

ESCUELA POLITÉCNICA DEL EJÉRCITO

DPTO. DE CIENCIAS DE LA COMPUTACIÓN

CARRERA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA

**IMPLEMENTACIÓN DE ORACLE PUBLIC CLOUD
ARCHITECTURE APLICADA A LA GESTIÓN DE
CATÁLOGO DE PRODUCTOS**

Previa a la obtención del Título de:

INGENIEROS DE SISTEMAS E INFORMÁTICA

POR: HERNÁN EDUARDO ALMACHI MONTEROS

PIETRO DENNIS ANDINO VELÁSQUEZ

SANGOLQUÍ, AGOSTO DE 2012

CERTIFICACIÓN

Certifico que el presente trabajo fue realizado en su totalidad por los Srs. HERNÁN EDUARDO ALMACHI MONTEROS y PIETRO DENNIS ANDINO VELÁSQUEZ como requerimiento parcial a la obtención del título de INGENIEROS DE SISTEMAS E INFORMÁTICA.

Sangolquí, Agosto de 2012

Ing. Jenny Ruiz
DIRECTORA DE TESIS

ESCUELA POLITÉCNICA DEL EJÉRCITO
CARRERA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
AUTORIZACIÓN

Nosotros, HERNÁN EDUARDO ALMACHI MONTEROS y
PIETRO DENNIS ANDINO VELÁSQUEZ

Autorizamos a la Escuela Politécnica del Ejército la publicación, en la biblioteca virtual de la Institución, del trabajo “IMPLEMENTACIÓN DE ORACLE PUBLIC CLOUD ARCHITECTURE APLICADA A LA GESTIÓN DE CATÁLOGO DE PRODUCTOS”, cuyo contenido, ideas y criterios son de nuestra exclusiva responsabilidad y autoría.

Sangolquí, Agosto de 2012

HERNÁN EDUARDO ALMACHI MONTEROS

PIETRO DENNIS ANDINO VELÁSQUEZ

DEDICATORIA

Dedico esta tesis a las 3 personas más importantes de mi vida:

A mi madre: Yolanda Monteros G., por su inmensa fortaleza, su inagotable apoyo, su valiosa dirección y consejo, por enseñarme a dar siempre lo mejor de mí y autosuperarme día tras día.

A mi padre: Hernán Almachi R., por su inmaculado ejemplo, por compartir y creer en todos los proyectos que me he propuesto, por siempre tener esa frase de aliento para sobrellevar todos los momentos de la vida, por su gran amor y paciencia.

A mi hermano: David Almachi M., por ser la persona más genial de este mundo y la más grande alegría en mi corazón.

Hernán

DEDICATORIA

Creaste los cielos y la tierra... Creaste el mar, la luna y las estrellas... Creaste el sol, las flores, creaste a la hormiga y creaste al león... Me diste una familia que se mantuvo en la aflicción... Porque estuviste con nosotros en la tormenta... Fuiste a la cruz y venciste... Nos salvaste con tu amor hecho sangre... Nos levantaste... Nos diste una nueva vida... Este esfuerzo y cada esfuerzo de mi vida te lo entrego a ti mi amado Señor Jesús.

A mi padre Edwin, soldado de Jesucristo

A mi madre Jeanette, mujer virtuosa.

A mi ñañita Yessenia, dulce princesita.

A mi abuelita Piedad, mi tía Norma y mi tío Giovanni, que me cuidaron de niño, que pintaron cada momento de alegría, esperanza, sabiduría y amor.

A Pablo Andino, Gandhi Jurado, David Diamond, Oscar Lasso y Ruth Velásquez, que me fueron a rescatar en el nombre de Jesús.

A mis amigos de Betesda. A mis leones de Judá.

A ti, que has abierto este trabajo: El anhelo de mi corazón es que te entregues a Jesucristo, que lo reconozcas como tu Señor y Salvador: *“Porque de tal manera amó Dios al mundo, que ha dado a su Hijo unigénito, para que todo aquel que en él cree, no se pierda, mas tenga vida eterna.”* (Juan 3:16, Sagrada Biblia - Palabra de Dios).

Pietro

AGRADECIMIENTOS

A los Ing. Jenny Ruiz y Geovanni Raura, por su acertada dirección en este proyecto.

A Hugo Zumárraga y Milton Recalde, por ser los mejores maestros y amigos en esta nueva etapa laboral.

A eMergeSoft, por ser la empresa baluarte en el desarrollo de este proyecto.

A la Escuela Politécnica del Ejército, por ser la brillante institución que me formó como profesional.

A mi compañero de tesis Pietro Andino, por su gran ayuda en la consecución de este proyecto.

A mis amigos, por su apoyo en todas las etapas de mi vida.

A la tierra, a la montaña, a las estrellas, a todo lo que fue y a todo lo que será.

Hernán

AGRADECIMIENTOS

A Dios, por darme la fortaleza para realizar este trabajo.

A mis padres y hermanita, por habernos atendido con tanto amor en nuestras reuniones de trabajo.

A Hernán, por todo lo que compartimos en este reto y aventura.

A los amigos que estuvieron pendientes del progreso de este Proyecto.

A Milton Recalde y Hugo Zumárraga, de eMergeSoft, por su enseñanza y su apoyo sincero.

A nuestros directores e informante de tesis: Ing. Jenny Ruiz, Ing. Geovanni Raura e Ing. Tatiana Noboa, respectivamente, por su valiosa colaboración para que este Proyecto sea exitoso.

A quienes hacen la Escuela Politécnica del Ejército, tan prestigiosa y hermosa universidad.

Pietro

ÍNDICE DE CONTENIDOS

RESUMEN	1
CAPÍTULO 1. INTRODUCCIÓN	2
1.1 Introducción	2
1.2 Justificación	2
1.3 Formulación del problema	3
1.4 Hipótesis	4
1.5 Objetivo general	4
1.6 Objetivos específicos	4
1.7 Alcance	5
1.7.1 Arquitectura	5
1.7.2 Módulos	6
1.8 Metodología	8
1.9 Factibilidad	9
1.9.1 Técnica y Económica	9
1.9.2 Software	9
1.9.3 Hardware	10
1.9.4 Administración	11
1.9.5 Operativa	11
CAPÍTULO 2. MARCO TEÓRICO	12
2.1 Realidad de la empresa	12
2.2 Arquitectura de la nube de Oracle (Oracle Cloud Computing)	12
2.2.1 Introducción	12
2.2.2 Definición de la computación en la nube	13
2.2.3 Definición de Oracle Cloud Computing	17
2.3 La metodología Programación Extrema (XP)	23
2.3.1 Introducción	23
2.3.2 Modelo de XP	24
2.3.3 Fases en XP	27
2.3.4 Roles en XP	28
2.3.5 Reglas y prácticas de XP	31
2.3.6 Valores en XP	39

2.4 Método Unificado Oracle (OUM)	40
2.4.1 Introducción	40
2.4.2 Características	41
2.4.3 Beneficios de OUM.....	41
2.4.4 Características clave	43
2.4.5 Implementación de un proyecto OUM.....	44
2.4.6 Procesos del proyecto para la continuidad	47
2.4.7 Actividades que determinan el compromiso en el ciclo de vida.....	53
2.4.8 Gestión de proyectos OUM.....	53
2.4.9 Actividades para el desarrollo y mantenimiento de la arquitectura empresarial.	56
2.4.10 Componentes del OUM.....	58
2.4.11 Requerimientos de software y hardware	59
2.4.12 Entorno de la herramienta de OUM	60
2.5 Oracle JDeveloper 11.1.1.1.5.0.....	61
2.5.1 Historia	61
2.5.2 Lenguajes soportados	62
2.5.3 Requerimientos de instalación	62
2.5.4 Roles en JDeveloper.....	63
2.5.5 Entorno de desarrollo	65
2.6 Oracle ADF 11.1.2.0.0.....	67
2.6.2 Introducción	67
2.6.3 Arquitectura.....	68
2.6.4 Componentes de negocio de ADF.....	69
2.7 Oracle Weblogic Server 11g.....	70
2.7.1 Historia.....	70
2.7.2 Sistemas de gestión de base de datos soportados por Oracle Weblogic Server ..	70
2.7.3 Conectores nativos	70
2.7.4 Estándares soportados	71
2.8 Base de datos Oracle 11g (Oracle Database 11g).....	74
2.8.1 Historia.....	74
2.8.2 Requisitos de Instalación.....	75
2.8.3 Estructura lógica.....	75
2.8.4 Estructura física.....	76

2.9 Oracle Enterprise Linux 5	77
2.9.1 Historia	77
2.9.2 Especificaciones del sistema	77
2.9.3 Sistemas de archivos soportados	78
2.9.4 Tecnologías de valor añadido Oracle	78
2.9.5 Seguridad.....	79
2.10 Aplicaciones tradicionales versus Cloud Computing	80
2.10.1 Análisis de características	80
2.10.2 Análisis estadístico	82
CAPÍTULO 3. DISEÑO.....	86
3.1 Introducción	86
3.2 Historias de usuario	86
3.3 Modelo de Casos de Uso	87
3.3.1 Relación Historias de usuario – Casos de uso.....	87
3.4 Plan de iteraciones	89
3.5 Especificación de diseño.....	90
3.6 Modelo de arquitectura	90
CAPÍTULO 4. PRUEBAS	91
4.1 Introducción	91
4.2 Pruebas de aceptación.....	91
4.2.1 Casos de prueba.....	92
4.3 Pruebas de rendimiento.....	93
4.3.1 Objetivos de las pruebas de rendimiento.....	93
4.3.2 Uso	93
4.3.3 Variables.....	94
4.3.4 Gráficos empleados	95
4.4 Hardware utilizado.....	96
4.5 Software utilizado	97
4.6 Pruebas de estrés	97
4.6.1 Test de rendimiento.....	97
4.6.2 Test por defecto.....	102
4.6.3 Test de carga sin exceso de peticiones	107
4.6.4 Test de sobrecarga.....	112

CAPÍTULO 5. CONCLUSIONES Y RECOMENDACIONES	118
5.1 Conclusiones	118
5.2 Recomendaciones	119
BIBLIOGRAFÍA	121
BIOGRAFÍA 1	126
BIOGRAFÍA 2	127
HOJA DE LEGALIZACIÓN DE FIRMAS	128

LISTADO DE TABLAS

Tabla 2.1: Fases de la Programación Extrema con sus características y resultados.....	27
Tabla 2.2: Análisis de características de sistemas tradicionales versus Computación en la nube	80
Tabla 3.2: Relación Historias de usuario – Casos de uso	87
Tabla 4.1: Características del servidor usado para las pruebas de aceptación	96
Tabla 4.2: Parámetros usados para el test de rendimiento	98
Tabla 4.3: Parámetros usados para el test por defecto	103
Tabla 4.4: Parámetros usados para el test de carga sin exceso de peticiones	108
Tabla 4.5: Parámetros usados para el Test de sobrecarga	113

LISTADO DE CUADROS

Cuadro 1.1: Costos de software en el Proyecto	10
Cuadro 1.2: Costos de hardware en el Proyecto	10
Cuadro 1.3: Costos de administración en el Proyecto	11
Cuadro 3.1: Historia de usuario <i>Autenticar usuarios</i>	86
Cuadro 4.1: Caso de prueba <i>Autenticar usuario</i>	92

LISTADO DE FIGURAS

Figura 1.1: Arquitectura en la nube de Oracle.....	5
Figura 2.1: Consumo de servicios en la nube.....	14
Figura 2.2: Niveles de la nube.....	16
Figura 2.3: Nube pública.....	18
Figura 2.4: Nube privada.....	19
Figura 2.5: Nube híbrida.....	20
Figura 2.6: Componentes para soportar Oracle PasS Platform (Plataforma Oracle PaaS) .	23
Figura 2.7: Modelo de desarrollo en XP.....	25
Figura 2.8: Fases de XP.....	28
Figura 2.9: Fases de implementación de un proyecto OUM.....	44
Figura 2.10: Integración del Área de enfoque de gestión con el Área de enfoque de implementación.....	54
Figura 2.11: Página inicial del entorno de la herramienta de OUM.....	60
Figura 2.12: Selección de roles en JDeveloper.....	64
Figura 2.13: Entorno de desarrollo JDeveloper 11g.....	65
Figura 2.14: Arquitectura ADF Fusion.....	68
Figura 2.15: Componentes de negocio de ADF.....	69
Figura 2.16: Instalaciones y dominios en Oracle Weblogic Server.....	72
Figura 2.17: Entorno de Oracle Weblogic Server.....	73
Figura 2.18: Estructura lógica de la base de datos Oracle.....	76
Figura 2.19: Estructura física de la base de datos Oracle.....	77
Figura 2.20: Estructura del sistema operativo Oracle Linux.....	79
Figura 2.21: Porcentaje de empresas pequeñas que utilizan Cloud Computing.....	82
Figura 2.22: Porcentaje de empresas medianas que utilizan Cloud Computing.....	82
Figura 2.23: Porcentaje de empresas grandes que utilizan Cloud Computing.....	82
Figura 2.24: Comparativa en la eficiencia en la recuperación de desastres.....	84

Figura 2.25: Segmentación de la demanda de la nube	84
Figura 4.1: Estructura de la interfaz del sistema de gestión de catálogo de productos Lithium ®	94
Figura 4.2: Resumen de desempeño – Test de rendimiento	99
Figura 4.3: Gráfico de máximos tiempos de respuesta – Test de rendimiento.....	100
Figura 4.4: Curva de spline – Test de rendimiento.....	101
Figura 4.5: Resumen de desempeño – Test por defecto	104
Figura 4.6: Gráfico de máximos tiempos de respuesta – Test por defecto.....	105
Figura 4.7: Curva de spline – Test por defecto.....	106
Figura 4.8: Resumen de desempeño – Test de carga sin exceso de peticiones	109
Figura 4.9: Gráfico de máximos tiempos de respuesta – Test de carga sin exceso de peticiones	110
Figura 4.10: Curva de spline – Test de carga sin exceso de peticiones	111
Figura 4.11: Resumen de desempeño – Test de sobrecarga.....	114
Figura 4.12: Gráfico de máximos tiempos de respuesta – Test de sobrecarga	115
Figura 4.13: Curva de spline – Test de sobrecarga	116

NOMENCLATURA UTILIZADA

XP: Xtreme Programming (Programación Extrema). Metodología ágil de desarrollo de software que pone énfasis en la adaptabilidad por sobre la previsibilidad.

OUM: Oracle Unified Method (Método Unificado Oracle). Metodología propia de la compañía Oracle, basada en estándares y fundamentada en el Proceso Unificado de desarrollo.

LDAP: Lightweight Directory Access Protocol (Protocolo Ligero de Acceso a Direcciones). Es un protocolo a nivel de la aplicación que nos permite acceder a un directorio organizado para obtener información de la red.

Oracle ADF: Oracle Application Development Framework (Framework de Desarrollo de Aplicaciones Oracle).

Data: Datos

Metadatos: Literalmente “datos sobre datos”. Son datos que describen a otros datos.

Binding: Ligadura o referencia a otro símbolo más complejo, que se usa frecuentemente y puede ser de cualquier tipo (numérico, cadena, variable, etcétera).

Front-end: Parte del software que interactúa con los usuarios.

Back-end: Parte del software que procesa las entradas desde el front-end.

Commit: Instrucción de guardar cambios en los datos de la base de datos.

Rollback: Instrucción de deshacer los cambios en los datos de la base de datos.

Task flow: Flujo de tareas.

RESUMEN

El presente Proyecto es una abstracción del negocio del catálogo de productos llevado a un sistema web cuyo fundamento es la arquitectura Oracle Public Cloud Computing. Se realizó el desarrollo del módulo de inventarios, encargado de la creación de productos y artículos disponibles como pedidos. El sistema cuenta con una capa de seguridad definida por un directorio LDAP albergado en el servidor de aplicaciones Oracle Weblogic.

En base a los requerimientos funcionales y de arquitectura, se acoplaron las metodologías Programación Extrema y Método Unificado Oracle, obteniendo una mayor rapidez en la codificación como beneficio principal de la Programación Extrema, así como el entendimiento del negocio y la retroalimentación de información de desarrollo por parte de Método Unificado Oracle.

El resultado final del Proyecto es el sistema piloto de catálogo de productos Lithium ®, con la respectiva documentación de historias de usuario, casos de uso, componentes de arquitectura, diagramas de secuencia, diagramas de base de datos, pruebas de aceptación y pruebas de rendimiento.

CAPÍTULO 1

INTRODUCCIÓN

1.1 Introducción

La constante evolución del mercado y la búsqueda de acceder a un mayor número de clientes, hace de las tecnologías de información, una herramienta base.

En tal virtud, es necesario que tales herramientas diversifiquen su arquitectura, para adaptarse a mercados cada vez más cambiantes, donde un proveedor pueda ampliar su línea de productos disponibles a la venta y sobretodo donde el modelo de prestación de servicios de negocio y tecnología permita al usuario acceder a un catálogo de servicios estandarizados y responder a las necesidades de su negocio de forma flexible y adaptativa.

Es aquí donde un sistema basado en arquitectura de la nube, puede hacer la diferencia en un negocio, permitiendo conectividad en diversos sistemas y plataformas sin la necesidad de instalar complementos, y sobretodo, la reducción sustancial de costos de implementación de infraestructuras de procesamiento para PYMES.

1.2 Justificación

El concepto de la nube ha revolucionado la informática en lo referente al desarrollo de aplicaciones. Actualmente, las empresas no necesitan obligatoriamente comprar hardware para alojar las aplicaciones, ni instalar software complementario para que estas funcionen. La oferta ha cambiado: ahora las grandes empresas desarrolladoras de software venden servicios, y al hacerlo, tienen la responsabilidad de que la aplicación funcione,

encargándose del hardware y la instalación del software necesario, brindándole al cliente un mayor grado de disponibilidad.

Con la premisa que las empresas necesitan automatizar la gestión de catálogo eficientemente, el desarrollar una aplicación de software que realice esta función, se acople a cualquier línea de negocio, y funcione como un servicio de la nube, le proporcionará a dicha aplicación, una plusvalía frente a sistemas tradicionales de gestión de catálogo.

En el ámbito del negocio, el presente Proyecto generará un aporte de orden económico a las empresas usuarias, ya que les permitirá diversificar su línea de negocios generando nuevos ingresos al aprovechar las características parametrizables del sistema. Además, se reducirán costos de hardware y licenciamiento de software para el cliente, al definir la aplicación como un servicio.

Dentro del ámbito tecnológico, se establecerá un hito de desarrollo al definir el manejo de los componentes visuales del sistema de forma parametrizable, gracias a la utilización de tecnologías como ADF (Oracle Application Development Framework), así como la relevante importancia en materia de seguridad al utilizar un directorio LDAP (Lightweight Directory Access Protocol) para el manejo de usuarios sobre un servidor de aplicaciones Oracle Weblogic.

1.3 Formulación del problema

En la actualidad las empresas requieren generar mayores ingresos de forma eficiente y, se enfrentan al reto del despliegue, gestión e interoperabilidad de su plataforma de ventas sobre diversas plataformas, lo cual genera grandes inversiones en la adquisición de hardware para el procesamiento de la aplicación y altos lapsos de inactividad frente a recuperación de desastres; además, un alto costo en desarrollo de software con objeto de la diversificación de la línea de negocio.

La implementación de arquitectura en la nube aplicada a la gestión de catálogo de productos permitirá a las organizaciones reducir tanto recursos financieros en la adquisición de hardware y software, tiempo de inactividad frente a la recuperación de desastres y costos en el desarrollo de software, frente a la diversificación de la línea de negocio.

1.4 Hipótesis

La implementación de arquitectura en la nube aplicada a la gestión de catálogo de productos, permitirá a las organizaciones pequeñas y medianas reducir recursos financieros en la adquisición de hardware y software; reducir el tiempo de inactividad frente a la recuperación de desastres y eliminar los costos de modificación de software frente a la diversificación de la línea de negocio.

1.5 Objetivo general

- Implementar un sistema piloto de arquitectura en la nube aplicada a la gestión de catálogo de productos.

1.6 Objetivos específicos

- Definir un modelo viable de arquitectura en la nube para la gestión de catálogo de productos basado en Oracle Application Development Framework (Oracle ADF).
- Implementar las metodologías Programación Extrema y Método Unificado de Oracle en el desarrollo de un sistema de gestión de catálogo de productos.
- Definir la administración de seguridad mediante Lightweight Directory Access Protocol (LDAP) definido en un servidor de aplicaciones Oracle Weblogic.

1.7 Alcance

1.7.1 Arquitectura

Se utilizará como diseño primario la arquitectura de la nube de Oracle, como muestra la Figura 1.1.

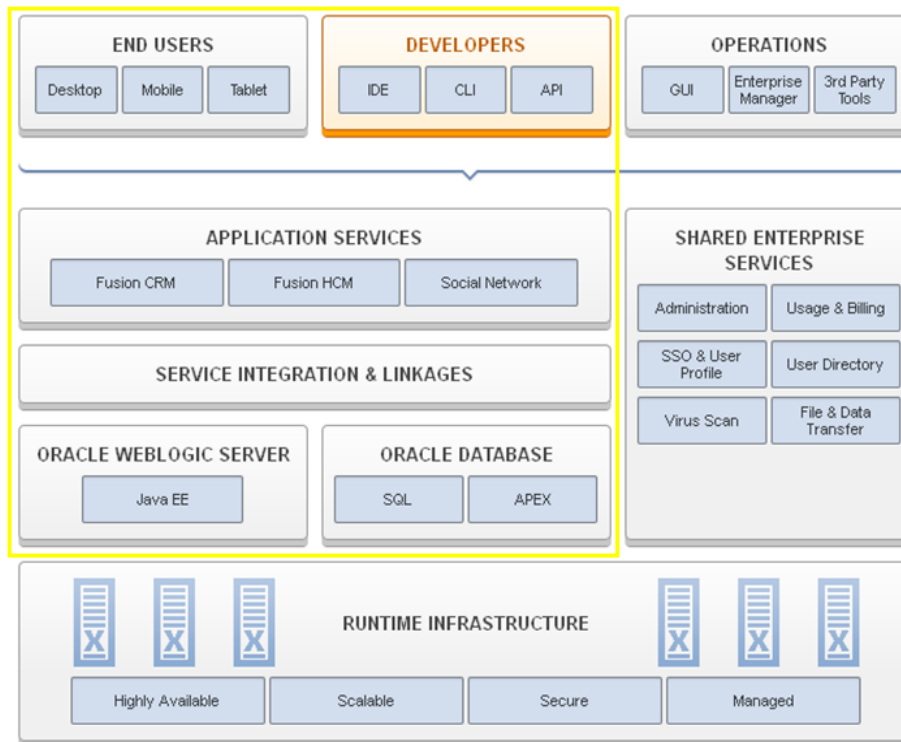


Figura 1.1¹: Arquitectura en la nube de Oracle

Se desarrollarán los siguientes segmentos de la arquitectura de la nube de Oracle:

- **Oracle WebLogic Server:** Se utilizará Oracle Weblogic Server como el servidor de aplicaciones.
- **Oracle Database:** Como base de datos se utilizará Oracle Database.
- **Developers:** Se utilizará el IDE² de desarrollo JDeveloper más el framework ADF.

¹ ORACLE. (2011). Cloud Architecture. Recuperado el 19 de Febrero de 2012, de: <http://cloud.oracle.com/mycloud/f?p=service:architecture:0>.

² Entorno de desarrollo integrado, por sus siglas en inglés.

- **EndUser:** Que se define como el cliente accediendo a la aplicación desde cualquier dispositivo con conectividad a internet.
- **ApplicationServices:** Sistema piloto de gestión de catálogo de productos.

El catálogo de productos permitirá al cliente ingresar su usuario y contraseña; seleccionar uno o varios productos y añadirlos a un carro de compras. Esto generará una transacción denominada *lista de pedidos*, en la cual se detallará el código de pedido, la persona que hizo el pedido y el detalle del mismo. A esta lista únicamente tendrá acceso un miembro del equipo de despacho con rol *empleado*.

1.7.2 Módulos

La aplicación contará con los siguientes módulos:

Pedido

- **Catálogo de productos:** En este módulo estarán definidos los productos y artículos disponibles a la venta; en esta sección se podrá realizar el pedido de productos.
- **Lista de pedidos:** En esta sección se registrará los pedidos hechos por clientes especificando el usuario, el detalle de la compra y el monto.

Personas

- **Personas:** En esta transacción se almacenarán los nombres, direcciones y teléfonos de los usuarios del sistema, así como el nombre de registro en el directorio LDAP.
- **Estructura de la organización:** En esta transacción se definirán cargos operativos de la empresa, de modo gráfico, como una estructura jerárquica.
- **Lista de precios:** Esta transacción almacenará n catálogos, los cuales tendrán diversos precios para los mismos artículos.

- **Relaciones:** Esta transacción tiene como funcionalidad atar un catálogo a un rol (empleado, cliente o proveedor) dentro de la línea de negocios.

Generales

- **Transacciones:** La funcionalidad de esta sección será definir una lista de transacciones y su ruta de acceso.
- **Roles:** En esta transacción se definirán los roles para los diversos usuarios.
- **Menús:** Esta transacción definirá una lista de menús que pueden ser configurables en el sistema.
- **Módulos:** Esta transacción definirá los módulos del sistema que albergarán los menús y transacciones definidos en el sistema.
- **Transacciones por rol:** En esta sección se podrá habilitar transacciones en menú para un rol.
- **Transacciones por menú:** En esta sección se podrá habilitar transacciones para un menú.
- **Menús por módulo:** Esta transacción permitirá definir menús para cada módulo.

Inventario

- **Unidades de productos:** Esta transacción permitirá definir unidades para registrar los productos.
- **Instancias de características:** Esta transacción permitirá definir valores puntuales para las características de un producto albergadas por un grupo.
- **Stock de artículos:** Esta transacción permitirá registrar la cantidad de artículos que existen por producto.
- **Productos:** En esta transacción se podrá insertar, editar y eliminar productos del catálogo.

- **Características por productos:** Esta transacción permitirá relacionar características a un producto.
- **Valores de características:** Esta transacción se encargará de asignar los posibles valores a las características de los artículos.
- **Categorías de productos:** Esta transacción permitirá administrar las categorías de productos.
- **Características:** Esta sección permitirá administrar la creación de artículos.
- **Grupos de características:** En esta sección se podrá definir conjuntos de características que potencialmente albergarán valores de características definidos.
- **Artículos:** En esta transacción se definirá productos atados a características con valores definidos.

1.8 Metodología

Este proyecto será desarrollado fundamentado en dos metodologías: Programación Extrema y Método Unificado Oracle.

La Programación Extrema se fundamenta en la comunicación, simplicidad, retroalimentación, coraje y respeto. Esta metodología es aplicada mediante reglas y mejores prácticas como el cliente en el sitio de desarrollo, el desarrollo iterativo e incremental, la planificación como una tarea constante, y la programación en parejas, las cuales se ajustan a las circunstancias en las que se desarrollará el presente Proyecto, promoviendo un desarrollo ágil y una solución informática de calidad.

Para complementar el desarrollo ágil que posibilitará la metodología Programación Extrema, se escogió el Método Unificado Oracle, por su bien definida estructura en cuanto a los recursos que ofrece para el desarrollo y documentación de proyectos Oracle. El Método Unificado Oracle está basado en el Proceso Unificado Ágil (RUP), siendo una

metodología que rescata los modelos vigentes de RUP adaptándolos a la nueva tendencia de desarrollo ágil, soportando UML³. Lo característico de este Método es que expone las mejores prácticas y normas de calidad Oracle, por lo cual se lo considera basado en estándares.

