**2012**

Manuel Samayoa

**Software Solutions**

|  |  |
| --- | --- |
| **Sergio Mancilla Escobar** | **2007-14244** |
| **Sonia Guamuch Aceituno** | **2006-11396** |
| **Pablo Daniel Tum Xitumul** | **2007-14469** |
| **Edgar Chivichon Chacón** | **2007-14774** |
| **Diego Josue Bedoya** | **2007-15269** |

01/01/2012

Sistema de Ventas de Software On-line (SIVSOF)

Arquitectura del Software

Versión 1.0

Sistema de Ventas de Software On-line (SIVSOF)

Arquitectura del Software

Versión 1.0

Historial de Revisiones

|  |  |  |  |
| --- | --- | --- | --- |
| **Fecha** | **Versión** | **Descripción** | **Autor** |
| 26/12/2011 | 0.0 | Versión inicial de arquitectura del software |  |
| 28/12/2011 | 1.0 | Versión preliminar de arquitectura del software |  |

Tabla de Contenidos

Arquitectura del Software 1

1. Introducción 1

1.1 Propósito 1

1.2 Alcance 1

1.3 Definiciones, Siglas y Abreviaciones 2

1.4 Referencias 2

1.5 Visión 2

2. Representación de la Arquitectura 2

3. Las Metas arquitectónicas y Limitaciones 3

4. La Vista de Casos de Uso 4

5. La Vista lógica 5

5.1 Visión 5

5.2 El desarrollo Significante de la Arquitectura de Paquetes 5

5.2.1 Capa de Presentación: 6

La capa de aplicación, muestra las pantallas o interfaces gráficas con las cuales el usuario interactuara o visualizara de la aplicación SIVSOF. 6

5.2.2 Capa de Lógica o Enlace: 9

La Capa Lógica o tambien conocida como de enlace, alberga los diagramas de Clases los cuales se utilizarán para realizar las operaciones solicitadas en el SIVSOF.

5.2.3 Capa de Web Services: 10

La Capa de Web Services muestra la estructura básica del servidor de aplicación, servidor de versiononamiento y de integración continua (IC) que usara la aplicación SIVSOF.

5.2.4 Capa de Datos: 11

Esta Capa de Datos, contiene todo lo referente a la Base de Datos, es decir, utilización de DBMS, diagrama Entidad / Relación.

5.3 Diagrama de Red 12

# Arquitectura del Software

# Introducción

La arquitectura del software forma parte del modelo de cómo se encuentra estructurado un Sistema especifico, sirviendo de comunicación entre las personas involucradas en el desarrollo y ayudando a realizar diversos análisis que orienten el proceso de toma de decisiones, con lo cual se logra tener un mejor entendimiento de la aplicación y por ende obtener una mejor calidad en la aplicación.

## Propósito

El documento de la Arquitectura del Software tiene como propósito principal el de proporcionar un medio o herramienta que sirva dentro del desarrollo y mantenimiento del software SIVSOF, adicionalmente provee una herramienta que represente de una manera precisa la aplicación y su implementación. Como herramienta a utilizar esta UML (Lenguaje Unificado de modelado), esta herramienta permite representar de manera semi-formal la estructura general del sistema mediante el uso de una gran gama de diagramas, además su representación gráfica puede ser usada para comunicarse con el usuario de una manera sencilla en la cual le sea más fácil su comprensión y aceptación.

## 

## Alcance

El alcance o limitaciones del presente documento de arquitectura de software es el de dar conocer la constitución del sistema, dicho en otras palabras, las vistas que forman parte del sistema, como por ejemplo, la vista de sistema, vista de componentes, vista de procesos, vista lógica y casos de uso.

En lo concerniente a la vista lógica se definirá la funcionalidad, diagrama de clases, diagrama de secuencias y paquetes. En la vista de componentes se describirá el manejo de Software, la reutilización, diagrama de cómo estará constituido el sistema físico, en otras palabras lo concerniente al hardware, en cuanto a la vista de procesos, se dará a conocer los procesos que participan en la implementación del sistema y por ultimo esta la vista de producción, que incluye el deployment que es la descripción del lugar en donde se publicaran los componentes.

