



UNIVERSIDAD SIMÓN BOLÍVAR  
DECANATO DE ESTUDIOS PROFESIONALES  
COORDINACIÓN DE INGENIERÍA DE LA COMPUTACIÓN

**CONFIGURACIÓN DE SAP ERP PARA UN MODELO DE INDUSTRIA DE  
CONSUMO MASIVO (CPG)**

Por:  
JULIO DE ABREU MOLINA

Realizado con la asesoría de:  
KENYER DOMINGUEZ

PASANTÍA LARGA  
Presentado ante la Ilustre Universidad Simón Bolívar  
como requisito parcial para optar al título de  
Ingeniero en Computación

**Sartenejas, Septiembre del 2013**



UNIVERSIDAD SIMÓN BOLÍVAR  
DECANATO DE ESTUDIOS PROFESIONALES  
COORDINACIÓN DE INGENIERÍA DE LA COMPUTACIÓN

ACTA FINAL PASANTÍA LARGA

**CONFIGURACIÓN DE SAP ERP PARA UN MODELO DE INDUSTRIA DE  
CONSUMO MASIVO (CPG)**

Presentado por:  
**JULIO DE ABREU MOLINA**

Esta Pasantía Larga ha sido aprobado por el siguiente jurado examinador:

---

KENYER DOMINGUEZ

---

PROFESOR 2

---

ANA CECILIA GARCIA REVERÓN

**Sartenejas, FECHA (dd/mm/aa)**

## Resumen

AQUÍ VA EL CONTENIDO DEL RESUMEN.

Como parte de este sistema, se planteó la evaluación de un método de detección de ataques de denegación de servicio mediante la estimación del parametro de Hurst utilizando el mecanismo de ventanas deslizantes. Los métodos se incorporaron en una herramienta de línea de comando versátil capaz de obtener series de tiempo apartir de una traza de datos tomados de la red y de graficar las estimaciones del parámetro de Hurst. La herramienta fue evaluada mediante el uso del algoritmo de Paxson con respecto a su precisión y con trazas de los escenarios de ataques de denegación de servicio producidas en el laboratorio Lincoln del *Massachusetts Institute of Technology*, durante la evaluación dirigida por la *Defense Advanced Research Projects Agency*, con respecto a su capacidad de detección de ataques. Los resultados muestran que el método de detección se puede implementar para su uso en tiempo real y se puede mejorar con respecto a la detección de ataques.

A mis padres...

# Agradecimientos

A Dios primeramente porque sin él no hubiese llegado a donde me encuentro ahora, por haberme permitido estudiar en la USB, por el milagro que me concedió al haber logrado algo que era prácticamente imposible para la mente humana: que la Universidad me haya dejado no sólo hacer el cambio de carrera, sino además escoger la carrera de mi elección en tan sólo los 6 meses.

A mis padres primero por darme el regalo de la vida. Segundo, porque me han dado todo su amor incondicional, y por hacerme un hombre de bien, por apoyarme en todo lo que hago, y por siempre creer en mí, en que podía llegar hasta este punto.

A mis hermanas Julianna y Juliet,

A mi tutor académico, el Profesor Kenyer Dominguez,

A mis amigos de la Universidad: Fabiana, Krisvely, Alejandro,

A mis tíos Miguel y Mary de Osuna,

A mis pastores Josué y Emily Prada,

A mi tutora industrial, la Ing. Ana Cecilia García,

A mis compañeros de pasantía en IBM: Juan Carlos, Estefanía, Elizabeth y Alejandro, por todo lo que hemos podido compartir durante estos meses, por brindarme su valiosa amistad, por ayudarme y brindarme su apoyo en todo momento. Juan y Elizabeth, gracias por el apoyo que me brindaron los primeros meses, ya que fueron para adaptarme a esta nueva etapa. Alejandro, por hacerme reír en las mañanas con tus ocurrencias. Estefania, gracias por convertirme en una gran amiga y un gran apoyo.

A una persona muy especial e importante para mi vida. Alguien con quien he podido compartir las victorias obtenidas a lo largo de este camino, pero también las derrotas. A mi mejor amigo, mi otro compañero de pasantías y hermano menor, Roberto Omaña. Te agradezco por tu valiosa amistad, por apoyarme en todo momento, por aconsejarme cuando lo he necesitado. Quiero que sepas, que aunque nuestras vidas tomen rumbos distintos, siempre contarás conmigo, y en mí tendrás a un hermano.

# Índice general

Índice general	VII
Índice de cuadros	XI
Índice de figuras	XII
Introducción	1
<b>1. Entorno Empresarial</b>	<b>3</b>
1.1. Descripción General . . . . .	3
1.2. Misión . . . . .	3
1.3. Visión . . . . .	3
1.4. Estructura Organizacional . . . . .	3
1.5. Ubicación del Pasante . . . . .	3
<b>2. Marco teórico</b>	<b>4</b>
2.1. Sistemas de Información ERP . . . . .	4
2.1.1. Definiciones . . . . .	4
2.2. SAP . . . . .	5
2.2.1. Definición . . . . .	5
2.2.2. Características de SAP R/3 . . . . .	6
2.2.2.1. Características Generales . . . . .	6
2.2.2.2. Características de Negocio . . . . .	6
2.2.2.3. Características de Flexibilidad . . . . .	6
2.2.2.4. Características Técnicas . . . . .	7
2.2.2.5. Otras Características . . . . .	7
2.2.3. Adaptación del Software a las Empresas . . . . .	7

2.2.4.	Módulos de SAP R/3 . . . . .	8
2.3.	Módulo de Ventas y Distribución (SD) . . . . .	9
2.3.1.	Estructura de una Empresa en SAP . . . . .	9
2.3.1.1.	Organización de Venta . . . . .	10
2.3.1.2.	Canal de Distribución . . . . .	10
2.3.1.3.	Sector . . . . .	10
2.3.1.4.	Área de Ventas . . . . .	11
2.3.1.5.	Oficina de Ventas . . . . .	11
2.3.1.6.	Puestos de Expedición . . . . .	11
2.3.1.7.	Puestos de Carga . . . . .	11
2.3.2.	Herramientas Principales . . . . .	12
2.3.2.1.	Manejo de Precios e Impuestos . . . . .	12
2.3.2.2.	Chequeo de Disponibilidad . . . . .	12
2.3.2.3.	Manejo de Crédito . . . . .	12
2.3.2.4.	Facturación . . . . .	12
2.3.2.5.	Determinación del Material . . . . .	12
2.3.2.6.	Determinación de Cuentas . . . . .	12
2.3.2.7.	Procesamiento de Textos . . . . .	13
2.3.3.	Clasificación de los Datos en el Módulo SD . . . . .	13
2.3.3.1.	Datos Maestros . . . . .	13
2.3.4.	Proceso de Ventas utilizando el Módulo SD . . . . .	13
2.3.5.	Relación existente entre el Módulo SD y otros Módulos . . . . .	13
2.4.	Lenguaje de Programación ABAP/4 . . . . .	14
2.4.1.	Estructura de un programa ABAP/4 . . . . .	15
2.4.1.1.	Includes en ABAP/4 . . . . .	17
2.4.2.	Herramientas provistas por ABAP/4 . . . . .	17
2.4.2.1.	Smartforms . . . . .	18
2.4.2.2.	Constructor de Formularios Inteligentes (Smart Form Builder)	18
2.4.2.3.	Template de impresión para el Smartform (Smart Form Print Form Template . . . . .	21
2.4.2.4.	Módulo de Funciones del Smart Form (Smart Form Function Module) . . . . .	21
2.4.2.5.	Programa de Impresión de Smart Forms (Smart Form Print Program) . . . . .	21

2.4.3. Sesión de Batch Input (Carga Masiva de Datos) . . . . .	21
<b>3. Marco Metodológico</b>	<b>23</b>
3.1. Descripción de la metodología . . . . .	23
3.2. Características Principales . . . . .	24
3.3. Fases de la metodología ASAP . . . . .	24
3.3.1. Fase 1: Preparación del Proyecto . . . . .	24
3.3.2. Fase 2: Business Blueprint . . . . .	26
3.3.3. Fase 3: Realización . . . . .	28
3.3.3.1. Simulación . . . . .	28
3.3.3.2. Validación . . . . .	29
3.3.3.3. Unión y Pruebas de Integración . . . . .	29
3.3.3.4. Conversión y Carga de los Datos . . . . .	31
3.3.3.5. Interfaces, Ampliaciones y Reportes . . . . .	32
3.3.4. Fase 4: Preparación Final . . . . .	33
3.3.4.1. Refinar el Sistema creado . . . . .	33
3.3.4.2. Planeacion de la preparación de la Salida en Vivo . . . . .	33
3.3.4.3. Entrenamiento del Usuario Final . . . . .	34
3.3.4.4. Transferencia de Conocimiento . . . . .	34
3.3.4.5. Administración del Sistema . . . . .	34
3.3.4.6. Migración de Datos . . . . .	35
3.3.4.7. Pruebas Finales y Entonación . . . . .	35
3.4. Aplicación para el Proyecto de Pasantía . . . . .	35
<b>4. Desarrollo</b>	<b>36</b>
4.1. Fase 1: Preparación Previa . . . . .	36
4.2. Fase 2: Business Blueprint . . . . .	37
4.2.1. Identificación del proceso de Master Data (Datos Maestros) . . . . .	38
4.2.2. Identificación del proceso de Sales (Pedidos de Ventas) . . . . .	41
4.2.3. Identificación del proceso de Shipping and Transportation (Embarque y Transporte) . . . . .	42
4.2.4. Identificación del proceso de Billing (Facturación) . . . . .	42
4.3. Fase 3: Realización (Construcción y Pruebas) . . . . .	43



4.3.1.	Configuración Base del Módulo de Ventas y Distribución para SSA Beverage . . . . .	43
4.3.1.1.	Configuración de la parte organizativa de la Empresa . . . .	43
4.3.2.	Elaboración de las Facturas Legales . . . . .	45
4.3.3.	Elaboración del programa de Carga Masiva de Datos para el Maestro de Clientes . . . . .	46
4.4.	Fase 4: Presentación y Resultados Finales . . . . .	46
<b>5.</b>	<b>Resultados</b>	<b>48</b>
5.1.	Máquina de prueba . . . . .	49
5.2.	Validación de los estimados . . . . .	49
<b>6.</b>	<b>Conclusiones y recomendaciones</b>	<b>54</b>
	<b>Bibliografía</b>	<b>57</b>
<b>A.</b>	<b>Información adicional de d2Hgr</b>	<b>62</b>
A.1.	Requerimientos de <i>software</i> y <i>hardware</i> . . . . .	63
A.2.	Ayuda de la línea de comando . . . . .	66
A.3.	Archivos del programa . . . . .	69
A.4.	Creación de <i>Xtdata</i> . . . . .	70

# Índice de cuadros

4.1. Configuraciones realizadas para establecer la relación entre las distintas unidades componentes de la estructura de SSA Beverage . . . . .	45
5.1. Computador usado para las pruebas . . . . .	49

# Índice de figuras

2.1. Clasificación de los datos en el Módulo SD . . . . .	13
2.2. Pantalla Inicial de la transacción SMARTFORMS . . . . .	19
2.3. Smart Form Builder . . . . .	20
2.4. Proceso de Batch Input ejecutado en SAP . . . . .	22
4.1. Estructura de la Empresa definida para el Módulo SD (Parte 1) . . . . .	38
4.2. Estructura de la Empresa definida para el Módulo SD (Parte 2) . . . . .	38
4.3. Organizaciones de Ventas definidas para el Módulo SD . . . . .	43
4.4. Canales de Distribución definidos para el Módulo SD . . . . .	43
4.5. Sectores definidos para el Módulo SD . . . . .	44
4.6. Áreas de Ventas definidas para el Módulo SD . . . . .	44
4.7. Puestos de Expedición definidos para el Módulo SD . . . . .	44
4.8. Parte de los Puestos de Carga definidos para el Módulo SD . . . . .	45
4.9. Ejemplo de Factura creada para el módulo SD . . . . .	47
5.1. Promedio y varianza del error absoluto de la estimación del parámetro de Hurst para el método estadístico R/S . . . . .	50
5.2. Promedio y varianza del error absoluto de la estimación del parámetro de Hurst para el método de gráficas de varianza-tiempo . . . . .	51
5.3. Promedio y varianza del error absoluto de la estimación del parámetro de Hurst para el método de varianza modificada de Allan . . . . .	51
5.4. Tiempo promedio utilizado para estimar una ventana de 1024 a 32768 datos para los métodos implementados . . . . .	52
A.1. Asignación de las Organizaciones de Ventas a la Sociedad . . . . .	62
A.2. Asignación de las Organizaciones de Ventas a los Canales de Distribución . .	62
A.3. Asignación de las Organizaciones de Ventas a los Sectores . . . . .	62

A.4. Parte de la asignación de las Oficinas de Ventas a las Áreas de Ventas . . . .	63
A.5. Parte de la asignación de las Organizaciones de Ventas y Canales de Distribución a los Centros de Distribución . . . . .	64
A.6. Parte de la asignación de los Puestos de Expedición a los Centros de Distribución	65

## Introducción

AQUÍ VA EL CONTENIDO DE ANTECEDENTES.

AQUÍ VA EL CONTENIDO DE LA JUSTIFICACIÓN.

En la Universidad de Tohoku de Sendai, Japón, en el laboratorio Kinoshita de la facultad de Ciencias de la Computación, se tiene un *framework* con el cual han creado un *Active Information Resource (AIR) - Network Management System (NMS)* [KIAK04] para gestionar las redes de computadoras. Mediante el lenguaje de programación orientado a agentes basado en Java, DASH (*Repository-based Agent Framework*) y su ambiente de dise no interactivo IDEA (*Interactive Design Environment*), la idea final es que la red y los elementos que la componen se pueda auto-gestionar mediante los agentes instalados en los elementos [USKS02], sin la necesidad de tener un administrador. El objetivo es identificar y solucionar los problemas de la red en tiempo real [AAK04] [KAIK07].

AQUÍ VA EL CONTENIDO DEL PLANTEAMIENTO DEL PROBLEMA.

En los últimos a nos ha habido un gran interés por determinar si los nuevos modelos basados en la autosimilaridad y la dependencia de largo alcance pueden ser utilizados para la detección de anomalías en las redes de datos [DDHT08]. Entre las anomalías más destacadas se encuentran las de ataques de denegación de servicio y su ampliación: el llamado ataque distribuido de denegación de servicio [Li06] [XLLH04].

En el trabajo de [Li06] se utilizó, con buenos resultados, el cálculo del grado de autosimilaridad para detectar anomalías en la red. Se realizó una implementación para un *framework* de autogestión de redes desarrollado en el laboratorio Kinoshita de la Universidad de Tohoku. Como se explica en la próxima sección la idea del proyecto de grado fue mejorar esta

implementación.

El objetivo principal del proyecto es la de la implementación de una solución *SAP SD* en una empresa de bebidas de consumo masivo *SSA CPG* para gestionar los procedimientos comerciales.

