



# Documento de Especificaciones, Requerimientos y Criterios de Aceptación de Software

## CONTENIDO

1.	INTRODUCCIÓN	2
1.1	Objetivos .....	3
1.1.1	Objetivo general	3
1.1.2	Objetivos específicos	3
1.1.3	Audiencia	3
1.2	Alcances y Limitaciones.....	4
1.3	Uso del Documento y Propiedad Intelectual .....	5
2.	DESCRIPCIÓN GENERAL	5
2.1	Especificación de funcionalidades .....	6
2.2	Viabilidades .....	6
2.3	Acuerdos con el cliente para la administración de requerimientos .....	6
3.	ARQUITECTURA DEL SISTEMA	7
3.1	Glosario de Términos y Abreviaturas.....	7
3.1.1	Arquitectura .....	7
3.1.2	Cliente .....	7
3.1.3	Servidor .....	7
3.1.4	Usuario .....	7
3.1.5	Modelo .....	7
3.1.6	LAN .....	7
3.1.7	WAN .....	8
3.2	Metas de Diseño .....	8
3.2.1	Sobre la Estructura Interna .....	8
3.2.2	Sobre la Seguridad .....	8
3.2.3	Sobre el Rendimiento.....	8
3.3	Características del sistema.....	9
3.3.1	De Diseño .....	9
3.3.2	De Tecnología.....	9
3.4	Descripción de la arquitectura .....	10

3.4.1	Modelo Cliente/Servidor .....	10
3.4.2	Estructura de la Arquitectura .....	11
3.4.3	Capas de Arquitectura .....	11
3.4.4	Ventajas de la Arquitectura .....	12
3.4.5	Desventajas de la Arquitectura .....	12
4.	ESPECIFICACIÓN DE REQUERIMIENTOS .....	13
4.1	Requerimientos funcionales .....	13
4.2	Requerimientos adicionales .....	14
4.3	Requerimientos No Funcionales .....	14
4.4	Requerimientos Técnicos .....	14
4.5	Requerimientos de Proceso .....	15
5.	ESTÁNDARES DE DESARROLLO .....	16
5.1	Estándar de variables .....	16
5.2	Convenciones de Codificación .....	16
5.3	Nomenclatura de Controles .....	17
5.4	Estandarización de Botones: .....	18
6.	ARTEFACTOS A ENTREGAR .....	20
7.	Anexos .....	21
7.1	Minuta de reunión .....	21
7.2	Entrevista .....	23

## 1. INTRODUCCIÓN

El documento de entrega especificaciones de requerimientos y criterios de aceptación de software (DERCAS), es un documento que se realiza previo al desarrollo de software, en la etapa de análisis.

La finalidad de la elaboración del DERCAS, es describir la especificación de un software en particular, el cual debe cumplir con ciertas funciones en un ambiente específico. Dentro del presente documento no se describe ningún detalle de diseño o implementación del producto, ya que estas labores no corresponden a la elaboración de un DERCAS y deben realizarse posteriormente en la etapa de diseño.

Para la elaboración del presente documento se baso el análisis de requerimientos en el estándar IEEE Std 830-1998, el mismo proporciona una guía de buenas prácticas para realizar el DERCAS apropiadamente y cubrir todos los ámbitos necesarios, previo a diseñar el sistema.

El DERCAS debe comunicar de manera clara, precisa y sin ambigüedades los requerimientos, objetivos y presunciones del sistema. Entonces, ¿Hacia quién está dirigido? Los lectores del DERCAS comprenden clientes y usuarios, analistas, desarrolladores, testers, gerentes, equipo de soporte, Etc. Es por tan amplia y variada audiencia de lectores por la cual el documento debe ser elaborado de la manera más clara posible.

## 1.1 Objetivos

### 1.1.1 Objetivo general

Integrar y describir detalladamente en el presente documento los principales requerimientos que serán implementados, en base al acuerdo entre ambas partes. Y servir como garantía para Universal Language y como respaldo para Professional Systems S.A de que únicamente los requisitos plasmados en este documento deben y serán realizados y no podrán modificarse después de ser aprobado y firmado este documento.

### 1.1.2 Obetivos específicos

- Presentar el enfoque de uso del presente documento de diseño a través de la definición de alcances, límites y lineamientos de uso.
- Normalizar la aplicación de los términos empleados en el presente documento.
- Definir metas de diseño bajo las cuales se procederá a construir el sistema UniSys.
- Definir las principales características de diseño y técnicas del sistema UniSys.
- Describir el marco general del sistema UniSys.
- Identificar y describir todos los componentes de software del sistema UniSys.
- Describir la totalidad de los casos de uso del sistema UniSys.
- Normalizar la fase de desarrollo bajo criterios definidos de estandarización.
- Describir la estructura de la arquitectura en la cual se implementará el sistema UniSys.

