

Universidad de las Ciencias Informáticas
Facultad 10



**“Pautas para el diseño de la Vista de Infraestructura de una
Arquitectura de Software de Gestión.”**

Trabajo de Diploma para optar por el título de
Ingeniero en Ciencias Informáticas.

Autor:

Adonys Betancourt León

Tutor:

Ing. Yusnier Matos Arias

Ciudad de la Habana, Julio de 2010

DECLARACIÓN DE AUTORÍA

Declaro que soy el único autor de este trabajo y autorizo al Centro de Informatización de Gestión de Entidades de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

Adonys Betancourt León

Autor

Ing. Yusnier Matos Arias

Tutor

DEDICATORIA

Dedico esta tesis a mi abuela Titica, que más que mi abuela, ha sido mi madre, mi padre, mi amiga y es la persona que más quiero en este mundo.

RESUMEN

Una infraestructura robusta y confiable que soporte el desarrollo de un proyecto de software es vital para el éxito del mismo. La misma debe estar constituida por las tecnologías adecuadas, así como por herramientas y servicios que respondan a las necesidades del proyecto. El mantenimiento de la Infraestructura Tecnológica (IT) desde el comienzo del proceso de desarrollo de software y hasta el despliegue del mismo es fundamental, dado que determina el comportamiento de las aplicaciones ante los usuarios finales en cualquiera de los escenarios que se definan.

El proyecto ERP-Cuba que se desarrolla en la Universidad de las Ciencias Informáticas (UCI) carece de una guía que permita realizar el trabajo del Arquitecto de Infraestructura de una forma organizada. Esta investigación está encaminada a crear un conjunto de pautas que permitan organizar y diseñar la Vista de Infraestructura de la arquitectura de dicho proyecto. Con el objetivo de crear esta guía se realiza un estudio de los diferentes puntos de vista presentados por varias empresas que poseen cierto grado de madurez en esta rama de la arquitectura. Tomando como punto de partida las actividades propuestas por estas empresas, además de las características y necesidades del proyecto ERP-Cuba, se definen las pautas para diseñar una Arquitectura de Infraestructura (AI) que cumpla con las exigencias de un proyecto de software de gestión.

PALABRAS CLAVES: arquitectura de infraestructura, despliegue, entorno de desarrollo, plataforma de servicios.

ÍNDICE

INTRODUCCIÓN	1
CAPÍTULO 1 FUNDAMENTACIÓN TEÓRICA	5
1.1 Arquitectura	5
1.1.1 Arquitectura de software.....	5
1.1.2 Arquitecto de software.....	6
1.1.3 Arquitecto de Infraestructura	7
1.2 Arquitectura de Infraestructura desde diferentes puntos de vista.....	8
1.2.1 Microsoft.....	8
1.2.1.1 Capacidades de la Optimización de la Infraestructura Principal.....	9
1.2.1.2 Niveles del modelo de optimización de la infraestructura principal	10
1.2.1.3 Enfoques por etapas.....	11
1.2.1.4 Actividades propuestas por Microsoft	12
1.2.2 The Open Group Architecture Framework	13
1.1.1.1 Pasos definidos por TOGAF para desarrollar una Arquitectura Tecnológica.	13
1.2.3 Instituto de Ingeniería de Software.....	16
1.3 Análisis de Propuestas Estudiadas.....	17
CAPÍTULO 2 PAUTAS PARA EL DESARROLLO DE LA ARQUITECTURA DE INFRAESTRUCTURA....	24
2.1 Áreas de trabajo de la Arquitectura de Infraestructura	24
2.1.1 Entorno de Desarrollo.....	24
2.1.2 Plataforma de Servicios.....	34
2.1.3 Despliegue.....	39
CAPÍTULO 3 VISTA DE INFRAESTRUCTURA DEL PROYECTO ERP-CUBA.....	44
3.1 Entorno de desarrollo	44
3.1.1 Análisis de los requerimientos tecnológicos.....	44
3.1.2 Selección de Sistema Operativo	45
3.1.3 Definición de los lenguajes de programación.....	46
3.1.4 Herramientas y tecnologías.....	46
3.1.5 Definición de políticas de trabajo y listas de verificación.....	50
3.2 Plataforma de Servicios	51

3.2.1	Módulos de la plataforma de servicios	52
3.2.1.1	Módulo de producción	52
3.2.1.2	Módulo de gestión	53
3.2.1.3	Módulo de investigación y superación	55
3.2.1.4	Módulo de monitoreo y control.....	55
3.2.2	Diseño Estructural de Tecnología	57
3.2.3	Políticas de trabajo.....	57
3.2.3.1	Sistema de salvas.....	58
3.2.3.2	Monitoreo y Control.	59
3.2.3.3	Lista de verificación	59
3.2.3.4	Scripts.....	60
3.3	Despliegue.....	61
3.3.1	Herramientas.....	61
3.3.2	Escenarios de despliegue	62
3.3.2.1	Pruebas de carga y estrés.....	62
3.3.2.2	Diseño lógico del centro de datos.....	63
3.3.2.3	Definición de escenarios.....	63
3.4	Validación de la propuesta.	64
	CONCLUSIONES GENERALES	67
	RECOMENDACIONES.....	68
	REFERENCIAS BIBLIOGRÁFICAS	69
	BIBLIOGRAFÍA.....	73

INTRODUCCIÓN

La AI es un área polémica dentro de las disciplinas arquitectónicas propuestas por los estándares actuales. Las actividades que se realizan en esta área dependen en gran medida del tipo de organización así como del tamaño y las condiciones tecnológicas de esta.

Definir una infraestructura estable para el proyecto es fundamental, dicha infraestructura debe tener en cuenta las necesidades tecnológicas y de información así como las herramientas y el posible crecimiento del proyecto; se debe brindar soporte desde el comienzo hasta el despliegue del mismo, incluso, después de este, debe garantizar los servicios para un soporte aceptable de las aplicaciones. En los proyectos de desarrollo de software de gestión, donde se hace indispensable el trabajo colaborativo, es recomendable que se garantice un conjunto de servicios que aporten valor agregado a la infraestructura: servicios de gestión de proyectos, sistemas de gestión documental, servidores de ficheros, repositorios de distribuciones y servidores de actualizaciones, además de otros servicios para promover la investigación y la superación de los miembros del proyecto. Es necesario, para garantizar la disponibilidad de estos servicios y aplicaciones, la implantación de sistemas de monitoreo que permitan obtener información actualizada y puedan contribuir con la toma de decisiones en cuanto a necesidades de escalabilidad y otras.

Según Microsoft la AI es un área relativamente nueva de la Arquitectura. Antiguamente se dedicaba toda la atención de la Arquitectura de IT hacia la Arquitectura de Información y Aplicación; pero actualmente varios factores han impulsado una creciente necesidad de esta nueva rama. Con la aplicación de esta disciplina no sólo se benefician las infraestructuras de las organizaciones, sino que la Arquitectura de IT logra una maduración en todo su conjunto; pues aún se encuentra en su infancia y falta mucho trabajo para estimular la creación de métodos, modelos y herramientas en esta disciplina. [10]

IBM plantea que esta es una Arquitectura diseñada para proporcionar todos los servicios necesarios detallados en la Arquitectura de Negocio e Información de una manera totalmente estable y segura. [20]

En el marco de este documento, se define Infraestructura como la Plataforma Tecnológica sobre la que funciona el proyecto de software, la constituyen las tecnologías hardware y software, así como las

herramientas, lenguajes de programación, aplicaciones y servicios usados durante todo el proceso de desarrollo, y se centra en el buen funcionamiento de dichos servicios y aplicaciones.

Partiendo de lo anterior, se puede definir AI como la disciplina arquitectónica que se encarga de gestionar todas esas tecnologías y herramientas de la manera más eficiente, garantizando niveles aceptables de escalabilidad, alta disponibilidad, estabilidad, y cierto grado de seguridad además de aportar valor agregado a las aplicaciones y servicios.

Generalmente, en los proyectos de software de gestión este tipo de actividades se lleva a cabo de manera intuitiva y las realizan diferentes personas dentro del proyecto, teniendo en cuenta las experiencias de cada uno y las necesidades que vayan surgiendo. Esto conlleva a que la Infraestructura no se gestione de forma organizada, evidenciándose la necesidad de contar con guías de actividades que orienten el trabajo para garantizar el correcto diseño y mantenimiento de la IT del proyecto.

Lo antes planteado se pone de manifiesto en el proyecto ERP-Cuba, donde no se gestiona la Infraestructura, no se tiene control sobre todas las herramientas usadas en el desarrollo, a ni existen estándares sobre las configuraciones básicas que estas deben tener. Se cuenta solamente con servicios elementales para el desarrollo: servidor de datos, servidor web y sistema de control de versiones. No existe un sistema de gestión de proyecto, ni gestión documental, se carece de servidores de ficheros y otros servicios y aplicaciones de valor agregado como blogs y portales; gestionándose las tareas de manera poco eficiente mediante documentos y hojas de cálculo, además de sobrecargar el servidor de control de versiones con documentos que, por no tributar a actividades directamente relacionadas con el desarrollo, no deberían estar alojadas en dicho servidor.

En general, no existe una guía de actividades para orientar el trabajo en cuanto a la gestión, control, configuración y mantenimiento de las tecnologías, herramientas, servicios y aplicaciones necesarias para el proyecto, tanto en el área de producción como en las tareas de gestión de las actividades de los integrantes de dicho proyecto.

Teniendo en cuenta la problemática anteriormente planteada se define como **problema a resolver**: ¿Cómo diseñar la Infraestructura Tecnológica del proyecto ERP-Cuba de forma que se organice el trabajo y el mantenimiento de las tecnologías y herramientas que soportan el desarrollo y la gestión del proyecto?

Con el fin de darle solución a dicho problema se tiene como **objeto de estudio** la AI orientada a proyectos de desarrollo de software, y el **campo de acción** se enmarca en la Infraestructura Tecnológica de proyectos de desarrollo de software de gestión.

Como **objetivo general** se propone diseñar la Infraestructura del proyecto ERP-Cuba siguiendo las pautas para el diseño de la Vista de Infraestructura de una Arquitectura de Software de Gestión.

Para dar cumplimiento al objetivo general se proponen los siguientes **objetivos específicos**:

- ✓ Realizar un estudio sobre diferentes fuentes que proponen actividades relacionadas con la AI.
- ✓ Describir las actividades que deben realizarse para diseñar y organizar la AI.
- ✓ Identificar responsabilidades, roles y artefactos para garantizar el buen funcionamiento de la AI dentro del proyecto.
- ✓ Validar la propuesta realizada aplicando las pautas definidas en el diseño de la Vista de AI del proyecto ERP-Cuba.

En este trabajo se crea presentar la definición de pautas para organizar y gestionar la IT del proyecto ERP-Cuba y diseñar la Vista de Infraestructura de la Arquitectura de dicho proyecto.

Para definir las pautas se realiza un estudio de las actividades que proponen fuentes reconocidas en el mundo del desarrollo de software así como conceptos que definen, y cómo estas pueden ser adaptadas al trabajo en la AI del proyecto. Se tienen en cuenta las características del proyecto, las condiciones tecnológicas y la disponibilidad de recursos de hardware para implementar su infraestructura.

Las pautas se presentan indicando actividades, tareas, roles y responsabilidades así como artefactos que se generan. Se identifican actividades para diseñar la Infraestructura y cada actividad contará con los siguientes atributos:

- ✓ Número: identifica el orden en que debe ser ejecutada.
- ✓ Actividad: nombre de la actividad con una pequeña descripción que explica brevemente el objetivo de la misma.
- ✓ Entrada: artefacto de entrada en caso de existir.
- ✓ Tareas: puede ser una o varias tareas que tributan a una actividad.
- ✓ Responsable: rol encargado de llevar a cabo la tarea.

- ✓ Salida: artefacto de salida.

Para evaluar los resultados de la aplicación de las pautas en el diseño de la Vista de Infraestructura del proyecto ERP-Cuba, se realiza una comparación del estado de la misma antes y después de aplicar dichas pautas.

La presente investigación se encuentra estructurada en 3 capítulos, a continuación se brinda una breve descripción de lo que trata cada uno:

Capítulo 1 Fundamentación Teórica: se abordan varios conceptos relacionados con la arquitectura en general. Se realiza un estudio de la AI desde el punto de vista de diferentes fuentes reconocidas y se definen las actividades (a grandes rasgos) que se deben realizar en esta disciplina arquitectónica, teniendo en cuenta su aplicación en un proyecto de desarrollo de software de gestión.

Capítulo 2 Pautas para el desarrollo de la Arquitectura de Infraestructura: se describen actividades que se realizan y roles que intervienen en el diseño e implantación de la AI de un sistema de gestión.

Capítulo 3 Vista de Infraestructura del proyecto ERP-Cuba: se presenta el diseño propuesto para la IT del proyecto ERP-Cuba, así como la evaluación de la aplicación de las pautas en la descripción de dicha vista arquitectónica.

CAPÍTULO 1 FUNDAMENTACIÓN TEÓRICA

Para poder apoyar a las organizaciones modernas y ajustarse a las necesidades actuales de las empresas es necesario introducirse con urgencia en una nueva rama de la arquitectura; la cual se hace muy necesaria para ayudar en la maduración de las arquitecturas en todo su conjunto.

La AI ayuda con la tecnología necesaria en una empresa o proyecto, para garantizar una estructura que soporte el desarrollo de la organización.

La presente investigación expone un conjunto de actividades que realizan algunas empresas con un alto grado de madurez en cuanto a la AI. Con el propósito de evaluar las actividades, además de los métodos y técnicas propuestas por dichas empresas y valorar en qué medida pueden ser adaptadas o utilizadas en la arquitectura del proyecto ERP-Cuba, se realiza un estudio de los aspectos que, teniendo en cuenta el alcance de la investigación, las necesidades reales y las características del proyecto, son esenciales para cumplir con el objetivo de este documento.

1.1 Arquitectura

1.1.1 Arquitectura de software

Existen muchas definiciones sobre arquitectura de software. Perry y Wolf en 1992 la definen como un conjunto de elementos que tienen una forma común. [27] Por su parte Bass, Clements, y Kazman en 1997 la definen como la estructura de las estructuras de un sistema, la cual comprende componentes de software, las propiedades visibles externas de estos componentes y las relaciones entre ellos. [3]

The Open Group of Architecture Framework (TOGAF) define la Arquitectura como la descripción formal de un sistema o un plan detallado del sistema a nivel de sus componentes que guían su implementación, dicho de otra forma: es la estructura de componentes, sus interrelaciones, los principios y guías que gobiernan su diseño y evolución a lo largo del tiempo. [36]

La definición oficial de arquitectura de software se ha acordado que sea la que brinda el documento de IEEE Std 1471-2000 que se formula así: “la arquitectura de software es la organización fundamental de un sistema representado por sus componentes, las relaciones entre ellos y con su entorno, y los principios que orientan su diseño y evolución.” [21]

1.1.2 Arquitecto de software

El rol del arquitecto ha sido siempre un tema polémico por las contradicciones que existen entre los que desempeñan dicho papel y el mundo de la informática en general. Puede definirse un arquitecto como la persona que hace arquitecturas y sus responsabilidades se restringen a hacer bien su trabajo. Esto puede incluir articular la visión arquitectónica con las necesidades del cliente, conceptualizar y experimentar con diferentes estrategias arquitectónicas; crear modelos, componentes y documentos de especificación de interfaces. [4]

Un arquitecto debe poseer ciertos rasgos y características para poder fundamentarse como un buen profesional y para realizar su trabajo con eficacia. Microsoft en su programa “Arquitecto Microsoft” considera algunas características comunes para todos los arquitectos independientemente del tipo que sea; a continuación se muestran algunas de las mismas:

Pensamiento visionario: durante la participación en un proyecto, el arquitecto debe considerar y proyectar la tecnología en el futuro, visionando los cambios que se producen en los negocios de los clientes, y la mejor manera de aprovechar las ventajas de la solución tecnológica actual en el futuro.

Investigar nuevas tecnologías: el arquitecto debe estar en continua investigación de nuevas tendencias tecnológicas, arquitectura de IT y las aplicaciones empresariales.

Comprender marcos de trabajo (frameworks) arquitectónicos y las mejores prácticas: los arquitectos entienden cuáles son los frameworks de arquitectura y cuáles empresariales, así como su valor en un proyecto. Los arquitectos seleccionan y usan metodologías en los proyectos, entienden el funcionamiento de frameworks, cómo la solución será desarrollada y el comportamiento antes y después del despliegue. Entienden el ciclo de vida de un proyecto y de una solución.

Seguir y divergir a la vez: cuando se trabaja en un entorno particular o en un proyecto específico, los arquitectos deben tener la capacidad de personalizar o modificar frameworks y/o las metodologías utilizadas para lograr una solución a un problema o requisito de negocio.

Poder para desarrollar rápidamente profundo conocimiento en una tecnología: ganando profundidad en múltiples tecnologías anteriores, el arquitecto puede asociar o transferir la capacidad de aprender otros métodos para investigar y ganar rápidamente experiencia en nuevas tecnologías.

Pueden trabajar con información ambigua o incompleta: los arquitectos deben colaborar en el proceso de indagación de la información para llegar a una solución, pero pueden empezar a trabajar con información limitada y según el proyecto progrese, tomar decisiones de compensación o equilibrio con el fin de mantener una solución que cumpla con los objetivos y poder satisfacer las exigencias de negocio que al principio fueron identificadas. Sin embargo, el arquitecto debe saber claramente si con la información limitada puede empezar a trabajar sin poner en riesgo el proyecto posteriormente por cambios drásticos o si este debe suspenderse antes de recopilar información mínima para comenzar las tareas. Es importante el trabajo conjunto de todo el equipo en este aspecto.