Los objetivos específicos planteados en este proyecto son los que se listan a continuación:

1. Adquirir los conocimientos de la Metodología Ascendant SAP y el funcionamiento del Sistema SAP en la funcionalidad Ventas y Distribución (SD).
2. Analizar los procesos de Ventas y Distribución con las funcionalidades SAP que los soportan. Adquirir los conocimientos generales del negocio, procesos y visión global de la solución SAP a implantar.
3. Configurar los campos del sistema a implantar para el módulo SD acorde a las necesidades del cliente, hacer los desarrollos necesarios y probar las funcionalidades SAP ERP SD que soportan los procesos comerciales.
4. Preparar los datos y el sistema, realizar la demostración y documentar los resultados correspondientes al módulo SD, para comprobar el funcionamiento y configuración del módulo de manera individual y en relación con otros módulos implicados del sistema.

# Capítulo 1

## Entorno Empresarial

Los conceptos más importantes sobre las series de tiempo, el parámetro de Hurst y su uso para la detección de ataques de denegación de servicio se presentan en este capítulo.

### 1.1. Descripción General

### 1.2. Misión

### 1.3. Visión

### 1.4. Estructura Organizacional

### 1.5. Ubicación del Pasante

# Capítulo 2

## Marco teórico

Los conceptos claves acerca de lo que es un sistema ERP, SAP, ABAP, Ascendant SAP son los que se presentan en este capítulo.

La sección 2.1 presenta las nociones básicas acerca de en qué consiste un Sistema de Información ERP. La sección 2.2 describe un poco lo que es SAP: sus características principales, una breve historia. Por otro lado, la sección ?? explica en que consiste el lenguaje de programación ABAP, sus principales características. La última sección define la metodología **Ascendant SAP (ASAP)**, a ser utilizada a lo largo de este proyecto.

### 2.1. Sistemas de Información ERP

Para entender en qué consisten los Sistemas de Información ERP, para ello habrá que definir dos conceptos fundamentales: Sistema de Información, y más aún, Sistemas ERP.

#### 2.1.1. Definiciones

**Definición 1.**



**Definición 2.** Un ERP (Sistema de Planificación de Recursos Empresariales, ó como se le conoce en Inglés: **Enterprise Resource Planning**), es un sistema que tiene la capacidad de la automatización e integración de todos los módulos de un área de negocio. En otras palabras, es capaz de manejar todas las áreas relacionadas con una empresa de forma automatizada e integrada.

Una característica fundamental de este tipo de sistemas es que está basado en módulos. Como consecuencia de ello, el mismo está compuesto por un conjunto de softwares o módulos.

Para poder tener acceso a los datos que están relacionados con cada módulo es necesario tener una Base de Datos centralizada, la cual almacena dicha información.

## 2.2. SAP

En esta sección se ofrece un panorama general acerca de SAP, qué es, cuáles son sus características principales y cuáles son los módulos que lo integra.

### 2.2.1. Definición

**SAP AG** es una empresa de la rama de la Computación que fue fundada por 4 ingenieros pertenecientes a IBM (**International Business Machines**), en la ciudad de Walldorf, Alemania, en el año de 1972. Por su origen alemán, las siglas SAP son un acrónimo de: *Sys-teme, Anwendungen Produkte in der Datenverarbeitung*, que traducido al castellano significa: **Sistemas, Aplicaciones y Productos en Procesamiento de Datos**.

El Software principal desarrollado por esta empresa es **SAP R/3** y el cual se encuentra disponible en 28 idiomas. Éste software es personalizable, utiliza la arquitectura cliente-servidor, es decir, que el cliente envía solicitudes al servidor y éste a su vez envía una respuesta al cliente. Éste software fue hecho en el lenguaje de programación **ABAP/4**.

### 2.2.2. Características de SAP R/3

De acuerdo al autor de [?], las características del software **SAP R/3** se pueden dividir en diferentes categorías, las cuales se mencionan a continuación:

#### 2.2.2.1. Características Generales

- Es un software altamente integrado y multifuncional, lo que trae como consecuencia que exista una estrecha relación entre las funciones del mismo.
- Es una aplicación que trabaja en tiempo real. En otras palabras, las actualizaciones de los datos son efectuadas a través de una conexión, y en ese mismo instante.

#### 2.2.2.2. Características de Negocio

- Este software contiene todas las funcionalidades necesarias para poder llevar a cabo el manejo de un negocio entero. Éste incorpora una aplicación llamada *Best Industry Practices*, que traducido al español quiere decir: *Mejores Practicas de la Industria*, y éste último es adecuado para una amplia gama de industrias y organizaciones.
- Este programa es capaz de soportar todos los procesos de negocio de la empresa.

#### 2.2.2.3. Características de Flexibilidad

- Este software es altamente configurable. En otras palabras, se puede adaptar a las necesidades de la empresa que lo utilice y a sus requerimientos. Para ello, se pueden realizar cambios que, dependiendo del número de factores que participen, éstos tendrán su grado de complejidad.
- Es capaz de dar apoyo a empresas que poseen subsidiarias en distintas partes del mundo.

- Este software es muy utilizado a nivel mundial, dado que está disponible en 28 idiomas, y a que posee la capacidad de adaptarse a la moneda, leyes y regulaciones, impuestos para un cierto país, etc.

#### **2.2.2.4. Características Técnicas**

- Este software tiene la capacidad de ser portable, dado que es multiplataforma, es decir, soporta cualquier sistema operativo, manejador de Base de Datos, etc.
- Posee un número mínimo de redundancia, lo que favorece a la consistencia de los datos almacenados. Adicionalmente, posee un manejador de alta seguridad de los datos, y puede manejar etructuras de datos complejas.

#### **2.2.2.5. Otras Características**

- Tiene la capacidad de manejar la misma información en cada módulo.
- Posee una única manera de ingreso al sistema. Éste es a través del *SAP GUI*.
- Tiene la capacidad de ser escalable, es decir, que está preparado para manejar el continuo crecimiento del trabajo sin disminuir la calidad.
- Tiene una interfaz gráfica amigable.

### **2.2.3. Adaptación del Software a las Empresas**

Para poder realizar la adaptación de SAP a las necesidades de una empresa en particular, están un conjunto de herramientas y utilidades destinado a ello. Para esto, se requiere de un conjunto de consultores, un equipo de proyecto y personal de Tecnología de la Información (IT), quienes serán los encargados de efectuar dicha adaptación.

Este proceso se puede realizar a través de dos métodos, los cuales se listan a continuación:

- **Cambios en la Configuración:** Aquí son modificadas las tablas relacionadas con los distintos módulos para poder realizar la adaptación.
- **Programación en el Lenguaje ABAP/4:** Esto implica modificar programas ya existentes en **SAP R/3** o crear programas nuevos.

#### 2.2.4. Módulos de SAP R/3

Debido a que SAP es un ERP, luego, SAP está dividido en diferentes módulos, con el fin de poder abarcar cada área de una empresa. Estos módulos son los que se listan a continuación:

1. Asset Management (Manejo de Aplicaciones - AM)
2. Financials (Finanzas - FI)
3. Controlling (Control - CO)
4. Human Resources (Recursos Humanos - HR)
5. Plant Maintenance (Mantenimiento de Planta - PM)
6. Production Planning (Planificación de la Producción - PP)
7. Project System (Sistema de Proyectos - PS)
8. Quality Management (Manejo de Calidad -QM)
9. Sales and Distribution (Ventas y Distribución - SD)
10. Materials Management (Manejo de Materiales - MM)
11. Services Management (Manejo de Servicios - SM)
12. Industry Specific Solutions (Soluciones Específicas para la Industria - IS)

13. Business Workflow (Flujo de trabajo del Negocio - WF)
14. Basis (Incluye el lenguaje de programación **ABAP 4** - BC).

## 2.3. Módulo de Ventas y Distribución (SD)

El **Módulo de Ventas y Distribución** (Sales and Distribution como es conocido en inglés) es un sub-sistema perteneciente a SAP, el cual se encarga de prestar apoyo a las distintas empresas en el área de las Ventas y distribución de productos y/o servicios.

Este módulo ayuda a las compañías a establecer un precio para sus productos, chequear órdenes de ventas que se mantienen abiertas, a tomar previsiones para necesidades futuras, etc. Adicionalmente, ayuda a dar mayor control a las actividades relacionadas con el área de ventas: desde el momento en que ocurre un pedido de algún(algunos) producto(s) y/o servicio(s) hasta su posterior entrega.

### 2.3.1. Estructura de una Empresa en SAP

Una empresa está conformada por un conjunto de unidades más pequeñas que facilitan el trabajo en el proceso de ventas y sus actividades asociadas.

Las unidades más importantes que componen a una empresa correspondientes al área de Ventas y Distribución son las que se listan a continuación:

- Organización(es) de Ventas
- Canal(es) de Distribución
- Sector(es)
- Áreas de Ventas

- Oficinas de Venta
- Puestos de Expedición
- Puestos de Carga

En los próximos puntos se ahondará mas en cada una de estas unidades, para así conocer en detalle en qué consiste cada una.

#### **2.3.1.1. Organización de Venta**

Una **Organización de Venta** dentro del Sistema SAP es una unidad que es muy utilizada para poder dar a conocer y distribuir los bienes y servicios que la empresa ofrece. A través de esta unidad, se pueden realizar las distintas negociaciones entre el cliente y la empresa, así como los términos y condiciones a los que llegan ambos, como por ejemplo el establecimiento de algún tipo de descuento, la forma de pago (si es de contado, en 15 días, etc).

#### **2.3.1.2. Canal de Distribución**

Un **Canal de Distribución** dentro de SAP es una unidad que es capaz de distribuir los bienes y servicios que son ofrecidos por la Organización de Ventas a los clientes que atiende. Para poder llevar esta tarea a cabo, es posible que sea necesario usar más de un canal de distribución. Es por ello que una Organización de Ventas tiene a su cargo al menos un Centro de Distribución, aunque pudieran ser varios, dependiendo ya del proceso que lleve la empresa en particular.

#### **2.3.1.3. Sector**

Un **Sector (o División por su nombre en inglés)** dentro de SAP es una unidad organizativa que permite agrupar cada uno de los productos o materiales que son ofrecidos por las Organizaciones de Ventas en diferentes categorías.

#### **2.3.1.4. Área de Ventas**

Un **Área de Ventas** es una relación existente entre una Organización de Ventas, un Canal de Distribución y un Sector. Todas las transacciones relacionadas con Ventas y Distribución son manejadas a través de las Áreas de Ventas.

#### **2.3.1.5. Oficina de Ventas**

Una **Oficina de Ventas** es el lugar físico donde la empresa realiza la venta de bienes y servicios. Es importante resaltar que estas unidades pueden estar ubicadas en distintos puntos geográficos. Un Área de Ventas tiene asociada a almenos una Oficina de Ventas, aunque puede tener varias bajo su cargo.

#### **2.3.1.6. Puestos de Expedición**

Un **Puesto de Expedición** es el lugar físico que posee la empresa para efectuar las entregas de los bienes producidos y adquiridos. Un puesto de expedición puede estar asociado a un Centro, pero no es limitativo. Es decir, que puede estar asociado a varios Centros. Algo importante que resaltar es que para que una entrega se pueda efectuar dentro de SAP, es necesario indicar en cuál Puesto de Expedición se desea hacer, ya que de lo contrario, no es posible.

#### **2.3.1.7. Puestos de Carga**

Un **Puesto de Carga** es una sub-división de los Puestos de Expedición.

## **2.3.2. Herramientas Principales**

### **2.3.2.1. Manejo de Precios e Impuestos**

A través de esta herramienta, el módulo puede evaluar los precios que son colocados a los productos o servicios de acuerdo a unas condiciones establecidas previamente.

### **2.3.2.2. Chequeo de Disponibilidad**

Con esta herramienta, el módulo puede evaluar la disponibilidad de un producto para un almacén especificado.

### **2.3.2.3. Manejo de Crédito**

Con esta herramienta es posible establecer límites de crédito para un cliente durante el proceso de ventas en el cual se encuentra envuelto.

### **2.3.2.4. Facturación**

Una vez que una Orden de Ventas es creada, se utiliza esta herramienta para crear la(s) factura(s) asociadas.

### **2.3.2.5. Determinación del Material**

Con esta herramienta es posible determinar un material específico de acuerdo a unas condiciones especificadas.

### **2.3.2.6. Determinación de Cuentas**

Ayuda a obtener ciertos detalles de los clientes basándose en unas condiciones específicas.



### 2.3.2.7. Procesamiento de Textos

Con esta herramienta se hace posible el manejo de textos entre los distintos documentos que se obtienen del proceso de ventas.

### 2.3.3. Clasificación de los Datos en el Módulo SD

De acuerdo al autor de [?], la data almacenada dentro del Módulo se puede clasificar como se muestra en la siguiente figura: Ahora se procederá a explicar la clasificación antes

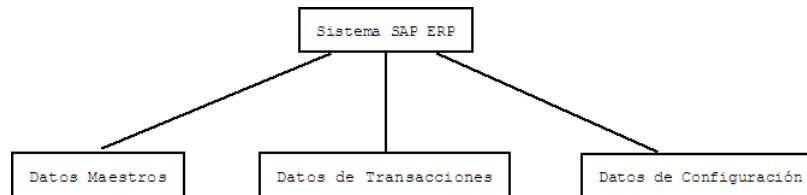


Figura 2.1: Clasificación de los datos en el Módulo SD

mencionada.

#### 2.3.3.1. Datos Maestros

Los Datos Maestros dentro del Módulo SD están compuestos por:

- Datos de Compañías
- Datos Maestros de Clientes

Cada una de estas entidades contienen a su vez atributos, jerarquías y tablas.

### 2.3.4. Proceso de Ventas utilizando el Módulo SD

### 2.3.5. Relación existente entre el Módulo SD y otros Módulos

Este módulo está fuertemente integrado con otros módulos de SAP, como por ejemplo: MM, WM, QM. En el momento en que un cliente realiza un pedido de algún producto,

posteriormente se chequea la disponibilidad del mismo en algún almacén; esto es posible gracias al módulo MM. Por otro lado, el módulo QM es el encargado de manejar la calidad y brindar soporte a un servicio prestado al cliente, ambos representados por un documento de ventas en SD.

## 2.4. Lenguaje de Programación ABAP/4

ABAP (Advanced Business Application Programming - Programacion de Aplicaciones de Negocio Avanzado) es un lenguaje de programación que fue diseñado en la década de los 80. Su uso principal es la de generar reportes con los cuales se les permite a las empresas construir sus propias aplicaciones para el manejo de las distintas áreas que lo componen (manejo de materiales, manejo del área financiera, manejo de las ventas, etc). Este es uno de los primeros lenguajes de programación que incluye dentro de su definición el concepto de Bases de Datos Lógicas. Dentro de las características que posee el lenguaje, se pueden mencionar las siguientes:

- Es un lenguaje basado en la programación estructurada. En otras palabras, contiene estructuras de control.
- Es un lenguaje interpretado, aunque existen versiones compiladas del mismo.
- Es muy utilizado para obtener dos tipos de programas: los que son usados para obtener por ejemplo un listado (modo reporte), y aquellos que son usados como transacciones (modo diálogo).
- Es un lenguaje orientado a eventos, es decir, que puede ser controlado desde el exterior a través de sentencias de eventos.
- Está integrado totalmente con el sistema **SAP R/3**.

