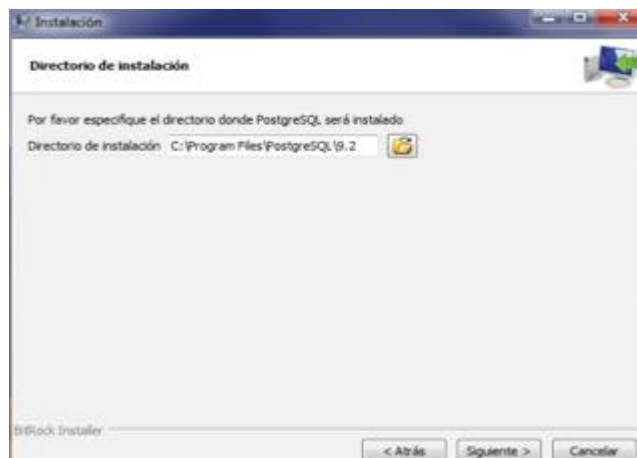
FIGURA 65: Inicio de Instalación PostgreSQL

Fuente: Propia

2. Hacer clic en siguiente.



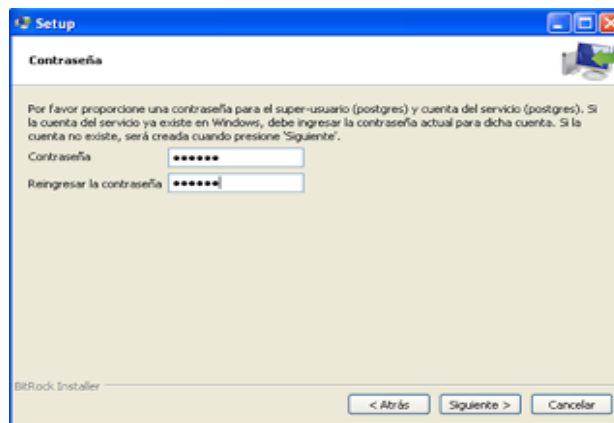
3. Poner la dirección o el directorio de instalación en donde se guardará el programa y hacer clic en siguiente.



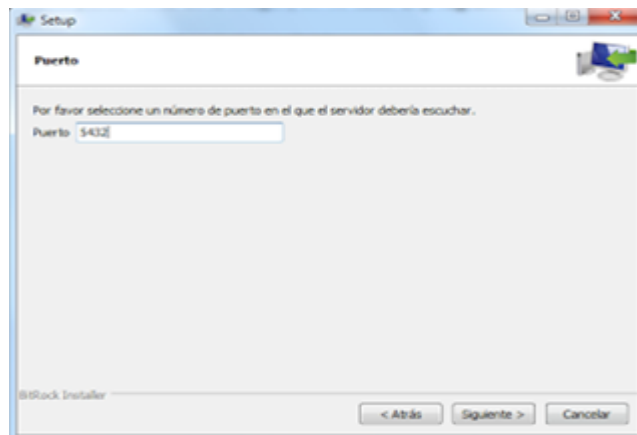
4. Poner la dirección en donde se guardará los datos y hacer clic en siguiente.



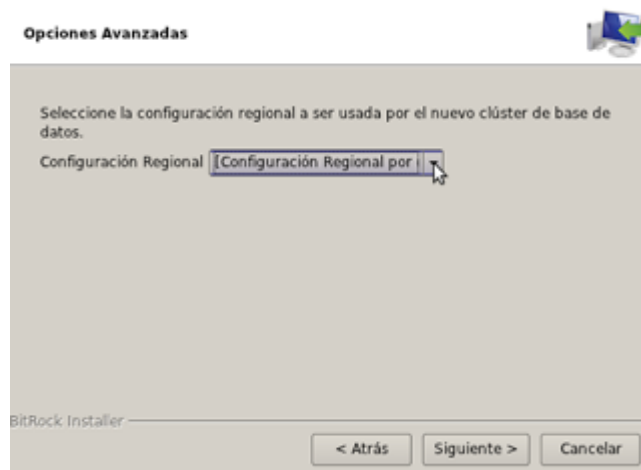
5. Ingresar usuario y contraseña de postgresql, hacer clic en siguiente.



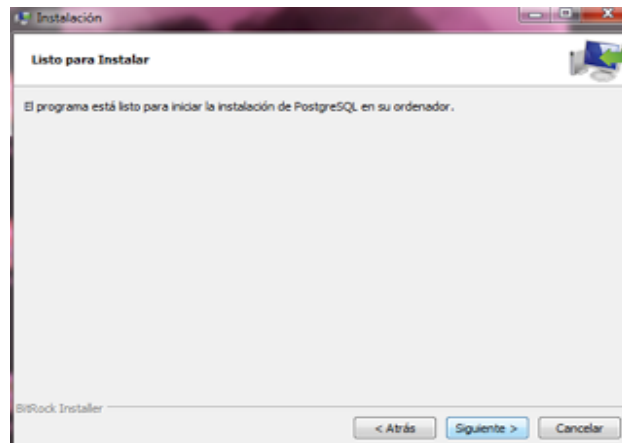
6. Dejar el puerto por default para comunicarse con el programa, el puerto es 5432:



7. Cambiar la configuración regional, dejar por default y hacer clic en siguiente:



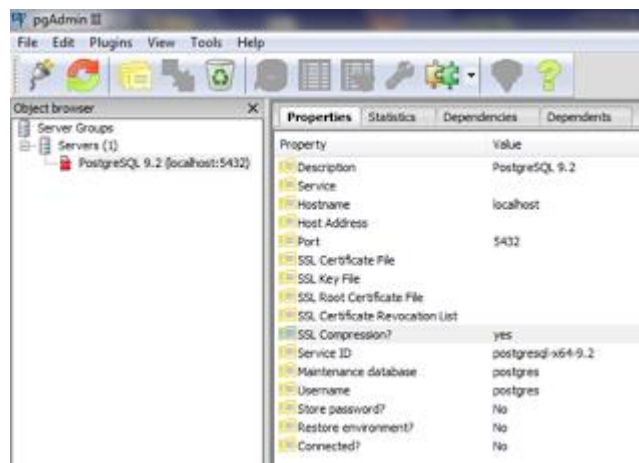
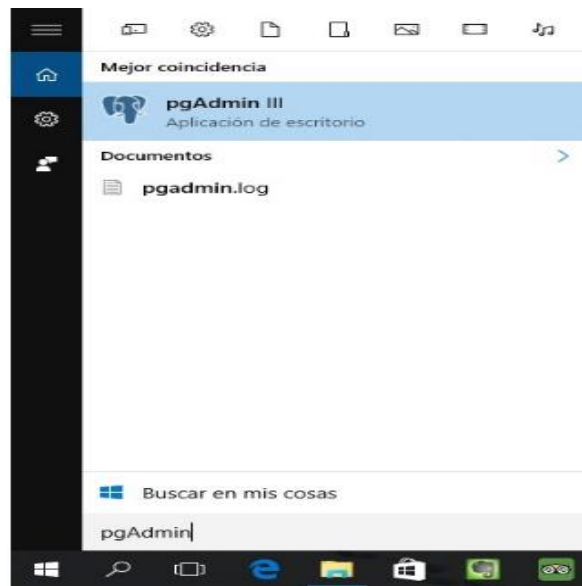
8. Hacer clic en siguiente.



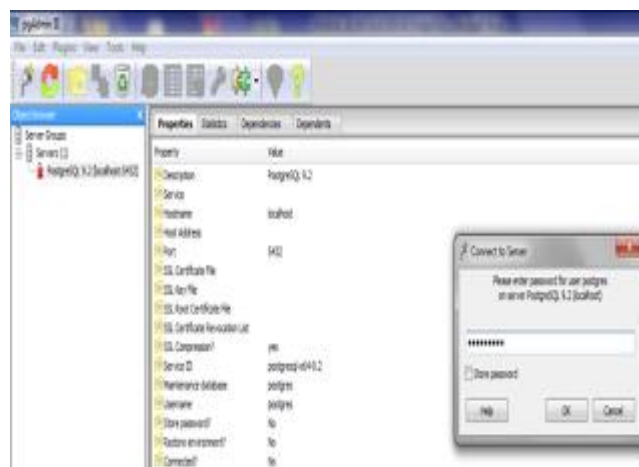
9. Hacer clic en terminar.

