

**UNIVERSIDAD NACIONAL DE SAN CRISTÓBAL
DE HUAMANGA**

**FACULTAD DE INGENIERÍA DE MINAS, GEOLOGÍA
Y CIVIL**

**ESCUELA DE FORMACIÓN PROFESIONAL DE
INGENIERÍA DE SISTEMAS**



**“APLICACIÓN WEB PARA ALMACÉN DE
COOPERATIVAS DE AHORRO Y CRÉDITO DEL PERÚ,
2009”**

**TESIS PRESENTADA POR:
BACH. ING. JAVIER SOLIS CASTRO**

**PARA OPTAR EL TÍTULO PROFESIONAL DE:
INGENIERO INFORMÁTICO**

AYACUCHO – PERÚ

2010

AGRADECIMIENTO

Debo expresar mi agradecimiento a todas las personas que integran el sistema cooperativo, quienes me han proporcionado los recursos necesarios creando un ambiente fértil para la investigación. A los miembros de mi Escuela Profesional, a mis compañeros de trabajo, a cada uno de mis amigos, a mis padres quienes contribuyeron con su tiempo, su asesoramiento y sus consejos.

Agradezco también al Ing. Efraín Porrás Flores, por su apoyo en la concepción del presente trabajo de investigación.

DEDICATORIA

A Gaudencio y Filomena mis padres, ejemplos vivos de amor, sacrificio y honestidad.

A mis hermanos y Kathia a quienes les robé tiempo que tuve por dedicarles.

CONTENIDO

Agradecimiento.....	i
Dedicatoria.....	ii
Contenido.....	iii
Resumen.....	v
Introducción.....	iv

CAPITULO I

PLANEAMIENTO DE LA INVESTIGACIÓN

1. Problema de Investigación.....	1
1.1. Fundamentación del Problema.....	1
1.2. Formulación del Problema.....	2
1.3. Objetivos.....	2
1.3.1 Objetivo General.....	2
1.3.2 Objetivos Específicos.....	2
1.4. Justificación.....	3
1.5. Delimitación del Problema.....	3
1.6. Hipótesis de la Investigación.....	3

CAPITULO II

REVISIÓN LITERARIA

2. Antecedentes.....	4
2.1 Marco Teórico.....	6
2.1.1 Software.....	6
2.1.2 ¿Qué es Xp?.....	11
2.1.2.1 Principios Básicos.....	13
2.1.2.2 Roles de Programación Extrema.....	17
2.1.2.3 Fases de Programación Extrema.....	18
2.1.3 Herramientas de Desarrollo	33
2.1.3.1 ¿Porqué usar Java?.....	33
2.1.3.2 Postgresql.....	37
2.1.3.3 Tecnología Web.....	39
2.1.3.4 Arquitectura Web.....	39
2.1.3.5 Mecanismos De Seguridad.....	41

CAPITULO III

METODOLOGÍA DE LA INVESTIGACIÓN

3.1 Tipo de Investigación.....	43
3.2 Diseño de Investigación.....	43
3.3 Técnica de Recolección de Datos.....	44
3.4 Técnica de Procesamiento de Datos.....	44
3.5 Estudio de Viabilidad.....	45

CAPITULO VI RESULTADOS

4.1	Instalación y Configuración de Herramientas	47
4.2	Diagrama físico de redes sobre la cual correrá la aplicación Web para cooperativas del Perú.....	48
4.3	Instalación de Java.....	49
4.4	Instalación de Project-Tomcat.....	50
4.4.1	Configuración de archivos.....	52
4.4.2	Configuración de Apache-Tomcat	53
4.4.3	Instalación de AJP mod_jk	54
4.5	Configuración de Niveles de Seguridad.....	56
4.6	Conectando la Base de Datos.....	58
4.7	Desarrollo de la Aplicación Web.....	61
A.	Fase de Exploración	61
B.	Fase de Planificación	68
C.	Fase de Iteración	76
D.	Fase de Producción	87
E.	Fase de Mantenimiento.....	88
4.8	Diario de Actividades.....	91

CAPITULO V CONCLUSIONES Y RECOMENDACIONES

5.1	Conclusiones.....	93
5.2	Recomendaciones.....	95
5.3	Referencias Bibliográficas.....	96
	Glosario.....	98
	Anexo A: Diseño Lógico de Aplicación Web de Almacén para Coopac's del Perú.....	102
	Anexo B: Diseño y Modelo de Base Datos de Aplicación Web de Almacén para Coopac's del Perú.....	103
	Anexo C: Diagrama de Clases de Aplicación Web.....	111
	Anexo D: Formatos de Requerimiento de Almacén.....	112
	Anexo E: Ranking de las Coopac del Perú 2009.....	117
	Anexo F: Índice de Figuras y Tablas	119

RESUMEN

La crisis financiera internacional trajo cambios del modelo económico de gestión financiera en el Perú. Los intermediadores financieros como bancos, financieras, las cajas municipales, rurales y cooperativas cambiaron sus estrategias, bajando a segmentos sociales no considerados mercados potenciales, implementado: asesoramiento, financiamiento, y tecnologías de información entre otras, como factores importantes de integrar y automatización.

La Federación de Cooperativas de Ahorro y Crédito del Perú(FENACREP), registra: la mayoría de cooperativas carecen de sistemas de información integrados, usan "Sistemas de Procesamiento de Transacciones". Como: OPEN BANK LITE (OBL), SICAT PLUS, SISCOM PLUS. La función de estos sistemas es procesar pagos, cobros, depósitos, etc. Estos sistemas no consideran la administración de almacenes, razón para desarrollar la "Aplicación Web para almacén de cooperativas de ahorro y crédito del Perú, 2009", cuya funcionalidad principal es el control de entradas y salidas de almacén.

La investigación centra el desarrollo de una aplicación Web para automatizar aspectos críticos del flujo de información en el Almacén de entradas/salidas, que controle, resguarde y apoye el proceso de mantener control de las existencias, usando tecnologías de Internet, usando programación extrema, un administrador de base de datos y un lenguaje de programación orientada a objetos.

Palabras Clave: Programación extrema, almacén de cooperativas de ahorro y crédito, lenguaje de programación orientada a objetos.

INTRODUCCIÓN

Hoy las empresas cooperativas agrupan más de 740 millones de miembros, emplean más personas que todas las empresas multinacionales existentes y tienen una importante cuota de participación en muchos sectores económicos básicos.

La falta de aplicaciones colaterales (como el de almacén) a medida que automaticen, controlen y apoyen en la toma de decisiones de costos mínimos a fin de que estas puedan competir frente a bancos, financieras, cajas, etc., que ofrecen el mismo servicio.

El capítulo I "Planteamiento de la Investigación", resume la situación actual del sistema de control de Almacén, identifica y formula los problemas, justifica la investigación a desarrollar y plantea los objetivos generales y específicos.

El capítulo II "Marco Teórico" detalla los temas que apoyarán como sustento teórico a la presente Tesis, además de plantear el sistema de hipótesis y el sistema de variables de la investigación.

En el capítulo IV "Desarrollo de la Investigación" se ejecuta la instalación y configuración necesaria del hardware, se analiza los datos obtenidos de los formatos de almacén, se plantea la solución de control de almacén, se implementa el modelo del negocio y se diseña y desarrolla el Sistema de Información Web que controle los procesos operativos y administrativos de almacén que opere bajo el modelo de negocios planteado.

CAPÍTULO I

PLANEAMIENTO DE LA INVESTIGACIÓN

1. PROBLEMA DE INVESTIGACIÓN

1.1. FUNDAMENTACIÓN DEL PROBLEMA

Desde su nacimiento con los Pioneros de Rochdale, el movimiento cooperativo no ha cesado de crecer en todo el mundo. "... hoy, los miembros de cooperativas en todo el planeta superan los 800 millones..."¹; y en el Perú el gerente general de la Federación Nacional de Cooperativas de Ahorro y Crédito del Perú FENACREP, MBA, Manuel Rabines Ripalda, en la ponencia: "*El rol del Estado en el financiamiento y promoción de otras formas asociativas, Las Coopac en el Perú*" refiere la existencia de 168 Coopac activas a nivel nacional, más de 674 mil asociados, presencia en 21 departamentos del país con 2,270 millones en activos.

Las cooperativas fomentan el ahorro y el financiamiento en sus diversas modalidades, además de brindar otros productos y servicios financieros a sus asociados a través de campañas, promociones, etc. El crecimiento de estas entidades en número y el crecimiento interno debido a la expansión a otras zonas geográficas (agencias u oficinas) tuvo como consecuencia nuevas necesidades y la generación de nuevos problemas de control y esto empeora por la falta de herramientas informáticas que apoyen el crecimiento, tal es el caso del almacén, el mismo con que cuentan todas las cooperativas y que en su mayoría se encuentran controladas ineficazmente de forma manual.

¹ Alianza Cooperativa Internacional para las Américas "*El Cooperativismo en el mundo*" tomado en <<http://www.aciamericas.coop/spip.php?article44>>

Consecuencia de ello es que se han tenido pérdidas sustanciales; mala reputación; disminución de prestigio; etc., por lo que se ha planteado el desarrollo de una aplicación Web, a fin de dar solución a los requerimientos y los problemas inherentes al área del almacén de la institución cooperativa del país.

1.2. FORMULACIÓN DEL PROBLEMA

1.2.1. PROBLEMA PRINCIPAL

¿Cómo automatizar los procesos y la necesidad de información operativa para almacén de las cooperativas de ahorro y crédito del Perú del año 2009?

1.2.2. PROBLEMAS SECUNDARIOS

- a. ¿Cómo implementar la funcionalidad para los procesos e información operativa del área de almacén?
- b. ¿En qué medida la seguridad protege la gestión de procesos e información del área de almacén?

1.3. OBJETIVOS

1.3.1. OBJETIVO GENERAL

Desarrollar la Aplicación Web para automatizar los procesos y la necesidad de información operativa del almacén de las cooperativas de ahorro y crédito del Perú, mediante el diagnóstico situacional, la reunión de los fundamentos teóricos, la selección de las herramientas tecnológicas, el modelado de la aplicación, el diseño de la base de datos y la implementación, con la finalidad de mejorar la ejecución de los procesos y brindar información operativa para el área de almacén.

1.3.2. OBJETIVOS ESPECÍFICOS

- a. Desarrollar la funcionalidad para los procesos e información operativa del área de almacén a fin de prestar apoyo en la administración de materiales.
- b. Evaluar la seguridad para la ejecución de procesos y obtención de

información del área de almacén a fin de brindar protección adecuada.

1.4. JUSTIFICACIÓN

La presente investigación se justifica por la importancia que cumple el concepto de la información para el control de entradas y salidas de requerimientos de material del almacén, pues su funcionamiento trasciende directamente a las áreas operativas de cooperativas.

El Comité técnico de adquisiciones y/o abastecimiento según política de la empresa necesita saber en forma oportuna y adecuada, las necesidades del almacén para cumplir con los requerimientos de las áreas en las cooperativas de ahorro y crédito del Perú.

El presente trabajo de investigación es necesario para mejorar la eficiencia en el control de entradas/salidas; ubicación de materiales; etc., para reducir los costos operativos, beneficiando con ello a los asociados y las cooperativas.

1.5. DELIMITACIÓN DEL PROBLEMA

La importancia del tema, por la potencia y la flexibilidad de la metodologías que hace posible la integración a los sistemas de información vigentes y a los planes estratégicos de negocios, se ha considerado al almacén de las Coopac's del Perú, donde las variables a investigar corresponden al año 2009.

1.6. HIPÓTESIS DE LA INVESTIGACIÓN

Si, automatizamos los procesos y la necesidad de información operativa para almacén de las COOPAC'S del Perú. Entonces logramos la generación de los procesos y contamos con información operativa.

CAPÍTULO II REVISIÓN LITERARIA

2.1 ANTECEDENTES

En el Perú las cooperativas de ahorro y créditos hicieron uso del sistema transaccional Open Bank Lite OBL por más de cinco años, sistema administrado y respaldado por la Federación Nacional de Cooperativas del Perú FENACREP a través de SISNIC SAC, pero las cooperativas debido al uso diario y el crecimiento constante sienten la carencia de módulos en el sistema transaccional y toman la iniciativa de implementar soluciones tecnológicas a estas deficiencias una de ellas es la presente desarrollada para solucionar de manera integrar el problema del control de almacén para las cooperativas de ahorro y crédito del Perú.

*El artículo de Gestiopolis "... En el estudio de la administración moderna el almacén es un medio para lograr economías potenciales y para aumentar utilidades de la empresa. Se piensa de una manera más integral sus funciones a las ventas, compras, control de inventarios, producción y distribución. También se le da al almacén la altura que debe tener dentro de las organizaciones en la selección de su personal, desde el jefe hasta el último puesto del almacén, aunque en nuestro medio el personal que labora en almacenes es mal visto..."*²

De la Tesis de Puig Rodríguez para obtener el título de Ingeniero Informático, de la universidad Oscar Lucero Moya: "Sistema de Almacén central" hace referencia al problema actual en almacén central: ¿Cómo hacer más fácil y ágil el proceso de envío de los vales de salida y los informes de recepción al Departamento Económico de la Universidad

² Tomado de: <<http://www.gestiopolis.com/canales/gerencial/articulos/32/almarsatcl.htm>>

de Holguín y la realización de estas solicitudes al almacén por parte del personal autorizado? De lo anterior podemos dar como Objeto de Investigación el proceso de presentación de los vales de salida y los informes de recepción al Departamento Económico de la Universidad de Holguín y el proceso de la solicitud de materiales al almacén.³

En una publicación de la Escuela Superior de Informática de la Universidad de Castilla "La Mancha" referente la gestión de proyectos de software Junio del 2005, citan: "... el almacén resulta extremadamente ineficiente y esto afecta absolutamente a todas las áreas de la empresa. Algunas de las carencias detectadas son:

- Debido a la distancia y a los problemas de comunicación con el almacén (por teléfono), no es posible conocer, con detalle y en tiempo real, el stock actual del almacén. Como consecuencia, no es posible predecir con exactitud si se podrán dar salida a los pedidos pendientes de servir del área solicitante.
- La ubicación en el almacén de los componentes y de los PC's no se encuentra optimizada. De hecho, actualmente se encuentran mezclados componentes con PC's a lo largo y a lo ancho del almacén, con las consiguientes ineficiencias que se producen tanto en el proceso de recepción de mercancías como en su posterior envío..."⁴

Otra publicación de la Escuela Superior de Informática de la Universidad de Castilla "La Mancha" referente a "Un caso práctico de Computación Ubicua en el almacén". Noviembre del 2004, El proyecto Sistema para Localización en Almacenes de Stocks SIGAS tenía que resolver los requerimientos y expectativas que se plantean en una organización a la

³ Yoan D. Puig "Sistema de Almacén central" Tesis para optar el título de Ingeniero Informático Univ. de Holguín Oscar Lucero Moya -CUBA Pp.06.

⁴ Tomado de: alarcos.inf-cr.uclm.es/doc/pgsi/doc/teo/8/pgsi-t8ej.pdf

hora de aplicar la computación. En el caso concreto de este proyecto fueron:

- Control de stocks: Acceso a la información referente a todos los artículos de un almacén, con sus diferentes características de precios, ubicaciones y existencias.
- Monitorización de los pedidos del día: Permitiendo llevar un seguimiento del estado de los pedidos en cada jornada de trabajo...⁵

En el REGLAMENTO ADMINISTRACIÓN DEPÓSITO DE ALMACENES Y SUBALMACENES de la universidad San Francisco Javier de Chuquisaca BOLIVIA, empresa de servicios donde se desarrolló un software y se reglamentó la administración dada la importancia del Almacén donde hace referencia *"El Encargado del Depósito de Almacenes en la Administración Central y los Auxiliares de los Sub-almacenes en las Unidades Desconcentradas utilizarán el método contable de valoración de inventarios PEPS Primeros en Entrar Primeros en Salir, significa que el costo de los bienes consumidos durante el periodo refleja el costo de productos comprados anteriormente en el tiempo, mientras que los productos mantenidos en el inventario reflejan costes más recientes."*⁶

2.2 MARCO TEÓRICO

2.2.1 Software

Palabra proveniente del idioma inglés y aceptado por la Real Academia Española, que define que el software es el *"Conjunto de programas, instrucciones y reglas informáticas para ejecutar ciertas tareas en una computadora"*

⁵ Tomado de: <<http://chico.inf-cr.uclm.es/cache/>>

⁶ Tomado de: <<http://www.usfx.info/edif/orgmet/administraciondealmacenes.pdf>>

Según el estándar 729 del Institute of Electrical and Electronics Engineers - IEEE, software "Es el conjunto de los programas de cómputo, procedimientos, reglas, documentación y datos asociados que forman parte de las operaciones de un sistema de computación".⁷

Bajo esta definición, el concepto de software va más allá de los programas de cómputo en sus distintos estados: código fuente, binario o ejecutable; también su documentación, datos a procesar e información de usuario es parte del software: es decir, abarca todo lo intangible, todo lo "no físico" relacionado.

El término *software* fue usado por primera vez en este sentido por John W. Tukey en 1957. En las ciencias de la computación y la ingeniería de software, el software es toda la información procesada por los sistemas informáticos: programas y datos. El concepto de leer diferentes secuencias de instrucciones desde la memoria de un dispositivo para controlar los cálculos fue introducido por Charles Babbage como parte de su máquina diferencial. La teoría que forma la base de la mayor parte del software moderno fue propuesta por vez primera por Alan Turing en su ensayo de 1936, "*Los números computables*", con una aplicación al problema de decisión.⁸

a. Funcionamiento del software

Según la norma ISO/IEC TR 9126-2:2003, la métrica externa proporciona a los usuarios, evaluadores, verificadores y desarrolladores, el beneficio de evaluar la calidad del software durante las pruebas o el funcionamiento.⁹

⁷ ISO/IEC TR 9126-3:2003. Software engineering. Product quality Part 3.

⁸ Cesar Liza A. Modelando con UML, principios y aplicaciones. Ed. RJ SRL. Perú. 2001.

⁹ Matsukawa M. Sergio. Análisis y Diseño Orientado a Objetos con UML y Rational Rose. Edit. Macro, Lima 2004.

b. Funcionalidad del software

Según Fernanda Scalone, la funcionalidad es la extensión o cantidad de posibilidades provistas por un sistema. Uno de los problemas más difíciles que encuentra un líder de proyecto es saber cuánta funcionalidad es suficiente. La presión por más funciones, conocida como "adornitis" (requerimientos desmedidos), siempre estará allí. Sus consecuencias son malas para proyectos internos y peor para productos comerciales. La "adornitis" es la combinación de dos problemas, uno más difícil que otro.¹⁰

El problema fácil es la pérdida de consistencia que puede resultar por el aumento inmoderado de funciones y propiedades, afectando la facilidad de uso. Lo que para una persona puede ser un rasgo insignificante puede ser una característica indispensable para otra. La solución aquí es trabajar una y otra vez en la consistencia del sistema en general, tratando de establecer cada cosa en un molde global. Un buen sistema de software está basado en un conjunto reducido de ideas poderosas; aún si contiene muchas funciones especializadas, estas son derivadas como consecuencia de dichos conceptos básicos.

El problema más difícil es evitar enfocarse demasiado en adornos que hagan olvidarse de otros factores de calidad. Muchos proyectos en su ferviente carrera por agregar funciones, el sistema pierde todo indicio de calidad. Al final, cuando se intenta hacer las cosas bien, el proyecto está en un escenario típico de noches y fines de semana de arduo trabajo que solo desgasta al equipo de desarrollo y no aumenta su productividad sino que la disminuye.

Con la presión de los usuarios o clientes, se ven obligados a liberar el sistema de forma prematura, el resultado neto es la pérdida de

¹⁰ Matsukawa Maeda, Sergio "Análisis y Diseño Orientado a Objetos con UML y Rational Rose" pp.137.

credibilidad y una reputación de retrasos. La solución aquí es usar las técnicas del método orientado a objetos que acentúan la calidad, para mantener constante el nivel de calidad en todos los aspectos del proyecto, se avanza en nuevas funciones hasta que se tienen maduras las anteriores.

Las sub-características de la "Funcionalidad" son: Aptitud, Precisión, Interoperatividad, Conformidad, Seguridad y Trazabilidad

c. Usabilidad del software

Según Fernanda Scalone, la facilidad de uso o usabilidad consiste en la simplicidad con que la gente puede aprender a usar el software y aplicarlo para resolver problemas. También incluye la facilidad de instalación, operación y monitoreo. La definición acentúa los diferentes niveles de experiencia de los usuarios potenciales. Este requerimiento representa uno de los mayores retos para los diseñadores de software interesados en facilidad de uso; es decir, cómo proveer orientación a los usuarios novatos y además al mismo tiempo, cómo no aburrir a los usuarios expertos.¹¹

Esta es una de las áreas en donde el método orientado a objetos es particularmente productivo, muchas técnicas orientadas a objetos, que de entrada parecen atender solo aspectos de diseño e implementación, también aportan ideas nuevas y poderosas en el desarrollo de interfaces del usuario. Los buenos diseñadores de interfaz del usuario siguen un principio sabio, hacer el menor número de suposiciones acerca del usuario. Cuando se diseña un sistema interactivo, se debe considerar que los usuarios podrán leer, mover un mouse, hacer "clic", teclear (espacio); no más que eso.

¹¹ Análisis y Diseño Orientado a Objetos con UML y Rational Rose pp.138

Según la norma ISO/IEC TR 9126-2:2003, la Ingeniería de Facilidad de Uso reúne una serie de técnicas con un enfoque centrado en el usuario que nos permite definir y manejar con un grado de precisión adecuado los niveles de facilidad de uso deseados para un sistema de software que se quiera desarrollar.

Dedicar recursos a mejorar la facilidad de uso de un sistema en el desarrollo de software no es un lujo. Actualmente es una necesidad debido a un mercado competitivo donde se demanda un nivel de facilidad de uso similar al existente en los productos actuales, que es muy difícil de conseguir si no se aplican técnicas para desarrollar software con un enfoque más centrado en el usuario y menos centrado en el desarrollador.¹²

Desarrollar un software sin preocuparse de su nivel de facilidad de uso constituye, hoy en día, un riesgo demasiado grande para las organizaciones que desarrollan software. La facilidad de uso es una medida que facilita a los usuarios específicos la realización de ciertas tareas, tales como: 1) aprendizaje del manejo del sistema, 2) disminución de la probabilidad de cometer errores, 3) efectividad y eficiencia en la realización de dichas tareas, y 4) satisfacción en el uso del sistema.

La facilidad de uso del sistema no es un atributo inherente al software, no puede especificarse independientemente del entorno de uso y de los usuarios que vayan a utilizar el sistema. La facilidad de uso no puede definirse como un atributo simple de un sistema, pues implica distintos aspectos dependiendo del tipo de sistema a construir. La facilidad de uso tiene los siguientes atributos: 1) Facilidad de aprendizaje, 2) Eficiencia, 3) Recuerdo en el tiempo, 4) Tasa de errores y 5) Satisfacción.

¹² ISO/IEC TR 9126-3:2003. Software engineering. Product quality part. 3.

El proceso de facilidad de uso consiste en un ciclo de diseño-evaluación-rediseño con un análisis previo donde se definen los niveles de facilidad de uso que se desean alcanzar.

Este proceso asegura que el producto de software obtenido está cercano al óptimo en cuanto a facilidad de uso, entendiendo por óptimo los niveles deseados de facilidad de uso.

Este proceso ayuda al desarrollador a conseguir las respuestas a las preguntas sobre los usuarios y las tareas que desempeñan en la fase de análisis, y da una base sobre la que se realiza el diseño de la interacción en la fase de diseño.

La medición de facilidad de uso utiliza 5 escalas diferentes y no incluye utilidad. Ellas son: 1) Tiempo de la tarea, 2) Errores, 3) Aprendizaje, 4) Aprender de nuevo, 5) Satisfacción y 6) Logro de la meta.

Las sub características de la "Facilidad de Uso" son: Comprensibilidad, Facilidad de aprendizaje, Operatividad, Explicitud, Adaptabilidad al usuario, Atractividad, Claridad, Facilidad de ayuda y Amigable al usuario.

2.2.2 ¿QUÉ ES XP?

A continuación se hizo un resumen de la publicación hecha el Departamento de Procesos, Información, Contabilidad y Finanzas de la Universidad del Valle. Sobre Programación Extrema:

a. Metodología ágil

Las metodologías ágiles (como por ejemplo XP, SCRUM, DSDM, Crystal, etc...) forman parte del movimiento de desarrollo ágil de software, que se basan en la adaptabilidad de cualquier cambio como medio para aumentar las posibilidades de éxito de un proyecto.

De forma que una metodología ágil es la que tiene como principios que:

- Los individuos y sus interacciones son más importantes que los procesos y las herramientas.
- El software que funciona es más importante que la documentación exhaustiva.
- La colaboración con el cliente en lugar de la negociación de contratos.
- La respuesta delante del cambio en lugar de seguir un plan cerrado.

Se puede decir que, este movimiento empezó a existir a partir de febrero de 2001, cuando se reunieron los representantes de cada una de estas metodologías y terminaron poniendo en común sus ideas en una declaración conjunta.¹³

b. Definición

La programación extrema es una metodología de desarrollo ligera (o ágil) basada en una serie de valores y de prácticas de buenas maneras que persigue el objetivo de aumentar la productividad a la hora de desarrollar programas.

Este modelo de programación se basa en una serie de metodologías de desarrollo de software en la que se da prioridad a los trabajos que dan un resultado directo y que reducen la burocracia que hay al rededor de la programación. Cuya característica principal es que sus ingredientes son conocidos desde el principio de la informática. Los autores de XP han seleccionado aquellos que han considerado mejores y han profundizado en sus relaciones y en como se refuerzan los unos con los otros. El resultado de esta selección ha sido esta metodología única y compacta. Por esto, aunque no está basada en

¹³ Tomado de: Revista digital Universidad del Valle - Colombia del Departamento de Procesos, Información, Contabilidad y Finanzas.