## Definiciones, Siglas y Abreviaciones

A continuación se definirán los términos y abreviaturas más comunes que se utilizarán en este documento:

* SO: Estas siglas hacen referencia a Sistema Operativo, a partir de ahora las usaremos para referirnos a un SO basado en Windows, Unix, Solaris, etc.
* SW: Es la abreviatura utilizada para referirnos al software.
* HW: Es la abreviatura utilizada para referirnos al hardware.
* IC: Será utilizado para referirnos a todo lo concerniente con la Integración Continua.
* IDE: Entorno de Desarrollo Integrado
* CU: Casos de Uso
* UML: Lenguaje de Modelado Unificado

## Referencias

Se utilizará el servicio de versionamiento proporcionado por el IDE Netbeans al igual que la aplicación Tortoise que también es utilizada para el versionamiento.

## Visión

Con la redacción de este documento se pretende describir la forma en que está formada la Arquitectura de Software del SIVSOF. Se especifica la representación arquitectónica del sw. La arquitectura del SW se verá representada mediante, las metas arquitectónicas, casos de uso, vista lógica, todo desde el punto de vista arquitectónico de los paquetes de plan, así como las vistas y elementos que lo componen.

A continuación se encontrarán dos secciones principales en las que se encontrará de forma más detalla y organizada la información acerca de la Arquitectura del Software del SIVSOF.

# Representación de la Arquitectura

El presente documento representa la arquitectura como un grupo de vistas; como la vista del caso, vista lógica, vista de proceso y vista de despliegue. No hay ninguna vista de aplicación separada descrita en el presente documento. Estas son las vistas en el Lenguaje Unificado de Modelado (UML) creadas en Enterprise Architecture y StarUML.

# Las Metas arquitectónicas y Limitaciones

En esta sección describe los requisitos del software y objetivos que tienen algún impacto significante en la arquitectura, de los cuales los principales que se pueden identificar son:

## Seguridad de Sesiones

Este requisito ser refiere a la seguridad que debe existir en el software para proteger la información al momento de intentos de robo de usuarios o contraseñas, por ejemplo, en el uso de sql injection la aplicación de sw debería ser capaz de enviar mensajes de error y no permitir la obtención de los datos solicitados por el atacante, de igual manera informar al administrador de la base de datos que el sistema ha sido comprometido y desplegar un informe detallando la situación.

## Portabilidad

El sistema debido a que es una aplicación web deberá poder ser corrido en cualquier navegador web de cualquier SO.

## Performance

La aplicación SIVSOF correrá desde un servidor y debido a las grandes demandas de solicitudes o visitas a la aplicación esta deberá de consumir la menor cantidad de recursos del servidor o computadora que sirva como servidor de la aplicación.

# La Vista de Casos de Uso

## Visión

La vista de Casos de Uso muestra los actores, acciones y relaciones que forman en conjunto la funcionalidad que será implementada en la plataforma de ejecución de la aplicación es decir el uso de JSP en Netbeans.

## cuCU del sistema.

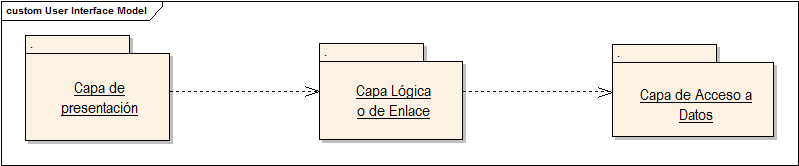
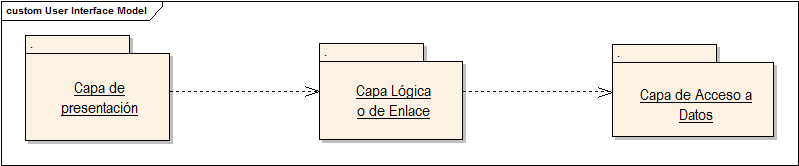
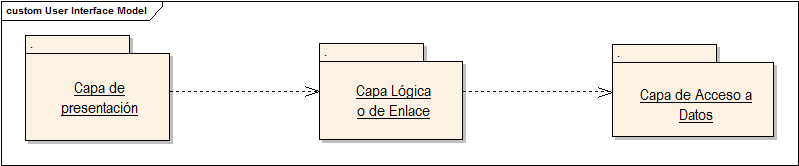
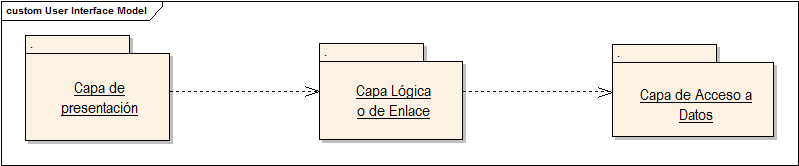
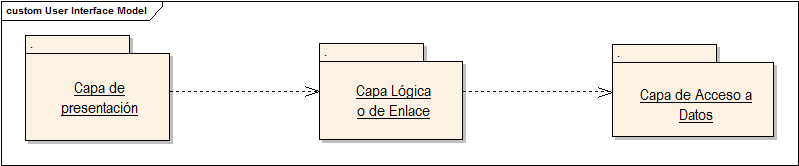
# 