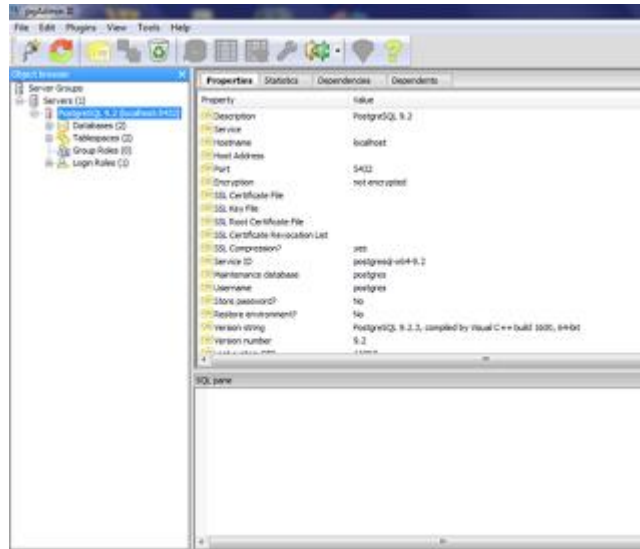


10. Buscar programa y hacer clic en PgAdminIII.



11. Ingresar contraseña y usuario de base de datos.





Backup y Restauración de Base de Datos en PostgreSQL y PgAdmin III

1. Acceder a PgAdmin III

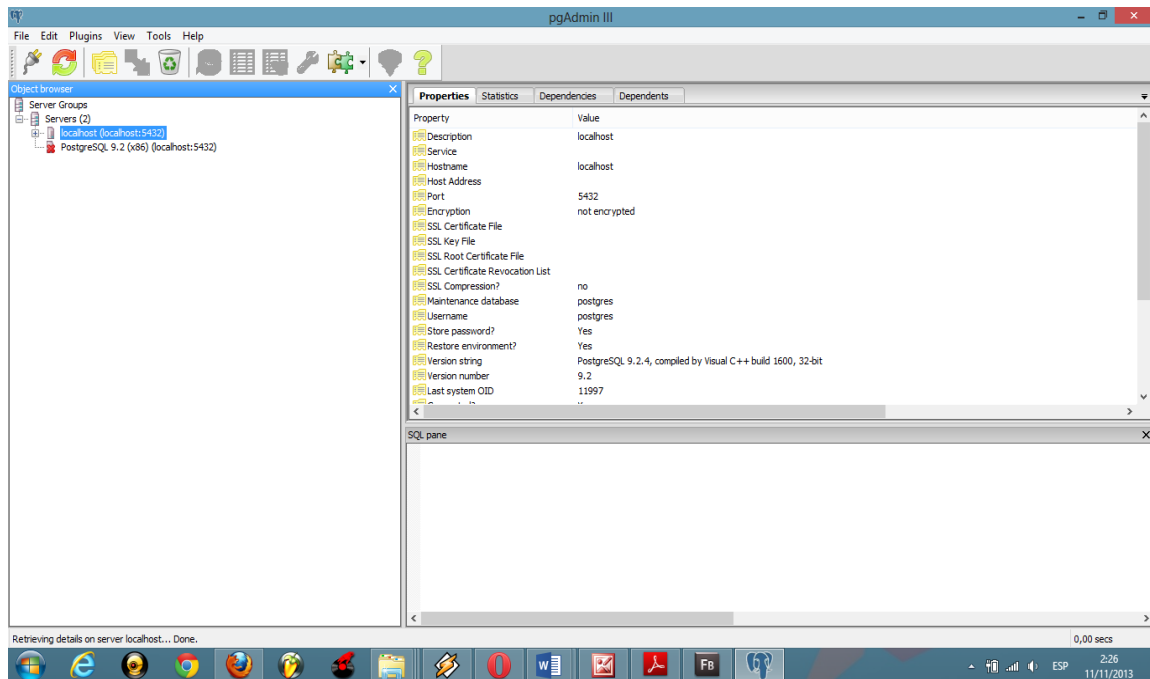
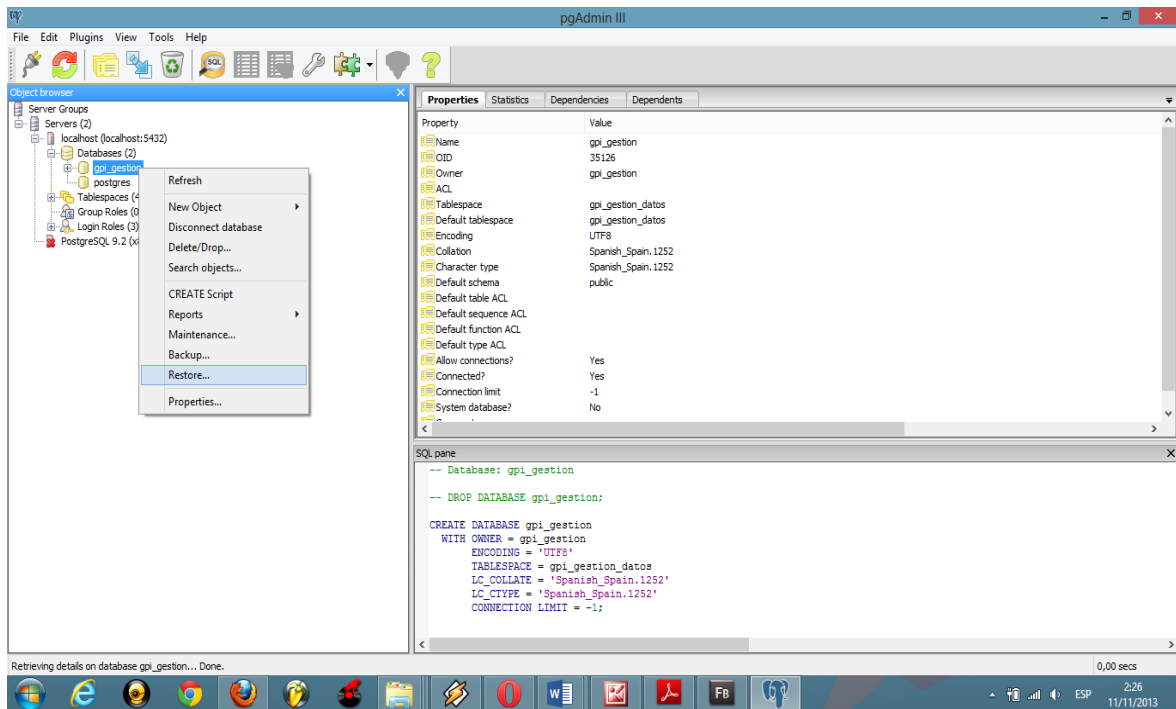


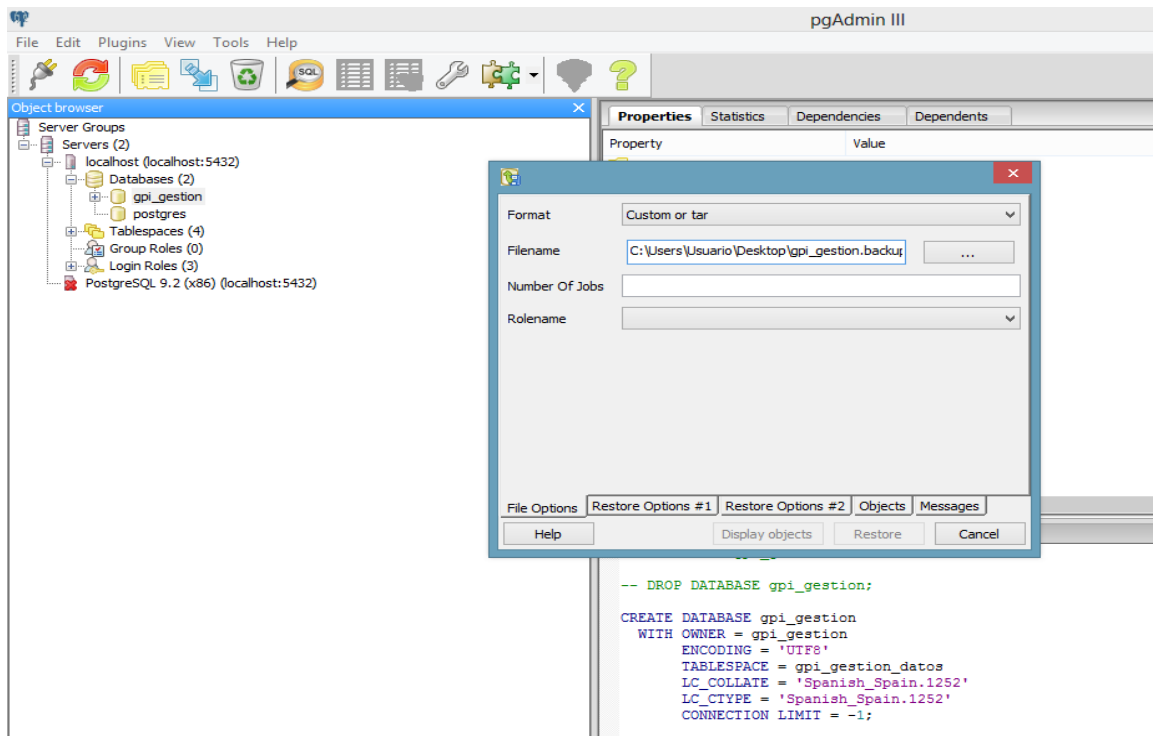
FIGURA 66: Inicio Restaurar BDD

Fuente: Propia

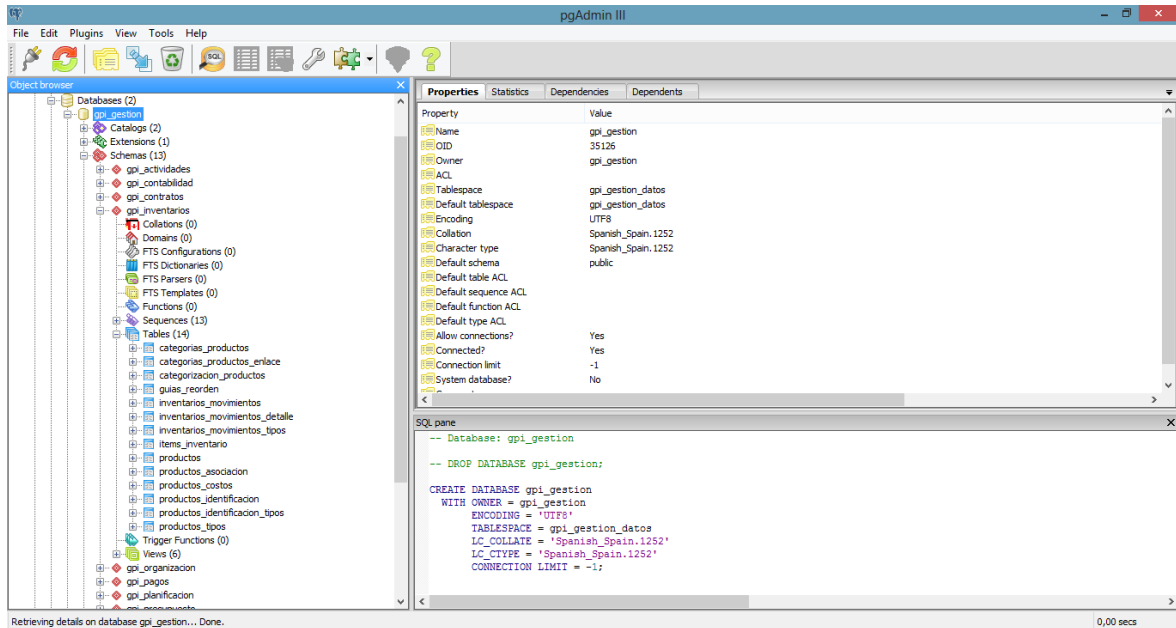
2. Crear base de datos con el nombre gpi_gestion, clic en restaurar y buscar el archivo de backup.