Buenas habilidades de comunicación: debe poseer un alto grado de comunicación social y buenos modales, dado que un arquitecto tiene que interactuar con clientes y muchas veces estos no entienden de tecnologías y hay que explicarles. Un arquitecto también debe ser capaz de transmitir conocimiento y experiencias a un equipo o grupo de personas ya que es él quien debe comunicar las soluciones propuestas para un determinado requerimiento. [13]

1.1.3 Arquitecto de Infraestructura

Según Leandro Doeyo, ex-director académico de la plataforma .NET para América Latina, el Arquitecto de Infraestructura se encarga de planificar el parque de hardware, la red, el software básico como el Sistema Operativo (SO), la base de datos y también herramientas de monitoreo que ayuden a controlar que lo anterior funcione correctamente. A este arquitecto no le concierne cómo las aplicaciones fueron desarrolladas o lo que las mismas hacen (en términos de negocio) sino, que las ve más como procesos que deben estar ejecutándose de forma adecuada. [14]

Simon Guest, consultor de AI de Microsoft, dice que el dominio del Arquitecto de Infraestructura incluye responsabilidades en el diseño del centro de datos, despliegue y mantenimiento de aplicaciones y servicios a lo largo de la organización. Este rol incluye trabajos con el arquitecto de soluciones para buscar escalabilidad, confiabilidad, manejabilidad, rendimiento, y seguridad, además de trabajar con el arquitecto empresarial para definir direcciones estratégicas. [18]

Un Arquitecto de Infraestructura debe poseer un conjunto de competencias tales como:

- Conocimientos avanzados sobre administración de sistemas, fundamentalmente GNU/LINUX y conocimiento medio sobre redes y arquitectura TCP/IP.

- Dominar la administración de bases de datos y tecnologías web.
- Conocimiento avanzado del protocolo ssh.
- Tener conocimientos medios sobre lenguajes de script como: Bash, Python, Perl.
- Dominar las tecnologías de almacenamiento o al menos saber cómo funcionan las mismas.
- Utilizar métodos para la evaluación de arquitecturas y sus atributos de calidad.
- Trabajar sobre herramientas de monitoreo de redes y servicios.
- Utilizar tecnologías de servidores.

1.2 Arquitectura de Infraestructura desde diferentes puntos de vista

A pesar de ser la AI una disciplina relativamente nueva, existen diferentes metodologías que presentan sus puntos de vista en lo relacionado a las actividades que se realizan en esta rama de la arquitectura, también exponen las responsabilidades de los roles que intervienen en esta disciplina. Algunas empresas como Microsoft han logrado cierto grado de madurez en cuanto a este tema.

1.2.1 Microsoft

Microsoft define diferentes perfiles arquitectónicos, dentro de los que se encuentra el Arquitecto de Infraestructura. Este es responsable de las decisiones del área de infraestructura, de mantener el entorno de Tecnología Informática (TI) y los usuarios finales, así como de comunicarse constantemente con los ingenieros que mantienen áreas específicas de la infraestructura. Se encargan de crear una arquitectura que cumple con los acuerdos de niveles de servicio, de las necesidades de los empresarios y apoya las aplicaciones y soluciones que se requieren para operar diariamente en las empresas.

Para optimizar la infraestructura Microsoft propone una metodología donde se definen una serie de actividades y modelos con el fin de mejorar la infraestructura de las empresas.

La Optimización de la Infraestructura (OI) de Microsoft se estructura según tres modelos de TI: optimización de la infraestructura principal, optimización de la infraestructura de plataforma de aplicación y optimización de la infraestructura de productividad empresarial.

Cada uno de los modelos de OI consta de cuatro niveles de madurez de procesos y clasificaciones de capacidad como grupos lógicos de requisitos para cada nivel de madurez. La OI principal se centra en elementos base de los servicios y componentes de TI, la OI de plataforma de aplicaciones se centra en

procedimientos recomendados para el desarrollo de software y la OI de productividad empresarial se centra en la infraestructura necesaria para maximizar la comunicación, colaboración y productividad del usuario final. La siguiente tabla destaca las capacidades de cada modelo de OI.

1.2.1.1 Capacidades de la Optimización de la Infraestructura Principal

El modelo de Optimización de la Infraestructura Principal define cinco capacidades que constituyen requisitos iniciales para elaborar una Infraestructura de TI más ágil. Estas cinco capacidades constituyen la base de cada uno de los niveles de madurez.

Administración de identidades y acceso.

Describe cómo los clientes deben administrar las identidades de personas y activos, soluciones que deben implementarse para administrar y proteger los datos de identidad, y cómo administrar el acceso a los recursos de los usuarios móviles corporativos, clientes o asociados desde fuera de un cortafuego (firewall).

Administración de escritorios, dispositivos y servidores.

Describe cómo se deben administrar los clientes, los escritorios, los dispositivos móviles y los servidores, además de cómo implementar revisiones, sistemas operativos y aplicaciones a través de la red.

Protección y recuperación de datos.

Proporciona una administración estructurada y ordenada de copias de seguridad, almacenamiento y restauración. A medida que prolifera la información y el almacenamiento de datos, mayor es la presión a la que se ven sometidas las organizaciones a la hora de proteger la información y ofrecer una recuperación efectiva y rentable siempre que sea necesario.

Seguridad y funciones de red.

Describe lo que los clientes deben considerar para implementar en su Infraestructura de TI con el fin de ayudar a garantizar que la información y comunicación queden protegidas del acceso no autorizado. Ofrece también un mecanismo para proteger la Infraestructura de TI frente a ataques de denegación y virus, al mismo tiempo que protege el acceso a recursos corporativos.

Proceso de TI y seguridad.

Ofrece orientación probada de procedimientos recomendados para diseñar, desarrollar, poner en funcionamiento y compatibilizar soluciones de una manera rentable al mismo tiempo que se garantiza la confiabilidad, la disponibilidad y la seguridad. Aunque es necesaria una tecnología sólida para cumplir con los requisitos de confiabilidad, disponibilidad y servicios de TI muy seguros, la tecnología por sí sola no es suficiente; también se necesita un buen trabajo de los procesos y del personal (capacidades, funciones y responsabilidades).

1.2.1.2 Niveles del modelo de optimización de la infraestructura principal

Además de las capacidades, el Modelo de Optimización de Infraestructura Principal asigna cuatro niveles de optimización (Básico, Estandarizado, Racionalizado y Dinámico) para cada capacidad. Las características de estos niveles de optimización son las siguientes:

Nivel de optimización 1: Básico.

La infraestructura de TI básica se caracteriza por procesos manuales y localizados, un control central mínimo, directivas de TI no existentes o no obligatorias, así como estándares de seguridad, copias de seguridad, administración e implementación de imágenes, cumplimiento y otras prácticas comunes de TI. El estado general de las aplicaciones y los servicios no se conocen debido a la falta de herramientas y recursos. De manera general, todas las revisiones, implementaciones de software y servicios se ofrecen de forma manual.

Nivel de optimización 2: Estandarizado.

La infraestructura estandarizada introduce controles a través del uso de estándares y directivas para administrar escritorios y servidores con el propósito de controlar la manera en la que se introducen los equipos en la red y para administrar recursos, directivas de seguridad y control de acceso mediante el uso de Active Directory (Programa de Directorio Activo de Microsoft). Los clientes del estado estandarizado reconocen el valor de los estándares básicos y ciertas directivas, pero creen que aún quedan aspectos por mejorar. Por lo general, todas las revisiones, implementaciones de software y servicios de escritorio se proporcionan con una interacción media y un costo medio elevado. Estas organizaciones disponen de un inventario razonable de hardware y software y comienzan a administrar licencias. Las medidas de seguridad se mejoran a través de un perímetro bloqueado, no obstante, la seguridad interna puede aún suponer un riesgo.

Nivel de optimización 3: Racionalizado.

La infraestructura racionalizada es donde los costos relacionados con la administración de escritorios y servidores se encuentran en su nivel más bajo y los procesos y directivas han sido optimizadas para desempeñar un papel más importante en el soporte y expansión del negocio. La seguridad es muy pro-activa y la respuesta a las amenazas y los desafíos es rápida y controlada. El uso de la implementación sin interacción ayuda a reducir los costos, el tiempo de implementación y las dificultades técnicas. El número de imágenes es mínimo y el proceso para administrar equipos de escritorio presenta una escasa interacción. Estos clientes disponen de un claro inventario de hardware y software y sólo adquieren las licencias y equipos que necesitan.

Nivel de optimización 4: Dinámico.

Los costos están totalmente controlados, es posible la integración entre usuarios y datos, escritorios y servidores; la colaboración entre los usuarios y los departamentos es omnipresente y por último, los usuarios móviles disponen de niveles de servicio y capacidades independientemente de la ubicación. Los procesos están totalmente automatizados y con frecuencia, integrados en la propia tecnología, lo que permite la alineación y la administración de TI según las necesidades de la empresa. Las inversiones adicionales en tecnología producen beneficios específicos, rápidos y medibles para la empresa. El uso de software de aprovisionamiento propio y de sistemas de cuarentena para garantizar la administración de revisiones y el cumplimiento con las directivas de seguridad establecidas permite a la organización de infraestructura dinámica automatizar los procesos, por lo que ayuda a mejorar la confiabilidad, a reducir los costos y a incrementar los niveles de servicio.

1.2.1.3 Enfoques por etapas

Microsoft recomienda un enfoque por fases para cumplir con los requisitos de cada una de las capacidades de OI, dichas fases se explican a continuación:

Evaluación: permite determinar las capacidades y recursos actuales dentro de la organización.

Identificación: permite determinar lo que necesita para alcanzar sus objetivos y qué capacidades se desean incorporar.

Evaluación y planeación: permite determinar lo que se necesita hacer para implementar las capacidades perfiladas en la fase Identificación.

Implementación: ejecuta el plan que se ha creado en la fase anterior.

1.2.1.4 Actividades propuestas por Microsoft

Las actividades propuestas por Microsoft son:

- ✓ Implementar un servicio de directorio principal para la autenticación.
- ✓ Implementar una solución para los entornos de desarrollo, la cual esté basada en imágenes y sea con poca interacción, es decir, los sistemas operativos se pueden implementar a través de la red mediante una carpeta compartida o de forma local usando un dispositivo de almacenamiento extraíble, como un CD, DVD, disco USB u otro dispositivo.
- ✓ Implementar uno o más firewall con configuración de bloqueo.
- ✓ Implementar servicios de red como servidores DNS (Domain Name Server o Servidor de Nombre de Dominio), DHCP (Dynamic Host Configuration Protocol o Protocolo de Configuración Dinámica de Servidor).
- ✓ Implementar una solución antivirus para equipos de escritorio y servidores.
- ✓ Implementar un sistema de copias de seguridad y restauración para los servidores.
- ✓ Implementar herramientas de protección de datos.
- ✓ Administrar y supervisar servidores críticos. [25]

Del conjunto de actividades anteriores no se tienen en cuenta en el presente trabajo, ni la implementación de un servicio de directorio principal para la autenticación, ni la implementación de servicios de red como servidores DNS, DHCP; debido a que el proyecto ERP-Cuba funciona dentro de una Infraestructura más abarcadora que es la implementada en la UCI y esta brinda los servicios de red necesarios, además de correo y mensajería instantánea, por lo que no es necesario implementarlos en la infraestructura del proyecto.

Las etapas definidas por Microsoft son fundamentales a la hora de desarrollar la Infraestructura de una empresa u organización por lo que las mismas se tienen en cuenta en el marco de este trabajo, estas plantean la evaluación e identificación de los recursos y capacidades del proyecto, que ayudarán a realizar un desarrollo satisfactorio.

1.2.2 The Open Group Architecture Framework

TOGAF es un Marco de trabajo (framework) de arquitectura que permite diseñar, evaluar y construir la arquitectura correcta para una organización. El centro de TOGAF es el ADM (Architecture Development Method o Método de Desarrollo de Arquitectura), un método confiable y aprobado para desarrollar una arquitectura de IT que satisfaga las necesidades del negocio.

Asociadas a la Arquitectura de Tecnología que define TOGAF, se encuentran otras actividades que por lo general otros autores las atribuyen a la AI. Debido a que muchas coinciden con las definidas dentro del proyecto ERP-Cuba, se tienen en cuenta durante el desarrollo de la investigación actual. Según TOGAF, esta disciplina describe las capacidades lógicas de software y hardware requeridas para soportar el despliegue del negocio, los datos y los servicios de aplicaciones. Además, incluye la Infraestructura de TI, redes, comunicaciones, procesamiento y estándares.

1.1.1.1 Pasos definidos por TOGAF para desarrollar una Arquitectura Tecnológica.

Paso 1: Crear una descripción de línea base en formato TOGAF.

El objetivo de este paso es convertir la descripción del sistema existente en los servicios.

Actividades

- ✓ Documentar preguntas clave para poner a prueba los méritos de la arquitectura de tecnología.
- ✓ Documentar los criterios para la selección de la arquitectura del portafolio de servicios.

Paso 2: Considerar los diferentes modelos de referencia de arquitectura, puntos de vista y herramientas.

El objetivo de este paso es realizar un análisis de las diferentes preocupaciones (los requisitos) o puntos de vista, y documentarlos pertinentemente.

Actividades

- ✓ Seleccionar los recursos pertinentes de la arquitectura de tecnología (modelos de referencia, patrones y otros).
- ✓ Seleccionar los puntos de vista relevantes.
- ✓ Considerar la elaboración de los puntos de vista, al menos en:

- Red informática / Vista de hardware.
 - Vista de estándares.
 - Vista de datos.
- ✓ Identificar las herramientas y técnicas adecuadas para la captura, el modelado y análisis, en asociación con los puntos de vista seleccionados.

Paso 3: Crear un modelo arquitectónico de construcción en bloques.

El objetivo de este paso es, en general, determinar cómo los servicios necesarios en el sistema se agrupan después de considerar todos los puntos de vista pertinentes del uso de la arquitectura.

Actividades

- ✓ Identificar los bloques de construcción de la arquitectura de tecnología.
- ✓ Para cada punto de vista, crear el modelo para el punto de vista específico necesario.
- ✓ Documentar la justificación para las decisiones de los bloques de construcción en el documento de la arquitectura.

Paso 4: Seleccionar el portafolio de servicios requerido por cada bloque de construcción.

El objetivo de este paso es seleccionar los portafolios de servicios para cada bloque de construcción generado en el paso 3.

Para cada Bloque de Construcción de la Arquitectura (ABB por sus siglas en inglés, Architecture Building Blocks) se debe crear una cartera de descripción de servicio como un conjunto de servicios. El conjunto de servicios debe ser probado para asegurar que la funcionalidad proporcionada cumple con los requisitos de la aplicación.

Actividades

- ✓ Producir servicios agrupados por afinidad.
- ✓ Comprobar grupos de afinidad contra las necesidades.
- ✓ Descripción del documento de gama de servicios para cada uno de los ABB.

Paso 5: Confirmar que las metas y objetivos del negocio se satisfagan.

El propósito de este paso es aclarar y revisar los objetivos de negocio y otros de la implementación de la arquitectura. Esto es necesario como control de que la arquitectura de tecnología cumple con estos objetivos.

Una lista de preguntas claves es usada para evaluar el modelo de la arquitectura y el portafolio de descripción de servicios, con el fin de poner a prueba su mérito y la integridad.

Actividades

- ✓ Llevar a cabo una revisión formal del puesto de control del modelo de arquitectura y bloques de construcción con las partes interesadas, validando que se cumplan los objetivos del negocio, utilizando una lista de preguntas clave.

Paso 6: Determinar el conjunto de criterios para elegir especificaciones y portafolio de especificaciones.

El objetivo de este paso es desarrollar un conjunto de criterios para la elección de las especificaciones y los portafolios de las especificaciones.

Actividades

- ✓ Lluvia de ideas para la elección de requisitos de los portafolios de especificaciones basándose en criterios previamente utilizados para el sistema existente y la extrapolación de los nuevos elementos arquitectónicos.
- ✓ Reunirse con los patrocinadores y presentar el estado actual para negociar una continua solicitud de los patrocinadores.

Paso 7: Completar la definición de la arquitectura.

El objetivo de este paso es especificar completamente la arquitectura de tecnología. Este es un proceso complejo e iterativo, en el que la selección de bloques de construcción y las interfaces tiene un gran impacto en cómo se cumplen los requisitos iniciales.

Actividades

- ✓ Asegurar una documentación clara de todas las interfaces para cada bloque de construcción (API, formatos de datos, protocolos, interfaces de hardware).
- ✓ Documentar plenamente cada elemento básico de los bloques de construcción de arquitectura.

- ✓ Documentar la justificación para las decisiones de los bloques de construcción en el documento de arquitectura.
- ✓ Generar el documento de arquitectura de tecnología.

Paso 8: Llevar a cabo un análisis de carencia.

El objetivo de este paso es identificar las áreas del sistema actual y destino, para las cuales no se han hecho las prestaciones en la arquitectura de tecnología. Esto es necesario a fin de identificar los proyectos que se realizarán en el marco de la implementación del sistema destino.

Actividades

- ✓ Identificar los bloques de construcción para ser prorrogados, clasificarlos según sean cambiados o modificados.
- ✓ Identificar bloques de construcción eliminados.
- ✓ Identificar los bloques de construcción nuevos. [35]

Dadas estas actividades presentadas por TOGAF se toman para desarrollar este trabajo el modelo de construcción en bloques y las demás actividades asociadas a esta, dado que se ajustan para la creación de diferentes áreas de trabajo, y permiten, en el diseño de una Plataforma de Servicios (PS), agrupar las funcionalidades y servicios por módulos, teniendo en cuenta afinidades tecnológicas y otros aspectos como consumo de recursos y otras compatibilidades. También es importante tener en cuenta el análisis de carencia, este permite identificar qué le falta a la Infraestructura según los requerimientos tecnológicos analizados y qué haría falta para suplir todas las necesidades.