### 1.1.3 Audiencia

Esta sección describe brevemente la audiencia a la qué va dirigida el documento, tener en cuenta desde el inicio de la elaboración del documento, nos permite visualizar en qué dirección elaborar el documento, ya que teniendo en cuenta quienes deben poder leer el documento sin problemas podemos determinar qué tipo de redacción es apropiada utilizar.

El presente documento debe ser entendible para todo tipo de audiencia, a continuación se detallan los posibles tipos de lectores y sus objetivos.

- *Clientes y usuarios:* Interesados en validar objetivos del sistema y descripción de alto nivel de la funcionalidad. Generalmente no están interesados en los requerimientos detallados del sistema.
- *Analistas (de Sistemas y de Requerimientos):* Deben revisar y validar que los requerimientos sean factibles, viables, funcionales/no funcionales. Así mismo de

ser necesario son los encargados de escribir especificaciones de otros sistemas que se integren con el sistema en cuestión.

- *Desarrolladores:* Ya que son los encargados de implementar los requerimientos y es preciso asegurarnos desde el inicio que tengan claro, qué es lo que deben implementar.
- *Tester:* Deben validar la satisfacción de los requerimientos.
- *Gerentes:* Deben tener conocimiento del proyecto, para poder medir, controlar y dar seguimiento al proceso de desarrollo.
- *Equipo de infraestructura y operaciones:* Deben dimensionar los equipos a utilizar para la implementación y así mismo establecer procedimientos de rutina.
- *Equipo de soporte de usuario:* Encargados de desarrollar un plan de capacitación de usuario, generar manuales de usuario final y de definir procedimientos para proporcionar soporte online.

## 1.2 Alcances y Limitaciones

Los alcances del presente documento son:

- Estructura y funcionalidad interna del sistema AgroSys, sin especificar los requerimientos e infraestructura de seguridad a nivel de conectividad de redes (VPN, esquemas DMZ, etc.).
- Metodología de desarrollo a utilizar sin especificar las bases de la planificación del desarrollo del sistema (calendarización de actividades, distribución de recursos, etc.)
- Estandarización para el desarrollo del sistema AgroSys, sin especificar ningún tipo de manual u otro documento de guía para el desarrollo del sistema.

Las limitaciones para la elaboración de la arquitectura son:

- Tiempo limitado para la fase de construcción e implementación para todos los procesos del giro de negocios.
- No se establece un manual de procedimientos administrativos para el mantenimiento del sistema (políticas de seguridad, backups, etc.).

### 1.3 Uso del Documento y Propiedad Intelectual

Este documento contiene material confidencial de uso exclusivo entre el personal ejecutivo de AgroGuate S.A. y el personal de TI de ConatusSys en su papel de desarrollador del software.

El presente documento **debe** utilizarse únicamente con los siguientes propósitos:

- Como manual de lineamientos y referencia bibliográfica durante las etapas de elaboración, construcción, desarrollo, implementación del software y cualquier futura agregación o modificación.
- Como consulta sobre la estructura de la arquitectura del sistema.
- Como guía y control de los requerimientos establecidos.
- Como punto de referencia para la toma de decisiones sobre futuras agregaciones o modificaciones al sistema AgroSys después de entregada la versión contemplada en este documento.

El presente documento **no debe** utilizarse para los siguientes fines:

- Como manual de usuario para la administración del sistema (no se establecen directrices que engloben funciones administrativas para el mantenimiento del sistema).
- Cualquier otra forma de uso no especificada en esta parte.

## 2. DESCRIPCIÓN GENERAL

El presente documento contiene los requerimientos del sistema a desarrollar para la organización AgroGuate S.A. Describe el enfoque de la utilización del sistema, así como las expectativas del mismo.

AgroSys es un sistema ad hoc, diseñado, desarrollado e implementado para AgroGuate, que proporciona una solución eficiente, segura y eficaz para la gestión y control de la información de sus productos, clientes, proveedores e inventario.