3. Ingresar la dirección del archivo backup en la caja Filename y en la lista de opciones Format poner "Custom o Tar" y presionar. Restore.



4. Listo. La información de toda la base de datos se importó.

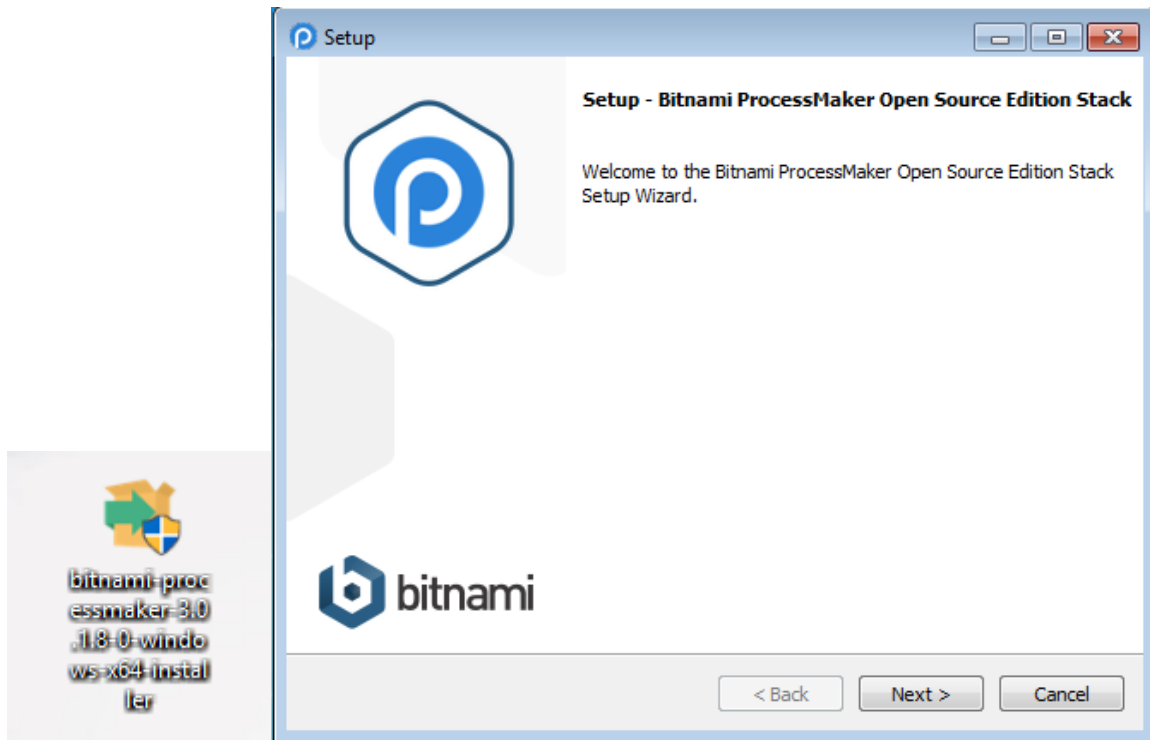


- **Instalación ProcessMaker**

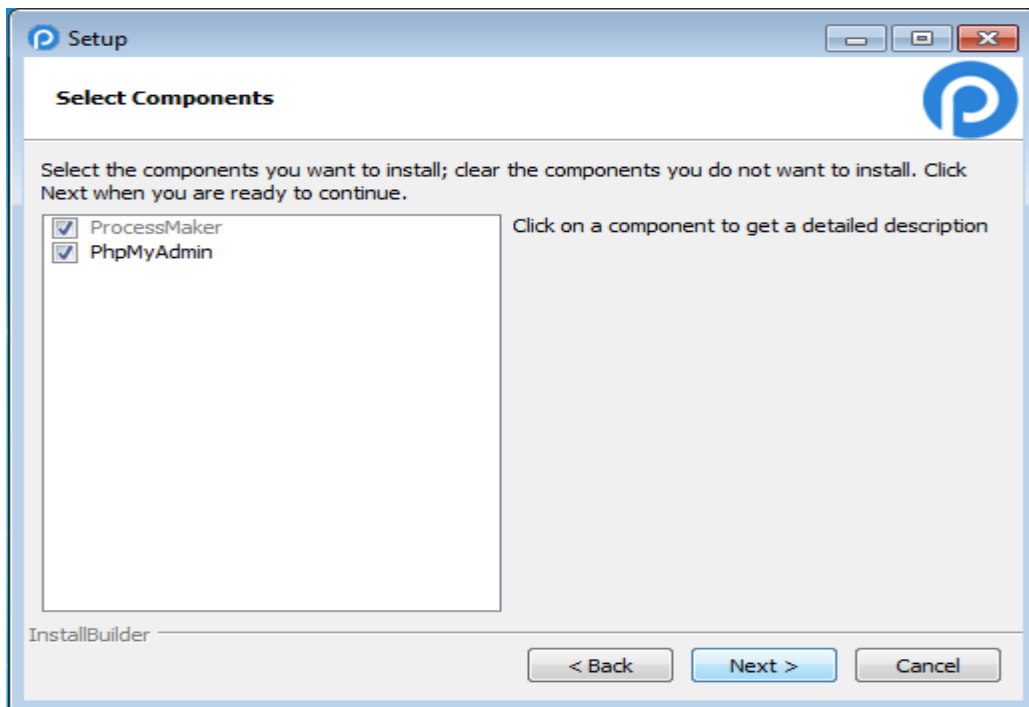
Requerimientos de Software

| Componente | Versiones soportadas | Descripción |
|----------------------|--|--|
| Plataforma | | |
| Windows | Windows 7, Windows 8, Windows 10, Windows Server 2012 R2 | |
| Linux | Cualquier versión (PHP) | Se instala en cualquier plataforma que soporte PHP |
| PHP | 5.3, 5.4, 5.5, 5.6 | De 3.0.1.7+ use PHP 5.4. De 3.0.1.8+ use PHP 5.5. |
| Base de datos | | |
| MySQL | 5.1, 5.5 | Planeado para conectar con base de datos externos. |
| Web Server | | |
| Apache | 2.2, 2.4 | |

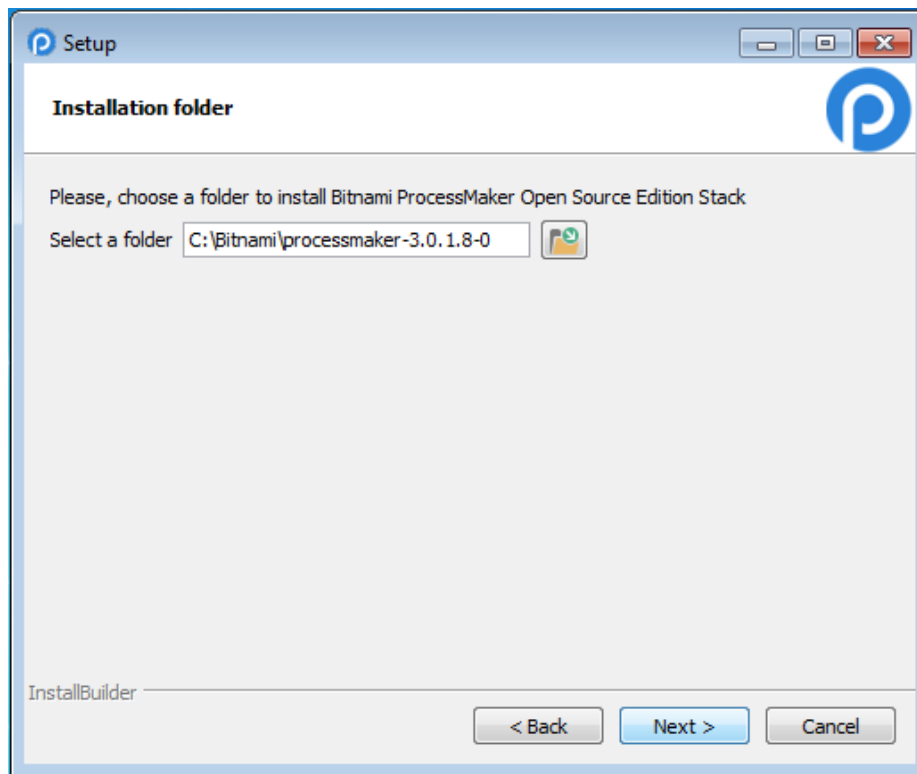
Hacer clic en el instalador ejecutable “bitnami-processmaker” para Windows x64 version 3.0.1.8, y luego hacer clic en Siguiente.