1.2.3 Instituto de Ingeniería de Software

El Instituto de Ingeniería de Software (Software Engineering Institute, SEI por sus siglas en inglés) es un centro de investigación y desarrollo que se encuentra en la Universidad de Carnegie Mellon. Su objetivo fundamental es ayudar a las organizaciones a mejorar las capacidades de Ingeniería de Software y a desarrollar o adquirir el software adecuado, dentro del presupuesto y tiempo específico. Es conocida por sus programas de gran categoría en Ciencias de la Computación y la Ingeniería. Según el SEI la AI tiene responsabilidades en el Entorno de Desarrollo (ED), hosting de aplicaciones y servicios a lo largo de la organización. [6]

El ED es una de las áreas fundamentales en las que se enmarca la AI, este punto es importante tenerlo en cuenta, dado que dentro de él se definen la mayoría de las herramientas necesarias para el desarrollo del proyecto. Los servicios son de fundamental importancia, brindan el soporte requerido para que la implementación se lleve a cabo con la mayor agilidad posible, posibilitando un desarrollo colaborativo, salvadas de información y recuperación de la misma en caso de desastres u otro factor de riesgo que se pueda dar, también provee soporte para una mejor gestión del proyecto. Estos son aspectos a tener en cuenta para definir las actividades que debe realizar un arquitecto de infraestructura.

La documentación que publica el SEI sobre este tema es muy escasa, teniendo acceso solamente a algunas definiciones, pero nunca a alguna metodología con actividades descritas que puedan servir de guía, se tuvieron en cuenta en esta investigación dado que son una entidad de suma importancia en temas sobre arquitectura y se conoce que consideran el rol de Arquitecto de Infraestructura al que le atribuyen actividades similares a las definidas en la investigación actual.

1.3 Análisis de Propuestas Estudiadas

Teniendo en cuenta las características del proyecto ERP-Cuba, formado por un equipo considerablemente grande de desarrolladores, con una amplia gama de roles que intervienen en el desarrollo del proyecto, se hace indispensable contar con determinados servicios para la gestión de información, tareas y actividades. Se necesita además una gran diversidad en las herramientas que se utilizan diariamente dadas las disímiles responsabilidades que tienen sus integrantes. Por tales motivos se optó por seguir lo definido por Microsoft en cuanto al ED y se toman otras consideraciones que hace TOGAF, como son: implementar un modelo de construcción en bloques o módulos para la asignación de procesos a servidores y una mejor organización que facilite el mantenimiento de las aplicaciones y servicios.

Las actividades asociadas al ED y al modelo de construcción en bloques serán extrapoladas al trabajo en el despliegue en la medida de lo posible y teniendo en cuenta las características de este proceso, con el objetivo de garantizar el soporte tecnológico del mismo y el funcionamiento adecuado de las aplicaciones que se despliegan. Microsoft, entre sus actividades, propone implementar determinados servicios de red, correo y mensajería que no serán tomados en cuenta en este documento dado que estos servicios se consumen de la infraestructura de la UCI, sobre la cual funciona el proyecto ERP-Cuba, siendo el objetivo

de este trabajo solamente la infraestructura de dicho proyecto, abarcando las necesidades propias de este y que no son suplidas por la macro-infraestructura UCI.

Teniendo en cuenta las necesidades organizativas y tecnológicas se propone diseñar una plataforma para el mantenimiento de las aplicaciones y servicios a lo largo del desarrollo, este, además, es uno de los puntos de vista planteados por SEI. Se propone, también, llevar a cabo un monitoreo y control periódico sobre los servidores de misión crítica como define Microsoft en sus propuestas.

Partiendo de lo anterior, y con el fin de organizar mejor la gestión de la IT, se definen tres áreas fundamentales hacia las que se dirige la AI: ED, PS y Despliegue.

Entorno de Desarrollo: es la infraestructura básica sobre la que funciona el proyecto. En esta área, se define el SO y la plataforma tecnológica para el desarrollo, lenguajes de programación, Entorno de Desarrollo Integrado (IDE, por sus siglas en inglés) y herramientas dirigidas al desarrollo.

Plataforma de Servicios: se define como el portafolio de servicios que soportan al desarrollo. Esta plataforma, por lo general, está constituida por los servicios elementales: Sistema de Control de Versiones, servidores de datos, servidores web o de aplicaciones y servidores de ficheros (FTP), además de otros servicios que dotan la plataforma de un valor agregado y se han convertido en fundamentales, como: Sistemas de Gestión de Proyectos, Sistemas de Gestión Documental y otros.

Despliegue: esta área la definen las actividades que se realizan en la misma, como: la definición de la Vista de Despliegue, incluidos los escenarios de despliegue. Es el “momento” en que se definen los escenarios de despliegue realizando para ello pruebas a la arquitectura para evaluar determinados atributos de calidad, como: pruebas de rendimiento, carga, estrés y tiempo de respuesta. Se diseña telemáticamente el Centro de Datos y se definen estrategias para la actualización de la tecnología en las entidades de despliegue.

Para cada una de estas áreas de trabajo se define un conjunto de actividades, roles y artefactos que se mencionan a continuación y serán descritos en el capítulo siguiente.

A continuación se muestra una tabla con las actividades definidas para cada una de las áreas de trabajo:

Tabla 2 Actividades de las Áreas de Trabajo

Área: Entorno de Desarrollo			
Nº	Actividad	Entrada	Salida
1	Análisis de los requerimientos tecnológicos del sistema.	Minuta de reunión con los clientes.	Análisis de requerimientos.
2	Definición de Sistemas Operativos.	Análisis de requerimientos.	Entorno de Desarrollo.
3	Definición de lenguajes de programación.	Análisis de requerimientos.	Entorno de Desarrollo.
4	Definición de herramientas y tecnologías.	Análisis de requerimientos.	Entorno de Desarrollo.
5	Definir configuraciones básicas de las tecnologías.	Entorno de Desarrollo.	Entorno de Desarrollo.
6	Implementación de la Plataforma de Desarrollo.	Entorno de Desarrollo.	Plataforma de Desarrollo.
7	Definición de políticas de trabajo y listas de verificación para garantizar el buen funcionamiento de la plataforma.	Plataforma de Desarrollo.	Políticas de trabajo, listas de verificación, estrategia de soporte.
Área: Plataforma de Servicios			
Nº	Actividad	Entrada	Salida
1	Definir módulos necesarios para la Plataforma de Servicios.		Diseño de Plataforma de Servicios. Diseño Estructural de Tecnología.
2	Definir políticas de trabajo y listas de verificación.	Diseño de Plataforma de Servicios.	Políticas de Trabajo para el mantenimiento de la Plataforma de Servicios.
3	Monitorizar los servicios y aplicaciones durante todo el proceso de desarrollo.		Informe de estado de los servicios y aplicaciones Informe de uso de los servicios. Informe sobre el consumo de ancho de banda.
Área: Despliegue			

Nº	Actividad	Entrada	Salida
1	Realizar pruebas de rendimiento, carga, estrés y tiempo de respuesta a las aplicaciones desarrolladas para definir escenarios de despliegue.		Escenarios de despliegue.
2	Diseño de Centro de Datos.		Diseño de Centro de Datos.

En una de estas áreas, PS, se decide implementar el modelo de construcción en bloques propuesto por TOGAF, donde se definen los módulos considerados como necesarios para soportar el desarrollo del proyecto.

Habiendo analizado las definiciones de las fuentes estudiadas, pero haciendo las variaciones necesarias para adaptarlas al proyecto, se presentan a continuación los roles identificados para llevar a cabo las actividades en las áreas definidas, así como las responsabilidades de cada uno de estos de forma general:

Roles que intervienen en la Arquitectura de Infraestructura.

Rol	Responsabilidad	Descripción
Arquitecto de Infraestructura	Planear y dirigir las pruebas al sistema para la evaluación de la arquitectura.	Diseñar las pruebas para la evaluación de la arquitectura. El Arquitecto de Infraestructura se encarga de proponer métodos de evaluación de los atributos de calidad de la arquitectura así como diseñar y planear dichas pruebas.
	Diseñar una estructura para la protección y recuperación de datos.	Proporcionar una administración estructurada y ordenada de copias de seguridad para el almacenamiento y recuperación de los datos. Se encarga de proponer las herramientas necesarias para proteger la información y ofrecer una recuperación efectiva siempre que sea necesario.
Responsable de	Definición y mantenimiento del ED.	Definición de herramientas y tecnologías para el

ED		desarrollo, como SO, lenguajes de programación, clientes de datos y otros.
Responsable de PS	Mantenimiento de aplicaciones y servicios a lo largo del proceso de desarrollo.	Garantizar la disponibilidad de los servicios y aplicaciones que conforman la Plataforma. Prever las necesidades de escalado y garantizar el funcionamiento adecuado de la PS.
	Monitoreo de aplicaciones y servicios.	Define métodos y métricas para la evaluación del comportamiento de los servicios y ayuda en la toma de decisiones en ese sentido.
	Proponer soluciones de escalabilidad, alta disponibilidad y seguridad.	Propone soluciones para garantizar el escalado, además de una alta disponibilidad en cualquiera de los escenarios donde se implemente la aplicación. Propone soluciones para lograr una alta flexibilidad ante los cambios o actualizaciones de la aplicación además de garantizar la seguridad y el buen funcionamiento de la misma.
	Optimización de aplicaciones.	Garantiza la alta disponibilidad de las aplicaciones y servicios a lo largo del proceso de desarrollo, proponiendo y llevando a cabo soluciones de optimización. El responsable de PS es un velador constante por el comportamiento de los servicios y aplicaciones, y es su responsabilidad decidir el momento para escalar dichos servicios y aplicaciones, teniendo en cuenta su comportamiento y el crecimiento en número de los usuarios, así como las actualizaciones de las tecnologías usadas.

Responsable de despliegue	Diseño y soporte telemático del despliegue.	Diseño de la vista de despliegue, incluidos los escenarios arquitectónicos para el funcionamiento óptimo de la aplicación, teniendo en cuenta los recursos mínimos para cada uno de los escenarios. Se incluye el diseño de componentes y la estrategia de despliegue. Define además la estrategia de actualizaciones de la aplicación.
	Diseño telemático del Centro de Datos.	Realiza el diseño lógico del Centro de Datos, configuración del sistema de réplicas, definición del sistema de almacenamiento y recursos software y hardware para el funcionamiento de la aplicación en escenarios con una gran densidad de usuarios, garantizando alta disponibilidad, seguridad y niveles aceptables de escalabilidad. Trabaja directamente con el arquitecto de datos buscando configuraciones óptimas para el funcionamiento saludable de todo el sistema.
Desarrollador	Instalación, configuración y mantenimiento de herramientas, servicios y aplicaciones.	Sus actividades están muy relacionadas con el área específica en la que se desempeña, ya sea en el ED, PS o en el Despliegue, su función es la instalación y mantenimiento de herramientas necesarias para el trabajo en dichas áreas.

Conclusiones

En el presente capítulo se realizó un estudio sobre la AI desde los puntos de vista de Microsoft, TOGAF y SEI. De dichas fuentes se tomaron ideas como son: la implementación de un ED constituido por

herramientas y tecnologías en dependencia de los roles que intervienen en el proyecto; la implementación de un modelo de construcción en bloques o módulos donde se definen portafolios de servicios para una mejor gestión de la PS.

Se definieron las áreas de trabajo en las que se enmarca la AI así como las actividades que se llevan a cabo en cada una de ellas, se definieron roles y responsabilidades para la realización de dichas actividades.

En el capítulo siguiente se explicarán con más detalles los objetivos de cada una de las actividades definidas así como los roles que intervienen y se explicará en qué consisten cada uno de los artefactos que se generan.

CAPÍTULO 2 PAUTAS PARA EL DESARROLLO DE LA ARQUITECTURA DE INFRAESTRUCTURA

En este capítulo se presentan las pautas para diseñar la infraestructura de la arquitectura para un proyecto de software de gestión. Como se planteó en el capítulo anterior, se definen tres áreas de trabajo que permiten organizar y distribuir cada actividad para un mejor cumplimiento de las pautas a seguir. Cada una de las pautas definidas está compuesta por un conjunto de actividades que poseen diferentes tareas, artefactos que se generan y roles responsables de realizar cada una de dichas actividades.

2.1 Áreas de trabajo de la Arquitectura de Infraestructura

2.1.1 Entorno de Desarrollo

El ED es donde se analizan los requisitos tecnológicos para el desarrollo del proyecto. Se definen las herramientas y tecnologías necesarias, como lenguajes de programación, IDEs de desarrollo, sistemas operativos, herramientas de modelado y para el trabajo con documentos. Se define además la plataforma de desarrollo, la cual permite contar con todas estas tecnologías y herramientas integradas en una solución reduciendo la complejidad del mantenimiento de dicha plataforma. Se define un conjunto de políticas de trabajo para que las actividades en esta área sean cumplidas con la mayor eficiencia posible.

2.1.1.1 Actividades fundamentales del Entorno de Desarrollo.

1. Análisis de los requerimientos tecnológicos del sistema.

El objetivo de esta tarea es especificar los requerimientos tecnológicos necesarios para realizar el desarrollo del proyecto atendiendo a las necesidades del mismo. Se realiza un análisis las diferentes características que debe tener el lenguaje de programación que se usará, el SO a usar en el desarrollo, la aplicación a desarrollar, entre otras.

Tareas:

- ✓ Analizar los requerimientos tecnológicos del sistema en cuanto a:
 - **Características de la aplicación:** centralizada o distribuida, densidad estimada de usuarios finales, tiempo de respuesta, seguridad.
 - **Características del desarrollo:** tecnologías y herramientas libres o propietarias, SO, lenguajes de programación, bibliotecas de software.

- **Necesidad de aplicaciones y servicios de soporte:** sistemas de control de versiones, sistemas de administración de proyectos, sistemas de gestión documental, servidores de ficheros, repositorios de herramientas.
- ✓ Redactar Informe con los requerimientos tecnológicos del sistema.

Artefacto: Análisis de requerimientos.

Posee todos los requerimientos tecnológicos y características de la aplicación y herramientas que se deben usar para la implementación del proyecto.

Responsable: Arquitecto de Infraestructura. Responsable del ED.

2. Definición de Sistemas Operativos (SO).

Generalmente esto depende del cliente, que es quien se decide por un SO en particular, pensando en costos futuros y en otros aspectos. Sin embargo, esto también está sujeto a una decisión del equipo y de las personas encargadas de la selección del mismo, dado que hay que tener en cuenta los requisitos tecnológicos y del sistema. Una mala selección del SO puede influir en el aumento del costo, tanto de mantenimiento, como en recursos humanos, de aquí la importancia de seleccionar el SO adecuado.

Teniendo en cuenta los requerimientos del sistema y las propuestas hechas tanto por parte del cliente, como por parte del equipo de desarrollo, junto con los analistas y arquitectos, se definen los SO a usar.

Al elegir un SO adecuado se prefiere uno que lo tenga “todo”, o al menos “casi todo”, pero esto es algo muy difícil de lograr la mayoría de las veces, por eso es bueno tener en cuenta un conjunto de elementos que ayudan a realizar una buena selección:

Entradas:

- ✓ Análisis de requerimientos

Tareas:

- ✓ Realizar un estudio de los diferentes SO teniendo en cuenta lo señalado en los requerimientos en cuanto a SO.
- ✓ Seleccionar los SOs que serán candidatos para realizar una comparación entre las características de los mismos.

- ✓ Realizar una comparación de los candidatos teniendo en cuenta los siguientes elementos:
 - **Fiabilidad:** se debe buscar un sistema robusto. Un sistema en el cual se pueda confiar lo suficiente como para realizar las tareas que se necesitan. La fiabilidad es la característica de los sistemas informáticos por la que se mide el tiempo de funcionamiento sin fallos. En el caso del hardware, se han conseguido altísimos grados de fiabilidad, mientras que en el software siguen existiendo bugs (errores) que dificultan el buen funcionamiento de los programas. Cuando uno de estos bugs aparece, es normal que el programa se bloquee, impidiendo al operador seguir trabajando con el sistema y obligando a reiniciar la máquina.
 - **Rendimiento:** El rendimiento se refiere a la capacidad de ejecutar las aplicaciones que se usan de la mejor manera posible. En el caso de tratarse de un sistema para servidor, el rendimiento será de suma importancia debido a la gran cantidad de datos a manejar en una red y a la cantidad de usuarios que usarán la máquina simultáneamente. El rendimiento es una proporción entre el resultado obtenido y los medios que se utilizaron. Se trata del producto o la utilidad que rinde alguien o algo.
 - **Seguridad:** es importante que los datos se mantengan seguros. Si el sistema que se elegirá será usado para el trabajo (ya sea de forma personal o en una gran empresa), ésta es una característica importante que puede ser definitiva en la decisión sobre un sistema en vez de otro.
 - **Controladores de dispositivos:** existen muchos dispositivos hoy en día que quizás se quieren utilizar junto al SO. Por ejemplo, si se necesita utilizar una tarjeta capturadora de televisión, una tarjeta de sonido especial, entre otras. Es de suma importancia que el SO que se seleccione disponga de controladores adecuados para los dispositivos que se usarán. Un controlador de dispositivo, llamado normalmente controlador (en inglés: device driver) es un programa informático que permite al SO interactuar con un periférico, haciendo una abstracción del hardware y proporcionando una interfaz (posiblemente estandarizada) para usarlo. Se puede esquematizar como un manual de instrucciones que le indica al SO, cómo debe controlar y comunicarse con un dispositivo en particular. Por tanto, es una pieza esencial, sin la cual no se podría usar el hardware. Existen tantos tipos de controladores como tipos de periféricos, y es común encontrar más de un controlador posible para el mismo dispositivo, cada uno ofreciendo un nivel distinto de funcionalidades. Por ejemplo, aparte de los oficiales (normalmente

disponibles en la página web del fabricante), se pueden encontrar también los proporcionados por el SO, o también versiones no oficiales hechas por terceros.