De esta manera, se buscará una codificación ágil del sistema de gestión de catálogo de productos mediante Programación Extrema, y una documentación sólida mediante el Método Unificado Oracle.

Referencia:

- ANEXO A. CARACTERÍSTICAS Y USO DE METODOLOGÍAS

1.9 Factibilidad

1.9.1 Técnica y Económica

El sistema será desarrollado utilizando los recursos proporcionados por la compañía eMergeSoft y alojado en el servidor de dicha empresa.

1.9.2 Software

El costo es 0 USD ya que Oracle no cobra licenciamiento durante el desarrollo; posteriormente los costos de licenciamiento son asumidos por parte del cliente cuando el sistema esté en producción.

³ Lenguaje Unificado de Modelado, por sus siglas en inglés.

Cuadro 1.1: Costos de software en el Proyecto

Detalle	Precio (dólares)	Cantidad	Total
Oracle JDeveloper	0	1	0
Oracle Weblogic Server	0	1	0
Oracle Linux	0	1	0
Oracle Database	0	1	0
Total	-	-	0

1.9.3 Hardware

Cuadro 1.2: Costos de hardware en el Proyecto

Detalle	Precio (dólares)	Cantidad	Total
Computador	800	2	1600
Impresora	120	1	120
Total	-	-	1720

1.9.4 Administración

Cuadro 1.3: Costos de administración en el Proyecto

Detalle	Precio	Cantidad	Total
Internet 1024/256 KBPS	35	4	140
Resma A4	5	4	20
Disco portable 320 GB	100	2	200
CD	0,50	20	10
Otros	20	N/A	20
Total	-	-	390

- El ítem “Otros” incluye 2 lápices, 2 esferos, 2 borradores, 2 correctores y 2 cuadernos.

1.9.5 Operativa

- La capacitación (asesoría técnica y de negocio) en Oracle Weblogic, administración de base de datos Oracle y desarrollo con Oracle ADF correrá por cuenta de la empresa eMergeSoft.

CAPÍTULO 2

MARCO TEÓRICO

2.1 Realidad de la empresa

eMergeSoft es una empresa radicada en la ciudad de Quito con 6 años en el mercado; cuenta con personal altamente calificado y certificado en tecnología Oracle. Sus principales actividades son: la consultoría y soporte en el sistema de base de datos Oracle (versiones 8i, 9i, 10g y 11g), Oracle Weblogic 11c y Oracle Linux 5; así como el desarrollo de sistemas de inventario y facturación basados en Oracle ADF.

eMergeSoft observó la necesidad de sus clientes, de la gestión de catálogo de productos en ambiente web, con seguridad administrada por el servidor de aplicaciones, y parametrización de productos en catálogo de acuerdo a la línea de negocio. Así nació la idea de desarrollar un software que gestione el catálogo de productos mediante el uso de la arquitectura de computación en la nube de Oracle.

2.2 Arquitectura de la nube de Oracle (Oracle Cloud Computing)

2.2.1 Introducción

En la década de los 90 apareció Internet. A raíz de Internet, la comunicación entre las personas comenzó a darse con una facilidad que no se había experimentado hasta el momento.

Simultáneamente, se comenzaban a desarrollar servidores con grandes capacidades de procesamiento y almacenamiento. Esto condujo a una nueva perspectiva del Internet: la comunicación entre las personas pasó a un segundo plano, y la visión de la potencialidad

de este, avanzó hacia la capacidad de mantener verdaderos sistemas informáticos con centros de datos remotos para gestionar servicios de información y aplicaciones.

Con esto, se abrió una nueva posibilidad de transformación para las aplicaciones tradicionales de software, pudiendo convertirse en servicios para ser consumidos desde cualquier dispositivo con conexión a Internet, el cual se metaforiza con una nube.

La nueva perspectiva favorece tanto a los desarrolladores de software como a los usuarios. Ambos obtienen dos beneficios directos al aplicar computación en la nube: desaparece el proceso de instalación del sistema y complementos necesarios para su funcionamiento, y se utiliza únicamente la cantidad necesaria de recursos (memoria, procesamiento, almacenamiento, ancho de banda). Esto conlleva una mayor velocidad en el desarrollo de aplicaciones, mayor innovación y reducción de costos, mientras que el nivel de versatilidad del negocio es más alto.

El paradigma está ya definido. De acuerdo al sistema que se necesite implementar, se implementarán las arquitecturas de la nube apropiadas.

El presente proyecto utiliza la arquitectura en la nube de Oracle (Oracle Cloud Computing), objeto de estudio del presente trabajo.

2.2.2 Definición de la computación en la nube

La computación en la nube es un paradigma que permite ofrecer servicios de computación a través de Internet⁴, del cual la nube es la metáfora, como una abstracción de la infraestructura que representa. De esta manera, cualquier dispositivo con acceso a Internet está en posibilidad de acceder a los servicios que este alberga y consumirlos, sin necesidad de conocer cómo estos funcionan a nivel técnico, como muestra la Figura 2.1.

⁴ WIKIPEDIA. (2012). Computación en la nube. Recuperado el 18 de febrero de 2012, de: http://es.wikipedia.org/wiki/Computaci%C3%B3n_en_nube.

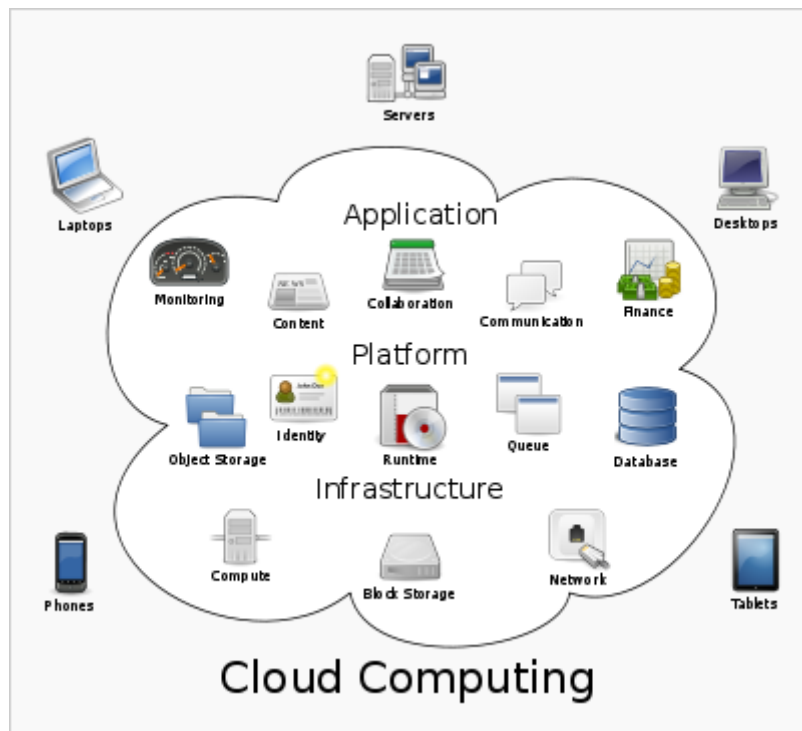


Figura 2.1⁵: Consumo de servicios en la nube

Categorías de la computación en la nube

La computación en la nube puede brindar servicios en 3 categorías:

- Software
- Plataforma
- Infraestructura

En cuanto al **software**, la computación en la nube ha demostrado ser eficaz como modelo de negocio. El software ejecutado usando servidores centralizados en Internet produce una reducción de costos importante. Asimismo, se eliminan gastos de mantenimiento, licencias y hardware necesarios para mantener dichos servidores. De esta manera, las empresas ejecutan aplicaciones de manera más fluida y eficiente.

⁵ WIKIPEDIA. (2012). Computación en la nube. Recuperado el 18 de Febrero de 2012, de: http://es.wikipedia.org/wiki/Computaci%C3%B3n_en_la_nube

En la sección de **plataforma** (Plataforma como servicio, Platform as a service), permite a los usuarios acceder a plataforma (con sus aplicaciones) en servidores centralizados, usando como base la infraestructura de la nube. Igualmente, hay reducción de costos, así como mayor facilidad de administración de las capas de hardware y software.

El segmento de la **infraestructura** (Infraestructura como servicio, Infrastructure as a service) es la parte fundamental del concepto, ya que permite a los clientes crear y usar aplicaciones, sin mantener centros de datos o servidores, sino accediendo a un servicio externo que es cobrado según los recursos consumidos (como tráfico de datos o tiempo).

Ventajas de la computación en la nube

La computación en la nube tiene las siguientes ventajas:

- **Rápida.-** Los servicios más básicos funcionan por sí solos y si fuera necesario servicios de software y bases de datos más complejos, ya no es necesario el proceso de adquisición de hardware con el impacto económico que esto representa.
- **Actual.-** Facilita la disponibilidad de las actualizaciones de software, elástica, móvil y económica.
- **Elástica.-** Puede crecer soportando grandes cargas de trabajo y estirándose sólo lo necesario, con lo cual se optimizan los recursos y por tanto su costo. Es móvil por su diseño que permite el acceso desde cualquier lugar a través de Internet.
- **Económica.-** Los clientes acceden a una infraestructura de la cual pagan exclusivamente lo que consumen.

Tipos de nube

Los tipos de nube que pueden ser utilizadas son:

- **Públicas.-** Es el tipo estándar, con servidores externos al usuario

- **Privadas.-** Mantienen la plataforma en las instalaciones de la empresa (cliente), siendo de uso exclusivo de esta, brindándole alta protección de datos y ediciones a nivel de servicio; comúnmente se usa para administración de hardware (máquinas, almacenamiento, infraestructura), pero también de software (aplicaciones).
- **Híbridas.-** Combinan recursos de una nube privada (infraestructura) con una nube pública (servicios), lo cual permite el control de aplicaciones propias y el uso de la nube pública sólo cuando es necesario.

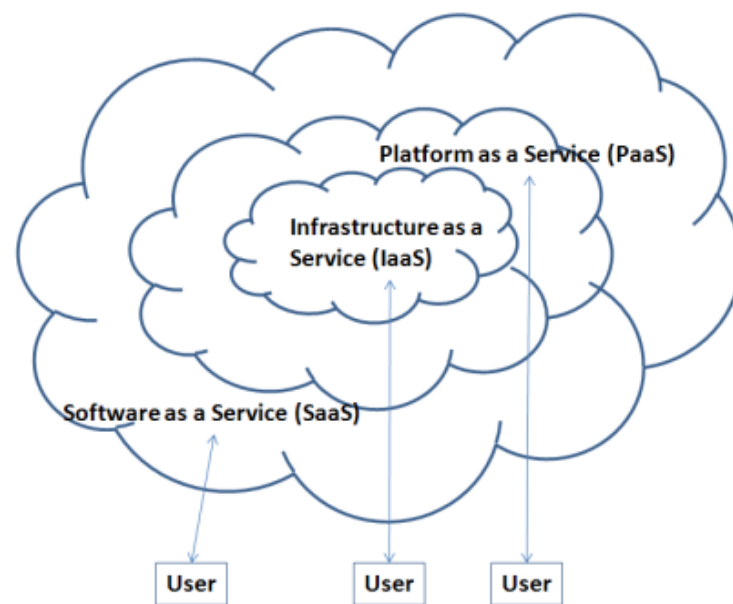


Figura 2.2⁶: Niveles de la nube

Como muestra la Figura 2.2, el usuario puede acceder a niveles más básicos o más profundos técnicamente en la nube. Ordenados desde el más básico hacia el más complejo: software, plataforma e infraestructura.

⁶GARCÍA J. TECHNOLOGY EVALUATION. (2010). Inteligencia empresarial. Recuperado el 19 de Febrero de 2012, de: http://www2.technologyevaluation.com/es/Research/ResearchHighlights/BusinessIntelligence/2010/09/research_notes/es/prn_PN_BI_JG_08_02_10_SP_1.asp.

2.2.3 Definición de Oracle Cloud Computing

La visión de Oracle sobre la Computación en la Nube, incluye de manera general los diferentes tipos de nube que existen y los diversos tipos de aplicaciones que se pueden desarrollar. De manera específica, Oracle busca el aprovisionamiento del servidor, el almacenamiento, la red y el software para soportar esta tecnología. Oracle se basa en el principio que la eficiencia es sobre todo.

La computación en la nube puede implementarse usando servidores de centros de datos (datacenters) empresariales, o mediante un servidor de la nube, que asuma todo el riesgo sobre la infraestructura.

Modelos de infraestructura

Para cambiar de un modelo de implementación de aplicación tradicional a uno basado en computación en la nube, se debe considerar varios aspectos. Escoger entre tipos de nube pública y privada, así como entre los APIs abiertos y los propietarios.

Nubes públicas, privadas e híbridas

Las empresas eligen en qué tipo de nubes implementar sus soluciones informáticas. Sin embargo, como principios básicos para esta elección, se pueden enunciar:

- Si la solución es temporal, lo mejor sería colocarla en una nube pública. Esto evita la compra de equipos de infraestructura para mantener la aplicación localmente.
- Si la solución es permanente y se necesita controlar requerimientos específicos acerca de la calidad del servicio o de la ubicación de datos, lo mejor sería colocarla en una nube privada o en una híbrida.

Las **nubes públicas** son administradas por terceros. Las aplicaciones de varios clientes se mezclan en los servidores, sistemas de almacenamiento y redes. Estas nubes suelen estar ubicados lejos de las instalaciones de las empresas. Los clientes las suelen escoger para reducir el riesgo y el costo, buscando una infraestructura flexible, pudiendo llegar a ser una extensión temporal de la infraestructura de la empresa.

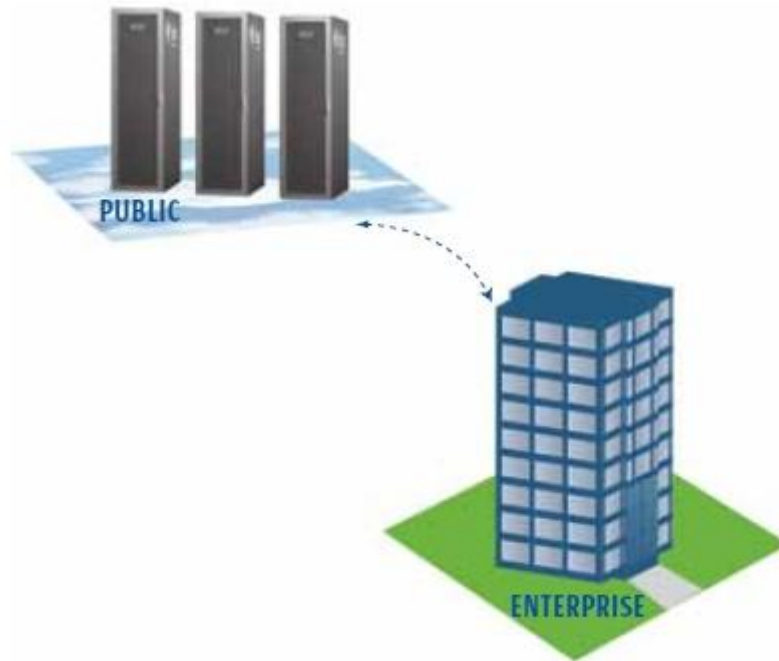


Figura 2.3⁷: Nube pública

Como muestra la Figura 2.3, una nube pública presta sus servicios a varios clientes, y es comúnmente implementado como un servicio de colocación.

Las **nubes privadas** son administradas y usadas por un solo cliente, dándole el control total sobre los datos, la seguridad y la calidad del servicio, ya que el cliente usa su propia infraestructura y controla cómo las aplicaciones son implementadas en esta. La gestión de este tipo de nubes puede ser hecha por los profesionales de tecnología de la propia empresa o mediante el modelo de alojamiento privado (hosted private model), donde un proveedor

⁷ SUN MICROSYSTEMS. (2009). Introduction to Cloud Computing Architecture. (10). Recuperado el 18 de Febrero de 2012, de: <http://www.oracle.com/us/dm/44034-cloudcomputing-332069.pdf>

de la nube (cloud provider) ofrece la instalación, configuración y operación de la infraestructura de la computación en la nube en el datacenter del cliente, brindándole la posibilidad de controlar los recursos de la nube, así como tener un constante crecimiento en experiencia acerca del establecimiento y operación del ambiente.

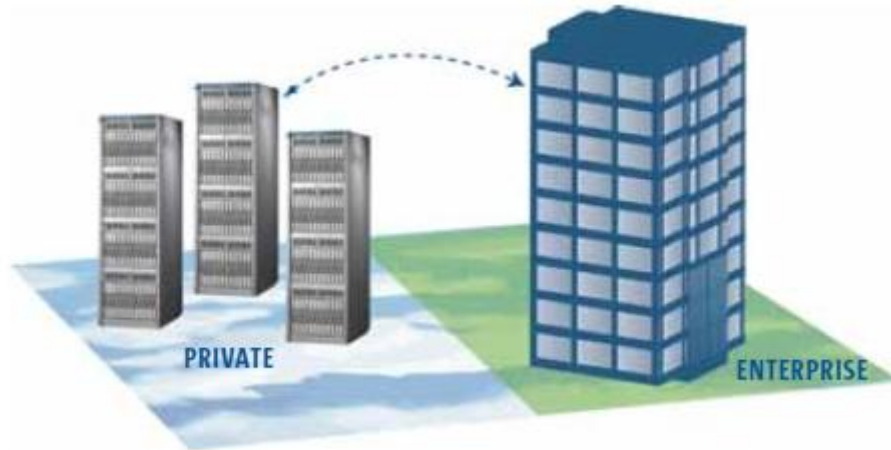


Figura 2.4: Nube privada

Como muestra la Figura 2.4, las nubes privadas están en control de un solo cliente. Este puede gestionar el establecimiento y operación por sí mismo o contratando a un proveedor de la nube (cloud provider).

Las **nubes híbridas** combinan los modelos público y privado. Este modelo de nube puede ser necesario en las siguientes circunstancias:

- Adicionar servicios de la nube pública a los de la nube privada, para mantener los niveles de servicio en aplicaciones con rápidas oscilaciones de la carga de trabajo, por ejemplo las nubes de almacenamiento que soportan las aplicaciones web 2.0⁸.
- Manejar los picos de cargas de trabajo planificadas.

⁸ Aplicaciones web enfocadas al usuario final con el objetivo de generar colaboración. El usuario suele manejar el contenido.

- Determinar la distribución más adecuada de las nubes públicas y privadas. Para esto se debe tener en cuenta la data y los recursos de procesamiento.

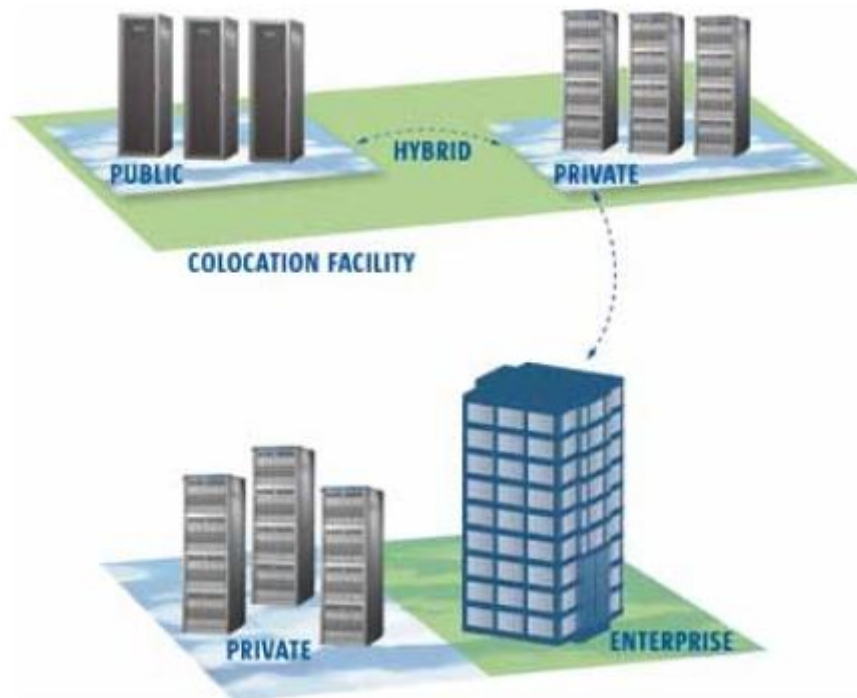


Figura 2.5⁹: Nube híbrida

Como muestra la Figura 2.5, las nubes híbridas combinan el modelo de nube pública y el de nube privada. Son más eficientes cuando ambos tipos de nube se encuentran ubicados en la misma implementación.

2.2.4 Capas de la arquitectura en la nube de Oracle

Los proveedores de servicios de la nube, generalmente ofrecen 3 categorías:

- Software como servicio
- Plataforma como servicio
- Infraestructura como servicio

⁹ SUN MICROSYSTEMS. (2009). Introduction to Cloud Computing Architecture. (11). Recuperado el 18 de Febrero de 2012, de: <http://www.oracle.com/us/dm/44034-cloudcomputing-332069.pdf>.

Software como servicio (Software as a service – SaaS)

Se trata de una aplicación de software en la nube, a la que pueden acceder múltiples clientes, comúnmente a través de exploradores web (web browsers). Entre las aplicaciones más destacadas se puede nombrar las aplicaciones de Google (Goggle Apps).

Oracle por su parte pone a disposición el Oracle CRM¹⁰ on Demand (Oracle CRM bajo demanda) con dos opciones: múltiples usuarios o un solo usuario. Oracle también da la posibilidad a proveedores de software independientes, de desarrollar sus propias aplicaciones y ofertarlas como SaaS. Esto es llamado por Oracle como *Oracle Platform for SaaS* (Plataforma Oracle para SaaS) y tiene cientos de clientes.¹¹

Plataforma como servicio (Platform as a service – PaaS)

La plataforma en la nube ofrece un servicio que permite desarrollar, gestionar y mostrar aplicaciones SaaS. La plataforma incluye:

- Bases de datos
- Capa media (middleware¹²)
- Herramientas de desarrollo

Las ofertas suelen especificar el lenguaje de programación o el API¹³; por ejemplo Java o Python. La base de dichas ofertas suele ser la arquitectura grid computing¹⁴

¹⁰ Siglas en inglés de Customer Relationship Management. Gestiona la relación de las empresas con sus clientes.

¹¹ DEMARESK, G. WANG, R. (2010) Oracle Cloud Computing. (4). Recuperado el 18 de Febrero de 2012, de: <http://www.oracle.com/us/technologies/cloud/oracle-cloud-computing-wp-076373.pdf>

¹² Software que permite a componentes de software y aplicaciones conectarse e intercambiar datos entre sí.

¹³ Siglas (en inglés) de Application Programming Interface. Es un conjunto de funciones y procedimientos que proporciona una biblioteca para ser usado por otro software como una capa de abstracción.

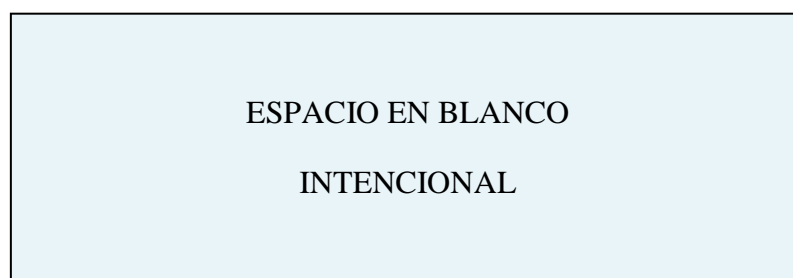
¹⁴ O computación grid. Es una tecnología desarrollada a principios de los años 90, que permite utilizar todo tipo de recursos informáticos sin un control centralizado. Es una nueva forma de computación distribuida donde los recursos están conectados por redes de área extensa como Internet.

(computación grid) virtualizada y agrupada en clústeres¹⁵. La computación grid proporciona escalabilidad y elasticidad, así como la puesta común de recursos (resource pooling). Oracle ofrece PaaS a proveedores de servicios de nube pública, así como a clientes empresariales la posibilidad de construir su propia nube pública. Esto es llamado *Oracle PaaS Platform* (Plataforma PaaS de Oracle).

Infraestructura como servicio (Infrastructure as a service – IaaS)

Este modelo ofrece servidores, almacenamiento y hardware de red como un servicio. Este hardware de infraestructura suele ser virtualizado, por lo cual el software de virtualización¹⁶, así como el sistema operativo, son considerados componentes de IaaS.

Oracle no ofrece servicios en la nube de IaaS. Sin embargo proporciona productos de hardware y software para que otros proveedores de IaaS habiliten sus servicios de nube pública. Estos productos de hardware y software también son ofertados a empresas para uso privado.



¹⁵ Conjunto de computadores que utilizan componentes de hardware comunes y son conectados mediante una tecnología de red. Se comportan como un solo computador.

¹⁶ Creación mediante software de una versión virtual de un recurso tecnológico; por ejemplo, una plataforma de hardware, un sistema operativo, un dispositivo de almacenamiento u otros recursos de red.

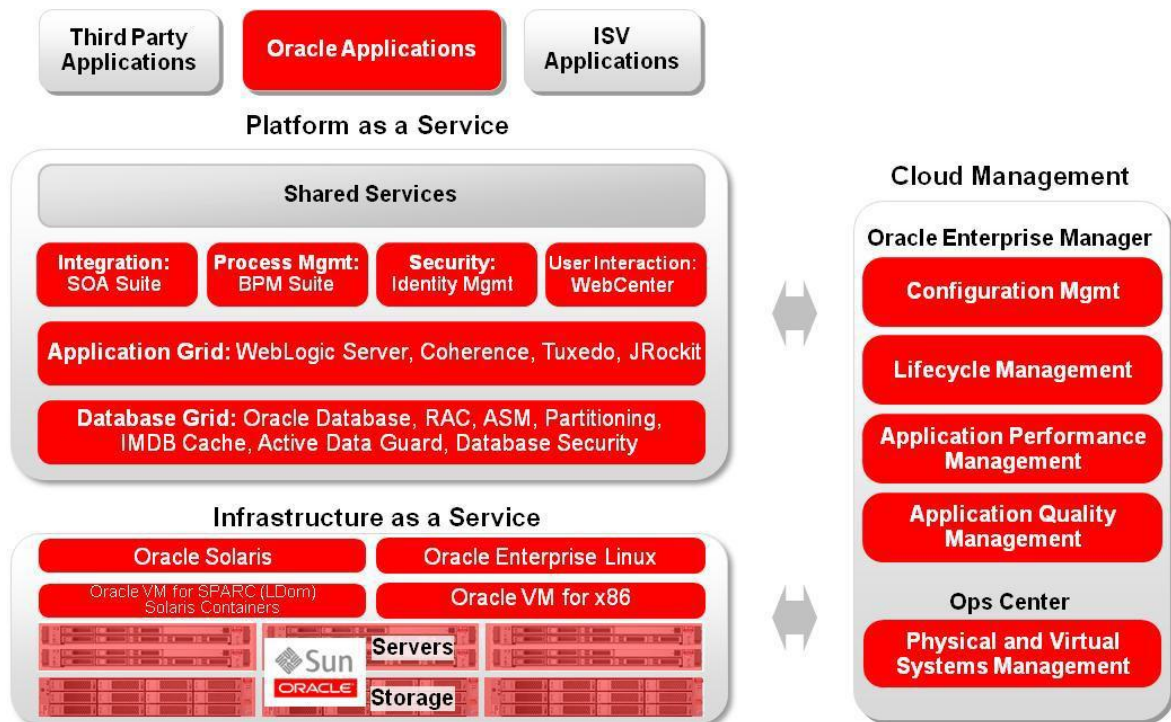


Figura 2.6¹⁷: Componentes para soportar Oracle PasS Platform (Plataforma Oracle PaaS) y Oracle IaaS Platform (Plataforma Oracle IaaS).