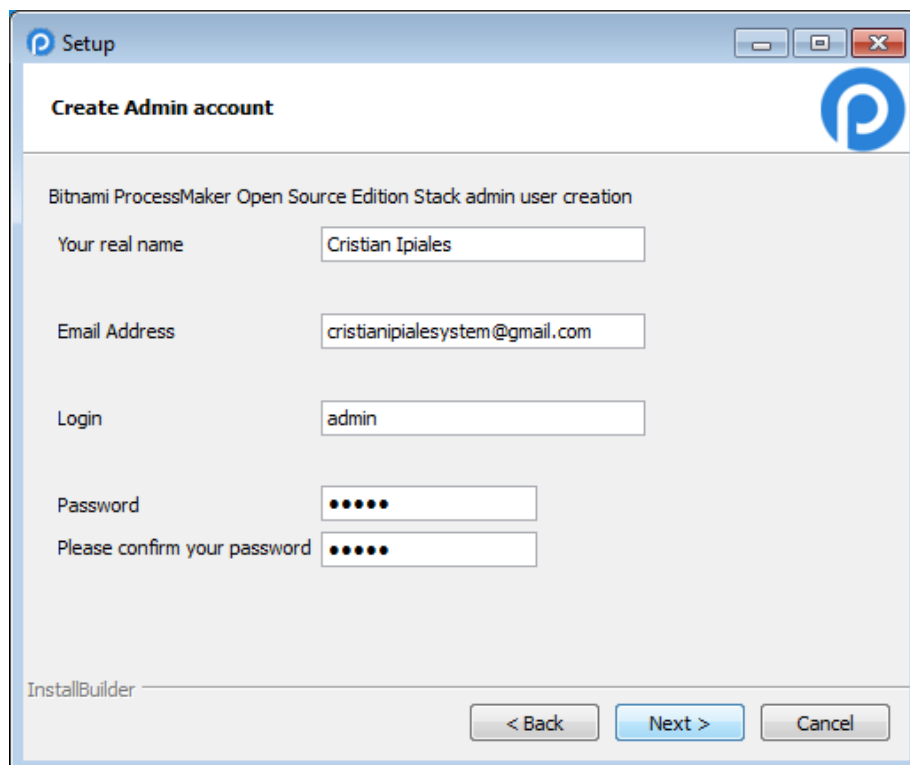
Seleccionar los componentes parte de la instalación ProcessMaker y PhpMyAdmin y hacer clic en siguiente.



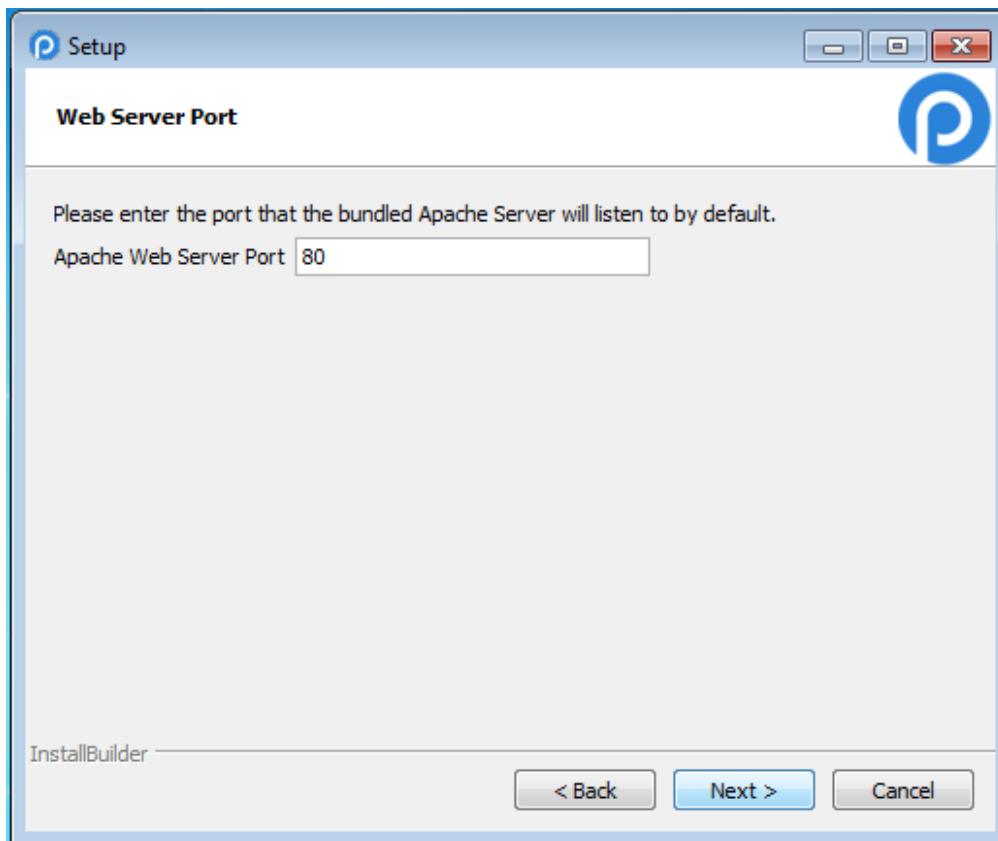
Elegir el lugar de instalación en el equipo y dar clic en siguiente.



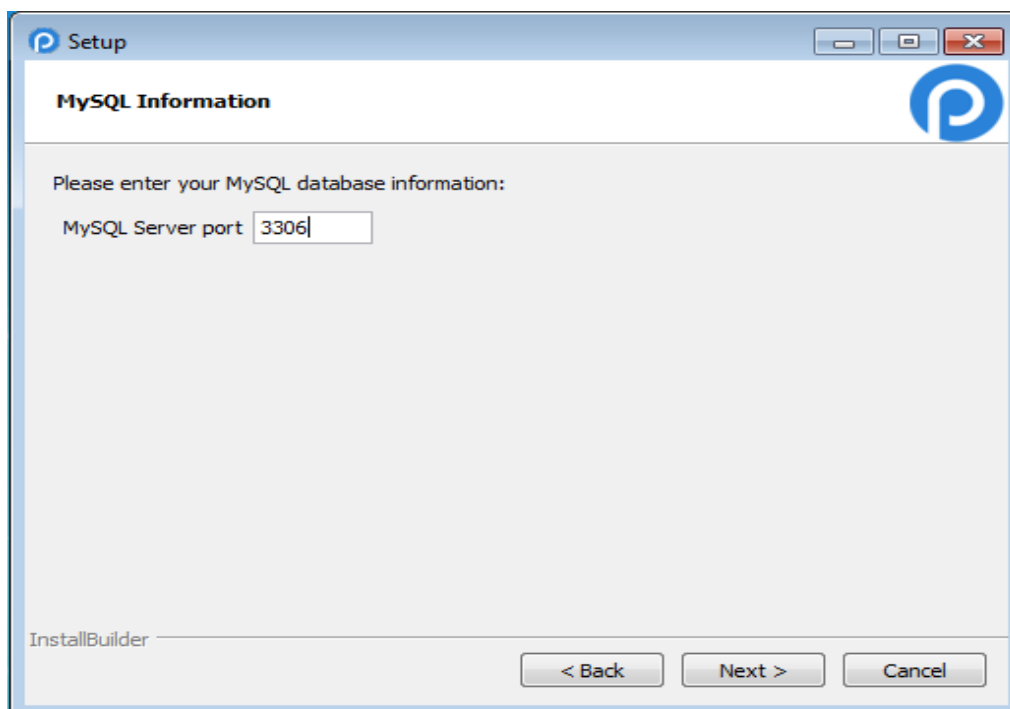
Crear una cuenta y un usuario administrador, dar clic en siguiente.