- **Aplicaciones comerciales y gratuitas:** existe una gran cantidad de aplicaciones ya sean comerciales o gratuitas para cada SO. Para algunos, como el caso de UNIX/Linux, existen millones de aplicaciones gratis, en cambio, para otros como Windows, existen más aplicaciones comerciales que gratuitas.
- **Soporte:** hoy en día existen varias opciones de SO entre los cuales elegir (no así hace algunos años atrás). Muchos de estos sistemas son gratuitos y no disponen de ningún soporte técnico. De acuerdo con las necesidades puede ser importante contar con personal especializado para que sirva de guía en el uso de un sistema determinado. El soporte técnico es un rango de servicios que proporcionan asistencia con el hardware o software de una computadora, o algún otro dispositivo electrónico o mecánico. En general los servicios de soporte técnico tratan de ayudar al usuario a resolver determinados problemas con algún producto en vez de entrenar o personalizar. La mayoría de las compañías que venden hardware o software ofrecen soporte técnico de manera telefónica o en línea. Las instituciones y compañías por lo general tienen sus propios empleados de soporte técnico. Existen a su vez múltiples lugares libres en la web respecto a soporte técnico, en los cuales los usuarios más experimentados ayudan a los novatos.
- **Costo del sistema y propiedad:** seguramente éste siempre será el factor más decisivo a la hora de seleccionar un SO. En algunos casos, el hecho de que exista un SO por el cual no deba pagar es muy tentador y por lo general lo hace casi definitivo.

Artefacto: Entorno de desarrollo.

Se obtiene el ED con el SO definido.

Responsable: Responsable del ED.

3. Definición de lenguajes de programación.

El objetivo de esta actividad es definir los lenguajes de programación que se usarán en el desarrollo del proyecto. Se seleccionan los lenguajes en que estará hecha la aplicación.

Entradas:

- ✓ Análisis de requerimientos.
- ✓ ED.

Tareas:

- ✓ Realizar un estudio profundo de los diferentes lenguajes de programación que se ajustan a lo señalado en el análisis de requerimientos.
- ✓ Seleccionar los lenguajes de programación candidatos para el desarrollo.
- ✓ Realizar comparaciones entre los lenguajes de programación candidatos teniendo en cuenta los siguientes aspectos:
 - Propuesta del cliente.
 - Velocidad de la ejecución del programa.
 - Calidad de los instrumentos disponibles.
 - Popularidad del lenguaje.
 - Potencia.
 - Multiplataforma
 - Soporta programación Orientada a Objetos, en caso de que sea necesario.
 - Facilidad de secuencias de comandos web del lado del cliente.
 - Facilidad de secuencias de comandos del servidor web.
 - Uso en proyectos similares por otras empresas.
 - Rapidez para aprender y desarrollar.
- ✓ Determinar los lenguajes de programación que serán usados durante el desarrollo, teniendo en cuenta los criterios planteados anteriormente.

Artefacto: Entorno de desarrollo.

Además del SO, el ED ahora tiene los lenguajes de programación definidos.

Responsable: Responsable del ED, Arquitecto de Infraestructura.

4. Definición de herramientas y tecnologías.

El objetivo de esta actividad es seleccionar un conjunto de herramientas necesarias para el desarrollo de la aplicación, se definen IDEs, editores de texto, SGBD (Sistema Gestor de Bases de Datos), entre otras herramientas. Una buena selección proporcionará un desarrollo más ágil.

Entradas:

- ✓ Análisis de requerimientos
- ✓ Entorno de Desarrollo (con el SO y el lenguaje de programación seleccionados).

Tareas:

- ✓ Realizar un estudio de los diferentes IDEs de programación que soportan el lenguaje seleccionado.
- ✓ Seleccionar los diferentes IDEs que son candidatos para usarse en el desarrollo del proyecto.
- ✓ Para la selección del IDE o los IDEs para el desarrollo, es importante tener en cuenta los siguientes aspectos:
 - **Ejecutar lenguaje de programación:** El IDE a seleccionar debe ser capaz de ejecutar los programas desarrollados en el lenguaje definido, esto ayuda mucho en el desarrollo, ya que no hay necesidad de usar otras herramientas para la ejecución.
 - **Auto-completamiento de código:** Una de las características fundamentales que se prefiere en la actualidad es esta, dado que permite un desarrollo más rápido y muchas veces ayuda en la solución de problemas.
 - **Debugger:** Muchas veces es fundamental para la solución de errores y problemas que presentan los programas desarrollados.
 - **Soporte para subversion:** Capacidad de integrarse con servidores Subversion, permite la realización de operaciones directas a un servidor de este tipo.
 - **Resaltado de sintaxis:** Diferencia de colores entre la diferentes instrucciones que son del lenguaje de programación y las palabras comunes que se utilizan, así como la declaración de variables, el uso de funciones y otros aspectos más.
 - **Informe de errores:** Este es un elemento fundamental, ya que cuando se está programando si se presenta un error de sintaxis, no hay necesidad de ejecutar el código para que este sea detectado.
 - **Multiplataforma:** Se puede usar en varias plataformas como son: Unix, Mac OS, Windows; y que no haya necesidad de cambiar código en los programas desarrollados cuando se extrapolen de una plataforma a otra.
- ✓ Realizar un estudio sobre los diferentes SGBD más usados en el mundo.
- ✓ Seleccionar el SGBD, para esto hay que tener en cuenta las siguientes características:

- **Abstracción de la información:** Los SGBD ahorran a los usuarios detalles acerca del almacenamiento físico de los datos. Da lo mismo si una base de datos ocupa uno o cientos de archivos, este hecho se hace transparente al usuario.
 - **Independencia:** La independencia de los datos consiste en la capacidad de modificar el esquema (físico o lógico) de una base de datos sin tener que realizar cambios en las aplicaciones que se sirven de ella.
 - **Consistencia:** En aquellos casos en los que no se ha logrado eliminar la redundancia, será necesario vigilar que aquella información que aparece repetida se actualice de forma coherente, es decir, que todos los datos repetidos se actualicen de forma simultánea.
 - **Seguridad:** La información almacenada en una base de datos puede llegar a tener un gran valor. Los SGBD deben garantizar que esta información se encuentre segura frente a usuarios malintencionados, que intenten leer información privilegiada; frente a ataques que deseen manipular o destruir la información; o simplemente ante las torpezas de algún usuario autorizado, pero inexperto.
 - **Respaldo:** Los SGBD deben proporcionar una forma eficiente de realizar copias de respaldo de la información almacenada en ellos, y de restaurar a partir de estas copias los datos que se hayan podido perder.
 - **Manejo de transacciones:** Una transacción es un programa que se ejecuta como una sola operación, esto quiere decir que el estado luego de una ejecución en la que se produce una falla, es el mismo que se obtendría si el programa no se hubiera ejecutado. Los SGBD proveen mecanismos para programar las modificaciones de los datos de una forma mucho más simple que si no se dispusiera de ellos.
 - **Tiempo de respuesta:** Lógicamente, es deseable minimizar el tiempo que el SGBD tarda en dar la información solicitada y en almacenar los cambios realizados.
- ✓ Realizar un estudio sobre los diferentes servidores web existentes.
 - ✓ El estudio que se propone realizar es sobre los servidores web de software.
 - ✓ Seleccionar los servidores web candidatos a ser usados en el proyecto.
 - ✓ Para realizar la selección del servidor web a usar, algunos aspectos que se deben tener en cuenta para seleccionar un servidor de este tipo son:

- **Licencia del software:** si se encuentra publicado bajo una licencia propietaria o libre, este es un factor importante ya que muchas veces no se cuenta con los recursos necesarios por lo que hay que recurrir a alternativas libres.
- **Uso:** cantidad de personas, grupos o instituciones que usan el servidor web, si es usado por un gran número de entidades, demuestra su aceptación en el mundo, teniendo en cuenta que su uso no sea debido a un monopolio.
- **Flexibilidad:** disposición de un sistema para la carga dinámica de módulos basado en plug-ins o módulos.
- **Multiplataforma:** capacidad para correr en más de una plataforma como pueden ser (Unix; Mac OS, Windows).
- **Conexiones seguras https:** si permite la realización de conexiones mediante el protocolo https, el cual brinda una mayor seguridad al intercambio de información que se efectúa.
- **Velocidad:** se requiere que el servidor sea rápido, eficiente, que sea capaz de responder varias peticiones a la vez con un alto grado de eficiencia, esto también está sujeto a los recursos de hardware.

Artefacto: Entorno de Desarrollo

Responsable: Responsable del ED.

5. Definir configuraciones básicas de las tecnologías y herramientas.

Para la utilización de las herramientas seleccionadas y para una mayor optimización de las mismas se debe tener un conjunto de configuraciones básicas estas a fin de que quienes las usen lo hagan de una manera fácil y eficiente. Esto es muy importante, porque muchas herramientas si no son configuradas de la manera correcta, pues simplemente no funcionan, o funcionan mal.

Entradas:

- ✓ Entorno de Desarrollo
- ✓ Análisis de requerimientos

Tareas:

- ✓ Estudiar diferentes aspectos referentes a los Sistemas Operativos seleccionados y definir configuraciones para los mismos, algunos aspectos son:
 - Configuración de firewall.
 - Configuración de los puertos del sistema.
 - Configuración de los accesos remotos a las computadoras.
- ✓ Definir configuraciones para cada una de las herramientas seleccionadas para el ED en caso necesario.

Artefacto: Entorno de Desarrollo.

Responsable: Responsable del ED.

6. Implementación de la plataforma de desarrollo

Uno de los principales objetivos de esta actividad es: crear una configuración de escritorio estándar para toda la organización que se base en una imagen común para cada versión del SO de la organización.

Una vez definido el SO, las herramientas y las tecnologías a usar, todos con sus respectivas configuraciones; se procede a implementar la Plataforma de Desarrollo; esta puede presentarse por una biblioteca o repositorio de herramientas, SO y librerías útiles para el proyecto. A estas tendrían acceso los miembros del proyecto para preparar sus entornos de trabajo. Esto posibilita tener instaladas sólo las herramientas que necesitan, evitando tener aplicaciones instaladas que no son necesarias para las tareas que se deben realizar.

Entradas:

- ✓ Entorno de Desarrollo

Tareas:

- ✓ Instalar y configurar los SO seleccionados, teniendo en cuenta que dicha tarea es realizada para la creación de imágenes estándares que se usarán en la mayoría de las computadoras de la organización.
- ✓ Instalar sobre dicho SO todas las herramientas definidas en el ED.
- ✓ Enumerar cualquier tecnología para la creación de imágenes de disco y las imágenes estándares usadas por la organización.

- ✓ Analizar cómo las tecnologías de creación de imágenes disponibles se usan para personalizar, compilar, capturar e implementar imágenes de disco para los sistemas operativos de equipos de escritorio.
- ✓ Seleccionar herramientas para la creación de imágenes de disco.
- ✓ Determinar todos los escenarios de configuración en los que debe funcionar la imagen de disco.
- ✓ Crear y probar las imágenes que serán implementadas en las estaciones de trabajo, para detectar errores que puedan ser corregidos antes de implantarse en los equipos clientes.
- ✓ Implementar los sistemas operativos de destino a través de la red mediante una carpeta compartida o de forma local usando un medio de almacenamiento extraíble, como un CD, DVD, disco duro USB u otro dispositivo. Los parámetros de configuración para cada equipo individual suelen proporcionarse manualmente durante el proceso de implementación.

Artefacto: Plataforma de Desarrollo.

La Plataforma de Desarrollo se presenta por una imagen con el SO instalado y este a su vez posee todas las herramientas y lenguajes de programación instalados, configurados y listos para usarse.

Responsable: Responsable del ED.

7. Definición de políticas de trabajo y listas de verificación para garantizar el soporte y el buen funcionamiento de la plataforma.

Se define un conjunto de políticas para el trabajo sobre la plataforma, y para el mantenimiento de la misma; se define cómo se realizan las actualizaciones, las salvadas de seguridad y los cambios. También se determina un conjunto de normas para el uso de las estaciones de trabajo y para el mantenimiento de las mismas en cuanto a las actualizaciones que se realizan sobre las herramientas que se están usando. Esto permite un trabajo organizado y disciplinado, que mejora la gestión del proyecto, el desarrollo y el aprovechamiento de los recursos. En caso de pérdida de datos, no se provocarán grandes retrasos, ni habrá problemas para ponerse al día con los datos perdidos.

Las listas de verificación son otro elemento a tener en cuenta, ya que al realizar chequeos sobre los servidores y servicios de los que se disponen, se tendrán parámetros a medir, por lo que la evaluación se realizará de una manera organizada, efectiva y objetiva.

La estrategia que se usará para el soporte es fundamental, ya que debe proporcionar un conjunto de métodos y actividades que mantendrán actualizados los ED y las estaciones de trabajo, de forma tal que se mantengan las tecnologías al día y existan pocos atrasos por errores y fallas.

Entradas:

- ✓ Plataforma de Desarrollo.

Tareas:

- ✓ Definir políticas de trabajo

Estas políticas ayudarán al mantenimiento de la plataforma de desarrollo. Es importante tener en cuenta acciones como:

- Analizar la aplicabilidad en la organización de las actualizaciones del SO, de los controladores de dispositivo y aplicaciones de software.
- Integrar las actualizaciones antes mencionadas a la imagen que define la plataforma de desarrollo o crear imágenes completamente nuevas.
- Probar y validar las imágenes finales antes de poder implementarlas en equipos cliente.

Artefactos: Políticas de trabajo

Define un conjunto de normas para las actividades que se realizan referentes a la Plataforma y ED.

Responsable: Responsable del ED.

2.1.2 Plataforma de Servicios

La PS pone a disposición un conjunto de servicios que ayudan a mejorar el desarrollo del proyecto, teniendo en cuenta los requerimientos de rendimiento, escalabilidad, seguridad, y alta disponibilidad para satisfacer la calidad de servicio requerida, evitando riesgos potenciales en la producción, mantenimiento y evolución.

2.1.2.1 Actividades Fundamentales de la Plataforma de Servicios

1. Definir módulos necesarios para la plataforma de servicios.

Se define un conjunto de módulos mediante los cuales se organizan los servicios telemáticos que ofrece la plataforma. Se deben tener en cuenta las tecnologías en las que se desarrollan las diferentes aplicaciones

garantizando la compatibilidad entre estas además del comportamiento habitual en cuanto a consumo de recursos, índice de uso por parte de los usuarios, entre otros.

El objetivo de esta actividad es separar los servicios necesarios para la organización por módulos, a fin de mejorar la administración de los mismos, tener un mayor control y organización y realizar un mejor aprovechamiento de los recursos con que cuenta la organización.

Tareas:

- ✓ Identificar módulos para la PS. Estos agruparán un conjunto de servicios teniendo en cuenta su comportamiento, actividad a la que tributan, requerimientos de hardware y otros aspectos.
- ✓ Seleccionar un portafolio de servicios para cada módulo identificado. Se seleccionan los servicios necesarios para satisfacer las necesidades de cada módulo.
- ✓ Seleccionar un conjunto de servicios que cumplirán con las necesidades del módulo en cuestión.

A continuación se definen cuatro módulos a tener en cuenta para la realización de esta tarea, los mismos pueden cambiar teniendo en cuenta las necesidades de la organización:

Módulo de producción:

Se necesita contar con un sistema de control de versiones para el trabajo colaborativo, lo cual ofrece un desarrollo más rápido. También se hace necesario un servidor de datos, el cual tendrá hospedado un gestor de bases de datos configurado adecuadamente con la seguridad y los niveles de acceso requeridos. Se necesita disponer de un servidor web en el caso de estarse desarrollando aplicaciones web. Un servidor de aplicaciones es otro de los servicios que se debe tener en cuenta.

Todos estos servicios deben contar con un alto nivel de disponibilidad dado que de ello depende que el desarrollo del proyecto no se atrase o se detenga.

Módulo de gestión:

Es necesario contar con determinados servicios para agilizar la gestión de tareas, errores, recursos y otros tipos de peticiones que tienen lugar durante el desarrollo y permite una gestión de proyecto controlada y disciplinada. Son necesarios sistemas de gestión documental, sistemas de gestión de proyectos. Son de gran utilidad además los servicios de correo y mensajería instantánea: Jabber e IRC (Internet Relay Chat). Estos son protocolos de comunicación en tiempo real basados en texto, que permiten debates entre dos o

más personas. Son muy importantes para el intercambio de información y conocimiento entre los desarrolladores. Si las condiciones lo permiten se puede contar además con servicios de VoIP (Voice Over IP): Servicio de telefonía IP que permite realizar llamadas telefónicas a través de Internet, esto permitirá una mejor comunicación entre los miembros del equipo de desarrollo. Se debe realizar un estudio profundo de las herramientas que se ajusten mejor a las necesidades para poder realizar todas las tareas necesarias para una buena gestión.

Módulo de investigación y superación:

Promover la investigación y la superación de los profesionales y otros miembros del equipo de desarrollo. Es esencial para garantizar que el trabajo tenga la calidad requerida. Para contribuir a originar la investigación existen varios servicios; pero ¿cómo lograr esto de una manera que todos puedan estar inmersos? Una de las estrategias que se ha utilizado es el uso de foros de discusión, blogs y wikis, entre otros, para abordar temas de interés, realizar convocatorias, impulsar la innovación y el desarrollo colaborativo. Esto crea una mayor interacción entre los miembros del equipo.

Módulo de monitoreo y control:

Abarca un conjunto de técnicas pendientes a mantener redes, servidores, y otros dispositivos, monitorizados periódicamente y con una planeación adecuada, de forma tal que se lleve un control estricto sobre el comportamiento y la evolución de los mismos. Tienen el objetivo de darle una continuidad adecuada a mecanismos de control y monitoreo, resolución de problemas y de suministro de recursos, hacer uso eficiente de la red y utilizar mejor los recursos, como por ejemplo: el ancho de banda; hacer la red más segura protegiéndola contra el acceso no autorizado; controlar cambios y actualizaciones de modo que se ocasionen la menor cantidad de interrupciones posibles en el servicio a los usuarios. Es necesario contar con un conjunto de aplicaciones bien configuradas que son un aliado indispensable para un Arquitecto de Infraestructura. Se necesitan aplicaciones que permitan emitir reportes de estado y uso de los servicios, consumo de ancho de banda, además de controlar el tráfico en un determinado espacio de red. Estos servicios ayudan a tomar decisiones tempranas en cuanto al escalado de la plataforma y otros.

