

**UTMACH**

UNIDAD ACADÉMICA DE INGENIERÍA CIVIL  
CARRERA DE ANÁLISIS DE SISTEMAS

TEMA:

**ANÁLISIS DE UN SISTEMA DE CONTROL DE PRODUCTOS DE CONSUMO MASIVO  
DE SUPERMERCADOS S.A. MEDIANTE LENGUAJE MODELADO UNIFICADO**

TRABAJO PRÁCTICO DEL EXAMEN COMPLEXIVO PREVIO A LA OBTENCIÓN DEL  
TÍTULO DE ANALISTA DE SISTEMAS

**AUTORA:**

0704404649-GARCIA ULLOA PATRICIA ALEXANDRA

MACHALA-EL ORO

## CESIÓN DE DERECHOS DE AUTOR

Yo, GARCIA ULLOA PATRICIA ALEXANDRA, con C.I. 0704404649, estudiante de la carrera de ANÁLISIS DE SISTEMAS de la UNIDAD ACADÉMICA DE INGENIERÍA CIVIL de la UNIVERSIDAD TÉCNICA DE MACHALA, en calidad de Autora del siguiente trabajo de titulación ANÁLISIS Y DISEÑO DEL SISTEMA DE CONTROL DE MERCADERÍA DE SUPERMERCADOS S.A. EN LA PROVINCIA DE EL ORO

- Declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional. En consecuencia, asumo la responsabilidad de la originalidad del mismo y el cuidado al remitirme a las fuentes bibliográficas respectivas para fundamentar el contenido expuesto, asumiendo la responsabilidad frente a cualquier reclamo o demanda por parte de terceros de manera EXCLUSIVA.
  
- Cedo a la UNIVERSIDAD TÉCNICA DE MACHALA de forma NO EXCLUSIVA con referencia a la obra en formato digital los derechos de:
  - a. Incorporar la mencionada obra al repositorio digital institucional para su democratización a nivel mundial, respetando lo establecido por la Licencia Creative Commons Atribución-NoComercial-CompartirIgual 4.0 Internacional (CC BY-NC-SA 4.0), la Ley de Propiedad Intelectual del Estado Ecuatoriano y el Reglamento Institucional.
  
  - b. Adecuarla a cualquier formato o tecnología de uso en internet, así como incorporar cualquier sistema de seguridad para documentos electrónicos, correspondiéndome como Autor(a) la responsabilidad de velar por dichas adaptaciones con la finalidad de que no se desnaturalice el contenido o sentido de la misma.

Machala, 25 de Noviembre de 2015



---

GARCIA ULLOA PATRICIA ALEXANDRA  
C.I. 0704404649

## **RESUMEN**

### **ANÁLISIS DE UN SISTEMA DE CONTROL DE PRODUCTOS DE CONSUMO MASIVO DE SUPERMERCADOS S.A. MEDIANTE LENGUAJE MODELADO UNIFICADO.**

Patricia Alexandra García Ulloa

Ing. Joffre Jeorwin Cartuche Calva

El trabajo presentado a continuación realiza un análisis y diseño del sistema de control de Mercadería de los Supermercados S.A en la Provincia de El Oro. Se empezará identificando los problemas y necesidades de la empresa para poder determinar los requerimientos funcionales y no funcionales que tendrá este sistema. con la finalidad de tener claro los procesos que deben ser tomados en cuenta. En el presente trabajo además se realizará una estimación del costo del Software por puntos de función. Se realizarán también los diferentes casos de uso y el diagrama de casos de uso, tomando como aliado a nuestro proceso El Lenguaje Modelado Unificado (UML). Todo esto para que se consiga un orden en la manipulación, ingreso y egreso de mercadería, tanto el local principal como de las sucursales.

#### Palabras Claves:

- Mercadería
- Supermercados S.A.
- Estimación
- Análisis
- Casos de uso.

