Proyecto Certificador de Desarrollo de SW 1

Tema Nº12:

Implementación - Planear Integración - Implementar Componentes - Integrar el Sistema

Indicador de logro Nº12:Desarrolla el Modelo de Implementación de un sistema de acuerdo a la metodología RUP - Lenguaje UML

**TEMA 01 Teoría de los**

Imagen que contiene Icono

Descripción generada automáticamente

**TEMA Nº12:**

IMPLEMENTACIÓN - PLANEAR INTEGRACIÓN

Objeto De La Experiencia

* Conoce los elementos del Modelo de Requerimientos utilizando metodología RUP y lenguaje unificado de modelado UML
* Elabora el Diagrama de Implementación de Componentes y Despliegue

Marco Teórico

Representación UML: Modelo, estereotipado como <<modelo de implementación>>.

El **modelo de implementación** puede tener las siguientes **propiedades**:

* Introducción: Una descripción textual que sirve como breve introducción al modelo.
* Subsistemas de implementación: Los subsistemas del modelo, que representan una jerarquía.
* Elementos de implementación: Los elementos del modelo, propiedad de los subsistemas.
* Relaciones: Las relaciones del modelo, propiedad de los subsistemas de implementación.
* Diagramas: Los diagramas del modelo, propiedad de los subsistemas de implementación.
* Vista de implementación: La vista de implementación del modelo, que es una vista de la arquitectura que muestra las capas y los subsistemas de implementación.

Un modelo de implementación es opcional.

Si elige crear un modelo de implementación, las decisiones de personalización claves son cómo relacionar el modelo de implementación y el modelo de diseño y qué elementos de implementación son lo suficientemente importantes para el modelo. Puede obtener asesoramiento de cómo tomar estas decisiones en el apartado [Directriz de producto de trabajo: Modelo de implementación](https://cgrw01.cgr.go.cr/rup/RUP.es/LargeProjects/core.base_rup/guidances/guidelines/implementation_model_5A4A2E5F.html). Consulte también el apartado [Técnica: Correlación de diseño a código](https://cgrw01.cgr.go.cr/rup/RUP.es/LargeProjects/core.base_rup/guidances/concepts/mapping_from_design_to_code_D1454680.html).

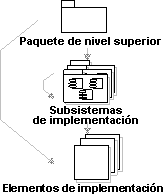
El modelo de implementación modela, principalmente, los subsistemas de implementación, que incluyen dependencias y otra información de gestión. En esta directriz se describe su objetivo y estructura.

En el entorno de programación, una implementación está compuesta por elementos de implementación, que incluyen archivos de código fuente, archivos binarios, y archivos de datos, organizados en directorios. Además de estos elementos de nivel bajo, a menudo aparece la necesidad de crear unidades de nivel más elevado de gestión, los subsistemas de implementación, que agrupan elementos de implementación y otros subsistemas de implementación.

El modelo de implementación modela, principalmente, los subsistemas de implementación, que incluyen dependencias y otra información de gestión. También puede modelar elementos clave de un subsistema de implementación, como archivos desplegables, o estructuras de directorio.

La notación en el modelo de implementación. Las flechas muestran posibles propiedades.

Existe, opcionalmente, un paquete que sirve como nodo de nivel más alto (raíz) en el modelo de implementación. Los paquetes, estereotipados como <<subsistemas de implementación>> agrupan los elementos de implementación (archivos y directorios) y otros subsistemas de implementación.



El modelo de implementación se centra en la preocupación de la organización física del software en términos de subsistemas de implementación y elementos de implementación. De forma opcional, puede crear un único modelo que solucione la implementación física y el diseño lógico en un único modelo. Esto es común en un enfoque de ingeniería directa e inversa que sincroniza los archivos de código fuente con un modelo de implementación/diseño combinado.