<<http://eisc.univalle.edu.co/materias/www/material/lecturas/xp.pdf>>

principios nuevos, sí que el resultado es una nueva manera de ver el desarrollo de software.

El objetivo que se perseguía en el momento de crear esta metodología era la búsqueda de un método que hiciera que los desarrollos fueran más sencillos. Aplicando el sentido común.

2.2.2.1 PRINCIPIOS BÁSICOS

La Programación Extrema se basa en 12 principios básicos agrupados en cuatro categorías:

a. Retroalimentación a escala fina:

- **El principio de pruebas:** se tiene que establecer un período de pruebas de aceptación del programa (llamado también *período de caja negra*) donde se definirán las entradas al sistema y los resultados esperados de estas entradas. Es muy recomendable automatizar estas pruebas para poder hacer varias simulaciones del sistema en funcionamiento. Para hacer estas simulaciones automatizadas, se pueden utilizar *Ambientes de Prueba* (Unit testing frameworks). Un buen ejemplo de un ambiente de prueba es el JUnit para Java (www.junit.org/index.htm)
- **Proceso de planificación:** en esta fase, el usuario tendrá que escribir sus necesidades, definiendo las actividades que realizará el sistema. Se creará un documento llamado Historias del usuario (User Stories). Entre 20 y 80 historias (todo dependiendo de la complejidad del problema) se consideran suficientes para formar el llamado Plan de Liberación, el cual define de forma específica los tiempos de entrega de la aplicación para recibir retroalimentación por parte del usuario. Por regla general, cada una de las Historias del usuario suelen necesitar de una a tres semanas de desarrollo.

Son muy importantes y tienen que ser una constante las reuniones periódicas durante esta fase de planificación. Estas pueden ser a diario, con todo el equipo de desarrollo para identificar problemas, proponer soluciones y señalar aquellos puntos a los que se les ha de dar más importancia por su dificultad o por su punto crítico.

- **El cliente en el sitio:** se le dará poder para determinar los requerimientos, definir la funcionalidad, señalar las prioridades y responder las preguntas de los programadores. Esta fuerte interacción cara a cara con el programador disminuye el tiempo de comunicación y la cantidad de documentación, junto con los altos costes de su creación y mantenimiento. Este representante del cliente estará con el equipo de trabajo durante toda la realización del proyecto.
- **Programación en parejas:** uno de los principios más radicales y en el que la mayoría de gerentes de desarrollo pone sus dudas. Requiere que todos los programadores XP escriban su código en parejas, compartiendo una sola máquina. De acuerdo con los experimentos, este principio puede producir aplicaciones más buenas, de manera consistente, a iguales o menores costes.

b. Proceso continuo en lugar de por lotes

- **Integración continua:** Permite al equipo hacer un rápido progreso implementando las nuevas características del software. En lugar de crear *builds* (o versiones) estables de acuerdo a un cronograma establecido, los equipos de programadores XP pueden reunir su código y reconstruir el sistema varias veces al día. Esto reduce los problemas de integración comunes en proyectos largos y estilo cascada.

- **Refactorización:** Permite a los equipos de programadores XP mejorar el diseño del sistema a través de todo el proceso de desarrollo. Los programadores evalúan continuamente el diseño y recodificar lo necesario. La finalidad es mantener un sistema enfocado a proveer el valor de negocio mediante la minimización del código duplicado y/o ineficiente.
- **Entregas pequeñas:** Colocan un sistema sencillo en producción rápidamente que se actualiza de forma rápida y constante permitiendo que el verdadero valor de negocio del producto sea evaluado en un ambiente real. Estas entregas no pueden pasar las 2 o 3 semanas como máximo.

c. Entendimiento compartido

- **Diseño simple:** Se basa en la filosofía de que el mayor valor de negocio es entregado por el programa más sencillo que cumpla los requerimientos. Simple Design se enfoca en proporcionar un sistema que cubra las necesidades inmediatas del cliente, ni más ni menos.
Este proceso permite eliminar redundancias y rejuvenecer los diseños obsoletos de forma sencilla.
- **Metáfora:** Desarrollada por los programadores al inicio del proyecto, define una historia de como funciona el sistema completo. XP estimula historias, que son breves descripciones de un trabajo de un sistema en lugar de los tradicionales diagramas y modelos UML (Unified Modeling Language). La metáfora expresa la visión evolutiva del proyecto que define el alcance y propósito del sistema.
Las tarjetas CRC (Clase, Responsabilidad y Colaboración) también ayudarán al equipo a definir actividades durante el diseño del sistema. Cada tarjeta representa una clase en la programación orientada a objetos y define sus

responsabilidades (lo que ha de hacer) y las colaboraciones con las otras clases (cómo se comunica con ellas).

- **Propiedad colectiva del código:** Un código con propiedad compartida. Nadie es el propietario de nada, todos son el propietario de todo. Este método difiere en mucho a los métodos tradicionales en los que un simple programador posee un conjunto de código. Los defensores de XP argumentan que mientras haya más gente trabajando en una pieza, menos errores aparecerán.
- **Estándar de codificación:** Define la propiedad del código compartido así como las reglas para escribir y documentar el código y la comunicación entre diferentes piezas de código desarrolladas por diferentes equipos. Los programadores las han de seguir de tal manera que el código en el sistema se vea como si hubiera estado escrito por una sola persona.

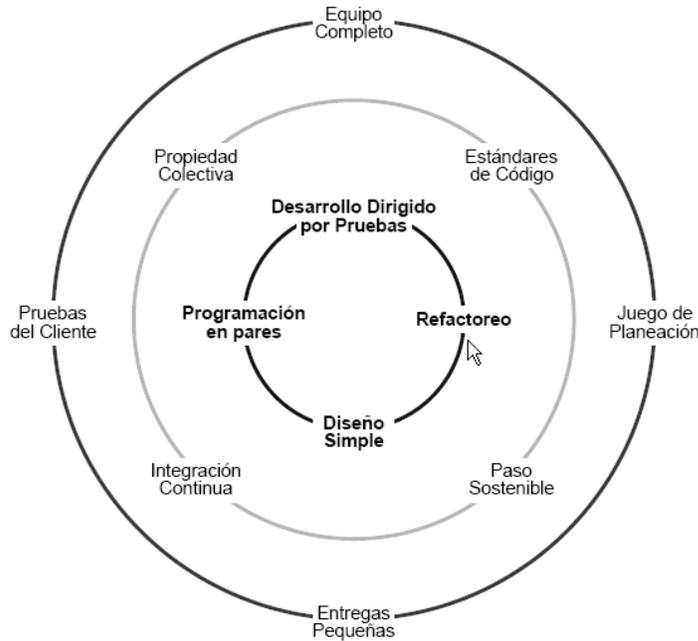
d. Bienestar del programador

- **La semana de 40 horas:** La programación extrema sostiene que los programadores cansados escriben código de menor cualidad. Minimizar las horas extras y mantener los programadores frescos, generará código de mayor calidad. Como dice Beck, está bien trabajar tiempos extra cuando es necesario, pero no se ha de hacer durante dos semanas seguidas.

Programación Extrema:

Figura N° 01: Las prácticas de Programación Extrema.

Prácticas XP



Fuente: Figura extraída de la Web de Ron Jeffries
(www.xprogramming.com)

2.2.2.2 ROLES DE PROGRAMACIÓN EXTREMA

Los roles que deben tener los miembros del equipo de acuerdo con la propuesta original de Kent Beck son:

- **Programador.** El programador escribe las pruebas unitarias y produce el código del sistema.
- **Cliente.** Escribe las historias de usuario y las pruebas funcionales para validar su implementación. Además, asigna la prioridad a las historias de usuario y decide cuáles se implementan en cada iteración centrándose en aportar mayor valor al negocio.

- **Encargado de pruebas (Tester).** Ayuda al cliente a escribir las pruebas funcionales. Ejecuta las pruebas regularmente, difunde los resultados en el equipo y es responsable de las herramientas de soporte para pruebas.
- **Encargado de seguimiento (Tracker).** Proporciona realimentación al equipo. Verifica el grado de acierto entre las estimaciones realizadas y el tiempo real dedicado, para mejorar futuras estimaciones. Realiza el seguimiento del progreso de cada iteración.
- **Entrenador (Coach).** Es responsable del proceso global. Debe proveer guías al equipo de forma que se apliquen las prácticas XP y se siga el proceso correctamente.
- **Gestor (Big boss).** Es el vínculo entre clientes y programadores, ayuda a que el equipo trabaje efectivamente creando las condiciones adecuadas. Su labor esencial es de coordinación.

2.2.2.3 FASES DE PROGRAMACIÓN EXTREMA

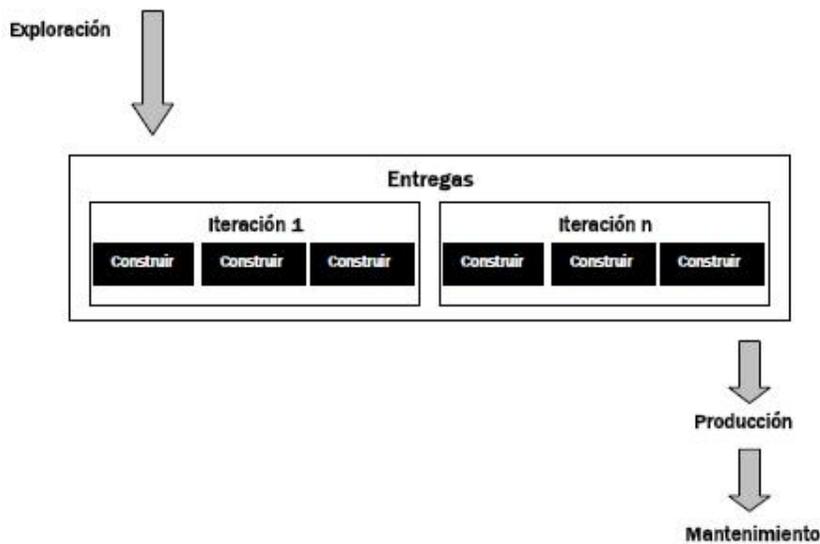
Tabla N° 01: Fases de la Programación Extrema

Exploración	Inicio de proyecto, exigencias de usuario de alto nivel, técnicos de prototipos.
Planificación	Prioridades de trabajo, interrupción en entregas y primero planifica.
Iteraciones	Pruebas y desarrollo del sistema. Incluye la planificación de iteración donde la interrupción de trabajo de bajo nivel ocurre. Los usuarios finales pueden trabajar aquí en la refinación del interfaz, la aseguración de la utilidad.
Producción	Despliegue de software en el entorno de producción del cliente.
Mantenimiento	Mantenimiento permanente, parches y mejoras.

FUENTE: Información de "Aprende XP en 24 horas" de L. Cancho C.

Hablando en términos generales, los proyectos PROGRAMACIÓN EXTREMA como este implican el derribo la visión de producto del cliente en liberaciones y luego el segmentar aquellas liberaciones en iteraciones. La planificación es vista como un proceso evolutivo de desarrollo refinado y afinado sobre la vida del proyecto. La Figura 02 muestra como trabaja esto.

Figura N° 02: En la Programación Extrema las entregas están hechas de iteraciones.

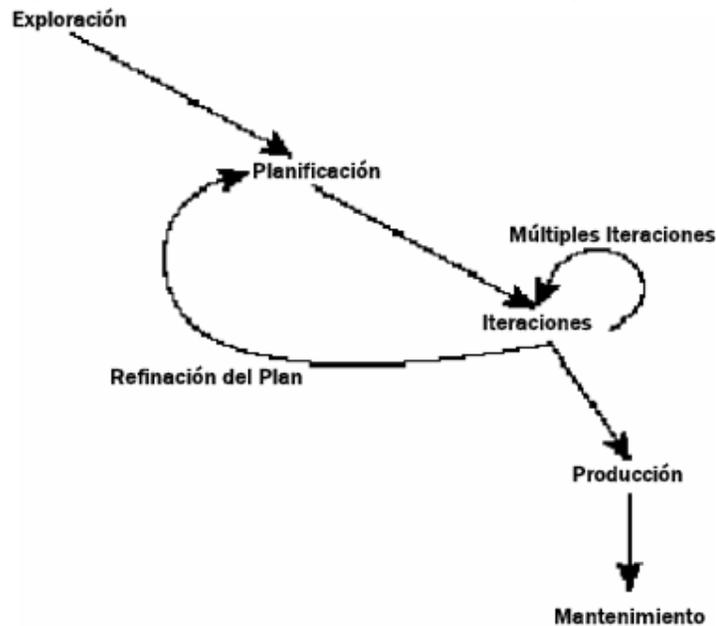


Durante la iteración hay múltiples construcciones, que ocurre con regularidad durante el proceso de desarrollo y como el equipo integra el nuevo código. El número de construcciones es dependiente del tipo de tecnología usada y el estilo de desarrollo que el equipo adopta.

¿Que es la versión? Consiste en que el trabajo del equipo de desarrollo del software terminado va a las manos del cliente.

¿Qué es la iteración? Es el que tiene un enfoque interno permitiendo al cliente local y los desarrolladores medir y ajustar el progreso del software.

Figura N° 03: El ciclo de vida del proyecto programación Extrema



Fuente: "Aprende XP en 24 horas" de L. Cancho C.

Después de la exploración, el equipo de desarrollo inicia un ciclo de planificación de iteraciones y continúa hasta que el software se considerará completo y listo para la producción. Producción o la liberación (entrega o versión) es el software completo que se mueve en un entorno de trabajo o vivo. La producción es importante porque significa que la empresa puede empezar a obtener el valor real de su nuevo software. Otra faceta de la producción del software es que los costos asociados con el fracaso son mucho mayores, lo que explica que la mayor parte de los clientes tengan cuidado para cubrir la prueba final de aceptación.

a. Explorando los Requisitos del Usuario

La Exploración se produce el descubrimiento de la visión y la misión del sistema. En esta fase del proyecto, el cliente se encarga de la elaboración de una declaración de visión. Esta declaración de visión debe estar alrededor de 20 a 30 palabras y lleva en al desarrollo de la metáfora del sistema. Los clientes y desarrolladores trabajan

juntos en esta fase, explorando opciones técnicas, la definición de requisitos, y completando una lista de historias de usuario.

a.1 Escritura de Historias de Usuario

En esta fase los desarrolladores estiman el tiempo de para ponerlos en práctica. Hay un entendimiento implícito que cualquier estimación en esta etapa se da a grandes rasgos y será refinada en el nivel de la iteración de acuerdo al progreso del proyecto. En algunos casos, las soluciones del esfuerzo por el desarrollo de prototipos de código de desarrolladores (pasan el tiempo realizando prototipos e investigar cuestiones técnicas) para conseguir una cifra de valoración más exacta. Los especialistas de tecnología o consultores podrían estar acostumbrados a validar enfoques de alto nivel, y ellos también pueden ayudar en este pico (esfuerzo de desarrollo de prototipos de código). El concepto de pico es una simple pero poderosa herramienta para ayudar al equipo de desarrolladores tanto a entender el esfuerzo y el típico enfoque del desarrollo que tendrá. A veces el equipo técnico se separará en un número de pares, cada uno de cual investigará una opción técnica en particular.

Al final de este pico el código es tirado, dejando el equipo de desarrollo con un entendimiento común del camino hacia adelante. La longitud de este pico puede ser en materia de días dependiendo de la magnitud del proyecto. Como el trabajo de los desarrolladores avanza pueden "bajar las herramientas" durante el desarrollo en los picos de los problemas técnicos mas difíciles, se trata de una excepción más que sobre una base regular.

a.2 Estimación y Descubrimiento

La fase de exploración es principalmente sobre el descubrimiento. Los clientes descubren lo que ellos quieren, y los desarrolladores descubren su enfoque probable y sus estimaciones.

Como los desarrolladores dan soluciones de picos, también realizan la estimación y el seguimiento. Esto funciona simplemente por desarrolladores que estiman el tiempo para completar sus tareas de pico, ellos ejecutan el pico y lo comparan a su estimación. De esta manera la fase de exploración tiene el beneficio añadido de permitir que el equipo pueda afinar sus habilidades de estimación. Estas habilidades serán vitales cuando se inicia la planificación de la iteración.

La duración de la exploración depende de la naturaleza del proyecto y las tecnologías en juego. La necesidad de pasar tiempo en los picos se reduce si el equipo de desarrollo utiliza herramientas familiares y técnicas. La longitud total de la fase de continuación podrían ser un par de semanas o meses, dependiendo del alcance y requisitos.

El resultado de exploración será la metáfora de sistema y la primera lista de historias de usuario. El equipo ahora conoce aproximadamente lo que se requiere, como lo harán y hacia dónde van. ¡Esto probablemente se cambia como progresos de trabajo, que es la razón por al qué usted ha seleccionado PROGRAMACIÓN EXTREMA!.

Tabla N° 02: Formato de Historia de Usuario.

Historia de Usuario	
Número:	Usuario:
Nombre historia:	
Prioridad en negocio: (Alta / Media / Baja)	Riesgo en desarrollo: (Alta / Media / Baja)
Puntos estimados:	Iteración asignada:
Programador responsable:	
Descripción:	
Observaciones:	

Fuente: Obtenido del Ejemplo de desarrollo de Software utilizando la metodología XP, DSIC, Universidad Politécnica de Valencia ,2006.

b. Creación del Plan de Proyecto

La planificación es una breve fase en la que los clientes y los desarrolladores se ponen de acuerdo sobre las características de la primera versión. Las características (historias de usuario) son entregados de una manera que hace tanto sentido del negocios como el técnico. La Tabla 4.2 se enumeran las responsabilidades de los clientes y desarrolladores durante la etapa de planificación.

Tabla 03: Responsabilidades Durante la etapa de Planificación.

Fase	Descripción
Clientes	Define usuario historias Define que historias tienen valor para el negocio. Define que historias se construirán para su versión.
Desarrolladores	Realizan estimaciones de cuanto tomará cada historia para construirlo. Advierte al cliente de los riesgos técnicos. Mide el progreso de equipo de desarrollo.

Fuente: Información de "Aprende XP en 24 horas" de L. Cancho C.

La Planificación implica dar y recibir entre el cliente y el desarrollador, pero no hay un equilibrio entre la viabilidad técnica y el valor del negocio. Los desarrolladores pueden generar sus propias historias de usuario "sin valor comercial" para las obras de infraestructura que soporta el sistema central. Un ejemplo de ello sería el establecimiento de la prueba y crear ambientes. Esto garantiza que todos los trabajos que el equipo está haciendo sean estimados y seguidos.

La práctica PROGRAMACIÓN EXTREMA "La Planificación del Juego" es la principal herramienta utilizada para la planificación. El Juego de Planificación implica a clientes y programadores que trabajan por historias de usuario estimando y priorizando.

El Juego de Planificación es un acontecimiento que es informal, de peso ligero y pragmático. Las necesidades de los usuarios están escritas en tarjetas de índice y son manejados/discutidos durante el

período de sesiones. PROGRAMACIÓN EXTREMA alienta el empleo de tarjetas de índice tanto para las historias de usuario como para planificación. El empleo de tarjetas de índice no es obligatorio y el equipo podría decidir usar otros instrumentos de colaboración a base de software.

La salida de esta fase es un programa de liberación, no una estructura de desglose de trabajo. Los clientes y desarrolladores ahora tienen una comprensión común de lo que contendrá la liberación. El siguiente paso es planificar las iteraciones individuales.

c. División de las Versiones en Iteraciones

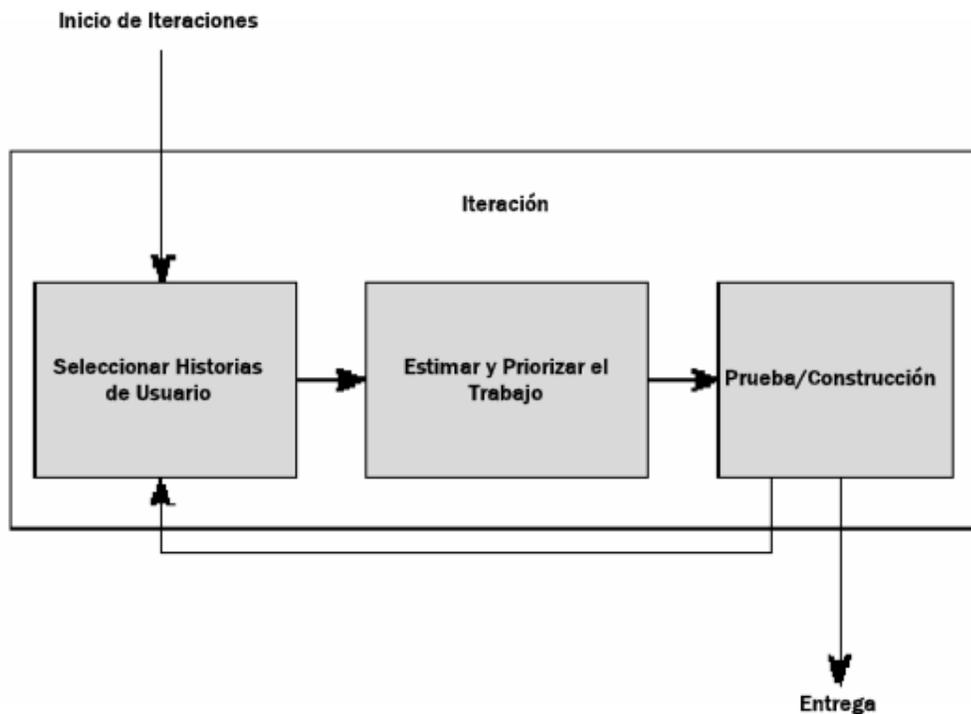
La liberación es reducida a un número de iteraciones con una longitud típica de una a cuatro semanas. La iteración inicial se enfoca en la arquitectura de componentes necesarios proporcionando un sistema de referencia. Esto puede ser llamado “Un Valor Cero del Negocio” porque la iteración de las historias de usuario es más probable que sean puramente técnicos y no tienen ninguna ventaja directa al negocio del cliente.

Posteriores iteraciones deben entregar el valor de negocio que sigue creciendo con el tiempo. El cliente elige las historias de usuarios que ellos quieren, y los desarrolladores deben completar la construcción. El “cliente puede cambiar su mente” este enfoque puede resultar frustrante para los desarrolladores. El único consuelo es que el encajonamiento es el tiempo de iteración. El Tiempo de encajonamiento significa que el alcance no sólo cambia las historias de usuario que están siendo desarrollados. El cliente no es capaz de cambiar las historias de usuario durante la iteración. El punto es que la duración del ciclo es tan corta que el riesgo o la posibilidad de cambio es menor.

c.1 El Proceso de Iteración

En la Iteración es donde el verdadero trabajo de desarrollo ocurre. En el principio de la iteración se llama a una reunión de planificación para permitir al equipo repartir el trabajo en tareas de programación. El cliente selecciona las historias de usuario del plan de liberación (entrega) total, poniendo prioridad sobre aquellos con mayor valor de negocio. Los errores en las pruebas de aceptación de la iteración anterior también serán seleccionados. La figura N° 03 muestra el proceso dentro de una iteración.

Figura N° 04: Proceso de una iteración del ciclo de vida XP.



Fuente: "Aprende XP en 24 horas"

El cliente puede optar por realizar pruebas de utilidad como para el progreso de los trabajos, que es particularmente importante en los casos en que el producto de software se enfrenta a los consumidores. En el caso de un sitio Web público la interfaz debe satisfacer al usuario en los distintos niveles de sus habilidades y experiencia. Dejando la

utilidad hasta que el punto de liberación (versión) no sea tan efectivo como la construcción de la utilidad en el desarrollo continúa. No existen reglas muy estrictas sobre como gestionar el desarrollo de su interfaz trabajando en pequeñas iteraciones. Se podría decidir que es justificado que un grupo de usuario este a tiempo completo. Se advierte, sin embargo, que ellos rápidamente se convertirán en usuarios expertos y esto podría reducir su valor como "usuarios genéricos".

Las historias de usuarios generalmente son demasiado extensas como para la ejecución de tareas de programación, por lo que el siguiente paso es desglosar las tareas necesarias de desarrollo para apoyar la historia. Los desarrolladores trabajan juntos para investigar las tareas que se tendrán que ejecutar. Las tareas están escritas en la parte posterior de la historia, en diferentes tarjetas de tarea, o en algún otro mecanismo de seguimiento.

La duración típica ideal para la programación de una tarea esta alrededor de uno a tres días. Lo ideal con los días de programación, son cuánto tiempo le tomaría para completar la tarea, si no hay distracciones. Las tareas más extensas y de mayor larga duración deben subdividirse en unidades más pequeñas.

Los desarrolladores se inscriben para las tareas y luego estiman cuanto tiempo sus propias tareas tomarán para completarse. Al igual que la estimación en general la mejor practica es para los desarrolladores quien estime el trabajo sea quien realmente hace el trabajo.

c.2 Medición de Rendimiento

En PROGRAMACIÓN EXTREMA las medidas de rendimiento (la cantidad de trabajo que el equipo está completando en el tiempo) por la velocidad. La realidad es que las estimaciones serán inexactas durante iteración cero. Esto se debe a que el equipo es nuevo, la claridad alrededor de los requisitos podrían fallar y los juegos de

herramientas están siendo acostado. El cliente y los desarrolladores deberían reconocer esto al principio. La clave es hacer un seguimiento real de la salida en contra de nuestras estimaciones y a continuación, podemos empezar el perfeccionamiento de la exactitud de los cálculos futuros.

d. Desarrollo y Pruebas

Los Desarrolladores comienzan el trabajo sobre una nueva tarea escribiendo la primera prueba y añadiéndolo a su marco de pruebas. La programación en pares sigue con cada compañero turnándose a la unidad para “manejar” el teclado de tiempo en tiempo. Como las clases o componentes son completados ellos son añadidos a la máquina de integración. La máquina de integración es un terminal de trabajo que ha sido designado únicamente para la construcción o el trabajo de integración. Durante la iteración, los desarrolladores utilizan la integración continua apoyada por la construcción automatizada de marcos.