- La salida de sus programas es multilingual.

### 2.4.1. Estructura de un programa ABAP/4

Un programa ABAP tiene la responsabilidad del procesamiento de datos dentro de una aplicación. Esto quiere decir que el programa debe ser dividido en distintas secciones que pueden ser asignadas a cada paso secuencial a ejecutar dentro del programa. Para poder lograr esto, es necesario modularizar los programas hechos en ABAP. Cada módulo o **Bloque de Procesamiento** como también se le conoce, consiste en un conjunto de instrucciones ABAP. Al momento de ejecutar un programa, es posible realizar la llamada de cada bloque. Es importante resaltar que no permiten anidamiento. Cada programa ABAP está compuesto por dos partes principales, las cuales se explican en los próximos bloques.

## DECLARACIÓN DE DATOS GLOBALES, CLASES Y SELECTION SCREENS (PANTALLAS DE SELECCION)

En esta parte es donde se llevan a cabo las declaraciones de los datos globales, de las pantallas de selección y de las clases. A continuación, se procederá a explicar cada uno de estos puntos.

### Datos Globales

Los Datos Globales es el conjunto de datos que son declarados con el fin de poder acceder a ellos y poder utilizarlos a lo largo del programa. Estos datos globales constan de:

- Variables
- Constantes
- Estructuras de Datos

- Tablas Internas

## **Clases**

Una clase en SAP es conocida como una plantilla para un objeto. También se puede definir como una descripción abstracta de un objeto.

## **Selection Screens (Pantallas de Selección)**

Las Pantallas de Selección son unas pantallas especiales que son definidas en ABAP/4 para permitir que los programas reciban datos provenientes de los usuarios a través de la entrada estándar (Teclado). El ambiente de ejecución de ABAP controla el flujo del procesamiento de las pantallas de selección. De esta manera, este ambiente genera un número de eventos antes de que las pantallas de selección sean mostradas, y otros luego después que las mismas hayan sido ejecutadas.

## **CONTENEDOR DE BLOQUES DE PROCESAMIENTO**

Esta segunda parte está compuesta por los Bloques de Procesamiento, entre los cuales se pueden encontrar los que se listan a continuación:

- Módulos de Diálogo
- Bloques de Eventos
- Procedimientos (métodos, subrutinas y módulo de funciones con sus datos locales)

## **BLOQUE DE PROGRAMA**

Consiste en todos los estados ABAP (salvo estados declarativos, los cuales van en la primera parte).

## BLOQUE DE PROCESO DE LLAMADAS

En esta tercera parte es posible hacer llamadas a otros bloques externos o internos al programa ABAP. Los módulos de Diálogo y los bloques de eventos son llamados desde afuera. Los procedimientos son llamados usando instrucciones ABAP.

### 2.4.1.1. Includes en ABAP/4

Un Include es un programa especial el cual se hace en SAP con el objetivo de lograr mayor modularización del programa principal. Este tiene dos funciones principales:

- Librería: Los programas Include permiten el uso del mismo código fuente en diferentes programas. Por ejemplo, en los Includes se pueden realizar las declaraciones globales, la definición de clases y procedimientos, etc.
- Orden: Los programas Include permiten manejar un cierto orden dentro de programas complejos. Por ejemplo, los grupos de funciones usan los programas Include para almacenar partes del programa.

### 2.4.2. Herramientas provistas por ABAP/4

Dentro de las herramientas más utilizadas de **ABAP/4**, están las que se listan a continuación:

- Smartforms (Formularios)
- Sesiones de Batch Input (Carga masiva de datos)

En la siguientes subsecciones se explicará más a detalle acerca de dichas herramientas.

#### **2.4.2.1. Smartforms**

Los **Smartforms** fueron introducidos dentro del lenguaje a través de la versión 4.6 de **SAP R/3**. Su función principal es la impresión y envío de documentos a través del correo electrónico, o por fax. Con esta herramienta es posible hacer el diseño de formularios, archivos PDF y documentos en general. Esta herramienta provee una interfaz capaz de construir y mantener la disposición y lógica del formulario. Una ventaja que tiene esta herramienta es el uso de una interfaz gráfica para modificar formularios, ya que con esto, se evita el uso de la programación. En un smartform, los datos son transmitidos a través de táblas dinámicas o estáticas. Adicionalmente permite incluir gráficos que pueden ser visibles en el formulario. En los próximos segmentos se explicará los componentes de esta herramienta.

#### **2.4.2.2. Constructor de Formularios Inteligentes (Smart Form Builder)**

Esta componente es la interfaz principal que es usada para la construcción de Smartforms a través de la pantalla inicial que se muestra en la Figura 2.2

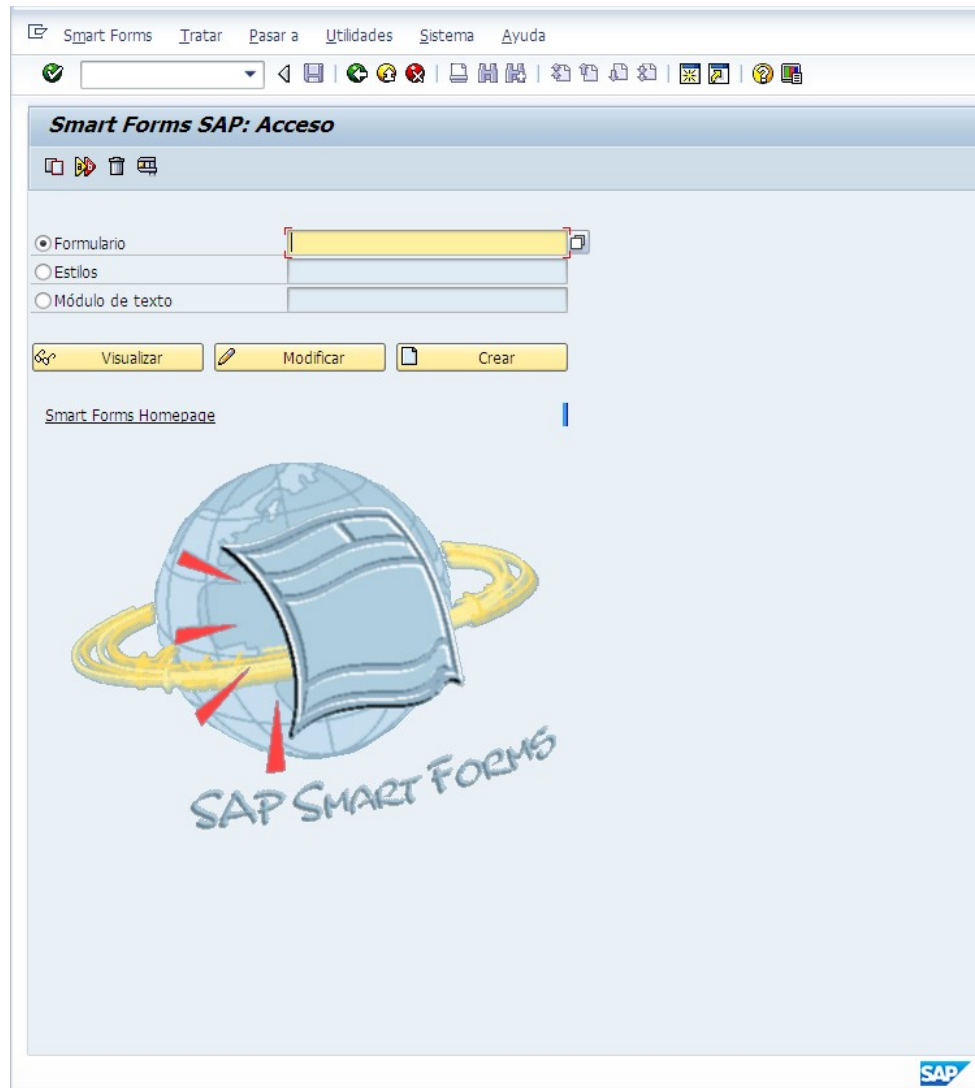


Figura 2.2: Pantalla Inicial de la transacción SMARTFORMS

Como se puede ver en la imagen anterior, con esta transacción es posible crear, modificar o visualizar un smartform. Al darle click a cualquier opción, la misma lleva a otra pantalla, la cual es la que permite la construcción de un smartform. Ésta es conocida como **Smart Form Builder**. En la Figura 2.3 se muestra a detalle dicha pantalla.

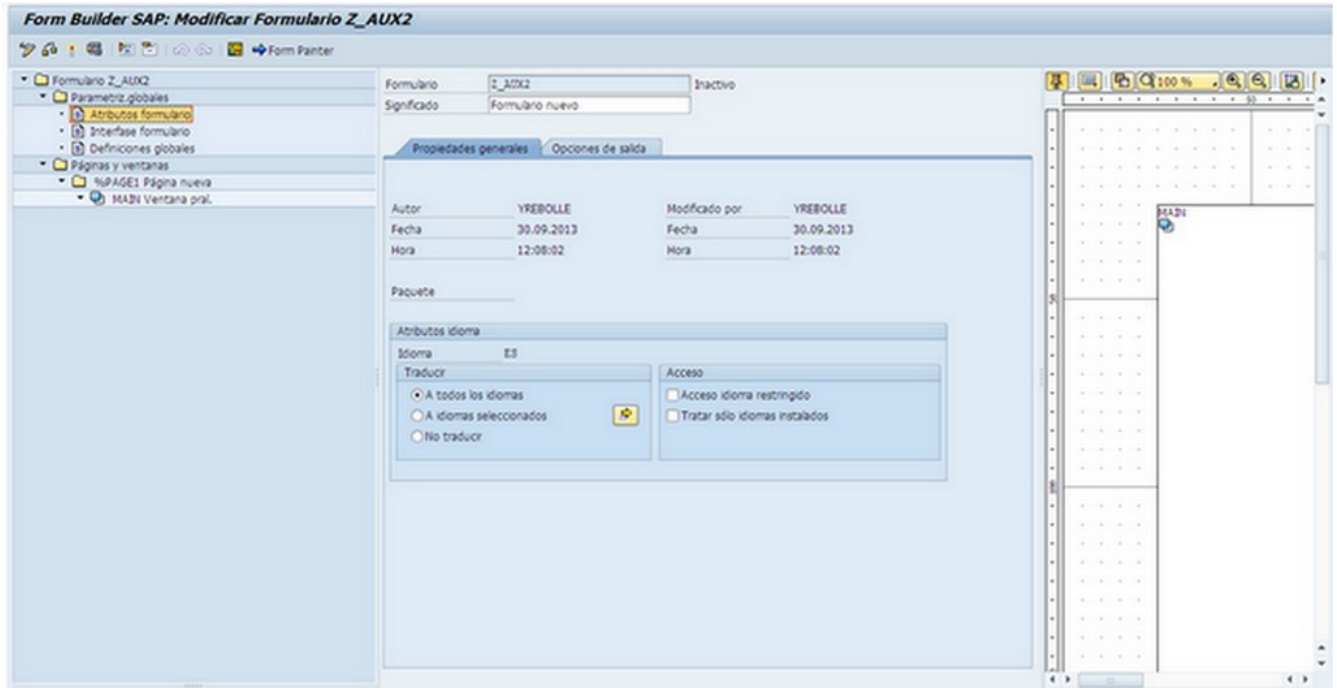


Figura 2.3: Smart Form Builder

En la figura se pueden ver varias partes bien diferenciadas, las cuales se explican a continuación:

- **Menú de Navegación:** Es el que está representado por la columna izquierda. Éste expone al formulario en forma de jerarquía, donde cada elemento es un nodo
- **Marco de Mantenimiento:** Está conformado por los atributos principales del formulario (Nombre y significado)
- **Lista de Campos:** Esta lista muestra los datos que están definidos actualmente en el formulario
- **Form Painter:** Ésta es una herramienta gráfica la cual permite diseñar cada una de las partes que tendrá el formulario.



#### **2.4.2.3. Template de impresión para el Smartform (Smart Form Print Form Template)**

Un template de impresión para el Smartform ofrece un diseño ya creado por el sistema para así crear un Smartform que pueda ser imprimible.

#### **2.4.2.4. Módulo de Funciones del Smart Form (Smart Form Function Module)**

El Módulo de Funciones del Smart Form es una secuencia de instrucciones generadas de forma automática al momento de la activación del mismo.

#### **2.4.2.5. Programa de Impresión de Smart Forms (Smart Form Print Program)**

El Programa de Impresión de Smartforms es el programa control encargado de manejar la información que es enviada al smartform y de su posterior impresión.

### **2.4.3. Sesión de Batch Input (Carga Masiva de Datos)**

Esta es una herramienta que provee ABAP/4 con el fin de introducir datos de manera no interactiva dentro del Sistema SAP. La Carga Masiva o por Lotes (Batch Input) es usado muy frecuentemente para transferir datos desde sistemas externos a un sistema SAP, o entre sistemas SAP. Una sesión de Batch Input, más específicamente, es un conjunto de una o varias llamadas a transacciones con los datos a ser procesados por dichas transacciones. Para poder lograr esta tarea, existen dos transacciones en SAP: la SHDB y la SM35. A partir de la ejecución de alguna de las transacciones antes mencionadas, el sistema se encarga de realizar una grabación sobre la(s) transacción(es) involucrada(s) junto con los datos en un formato especial el cual puede ser interpretado por SAP. Cuando el sistema ejecuta una sesión, utiliza la data almacenada en dicha sesión para comenzar la simulación de la entrada on-line de los datos. El sistema llama a las transacciones y carga los datos en ella. En la

figura 2.4 se muestra un esquema general de este proceso.

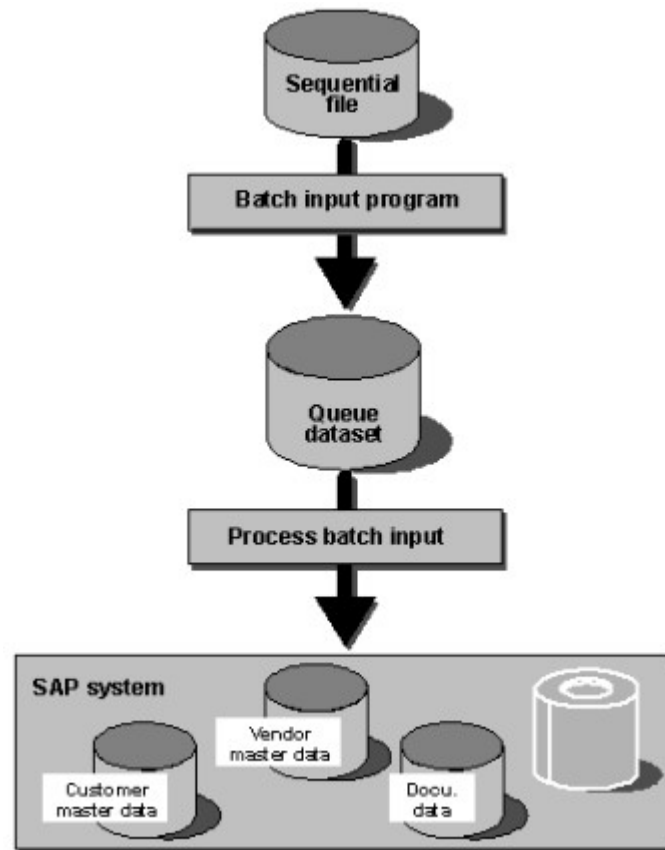


Figura 2.4: Proceso de Batch Input ejecutado en SAP

En la imagen anterior se puede apreciar los pasos a seguir para poder llevar a cabo una sesión de Batch Input. Para ello, el sistema a través de una interfaz recibe el archivo que contiene los datos a ser cargados, y éste es pasado al programa de Carga Masiva (Batch Input), el cual se encarga de procesar dichos datos y colocarlos en una cola, para luego procesar uno por uno a través de la simulación de la ejecución de la transacción involucrada.