## **SUMMARY**

### **ANALISIS DE UN SISTEMA DE CONTROL DE PRODUCTOS DE CONSUMO MASIVO DE SUPERMERCADOS S.A. MEDIANTE LENGUAJE MODELADO UNIFICADO.**

The work presented below an analysis and design of control system Merchandise Supermarket SA in the province of El Oro. It will begin by identifying the problems and needs of the company to determine functional and non-functional requirements that will have this system . in order to be clear about the processes that must be taken into account. In this paper also estimate the cost of the Software will be made by function points. The different use cases and use case diagram, taking as our ally Unified Process Methodology Language (UML) is also performed. All this for an order to be achieved in handling, entry and exit of goods, both local and main branches.

#### Keywords:

- Merchandise
- Supermarket S.A.
- Estimation
- Analysis
- Use cases.

## INDICE DE CONTENIDOS

Portada.....	i
Frontispicio.....	ii
Cesión de Derechos de Autor.....	iii
Resumen.....	iv
Summary.....	v
Índice de Contenido .....	v
Índice de Tablas .....	vi
<b>1 INTRODUCCIÓN .....</b>	<b>1</b>
<b>1.1 Marco Contextual.....</b>	<b>1</b>
<b>1.2 Problema.....</b>	<b>1</b>
<b>1.3 Objetivo General .....</b>	<b>2</b>
<b>2. DESARROLLO.....</b>	<b>2</b>
<b>2.1 Marco Teórico .....</b>	<b>2</b>
<b>2.1.1 Metodologías de Software.....</b>	<b>2</b>
<b>Metodología RUP.....</b>	<b>2</b>
<b>Estructura de la Metodología RUP.....</b>	<b>2</b>
<b>Lenguaje Unificado de Modelado.....</b>	<b>3</b>
<b>2.1.2 Requerimientos del Software.....</b>	<b>3</b>
<b>Requisitos Funcionales.....</b>	<b>3</b>
<b>Requisitos No Funcionales.....</b>	<b>3</b>
<b>2.1.3 Estimación del proyecto software por puntos de Función.....</b>	<b>3</b>
<b>Estimación de puntos de función.....</b>	<b>4</b>
<b>Terminología.....</b>	<b>4</b>
<b>Estimación del esfuerzo del proyecto.....</b>	<b>5</b>
<b>Estimación del tiempo del proyecto.....</b>	<b>6</b>
<b>Estimación del costo del proyecto.....</b>	<b>7</b>
<b>2.1.4 Diagrama de Casos de Uso.....</b>	<b>7</b>
<b>2.2. Marco Metodológico .....</b>	<b>7</b>
<b>2.3. Resultados .....</b>	<b>10</b>
<b>3. CONCLUSIONES .....</b>	<b>11</b>
<b>4. REFERENCIAS BIBLIOGRÁFICAS .....</b>	<b>11</b>
<b>5. ANEXOS. ....</b>	<b>12</b>

## INDICE DE TABLAS

DESCRIPCION	PÁGINAS
Tabla 1: Factores que determinan el nivel de complejidad	5
Tabla 2: Cálculo de los Puntos de función	5
Tabla 3: Niveles de Influencia para factor de ajuste	6
Tabla 4: Horas por Punto de Función según L. Programación	6
Tabla 5: Tipos de relaciones de casos de uso	7
Tabla 6: Conteo de puntos de función Archivo Logico interno	8

# 1 INTRODUCCION

Mediante el análisis y la planificación consciente, un profesional tiene derecho a esperar un grado de éxito mucho mayor que si toma las decisiones al azar o por el método del ensayo y error. (Berroni, 2000)

Apoiados en este pensamiento vamos a utilizar la metodología RUP, pues el desarrollo de nuestro sistema requiere que hagamos un estudio analítico del problema, que detectemos todos los requerimientos funcionales y no funcionales para poder desarrollar el sistema con un margen mínimo de errores.

