

**INTERACCIÓN DE DISPOSITIVOS MÓVILES CON ARQUITECTURAS DE
CÓMPUTO AVANZADO BAJO UN MODELO DE INFRAESTRUCTURA COMO
SERVICIO**

ELIZABETH TATIANA CAMACHO PEREZ

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FISICOMECÁNICAS
ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
BUCARAMANGA**

2013

**INTERACCIÓN DE DISPOSITIVOS MÓVILES CON ARQUITECTURAS DE
CÓMPUTO AVANZADO BAJO UN MODELO DE INFRAESTRUCTURA COMO
SERVICIO**

ELIZABETH TATIANA CAMACHO PEREZ

Trabajo de grado para optar el título de Ingeniero de Sistemas

Director

CARLOS JAIME BARRIOS HERNANDEZ

Doctor in ComputerScience

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FISICOMECÁNICAS
ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
BUCARAMANGA
2013**

AGRADECIMIENTOS

- Al **Ph.D. Carlos Jaime Barrios Hernández**, Director del proyecto por toda sus enseñanzas y por su tiempo para la realización del proyecto.
- Grupo de Supercomputación y Calculo Científico de la UIS, (**SC3-UIS**), por su colaboración y apoyo.
- A la **Universidad Industrial de Santander** y al **CENTIC**, por proveer los recursos necesarios para el desarrollo del proyecto

DEDICATORIA

A mi mamá Pilar Pérez por el apoyo incondicional y la confianza que me brindo a lo largo de la carrera.

A Mario Gómez por su paciencia.

Al grupo de Supercomputación y Calculo Científico de la UIS por guiarme a través de todo este proceso y permitirme desarrollar la tesis.

CONTENIDO

	Pág.
INTRODUCCIÓN	19
1. PRESENTACIÓN DEL PROYECTO	20
1.1. ORIENTACIÓN SOBRE EL CONTENIDO DEL INFORME	20
1.2. ANTECEDENTES DEL PROYECTO	21
1.3. DEFINICIÓN DEL PROBLEMA	22
1.4. OBJETIVO GENERAL	23
1.5. OBJETIVOS ESPECIFICOS	23
1.6. JUSTIFICACIÓN	24
1.7. ALCANCES Y LIMITACIONES	24
2. MARCO TEÓRICO	25
2.1. GRID COMPUTING	25
2.2. COMPUTACIÓN MÓVIL.	27
2.3. COMPUTACIÓN MÓVIL EN COLOMBIA.	32
2.4. SISTEMA OPERATIVOS MOVILES	33
2.5. SISTEMA OPERATIVO ANDROID	35
2.6. SELECCIÓN DE SISTEMA OPERATIVO PARA EL DISEÑO DE LA APLICACIÓN.	37
2.7. CÓDIGO DEL LADO DEL CLIENTE (CLIENTSIDE SCRIPTS).	38
2.8. CÓDIGO DEL LADO DEL SERVIDOR (SERVER SIDE SCRIPTS).	38
2.9. TECNOLOGÍA UTILIZADA.	39
2.10. BASES DE DATOS	41
2.10.1 Modelos de Bases de Datos.	41
2.11. MySQL.	45
3. MARCO METODOLOGICO	46
4. CONCEPCIÓN Y DISEÑO DE LA HERRAMIENTA Y LABORES DE ADMINISTRACIÓN.	48

4.1 DEFINICION UML	48
4.2 USUARIOS DE LA PLATAFORMA LOGINGRID:	49
4.3 CASO DE USO GLOBAL:	50
4.4 MÓDULO DISEÑO INTERFAZ DEL SERVIDOR.	50
4.4.1 DIAGRAMA DE CASOS DE USO DISEÑO INTERFAZ DEL SERVIDOR.	50
4.4.2 DIAGRAMA DE CASOS DE USO DISEÑO INTERFAZ DEL SERVIDOR.	51
4.4.3 DIAGRAMA DE ACTIVIDADES PARA EL CASO DE USO ENVIAR SOLICITUD.	51
4.4.4 DIAGRAMA DE ACTIVIDADES PARA EL CASO DE USO SELECCIONAR NODOS.	52
4.5 MÓDULO DISEÑO DE SINCRONIZACIÓN.	53
4.5.1 DIAGRAMA DE CASO DE USO DEL MÓDULO DISEÑO DE SINCRONIZACIÓN.	54
4.5.2 DIAGRAMA DE ACTIVIDADES PARA EL CASO DE USO ENVIAR INFORMACIÓN.	54
4.6 Módulo aplicación para el dispositivo	55
4.6.1 MODELO DE CASOS DE USOS	56
4.6.2 MODELO CASO DE USO REGISTRO	56
4.6.3 MODELO CASO DE USO RESERVA	57
4.7 DIAGRAMA DE PAQUETES	58
4.8 DIAGRAMA DE COMPONENTES.	59
4.9 DISEÑO DE COMPONENTES MODULARES QUE INTEGRAN LA APLICACIÓN WEB	63
4.10 ANÁLISIS DE REQUERIMIENTOS.	65
4.10.1 MÓDULO INTERFAZ DEL SERVIDOR.	65
4.10.2 MÓDULO SINCRONIZACIÓN	66
5. ANALISIS EVALUACIÓN DE RENDIMIENTO	68
5.1 PRUEBAS DE INTEGRACIÓN.	68
5.2 MODULO DISEÑO INTERFAZ DEL SERVIDOR	68
5.2.1 DESCRIPCIÓN DE LA PRUEBA.	68
5.3 MÓDULO DE SINCRONIZACIÓN	69

5.3.1 DESCRIPCIÓN DE LA PRUEBA.	69
5.3.2 PROCEDIMIENTO DE LA PRUEBA.	69
5.3.3 RESULTADOS DE LA PRUEBA	70
5.4 MÓDULO APLICACIÓN	70
5.4.1DESCRIPCIÓN DE LA PRUEBA.	70
5.4.2 PROCEDIMIENTO DE LA PRUEBA	70
6. CONCLUSIONES	71
RECOMENDACIONES	73
BIBLIOGRAFIA Y FUENTES DE INFORMACION	75

LISTA DE TABLAS

	Pág.
Tabla 1 Sistemas computacionales que se relacionan con la movilidad, la adaptación, la conectividad inalámbrica, entre otros.....	28
Tabla 2 Aplicaciones de la computación móvil e inalámbrica Fuente: Revista ACIS, edición no.87 noviembre 2003 - enero de 2004.....	29
Tabla 3 Algunos de las áreas de investigación y últimos avances en Computación Móvil.....	31
Tabla 4 Usuarios del sistema	55
Tabla 5 Caso de uso REGISTRO.....	57
Tabla 6 Caso de uso RESERVA	58
Tabla 7 Procedimiento de la prueba. Módulo interfaz.	68
Tabla 8 Resultado primera prueba. Módulo Interfaz del servidor	69
Tabla 9 Procedimiento de la prueba. Módulo sincronización.	69
Tabla 10. Resultado prueba. Módulo sincronización.....	70
Tabla 11 Procedimiento de la prueba Aplicación.	70
Tabla 12 Cumplimiento de objetivos	74