La memoria RAM y los procesadores son otros dos aspectos que hay que tener en cuenta cuando se realiza el monitoreo, ayudando esto a mejorar los servicios y a no sobrecargar los servidores. En el caso

de los servidores dedicados al almacenamiento es fundamental mantener el control mediante sistemas de alerta, y mecanismos que permitan tener reportes actualizados sobre el espacio disponible en discos, garantizando la toma de decisiones y la aplicación de medidas proactivas para evitar complicaciones mayores que influyan en la calidad de las actividades del equipo.

Se hace necesaria la selección de herramientas de monitoreo adecuadas, estas deben ser bien escogidas para que se puedan obtener, a través de reportes emitidos, los datos necesarios para ejercer el control y en la manera más legible posible, permitiendo emitir informes coherentes respecto al comportamiento de la red y otros recursos.

Otro aspecto fundamental es seleccionar herramientas que sean capaces de emitir avisos y notificaciones en caso de la ocurrencia de alteraciones en las aplicaciones, así como en los registros de logs de las mismas. Así como garantizar la revisión de los logs de las aplicaciones periódicamente, con el propósito de adelantarse a las fallas que las aplicaciones no detecten.

Artefacto: Diseño de Plataforma de Servicios, Diseño Estructural de Tecnología.

El diseño de la PS es el diseño telemático de la misma.

Responsable: arquitecto principal de Infraestructura, responsable de la PS.

2. Definir políticas de trabajo y listas de verificación.

Se definen las políticas para el mantenimiento de la PS. Cómo se realizarán las salvadas, las actualizaciones de los servicios, la revisión de los mismos y cada qué tiempo se efectuará, además de identificar cuáles serán los parámetros a medir en cada evaluación que se realice.

Entradas:

- ✓ Diseño de Plataforma de Servicios.

Tareas:

- ✓ Definir políticas de trabajo. Para ello es importante tener en cuenta las siguientes acciones:
 - Crear un plan de respuesta ante incidentes.
 - Crear un plan de copia de seguridad y recuperación para los servidores críticos de la

organización, teniendo en cuenta:

- Modos de copia de seguridad (en línea, sin conexión).
- Tipos de copia de seguridad (completa, incremental, diferencial).
- Supervisar y administrar los recursos de administración del almacenamiento para copia de seguridad y recuperación.
- Crear listas de verificación para la evaluación de los servidores, estas deben variar según el funcionamiento de los mismos.
- Definir un período de tiempo mediante el cual se revisarán los reportes emitidos por el monitoreo de la red y de los servidores, para detectar errores y evaluar los servicios haciendo uso de listas de verificación.

Artefacto: Políticas de trabajo para el mantenimiento de la Plataforma de Servicios.

Responsable: Responsable de la PS.

3. Monitorizar y supervisar los servicios y aplicaciones durante todo el proceso de desarrollo.

Realizar un monitoreo de todos los equipos, los servicios y los servidores, de todo el proyecto en general, durante la etapa de desarrollo. Esta es una actividad fundamental que posibilita detectar errores rápidamente, por lo que la falla que pueda presentarse durará poco, y posibilitará que el desarrollo del proyecto siga funcionando correctamente. Hay que mantener un control estricto sobre el comportamiento del ancho de banda, los procesadores, el consumo de memoria RAM y también sobre la capacidad de almacenamiento de los dispositivos, para mantener una alta disponibilidad, fundamental para el desarrollo.

Tareas:

- ✓ Realizar un inventario de todos los servidores de la Infraestructura de la organización.
- ✓ Establecer qué se necesita supervisar y derivar un modelo de estado que defina lo que significa para un sistema o servicio ser correcto (funcionar en condiciones normales) o ser incorrecto (presentar errores o rendimiento reducido).
- ✓ Supervisar la disponibilidad de los servicios, para esto se debe seleccionar el software necesario teniendo en cuenta los siguientes aspectos:
 - Capacidad de recopilar la información de atributos del servidor y aplicar reglas específicas para supervisar en función de los atributos definidos.

- Capacidad para obtener datos de registros de eventos y otros proveedores de la manera definida por algunas reglas específicas.
- Capacidad para recopilar datos de rendimiento basados en contadores de rendimiento.
- Capacidad para generar alertas basadas en criterios especificados en reglas.
- ✓ Cuantificar, evaluar y mantener un alto nivel del servicio de TI haciendo uso de los datos de supervisión, este nivel de servicio se basa en:
 - Disponibilidad: supervisar la disponibilidad de los servidores mediante la comunicación con los mismos para asegurarse de que estén ejecutándose.
 - Rendimiento: supervisar los contadores de rendimiento para asegurarse de que los servidores estén ejecutándose dentro de unos parámetros aceptables.
 - Capacidad: supervisar la capacidad del disco y realizar análisis y planeación de la capacidad.
 - Reconocimiento de errores: identificar errores o condiciones que repercutan en los tres aspectos anteriores de los niveles de servicio.
- ✓ Realizar un control de las actividades de administración de disponibilidad para los servidores críticos.
- ✓ Verificar que los servicios son usados por los miembros de la organización.
- ✓ Emitir informes sobre el estado de todos los servicios de la organización y el uso de los mismos.
- ✓ Monitorear y evaluar el consumo de ancho de banda.

Artefacto: Informes de estado de los servicios y aplicaciones, Informe de uso de los servicios, Informe sobre el comportamiento y uso de la red.

Responsable: Responsable de la Plataforma de Servicios.

2.1.3 Despliegue

El despliegue es otro de los temas importantes en un proyecto, pues en esta fase es donde se pone en funcionamiento el proyecto desarrollado. Es importante primero diseñar escenarios de despliegue a los que se le realicen un conjunto de pruebas para asegurar que funcionan correctamente, por lo que hay que realizar pruebas de cargas y estrés, simular que el proyecto ya está funcionando como si fuera real y con todas las condiciones que necesita el mismo y las configuraciones realizadas.

Es necesario definir una estrategia de soporte al mismo, dado que este debe ser mantenido tecnológicamente y se le debe dar un seguimiento después de puesta en funcionamiento la aplicación. Además, hay que realizar actualizaciones del sistema en los diferentes despliegues realizados, por lo que se tienen que diseñar políticas y normas para la realización de estas tareas. Es fundamental que esto ocurra de una manera eficiente y sin problemas, dado que es la parte donde el cliente ya interactúa con su producto y de ello depende en gran medida el prestigio del equipo.

Otro aspecto fundamental a tener en cuenta en el despliegue es el crecimiento de las bases de datos. Un sistema de almacenamiento mal gestionado puede dar al traste con el buen funcionamiento del producto, por tal motivo es necesario realizar estimaciones lo más precisas posibles en cuanto al tamaño y el posible crecimiento de los datos y a la capacidad de almacenamiento. Actualmente existen metodologías o métodos empíricos bastante precisos para estimar el crecimiento de las bases de datos, sin embargo, el tema no es tan simple, es necesario realizar un estudio profundo de la base de datos en cuestión así como el tipo de aplicación, de cliente y la cantidad de estos que interactúan con el sistema para tener información más confiable.

En esta actividad se hace indispensable la participación de un arquitecto de datos u otro miembro del equipo de desarrollo que conozca detalladamente el modelo de datos y la base de datos en sí, además de arquitectos de sistema y analistas que tengan detalles del negocio.

2.1.3.1 Actividades fundamentales del Despliegue

1. Realizar pruebas de rendimiento, carga, estrés y tiempo de respuesta a las aplicaciones desarrolladas para definir escenarios de despliegue.

Las pruebas de rendimiento se realizan tanto para determinar cómo responde un sistema ante cierta carga, para validar otros atributos relacionados con la calidad, como pueden ser la escalabilidad, el uso de los recursos, entre otros. Estas pruebas pueden tener distintos propósitos, por ejemplo, pueden demostrar que el sistema cumple los criterios de rendimiento, puede medir qué partes del sistema o qué carga hacen que el sistema rinda de forma incorrecta.

Existen distintos tipos de pruebas de rendimiento que ayudan a mejorar las capacidades la aplicación maximizando el beneficio del negocio, estas pueden ser:

Pruebas de carga: se realizan para comprender el comportamiento de una aplicación ante una carga determinada. Esta carga puede ser el número de usuarios esperados ejecutando un número de transacciones durante un tiempo determinado. El resultado de este test dará el tiempo de respuesta de todas las transacciones críticas.

Pruebas de estrés: estas pruebas son utilizadas normalmente para someter la aplicación al límite de su funcionamiento, mediante la ejecución de un número de usuarios muy superior al esperado. Este test tiene como finalidad, determinar la robustez de una aplicación cuando la carga es extrema y ayuda a determinar si la aplicación se comportará correctamente en dichas situaciones.

Pruebas de resistencia: este test se realiza con el fin de determinar si la aplicación puede mantener la carga esperada de manera continua y durante un largo tiempo.

Tareas:

- ✓ Seleccionar software para la realización de pruebas de carga, estrés y tiempo de respuesta de las aplicaciones desarrolladas.
- ✓ Realizar dichas pruebas a las aplicaciones después de seleccionado el software para dicha tarea.
- ✓ Detectar errores y anomalías en las aplicaciones.
- ✓ Definir, a partir de los resultados obtenidos de las pruebas realizadas, los Escenarios de Despliegue, de forma tal que estos ofrezcan rendimiento, escalabilidad y disponibilidad ante las transacciones que se realizarán en los entornos de producción reales.

Artefacto: Escenarios de despliegue, Informe de resultados de pruebas.

Responsable: diseñador de despliegue, desarrolladores.

2. Diseño de Centro de Datos.

La mayoría de los Centros de Datos ofrecen: acceso a Internet, hosting de aplicaciones, distribución de contenidos, almacenamiento de datos y backup, administración de bases de datos, encendido a prueba de fallos, infraestructura de cableado de alto desempeño y seguridad. El diseño de un Centro de Datos depende en gran medida de los recursos con los que cuenta la organización dado que estos son bastante costosos debido a los requerimientos que deben tener los mismos, uno de los principales es la disponibilidad y deben estar siempre disponibles. Aquí se concentran todos los recursos necesarios para

el procesamiento de la información en la organización. Generalmente todos estos recursos consisten en computadoras y redes de comunicaciones.

Entradas:

- ✓ Informe de resultados de pruebas.
- ✓ Escenarios de Despliegue.

Tareas:

- ✓ Realizar un estudio detallado de la tecnología que se va a usar.
- ✓ Analizar todos los detalles tecnológicos (red, clúster de datos, clúster web, tipos de servidores, entre otros) y las herramientas a instalar.
- ✓ Definir los servicios y las configuraciones elementales y necesarias para el funcionamiento de las aplicaciones.
- ✓ Realizar un balance de los recursos disponibles, de la densidad inicial de usuarios y de los servicios que serán hospedados.
- ✓ Verificar si existe una macro-infraestructura en la organización, dado que si existe una, es muy probable que el Centro de Datos pueda ser integrado con la misma y utilice servicios de mensajería y todos los servicios de red adicionales como DNS, servidores de ficheros y sistemas de almacenamiento implementados.
- ✓ Realizar un análisis de los siguientes elementos que son de esencial importancia para garantizar el funcionamiento de las aplicaciones desarrolladas:
 - **Servidores de datos:** si es necesario se implementan clústeres de mediano o alto rendimiento, pero en la mayoría de los casos con un servidor de datos es suficiente, aquí influye mucho la densidad de usuarios y el tipo de aplicación.
 - **Servidores web:** en dependencia de los recursos de los que se dispone, al igual que sucede con los servidores de datos, se pueden implementar servicios redundantes para garantizar disponibilidad.
 - **Sistema de almacenamiento:** es esencial contar con un sistema de almacenamiento correctamente configurado y asegurado. Este sistema, abarca otros subsistemas o módulos como son las salvadas automáticas y la recuperación de información, de ahí su importancia.

- ✓ Realizar un diseño lógico del Centro de Datos, el cual debe estar bien encaminado para obtener una buena seguridad. Algunas actuaciones para esto son:
 - Creación de Zonas Desmilitarizadas (DMZ).
 - Segmentación de redes locales y creación de Redes Virtuales (VLAN).
 - Despliegue y configuración de la electrónica de red: pasarela, conmutadores, encaminadores.
 - Creación de los entornos de explotación, desarrollo de aplicaciones y gestión de red.
 - Separación de los servicios por servidores.
 - Instalación de los servidores y periféricos.
 - Creación de una estructura para la salva y recuperación de datos.
 - Usar equipamiento confiable y fundamentalmente sistemas redundantes que garanticen la disponibilidad en un alto grado. Esto depende en gran medida de los recursos con que se cuente, y de las necesidades reales del sistema y del cliente.

Artefacto: Diseño de Centro de Datos.

Responsable: arquitecto de infraestructura, responsable de despliegue, desarrolladores.

Conclusiones

Con el desarrollo de este capítulo se obtiene un conjunto de pautas que permiten diseñar la AI para un software de gestión. La propuesta se divide en tres áreas de trabajo fundamentales; la misma proporciona un mayor desempeño en las actividades que se realizan en esta disciplina de la Arquitectura y ayuda en la gestión de la IT de una manera más organizada.

CAPÍTULO 3 VISTA DE INFRAESTRUCTURA DEL PROYECTO ERP-CUBA

En el presente capítulo se crea la Vista de AI del proyecto ERP-Cuba de manera organizada siguiendo las pautas definidas para diseñar una IT de forma correcta, validando de esta forma la propuesta realizada en el Capítulo 2. La implementación de la Infraestructura del proyecto ERP-Cuba garantiza el uso de tecnología licenciada bajo software libre, la creación de Entornos de Desarrollo proporciona todas las herramientas necesarias para desarrollar el proyecto, además, la PS ofrece recursos necesarios para garantizar el funcionamiento de los procesos dentro del proyecto y una buena gestión del mismo. También propone un diseño básico del Centro de Datos, con los aspectos elementales a tener en cuenta para el buen funcionamiento de las aplicaciones desplegadas.

El proyecto ERP-Cuba trabaja en conjunto con el UCID (Unidad de Compatibilización, Integración y Desarrollo), este último posee varias herramientas definidas para el desarrollo que, para evitar incompatibilidades se mantienen en la IT del proyecto ERP-Cuba. Estas herramientas fueron estudiadas y se consideran adecuadas para continuar usando las mismas.

3.1 Entorno de desarrollo

Para la selección de herramientas en esta área se tienen en cuenta un conjunto de tecnologías que ya están definidas en el UCID, como son: lenguajes de programación, servidor de subversión, clientes de datos, clientes web, clientes de servidor de versiones y los IDEs. De todas estas herramientas se realiza una descripción. Además, existe otro conjunto de herramientas que fue necesario incorporar en esta área de trabajo, para la selección de las mismas se tuvo en cuenta algunos criterios de comparación con respecto a otras tecnologías similares.

3.1.1 Análisis de los requerimientos tecnológicos

El proyecto ERP-Cuba brinda una solución nacional basada en software libre, el mismo se desarrolla en Cuba, este es un país con un embargo económico que le imposibilita realizar desarrollos con tecnologías propietarias, por lo tanto, requiere de tecnologías y herramientas libres y de código abierto siempre que sea posible.

Este proyecto desarrolla una aplicación web para diversos escenarios, ya sea con pocos o muchos usuarios. Se requiere procesamiento de información centralizada dado que se despliega por todo el país

con acceso a bases de datos específicas para cada uno de los ministerios o institutos. Dada la diversidad de escenarios puede manejar pequeños y grandes volúmenes de información; por tanto se hace necesario gestores de bases de datos potentes y con posibilidades de soporte y adaptación.

CEDRUX (aplicación web desarrollada por el ERP-Cuba), está pensado para que sea multiplataforma, debido a esto los lenguajes de programación y librerías también deben serlo. Posee un equipo de desarrollo grande, por tanto, requiere de potentes herramientas para el trabajo colaborativo. También existe una gran variedad de roles en el equipo lo que evidencia la necesidad de una alta gama de herramientas, tanto de modelado como para el trabajo con documentos.

El proyecto ERP-Cuba desarrolla una aplicación de gestión para la web. Teniendo en cuenta esto, y todos los requerimientos mencionados anteriormente, se debe prestar especial atención a la selección de los recursos tecnológicos con los que se llevará a cabo el desarrollo y la implantación. Entre las principales tecnologías se encuentran:

- **SO a usar:** debe ser flexible, rápido, capaz de soportar diferentes tecnologías de hardware, configurable a las necesidades de la empresa; robusto, seguro y fiable.
- **Lenguajes de programación:** debe ofrecer una gran rapidez de ejecución, además de ser robusto y potente.
- **Servidor web:** debe transferir hipertextos, páginas web o páginas HTML (HyperText Markup Language): textos complejos con enlaces, figuras, formularios, botones y objetos incrustados como animaciones o reproductores de música.
- **Servidor de datos:** debe ser robusto y tener alta seguridad, con el fin de salvaguardar los datos de la entidad.

3.1.2 Selección de Sistema Operativo

Ubuntu 10.04

Es el SO basado en Linux más usado en la UCI. Contiene un conjunto de aplicaciones necesarias tanto para desarrollar como para un servidor, posee: navegador web, suite de oficina, aplicaciones multimedia, mensajería instantánea, IDEs de desarrollo, SGBD, servidores de correo. Es libre y contiene una gama de opciones de soporte gratuito que se encuentran disponibles en la Comunidad Ubuntu, incluyendo foros, canales de IRC y listas de correo.