Al final de cada iteración, los clientes realizarán las pruebas de aceptación de lo que han escrito. Cualquier historia de usuario que no tenga la aceptación se determinará durante la próxima iteración.

d.1 Programación en Pares.

- **Programar las tareas de ingeniería por cada par de programadores.**

El programador implementa una tarea de ingeniería (la unidad más pequeña de la programación) lo cual será integrado con el resto del sistema mediante la refactorización.

- **Programar para reducir la probabilidad de errores.**

El desarrollo del sistema es impulsado por las pruebas, una vez que

corran todos los casos de prueba para el código implementado se estará añadiendo valor al sistema.

- **Codificar primero la prueba unitaria automatizada de cada método o clase a la vez.**

Pregunte primero ¿Cuáles son los casos de prueba para esta tarea? Una vez definido los casos de prueba verifique la estructura de algunos de los actuales casos de prueba que podrán ser aplicados para un determinado método o clase, una vez refactorizando el código de la clase estará sometido necesariamente a una serie de casos de prueba, lo cual nos garantiza si es posible integrar la clase al resto del sistema.

d.2 Pruebas Unitarias

La creación de las pruebas unitarias ayuda al desarrollador, para considerar realmente lo que hay que hacer. Los requisitos se establecen firmemente exactos por medio de ensayos. No puede haber ningún malentendido, una especificación escrita en forma de código ejecutable. También se tiene información inmediata mientras se trabaja. A menudo no está claro cuando un desarrollador ha acabado la funcionalidad necesaria. El ámbito de aplicación puede ocurrir como ampliaciones y condiciones de error se consideran. Si vamos a crear nuestras primeras pruebas unitarias entonces sabemos cuándo se realizan, el plazo de todas las pruebas unitarias.

También hay un beneficio para el diseño del sistema. A menudo son muy difíciles las pruebas unitarias de algunos sistemas de software. En estos sistemas primeramente se construye el código y segundo las pruebas, a menudo por un equipo totalmente diferente. Mediante la creación de su primera pruebas su diseño se verá influido por un deseo de poner a prueba todo lo de valor a su cliente. Su diseño reflejará este por ser más fácil de probar. Puede crear una prueba para definir algunos pequeños

aspectos del problema en la mano. Luego de crear el código más simple hay que hacer que pase la prueba. Luego de crear una segunda prueba. Ahora se agrega el código que acaba de ser creada para que esta nueva prueba pase, No hasta que haya una tercera prueba. Usted continuará hasta que no quede nada para poner a prueba. El código creado es simple y conciso, sólo la aplicación de las características que quería. Otros desarrolladores pueden ver la forma de utilizar este nuevo código de la navegación por las pruebas.

d.3 Pruebas de Aceptación

Las pruebas de aceptación son creadas a partir de las historias de usuario que son seleccionados en la reunión de planificación de iteración y se traduce en pruebas de aceptación. El cliente especifica los escenarios para pruebas cuando una historia de usuario ha sido aplicada correctamente. Una historia puede tener una o varias pruebas de aceptación.

Cada prueba de aceptación representa algunos de los resultados esperados del sistema. Los clientes son los responsables de verificar la exactitud de las pruebas de aceptación y la revisión de los resultados de las pruebas para decidir cuales son de alta prioridad.

Una historia de usuario no se considerará completa hasta que ha pasado pruebas de aceptación. Esto significa que las nuevas pruebas de aceptación se deben crear en cada iteración.

El aseguramiento de la calidad (QA) es una parte esencial del proceso de XP. En algunos proyectos de control de calidad es realizada por un grupo aparte, mientras que en otros QA serán integrados en el desarrollo de sí mismo. En cualquier caso XP requiere el desarrollo que tiene relación mucho más estrecha con la garantía de la calidad.

Las pruebas de aceptación deben ser automatizadas por lo que se puede ejecutar con frecuencia. Las pruebas de aceptación de puntuación son publicadas en el equipo.

Tabla N° 04: Formato de Pruebas de Aceptación

Caso de Prueba de Aceptación	
Código:	Historia de Usuario (Nro. y Nombre):
Nombre:	
Descripción:	
Condiciones de Ejecución:	
Entrada / Pasos de ejecución:	
Resultado Esperado:	
Evaluación de la Prueba:	

Fuente: Ejemplo de desarrollo software utilizando la metodología XP, LSI-PSW-LDS Departamento Sistemas Informáticos y Computación (DSIIC) – Universidad Politécnica de Valencia, 2006.

d.4 Despliegue de Software en la Producción

Al final de la versión, el producto es verificado y certificado para su despliegue en el entorno de producción del cliente. Esto es también el punto donde un experto de usabilidad se reuniría con los usuarios de la muestra para realizar pruebas de aceptación sobre la última construcción. Las pruebas sobre verdaderos usuarios asegurarán que el flujo del interfaz apoya a usuarios en el logro de sus tareas.

Descubierto todas las preguntas que dificultan la productividad de los usuarios deben reconciliarse y lo que significa que la construcción todavía no es completa.

Esta fase podría incluir algún sistema de sintonía o de aplicación, dependiendo del ambiente objetivo. La sintonía puede ser ejecutada en el entorno de la organización, suponiendo en la producción espejos. (Los clientes normalmente niegan el acceso a los

desarrolladores de código de producción, de todos modos.) El objetivo no es seguir la evolución funcional, pero para estabilizar el sistema.

En la teoría, cualquier sistema desarrollado usando PROGRAMACIÓN EXTREMA puede ser desplegado en la producción tan pronto como la primera iteración. (La primera iteración podría ser una fase puramente técnica donde el equipo establece el sistema base.) El cliente seleccionará el despliegue de la iteración en función del valor de negocio y los factores externos, tales como la migración de datos o la integración de sistema heredados.

El calendario de despliegue será probablemente dependiente de las limitaciones externas, no a la capacidad del equipo para la producción de software con la suficiente rapidez.

El enfoque del despliegue requiere un grado de planificación y gestión de riesgos ya que el equipo de desarrollo ahora entra en el ambiente de producción del cliente. En la producción del sistema usted tiene que ser consciente que el personal de operaciones requerirá guías de despliegue u otra documentación de apoyo. Cualquier documentación que el equipo de desarrollo escribe será de acuerdo al enfoque ligero de PROGRAMACIÓN EXTREMA; lo que debería ser suficiente, pero no más.

Si la Gran Explosión o la migración de por fase son usadas para el despliegue es una decisión que debe ser hecha por todas las partes interesadas: los desarrolladores, clientes, operaciones y usuarios. La Tabla 04 hará la comparación de estos enfoques.

Tabla 05: Comparación de enfoques de despliegue.

Enfoque	Descripción
Explosión (Big Bang)	Singularice corte después de la terminación del sistema. Este enfoque es usado cuando un sistema existente está siendo substituido y gestionado en un período de transición que es demasiado complejo o costoso. El despliegue es más sencillo, pero los riesgos son más altos y los fracasos pueden ser críticos.
Gradual	El nuevo sistema se extenderá en piezas discretas de funcionalidad. Esto funciona bien cuando el sistema se puede, lógicamente, dividir en secciones. Desarrollando el sistema durante un periodo prolongado que puede reducir el riesgo, pero si exponer a la empresa a un largo plazo de perturbaciones.
Paralelo	Esto implica tanto en el funcionamiento de los sistemas y los nuevos al lado de la otra por un periodo de tiempo. El alto coste asociado con múltiples entradas de datos y presentación de informes, tiende a la reducción de los riesgos superados a los aspectos de este enfoque.
Campo Verde	El sistema se despliega en un entorno donde no existe ningún sistema. Este podría ser un inicio o una nueva línea del negocio que tiene ahora el apoyo de productos de software. Este es el escenario ideal para el despliegue, porque los riesgos son menores y las preguntas de la integración con otros sistemas son mínimas.

e. Manteniendo el Sistema Después de la Versión o Liberación

Según Kent Beck, Martín Fowler, "... al final de cada iteración, usted debe probar no sólo que la historia se ha terminado si no que

también funcione. A veces las pruebas automatizadas pueden ser creadas y administradas por el cliente al final de una iteración.”¹⁴

Tras el despliegue del sistema, este entra en un permanente estado de mantenimiento. Con Programación Extrema que ha sido utilizado constantemente en la evolución, refactorización y perfeccionamiento del sistema. Lo que ayudan a la capacidad de parche o actualización del sistema con la ayuda su establecimiento y verificando la plataforma automatizadas de pruebas. Esto significa que puede seguir adelante y libremente realizar cambios en el código con confianza.

El despliegue requiere más atención ahora que se trata de un sistema de producción. Tendrá que crear la migración de datos y el sistema de estrategias de parches. Habrá un impacto sobre el equipo de desarrollo de la producción, como los desarrolladores de hacer malabares entre su tiempo de apoyo a la producción y los nuevos trabajos.

Este es un hecho de la vida, por lo que en lugar de hacer caso omiso de este menos que al estado ideal, el equipo de registro y medida de su nuevo rendimiento. Las estimaciones para la siguiente o actual iteración se ajustarán en consecuencia a la voluntad.

2.2.3 HERRAMIENTAS DE DESARROLLO

2.2.3.1 ¿Porqué usar Java?

Java es un lenguaje orientado a objetos. Esto significa que posee ciertas características que hoy día se consideran estándares en los lenguajes OO:

¹⁴ Kent Beck, Martin Fowler: Planning “*Extreme Programming Explored*”, Addison-Wesley Editorial “Pearson Education, 1999”, Addison Wesley, 2000.

- Objetos
- Clases
- Métodos
- Subclases
- Herencia simple
- Enlace dinámico
- Encapsulamiento

Estos conceptos no son simples, por lo que se explicará su significado más adelante.

Para programar orientado a objetos es necesario primero *diseñar un conjunto de clases*. La claridad, eficiencia y mantenibilidad del programa resultante dependerá principalmente de la calidad del diseño de clases. Un buen diseño de clases significará una gran economía en tiempo de desarrollo y mantención¹⁵.

Porque la ventaja potencial más importante de un lenguaje OO está en las *bibliotecas de clases* que se pueden construir para él. Una biblioteca de clases cumple el mismo objetivo de una biblioteca de procedimientos en un lenguaje como C. Sin embargo:

Una biblioteca de clases es mucho más fácil de usar que una biblioteca de procedimientos, incluso para programadores sin experiencia en orientación a objetos. Esto se debe a que las clases ofrecen mecanismos de abstracción más eficaces que los procedimientos.

a. Java posee bibliotecas de clases estándares

Toda implementación de Java debe tener las siguientes bibliotecas de clases:

- a. Manejo de archivos

¹⁵ Tomado de: Revista digital emitida por Universidad de Chile - Departamento de Ciencias de la Computación. <<http://www.dcc.uchile.cl/~lmateu/Java/Apuntes/java>>

- b. Comunicación de datos
- c. Acceso a la red Internet
- d. Acceso a bases de datos
- e. Interfaces gráficas

La interfaz de programación de estas clases es estándar, es decir en todas ellas las operaciones se invocan con el mismo nombre y los mismos argumentos.

b. Java es multiplataforma

Los programas desarrollados en el lenguaje Java pueden ejecutarse en cualquiera de las siguientes plataformas, sin necesidad de hacer cambios:

- Microsoft Windows
- Mac OS
- Unix
- Linux

La compatibilidad es total debido a que se ejecuta sobre una **Máquina virtual Java** (en inglés *Java Virtual Machine*, **JVM**) que es un máquina virtual de proceso nativo, es decir, ejecutable en una plataforma específica, capaz de interpretar y ejecutar instrucciones expresadas en un código binario especial (el Java bytecode), el cual es generado por el compilador del lenguaje Java.

La gran ventaja de la máquina virtual java es aportar portabilidad al lenguaje de manera que desde Sun Microsystems se han creado diferentes máquinas virtuales java para diferentes arquitecturas y así un programa .class escrito en un Windows puede ser interpretado en un entorno Linux. Tan solo es necesario disponer de dicha máquina virtual para dichos entornos. De ahí el famoso axioma que sigue a Java, "escríbelo una vez, ejecútalo en cualquier parte", o "Write once, run anywhere".

c. Java es robusto:

- Java siempre chequea los índices al acceder un arreglo.
- Java realiza chequeo de tipos durante la compilación (al igual que C). En una asignación entre punteros el compilador verifica que los tipos sean compatibles.

Además, Java realiza chequeo de tipos durante la ejecución (cosa que C y C++ no hacen). Cuando un programa usa un *cast* para acceder un objeto como si fuese de un tipo específico, se verifica durante la ejecución que el objeto en cuestión sea compatible con el *cast* que se le aplica. Si el objeto no es compatible, entonces se levanta una excepción que informa al programador la línea exacta en donde está la fuente del error.

- Java posee un recolector de basuras que administra automáticamente la memoria. Es el recolector el que determina cuando se puede liberar el espacio ocupado por un objeto. El programador no puede liberar explícitamente el espacio ocupado por un objeto.
- Java no posee aritmética de punteros, porque es una propiedad que no se necesita para programar aplicaciones. En C sólo se necesita la aritmética de punteros para programa *malloc/free* o para programar el núcleo del sistema operativo.

d. Java es flexible frente a otros

Pascal también es otro lenguaje robusto, pero logra su robustez prohibiendo tener punteros a objetos de tipo desconocido. Lamentablemente esta prohibición es demasiado rígida. Aunque son pocos los casos en que se necesita tener punteros a objetos de tipo desconocido, las contorsiones que están obligados a realizar los programadores cuando necesitan estos punteros dan origen a programas ilegibles.

Lisp por su parte es un lenguaje flexible y robusto. Todas las variables son punteros a objetos de cualquier tipo (un arreglo, un elemento de lista, etc.). El tipo del objeto se encuentra almacenado en el mismo objeto. Durante la ejecución, en cada operación se chequea que el tipo del objeto manipulado sea del tipo apropiado. Esto da flexibilidad a los programadores sin sacrificar la robustez. Lamentablemente, esto hace que los programas en Lisp sean poco legibles debido a que al estudiar su código es difícil determinar cuál es el tipo del objeto que referencia una variable.

Java combina flexibilidad, robustez y legibilidad gracias a una mezcla de chequeo de tipos durante la compilación y durante la ejecución. En Java se pueden tener punteros a objetos de un tipo específico y también se pueden tener punteros a objetos de cualquier tipo. Estos punteros se pueden convertir a punteros de un tipo específico aplicando un *cast*, en cuyo caso se chequea en tiempo de ejecución de que el objeto sea de un tipo compatible.

2.2.3.2 POSTGRESQL

POSTGRESQL es un sistema de gestión de base de datos relacional orientada a objetos y libre, publicado bajo la licencia BSD.

Como muchos otros proyectos de código abierto, el desarrollo de POSTGRESQL no es manejado por una sola empresa sino que es dirigido por una comunidad de desarrolladores y organizaciones comerciales las cuales trabajan en su desarrollo. Dicha comunidad es denominada el PGDG (POSTGRESQL Global Development Group).¹⁶

¹⁶ Tomado de: <<http://blog.taragana.com/index.php/archive/five-reasons-to-use-postgresql/es/>>

a. Razones para utilizar POSTGRESQL:

a.1 POSTGRESQL es multiplataforma

POSTGRESQL es un motor de base de datos que está disponible para cada sistema operativo (multiplataforma), por lo que es el más adecuado para las cooperativas de ahorro y crédito del Perú, dado que en la realidad cuentan con motores de bases de datos diversos.

a.2 Código fuente libre y de alta calidad.

a.3 Fiabilidad y estabilidad

- Es capaz de manejar prácticamente cualquier tarea que se extiende de la creación de la tabla simple a la réplica de manejo a través de los servidores múltiples.
- El producto Navicat PostgreSQL ofrece las características similares al pgAdmin III y se empaqueta en un interfaz muy bien-diseñado. Una herramienta buena, en Internet es phpPgAdmin.
- POSTGRESQL es considerado estable por tener procedimientos almacenados, funciones y vistas.
- POSTGRESQL, es fiable ya que grandes compañías como SKYPE trabajan con bases de datos gigantescas.

a.4 Este motor trabaja con lenguajes de programación múltiples

PostgreSQL trabaja con C++, C#, JDBC, Perl, PHP, Python, Ruby, Tcl, Ada, etc.

a.5 Soporte técnico

Soporte técnico profesional tanto de la comunidad así como de empresas especializadas la cual es pagada pero de costos simbólicos.

a.6 Esta diseñada para entornos con altos volúmenes de tráfico/Transacciones.

a.7 Licencias libres

POSTGRESQL se puede usar sin restricciones, tanto en servidores como por conexiones.

a.8 POSTGRESQL es extensible.

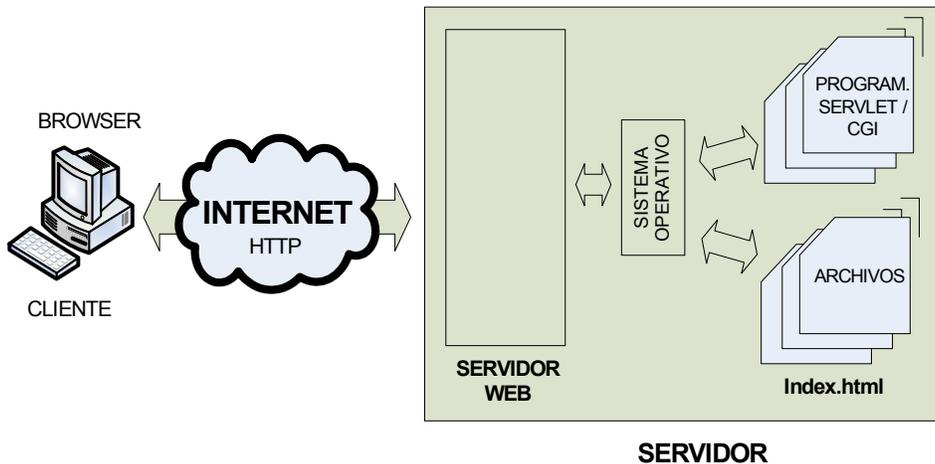
2.2.3.3 TECNOLOGÍA WEB

Servidor Web. Es un programa que sirve datos en forma de páginas Web, hipertextos o páginas HYPERTEXT MARKUP LANGUAGE: HTML; textos con enlaces, figuras, formularios, botones y objetos. La comunicación de estos datos entre cliente y servidor se hace por medio un protocolo, concretamente el protocolo HTTP.

2.2.3.4 ARQUITECTURA WEB

Para abrir una página Web en un navegador, normalmente se tecllea el correspondiente URL o se pica en el hiperenlace oportuno. Una vez que se solicita esta petición mediante el protocolo HTTP y la recibe el servidor Web, éste localiza la página Web en su sistema de ficheros y la envía de vuelta al navegador que la solicitó.

Figura N° 05: Arquitectura Web básica



Fuente: Tesis, L Berrocal "Sistema de Información Web para el Turismo Receptivo de Ayacucho bajo el modelo Network Marketing".

a. APLICACIONES WEB

Cuando se comenzó con el uso de la informática en una empresa se hizo bajo un modelo cliente/servidor. Los usuarios disponían de una terminal con una pequeña pantalla conectada a un servidor donde coexistían los datos y el software. Más tarde, con la aparición del PC, el modelo cambió y la lógica y los datos pasaron a residir completamente en el PC, con los consecuentes problemas de administración, soporte y coste de las licencias. Esto no varió demasiado durante bastante tiempo, pero ahora, parece que volvemos al principio. La era del PC ha terminado para dejar paso a la era Internet y es justamente ahí donde se van a alojar nuestros programas y datos. Con las Aplicaciones Web se recupera el papel del servidor que se convierte ahora en un servidor Web. Los datos se almacenan en bases de datos accesibles desde un navegador Web o una terminal móvil gracias a la lógica que se ejecuta en el servidor y al diseño del interfaz que es transferido a dichas terminales. Se conocen buenos ejemplos de este tipo de programas, como puede ser el correo electrónico basado en Web o las comunidades en línea.¹⁷

Una Aplicación Web es un conjunto de páginas Web enlazadas que visualizan diferentes partes de la información que se quiere mostrar a través de ella. Constituye una de las mejores herramientas para divulgar, gestionar y compartir la información por lo que trae consigo un aumento de la eficiencia en cuanto a la manipulación de gran cantidad de información.

¹⁷."Sistema de Almacén central" Tesis Ingeniería Yoan D. Puig Rodríguez de la Universidad de Holguín Oscar Lucero Moya -CUBA. pp.16.

b. El Servidor HTTP Apache, es un software (libre) de código abierto para plataformas Unix, Windows, Macintosh y otras, que implementa el protocolo HTTP/1.1 y la noción de sitio virtual.¹⁸

c. Apache Tomcat, es una aplicación de software de código abierto del Java Servlet y JavaServer Pages tecnologías. Java Servlet y JavaServer especificaciones páginas se desarrolló en el marco del Java Community Process.

d. Apache Tomcat, es desarrollado en un entorno abierto y participativo y publicado bajo la licencia Apache Software. Apache Tomcat está destinado a ser una colaboración de los mejores de los desarrolladores de la raza de todo el mundo. este invita a participar en este proyecto de desarrollo abierto.

2.2.3.5 MECANISMOS DE SEGURIDAD

No existe un único mecanismo capaz de proveer todos los servicios anteriormente citados, pero la mayoría de ellos hacen uso de técnicas criptográficas basadas en el cifrado de la información. Los más importantes son los siguientes:

- **Intercambio de autenticación.** Corrobora que una entidad, ya sea origen o destino de la información, es la deseada.
- **Cifrado.** Garantiza que la información no sea inteligible para individuos, entidades o procesos no autorizados.
- **Integridad de datos.** Este mecanismo implica el cifrado de una cadena comprimida de datos a transmitir. Este mensaje se envía al receptor junto con los datos ordinarios. El receptor repite la compresión y el cifrado posterior de los datos y compara el

¹⁸ Tomado de la tesis, L Berrocal "Sistema de Información Web para el Turismo Receptivo de Ayacucho bajo el modelo Network Marketing".

resultado obtenido con el que le llega, para verificar que los datos no han sido modificados.

- **Firma digital.** Este mecanismo implica el cifrado, por medio de la clave secreta del emisor, de una cadena comprimida de datos que se va a transferir. La firma digital se envía al receptor junto con los datos ordinarios. Este mensaje se procesa en el receptor, para verificar su integridad.
- **Control de acceso.** Esfuerzo para que sólo aquellos usuarios autorizados accedan a los recursos del sistema o a la red.
- **Tráfico de relleno.** Consiste en enviar tráfico falseado junto con los datos válidos para que el enemigo no sepa si se está enviando información, ni qué cantidad de datos útiles se está transfiriendo.
- **Control de encaminamiento.** Permite enviar determinada información por determinadas zonas clasificadas. Asimismo posibilita solicitar otras rutas, en caso que se detecten persistentes violaciones de integridad en una ruta determinada.

CAPÍTULO III

METODOLOGÍAS DE LA INVESTIGACIÓN

3.1. TIPO DE INVESTIGACIÓN

El tipo de investigación usado en la presente es la “Aplicada” (activa o dinámica). Se usa este tipo con el propósito de solucionar el problema antes que cualquier formulación de teorías; se busca conocer para hacer, para actuar, para construir, para modificar; se basa en la aplicación inmediata de una alternativa de solución sobre la realidad de la forma como conducen el almacén en las Coopac's de ahorro y crédito del Perú, planteando soluciones concretas, reales, factibles y necesarias a los problemas determinados.

3.2. DISEÑO DE INVESTIGACIÓN

Para el tipo de investigación elegida, el diseño adecuado es el no-experimental, porque nos permite observar de manera directa el sistemas manual del funcionamiento del almacén en las Coopac's del Perú, en su contexto natural para describirlas y analizarlos. En otras palabras nos permite investigar de una manera sistemática e/y empírica.

Diseño no-experimental transversal; una manera más específica de mostrar el diseño utilizado es el no-experimental transversal, que nos permite recolectar datos de un solo momento en un tiempo único cuyo propósito es describir las variables y analizar sus incidencias.

3.3 TÉCNICA DE RECOLECCIÓN DE DATOS

3.3.1 Método de Entrevista

Se empleó el método de entrevista directa, al personal encargado y capacitado.

3.3.2 Documentos y Materiales

- Credencial
- Carta de presentación
- Útiles

3.3.3 Muestreo

Se uso del tipo panel, el cual se emplea para las Coopac's del Perú formales (reconocidas por FENACREP) autorizadas a operar con la finalidad de medir los distintas formas de llevar un almacén.

Para el caso se entrevistó al personal de 07 cooperativas y son:

- Cooperativa de Ahorro y Crédito Santa María Magdalena.
- Cooperativa de Ahorro y Crédito San Cristóbal de Huamanga.
- Cooperativa de Ahorro y San Pedro de Andahuaylas.
- Cooperativa de Ahorro y Crédito San Martín de Porres de Tarapoto.
- Cooperativa de Ahorro y Crédito Quillacoop de Quillabamba.
- Cooperativa de Ahorro y Crédito Santo Domingo de Guzmán.
- Cooperativa de Ahorro y Crédito Señor de Maynay.

Ver Anexo D.

3.4 TÉCNICA DE PROCESAMIENTO DE DATOS

Se realiza una exhaustiva tarea de depuración de la información revisando los datos obtenidos de cada una de las entrevistas realizadas en campo. Así mismo se hace la respectiva consistencia con la finalidad

de sanear algunos errores al captar los datos o por la falta de veracidad del informante.