LISTA DE FIGURAS

	Pág.
Figura 1 MODELO DE ACCESO A PHP	40
FIGURA 2.DIAGRAMA DE METODOLOGÍA XP	47
FIGURA 3.DIAGRAMA DE CASOS DE USO DISEÑO GLOBAL.....	50
FIGURA 4.DIAGRAMA DE CASOS DE USO DISEÑO INTERFAZ DEL SERVIDOR.....	51
FIGURA 5.DIAGRAMA DE ACTIVIDADES PARA EL CASO DE USO ENVIAR SOLICITUD.....	52
FIGURA 6.DIAGRAMA DE ACTIVIDADES PARA EL CASO DE USO SELECCIONAR NODO.....	53
FIGURA 7.DIAGRAMA DE CASOS DE USO DE SINCRONIZACIÓN.....	54
FIGURA 8.DIAGRAMA DE ACTIVIDADES PARA EL CASO DE USO ENVIAR INFORMACIÓN.....	55
FIGURA 9. CASOS DE USO.....	56
FIGURA 10.CASOS DE USO REGISTRO.....	56
FIGURA 11.CASOS DE USO RESERVA	57
FIGURA 12. DIAGRAMA DE NUEVO MODULO	58

FIGURA 13	DIAGRAMA DE ESTRUCTURA DEL SISTEMA.....	60
FIGURA 14.	DIAGRAMA DE SECUENCIA.....	61
FIGURA 15.	DIAGRAMA DE REGISTRO	62
FIGURA 16.	DIAGRAMA DE RESERVA.....	63
FIGURA 17.	DIRECTORIO RAÍZ DEL PROYECTO	64
FIGURA 18.	INICIO DE SESIÓN	65
FIGURA 19.	REPRESENTACIÓN GENERAL DEL SISTEMA.....	66

RESUMEN

TITULO: INTERACCIÓN DE DISPOSITIVOS MÓVILES CON ARQUITECTURAS DE CÓMPUTO AVANZADO BAJO UN MODELO DE INFRAESTRUCTURA COMO SERVICIO¹

AUTORES: ELIZABETH TATIANA CAMACHO PEREZ²

PALABRAS CLAVE: Clúster, Computación de alto rendimiento (HPC), computación distribuida, ComputeMode.

DESCRIPCION: El uso de recursos para la realización de cálculos que necesiten gran capacidad de procesamiento y tiempo de cómputo, es cada vez más importante en los campus Universitarios. El contar con máquinas que permitan crear diferentes clústers y que a su vez, estos clústers puedan interactuar con otros dentro del campus universitario, es fundamental para el aprovechamiento de los diferentes recursos que no son usados en tiempos no laborales. Contar con diferentes recursos que puedan ayudar en el campo de la computación distribuida, resulta de gran importancia para darle un valor agregado a la Universidad que lo está desarrollando. Actualmente se puede contar con diferentes herramientas que puedan ayudar a la creación de los clúster. Entre estas, una de la más completa y sencillas es la herramienta ComputeMode. El uso de esta herramienta permite crear un clúster ligero, agregando recursos de cómputo no usados.

Es por ello que el presente trabajo pretende analizar como primera instancia la manera en la que se podría dar la interacción entre recursos distribuidos de una infraestructura de cómputo en redes heterogéneas y con base a este análisis poder diseñar y desarrollar una estrategia que pueda resolver la limitación antes mencionada y que además se aprovechen los diferentes recursos con los que cuenta el campus Universitario.

¹Trabajo de grado en la Modalidad de Investigación

²Facultad de ingenierías Físico-mecánicas. Escuela de ingeniería de sistemas e informática.

Director: PhD: Carlos Jaime Barrios Hernández.

ABSTRACT

TITLE: INTERACTION WITH MOBILE DEVICES ADVANCED COMPUTER ARCHITECTURES under a model of infrastructure as a service

Authors: Elizabeth TATIANA PEREZ CAMACHO

KEYWORDS: Cluster, High Performance Computing (HPC), distributed computing, ComputeMode.

DESCRIPTION: The use of resources to perform calculations that need large processing power and computation time is increasingly important on college campuses. Having machines that create different clusters and in turn, these clusters can interact with others within the university campus, is essential for the utilization of the different resources that are not used in the not work (night shifts, holidays and late week). Also, having the possibility of having more and more with different resources that can help in the area of distributed computing is of great importance to give added value to the University that is being developed. Currently you can have different tools that can help create the cluster. Among these, one of the most complete and easy is the tool ComputeMode. Use this tool to create a light cluster, adding unused computing resources, but has the disadvantage when interacting with other nodes that are not in the same hierarchy of network, ie, preventing the use of other resources that are not within the same subnet of each cluster server.

This paper analyzes the way the first instance in which the interaction could give resources distributed computing infrastructure in heterogeneous networks and based on this analysis to design and develop a strategy that can solve the above limitation and also take advantage of the best way the different resources are there on campus.

Degree Work

Faculty of Physical Engineering – Mechanical.School of Systems Engineering.

Director: PhD: Carlos Jaime Barrios Hernández.

GLOSARIO

GRID:El término grid se refiere a una infraestructura que permite la integración y el uso colectivo de servidores y recursos de alto rendimiento

SSH: (Secure Shell, intérprete de órdenes segura), protocolo y programa cuya función

Es acceder a máquinas remotas a través de una red. Permite manejar por completo un computador mediante un intérprete de comandos.

EL NÚCLEO: Es el encargado de dar paso seguro a los distintos programas de acceso al hardware, así como la gestión de desarrollar procesos en el sistema de archivos y la memoria.

EL MIDDLEWARE: Conjunto de módulos que integra las diversas aplicaciones que hacen parte de los dispositivos móviles como el sistema de mensajería y comunicaciones, códec multimedia, etc.

ENTORNO DE EJECUCIÓN DE APLICACIONES. Este componente es un gestor de aplicaciones e interfaces programables que facilitan la creación de software.

ESCALABILIDAD: Es la posibilidad de aumentar la capacidad de clientes y servidores por separado. Cualquier elemento puede ser aumentado (o mejorado) en cualquier momento, o se pueden añadir nuevos nodos a la red (clientes y/o servidores).

INTERFAZ DE USUARIO. Este elemento es el medio o instrumento por el cual el usuario se comunica y relaciona con el dispositivo móvil, la cual incluye los menús, teclado, aplicaciones, listas, botones, gráficos, etc.

ARQUITECTURA CLIENTE-SERVIDOR: Es un modelo para el desarrollo de sistemas de información, en el que las transacciones se dividen en procesos

independientes que cooperan entre sí para intercambiar información, servicios o recursos. Se denomina cliente al proceso que inicia el diálogo o solicita los recursos, y servidor al proceso que responde a las solicitudes.