# Capítulo 3

## Marco Metodológico

Para el desarrollo de un buen software es necesario la utilización de una metodología, ya que la misma brinda una serie de mecanismos y fases para el desarrollo organizado de la aplicación en cuestión. Para este proyecto, se decidió utilizar la metodología **ASAP (Ascendant SAP)**, ya que es la más utilizada para el desarrollo de aplicaciones en SAP ERP.

### 3.1. Descripción de la metodología

**ASAP (Ascendant SAP ó Accelerated SAP como también se le conoce)** es una metodología que fue diseñada por SAP para agilizar el desarrollo de sus aplicaciones, ya que con la metodología actual, un desarrollo se podía llevar mucho más tiempo del esperado. Mientras un proyecto usando la metodología convencional se puede llevar dos años o más en realizarse, el mismo proyecto basándose en la metodología **ASAP**, puede ser realizado en menos de un año. Es importante resaltar que esta tecnología no es utilizada en todos los desarrollos con SAP, ya que la misma se recomienda para aquellas empresas que no tienen una extensa modificación en sus requerimientos, o para aquellas empresas que requieren del uso de la re-ingeniería.

## 3.2. Características Principales

ASAP ha sido diseñado con el objetivo de estandarizar y de llevar de una forma coordinada una implementación SAP. Esta metodología posee las siguientes características:

1. Es capaz de optimizar tiempo, calidad y recursos.
2. Es capaz de aprovechar las mejores prácticas del negocio.
3. Es capaz de entregar un proceso orientado a un mapa de proyecto (Hoja de Ruta ASAP). Una hoja de ruta ASAP es un gráfico que presenta los pasos o fases a seguir, como se muestra en el siguiente ejemplo:

## 3.3. Fases de la metodología ASAP

Esta metodología está dividida en 4 fases, las cuáles se listan en la siguiente sección.

### 3.3.1. Fase 1: Preparación del Proyecto

En esta fase se evalúan dos factores críticos del proyecto: en primer lugar la preparación de la organización del mismo, ya que en este punto se realizan algunas tareas de gestión que son claves, como por ejemplo: Proveer un compromiso de la alta gerencia y apoyo, establecer metas claras y objetivos claros, acordar en los próximos pasos a dar dentro del proyecto, proveer un proceso eficiente de toma de decisiones, escoger un equipo que sea calificado y que represente las distintas áreas funcionales. En segundo lugar, la planificación del proyecto, ya que en este punto se deben identificar aquellos elementos que sean críticos, dentro de los cuales se puede mencionar los que se listan a continuación:

- Principios Rectores: Estos son principios de alto nivel que pueden ser establecidos al inicio del proyecto. Estos definen y comunican la visión de la empresa, ayuda a mantener

el proyecto enfocado, y en el caso en que exista algún conflicto, sirve de base para su solución.

- Principios estratégicos: Estos son principios de negocio que direccionan las estrategias a utilizar. Al tener unas estrategias bien definidas, la implementación se torna mas fácil, y así se pueden lograr los objetivos establecidos.
- Impulsores del proyecto: Estos son los encargados de escoger el software ERP para una implementación particular. Es importante recordar que SAP tiene distintas soluciones para ciertos tipos de empresas, como por ejemplo IS-OIL para aquellas empresas cuyo producto tenga relación con hidrocarburos, entre otros.
- Presupuestos, estándares y indicadores: Al inicio del proyecto se debe establecer un presupuesto sobre el costo del mismo, además que se deben definir los estándares que va a utilizar y los distintos indicadores.

En tercer lugar, el equipo de implementación, ya que es el responsable de que el proyecto se pueda llevar a cabo. Generalmente este equipo se encuentra integrado por consultores pertenecientes a organizaciones externas y por empleados internos de la compañía. Este equipo está dividido en los siguientes grupos:

- Personal interno de la compañía o Cliente.
- Personal de Implementación e Integración: Éste incluye a empleados de la empresa desarrolladora o de sub-contratistas.

El equipo de clientes está integrado por:

- Miembros del equipo central: Tienen dedicación al 100 del tiempo disponible dedicado al proyecto.

- Equipo de extensión: Son los que tienen dedicación del 20-50 del tiempo al proyecto, dependiendo de la fase en la cual se encuentre.

En este punto, los equipos definidos son organizados por los distintos módulos o funcionalidades. Por ejemplo, se pueden tener equipos para el módulo de Finanzas (FI), Ventas y Distribución (SD), Gestión de Materiales (MM), entre otros. Es fundamental que el equipo escogido tenga las siguientes capacidades:

- Analizar el impacto del nuevo sistema ERP sobre los procesos del negocio contra el proceso actual.
- Analizar los requerimientos funcionales y de implementación.
- Diseñar un sistema integrado.
- Proveer de conocimiento a los empleados durante el proyecto.

### **3.3.2. Fase 2: Business Blueprint**

El objetivo principal de esta fase es comprender el funcionamiento actual de la empresa, para así poder determinar los requerimientos de implementación basados en las necesidades que la organización pueda presentar en un futuro. Para esto, se realiza un análisis exhaustivo del negocio de la compañía, cómo se desenvuelve actualmente, e identificando las funcionalidades soportadas por el sistema actual. Luego se compara las prácticas existentes y sus funcionalidades con las que son soportadas por SAP. Durante esta fase, los ejecutivos y gerentes de la compañía son entrevistados. Dichas entrevistas son realizadas dentro de grupos pequeños, o de manera individual. Luego, basado en las respuestas obtenidas, los consultores pueden entender y definir los siguientes parámetros:

- El negocio de la compañía

- La forma de operar
- Elementos críticos del negocio
- Procesos deseables a llevar a cabo en el negocio
- Requerimientos del negocio y de funcionalidades
- El alcance que tendrá el proyecto
- Los riesgos que posee el desarrollo del proyecto

Al final de esta etapa es elaborado un documento llamado **Documento de Proyecto (Blueprint Document como se le conoce en inglés)**. Este documento puede ser descrito como un modelo visual de la empresa. En este documento se detalla lo siguiente:

- Funcionalidad ya existente
- Funcionalidad a desarrollar
- Procesos actualmente en operacion
- Alcance de la implementación
- Estructura organizacional
- Funcionalidad diferida
- Riesgos potenciales
- 

Una decisión fundamental que se toma en esta fase es la definición de una estructura organizacional SAP basada en los procesos de organización del negocio. Esta estructura determina como los datos son definidos dentro del sistema, la complejidad de los datos de entrada, y el

tamaño de los archivos que contienen datos maestros. La estructura definida debe haberse analizado bien en esta etapa, ya que cualquier cambio que fuera realizado en las fases siguientes, derivaría en un costo elevado. Uno de los elementos más importantes de la estructura organizacional es el código que se le asigna a la compañía, ya que éste es el elemento más alto dentro de esta estructura armada. El código de la compañía una unidad legal e organizacionalmente independiente. Éste representa a una unidad de contabilidad independiente, lo cual hace que posea sus propias componentes financieras. Una de las partes de esta estructura es el **Área de Control**. La misma simboliza a un elemento organizacional con la cual se manejan como estructura organizativa es la estructura del negocio. Esto no es más que la organización de la empresa en sí. En otras palabras, las distintas áreas funcionales, como Logística, Recursos Humanos, etc.

### **3.3.3. Fase 3: Realización**

Durante esta fase, el sistema es configurado basado en los requerimientos obtenidos en la fase anterior, y luego es probado. Esta fase no es rígida, es decir, que la aplicación de la misma es progresiva. En otras palabras, se construye, se prueba, se refinan los detalles y se vuelve a probar. En las próximas sub-secciones se describen un poco las etapas por las cuales se pasa dentro de esta fase.

#### **3.3.3.1. Simulación**

El primer paso que se lleva en esta etapa es la configuración. Aquí, los distintos consultores configuran el sistema SAP de acuerdo al documento del Negocio previamente definido. Esta configuración cubre el 80 de los procesos del negocio de la compañía y de las transacciones diarias. Este proceso implica modificar el software de SAP a través de procedimientos no programables, como por ejemplo: modificaciones de entrada a tabla, herramientas base, etc.



El siguiente paso dentro de esta etapa es la Reproducción. Éste consiste en introducir un conjunto reducido de usuarios en el nuevo sistema para hacer las pruebas a la configuración realizada previamente, y así poder obtener una retroalimentación. Las reproducciones son realizadas de forma periódica basándose en las preferencias de los usuarios.

#### **3.3.3.2. Validación**

Durante esta etapa, el diseño es refinado y finalizado. El equipo de trabajo refina el sistema de tal modo que todos los requerimientos del negocio se encuentren configurados. Un punto importante durante la realización de esta fase es la elaboración de una lista de procesos maestros. El equipo involucrado comienza a desarrollar los procedimientos del procedimiento del Negocio (Guía de Configuración), con la cual se documenta toda la configuración realizada al sistema.

#### **3.3.3.3. Unión y Pruebas de Integración**

Para que un software sea colocado en producción debe haber recibido todas las pruebas necesarias. Un software que no es probado de la manera adecuada, puede traer consecuencias a la hora de ponerse en reproducción, como por ejemplo que se presente alguna falla inesperada, lo que derivaría en descartar el programa. El hecho de realizar una prueba a un sistema ERP integrado representa un gran desafío, pero dentro de los beneficios a obtener están los siguientes:

- Se confirma que el proceso trabaja adecuadamente
- Se obtiene una configuración ágil
- Tiene rendimiento garantizado
- La integración es mejorada

- Sus costos son bajos
- Los riesgos se reducen al mínimo

Las pruebas de todo el sistema implementado se realiza en dos grandes grupos: El primer grupo es el conjunto de pruebas unitarias. Estas consisten en realizar pruebas a pequeñas transacciones, como por ejemplo: crear una orden de venta, crear un cliente, etc. Para esto, se hace por cada área funcional o módulo cumpliendo un ciclo básico, como por ejemplo, en Ventas y Distribución el ciclo básico consiste en: Realizar el pedido de Venta, luego efectuar la entrega para así finalmente generar la factura correspondiente. El segundo grupo de pruebas está conformado por las **Pruebas Integrales**. Éstas consisten en crear un conjunto de escenarios que involucren todas las áreas funcionales implementadas. Estas pruebas son diseñadas con una perspectiva orientada a procesos. Por ejemplo, se prueba cada paso requerido para realizar un Pedido de Venta. En este punto, es fundamental contar con la presencia de los usuarios finales, ya que son ellos los que pueden brindar una retroalimentación oportuna, y así poder detectar posibles fallas. Existen dos maneras de aplicar estas pruebas unitarias. La primera, se conoce como **Aseguramiento de Calidad (QA por su nombre en inglés)**, es realizada por un conjunto de miembros de tiempo completo en compañía de usuarios del negocio pertenecientes a las distintas áreas funcionales. Cada grupo es asignado a un pequeño escenario, y es responsable de verificar que todo el proceso involucrado se lleve a cabo sin errores desde el inicio hasta el final. La segunda opción es tener distintos grupos que realicen las distintas pruebas por cada área funcional. En el caso de que la prueba involucre varias áreas funcionales, por ejemplo, se tienen las áreas funcionales A, B y C. En el momento en que el área A necesite enviar recursos al área B, la responsabilidad de las pruebas pasa del miembro del equipo A que está ejecutando la prueba, a algún miembro del equipo B.

#### **3.3.3.4. Conversión y Carga de los Datos**

Para que el sistema implementado en SAP pueda funcionar correctamente, es necesario realizar la carga de una gran cantidad de datos. Existen dos métodos para la migración de los datos dentro del sistema SAP, estos son los que se listan a continuación:

- Entrada por Lote: Consiste en simular la entrada de datos a través de las pantallas de las distintas transacciones.
- Entrada Directa: Se recibe el archivo de forma directa, se procesa, se realizan los chequeos previos a los datos, y luego se realiza la actualización en la Base de Datos.

Es importante que la migración de los datos se realice lo más temprano posible. Se recomienda agrupar los datos en pequeños grupos, e ir cargándolos poco a poco. Sin embargo, esto deriva en una limitación de la efectividad a la hora de continuar con las configuraciones, dado que el sistema se encuentra incompleto y puede presentar fallas a la hora de la carga de estos datos. En SAP, los datos se pueden agrupar en 5 grupos principales:

- Datos Maestros Automatizados
- Datos Maestros Manuales
- Datos de Transacción Automatizados
- Datos de Transacción Manuales
- Datos de Limpieza

Existe un problema muy importante que hay que tomar en cuenta, y que por lo general se ignora o no se le da la debida importancia: El problema de los datos sucios o inconsistentes. Esto es fundamental, ya que pueden generar problemas con el sistema implementado en SAP por la calidad de los datos que han sido cargados. Otro problema con el que se debe estar

alerta es con la duplicidad de los datos, ya que por ejemplo, se puede tener un material con múltiples entradas. Esto se debe a que dentro de la entrada de un material, el único campo unívoco correspondiente a dicho material es un código asignado por SAP. Existen dos maneras de chequear que los datos no tengan duplicados: La primera, se realiza en el sistema original de donde provienen los datos. La segunda, se realiza una vez que los datos hayan sido ingresados a SAP, se procede a su chequeo correspondiente. Las personas encargadas del chequeo de los datos son los usuarios finales, ya que, son ellos quienes conocen dichos datos y pueden ser capaces de descartar cualquier inconsistencia que pueda ser encontrada.

### 3.3.3.5. Interfaces, Ampliaciones y Reportes

Para poder asegurar aquellos pequeños sub-sistemas que se pierden del sistema original a la hora de adaptar el negocio de una empresa en SAP, es necesario la creación de interfaces. Las interfaces en SAP se encargan de ayudar en la integración del proceso del negocio y la sincronización de los datos entre dos o más sistemas SAP, o entre SAP y un sistema externo. Adicionalmente, para poder adaptar las implementaciones realizadas a las necesidades específicas de la empresa, es posible que sea necesario la creación de una **Ampliación**. Una Ampliación básicamente es una modificación pequeña que se les puede aplicar a programas estándares ya existentes dentro del Sistema SAP. Para esto, es necesario que dentro del equipo del proyecto, existan personas que sepan programar en ABAP, ya que dichas ampliaciones son realizadas en este lenguaje. El problema con esto, es que necesita ser probado constantemente, sobre todo cuando existan actualizaciones al software de SAP. Otra herramienta que suele ser importante durante esta fase son los reportes realizados, ya que hay muchas cosas que quizás la empresa necesite, que con las transacciones existentes no se pueden realizar. Los requerimientos para realizar dichos reportes deben ser establecidos lo más temprano posible, para que así puedan ser agrupados de acuerdo a la prioridad que tenga dicho reque-

rimiento. Para la realización de dichos requerimientos, es necesario contar con el grupo de programadores en ABAP.