3.4.1 Codificación

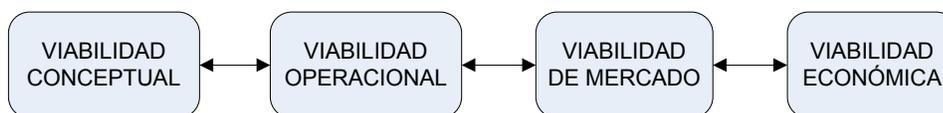
En esta fase se forma un grupo de valores numéricos asignando a todas las preguntas abiertas y las respuestas de tal forma que los datos puedan ser agrupados en según respuestas comunes y darles la codificación respectiva.

3.5 ESTUDIO DE VIABILIDAD

Este procedimiento se realizó para saber si el proyecto elegido como alternativa es posible llevarlo a cabo; es decir, si es viable desde el punto de vista legal, técnico, económico, comercial y financiero.

No existe un método categóricamente universal para llevar a cabo un estudio de viabilidad ya que cada proyecto que se evalúa es diferente. Sin embargo se sigue un enfoque genérico dividido en 4 aspectos esenciales a toda iniciativa empresarial. Ver la siguiente figura.

Figura N° 06: Fases genéricas del estudio de viabilidad



Fuente: L. Berrocal Tesis: "Sistema de Información Web para el Turismo Receptivo de Ayacucho bajo el modelo Network Marketing

La idea de establecer el sistema se analiza en función de cada uno de éstos aspectos, aunque la secuencia no es estrictamente necesaria pero ayuda en la planificación y ejecución del estudio.

3.5.1 Viabilidad Conceptual

Es necesario realizar un análisis crítico y exhaustivo en base a los fundamentos teóricos del proyecto.

3.5.2 Viabilidad Operacional

Se evalúa objetivamente los siguientes aspectos relacionados a la operación y administración del negocio propuesto: recursos humanos apropiados y experimentado (personal nuevo ingresa con proceso de inducción), infraestructura disponible, capacidad tecnológica, requisitos legales, ambiente competitivo, disponibilidad de tiempo, entre otros.

3.5.3 Viabilidad de Mercado

Probablemente sea el componente más crítico debido a lo complejo que resulta este análisis que para propósitos de esta investigación se incluye como mínimo: el mercado potencial y prioritario, la participación proyectada del mercado y las proyecciones de ventas.

3.5.4 Viabilidad Económica

Este análisis financiero conlleva los siguientes pasos: análisis de las fuentes y usos de fondos, proyecciones de ingresos y gastos, análisis del punto de equilibrio, estimación del periodo de retorno de inversión y la estimación de la tasa de ganancias.

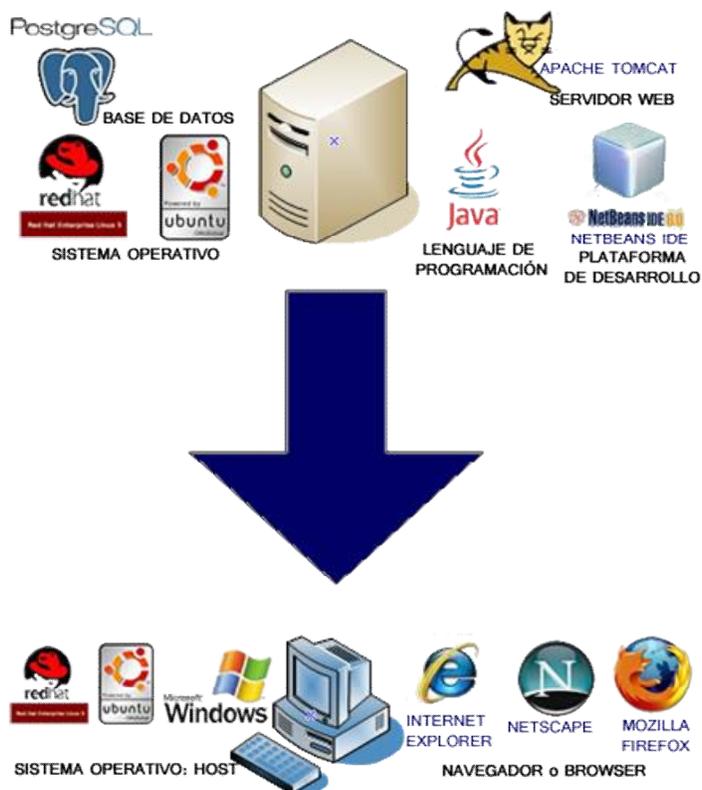
CAPITULO IV

RESULTADOS

4.1 INSTALACIÓN CONFIGURACIÓN DE HERRAMIENTAS

Se va a instalar la versión 11.0 de openSuse Linux en español. Pudiéndose también usar Ubuntu, Red Hat Linux Enterprise, etc. Esto debido a que con ello se reducen gastos de licenciamiento de uso de software, para el caso usaremos para la instalación de un servidor de base de datos PostgreSQL, apache 2.0.55+jboss 4.0.4 para trabajo en un entorno de desarrollo j2ee con netbeans 6 bajo ubuntu 7.04. Quedando como se puede apreciar en la Figura N°07.

Figura N°07: Plataforma Aplicación Web.



Fuente: Adecuación desarrollo propio.

4.1.1 Modo de instalación

Se selecciona "Instalación nueva" y se marca solamente la casilla para configuración automática; luego presionar siguiente.

4.1.2 Particionamiento

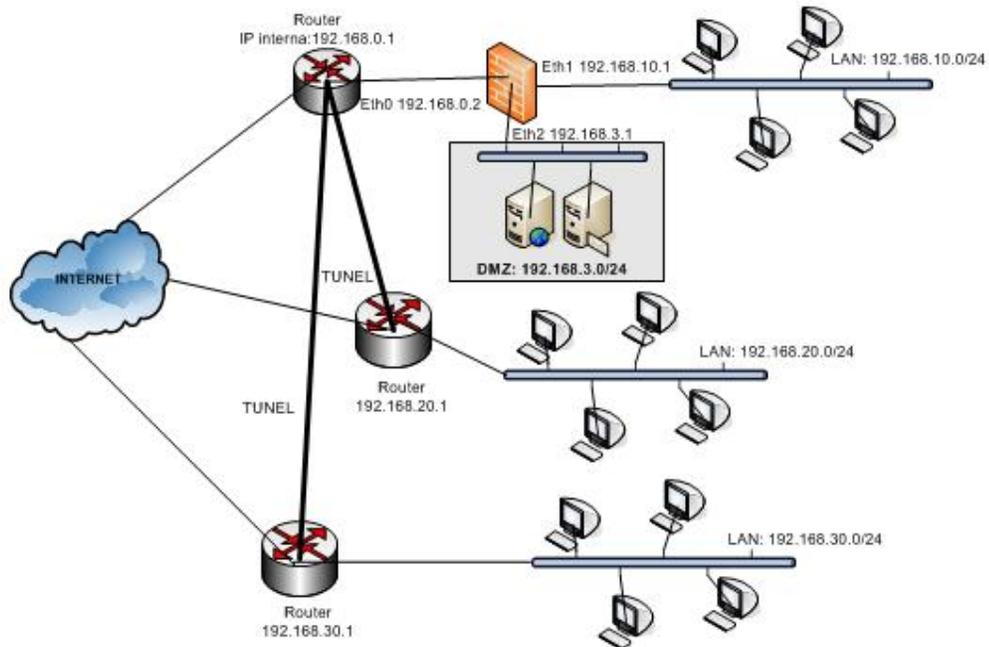
Al particionar el disco duro se hizo 3 particiones:

- Una para el intercambio (Swap).
- Una para la raíz de Linux(/).
- Y la ultima para la carpeta/home.

4.2 Diagrama físico de redes sobre la cual correrá la aplicación Web.

La siguiente figura N° 08, muestra la red donde se ubica la zona desmilitarizada (DMZ) El mismo que es válido para el 90% de cooperativas del Perú, ya que ese mismo porcentaje de cooperativas que tienen agencias o sucursales.

Figura N° 08: Diagrama Físico de redes de la Aplicación Web.



Fuente: Adecuación desarrollo propio.

4.3 INSTALACIÓN DE JAVA

Para escribir y compilar un programa java, applets o servlets hay que descargar el kit de desarrollo Java (JDK), que dispone de un compilador, bibliotecas de clases y herramientas para compilar y ejecutar código Java. En la presente se va a utilizar el JDK de Sun.

Pudiendo descargarse el JDK de Sun según desea usar:

<http://java.sun.com/javaee/>

<http://java.sun.com/j2se/1.4.2/>

<http://java.sun.com/j2se/1.4.2/download.html>

Nota: El entorno de ejecución Java serán suficientes para configurar el entorno de servidor, pero el Software Development Kit (SDK) es necesaria si uno quiere escribir y compilar programas Java. El "Forte SDK" es una completa interfaz gráfica de usuario interactiva impulsado por un entorno de desarrollo. El SDK está disponible en formato RPM y tar.

- `j2sdk-1_4_0-linux-i386-rpm.bin`
- `chmod +x j2sdk-1_4_0-linux-i386-rpm.bin` `chmod +x j2sdk-1_4_0-linux-i386-rpm.bin`
- `./j2sdk-1_4_0-linux-i386-rpm.bin ./j2sdk-1_4_0-linux-i386-rpm.bin`
- `rpm -ivh j2sdk-1_4_0-fcs-linux-i386.rpm` `rpm-ivh j2sdk-1_4_0-fcs-linux-i386.rpm`

Quedando instalado en `/usr/java/j2sdk1.4.0/`.

4.3.1 Configuración:

Establezca la variable de entorno PATH. Añadir declaración a `$HOME / script .bash_profile` o el `$HOME / .bashrc` o Shell, que controla el entorno.

```
PATH = /usr/java/j2sdk1.4.0/bin: $PATH
export PATH
export JAVA_HOME = /usr/java/j2sdk1.4.0
export CLASSPATH = /usr/java/j2sdk1.4.0/lib/tools.jar: /
usr/java/j2sdk1.4.0/jre/lib/rt.jar
```

El script de shell puede ser re-ejecutado con el comando: `bashrc`

4.3.2 Prueba:

Programa de prueba siguiente: Test.java

```
public class prueba
{
public static void main (String [] args)
{
System.out.println ("Hola nueva Aplicación");
}
}
```

Compile: `javac Test.jav`

(o `/usr/java/j2sdk1.4.0/bin/javac Test.java`),

Debe tener en cuenta que el nombre de archivo y el nombre de la clase son los mismos. Esto se convirtió en un requisito en el JDK 1.4. El resultado de la compilación es el archivo: `Test.class`

Run:

```
[prompt] $ java Test
```

```
"Hola nueva Aplicación";
```

```
O en: /usr/java/j2sdk1.4.0/bin/java Test (o /
usr/java/j2sdk1.4.0/bin/java Test)
```

4.4 INSTALACIÓN DE PROJECT-TOMCAT:

Tomcat es el contenedor de Java "o procesador de Java Servlets y Java Server Pages (JSP). Se debe tener en cuenta también que Java debe estar instalado para que Tomcat pueda funcionar. Me centraré en el uso de Tomcat con Apache, señalando que se hizo la instalación por defecto de Tomcat; Tomcat puede trabajar independiente del servidor web http y contenedor de servlets.

El Proyecto de Yakarta se llama: Apache Tomcat Java Servlets y JSP
Página de inicio.

Descarga Tomcat 4.0 RPM (Lanzamiento: Catalina. Servlet 2.3 y JSP 1.2 Especificaciones.) los archivos necesarios son:

- `regexp-1.2-1.noarch.rpm`
- `servletapi4-4.0.3-1.noarch.rpm` - `servlet.jar` archivos
- `Xerces-J-1.4.4-1.noarch.rpm` - Xerces es un analizador de XML disponible en la Apache Software Foundation.
- `Tomcat4-webapps-4.0.3-1.noarch.rpm`
- `Tomcat4-4.0.3-1.noarch.rpm`

Debe tener los siguientes controladores instalados.

```
rpm-ivh regexp-1.2-1.noarch.rpm servletapi4-4.0.3-1.noarch.rpm Tomcat4-  
webapps-4.0.3-1.noarch.rpm Tomcat4-4.0.3-1.noarch.rpm Xerces -j-1.4.4-  
2.noarch.rpm
```

Enlaces simbólicos generados a partir de `/usr/share/java/ ...` a `/var/tomcat4/`

Documentación:

- Tomcat 4.0 Documentación puede encontrarla en:
(<http://tomcat.apache.org/tomcat-4.0-doc/index.html>)
- También puede buscar documentación en disco local en:
`/usr/share/doc/tomcat4-4.0.3/` notas de la versión.
- Los archivos de documentación local están en:
`/var/tomcat4/webapps/tomcat-docs/`

Config file: `/etc/tomcat4/conf/tomcat4.conf`

Cambiar la siguiente referencia de `JAVA_HOME` a:

```
JAVA_HOME = /usr/java/j2sdk1.4.0
```

Notas:

- Variable de entorno `JAVA`, parecieran estar aquí y no en un script de inicio.
- Opciones de variable de entorno:
 - `CATALINA_HOME`: El directorio de instalación de Tomcat

- Ejemplo CATALINA_OPTS: `export CATALINA_OPTS="-Djava.ext.dirs=/dummy_dir"`
De la línea de comandos para pasar al entorno de ejecución Java
- CLASSPATH: Opcional - se ejecuta sin esto, pero yo lo utilizo cuando añado los archivos JAR personalizados. (`CLASSPATH = $CLASSPATH: /var/tomcat4/lib: /usr/share/postgresql:/usr/java/jre/lib:/usr/java/j2sdk1.4.0/lib:`)
- Tomcat se ejecutará como usuario Tomcat4.

4.4.1 Configuración de Archivos:

- `/var/tomcat4/conf/server.xml` - configuración del contenedor Servlet: Define los servicios ofrecidos, el número de puerto TCP para los servicios, SSL, configuración JDBC para PostgreSQL, MySQL (usuario/contraseña), Oracle, ODBC, etc. Más adelante se va a configurar el Apache para utilizar uno de estos servicios.
- `/var/tomcat4/conf/web.xml` - Tomcat integrado en el archivo de configuración global del servidor http: Define servlets para ser procesados por Tomcat. Algunos están predefinidos para realizar las tareas pre-configurados como SSI, JSP, etc.
- `/var/tomcat4/conf/catalina.policy` - Política de seguridad para los permisos de Tomcat.
- `/var/webapps/examples/WEB-INF/web.xml` - archivo de configuración de aplicaciones específicas.

Iniciando el servicio: Servicio de Tomcat4 inicio (o: `/etc/init.d/tomcat4 de inicio`).

`/etc/rc.d/init.d/tomcat4 start`

O ejecutar directamente los guiones de inicio:

`/etc/rc.d/init.d/tomcat4 start`

Para añadir a init proceso de arranque:

/sbin/chkconfig - add Tomcat4

Tomcat (socket de escucha y espera de las solicitudes) se iniciará antes que Apache, de forma que Apache tiene un socket a la que pueden conectarse y comunicarse. Tomcat también se detendrá después de Apache. Tenga paciencia ya que hay un montón de cosas pasando. Utilice el comando top para iniciar todos los procesos Java que se iniciaron. Cuando los procesos que se iniciaron ya no son de Java, a continuación, Tomcat es probable que esté listo.

Pruebas: Escribir en el browser en la parte de dirección:

http://localhost:8180/

http://localhost:8180/examples/jsp/index.html - JSP muestras

4.4.2 Configuración de Apache-Tomcat

Apache es un servidor Web rápido y configurable. En esta sección se explica el uso de Apache como servidor Web principal, pero utilizando para procesar Tomcat JSP y Servlets. Esta configuración utiliza la AJP (Apache JServ Protocol) elemento conector (mod_jk) para realizar esta tarea. mod_jk El mod_jk módulo de Apache enviará las peticiones de servlets de Tomcat usando sockets TCP / IP para las comunicaciones. Mod_jk módulo funciona con las versiones de Tomcat 4.x. y para todas las versiones anteriores.

Existen numerosos módulos de conexión de Apache:

Tabla N° 05: Detalle de los Módulos de Conexión de apache

Nombre del conector	Nombre del módulo de Apache	Protocolo de conexión	Versión de Tomcat.
AJP	mod_jk	AJP1.2, AJP1.3	3, 4
Coyote JK2	mod_jk	AJP/JK1.3, AJP/JK1.4	4,1
WARP	mod_webapp	WARP	4
JServ	mod_jserv	??	Viejo

Fuente: <http://www.agapea.com/libros/Apache-Tomcat-5-ISBN-8441517800-i.htm>

4.4.3 Instalación AJP mod_jk:

- Primero verifique si Apache está compilado para el soporte DSO: `/usr/sbin/httpd -l`.

Si una de las líneas de `mod_so.c` contiene el resultado entonces el soporte está disponible.

- Descargar mod_jk RPM: http://jakarta.apache.org/builds/jakarta-tomcat-4.0/release/v4.0.3/rpms/mod_jk-1.3-1.0-1.4.0.2.i386.rpm

Este módulo es dependiente de la versión de Apache se está utilizando. Para este caso se está usando Apache 1.3.

- Descargar una versión en EAPI de mod_jk:
http://jakarta.apache.org/builds/jakarta-tomcat/release/v3.3/bin/linux/i386/mod_jk-3.3-ap13-eapi.so

4.4.3.1 Modificaciones del archivo de configuración:

Los cambios de configuración de archivo examinan a continuación se refieren a los liberados con el mod_jk RPM-1.3-1.0-1.4.0.2.i386.rpm.

- Definir los puertos TCP / IP se utiliza para las comunicaciones entre Apache y Tomcat:
 - Ver archivo tomcat-configuración de apache: `/etc/httpd/conf/workers.properties` (Un host independiente también puede ser especificado.) (Si está definiendo múltiples JVM / servlet procesadores, utilizados en el hosting virtual, cada uno debe ser asignado un puerto diferente.)
 - Los servicios de Tomcat se definen en `/var/tomcat4/conf/server.xml` (Ver "Conector" definición) y en el archivo de inclusión `mod_jk.conf` Apache. Parámetros de recursos también puede ser configurado para controlar el número máximo de subprocessos que puede ser generado.

- Módulo de carga para la comunicación con Tomcat. httpd.conf (Añadir a httpd.conf)
 - (ruta de URL de los servlets) y / o tipos de archivo (los tipos de archivos JSP) que están destinados a la transformación de Tomcat. Todas las solicitudes en la ruta dada (ubicación servlet) o solicitud de un determinado tipo de archivo (por ejemplo *.jsp) y luego serán procesados por Tomcat. (véase la Directiva JkMount en Apache archivo de inclusión mod_jk.conf) Todo lo demás es servido por Apache.

Modificar: /etc/httpd/conf/ **workers.properties**

Añadir a: /etc/httpd/conf/httpd.conf

```
LoadModule jk_module /usr/lib/apache/mod_jk.so
```

OR

```
LoadModule jk_module    modules/mod_jk.so
```

...

...

```
AddModule mod_jk.c
```

...

...

```
Include /etc/httpd/conf/mod_jk.conf
```

Nota:

La sentencia modules/mod_jk.so es una referencia relativa de /etc/httpd/. El directorio /etc/httpd/modules/ es flexible vinculado a /usr/lib/apache/

File: /etc/httpd/conf/mod_jk.conf (**Cambiar las referencias de "tomcat" a "Tomcat4" en todo el archivo, de lo contrario el archivo por defecto está bien.**)

```

</IfModule Mod_jk.c>
  JkWorkersFile /etc/httpd/conf/workers.properties
  JkLogFile /var/log/httpd/mod_jk.log
  JkLogLevel info
  # Contexto raíz monturas para Tomcat
  # Formato: JkMount URL_PREFIX WORKER_NAME

  JkMount / *.jsp ajp13

  JkMount/servlet/ * ajp13

  # La siguiente línea hace apache consciente de la ubicación de
  /examples contextAlias /examples:
  <Directory "/var/tomcat4 /webapps/examples">
    Options Indexes FollowSymLinks
  </Directory>

  # La siguiente línea levanta todos los archivos JSP y el / servlet / uri
  a Tomcat
  JkMount/examples/servlet/* ajp13
  JkMount/examples/ *.jsp ajp13
  #La siguiente línea prohíbe a los usuarios el acceso directo WEB-INF
  <Location "/examples/WEB-INF/">

    AllowOverride None AllowOverride None
    deny from all deny from all
  </Location> </ Location>
  ...

  ...
</IfModule> </ IfModule>

```

4.5 CONFIGURACIÓN DE NIVELES DE SEGURIDAD

La seguridad en sus diferentes niveles para "montar" un servidor Web es un punto muy importante a considerar e implementar, ya sea en el servidor, en el administrador de base de datos o en la programación misma.

4.5.1 Seguridad en Apache

Cuando un servidor Web recibe una petición, antes de devolver el resultado, lleva a cabo varias acciones para verificar que esté

autorizada; estas acciones se pueden agrupar en tres tipos: Autenticación, Autorización y Control de Acceso.

El proceso por el cual se verifica la identidad de un usuario, es la autenticación que puede estar gestionada por distintos módulos, dependiendo de la forma de implementación. Se decide llevarla a cabo gestionando ficheros con listas de usuarios y contraseñas (encriptadas), utilizando el módulo `mod_auth`.

```
htpasswd -c ruta/passwords super@te
```

Este código crea el archivo de usuarios, te pide la clave para ser codificada.

```
htpasswd -c ruta/groups
```

Este otro crea el archivo de grupos donde se introducirá los usuarios autenticados.

Por último se crea un fichero con el nombre `.htaccess` en la carpeta que se quiere proteger, conteniendo las siguientes líneas:

```
AuthType Basic  
AuthName "Sistema_Almacen"  
AuthUserFile /ruta/passwords  
AuthGroupFile /ruta/groups  
Require group usuariosAutenticados
```

Las directivas de `mod_auth` necesarias para configurar la autenticación y autorización son el `AuthUserFile` y el `AuthGroupFile` que sirve para especificar la ruta donde se almacenará el fichero de usuarios y grupos respectivamente.

Las directivas necesarias para complementar la configuración del módulo son: `AuthType`, `AuthName` y `Require`.

El proceso por el cual se verifica que un usuario con una identidad conocida, tiene acceso al recurso solicitado, es la autorización que una de sus formas está gestionada mediante la directiva <directory> en el fichero principal de configuración.

Para lograr estos cambios es necesario tener la directiva AllowOverride con el valor AuthConfig.

4.6 CONECTANDO LA BASE DE DATOS:

Se hará la conectividad de la bases de datos en postgresQL:

JDBC y PostgreSQL archivos JAR: El "classpath" puede establecer la variable de entorno de ejecución Java de manera que se encuentran las bibliotecas apropiadas Java (archivos JAR). Las variables de entorno de Tomcat, se pueden establecer en /etc/tomcat4/conf/tomcat4.conf. También se puede establecer la variable CLASSPATH para incluir PostgreSQL JDBC archivos JAR; ya no la variable de entorno CLASSPATH en el fichero de configuración sino que se empleó la ruta de acceso predeterminada al realizar los siguientes pasos:

Lista de controladores de PostgreSQL JDBC:

```
[prompt]# rpm -ql postgresql-jdbc-7.1.3-2
/usr/share/pgsql/jdbc7.0-1.1.jar
/usr/share/pgsql/jdbc7.1-1.2.jar
Lugar JDBC bibliotecas JAR en la ruta donde se pueden encontrar:
cp /usr/share/pgsql/jdbc7.1-1.2.jar /var/tomcat4/lib cp/
usr/share/pgsql/jdbc7.1-1.2.jar / var/tomcat4/lib
```

4.6.1 Conectando:

Java Servlet se ejecutan en Tomcat, PostgreSQL mediante JDBC de acceso:

Fuente de Java Servlet archivo:

```
// File: muestrabase.java
/* Un servlet para mostrar el contenido de la base de datos
PostgreSQL Base */
import java.io.*;
```

```

import java.sql.*;
import java.text.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class ShowBedrock extends HttpServlet
{
    public String getServletInfo()
    {
        return "Servlet se conecta a la base de datos PostgreSQL y muestra el resultado de un SELECT";
    }
    private Connection dbcon;
    //Conexión para el conectar la base de datos bdalmacen
    // "Init" establece una conexión de base de datos
    public void init (ServletConfig config) throws ServletException
    {
        String LoginUser String = "postgres";
        String loginPasswd = "ultrasecreta";
        String LoginUrl String = "jdbc: postgresql: //localhost/bdalmacen";
        // Cargar el controlador de PostgreSQL
        try
        {
            Class.forName ( "org.postgresql.Driver");
            dbcon = DriverManager.getConnection (loginUrl, loginUser, loginPasswd);
        }
        catch (ClassNotFoundException ex)
        {
            System.err.println ( "No se ha encontrado la clase:" + ex.getMessage ());
            throw new ServletException ( " Error Clase no encontrada");
        }
        catch (SQLException ex)
        {
            System.err.println ( "SQLException:" + ex.getMessage ());
        }
    }
    // Usar http GET
    public void doGet (HttpServletRequest request,
        HttpServletResponse response)
        throws IOException, ServletException
    {
        response.setContentType ( "text/html"); //Respuesta de tipo MIME
        //Flujo de salida a STDOUT

```

```

PrintWriter out = response.getWriter ();
out.println("<HTML> <Head> <Title>Almacenes COOPAC</Title></
head>");
out.println ( "<BODY> <H1> Base de Datos Almacen </ H1>");
try
{
//hacemos nuestra declaración
statement statement = dbcon.createStatement();
String query = "SELECT NOMBRE, AREA,;";
query + = "Almacen";
query + = "FROM usuarios";
// Realizar la consulta
ResultSet rs=statement.executeQuery (query);
out.println ( "<borde de la tabla>");
// Itera a través de cada fila de rs
while (rs.next ())
{
String m_name = rs.getString ( "NOMBRE");
String m_dept = rs.getString ( "AREA");
String m_jobtitle = rs.getString ( " ALMACEN");
out.println ( "<tr> +
" <td>" m_name + " </ td>" +
" <td>" m_dept + " </ td>" +
" <td>" m_jobtitle + " </ td>" +
" </tr>");
}
out.println ( "</table> </body> </html>");
statement.close ();
}
catch (Exception ex)
{
out.println ( "<HTML>" +
"<Head> <Title>" +
"Base: Error" +
"</Title></head> \ n<body>" +
"<P> Error de SQL en doGet:" +
ex.getMessage()+"</P></body> </ HTML>");
return;
}
out.close ();
}
}

```

Notas:

- *public void doGet (...*

Uso de http GET. Para http POST use las operaciones doPost

- `String m_name=rs.getString ("nombre");`

También se pueden utilizar nombres de campo de base de datos o números de campo; es decir:

```
String m_name=rs.getString (1);
```

```
String m_dept=rs.getString (2);
```

Set CLASSPATH variable de entorno:

```
export CLASSPATH = $ CLASSPATH: / var/tomcat4/lib/jdbc7.1-1.2.jar
```

O

```
export CLASSPATH = /usr/java/j2sdk1.4.0/lib/tools.jar:/usr/java/j2sdk1.4.0/jre/lib/rt.jar:/var/tomcat4/common/lib/servlet.jar:/var/ tomcat4/lib/jdbc7.1-1.2.jar
```

4.6.2 Compile:

```
[prompt] # cd /var/tomcat4/webapps/examples/WEB-INF/classes  
[prompt] # javac muestrabase.java
```

PostgreSQL configuración: `/var/lib/pgsql/data/postgresql.conf`

```
Set: tcpip_socket = true
```

Esto permite JDBC para conectarse a PostgreSQL.