CLIENTE WEB: programa asociado con el usuario, que formula peticiones al servidor.

HIPERTEXTO: Cualquier texto disponible en el World Wide Web que contenga enlaces con otros documentos.

HTML (HiperTextMarkup Language, Lenguaje Marcado de Hipertexto): Es un lenguaje empleado para describir el interior de los documentos Web, basado en el uso de etiquetas. Permite describir hipertextos con enlaces (hiperlinks) que se conducen a otros documentos o fuentes de información relacionadas y con inserciones multimedia (gráficos, audio, video, etc.).

HTTP (HiperText Transfer Protocol, Protocolo de Transferencia de Hipertexto): Es un conjunto de normas usado para describir el modo de envío de los documentos HTML por Internet, que proporciona para que los navegadores hagan peticiones y los servidores entreguen respuestas.

INTERNET: Red global de comunicaciones que interconecta computadores y bases de datos distribuidas por todo el planeta.

IP (Internet Protocol, Protocolo de Internet): En un conjunto de normas que provee las funciones básicas de direccionamiento en Internet y en cualquier red TCP/IP (Transfer Control Protocol/Internet Protocol, Protocolo de Control de Transferencia/Protocolo de Internet). Este protocolo se encarga de poner una etiqueta con la dirección adecuada a cada paquete, ya que cada computador conectado a la red tiene una dirección de Internet única que lo distingue de cualquier otra computadora en el mundo.

JAVA: Es un lenguaje de programación orientado a objetos desarrollado por SUN Microsystems. Está diseñado para usarse en entorno distribuido de Internet.

JAVASCRIPT: Es un lenguaje de programación usado para crear programas que se ejecutan en el lado del cliente para realizar acciones dentro del ámbito de una página Web.

PÁGINA WEB: Es un documento de Internet que permite el hipertexto (permite avanzar de una página a otra enlazando el hipermedia). Presenta documentos con texto, imagen estática y en movimiento, audio, video, etc. Y utiliza el estándar HTML.

PAGINA WEB DINAMICA: Es una página Web cuyo contenido es calculado por el servidor en el momento en que el usuario accede a ella. Normalmente el contenido se obtiene desde una base de datos.

PORTABLE: La portabilidad de un software se define como su grado de dependencia de la plataforma en la que se ejecuta. La portabilidad es mayor cuanto menor es su dependencia del software de plataforma.

SCRIPT: Es una aplicación informática escrito en un lenguaje específico de programación que tiene un conjunto de instrucciones y normalmente funciona sobre otras aplicaciones que ya están en funcionamiento.

SERVIDOR WEB: Es un servidor que almacena las páginas de un sitio Web y envía páginas Web en respuesta a las peticiones HTTP hechas desde los navegadores de los clientes.

UML (Unified Modeling Language, Lenguaje de Modelamiento Unificado): Es un lenguaje gráfico para visualizar, especificar y documentar cada una de las partes que comprende el desarrollo de software. UML entrega una forma de

modelar cosas conceptuales como los procesos de negocio u funciones de sistema, además de cosas concretas como lo son escribir clases en un lenguaje determinado, esquemas de bases de datos y componentes de software reusable.

URL (Uniform Resource Locator, Localizador Uniforme de Recursos):

Cadenas de caracteres que definen la localización y el acceso a documentos de hipertexto o programas en Internet. Un URL tiene el siguiente formato: Esquema://computadora/ruta.

WWW (World Wide Web): Sistema de arquitectura Cliente/Servidor para distribución y obtención de información en Internet, basada en hipertexto e hipermedia.

WIFI: es el término usado para referirse al estándar IEEE 802.11 lanzado en 1997, y que se centra en las redes LAN (Local Area Network) sin cables o Wireless LAN (WLAN).

Permite comunicar computadores y otros dispositivos de manera inalámbrica, y enlazarlos con redes corporativas para que tengan acceso a la conexión de Internet y otros recursos como impresoras, bases de datos, archivos, etc. Tiene un radio de alcance de 90 metros, y ofrece tasas de transferencia entre 11 – 54 Mbps (Megabits por segundo), que disminuyen por efecto de la distancia.

XML: abreviatura inglesa utilizada para Extensible Markup Language, que en español significa Lenguaje de Marcado Extensible. Es una especificación multiplataforma que permite crear etiquetas personalizadas y obtener la funcionalidad que no proporciona otros lenguajes de marcado utilizados para aplicaciones Web, como HTML. Es especialmente útil en el manejo de datos.

INTRODUCCIÓN

El uso natural de sistemas de cómputo de tal manera que se permita reconocer y utilizar recursos de una manera fácil y percibirlos como objetos no diferenciados, es una exigencia en diferentes contextos, pues se requiere la integración de dispositivos alrededor de escenarios donde el ser humano se apoye en la tecnología de manera transparente.

Las tecnologías de información y comunicación son dinámicas y se actualizan rápidamente, generando nuevos requerimientos y necesidades de mayor nivel; exigiendo una adecuada utilización de los recursos y del tiempo de ejecución. Un ejemplo de esto es acceder a la información de una manera rápida, eficiencia necesaria para las nuevas tecnologías que implican el uso de dispositivos móviles o ubicuos, que hoy en día, hacen parte de la cotidianidad humana.

Teniendo en cuenta esas exigencias, es muy importante identificar los requerimientos de interacción de dispositivos móviles con arquitecturas de cómputo avanzado, para diseñar soluciones que no solo permitan la simple interconexión de los dispositivos con los recursos de cómputo avanzado, sino una interacción que permita utilizarlos de acuerdo a las necesidades.

PRESENTACIÓN DEL PROYECTO

1.1. ORIENTACIÓN SOBRE EL CONTENIDO DEL INFORME

Este documento contiene un informe sobre cada una de las etapas llevadas a cabo en el desarrollo del proyecto, planteadas en el cronograma de actividades presentado en el plan; la información se encuentra distribuida de la siguiente manera:

CAPITULO 1. Presentación del Proyecto: Se hace un análisis de los antecedentes del proyecto, definición del problema, objetivos generales, específicos, justificación, alcances y limitaciones del proyecto.

CAPITULO 2. Marco Teórico: Se presentan los conceptos utilizados en el desarrollo técnico del proyecto.

CAPITULO 3. Marco Metodológico: En este capítulo se menciona el procedimiento metodológico que se siguió para la elaboración del proyecto y las razones por la que fue elegida dicha metodología.

CAPITULO 4. Concepción y diseño de la Herramienta y Labores de Administración: se presenta el análisis de requisitos y el diseño de la herramienta.

CAPITULO 5. Análisis evaluación de rendimiento y funcionalidad del sistema: se presenta un informe de las pruebas realizadas al prototipo desarrollado y los resultados obtenidos.

CAPITULO 6. Conclusiones del trabajo realizado.

CAPITULO 7. Recomendaciones y sugerencias a tener en cuenta en la elaboración de futuros proyectos.