Se obtienen actualizaciones de seguridad para el escritorio y el servidor, está disponible para todos bajo los mismos términos libres, las mismas hacia nuevas versiones de Ubuntu son gratis. Es un SO veloz, los programas funcionan con rapidez al igual que la realización de las tareas. [5]

3.1.3 Definición de los lenguajes de programación

PHP 5.2

Es un lenguaje script procesado en el lado del servidor, el mismo se muestra como código embebido dentro de una página HTML. Posee una comunidad de desarrolladores que brinda soporte, como producto de código abierto, PHP goza de la ayuda de un gran grupo de programadores, permitiendo que los fallos de funcionamiento se encuentren y reparen rápidamente. Además es uno de los lenguajes más usados para la creación de aplicaciones web. [28]

JavaScript 1.8.1

JavaScript es un lenguaje con muchas posibilidades, utilizado para crear pequeños programas que luego son insertados en una página web y en programas más grandes. Con JavaScript se puede crear diferentes efectos e interactuar con los usuarios.

Este lenguaje posee varias características, entre ellas se puede mencionar que es un lenguaje basado en acciones que poseen menos restricciones. Además, es un lenguaje que utiliza Windows y sistemas X- Windows, gran parte de la programación en este lenguaje está centrada en describir objetos, escribir funciones que respondan a movimientos del mouse, aperturas, utilización de teclas, cargas de páginas, entre otros. [24]

Ext js 3.0

Ext es una librería JavaScript ligera y de alto rendimiento, compatible con la mayoría de navegadores que permite crear páginas e interfaces web dinámicas. [31]

Se utiliza para la creación de aplicaciones enriquecidas del lado del cliente. Sus características principales son: gran desempeño, componentes de interfaz de usuarios personalizables, además de buen diseño y documentación. [12]

3.1.4 Herramientas y tecnologías

PgAdmin3 1.10.2

Es una aplicación gráfica para administrar bases de datos en PostgreSQL, esta es la más popular con licencia Open Source. Está escrita en C++ usando la librería gráfica multiplataforma wxWidgets, lo que permite que se pueda usar en Linux, FreeBSD, Solaris, Mac OS X y Windows. Está diseñado para responder a las necesidades de todos los usuarios, permite escribir consultas SQL simples, así como desarrollar bases de datos complejas. La interfaz gráfica soporta todas las características de PostgreSQL, y facilita la administración. La aplicación también incluye un editor SQL con resaltado de sintaxis, un editor de código de la parte del servidor y un agente para lanzar scripts programados. La conexión al servidor puede hacerse mediante conexión TCP/IP o Unix Domain Sockets (en plataformas Unix), y puede encriptarse mediante SSL para mayor seguridad. [39]

OpenOffice 3.5

OpenOffice es un paquete de Ofimática de software libre. Son muchas las actividades que se pueden realizar con él, gracias a todo el soporte que posee y a las grandes potencialidades que presenta.

Entre sus principales características se pueden listar las siguientes:

- **Soporte Web 2.0 con componentes Wiki:** se puede editar blogs de Wordpress y Movable Type y Wikis directamente desde el programa.
- **Soporte MS Office XML:** estable y completo.
- **Calendario para Thunderbird:** con manejo de tareas, vistas de semana y días, control de tiempos libres y ocupados.
- **Reemplazo de Outlook:** una nueva aplicación que integra el correo electrónico con calendario (con invitaciones a reuniones y publicación de horarios libres y ocupados), gestión de tareas y conectividad con servidores CalDAV, Google Calendar, Sun Calendar y soporte iCal. [40]

Visual Paradigm 3.0

Visual Paradigm es una herramienta CASE empleada para visualizar y diseñar elementos de software, para ello utiliza el lenguaje UML y proporciona a los desarrolladores una plataforma que les permite diseñar un producto con calidad de forma rápida. Facilita la interoperabilidad con otras herramientas CASE como Rational Rose y se integra con diversos IDEs como: Netbeans y Eclipse. Está disponible en varias ediciones: Enterprise, Professional, Community, Standard, Modeler y Personal. Genera código y

realiza ingeniería inversa para diferentes lenguajes de programación como: Java, C++, PHP y ADA, entre otros. Además, exporta e importa los diagramas en el estándar XML.

Características:

- Entorno de creación de diagramas para UML 2.0.
- Disponibilidad de múltiples plataformas.
- Disponibilidad de integrarse en los principales IDEs. [32]

Netbeans 6.8

Netbeans es una gran herramienta para construir aplicaciones web y para desarrolladores de software. Es un IDE que está disponible como código de software abierto. Una de sus mejores características es que se puede ejecutar en la mayoría de los SO como son Windows, Linux, Mac OS X y Solaris.

Con él se pueden desarrollar aplicaciones usadas a través del escritorio o la web. Las aplicaciones pueden ser desarrolladas en diferentes lenguajes de programación, como son, Java, Ruby (con soporte para Rails), C y C++. Posee muchas capacidades visuales útiles que sirven tanto para el desarrollo de las aplicaciones como para la realización de pruebas y corrección de errores. [26]

ZendStudio 5.5.0

Es usado por muchos desarrolladores, se considera uno de los mejores IDE actualmente. Fue desarrollado por uno de los mayores impulsores de PHP: Zend, orientado a desarrollar aplicaciones web en PHP para las que proporciona un buen número de ayudas desde la creación y gestión de proyectos hasta la depuración del código. [42]

Entre sus principales características se encuentra:

- Fácil de instalar.
- Brinda Soporte técnico.
- Posee Soporte para PHP 4 y PHP 5.
- Detección de errores en tiempo real.
- Soporte para JavaScript, HTML y CSS.
- Comunicación SSL.

- Depuración de scripts PHP.
- Mecanismos de depuración rápida.
- Conectividad a bases de datos con integración de controladores JDBC.
- Subversion (SVN). [25]

Eclipse 3.5.2

Eclipse es un IDE de código abierto el cual es utilizado entre los desarrolladores de Java debido a las herramientas de desarrollo (JDT) que posee y su arquitectura brinda una alta extensibilidad para plugin; el mismo se integra a diferentes plataformas.

Ventajas:

- Emplea plugins para proporcionar todas sus funcionalidades como plataforma.
- Dispone de un editor de texto con resaltado de sintaxis.
- La compilación del código es en tiempo real. Tiene pruebas unitarias con JUnit.
- Control de versiones con CVS (Control Version System o Sistema de Control de Versiones).
- A través de plugins libremente disponibles es posible añadir control de versiones con Subversion e integración con Hibernate.

Firefox 3.0

Es un navegador web de la fundación Mozilla que se caracteriza por su navegación por pestañas. Es libre y de código abierto, siendo un software multiplataforma, disponible para Microsoft Windows, Mac y GNU/Linux.

Entre sus principales características están:

- Ofrece control sobre las páginas que están siendo visitadas, bloqueando las ventanas emergentes.
- Posee restauración de sesión.
- Navegación por pestañas.
- Este navegador no permite que una página web descargue, instale, o ejecute programas en el ordenador sin el consentimiento explícito.

- Posee marcadores dinámicos y soporte para revisar canales RSS (Really Simple Syndication: estructura de datos XML para syndicar noticias). [38]

RapidSvn 0.12

RapidSvn es una interfaz para el sistema de revisión Subversión escrito en C++ usando el framework wxWidgets. Está liberado bajo la Licencia Pública General de GNU v3. Entre sus principales características se encuentran:

- **Simple:** brinda una interfaz fácil de usar.
- **Eficiente:** es simple para los principiantes, pero lo suficientemente flexible para aumentar la productividad de los usuarios experimentados de subversion.
- **Portátil:** funciona en cualquier plataforma en la que Subversión y wxWitdgets se puede ejecutar: Linux, Windows, Mac OS X y Solaris, etc.
- **Rápido:** completamente escrito en C++. [17]

KdeSvn 1.5.3

KdeSvn es un cliente de Subversion VCS, posee una interfaz simple de usar; usa la API de desarrollo de subversion, en lugar de analizar los resultados de la herramienta de la línea de comandos como la mayoría de los otros clientes como RapidSvn. Provee una buena integración con el escritorio de KDE. [16]

3.1.5 Definición de políticas de trabajo y listas de verificación

Políticas de trabajo

Para mantener la correcta atención y seguimiento del ED, es necesario:

- Mantener actualizados los documentos de definición del ED.
- Mantener actualizados los documentos de la Infraestructura concernientes a los recursos de hardware y tecnologías software definidas.
- Mantener copias de seguridad de la documentación generada.
- Hacer imágenes periódicamente de los SO con las herramientas usadas y funcionando.
- Documentar las configuraciones para optimizar y calibrar los SO usados.
- Probar las actualizaciones de las aplicaciones y analizar si dichas aplicaciones serán actualizadas en la plataforma de desarrollo.

- Actualizar las imágenes con nuevas versiones de los programas en caso de que se requiera o que se decida hacerlo e instalar nuevos programas si es necesario.
- Actualizar las estaciones de trabajo con las nuevas imágenes.

Lista de verificación

La lista de verificación se elaboró teniendo en cuenta las siguientes preguntas y dando a ellas diversos pesos en dependencia de las respuestas de los encuestados en cada caso:

- ¿Existe el documento Entorno de Desarrollo?
- ¿Se realiza una descripción general del Entorno de Desarrollo?
- ¿Existen políticas de trabajo que permitan brindar soporte al Entorno de Desarrollo?
- ¿Se encuentran actualizadas las imágenes teniendo en cuenta las necesidades de la organización?
- ¿Se encuentran actualizadas las estaciones de trabajo a partir de las imágenes?
- ¿Las imágenes cuentan con todas las aplicaciones necesarias para el desarrollo?
- ¿Existen salvadas de la documentación y de las imágenes?

3.2 Plataforma de Servicios

La PS está compuesta por diferentes módulos, siguiendo el modelo de construcción en bloques, para cada uno de estos módulos se define un portafolio de servicios agrupados por el tipo de actividad a la que tributan. Los servicios se encuentran organizados teniendo en cuenta compatibilidades entre las tecnologías sobre las cuales están implementados los servicios, recursos que consumen y comportamiento habitual además del uso promedio que le dan los usuarios, tratando de aprovechar al máximo los recursos hardware con los que se cuenta.

En esta área también se tienen en consideración un conjunto de herramientas ya definidas por el UCID para su uso.

3.2.1 Módulos de la plataforma de servicios

3.2.1.1 Módulo de producción

Agrupar un conjunto de servicios que apoyan el desarrollo de la aplicación y la entidad. Ofrece herramientas que brindan soporte al ED, por ejemplo se puede mencionar servidor web, servidor de datos y servidor de subversión.

Servidor FTP: Proftpd 1.3

El servicio de FTP es uno de los más viejos y utilizados para la transferencia de archivos, la mayoría de los servidores webs lo necesitan y lo utilizan de alguna u otra forma para actualizar sus páginas estáticas, subir imágenes o para permitir la descarga de archivos a sus visitantes. Es seguro, sumamente flexible, modular y fácil de configurar.

Entre todas las características que tiene, las más rescatables son: permite autenticar usuarios con casi cualquier cosa, se pueden utilizar servidores virtuales de FTP, se pueden tener múltiples servidores brindando servicio de FTP anónimo, es modular (lo que permite extender su funcionalidad ampliamente), un usuario con acceso por FTP únicamente no requiere de una configuración especial, y su código es libre, está licenciado bajo GPL. [8]

Servidor web: Apache 2.2

Es actualmente es uno de los servidores web más utilizado en el mundo. Su potencia, fiabilidad y sencillez de configuración lo hacen sin duda un fuerte rival para sus competidores.

En cuanto a las características que posee Apache se encuentran las siguientes: funciona sobre diferentes plataformas y SO, ofrece tecnología libre, es un servidor web configurable, capaz de extender su funcionalidad y la calidad de sus servicios, es muy sencillo ampliar las capacidades del servidor, actualmente existen muchos módulos que son adaptables, trabaja junto a varios lenguajes de programación como: PHP, Perl, Java, JSP (Java Server Pages) y otros lenguajes scripts que son usados en los sitios dinámicos. Permite la creación de log según las necesidades del administrador, de este modo se puede tener un mayor control de lo que sucede en el servidor. [7]

Servidor de Datos: PostgreSQL 8.3

Es un SGBD Objeto-Relacional de software libre. Tiene prácticamente todo lo que tienen los gestores comerciales haciendo de él una muy buena alternativa GPL (General Public License).

Posee una gran escalabilidad, haciéndolo idóneo para su uso en sitios web que posean un gran número de peticiones diarias. Posee características que lo convierten en uno de los SGBD de código abierto (gratuito y con código fuente disponible) más avanzado del mundo, tales como:

- Soporta distintos tipos de datos: además del soporte para los tipos base (int, string, long, char entre otros), también soporta datos de tipo fecha, monetarios, elementos gráficos, datos sobre redes (MAC, IP), cadenas de bits. También permite la creación de tipos propios.
- Incorpora una estructura de datos en arreglos y funciones de diversa índole: manejo de fechas, geométricas y orientadas a operaciones con redes.
- Permite la declaración de funciones así como la definición de disparadores.
- Soporta el uso de índices, reglas y vistas.
- Incluye herencia entre tablas.
- Permite la gestión de diferentes usuarios, como también los permisos asignados a cada uno de ellos. [23]

Control de versiones: Subversion 1.5

Subversion, también conocido como svn, es un sistema de control de versiones parecido a «Concurrent Versions System» (CVS). Posee soporte para Apache2 mediante WebDAV, y un exhaustivo manual documentado sobre el uso y despliegue de Subversión en los sistemas.

Esta herramienta para la comunidad Open Source es potente, flexible y eficaz. Entre las nuevas funciones de Subversion-1.5 se encuentran: renombrar, mover y copiar ficheros, estructura en árbol de directorios, ficheros metadatos que presentan una amplia serie de propiedades, posibilidad de hacer un seguimiento de las versiones y excelente gestión de ficheros binarios. [11]

3.2.1.2 Módulo de gestión

Este módulo está orientado a suplir las necesidades de gestión de la información dentro del proyecto, comprende servicios de gestión documental así como gestión de proyectos y Gestión de Activos y medios.

Algunas herramientas usadas en este módulo como: Redmine y Alfresco, se estaban probando en la Universidad y se comentaba acerca su uso definitivo, por lo que este fue un factor importante para la selección de las mismas. Haciendo un estudio y viendo que dichas herramientas satisfacían las necesidades del proyecto en cuanto a su uso, se decide implantarlas.

Sistema de gestión de proyectos: Redmine 0.8.

Es un servidor de gestión de proyectos plenamente desarrollado. Una opción interesante para compañías que manejan diversos proyectos a la vez y en las que varias personas están involucradas en el avance de los mismos. Plantea opciones muy útiles como la creación de usuarios con privilegios bien determinados, la gestión de múltiples proyectos a la vez, además se trabaja sobre una interfaz web, que se puede alojar en cualquier servidor de la empresa o incluso en el hosting de preferencia. Dado que se trabaja sobre una interfaz web Redmine se puede montar sobre distintas plataformas y también trabaja con varios motores para bases de datos. [19]

Sistema de gestión documental: Alfresco 3.0.

Es un sistema de administración de contenidos de código libre y abierto, basado en estándares abiertos y de escala empresarial. Está diseñado para usuarios que requieren un alto grado de modularidad y rendimiento escalable. Este incluye un repositorio de contenidos, un framework de portal web para administrar y usar contenido estándar en portales, una interfaz CIFS (Sistema de Archivo Común de Internet) que provee compatibilidad de sistemas de archivos en Windows y SO similares a Unix, un sistema de administración de contenido web, capacidad de visualizar aplicaciones web y sitios estáticos. Está desarrollado en Java.

Alfresco cuenta con disímiles características para una adecuada gestión documental y archivística, es versionado a nivel de repositorio (similar a Subversion), superposición transparente, cuenta con gestión de registros y gestión de imágenes, posee una publicación integrada, servicios descentralizados, soportes en varios idiomas, empaquetamiento de aplicación portable y soporte multiplataforma. [9]

Herramienta para el monitoreo de activos: GLPI

GLPI (Gestionnaire Libre de Parc Informatique o Gestión Libre de Parque Informático) es un administrador de recursos de información con una interfaz para su administración. Es utilizada para establecer bases de datos con un inventario de la empresa (PCs, impresoras, escáners, software).

Dispone de funciones que facilitan el trabajo de los administradores así como un sistema de seguimiento de trabajo con notificación por correo, además de poder establecer la topología de la red. Esta herramienta fue la encontrada para el control de activos. [34]

3.2.1.3 Módulo de investigación y superación

Este módulo ofrece diversas maneras de fomentar la superación cumpliendo con uno de los objetivos estratégicos de la organización, mediante el uso de portales y blogs.

Portales

Los servicios que comúnmente ofrecen los portales incluyen un directorio de sitios web, facilidades para la localización de sitios, noticias, información del clima, buzones de correo, información de la bolsa, directorios telefónicos, mapas y en algunas ocasiones foros de discusión con temas de interés general. [22]

El proyecto ERP-Cuba cuenta con un portal web desarrollado en Drupal, el mismo posee un blog para la discusión de temas de interés para la organización.

3.2.1.4 Módulo de monitoreo y control

Agrupar un conjunto de herramientas que son un aliado indispensable en el monitoreo y control de los servicios que se brindan, las herramientas usadas permiten valorar el estado de los servidores y detectar problemas que puedan ocurrir en los mismos.

Bacula 5.0.1

Software de código abierto, un sistema de copias de seguridad de nivel empresarial para redes heterogéneas. Diseñado para automatizar tareas que a menudo requieren la intervención de un administrador de sistemas. Soporta GNU/Linux, Unix, clientes de copias de seguridad de Windows y una gama de dispositivos profesionales de copia de seguridad, incluyendo las bibliotecas de cinta. Los administradores y operadores pueden configurar el sistema a través de una línea de comandos de consola, GUI o interfaz Web. Es una de las herramientas Open Source preparadas para funcionar en entornos empresariales.