## **2.1 Especificación de funcionalidades**

Para obtener la especificación de las funcionalidades esperadas por el cliente, se hizo uso de varios métodos para obtener un panorama general y lo más claro posible del resultado final que se desea obtener, como resultado de las diversas reuniones con el cliente se realizaron minutas y así mismo se realizaron entrevistas a usuarios de distintos niveles de la organización, para obtener un panorama general de la situación actual y la situación esperada. (Anexo 1 y 2, respectivamente)

## **2.2 Viabilidades**

En esta sección se realizó el análisis de viabilidad técnica, económica y operativa de la realización e implementación del proyecto, esta fase es de suma importancia ya que nos permite visualizar las diversas limitaciones que se le pueden presentar al proyecto, ya sea falta de capacidad del usuario final para interactuar adecuadamente con el sistema, infraestructura inadecuada de la organización para soportar la solución, hasta el factor económico.

Se adjunta el documento de análisis de viabilidad, realizado en una fase previa del proceso.

## **2.3 Acuerdos con el cliente para la administración de requerimientos**

Una vez especificados y aceptados los requerimientos por parte del Gerente General de la empresa AgroGuate S.A, estos no podrán ser modificados o agregar nuevos requerimientos al sistema AgroSys, ya que agregarlos significaría un retraso en las actividades planificadas según el cronograma de trabajo y esto implicaría la no culminación del proyecto en su totalidad y en el tiempo especificado. Por ende, el Gerente General debe revisar a detalle y con mucha precaución el presente documento antes de aceptar los términos aquí expuestos.

Inicialmente, se hizo un levantamiento general de requerimientos y expectativas del cliente. Posteriormente se acordaron diversos puntos con el cliente en relación a la toma de requerimientos; los cuales se listan a continuación.

- El personal de Agroguate S.A. se compromete a tener disposición para atender a los consultores y apoyarles, para afinar los detalles de los requerimientos.
- Posterior a reuniones de toma de requerimientos ConatusSys, debe elaborar un documento formal, en donde se listen formalmente los requerimientos recabados.
- Agroguate S.A. en conjunto con ConatusSys, debe revisar el documento y especificación de cada uno de los requerimientos, realizar cambios en



común acuerdo y documentar cada uno de los cambios realizados y la razón del cambio de los mismos.

### **3. ARQUITECTURA DEL SISTEMA**

La arquitectura del Software es la organización fundamental del sistema que incluye a sus componentes, sus relaciones entre ellos y el ambiente y los principios que dictan su diseño y evolución.

#### **3.1 Glosario de Términos y Abreviaturas**

##### **3.1.1 Arquitectura**

La arquitectura de software define, de manera abstracta, los componentes que llevan a cabo alguna tarea de computación, sus interfaces y la comunicación entre ellos.

##### **3.1.2 Cliente**

Inicia el diálogo o la solicitud de un servicio, generalmente son computadoras personales o estaciones de trabajo con capacidades limitadas para el procesamiento de información.

##### **3.1.3 Servidor**

Computadora o conjunto de ellas que atienden las solicitudes realizadas por los clientes.

##### **3.1.4 Usuario**

Persona que utiliza o trabaja con algún componente de la arquitectura del sistema y que tiene autorización para hacerlo.

##### **3.1.5 Modelo**

Un modelo es una representación de un objeto, sistema o idea.

##### **3.1.6 LAN**

LAN son las siglas de Local Area Network, Red de área local. Una LAN es una red que conecta los ordenadores en un área relativamente pequeña y predeterminada (como una habitación, un edificio, o un conjunto de edificios).

### 3.1.7 WAN

Las Redes de área amplia (WAN) son redes informáticas que se extienden sobre un área geográfica extensa.

## 3.2 Metas de Diseño

La arquitectura del sistema se diseñará en torno a metas específicas a cumplir, las cuales describen las restricciones mínimas de aceptación para el funcionamiento del sistema. Éstas se presentan a continuación, agrupadas en tres (3) categorías:

### 3.2.1 Sobre la Estructura Interna

- Se debe modular en forma óptima, la programación de los componentes involucrados.
- La reutilización de código (en funciones, componentes, declaración de constantes, etc.) es crítica para el desarrollo exitoso.
- La separación e independencia de funciones también es crítica para el desarrollo exitoso del sistema.
- Minimizar la redundancia de información en el modelo entidad-relación de las bases de datos.

### 3.2.2 Sobre la Seguridad

- Se debe dejar constancia de todo mensaje enviado y recibido en forma de una bitácora específica por cada parte.
- Únicamente las personas las cuales tengan un usuario definido en el sistema podrán entrar al mismo.
- El sistema debe denegar el acceso a cualquier función no autorizada, según los privilegios asignados al usuario.
- Las bitácoras deberán cumplir con los siguientes requerimientos:
  - Permitir a los administradores del sistema comprender las transacciones y accesos fácilmente.
  - Permitir encontrar situaciones extrañas o poco comunes en las transacciones diarias de los usuarios.
  - Buscar por usuarios todo el record de sus actividades.