1.2. ANTECEDENTES DEL PROYECTO

El uso de sistemas de cómputo avanzado de tal manera que permita reconocer y utilizar recursos de una manera fácil y percibirlos como objetos no diferenciados, es una exigencia en diferentes contextos, pues se requiere la integración de dispositivos alrededor de escenarios donde el ser humano se apoye en la tecnología de manera transparente.

Las tecnologías de información y comunicación son dinámicas y se actualizan rápidamente, generando nuevos requerimientos y necesidades de mayor nivel; exigiendo una adecuada utilización de los recursos y del tiempo de ejecución. Un ejemplo de esto es acceder a la información de una manera rápida, eficiencia necesaria para las nuevas tecnologías que implican el uso de dispositivos móviles o ubicuos, que hoy en día, hacen parte de la cotidianidad humana.

En un contexto más específico, los usuarios de arquitecturas de gran escala o de cómputo avanzado³ igualmente son usuarios de dispositivos como Smartphone, tabletas digitales o simplemente, usuarios de portales que permiten la interacción con servicios web. Los usuarios exigen una transparencia en la utilización de los recursos de súper cómputo y una pervasividad o “omnipresencia” de los servicios, utilizando lo que comúnmente puede tener a su mano (como cualquiera de los dispositivos que se mencionó anteriormente) para realizar parte de sus tareas, monitorear su ejecución y obtener la información que genera.

Teniendo en cuenta esas exigencias, es muy importante identificar los requerimientos de interacción de dispositivos móviles con arquitecturas de cómputo avanzado, para diseñar soluciones que no solo permitan la simple interconexión de los dispositivos con los recursos de cómputo avanzado, sino una interacción que permita utilizarlos de acuerdo a las necesidades.

³Teniendo en cuenta que aquello que clasificamos como computo avanzado, toma como referencia la computación estándar, es decir, aquello que ofrece más rendimiento y servicios que las arquitecturas comúnmente accesible.

1.3. DEFINICIÓN DEL PROBLEMA

La necesidad de una interacción más rápida entre el usuario y la plataforma de cálculo es cada vez mayor, esto se debe a que el usuario requiere realizar un seguimiento continuo de su aplicación desde un dispositivo móvil, con el fin de suplir los requerimientos que van surgiendo en cada situación, estando desde cualquier lugar y bajo cualquier circunstancia, es por ello que la GRID UIS-2 desea facilitar la interacción más transparente y eficiente entre el usuario y la plataforma.

Los sistemas operativos móviles También conocidos como S.O. móvil hoy mundialmente conocidos, debido al desarrollo que le han dado grandes marcas tecnológicas en el mercado. A diferencia de los sistemas operativos para ordenadores, estos son más simples, y gozan de conectividad inalámbrica; entre sus características más versátiles esta la interfaz de usuario, que permite la interacción entre este y el diseño de las aplicaciones.

La brecha entre el hardware de los dispositivos móviles y el pc de escritorio, es denominada como 'Capa' o 'máquina virtual', ella facilita al usuario experto las herramientas e interfaces adecuadas para realizar sus tareas informáticas, ahorrándole trabajo y tiempo, a fin de evitarles procesos muy largos.

1.4. OBJETIVO GENERAL

Diseñar una Aplicación de interacción de dispositivos móviles con Arquitecturas de Computo Avanzado, a través de un servicio dirigido al usuario.

1.5. OBJETIVOS ESPECIFICOS

- Analizar la interacción de plataformas de cómputo de alto rendimiento con dispositivos móviles, identificando posibilidades tecnológicas.
- Definir requerimientos, que necesita una aplicación que interactúe con arquitecturas de cómputo avanzado desde dispositivos móviles.
- Proponer lineamientos de desarrollo para aplicaciones localizadas en plataformas de cómputo avanzado que requieran acceso desde dispositivos móviles.
- Desarrollar una aplicación que permita acceder a recursos específicos en infraestructura de plataformas de cómputo avanzado bajo un modelo de infraestructura como servicio.

1.6. JUSTIFICACIÓN

Los servicios que se implementan dentro de la GRIDUIS-2, con el pasar del tiempo sufren modificaciones y reestructuraciones en pro de las necesidades de sus usuarios, ya sea para los estudiantes, profesores o para los científicos. Es por eso que realizar el diseño de una aplicación para la interacción de todos los usuarios, dentro de un ambiente académico, se hace de manera amigable, en donde todos se puedan sentir miembros de una comunidad a la cual visiten con agrado y pertenencia.

La implementación de los servicios reestructurados, se realiza con el propósito de facilitar a los usuarios manejo y desarrollo de diferentes actividades de una manera sistematizada y transparente, logrando optimizar las funcionalidades que estos servicios ofrecen.

1.7. ALCANCES Y LIMITACIONES

Para analizar y diseñar la interfaz de usuario, se tuvo en cuenta los aspectos que dicta el estándar UML para la creación de proyectos, como lo son la fase de obtención de requerimientos, análisis, diseño, implementación y pruebas, empleando la menor cantidad de tiempo y recursos posibles. Considerando que en el diseño de una aplicación la principal función es la interacción con usuarios y otras clases de actores secundarios como el servidor.

Al término de la fase de implementación, e inicio de la fase de pruebas, los datos deben tener la propiedad de ser recibidos y aceptados por una aplicación desarrollada en Java, el cual procesará dicha información y retornará una serie de resultados por medio del servicio web a la plataforma, que a su vez interpretará dichos datos y los mostrará al usuario.

MARCO TEÓRICO

En el desarrollo de este proyecto se consultó una amplia bibliografía, que cubre conceptos, teorías, tanto de diseño como de gestión de datos y otros temas que son básicos para un entendimiento y desarrollo de la aplicación. Al momento de referirse a dichos temas de una manera apropiada, resultaría un documento realmente extenso.

Para iniciar un análisis sobre las interacciones de las plataformas de cómputo de alto rendimiento con dispositivos móviles, se presentara a continuación una breve reseña identificando sus posibilidades tecnológicas

2.1. GRID COMPUTING

Existen conceptos que se han desarrollado sobre que es una grid y de ellos podemos hablar del concepto elaborado por el Grid Computing Information Center, una de las asociaciones dedicada exclusivamente al desarrollo de esta tecnología, la cual define grid como un “tipo de sistema paralelo y distribuido que permite compartir, seleccionar y reunir recursos autónomos geográficamente distribuidos en forma dinámica y en un tiempo de ejecución, dependiendo de su disponibilidad, capacidad, desempeño, costo y calidad de servicio requerida por sus usuarios”⁴ esta definición, busca aprovechar la acción conjunta que surge de la cooperación entre recursos computacionales y proveerlos como servicios.