# La Vista lógica

## Visión

La vista lógica se encarga de listar y mostrar los componentes del diseño y las relaciones de una forma independiente de todos los detalles y dependencias técnicas y del como la funcionalidad será implementada en la plataforma de ejecución, es decir con JSP en Netbeans como IDE. Se detallara y describirán las interfaces de usuario, diagrama de clases junto con su interacción y el diagrama relacional en cuanto a la BD.

## El desarrollo Significante de la Arquitectura de Paquetes

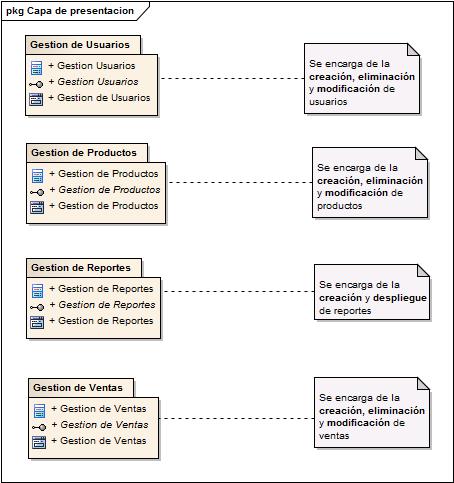
*********Diagrama de Capas de SIVSOF***

### Capa de Presentación:

### Esta capa, está formada por la o las interfaces del usuario las cuales sirven como medio de interacción entre la aplicación y el usuario. Esta interacción es el primer paso para que se lleve a cabo una actividad (compra de un producto, creación de usuarios, etc.). La interfaz de usuario será implementada en un ambiente web mediante JSP y el IDE Netbeans.

***Interacción:*** La interacción con la capa lógica será por medio de llamadas a métodos implementados en JSP e hibernate, que servirán para ejecutar acciones definidas por el usuario.

*Packages:*



Interfaces:

*Gestión Usuarios:*



*Gestión Productos:*



*Gestión Ventas:*



*Gestión Reportes:*



### Capa de Lógica o Enlace:

### Esta Capa Lógica o Enlace, contiene el diagrama de clases que se utilizara para poder realizar las operaciones requeridas en el sistema SIVSOF.



### Capa de Web Services:

La Capa de Web Services muestra la estructura básica del servidor de aplicación, servidor de versiononamiento y de integración continua (IC) que usara la aplicación SIVSOF.

Escenario IC





Commit

Generación

Mecanismo de

Retroalimentación





Commit

Poll

Servidor IC



Commit

Build

Script

Repositorio

SVN

### Capa de Datos:

### Esta Capa de Datos, contiene todo lo referente a la Base de Datos, es decir, utilización de DBMS, diagrama Entidad / Relación.