### 3.2.3 Sobre el Rendimiento

- El tiempo de respuesta de cualquier transacción iniciada debe tardarse a lo más 40 segundos. A excepción de la generación y acceso a reportes o a ficheros externos, cuya tasa de respuesta variará dependiendo de las capacidades de la computadora que realice la solicitud al servidor.

- Los componentes de software identificados en la arquitectura del sistema en los sitios locales, deberán ser livianos (eficientes en tamaño, manejo de recursos y un mínimo aceptable de referencias a otros componentes) y simples en su funcionamiento. Esto con el fin de evitar posibles causas de fallo en el sistema del servidor.
- Cualquier agregación de funciones al sistema o cambios en ciertas configuraciones del sistema en un futuro podrá afectar en una escala mínima el rendimiento del sistema. Sin embargo la agregación de máquinas cliente que utilicen el sistema podría afectar a un futuro el rendimiento si su tasa de transacciones es elevada. Según el análisis de los procesos de negocio de AgroGuate S.A., la tasa de transacciones actualmente es mínima y el riesgo de saturación de transacciones es bajo.

### 3.3 Características del sistema

A continuación se listan las principales características y aspectos uniformes que describen detalles de la arquitectura general del sistema.

#### 3.3.1 De Diseño

Las principales características de diseño del sistema son:

- El sistema trabajara bajo una arquitectura Cliente/Servidor.
- Se verificará y registrará la información correspondiente a la configuración de permisos y accesos de cada usuario, por cuenta y transacción del sistema.
- Repositorio de información o Base de Datos centralizada.

#### 3.3.2 De Tecnología

Las principales características técnicas son:

- Las tecnologías empleadas en la arquitectura del sistema son:
  - Programación y desarrollo: Visual Basic .Net
  - Comunicaciones y redes: HTTP.
- La comunicación entre las partes se basará en el modelo de referencia TCP/IP, utilizando el protocolo HTTP como medio de transporte.
- Respecto al software a utilizar, se listan las siguientes características:
  - La arquitectura del sistema se basará en tecnologías implementadas por Microsoft Corp.
  - La plataforma de los sistemas operativos de los servidores en donde se instalarán los componentes del sistema, deberá corresponder a una arquitectura con procesadores de la familia Intel x86 o compatibles de 32-bits y sistema operativo de la familia MS Windows XP o mayor. En caso de contar con otro sistema operativo, se procederá a instalar una máquina

virtual con un sistema operativo afín a la arquitectura. Para ello se utilizará VMWare o aplicación similar para correr la máquina virtual.

- Las herramientas de desarrollo del sistema se basan en tecnologías de Microsoft Corp.

### 3.4 Descripción de la arquitectura

Debido a las necesidades de hacer el sistema accesible para los diferentes usuarios, se decidió seguir la arquitectura Cliente-Servidor, la cual aumenta las capacidades de la arquitectura conocida como Modelo Cliente/Servidor y a su vez organiza todos los componentes que conforman al sistema según su función.

Para realizar el modelo de programación se decidió utilizar un modelo de capas, este tipo de arquitectura nos permite separar la capa de aplicación de la capa donde se encuentra almacenada la información, en este tipo de arquitectura los requerimientos de la máquina cliente, son mínimos ya que únicamente realizan la petición al servidor y es el servidor quien se encarga de realizar la transacción.

#### 3.4.1 Modelo Cliente/Servidor

La arquitectura Cliente/Servidor es el modelo más utilizado para desarrollar aplicaciones. Es un modelo para el desarrollo de aplicaciones en el que las operaciones a realizar se dividen en procesos independientes que cooperan entre sí para proporcionar un servicio o una respuesta.

Este tipo de arquitectura consta de tres componentes: al proceso el cual inicia el diálogo o la solicitud de un servicio se le denomina Cliente, generalmente son computadoras personales o estaciones de trabajo con capacidades limitadas para el procesamiento de información. El proceso que atiende a las solicitudes realizadas por los clientes se le denomina Servidor, las cuales son computadoras más poderosas que los Clientes. El último componente de esta arquitectura es el medio físico mediante el cual se comunican los Clientes con el Servidor, estos medios son generalmente redes de área local (LAN) o redes de área amplia (WAN). En este caso se utilizara una red inalámbrica de área local (WLAN).

En este modelo se resaltan las principales funciones tanto de los clientes como de los servidores. Los Clientes son los procesos que interactúan con el usuario y generalmente lo hacen de forma gráfica mediante Interfaces Gráficas de Usuario (GUI) y son estos últimos quienes envían peticiones al Servidor y reciben la respuesta.