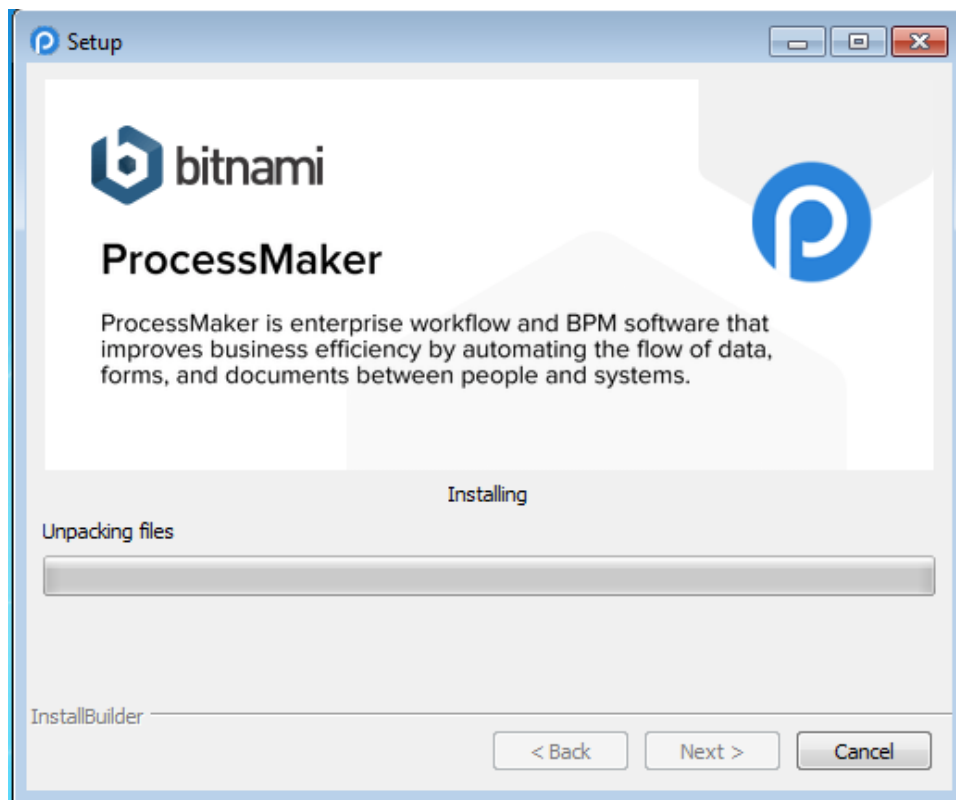
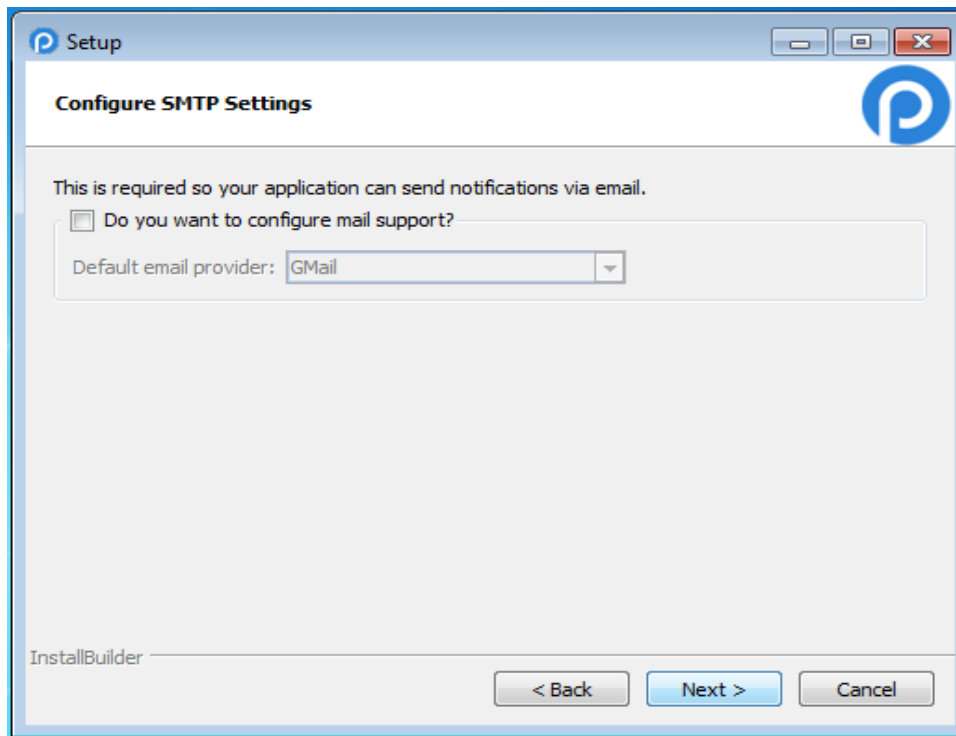
Elegir el puerto de instalación para apache integrado de processmaker.



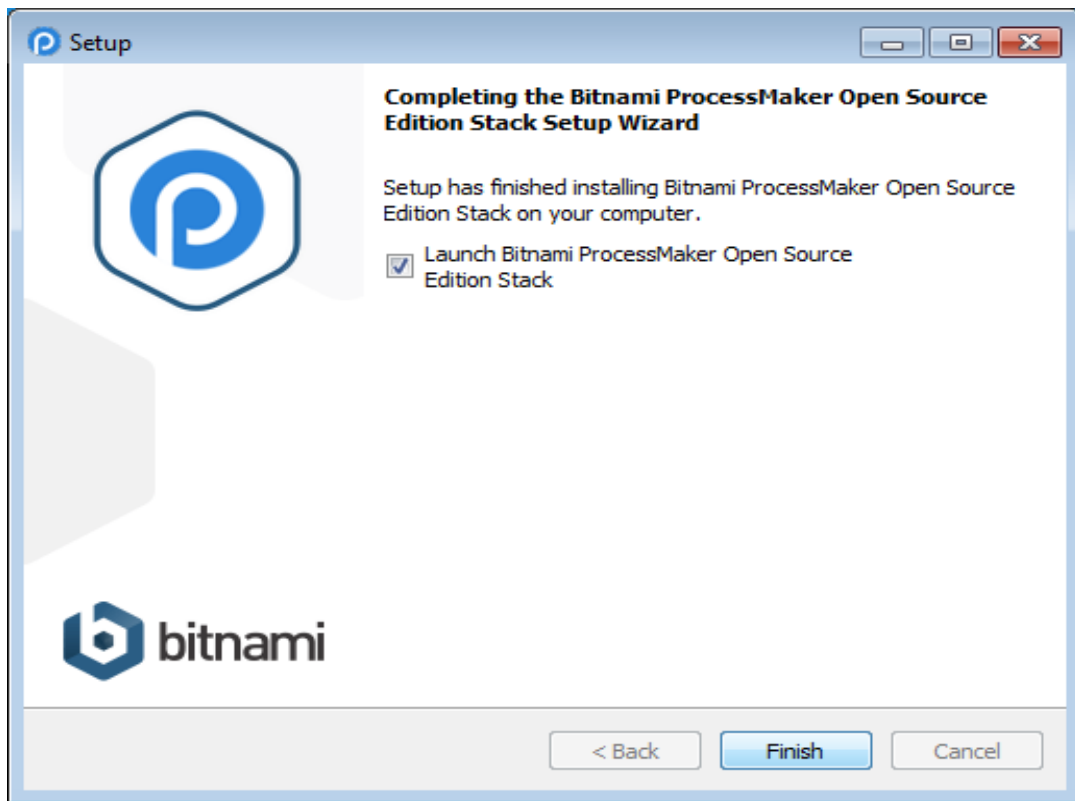
Elegir el puerto de instalación de MySQL y dar clic en siguiente.



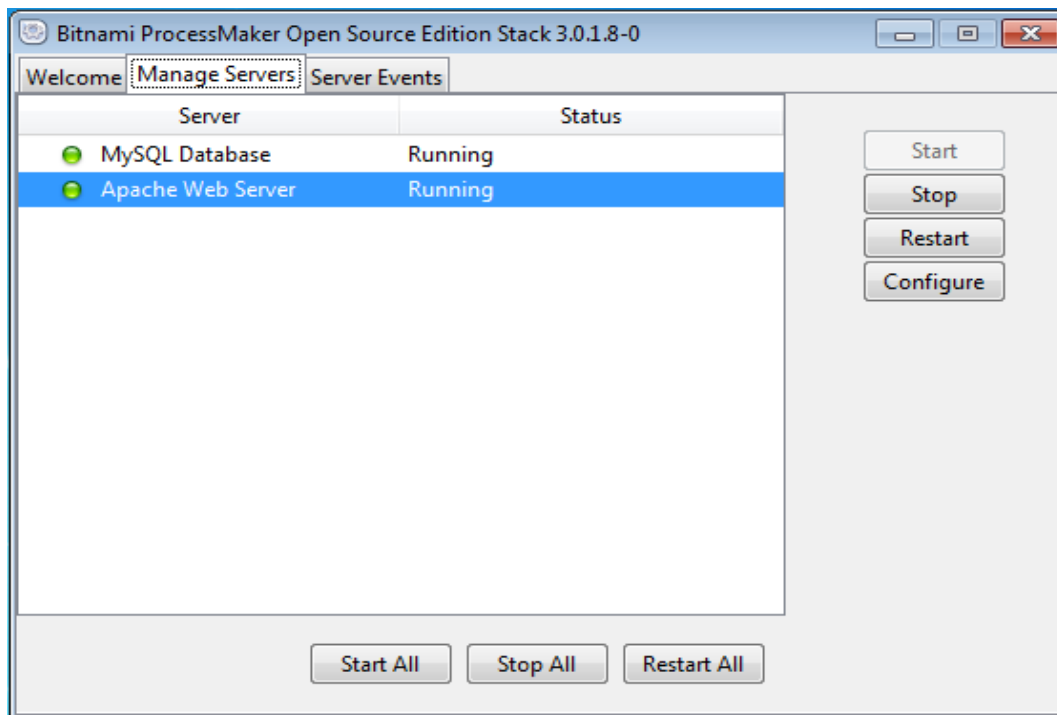
Configurar si es necesario el envío de notificaciones por email y hacer clic en siguiente.