4.6.3 Prueba:

Escribir en el browser.

- Tomcat sólo: `http://localhost:8180/almacen/servlet/muestrabase`
- Apache Solicitud: `http://localhost/almacen/servlet/muestrabase`

4.7 DESARROLLO DE LA APLICACIÓN WEB

A. FASE DE EXPLORACIÓN

A.1 Historia de Usuario

En las reuniones con los clientes, se definieron una serie de historias de usuario, que son mencionadas en la siguiente tabla N° 06:

Tabla N° 06: Historias de usuario.

N°	Historia de Usuario
01	Emitir requerimiento de artículos
02	Otorgar requerimiento de artículos
03	Entregar requerimiento de artículos
04	Registrar Ingreso de artículo
05	Control de acceso de usuarios.
06	Mantener requerimiento de artículo
07	Mantener datos del empleado
08	Mantener usuario
09	Mantener artículo
10	Mantener área
11	Mantener agencia
12	Mantener cargo
13	Reporte de stock mínimo
14	Reporte de inventario valorado
15	Reporte de inventario físico
16	Reporte de requerimiento de artículo

A.2 Descripción de las Historias de Usuario

Tabla N° 07: Emitir Requerimiento de Artículo

Historia de Usuario	
Número: 1	Usuario: Empleado
Nombre historia: Emitir requerimiento de artículos	
Descripción: El Actor, tales como el encargado de cada área emiten un requerimiento de artículos ante la falta de artículos en su oficina; debiendo indicar la fecha de emisión, el carácter de urgencia, descripción, unidad de medida, cantidad, el área al que pertenece.	
Observaciones:	

Tabla N° 08: Otorgar requerimiento de artículos

Historia de Usuario	
Número: 2	Usuario: Administrador
Nombre historia: Otorgar requerimiento de artículos	
Descripción: EL Actor (Jefe de abastecimiento, el gerente, administrador) ubica por número de requerimiento de artículo pendientes de aprobar (debe contener fecha de requerimiento, artículo, unidades solicitadas, para aprobar dicho requerimiento).	
Observaciones:	

Tabla N° 09: Entregar requerimiento de artículos

Historia de Usuario	
Número: 3	Usuario: Almacenero
Nombre historia: Entregar requerimiento de artículos	
Descripción: EL Actor (Almacenero) ubica por un número de requerimiento de artículo, visualizando los datos del solicitante, los artículos a entregar, fecha de requerimiento, fecha de aprobación, y el nivel de prioridad, etc. el estado de requerimiento, la fecha de entrega, el carácter de requerimiento, etc. el almacenero Selecciona un artículo en stock por atender.	
Observaciones:	

Tabla N° 10: Registro de Ingreso de Artículos.

Historia de Usuario	
Número: 4	Usuario: Almacenero
Nombre historia: Registro de Ingreso de Artículos	
Descripción: EL Actor (Almacenero) para actualizar el stock de artículos, debe indicar fecha de ingreso de artículo, numero de documento de ingreso, datos del almacenero, el almacenero Selecciona un artículo en stock para actualizar, el sistema visualiza código, descripción, unidad, precio Unitario, stock mínimo, cantidad, línea de artículo y almacén. El sistema solo permite ingresar la cantidad de artículo para su actualización.	
Observaciones:	

Tabla N° 11: Control de acceso de usuarios.

Historia de Usuario	
Número: 5	Usuario: Todos Actores
Nombre historia: Control de acceso de usuarios.	
Descripción: Antes de iniciar la aplicación se solicita el nombre de usuario y su clave para que tenga acceso a los datos que corresponden a su categoría de usuario.	
Observaciones:	

Tabla N° 12: Mantener Requerimiento de Artículo.

Historia de Usuario	
Número: 6	Usuario: Todos Actores
Nombre historia: Mantener Requerimiento de Artículo.	
Descripción: EL Actor (Almacenero, administrador) ubica por un número de requerimiento de artículo , visualizando los datos del solicitante, los artículos solicitadas(código, descripción, unidad, precio unitario), fecha de requerimiento, carácter de requerimiento y el nivel de prioridad, etc. el actor está en la capacidad modificar carácter de requerimiento, prioridad y de eliminar, agregar y/o modificar un artículo de requerimiento.	
Observaciones:	

Tabla N° 13: Mantener Datos del Empleado.

Historia de Usuario	
Número: 7	Usuario: Administrador
Nombre historia: Mantener Datos del Empleado.	
Descripción: El administrador una vez admitido al sistema está en la capacidad de crear, modificar, eliminar empleados del sistema y asignar cargo, área y agencia donde se desempeñara.	
Observaciones:	

Tabla N° 14: Mantener Usuario.

Historia de Usuario	
Número: 8	Usuario: Administrador
Nombre historia: Mantener Usuario.	
Descripción: El administrador una vez admitido al sistema está en la capacidad de crear, modificar y dar de baja a un usuario y dar permisos según el tipo de usuario para el acceso sistema.	
Observaciones:	

Tabla N° 15: Mantener Artículo.

Historia de Usuario	
Número: 9	Usuario: Almacenero
Nombre historia: Mantener Artículo.	
Descripción: Todo Actor una vez admitido al sistema está en la capacidad de crear, modificar, eliminar artículos, asignar línea de artículo y almacén del sistema.	
Observaciones:	

Tabla N° 16: Mantener Área.

Historia de Usuario	
Número: 10	Usuario: Administrador
Nombre historia: Mantener Área.	
Descripción: Todo Actor una vez admitido al sistema está en la capacidad de crear, modificar y eliminar área.	
Observaciones:	

Tabla N° 17: Mantener Agencia.

Historia de Usuario	
Número: 11	Usuario: Administrador
Nombre historia: Mantener Agencia.	
Descripción: Todo Actor una vez admitido al sistema está en la capacidad de crear, modificar, eliminar agencia y también están en la capacidad de asignar o eliminar áreas de Agencia.	
Observaciones:	

Tabla N° 18: Mantener Cargo.

Historia de Usuario	
Número: 11	Usuario: Administrador
Nombre historia: Mantener Agencia.	
Descripción: Todo Actor una vez admitido al sistema está en la capacidad de crear, modificar y eliminar cargos del sistema.	
Observaciones:	

A.3 Arquitectura en Capas de la Aplicación

La interfase al usuario se presenta mediante un browser o navegador de Internet, haciendo uso de la arquitectura Web "Cliente-Servidor".

La arquitectura de la aplicación para prestar servicios Web está compuesta por las siguientes capas (tiers, ver Figura N° 09):

- Capa de Aplicación: páginas Web servidas mediante un servidor Web (Apache) que contiene las capas de lógica del negocio y manejo de datos.
- Capa Lógica del Negocio: se resuelve la manera funcional pero simple la lógica del negocio que recibe/envía la información a la capa de presentación, mediante la interacción directa con la capa de manejo de datos.
- Capa de Manejo de Datos: se utiliza un motor de base de datos relacional, el mismo que es accedido mediante un esquema de persistencia que la hace transparente para la aplicación: el mapeo objetos-relacional (ORM).

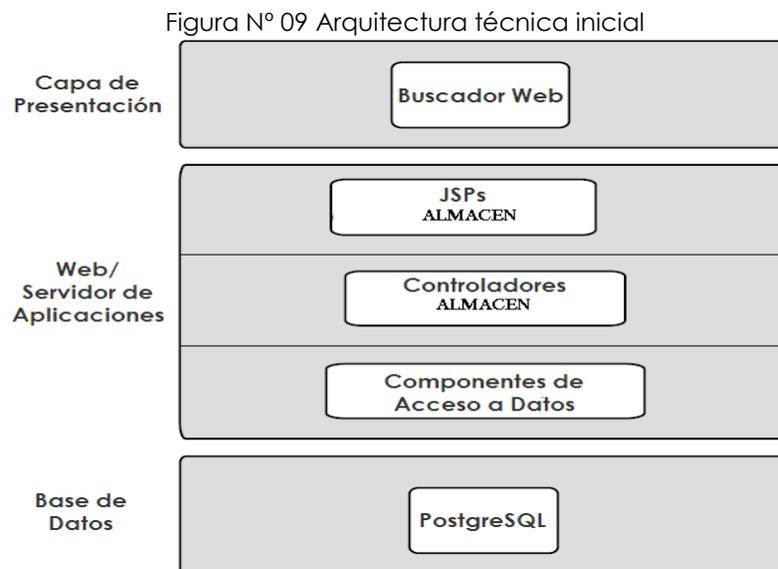
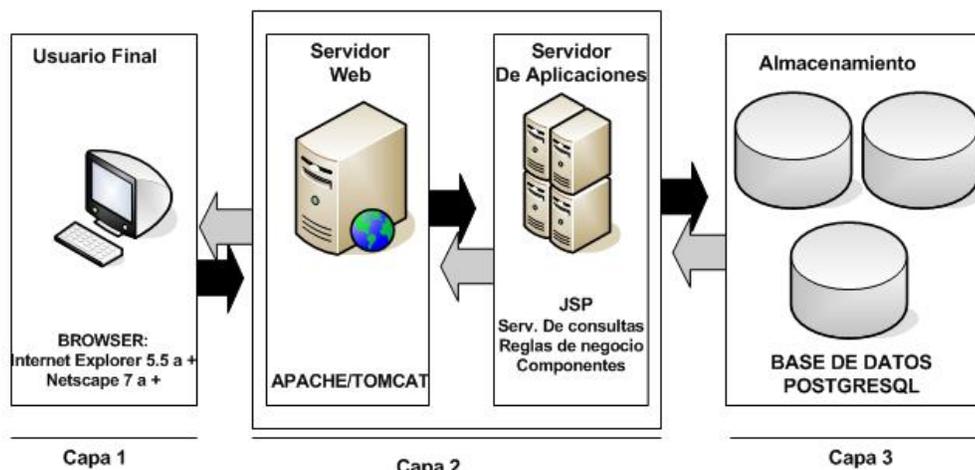


Figura N° 10 Arquitectura técnica de la Aplicación Web.



Fuente: adecuación de <http://oness.sourceforge.net/proyecto/>

Como se puede ver es una aplicación de tres capas, Para resolver los problemas se habían generalizado en el sector Cooperativo: una capa que servirá para guardar los datos (base de datos), una capa para centralizar la lógica de negocio y vista Web (modelo) y por último una interfaz gráfica que facilite al usuario el uso del sistema. Ver figura N°10 de Derecha a Izquierda.

La arquitectura utilizada por la presente, define tres capas bien diferenciadas, si bien gracias al soporte que proporciona una capa de modelo basada en Enterprise JavaBeans (EJB) posibilitase una arquitectura de cuatro capas estas funcionalidades están más orientadas a grandes sistemas o sistemas críticos como pueden ser replicación, trabajo en cluster, etc.

A.4 Plan de Alto Nivel.

En las reuniones con el cliente, se definieron una serie de Historias, de las cuales se separaron en partes (iteraciones) que son mencionadas en la siguiente tabla N° 20:

Tabla N° 19: Plan de alto nivel

N°	Historia de Usuario	Esfuerzo(Semana)
01	Emitir requerimiento de artículos	2
02	Otorgar requerimiento de artículos	2
03	Entregar requerimiento de artículos	1
04	Registrar Ingreso de Artículo	1
05	Control de acceso de usuarios.	3
06	Mantener Requerimiento de Artículo	2
07	Mantener Datos del Empleado	2
08	Mantener Usuario	3
09	Mantener Artículo	3
10	Mantener Área	2
11	Mantener Agencia	2
12	Mantener Cargo	2
13	Reporte de Stock Mínimo	1
14	Reporte de Inventario Valorado	1
15	Reporte de Inventario Físico	1
16	Reporte de Requerimiento de Artículo	2

B. FASE DE PLANIFICACIÓN

B.1 Historia de Usuario.

En esta fase es donde se estima tiempo de desarrollo de cada historia de usuario, riesgo y otros factores.

Tabla N° 20: Emitir requerimiento de artículos

Historia de Usuario	
Número: 1	Usuario: Jefes de Área
Nombre historia: Emitir requerimiento de artículos	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 2	Iteración asignada: 1
Programador responsable: SOLIS CASTRO, Javier	
<p>Descripción: El usuario (jefes de áreas), elige la opción "emitir requerimiento de artículo" del menú requerimiento artículo, el sistema presenta la interface "emitir requerimiento de artículo" con fecha, código de requerimiento y datos del solicitante. El usuario ingresa descripción del requerimiento y selecciona prioridad de requerimiento.</p>	

El usuario busca y selecciona un artículo, el sistema muestra detalles del artículo, luego ingresa la cantidad.
El usuario adiciona el artículo al requerimiento de artículo repitiendo los pasos anteriores hasta completar su requerimiento, finalmente, registra.
Observaciones:

Tabla N° 21: Otorgar requerimiento de artículos

Historia de Usuario	
Número: 2	Usuario: Administrador
Nombre historia: Otorgar requerimiento de artículos	
Prioridad en negocio: media	Riesgo en desarrollo: Alta
Puntos estimados: 2	Iteración asignada: 2
Programador responsable: SOLIS CASTRO, Javier	
Descripción:	
El usuario (Administrador), elige la opción “Otorgar Requerimiento Artículo” del menú Requerimiento artículo, el sistema presenta la interface “Otorgar Requerimiento Artículo”.	
El usuario ingresa el número de requerimiento y busca requerimiento de artículo, el sistema muestra datos del solicitante, descripción, prioridad, fecha de emisión y detalle de requerimiento.	
El usuario selecciona un detalle para modificar, el sistema muestra detalles del artículo seleccionado, luego ingresa la cantidad y adiciona al detalle del requerimiento del artículo.	
El usuario según el stock del requerimiento del artículo otorgar los artículos solicitados y finalmente registra.	
Observaciones:	

Tabla N° 22: Entregar requerimiento de artículos

Historia de Usuario	
Número: 3	Usuario: Almacenero
Nombre historia: Entregar requerimiento de artículos	
Prioridad en negocio: Media	Riesgo en desarrollo: Alta
Puntos estimados: 1	Iteración asignada: 2
Programador responsable: SOLIS CASTRO, Javier	
<p>Descripción: El usuario (almacenero) elige la opción "Registrar Salida Artículo" del menú requerimiento artículo, el sistema presenta la interface "Registrar Salida de Artículo" con fecha de requerimiento de salida de artículo. El usuario ingresa el número de requerimiento de artículo aprobado y busca, el sistema muestra fecha de aprobación de requerimiento, datos del solicitante, prioridad y urgencia del requerimiento y detalle del requerimiento solo aprobados. El usuario según el stock del artículo selecciona los artículos a ser entregados al solicitante del detalle de requerimiento de artículo aprobados finalmente registra.</p>	
Observaciones:	

Tabla N° 23: Registrar Ingreso de Artículo

Historia de Usuario	
Número: 4	Usuario: Almacenero
Nombre historia: Registrar Ingreso de Artículo	
Prioridad en negocio: Media	Riesgo en desarrollo: Media
Puntos estimados: 1	Iteración asignada: 2
Programador responsable: SOLIS CASTRO, Javier	
<p>Descripción: El usuario (Almacenero), elige la opción "Registrar Ingreso Artículo" del menú requerimiento artículo, el sistema presenta la interface "Registrar Ingreso Artículo" Con fecha de registro, código de ingreso de artículo. El usuario busca y selecciona un artículo, el sistema muestra detalles del artículo, luego ingresa la cantidad, precio unitario. El usuario adiciona el artículo al ingreso de artículos, repitiendo los pasos anteriores hasta completar el ingreso de artículos, finalmente, registra y actualiza el stock de artículos.</p>	

Tabla N° 24: Control de acceso de usuarios

Historia de Usuario	
Número: 5	Usuario: Empleados
Nombre historia: Control de acceso de usuarios	
Prioridad en negocio: Media	Riesgo en desarrollo: Alta
Puntos estimados: 3	Iteración asignada: 3
Programador responsable: SOLIS CASTRO, Javier	
<p>Descripción: El Usuario ingresa su login y contraseña, el sistema busca y compara si existe el login y contraseña en la base de datos de usuarios. Si los datos son correctos, el sistema muestra la pagina principal para comercializar producto tara, con las historias de usuario permitidas de acuerdo al actor directo o indirecto. Si los datos son incorrectos o no existen, el sistema muestra un mensaje de error de login o contraseña.</p>	
Observaciones:	

Tabla N° 25: Mantener Requerimiento de Artículo

Historia de Usuario	
Número: 6	Usuario: Administrador, Empleados
Nombre historia: Mantener Requerimiento de Artículo	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 3	Iteración asignada: 1
Programador responsable: SOLIS CASTRO, Javier	
<p>Descripción: El usuario (Administrador), elige la opción "Mantener Requerimiento Artículo" del menú Requerimiento artículo, el sistema presenta la interface "Mantener Requerimiento Artículo". El usuario ingresa el número de requerimiento y busca requerimiento de artículo, el sistema muestra datos del solicitante, descripción, prioridad, fecha de emisión y detalle de requerimiento. El usuario puede modificar descripción, prioridad del requerimiento. El usuario selecciona un detalle para modificar, el sistema muestra</p>	

<p>detalles del artículo seleccionado, luego ingresa la cantidad y adiciona al detalle.</p> <p>El usuario selecciona un detalle para eliminar del requerimiento, el sistema según el código elimina.</p> <p>El usuario busca y selecciona un artículo, el sistema muestra detalles del artículo, luego ingresa la cantidad.</p> <p>El usuario adiciona el artículo al requerimiento de artículo repitiendo los pasos anteriores hasta completar su requerimiento, finalmente, Actualiza todo el requerimiento.</p>
Observaciones:

Tabla N° 26: Mantener Datos del Empleado

Historia de Usuario	
Número: 7	Usuario: Administrador
Nombre historia: Mantener Datos del Empleado	
Prioridad en negocio: Media	Riesgo en desarrollo: Baja
Puntos estimados: 2	Iteración asignada: 4
Programador responsable: SOLIS CASTRO, Javier	
<p>Descripción:</p> <p>El usuario puede realizar operaciones como:</p> <p>Ingresar nuevo empleado, el usuario hace clic en el botón nuevo y el sistema muestra código de empleado, el usuario ingresa DNI, apellidos, nombres, dirección, email y teléfono, selecciona cargo, área finalmente registra en la base de datos.</p> <p>Mantener Datos del Empleado, el usuario busca y selecciona un empleado, el sistema muestra datos del empleado, cargo y agencia, luego ingresa los nuevos datos finalmente registra en la base de datos.</p>	
Observaciones:	

Tabla N° 27: Mantener Usuario

Historia de Usuario	
Número: 8	Usuario: Administrador
Nombre historia: Mantener Usuario	
Prioridad en negocio: Baja	Riesgo en desarrollo: Media
Puntos estimados: 3	Iteración asignada: 4
Programador responsable: SOLIS CASTRO, Javier	
<p>Descripción: El usuario puede realizar operaciones como: Crear cuenta, el usuario hace clic en el botón nuevo y el sistema muestra código de usuario, el usuario ingresa usuario y password y selecciona el estado (vigente, suspendido, inactivo) y selecciona los permisos y finalmente busca un empleado y le asigna la cuenta y registra en la base de datos. Mantener cuenta, el usuario busca y selecciona un usuario, el sistema muestra datos del empleado, permisos y cuenta, luego ingresa los nuevos datos finalmente registra en la base de datos.</p>	
Observaciones:	

Tabla N° 28: Mantener Artículo

Historia de Usuario	
Número: 9	Usuario: Almacenero, Administrador
Nombre historia: Mantener Artículo	
Prioridad en negocio: Baja	Riesgo en desarrollo: Media
Puntos estimados: 3	Iteración asignada: 5
Programador responsable: SOLIS CASTRO, Javier	
<p>Descripción: El usuario (Administrador), elige la opción "Mantener Artículo" del menú Administración, el sistema presenta la interface "Mantener Artículo". El usuario puede realizar operaciones como: Ingresar nuevo artículo, el usuario hace clic en el botón nuevo y el sistema muestra código de artículo, el usuario ingresa descripción de artículo, unidad, cantidad, precio unitario, stock mínimo, selecciona línea artículo y agencia y registra en la base de datos. Modificar área, el usuario busca y selecciona un artículo, el sistema muestra detalle del artículo, luego ingresa los nuevos datos finalmente registra en la base de datos.</p>	

Tabla N° 29: Mantener Área

Historia de Usuario	
Número: 10	Usuario: Administrador
Nombre historia: Mantener Área	
Prioridad en negocio: Baja	Riesgo en desarrollo: Media
Puntos estimados: 1	Iteración asignada: 5
Programador responsable: SOLIS CASTRO, Javier	
<p>Descripción: El usuario (Administrador), elige la opción "Mantener Área" del menú Administración, el sistema presenta la interface "Mantener Área". El usuario puede realizar operaciones como: Ingresar nuevo área, el usuario hace clic en el botón nuevo y el sistema muestra código de área, el usuario ingresa descripción de área y registra en la base de datos. Modificar área, el usuario busca y selecciona un área, el sistema muestra detalle de área, luego ingresa una nueva descripción y finalmente registra en la base de datos.</p>	

Tabla N° 30: Mantener Agencia

Historia de Usuario	
Número: 11	Usuario: Administrador
Nombre historia: Mantener Agencia	
Prioridad en negocio: Baja	Riesgo en desarrollo: Baja
Puntos estimados: 2	Iteración asignada: 5
Programador responsable: SOLIS CASTRO, Javier	
<p>Descripción: El usuario (Administrador), elige la opción "Mantener Agencia" del menú Administración, el sistema presenta la interface "Mantener Agencia". El usuario puede realizar operaciones como: Ingresar nueva agencia, el usuario hace clic en el botón nuevo y el sistema muestra código de Agencia, el usuario ingresa descripción de cargo y registra en la base de datos. Modificar Agencia, el usuario busca y selecciona una agencia, el sistema muestra detalle de Agencia y áreas pertenecientes, el usuario ingresa una nueva descripción, selecciona áreas para asignar o para eliminar y registra en la base de datos.</p>	

Tabla N° 31: Mantener Cargo

Historia de Usuario	
Número: 12	Usuario: Administrador
Nombre historia: Mantener Cargo	
Prioridad en negocio: Media	Riesgo en desarrollo: Baja
Puntos estimados: 1	Iteración asignada: 4
Programador responsable: SOLIS CASTRO, Javier	
<p>Descripción: El usuario (Administrador), elige la opción "Mantener Cargo" del menú Administración, el sistema presenta la interface "Mantener Cargo". El usuario puede realizar operaciones como: Ingresar nuevo cargo, el usuario hace clic en el botón nuevo y el sistema muestra código de cargo, el usuario ingresa descripción de cargo y registra en la base de datos. Modificar cargo, el usuario busca y selecciona un cargo, el sistema muestra detalle de cargo, luego ingresa una nueva descripción y finalmente registra en la base de datos.</p>	
Observaciones:	

B.2 Plan de Versión

En esta fase es donde se estima tiempo de desarrollo de cada historia de usuario, riesgo y otros factores.