Algunas de las funciones de los Clientes son:

- Enviar solicitudes de procesamiento a Servidores.

- Proporcionar información necesaria al Servidor para atender las peticiones.
- Manejo de Interfaces Graficas de Usuario.
- Captura y validación de datos

Por otra parte, los servidores son quienes computan las solicitudes hechas por los Clientes y regresan el resultado del cómputo de datos a cada Cliente. Algunas de las funciones del Servidor son:

- Recepción de solicitudes de Clientes.
- Cómputo de datos solicitados por el Cliente.
- Acceso a Bases de Datos.

Las principales características de la arquitectura Cliente/Servidor son:

- El Servidor presenta a todos los Clientes una Interfaz única y bien definida.
- El Cliente no necesita saber la lógica y el funcionamiento del Servidor.
- El Cliente no depende de nada del Servidor, ni su ubicación, ni su tipo de equipo, ni su sistema operativo.
- Los cambios en el Servidor implican pocos o ningún cambio a los Clientes.
- Un Servidor puede atender a uno o más Clientes.

### 3.4.2 Estructura de la Arquitectura

La arquitectura Cliente/Servidor funciona bajo la siguiente estructura:

### 3.4.3 Capas de Arquitectura

La arquitectura a utilizar se basa en la distribución o separación de la capa de presentación y la capa de servicio o lógica de negocio, en la cual los diferentes componentes de software involucrados en el sistema, pueden comunicarse con un servidor de base de datos en el cual pueden realizar sus operaciones diarias de inserción, modificación, consulta o eliminación.

En el Servidor radicará toda la lógica del funcionamiento general del sistema, así como la base de datos central y repositorios de propósito general sobre la comunicación con los diferentes componentes.

En el servidor radicará una base de datos local con información de configuración de permisos y accesos.

#### 3.4.4 Ventajas de la Arquitectura

- Centralización del control: los accesos, recursos y la integridad de los datos son controlados por el servidor de forma que un programa cliente defectuoso o no autorizado no pueda dañar el sistema. Esta centralización también facilita la tarea de poner al día datos u otros recursos.
- Escalabilidad: se puede aumentar la capacidad de clientes y servidores por separado. Cualquier elemento puede ser aumentado (o mejorado) en cualquier momento, o se pueden añadir nuevos nodos a la red (clientes y/o servidores).
- Fácil mantenimiento: al estar distribuidas las funciones y responsabilidades entre varios ordenadores independientes, es posible reemplazar, reparar, actualizar, o incluso trasladar un servidor, mientras que sus clientes no se verán afectados por ese cambio (o se afectarán mínimamente). Esta independencia de los cambios también se conoce como encapsulación.
- Existen tecnologías, suficientemente desarrolladas, diseñadas para el paradigma de C/S que aseguran la seguridad en las transacciones, la amigabilidad de la interfaz, y la facilidad de empleo.

#### 3.4.5 Desventajas de la Arquitectura

- La congestión del tráfico ha sido siempre un problema en el paradigma de C/S. Cuando una gran cantidad de clientes envían peticiones simultáneas al mismo servidor, puede ser que cause muchos problemas para éste (a mayor número de clientes, más problemas para el servidor).
- Si el servidor está caído, las peticiones de los clientes no pueden ser satisfechas.
- El software y el hardware de un servidor son generalmente muy determinantes. Un hardware regular de un ordenador personal puede no poder servir a cierta cantidad de clientes. Normalmente se necesita software y hardware específico, sobre todo en el lado del servidor, para satisfacer el trabajo. Por supuesto, esto aumentará el costo.
- El cliente no dispone de los recursos que puedan existir en el servidor.

## 4. ESPECIFICACIÓN DE REQUERIMIENTOS

Los requerimientos son aquellas características específicas, con las que el software DEBE contar y DEBE cumplir, para ser aceptado por el cliente.