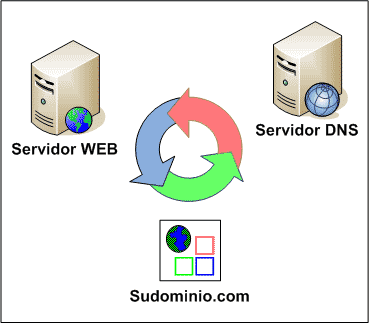
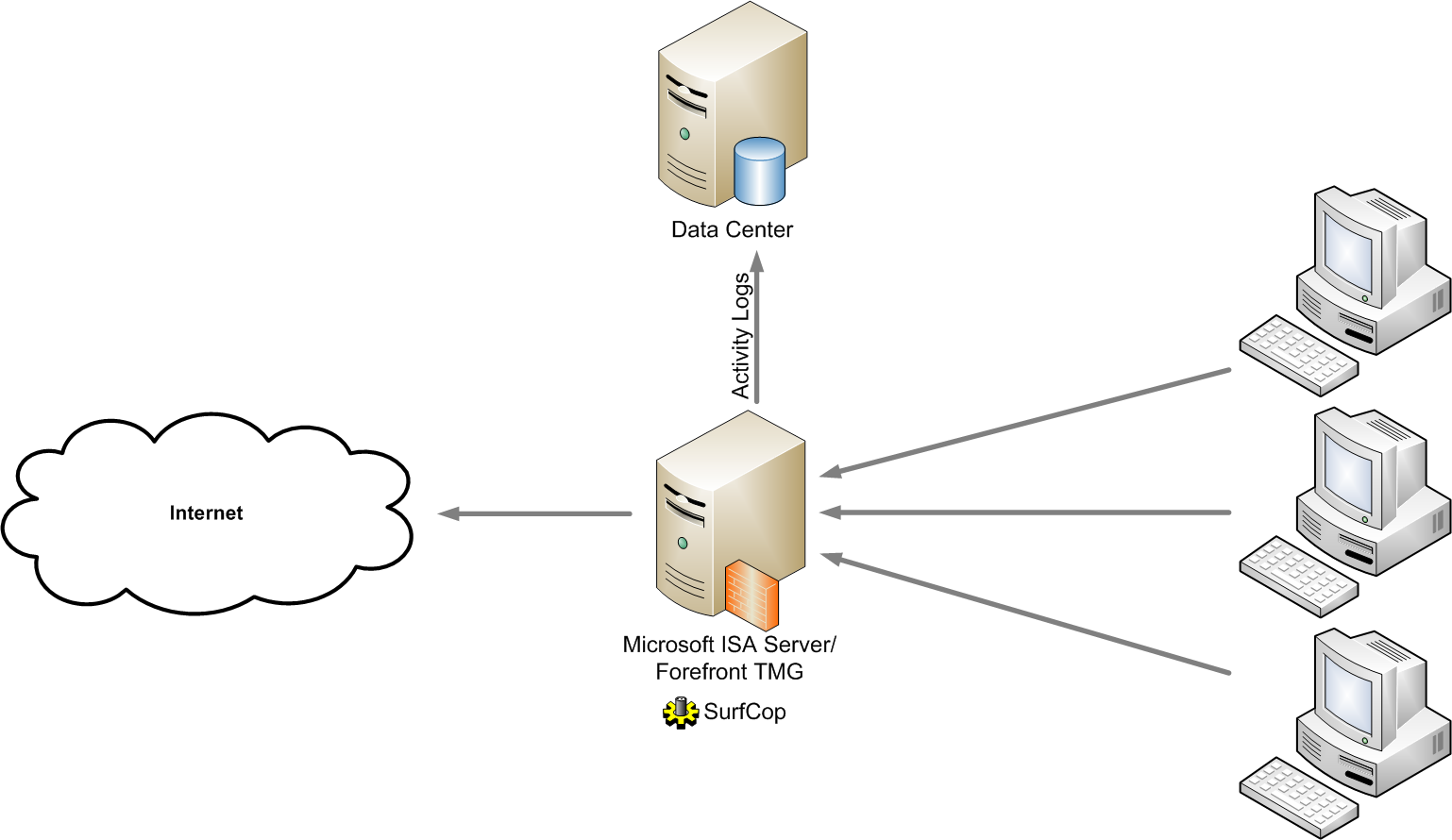
**DBMS:** se utilizar PostgreSQL 9.1

***Modelo Entidad Relación***



## Diagrama de Red

Cliente



PostgreSQL





Explorador web

Switch



Firewall

Router