Posee varios componentes:

Director: es la aplicación que se encarga de supervisar todo lo relacionado con Bacula.

Console: es la encargada de comunicarse con Bacula Director.

File: aplicación que debe estar instalada en la máquina para hacer copias de seguridad.

Storage: esta aplicación realiza la lectura y escritura del espacio de almacenamiento.

Catalog: aplicación responsable de las bases de datos usadas.

Monitor: permite al administrador mantener el historial del estado de varias herramientas de Bacula. [15]

Webmin 1.420

Webmin es una interfaz basada en web que permite la administración de sistemas Unix. La idea general de Webmin es poner de forma accesible la configuración de la mayoría de programas, servicios y servidores que se usan normalmente en el SO (Unix, Linux), de modo que todo se logra mediante formularios webs, pero a su vez, se pueden editar los archivos de configuración mediante un editor de texto. Es un programa modular, lo que posibilita instalar nuevos módulos. [41]

No se encontró otra herramienta parecida a esta.

Nagios3 3.2.0

Es un monitor de sistemas y redes usado en GNU/Linux, el cual controla toda la IT, asegurando los procesos en los sistemas, servicios y negocios con el fin de que estos funcionen adecuadamente.

En caso de falla, posee la capacidad de alertar al equipo técnico sobre el problema, antes que este afecte los procesos en el negocio, usuarios finales o consumidores.

Entre sus características principales figuran la monitorización de servicios de red (SMTP, POP3, HTTP, SNMP), la monitorización de los recursos de sistemas hardware (carga del procesador, uso de los discos, memoria, estado de los puertos), independencia de sistemas operativos, posibilidad de monitorización remota mediante túneles SSL cifrados ó SSH, y la posibilidad de programar plugins específicos para nuevos sistemas. [30]

Webalizer 2.01.10

Desarrollado en el lenguaje de programación C, permite generar reportes de páginas web. Mediante esos reportes, se puede observar el número de personas que han entrado en la web donde se halla instalado (o

ejecutado). Este programa no sólo brinda reportes cuantitativos, sino que también proporciona reportes gráficos, lo que hace más elegante y sencillo de observar las estadísticas de las páginas web. Entre otras cosas Webalizer es sumamente útil para saber qué archivos son los que poseen más número de descargas en las páginas web. También genera gráficas en formato HTML para que puedan ser observadas desde un navegador web. [29]

3.2.2 Diseño Estructural de Tecnología

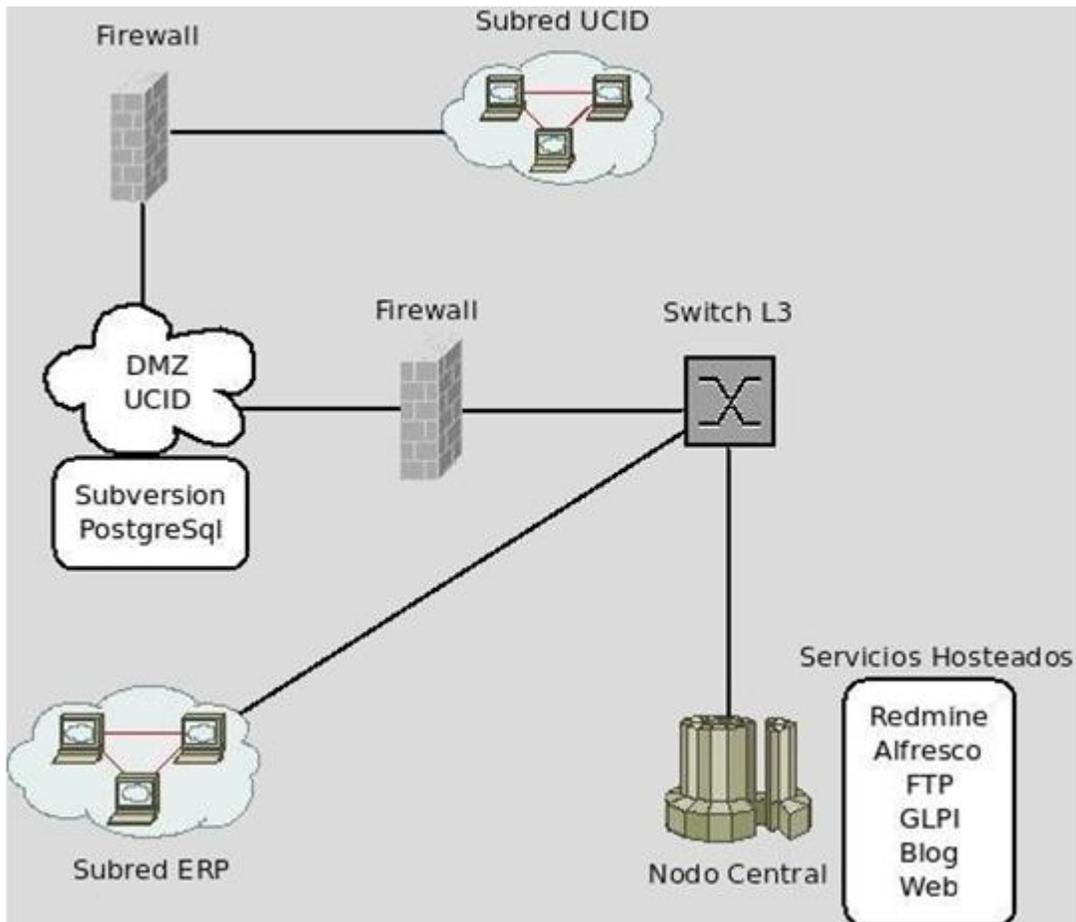


Figura 1 Diseño Estructural de Tecnología

3.2.3 Políticas de trabajo

Para mantener un buen funcionamiento en el trabajo con los servicios telemáticos que soportan el desarrollo del software es necesario cumplir disciplinadamente los siguientes requisitos:

- Tener las configuraciones de los servidores siempre resguardadas, así como las bases de datos que usan los mismos. Esto asegura en gran medida la restauración en casos de desastres y/o pérdida de información, minimizando el tiempo de afectación. Se recomienda instalar un sistema de salvas centralizadas y automatizar el proceso mediante scripts o herramientas para dichos fines.
- Probar con anterioridad (en servidores o PCs de prueba) los cambios de las configuraciones y adaptaciones de los servicios, así como integraciones entre estos para llevarlas a cabo posteriormente en los servidores definitivos.
- Documentar los procedimientos para la instalación y configuración de los servidores y servicios telemáticos.
- Mantener actualizados los registros de los cambios y configuraciones en los servicios. Llevar una bitácora con la documentación de dichos registros debidamente identificados.

3.2.3.1 Sistema de salvas

Realizar las siguientes salvas:

- Base de datos de las aplicaciones que se desarrollan.
- Base de datos de las aplicaciones que se prueban (piloto, despliegue).
- Bases de datos de todos los servicios de la Plataforma.
- Servidor FTP.
- Datos y configuraciones del servidor de subversion.
- Ficheros de configuración de todos los servicios de la plataforma.
- Ficheros de trazas de las herramientas de monitoreo.
- Reportes de las herramientas de monitoreo.

Procedimiento de realización de salvas automáticas

Cada servidor al que se le realiza salva posee un script implementado en el lenguaje de programación bash, que cuando es ejecutado realiza la salva de los datos correspondientes ubicándolos en un lugar específico.

En el servidor donde se almacenan todas las salvas, se hace un fichero script que ejecute cada uno de los ficheros en los servidores correspondientes y copia el resultado de ello (salvas) hacia el destino final (servidor de salvas). Mediante una tarea programada se ejecuta el fichero en el servidor donde se

almacenarán todas las salvas (servidor de salvas) en un horario conveniente, preferentemente por la madrugada (3 am).

Además de las salvas que se realizan mediante scripts, la plataforma se integra con el servicio de salvas automáticas que provee la Dirección Técnica de la UCI, estas se realiza mediante el uso de Bacula con los clientes previamente configurados y los servidores a los que se realizan las salvas.

3.2.3.2 Monitoreo y Control.

El monitoreo periódico de los servicios y del comportamiento de la red, se realizan mediante el uso de herramientas que sean capaces de sacar estadísticas y que puedan detectar cambios que ocurran tanto en los sistemas como en las redes que se monitorean, además de emitir reportes al aparecer ciertas anomalías. Los mismos se realizan sobre el ancho de banda, consumo de los recursos y servicios que se ofrecen, sobre el uso de memoria RAM de los servidores, así como el comportamiento de los procesadores y la cantidad de tráfico que circula por los diferentes puntos de acceso a la red, y cuáles servicios son más consumidos que otros con el fin de mejorar los mismos y disponer de balanceadores de carga que regulen el tráfico evitando interrupciones como cuellos de botella, sobrecarga de servidores y otros.

3.2.3.3 Lista de verificación

La siguiente lista de verificación se define para la PS, igual que la definida para el ED esta asigna pesos a las diversas preguntas que contiene, dichas preguntas se muestran a continuación:

- ¿Se realiza una descripción de cada uno de los servidores y máquinas clientes con los que se cuentan, con suficiente detalle que permitan determinar una breve descripción, software base y servicios?
- ¿Se realizan salvas de los principales servidores de misión crítica en la organización?
- ¿Se monitorean y controlan los servicios que se brindan?
- ¿Existe un monitoreo y control del uso del ancho de banda de la red?
- ¿Se almacenan los reportes o estadísticas emitidos por las herramientas de monitoreo?
- ¿Estado en que se encuentran los servicios, bien o mal, en caso de ser mal, por qué?
- ¿Se lleva un control del consumo de recurso de los servidores?
- ¿Se realizan mejoras en caso de que sea necesario a partir del análisis de los reportes que se realizan?

- ¿Se están consumiendo los recursos de forma adecuada por parte de los miembros del proyecto?

3.2.3.4 Scripts

Un script es un programa escrito en un lenguaje interpretado (en lugar de compilado). Este consiste de una serie de instrucciones que serán utilizadas por otra aplicación o que pueden ser ejecutadas de forma manual. Los scripts son muy usados en la automatización de tareas y en la administración de sistemas operativos, dado que se pueden crear ficheros con instrucciones que son ejecutados mediante el uso de un cron (en sistemas UNIX cron es un administrador regular de procesos en segundo plano que ejecuta procesos) en el sistema.

Bash 4.1

Bash es un intérprete de comandos que funciona como interfaz de usuario con el SO. Es el encargado de traducir los comandos de los usuarios a instrucciones que el SO pueda entender y viceversa, traducir el resultado devuelto por el SO a un lenguaje que los usuarios puedan entender.

Es uno de los intérpretes de comandos más usados y de los más potentes. Programar en este lenguaje es fácil, el mismo posee muchas posibilidades de uso. Ha sido escrito por el Proyecto GNU y pertenece a la categoría de interfaz de usuarios en modo carácter o texto. [1]

Perl 5.10.1

Perl es un lenguaje creado a principio de los noventa por Larry Wall y su distribución es gratuita. Fue pensado para la manipulación de cadenas de caracteres, archivos y procesos. Se percibe habitualmente como un lenguaje intermedio entre los Shell Scripts y la programación en C. El lenguaje Perl no es precompilado, pero aún así es más rápido que la mayoría de lenguajes interpretados, en especial que el Bourne Shell. Esto se debe a que los programas en Perl son analizados, interpretados y compilados por el intérprete Perl antes de su ejecución.

Estas características hacen que el mantenimiento y la depuración de un programa en Perl sean mucho más sencillos que el mismo programa escrito en C.

Perl es un lenguaje muy utilizado en los siguientes campos:

- **La administración de sistemas operativos:** Debido a sus características es muy potente en la creación de pequeños programas que pueden ser usados como filtros para obtener información de

ficheros, y realizar búsquedas. Además, aunque Perl nació en un entorno Unix, hay versiones para casi todas las plataformas existentes.

- **La creación de formularios en la web:** Se utiliza para la creación de scripts CGI (Common Gateway Interface). Estos scripts realizan el intercambio de información entre aplicaciones externas y servicios de información, es decir, se encarga de tratar y hacer llegar la información que el cliente web manda al servidor a través de su formulario. [37]

3.3 Despliegue

Se define un conjunto de herramientas que permiten realizar pruebas de carga y estrés sobre los servidores, donde se obtienen resultados que ayudan a diseñar escenarios de despliegue lo más reales posibles. También se lleva a cabo el diseño tecnológico de un centro de datos, y se proporciona una estrategia de soporte que permite mantener un control sobre los recursos.

3.3.1 Herramientas

JMeter 2.3.4

Desarrollado en Java dentro del proyecto Jakarta, que permite realizar pruebas de rendimiento y pruebas funcionales sobre aplicaciones web. Es una herramienta de carga para llevar a cabo simulaciones sobre cualquier recurso de software.

Inicialmente fue diseñado para pruebas de estrés en aplicaciones web, hoy en día su arquitectura ha evolucionado no sólo para llevar a cabo pruebas en componentes habilitados en Internet (HTTP), sino además: en bases de datos, programas en Perl, y prácticamente cualquier otro medio.

Además, posee la capacidad de realizar desde una solicitud sencilla hasta secuencias de peticiones que permiten diagnosticar el comportamiento de una aplicación en condiciones de producción. En este sentido, simula todas las funcionalidades de un navegador, o de cualquier otro cliente, siendo capaz de manipular resultados en determinada requisición y reutilizarlos para ser empleados en una nueva secuencia. [2]

Dia 0.97.1

Dia es un editor de diagramas que ha nacido, como la respuesta Open Source a Visio, la herramienta de Microsoft. Es totalmente compatible con ella, por lo tanto, permite importar y exportar diagramas en

formato VDX, además de XML, SVG, XFIG, WMF, PNG y EPS. Es muy completo y permite la creación de todo tipo de diagramas, eléctrico, mecánico, hidráulico, electrónico, de flujo, de red, entre otros

Es una aplicación GPL y multiplataforma, pequeña, liviana y fácil de usar. [33]

3.3.2 Escenarios de despliegue

Los escenarios de despliegue se definen teniendo en cuenta el tipo de aplicación que se desarrolla y los usuarios finales a los que está dirigida. Siendo una aplicación que está pensada para funcionar tanto en ambientes relativamente pequeños como en otros más amplios.

Para diseñar los escenarios con un grado aceptable de confiabilidad se simulan situaciones lo más real posible con los recursos disponibles y se les realiza determinadas pruebas que en este caso han sido pruebas de rendimiento para evaluar el tiempo de respuesta de la aplicación y pruebas de carga y estrés, variando parámetros como la memoria RAM, la cantidad de usuarios que se conectan de forma concurrente y en la mayoría de los casos invocando las funcionalidades identificadas como críticas en el negocio de la aplicación.

A continuación se muestran los datos de pruebas realizadas y algunos de los escenarios definidos después de analizar los resultados de las pruebas mencionadas anteriormente.

3.3.2.1 Pruebas de carga y estrés

Se realizaron un conjunto de pruebas a la aplicación web para comprobar su velocidad de respuesta a peticiones HTTP con la herramienta Jmeter.

Características del servidor:

- RAM: 1 Gb
- Capacidad disco duro: 250 Gb
- Procesador: Intel ® Pentium 4 a 3.0 Ghz
- Red: 100 Mbps

En pruebas de rendimiento a una aplicación es muy importante definir los rangos de éxito por los cuales se registrará la aplicación para poder decir cómo se comporta en dependencia de la cantidad de usuarios que

se conectan a la vez. En este caso se han definido varias categorías en dependencia del tiempo de respuesta para así poder dar un criterio de cómo se comportará la aplicación.

Los rangos definidos son:

- De 0 a 1 segundo de rendimiento es óptimo.
- De 1 a 3 segundos de rendimiento es alto.
- De 3 a 6 segundos el rendimiento es medio.
- De 6 segundos en lo adelante el rendimiento es bajo.

Una vez terminadas las pruebas a la aplicación, teniendo el criterio de éxito y los intervalos arrojados por las pruebas, se llega a la conclusión de que la aplicación puede tener un rendimiento Alto cuando acceden hasta 25 usuarios simultáneamente, un rendimiento Medio cuando acceden hasta 40 usuarios aproximadamente y tendría un rendimiento Bajo cuando la cantidad de usuarios sea mayor de 50.

3.3.2.2 Diseño lógico del centro de datos

El Centro de Datos que se propone cuenta con los recursos elementales para poder implementar la PS. Hay otros elementos que se deben tener en cuenta, pero estos dependen más de la organización y de las políticas de seguridad que tengan definidas. El Centro de Datos debe contar con: servidor web, servidor de datos, sistema de almacenamiento.

También se puede considerar otro conjunto de elementos como son: la creación de una DMZ para controlar el acceso a la organización, dispositivos de hardware para brindar servicios a clientes inalámbricos, creación de una LAN.

3.3.2.3 Definición de escenarios

Teniendo en cuenta las pruebas anteriormente realizadas a la aplicación desarrollada se realiza el diseño de varios escenarios de despliegue.

Escenario 1: sin conexión a internet

- 1 PC escritorio como servidor web, 1 GB RAM, 4 HDD, Pentium 4 o superior.
- 1 PC escritorio como servidor de datos, 1 GB RAM, 80 HDD, Pentium 4 o superior.
- 10 PC escritorio como estaciones de trabajo, 256 MB RAM, Pentium 4 o superior.

Escenario 2: con conexión a Internet

- 1 PC escritorio como servidor web, 1 GB RAM, 4 HDD, Pentium 4 o superior.
- 1 PC escritorio como servidor de datos, 1 GB RAM, 80 HDD, Pentium 4 o superior.
- 10 PC escritorio como estaciones de trabajo, 256 MB RAM, Pentium 4 o superior.
- 1 PC escritorio como router y firewall 1GB RAM, 4 HDD, Pentium 4 o superior, 2 tarjetas de red.
- 1 Switch L2.