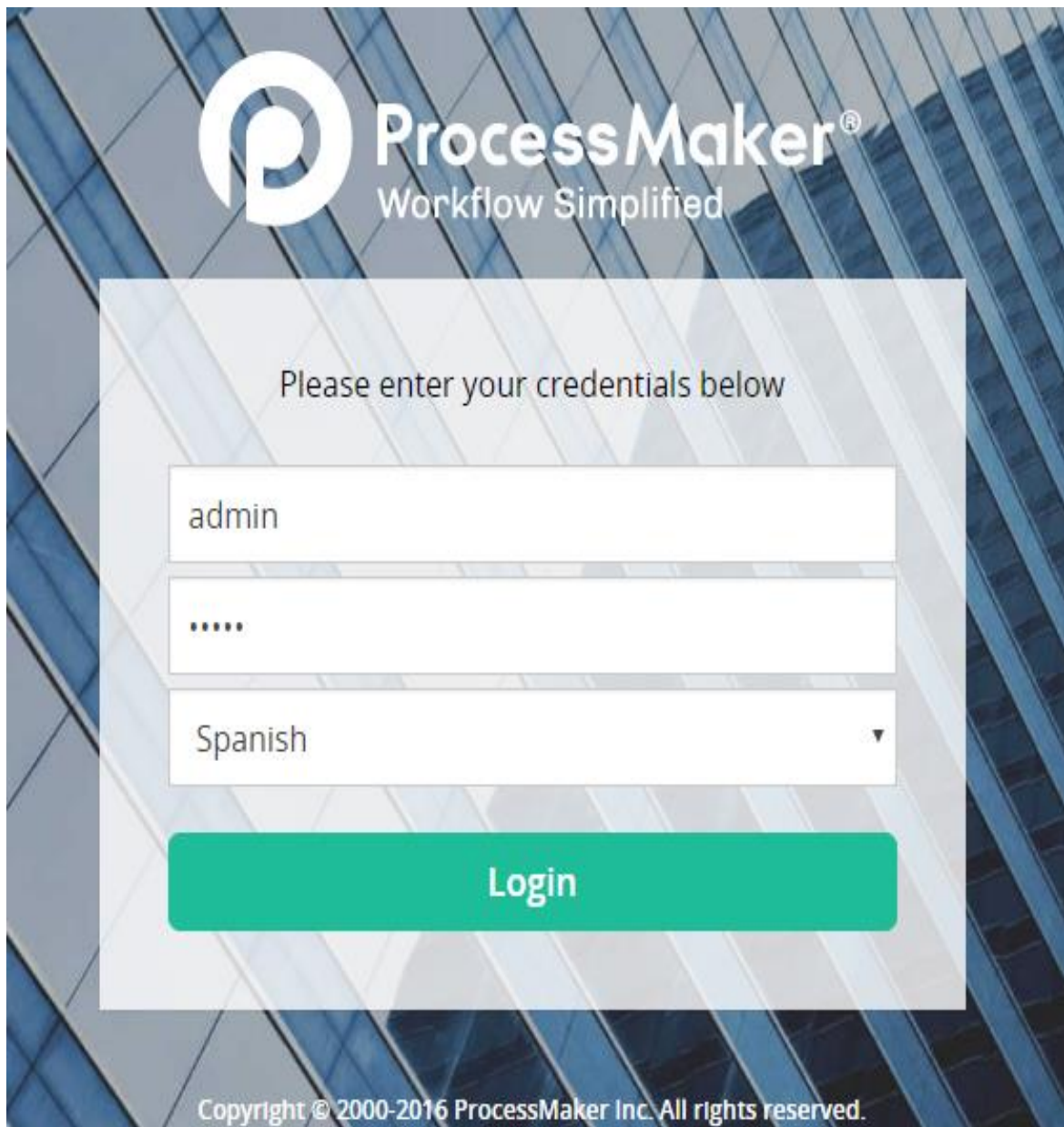


Finalizar la instalación



El administrador ProcessMaker inicia.





Por último, copiar appserv.rar de los archivos incluidos en el CD y extraer en el disco local D:

D:\appserv\www

Para revisar el correcto funcionamiento del sistema en el navegador web digitar la siguiente url de la página <http://localhost/GpiGestionBackend/public/bin-debug/GpiGestionFrontend.php>



Sub Dirección de GESTIÓN TECNOLÓGICA
© Copyright GPI
2015

FIGURA 67: Página de inicio del sistema.

Fuente: Propia

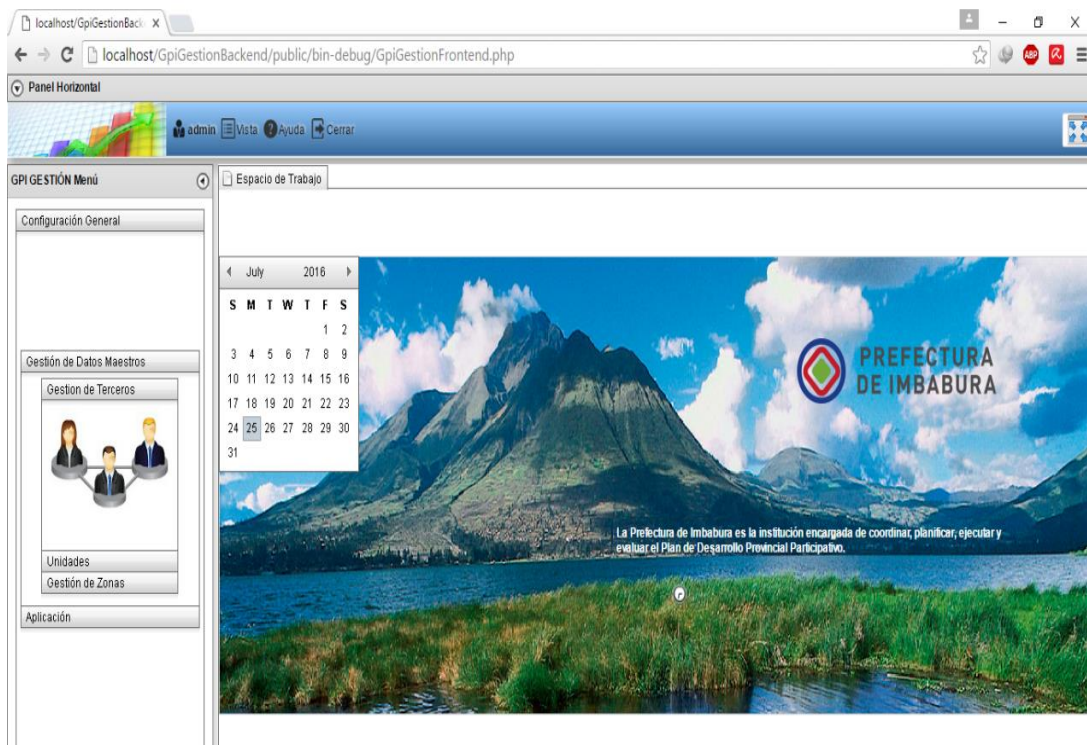


FIGURA 68: Menú de inicio del sistema.

Fuente: Propia

PRODUCTO - COMPUTADORES PORTATILES

| Id | Serie | Producto | Descripción | Existencias | Bodega | Organización Interna |
|----|-------|-----------------|-------------------------|-------------|----------------|------------------------|
| 5 | 123 | CAJA | CAJAS DE CARTONS | 200 | Bodega Norte | Administración General |
| 7 | 125 | COMPUTADORES PO | Computadores personales | 10 | Bodega Central | Administración General |
| 6 | 124 | COMPUTADORES PO | Computadores personales | 110 | Bodega Central | Consejo Provincial |
| 24 | 1234 | 425/65 R22.5 | quimico | 23 | Bodega Central | Administración General |

Items: 1 - 4 de 4

Costos

| Costo Inicio | Costo Fin | \$ Costo Estimado | Institución | Organización Interna |
|--------------|------------|-------------------|---------------------------------|---------------------------------|
| 2013-12-01 | 2013-12-31 | \$1459 | GOBIERNO PROVINCIAL DE IMBABURA | Gobierno Provincial de Imbabura |

Items: 1 - 1 de 1

FIGURA 69: Gestión de Ítems de Inventario.

Fuente: Propia

CATEGORIAS MENÚ

- 0 PRODUCTOS DE LA AGRICULTURA, SILVIC
- 1 MINERALES; ELECTRICIDAD, GAS Y AGUA
- 2 PRODUCTOS ALIMENTICIOS, BEBIDAS Y TA
- 3 OTROS BIENES TRANSPORTABLES, EXCEP
- 4 PRODUCTOS METALICOS, MAQUINARIA Y E
- 5 ACTIVOS INTANGIBLES ; TERRENOS, CONS
- 6 SERVICIOS COMERCIALES DE DISTRIBUCI
- 7 SERVICIOS FINANCIEROS Y SERVICIOS OC
- 8 SERVICIOS PRESTADOS A LAS EMPRESAS
- 9 SERVICIOS PARA LA COMUNIDAD, SOCIALE

Productos Tipos

| Codigo | Nombre |
|------------|--|
| 54614.00.1 | - SERVICIOS DE INSTALACION DE TODOS LOS TIPOS DE ANTENA PARA EDIFICIOS RESIDENCIALES, INCLUIDAS LAS ANTE |
| 54640.00.1 | - SERVICIOS DE INSTALACION PARA EL SUMINISTRO DE DIVERSOS FLUIDOS (POR EJEMPLO, OXIGENO EN LOS HOSPITA |
| 71951.01.1 | ANALISIS DE INFORMACION CONFIDENCIAL SOBRE MERCADOS |
| 43540.00.1 | ASCENSORES PARA PERSONAS, MERCANCIAS |

Items: 1 - 500 de 5441

Productos Específicos

| Codigo | Nombre |
|-----------|--|
| 546140011 | - SERVICIOS DE INSTALACION DE TODOS LOS TIPOS DE ANTENA PARA EDIFICIOS RESIDENCIALES, INCLUIDAS LAS ANTE |
| 361132913 | (12.4 R 38) ROW CROP |
| 321940016 | (DIAFANO) |
| 361130011 | 10.00.R15 |

Items: 1 - 500 de 23539

FIGURA 70 Clasificación de Productos.