### 4.1 Requerimientos funcionales

- Identificación y autenticación del usuario en el sistema y llenado de bitácoras de acceso y transacciones de todas las operaciones que éste realice.
- Ingresar nuevo producto y tipos de productos al sistema, modificar sus datos o dar de baja al mismo.
- Ingresar nuevo proveedor al sistema, modificar sus datos o dar de baja al mismo.
- Ingresar nuevo cliente al sistema, modificar sus datos o dar de baja al mismo.
- Ingresar un registro de empleados al sistema, modificar sus datos o dar de baja al mismo.
- Registrar el pago de una factura, así como el tipo de pago, la cabecera y el detalle de la misma.
- Seguridad de acceso discrecional de primer nivel (bloqueo y asignación de privilegios por menú y no por función). Manejo de perfiles, usuarios y consulta de bitácoras.
- El usuario debe tener posibilidad de realizar consultas de existencia de productos online.
- Gestionar inventario multibodega.
- El sistema debe permitir dar de baja del inventario los productos al momento de ser retirados de bodega o almacén
- Debe permitir almacenar información de los usuarios
- Almacenar y generar órdenes de compra
- Debe ser capaz de consultar el historial de comprar de un comprador específico
- El sistema debe permitir almacenar el medio por el cual el consumidor realiza el pago.
- Control de fecha de caducidad de los productos.
- Debe permitir identificar productos con mayor demanda.
- Debe permitir identificar productos con menor demanda.
- Debe facilitar la ubicación de los productos.

- Control de cuentas por cobrar.

#### 4.2 Requerimientos adicionales

- El sistema debe permitir imprimir los documentos.
- Gestión de accesos al sistema por medio del manejo de una bitácora de accesos.
- Creación del sistema sin migración de datos, es decir, que AgroGuate S.A. asignará al personal apropiado para realizar la digitación de toda su información pasada y futura.

#### 4.3 Requerimientos No Funcionales

- Tamaño estándar de las interfaces de usuario para una facilidad de navegación.
- Combobox automáticos para facilitar el llenado de datos.
- Botones de minimizar y maximizar en pantalla general y subpantallas a excepción de cuadros de diálogos.
- Colores, iconos, botones, tamaños y letras definidas por AgroGuate S.A. Ver sección de Estándares de Desarrollo.

#### 4.4 Requerimientos Técnicos

- Para el correcto funcionamiento del sistema se debe contar con los siguientes sistemas operativos:
  - Windows XP service pack 3
  - Windows 7
- En caso de que no se contara con ninguno de los sistemas operativos antes mencionados, se procederá a instalar una máquina virtual, con alguno de los sistemas operativos compatibles con el sistema AgroSys.
- Requisitos mínimos y máximos de software y hardware para instalar el sistema:

MÍNIMO	MÁXIMO
Windows XP sp3	Windows 7
32 bits	64 bits
Frameworks 3	Frameworks 4
Windows installer 3.1	Windows installer 3.1
Procesador Intel Pentium 4 ó Core 2 duo	Procesadores core i 7
512 de RAM mínima	2 Gb máximo
MySQL server 5.1	MySQL server 5.1



Monitor	Monitor
Teclado	Teclado
Mouse	Mouse
CD-room	CD-room

#### 4.5 Requerimientos de Proceso

La metodología de desarrollo utilizada para la creación del sistema SUL será RUP. A continuación se proporcionan detalles de dicha metodología:

- Las siglas RUP en ingles significa Rational Unified Process (Proceso Unificado de Rational) es una metodología para el desarrollo de software que proporciona un enfoque disciplinado para asignar tareas y responsabilidades dentro de una organización de desarrollo. Su meta es asegurar la producción de software de alta calidad que resuelva las necesidades de los usuarios dentro de un presupuesto y tiempo establecidos.
- El ciclo de vida de RUP; RUP divide el proceso en 4 fases, dentro de las cuales se realizan varias iteraciones en número variable según el proyecto y en las que se hace un mayor o menor hincapié en los distintas actividades. En las iteraciones de cada fase se hacen diferentes esfuerzos en diferentes actividades
  - Inicio: Se hace un plan de fases, se identifican los principales casos de uso y se identifican los riesgos. Se define el alcance del proyecto
  - Elaboración: se hace un plan de proyecto, se completan los casos de uso y se eliminan los riesgos
  - Construcción: se concentra en la elaboración de un producto totalmente operativo y eficiente y el manual de usuario
  - Transición: se instala el producto en el cliente y se entrena a los usuarios. Como consecuencia de esto suelen surgir nuevos requisitos a ser analizados.
- En RUP se definen nueve flujos de trabajo distintos, separados en dos grupos.
- Modelado del negocio.
  - Requisitos.
  - Análisis y diseño.
  - Implementación.
  - Pruebas
  - Despliegue.
- Los flujos de trabajo de apoyo son:
  - Administración del proyecto.
  - Configuración y control de cambios.
  - Entorno.

- Dichos flujos de trabajo serán visitados una y otra vez a lo largo de todo el proceso.