La organización de los subsistemas de implementación puede ser más o menos próxima al modelo de diseño, dependiendo de cómo decida correlacionar entre estos dos modelos. Esta es una decisión de proceso que debe capturarse en las directrices de diseño específicas al proyecto. Cuando la correlación es exacta, es decir, cada subsistema de implementación también es un subsistema de diseño, puede crear diagramas que se centran en un único subsistema de diseño, resumiendo el diseño y su implementación.

1. Planeación de la Integración

* 1. Planificar la Integración del Sistema

En esta tarea se describe cómo planificar la integración del sistema.

Disciplinas: Implementación

La finalidad de la implementación es:

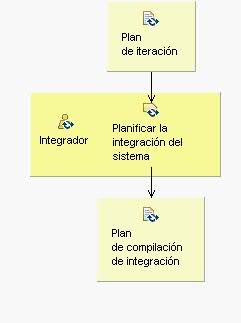
* Definir la organización del código, en términos de los subsistemas de implementación organizados en capas
* Implementar los elementos de diseño en términos de los elementos de implementación (archivos de origen, binarios, programas ejecutables y otros)
* Probar y desarrollar componentes como unidades
* Integrar los resultados producidos por los implementadores individuales (o equipos) en un sistema ejecutable.

La disciplina de implementación limita su ámbito a la forma en que las clases individuales deben pasar por la prueba de unidad. La prueba del sistema y la de integración se describen en la disciplina de prueba.

La implementación está relacionada con otras disciplinas:

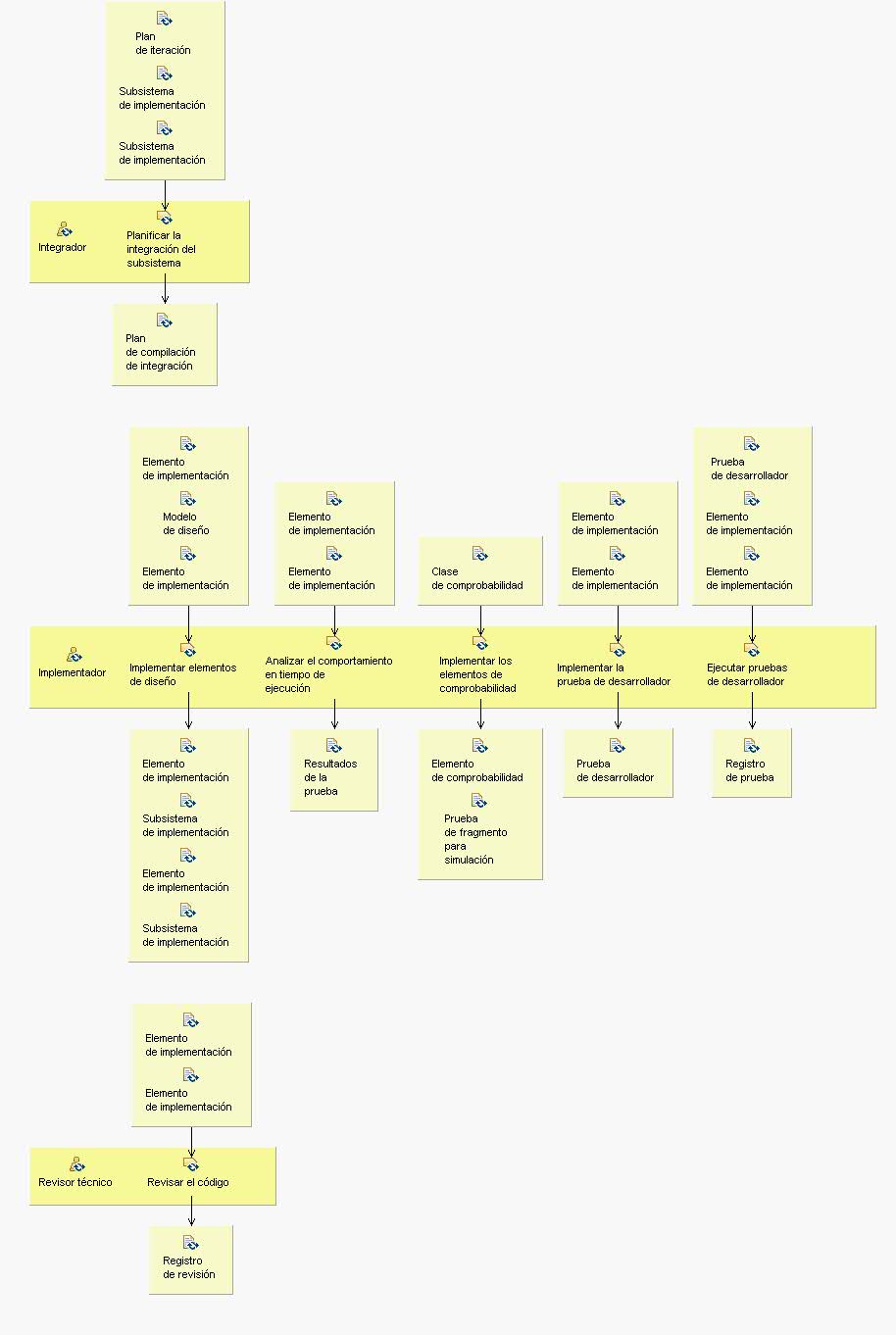
* La disciplina de [**requisitos**](https://cgrw01.cgr.go.cr/rup/RUP.es/LargeProjects/core.base_rup/disciplines/rup_requirements_discipline_8EEB5209.html) describe cómo capturar los requisitos que debe cumplir la implementación en un modelo de guión de uso.
* La disciplina de [**análisis y diseño**](https://cgrw01.cgr.go.cr/rup/RUP.es/LargeProjects/core.base_rup/disciplines/rup_analysis_design_discipline_29760231.html) describe cómo desarrollar un modelo de diseño. El modelo de diseño representa la intención de la implementación y es la entrada principal de la disciplina de implementación.
* La disciplina de [**prueba**](https://cgrw01.cgr.go.cr/rup/RUP.es/LargeProjects/core.base_rup/disciplines/rup_test_discipline_9DFAFB2F.html) describe cómo realizar el test de integración para cada compilación durante la integración del sistema. También describe cómo realizar las pruebas del sistema para verificar que todos los requisitos se han cumplido, así como la forma en que se detectan y remiten los defectos.
* La disciplina de [**entorno**](https://cgrw01.cgr.go.cr/rup/RUP.es/LargeProjects/core.base_rup/disciplines/rup_environment_discipline_41478973.html) describe cómo desarrollar y mantener artefactos de soporte que se utilicen durante la implementación, como la descripción del proceso, las directrices de diseño y las de programación.
* La disciplina de [**despliegue**](https://cgrw01.cgr.go.cr/rup/RUP.es/LargeProjects/core.base_rup/disciplines/rup_deployment_discipline_A1C86C42.html) describe cómo utilizar el modelo de implementación para producir y entregar el código al cliente final.
* La disciplina de [**gestión de proyectos**](https://cgrw01.cgr.go.cr/rup/RUP.es/LargeProjects/core.base_rup/disciplines/rup_project_management_discipline_F303DA84.html) describe la mejor forma de planificar el proyecto. Algunos aspectos importantes del proceso de planificación son el plan de iteración, la gestión de cambios y los sistemas de seguimiento de defectos.

Flujo de Trabajo

****

1. Implementación de los Componentes
   1. Planear Integración de Componentes

Esta actividad completa una parte de la implementación de forma que se puede entregar para la integración.

****Flujo de Trabajo

* 1. Implementar Componentes

El diagrama de componentes proporciona una visión física de la construcción del sistema de información. Muestra la organización de los componentes software, sus interfaces y las dependencias entre ellos.