2.3 La metodología Programación Extrema (XP)

2.3.1 Introducción

La Programación Extrema (XP por sus siglas en inglés) es una nueva metodología de desarrollo de software surgida al final de la década de 1990, como resultado de una búsqueda de simplicidad y agilidad. Kent Beck es el gurú de la Programación Extrema y el más influyente en su difusión¹⁸. Asimismo, es un referente el caso de éxito obtenido a finales de 1998, en que Beck introdujo XP en un proyecto de desarrollo llamado C3 (Chrysler Comprehensive Compensation) para la firma Chrysler.

¹⁷ DEMARESK, G. WANG, R. (2010). *Oracle Cloud Computing*. (9). Recuperado el 18 de Febrero de 2012, de: <http://www.oracle.com/us/technologies/cloud/oracle-cloud-computing-wp-076373.pdf>.

¹⁸ Debido a los libros que ha publicado, especialmente *XP Explained*, publicado en 1999.

Las metodologías tradicionales se han mostrado en general como muy pesadas y poco eficientes, con un exceso de burocracia que muchas veces retrasa el desarrollo de los sistemas. En respuesta a esto, han surgido las denominadas metodologías ágiles, las cuales buscan un equilibrio entre el abuso de procesos y la ausencia de estos en el ciclo de desarrollo del software. Estas metodologías ágiles se diferencian de las clásicas en que sus métodos son adaptables (no hay resistencia al cambio) y son orientados a la gente y no al proceso (no se asume la capacidad de un equipo para desarrollar un proceso, sino que busca las mejores condiciones para su trabajo).

Dentro de las mencionadas nuevas metodologías ágiles, XP es una de las más exitosas. La misma está diseñada para entregar el software que el cliente necesita cuando lo necesita. XP motiva a los desarrolladores a solventar los requerimientos de los clientes aún en etapas avanzadas del desarrollo. De igual manera, resalta la importancia del trabajo en equipo, donde todos los miembros buscan entregar software de calidad.

2.3.2 Modelo de XP

En las metodologías tradicionales, se realiza una definición de las etapas de desarrollo y un plan para el proceso de desarrollo. XP por el contrario propone un ciclo dinámico donde se prevé que el cliente no defina lo que realmente requiere al principio del proyecto.

XP se basa en iteraciones, que son ciclos cortos de desarrollo, con entregables funcionales al final de cada ciclo. En cada iteración se realiza el análisis, diseño, desarrollo y pruebas, utilizando reglas y prácticas que caracterizan a XP.

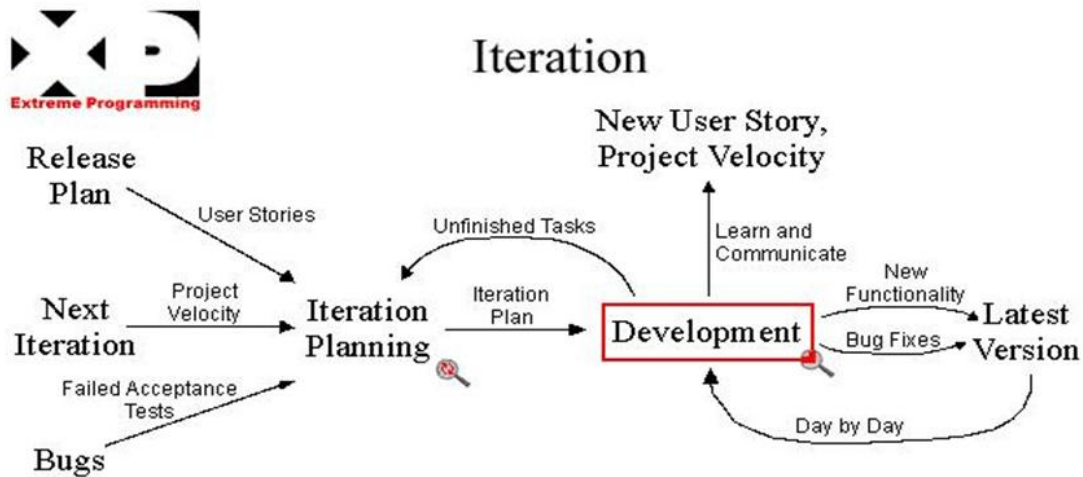


Figura 2.7¹⁹: Modelo de desarrollo en XP

El modelo de XP define 4 variables para cualquier proyecto de software, las cuales están relacionadas entre sí:

- Costo
- Tiempo
- Calidad
- Alcance

Costo

Para definir el costo, se debe tomar en cuenta que la cantidad de programadores que se contrate debe ser incremental; esto facilita la gestión de la comunicación en el equipo, basándose en la premisa de que es más sencillo mantener una comunicación entre pocas personas; si se añade más programadores se debe tener en cuenta el proceso de adaptación al desarrollo. Es importante indicar que XP busca que el costo de solucionar un problema no sea exponencial en el tiempo transcurrido de desarrollo, como sucede en las metodologías tradicionales.

¹⁹ BABELIAS. (2012). Programación extrema. Recuperado el 18 de Febrero de 2012, de: <http://babelias-blog.com/2012/02/programacion-extrema/>.

Tiempo

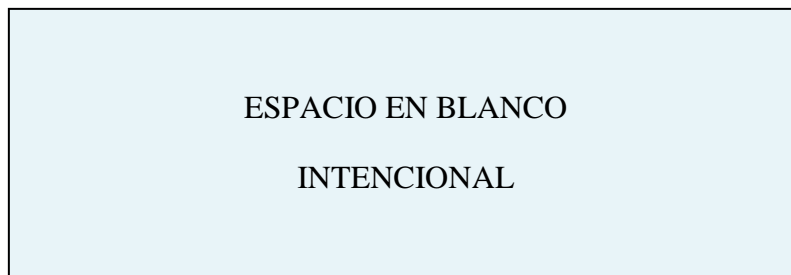
El tiempo se mide acertadamente cuando se mira a un proyecto de manera real, al final del mismo. Por lo tanto, si se hace que un proyecto finalice cada cierto tiempo breve, la estimación de este va a ser mejor. Esto significa la aplicación de programación iterativa e incremental.

Calidad

La calidad se puede medir externa e internamente. La calidad externa es percibida por el cliente, mediante la evaluación de la solventación de sus requerimientos, así como del performance del sistema. La calidad interna es valorada por los desarrolladores, quienes evalúan la claridad de la especificación de requerimientos así como la eficacia de las herramientas de hardware y de software que utilizan.

Alcance

El alcance es la variable más importante a especificar, debido a que definirá el sistema como tal. Mientras el alcance sea menor, la calidad tiende a ser mayor. La negociación del alcance entre el equipo debe enfocarse a acordar el menor alcance posible, debido a que siempre el cliente establece un alcance mayor al que realmente necesita.



2.3.3 Fases en XP

Aunque XP es una metodología dinámica, se pueden observar 4 fases en su desarrollo.

Tabla 2.1: Fases de la Programación Extrema con sus características y resultados.

Fase	Características	Resultado
Exploración	<ul style="list-style-type: none"> - Se define el alcance general del proyecto. - El cliente redacta <i>historias de usuario</i>, que son notas en lenguaje directo y no técnico con sus requerimientos. - El equipo desarrollador realiza la estimación del tiempo que le tomará por cada historia de usuario. Esta es una estimación primaria que puede variar cuando se analice más profundamente en las iteraciones. 	Visión general del sistema con el plazo estimado para desarrollarlo.
Planificación	<ul style="list-style-type: none"> - El equipo acuerda la prioridad de desarrollo de las historias de usuario y las entregas respectivas. 	Plan de entregas (Release plan)
Iteraciones	<ul style="list-style-type: none"> - Fase principal en el ciclo de desarrollo de XP. - El cliente participa activamente. Se analizan las historias de usuario con él para obtener el detalle del que las historias carecen en un inicio. - Se desarrollan las funcionalidades con su respectivo entregable (implementación de las historias asignadas a la iteración). - Son utilizadas para medir el avance del 	Plan de iteraciones (Iteration plan) y funcionalidades desarrolladas

	proyecto. Un entregable sin errores indica en gran manera si un proyecto avanza correctamente.	
Puesta en producción	<ul style="list-style-type: none"> - Implementación de la funcionalidad completa del sistema, uniendo los módulos funcionales que fueron entregados sin errores. - No se realizan ya desarrollos funcionales, pero sí se puede realizar ajustes (fine tuning). 	Versión del sistema

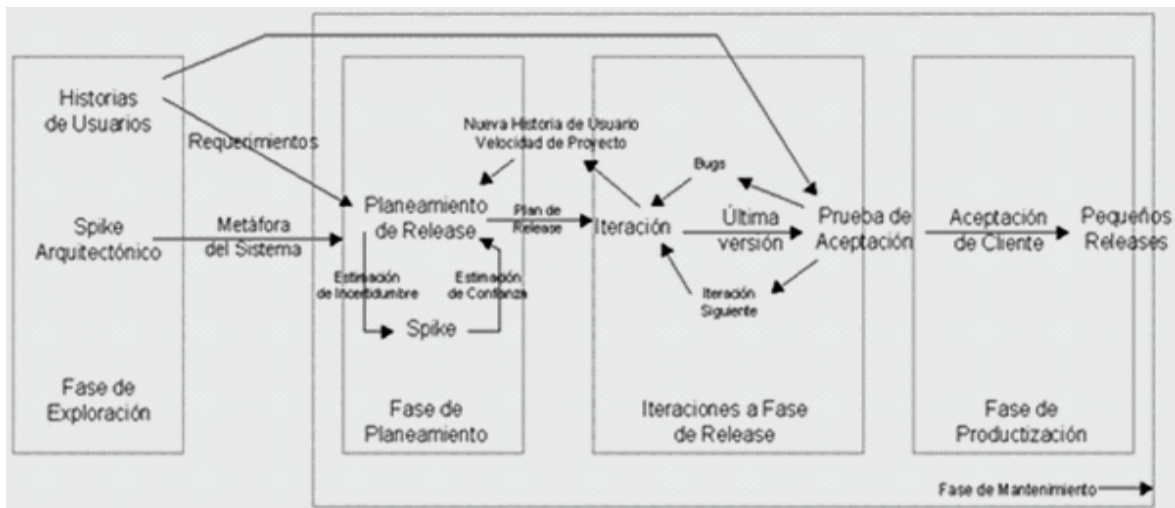


Figura 2.8²⁰: Fases de XP

2.3.4 Roles en XP

En XP existen 3 roles principales dentro del equipo de trabajo:

- Cliente
- Gerente de proyecto (también llamado jefe de proyecto)
- Desarrolladores

²⁰ VILLEGAS, A. A propósito de programación extrema XP (eXtreme Programming). Recuperado el 18 de Febrero de 2012, de: <http://www.monografias.com/trabajos51/programacion-extrema/programacion-extrema2.shtml>.

Derechos y responsabilidades del cliente

Los derechos y responsabilidades del cliente son:

- Decidir qué se implementa y con qué prioridad.
- Conocer el estado actual del proyecto y su progreso.
- Modificar su especificación de requerimientos en cualquier momento.

Derechos y responsabilidades del gerente de proyecto

Los derechos y responsabilidades del gerente del proyecto son:

- Conocer el estado actual del proyecto y su progreso.
- Supervisar la aplicación de la metodología.
- Verificar el cumplimiento de cronogramas.
- Verificar la calidad del software.
- Reunirse con cliente y desarrolladores en cualquier momento.
- Tomar decisiones en cuanto al recurso humano (contrataciones y despidos).

Derechos y responsabilidades de los desarrolladores

Los derechos y responsabilidades de los desarrolladores son:

- Decidir cómo se implementan las funcionalidades.
- Crear el software con la mayor calidad posible.
- Recibir aclaraciones del cliente acerca de sus requerimientos en cualquier momento.
- Estimar el esfuerzo y el cronograma de entregas en base a su experiencia en anteriores proyectos, en el proyecto actual o con ayuda de desarrolladores amigos.
- Modificar las estimaciones en base a nuevos hallazgos durante el desarrollo.

En XP se pueden establecer también otros roles, de acuerdo al recurso humano disponible. Estos roles son:

- Probador
- Rastreador
- Entrenador
- Consultor

Derechos y responsabilidades del probador

- Ayudar al cliente a escribir pruebas funcionales.
- Verificar la ejecución de las pruebas funcionales regularmente.

Derechos y responsabilidades del rastreador

- Cerrar el ciclo de retroalimentación indicando el estado de las estimaciones de tiempo actuales.
- Realizar proyecciones a futuro acerca de si los objetivos serán cumplidos.
- Lleva un registro historial de las pruebas funcionales y los errores reportados.

Derechos y responsabilidades del entrenador

- Guiar a los miembros del equipo a aplicar las reglas y mejores prácticas de XP de acuerdo a las circunstancias en que se encuentren.
- Disminuir la dependencia del equipo en él progresivamente.

Derechos y responsabilidades del consultor

- Asesorar en el área técnica en caso de que se presenten dudas o inconvenientes.

En caso que no se cuente con el recurso humano, se puede asignar la función de estos roles a los existentes, de acuerdo a los atributos de las personas.

2.3.5 Reglas y prácticas de XP

Las reglas y prácticas caracterizan en gran manera a XP. Estas se pueden agrupar en 4 grupos:

- Reglas y prácticas para la Planificación
- Reglas y prácticas para el Diseño
- Reglas y prácticas para el Desarrollo
- Reglas y prácticas para las Pruebas

Reglas y prácticas para la Planificación

La planificación en XP se diferencia a la de las metodologías tradicionales en lo siguiente:

- No es un evento, sino una tarea continua y dinámica. No son predicciones del futuro sino estimaciones que pueden ser modificadas de acuerdo a las circunstancias.
- Es simple, ya que no se requiere un experto en sistemas de gerenciamiento de proyectos para realizarla.
- Es realizada por quienes luego desarrollarán el proyecto.

Se deben tomar en cuenta las siguientes reglas y prácticas, las cuales deben ser realizadas cronológica e iterativamente:

- Establecer un diálogo continuo entre los miembros del equipo.
- Recopilar historias de usuario que sustituyen a los tradicionales casos de uso.
- Los programadores deben estimar el tiempo en que desarrollarán las historias de usuario.

- Si en alguna historia de usuario no se puede establecer la complejidad del desarrollo de la misma, se deben realizar pequeños programas de prueba llamados *spikes*. Esto reduce los riesgos de percepción.
- El equipo se debe reunir para establecer el Plan de Entregas (Release Plan).
- Proceder a la fase de iteraciones. En cada iteración se desarrolla, prueba e instala historias de usuario que formen una funcionalidad.

Historias de usuario (user stories)

Se utilizan las historias de usuario, las cuales tienen las siguientes características:

- Reemplazan a los documentos de especificación funcional y a los casos de uso.
- Son fichas escritas por el cliente en su propio lenguaje, conteniendo sus requerimientos en una descripción corta.
- Los desarrolladores hacen una breve estimación del tiempo de trabajo. En la implementación se habla en detalle de las historias de usuario con respecto a este aspecto con el cliente.
- Son desarrolladas en un tiempo entre 1 y 4 semanas. Si la estimación es mayor a 4 semanas, la historia debe ser dividida en varias; si es de menos de 1 semana, se debe combinar con otra historia.

Plan de entregas (Release Plan)

Además se utiliza un Plan de entregas, el cual tiene las siguientes características:

- Establece qué historias de usuario serán agrupadas para una entrega.
- Establece el cronograma de entregas.
- Es el resultado de la reunión entre los actores conocida como “juego de planeamiento”.

- Comúnmente el cliente ordenará y agrupará las historias de usuario según la prioridad que considere. El cronograma se hace mediante las estimaciones de tiempo realizada por los desarrolladores.
- Posteriormente, de algunas iteraciones, se debe hacer una nueva reunión para revisar el plan y reajustarlo si es necesario.

Plan de iteraciones (Iteration plan)

- Se desarrollan las historias de usuario, probándolas en un ciclo de iteración.
- Al inicio de cada ciclo se realiza una reunión de planificación de las iteraciones. Las historias de usuario se traducen en instrucciones de programación y se establecen las pruebas de aceptación.
- Las pruebas de aceptación se realizan al final del ciclo en cuestión, pero también de los posteriores para verificar que no se vieron afectadas por los mismos.
- Si las pruebas fallan en un ciclo anterior se analiza su corrección y las medidas para que vuelva a ocurrir.

Reuniones diarias de seguimiento (Stand-up meeting)

- Se realizan para mantener la comunicación entre el equipo y compartir problemas y soluciones.
- La mayoría de participantes escuchan, sin aportar detalles.
- Por cuestión de optimizar el tiempo, se hacen en círculo y de pie.

Reglas y prácticas para el Diseño

XP promueve los diseños simples y fáciles de entender. A continuación algunos aspectos importantes.

Simplicidad

- Se busca la facilidad en la implementación.
- No se debe adelantar implementación de funcionalidades que no corresponden a la iteración que se está desarrollando. Se debe confiar en que luego se tendrá la capacidad de implementar lo que sea necesario.

Soluciones “spike”

- Son pequeños programas para realizar pruebas cuando el desarrollo presenta inconvenientes técnicos o cuando es difícil estimar el tiempo.
- Luego de la evaluación correspondiente el programa “spike” es desechado.

Recodificación o refactorización del código

- Buscan mantener el código más simple, conciso y entendible que pueda implementar la funcionalidad deseada. Para ello es necesario escribir nuevamente parte del código.
- Se debe recodificar cada vez que sea necesario.
- Genera una influencia positiva sobre futuras iteraciones cuando sea necesario realizar modificaciones.

Metáforas

- Sirven para explicar el propósito del proyecto y guiar su estructura y arquitectura.
- El equipo debe estar de acuerdo para poder hablar un mismo idioma.

- Suelen ser difíciles de implementar y se cuestiona su uso.

Reglas y prácticas en el Desarrollo del código

Disponibilidad del cliente

- Se requiere tener al cliente disponible durante todo el proyecto.
- El cliente debe involucrarse en el grupo, a más de apoyar a los desarrolladores.
- Al comienzo debe proporcionar las historias de usuarios. Pero los detalles de las mismas deben ser proporcionados por él en la etapa de desarrollo. Estos detalles deben ser proporcionados *cara a cara* a los desarrolladores. En otras metodologías el dar detalles cara a cara es suplantado por la realización de documentos de especificaciones, lo cual es más tedioso y consume más tiempo.
- El cliente puede prevenir desarrollos no deseados, evitando una corrección difícil.

Uso de estándares

- Se busca una programación estandarizada, entendible para el equipo y fácil de recodificar.

Programación dirigida por las pruebas (Test-driven programming)

- Antes de programar, se debe escribir las pruebas que el sistema debe pasar. Estas son las llamadas pruebas unitarias (unit tests).
- El desarrollo debe ser el mínimo para pasar las pruebas previamente definidas; es decir que estas pruebas dirigen el desarrollo.

Programación en pares

- Se deben ubicar dos programadores por ordenador. Mientras uno codifica, el otro supervisa.
- Aunque parezca que se duplica el tiempo y los costos en recursos humanos, se minimizan los errores, lográndose mejores diseños, compensando la inversión en horas y en la calidad del producto que suele ser mejor que con programación individual.
- La mayoría de errores son descubiertos al momento de la codificación al ser dos personas las que lo revisan.
- La cantidad de defectos encontrados en las pruebas estadísticamente es menor.
- El equipo resuelve problemas más rápidamente.
- Las personas aprenden más tanto del sistema como de su desarrollo
- Al final del proyecto, más personas conocen los detalles de cada parte del código.
- Las personas aprenden a trabajar juntas, haciendo al grupo más dinámico, la información fluye de mejor manera y el ambiente laboral es mejor.

Integraciones permanentes

- Los desarrolladores deben trabajar con la última versión para evitar problemas y retrasos en el proyecto.
- Se debe publicar lo antes posible las nuevas versiones, así no sean las últimas, con la condición de que no tengan errores.
- Se aconseja que cada día haya una nueva versión.
- La actualización se hace una a una por cada pareja (no todas a la vez).

Propiedad colectiva del código

- Todo el equipo puede contribuir con nuevas ideas que apliquen a cualquier parte del proyecto.
- Cualquier pareja puede tomar la responsabilidad de hacer un cambio en el código (recodificación) sin negociar con su “dueño”, evitando el viejo paradigma de que “todos son responsables” cuando muchas veces en realidad nadie se hacía responsable.
- Las verificaciones deben ser permanentes para que la solución se mantenga correcta.

Ritmo sostenido

- Tiene como objetivo planificar el trabajo para tener un ritmo constante pero sin sobrecargar al equipo.
- Al retrasarse un proyecto, el trabajar tiempo extra puede resultar perjudicial, ya que se produce una desmotivación en el equipo, reduciéndose la calidad del producto.
- Mientras sea posible, se debe negociar el plan de entregas (Release plan), reuniéndose nuevamente con el cliente, desarrolladores y gerente.
- Si se agregan desarrolladores a proyectos avanzados, se corre el riesgo de que el inconveniente no se resuelva.

Reglas y prácticas en las Pruebas

Pruebas unitarias

Las pruebas unitarias son preparadas por el usuario para verificar la funcionalidad así como las validaciones correspondientes. Se debe seguir las siguientes recomendaciones:

- Todos los módulos deben pasar las pruebas unitarias antes de ser publicados.

- Las pruebas deben ser definidas antes de la codificación.
- Habilita la propiedad colectiva de código
- El sistema, el código y las pruebas deben ser guardados en caso de que otros programadores necesiten modificar código.

Detección y corrección de errores

- Al encontrar un error (bug) se debe proceder inmediatamente a su corrección.
- Se debe tomar precauciones para que errores similares no se vuelvan a dar.
- Se deben generar nuevas pruebas para verificar la corrección del error.

Pruebas de aceptación

- Son creadas en cada ciclo de iteración basándose en las historia de usuarios.
- El cliente debe definir uno o varios escenarios para verificar que una historia de usuario ha sido correctamente implementada.
- Son consideradas como *pruebas de caja negra* (“black box system tests”).
- Los clientes están encargados de cerciorarse que los resultados de estas pruebas sean correctos.
- Si varias pruebas fallan, el cliente debe indicar la prioridad de resolución.
- Cuando una historia de usuario pasa por todas las pruebas de aceptación puede considerarse como terminada.
- Se aconseja publicar los resultados de las pruebas para que todo el equipo tenga conocimiento de esto.

2.3.6 Valores en XP

Para que el proyecto tenga éxito, el equipo debe tener los siguientes valores:

- Comunicación
- Simplicidad
- Retroalimentación
- Coraje

Comunicación

La comunicación permanente evita problemas. Debido a que la documentación es escasa, el diálogo cara a cara entre los miembros del equipo es el método fundamental para la comunicación.

Simplicidad

XP pone énfasis en la sencillez en todo lo que se haga. Esto hace que el código sea entendible y más fácil de mejorar mediante recodificaciones continuas.

Retroalimentación

La retroalimentación debe ser permanente. Esta viene dada por los comentarios del cliente acerca de la funcionalidad desarrollada y acerca de la siguiente iteración. Además, por los resultados de las pruebas unitarias, que les sirve a los desarrolladores para evaluar su trabajo.

Coraje

Si se presentan graves problemas en el diseño o en el desarrollo, se debe tener el coraje suficiente para solucionarlo, sin importar que esto implique desechar algo en lo que se había consumido gran cantidad de tiempo y esfuerzo.

Estos cuatro valores de XP deben ser desarrollados en un ambiente de **respeto** mutuo entre los miembros del equipo, el cual puede ser considerado el quinto valor de la metodología.

2.4 Método Unificado Oracle (OUM)

2.4.1 Introducción

El método Unificado Oracle (OUM por sus siglas en inglés) es una metodología desarrollada por Oracle, aparecida en 2006.

OUM es un método basado en estándares, diseñado para soportar todo el ciclo de vida del software. La implementación de OUM es rápida, adaptable y enfocada al negocio. Oracle provee un proyecto integral y un marco de gestión de programas y materiales de apoyo por parte de Oracle a la empresa de TI en los aspectos de estrategia, arquitectura y gobernanza²¹.

El Método Unificado Oracle presenta de manera organizada pero flexible el enfoque de los proyectos de software con el objetivo de acelerar los mismos y obtener requerimientos críticos y dependencias. Con OUM puede navegar a lo largo del ciclo de vida de software logrando objetivos de negocio cuantificables.

Para aplicar OUM se necesita instalar la herramienta de software correspondiente, el cual está disponible para su descarga en la página web oficial de Oracle. El único requisito para obtenerla es ser partner diamante, platino u oro.²²

²¹ Alineamiento de las TI con las metas y estrategias del negocio, heredándolas a todos los departamentos de la empresa para que la estructura organizativa trabaje eficientemente en dichas metas y estrategias.

²² ORACLE (2012). Oracle Linux. Recuperado el 20 de Febrero de 2012, de: <http://www.oracle.com/us/technologies/linux/025994.htm>.

2.4.2 Características

El Método Unificado Oracle posee las siguientes características:

- Basado en estándares
- Iterativo e incremental
- Soporta agilidad y disciplina

Basado en estándares

El Método Unificado Oracle se basa en uno de los estándares de facto de la ingeniería de software: el Proceso Unificado. OUM extiende el mismo, acomodándolo según la experiencia de campo y la contribución intelectual de profesionales Oracle.

Iterativo e incremental

El Método Unificado Oracle reconoce los beneficios del desarrollo iteracional por lo cual se alinea al mismo. El método está diseñado para que cualquier de las tareas que plantea puedan repetirse. El objetivo de esto es mejorar las mismas ya sea por un análisis propio o por parte de los clientes.

Soporta agilidad y disciplina

OUM está diseñado para soportar varios tipos de proyectos. Por ello debe ser flexible y escalable. El punto de equilibrio adecuado varía según el riesgo y escalabilidad del proyecto.

2.4.3 Beneficios de OUM

Los beneficios de OUM incluyen:

- Mayor esfuerzo concentrado
- Flexibilidad integrada

- Ahorro de tiempo
- Mayor calidad
- Mayor rentabilidad
- Reducción de riesgos

Mayor esfuerzo concentrado

OUM busca que en los proyectos se definan claramente el negocio y los requerimientos para crear modelos de arquitectura. OUM controla el desarrollo de la aplicación, busca el entendimiento del negocio y brinda una base firme para la aceptación del cliente.

Flexibilidad integrada

Los procesos de ejecución de tareas indicado por OUM se puede aplicar a diferentes tipos de implementación y desarrollo de software.

Ahorro de tiempo

OUM ha sido desarrollado en base a los aportes de profesionales Oracle. Los equipos de proyectos de software tienen a su disposición la valiosa ventaja de poder aprovechar estas prácticas que están basadas en estándares de la industria.

Mayor calidad

OUM incorpora el concepto de iteraciones en las pruebas a lo largo del ciclo de vida del software, no sólo al final del mismo.

Mayor rentabilidad

OUM busca controlar y optimizar los gastos mediante un modelo de desglose que hace que sólo se gaste en lo necesario.