## 5. ESTÁNDARES DE DESARROLLO

### 5.1 Estándar de variables

Para la utilización de variables se define el siguiente estándar:

- Todas las variables deben ser declaradas antes de ser utilizadas
- Las primeras 3 letras de la variable, deben indentificar el módulo al que pertenecen, por ejemplo:
- La segunda parte de las variables, en caso sea posible, debe evitarse utilizar abreviaturas, con la finalidad de hacer el código lo más entendible posible.
- El nombre de las variables debe escribirse con minúsculas y separar la primera parte de la variable, con la segunda con guión bajo (\_).

### 5.2 Convenciones de Codificación

- Todas las sentencias de control, deben contar con su respectiva llave de apertura y de cierre.
- Separar con un espacio en blanco las condiciones en una sentencia de control.
- Uso de comentarios: Utilizar comentarios en casos necesarios, no abusar del uso de los mismos, ya que esto únicamente puede dificultar la lectura del código.
- Cada línea debe contener una única instrucción
- Utilización de líneas en blanco, las líneas en blanco ayudan a mejorar la legibilidad entre secciones de código que están relacionadas lógicamente.
- Evitar que el ancho de una línea exceda de 80 caracteres, no es una cláusula restrictiva pero es recomendable.

### 5.3 Nomenclatura de Controles

Nombre de Control	Abreviatura	Ejemplo
Button	Btn	btnBoton
CheckBox	Ckb	ckbCheque
CheckedListBox	Cklb	cklbListaCheque
ComboBox	Cmb	cmbListaDesplegable
ContextMenuStrip	Cms	cmsMenuContexto
DataGridView	Dgv	dgvConsulta
DataSet	Ds	dsDatos
DateTimePicker	Dtp	dtpFecha
Form	Frm	frmForma
GroupBox	Gb	gbPanel
HelpProvider	Hp	hpAyuda
Label	Lbl	lblTitulo
LinkLabel	Lnkl	lnklHipervinculo
ListBox	Lb	lbListaCuadro
MenuStrip	Ms	msMenuOpcion
MonthCalendar	Mc	mcMes
NotifyIcon	Ni	niNotificacion
Panel	Pnl	pnlPanel
PictureBox	Pctb	pctbImagen
ProgressBar	Pgrb	pgrbBarraProgreso
RadioButton	Rbtn	rbtnOpcion
RichTextBox	Rtb	rtbDocumento
TabControl	Tc	tcTabulador
TextBox	Txt	txtTexto
Timer	Tmr	tmrTiempo
ToolStrip	Ts	tsBarra
ToolStripButton	Tsb	tsbBoton
ToolStripContainer	Tsc	tscContenedor
ToolStripLabel	Tsl	tslTitulo
ToolTip	Tt	ttPista

#### 5.4 Estandarización de Botones:



Ayuda: Al presionar este botón, el usuario podrá acceder fácilmente a los temas de ayuda para consultar el funcionamiento de cualquier módulo del sistema.



Buscar: Al presionar este botón, el sistema buscará conjunto de datos, según el o los filtros aplicados.



Cancelar: Al presionar este botón, el sistema cancelará la inserción o edición que se iba a llevar de un registro.



Iniciar Sesión: Al presionar este botón, el usuario podrá ingresar al sistema, siempre y cuando haya ingresado un usuario y contraseña correcta.



Cerrar Sesión: Al presionar este botón, el usuario podrá cerrar la sesión e iniciar una nueva sesión. El sistema preguntará si el usuario está seguro de cerrar o no su sesión.



Editar: Al presionar este botón, el sistema mostrará los campos del registro a modificar, y permitirá al usuario editarlos para luego guardarlos.



Eliminar: Al presionar este botón, el sistema preguntará al usuario si está seguro de eliminar el registro seleccionado. En caso que confirme el usuario, el registro será eliminado.



Guardar: Al presionar este botón, el sistema guardará cualquier nuevo registro, o los cambios realizados de un registro.



Nuevo: Al presionar este botón, el usuario podrá ingresar un nuevo registro.

## 6. ARTEFACTOS A ENTREGAR

- Sistema AgroSys
  - Módulo Empleados
  - Módulo Ventas
  - Módulo de Compras
  - Facturación
  - Sucursales
  - Módulo de Seguridad
  - Módulo de Clientes
  - Módulo de Proveedores
  - Inventario
- 1 Manual de usuario y 1 Manual técnico
- Viabilidad técnica, económica y legal
- Minutas de Reunión
- DERCAS
- Diagramas:
  - Diagrama de flujo de datos
  - Diagrama cero
  - Diagrama hijos
  - Diagramas Casos de uso
  - Diagramas de actividad
  - Diagrama de secuencia
  - Diagrama de componentes
- Script Base de Datos
- Máquina Virtual en S.O no compatibles
- Utilidades para la implementación del sistema