Fuente: Propia

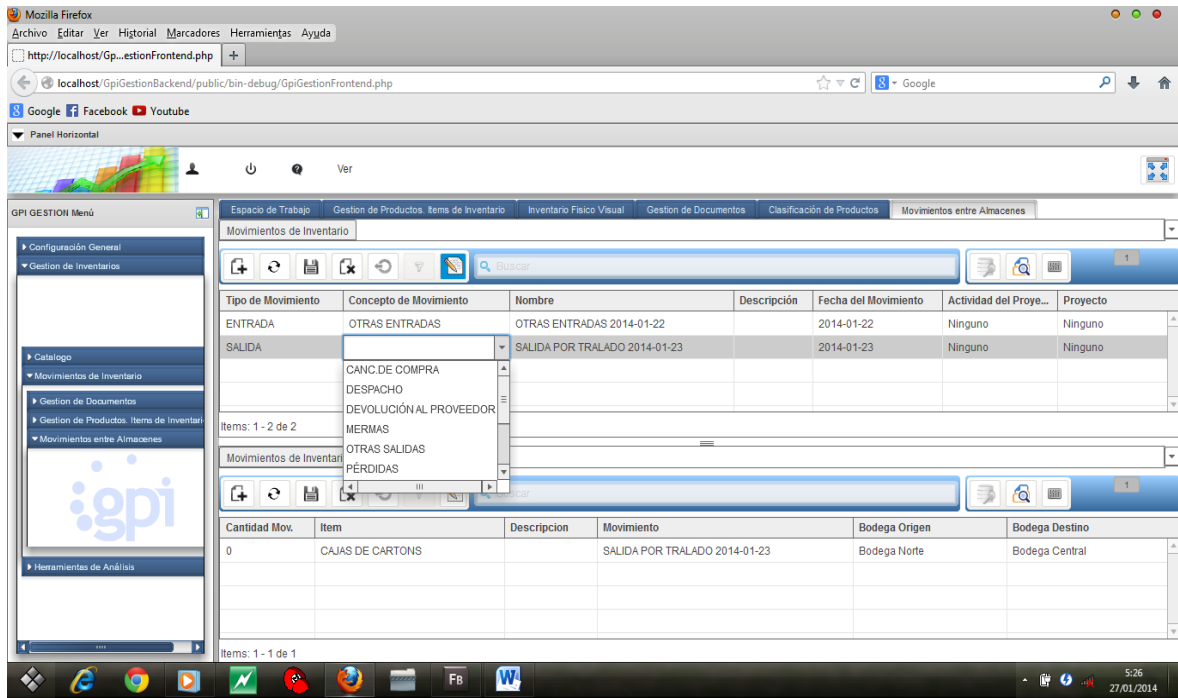


FIGURA 71: Movimientos de Inventario.

Fuente: Propia

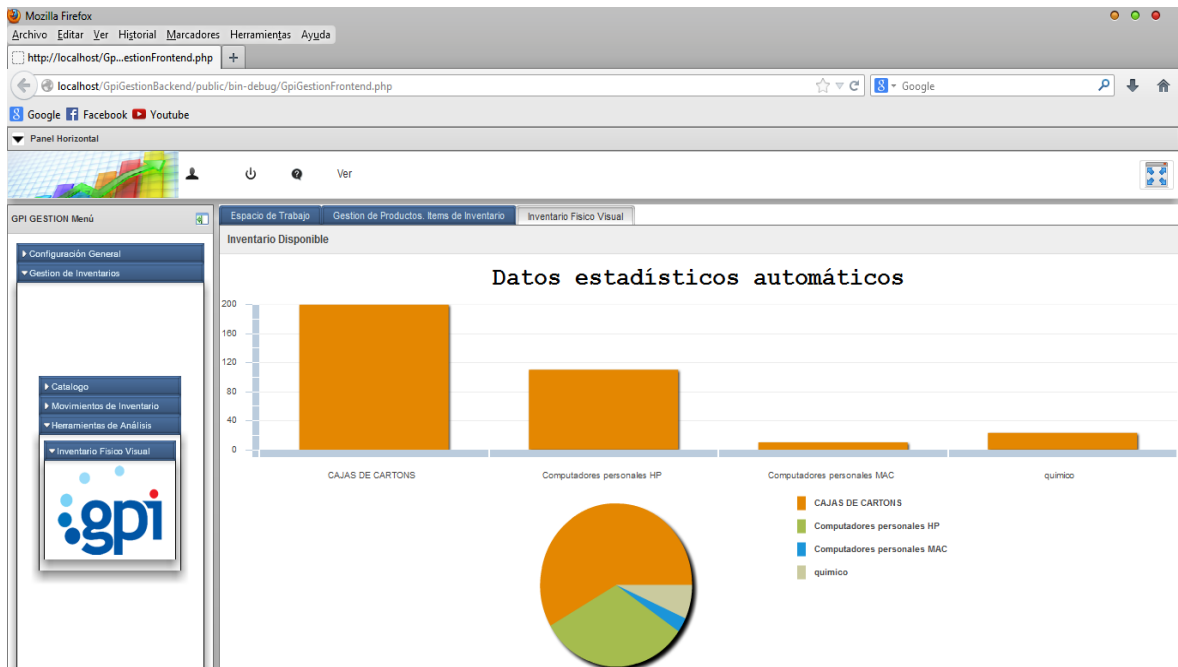


FIGURA 72: Datos Estadísticos de Ítems de Inventario.

Fuente: Propia

CAPÍTULO V

5 CONCLUSIONES Y RECOMENDACIONES

5.1 CONCLUSIONES

- 1) El Gobierno Provincial como entidad pública a fin de reducir costos, se planteó realizar una suite de sistemas con interfaz amigable para el usuario que permitan realizar transacciones en forma automática, rápida y eficiente.
- 2) Scrum es la metodología ágil de trabajo en equipo que proporciona un enfoque disciplinado en la asignación de responsabilidades, cada iteración ejecutados en periodos cortos y fijos proporcionan un resultado completo o parcial que puede ser entregado al cliente cuando lo solicite.
- 3) RUP es un proceso de Ingeniería de Software que proporciona una metodología efectiva para modelar y documentar un proyecto de software.
- 4) La utilización de software libre, proporciona el acercamiento a las nuevas tecnologías como parte del proceso educativo, además de los beneficios económicos ya que no es necesario pagar por licencias de software.
- 5) El Sistema de Gestión de Inventarios está completamente integrado a los demás módulos, es funcional y de fácil uso, diseñado y desarrollado exclusivamente para cubrir todas las necesidades de los interesados.
- 6) Al automatizar los procesos de gestión de inventarios se obtiene:
 - a. Ahorro de tiempo y de costos al momento de realizar las transacciones.
 - b. De manera rápida y actualizada aquella información relacionada con los inventarios.

- 7) La documentación, información, foros, blogs sobre Flex es aún muy escasa en el manejo con base de datos y su integración a Flex 4 es una versión más actualizada y trae muchos cambios.
- 8) La utilización de Zend Framework, ayudó a desarrollar la aplicación de manera confiable y rápida, además ofrece un gran rendimiento, robusta implementación Patrón Modelo, Vista, Controlador y una abstracción de base de datos fácil de usar.
- 9) El diseño de la base de datos fue desarrollada en base a las necesidades, tamaño de la información, facilidad de acceso, extracción de la información requerida y enfoque Multi-Empresas que permita operar con ilimitada cantidad de empresas o instituciones en el mismo sistema.

5.2 RECOMENDACIONES

- 1) Las instituciones y organizaciones de gobierno como parte del cambio y avance tecnológico deben ejecutar sus procesos en un sistema web el cual permita realizar transacciones en forma automática, rápida y eficiente.
- 2) El funcionamiento de cada módulo del sistema y la relación que existe entre ellos es importante para que la ejecución de cada uno de los procesos del Gobierno Provincial de Imbabura sean eficaces y de calidad, sin equivocaciones.
- 3) Se recomienda utilizar la metodología RUP y SCRUM conjuntamente en proyectos de Desarrollo de Software, así la información del desarrollo e implementación estará documentada, además de mantener un orden, control, priorización de tiempo para obtener resultados óptimos de las actividades.

- 4) Se recomienda utilizar software libre para el desarrollo y la implementación de aplicaciones o sistemas web, ya que el software libre mantiene las características del software comercial.
- 5) Se requiere de una adecuada administración de los módulos y base de datos para la extracción fácil y ágil de la información de base de datos.
- 6) Antes de seleccionar una herramienta para el desarrollo de un sistema se debería analizar las ventajas y desventajas de dicha herramienta. Luego de estudiar la herramienta y empezar a desarrollar el aplicativo, se dieron las siguientes desventajas:
 - a. La demora en la carga de los módulos y extracción de datos si se tiene mucha información en la Base de Datos.
 - b. La inadecuada actualización de cambios hechos sobre las formas y módulos.
- 7) Tener conocimientos básicos PHP para la utilización de Zend Framework, instalación fácil y simple para consultar base de datos, sin tener que escribir ninguna consulta SQL.
- 8) Amplios conocimientos en diseño y administración de Base de Datos para un enfoque Multi-Empresa.

GLOSARIO

- **Introducción.**

El presente glosario forma parte del sistema “GPI_INVENTARIOS” y es una compilación de términos empleados en todo el sistema.

Representa una guía general de las definiciones y conceptos que se usan con frecuencia y tienen una relevante importancia para la adecuada comprensión dentro del sistema.

- **Propósito**

La función principal del glosario es identificar conceptos y términos del sistema.

- **Glosario**

Actor. Alguien o algo externo al sistema que interactúa con él.

Administrar. (Aspectos normativos y técnicos del funcionamiento de una organización), es llevar a cabo aquellos actos que se realizan sobre la empresa con la finalidad de conservarla o explotarla siguiendo la política marcada para alcanzar los objetivos planteados.

Administración de inventarios. Es la eficiencia en el manejo adecuado del registro, de la rotación y evaluación del inventario de acuerdo a como se clasifique y qué tipo de inventario tenga la empresa, ya que a través de todo esto determinaremos los resultados (utilidades o pérdidas) de una manera razonable.

Base de datos. Las bases de datos permiten almacenar, buscar, ordenar y recuperar datos de forma eficiente.

Bodega o Almacén General (Establecimiento). Es un espacio destinado, bajo ciertas condiciones, al almacenamiento de distintos bienes.

Caso de Uso. Secuencia de acciones que el sistema realiza, la cual proporciona un resultado de valor observable.

Catálogo de bienes. Relación ordenada de claves y descripción de Producto que serán utilizados para el correcto control y registro de los inventarios.

Categoría de Producto. Una categoría de producto consiste en todos los productos que ofrecen la misma funcionalidad general.

Control de Inventarios. Es la técnica que permite mantener la existencia de productos a niveles deseados.

Costo. Denominado al valor monetario asignado a un bien o servicio.