Reducción de riesgos

Las iteraciones que maneja OUM se aplican también en el análisis de riesgos, con lo cual se atiende a los mismos lo más rápido posible según su criticidad, en el ciclo de vida del proyecto.

2.4.4 Características clave

OUM fue diseñado con las siguientes características clave:

- Flexible
- Escalable
- Organizado en vistas

Flexible

OUM permite a la empresa, escoger la estrategia, técnicas y tareas necesarias para su proyecto. OUM da lineamientos necesarios para que el plan del proyecto se adapte a la situación. Al unir actividades y tareas de diferentes maneras, OUM se muestra aplicable a una gran variedad de proyectos de desarrollo de TI y proyectos de implementación.

Escalable

No importa la complejidad del proyecto que se quiera desarrollar, el Método Unificado Oracle proporciona la escalabilidad necesaria para aplicarse a proyectos únicos, ayudando a determinar qué tareas deben incluirse en el plan del proyecto.

Organizado en vistas

El material que el Método Unificado Oracle pone a disposición está organizado en vistas. Las vistas proporcionan un enfoque inicial del plan de trabajo. Cada vista da acceso a una guía y una estructura definida de la estructura de trabajo.

2.4.5 Implementación de un proyecto OUM

Área de enfoque de implementación (Implement focus area)

El Área de enfoque de implementación (Implement focus area) provee un marco de trabajo (framework) para desarrollar aplicaciones Oracle basadas en el negocio. El Método Unificado Oracle utiliza fases y procesos que incluyen puntos de control de calidad y permiten la coordinación de las actividades a lo largo del mismo. En cada fase, el equipo realiza tareas en varios procesos.

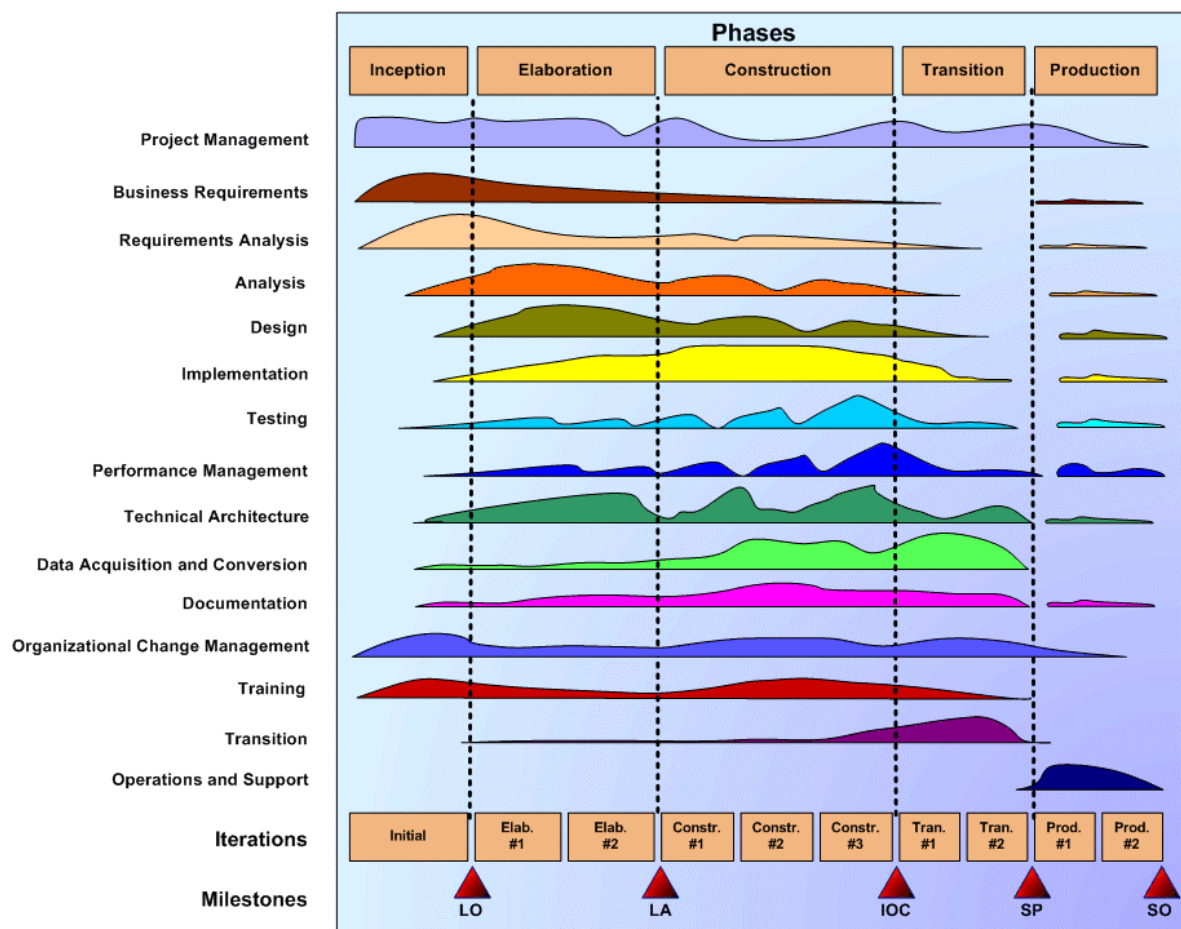


Figura 2.9²³: Fases de implementación de un proyecto OUM

²³ ORACLE. (2011). *Oracle Unified Method (OUM)*. (7). Recuperado el 20 de Febrero de 2012, de: <http://www.oracle.com/us/products/consulting/resource-library/oracle-unified-method-069204.pdf>.

Fases de control

OUM tiene las siguientes fases de control:

- Fase de Comienzo (Inception Phase)
- Fase de Elaboración (Elaboration Phase)
- Fase de Construcción (Construction Phase)
- Fase de Transición (Transition Phase)
- Fase de Producción (Production Phase)

Fase de Comienzo (Inception Phase)

El objetivo de esta fase es lograr un acuerdo acerca de los objetivos del proyecto entre las partes involucradas. Esta fase es crítica ya que se determinan tres aspectos básicos para el desarrollo de un proyecto: el alcance, los requerimientos de alto nivel y los riesgos.

Esta fase sirve para poner en marcha un proyecto, analizar su estrategia de negocio, y obtener la confirmación, documentación y prioridad de los requerimientos de alto nivel. También el equipo del proyecto se consolida y se desarrolla el plan inicial del proyecto.

Fase de Elaboración (Elaboration Phase)

El objetivo de esta fase es detallar los requerimientos, dividir la solución, crear los prototipos necesarios y el modelo básico de la arquitectura del sistema. Esto crea una base para el diseño e implementación del proyecto en la fase de construcción. La arquitectura evoluciona a partir de los requerimientos más importantes y los riesgos existentes, y se evalúa varios prototipos. Durante esta fase es verificado el entendimiento de los requerimientos de negocio del cliente, con lo cual se reducen riesgos en el desarrollo.

Fase de Construcción (Construction Phase)

El objetivo de esta fase es obtener la solución de software a partir de los modelos de requerimientos detallados. La primera versión que entra a producción se llama *beta*. Se completa el desarrollo de la aplicación, se verifica que los componentes se ensamblen entre sí y se prepara la aplicación para las pruebas de aceptación y la posterior implementación.

El desarrollo de la aplicación es completado según un número predefinido de iteraciones, actualizándose repetidamente los modelos (como el de Casos de uso o el de Diseño) conforme los requerimientos se vayan entendiendo mejor. Cuando todas las iteraciones se han terminado para cada partición, el sistema completo es probado. El sistema probado es el resultado final de esta fase.

Fase de Transición (Transition Phase)

El objetivo de esta fase es instalar el software en producción, realizar las pruebas de aceptación e iniciar la aplicación en vivo. La organización acepta el software y enseguida se prepara para la utilizarlo. Si la aplicación va a reemplazar a otra, se escoge una fecha con el menor impacto para el desempeño de la organización.

La fase de Transición puede tener varias iteraciones. Esta fase incluye la preparación para el lanzamiento de la aplicación y la realización de ajustes menores basados en la retroalimentación del usuario.

Fase de Producción (Production Phase)

El objetivo de esta fase es usar el software desarrollado, evaluar su desempeño y dar soporte a los usuarios. Se supervisa el sistema: su continuidad, performance, operación y mantenimiento de los sistemas de apoyo. Se responde las solicitudes de ayuda por parte de los usuarios; se reporta errores y solicitudes de prestaciones de los usuarios; y se maneja un

proceso de control de cambios para que los defectos y nuevas características se registren y prioricen para futuras versiones.

2.4.6 Procesos del proyecto para la continuidad

Las tareas de OUM se agrupan en procesos que contienen tareas relacionadas entre sí. Los miembros del equipo son asignados a las mismas de acuerdo a su especialización y background (conocimiento). El Método Unificado Oracle tiene los siguientes procesos:

- Requerimientos de negocio (Business Requirements)
- Requerimientos de análisis (Requirements Analysis)
- Análisis (Analysis)
- Diseño (Design)
- Implementación (Implementation)
- Pruebas (Tests)
- Gestión del performance (Performance management)
- Arquitectura técnica (Technical Architecture)
- Adquisición y conversión de datos (Data Acquisition and Conversion)
- Documentación (Documentation)
- Gestión organizacional del cambio (Organizational change management)
- Entrenamiento (Training)
- Transición (Transition)
- Operaciones y soporte (Operations and support)

Requerimientos de Negocio (Business Requirements)

En este proceso, los requerimientos de negocio o nuevas mejoras son identificados, procesados y priorizados por usuarios habilitados y analistas experimentados. Parte de los

requisitos de alto nivel y documentos de alcance. Si dichos requerimientos no han sido obtenidos, se puede comenzar a partir del alcance y los objetivos. El resultado de este proceso son los objetivos y metas de negocio, y la lista de requerimientos funcionales y no funcionales.

Requerimientos de Análisis (Requirements Analysis)

En este proceso se hace un análisis más minucioso de los requerimientos funcionales y suplementarios mediante un Modelo de Casos de Uso. Este ayuda a verificar que los modelos y procesos de sistemas asociados satisfagan los requerimientos de negocio. El Modelo de Casos de Uso documenta en detalle los requerimientos del sistema de manera que el cliente y los desarrolladores los puedan entender. Los resultados de este proceso son: el Modelo de Casos de Uso, un prototipo de la interfaz de usuario, y una descripción de alto nivel de la arquitectura de software.

Análisis (Analysis)

En este proceso, los requerimientos obtenidos son analizados y proyectados sobre el set de productos COTS²⁴, para determinar qué requerimientos se pueden cumplir directamente por el software que se está desarrollando y qué requisitos deberán ser cumplidos mediante el desarrollo software de aplicación personalizada o extensiones personalizadas.

Este proceso de Análisis no sólo busca obtener un mapa del producto, sino principalmente redefinir y estructurar los requerimientos a través de un modelo conceptual: el Modelo de Análisis, que es el conjunto de todos los componentes analizados. El resultado de este proceso es el Modelo de Análisis Revisado (Reviewed Analysis Model), el cual incluye un grupo de diagramas de clases, que surgen de los casos de uso.

²⁴ Siglas en inglés de Commercial Off-The-Self. Describe los productos de software o de hardware que están hechos a medida y están disponibles para la venta al público; estos productos se instalan en los sistemas existentes y no necesitan personalización.

Diseño (Design)

En el proceso de Diseño, el sistema es formado para cumplir los requisitos funcionales y suplementarios, basándose en la arquitectura creada y confirmada durante el proceso de Análisis. El diseño es el enfoque cuando está finalizando la fase de Elaboración y están empezando las iteraciones de la de Construcción. Los principales productos de este proceso se combinan en última instancia formando el Modelo de Diseño (Design Model) que será usado en el proceso de Implementación (Implementation). El producto principal de este proceso es el Modelo de Diseño Revisado (Reviewed Design Model), que incluye un modelo de objetos que describe la elaboración del diseño de casos de uso y diseño de clases, que detalla el análisis de clases definido en el Modelo de Análisis (Analysis Model).

Implementación (Implementation)

En el proceso de Implementación, la aplicación final se desarrolla principalmente mediante iteraciones, basándose en los resultados del proceso de Diseño. Se desarrolla el código fuente de los componentes, scripts y ejecutables. Los desarrolladores realizan pruebas de los componentes de software. La Implementación es el proceso fundamental en la fase de Construcción, la cual comenzó al final de la fase de Elaboración a fin de aplicar la arquitectura base. En la fase de Transición se manejan los defectos y errores detectados en las pruebas y liberación del sistema. El resultado de este proceso es la iteración final de construcción, lista para ser probada.

Pruebas (Tests)

El proceso de Pruebas se enfoca en el control de calidad y conformidad de todos los elementos del sistema. Está relacionado con las tareas del *Proceso de Gestión de Calidad* (Quality Management Process) del Método de Gestión de Proyectos Oracle (Oracle's

Project Management Method), que es parte del Área de gestión de enfoque de OUM (OUM's Manage focus area), y con la definición y refinamiento de los requerimientos obtenidos en el proceso de Requerimientos de Negocio (Business Requirements). El proceso de Pruebas está dirigido a las funciones del proyecto, y en el caso de proyectos que necesiten asociarse con interfaces de otros sistemas, también prueba estos. Las pruebas están a cargo de todas las personas involucradas en el proyecto, las cuales deben consolidarse como un equipo.

Gestión del Performance (Performance Management)

El proceso de Gestión del performance tiene como objetivo realizar y ejecutar una estrategia de manera proactiva, para la gestión del rendimiento en todo el ciclo de vida del software. Este proceso no significa realizar una prueba o ajuste aislado (aunque esto puede ser parte de la estrategia), sino influir en la Arquitectura técnica del sistema, con la cual está incondicionalmente relacionado.

Arquitectura Técnica (Technical Architecture)

El proceso de Arquitectura técnica busca diseñar una arquitectura del sistema que soporte la visión del negocio. Las tareas en este proceso tienen como propósito identificar y documentar los requerimientos relacionados con el establecimiento y mantenimiento de la aplicación y el entorno de infraestructura técnica. Los procesos y procedimientos necesarios para implementar, monitorear y mantener los diferentes ambientes son definidos y probados.

Adquisición y conversión de datos (Data Acquisition and Conversion)

Este proceso tiene por objetivo crear los componentes necesarios para extraer, transformar, transportar y cargar el patrimonio de datos de origen para poder soportar las

necesidades de información del nuevo sistema. Se definen explícitamente los datos que se convertirán, así como sus fuentes. Estos datos pueden ser necesarios para pruebas del sistema, capacitación, pruebas de aceptación y producción. En algunos casos, es beneficioso realizar la conversión de datos en estado temprano de los mismos, para proveer las revisiones más realistas de los componentes durante las iteraciones de desarrollo.

Documentación (Documentation)

Para que el software tenga su transición a producción, lograr la aceptación del cliente y mantener activos los procesos de negocio, es necesario que exista una documentación de calidad. Los requisitos y estrategia para conseguirlo varían en función del alcance, la tecnología y los requerimientos. En el caso de proyectos que manejen la tecnología de Oracle, la documentación que provee esta compañía es la base. El proceso de Documentación tiene el objetivo de aportar a los manuales suministrados por Oracle, con la documentación específica del nuevo software desarrollado y sus procedimientos de negocio.

Gestión organizacional del cambio (Organizational change management)

Este proceso se inicia a nivel estratégico con los ejecutivos. Se identifican los desafíos personales y organizacionales de la adopción de la tecnología, para establecer un enfoque sistemático, rentable y con énfasis en el tiempo. Se busca reducir el riesgo, el cual se adapta a cada una de las necesidades de la organización. Además de aumentar la tasa de aprobación de usuarios, su planificación y la gestión de cambio, permite a la organización mantener el nivel de productividad a través de transiciones tecnológicas que suelen ser dificultosas. En síntesis, la organización podrá cumplir los cronogramas y los objetivos de negocio, y maximizar las ganancias.

Entrenamiento (Training)

El objetivo del proceso de Training es asegurarse que tanto el equipo de desarrollo como los usuarios están capacitados, el primero para realizar las tareas para iniciar el proyecto, y el segundo, para ejecutar las tareas de funcionamiento del nuevo sistema.

Transición (Transition)

El objetivo del proceso de Transición es instalar el sistema. Se requieren procedimientos para dicha instalación, los cuales son llevados a cabo en producción. El proceso se inicia tempranamente definiendo los requerimientos necesarios a la fecha corte del nuevo sistema. Luego se aplica una estrategia que incluye el desarrollo de un plan de instalación, la preparación del ambiente de producción, la realización de la fecha de corte, y la clausura de todos los sistemas heredados.

Operaciones y Soporte (Operations and Support)

El proyecto no termina abruptamente cuando se ha entregado la solución para producción, ya que en los meses posteriores a esto puede determinarse el éxito o fracaso del proyecto. Aún los auditores internos no han realizado la evaluación final del sistema, y los usuarios muy probablemente tendrán unos pocos problemas por descubrir. Hay ciertos requerimientos que no se han implementado por tener una prioridad menor. También hay requerimientos agrupados como posibles necesidades y otros agrupados como necesidades comprobadas; ambos podrán ser considerados prioridades en el plan de mejora, a partir del cual se definen las actualizaciones.

El proceso de Operaciones y soporte se encarga de monitorear y responder a los inconvenientes del sistema, actualizar el mismo corrigiendo errores y problemas en el

rendimiento, evaluar el sistema en producción, y realizar un plan de mejoras en cuanto funcionalidad, rendimiento y seguridad.

2.4.7 Actividades que determinan el compromiso en el ciclo de vida

Una fase contiene actividades, y una actividad, tareas. OUM maneja este esquema para determinar de mejor manera el compromiso en el ciclo de vida. Las actividades permiten al gerente o jefe de proyecto racionalizar la creación y gestión de la Estructura de Desglose de Trabajo (Work Breakdown Structure, WBS) para el compromiso. Los jefes de proyecto están en potestad de trabajar a nivel de actividades en lugar que a nivel de tareas.

2.4.8 Gestión de proyectos OUM

Área de Enfoque de Gestión (Manage Focus Area)

El Área de Enfoque de Gestión (Manage Focus Area) provee un marco en que cualquier tipo de proyecto puede ser planeado, estimado, controlado y completado consistentemente, lo cual es requerido en el entorno empresarial actual: el Método Oracle de Gestión de Proyectos (Oracle's Project Management Method – PJM).

La gestión de proyectos permite al jefe del proyecto negociar la entrega de la aplicación de acuerdo a un nivel de calidad, mientras planifica y controla el alcance, costo y cronograma.

El área de enfoque de gestión tiene 3 fases:

- Fase de puesta en marcha del proyecto (Project Start up Phase)
- Fase de ejecución y control del proyecto (Project Execution and Control Phase)
- Fase de cierre del proyecto (Project Closure Phase)

Manage Focus Area Integration with the Implement Focus Area

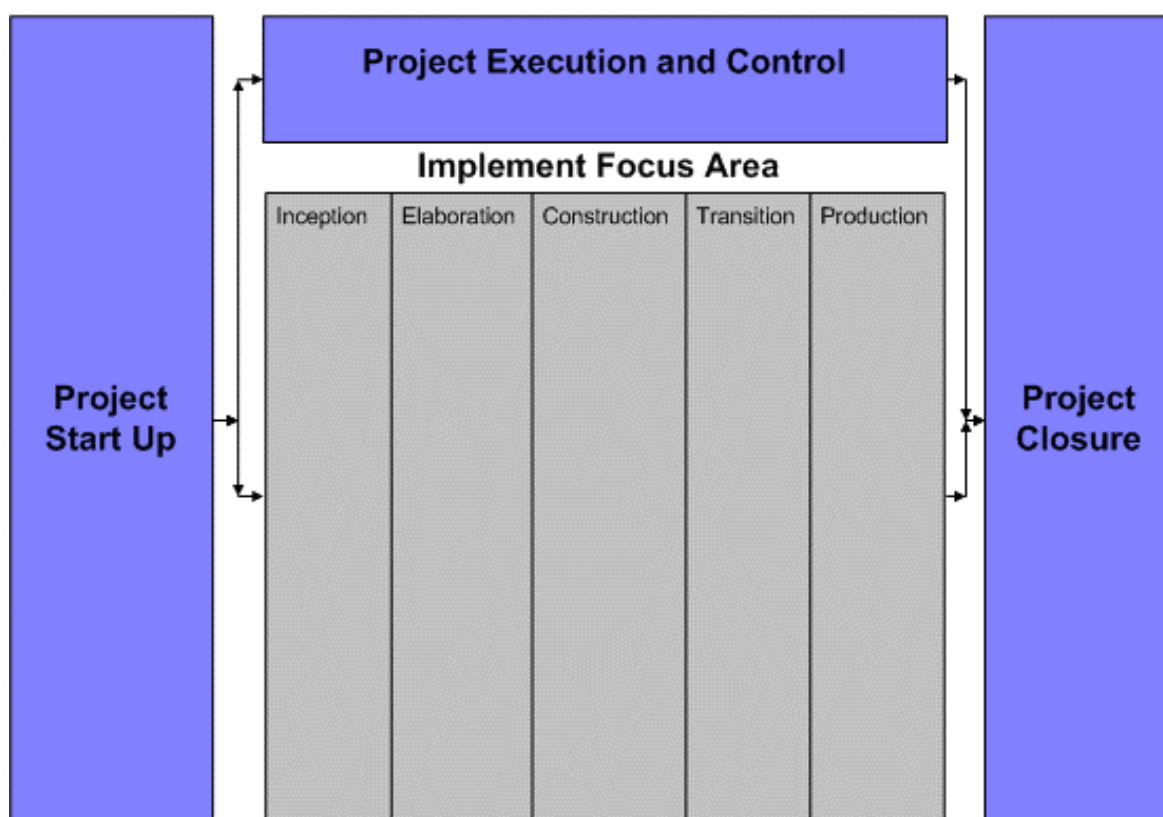


Figura 2.10²⁵: Integración del Área de enfoque de gestión con el Área de enfoque de implementación

Fase de Puesta en marcha del Proyecto (Project Start Up Phase)

La Fase de Puesta en marcha del Proyecto (Project Start Up Phase) precede a la fase de Comienzo del proyecto (Inception phase). La puesta en marcha del proyecto significa que todas actividades de la planificación se ejecutarán y las reglas, procedimientos y estrategias serán definidas para cada uno de los procesos, que rigen el proceder de los compromisos.

²⁵ ORACLE. (2011). *Oracle Unified Method (OUM)*. (13). Recuperado el 20 de Febrero de 2012, de: <http://www.oracle.com/us/products/consulting/resource-library/oracle-unified-method-069204.pdf>.

Fase de Ejecución y Control del Proyecto (Project Execution and Control Phase)

La Fase de Ejecución y Control del Proyecto (Project Execution and Control Phase) se ejecuta simultáneamente con las fases pertenecientes al Área de enfoque de la implementación (Implementation focus area). Su objetivo es proveer una visibilidad adecuada del progreso del proyecto, para que la gestión pueda tomar acciones efectivas cuando el performance del proyecto se desvía significativamente de lo que se tenía planificado. Se monitorea y revisa los logros y resultados de los proyectos versus la Estructura de Desglose de Trabajo (WBS) documentada, estimaciones, cronograma, plan de recursos y presupuesto de gastos, y se ajusta estos planes a los logros y resultados reales.

Fase de Cierre del Proyecto (Project Closure Phase)

La Fase de Cierre del Proyecto (Project Closure Phase) se ejecuta después de la fase de Producción (Production Phase). Desde el punto de vista administrativo y contractual, el proyecto está en estado de “cerrado”. Para ello los productos se han entregado al cliente completamente y han cumplido sus expectativas, consiguiendo su aceptación final. Además se entregan al cliente los documentos necesarios para reuso, recopilación y conservación del sistema.

El Área de Enfoque de Gestión (Management Focus Area) contiene trece procesos:

- Oferta de Transición (Bid Transition)
- Gestión del Alcance (Scope Management)
- Gestión Financiera (Financial Management)
- Gestión de Trabajo (Work Management)
- Gestión de Riesgos (Risk Management)

- Gestión de Problemas e Inconvenientes (Issue and Problem Management)
- Gestión de Personal (Staff Management)
- Gestión de la Comunicación (Communication Management)
- Gestión de la Infraestructura (Infrastructure Management)
- Gestión de Adquisiciones (Procurement Management)
- Gestión del Cambio Organizacional (Organization Change Management)

Los procesos mencionados forman un amplio conjunto de tareas necesarias para gestionar el desarrollo e implementación de proyectos basados en Oracle.

Cada proyecto incluye la mayoría o todos estos procesos; esto no es responsabilidad de Oracle.

2.4.9 Actividades para el desarrollo y mantenimiento de la arquitectura empresarial

Área de Enfoque de Visión (Envision Focus area)

El Área de enfoque de Visión (Envision focus area) provee un marco para el desarrollo y mantenimiento de la estrategia empresarial de TI, la arquitectura y la gobernanza. La guía de Visión ayuda al equipo del proyecto en la transición de la planificación a nivel empresarial y actividades estratégicas hacia la identificación e iniciación del proyecto en sí.

El Área de enfoque de Visión (Envision focus area) tiene dos fases:

- Fase de Iniciamiento (Initiate Phase)
- Fase de Mantenimiento y Evolución (Maintain and Evolve Phase)

Fase de Iniciamiento (Initiate Phase)

La Fase de Iniciamiento (Initiate Phase) es usada para realizar una serie de tareas fundamentales; estas tienen un rango de objetivos y aplicabilidad dependiendo del proyecto. En su aplicación más simple, la *Fase de Iniciamiento* puede establecer la visión de uno o más proyectos destinados a lograr ciertos objetivos de negocio; en su aplicación más compleja, puede definir un amplio conjunto de procesos de TI a nivel empresarial que tendrán su continuación en la Fase de Mantenimiento y Evolución (Maintain and Evolve Phase).

Fase de Mantenimiento y Evolución (Maintain and Evolve Phase)

La Fase de Mantenimiento y Evolución (Maintain and Evolve Phase) conforma la base para la gobernanza y gestión de procesos y estrategias de negocio a nivel empresarial, centrándose en la tecnología de la información que tiene relación con la arquitectura y prácticas de negocio.

El Área de Enfoque de Visión (Envision focus area) contiene seis procesos:

- Mapa de Ruta de la Visión (Envision Roadmap)
- Análisis Empresarial de Negocio (Enterprise Business Analysis)
- Gestión de Cambio Organizacional (Organizational Change Management)
- Arquitectura Empresarial (Enterprise Architecture)
- Gestión de la Cartera de TI (IT Portfolio Management)
- Gobernanza (Governance)

La Visión busca adaptarse a las necesidades específicas de las empresas. No es probable que todos los procesos del Área de Enfoque de Visión (Envision focus area) se

ejecuten para un proyecto de una empresa, ni que la Visión busque contener una lista exhaustiva de procesos a nivel empresarial.

2.4.10 Componentes del OUM

OUM suministra los siguientes componentes para su aplicación:

- Materiales para Visión General (Overview Materials)
- Directrices (Guidelines)
- Plantillas (Templates)
- Estructura de Desglose de Trabajo A Medida (Tailored Work Breakdown Structure)

Materiales para Visión General (Overview Materials)

Los Materiales para Visión General (Overview Materials) proporcionan un enfoque general acerca del alcance de los proyectos de TI, incluyendo los estándares de la industria, en los cuales se basa OUM.