3.4 Validación de la propuesta.

Para validar las pautas propuestas para diseñar una AI se decide establecer una comparación entre la Infraestructura del Proyecto ERP-Cuba antes de aplicada dicha propuesta en el diseño de la misma y después de aplicada la propuesta. Para esto se usarán un conjunto de aspectos que ayudan a valorar el estado en que se encuentra la Infraestructura del proyecto.

Tabla 3 Comparación de la Infraestructura del proyecto ERP-Cuba antes y después de diseñada la Vista de Infraestructura.

Entorno de Desarrollo

Requisitos	Antes	Después
Imágenes estándar definidas para equipos de escritorio	No	Sí
Establecimiento de herramientas de implementación para instalaciones de red o de imagen sin conexión.	No	Sí
Establecimiento de un plan para evaluar y probar todo el software adquirido.	No	Sí
Definición de un conjunto estándar de imágenes de disco (SO y aplicaciones) para todos los tipos de hardware.	No	Sí
Prueba y certificación de la compatibilidad de un 80% de las aplicaciones nuevas o actualizadas antes de implementarlas en los equipos de escritorio.	No	Sí
Instalación de todos los sistemas operativos y actualizaciones de seguridad de aplicaciones de software.	No	Sí

Plataforma de Servicios

Requisitos	Antes	Después
Agrupamiento de servicios según funcionamiento, requerimientos de hardware u otra forma de agrupación.	Sí	Sí
Valores agregados en la organización mediante la implementación de servicios.	No	Sí
Creación de un plan de copia de seguridad de datos y un plan de recuperación para el 80% o más de todos los servidores de la organización.	No	Sí
Uso de simulacros para probar los planes.	No	No
Instalación de software de supervisión de disponibilidad.	No	Sí
Supervisión de un 80% de los servidores críticos en cuanto a rendimiento, eventos y alertas.	No	Sí

Despliegue

Requisitos	Antes	Después
Diseño de Escenarios de Despliegue basados en pruebas realizadas a las aplicaciones desarrolladas	No	Sí
Diseño de un Centro de Datos para hospedar los servicios de la organización	No	Sí

El análisis anterior evidencia la factibilidad de las pautas propuestas para el diseño de la Vista de AI, pues se muestra gestión poco eficiente que existía en el proyecto ERP-Cuba. Anteriormente no se contaba con imágenes para las computadoras de producción, ni poseían servicios automatizados que ayudaran en la gestión del proyecto. Asimismo, tampoco tenían servidores dedicados al compartimiento de información, herramientas y documentos como servidores FTP, no se gestionaba la superación de los miembros del proyecto mediante el uso de portales web, foros o blogs.

Con el diseño de la Vista de Infraestructura, el proyecto adquiere un conjunto de características que mejoran el desarrollo del mismo. Actualmente poseen una plataforma de desarrollo bien definida para las

estaciones de trabajo, la cual es mantenida y actualizada periódicamente. También cuentan con sistemas para la gestión de proyecto, gestión documental, servidores de ficheros; todos estos servicios se encuentran separados por módulos, lo que permite realizar un mejor control y mantenimiento de los mismos.

De la misma manera se lleva a cabo un control sobre la disponibilidad de los servidores que brindan servicios a la organización mediante el uso de un conjunto de herramientas destinadas a este fin. Se han diseñado Escenarios de Despliegue basados en pruebas realizadas a las aplicaciones creadas, basadas en Entornos de Desarrollo reales.

Conclusiones

En este capítulo queda presentada la Vista de Infraestructura del proyecto ERP-Cuba. Para el diseño y la implementación de la misma se realizan las actividades propuestas en el Capítulo 2: Pautas para el desarrollo de la AI.

Se presentaron las tecnologías y herramientas fundamentales usadas para el desarrollo y la gestión de las actividades del proyecto.

Las tecnologías y herramientas seleccionadas son distribuidas bajo licencias GPL o equivalentes, y en su mayoría multiplataforma. Teniendo en cuenta las condiciones económicas del país y de la universidad, además de su política de migración a software libre, seleccionar herramientas de este tipo constituye un aporte considerable en ese sentido.

Se realizó una comparación del estado de la Infraestructura del proyecto antes y después de aplicar las pautas propuestas para su diseño, implementación y control. Mostrando un conjunto de aspectos que se tuvieron en cuenta en dicha comparación, se evalúan los resultados finales como positivos, se evidencia que la aplicación de pautas para organizar y mejorar la gestión de la Infraestructura del proyecto contribuye a una mejora en el desarrollo y en la gestión de las actividades del proyecto.

CONCLUSIONES GENERALES

Al culminar la presente investigación se dio cumplimiento a los objetivos planteados, alcanzando los resultados propuestos. Se logró desarrollar una guía, presentada en pautas, que propone las actividades fundamentales a realizar en el área de AI, se definen roles que llevan a cabo dichas actividades además de artefactos generados en cada una de las mismas.

Se reafirma así la necesidad de contar con guías de actividades bien definidas para garantizar la gestión adecuada de la Infraestructura de un proyecto de desarrollo de software.

De manera general, se concluye que lo anterior se ve demostrado a través de lo siguiente:

- Se analizaron los fundamentos teóricos, demostrando la necesidad de desarrollar una guía para diseñar, implementar y gestionar la Infraestructura Tecnológica de los proyectos de desarrollo de software de gestión.
- Se realizó un estudio de guías propuestas por fuentes como Microsoft, TOGAF y SEI, punteras en el desarrollo de software y con una madurez reconocida en cuanto a la gestión de sus infraestructuras tecnológicas.
- Se presentaron las pautas para el diseño de la Vista de Infraestructura de un proyecto de desarrollo de software de gestión y se validaron las mismas mediante su aplicación en el proyecto ERP-Cuba, demostrándose los aportes de aplicar dichas pautas.

RECOMENDACIONES

Con el objetivo de continuar el tema de investigación de este trabajo, para incrementar sus resultados en las áreas productivas se recomienda:

Continuar las investigaciones sobre el objeto de estudio que motivó a este trabajo, profundizando en los problemas identificados.

Desplegar los resultados de este trabajo en los proyectos de la UCI, a fin de extender las mejoras que esta solución proporciona.

Revisar este informe con el objetivo de su publicación, partiendo de sus contenidos, para profundizar en temas que permitan su presentación en talleres y otros eventos científicos nacionales.

REFERENCIAS BIBLIOGRÁFICAS

1. Antofalinux.cl. Bash1 - Introducción y ficheros de configuración. Marzo 2007. [cited 23 Marzo 2010]. Available from world wide web: <http://www.antofalinux.cl/index.php?option=com_content&view=article&id=56:bash1&catid=21:avanzados&Itemid=9>.
2. Apache Software Foundation. JMeter - Apache JMeter. JMeter - Apache JMeter 2009. [cited 4 Junio 2010]. Available from world wide web: <<http://jakarta.apache.org/jmeter/>>.
3. Bass, Len, Paul Clements, y Rick Kazman. Software Architecture in Practice. 2da Edición Addison-Wesley Professional, Abril 2003 [cited 8 Enero 2010]. Available from world wide web: <<http://www.informit.com/store/product.aspx?isbn=0321154959>>.
4. Bredemeyer, Dana, y Ruth Malan. Who: Software Architect Role. Junio 2000. [cited 8 Enero 2010]. Available from world wide web: <<http://www.bredemeyer.com/who.htm>>.
5. Canonical Ltd. Ubuntu - Linux for Human Beings! 2008. [cited 24 Enero 2010]. Available from world wide web: <<https://help.ubuntu.com/9.04/about-ubuntu/C/>>.
6. Carnegie Mellon University. About Us | Overview. 2010. [cited 26 Enero 2010]. Available from world wide web: <<http://www.sei.cmu.edu/about/>>.
7. Ciberaula. Una Introducción a Apache. 2010. [cited 15 Febrero 2010]. Available from world wide web: <http://linux.ciberaula.com/articulo/linux_apache_intro/>.
8. Coletti, Daniel Estevan. ProFTPD - un servidor FTP para profesionales. 2010. [cited 26 Marzo 2010]. Available from world wide web: <<http://beta.redes-linux.com/manuales/ftp/proftpd.pdf>>.
9. Complusoft. Alfresco - Gestor documental - Complusoft - Software Development Company. 2010. [cited 18 Marzo 2010]. Available from world wide web: <<http://www.complusoft.es/soluciones/alfresco>>.
10. Daniel Jumelet, Infrastructure Architecture. marzo 2007. [cited 1 Julio 2010]. Available from world wide web: <http://msdn.microsoft.com/es-es/library/bb402960.aspx>
11. Debian.org. Debian - Details of package subversion in etch-backports. 2010. [cited 7 Junio 2010]. Available from world wide web: <<http://packages.debian.org/es/etch-backports/subversion>>.
12. Desarrolloweb.com. Ext JS. 2010. [cited 7 Junio 2010]. Available from world wide web: <<http://www.desarrolloweb.com/wiki/ext-js.html>>.
13. Díaz Verdecia, Osvaldo, y Virgen Damaris Quevedo Campins. Una guía práctica de Arquitectura de Software. Mayo 2009.

14. Doeyo, Leandro. Arquitectura de Software en Forma de Cuentito - Windows Live. Noviembre 2007. [cited 22 Febrero 2010]. Available from world wide web: <<http://diegumzone.spaces.live.com/blog/cns!1AD5096D63670065!1227.entry>>.
15. DragonJAR. 10 extraordinarias herramientas para hacer copias de seguridad en Linux. 10 extraordinarias herramientas para hacer copias de seguridad en Linux Enero 2010. [cited 15 Enero 2010]. Available from world wide web: <<http://www.dragonjar.org/herramientas-para-backup-linux.xhtml>>.
16. Edgewall Software. Kdesvn project. Kdesvn project Octubre 2010. [cited 1 Octubre 2010]. Available from world wide web: <<http://kdesvn.alwins-world.de/>>.
17. GNU FDL. RapidSVN. Octubre 2009. [cited 1 Octubre 2010]. Available from world wide web: <http://rapidsvn.org/index.php/Main_Page>.
18. Guest, Simon. Architect Personas - simon.says - Site Home - MSDN Blogs. Diciembre 2005. [cited 14 Febrero 2010]. Available from world wide web: <<http://blogs.msdn.com/b/smguest/archive/2005/12/12/503001.aspx>>.
19. Hilillos Blog. Planner | Hilillos-blog. Junio 2009. [cited 23 Marzo 2010]. Available from world wide web: <<http://www.hilillos-blog.org/tag/planner/>>.
20. IBM, Exploring IT architecture disciplines, Part 4: Update your infrastructure architecture to serve and protect your enterprise. 19 sep 2006. [cited 1 Julio 2010]. Available from world wide web: <http://www.ibm.com/developerworks/architecture/library/ar-infraoverview/>
21. IEEE-SA. IEEE Standards Description: 1471-2000. 2004. [cited 12 Enero 2010]. Available from world wide web: <http://standards.ieee.org/reading/ieee/std_public/description/se/1471-2000_desc.html>.
22. Informática Milenium, S.A.de C.V. ¿Que son los Portales en Internet? ¿Que son los Portales en Internet? 2009. [cited 20 Enero 2010]. Available from world wide web: <<http://www.informaticamilenium.com.mx/paginas/mn/articulo25.htm>>.
23. ItecoPartner s.l.u. PyQt4 , SQLite, MySQL, PostgreSQL | iTecnoPartner. Abril 2010. [cited 7 Junio 2010]. Available from world wide web: <<http://www.itecnopartner.com/?q=en/content/pyqt4-sqlite-mysql-postgresql>>.
24. Maestro del Web. ¿Qué es Javascript? 2007. [cited 5 Marzo 2010]. Available from world wide web: <<http://www.maestrosdelweb.com/editorial/%C2%BFque-es-javascript/>>.
25. Microsoft Corporation. Guía de recursos de implementador para la optimización de la infraestructura principal: del nivel básico al estandarizado. Mayo 2007. [cited 4 Enero 2010]. Available from world wide web: <<http://technet.microsoft.com/es-es/library/bb821254.aspx>>.

26. Oracle Corporation. NetBeans 6.8 IDE - PHP Development. NetBeans 6.8 IDE - PHP Development 2010. [cited 1 Enero 2010]. Available from world wide web: <<http://netbeans.org/features/php/index.html>>.
27. Perry, Dewayne E., y Alexander L. Wolf. Foundations for the study of software architecture. ACM SIGSOFT Software Engineering Notes 17:4, Octubre 1992, 40-52.
28. Rodas Hinostroza, Raul. LinuxCentro.net - Características de PHP. Febrero 2007. [cited 12 Febrero 2010]. Available from world wide web: <<http://www.linuxcentro.net/linux/staticpages/index.php?page=CaracteristicasPHP>>.
29. Rosado, David. Como usar Webalizer para que nos genere estadísticas de nuestros Sitios. Octubre 2007. [cited 25 Febrero 2010]. Available from world wide web: <<http://www.alcancelibre.org/article.php/20071008203343467>>.
30. Scribd. NAGIOS. NAGIOS Octubre 2009. [cited 17 Enero 2010]. Available from world wide web: <<http://www.scribd.com/doc/21931635/NAGIOS>>.
31. Simple Machines LLC. Ext JS - En Español | Bienvenido. 2008. [cited 15 Enero 2010]. Available from world wide web: <<http://www.extjses.com/>>.
32. SlideShare Inc. Visual Paradigm For Uml. 2010. [cited 7 Junio 2010]. Available from world wide web: <<http://www.slideshare.net/vanquishdarkenigma/visual-paradigm-for-uml>>.
33. SmallSquid.com. Dia, un editor de diagramas gratuito y compatible con Visio - AplicacionesEmpresariales.com. Diciembre 2007. [cited 16 Enero 2010]. Available from world wide web: <<http://www.aplicacionesempresariales.com/dia-un-editor-de-diagramas-gratuito-y-compatible-con-visio.html>>.
34. Softwarelibrevenezuela. Gestion Libre del Parque Informatico, GLPI. Marzo 2009. [cited 8 Marzo 2010]. Available from world wide web: <<http://softwarelibrevenezuela.blogspot.es/i9-03/>>.
35. The Open Group. Phase D: Technology Architecture. 2006a. [cited 26 Enero 2010]. Available from world wide web: <<http://www.opengroup.org/architecture/togaf8-doc/arch/toc.html>>.
36. The Open Group. The Open Group Architecture Framework Version 8.1.1. 2006b. [cited 25 Febrero 2010]. Available from world wide web: <<http://www.opengroup.org/architecture/togaf8-doc/arch/toc.html>>.
37. Tufuncion.com. 8 lenguajes de programación que deberías aprender | TuFuncion. 2006. [cited 5 Marzo 2010]. Available from world wide web: <<http://www.tufuncion.com/diferentes-lenguajes-programacion>>.
38. Tufuncion.com. Las 10 mejores características de Firefox | TuFuncion. Las 10 mejores características de Firefox | TuFuncion Agosto 2007. [cited 1 Enero 2010]. Available from world wide web:

<<http://www.tufuncion.com/caracteristicas-firefox>>.

39. Ubuntu. PgAdmin III - Guía Ubuntu. Marzo 2008. [cited 11 Febrero 2010]. Available from world wide web: <http://www.guia-ubuntu.org/index.php?title=PgAdmin_III>.
40. Villanueva, Constanza. OpenOffice, el paquete de Office gratis. Ahora ya puedes tener Word Gratis, Excel Gratis, PowerPoint Gratis, Access Gratis, Publisher gratis, Ofimática gratis. Enero 2008. [cited 28 Febrero 2010]. Available from world wide web: <<http://openoffice.es/>>.
41. Webmin. Webmin. 2009. [cited 21 Marzo 2010]. Available from world wide web: <<http://webmin.com/>>.
42. Zend Technologies Ltd. Zend Studio - The Professional PHP IDE - Zend.com. Zend Studio - The Professional PHP IDE - Zend.com 2009a. [cited 1 Enero 2010]. Available from world wide web: <<http://www.zend.com/products/studio/>>.

BIBLIOGRAFÍA

Amartino, Mariano. PHP y el proceso del lado del servidor. 2010. [cited 23 Enero 2010]. Available from world wide web: <<http://www.uberbin.net/archivos/internet/php-y-el-proceso-del-lado-del-servidor.php>>.

Apache. Apache Resource. [cited 23 Marzo 2010]. Available from world wide web: <<http://www.apache.com/>>.

Microsoft Corporation. Infraestructura tecnológica. 2010. [cited 15 Abril 2010]. Available from world wide web: <<http://www.microsoft.com/business/smb/es-mx/infraestructura-tecnol%C3%B3gica/default.msp>>.

Pecos, Daniel. PostgreSQL. 2010. [cited 6 Marzo 2010]. Available from world wide web: <http://www.netpecos.org/docs/mysql_postgres/x15.html>.

PostgreSQL Global Development Group. PostgreSQL: The world's most advanced open source database. 2010. [cited 16 Enero 2010]. Available from world wide web: <<http://developer.postgresql.org/pgdocs/postgres/pgbench.html>>.

Programación en castellano. Visual Paradigm for UML. Programación en Castellano. 2010. [cited 6 Febrero 2010]. Available from world wide web: <http://www.programacion.com/noticia/visual_paradigm_for_uml_1363>.

QuinStreet Inc. JavaScript.com (TM) - The Definitive JavaScript Resource: JavaScript Tutorials, Free Java Scripts, Source Code and Other Scripting Resources. 2010. [cited 19 Febrero 2010]. Available from world wide web: <<http://www.javascript.com/>>.

Diccionario informático - D-F. Junio 2010a. [cited 4 Junio 2010]. Available from world wide web: <http://www.hard-h2o.com/diccionario-informatico_d-f.html>.

Tutorial de Perl. Enero 2010b. [cited 17 Enero 2010]. Available from world wide web: <<http://www.ulpgc.es/otros/tutoriales/perl/indice.htm>>.

The ProFTPD Project. The ProFTPD Project: Home. *The ProFTPD Project: Home* 1999. [cited 15 Enero 2010]. Available from world wide web: <<http://www.proftpd.org/>>.