Tabla N° 32: Plan de versión

N°	Historia de Usuario	Prioridad	Riesgo	Esfuerzo	iteración
01	Emitir requerimiento de artículos	Alta	Alta	2	1
02	Otorgar requerimiento de artículos	Media	Alta	2	2
03	Entregar requerimiento de artículos	Media	Alta	1	2
04	Registrar Ingreso de Artículo	Media	Media	1	2
05	Control de acceso de usuarios.	Media	Alta	3	3
06	Mantener Requerimiento de Artículo	Alta	Alta	3	1
07	Mantener Datos del Empleado	Media	Baja	2	4
08	Mantener Usuario	Baja	Media	3	4

09	Mantener Artículo	Baja	Media	3	5
10	Mantener Área	Baja	Baja	1	5
11	Mantener Agencia	Baja	Baja	2	5
12	Mantener Cargo	Media	Baja	1	4
13	Reporte de Stock Mínimo	Media	Baja	1	6
14	Reporte de Inventario Valorado	Baja	Media	1	6
15	Reporte de Inventario Físico	Baja	Baja	1	6
16	Reporte de Requerimiento de Artículo	Baja	Baja	2	6

C. FASE DE ITERACIÓN.

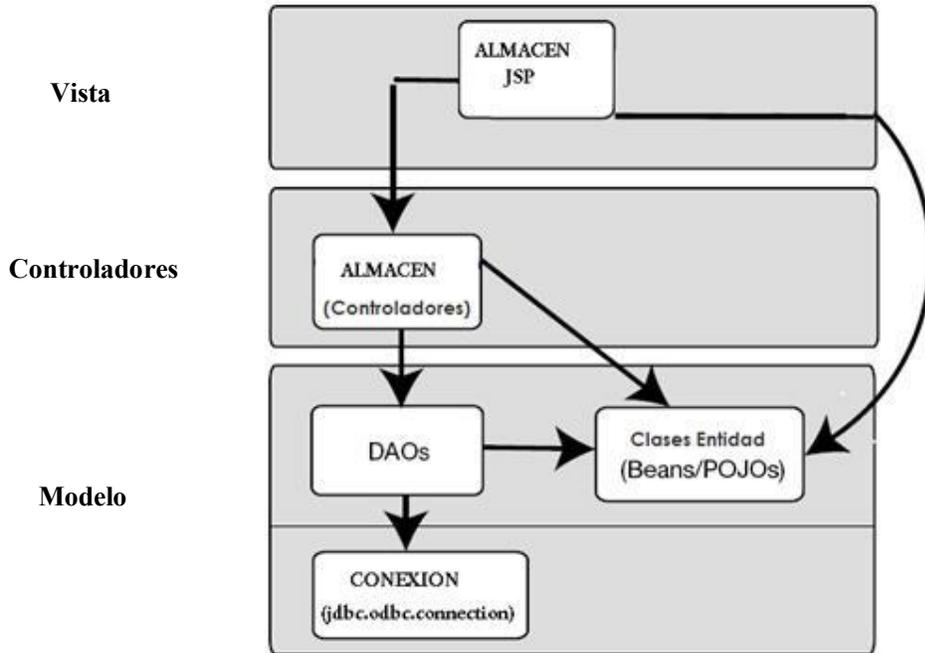
En esta se obtiene: arquitectura técnica, tareas de ingeniería; plan de iteración; casos de prueba de aceptación; GUI; tarjetas CRC; base de datos física; código fuente para clases entidad; pruebas unitarias; código fuente para tarea de ingeniería; reporte de pruebas unitarias; reporte de pruebas de integración y finalmente de aceptación.

C.1 Arquitectura Técnica

La figura N° 11, muestra los componentes para implementar una historia de usuario, el componente almacén (jsp) utiliza los JSPs (Java Server Pages) y los Beans (clases entidad) para mostrar la interfaz gráfica de usuario, los controladores usan los DAOs (Data access objects) y los Beans para realizar accesos a la base de datos, el componente JDBC(Java Database Connectivity) sirve para establecer una conexión entre la Base de Datos y la aplicación para realizar diferentes operaciones.

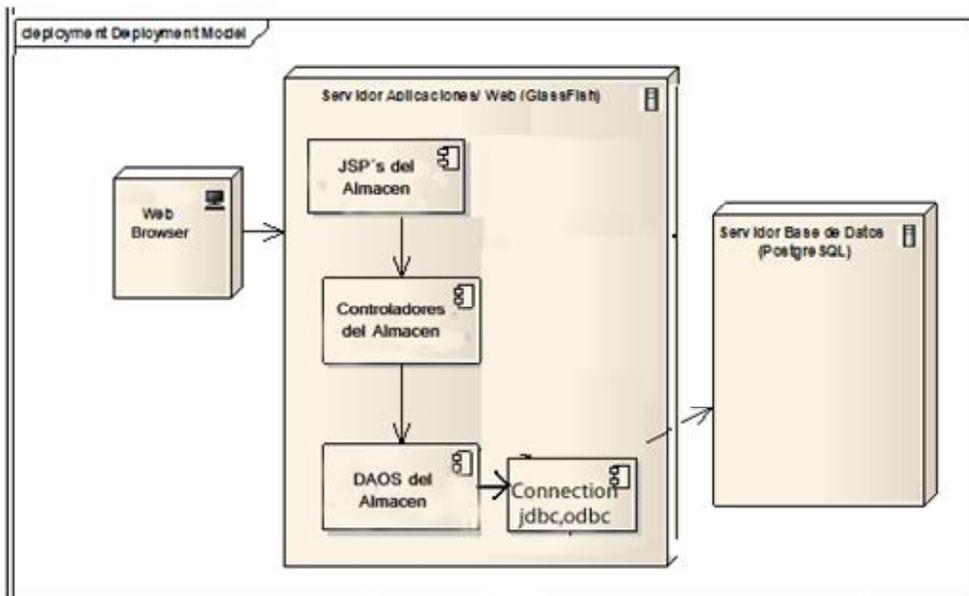
La figura N°12 del diagrama de despliegue, organiza los nodos de procesamiento con sus componentes utilizados; se observa así, la arquitectura conformada por el cliente, servidor Web GlassFish y servidor de base de datos postgresSQL.

Figura N° 11: Arquitectura técnica. Diagrama de componentes



Fuente: Figura adecuada de la exposición del Ing. Carlos G. Gavidia de (<http://www.slideshare.net/cptanalatrister/arquitectura-y-diseo-de-aplicaciones-java-ee>)

Figura N° 12: Arquitectura técnica. Diagrama de despliegue.



Fuente: Figura adecuada de la exposición del Ing. Carlos G. Gavidia de (<http://www.slideshare.net/cptanalatrister/arquitectura-y-diseo-de-aplicaciones-java-ee>)

C.1.1 Requerimientos de Arquitectura Técnica para la Aplicación Web

Al hacer una revisión del número posible de conexiones que se puedan hacer a la Aplicación Web desde una cooperativa (Señor de Maynay) que a enero del 2009, contaba con 17 trabajadores con 2 años y 3 meses de funcionamiento, siendo este uno de los extremos y la de la Cooperativa Santa María Magdalena con más de 200 trabajadores la que tiene 47 años de existencia, siendo la Cooperativa más grande del país en asociados y de Agencias o sucursales. Del mismo modo las cooperativas del país trabajan con el protocolo TCP/IP.

Por lo que se recomienda que el servidor Web, servidor de aplicaciones y de Base de datos sea el mismo hardware físicamente, ya que el requerimiento con proyección al futuro no es alto.

Tabla N° 33: Detalle hardware.

CAC Señor de Maynay	CAC Santa María Magdalena
Servidor PS HP ProLiant ML110 G6 X3430, 2 GB RAM, 2 Und. 250 GB LFF 300 W	Servidor de torre SATA de HP ProLiant ML150 G6 E5502 de núcleo a 1,86 GHz
Intel® Xeon® X3430 (4 Core, 2,40 GHz, 8 MB L3, 95W), Adaptador de servidor NC- Express Gigabit Ethernet.	Intel® Xeon® E5502 (2 núcleos, 1,86 GHz, 4 MB L3, 80W), Adaptador de servidor NC- Express Gigabit Ethernet.
Sin RAID	RAID SATA HP Smart Array B110i
Costo Aprox. 750.00 US\$	Costo Aprox. 940.00 US\$

C.2 Tareas de Ingeniería

Historia de Usuario: Emitir Requerimiento de Artículos

Tabla N° 34: Tarea de ingeniería. Autenticar usuario

Tarea de Ingeniería	
Número tarea de ingeniería: 1	Número historia de usuario: 01
Nombre tarea: Emitir requerimiento de artículos	
Tipo de tarea : Desarrollo	Puntos estimados: 2
Fecha inicio: 02/08/2009	Fecha fin: 03/09/2009
Programador responsable: SOLIS CASTRO, Javier	
Descripción: El usuario (jefes de áreas), elige la opción "emitir requerimiento de artículo" del menú requerimiento artículo, el sistema presenta la interface "emitir requerimiento de artículo" con fecha, código de requerimiento, datos del solicitante y prioridades.	

Tabla N° 35: Tarea de ingeniería. Agregar artículo

Tarea de Ingeniería	
Número tarea de ingeniería: 2	Número historia de usuario: 01
Nombre tarea: Agregar Artículo	
Tipo de tarea : Desarrollo	Puntos estimados: 2
Fecha inicio: 02/08/2009	Fecha fin: 03/09/2009
Programador responsable: SOLIS CASTRO, Javier	
Descripción: El usuario busca y selecciona un artículo, el sistema muestra detalles del artículo, luego ingresa la cantidad. El usuario adiciona el artículo al requerimiento de artículo repitiendo los pasos anteriores hasta completar su requerimiento.	

Tabla N° 36: Tarea de ingeniería. Cargar Stock de artículo

Tarea de Ingeniería	
Número tarea de ingeniería: 3	Número historia de usuario: 01
Nombre tarea: Cargar Stock de Artículo	
Tipo de tarea : Desarrollo	Puntos estimados: 2
Fecha inicio: 02/08/2009	Fecha fin: 03/09/2009
Programador responsable: SOLIS CASTRO, Javier	
Descripción: El sistema realiza la búsqueda del stock del artículo según el código del artículo y estado finalmente es encapsulado por la clase artículo.	

Tabla N° 37: Tarea de ingeniería. Cargar Línea Artículo

Tarea de Ingeniería	
Número tarea de ingeniería: 4	Número historia de usuario: 01
Nombre tarea: Cargar Línea Artículo	
Tipo de tarea : Desarrollo	Puntos estimados: 2
Fecha inicio: 02/08/2009	Fecha fin: 03/09/2009
Programador responsable: SOLIS CASTRO, Javier	
Descripción: El sistema realiza la búsqueda de línea de artículo según el código del artículo y estado finalmente es encapsulado por la clase artículo.	

Tabla N° 38: Tarea de ingeniería. Grabar Requerimiento Artículo

Tarea de Ingeniería	
Número tarea de ingeniería: 5	Número historia de usuario: 01
Nombre tarea: Grabar Requerimiento Artículo	
Tipo de tarea : Desarrollo	Puntos estimados: 2
Fecha inicio: 02/08/2009	Fecha fin: 03/09/2009
Programador responsable: SOLIS CASTRO, Javier	
Descripción: El usuario selecciona prioridad, ingresa urgencia de requerimiento de artículo finalmente registra en la base de datos y muestra un mensaje de la operación realizada.	

Tabla N° 39: Tarea de ingeniería. Autenticar usuario

Tarea de Ingeniería	
Número tarea de ingeniería: 1	Número historia de usuario: 03
Nombre tarea: Autenticar Usuario	
Tipo de tarea : Desarrollo	Puntos estimados: 1
Fecha inicio: 02/08/2009	Fecha fin: 03/09/2009
Programador responsable: SOLIS CASTRO, Javier	
Descripción: El Usuario ingresa su login y contraseña, el sistema busca y compara si existe el login y contraseña en la base de datos de usuarios. Si los datos son correctos, el sistema muestra la pagina principal de almacén de producto, con las historias de usuario permitidas de acuerdo privilegios dadas al usuario. Si los datos son incorrectos o no existen, el sistema muestra un mensaje de error de login o contraseña.	

Tabla N° 40: Tarea de ingeniería. Cargar Permisos

Tarea de Ingeniería	
Número tarea de ingeniería: 2	Número historia de usuario: 03
Nombre tarea: Cargar Permisos	
Tipo de tarea : Desarrollo	Puntos estimados: 1
Fecha inicio: 02/08/2009	Fecha fin: 03/09/2009
Programador responsable: SOLIS CASTRO, Javier	
<p>Descripción: Si el usuario existe en la base de datos, el sistema verificado el estado (habilitado, suspendido) si está habilitado, el sistema buscar los permisos asignados en la base de datos de permisos, el sistema muestra la pagina principal del sistema con todos los permisos. Si el usuario no tiene permisos el sistema muestra la pagina principal sin menú.</p>	

Tabla N° 41: Tarea de ingeniería. Buscar Empleado.

Tarea de Ingeniería	
Número tarea de ingeniería: 3	Número historia de usuario: 03
Nombre tarea: Buscar Empleado	
Tipo de tarea : Desarrollo	Puntos estimados: 1
Fecha inicio: 02/08/2009	Fecha fin: 03/09/2009
Programador responsable: SOLIS CASTRO, Javier	
<p>Descripción: El sistema realiza la búsqueda del empleado según el código del usuario en la base de datos de empleados y muestra en la página principal del sistema.</p>	

Tabla N° 42: Tarea de ingeniería. Cargar Cargo

Tarea de Ingeniería	
Número tarea de ingeniería: 4	Número historia de usuario: 03
Nombre tarea: Cargar Cargo	
Tipo de tarea : Desarrollo	Puntos estimados: 1
Fecha inicio: 02/08/2009	Fecha fin: 03/09/2009
Programador responsable: SOLIS CASTRO, Javier	
<p>Descripción: El sistema realiza la búsqueda de cargo según el código del usuario en la base de datos y muestra en la página principal del sistema.</p>	

Tabla N° 43: Tarea de ingeniería. Cargar Agencia.

Tarea de Ingeniería	
Número tarea de ingeniería: 5	Número historia de usuario: 03
Nombre tarea: Cargar Agencia	
Tipo de tarea : Desarrollo	Puntos estimados: 1
Fecha inicio: 02/08/2009	Fecha fin: 03/09/2009
Programador responsable: SOLIS CASTRO, Javier	
Descripción: El sistema realiza la búsqueda de Agencia según el código del usuario, estado y código de empleado en la base de datos y muestra en la página principal del sistema.	

Tabla N° 44: Tarea de ingeniería. Cargar Área.

Tarea de Ingeniería	
Número tarea de ingeniería: 6	Número historia de usuario: 03
Nombre tarea: Cargar Área	
Tipo de tarea : Desarrollo	Puntos estimados: 1
Fecha inicio: 02/08/2009	Fecha fin: 03/09/2009
Programador responsable: SOLIS CASTRO, Javier	
Descripción: El sistema realiza la búsqueda de Área según el código del usuario, estado y código de empleado en la base de datos y muestra en la página principal del sistema.	

C.3 Plan de Iteración

Tabla N° 45: Plan de iteración (Primera)

N° HU	N° TI	Fecha Inicio	Fecha Fin	Programador
01	1	01/09/2009	03/09/2009	SOLIS CASTRO, Javier
01	2	01/09/2009	03/09/2009	SOLIS CASTRO, Javier
01	3	03/09/2009	05/09/2009	SOLIS CASTRO, Javier
01	4	03/09/2009	05/09/2009	SOLIS CASTRO, Javier
01	5	05/09/2009	07/09/2009	SOLIS CASTRO, Javier
03	1	05/09/2009	07/09/2009	SOLIS CASTRO, Javier
03	2	07/09/2009	09/09/2009	SOLIS CASTRO, Javier
03	3	07/09/2009	09/09/2009	SOLIS CASTRO, Javier
03	4	07/09/2009	09/09/2009	SOLIS CASTRO, Javier
03	4	07/09/2009	09/09/2009	SOLIS CASTRO, Javier

C.4 Artefacto Interfaz Gráfica de Usuario (GUI).

Diseño de interfaces con la participación del cliente. Las interfaces para las dos historias de usuario se muestran en las figuras 13, 14 y 15.

Figura N° 13:
GUI de la Historia de usuario: Emitir Requerimiento de Artículo.

HOJA DE REQUERIMIENTO DE ARTICULOS							
HOME CERRAR SESION							
SOLICITANTE							
AGENCIA :	San Francisco	AREA USUARIA:	Sistemas	DESCRIPCION:	Material Para Exámenes		
NOMBRE SOLICITANTE:	SOLIS CASTRO, Javier			CARGO SOLICITANTE:	Jefe de Contabilidad		
N° REQUERIMIENTO:	003	FECHA DE REQUERIMIENTO:	25/01/2010	PRIORIDAD:	Segunda		
BUSCAR ARTICULO							
ITEM	CODIGO	DESCRIPCION	UNIDAD	STOCK			
1							
CANTIDAD :	<input type="text"/>	<input type="button" value="Agregar Articulo"/>	<input type="button" value="Buscar Articulo"/>				
DETALLE DE REQUEMIENTO							
ITEM	CODIGO	DESCRIPCION	UNIDAD	CANTIDAD	PRECIO UNITARIO(\$/,)	SUB TOTAL(\$/,)	ESTADO
1	1	Tinta Seca	Unidad	4.0	1.2	4.8	pendiente
2	2	A4-75gr	Millar	3.0	14.5	43.5	pendiente
						TOTAL(\$/,)	48.3
<input type="button" value="Guardar Requerimiento"/> <input type="button" value="Imprimir"/> <input type="button" value="Cancelar"/>							

Figura N° 14:
GUI de la Historia de usuario: Control de Acceso

Usuario

Contraseña

Figura N° 15:
 GUI de la Historia de usuario: GUI N° 3.6: Menú Principal del Sistema



C.5 Tarjetas Clase, Responsabilidad y Colaboración (CRC)

Tabla N° 46: Tarjeta CRC. Requerimiento de Artículo.

Clase: RequerimientoArticulo	
Responsabilidades: Verificar código usuario, código requerimiento y fecha de requerimiento. Mostrar datos del solicitante, cargo, área, agencia, fecha de solicitud, código de requerimiento.	Colaboradores: Empleado

Tabla N° 47: Tarjeta CRC. Estado de Requerimiento de Artículo.

Clase: EstadoRequerimientoArticulo	
Responsabilidades: Busca código de requerimiento, estado, fecha de Requerimiento de Artículo.	Colaboradores: Requerimiento Artículo

Tabla N° 48: Tarjeta CRC. Detalle de Requerimiento de Artículo.

Clase: DetalleRequerimientoArticulo	
Responsabilidades: Verifica cantidad y stock de artículo. Mostrar código, descripción, unidad, precio unitario, sub total y total de detalle de requerimiento de artículo.	Colaboradores: Artículo RequerimientoArticulo

Tabla N° 49: Tarjeta CRC. Artículo

Clase: Artículo	
Responsabilidades: Buscar código, descripción, unidad y código de Línea Artículo. Enviar código, descripción, unidad, línea Artículo y stock Artículo a Detalle Requerimiento Artículo.	Colaboradores: LineaArticulo

Tabla N° 50: Tarjeta CRC. Stock Artículo

Clase: StockArticulo	
Responsabilidades: Buscar código, stock, cantidad de Artículo. Enviar código, stock, cantidad y estado de Stock Artículo a Artículo.	Colaboradores: Artículo

Tabla N° 51: Tarjeta CRC. Línea Artículo

Clase: LineaArticulo	
Responsabilidades: Buscar código y denominación de Línea de Artículo. Enviar código, denominación Línea de Artículo a Artículo.	Colaboradores:

Tabla N° 52: Tarjeta CRC. Usuario.

Clase: Usuario	
Responsabilidades: Verificar login, contraseña y estado. Mostrar apellido, nombre, dirección, teléfono y email del empleado.	Colaboradores: Empleado Permiso

Tabla N° 53: Tarjeta CRC. Permiso

Clase: Permiso	
Responsabilidades: Buscar código, denominación y estado de Roles del usuario. Mostrar Permisos del Usuario.	Colaboradores: Empleado Permiso

Tabla N° 54: Tarjeta CRC. Empleado

Clase: Empleado	
Responsabilidades: Buscar apellido, nombre, dirección, DNI de empleado. Enviar apellido, razón social, dirección, DNI empleado a Usuario.	Colaboradores: Usuario

Tabla N° 55: Tarjeta CRC. Cargo

Clase: Cargo	
Responsabilidades: Buscar código y denominación de Cargo. Enviar código, denominación y estado de Cargo a Empleado.	Colaboradores: Empleado Usuario

Tabla N° 56: Tarjeta CRC. Área

Clase: Área	
Responsabilidades: Buscar código y denominación de Área. Enviar código, denominación y estado de Área a Empleado.	Colaboradores: Empleado Usuario

Tabla N° 57: Tarjeta CRC. Agencia.

Clase: Agencia	
Responsabilidades: Buscar código y denominación de Agencia. Enviar código, denominación y estado de Agencia a Empleado.	Colaboradores: Empleado Usuario

D. FASE DE PRODUCCIÓN

D.1 Diseño y Base de Datos Física

Ver Anexo C y B.

D.2 Código Fuente Para Clases Entidad

La figura N°16: muestra el código fuente para las clases entidad; de la historia de usuario "Control de Acceso de Usuario"; para la clase "Usuario"; declarando atributos; el constructor y los métodos "get" y "set". En java estas clases también se denominan "beans" o "POJOS" (clase primitivas).

Figura N° 16: Código fuente de la clase entidad Usuario.

```

1 package org.almacen.bean;
2 import java.io.Serializable;
3 import java.util.Date;
4 import java.util.List;
5 public class Usuario implements Serializable {
6     private int codigo;
7     private String password;
8     private String login;
9     private Date fechaCreacion;
10    private boolean estado;
11    private //relacion de clases
12    int codigoEmpleado;
13    private Empleado empleado;
14    private List lispermiso;
15    public int getCodigo() {
16        return codigo;
17    }
18    public void setCodigo(int codigo) {
19        this.codigo = codigo;
20    }
21    public String getPassword() {
22        return password;
23    }
24    public void setPassword(String password) {
25        this.password = password;
26    }
27    public String getLogin() {
28        return login;
29    }
30

```

D3. Diseño lógico de la aplicación Web para almacén de cooperativas de ahorro y crédito del Perú.

Ver Anexo A

E. FASE DE MANTENIMIENTO

E.1 Prueba Unitaria

La prueba unitaria se realiza con el framework "JUnit", para probar los métodos de las clases control, usando los métodos de JUnit "testProcessRequest", "testDoGet", "testDoPost", "testGetServletInfo", "tearDown", "setUp", "setUpClass", "tearDownClass". La salida de la prueba se verifica con "AssertTrue" que prueba el valor de verdad y "AssertEquals" que prueba la igualdad de dos expresiones.

La figura N°17: muestra la implementación usando el Servlets de Java para la autenticación de usuario y los permisos autorizados al usuario.

Figura N° 17: Código fuente para el historia de usuario. Control de Acceso de Usuarios.

```
17 public class ControlSeguridad extends HttpServlet {
18     protected void processRequest(HttpServletRequest request, HttpServletResponse response)
19     throws ServletException, IOException {response.setContentType("text/html;charset=
20         String loggin = request.getParameter("loginn");
21         String password = request.getParameter("contrasenia");
22         ServletContext ctx = this.getServletConfig().getServletContext();
23         try {
24             DAOSeguridad ds = new DAOSeguridad();
25             List permiso = null;
26             List lisReq = new LinkedList();
27             List lisReporte = new LinkedList();
28             List lisAdmin = new LinkedList();
29             Agencia_Area agArea = null;
30             Conexion cn = new Conexion();
31             ds.connection(cn.conectar());
32             Usuario usuario = ds.autenticarUsuario(loggin, password);
33             if (usuario.getPassword().equalsIgnoreCase(password)) {
34                 if (usuario.isEstado()) {
35                     agArea = ds.buscarAgenciaArea(usuario.getCodigoEmpleado());
36                     ctx.setAttribute("usuario", usuario);
37                     ctx.setAttribute("areaAgencia", agArea);
38                     ctx.setAttribute("autorizado", "Si");
39                     permiso = ds.permiso(usuario.getCodigo());
40                     ds.cerrar();
41                     for (int i = 0; i < permiso.size(); i++) {
42                         Menu m = (Menu) permiso.get(i);
43                         if (m.getOrden() == 1) {
44                             lisReq.add(m);
45                         }
46                     }
47                 }
48             }
49         }
50     }
51 }
```

E.1.1 Reporte de Pruebas Unitarias.

Tabla N° 58: Reporte de pruebas unitarias. Primera iteración.

N° HU	N° TI	Nombre Clase Control	Resultado
05	1	ControlSeguridad (autentica usuario)	Satisfactorio
	2	SCuentaUsuario	Satisfactorio
	3	SEmpleado	Satisfactorio
01	1	SRequerimientoArticulo	Satisfactorio
	2	ControlArticulo	Satisfactorio

E.1.2 Reporte de Prueba Integración.

Tabla N° 59: Reporte de prueba integración. Primera iteración.

N° HU	N° TI	Nombre Tarea de ingeniería	Resultado
05	1	Autenticar usuario	Satisfactorio
	1	Buscar Empleado	Satisfactorio
	1	Buscar Cargo	Satisfactorio
	1	Buscar Área	Satisfactorio
	1	Buscar Rol	Satisfactorio
01	1	Buscar Artículo	Satisfactorio
		Buscar Stock Artículo	Satisfactorio
		Buscar Línea Artículo	Satisfactorio
	2	Agregar detalle Requerimiento de Artículo.	Satisfactorio
	3	Guardar Requerimiento Artículo	Satisfactorio
		Guardar Estado Requerimiento Artículo	Satisfactorio

E.2 Reporte de Prueba de Aceptación.

Tabla N° 60: Reporte de prueba de aceptación. Mantener Empleado.

N° CASO PRUEBA	1
Propósito	Verificar datos del empleado seleccionado.
ACTIVIDAD	
Inicialización	Seleccionar empleado
Descripción de de de entrada	El usuario hace clic en la opción "Mantener Empleado" ingresa cuenta de usuario o deja en blanco y selecciona un empleado.
RESULTADOS	
Esperados	83364305, Ronald, Gomez, Arostegui, Cuzco 376 ,966674245 . Jefe de RRHH, Ica, RRHH
Reales	83364305, Ronald, Gomez, Arostegui, Cuzco 376 ,966674245 . Jefe de RRHH, Ica, RRHH

Tabla N° 61: Reporte de prueba de aceptación. Mantener Cuenta de Usuario.

N° CASO PRUEBA	2
Propósito	Verificar datos de cuenta de usuario seleccionado.
ACTIVIDAD	
Inicialización	Seleccionar cuenta de usuario.
Descripción de los datos de entrada	El usuario en la página "Mantener Cuenta Usuario" ingresa cuenta de usuario del empleado o deja en blanco y selecciona una cuenta.
RESULTADOS	
Esperados	KRobles.Kathia1234.Activo.Robles Huaman, Kathia. 83364305. San Martín 104. 966881213. kathiaroblesh@hotmail.com Jefe de Operaciones.
Reales	KRobles.Kathia1234.Activo.Robles Huaman, Kathia. 83364305. San Martín 104. 966881213. kathiaroblesh@hotmail.com Jefe de Operaciones.