## 7. Anexos

### 7.1 Minuta de reunión

INFORMACIÓN			
<b>NOMBRE DEL PROYECTO:</b>	Implementación Automatización	<b>CÓDIGO DEL PROYECTO:</b>	1.2
<b>PREPARADA POR:</b>	Magaly Velásquez/Rebeca Carrera	<b>FECHA:</b>	27/09/2013
<b>HORA INICIO :</b>	02:30pm	<b>HORA FIN:</b>	05:00pm

#### 1. PROPÓSITO DE LA REUNIÓN

- Reunión toma de requerimientos 1

#### 2. ASISTENCIA A LA REUNIÓN

Nombre	Empresa/Departamento	e-mail	Asistencia
Ramiro González	Agroguate S.A./ Administración		P
Julia Estrada	Agroguate S.A./ Operaciones		P
Mario Aguirre	Agroguate S.A./ Comercial		P
Rodolfo Chacón	Agroguate S.A/ Producción		P
Rebeca Carrera	ConatusSys		P
Magaly Velásquez	ConatusSys		P

\*Códigos para asistencia: P: Presente, T: Tardía, R: Representante, A: Ausente

#### 3. AGENDA

- Toma de requerimientos funcionales y no funcionales

#### 4. ASUNTOS TRATADOS

- Requerimientos.

#### 5. ACUERDOS

##### Requerimientos

- La organización debe adaptar sus procesos al software que utilizan actualmente, y se requiere que sea el

software el que deba adaptarse a los procesos de la organización.

- La información debe encontrarse actualizada en todo momento y las operaciones deben ser atómicas.
- El sistema de control de inventarios debe ser eficiente y multibodega, es decir que debe permitir consultar existencias online de las demás sucursales remotamente.
- Control de clientes.
- ConatusSys, debe estudiar cada uno de los distintos procesos y adaptar el SW a los procesos de la organización.
- El sistema requerido debe permitir la comunicación e integración de los resultados de cada uno de los distintos departamentos que conforman Agroguate S.A.
- Debe ser una herramienta de apoyo en la alineación de procesos y objetivos de Agroguate S.A.
- Debe contar con control de accesos y permitir restringir el acceso a la información de acuerdo a las responsabilidades de cada empleado.
- Optimizar el tiempo de respuesta de obtención de información puntual a través del sistema
- El cliente no considera necesario que el sistema sea multimonedas, ya que todas sus transacciones las realizan en quetzales.
- Generación de distintos niveles de reportes financieros, de inventarios, entre otros.

#### 6. PENDIENTES

Acción	Asignado a	Fecha límite	Estado
Realizar entrevistas a los usuarios de Agroguate S.A.	Rebeca Carrera	01/10/2013	Pendiente
Especificación formal de requerimientos.	Magaly Velásquez	01/10/2013	Pendiente

#### 7. APROBADO POR

	Nombre	Firma	Fecha
Gerente de Proyecto : ConatusSys	Magaly Velásquez		30/09/2013
Gerente de Proyecto del cliente:	Ramiro González		30/09/2013

Este documento será enviado vía correo electrónico a todos los participantes, solicitándoles manifestar su acuerdo o en caso contrario sus observaciones a lo registrado en la misma, de no recibirse observaciones o acuerdos en las próximas 24 horas, se estará dando por válida la minuta.



## 7.2 Entrevista

### Entrevista toma de requerimientos

1. ¿Desde su punto de vista, cuál es el problema a resolver?
2. ¿Quién será el usuario final de la aplicación?
3. ¿Cuenta el usuario final con los conocimientos necesarios de computación para utilizar adecuadamente la herramienta?
4. ¿Qué es lo que espera obtener del sistema? ¿Cómo espera que el software impacte en las actividades diarias de la organización?
5. ¿Hay algún ente regulador, ya sea fiscal o auditor que intervenga en alguna actividad ligada con alguna parte del sistema?
6. ¿Qué tipo de información debe ser procesada por el sistema y qué debe hacer el sistema con la misma?
7. ¿Qué tipo de interacción con el usuario debe proporcionar el sistema?
8. ¿Cómo podría caracterizarse un buen resultado generado por la solución?
9. ¿Cuáles problemas debería atacar esta solución?
10. ¿Podría usted describir o mostrar el ambiente de negocios en el que se utilizará la solución?
11. ¿Los aspectos especiales del desempeño o las restricciones afectarán la forma en que se busque la solución?
12. ¿Cuál será el beneficio de una solución exitosa?