Se utilizará para la estimación del Software la técnica de puntos de función, para lo cual tendremos que identificar todas las funciones que ofrece el Sistema y estimar el esfuerzo, el tiempo y el costo de nuestro proyecto.

El presente estudio tendrá como aliado la Herramienta UML (Lenguaje Modelado Unificado) que servirá para poder realizar, documentar y modelar todos y cada uno de los procesos necesarios, utilizando Casos de Uso y diagramas respectivos.

## 1.1. Marco Contextual

“La Ingeniería de Software se considera la rama de la ingeniería que aplica los principios de la ciencia de la computación y las matemáticas para lograr soluciones costo-efectivas a los problemas de desarrollo de software, es decir, permite elaborar consistentemente productos correctos, utilizables y costo-efectivos. La misma requiere llevar a cabo varias tareas, una de ellas es el análisis de requisitos. El análisis de requisitos permite extraer los requisitos de un producto de software. La Ingeniería de Software es una tecnología que indica “COMO” construir técnicamente un Software: económico, fiable y que funcione eficientemente” (Booch, 2002)

“Al ser el Software un bien en el que se invierten grandes cantidades, se hace muy importante poder controlarlo. En los años recientes, el gasto en tecnologías de información ha trasladado su énfasis del hardware al software, provocando que la relación entre el segundo y el primero suba de 32.5% en 2006 a 40% en 2012” (Duran, 2013, pág. 15)

De hecho se conoce que grandes proyectos han fracasado al no estar a tiempo o dentro del presupuesto por una mala estimación de esfuerzo o duración, o de las capacidades requeridas de los ingenieros y de la empresa.

## 1.2. Problema

Supermercados S.A. es una empresa dedicada a la comercialización al detalle de diversos productos de consumo masivo. Cuenta con una cadena de locales de autoservicio en distintas zonas del área metropolitana, y planea expandir su influencia a otras zonas de la Provincia del El Oro.

En sí el problema radica en la necesidad de la comunicación de la información entre el local principal y todas sus sucursales.

Dada la problemática existente nos planteamos el siguiente problema:

¿Cómo mejorar el proceso de control de productos de consumo masivo de la Empresa Supermercados S.A.?

### 1.3. Objetivo General

Analizar un Sistema de Control de Mercadería mediante el Lenguaje UML y técnicas de estimación para el mejoramiento el proceso de Control de Productos de Consumo Masivo de la Empresa Supermercados S.A.

## 2. DESARROLLO

### 2.1. Marco Teórico

#### 2.1.1. Metodologías de Software

“Las Metodologías que se utilizan para desarrollar un Software se componen de una serie de procedimientos, tácticas y mejoras a la documentación para el desarrollo de software. Se puede decir que se asemeja a la cocina en donde se sigue un recetario que nos va indicando paso a paso como realizar obtener el producto informático deseado, nos va diciendo a que personas debemos tomar en cuenta para lograr un buen producto , también nos indica que roles debe tener cada integrante del equipo. Mencionaremos también que toda la información es detallada tanto para iniciar una actividad o aquella que debe ser resultado de una actividad” (Carpio, 2008)

Para el presente trabajo debemos tener como referencia una metodología de desarrollo de Software, la cual guíe el camino a seguir para resolver el problema suscitado. Entonces mencionaremos que utilizaremos la Metodología RUP.

##### 2.1.1.1. Metodología RUP

“La Metodología RUP es una metodología cuyo fin es entregar un producto de Software. Se estructura todos los procesos y se mide la eficiencia de la Organización. Es un proceso de desarrollo de Software el cual utiliza el lenguaje Unificado de Modelado UML, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos” (Luna, 2013)

Es decir que esta metodología involucra al Lenguaje Unificado de Modelado UML, afin con casos de uso y demás diagramas.