Tabla N° 62: Reporte de prueba de aceptación. Mantener Artículo.

N° CASO PRUEBA	3
Propósito	Verificar datos del Artículo seleccionado.
ACTIVIDAD	
Inicialización	Seleccionar Artículo
Descripción de los datos de entrada	El usuario en la página "Mantener Artículo" ingresa el nombre del artículo o deja en blanco y selecciona un Artículo.
RESULTADOS	
Esperados	Bond A4-75gr, Millar, 7.0, 14.5, 12.0, Papel, Ayacucho-Central.
Reales	Bond A4-75gr, Millar, 7.0, 14.5, 12.0, Papel, Ayacucho-Central.

4.8 Diario de Actividades

El diario de actividades son las anotaciones de las reuniones que tuvimos con el cliente a lo largo del desarrollo del proyecto.

Tabla N° 63: Diario de Actividades Aplicación Web para Almacén de Cooperativas De Ahorro Y Crédito Del Perú.

Fecha (dd/mes) 2009	Actividad Realizada	Tiempo Dedicado (en horas)	Observaciones
17/10	Primera reunión con el cliente.	3	No sabemos lo que hacemos y es frustrante.
21/10	Revisión con el grupo de las hojas de cliente	2	Sigue complicado.
23/10	Reunión con el grupo para hacer y acordar las historias de usuario del cliente.	4	Empezamos a enterarnos.
27/10	Reunión con el cliente para ver las historias de usuario.	2	Productiva
30/10	Rehacer las historias de usuario pedidas por el cliente.	2	Poco productiva
03/11	Reunión con el cliente para ver las historias de usuario y hacer un prototipo de la interfaz.	4	Nos entendemos
07/11	Reorganización de historias y hacer las interfaz con el grupo.	4	Productivo
11/11	Reunión con el cliente para ver la interfaz y si esta de acuerdo con ellas.	2	
17/11	Generación de prototipos y presentación de las interfaces y las historias de usuario.	4	Nos entendemos
24/11	Reunión con el cliente para	3	

	comenzar a hacer la base de datos.		
27/11	Reunión con el grupo para hacer la base de datos.	2	Superada satisfactoriamente
05/12	Reunión con el cliente para ver la base de datos y ver si esta bien.	3	Productiva
06/12	Trabajo en grupo y revisión de la DB y hacer el informe conforme el trabajo.	4	Productiva
08/12	Corrección de la base de datos con el cliente, generación de prototipos y presentación del informe.	3	Productiva
11/12	Reunión con el grupo para separar la base de datos según las historias de usuario.	1	Productiva
15/12	Cita con el cliente para la corrección con el cliente de las historias de usuario con sus respectivas bases de datos.	4	Productiva
17/12	Reunión del grupo para hacer el informe conforme lo manda el cliente y avanzar el código.	3	Productiva
23/12	Revisión del informe y del código por el cliente.	2	Productiva
28/12	Reunión con el grupo de XP para ver la corrección del informe y del código.	3	Productiva

Fuente: Elaboración propia

CAPITULO V

CONCLUSIONES Y RECOMENDACIONES

5.1 CONCLUSIONES

Con la implementación de la Aplicación Web que apoyará la gestión de la información en el almacén de las Cooperativas de ahorro y Crédito del Perú, se dan cumplimiento a los objetivos propuestos inicialmente. El resultado será un producto informático de alta calidad, cuyo objetivo primordial es favorecer los procesos de realización de las solicitudes de materiales; útiles de escritorio; etc. de las diferentes áreas al Almacén y la presentación de los mismos como documentos de conformidad luego de su recepción. A través de esta Aplicación Web se logrará una mayor confiabilidad y rapidez en el flujo de información por los canales y controles necesarios para que éstos se realicen con la mayor eficiencia y seriedad posible, provocando que el trabajo en las áreas o departamentos involucrados se lleve a cabo con una óptima calidad y eficiencia.

Mientras se implementaba el sistema se arribaron a las conclusiones siguientes:

- La presente aplicación que controla los procesos de Entradas y Salidas del Almacén para las COOPAC'S, la cual funciona sobre la arquitectura Web, eleva al área de almacén al nivel de otras áreas de trasendentales como: ventas, logística, operaciones; pues facilita la administrando y control de las entradas y salidas de almacén proporcionando información de control y táctica a la gerencia, permitiendo mejores resultados financieros.
- En el proceso de desarrollo software la funcionalidad fue incluida

al describir las historias de usuario para el almacén, en la cual se detallan los requerimientos que esta aplicación cumple para cada usuario de la aplicación Web.

- Toda metodología debe ser adaptada al contexto del proyecto (recursos técnicos y humanos; tiempo de desarrollo; tipo de sistema; envergadura; etc.). Las metodologías ágiles ofrecen una solución a medida para una gran cantidad de proyectos pequeños y con un alto grado de predictibilidad. Una cualidad en una metodología ágil es su sencillez, tanto en su aprendizaje como en su aplicación. Sin embargo hay que tener presente una serie de restricciones para su aplicación tales como: están dirigidas a equipos pequeños o medianos y el entorno físico debe ser un ambiente que permita la comunicación y colaboración entre todos los miembros del equipo durante todo el tiempo.

5.2 RECOMENDACIONES

- Reglamentar el uso de la Aplicación Web, así como capacitar al personal que interactuará con la aplicación con la finalidad de que este se convierta en una herramienta de control de las E/S de materiales.
- Se recomienda investigar a mayor detalle la migración de motores de bases de datos de distintos propietarios, o la creación de un Aplicativo que cumpla esta función ya que sería de mucha utilidad para esta aplicación y otras que existen en el mercado.
- Cuando las COOPAC's tengan necesidades mayores en almacenamiento, velocidad, etc. Estas puede optar por EnterpriseDB PostgreSQL, versión que reimplementa características tales como tipos de datos, disparadores, opiniones y cursores que copian el comportamiento de Oracle, perceptiblemente más caro.
- El uso de esta aplicación puede extenderse a otras Entidades como cajas municipales y rurales, puesto que el servicio es el mismo; y por ende el funcionamiento de su almacén; también tienen los mismos planes de expansión, etc. Por lo que podrían beneficiarse con la administración y control de las entradas y salidas de material y proporcionar información de control y táctica.

5.3 REFERENCIAS BIBLIOGRÁFICAS

- 1) Alianza Cooperativa Internacional para las Américas "El Cooperativismo en el mundo" tomado en <http://www.aciamericas.coop/spip.php?article44>.
- 2) Hugo Pardo Kuklinski, de la tesis doctoral: "Un modelo de aplicación Web institucional universitaria", México, 2006 pp: 28.
- 3) Powell, En cuanto a publicación de libros, el libro "Web Site Engineering: beyond Web Page Design", Australia, 2005.
- 4) Tomado de:
<http://www.biblioteca.uson.mx/digital/tesis/docs/17827/Capitulo4.pdf>.
- 5) Tompkins, 1988. Citado por: <<http://www.gestiopolis.com/canales/gerencial/articulos/32/almarsatcl.htm>>
- 6) Tomado de: <<http://www.gestiopolis.com/canales/gerencial/articulos/32/almarsatcl.htm>>
- 7) Yoan D. Puig, "Sistema de Almacén central" Tesis para optar el título de Ingeniero Informático Universidad de Holguín Oscar Lucero Moya -CUBA. pp.06.
- 8) Tomado de: alarcos.inf-cr.uclm.es/doc/pgsi/doc/teo/8/pgsi-t8ej.pdf.
- 9) Tomado de: < <http://chico.inf-cr.uclm.es/cache/>>
- 10) Tomado de: <<http://www.usfx.info/edif/orgmet/administraciondealmacenes.pdf>>
- 11) ISO/IEC TR 9126-2:2003. Software engineering. Product quality Part 2. External metrics. 2003.
- 12) Cesar Liza Ávila Modelando con UML, principios y aplicaciones. Ed. RJ SRL. Perú. 2001.
- 13) Matsukawa M. Sergio. Análisis y Diseño Orientado a Objetos con UML y Rational Rose. Edit. Macro, Lima 2004.
- 14) Luis Cancho C. "Aprende XP en 24 horas" Perú. 2007.

- 15) Revista digital Universidad del Valle - Colombia del Departamento de Procesos, Información, Contabilidad y Finanzas DPIC.
<<http://eisc.univalle.edu.co/materias/www/material/lecturas/xp.pdf>>
- 16) Tomado de: <<http://www.extremeprogramming.org/>>.
- 17) Tomado de:
<<http://users.dsic.upv.es/asignaturas/facultad/lsi/ejemploxp/>>
- 18) Kent Beck, Martin Fowler: Planning "Extreme Programming Explored", Addison-Wesley Editorial "Pearson Education, 1999", Addison Wesley, 2000.
- 19) Ron Jeffries, Ann Anderson, Chet Hendricrickson, "Extreme Programming Installed", Editorial "Pearson Education, 1999", Addison Wesley, 2000.
- 20) William C. Wake, "Extreme Programming Explored", Editorial Pearson Education, Addison Wesley, Copyright 2000, All Rights Reserved.
- 21) Revista digital Universidad de Chile - Departamento de Ciencias de la Computación DECC.
<<http://www.dcc.uchile.cl/~lmateu/Java/Apuntes/java>>
- 22) Tomado de: <<http://blog.taragana.com/index.php/archive/five-reasons-to-use-postgresql/es>>
- 23) Leoncio Berrocal Ll. "Sistema de Información Web para el Turismo Receptivo de Ayacucho bajo el modelo Network Marketing" " Tesis para optar el título de Ingeniero Informático.

GLOSARIO

Aplicación. Agrupación lógica de funcionalidades bajo un conjunto de programas informáticos que representan una unidad estructural.

Browser. Navegador o navegador web (del inglés, web browser) es un programa que permite visualizar la información que contiene una página web (ya esté esta alojada en un servidor dentro de la World Wide Web o en uno local).

Clase. Son declaraciones o abstracciones de objetos, lo que significa, que una clase es la definición de un objeto.

Comité técnico de adquisiciones. Área creada por algunas Cooperativas en el Perú integrada por directivos, gerencia y personal administrativo.

Confidencialidad. Conjunto de métodos y técnicas que garantizan que el acceso a la información solo se produce por los usuarios autorizados para ello.

Consistencia de Datos. Uniformidad de significantes y presentación de las informaciones circulantes, lo que se denomina un lenguaje común de datos.

COOPAC. Es usada el Perú para abreviar Cooperativa de ahorro y crédito

CRC. (Clase – Responsabilidad – Colaboración) Forma de diseñar las clases, de manera simple, en la fase de planificación desarrollada entre cliente desarrolladores.

Diagrama de Flujo. Representación gráfica de la circulación de informaciones entre actividades.

Entidad. Elementos físicos o lógicos, reales o abstractos que intervienen en la organización, formando parte de ella o de su dinámica funcional.

Estabilidad. Garantía en el funcionamiento continuado.

FENACREP. Federación Nacional de Cooperativas de ahorro y crédito del Perú.

GUI. Interfaz Gráfica de Usuario, conocida también como GUI, es un programa informático que actúa de interfaz de usuario utilizando un conjunto de imágenes y objetos gráficos para representar la información y acciones disponibles en la interfaz.

HTML. (Hypertext Markup Language): Lenguaje usado para escribir documentos para servidores World Wide Web.

HTTP. (Hypertext Transport Protocol): Es un conjunto de especificaciones para el intercambio de ficheros (texto, gráfico, imagen, sonido, vídeo) en la Web.

IEEE. Corresponde a las siglas de Instituto de Ingenieros Electricistas y Electrónicos. Es una asociación técnico-profesional mundial dedicada a la estandarización, entre otras cosas.

Implementación. Puesta en marcha la construcción de una aplicación informática.

Interface. Interfaz. Conexión entre dos dispositivos de hardware; entre dos aplicaciones o entre un usuario y una aplicación que facilita el intercambio de datos.

ISO/IEC. Estándar internacional para la evaluación del Software; supervisado por el proyecto SQuaRE, ISO 25000:2005, el cuál sigue los mismos conceptos.

JUnit. Conjunto de bibliotecas creadas por Erich Gamma y Kent Beck que son utilizadas en programación para hacer pruebas unitarias de aplicaciones Java. Es JUnit es un conjunto de clases (framework)

Macromedia Dreamweaver. Herramienta para el desarrollo de aplicaciones Web.

Modelo Entidad-Relación. Método de representación de la información basado en la definición de un conjunto de objetos de la organización y la relación entre estos.

Módulo. En programación, un módulo es un software que agrupa un conjunto de subprogramas y estructuras de datos. Los módulos son unidades que pueden ser compiladas por separado y los hace reusable y permite que múltiples programadores trabajen en diferentes módulos en forma simultánea, produciendo ahorro en los tiempos de desarrollo.

MySQL. Base de Datos relacional multiplataforma de código abierto, muy popular en aplicaciones Web.

Novell NetWare. Sistema operativo de red. Es una de las plataformas de servicio más fiable para ofrecer acceso seguro y continuado a la red y los recursos de información; sobre todo en cuanto a servidores de archivos.

Objeto. Se define como la unidad que en tiempo de ejecución realiza las tareas de un programa. También a un nivel más básico, se define como la instancia de una clase.

Open Bank Life (OBL). Sistema transaccional desarrollado en Canadá por SISDE. Se adquirió por la asociación de cooperativas SISNIC S.A.C. Este sistema fue muy usado el año 2003 y 2004.

Open Source. Código abierto ó código libre. Software que distribuye de forma libre su código fuente, de forma que los desarrolladores pueden hacer variaciones, mejoras ó reutilizarlo en otras aplicaciones.

PC. Una computadora personal u ordenador personal, también conocido como PC (sigla en inglés de personal computer).

PHP. Es una ambiente *script* del lado del servidor que permite crear y ejecutar aplicaciones Web dinámicas e interactivas. Se caracteriza por ser una tecnología *open source* y por su fácil desarrollo y mantenimiento.

Sistema. "Ser" complejo y organizado, compuesto por un conjunto de partes o componentes orientado a la producción de algo considerado un elemento unitario.

Sistemas de Información. Conjunto de componentes físicos, lógicos, estructurales, logísticos y organizacionales orientados a la captura, almacenamiento, tratamiento y explotación de la información.

Sistema Operativo. Conjunto de programas que representan la vía de comunicación entre los programas de usuarios y utilidades con el hardware.

Software. Programas de sistema, utilerías o aplicaciones expresados en un lenguaje de máquina.

SQL. (Structured Query Language): Lenguaje de preguntas estructurado para acceder a las Bases de Datos.

Spike. Es una solución muy simple al programa para explorar posibles soluciones

UML. (Unified Modeling Language): Es una notación Standard para modelar objetos del mundo real como primer paso en el desarrollo de programas orientados a objetos.

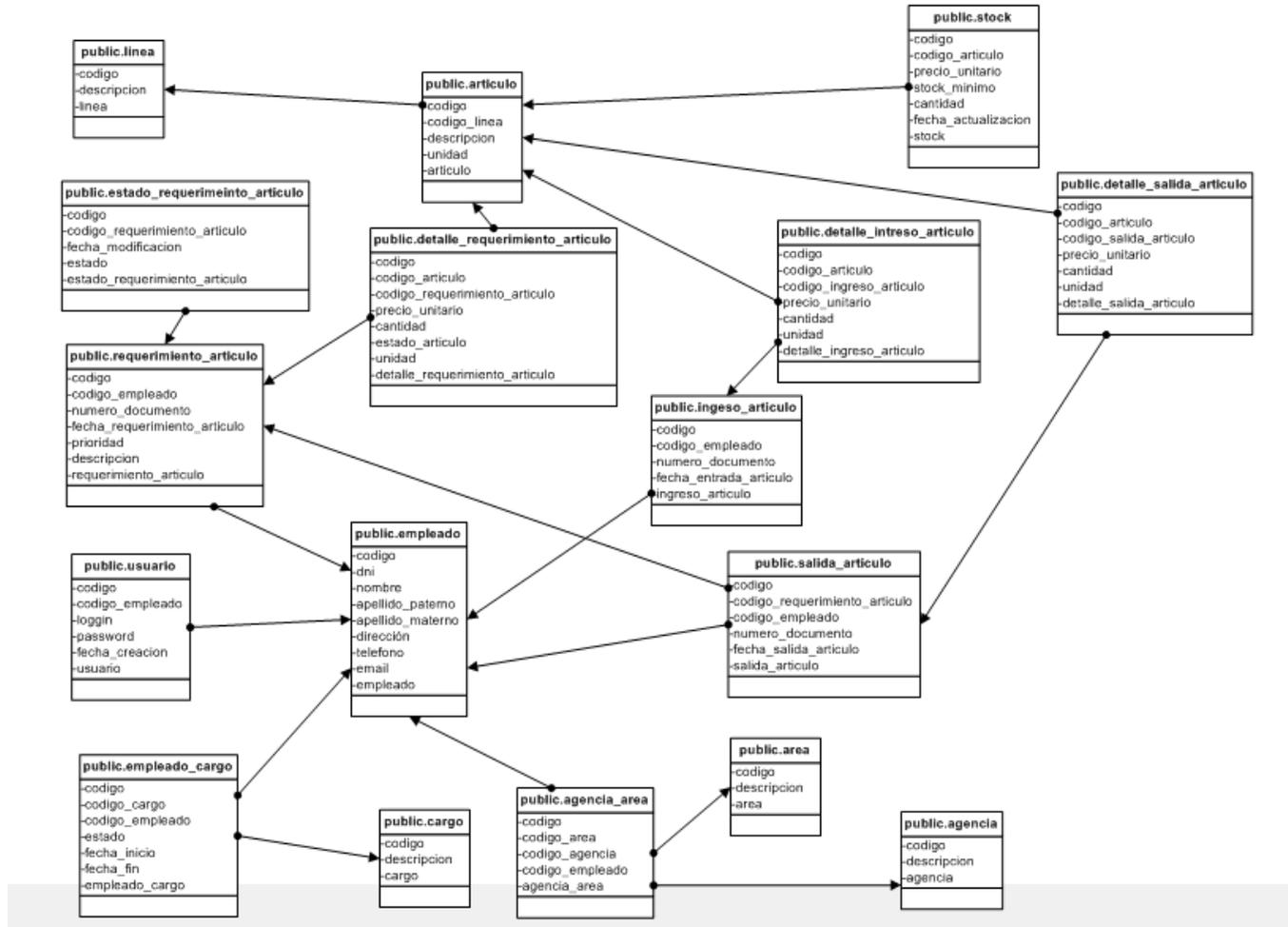
Tar. Refiere en Informática a un formato de archivos ampliamente usado en entornos UNIX, identificados con la extensión tar.

URL. (Uniform Resource Locator): Dirección de un recurso en la Web. Tiene el formato: protocolo//máquina.dominio/puerto/ruta/recurso.

WWW. (World Wide Web): Red de documentos HTML intercomunicados y distribuidos entre servidores del mundo entero.

XP. La programación extrema o Extreme Programming es un enfoque de la ingeniería de software formulado por Kent Beck, autor del primer libro sobre la materia, Extreme Programming Explained: Embrace Change (1999). Este programa es el más destacado de los procesos ágiles de desarrollo de software. Al igual que éstos, la programación extrema se diferencia de las metodologías tradicionales principalmente en que pone más énfasis en la adaptabilidad que en la previsibilidad.

ANEXO A:
DISEÑO LÓGICO DE LA APLICACIÓN WEB DE ALMACEN PARA COOPAC'S DEL PERÚ



ANEXO B:**DISEÑO DE BASE DE DATOS DE APLICACIÓN WEB DE ALMACEN PARA COOPAC'S DEL PERÚ****Tabla: EMPLEADO**

Nombre del Campo	Tipo de Dato	Nulo	PK	Único
codigo	serial	NOT NULL	Y	Y
dni	Varchar (8)	NOT NULL	N	Y
nombre	Varchar(35)	NOT NULL	N	N
apellido_parterno	Varchar (35)	NOT NULL	N	N
apellido_materno	Varchar (35)	NOT NULL	N	N
dirección	Varchar (100)	NOT NULL	N	N
teléfono	Varchar (16)	NULL	N	N
Email	Varchar (40)	NULL	N	N

Tabla: CARGO

Nombre del Campo	Tipo de Dato	Nulo	PK	Único
codigo	serial	NOT NULL	Y	Y
descripción	text	NOT NULL	N	N

Tabla: USUARIO

Nombre del Campo	Tipo de Dato	Nulo	PK	Único
codigo	serial	NOT NULL	Y	Y
login	Varchar (20)	NOT NULL	N	Y
password	Varchar(20)	NOT NULL	N	Y
apellido_parterno	Varchar (35)	NOT NULL	N	N
fecha_creacion	date	NOT NULL	N	N
estado	boolean	NOT NULL	N	N
codigo_empleado	integer (16)	NOT NULL	N	N

Tabla: AGENCIA

Nombre del Campo	Tipo de Dato	Nulo	PK	Único
codigo	serial	NOT NULL	Y	Y
descripcion	text	NOT NULL	N	N

Tabla: ÁREA

Nombre del Campo	Tipo de Dato	Nulo	PK	Único
codigo	serial	NOT NULL	Y	Y
descripcion	text	NOT NULL	N	N

Tabla: AGENCIA_AREA

Nombre del Campo	Tipo de Dato	Nulo	PK	Único
codigo	serial	NOT NULL	Y	Y
codigo_area	integer	NOT NULL	N	N
codigo_agencia	integer	NOT NULL	N	N
estado	boolean	NOT NULL	N	N
codigo_empleado	integer (16)	NOT NULL	N	N
fecha_inicio	date	NOT NULL	N	N
fecha_fin	date	NOT NULL	N	N

Tabla: EMPLEADO_CARGO

Nombre del Campo	Tipo de Dato	Nulo	PK	Único
codigo	Serial	NOT NULL	Y	Y
codigo_cargo	integer	NOT NULL	N	N
codigo_empleado	integer	NOT NULL	N	N
estado	boolean	NOT NULL	N	N
codigo_empleado	integer (16)	NOT NULL	N	N
fecha_inicio	Date	NOT NULL	N	N
fecha_fin	Date	NOT NULL	N	N

Tabla: INGRESO ARTICULO

Nombre del Campo	Tipo de Dato	Nulo	PK	Único
codigo	Serial	NOT NULL	Y	Y
numero_documento	Integer	NOT NULL	N	N
fecha_entrada_articulo	Date	NOT NULL	N	N
Codigo_empleado	Integer	NOT NULL	N	N

Tabla: DETALLE_INGRESO_ARTICULO

Nombre del Campo	Tipo de Dato	Nulo	PK	Único
codigo	Serial	NOT NULL	Y	Y
precio_unitario	Double precision	NOT NULL	N	N
cantidad	Doble precision	NOT NULL	N	N
unidad	Varchar(20)	NOT NULL	N	N
codigo_articulo	integer	NOT NULL	N	N
Codigo_ingreso_articulo	integer	NOT NULL	N	N

Tabla: LINEA

Nombre del Campo	Tipo de Dato	Nulo	PK	Único
Codigo	Serial	NOT NULL	Y	Y
descripción	Text	NOT NULL	N	N

Tabla: ARTICULO

Nombre del Campo	Tipo de Dato	Nulo	PK	Único
codigo	Serial	NOT NULL	Y	Y
descripción	Text	NOT NULL	N	N
unidad	Varchar(20)	NOT NULL	N	N
codigo_linea	integer	NOT NULL	N	N
Codigo_agencia	integer	NOT NULL	N	N

Tabla: STOCK

Nombre del Campo	Tipo de Dato	Nulo	PK	Único
codigo	Serial	NOT NULL	Y	Y
precio_unitario	Double precision	NOT NULL	N	N
Stock_minimo	Doble precision	NOT NULL	N	N
cantidad	Doble precision	NOT NULL	N	N
fecha_actualizacion	Date	NOT NULL	N	N
codigo_articulo	integer	NOT NULL	N	N

Tabla: SALIDA_ARTICULO

Nombre del Campo	Tipo de Dato	Nulo	PK	Único
codigo	Serial	NOT NULL	Y	Y
numero_documento	Integer	NOT NULL	N	N
fecha_salida_articulo	date	NOT NULL	N	N
codigo_requerimiento_articulo	Integer	NOT NULL	N	N
codigo_empleado	integer	NOT NULL	N	N

Tabla: DETALLE_SALIDA_ARTICULO

Nombre del Campo	Tipo de Dato	Nulo	PK	Único
codigo	Serial	NOT NULL	Y	Y
precio_unitario	Double precision	NOT NULL	N	N
cantidad	Double precision	NOT NULL	N	N
unidad	Varchar(20)	NOT NULL	N	N
codigo_articulo	Integer	NOT NULL	N	N
codigo_salida_articulo	Integer	NOT NULL	N	N

Tabla: REQUERIMIENTO_ARTICULO_ORIGINAL

Nombre del Campo	Tipo de Dato	Nulo	PK	Único
codigo	Serial	NOT NULL	Y	Y
numero_documento	Integer	NOT NULL	N	N
fecha_requerimiento_articulo	date	NOT NULL	N	N
prioridad	Varchar(30)	NOT NULL	N	N
descripcion	text	NOT NULL	N	N
Codigo_empleado	integer	NOT NULL	N	N

Tabla: DETALLE_REQUERIMIENTO_ARTICULO_ORIGINAL

Nombre del Campo	Tipo de Dato	Nulo	PK	Único
codigo	Serial	NOT NULL	Y	Y
Precio_unitario	Double precision	NOT NULL	N	N
cantidad	Double precision	NOT NULL	N	N
unidad	Varchar(20)	NOT NULL	N	N
codigo_articulo	integer	NOT NULL	N	N
Estado_articulo	boolean	NOT NULL	N	N
Codigo_requerimiento_articulo_original	Integer	NOT NULL	N	N