Estos diagramas pueden incluir paquetes que permiten organizar la construcción del sistema de información en subsistemas y que recogen aspectos prácticos relacionados con la secuencia de compilación entre componentes, la agrupación de elementos en librerías, etc.

.

Componentes

* Un componente representa una pieza del software reutilizable. Que conforma un conjunto de interfaces y proporciona su implementación.
* Modela artefactos tales como ejecutables, bibliotecas, tablas, ficheros, documentos,
* Representa el empaquetamiento físico de elementos lógicos tales como clases, interfaces,
* Residirán en los nodos del sistema



Tipos de Componentes

* Despliegue

Necesarios y suficientes para formar un sistema ejecutable: librerías Dinámicas (dll), ejecutables (exe, jar, ear)

* Productos del trabajo

Permanecen al final del proceso de desarrollo: archivos código fuentes, ficheros de datos.

Con ellos, se crean los componentes de despliegue

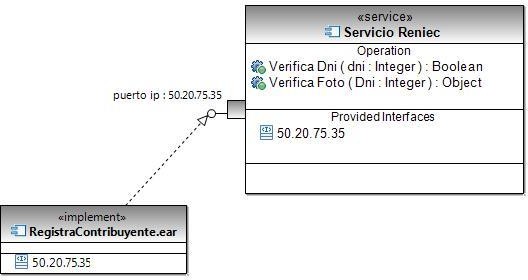
* De ejecución

Se crean durante la ejecución: objeto COM, instanciado a partir de una dll.

Servicios web, al momento de instalar un ear.

Diagrama De Componentes

Un diagrama de componentes representa cómo un sistema de software es dividido en componentes y muestra las dependencias entre estos componentes. Los componentes físicos incluyen archivos, cabeceras, bibliotecas compartidas, módulos, ejecutables, o paquetes. Los diagramas de Componentes prevalecen en el campo de la arquitectura de software, pero pueden ser usados para modelar y documentar cualquier arquitectura de sistema.



Nodo

Un nodo es un elemento físico que existe en tiempo de ejecución y representa un recurso computacional que puede tener memoria y capacidad de procesamiento.

* En redes de computadoras cada una de las máquinas es un nodo, y si la red es Internet, cada servidor

constituye también un nodo.

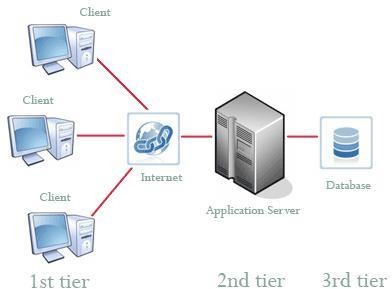
* Los componentes se ejecutan en nodos.
* Los nodos representan el despliegue físico de los componentes.



Servidor de Aplicaciones

Es un recurso computacional dedicada a la ejecución eficaz de los procedimientos (programas, rutinas, secuencias de comandos) para apoyar la construcción de aplicaciones

Se denomina servidor de aplicaciones a un servidor en una red de computadores que ejecuta ciertas aplicaciones



Cluster

El término clúster se aplica a los conjuntos o conglomerados de computadoras construidos mediante la utilización de componentes de hardware comunes y que se comportan como si fuesen una única computadora.

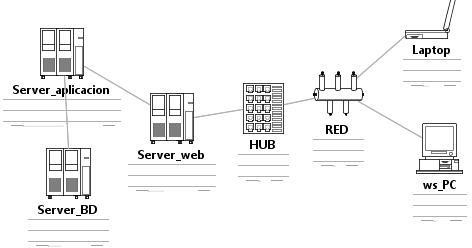
De un clúster se espera que presente combinaciones de los siguientes servicios:

* Alto rendimiento: Son clusters en los cuales se ejecutan tareas que requieren de gran capacidad computacional. El llevar a cabo estas tareas puede comprometer los recursos del clúster por largos periodos de tiempo.
* Alta eficiencia: Son clusters cuyo objetivo de diseño es el ejecutar la mayor cantidad de tareas en el menor tiempo posible.
* Alta disponibilidad: Son clusters cuyo objetivo de diseño es el de proveer disponibilidad y confiabilidad. Estos clusters tratan de brindar la máxima disponibilidad de los servicios que ofrecen. La confiabilidad se provee mediante software que detecta fallos y permite recuperarse frente a los mismos, mientras que en hardware se evita tener un único punto de fallos

****

Diagrama de Despliegue

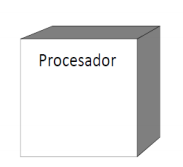
* Muestra la configuración de los nodos que participan en la ejecución y de los componentes que residen en los nodos.
* Incluye nodos y arcos que representan conexiones físicas entre nodos.
* Modelado de sistemas empotrados, sistemas cliente-servidor, sistemas distribuidos, servidores de aplicaciones



Tipos de Nodos

Procesador

Nodo con capacidad de procesamiento por lo que puede ejecutar un componente



Dispositivo

Nodo sin capacidad de procesamiento. Por lo general representa una máquina para la implementación del sistema, tal como: PC´, impresora, router, switch, hub, laptop, dispositivo móvil, etc.

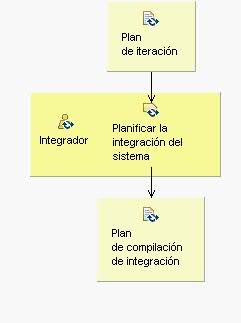


1. Integración del Sistema

* 1. **Integrar los módulos del Sistema**

En esta tarea se describe cómo planificar la integración de los módulos del sistema.

Flujo de Trabajo

****

1. Caso práctico Nro. 3

La empresa INFOTEL BUSINESS, es una empresa que se dedica a la comercialización y distribución de productos informáticos.

Tiene 15 años en el mercado informático y sus procesos principales son Gerencia, Ventas, Compras y Almacén. Tiene una cartera de clientes muy fidelizada y sus proveedores les proporcionan los productos informáticos de las mejores marcas y modelos con tecnología moderna.

Proceso De Ventas

Un Cliente solicita un pedido de venta a un Vendedor. El Vendedor antes de Generar el Pedido de Venta, Consulta el Stock de Productos y el Estado del Cliente.

Si todo está correcto, el Vendedor Genera el Pedido de Venta. Luego el Cajero Consulta el Pedido de Venta para Generar el Documento de Venta previa Consulta de la Forma de Pago y del Tipo de Documento de Venta.

El Tipo de Documento de Venta puede ser Factura de Venta o Boleta de Venta (Ticket electrónico). La Forma de Pago, puede ser Contado o Crédito. Después el Cajero Genera el Documento de Venta. El Auxiliar de Almacén Consulta el Documento de Venta y Genera la Guía de Salida, para entregar los productos y los documentos de venta al Cliente.

Proceso De Compras

El Asistente de Compras generar los pedidos de compra, previa consulta del stock de productos y del proveedor. Luego envía el pedido de compra a varios proveedores para solicitarles una cotización, después el Asistente de Compra registra las cotizaciones aceptadas por el Jefe de Compras.

El Jefe de Compras verifica la cotización y genera la orden de compra para enviarla al proveedor seleccionado, indicando la forma de pago. El proveedor genera una factura de compra y guía de entrada de productos para enviarla al Jefe de Compras quien registra la factura de compra previa consulta de la guía de entrada de productos. Luego el asistente de almacén, registra la guía de entrada y actualiza el stock de productos

Proceso De Almacén

El Asistente de Almacén genera el pedido de productos faltantes previa consulta del stock de productos y del proveedor.

El Jefe de Almacén registra las guías de entrada, las guías de salida, actualiza el stock de productos y genera el inventario que puede ser diario, semanal, mensual o anual.