Otra definición más estructurada expuesta por Foster, Kesselman y Tuecke, precursores de la computación grid y a ellos se les atribuye el término grid, se plantea la existencia de *organizaciones virtuales (OV)* como puntos de partida de este análisis. Una *organización virtual* es “un conjunto de individuos y/o instituciones definida por reglas que controlan el modo en que comparten sus recursos”. Son organizaciones unidas para lograr objetivos comunes

⁴Computación Grid, Textos científicos, 2007: <http://www.textoscientificos.com/redes/computacion-grid/>

Sin embargo, existen varios requerimientos y problemas subyacentes tales como la necesidad de relaciones flexibles para compartir recursos, niveles de control complejos y precisos, variedad de recursos compartidos, modos de funcionamiento, calidad de servicio, etc. Las tecnologías actuales no proporcionan espacio para la variedad de recursos involucrados o no aportan la flexibilidad y control de las relaciones cooperativas necesarias para establecer las OV's.

Para analizar la interacción de plataformas de cómputo de alto rendimiento con dispositivos móviles, se propone la *grid* bajo un modelo de infraestructura como servicio, para "compartir recursos en forma coordinada y resolver problemas en organizaciones virtuales multi-institucionales de forma dinámica". De esta manera, varias instituciones pueden formar distintas OV's e incluso formar parte de más de una al mismo tiempo, realizando diferentes roles e integrando distintos recursos. La *grid* permite el intercambio, selección e integración de una amplia variedad de recursos entre los que se pueden nombrar: computadores, súper computadores, sistemas de almacenamiento, dispositivos especializados, etc. Que están generalmente distribuidos geográficamente y pertenecen a diferentes organizaciones, con el fin de resolver problemas que impliquen cómputo a gran escala y uso intensivo de datos en la ciencia, industria y el comercio. La computación *grid* fue desarrollada en ámbitos científicos a principios de los años 1990 y su entrada al mercado comercial siguiendo la idea de la llamada Utility computing supone una revolución. Los recursos que se distribuyen en la red por medio de la *grid*, se distribuyen de forma transparente pero manteniendo pautas de seguridad y políticas de gestión de carácter tanto técnico como económico. Su objetivo será compartir una serie de recursos en la red de manera uniforme, segura, transparente, eficiente, ofreciendo un único punto de acceso a un conjunto de recursos distribuidos geográficamente en diferentes dominios de administración. Estos recursos globalizados no están conectados o enlazados firmemente, es decir, no se encuentran en el mismo lugar geográfico. Las principales diferencias entre la computación *grid* y la computación clúster es la forma en que los recursos son administrados, en el caso de los clúster la

asignación de los recursos es llevada a cabo centralizadamente por un administrador de recursos, y todos estos recursos trabajan juntos cooperativamente como un solo recurso unificado en cambio, en el caso de la grid la administración tiende a ser más descentralizada y no intenta proveer una sola imagen del sistema.

2.2. COMPUTACIÓN MÓVIL.

Definición. La conceptualización de la computación móvil está en constante evolución. Desde un punto de vista tecnológico y de las muchas definiciones generadas, se pueden tomar dos que se ajustan al concepto:

- La computación móvil está asociada con la movilidad de hardware, datos y software.
- La computación móvil es una tecnología que habilita el acceso a recursos digitales en cualquier momento y desde cualquier lugar.

Estas dos definiciones, son lo suficientemente discriminatorias, para determinar cuándo una solución informática puede ser clasificada como móvil.

A continuación se presenta características de diferentes sistemas computacionales que se relacionan con conceptos de movilidad, adaptación, conectividad inalámbrica, entre otras:

SISTEMAS COMPUTACIONALES QUE SE RELACIONAN CON LA MOVILIDAD

SISTEMA COMPUTACIONAL	DEFINICION	CONCEPTOS
Computación móvil	Es el uso de un dispositivo de cómputo portable con capacidad de movilidad.	Movilidad
Computación Inalámbrica	Es el uso de un dispositivo de cómputo dentro de una red Inalámbrica, mediante enlaces basados en radiofrecuencia o infrarrojo.	Banda Ancha Seguridad Movilidad Protocolos Estándares
Computación Ubicua	Es un ambiente de cientos o miles de dispositivos de cómputo baratos y con bajo ancho de banda, que ofrecen una gran capacidad de cómputo pero que a la vez parecen invisibles	Ubicuidad
Computación Nómada o Adaptable	Es el uso de un dispositivo de cómputo con la capacidad de mudarse de un ambiente computacional a otro.	Adaptación Hot Spot
Computación Desconectada	Es el uso de un dispositivo de Cómputo que tiene la habilidad de seguir trabajando mientras esta desconectado de la infraestructura de comunicaciones.	Sincronización Caching de archivos

Tabla 1 Sistemas computacionales que se relacionan con la movilidad, la adaptación, la conectividad inalámbrica, entre otros Fuente: Autor

La funcionalidad de la computación móvil, se ha convertido en un fuerte apoyo a las características que brinda la computación tradicional, y no como se piensa en algunos sectores que viene a reemplazarla. Se habla de apoyo porque permite realizar procesos que la computación tradicional no realiza de forma óptima y más aún, si reconocemos que en algunos ambientes el factor de la movilidad juega un papel muy importante, por ejemplo es el caso del sector salud, en la toma de datos para pacientes en sitio, tal es el caso del personal que atiende en las ambulancias, por otro lado en los sectores que trabajan atención de desastres; también se puede vislumbrar en sectores académicos. es necesario establecer las diferencias entre los términos inalámbrico y móvil. Si bien es cierto, que el uso de las redes inalámbricas fortalece el potencial de las aplicaciones móviles, es falso que el solo hecho de que una aplicación haga uso de redes inalámbricas la clasifique como móvil. Y, también es falso que sin el uso de redes inalámbricas sea imposible hacer aplicaciones móviles.

A continuación se presenta una tabla donde se ejemplifican estas diferencias:

APLICACIONES DE LA COMPUTACIÓN MÓVIL E INALÁMBRICA

INALÁMBRICO	MOVIL	APLICACIÓN
NO	NO	Un PC de escritorio de hogar
NO	SI	Computador portátil utilizado en cualquier lugar sin conexión inalámbrica o sin tarjeta de expansión inalámbrica
SI	NO	PC con tarjeta de conexión wifi inalámbrica
SI	SI	Un PDA con conexión inalámbrica integrada o con una conexión inalámbrica externa

Tabla 2 Aplicaciones de la computación móvil e inalámbrica Fuente: Revista ACIS, edición no.87 noviembre 2003 - enero de 2004

En el campo de la computación móvil, han surgido diferentes avances en investigación de hardware, software y comunicaciones. La siguiente tabla presenta algunos avances en esta área y su relación con los comités de investigación del grupo de interés

ÁREAS DE INVESTIGACIÓN Y ÚLTIMOS AVANCES EN COMPUTACIÓN MÓVIL.