Directrices (Guidelines)

Las Directrices (Guidelines) son pautas para cada aspecto de las fases, procesos, tareas, roles y productos de trabajo en soluciones de negocio basadas en Oracle.

Plantillas (Templates)

Las Plantillas (Templates) facilitan la creación de documentos entregables de alta calidad.

Estructura de Desglose de Trabajo A Medida (Tailored Work Breakdown Structure)

La Estructura de Desglose de Trabajo A Medida (Tailored Work Breakdown Structure) permite iniciar fácilmente la gestión de los proyectos.

2.4.11 Requerimientos de software y hardware²⁶

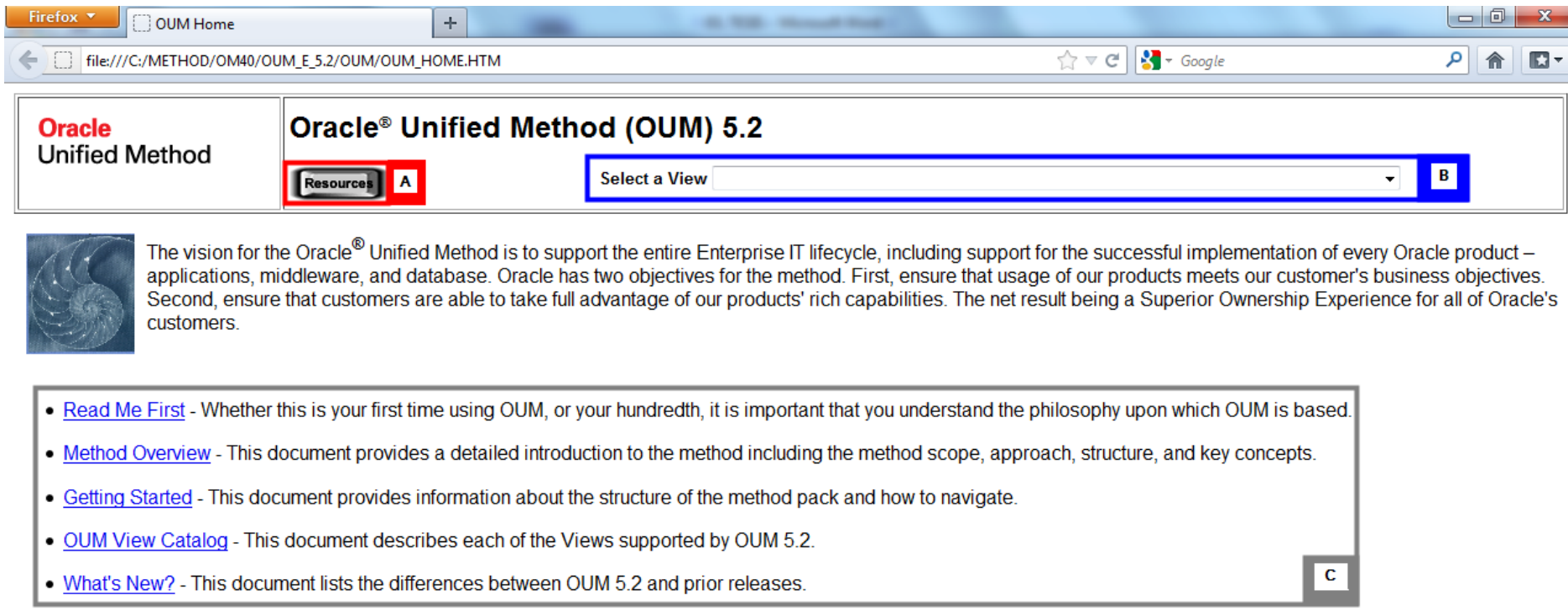
Es necesario el siguiente software y hardware para realizar la instalación de la herramienta OUM 5.2:

- Velocidad de procesador: 1 gigahercio (GHz).
- Espacio en disco: 250 megabytes (MB).
- Memoria RAM: 520 megabytes (MB).
- Microsoft Windows 7/XP/2000/NT 4.0.
- Microsoft Office 2000 o posterior.
- Microsoft Visio 5.0 o posterior (opcional).
- Adobe Acrobat Reader 6.0 o posterior.
- Mozilla Firefox 2.0.0.16 o posterior / Microsoft Internet Explorer 6.0 o posterior / Netscape 7.0 o posterior.

ESPACIO EN BLANCO
INTENCIONAL

²⁶ ORACLE. (2011). Oracle Unified Method (OUM). (16). Recuperado el 20 de Febrero de 2012, de: <http://www.oracle.com/us/products/consulting/resource-library/oracle-unified-method-069204.pdf>.


2.4.12 Entorno de la herramienta de OUM



Oracle
Unified Method

Oracle® Unified Method (OUM) 5.2

Resources **A** **B**

 The vision for the Oracle® Unified Method is to support the entire Enterprise IT lifecycle, including support for the successful implementation of every Oracle product – applications, middleware, and database. Oracle has two objectives for the method. First, ensure that usage of our products meets our customer's business objectives. Second, ensure that customers are able to take full advantage of our products' rich capabilities. The net result being a Superior Ownership Experience for all of Oracle's customers.

- [Read Me First](#) - Whether this is your first time using OUM, or your hundredth, it is important that you understand the philosophy upon which OUM is based.
- [Method Overview](#) - This document provides a detailed introduction to the method including the method scope, approach, structure, and key concepts.
- [Getting Started](#) - This document provides information about the structure of the method pack and how to navigate.
- [OUM View Catalog](#) - This document describes each of the Views supported by OUM 5.2.
- [What's New?](#) - This document lists the differences between OUM 5.2 and prior releases.

ORACLE
Copyright © 2006, 2009, Oracle. All rights reserved.

Figura 2.11: Página inicial del entorno de la herramienta de OUM

Recursos

En la Figura 2.11, sección A, se muestra el botón “Resources”, el cual da acceso a una página que agrupa algunos eslavones claves para la aplicación de OUM. Esto facilita el acceso a estos recursos: información acerca de OUM, primeros pasos, descripción del método OUM, preguntas más frecuentes, catálogo de vistas, referencias y publicaciones, manual de uso de las plantillas de OUM, y, comentarios y preguntas. Esta sección también contiene el diagrama de contexto de la aplicación de OUM, que muestra de manera macro la gestión de proyectos con esta metodología.

Selección de vistas

En la Figura 2.11, sección B, se muestra un combobox que permite elegir entre las vistas de OUM.

Accesos directos

En la Figura 2.11, sección C, se muestran accesos directos hacia los siguientes recursos: lectura inicial antes del primer uso de OUM, descripción del método, primeros pasos, catálogo de vistas, y, las nuevas características de la versión 5.2.

2.5 Oracle JDeveloper 11.1.1.1.5.0

2.5.1 Historia

JDeveloper es un entorno de desarrollo parte de la empresa Oracle, su primera versión aparece en Marzo de 1998 con la versión 1.0; posteriormente en abril de 1999 JDeveloper

incorpora una capa de acceso a base de datos así como la primera versión de BC4J ²⁷ que incluye el uso de metadatos como única vía de configuración.

En mayo de 2002, se libera la versión 9.0.2, que contiene herramientas de modelamiento UML; en abril de 2004, aparece la versión 9.0.5, en la cual se añade el framework ADF; finalmente surgen nuevas versiones de JDeveloper hasta la versión 11.1.2.0.0 ²⁸

2.5.2 Lenguajes oportados

JDeveloper soporta los siguientes lenguajes de programación:

- Java
- HTML
- XML
- PL/SQL
- Javascript
- PHP

2.5.3 Requerimientos de instalación

Sistemas operativos soportados

JDeveloper certifica su funcionamiento con los siguientes sistemas operativos²⁹:

- Windows XP (Service Pack 2)
- Windows Server 2003 (Release 2)

²⁷ Componente de negocio de java que permite el encapsulamiento y mapeo objeto – relación con la base de datos, abstrayendo su información general y de conexión.

²⁸ ORACLE. (2011). A History of Oracle JDeveloper and Oracle ADF Releases. Recuperado el 18 de Febrero de 2012, de: <http://www.oracle.com/technetwork/developer-tools/jdev/jdev-history-099970.html>.

²⁹ MILLS, D., KOLETZKE, P., ROY-FADERMAN, A. (2010). Oracle JDeveloper 11g Handbook. USA: Oracle.

- Windows Server 2000 (Service Pack 4)
- Red Hat Enterprise Linux 3.0
- Red Hat Enterprise Linux 4.0
- Mac OS X 10.4.x

Requerimientos del sistema

JDeveloper funciona con estos requisitos mínimos:

- Procesador dual core
- Memoria RAM de 2GB
- Espacio en disco duro de 1.4GB

2.5.4 Roles en JDeveloper

Este IDE de desarrollo presenta 5 roles para delimitar el trabajo:

- **Rol por defecto.**- Permite acceder a todos los componentes del IDE.
- **Personalización de desarrollador.**- Permite configurar el producto para personalizar los metadatos.
- **Edición de Base de datos.**- Incluye exclusivamente herramientas de manejo de base de datos.
- **Edición Java EE³⁰.**- Incluye exclusivamente herramientas para el desarrollo en Java EE.

³⁰ Plataforma de desarrollo y ejecución de aplicaciones java con arquitectura *n* capas distribuidas, incluye JDBC, RMI, JMS, servicios web, XML.

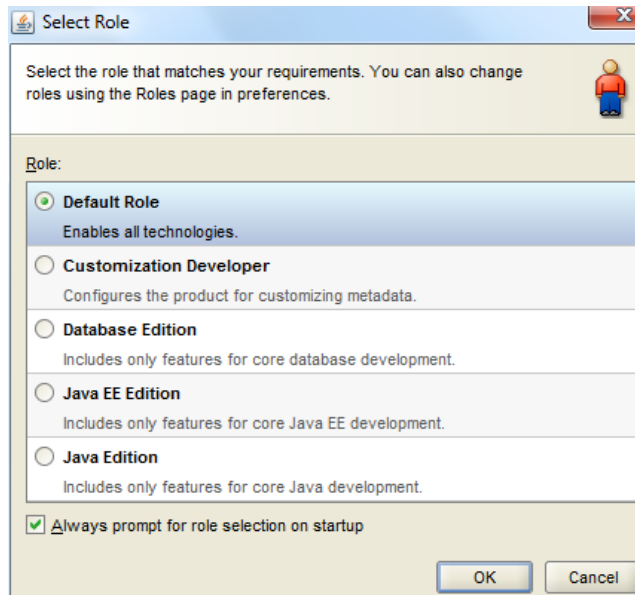


Figura 2.12: Selección de roles en JDeveloper

ESPACIO EN BLANCO
INTENCIONAL

2.5.5 Entorno de desarrollo

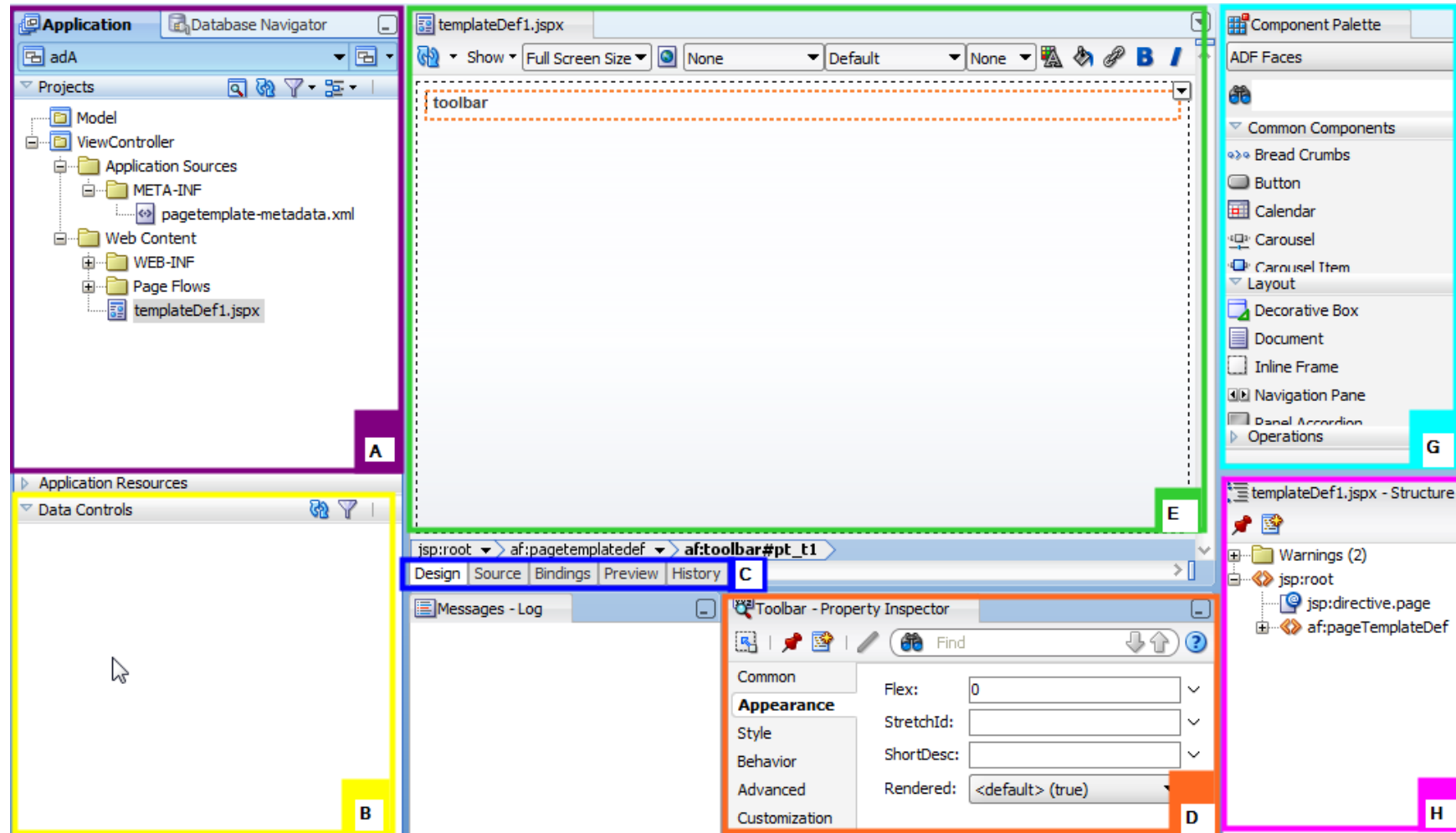


Figura 2.13 Entorno de desarrollo JDeveloper 11g

Navegador de aplicación

En la Figura 2.13, sección A, se muestra el navegador de aplicación que permite administrar módulos, clases java, paginas jsf, jsp, task flow.

Data Controls

En la Figura 2.13, sección B, se muestra el control de datos que permite utilizar las vistas relacionadas en un Application module como pantallas.

Pestañas de control

En la Figura 2.13, sección C, se muestran las siguientes pestañas:

- **Design.-** Muestra el diseño físico de la página diseñada.
- **Source.-** Muestra el código generado por diseño físico de la página desarrollada.
- **Bindings.-** Muestra las variables bind, los métodos y operaciones asociadas con la página desarrollada.
- **Preview.-** Muestra una imagen previa de la página desarrollada cuando se despliegue en un browser.
- **History.-** Despliega dos ventanas que tienen el código de la pantalla desarrollada: la primera, a la fecha de corte, y la segunda, a la fecha actual.

Properties

En la Figura 2.13, sección D, se muestra las propiedades de los objetos que se seleccionen.

Design Screen

En la Figura 2.13, sección E, se muestra donde crear las pantallas de la aplicación; esta sección funciona en relación a la sección C.

Component palette

En la Figura 2.13, sección G, se muestra la paleta de componentes.

Design Screen

En la Figura 2.13, sección H, se muestra la ventana de estructura jerárquica dentro de la página desarrollada.

2.6 Oracle ADF 11.1.2.0.0

2.6.1 Historia

Oracle ADF aparece por primera vez en abril de 2004, para la versión 9.0.5 de JDeveloper; actualmente está definida la versión más estable (11.1.2.0.0) liberada en junio de 2011.

2.6.2 Introducción

Oracle ADF es un framework de desarrollo construido en base a los estándares Java EE y definido en base al patrón de diseño MVC³¹. Oracle ADF es parte del IDE de desarrollo JDeveloper desde abril de 2004, para la versión 9.0.5.

³¹ Siglas de Modelo Vista Controlador. Es un patrón de diseño de software que diferencia tres secciones: la interfaz de usuario, los datos de la aplicación y la lógica de negocio.

2.6.3 Arquitectura

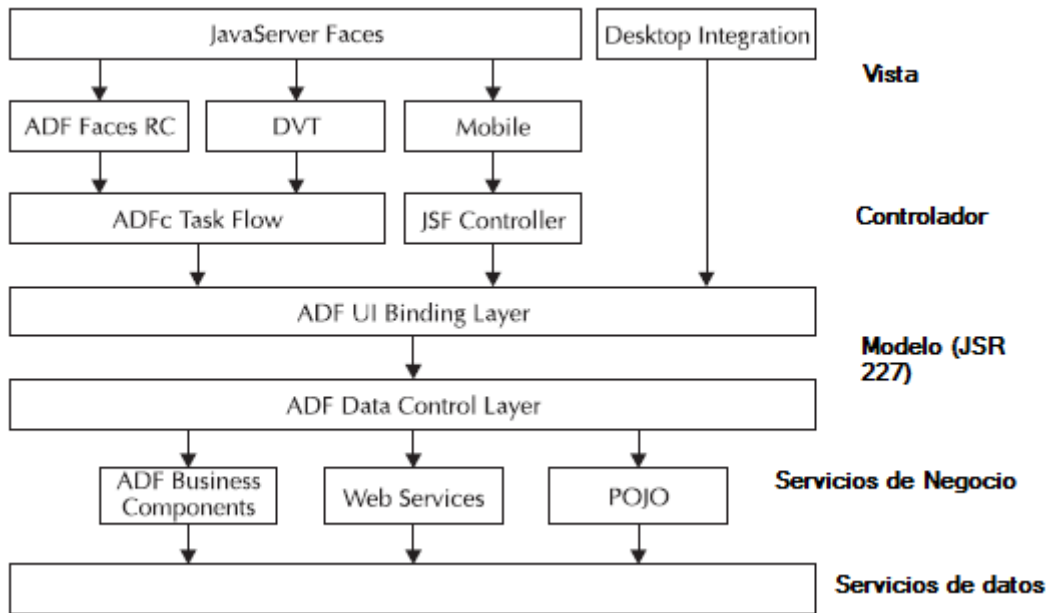


Figura 2.14³²: Arquitectura ADF Fusion

La Arquitectura ADF tiene los siguientes componentes, como muestra la Figura 2.14:

- **Servicios de negocio.-** Representa el back-end del modelo de datos que interactúa con la base de datos y administra la lógica de la aplicación.
- **Modelo.-** Facilita que la capa de controlador y la capa de vista utilicen diversas implementaciones de la capa de servicios de negocio.
- **Controlador.-** Define procesos de control de flujo de la aplicación.
- **Vista.-** Define la interfaz de la aplicación.

³² NIMPHIUS, F., MUNSINGER, L. (2010). Oracle Fusion Developer Guide. (4). USA: Oracle.

2.6.4 Componentes de negocio de ADF

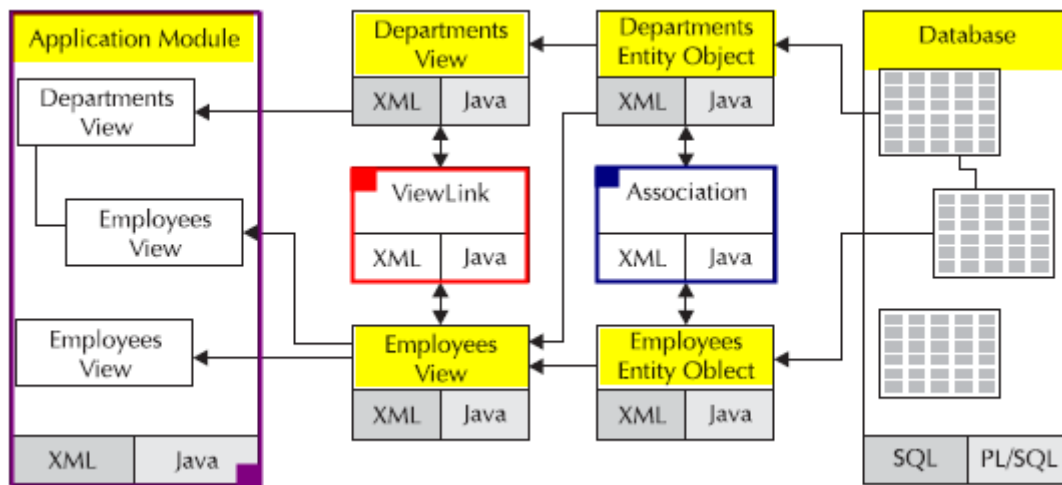


Figura 2.15³³: Componentes de negocio de ADF

Como muestra la Figura 2.15, ADF tiene los siguientes componentes:

- **Modulo de Aplicación (Application Module).**- Representan el modelo de datos representado por agrupaciones de vistas y view links.
- **Objeto Vista (View Object).**- Recopilación de datos de una base de datos asociada a un objeto entidad por medio de un script sql.
- **Objeto Entidad (Entity Object).**- Objeto java que representa una fila de una tabla, en una base de datos.
- **Asociación (Association).**- Define la relación entre dos objetos entidad; una asociación automáticamente toma los constraints³⁴ de la base de datos.
- **Enlace de vista (View link).**- Representa la relación entre dos objetos vista; de esta relación se genera la relación maestro - detalle³⁵.

³³ NIMPHIUS, F., MUNSINGER, L. (2010). *Oracle Fusion Developer Guide*. (6). USA: Oracle.

³⁴ Restricción en una base de datos que puede ser: not null, unique, check, primary key, foreign key.

³⁵ Es una conexión entre dos bloques de datos asociados a tablas, y es determinada por la relación entre la clave primaria y la clave foránea.

2.7 Oracle Weblogic Server 11g

2.7.1 Historia

La primera versión de Weblogic Server aparece en noviembre de 1997 y recibe el nombre de *Tengah*; posteriormente, en noviembre de 2006, surge la versión más estable: la 9.0. Actualmente está definida la versión 12c, liberada en diciembre de 2011.

2.7.2 Sistemas de gestión de base de datos soportados por Oracle Weblogic Server

Oracle Weblogic Server soporta los siguientes sistemas de gestión de base de datos:

- Oracle
- DB2
- Microsoft SQL Server
- MySQL Enterprise

2.7.3 Conectores nativos³⁶

Oracle Weblogic Server incluye los siguientes conectores nativos:

- CORBA connectivity
- COM+ Connectivity
- IBM WebSphere MQ connectivity
- Java EE Connector Architecture
- Native enterprise-grade JMS messaging
- WebLogic/Tuxedo Connector

³⁶ WIKIPEDIA. (2012). Oracle WebLogic Server. Recuperado el 19 de Febrero de 2012, de: http://en.wikipedia.org/wiki/Oracle_WebLogic_Server

2.7.4 Estándares soportados

Oracle Weblogic Server soporta los siguientes estándares:

- **BPEL.-** Lenguaje de Ejecución de Procesos de Negocio con Servicios Web, lenguaje basado en XML, cuyo propósito es el control en la invocación de servicios web.
- **JAAS.-** Servicio java de autenticación y autorización es un framework que añade una capa de verificación a la arquitectura java, para el control de acceso y ejecución de código.
- **Java EE.-** Plataforma de desarrollo java.
- **JPA.-** Api de persistencia java, su objetivo es mantener los beneficios de la programación orientada a objetos al interactuar con una base de datos
- **JMX.-** Gestor de extensiones java
- **SNMP.-** Protocolo Simple de Administración de Red

2.7.5 Organización de dominios

Dominio

Un dominio en Oracle Weblogic Server es un grupo de recursos que tiene incluido una instancia del servidor de aplicaciones (servidor de administración), encargada de la administración de dicho dominio.

Un dominio consta de:

- **Servidor de administración.-** Se encarga de la gestión del dominio, así como de la actualización de los documentos de configuración del dominio y su distribución a los servidores gestionados.

- **Servidor gestionado.-** Es el conjunto de componentes de aplicaciones, los servicios web y sus recursos asociados. Cuando se inicializa un servidor gestionado este se carga con el documento de configuración definido por el servidor de administración.
- **Recursos y servicios.-** Engloban proveedores de seguridad, adaptadores de recursos, servicios de diagnóstico y monitoreo, etc.

Opciones de instalación de dominios

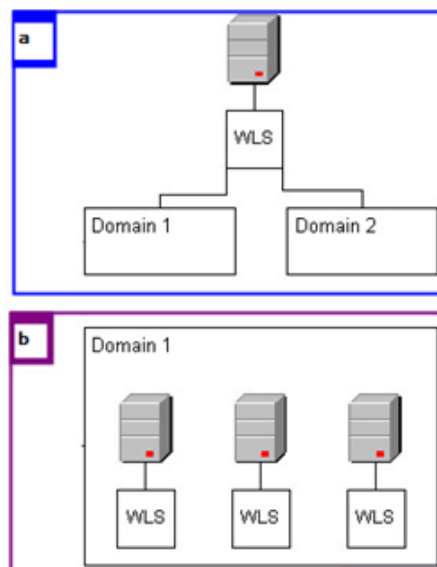


Figura 2.16³⁷: Instalaciones y dominios en Oracle Weblogic Server.

Como muestra la Figura 2.16, existen dos opciones a la hora de definir las instalaciones por dominio de Oracle Weblogic Server:

- Múltiples instalaciones para un solo dominio.
- Múltiples dominios para una sola instalación.

³⁷ ORACLE. (2012). WebLogic Server Domains. Recuperado el 19 de Febrero de 2012, de: http://docs.oracle.com/cd/E24329_01/web.1211/e24446/domains.htm.

2.7.6 Entorno de Oracle Weblogic Server

ORACLE WebLogic Server® Administration Console

Directorio Raíz | Desconexión | Preferencias | Grabar | Ayuda |

Bienvenido, weblogic | Conectado a: DefaultDomain

Directorio Raíz > Resumen de Entorno > DefaultDomain > Resumen de Entorno > Resumen de Despliegues > Resumen de Entorno

Resumen de Entorno

WebLogic Server puede alojar las aplicaciones en diferentes instancias del servidor, cada una de las cuales se puede ejecutar en computadoras diferentes y especificar su propia dirección de red. También puede agrupar los servidores en clusters para asegurarse de que las aplicaciones estén siempre disponibles si una de las instancias del servidor falla.

Utilice esta sección de la Consola de Administración para crear, configurar y controlar servidores y clusters.