### **3.3.4. Fase 4: Preparación Final**

En esta última fase hay varias tareas que deben ser llevadas a cabo para la culminación de un proyecto realizado en SAP. Estas tareas son las que se listan a continuación:

#### **3.3.4.1. Refinar el Sistema creado**

Una vez que se han realizado todas las pruebas al sistema y se haya recibido la retroalimentación del usuario final, se procederá a hacer las modificaciones pertinentes para adaptar el sistema a los posibles cambios que puedan surgir. Es posible que tanto las configuraciones, como las interfaces e ampliaciones tengan que sufrir alguna modificación.

#### **3.3.4.2. Planeación de la preparación de la Salida en Vivo**

Este plan consiste en el conjunto de actividades que deben ser ejecutadas las últimas semanas antes de salir en vivo. Este último término se refiere a la ejecución integral y puesta en producción de todo el sistema por parte de los usuarios finales. Algunas de estas actividades a realizar, son las que se listan a continuación:

- Tareas variadas
- Establecer un calendario y los hitos principales.
- Estimar tiempo de carga de datos por cada sub-carga
- Asignación de cada tarea a una persona
- Establecer un período y el procedimiento para desconectar el sistema legal previo.

- Procedimiento de limpieza de datos

Este plan puede ser revisado por los gerentes del proyecto, los ejecutivos, equipo técnico y líderes para su posterior aprobación.

#### **3.3.4.3. Entrenamiento del Usuario Final**

El autor señala que una regla general que se debería seguir en todo desarrollo en SAP es que el 10 de todo el tiempo invertido en el desarrollo deberá ser tomado para el entrenamiento. De este tiempo, al menos el 1 debería ser tomado para el entrenamiento de los ejecutivos de la empresa. El entrenamiento se hace necesario, ya que las personas que laboran en una empresa, por lo general realizan sus tareas diarias de una forma ya establecida. Por lo tanto, es necesario que sean enseñados para poder adaptarse a las nuevas tecnologías.

#### **3.3.4.4. Transferencia de Conocimiento**

En este punto es importante que todos los conocimientos adquiridos por los consultores durante el proceso del desarrollo del proyecto, como lo es la instalación detallada, sean enseñados a los empleados de la compañía, para que así, puedan replicar el sistema en otros lugares. Ellos deben transmitirles los conocimientos acerca de SAP adicionalmente, ya que los empleados en muchas ocasiones no tienen conocimiento acerca de este tipo de tecnologías.

#### **3.3.4.5. Administración del Sistema**

En este punto, el equipo encargado de realizar las pruebas al sistema y a los servidores es el equipo de Técnicos (Basis). Aquí es donde se verifica si son necesarios más servidores o más hardware.

#### **3.3.4.6. Migración de Datos**

En esta etapa se realiza la migración de los datos restantes desde el sistema existente al nuevo sistema creado en SAP. En este momento, el sistema antiguo permanece funcionando por un tiempo hasta que toda la data migrada sea validada.

#### **3.3.4.7. Pruebas Finales y Entonación**

En este punto se realizan pruebas de volúmen y se procede a colapsar el sistema, para verificar que puede atender gran cantidad de solicitudes concurrentes, y se realizan las modificaciones pertinentes. En este punto comienza la puesta en vivo del sistema completo.

### **3.4. Aplicación para el Proyecto de Pasantía**

Para este proyecto, se llegaron aplicar cada una de las fases de esta metodología, en su respectivo orden.

En el próximo capítulo se va a explicar el conjunto de actividades realizadas en las fases que formaron parte de este proyecto.

# Capítulo 4

## Desarrollo

En este capítulo se describe de manera detallada todo el proceso de desarrollo de este proyecto de pasantía. Para ello, se va a describir como fueron desarrolladas cada una de las fases involucradas.

### 4.1. Fase 1: Preparación Previa

Durante esta fase se realizaron diferentes reuniones organizadas por la tutora industrial con el fin de tener un entrenamiento en el software SAP. Para ello, distintos empleados de la compañía estuvieron en dichas reuniones para dictar los talleres. Los talleres recibidos fueron acerca de los siguientes módulos:

- Módulo de Gestión de Calidad (QM)
- Módulo de Gestión de Materiales (MM)
- Módulo de Gestión Financiera (FI)
- Módulo de Controlling (CO)



- Módulo de Ventas y Distribución (SD)

Para cada módulo fueron explicados los conceptos básicos que están involucrados en cada uno. Específicamente en el taller de Ventas y Distribución (SD), se explicó el ciclo de ventas general. Para ello, la persona encargada, explicó por detalle cada una de las etapas por las cuales pasa un proceso de ventas en general (Realización del Pedido de Ventas, Entrega, Facturación). Adicionalmente, fue impartido un taller sobre el lenguaje de Programación ABAP/4, para conocer un poco acerca de la estructura del lenguaje. Durante este taller, se recibieron conocimientos acerca de: estructuras de control, herramientas que posee el lenguaje para emitir reportes como por ejemplo: smartforms, que son una especie de formulario que se pueden imprimir. Además, se recibió material acerca de otro template que fue desarrollado para otro tipo de industria, pero que sirvió como base para ver como el módulo de Ventas y Distribución es puesto en práctica, y para servir como ejemplo a la hora de implantar dicho módulo a la empresa de bebidas de consumo masivo.

## 4.2. Fase 2: Business Blueprint

Durante esta fase fueron desarrolladas las siguientes actividades:

- Identificar el proceso de Master Data (Datos Mestros) a desarrollar en SAP.
- Identificar el proceso de Sales (Pedidos de Venta) a desarrollar en SAP.
- Identificar el proceso de Shipping and Transportation (Embarque y transporte) a ser desarrollado en SAP.
- Identificar el proceso de Billing (Facturación) a ser desarrollado en SAP.

Como fue explicado en el capítulo anterior, en esta fase se detallan cada uno de los procesos del negocio de la compañía para la cual se está llevando el proyecto; esto involucra

conocer los procesos actuales y como se podrían aplicar dichos procesos en SAP.

Para el caso de la empresa de bebidas de Consumo Masivo para la cual se está trabajando, se van a analizar cada uno de los procesos involucrados en Ventas y Distribución, para ello, se procederá en las sub-secciones siguientes a detallar cada una de las actividades mencionadas en la lista anterior.

#### 4.2.1. Identificación del proceso de Master Data (Datos Maestros)

En esta actividad, los datos maestros que fueron estudiados e identificados para el caso de la empresa de bebidas de consumo masivo para la cual se está realizando esta configuración, consistieron en los datos que forman parte de la estructura organizativa de la empresa. Para ello se realizó un estudio con otras empresas que manejan el mismo rubro para saber cuál es el camino a seguir. En el siguiente gráfico se muestra la organización de SSA BEVERAGE:

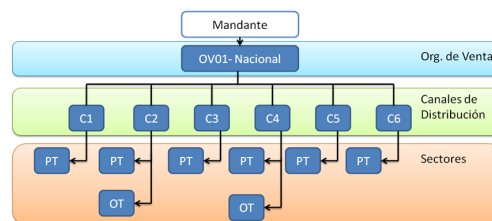


Figura 4.1: Estructura de la Empresa definida para el Módulo SD (Parte 1)



Figura 4.2: Estructura de la Empresa definida para el Módulo SD (Parte 2)

Para las Organizaciones de Ventas, se fijaron dos para configurar en la siguiente etapa, y son las que se muestran en la siguiente lista:

- Nacional: Se va a encargar de atender a los clientes nacionales

- Internacional : Se va a encargar de atender las exportaciones de los productos

Luego se definieron los Canales de Distribución. Para este caso, se decidió fijar 7 canales de distribución. Estos son los que se listan en la siguiente lista:

- Distribución Directa: Este canal será utilizado para atender a los clientes en general.
- Intermediario: Este canal será utilizado para atender a aquellos clientes que tengan algún intermediario con el que a través de ellos realicen alguna compra de materiales.
- Eventos: Este canal será utilizado para captar clientes a través de eventos que realice la compañía.
- Institucionales: Esta canal será empleado para atraer a clientes a través de instituciones sin fines de lucro.
- Mayoristas: Este canal será empleado para atraer a aquellos clientes se compren al mayoreo.
- Traslados: Este canal será utilizado para aquellos clientes que estén a lo largo del país.
- Exportaciones: Este canal será utilizado para aquellos clientes que estén en el exterior.

A continuación, se procedió a establecer los sectores en los cuales serán agrupados los materiales a producir, como se muestra en la siguiente lista:

- Producto Terminado: Consiste en el grupo conformado por todas aquellas bebidas que vayan a ser vendidas por la empresa, que ya han sido elaboradas.
- Otros: En este grupo entran aquellos materiales que fueron usados para la elaboración de los productos finales, del cual quedó un remanente, como por ejemplo, del mosto sobrante de la elaboración de la cerveza.

Luego se procedió a establecer varias oficinas de ventas. Para esto, se decidió mantener la misma distribución de los Centros de Distribución que fueron colocados dentro del módulo de Gestión de Materiales (MM), en la cual se dividió al territorio en varias regiones. Luego, se definió una oficina por cada región, quedando las mismas de la siguiente manera:

- Capital
- Central
- Los Llanos
- Nororiental
- Insular
- Occidental
- Andes
- Zuliana

Posteriormente, se hizo la identificación de los puestos de expedición por cada centro de distribución. Dada la distribución de las oficinas de ventas en el párrafo anterior, y que las mismas presentan una distribución equivalente a los Centros de Distribución, se decidió establecer dos puestos de carga por cada uno, como se muestra a continuación:

- Despacho: Este puesto se encarga de la entrega de los materiales a los clientes.
- Retorno: Este puesto se encarga de recibir aquellos materiales que son enviados por los clientes, como por ejemplo: botellas retornables, etc.

Para establecer los puestos de expedición en la siguiente etapa, fueron identificados dos puestos de carga por cada puesto de expedición.

### 4.2.2. Identificación del proceso de Sales (Pedidos de Ventas)

Dentro del proceso de ventas que es manejado por el módulo SD, éste es el punto de partida, ya que el primer paso para poder realizar el ciclo de una venta de un material es la realización de un Pedido de Ventas. Para ello, es creado un documento de Pedido de Ventas que contiene varios tipos de información, como por ejemplo:

- Información personal del cliente
- Información del(los) material(es) a adquirir
- Información sobre las condiciones de pago

Para ello, es necesario identificar dos puntos importantes en esta etapa:

- Tipo de Pedido a realizar: Representa el tipo de Documento a crear
- Clases de Mensajes: Contiene las rutinas de impresión de documentos
- Tipos de Posición: Cada renglón del documento relacionado con cada material representa una posición del documento
- Clases de condición: Como su nombre lo indica, son condiciones. Éstas son establecidas a los precios de los productos
- Esquemas de Cálculo: Contiene las clases de Condición a utilizar

Para este proyecto, se decidió utilizar el Pedido Estándar, el cual permite realizar un pedido normal. Para las clases de mensajes se decidió adoptar la estándar, al igual que los tipos de posición y las clases de condición.

### 4.2.3. Identificación del proceso de Shipping and Transportation (Embarque y Transporte)

El segundo paso dentro del ciclo general de una venta en SAP es el proceso de Entrega (Shipping), el cual consiste en generar un Documento de Entrega, en el cual consta, además de los datos provenientes del Documento de Ventas al cual hace referencia, los datos relacionados con el puesto de expedición en el cual será entregado el material solicitado. Para esto es necesario definir el tipo de Documento a utilizar. Dado que en el segmento anterior se decidió definir un Pedido Estándar, para esta parte del ciclo de Ventas se decidió definir un Documento de Entrega Estándar, ya que es el que corresponde con el tipo de pedido establecido anteriormente.

### 4.2.4. Identificación del proceso de Billing (Facturación)

El tercer y último paso dentro del ciclo general de una venta en SAP es el proceso de Facturación (Billing). Para este caso se define un tipo de documento de Facturación acorde con el tipo de Pedido de Ventas y tipo de Entrega definidos anteriormente. Para el pedido Estándar definido más atrás, fueron definidos dos tipos de Documentos de Facturación: El primero, la Facturación Estándar, el cual es utilizado para crear la factura de un pedido existente, para así cerrar el ciclo normal de la venta de un material. El segundo, consiste en la **Anulación de Factura**, éste es usado para una vez que haya sido creada la factura de un pedido, y halla ocurrido algún problema durante el proceso en el cual fue realizado dicho pedido, el mismo se pueda anular, y así se pueda volver a crear.

### 4.3. Fase 3: Realización (Construcción y Pruebas)

Las actividades desarrolladas en esta fase se dividieron en dos grupos principales: Configuración Base del Módulo de Ventas y Distribución (SD) para SSA Beverage y Elaboración de las facturas legales. A continuación, se procederá a detallar cada uno de los dos grupos.