Data Modeler. Herramienta para el modelado de bases de datos.

Gestionar. (Aspecto económico), es aplicar las reglas, procedimientos y métodos operativos para llevar a cabo con eficacia las actividades económicas que permiten lograr esos objetivos marcados por la empresa.

Gestión de inventarios. Es poner a disposición de las áreas de producción o comercial una determinada cantidad de producto en el momento preciso, en el lugar oportuno y con el mínimo coste posible.

Inventario. Relación de recursos materiales existentes en un almacén.

Inventario Inicial. Inventario que refleja la cantidad de existencia que una empresa tiene al final del proceso contable al iniciar sus actividades después de hacer un conteo físico.

Inventario Final. Inventario que refleja la cantidad de existencia que una empresa tiene al final del proceso contable, coincidirá con el inventario inicial del siguiente periodo.

Inventario Físico. Es el inventario de la mercancía, verificación o confirmación de materiales existentes o productos de una empresa.

Manejo del inventario. El proceso de asegurar la disponibilidad de los productos a través de actividades de administración de inventario como planeación, posicionamiento de stock, y supervisión de la edad del producto.

Nivel óptimo de inventario. Existencia recomendable de recursos materiales para el funcionamiento en un almacén.

Paquetes. Agrupaciones de casos de uso y actores por funcionalidad que proveen.

Productos. Un producto es un conjunto de características y atributos tangibles (forma, tamaño, color...) e intangibles (marca, imagen de empresa, servicio...) para satisfacer un deseo o una necesidad.

Recursos materiales. Son todos aquellos bienes que posee la institución, tanto los que hayan sido entregados como aportes de los propietarios, como los que hayan sido recibidos por adquisiciones a terceros a cualquier título: compra, trueque, dación en pago o donaciones, siempre que sean susceptibles de ser valorizadas.

Stakeholder. Parte interesada del Proyecto.

Toma de inventario físico. La toma de inventario, es un proceso que consiste en verificar físicamente los bienes con que cuenta cada entidad, a una fecha dada; con el fin de asegurar su existencia real. La toma de inventarios permite contrastar los resultados obtenidos valorizados, con los registros contables, a fin de establecer su conformidad, investigando las diferencias que pudiera existir y proceder a las regularizaciones del caso.

Valuación. Del inventario en el cual cada artículo puede asociarse con una factura específica, que sirve para su valuación correspondiente.

REFERENCIAS

Adobe. (2014). *ActionScript Technology Center*. Obtenido de www.adobe.com/devnet/actionscript.html

Adobe, F. P. (2014). *Adobe Flash Player*. <http://adobe.com>.

adobe.com. (2014). <http://www.adobe.com/>.

AGESIC. (s.f.). *Seguridad del Portal*. Obtenido de agesic.gub.uy/innovaportal/file/549/1/Capitulo_5_Seguridad_v1_0.pdf

applesfera. (2014). <http://www.applesfera.com>.

Bert Bos, C. C. (2014). *Cascading Style Sheets*. Obtenido de W3c: www.w3.org/Style/CSS/

Bista, S. (12 de 03 de 2015). *ProcesMaker*. Obtenido de <http://www.processmaker.com/es>

Ciber. (2014). *Ciberaula. Una Introducción a APACHE*. Obtenido de http://linux.ciberaula.com/articulo/linux_apache_intro

CIBERAULA. (2014). Linux apache intro. *Linux apache intro*. Obtenido de http://linux.ciberaula.com/articulo/linux_apache_intro

Cindy, E. B. (s.f.). *Bligoo*. Obtenido de http://elblogdecindy.bligoo.com/media/users/17/874375/files/173246/sistema_operativo_definicion.pptx

Diagramas HIPO. (2014). Obtenido de Diagramas HIPO: <http://www.buenastareas.com/ensayos/Venezuela-200-A%C3%B1os/3528476.html>

Diego. (2014). *IBM*. Obtenido de Developers Works: <http://www.ibm.com/>

Ecured. (2014). *Base de Datos*. Obtenido de <http://www.ecured.cu/index.php/PostgreSQL>

extremeprogramming. (s.f.).

<http://extremeprogramming.host56.com/ARTICULO7.php>.

Flex Enterprise Services. (2014).

FundaWeb. (2014). *FundaWeb*. Obtenido de

<http://fundaweb.wikispaces.com/Lenguajes>

glosario. (2014). *GLOSARIO DE INFORMÁTICA*. Obtenido de

<http://www.internetglosario.com/>

GPI. (2014). Obtenido de <https://secure.urkund.com/view/document/19034741-557364-182231/download>

GPI. (2014). Documentación Institucional. *Centro de Documentación Institucional*. Obtenido de www.imbabura.gob.ec

GPI. (2014). Estatuto orgánico de gestión organizacional. *Organización por procesos*. Ibarra, Imbabura, Ecuador.

GPI. (2014). Plan estratégico institucional. *PLAN ESTRATÉGICO INSTITUCIONAL 2014 - 2019*. Ibarra, Imbabura, Ecuador. Obtenido de <http://www.imbabura.gob.ec/transparenciagpi/K/Plan-Estrategico-Institucional-2014-2019.pdf>

GPI. (2014). Prefectura de Imbabura. *Proforma*. Ibarra, Imbabura, Ecuador. Obtenido de <http://www.imbabura.gob.ec/>

Group, O. M. (2014). *BPMN*. Obtenido de <http://www.bpmn.org/>

Group, T. P. (2014). *PHP*. Obtenido de <http://php.net/>

IBM. (2014). BPM. *BPM*. Obtenido de <http://www.ibm.com/>

IS. (2014). Obtenido de Casos de Uso:

<https://users.dcc.uchile.cl/~psalinas/uml/casosuso.html>

IS. (2014). *Ingeniería Software*. Obtenido de www.informatica.us.es

ISTICC. (2014). Artículo. *SISTEMA DE GESTIÓN Y CONTROL DEL INVENTARIO DE CONSUMO INTERNO*. Obtenido de <http://repositorio.utn.edu.ec/bitstream/123456789/1070/2/04%20ISC%20230-Articulo-cientifico.pdf>

jmhogua. (2014). *Lógica de Negocios*. Obtenido de <http://jmhogua.blogspot.com/2007/02/capa-lgica-de-negocios.html>

jotadeveloper.com. (2014). <http://blog.jotadeveloper.com/analisis-de-un-control-de-inventario>.

Kickbill. (2014). *Tus primeros pasos con Zend Framework*. Obtenido de <http://www.kickbill.com/?p=1232>

Lógica Negocio. (s.f.). Obtenido de SISTEMA DE NOTIFICACIÓN EN LÍNEA: <http://trac.ingenian.com>

LUNA, C. (2014).

Madeja. (2014). ZEND. *ZEND*. Obtenido de <http://www.juntadeandalucia.es/servicios/madeja/printpdf/576>

Niccolai, G. (s.f.). *Minimal XML*. Obtenido de <http://mxml.sourceforge.net/>

org. (2014). *Wikipedia*. Obtenido de wikipedia.org.

Pérez, D. (2014). *Webmaster*. Obtenido de Administrador de Sistemas: <https://secure.arkund.com/view/document/19034741-557364-182231/download>

postgresql. (2014). *PostgreSQL*. Obtenido de www.postgresql.org/es/

Postgresqlql. (s.f.). www.Postgresqlql.org.pe/articles.

ProcessMaker. (22 de 03 de 2014). *ProcessMaker*. Recuperado el 12 de 03 de 2015, de ProcessMaker: <http://wiki.processmaker.com/index.php/2.0/Casos>

processmaker. (2016). Obtenido de processmaker: <https://www.processmaker.com/es>

Proyectos Ágiles. (s.f.). Obtenido de Proceso Scrum:

<http://www.proyectosagiles.org>

quest. (2014). <http://www.quest.com>.

robotlegs-mvcs. (2014). Obtenido de <http://www.video-flash.de/>

Rodrigo. (2014). *Zend*. Argentina: Edición: Stephanie Falla Aroche.

RUP, An Introduction. (2014). Obtenido de <http://www.ibm.com/>

SCRUM. (2014). Obtenido de SCRUM: <http://www.buenastareas.com/ensayos>

Sereni, G. (s.f.). *Flex Argentina*. Obtenido de

<http://flexar.blogspot.com/2009/01/04-introducin-mxml.html>

serviciostic. (s.f.). *Tecnologías*. . Obtenido de <http://www.serviciostic.com/las-tic/definicion-de-tic.html>

SO. (2014). Sistema operativo definición. *Sistema operativo definición*. Obtenido de

http://elblogdecindy.bligoo.com/media/users/17/874375/files/173246/sistema_operativo_definicion.pptx

Software libre. (17 de Abril de 2014). Obtenido de

<https://www.gcfaprendelibre.org>

Software, Q. (2014). *Toad Data Modeler*. Obtenido de

<http://www.questsoftware.es/toad-data-modeler/>

Souto, R. (2014). *Guía Zend*. Maestros del Web.

TIBCO. (2014). *TIBCO BUSINESS STUDIO*. Obtenido de

<http://tap.tibco.com/storefront/trialware/tibco-business-studio-community-edition/prod15312.html>

Utn. (2014). Resumen Ejecutivo. *Resumen Ejecutivo*. Obtenido de <http://repositorio.utn.edu.ec/bitstream/123456789/1872/2/04%20ISC%20231%20ResumenEjecutivo-Espanol.pdf>

Visual, I. (s.f.). http://www.islavisual.com/articulos/desarrollo_web/scrum.jpg.

ANEXOS

Anexo A: Manual de usuario se encuentra adjunto en el CD

A1: Manual de Usuario Módulo GPI_INVENTARIOS

Anexo B: Manual Técnico se encuentra adjunto en el CD