Campo relacionado	Área de Investigación	Avances	Comités deACM
Hardware	Tecnologías de visualización. Empaquetamiento de circuitos. Circuitos desarrollados en plástico (Poly-tronica). Circuitos de uso específico. Ahorro de energía	Anteojos, visores y dispositivos “manos libres”. Reducción del tamaño del terminal móvil. Chips, displays y baterías flexibles fabricadas en plástico a bajo costo. Dispositivos más livianos y ergonómicos. Mayor duración de la batería.	Otras Organizaciones como IEEE.
Software	<ul style="list-style-type: none"> · Multimedia en Dispositivos inalámbricos. · Adaptación en Código Móvil. · Modelamiento de la movilidad. · Nuevas arquitecturas de desarrollo para 	<ul style="list-style-type: none"> · Aplicaciones en varios sectores productivos. · MMS, video streaming en PDAs. · Comercio electrónico móvil. · Sistemas basados 	Wireless Mobile Multimedia (WMoM). · Wireless Sensor Networks and Applications (WSNA).

	dispositivos móviles. · Manejo de bases de datos con características como sincronización	en localización. · Agentes móviles · Plataformas de desarrollo abiertas como J2ME. · Plataformas propietarias como .Net, Visual Age ME, Oracle 9i Lite, CodeWarrior etc.	· Modeling Analysis and Simulation of Wireless and Mobile Systems (MSWiM). · Mobile Commerce (MCommerce).
Comunicaciones	WirelessLANs. · Redes Móviles y PCS. · Nuevos protocolos de comunicaciones. · Redes Ad-hoc. · Seguridad Wireless.	· IEEE 802.11 a,b,g. · Wireless ATM. · UMTS, GPRS. · CDMA 1x, CDMA 2000. · Mobile IPv6, Mobile. IPv4, IPv6, TCP Wireless y otros.	Wireless Security (WiSE). · Discrete Algorithms and Methods for Mobile Computing (Dial – M). · Otras organizaciones: IEEE, IETF.

Tabla 3 Algunos de las áreas de investigación y últimos avances en Computación Móvil. Fuente: Autor

2.3. COMPUTACIÓN MÓVIL EN COLOMBIA.

En los últimos años, la computación móvil, ha llegado al mercado en diferentes formas. Los dispositivos no sólo ayudan en la administración de citas y contactos, sino también proporcionan una herramienta para reemplazar los procesos de negocios basados en papelería con aplicaciones basadas en formas. La eficiencia y precisión aumentada para capturar rápidamente los datos⁵ en un dispositivo de computación puede dar como resultado mayor productividad del empleado, un reporte del negocio más rápido para la toma de decisiones, y costos operacionales reducidos.

En nuestro país el auge de la computación móvil se ha visto principalmente en el sector corporativo, sin importar el tamaño o la actividad económica de la empresa. Varias compañías han optimizado su operación de ventas y distribución, gracias a las aplicaciones que les brindan estos equipos. Entre las empresas que tienen herramientas que se ejecutan sobre estos dispositivos están Wella, IMUSA, Unilever, Nestlé, Alpina y Seguros MAPFRE.

Colombia es el tercer mercado latinoamericano más evolucionado en aplicaciones móviles después de Brasil y México, gracias a un grupo de empresas que han realizado desarrollo con gran madurez en el tema.

Los dispositivos móviles permiten a una organización extender su infraestructura de computación tradicional más allá de los alcances de su negocio, a su fuerza de trabajo móvil de maneras muy importantes como: administración de información personal, documentos, aplicaciones de línea de negocios, servicios financieros, servicios de salud, hotelería y turismo, industria manufacturera y venta al por menor y distribución.

⁵MARTINEZ, Richard. Cómo automatizar las ventas. Disponible en: <http://www.multitel.com.co/noticiadestacada40.html>

2.4. SISTEMA OPERATIVOS MOVILES

También son conocidos como S.O. móvil (Sistema Operativo Móvil), hoy mundialmente populares gracias al desarrollo que le han dado las grandes marcas tecnológicas en el mercado. A diferencia de los sistemas operativos para ordenadores, estos son más simples, gozan de conectividad inalámbrica; entre sus características más versátiles se halla la interfaz de usuario que permite la interacción entre este y el diseño de las aplicaciones..

Los sistemas operativos de los ordenadores personales o computadores de escritorio, así como controlan estos aparatos electrónicos, en el caso de los S.O móviles la diferencia radica en su orientación a la "conectividad inalámbrica", los cuales a través de diversas funcionalidades y aplicaciones permiten la usabilidad de los dispositivos.

La amplia, pero poco conocida brecha entre el hardware de los dispositivos móviles y el usuario, denominada como 'Capa' o 'máquina virtual', facilita al cibernauta básico o experto las herramientas e interfaces adecuadas para realizar sus tareas informáticas, ahorrándole trabajo y tiempo, a fin de evitarles complicados procesos.

La 'Capa' o 'máquina virtual' del Sistema Operativo Móvil está integrada de los siguientes elementos:

- **Kernel ó Núcleo.** Es el encargado de dar paso seguro a los distintos programas de acceso al hardware, así como la gestión de desarrollar procesos en el sistema de archivos y la memoria.
- **El middleware.** Conjunto de módulos que integra las diversas aplicaciones que hacen parte de los dispositivos móviles como el sistema de mensajería y comunicaciones, códec multimedia, etc.
- **Entorno de ejecución de aplicaciones.** Este componente es un gestor de aplicaciones e interfaces programables que facilitan la creación de software.

- **Interfaz de usuario.** Este elemento no es más que el medio o instrumento por el cual el usuario se comunica y relaciona con el dispositivo móvil, la cual incluye los menús, teclado, aplicaciones, listas, botones, gráficos, etc.

Estos son sistemas operativos más conocidos en el mercado.

- **Palm OS.** Diseñado por la hoy conocida Access SystemsAmericas para las agendas electrónicas o PDAs. Las aplicaciones de este sistema operativo móvil incluye libreta de direcciones, calculadora, calendario, gastos, libreta de notas, tareas y notas. Conectividad para infrarrojos y bluetooth.
- **Symbian.** Fue fabricado luego de la alianza de varias empresas del sector como Nokia, Samsung, Sony Ericsson, LG, Motorola, Lenovo, entre otras. Actualmente, su incidencia en el mercado es del 30.6%
- **Windows Phone.** Desarrollado por Microsoft, el cual ha desarrollado 8 versiones. El más reciente es el Windows Phone 8.
- **iOS.** Conocido anteriormente como iPhone OS, este S.O móvil es de Apple el cual inicialmente fue diseñado para el iPhone, pero posteriormente fue aplicado también para los demás dispositivos móviles de la compañía. Su interfaz de usuario se distingue por usar pantalla multitouch, elementos deslizadores, interruptores y botones. En el mercado tiene una penetración del 16%.
- **Android.** Sistema Operativo móvil desarrollado por Google para smartphones, tabletas, portátiles, netbooks, Google TV, relojes de pulseras, auriculares y demás dispositivos. Cada una de las versiones de Android recibe el nombre de un postre en inglés. Actualmente, existen más de 400 mil aplicaciones de este sistema operativo.
- **Blackberry OS.** Desarrollado por Research In Motion para BlackBerry. Su sistema permite la realización de multitareas, servicio de mensajería instantánea a través del sistema de PIN. De acuerdo a los fabricantes este sistema operativo móvil está orientado para el desarrollo de tareas profesionales que permiten sincronizar el dispositivo con agenda, correos electrónicos, calendario y contactos.