##### 2.1.1.2. Estructura de la Metodología RUP

Según (Loor, 2010):La Disciplina de desarrollo de la Metodología RUP está estructurada de la siguiente manera:

- **Ingeniería de negocios:** es aplicada para entender las necesidades y la cultura empresarial dentro de la organización.
- **Requerimientos:** para trasladar las necesidades a un sistema automatizado
- **Análisis y Diseño:** permite representar los requerimientos obtenidos en una arquitectura de software.
- **Implementación:** Se crea un software que se ajuste a la arquitectura alcanzada y que tenga el comportamiento deseado
- **Pruebas:** permite asegurar que el comportamiento requerido sea el correcto y que todo lo solicitado este presente



Para dar solución al problema pasaremos por la Fase de Ingeniería de negocios, de requerimientos en su totalidad, y se desarrollará en su mayoría la fase de Análisis y Diseño.

### **2.1.1.3. Lenguaje Unificado de Modelado**

“El lenguaje Unificado de Modelado (UML) nos sirve para realizar la visualización, especificación y construcción documentada de todos quienes intervienen en un sistemas y que llevan gran cantidad de software. UML nos ofrece una forma estándar y accesible de trazar los esbosos de un sistema, tomando en cuenta tanto lo conceptual como todo lo que tiene que ver en el proceso y las funciones que tendrá el sistema, así también las cosas concretas, tales como las clases escritas en un lenguaje de programación específico, esquemas de bases de datos y componentes software reutilizables.” (Jacobson, 2000)

El Lenguaje Unificado de Modelado nos dará las pautas para crear, diseñar y modelar los casos de uso que se requieran para el presente estudio.

### **2.1.2. Requerimientos del Software**

#### **2.1.2.1. Requisitos Funcionales**

“Un requisito funcional es aquel que describe las funciones del sistema de software o quienes lo componen. Podemos describir a la función como un conjunto de input y output, es decir entradas y salidas..” (Duran, 2013)

Para definir cada uno de los requisitos funcionales que tiene nuestro presente caso de estudio se tomará en cuenta cada uno de los Proceso generales establecidos y mencionados en el problema.

#### **2.1.2.2. Requisitos No Funcionales**

“Un requisito no funcional o atributo de calidad es, en la ingeniería de sistemas y la ingeniería de software, un requisito que especifica criterios que pueden usarse para juzgar la operación de un sistema en lugar de sus comportamientos específicos, ya que éstos corresponden a los requisitos funcionales. Por tanto, se refieren a todos los requisitos que no describen información a guardar, ni funciones a realizar.” (Pressman, 2002)

Es decir que dentro de los requisitos no funcionales se podrá mencionar cuestiones relacionadas con el tiempo de respuesta, con el hardware y software mínimo para el buen funcionamiento del programa, etc.

### **2.1.3. Estimación del proyecto software por puntos de Función**

“El Método de Puntos de Función, es un metodos que se lo desarrollo en el año de 1979 por Allan Albrecht, quien definió conceptos para medir el software a partir de valoraciones de funcionalidades entregadas al usuario y no a partir de aspectos técnicos, con la intención de producir valoraciones independientes de la tecnología y fases del ciclo de vida utilizado. Este método realiza la valoración de todo el proyecto y finalmente se puede aplicar un factor de ajuste, que puede depender de características generales del sistema como por ejemplo los requisitos no funcionales” (Rodríguez, 2010)

La estimación del proyecto software por puntos de función permite una conexión del usuario o del cliente que pueda entender desde el principio como se va desarrollando sus sistema.

### 2.1.3.1. Estimación de puntos de función

“Una mejor forma de estimar el esfuerzo requerido para desarrollar un producto de software es la determinación de los puntos de Función asociados a el. Los puntos de función se están convirtiendo en un estándar aceptado para la medición del tamaño del software.” (Elliot, 2005)

Debido a esta particularidad la gran mayoría de personas utiliza este método para poder estimar el tiempo, el esfuerzo y el costo de un proyecto software.