#### 4.3.1. Configuración Base del Módulo de Ventas y Distribución para SSA Beverage

Para poder realizar las configuraciones, hubo que dividirlas en varios grupos, los cuales se muestran a continuación:

##### 4.3.1.1. Configuración de la parte organizativa de la Empresa

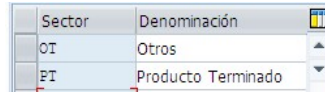
Este grupo fue compuesto por las configuraciones realizadas para definir la estructura de SSA Beverage. Para ello, se ingresó a SAP, y a través del Customizing, fue posible efectuarlas. Se comenzó configurando las Organizaciones de Ventas, Canales de Distribución, Sectores y Áreas de Ventas definidos en la fase anterior. A continuación, se muestran figuras de las tres configuraciones realizadas: Luego, se procedió a realizar la configuración de las unidades

OrgVt	Denominación
OV01	Nacional
OV02	Internacional

Figura 4.3: Organizaciones de Ventas definidas para el Módulo SD

Canal distrib.	Denominación
C1	Distribucion directa
C2	Intermediario
C3	Eventos
C4	Institucionales
C5	Mayoristas
C6	Traslados
C7	Exportaciones

Figura 4.4: Canales de Distribución definidos para el Módulo SD



Sector	Denominación
OT	Otros
PT	Producto Terminado

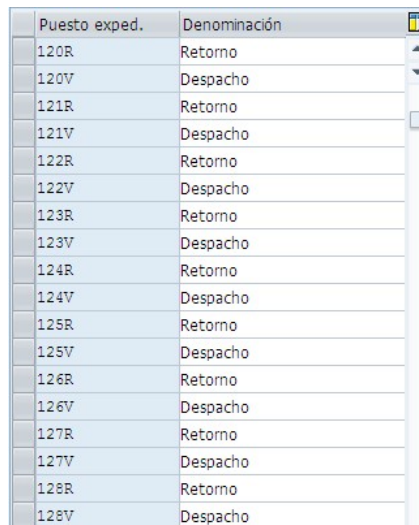
Figura 4.5: Sectores definidos para el Módulo SD



OrgVt	Denominación	CDs	Denominación	Se	Denominación	Status
OV01	Nacional	C1	Distribucion directa	PT	Producto Terminado	
OV01	Nacional	C2	Intermediario	OT	Otros	
OV01	Nacional	C2	Intermediario	PT	Producto Terminado	
OV01	Nacional	C3	Eventos	PT	Producto Terminado	
OV01	Nacional	C4	Institucionales	OT	Otros	
OV01	Nacional	C4	Institucionales	PT	Producto Terminado	
OV01	Nacional	C5	Mayoristas	PT	Producto Terminado	
OV01	Nacional	C6	Traslados	PT	Producto Terminado	
OV02	Internacional	C7	Exportaciones	PT	Producto Terminado	

Figura 4.6: Áreas de Ventas definidas para el Módulo SD

encargadas de la entrega de los materiales (Puestos de Expedición y Puestos de Carga). Para el primero se muestra la figura que muestra la configuración realizada, y para el segundo, en la imagen se muestra parte de las configuraciones realizadas. Después que fueron creadas



Puesto exped.	Denominación
120R	Retorno
120V	Despacho
121R	Retorno
121V	Despacho
122R	Retorno
122V	Despacho
123R	Retorno
123V	Despacho
124R	Retorno
124V	Despacho
125R	Retorno
125V	Despacho
126R	Retorno
126V	Despacho
127R	Retorno
127V	Despacho
128R	Retorno
128V	Despacho

Figura 4.7: Puestos de Expedición definidos para el Módulo SD

cada una de las unidades antes mencionadas dentro de SAP, las configuraciones posteriores consistieron en crear las relaciones entre las mismas. En la tabla se muestra una lista con las configuraciones realizadas. En el apéndice se podrán ver las imágenes que reflejan las





Figura 4.8: Parte de los Puestos de Carga definidos para el Módulo SD

<b>Configuraciones Realizadas</b>	
Asignar las Organizaciones de Ventas creadas a la Sociedad	
Asignar los Canales de Distribución a las Organizaciones de Ventas	
Asignar los Sectores a las Organizaciones de Ventas	
Asignar las Oficinas de Ventas a las Áreas de Ventas	
Asignar las Organizaciones de Ventas y Canales de Distribución a los Centros de Distribución	
Asignar los Puestos de Expedición a los Centros de Distribución	

Cuadro 4.1: Configuraciones realizadas para establecer la relación entre las distintas unidades componentes de la estructura de SSA Beverage

configuraciones antes mencionadas.

### 4.3.2. Elaboración de las Facturas Legales

Para el desarrollo de las Facturas Legales fue necesario el uso de los smartforms, ya que los mismos producen formularios imprimibles. Para ello se tomó como ejemplo un formulario estándar ya existente en el sistema para hacer el diseño de uno propio. El mismo consta de varias partes bien diferenciadas:

- Pesos y Volumen: Contiene información sobre los pesos y volúmenes totales de los materiales adquiridos.
- Condiciones de Expedición y Pago: Contiene información sobre la forma en la cual se realiza la expedición y sobre la forma de pago.
- Dirección del Cliente
- Datos Generales: Contiene datos generales del cliente, como: número del cliente, nombre,

etc.

- Posiciones de Factura: Contiene información detallada de cada material adquirido, como por ejemplo: Número de material, descripción, cantidad, precio unitario, etc.

En la Figura

### 4.3.3. Elaboración del programa de Carga Masiva de Datos para el Maestro de Clientes

Para la elaboración de este programa fue necesario el uso de las siguientes herramientas SAP:

- Editor ABAP
- Uso de Grabaciones (Transacción SHDB ó SM35)
- Uso de Includes
- Uso de la transacción XD01 (Transacción para Creación de un registro en el Maestro de Clientes)
- Selection Screens (Pantallas de Selección)

El Editor ABAP fue utilizado para el desarrollo del programa. Para esto, decidió modularizar el mismo. En otras palabras, el programa principal sólo iba a contener el uso de variables y llamadas a procedimientos. Para poder lograr esto, fue necesario la creación de dos Includes: uno para la declaración de variables, constantes, estructuras de datos, etc. Y otro, para la definición de funciones. Para poder recibir el archivo por pantalla, fue necesario el uso de los **Selection Screens**, los cuales permiten colocar la dirección del archivo. La primera función a implementar fue una que permitiera la lectura de un archivo en dos formatos: Texto Plano

(.txt) y de Celdas (.xls ó Excel como se le conoce comercialmente). En esta función, se llama a una función estándar de SAP la cual permite leer un archivo en excel y transformar la data a una estructura especial. Luego dicha estructura fue transformada a una estructura definida en uno de los Includes. El proceso para un archivo plano fue análogo. Luego se procedió a crear una grabación para Batch Input a través de la transacción **SM35**, con la cual, usando la transacción **XD01** como base, se procedió a realizar la simulación de la carga de datos. Luego, el código generado por dicha grabación, fue utilizado dentro de otra función creada para procesar la data cargada anteriormente. De esta manera, se logra la carga masiva de datos en el Maestro de Clientes para el módulo de Ventas y Distribución.

#### 4.4. Fase 4: Presentación y Resultados Finales

Aqui iran los resultados finales



SSA BEVERAGE C.A.  
Av. Principal Los Ruices, Edif. 1, Caracas, Edo Miranda, Venezuela 1060.  
0212-235-4578

Empresa  
DYM3000 C.A  
VENEZUELA

**CONDICIONES**  
**Pago:**  
Hasta el 30.09.2013  
se obtiene el 2,000  
% de descuento  
  
Hasta el 15.10.2013  
sin deducción

**Entrega:**  
FOB  
FOB

**Factura**

**INFORMACIÓN DE PAGO**  
**Número/Fecha del documento:**  
90000006 / 19.08.2013  
**Número/Fecha de la nota de entrega:**  
80000006 / 19.08.2013  
**Número/Fecha del Pedido:**  
13 / 19.08.2013  
**Número/Fecha de referencia:**  
52 / 19.08.2013

**Cliente:**  
12  
**Moneda:**  
VEF  
**Importe de la factura:**  
23.000,00

**PESOS - VOLÚMENES**  
**Peso Bruto:** 900 KG  
  
**Peso Neto:** 900 KG

**DETALLES DE FACTURA**

Posición	Material	Denominación	Cantidad	Unidad			
000001	840000001	Refresco Limón Soda	100	CAJ			
	Total		230,00	VEF	1	CAJ	23.000,00
	<b>Importe de la factura</b>						23.000,00
	<b>Importe con derecho a descuento</b>						0,00

Figura 4.9: Ejemplo de Factura creada para el módulo SD

# Capítulo 5

## Resultados

AQUI VA EL CONTENIDO DE RESULTADOS.

Los resultados que se quieren evaluar están divididos en tres partes: Primero, es necesario verificar la calidad de estimación del parámetro de Hurst usando los métodos implementados, segundo, es importante medir el tiempo necesario para cada estimación y, por último, es importante comprobar si se pueden detectar ataques de denegación de servicio mediante la variación del parámetro y el uso del mecanismo de ventanas deslizantes.

La descripción del computador utilizado para ejecutar las pruebas se presenta en la sección 5.1. Los archivos utilizados y los resultados de la verificación de la estimación del parámetro de Hurst y del tiempo necesario para la estimación se presentan en la sección 5.2. Estos resultados muestran qué tan preciso es la implementación de los métodos y cuánto tiempo en promedio necesita la herramienta para estimar el valor del parámetro de Hurst en una ventana. La metodología para la evaluación de la detección de ataques de denegación se muestra en la sección. Por otro lado en la sección se muestran los resultados de utilizar hilos de ejecución para observar el cambio del parámetro de Hurst en el tiempo en modo *offline* con trazas producidas por `tcpdump`.

## 5.1. Máquina de prueba

Todas las pruebas fueron realizadas con el computador cuyas características aparecen en el cuadro 5.1. La herramienta **d2Hgr** fue compilada sobre esa máquina, exclusivamente con optimización gcc de nivel 2, que optimiza el binario resultante de la compilación con respecto a bloques repetitivos y cálculos punto flotante.

CPU	Althon 64 X2 4200+
RAM	4 GB
Distribución	Slackware 13.0 x86_64
Kernel Linux	2.6.29.6
Versión gcc	4.3.3

Cuadro 5.1: Computador usado para las pruebas

## 5.2. Validación de los estimados

Se tenía que probar que las implementaciones de los 3 métodos producían estimaciones razonables del parámetro de Hurst para saber de antemano que tan confiable podrian ser la estimaciones cuando se usasen junto con el mecanismo de ventanas deslizantes. Para la validación de los métodos implementados se probaron los mismos con datos producidos por el algoritmo de Paxson [Pax95].

El algoritmo de Paxson permite crear una serie de tiempo pseudo-aleatoria en el dominio del tiempo de forma rápida, y permite especificar el parámetro de Hurst a ser simulado y el número de datos que debe tener la serie de tiempo resultante, manteniendo la media en 0 y la varianza en 1, con valores uniformemente distribuidos en el intervalo  $[0, 2\pi]$ . Mediante el algoritmo se puede simular un proceso estacionario autosimilar con dependencia de largo alcance [Pax95].

La implementación del algoritmo de Paxson utilizada es la que viene en el paquete **fArma**

de R. La validación contó con 6 tamaños diferentes para la serie de tiempo  $X_k$  generada por el algoritmo de Paxson. Estos tamaños fueron  $\{1024, 2048, 4096, 8192, 16384, 32768\}$ . Por cada tamaño se generaron 10 series de tiempo  $X_k$  para cada uno de los 9 valores del parámetro de Hurst que se quería simular. Estos valores fueron  $\{0,55, 0,60, 0,65, 0,70, 0,75, 0,80, 0,85, 0,90, 0,95\}$ .

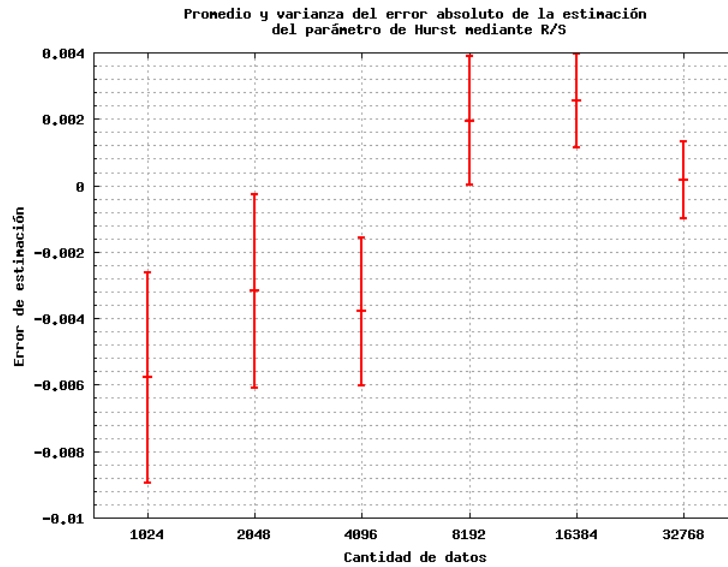


Figura 5.1: Promedio y varianza del error absoluto de la estimación del parámetro de Hurst para el método estadístico R/S

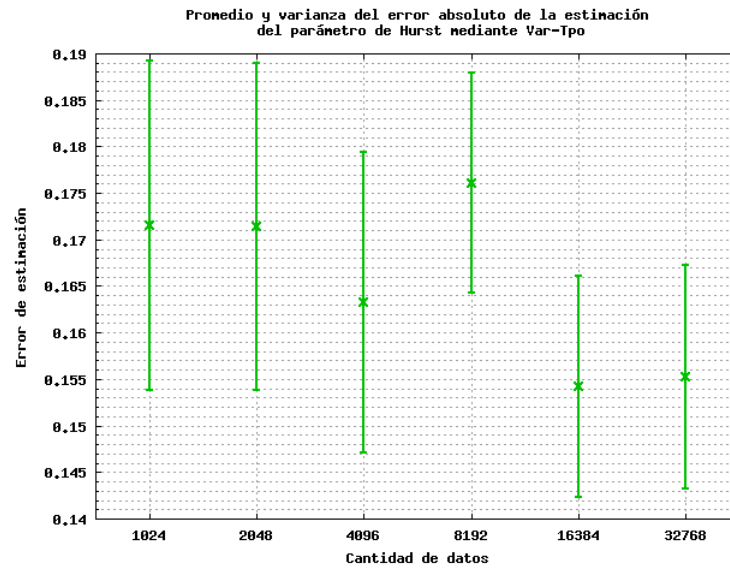


Figura 5.2: Promedio y varianza del error absoluto de la estimación del parámetro de Hurst para el método de gráficas de varianza-tiempo

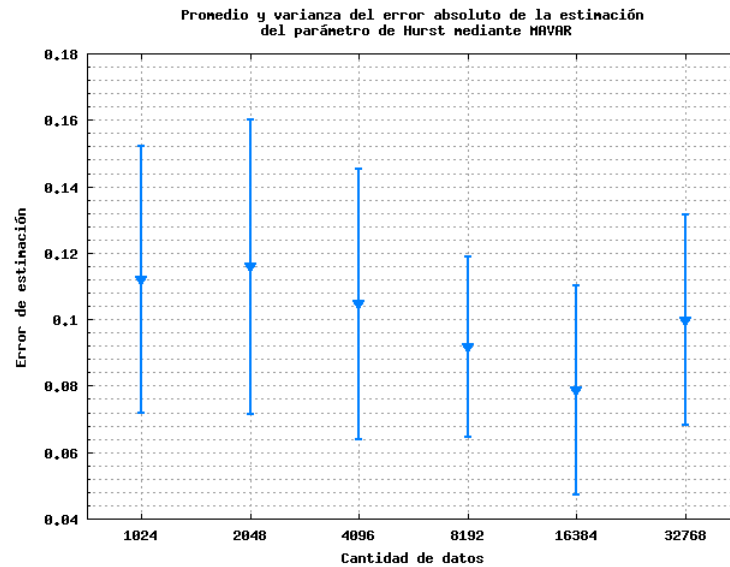


Figura 5.3: Promedio y varianza del error absoluto de la estimación del parámetro de Hurst para el método de varianza modificada de Allan



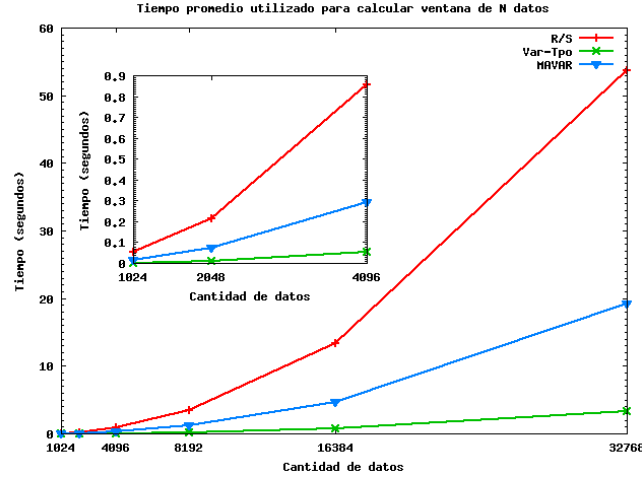


Figura 5.4: Tiempo promedio utilizado para estimar una ventana de 1024 a 32768 datos para los métodos implementados

Observando las figuras 5.1, 5.2 y 5.3, lo que se puede evidenciar es que, en general, el promedio del error absoluto de las estimaciones del parámetro de Hurst de los métodos mejoran y la varianza disminuye cuando se toman en cuenta un mayor número de datos para la ventana de estimación. Sin embargo, para los dos métodos que dependen de cálculos con la varianza el promedio del error absoluto de la estimación, incluso con el mayor número de datos es mayor a 0,1. El método del estadístico R/S es el método que se comporta mejor ya que tiene, en general, un error absoluto muy cerca a 0 con una varianza mínima.