Sección	Descripción
Servidores	Un servidor es una instancia de WebLogic Server que se ejecuta en su propia Java Virtual Machine (JVM) y que tiene su propia configuración.
Clusters	Un cluster es un despliegue en el que varias instancias de WebLogic Server (servidores) se ejecutan simultáneamente y trabajan juntas para aumentar la escalabilidad y la fiabilidad. Un cluster se presenta a los clientes como una sola instancia de WebLogic Server. Los servidores que forman un cluster se pueden ejecutar en la misma máquina o pueden estar ubicados en máquinas diferentes.
Hosts Virtuales	Un host virtual es un juego de nombres de host a los que responden las instancias de WebLogic Server (servidores) o los clusters. Al utilizar el alojamiento virtual, puede utilizar DNS para especificar uno o más nombres de host que se asignan a las direcciones IP de un servidor o cluster. También puede especificar las aplicaciones web que sirve cada host virtual.
Destinos Migrables	Un destino migrable es un destino que está activo como máximo en un servidor de un cluster a la vez.
Máquinas	Una máquina es la representación lógica de la computadora que aloja una o más instancias de WebLogic Server (servidores). WebLogic Server utiliza los nombres de las máquinas configuradas para determinar el servidor óptimo de un cluster al que se pueden delegar determinadas tareas, como la replicación de sesiones HTTP. El servidor de administración utiliza la definición de la máquina junto con la aplicación del gestor de nodos para iniciar servidores remotos.
Gestores de Trabajos	Un gestor de trabajo define un juego de clases de solicitudes y limitaciones de thread que gestionan el trabajo realizado por los servidores de WebLogic Server. Las aplicaciones J2EE, los módulos de aplicaciones web, los EJB y las aplicaciones RMI pueden especificar un gestor de trabajos determinado con el que gestionar las solicitudes de trabajos.
Clases de Inicio y Cierre	Las clases de inicio y cierre son programas Java que se crean para proporcionar servicios personalizados en todo el sistema para las aplicaciones. Las clases se agregan a la classpath de WebLogic Server y, a continuación, se configuran para cargarse y ejecutarse al iniciar o cerrar un servidor.

Centro de Cambios

Ver Cambios por Página

La edición de la configuración está activada. Los futuros cambios se activarán automáticamente a medida que modifique, agregue o suprima elementos de este dominio.

Estructura de Dominio

- DefaultDomain
 - Entorno
 - Despliegues
 - Servicios
 - Dominios de Seguridad
 - Interoperabilidad
 - Diagnósticos

¿Cómo...

- Crear Servidores Gestionados
- Iniciar y Parar Servidores
- Crear un Cluster
- Configurar Conexiones de Red por Defecto
- Configurar Clases de Inicio

Estado del Sistema

Estado de Servidores en Ejecución

- Failed (0)
- Critical (0)
- Overloaded (0)
- Advertencia (0)

Figura 2.17: Entorno de Oracle Weblogic Server

En la Figura 2.17 se muestran las siguientes secciones del entorno de Oracle Weblogic Server.

- A) **Estructura de dominio.**- Permite acceder al menú principal del servidor de administración
- B) **Sección de ayuda.**- Permite desplegar el menú de ayuda de Oracle Weblogic Server.
- C) **Estado del sistema.**- Permite medir 5 estados de los servidores de ejecución: fracaso, crítico, sobrecarga, advertencia, ok.
- D) **Sección de identificación de dominio.**- Muestra el dominio al cual hace referencia la consola de administración.
- E) **Resumen de entorno.**- Esta sección carga todas las pantallas señaladas por la estructura.

2.8 Base de datos Oracle 11g (Oracle Database 11g)

2.8.1 Historia

En el año 1989, Oracle lanza al mercado la versión 6 de su base de datos; es la primera vez que se incluye lenguaje PL/SQL. Posteriormente, en el año 1995, con la versión 8, la base soporta objetos (ORBDMS (Object Relational Database Management System)). A continuación, en el año 2001, con la versión 9i, se presentan las opciones de alta disponibilidad y clustering³⁸. En el año 2003 se lanza la versión 10g; es la primera base de datos en estar definida para el sistema operativo Linux de 64 bits. Finalmente, en el año 2007, se lanza la versión 11g, con capacidades de administración de transacciones y minería de datos.

³⁸ Construcción de computadoras mediante la utilización de hardwares comunes, que se compartan como se fuesen una sola computadora.

2.8.2 Requisitos de Instalación

Requisitos de hardware

- Memoria Ram: 1GB
- Memoria virtual: 2GB
- Espacio en disco: 4.55GB (instalación básica), 4.92 GB (instalación avanzada)
- Adaptación de video: 256 colores
- Procesador: 550Mhz
- Arquitectura del procesador: Intel(x86), AMD64, Intel EM64T

Requisitos de software

- Windows 2000 Service Pack1 / Windows Server 2003 / Windows XP Professional / Windows 7
- Oracle Linux 6

2.8.3 Estructura lógica

La estructura lógica comprende 2 secciones:

- **Área lógica de almacenamiento.-** Conjunto de tablespaces³⁹.
- **Colección de objetos.-** La agrupación os objetos de distintos esquemas⁴⁰

³⁹ Un tablespace es una unidad de almacenamiento lógica. Un tablespace contiene uno o más datafiles. El tamaño de los datafiles puede ser fijo o con crecimiento, definiendo el límite superior.

⁴⁰ Conjunto de objetos de la base de datos (tablas, vistas, secuencias, procedimientos almacenados, sinónimos, índices, clusters y enlaces con otras bases de datos). Si la base de datos es relacional, el esquema define los campos, las tablas y las relaciones entre cada uno de ellos.

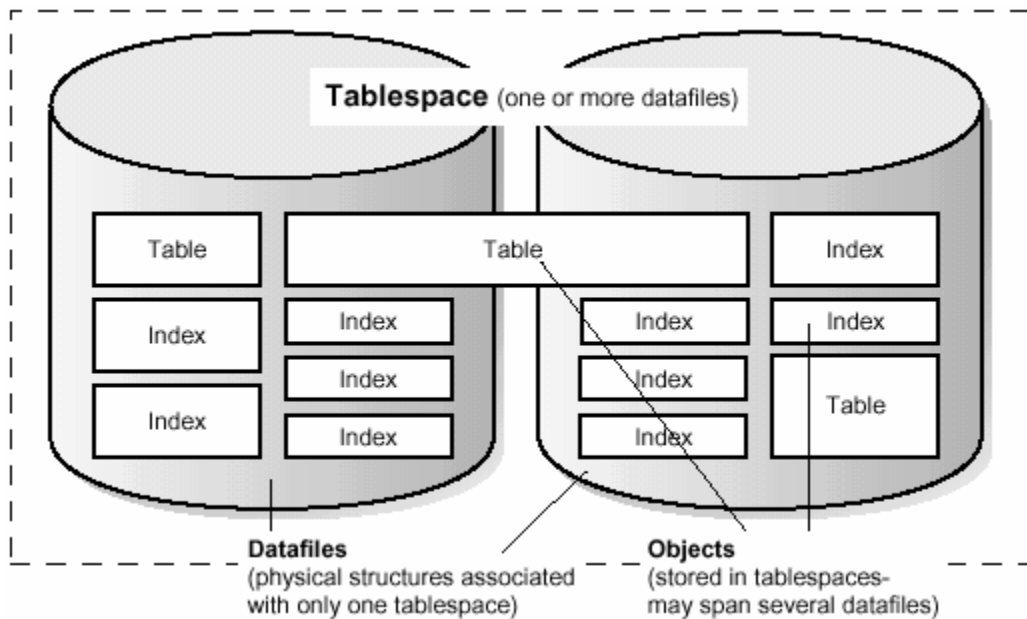


Figura 2.18⁴¹: Estructura lógica de la base de datos Oracle

2.8.4 Estructura física

La estructura física la conforman los siguientes archivos:

- **Ficheros de datos.-** Contienen los datos de la base de datos, son de tamaño fijo, y se establecen cuando se crea la base o cuando se crean los tablespaces.
- **Ficheros de redo log.-** Registra los cambios en la base y permítela recuperación de datos de la base de datos ante un imprevisto.
- **Ficheros de control.-** Almacena información de la estructura física de la base de datos

⁴¹ FERNANDEZ, J. Estructura de una base de datos Oracle. (1). Recuperado el 10 de Septiembre de 2011, de: <http://dis.um.es/~jfernand/0405/dbd/DBD04T06-oracle.pdf>

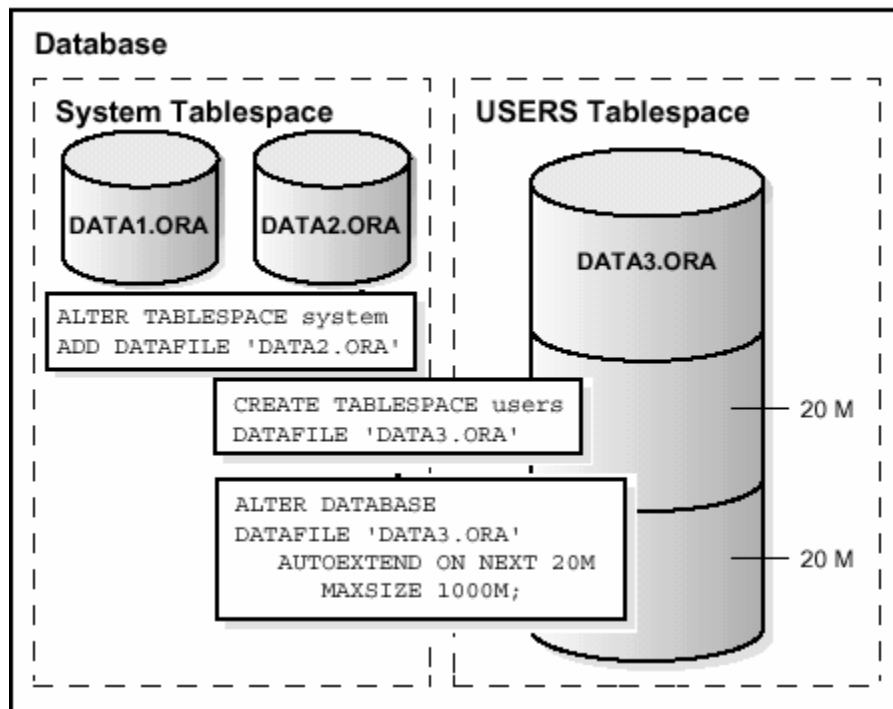


Figura 2.19⁴²: Estructura física de la base de datos Oracle

2.9 Oracle Enterprise Linux 5

2.9.1 Historia

Oracle comenzó proporcionando soporte para el sistema gestor de base de datos Oracle sobre el sistema operativo Linux. Posteriormente el 26 de octubre de 2006, Oracle lanza su sistema operativo Oracle Linux⁴³, basado en Red Hat⁴⁴.

2.9.2 Especificaciones del sistema⁴⁵

- Arquitectura: x86, x86-64, IA64 (Itanium)
- Memoria RAM mínima: 512MB
- Espacio en disco: 1GB

⁴² FERNANDEZ, J. Estructura de una base de datos Oracle. (3). Recuperado el 10 de Septiembre de 2011, de: <http://dis.um.es/~jfernand/0405/dbd/DBD04T06-oracle.pdf>.

⁴³ WIKIPEDIA (2012). Oracle Linux. Recuperado el 20 de febrero de 2012, de: http://en.wikipedia.org/wiki/Oracle_Linux.

⁴⁴ Distribución del sistema operativo Red Hat desarrollada por la compañía del mismo nombre.

⁴⁵ ORACLE. (2012). Oracle Linux. Recuperado el 20 de febrero de 2012, de: <http://www.oracle.com/us/technologies/linux/025994.htm>.

- Máxima memoria RAM soportada: 64GB (x86), 2TB (x86_64)
- Máxima memoria virtual (espacio de usuario, kernel): GB/1GB (x86), 128TB/128TB (x86_64)
- Número máximo de CPUs, núcleos: 32 (x86), 256 (x86_64)
- Versión de kernel base: 2.6.18
- Compilador: GCC 4.1
- Librería C: GNU libc 2.5
- Ambiente de escritorio: GNOME 2.16

2.9.3 Sistemas de archivos soportados⁴⁶

Los sistemas de archivos soportados por Oracle Linux son:

- ext4 máximo tamaño de archivo de sistema: 16TB
- ext3 máximo tamaño de archivo de sistema: 2TB/16TB
- OCFS2 máximo tamaño de archivo de sistema: 4PB

2.9.4 Tecnologías de valor añadido Oracle⁴⁷

- **ASMLib.-** Librería de soporte para la administración de almacenamiento automático de la base de datos.
- **OCFS2.-** Sistema de archivos de disco compartido para almacenar archivos de la base de datos Oracle, para bases de datos en clústers.
- **Yum.-** Servidor público que ofrece plugins para Oracle Linux y Oracle VM Server⁴⁸.

⁴⁷ ORACLE (2012). Oracle Linux. Recuperado el 20 de Febrero de 2011, de: <http://www.oracle.com/us/technologies/linux/025994.htm>.

⁴⁸ Servidor de virtualización de Oracle.

2.9.5 Seguridad

Una de las principales ventajas de Oracle Linux es su kernel inquebrantable que corre directamente sobre el sistema operativo, lo que representa gran velocidad de procesamiento y la facilidad de añadir nuevas características para el sistema, sin la necesidad de reinstalar el sistema operativo.

Oracle Linux es un sistema modular, en el cual el núcleo o Kernel interactúa sobre el hardware permitiendo establecer horarios de acceso a los recursos del sistema en base a las aplicaciones procesadas.

Las aplicaciones corren sobre el espacio de usuario y llaman a un grupo de librerías del Kernel para acceder a los servicios.

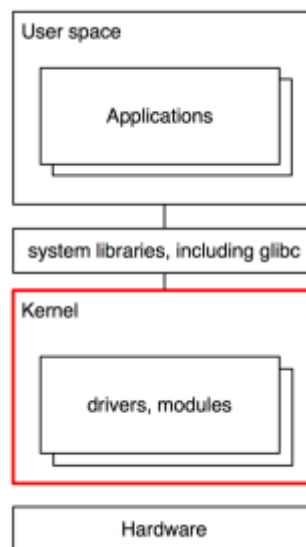


Figura 2.20⁴⁹: Estructura del sistema operativo Oracle Linux

⁴⁹ ORACLE. (2011). Certification with Oracle Linux. Recuperado el 20 de Febrero de 2011, de: <http://www.oracle.com/us/technologies/027626.pdf>.

2.10 Aplicaciones tradicionales versus Cloud Computing

Con el objeto de sustentar el crecimiento y nivel de desempeño de la Computación en la Nube a nivel mundial, se realizó un análisis bibliográfico, así como la tabulación e interpretación de resultados, basados en estadísticas existentes sobre este tema.

2.10.1 Análisis de características

Tabla 2.2: Análisis de características de sistemas tradicionales versus Computación en la nube

Categoría	Sistemas tradicionales	Computación en la nube
Sistema operativo	Requiere de sistemas operativos específicos	Multiplataforma
Almacenamiento	Requiere de dispositivos de almacenamiento de gran tamaño	No requiere de dispositivos de almacenamiento de gran tamaño
Internet	No requiere internet	Requiere internet
Dependencia de proveedor	Ninguna dependencia	Alta dependencia
Movilidad	Funcionalidad en sitio o por medio de VPN	Movilidad sin proceso adicional
Esquema de compra	Se compra hardware como activo y se implementa	Se compra servicios

	una arquitectura propia	
Esquema de negocio	Pago por activos fijos y recursos humanos (administrativo)	Pago por transaccionalidad
Esquema de escalamiento Técnico	Estático	Dinámico
Rango de prestación de Servicios	Nivel local	Nivel mundial

ESPACIO EN BLANCO
INTENCIONAL

2.10.2 Análisis estadístico

Porcentaje de empresas que utilizan Cloud Computing

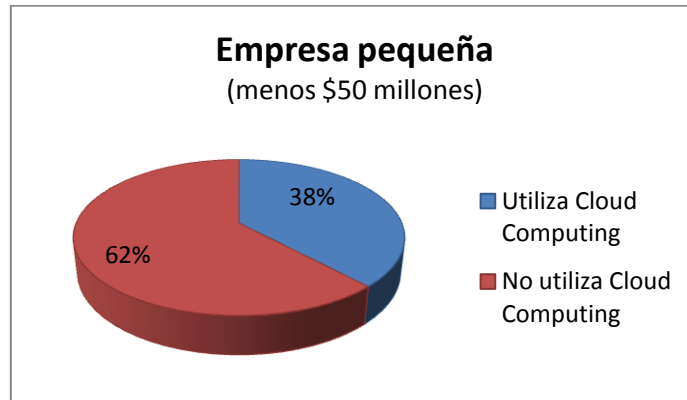


Figura 2.21: Porcentaje de empresas pequeñas que utilizan Cloud Computing.

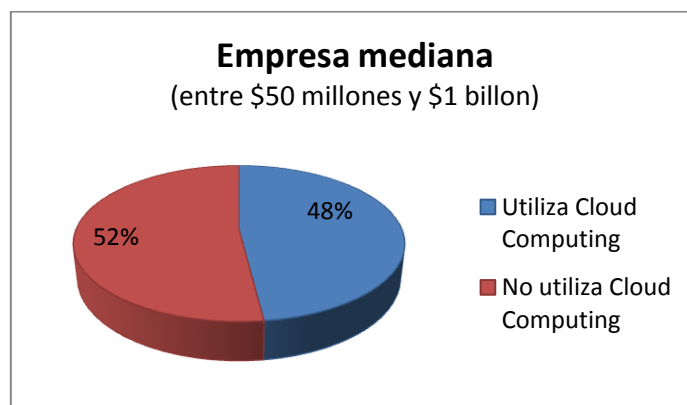


Figura 2.22: Porcentaje de empresas medianas que utilizan Cloud Computing

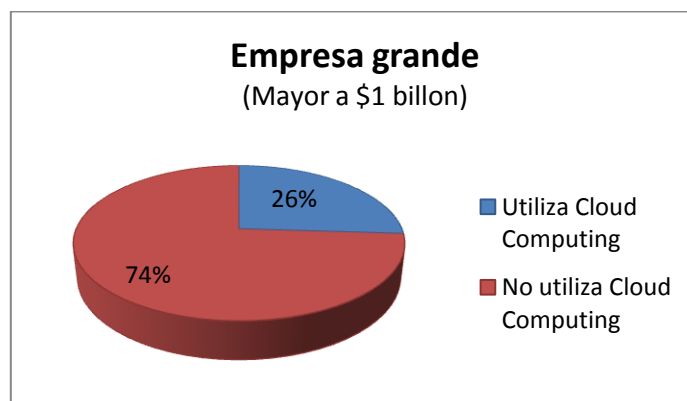


Figura 2.23: Porcentaje de empresas grandes que utilizan Cloud Computing

Como se muestra en las Figuras 2.21, 2.22 y 2.23⁵⁰, el nivel de uso de la computación en la nube, tiene su mayor porcentaje de acción en las empresas medianas. Si una **empresa mediana** tiene la visión de convertirse en una empresa grande, debe optimizar su infraestructura, reduciendo costos operativos y de mantenimiento.

Como se muestra en las Figuras 2.21, 2.22 y 2.23, en el nivel de uso de la computación en la nube, a las empresas medianas, le siguen las empresas pequeñas; esto se debe a que una **empresa pequeña** carece de datacenters debido a su bajo nivel de transaccionalidad; para ello la computación en la nube es una opción altamente viable que le permitirá reducir costos de hardware, mantenimiento y operación.

Como se muestra en las Figuras 2.21, 2.22 y 2.23, dentro de las empresas grandes se define el menor porcentaje de uso de la computación en la nube. Esto se debe a que una **empresa grande** tiene un datacenter altamente estructurado y un nivel alto de transaccionalidad. El impacto sobre el cambio de hardware, software y recurso humano técnico relacionado a un datacenter, es mayor que en los dos tipos de empresas restantes. Cabe destacar, que la relación entre el alto nivel de transaccionalidad de una empresa grande versus el esquema comercial de cobro por transacción de Cloud Computing versus el costo de mantenimiento y operación de un datacenter propio en una empresa grande, definen a Cloud Computing como una opción en vías de desarrollo; pero no viable a corto plazo en empresas grandes.

⁵⁰ OnLINE TECH. (2011). 2011 Cloud & IT Disaster Recovery Statistics. Recuperado el 19 de Febrero de 2012, de: <http://resource.onlinetech.com/2011-cloud-it-disaster-recovery-statistics/>.

Eficiencia en la recuperación de desastres

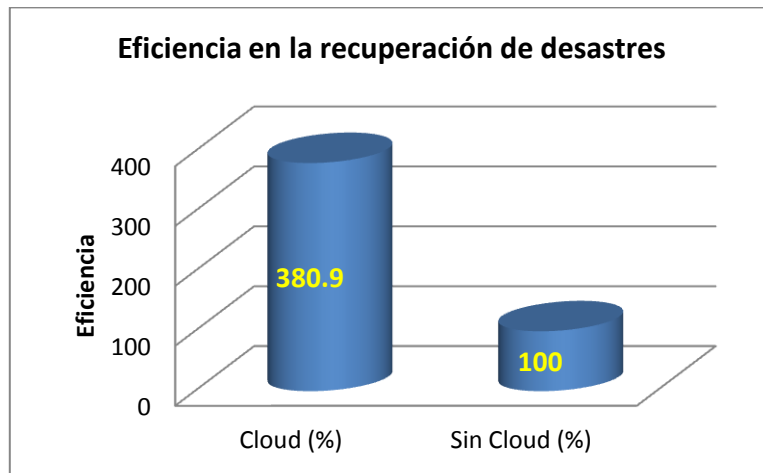


Figura 2.24⁵¹: Comparativa en la eficiencia en la recuperación de desastres

En la Figura 2.24 se muestra el nivel de efectividad en recuperación de desastres del 380.9% para aplicaciones sobre la nube frente a aplicaciones que no utilizan computación en la nube; esto se debe a las normas de calidad establecidas por las empresas que proveen servicios de computación en la nube.

2.10.3 Demanda de la nube

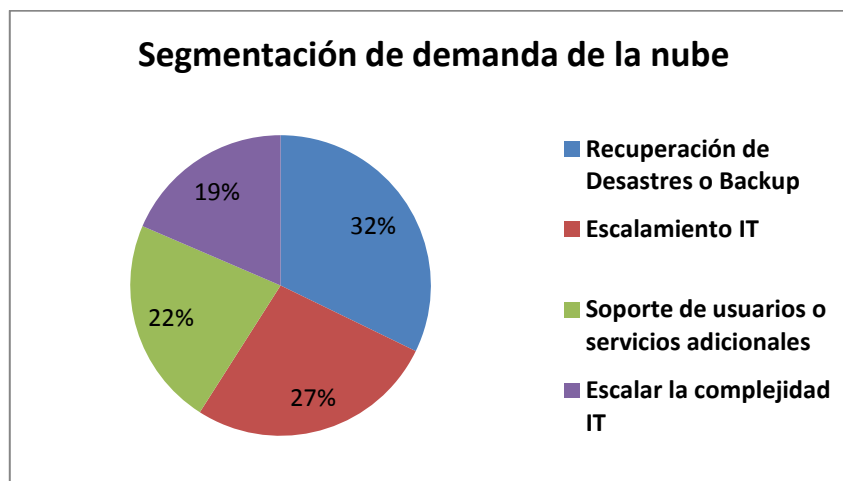


Figura 2.25⁵²: Segmentación de demanda de la nube

⁵¹ Datos basados en un promedio de 8 horas de inactividad para data centers y 2.1 horas de inactividad para esquemas en la nube; se aplicó regla de tres utilizando la siguiente relación $\forall 8 \text{ horas} = 100\%$ y $2.1 \text{ horas} = x$

De acuerdo a la Figura 2.25, las empresas requieren computación en la nube, en mayor porcentaje, para afrontar recuperación de desastres o gestión de respaldos. La segunda razón para utilizar computación en la nube es el escalamiento IT, que se refiere a nuevas implementaciones de tecnología para superar los desafíos de la línea de negocio en la línea del tiempo. La tercera razón para el uso de computación en la nube está ligada a servicios adicionales que la empresa necesita proveer a sus clientes; esto generalmente ocurre como respuesta a la diversificación de la línea de negocio y adaptación al cambio del frente comercial. La cuarta razón para el uso de computación en la nube es escalar la complejidad IT, que comprende la renovación total de los sistemas centrales de una empresa; esta renovación debe producirse paralelo al uso y mantenimiento del sistema antiguo.

⁵² Datos basados en: OnLINE TECH (2011). 2011 Cloud & IT Disaster Recovery Statistics. Recuperado el 19 de Febrero de 2012 de: <http://resource.onlinetech.com/2011-cloud-it-disaster-recovery-statistics/>. Se realizó la sumatoria de los datos originales para los siguientes segmentos: Recuperación de desastres o backup, escalamiento IT, soporte de usuarios y servicios adicionales, y se la definió como universo; posteriormente, utilizando regla de tres, se derivó los porcentajes de los segmentos basados en el universo (100%).

CAPÍTULO 3

DISEÑO

3.1 Introducción

El diseño del sistema de gestión de catálogo de productos Lithium ® comprende las etapas de exploración, planificación e iteraciones, de acuerdo a la metodología XP.

3.2 Historias de usuario

El cliente definió las siguientes historias de usuario para el sistema de gestión de catálogo de productos Lithium ®.

Cuadro 3.1 Historia de usuario *Autenticar usuario*

HISTORIA DE USUARIO			
Hugo Zumárraga			
Nombre	Autenticar usuarios	Número	1
Prioridad	Alta	Iteraciones	2
Riesgo	Alto	Versión	1
Descripción	Validar el ingreso de los usuarios al sistema, ya sea administrador, empleado o cliente.		
Observaciones			

Referencia:

- ANEXO B. HISTORIAS DE USUARIO

3.3 Modelo de Casos de Uso

La metodología XP plantea la suplantación de los tradicionales casos de uso por las historias de usuario; sin embargo los casos de uso definen más detalladamente los requerimientos del sistema, y son de carácter técnico. XP es flexible en cuanto a esto, ya que permite la adopción de herramientas de las metodologías existentes. En este caso, las plantillas de casos de uso de negocio y casos de uso de desarrollo de OUM.

3.3.1 Relación Historias de usuario – Casos de uso

Las historias de usuario tienen relación directa con los casos de uso. Las primeras están escritas en lenguaje no técnico y son el resultado de los requerimientos expresados por el cliente. Los casos de uso han tomado los datos obtenidos por las historias de usuario, detallándose en mayor medida y con carácter técnico.

A continuación la relación existente entre las historias de usuario obtenidas y los casos de uso.

Tabla 3.2 Relación Historias de usuario – Casos de uso

Historia de usuario	Casos de uso
Administrar productos y clasificarlos por categorías	- Administrar categoría - Administrar producto
Administrar unidades de medida	- Administrar unidad de medida
Administrar características, armar combos	- Administrar característica

de ellas y asignarles a los productos	<ul style="list-style-type: none"> - Relacionar característica a producto - Administrar valor de características - Administrar grupo de características - Administrar instancia de característica
Administrar artículos y su correspondiente stock	<ul style="list-style-type: none"> - Administrar artículo - Administrar stock de artículo
Crear lista de precios y asignar a clientes	<ul style="list-style-type: none"> - Administrar lista de precios - Administrar relación lista de precios – persona
Ordenar artículos	<ul style="list-style-type: none"> - Ordenar artículo
Crear lista de pedidos	<ul style="list-style-type: none"> - Revisar lista de pedidos
Gestionar roles	<ul style="list-style-type: none"> - Administrar rol
Administrar módulos	<ul style="list-style-type: none"> - Administrar modulo
Gestionar menus	<ul style="list-style-type: none"> - Administrar menu
Gestionar transacciones	<ul style="list-style-type: none"> - Administrar transacción
Administrar menús, módulos y transacciones	<ul style="list-style-type: none"> - Gestionar transacción por rol - Gestionar menú por módulo - Gestionar transacción por menu
Administrar personas	<ul style="list-style-type: none"> - Administrar persona
Gestionar estructura de la empresa	<ul style="list-style-type: none"> - Gestionar estructura de la

	organización
--	--------------

Referencias:

- ANEXO F. RA.015 CASOS DE USO DE NEGOCIO
- ANEXO G. RA.023 CASOS DE USO DE DESARROLLO

3.4 Plan de iteraciones

El Plan de iteraciones para el desarrollo del sistema de gestión de catálogo de productos Lithium ® consiste en un cronograma que contempla 2 iteraciones por cada grupo de tareas, de acuerdo las historias de usuario correspondientes y sus respectivos módulos. Las pruebas de aceptación y las pruebas de rendimiento se realizarán en 1 iteración al final de la versión del sistema.