Según (Doria, 2006) para poder determinar los puntos de función se debe realizar lo siguiente:

-Identificar archivos:

- Asignar a cada uno un tipo (ILF, EIF)

- Identificar la cantidad de Data Element Type (Det) y Record Element Type (Ret)

- Asignar a cada uno un nivel de complejidad (alta, media, baja) en función de la Cantidad de Det y Ret que tengan

Es por esto que resulta necesario conocer esta terminología:

#### **Terminología:**

**Entradas Internas (EI):** Procesos en los que se introducen datos y que suponen la actualización de cualquier archivo interno.

**Salidas Externas (EO):** Procesos en los que se envía datos al exterior de la aplicación.

**Consultas Externas (EQ):** Procesos consistentes en la combinación de una entrada y una salida, en el que la entrada no produce ningún cambio en ningún archivo y la salida no contiene información derivada.

**Archivos Lógicos Internos (ILF):** Grupos de datos relacionados entre sí internos al sistema.

**Archivos Lógicos Externos (EIF):** Grupos de datos que se mantienen externamente.

**Data Element Type (DET) :**Es una campo único (no repetitivo) reconocible por el usuario

**Record Element Type (RET):** Es un subconjunto de campos de un archivo, reconocible como tal por el usuario.

**File Type Referenced (FTR):** Es un tipo de archivo al que se hace referencia en una transacción; tiene que ser un ILF o EIF

Una vez identificados y contados los ILF y los EIF con sus DET y RET se debe determinar el nivel de complejidad de cada uno de acuerdo a la siguiente tabla:

**Tabla 1: Factores que determinan el nivel de complejidad**

	Para ILF / EIF	N° de Campos		
		1 a 19 DET	20 a 50 DET	51 o más DET
N° registros	1 RET	Baja	Baja	Media
	2 a 5 RET	Baja	Media	Alta
	6 o más RET	Media	Alta	Alta

Elaborado por James Elliot (2010)

### 2.1.3.2. Estimación del esfuerzo del proyecto

Se analizan y determinan todas las entradas (procesos que hacen llegar datos a la aplicación desde el exterior, desde un usuario u otra aplicación), Salidas (Son todos aquellos procesos que hacen llegar datos desde la aplicación hacia el exterior, a un usuario o a otra aplicación), Consultas (todos aquellos procesos que están formados por una combinación de entradas y salidas, produciendo una consulta a los datos), Ficheros de Interfaz (ficheros internos de otra aplicación. Una vez determinados e los ubica en la siguiente tabla dependiendo de su complejidad y los puntos de Función sin ajustar. (Carpio, 2008)

A continuación una tabla en donde se muestra el factor de Ponderación para hacer el cálculo de los puntos de función

**Tabla 2. Cálculo de los puntos de función**

Parámetro de Medición	Factor de ponderación				
	Cuenta	Simple	Medio	Complejo	
Número de entradas de usuario	<input type="checkbox"/> x	3	4	6	=
Número de salidas de usuario	<input type="checkbox"/> x	4	5	7	=
Número de peticiones de usuario	<input type="checkbox"/> x	3	4	6	=
Número de archivos	<input type="checkbox"/> x	7	10	15	=
Número de interfaces externas	<input type="checkbox"/> x	5	7	10	=
Cuenta=Total					

Elaborado por: **Castillo Carlos (2006)**

Según (Doria, 2006), existen 14 niveles de influencia que determinarán el factor de ajuste, niveles que se muestran a continuación:

**Tabla 3: Niveles de influencia para Factor de Ajuste**

NIVELES DE INFLUENCIA	
Comunicación de datos	0-5
Procesamiento distribuido	0-5
Objetivos de Rendimiento	0-5
Configuración del equipamiento	0-5
Tasa de transacciones	0-5
Entrada de datos en línea	0-5
Interfase con el usuario	0-5
Actualizaciones en línea	0-5
Procesamiento complejo	0-5
Reusabilidad del código	0-5
Facilidad de implementación	0-5
Facilidad de operación	0-5
Instalaciones múltiples	0-5
Facilidad de cambios	0-5
Factor de Ajuste	

Elaborado por : David Morales (2008)

Después todas esas puntuaciones se suman y con la siguiente ecuación se obtiene el factor de ajuste de complejidad (FA):

$$PFA = PFSA * (0.01 * FACTORAJUSTE) + 0.65$$

El valor de FA puede variar de 0.65 a 1.35.