Se pide elaborar el Modelo de Implementación, Solución de Escritorio en C#, Motor de Base de Datos MS SQL SERVER. Utilizar IBM Rational Software Architect Designer V9.6

4.1 Organización del Diagrama de Implementación

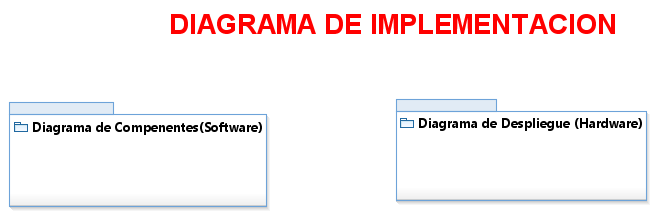
4.2 Diagrama de Componentes (Ms Visula Studio .Net, C#)

5.3 Diagrama de Despliegue

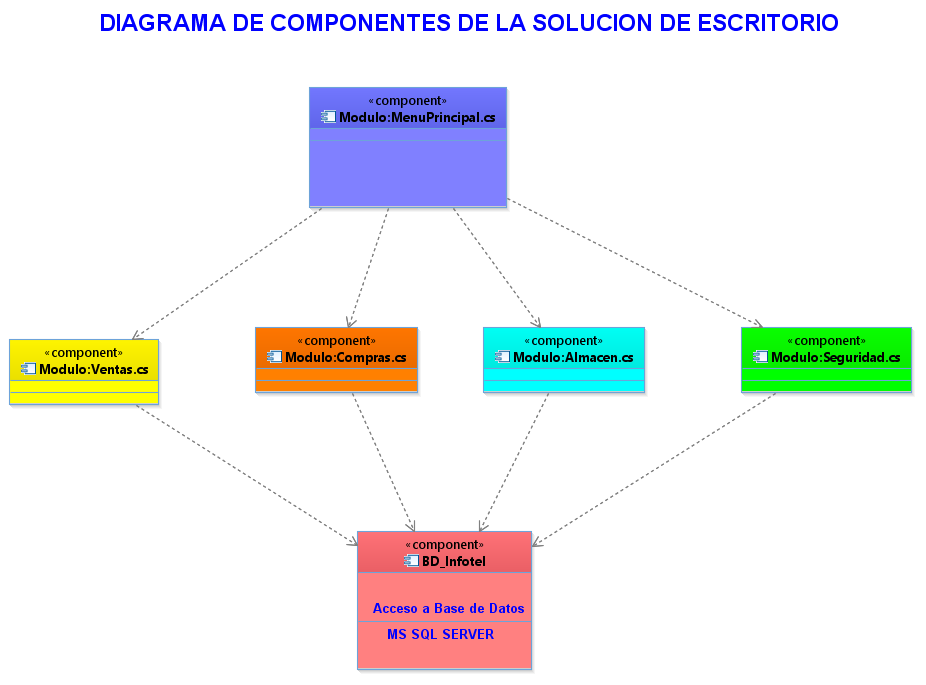
Solución del Casos Nro. 3

Modelo De Implementación

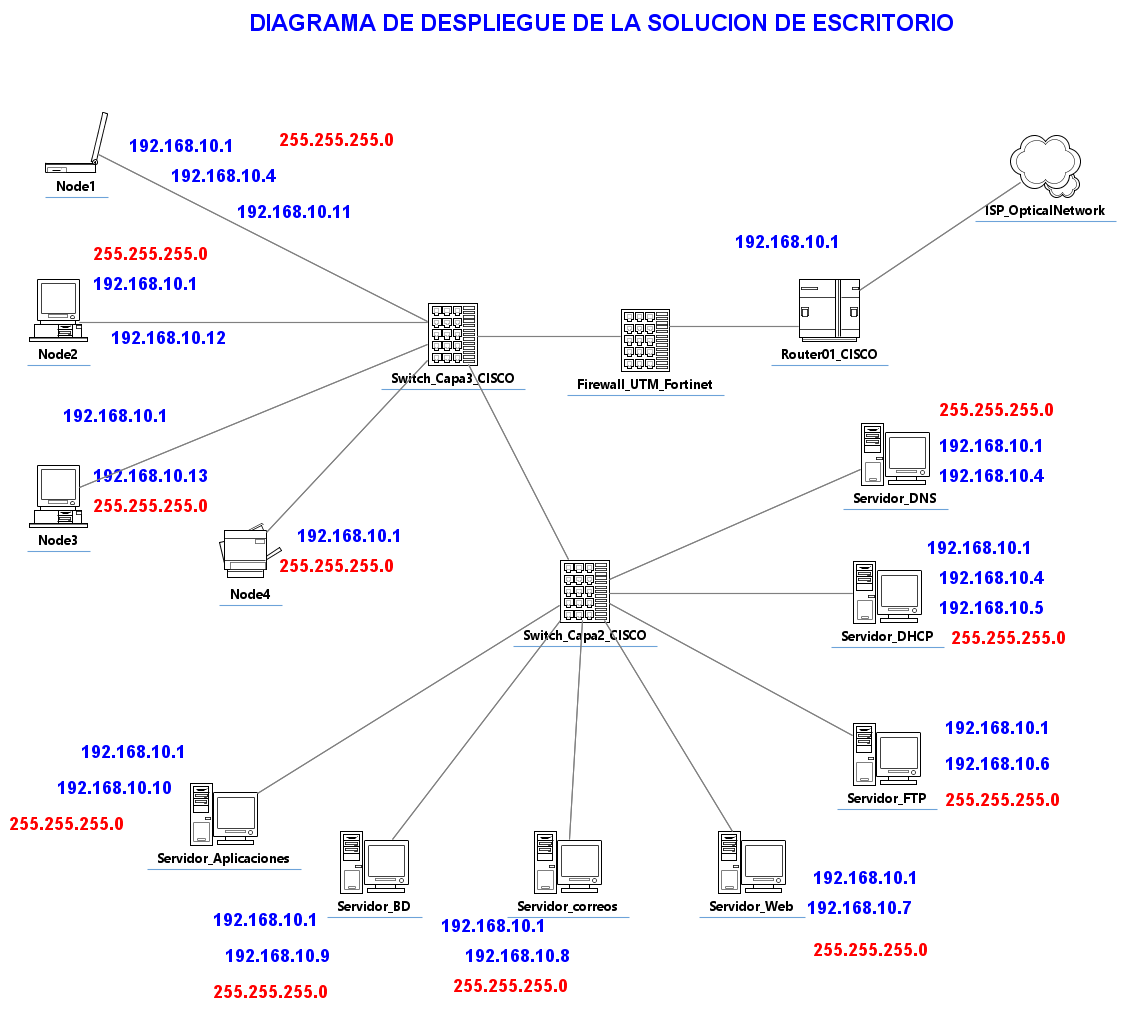
4.1 Organización del Diagrama de Implementación

****

4.2 Diagrama de Componentes

****

5.3 Diagrama de Despliegue

****

**Actividad:**

Ingresa a la plataforma virtual, luego desarrolla la siguiente actividad propuesta:

1. **CUESTIONARIO TÉCNICO**
2. Tomando como ejemplo el Caso Nro. 03, de la presente guía Modelo de Implementación, elabore la Organización del Diagrama de Implementación de su proyecto en grupo a desarrollar. Deberá utilizar metodología RUP y además IBM Rational Software Architect Designer V9.6.
3. Identifique los componentes de hardware y software del Sistema en base a su proyecto en grupo a desarrollar.
4. Elaborar el Modelo de Implementación diagrama de Componentes y Despliegue de su proyecto en grupo a desarrollar. Deberá utilizar metodología RUP y además IBM Rational Software Architect Designer V9.6.
5. **CONCLUSIONES DE LA EXPERIENCIA**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_