De acuerdo a [WY08] si queremos analizar los datos en tiempo real para una red se tendría que hacer una estimación como mínimo cada segundo, por lo que una estimación no puede tardar más de un segundo en realizarse. Esto es seriamente limitante ya que cada método tiene un tiempo de ejecución muy diferente. Observando la figura 5.4, con esta restricción la única posibilidad que se tiene es limitar al método del estadístico R/S a 4096 datos, al método de gráficas varianza-tiempo a 16834 datos y al método de varianza modificada de Allan a 4096 datos. Dado que para comparar los métodos es importante usar la misma cantidad de datos, decidimos trabajar con 4096 datos. El limitar el número de datos va a afectar la estimación

del parámetro de Hurst directamente como se muestra en las figuras 5.1, 5.2 y 5.3.

A pesar de estos problemas, seguimos adelante con el proyecto porque no queremos una sola estimación puntual y precisa de la traza, sino detectar su variación, por lo que optamos por un buen balance entre el tiempo de ejecución y el número de datos a utilizar [WY08].

## Capítulo 6

# Conclusiones y recomendaciones

AQUÍ VAN LAS CONCLUSIONES Y RECOMENDACIONES.

Como resultado de este proyecto se logró implementar una herramienta de línea de comando que pudiese crear series de tiempo a partir de trazas producidas por `tcpdump` para estimar el parámetro de Hurst y graficar sus resultados. La estimación se hizo mediante el mecanismo de ventana deslizante, limitando el tamaño de datos para cada ventana de forma de probar si de esta forma, utilizando las técnicas descritas, se podían obtener estimaciones del parámetro de Hurst en tiempo real que permitieran detectar ataques de denegación de servicio.

Los resultados muestran que aunque la herramienta implementada logra crear una serie de tiempo que puede estimar el parámetro de Hurst con un error aceptable y puede graficar sus resultados, su detección de ataques de denegación de servicio en tiempo real es mejorable. Si bien se logra detectar cambios drásticos, en algunos casos cerca de la región del ataque, la cantidad de falsos positivos y errores hace su uso casi imposible.

Para seguir trabajando en la detección de ataques de denegación de servicio usando el parámetro de Hurst se presentan las siguientes recomendaciones:

- Emplear algún mecanismo de muestreo de los valores de  $n$  utilizados en la estimación del parámetro de Hurst en una ventana con todos los métodos para mejorar la precisión y posiblemente disminuir el tiempo necesario para una estimación:

En la herramienta implementada, no se usó ningún mecanismo de muestreo en la estimación del parámetro de Hurst, implementando los métodos como fueron escritos en el capítulo 2. Esto significa que se tomaron en cuenta todos los tamaños  $n$  viables para la creación de las gráficas log – log. Parte de la lentitud de los métodos en dar un resultado está en tomar en cuenta todos los  $n$  posibles. Por otro lado, de acuerdo con [BP05] usando el método de varianza modificada de Allan, el tomar todos los  $n$  acercándose a  $N/3$  comienza a afectar seriamente la estimación. Sin embargo, en el artículo no se presenta una forma determinística que describa hasta que punto se debería tomar en cuenta los  $n$  para una estimación.

- Encontrar algún método de utilizar la información obtenida en la estimación de una ventana de tal forma que los cálculos hechos para la primera ventana aceleren la estimación de la siguiente ventana [WY08].
- Al poder estimar una ventana en menor tiempo también sería recomendable aumentar el número de datos utilizados para una estimación. Cómo no se quiere aumentar el tiempo de traza real  $w$  usado en una estimación, esto implicaría usar un  $c$  más pequeño [HDTI00].
- En la estimación de una ventana, si se logra encontrar una forma de dividir el cálculo de tal forma de poder aplicar una paralelización, el uso de más de un hilo de ejecución para estimación de la ventana podría ser probado [HDTI00].
- En lugar de utilizar un único tamaño de  $s$  en el mecanismo de ventana deslizante, se podría usar un valor flexible a los cambios de la red, de tal forma que si el parámetro

de Hurst cambia repentinamente entre dos ventanas, se acorte  $s$  con tal de verificar si en realidad hay una sucesión de paquetes uniformemente espaciados que puedan ser parte un ataque de denegación de servicio, minimizando la cantidad de falsos positivos [WY08].

# Bibliografía

- [AAK04] Sameera Abar, Toru Abe, and Tetsuo Kinoshita. A next generation knowledge management system architecture. In *AINA (2)*, pages 191–195, 2004.
- [BKM96] Sabyasachi Basu, Steve Klivansky, and Amarnath Mukherjee. Time series models for internet traffic, 1996.
- [BP05] Stefano Bregni and Luca Primerano. Using the modified allan variance for accurate estimation of the hurst parameter of long-range dependent traffic. *CoRR*, abs/cs/0510006, 2005.
- [Bre02] Stefano Bregni. *Synchronization of Digital Telecommunications Networks*. John Wiley & Sons, Inc., New York, NY, USA, 2002.
- [CB96] Mark E. Crovella and Azer Bestavros. Self-similarity in world wide web traffic evidence and possible causes. *IEEE/ACM Transactions on Networking*, 5:835–846, 1996.
- [DDHT08] R. Dobrescu, M. Dobrescu, D. Hossu, and S. Taralunga. Using internet traffic self-similarity for detection of network anomalies. In *11th International Conference on Optimization of Electrical and Electronic Equipment, 2008. OPTIM 2008*, pages 81–86, 2008.

- [FP01] Sally Floyd and Vern Paxson. Difficulties in simulating the internet. *IEEE/ACM Transactions on Networking*, 9:392–403, 2001.
- [Gon03] Fengmin Gong. Deciphering detection techniques: Part 3 denial of service detection. *White paper, McAfee Network Security Technologies Group and Network Associates*, 2003.
- [HDTI00] T. Hagiwara, H. Doi, Hideki Tode, and Hiromasa Ikeda. High-speed calculation method of the hurst parameter based on real traffic. In *LCN*, pages 662–669, 2000.
- [HZ97] Oliver Yang H.F. Zhang, Y.T. Shu. Estimation of hurst parameter by variance-time plots. *IEEE*, 0-7803-3905-3, 1997.
- [ITU<sup>+</sup>07] Ryuji Igarashi, Akinori Takahashi, Hiroshi Ueda, Yutaka Nasuno, Yukio Iwaya, Masato Sakata, and Tetsuo Kinoshita. A proposal for real time hurst parameter derivation. *IEEJ Transactions on Electronics, Information and Systems*, 127(6):968–969, 2007.
- [JMR04] David Dittrich Jelena Mirkovic, Sven Dietrich and Peter Reiher. *Internet Denial of Service: Attack and Defense Mechanisms*. Prentice Hall PTR, 2004.
- [KAIK07] Susumu Konno, Sameera Abar, Yukio Iwaya, and Tetsuo Kinoshita. Effectiveness of autonomous network monitoring based on intelligent-agent-mediated status information. In *IEA/AIE*, pages 1078–1087, 2007.
- [KFR02] Thomas Karagiannis, Michalis Faloutsos, and Rudolf H. Riedi. Long-range dependence: Now you see it, now you don’t! In *In IEEE GLOBECOM, Global Internet Symposium*, page 2002, 2002.

- [KIAK04] Susumu Konno, Yukio Iwaya, Toru Abe, and Tetsuo Kinoshita. Design of network management support system based on active information resource. *Advanced Information Networking and Applications, International Conference on*, 1:102, 2004.
- [Kle76] L. Kleinrock. *Queueing Systems, Volume 2: Computer Applications*. Wiley, 1976.
- [Li06] Ming Li. Change trend of averaged hurst parameter of traffic under ddos flood attacks. *Computers & Security*, 25(3):213–220, 2006.
- [LTWW93] Will E. Leland, Murad S. Taqqu, Walter Willinger, and Daniel V. Wilson. On the self-similar nature of ethernet traffic, 1993.
- [LW91] Will E. Leland and Daniel V. Wilson. High time-resolution measurement and analysis of lan traffic: Implications for lan interconnection. In *INFOCOM*, pages 1360–1366, 1991.
- [Man82] Benoit B. Mandelbrot. *The Fractal Geometry of Nature*. W. H. Freeman, August 1982.
- [MM02] Richard Mortier and C Richard Mortier. Internet traffic engineering, 2002.
- [MVN97] Sándor Molnár, Attila Vidács, and Arne A. Nilsson. Bottlenecks on the way towards fractal characterization of network traffic: Estimation and interpretation of the hurst parameter, 1997.
- [MVS01] David Moore, Geoffrey Voelker, and Stefan Savage. Inferring internet denial-of-service activity. In *In Proceedings of the 10th Usenix Security Symposium*, pages 9–22, 2001.



- [Odl01] Andrew Odlyzko. Internet growth: Myth and reality, use and abuse. *Journal of Computer Resource Management*, 102:23–27, 2001.
- [Pax95] Vern Paxson. Fast approximation of self-similar network traffic. Technical report, Lawrence Berkeley Laboratory and EECS Division, University of California, 1995.
- [PF95] Vern Paxson and Sally Floyd. Wide-area traffic: The failure of poisson modeling. *IEEE/ACM Transactions on Networking*, 3:226–244, 1995.
- [PW00] Kihong Park and Walter Willinger. *Self-Similar Network Traffic and Performance Evaluation*. John Wiley & Sons, Inc., New York, NY, USA, 2000.
- [SSO07] Oleg Sheluhin, Sergey Smolskiy, and Andrew Osin. *Self-Similar Processes in Telecommunications*. John Wiley & Sons, Inc., New York, NY, USA, 2007.
- [SV01] Biplab Sikdar and Kenneth S. Vastola. The effect of tcp on the self-similarity of network traffic. In *Proc. of the 35th Conf. on Information Sciences and Systems*, pages 21–23, 2001.
- [TIU<sup>+</sup>13] Akinori Takahashi, Ryuji Igarashi, Hiroshi Ueda, Yutaka Nasuno, Yukio Iwaya, and Tetsuo Kinoshita. Online network observation to detect traffic anomaly. *IEICE technical report*, 107(221):57–62, 20070913.
- [USKS02] Takahiro Uchiya, Takuo Suganuma, Tetsuo Kinoshita, and Norio Shiratori. An architecture of active agent repository for dynamic networking. In *AAMAS '02: Proceedings of the first international joint conference on autonomous agents and multiagent systems*, pages 1266–1267, New York, NY, USA, 2002. ACM.
- [WP98] Walter Willinger and Vern Paxson. Where mathematics meets the internet. *Notices of the American Mathematical Society*, 45:961–970, 1998.

- [WWT03] Rolf H. Riedi Walter Willinger, Vern Paxson and Murad S. Taqqu. Long-Range Dependence and Data Network Traffic. In G. Oppenheim P. Doukhan and M. Taqqu, editors, *Theory and Applications of Long-Range Dependence*, pages 373–407. Birkäuser, Berlin, 2003.
- [WY08] Jiangtao Wang and Geng Yang. An intelligent method for real-time detection of ddos attack based on fuzzy logic. *Journal of Electronics (China)*, 25(4):511–518, 2008.
- [XLLH04] Y. Xiang, Y. Lin, W.L. Lei, and S.J. Huang. Detecting ddos attack based on network self-similarity. *IEE Proceedings - Communications*, 151(3):292–295, 2004.

# Apéndice A

## Información adicional de d2Hgr

Asignación de organización de ventas a sociedad				
OrgVt	Denominación	So.	Nombre empresa	Status
OV01	Nacional	1000	SSA BEVERAGE	
OV02	Internacional	1000	SSA BEVERAGE	

Figura A.1: Asignación de las Organizaciones de Ventas a la Sociedad

Asignación de organización de ventas a canal de distribución				
OrgVt	Denominación	CDIs	Denominación	Status
OV01	Nacional	C1	Distribucion directa	
OV01	Nacional	C2	Intermediario	
OV01	Nacional	C3	Eventos	
OV01	Nacional	C4	Institucionales	
OV01	Nacional	C5	Mayoristas	
OV01	Nacional	C6	Traslados	
OV02	Internacional	C7	Exportaciones	

Figura A.2: Asignación de las Organizaciones de Ventas a los Canales de Distribución

Asignación de organización de ventas a sector				
OrgVt	Denominación	Se	Denominación	Status
OV01	Nacional	OT	Otros	
OV01	Nacional	PT	Producto Terminado	
OV02	Internacional	PT	Producto Terminado	

Figura A.3: Asignación de las Organizaciones de Ventas a los Sectores

AQUÍ VA EL CONTENIDO DE LOS APÉNDICES.

OrgVt	Denominación	CDIs	Denominación	Se	Denominación	OFVta	Denominación	Status
OV01	Nacional	C1	Distribucion directa	PT	Producto Terminado	A001	Capital	
OV01	Nacional	C1	Distribucion directa	PT	Producto Terminado	A002	Central	
OV01	Nacional	C1	Distribucion directa	PT	Producto Terminado	A003	Los Llanos	
OV01	Nacional	C1	Distribucion directa	PT	Producto Terminado	A004	Nororiental	
OV01	Nacional	C1	Distribucion directa	PT	Producto Terminado	A005	Insular	
OV01	Nacional	C1	Distribucion directa	PT	Producto Terminado	A006	Guayana	
OV01	Nacional	C1	Distribucion directa	PT	Producto Terminado	A007	Occidental	
OV01	Nacional	C1	Distribucion directa	PT	Producto Terminado	A008	Andes	
OV01	Nacional	C1	Distribucion directa	PT	Producto Terminado	A009	Zulana	
OV01	Nacional	C2	Intermediario	OT	Otros	A001	Capital	
OV01	Nacional	C2	Intermediario	OT	Otros	A002	Central	
OV01	Nacional	C2	Intermediario	OT	Otros	A003	Los Llanos	
OV01	Nacional	C2	Intermediario	OT	Otros	A004	Nororiental	
OV01	Nacional	C2	Intermediario	OT	Otros	A005	Insular	
OV01	Nacional	C2	Intermediario	OT	Otros	A006	Guayana	
OV01	Nacional	C2	Intermediario	OT	Otros	A007	Occidental	
OV01	Nacional	C2	Intermediario	OT	Otros	A008	Andes	
OV01	Nacional	C2	Intermediario	OT	Otros	A009	Zulana	
OV01	Nacional	C2	Intermediario	PT	Producto Terminado	A001	Capital	
OV01	Nacional	C2	Intermediario	PT	Producto Terminado	A002	Central	
OV01	Nacional	C2	Intermediario	PT	Producto Terminado	A003	Los Llanos	
OV01	Nacional	C2	Intermediario	PT	Producto Terminado	A004	Nororiental	
OV01	Nacional	C2	Intermediario	PT	Producto Terminado	A005	Insular	
OV01	Nacional	C2	Intermediario	PT	Producto Terminado	A006	Guayana	

Figura A.4: Parte de la asignación de las Oficinas de Ventas a las Áreas de Ventas

Esta parte del apéndice contiene el manual del usuario de la herramienta implementada, la descripción de sus archivos y el código fuente de algunos módulos de posible interés.