Una vez calculado el FA ya calculamos los puntos de función (PF):

### 2.1.3.3. Estimación del tiempo del proyecto

Según (Doria, 2006), para estimar el tiempo del proyecto lo primero que se debe hacer es tomar en cuenta la siguiente tabla en donde se ubica el promedio de horas por punto de función según el lenguaje en el que se programe:

**Tabla 4: Horas por punto de Función según el lenguaje de programación**

Entorno y Lenguaje	Horas por punto de Función
Lenguajes 2G	25
Lenguajes 3G	15
Lenguajes 4 G	7

Se multiplica los puntos de función por el promedio de Horas por punto de Función, esto es:

$$\text{Esfuerzo H/H} = PF \times HPF$$

se divide el esfuerzo del proyecto para el número de horas que se trabaje al día, y luego se divide para el número de días que se trabaje por mes.

#### 2.1.3.4. Estimación del costo del proyecto

Para estimar el Costo del proyecto se multiplica el sueldo del programador por los meses que se demore, y a eso se le suma los otros gastos que genere el sistema.

#### 2.1.4. Diagrama de Casos de Uso

Es un modelo de sistema que contiene un conjunto de casos de uso y actores con sus relaciones respectivas.

“Un actor es una idealización de una persona externa, de un proceso, o de una cosa que interactúa con un sistema, un subsistema o una clase” (Booch, 2002)

Los casos de uso son todos los comportamientos que pueden ser descritos y manejados.

Los casos de uso se pueden conectar con diferentes tipos de líneas como se representan en la siguiente tabla:

**Tabla 5: tipos de relaciones de casos de uso**

RELACION	FUNCION	NOTACION
Asociación	La línea de comunicación entre un actor y un caso de uso en el que participa	_____
Extensión	La inserción de comportamiento adicional en un caso de uso base que no tiene conocimiento sobre él.	<<extend>> - - - - ->
Generalización	Una relación entre un caso de uso general y un caso de uso más específico, que hereda y añade propiedades a aquel.	_____▷
Inclusión	Inserción de comportamiento adicional en un caso de uso base, que describe explícitamente la inserción	<<include>> - - - - ->

**Elaborado por: Ajila Luis (2005)**

## 2.2. Marco Metodológico

Para empezar mencionaremos que para el presente estudio nos fundamentamos en la Metodología RUP, la cual posee las siguientes etapas:

- Requerimientos:
- Análisis y Diseño:
- Implementación:
- Pruebas:

### 1. Etapa de Requerimientos:

### a) Requisitos Funcionales y No Funcionales

Lo primero que se hizo en este caso es determinar los requisitos funcionales y no funcionales:

Estos requisitos se los especifican en las tablas a partir del anexo 2.

### b) Estimación del proyecto software por puntos de función

- Tomando en cuenta lo antes investigado lo primero que debemos hacer es identificar a los archivos, tomando como referencia los Requisitos, se le asigna a cada uno un tipo (ILF, EIF ) y se identifica la cantidad de Data Element Type (Det) y Record Element Type (Ret)

**Tabla 6:Conteos de puntos de función archivos lógicos interno**

ERS	FICHEROS LÓGICOS INTERNOS	DET	NÚMERO DE DET	NÚMERO DE RET	COMPLEJIDAD
REQ-01.1	Pedidos	Numero de Pedido Fecha de Pedido Código de Producto Descripción Unidad Cantidad Total de Items	7	2	Media
REQ-01.2	Producto	Código de Producto código de Categoría código de proveedor Stock Actual Nombres Unidad de Medida Precio de Proveedor Costo Iva Precio de Venta Utilidad Stock Mínimo	11	3	Alta
REQ-01.3	Categoría	código de categoría Nombre Detalle	3	1	Baja
REQ-01.4	Proveedor	Código de proveedor cedula nombres Representante Dirección Ciudad Teléfono Código postal fax	9	1	Baja