Tabla: REQUERIMIENTO_ARTICULO

Nombre del Campo	Tipo de Dato	Nulo	PK	Único
codigo	Serial	NOT NULL	Y	Y
numero_documento	Integer	NOT NULL	N	N
fecha_requerimiento_articulo	date	NOT NULL	N	N
Estado_requerimiento	Varchar(20)	NOT NULL	N	N
prioridad	Varchar(30)	NOT NULL	N	N
descripcion	text	NOT NULL	N	N
Codigo_empleado	integer	NOT NULL	N	N

Tabla: DETALLE_REQUERIMIENTO_ARTICULO

Nombre del Campo	Tipo de Dato	Nulo	PK	Único
codigo	Serial	NOT NULL	Y	Y
Precio_unitario	Double precision	NOT NULL	N	N
cantidad	Double precision	NOT NULL	N	N
unidad	Varchar(20)	NOT NULL	N	N
codigo_articulo	integer	NOT NULL	N	N
Estado_articulo	boolean	NOT NULL	N	N
Codigo_requerimiento_articulo	Integer	NOT NULL	N	N

Tabla: ESTADO_REQUERIMIENTO_ARTICULO

Nombre del Campo	Tipo de Dato	Nulo	PK	Único
codigo	Serial	NOT NULL	Y	Y
fecha_modificacion	date	NOT NULL	N	N
estado	Varchar(20)	NOT NULL	N	N
codigo_empleado	integer	NOT NULL	N	N
codigo_requerimiento_articulo	integer	NOT NULL	N	N

Tabla: MENU

Nombre del Campo	Tipo de Dato	Nulo	PK	Único
Codigo	Serial	NOT NULL	Y	Y
nombre	Varchar(100)	NOT NULL	N	N
url	Varchar(120)	NOT NULL	N	N
estado	Boolean	NOT NULL	N	N
orden	Integer	NOT NULL	N	N

Tabla: PEMISO

Nombre del Campo	Tipo de Dato	Nulo	PK	Único
Codigo	Serial	NOT NULL	Y	Y
estado	Boolean	NOT NULL	N	N
codigo_usuario	Integer	NOT NULL	N	N
codigo_menu	Integer	NOT NULL	N	N

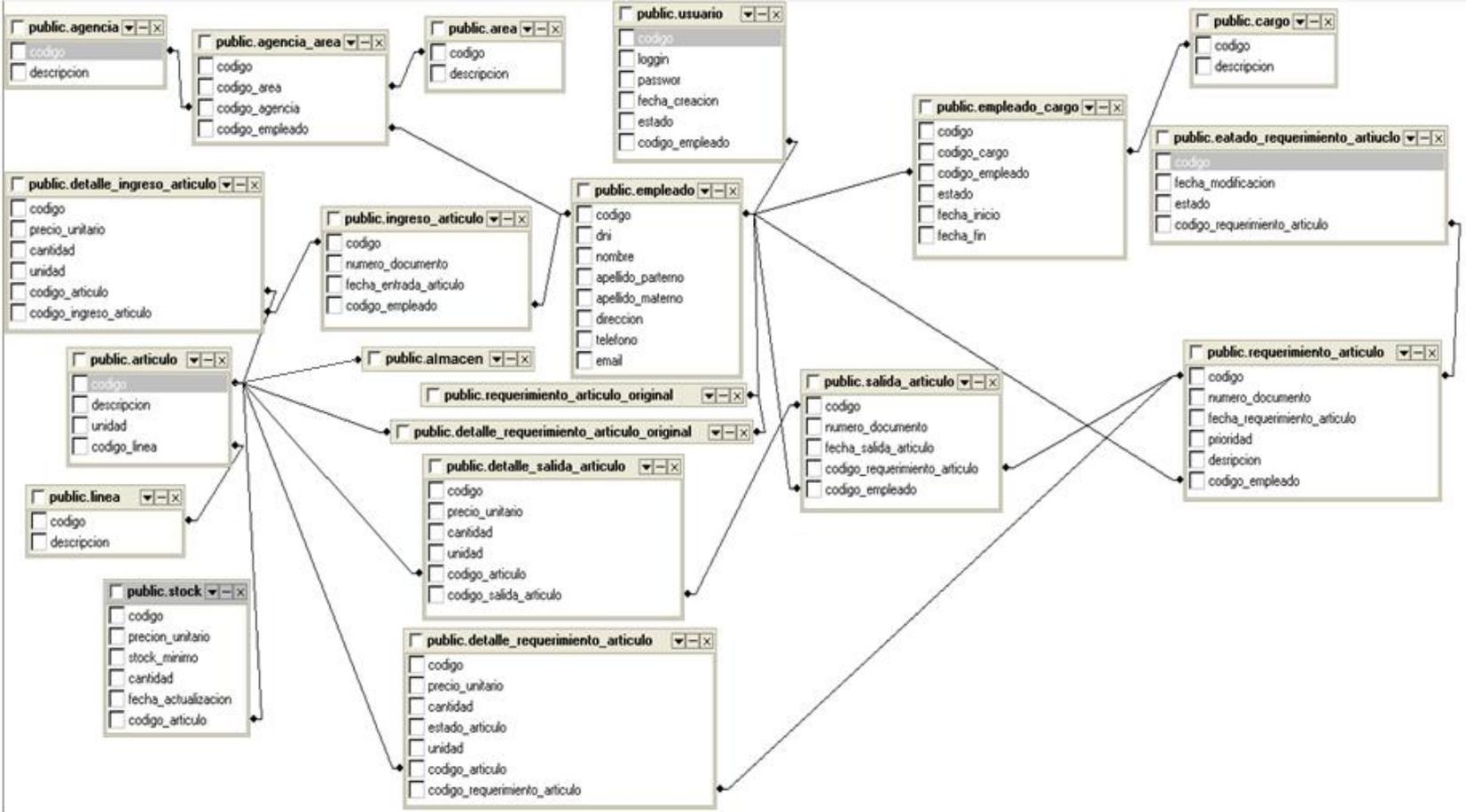
Tabla: PERSONA_TRABAJA

Nombre del Campo	Tipo de Dato	Nulo	PK	Único
Codigo	Serial	NOT NULL	Y	Y
Codigo_empleador	Integer	NOT NULL	N	N
codigo_agencia_area	Integer	NOT NULL	N	N
Fecha_asignacion	date	NOT NULL	N	N
Fecha_fin	date	NOT NULL	N	N
estado	Boolean	NOT NULL	N	N

Tabla: ALMACEN

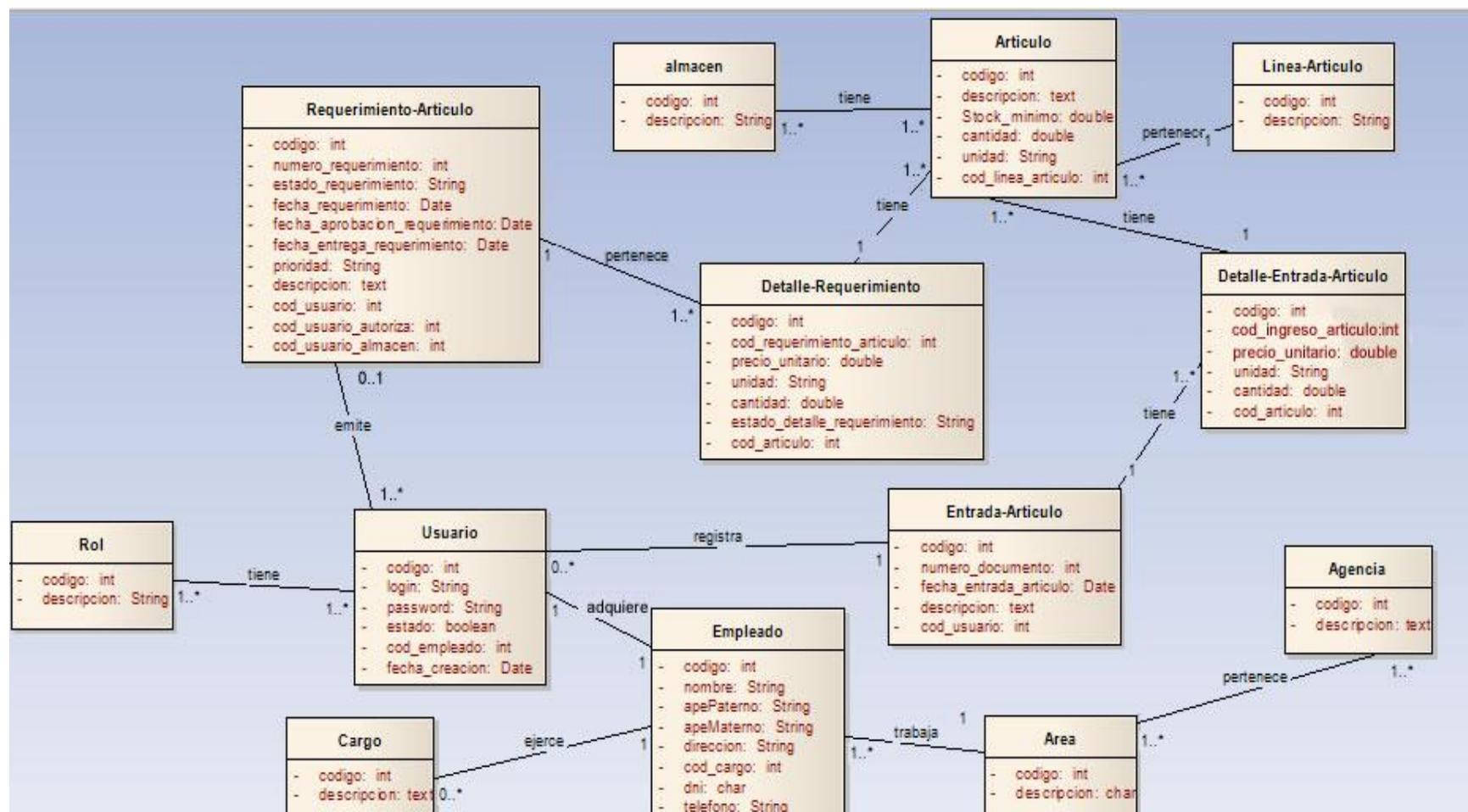
Nombre del Campo	Tipo de Dato	Nulo	PK	Único
codigo	serial	NOT NULL	Y	Y
descripcion	text	NOT NULL	N	N

MODELO DE BASE DATOS DE APLICACIÓN WEB DE ALMACEN PARA COOPAC'S DEL PERÚ.



ANEXO C:

DIAGRAMA DE CLASES DE APLICACIÓN WEB DE ALMACEN PARA COOPAC'S DEL PERÚ.





COOPERATIVA DE AHORRO Y CREDITO SAN PEDRO DE ANDAHUAYLAS

REQUERIMIENTO DOCUMENTAL N° 000014-2009-AGENCIA ANDAHUAYLAS

Fecha: _____

RESPONSABLE / SOLICITANTE:

SOLICITANTE:

DEPENDENCIA RECEPTORA DE LA SOLICITUD:

ENCARGADO:

ESPECIFICACIONES DE LA CONSULTA

ITEM	CANTIDAD	UNIDAD MEDIDA	DESCRIPCIÓN DEL REQUERIMIENTO
1			
2			
3			
4			
5			
6			
7			
8			
9			
10			
11			
12			

JU STIFICACIÓN DEL REQUERIMIENTO:

Elaborado por: <u> Sistemas </u> Fecha: <u> 28-12-2009 </u> Firma: _____	Autorizado por: _____ Fecha: _____ Firma: _____	Recibido por: _____ Fecha: _____ Firma: _____
------------------------------------------------------------------------------------	-------------------------------------------------------	-----------------------------------------------------

Fecha estimada de respuesta al requerimiento:

Fecha real de respuesta al requerimiento:

FUENTE: CAC San Pedro de Andahuaylas.

COOPERATIVA DE AHORRO Y CREDITO SAN MARTIN DE PORRES DIRECCION DE RECURSOS MATERIALES Y SERVICIOS DEPARTAMENTO DE ALMACENES E INVENTARIOS SALIDA DE ALMACEN						Fecha : _____ Hora : _____					
Unidad responsable: Destino : _____				Bodega : Central Requisición : _____		Pagina : 1/1 No. Salida : _____					
o	Requis	Lote	Pedido-No.	Característica del Artículo	Cantidad	Unidad	Precio unitario	-%	+%	Precio total	Total
<p>S E P.</p> <p>SECRETARIA GENERAL DE LA CIUAP</p> <p>TOTAL:</p>											
AUTORIZA			INSPECCIONA			ENTREGA					
NOTA: Estampar sello oficial de la dependencia al recibir Favor de no hacer anotaciones en esta salida porque se invalida											RECIBIMOS DE CONFORMIDAD LOS BIENES QUE SE DESCRIBEN. Nombre : _____ Fecha : _____ Hora : _____ Firma : _____

FUENTE: CAC San Martín de Porres de Tarapoto



INFORME N° [REDACTED] COSEMA/LHG

A: [REDACTED]
 Encargada Del Área de Contabilidad y Administración

DE: LIA LIZBEL HILARES GAVILAN
 Encargada / Agencia Pichari

Asunto: REQUERIMIENTOS DE COMPRA DE PREMIOS MASTERCOP

Fecha: [REDACTED]

Por medio del presente me dirijo a usted. Con la finalidad de comunicarle que los socios de la agencia pichari están participando en la campaña del mastercoop y se les tiene que hacer la entrega de premios correspondientes, por lo cual se le solicita la compra de los premios, según el siguiente detalle:

N°	RELACION DE SOCIOS	TIEMPO	MONTO	PREMIO MASTERCOP
01	PEREZ SIQUINCHA ANGEL	6 MESES	1,000.00	01 POLO
02	TINPO QUISPLYENI YANET	6 MESES	1,000.00	01 POLO
03	LOAYZA FLORES LEONORA	6 MESES	1,000.00	01 POLO
04	QUINTAN VILLAR MAURKA	6 MESES	1,000.00	01 POLO
05	FLORES TORRE JOSE	6 MESES	2,000.00	01 JUEGO DE VAJILLA
06	CABRERA GUILLEN ABELARDO	6 MESES	2,000.00	01 JUEGO DE VAJILLA
07	CABRERA GUILLEN ABELARDO	6 MESES	2,000.00	01 JUEGO DE VAJILLA
08	LLANTOY VILCA ALBINO	6 MESES	2,000.00	01 JUEGO DE VAJILLA
09	PENA CHAVEZ RUBEN	6 MESES	1,000.00	01 JUEGO DE CUBIERTOS
10	GAVILAN YAUYO MARIA	6 MESES	3,000.00	01 JUEGO DE CUBIERTOS
11	RIMACHI GONZALES ESTEBAN	6 MESES	4,000.00	01 SANGUCHERA

Atentamente:

 SEÑOR DE MAYNAY
 Cooperativa de Ahorro y Crédito
 Lia Lizbel Hilares Gavilan
 Encargada / Pichari

ANEXO E: Ranking de las Coopac del Perú 2009

FEDERACION NACIONAL DE COOPERATIVAS DE AHORRO Y CREDITO DEL PERU – FENACREP

RANKING DE LAS COOPAC

Al 30 Setiembre De 2009

(En Miles de Nuevos Soles)

COOPAC CON ACTIVOS SUPERIORES A 1500 U.I.T.

Rank Activos	COOPERATIVAS	ACTIVO TOTAL		
		Monto	%	%Acum
1	ABACO	460,030	15.21	15.21
2	AELUCOOP	354,850	11.73	26.94
3	PACIFICO	282,790	9.35	36.28
4	SANTA MARIA MAGDALENA LTDA. - A	126,358	4.18	40.46
5	SAN CRISTOBAL DE HUAMANGA LTDA.	97,305	3.22	43.68
6	SANTO CRISTO DE BAGAZAN	93,105	3.08	46.75
7	SAN MARTIN DE PORRES - T	91,982	3.04	49.79
8	ILO LTDA.	86,901	2.87	52.67
9	SAN ISIDRO	84,088	2.78	55.45
10	SAN PEDRO 116	78,478	2.59	58.04
11	SANTO DOMINGO DE GUZMAN LTDA. - C	75,662	2.50	60.54
12	SANTA CATALINA	64,294	2.13	62.67
13	PETROPERU Ltda.	58,699	1.94	64.61
14	QUILLABAMBA	51,231	1.69	66.30
15	LEON XIII LTDA.	51,091	1.69	67.99
16	DEL SUR	45,012	1.49	69.48
17	FINANTEL Ltda.	44,801	1.48	70.96
18	TOQUEPALA	43,257	1.43	72.39
19	LA ESPERANZA DE MARCONA	37,629	1.24	73.63
20	CUAJONE	35,999	1.19	74.82
21	SAN HILARION	31,746	1.05	75.87
22	TUMAN	29,072	0.96	76.83
23	SAN FRANCISCO	28,358	0.94	77.77
24	CRL. FRANCISCO BOLOGNESI	27,442	0.91	78.68
25	LA REHABILITADORA	27,178	0.90	79.57
26	EFIDE	26,877	0.89	80.46
27	HUANCAVELICA LTDA. 582 - H	23,620	0.78	81.24
28	LOS ANDES COTARUSI AYMARAES	23,007	0.76	82.00
29	TOCACHE	22,653	0.75	82.75
30	DE LOS TRAB. DEL SECTOR SALUD	20,549	0.68	83.43
31	DE LOS EMPLEADOS DE IBM	19,856	0.66	84.09
32	QUINCE DE SETIEMBRE LTDA.	18,713	0.62	84.71
33	DE LOS TRAB. DE VOLCAN CIA. MINERA	17,240	0.57	85.28
34	DE OFICIALES PNP CRNL. HUMBERTO FLORES HIDALGO	16,729	0.55	85.83
35	VEINTINUEVE DE AGOSTO DE LA PNP	15,572	0.51	86.34
36	LA PORTUARIA	15,507	0.51	86.86
37	PARROQUIA SAN LORENZO TRUJILLO Ltda.	14,868	0.49	87.35
38	S.O. 3ERA. S.F.P. JOSE DEL CARMEN HUAMAN MUÑOZ	14,184	0.47	87.82
39	NSTRA. SRA. DEL ROSARIO - C	13,553	0.45	88.26
40	SAN JOSE CARTAVIO	13,233	0.44	88.70
41	LA FLORIDA - CREDIFLORIDA	12,616	0.42	89.12
42	EL DORADO	12,055	0.40	89.52

43	CHIQUINQUIRA LTDA.	10,432	0.34	89.86
44	PROMOCION TAHUANTINSUYO	9,988	0.33	90.19
45	AOPCOOP	9,941	0.33	90.52
46	PARROQUIA SANTA ROSA DE LIMA	9,541	0.32	90.84
47	MARBELLA DE MAGDALENA	9,287	0.31	91.14
48	AFIANZA	8,535	0.28	91.43
49	NORANDINO	8,501	0.28	91.71
50	SAN PEDRO DE HUANCABAMBA	8,382	0.28	91.98
51	TRAB. EMP. LUZ Y FUERZA ELECTRICA-CREDICOOP	8,364	0.28	92.26
52	FINANZAS SOLIDARIAS MONSEÑOR MARCOS LIBARDONI	8,001	0.26	92.52
53	SEÑOR DE LOS MILAGROS DE ATACUCHA	7,933	0.26	92.79
54	SAN JOSE LTDA. - IQUITOS	7,476	0.25	93.03
55	FORTALECER	7,318	0.24	93.28
56	VIRGEN DE LAS NIEVES LTDA. 278	6,591	0.22	93.49
57	SERVICOOP	6,531	0.22	93.71
58	PARQUE INDUSTRIAL	6,166	0.20	93.91
59	SAN PIO X Ltda.	6,162	0.20	94.12
60	SERVICIO ADUANERO	5,713	0.19	94.31
61	EMPHOST	5,640	0.19	94.49
62	TRAB. IPSS SAN JORGE	5,486	0.18	94.67
63	TRAB. DEL BANCO CENTRAL DE RESERVA DEL PERÚ	5,343	0.18	94.85

Fuente: INFORME ANUAL DE LA FEDERACIÓN NACIONAL DE COOPERATIVAS DE AHORRO Y CRÉDITO DEL PERÚ.

ANEXO F: ÍNDICE DE FIGURAS Y TABLAS

Figuras	Página
Figura N° 01: Las prácticas de Programación Extrema.	17
Figura N° 02: En la Programación Extrema las entregas están hechas de iteraciones.	19
Figura N° 03: El ciclo de vida del proyecto programación Extrema	20
Figura N° 04: Proceso de una iteración del ciclo de vida XP	25
Figura N° 05: Arquitectura Web básica	39
Figura N° 06: Fases genéricos del estudio de viabilidad	45
Figura N° 07: Plataforma Aplicación Web.	47
Figura N° 08: Diagrama Físico de redes de la aplicación Web.	48
Figura N° 09: Arquitectura técnica inicial	66
Figura N° 10: Arquitectura técnica de la Aplicación Web.	67
Figura N° 11: Arquitectura técnica. Diagrama de componentes	77
Figura N° 12: Arquitectura técnica. Diagrama de despliegue.	77
Figura N° 13: GUI de la Historia de usuario: Emitir Requerimiento de Artículo.	83
Figura N° 14: GUI de la Historia de usuario: Control de Acceso	83
Figura N° 15: GUI de la Historia de usuario: GUI N° 3.6: Menú Principal del Sistema.	84
Figura N° 16: Código fuente de la clase entidad. Usuario.	87
Figura N° 17: Código fuente para el historia de usuario. Control de Acceso de Usuarios.	88

ÍNDICE DE TABLAS

Tabla	Página
Tabla N° 01: Fases de la Programación Extrema	18
Tabla N° 02: Formato de Historia de Usuario.	22
Tabla N° 03: Responsabilidades Durante la etapa de Planificación	23
Tabla N° 04: Formato de Pruebas de Aceptación	30
Tabla N° 05: Detalle de los Módulos de Conexión de apache.	53
Tabla N° 06: Historias de usuario.	62
Tabla N° 07: Emitir Requerimiento de Artículo	62
Tabla N° 08: Otorgar requerimiento de artículos	62
Tabla N° 09: Entregar requerimiento de artículos	63

Tabla N° 10: Registro de Ingreso de Artículos.	63
Tabla N° 11: Control de acceso de usuarios.	63
Tabla N° 12: Mantener Requerimiento de Artículo.	64
Tabla N° 13: Mantener Datos del Empleado.	64
Tabla N° 14: Mantener Usuario.	64
Tabla N° 15: Mantener Artículo.	65
Tabla N° 16: Mantener Área.	65
Tabla N° 17: Mantener Agencia.	65
Tabla N° 18: Mantener Cargo.	65
Tabla N° 19: Plan de alto nivel	68
Tabla N° 20: Emitir requerimiento de artículos	68
Tabla N° 21: Otorgar requerimiento de artículos	69
Tabla N° 22: Entregar requerimiento de artículos	70
Tabla N° 23: Registrar Ingreso de Artículo	70
Tabla N° 24: Control de acceso de usuarios	71
Tabla N° 25: Mantener Requerimiento de Artículo	71
Tabla N° 26: Mantener Datos del Empleado	72
Tabla N° 27: Mantener Usuario	73
Tabla N° 28: Mantener Artículo	73
Tabla N° 29: Mantener Área	74
Tabla N° 30: Mantener Agencia	74
Tabla N° 31: Mantener Cargo	75
Tabla N° 32: plan de versión	75
Tabla N° 33: Detalle de hardware	78
Tabla N° 34: Tarea de ingeniería. Autenticar usuario	79
Tabla N° 35: Tarea de ingeniería. Agregar artículo	79
Tabla N° 36: Tarea de ingeniería. Cargar Stock de artículo	79
Tabla N° 37: Tarea de ingeniería. Cargar Línea Artículo	80
Tabla N° 38 Tarea de ingeniería. Grabar Requerimiento Artículo	80
Tabla N° 39: Tarea de ingeniería. Autenticar usuario	80
Tabla N° 40: Tarea de ingeniería. Cargar Permisos	81
Tabla N° 41: Tarea de ingeniería. Buscar Empleado.	81
Tabla N° 42: Tarea de ingeniería. Cargar Cargo	81
Tabla N° 43: Tarea de ingeniería. Cargar Agencia.	82
Tabla N° 44: Tarea de ingeniería. Cargar Área.	82
Tabla N° 45: Plan de iteración (Primera)	82
Tabla N° 46: Tarjeta CRC. Requerimiento de Artículo.	84
Tabla N° 47: Tarjeta CRC. Estado de Requerimiento de Artículo.	84
Tabla N° 48: Tarjeta CRC. Detalle de Requerimiento de Artículo.	85
Tabla N° 49: Tarjeta CRC. Artículo.	85
Tabla N° 50: Tarjeta CRC. Stock Artículo	85

Tabla N° 51: Tarjeta CRC. Línea Artículo	85
Tabla N° 52: Tarjeta CRC. Usuario.	86
Tabla N° 53: Tarjeta CRC. Permiso	86
Tabla N° 54: Tarjeta CRC. Empleado	86
Tabla N° 55: Tarjeta CRC. Cargo	86
Tabla N° 56: Tarjeta CRC. Área	86
Tabla N° 57: Tarjeta CRC. Agencia.	87
Tabla N° 58: Reporte de pruebas unitarias. Primera iteración.	89
Tabla N° 59: Reporte de prueba integración. Primera iteración.	89
Tabla N° 60: Reporte de prueba de aceptación. Mantener Empleado.	89
Tabla N° 61: Reporte de prueba de aceptación. Mantener Cuenta de Usuario.	90
Tabla N° 62: Reporte de prueba de aceptación. Mantener Artículo.	90
Tabla N° 63: Diario de Actividades Aplicación Web para Almacén de Cooperativas De Ahorro Y Crédito Del Perú.	91