Los grupos de tareas son:

- Modelamiento de base de datos
- Generación de componentes de negocio
- Generación de application module
- Generación de regiones
- Generación de clases base
- Pruebas de aceptación
- Pruebas de estrés

Adicionalmente se definió el Cronograma de Documentación.

Referencias:

- ANEXO E. PLAN DE ITERACIONES

3.5 Especificación de diseño

La especificación de diseño consiste en una descripción de las clases utilizadas en el sistema de gestión de catálogo de productos Lithium ®, los diagramas de secuencia y los modelos de base de datos de este.

Referencias:

- ANEXO H. DS.140 ESPECIFICACIÓN DE DISEÑO

3.6 Modelo de arquitectura

El modelo de arquitectura es una descripción de los componentes de arquitectura utilizados en el desarrollo del sistema de gestión de catálogo de productos Lithium ®.

Referencias:

- ANEXO I. EA.030 COMPONENTES DE ARQUITECTURA

CAPÍTULO 4

PRUEBAS

4.1 Introducción

Para garantizar la funcionalidad y rendimiento del sistema piloto de gestión de catálogo de productos Lithium ®, se han determinado una serie de pruebas.

El desempeño del sistema en las pruebas de aceptación está directamente relacionado con la programación de la funcionalidad, mientras que el desempeño del sistema en las pruebas de rendimiento está directamente relacionado con el hardware que cumple la función de servidor.

4.2 Pruebas de aceptación

El objetivo de las pruebas de aceptación es verificar si el sistema de gestión de catálogo de productos cumple con los flujos descritos en los casos de uso. Según la metodología Programación Extrema, el cliente debe realizar estas pruebas, para así *aceptar* el software que está siendo entregado por los desarrolladores. Se determinó casos de prueba para cada caso de uso del sistema.

4.2.1 Casos de prueba

Cuadro 4.1 Caso de prueba *Autenticar usuario*

Caso de prueba 1	
Código	Aut_usu
Caso de uso	Autenticar usuario
Objetivo	Detectar posibles defectos de la funcionalidad Autenticar usuario.
Requisitos asociados	
Variables de entrada	Usuario, contraseña
Flujo normal	<ol style="list-style-type: none"> 1. Ingresa nombre de usuario y contraseña. 2. Presiona clic en el botón “Ingresar”. 3. Verifica usuario y contraseña en el directorio LDAP del servidor Oracle Weblogic y presenta la pantalla principal del sistema.
Resultado normal esperado	El usuario ingresa al sistema.
Flujo alternativo	3.1a <datos obligatorios no se han ingresado> --- 3.1b <Los datos ingresados son incorrectos>
Resultado alternativo esperado	3.1a Presenta un mensaje indicando que un valor es requerido y que debe ser ingresado. --- 3.1b Presenta un mensaje indicando que los datos ingresados no son correctos.
Evaluación de prueba	
Fecha de ejecución	
Ejecutado por	Hugo Zumárraga
Lugar de ejecución	EMergeSoft
Resultado	Satisfactorio
Observaciones	Ninguna
Gravedad del error	N/A

Notas del programador	--
Estado	Correcto
Acciones de corrección	N/A
Corregido por	N/A

Referencia:

- ANEXO J. CASOS DE PRUEBA

4.3 Pruebas de rendimiento

4.3.1 Objetivos de las pruebas de rendimiento

El objetivo de las pruebas de rendimiento es determinar los niveles de carga máximos para el servidor de aplicaciones y los niveles de respuesta del sistema frente a diferentes niveles de carga y su proyección en la experiencia del usuario.

4.3.2 Uso

Un escenario de prueba está considerado como la totalidad de las transacciones ejecutadas sistemáticamente.

El software de simulación de carga Jmeter realiza la grabación del escenario o plan de pruebas, registrando como etiqueta, cada página visitada por el usuario. Para el presente sistema no existe tal nivel de detalle, en la captura de las etiquetas que hacen referencia a una página visitada, en la ejecución de la prueba, puesto que la interfaz del sistema incorpora una región estática principal (.jspx), que carga regiones (.jsff) dinámicamente, como muestra la Figura 4.1.

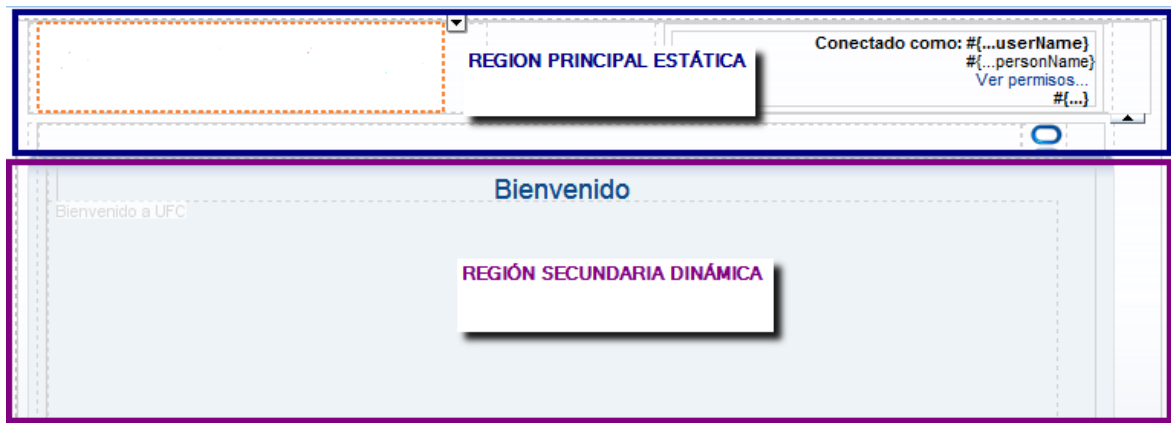


Figura 4.1: Estructura de la interfaz del sistema de gestión de catálogo de productos

Lithium ®

4.3.3 Variables

Se utilizarán 3 variables:

- **Número de hilos.-** Representa el número de usuarios a ejecutarse en el escenario de prueba.
- **Período de subida.-** Representa el tiempo en el cual un nuevo usuario ejecutará el escenario de prueba
- **Contador del bucle.-** Representa el número de iteraciones para un escenario de prueba

Fórmulas de generación de carga

$$\text{Periodo de carga} = \frac{\text{periodo de subida}}{\text{número de hilos}}$$

Por ejemplo, para generar el escenario cada 30 segundos, hasta llegar a 50 usuarios (número de hilos), despejando de la fórmula de *Periodo de carga* se define el *Período de subida*:

$$\text{Periodo de subida} = \text{Periodo de carga} * \text{Número de hilos}$$

*Periodo de subida = 30 segundos * 50 hilos*

Periodo de subida = 150 segundos

4.3.4 Gráficos empleados

Para las pruebas de rendimiento, se emplearon los siguientes gráficos:

- Gráfico de tiempos máximos de respuesta
- Curva de spline
- Tabla de resumen

Tiempos máximos de respuesta

Denota los máximos tiempos (milisegundos) de ejecución de fragmento en un escenario.

Curva de spline

Es un conjunto de coordenadas (puntos de control) que definen una curva generada por métodos de interpolación. Esta curva permite establecer la fluctuación de tiempo de respuesta de la aplicación.

Tabla de resumen de desempeño

La tabla de muestra definida por el software de generación de carga Jmeter, contiene los siguientes campos:

- **Media.-** Media aritmética de los tiempos de respuesta de la aplicación.
- **Mediana.-** Mediana aritmética de los tiempos de respuesta de la aplicación.

- **Línea de 90%.-** Tiempo de respuesta en el que se encuentra el 90% de las solicitudes.
- **Min.-** Mínimo tiempo de respuesta para una solicitud.
- **%Error.-** Porcentaje de una solicitud con errores
- **Rendimiento.-** Es la cantidad de solicitudes que el servidor procesa por hora.
- **Kb/Sec.-** Cantidad de kilobytes que el servidor procesa por segundo
- **Max.-** Máximo tiempo de respuesta para una solicitud

4.4 Hardware utilizado

La Tabla 4.1 indica las características del servidor utilizado para las pruebas de aceptación.

Tabla 4.1 Características del servidor usado para las pruebas de aceptación

Característica	Valor
Procesador	Intel Core i3-2100 @ 3.10 Ghz
Memoria RAM	16 GB
Espacio en disco duro	500 GB
Adaptador de red	Intel ® 82579V Gigabit Network Connection

4.5 Software utilizado

El software empleado para realizar las pruebas de estrés es JMeter, definido como una herramienta de generación de carga con especial tendencia a aplicaciones web. Es parte de Apache Jakarta⁵³. JMeter soporta pruebas unitarias con bases de datos JDBC, FTP, servicios web, JMS y HTTP. Actualmente se encuentra en la versión 2.6 r1237317.

4.6 Pruebas de estrés

Las pruebas de estrés consisten en 4 tipos:

- Test de rendimiento
- Test por defecto
- Test de carga sin exceso de peticiones
- Test de sobrecarga

4.6.1 Test de rendimiento

El objetivo de esta prueba es la revisión del despliegue de todas las transacciones del sistema, sin generar carga de concurrencia al servidor de aplicaciones.

Para esta prueba se ha definido los parámetros especificados a continuación en la Tabla 4.2

⁵³ Proyecto de Apache Software Foundation que crea y mantiene software de código abierto (open source) para la plataforma Java.

Tabla 4.2 Parámetros usados para el test de rendimiento.

Detalle	Valor
Número de hilos	1
Período de subida	1
Contador de bucle	1

Esta configuración indica la conexión de 1 usuario en 1 segundo, quien navegará 1 vez por la aplicación.

ESPACIO EN BLANCO
INTENCIONAL

Resumen de desempeño

Etiqueta	# Muestras	Media	Mín	Máx	Desv. Estándar	% Error	Rendimiento	Kb/sec	Media de Bytes
/embp	1	10	10	10	0,00	0,00%	100,0/sec	525,29	5379,0
/embp/	1	4	4	4	0,00	0,00%	250,0/sec	1413,33	5789,0
/embp/jsessio...	1	4	4	4	0,00	0,00%	250,0/sec	1413,33	5789,0
/embp/adfAuth...	3	5	4	6	0,82	0,00%	5,8/sec	30,66	5379,0
/embp/faces/lo...	6	3	2	5	1,00	0,00%	11,6/sec	65,36	5789,0
/embp/faces/lo...	1	23	23	23	0,00	0,00%	43,5/sec	30,49	718,0
/embp/faces/m...	25	6	2	45	7,97	0,00%	54,7/sec	72,23	1352,1
/embp/faces/m...	31	6	4	11	1,61	0,00%	72,1/sec	35,69	507,0
/embp/faces/u...	1	4	4	4	0,00	0,00%	250,0/sec	142,58	584,0
/embp/faces/m...	1	3	3	3	0,00	0,00%	333,3/sec	1884,44	5789,0
Total	71	6	2	45	5,39	0,00%	129,1/sec	220,95	1752,7

Figura 4.2: Resumen de desempeño – Test de rendimiento

Gráfico de máximos tiempos de respuesta

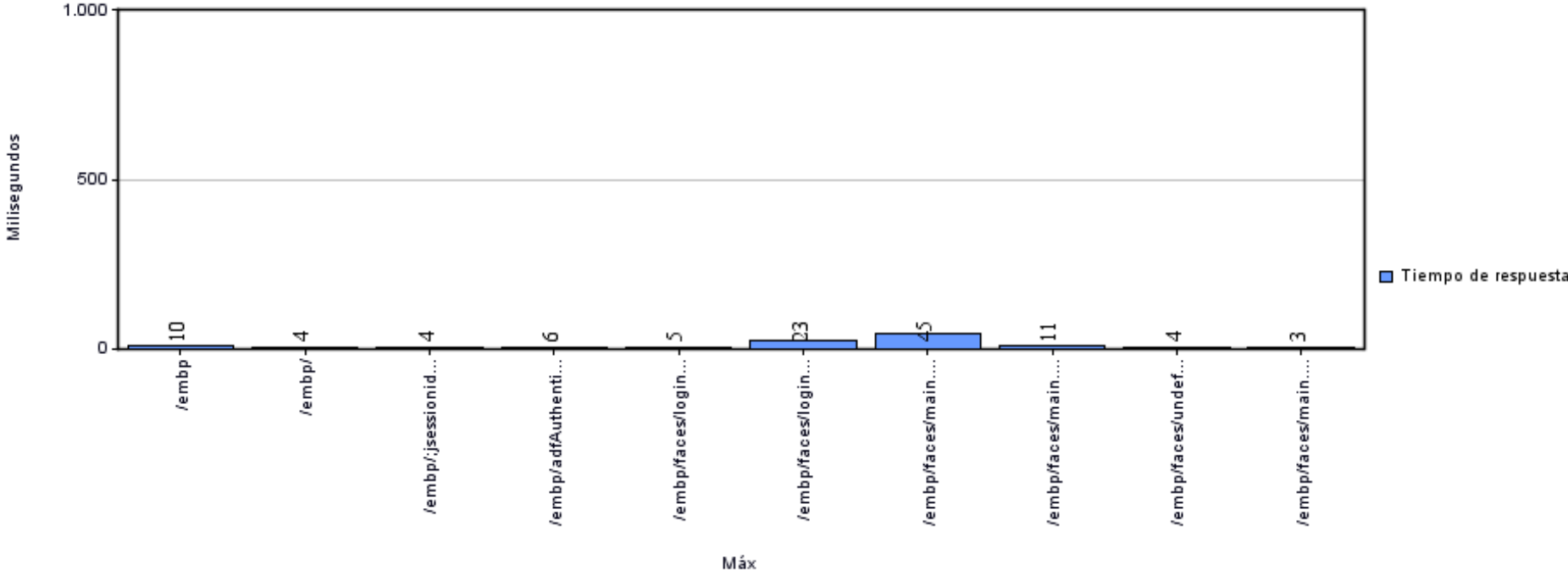


Figura 4.3: Gráfico de máximos tiempos de respuesta – Test de rendimiento

Curva de spline



Figura 4.4: Curva de spline – Test de rendimiento

Análisis de test de rendimiento

La tabla de resumen de rendimiento (Fig. 4.2) muestra un porcentaje total de error de 0% lo cual garantiza al 100% el correcto despliegue funcional de la aplicación.

En el gráfico de máximo tiempo de respuesta (Fig. 4.3) se denota como el más alto valor de carga de una transacción 45ms, lo que proporciona un parámetro que garantiza la velocidad de despliegue de los componentes gráficos de la aplicación.

La curva de spline (Fig. 4.4) muestra fluctuaciones en el tiempos de respuesta entre 2 y 45 ms; en este nivel de fluctuación el usuario percibe que el sistema está reaccionando instantáneamente.

Los resultados proporcionados por esta prueba garantizan el correcto despliegue y funcionalidad de la aplicación.

4.6.2 Test por defecto

El objetivo de esta prueba es controlar el correcto comportamiento de la interfaz frente a una carga menor a la mitad del estándar de concurrencia normal.

Para esta prueba se ha definido los siguientes especificados a continuación en la Tabla 4.3.

Tabla 4.3 Parámetros usados para el Test por defecto.

Detalle	Valor
Número de hilos	50
Período de subida	100
Contador de bucle	1

Esta configuración indica la conexión de 50 usuarios en 100 segundos; estos usuarios navegarán 1 vez por la aplicación.

ESPACIO EN BLANCO
INTENCIONAL

Resumen de desempeño

Etiqueta	# Muestras	Media	Mín	Máx	Desv. Estándar	% Error	Rendimiento	Kb/sec	Media de Bytes
/embp	50	163	6	3005	469,21	0,00%	30,4/min	2,66	5375,0
/embp/	50	68	3	1625	239,67	0,00%	30,4/min	2,86	5786,0
/embp/jsessio...	50	102	3	3226	456,01	0,00%	30,4/min	2,86	5786,0
/embp/adfAuth...	150	75	4	2136	208,63	0,00%	1,5/sec	7,88	5375,0
/embp/faces/lo...	300	46	2	2898	210,17	0,00%	3,0/sec	16,97	5785,7
/embp/faces/lo...	50	193	9	2756	476,06	0,00%	30,2/min	0,35	719,0
/embp/faces/m...	1250	28	2	3232	137,19	0,00%	12,5/sec	16,55	1352,4
/embp/faces/m...	1550	24	4	2706	101,83	0,00%	15,5/sec	7,71	508,0
/embp/faces/u...	50	57	3	2568	358,74	0,00%	30,3/min	0,29	583,3
/embp/faces/m...	50	10	2	337	46,65	0,00%	30,4/min	2,86	5786,0
Total	3550	36	2	3232	171,54	0,00%	35,4/sec	60,59	1752,6

Figura 4.5: Resumen de desempeño – Test por defecto

Gráfico de máximos tiempos de respuesta

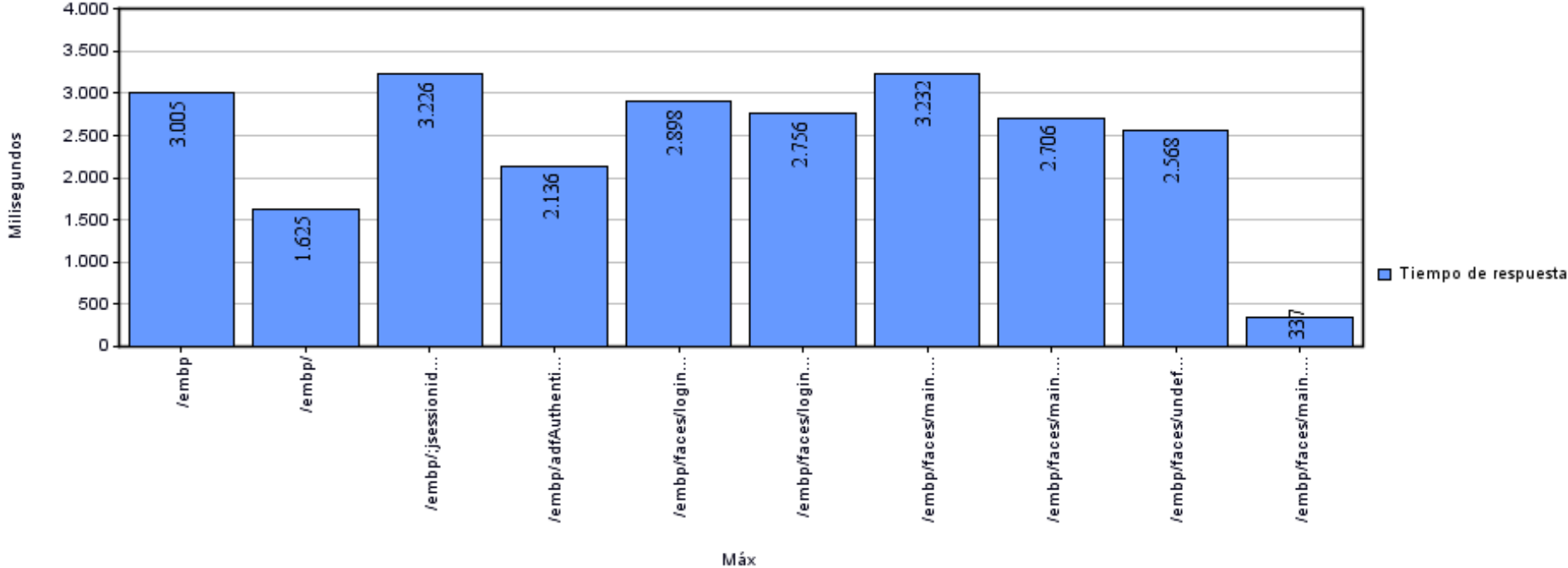


Figura 4.6: Gráfico de máximos tiempos de respuesta – Test por defecto

Curva de spline



Figura 4.7: Curva de spline – Test por defecto

Análisis de test por defecto

La tabla de resumen de rendimiento (Fig. 4.5) muestra un porcentaje total de error de 0% lo cual garantiza al 100% el correcto despliegue funcional de la aplicación.

En el gráfico de máximos tiempos de respuesta (Fig. 4.6) se denota como el más alto valor de carga de una transacción 3232m; este valor se encuentra dentro del rango aceptable para el cual usuario no percibe un retraso en el despliegue del sistema.

La curva de spline (Fig. 4.7) muestra fluctuaciones en el tiempos de respuesta entre 2 y 3232 ms; en este nivel de fluctuación el usuario percibe que el sistema está procesado su requerimiento en un límite promedio.

Los resultados determinados en esta prueba muestran el correcto procesamiento de los datos reflejados en el despliegue de componentes.

4.6.3 Test de carga sin exceso de peticiones

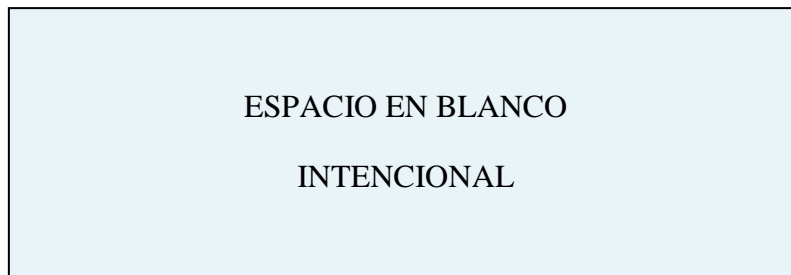
El objetivo de esta prueba es revisar el correcto funcionamiento del sistema frente a un escenario de concurrencia normal.

Para esta prueba se ha definido los parámetros especificados a continuación en la Tabla 4.4.

Tabla 4.4 Parámetros usados para el test de carga sin exceso de peticiones.

Detalle	Valor
Número de hilos	10
Período de subida	0
Contador de bucle	5

Esta configuración indica una conexión de 10 usuarios; estos usuarios navegarán 5 veces por la aplicación sin tiempo límite.



Resumen de desempeño

Etiqueta	# Muestras	Media	Mín	Máx	Desv. Estándar	% Error	Rendimiento	Kb/sec	Media de Bytes
/embp	50	138	6	989	202,37	0,00%	49,6/min	4,34	5375,0
/embp/	50	71	3	636	137,80	0,00%	49,8/min	4,69	5785,2
/embp/jsessio...	50	109	4	1678	277,57	0,00%	49,7/min	4,68	5786,0
/embp/adfAuth...	150	160	5	1536	282,30	0,00%	2,1/sec	11,18	5375,0
/embp/faces/lo...	300	104	3	1932	206,08	0,00%	4,3/sec	24,09	5785,5
/embp/faces/lo...	50	675	17	9495	1575,68	0,00%	49,2/min	0,58	719,0
/embp/faces/m...	1250	173	2	5397	349,56	0,00%	17,8/sec	23,52	1352,4
/embp/faces/m...	1550	203	5	4380	417,40	0,00%	22,1/sec	10,97	508,0
/embp/faces/u...	50	133	13	802	192,84	0,00%	46,7/min	0,44	583,2
/embp/faces/m...	50	104	5	576	159,20	0,00%	44,5/min	4,19	5786,0
Total	3550	182	2	9495	410,38	0,00%	50,3/sec	86,01	1752,6

Figura 4.8: Resumen de desempeño – Test de carga sin exceso de peticiones

Gráfico de máximos tiempos de respuesta

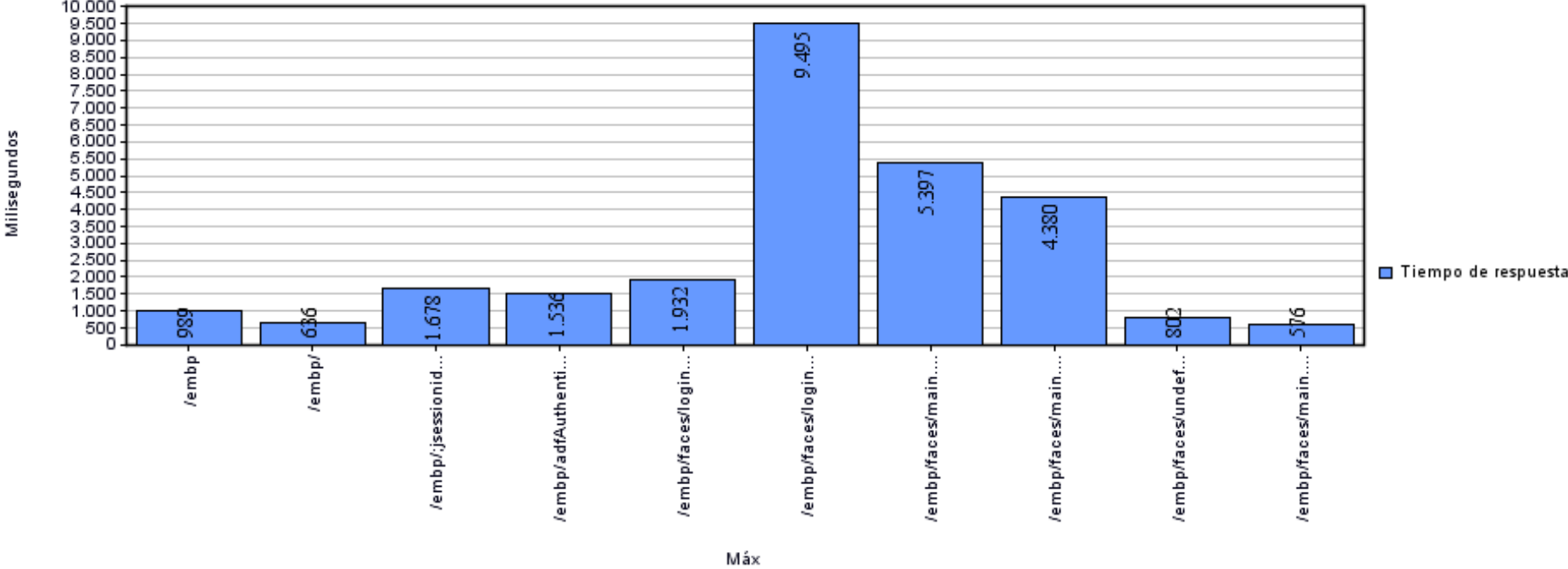


Figura 4.9: Gráfico de máximos tiempos de respuesta – Test de carga sin exceso de peticiones.