<b>REQ-01.5</b>	Local	código de local dirección ciudad teléfono fax distrito	6	1	Baja
<b>REQ-02,1</b>	Guía	Código de guía código de local Código de transportista código de producto Código de pedido fecha de salida Precio de venta cantidad	8	5	Alta
<b>REQ-02.2</b>	Transportista	Código de transportista Cédula de transportista nombre dirección teléfono	5	1	Baja
<b>REQ-03.1</b>	Orden	Código de la orden Código proveedor Fecha de entrega fecha de orden Código de producto cantidad solicitada cantidad recibida Precio proveedor	8	3	Alta

Elaborado por: Patricia García Ulloa

- Asignar a cada uno un nivel de complejidad (alta, media, baja) en función de la Cantidad de Det y REt que tengan. Ver anexo 4

Luego de obtener los puntos de función sin ajustar. Ahora se estima un factor de ajuste siguiendo también otros estándares y se hace el cálculo del esfuerzo requerido:

$$PFA = PFSA * (0.65 + (0.01 * \text{FACTORAJUSTE}))$$

$$PFA = 219 * (0.65 + (0.01 * 33))$$

$$PFA = 219 * 0.98 = 214.62$$

Este sistema se lo programará en Lenguaje de Cuarta generación, las horas promedio por punto de función en este lenguaje es 7, por lo tanto se multiplica 214 \*7 y se obtiene 1426.88. Este es el esfuerzo que se requiere.

$$\text{Esfuerzo} = PFA * \text{NUM.HORAS}$$

$$\text{Esfuerzo} = 214 * 7$$

$$\text{Esfuerzo} = 1498 \text{ h/h}$$

Se estima el tiempo:

$$\text{Tiempo} = (1498 / 8) / 20$$

**Tiempo= 9.3 meses**, es decir aproximadamente 9 meses teniendo un solo programador.

Para estimar el costo se multiplicará el sueldo de cada trabajador por los meses establecidos.

$$\text{Costo} = (\text{desarrolladores} * \text{duración} * \text{sueldo}) + \text{otros costos}$$

$$\text{Costo} = (1 * 9 * 500) + 5000$$

**Costo=9500 dólares**

### **Etapas de Análisis y Diseño del Sistema:**

Luego de estimar estos tres parámetros tan importantes realizamos el diagrama conceptual del problema, el cual queda establecido de la siguiente manera:

### **Diagrama Conceptual del Problema**

**Ver anexo 3**

### **Casos de uso extendido**

Dentro de esta etapa de Diseño también diagramamos los casos de uso extendido y el diagrama general de Casos de Uso. Para diagramarlos tomamos en cuenta los distintos procesos que se mencionan en el enunciado:

- Reposición de mercadería
- Despacho de mercadería desde el local central a una sucursal
- Solicitud de Mercadería a un proveedor
- Recepción de Mercadería

En estos procesos aparecen los siguientes actores:

- Bodeguero de la Sucursal
- Bodeguero del almacén central
- Encargado del departamento de Compras
- Proveedor