2.5. SISTEMA OPERATIVO ANDROID

Android es un sistema operativo orientado a dispositivos móviles, basado en una versión modificada del núcleo Linux. Inicialmente fue desarrollado por Android Inc., una pequeña empresa, que posteriormente fue comprada por Google; en la actualidad lo desarrollan los miembros de la Open Handset Alliance (liderada por Google).

Su presentación se realizó el 5 de noviembre de 2007 junto con la fundación Open Handset Alliance, en un consorcio de numerosas compañías de hardware, software y telecomunicaciones comprometidas con la promoción de estándares abiertos para dispositivos móviles.

Android consiste en un sistema abierto, multitarea, que permite a los desarrolladores acceder a las funcionalidades principales del dispositivo mediante aplicaciones, cualquier aplicación puede ser reemplazada libremente, además desarrollarlas por terceros, a través de herramientas proporcionadas por Google, y mediante los lenguajes de programación Java y C.

El código fuente de Android está disponible bajo diversas licencias de software libre y código abierto, Google liberó la mayoría del código de Android bajo la licencia Apache. Todo esto permite que un desarrollador no solo pueda modificar su código sino también mejorarlo. A través de esas mejoras puede publicar el nuevo código y con el ayudar a mejorar el sistema operativo para futuras versiones.

Android depende de un Linux versión 2.6 para los servicios base del sistema como seguridad, gestión de memoria, gestión de procesos, stack de red, y modelo de drivers. El núcleo también actúa como una capa de abstracción entre el hardware y lo demás del stack de software.

Otras características que presenta android es un navegador web integrado basado en el motor WebKit, soporte para gráfico 2D y 3D basado en la especificación OpenGL, soporte multimedia para audio, video e imágenes en varios formatos, conectividad Bluetooth, EDGE, 3G, Wifi, entre otros.

A lo largo de estos años, se han lanzado diversas versiones de Android, todas presentan curiosamente nombres de postres: versión 1.5-Cupcake (Magdalena), v1.6-Donut (Dona), v2.0/v2.1-Eclair (Relámpago o Pepito, un postre francés), v2.2-Froyo (acrónimo para “Frozen Yogurt”, Yogur congelado), v2.3-Gingerbread (Pan de Jengibre) y v3.0/v3.1 Honeycomb (Panal de miel).

Cabe resaltar, que la libertad del código de Android ha logrado que en poco tiempo se implante en multitud de dispositivos electrónicos, desde móviles hasta computadoras portátiles, netbooks, microondas, lavadoras, marcos digitales, navegadores GPS, relojes e incluso en navegadores de abordo de coches.

Esto convierte a Android en un sistema operativo multifuncional, que garantiza cada vez más su crecimiento y expansión en el mercado y fabricación tecnológica.

2.6. SELECCIÓN DE SISTEMA OPERATIVO PARA EL DISEÑO DE LA APLICACIÓN.

Al iniciar con el diseño del prototipo fue necesario seleccionar que sistema operativo se utilizaría para la aplicación y se estudiaron los casos de Android, BlackBerry y IOS de iPhone.

Diseñar una aplicación para iPhone no permite realizar pruebas si no se ha pagado previamente a Apple por el registro en el programa de desarrolladores. El costo es de 90\$ dólares para utilizarlo por un año, con ese pago se proporciona perfiles de despliegue para que realice pruebas en un teléfono a la aplicación, solo se podrá probarlo con el simulador en el ordenador y, desde luego, no es lo mismo. Siempre es necesario ver la aplicación funcionando en el teléfono real para poder detectar problemas.

Primero hay que comprender bien el proceso, saber para qué sirve cada certificado y cuidar que se introducen en el iTunes Connect, se tendrá que poner imágenes para iTunes, imágenes de volcados de pantalla y calificar la aplicación según su contenido, calificación de edad y luego se envía el binario. Eso se realiza desde el Xcode luego se procede a la espera.

La primera vez que sea enviada la aplicación tardará una semana en asignar un revisor y casi 4 días en dar el ok a la aplicación.

Proceso muy similar hay que desarrollar si se desea crear la aplicación para dispositivos BlackBerry.

Por esa razón se decidió generar el desarrollo del diseño del prototipo en Android cuyas características ya fueron explicadas anteriormente.

2.7. CÓDIGO DEL LADO DEL CLIENTE (CLIENTSIDE SCRIPTS).

Hace referencia al código que se ejecutan en los navegadores que las computadoras clientes tienen instalados. Estos códigos, no hacen necesario que el servidor Web cumpla determinados requisitos. Las tecnologías más comunes de este tipo son:

- **JavaScript:** es un lenguaje de programación interpretado, es decir, que no requiere compilación, utilizado principalmente en páginas Web, con una sintaxis semejante a la del lenguaje Java y el lenguaje C. Permite la creación de ventanas, mostrar y cambiar texto e imágenes en movimiento, validar entradas de un determinado formulario antes de enviarlo al servidor.
- **Java Applets:** Desarrollado por Sun Microsystems. Los applets son programas escritos en lenguaje de programación Java, se incrustan en el código fuente de la página Web y se ejecutan en el navegador del cliente gracias a la Máquina Virtual de Java (Java Virtual Machine, JVM) que éste lleva incorporado. Pueden lograr efectos para el texto, sonido e imágenes.

2.8. CÓDIGO DEL LADO DEL SERVIDOR (SERVER SIDE SCRIPTS).

Estos códigos se ejecutan en el servidor. Para su funcionamiento, el programa se ejecutará en el servidor con los datos o peticiones que el usuario envía desde su navegador y el servidor muestra los resultados del programa en una página HTML que el usuario verá normalmente en su navegador. Los más usados son:

- **ASP (Active Server Pages).** Es altamente utilizado en la gestión de Bases de Datos ya que puede conectarse a SQL, Access, Oracle u otras. Requiere de una computadora configurada como Servidor Web de Microsoft (Microsoft Web Server), en este caso, el navegador del cliente es indiferente pues el trabajo se realiza del lado del Servidor.

- PHP. Es un lenguaje similar al usado en la tecnología ASP de código abierto (Open Source) y gratuito. Su gran potencia se encuentra en la interacción con los motores de bases de datos más usados: Oracle, Sybase, MySQL.
- JSP (Java Server Pages). Es una tecnología que permite la generación dinámica de páginas Web combinando código JAVA (scriptlets) con un lenguaje marcado como HTML o XML.

2.9. TECNOLOGÍA UTILIZADA.

La tecnología usada para el diseño del sitio Web es PHP, los nuevos servicios son desarrollados con esta tecnología ya que permite desarrollar aplicaciones independientes de la plataforma y portables a otros sistemas operativos y servidores Web.