Curva de spline

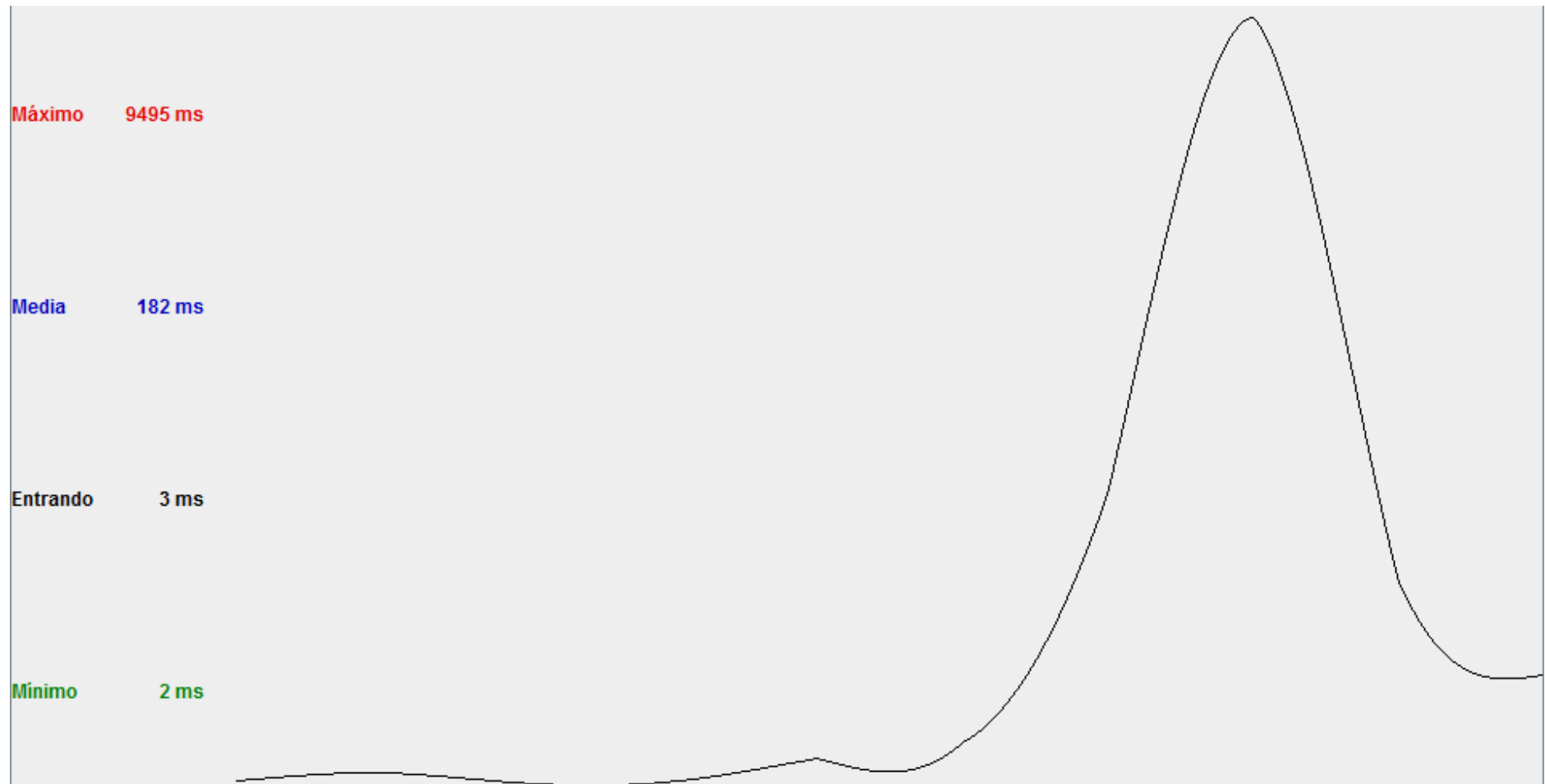


Figura 4.10: Curva de spline – Test de carga sin exceso de peticiones

Análisis de test de carga sin exceso de peticiones

La tabla de resumen de rendimiento (Fig. 4.8) muestra un porcentaje total de error de 0% lo cual nos garantiza al 100% el correcto despliegue funcional de la aplicación.

En el gráfico de máximos tiempos de respuesta (Fig. 4.9) se denota como el más alto valor de carga de una transacción 9495ms; para este tiempo, el usuario todavía se encuentra atento al sistema y siente una ligera demora en su solicitud.

La curva de spline (Fig. 4.10) muestra fluctuaciones en el tiempos de respuesta entre 2 y 9495 ms; en este nivel de fluctuación el usuario percibe una demora, mas aún no la interpreta como una interrupción.

Los resultados de la prueba muestran tiempos de respuesta satisfactorios para un nivel de carga normal en el funcionamiento de la aplicación.

4.6.4 Test de sobrecarga

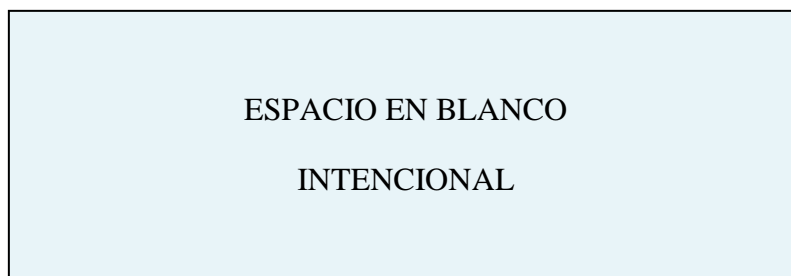
El objetivo de esta prueba es definir el comportamiento del sistema frente a un escenario de saturación del servidor y establecer los máximos tiempos de respuesta.

Para esta prueba se ha definido los siguientes parámetros especificados a continuación en la Tabla 4.5.

Tabla 4.5 Parámetros usados para el Test de sobrecarga

Detalle	Valor
Número de hilos	50
Período de subida	5
Contador de bucle	1

Esta configuración indica una conexión de 50 usuarios en 5 segundos. Estos usuarios navegarán 1 vez la aplicación.



Resumen de desempeño

Etiqueta	# Muestras	Media	Mín	Máx	Desv. Estándar	% Error	Rendimiento	Kb/sec	Media de Bytes
/embp	50	585	74	1347	296,63	0,00%	9,2/sec	48,04	5376,0
/embp/	50	219	3	704	232,84	0,00%	9,3/sec	52,38	5787,0
/embp/jsessio...	50	187	4	710	185,27	0,00%	9,1/sec	51,18	5787,0
/embp/adfAuth...	150	1343	6	12253	2249,85	0,00%	2,3/sec	12,01	5376,0
/embp/faces/lo...	300	463	3	7177	1044,20	0,00%	4,6/sec	25,85	5787,0
/embp/faces/lo...	50	225	32	460	128,52	0,00%	8,0/sec	5,60	719,0
/embp/faces/m...	1250	873	13	16611	1811,35	0,00%	19,2/sec	25,32	1353,5
/embp/faces/m...	1550	845	19	16760	1619,15	0,00%	24,2/sec	12,04	509,0
/embp/faces/u...	50	236	101	637	136,37	0,00%	7,2/sec	4,13	584,2
/embp/faces/m...	50	884	7	8596	1608,66	0,00%	1,1/sec	6,02	5786,2
Total	3550	805	3	16760	1640,15	0,00%	53,9/sec	92,33	1753,6

Figura 4.11: Resumen de desempeño – Test de sobrecarga

Gráfico de máximos tiempos de respuesta

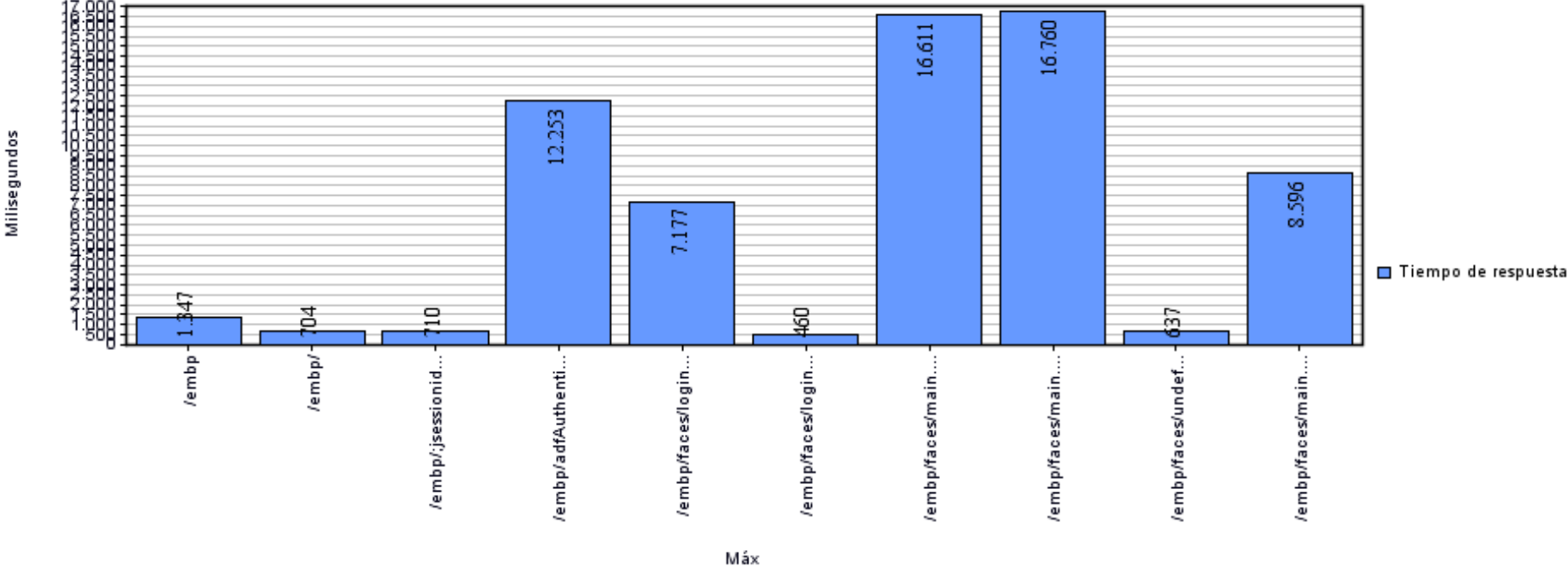


Figura 4.12: Gráfico de máximos tiempos de respuesta – Test de sobrecarga

Curva de spline

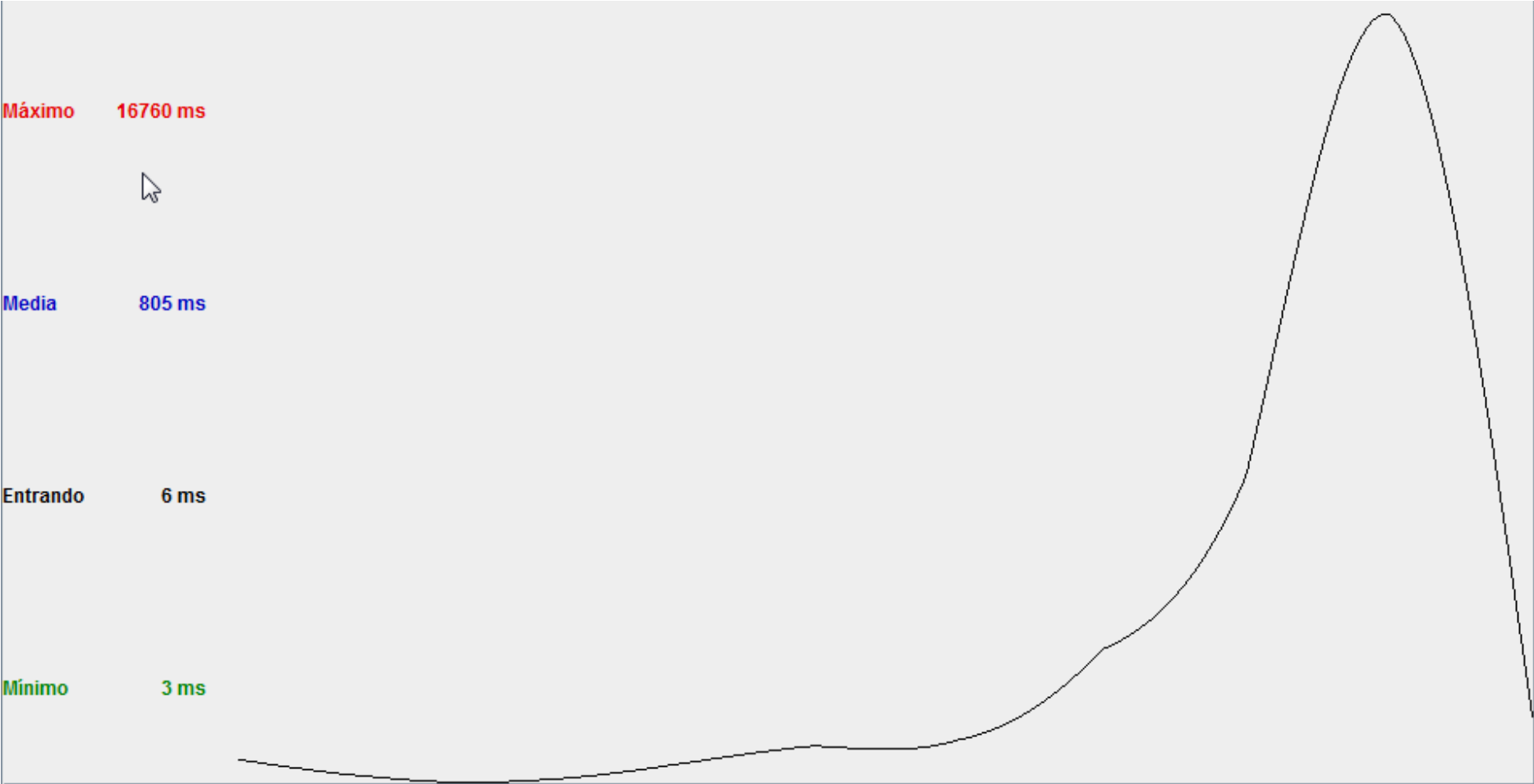


Figura 4.13 Curva de spline – Test de sobrecarga

Análisis del test de sobrecarga

La tabla de resumen de rendimiento (Fig. 4.11) muestra un porcentaje total de error de 0% lo cual nos garantiza al 100% el correcto despliegue funcional de la aplicación.

En el gráfico de máximos tiempos de respuesta (Fig. 4.12) se denota como el más alto valor de carga de una transacción 16760; para este tiempo, el usuario ha perdido totalmente su atención en el servicio web.

La curva de spline (Fig. 4.13) muestra fluctuaciones en los tiempos de respuesta entre 2 y 16760ms; en este nivel de fluctuación el usuario realiza otras tareas mientras espera que su ordenador termine de cargar la transacción.

Los resultados de la prueba muestran que pese a la sobrecarga del servidor de aplicaciones, no se generó ningún error lo cual, nos garantiza la correcta definición de la arquitectura del sistema en función del hardware utilizado.

CAPÍTULO 5

CONCLUSIONES Y RECOMENDACIONES

5.1 Conclusiones

- El modelo de arquitectura en la nube de Oracle definido sobre Oracle ADF (framework de desarrollo de aplicaciones de Oracle), generó una aplicación, con estructura de módulos, definidos para instancia de despliegue como librerías, instaladas en el servidor de aplicaciones Oracle Weblogic 11g; el uso de una librería permite realizar mantenimiento de un módulo, sin afectar la transaccionalidad total del sistema en tiempo real.
- Se utilizó una relación parametrizable de características de productos; esto permite al cliente diversificar su línea de negocio sin costos adicionales de desarrollo.
- El uso de Programación Extrema y Método Unificado Oracle en el desarrollo del sistema de catálogo de productos Lithium ® fue factible debido al análisis previo de fortalezas y mejores prácticas de ambas metodologías, en base al nivel de abstracción del negocio, la complejidad de la infraestructura base del sistema, la estimación de tiempo de Proyecto, el contingente humano disponible y el monto asignado al Proyecto.
- Para colaborar con el desarrollo del sistema y complementar la documentación se elaboraron los siguientes documentos: Especificación de requerimientos de software (basado en la norma IEEE830), Documento de visión, Manual de instalación y Manual de usuario. Esto fue factible debido a que XP permite la utilización de herramientas de otras metodologías o de la ingeniería de software.

- De la experiencia en la aplicación de las metodologías de desarrollo Programación Extrema y Método Unificado Oracle en el sistema de gestión de catálogos Lithium ®, se encontró que la Programación Extrema promueve la codificación mientras que el Método Unificado Oracle, la documentación. Esto es importante debido a que un software debe ser desarrollado y documentado eficiente y correctamente. Utilizando XP y OUM se encontró este equilibrio.
- Se empleó un protocolo ligero de acceso a directorios (LDAP) sobre el servidor de aplicaciones Oracle weblogic 11g, el cual valida y administra la autenticación de usuario, garantizando la seguridad de acceso a la aplicación eliminando la vulnerabilidad a un proceso de sql injection⁵⁴, como sucede en sistemas que almacenan usuarios y contraseñas en tablas de una base de datos.

5.2 Recomendaciones

- Para realizar un proyecto utilizando framework de desarrollo ADF se deben utilizar las bases de datos certificadas por Oracle como son: Oracle Database 11g, Sql Server 2008 e IBM DB2 9.5.
- Para el despliegue de un proyecto utilizando el framework de desarrollo ADF se debe utilizar los servidores de aplicaciones certificados por Oracle: IBM WebSphere 7.0.0.13 y Oracle Weblogic 11g.
- Para la oferta de un proyecto utilizando ADF, es conveniente desarrollar una prueba conceptual basada en una sección del proyecto propuesto, en la cual el cliente pueda cuantificar y cualificar las capacidades y beneficios en la adopción de esta nueva tecnología.

⁵⁴ Ingreso de código sql invasor en el código programado para alterar el funcionamiento normal de la base de datos, siendo el código de carácter malicioso o espía.

- Realizar un análisis del proyecto de desarrollo de software y la arquitectura a utilizar, para definir metodologías acordes al nivel de abstracción relativo a la complejidad de negocio, para promover un desarrollo eficiente de software.
- Se sugiere al Departamento de Ciencias de la Computación de la Escuela Politécnica del Ejército, la revisión de casos de éxito de implementaciones internacionales de software como parte de su malla curricular, para diversificar la visión y entendimiento de negocio de los estudiantes.

BIBLIOGRAFÍA

1. ORACLE. (2011). Cloud Architecture. Recuperado el 19 de Febrero de 2012, de: <http://cloud.oracle.com/mycloud/f?p=service:architecture:0>.
2. WIKIPEDIA. (2009). Computación en la nube. Recuperado el 19 de Febrero de 2012 de: http://es.wikipedia.org/wiki/Computaci%C3%B3n_en_la_nube.
3. BECK, K. (1999). Extreme Programming explained. USA: Addison-Wesley Professional.
4. BECK K., FOWLER M. (2000), Planning Extreme Programming. USA: Addison-Wesley Professional.
5. JOSKOWICZ, J. (2008). Reglas y prácticas en eXtreme Programming. Recuperado el 18 de Febrero de 2012, de: <http://ie.fing.edu.uy/~josej/docs/XP%20-%20Jose%20Joskowicz.pdf>.
6. GARCÍA J., TECHNOLOGY EVALUATION CENTERS. (2010). Inteligencia empresarial. Recuperado el 19 de febrero de 2012, de: http://www2.technologyevaluation.com/es/Research/ResearchHighlights/BusinessIntelligence/2010/09/research_notes/es/prn_PN_BI_JG_08_02_10_SP_1.asp.
7. SUN MICROSYSTEMS. (2009). Introduction to Cloud Computing Architecture. Recuperado el 18 de Febrero de 2012, de: <http://www.oracle.com/us/dm/44034-cloudcomputing-332069.pdf>.
8. DEMARESK, G. WANG, R. (2010). Oracle Cloud Computing. Recuperado el 18 de Febrero de 2012, de: <http://www.oracle.com/us/technologies/cloud/oracle-cloud-computing-wp-076373.pdf>.
9. BABELIAS. (2012). Programación extrema. Recuperado el 18 de Febrero de 2012, de: <http://babelias-blog.com/2012/02/programacion-extrema/>.

10. VILLEGAS, A. A propósito de programación extrema XP (eXtreme Programming). Recuperado el 18 de febrero de 2012, de:
<http://www.monografias.com/trabajos51/programacion-extrema/programacion-extrema2.shtml>.
11. ORACLE (2012). Oracle Linux. Recuperado el 20 de Febrero de 2012, de:
<http://www.oracle.com/us/technologies/linux/025994.htm>.
12. ORACLE. (2011). Oracle Unified Method (OUM). Recuperado el 20 de Febrero de 2012, de: <http://www.oracle.com/us/products/consulting/resource-library/oracle-unified-method-069204.pdf>.
13. ORACLE. (2011). A History of Oracle JDeveloper and Oracle ADF Releases. Recuperado el 18 de Febrero de 2012, de:
<http://www.oracle.com/technetwork/developer-tools/jdev/jdev-history-099970.html>.
14. MILLS, D., KOLETZKE, P., ROY-FADERMAN, A. (2010). Oracle JDeveloper 11g Handbook. USA: Oracle.
15. NIMPHIUS, F., MUNSINGER, L. (2010). Oracle Fusion Developer Guide. USA: Oracle.
16. WIKIPEDIA. (2012). Oracle WebLogic Server. Recuperado el 19 de Febrero, de 2012 de: http://en.wikipedia.org/wiki/Oracle_WebLogic_Server.
17. ORACLE. (2012). WebLogic Server Domains. Recuperado el 19 de Febrero de 2012. de: http://docs.oracle.com/cd/E24329_01/web.1211/e24446/domains.htm.
18. FERNANDEZ, J. Estructura de una base de datos Oracle. (3). Recuperado el 10 de Septiembre de 2011, de: <http://dis.um.es/~jfernand/0405/dbd/DBD04T06-oracle.pdf>.

19. AGUILAR, A. (2002). Programación Extrema y Software Libre. Recuperado el 20 de Febrero de 2012, de:
<ftp://jano.unicauca.edu.co/cursos/EnfasisIV/Transp/Anteriores/gasuxp.pdf>.
20. WIKIPEDIA (2012), Oracle Linux. Recuperado el 20 de febrero de 2012, de:
http://en.wikipedia.org/wiki/Oracle_Linux.
21. OnLINE TECH (2011). 2011 Cloud & IT Disaster Recovery Statistics. Recuperado el 19 de Febrero de 2012 de: <http://resource.onlinetech.com/2011-cloud-it-disaster-recovery-statistics/>.
22. ORACLE. (2012), How to Use Key Bindings. Recuperado el 18 de Febrero de 2012, de: <http://docs.oracle.com/javase/tutorial/uiswing/misc/keybinding.html>.
23. ORACLE. (2010). Oracle Application Development Framework Overview. Recuperado el 18 de Febrero de 2012, de:
<http://www.oracle.com/technetwork/developer-tools/adf/adf-11-overview-1-129504.pdf>.
24. WIKIPEDIA. (2012), Oracle Application Development Framework. Recuperado el 18 de Febrero de:
http://en.wikipedia.org/wiki/Oracle_Application_Development_Framework.
25. WIKIPEDIA. (2012). Oracle Database. Recuperado el 18 de Febrero de 2012 de:
http://es.wikipedia.org/wiki/Oracle_Database.
26. ORACLE. (2012). Oracle Linux FAQ. Recuperado el 18 de Febrero de 2012, de:
<http://www.oracle.com/us/technologies/027617.pdf>.
27. ORACLE. (2012). Oracle Database Cloud Service. Recuperado el 18 de Febrero de 2012, de: https://cloud.oracle.com/mycloud/service_database_architecture.html.

28. WIKIPEDIA. (2012). Sql Injection. Recuperado el 18 de Febrero de 2012, de:
http://en.wikipedia.org/wiki/Sql_Injection.
29. WIKIPEDIA. (2012). Lenguaje Unificado de Modelado. Recuperado el 18 de Febrero de 2012, de: <http://es.wikipedia.org/wiki/UML>.
30. WIKIPEDIA. (2012). Stakeholder. Recuperado el 14 de abril de 2012, de:
<http://es.wikipedia.org/wiki/Stakeholder>.
31. UDLAP. Capítulo 5: Pruebas. Recuperado el 14 de abril de 2012, de:
http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/sanchez_r_ma/capitulo5.pdf
32. ORACLE. (2010). Oracle Application Development Framework Overview.
Recuperado el 18 de Febrero de 2012, de:
<http://www.oracle.com/technetwork/developer-tools/adf/adf-11-overview-1-129504.pdf>.
33. WIKIPEDIA. (2012). CRUD. Recuperado el 7 de Julio de 2012, de:
<http://es.wikipedia.org/wiki/CRUD>.
34. WIKIPEDIA. (2012). Front-end y back-end. Recuperado el 6 de Julio de 2012, de:
http://es.wikipedia.org/wiki/Front-end_y_back-end.
35. WIKIPEDIA. (2012). Metadato. Recuperado el 6 de Julio de 2012, de
<http://es.wikipedia.org/wiki/Metadato>.
36. WIKIPEDIA. (2012). OCFS2. Recuperado el 6 de Julio de 2012, de:
<http://en.wikipedia.org/wiki/OCFS2>.
37. WIKIPEDIA. (2012). Entorno de desarrollo integrado. Recuperado el 8 de Julio de 2012, de: http://es.wikipedia.org/wiki/Entorno_de_desarrollo_integrado.
38. WIKIPEDIA. (2011). Binding. Recuperado el 8 de Julio de 2012, de:
<http://es.wikipedia.org/wiki/Binding>.

39. WIKIPEDIA. (2012). Red privada virtual. Recuperado el 8 de Julio de 2012, de:
http://es.wikipedia.org/wiki/Red_privada_virtual.
40. WIKIPEDIA. (2012). JavaServer Pages. Recuperado el 8 de Julio de 2012, de:
<http://es.wikipedia.org/wiki/JSP>.
41. WIKIPEDIA. (2012). Extensible Markup Language. Recuperado el 8 de Julio de 2012, de: http://es.wikipedia.org/wiki/Extensible_Markup_Language.

BIOGRAFÍA 1

Nombre: Hernán Eduardo Almachi Monteros

Lugar y Fecha de Nacimiento: Quito, 9 de diciembre de 1987

Educación Primaria: Unidad Educativa “La Salle”

Educación Secundaria: Unidad Educativa “La Salle”

Cursos:

- CCNA 1 - ESPE - 2009
- CCNA 2 - ESPE - 2010
- CCNA 3 - ESPE - 2010
- CCNA 4 - ESPE - 2011
- Suficiencia en el idioma inglés - ESPE - 2010
- Oracle Database Administration 11g - 2011
- Microsoft IT Pro Mini Camp 2012

Historial laboral:

- Analista de Riesgos B. - Banco Pichincha - 2010
- Desarrollador de software - eMergeSoft - 2011 a la actualidad

BIOGRAFÍA 2

Nombre: Pietro Dennis Andino Velásquez

Lugar y Fecha de Nacimiento: Quito, 9 de febrero de 1988

Educación Primaria: Unidad Educativa Experimental “Manuela Cañizares”

Educación Secundaria: Colegio “San Gabriel”

Reconocimientos:

- Portaestandarte de Pichincha - Unidad Educativa Experimental “Manuela Cañizares” - 1999
- Mérito Académico, Segundo Puesto - Colegio “San Gabriel” - 2000

Cursos:

- CCNA 1 - ESPE - 2007
- Suficiencia en el idioma inglés - ESPE - 2008
- Oracle Database Administration 11g - eMergeSoft - 2011

Historial laboral:

- Asistente de proyectos de software - Inveligent - 2010
- Administrador de bases de datos en FISA group - eMergeSoft - 2011 a la actualidad

HOJA DE LEGALIZACIÓN DE FIRMAS

ELABORADO POR

Hernán Eduardo Almachi Monteros

Pietro Dennis Andino Velásquez

DIRECTOR DE LA CARRERA

Ing. Mauricio Campaña

Lugar y Fecha: SANGOLQUÍ, AGOSTO DE 2012