### **2.3. Resultados**

### **Requerimientos Funcionales y No funcionales**

Ver anexo 2

### **Estimación de Software**

Esfuerzo Requerido: 1498 horas por hombre

Tiempo del Proyecto: 9 meses con un solo desarrollador



Costo del Proyecto: 9500 dólares

## Diagramas de Caso de Uso

Ver anexo 5

### 3. CONCLUSIONES

- Se analizó con éxito un Sistema de Control de Mercadería mediante el Lenguaje UML y técnicas de estimación logrando el mejoramiento del proceso de Control de Productos de Consumo Masivo de la Empresa Supermercados S.A
- El análisis y el diseño tiene un papel importante para obtener mejores resultados en el desarrollo del sistema, pues consiste en planificar y tener una idea clara del problema y también de la solución
- Corroboramos que la estimación por puntos de función es muy eficiente para medir el esfuerzo, el tiempo y el costo de un proyecto de software, y es mucho más entendible para el usuario que otros métodos antes desarrollados.
- La determinación de requisitos Funcionales permite escudriñar los detalles que hará el sistema y favorece en la reducción de la cantidad de errores en el proceso.
- Los diagramas de Caso de Uso permiten ver un modelo gráfico entendible para cada uno de los usuarios.

### 4. REFERENCIAS BIBLIOGRAFICAS

- (1)Berroni, M. (2000). Lenguaje Modelado Unificado. En J. Iñiguez, *Herramientas del diseño de software* (pág. 200). España: Nuevas Hert.
- (2)Berru, D. (2004). *Casos de Uso extendido*. Barcelona: Mc Graw Hill.
- (3)Carpio, C. (2008). *Ingeniería de Programas*. Recuperado el 10 de Octubre de 2015, de [es.wikipedia.org/wiki/casosdeuso](http://es.wikipedia.org/wiki/casosdeuso)
- ( 4)Duran, S. (2006). Boletín de política Informática. *Informaticos*, 10.
- (5)Lluis, C. (2000). *Análisis Orientado a objetos*. Recuperado el 15 de Octubre de 2015, de <https://www.lsi.us.es/docs/informes/EstadoActual.pdf>
- (6)Pressman, R. (5 de Abril de 2002). *Ingeniería del Software enfoque práctico*. Recuperado el 5 de Octubre de 2015, de [es.wikipedia.org/wiki/RequisitoNoFuncional](http://es.wikipedia.org/wiki/RequisitoNoFuncional)
- (7)Stellman, A. (2010). *Proyectos aplicados a software*. Cambridge: Lectores activos.

# URKUND

## Urkund Analysis Result

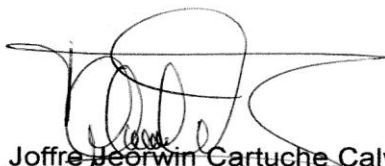
**Analysed Document:** ANALISIS Y DISEÑO DEL SISTEMA DE CONTROL DE  
MERCADERIA DE SUPERMERCADOS S.A. DE LA PROVINCIA DE  
EL ORO.docx (D16371509)  
**Submitted:** 2015-11-25 09:54:00  
**Submitted By:** pagu2005@hotmail.com  
**Significance:** 7 %

### Sources included in the report:

<http://www.pmainformatica.com/2015/04/estimacion-puntos-funcion-introduccion.html>  
<http://www.softwaremetrics.com/esp/>  
<https://prezi.com/uxaoyjxqxdsr/copy-of-copy-of-diagrama-de-caso-de-uso/>  
<https://prezi.com/ox8y9pmfjagf/diagrama-de-caso-de-uso/>  
<http://www.galeon.com/rcruz0423/docs/clase2p2.pdf>  
<https://andresvasquez.wordpress.com/2008/06/30/aprendiendo-uml-en-24-horas-libro/>  
<http://www.slideshare.net/guest79a3dc68/objeto-de-aprendizaje-3840895>  
[http://www.mat.uson.mx/mireles/AnalisisOO\\_archivos/slide0092.htm](http://www.mat.uson.mx/mireles/AnalisisOO_archivos/slide0092.htm)  
<http://www.slideshare.net/pervys/estimacin-software-por-puntos-de-funcin>  
<http://www.slideshare.net/luusanchez/estimacin-por-puntos-de-funcin>

### Instances where selected sources appear:

15



Ing. Joffre Jeorwin Cartuche Calva

C.I: 0703744193