## A.1. Requerimientos de *software* y *hardware*

El programa resultante debería poder ser utilizado sobre cualquier maquina con más de 32 MB libres de RAM. Aunque no hay limitaciones para el procesador a usar, entre más nuevo el procesador y mayor número de núcleos tenga, más rápido hará los calculos la herramienta. En el caso de la memoria, entre más tenga disponible la herramienta, mayor cantidad de datos podrá utilizar en los cálculos.

- Para su compilación, el programa requiere una versión actualizada de gcc, el compilador C GNU. Se ha utilizado varias versiones para su compilación por lo cualquier versión

OrgVt	Denominación	CDistClima	Denominación	Ce.	Nombre 1	Status
0001	Org.ventas 0001	01	Canal distribución 1	0001	Werk 0001	
OV01	Nacional	C1	Distribucion directa	101	Centro Cerveza Venezolana	
OV01	Nacional	C1	Distribucion directa	120	Centro Región Capital	
OV01	Nacional	C1	Distribucion directa	121	Centro Región Central	
OV01	Nacional	C1	Distribucion directa	122	Centro Región de los Llanos	
OV01	Nacional	C1	Distribucion directa	123	Centro Región Nororiental	
OV01	Nacional	C1	Distribucion directa	124	Centro Región Insular	
OV01	Nacional	C1	Distribucion directa	125	Centro Región de Guayana	
OV01	Nacional	C1	Distribucion directa	126	Centro Región C. Occidental	
OV01	Nacional	C1	Distribucion directa	127	Centro Región de los Andes	
OV01	Nacional	C1	Distribucion directa	128	Centro Región Zulana	
OV01	Nacional	C2	Intermediario	120	Centro Región Capital	
OV01	Nacional	C2	Intermediario	121	Centro Región Central	
OV01	Nacional	C2	Intermediario	122	Centro Región de los Llanos	
OV01	Nacional	C2	Intermediario	123	Centro Región Nororiental	
OV01	Nacional	C2	Intermediario	124	Centro Región Insular	
OV01	Nacional	C2	Intermediario	125	Centro Región de Guayana	
OV01	Nacional	C2	Intermediario	126	Centro Región C. Occidental	
OV01	Nacional	C2	Intermediario	127	Centro Región de los Andes	
OV01	Nacional	C2	Intermediario	128	Centro Región Zulana	
OV01	Nacional	C3	Eventos	120	Centro Región Capital	
OV01	Nacional	C3	Eventos	121	Centro Región Central	
OV01	Nacional	C3	Eventos	122	Centro Región de los Llanos	
OV01	Nacional	C3	Eventos	123	Centro Región Nororiental	

Figura A.5: Parte de la asignación de las Organizaciones de Ventas y Canales de Distribución a los Centros de Distribución

mayor a la 4,2 debería funcionar.

- La librería `libpcap` es necesaria para darle las funcionalidades de manipulación de trazas `tcpdump` al programa. A partir de la versión 0,8 se encuentran todas las funciones que utiliza el programa.
- La compilación de la herramienta se hace más fácil con `make`, programa GNU. Cualquier versión mayor a 3,6 no debería dar problemas.
- El programa `gnuplot` es indispensable para poder graficar los resultados creados por el programa. La versión más utilizada durante su desarrollo fue la 4,2 por lo que es la que se recomienda.

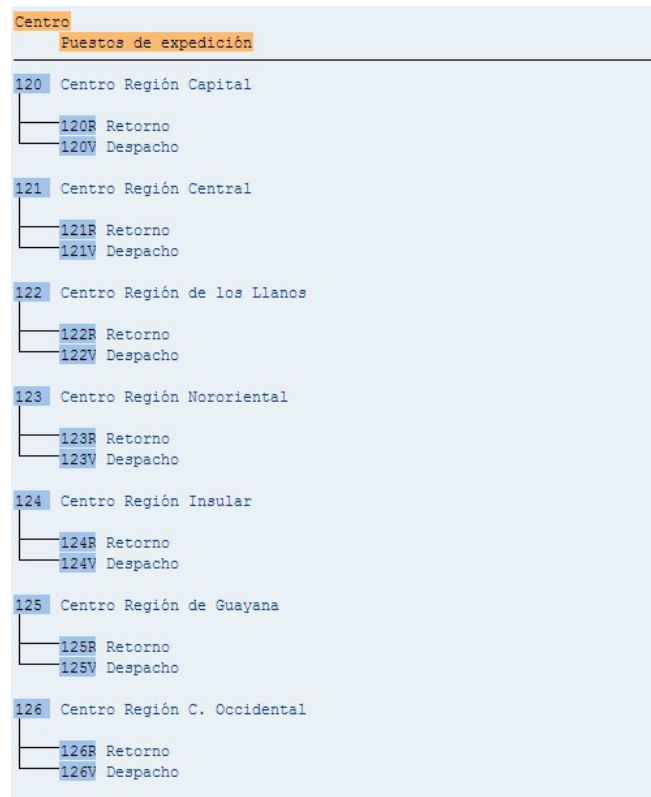


Figura A.6: Parte de la asignación de los Puestos de Expedición a los Centros de Distribución

- La graficación sólo se puede hacer sobre un sistema operativo bajo el estándar POSIX<sup>1</sup> con la herramienta gnuplot instalada. Esto se debe a una necesidad de la interfaz gnuplot en ANSI C que utiliza un “pipe” tipo POSIX para comunicarse directamente con el programa gnuplot instalado en la máquina. Sin embargo, dentro de las opciones del programa se puede pasar la información obtenida en la estimación a archivos de texto para su posterior análisis.
- La librería pthreads da las funciones necesarias para utilizar hilos de ejecución en el programa.

<sup>1</sup>”Portable Operating System Interface [for Unix]”son una familia de estándares de llamadas al sistema operativo definidos por la IEEE y especificados formalmente en el IEEE 1003. La gran mayoría de las distribuciones GNU/Linux siguen los estándares aunque no están oficialmente certificados.

## A.2. Ayuda de la línea de comando

La herramienta toma como parámetro principal un archivo `tcpdump` o un CSV de una serie de tiempo pseudo-aleatoria generada con la herramienta R. Aparte se puede escoger una velocidad de captura, una ventana, una ventana deslizante, filtrar paquetes, delimitar corridas, correr con varios hilos de ejecución, además de estimar el parámetro de Hurst con los tres métodos mencionados. Se puede también obtener el cambio del parámetro de Hurst en el tiempo o graficar la estimación de una ventana en particular. La ayuda de la línea de comando se muestra abajo:

```
Usage: ./d2Hgr -[f|g] file -[i|nrlvxmu] [OPTIONS]
```

Estimate and graph Hurst parameter calculations from a file.

- `-f file` Specifies a `tcpdump` dump file.
- `-g file` Specifies a CSV file with a simulated network traffic stream.  
This option cannot be used with the time based or `tcpdump` options since the stream is simulated and is static in it's definition.
- `-i` Prints only the protocol statistic information available from the `tcpdump` file.
- `-n` Tells the program if you wish to graph the packets per delta time graph. Affected by `-p`, `-o`, `-d`, `-c`, `-b` `-e`, `-y` flags.
- `-r` Tells the program to calculate the Hurst parameter changes over time by use of the R/S statistic. Affected by `-p`, `-o`, `-d`, `-c`, `-w`, `-s`, `-j`, `-t`, `-b`, `-e`, `-y` flags.

- l num     Tells the program to graph the Pox diagram that creates the Hurst data point for a given window number between 0 and Datapoints designated by num. Affected by -p, -o, -d, -c, -w, -s, -j, -b, -e, -y flags.
- v        Tells the program to calculate the Hurst parameter changes over time by using the Variance-time plot technique. Affected by -p, -o, -d, -w, -s, -j, -t, -b, -e, -y flags.
- x num     Tells the program to graph the Variance-time plot that estimates the Hurst parameter for a given window number designated by num. Affected by -p, -o, -d, -c, -w, -s, -j, -b, -e, -y flags.
- m        Tells the program to calculate the Hurst parameter changes over time by use of the Modified Allan Variance. Affected by -p, -o, -d, -c, -w, -s, -j, -t, -b, -e, -y flags.
- u num     Tells the program to graph the Modified Allan Variance that estimates the Hurst parameter for a given window number designated by num. Affected by -p, -o, -d, -c, -w, -s, -j, -t, -b, -e, -y flags.

#### OPTIONS:

- p proto   Specifies which protocol to be taken into account when doing calculations. Default is all.  
  
Implemented protocols are:  
  
ip tcp udp icmp sctp ftp ssh telnet smtp dns dhcp http pop3 ntp  
imap snmp ldap https smtps ldaps imaps pop3s nfs squid
- o dir     Output directory for results. Default is current directory.



- d Tells the program if the results should be placed in text files for later use. This option doesn't graph. Useful for using this program where gnuplot is not available.
- y Tells the program to graph and print the resulting data files.
- a Tells the program not to graph or create data files.
- c sec Specify a packet capture speed. This makes the program not look for one. Example: 0.01 = 0.01 seconds.
- w sec Window time size. This makes the program not look for one. Example: 60 = 60 seconds.
- s sec Slide time size. This makes the program not look for one. Example: 1 = 1 seconds.
- j num Tells the program to use log base num for all the calculations of blocks for the R/S statistic, Variance-time plot and Modified Allan Variance. Default is 2.
- t num Does Hurst calculation using threads. If 0 is placed, then the default number of threads (4) is used.
- b sec Tells the program from which second in the time data to begin the calculation. Default is 0s.
- e sec Tells the program how many seconds after the beginning point to include in the calculation. Default is the whole tcpdump time.
- k num Average Hurst parameter change with which to try to detect attacks.
- K num Standard deviation of Hurst parameter change with which to detect attacks.
- q num Average Hurst parameter value with which to try to detect attacks.
- Q num Standard deviation of Hurst parameter value with which to detect

```

        attacks.

-h      Print this help.

```

### A.3. Archivos del programa

El programa de C llamado **d2Hgr**, consiste de 18 archivos de código fuente que incluye 9 archivos de cabecera. Los archivos contienen la siguiente información:

- **config.h**: Cabecera de funciones para parsear las opciones de línea de comando.
- **config.c**: Implementación de las funciones que parsean las opciones de la línea de comando.
- **d2Hgr.h**: Archivo que contiene las definiciones de las funciones para leer los archivos producidos por **tcpdump**.
- **d2Hgr.c**: Implementación de las funciones para leer la información del archivo producido por **tcpdump** y poder obtener la información necesaria para su posterior análisis.
- **externvars.h**: Archivo que contiene todas las estructuras especiales para el uso del programa.
- **filter.h**: Archivo que contiene la definición de las funciones para construir la expresión de filtro de **libpcap** para usar durante la corrida del programa.
- **filter.c**: Archivo que contiene la implementación de todas las funciones de **filter.h**.
- **flows.h**: Cabecera de funciones para contabilizar los flujos de IPv4.
- **flows.c**: Archivo que contiene la definición de las funciones para contabilizar los flujos de IPv4.

- **gnuplot.i.h**: Cabecera para el archivo `gnuplot.i.c` que define las funciones de la interfaz `gnuplot` para que el programa pueda graficar los resultados. Esta es una versión modificada de la interfaz ANSI C de N. Devillard para `gnuplot`.
- **gnuplot.i.c**: Implementación de las funciones de N. Devillard para su interfaz de ANSI C con `gnuplot`.
- **graph.h**: Archivo que contiene la definición de las funciones para graficar los resultados.
- **graph.c**: Archivo que contiene la implementación de las funciones para graficar los resultados.
- **hurst.h**: Archivo que contiene las definiciones de los métodos para aproximar el parámetro de Hurst.
- **hurst.c**: Archivos que contiene las implementaciones de los métodos para aproximar el parámetro de Hurst y los métodos para extraer la información sobre los paquetes una vez leídos por las funciones de `d2Hgr.h`.
- **main.c**: El archivo que contiene el `main` del programa.
- **detect.h**: El archivo que contiene las definiciones de los métodos para la detección de ataques de denegación de servicio.
- **detect.c**: El archivo que contiene las implementaciones de los métodos para la detección de ataques de denegación de servicio.

## A.4. Creación de *Xtdata*

```

1 // Reset j for next cycle.
2 float j = 0.0;

```

```

3 // Counter to know in which delta_time section we are on the xtdata
4 // array.
5 int arrcount = 0;
6 // Temporal storage for the number of packets seen.
7 int numpktseen = 0;
8
9 // While the added time is less than the total time and we haven't
10 // run out of packets, run this cycle to see how many packets we see
11 // per delta_time section.
12 while (arrcount < datapoints && timecnt < packets &&
13        j <= time_int ) {
14
15     // Add the next timedata to j and pass to the next packet time
16     // diff.
17     j += timedata[timecnt++];
18
19     // We are seeing one new packet, so add it to our temporal
20     // storage.
21     numpktseen++;
22
23     if (j >= (arrcount+1)*(delta_time)) {
24         // If we've surpassed one of the delta_times, fill the
25         // information for the delta_time section.
26         xtdata[arrcount++] = numpktseen - 1;
27
28         // Boolean variables
29         short int passed = 0;
30         short int carry = 1;
31
32         // Since we're not sure if the packet jumps through too
33         // many sections, we check for it, filling each

```

```

34      // sectioned passed with 0, except the first one which has
35      // at least one.
36      while (arrcount < datapoints &&
37             j > (arrcount)*(delta_time)) {
38          if (passed == 0) {
39              xtdata[arrcount++] = 1;
40              passed = 1;
41              carry = 0;
42          } else {
43              xtdata[arrcount++] = 0;
44          }
45      }
46
47      // Finally we reset the temporal storage for the next cycle.
48      if (carry == 1) {
49          numpktseen = 1;
50      } else {
51          numpktseen = 0;
52      }
53  }
54  }

```

Código fuente A.1: Creación de *Xtdata*