Las páginas PHP y servlets se ejecutan en una máquina virtual de Java, lo cual permite que se puedan usar en cualquier tipo de computadora, siempre que exista una máquina virtual de Java para ella. Cada PHP se ejecuta en su propio contexto (llamado también hilo o hebra); sin embargo no se ejecuta cada vez que recibe una petición, sino que persiste de una petición a la siguiente, de forma que no se pierde tiempo en invocarlo (cargar programa e interpretarlo). Su persistencia le permite también hacer una serie de actividades de forma más eficiente: conexión a bases de datos y manejo de sesiones.

MODELO DE ACCESO A PHP

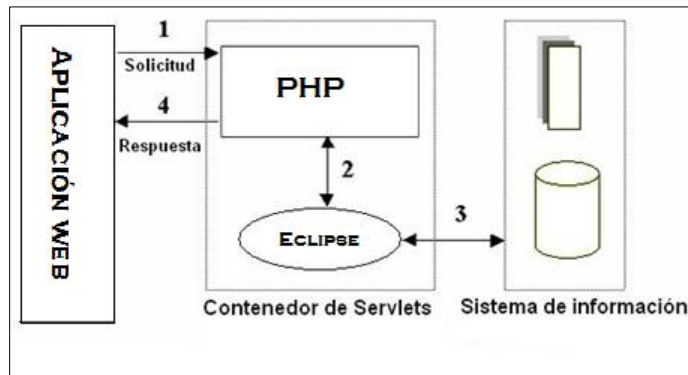


Figura 1 MODELO DE ACCESO A PHP Fuente: Autor

Un usuario desde un navegador Web cliente hace una petición que es enviada a un archivo PHP. Este archivo accede a componentes del servidor que generan contenido dinámico y lo presentan en el navegador.

Después de recibir la petición del cliente, el archivo PHP pide información de ECLIPSE si es necesario. ECLIPSE puede, en turnos, pedir información de otro PHP o de una base de datos. Una vez el ECLIPSE genera el contenido, el archivo PHP puede consultar y presentar el contenido al navegador.

La primera vez que un archivo PHP es solicitado, este es compilado en un objeto. La respuesta del objeto es HTML, el cual es interpretado por el navegador para ser presentado al usuario. Después de la compilación, el objeto de la página compilada es almacenado en la memoria principal de la computadora con configuración de servidor. En las peticiones posteriores a esta página, el servidor revisa si el archivo PHP ha cambiado. Si no ha cambiado, el servidor utiliza el objeto de la página compilada guardado en memoria para generar la respuesta al cliente, en caso contrario el servidor automáticamente compila el archivo de la página y procede a reemplazar el objeto en la memoria.

2.10. BASES DE DATOS

Una base de datos es un conjunto de datos relacionados entre sí, que pertenecen a un mismo contexto y que son almacenados porque se consideran necesarios para una determinada organización o negocio.

2.10.1 Modelos de Bases de Datos.

Las bases de datos se pueden clasificar de acuerdo a su modelo de administración de datos. Algunos modelos con frecuencia utilizados en las bases de datos son:

- Bases de Datos Jerárquicas.

Son bases de datos que almacenan de una manera similar a un árbol (invertido), en donde un nodo padre de información puede tener varios hijos. El nodo que no tiene padres es llamado raíz, y a los nodos que no tienen hijos se los conoce como hojas.

- Base de Datos de Red.

En este modelo se permite que un mismo nodo tenga varios padres. Ofrece una solución eficiente al problema de redundancia de datos; sin embargo, la dificultad para administrar los datos en una base de datos de red ha conllevado a que sea un modelo usado más por programadores que por usuarios finales.

- Base de Datos Relacional.

Es el modelo más utilizado en la actualidad para modelar problemas reales y administrar datos dinámicamente. Su principal idea es el uso de "relaciones". Estas relaciones podrían considerarse en forma lógica como conjuntos de datos, también llamados tuplas. Cada relación es una tabla que está compuesta por registros (las filas de una tabla), que representan las tuplas, y campos (las columnas de una tabla). Los datos pueden ser recuperados o almacenados

mediante "consultas" que ofrecen una amplia flexibilidad para administrar la información.

El lenguaje más habitual para construir las consultas a bases de datos relacionales es el Lenguaje Estructurado de Consultas (StructuredQueryLanguage, SQL), un estándar implementado por los principales manejadores de bases de datos relacionales.

- Acceso a Base de Datos.

Para desarrollar aplicaciones que conecten bases de datos, se utilizan interfaces y programas estándar que envían demandas escritas en SQL, y procesan los resultados. Para conectarse a un motor de bases de datos determinado, se necesita una interfaz estándar o controlador (en inglés: driver) que es intermediario entre la aplicación y la base de datos.

- **Conectores más Utilizados**
- **ODBC.** Es un programa de interfaz de aplicaciones (API) para acceder a datos en sistemas manejadores de bases de datos tanto relacionales como no relacionales, utilizando para ello el lenguaje de consulta estructurado (SQL). Se administran a través de la ventana ODBC del Panel de Control, En computadoras con sistema operativo Microsoft Windows.
- **MDB.** Servidor de bases de datos casi profesional. Esta aplicación permite trabajar con tablas de base de datos creadas en Microsoft Access 97/2000. Es posible abrir tablas en Lenguaje de consulta estructurado, visualizarlas, navegar, crear y borrar índices, fijar relaciones, copiar, etc.
- **JDBC.** La conectividad de bases de datos Java (Java DatabaseConnectivity, JDBC) es una especificación de la interfaz de aplicación de programa (ApplicationProgramming Interface, API) para conectar los programas escritos en Java a los datos en bases de datos de mayor uso.

Los servlets y las páginas PHP usan JDBC prácticamente de la misma manera que cualquier otra aplicación en Java, típicamente los datos del controlador JDBC, la cadena de conexión, el nombre de usuario y la contraseña para conectarse a la base de datos son codificados dentro del programa.

Las operaciones básicas realizadas durante la ejecución de un controlador JDBC son:

- Cargar un controlador JDBC.
- Utilizar ese controlador para abrir una conexión con la base de datos.
- Emitir instrucciones SQL a través de la conexión.
- Procesar los conjuntos de resultados devueltos por las operaciones SQL.

Manejadores o Gestores de Bases de Datos.

Son un tipo de software específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan, para almacenar y posteriormente acceder a los datos de forma rápida y estructurada.

Las funciones principales de un gestor de bases de datos (DataBase Manager System, DBMS) son:

- Crear y organizar la Base de datos.
- Establecer y mantener las trayectorias de acceso a la base de datos de tal forma que los datos se puedan acceder rápidamente.
- Manejar los datos de acuerdo a las peticiones de los usuarios.
- Registrar el uso de las bases de datos.
- Interacción con el manejador de archivos. Esto a través de las sentencias en Lenguaje Manipulador de Datos (Data Manipulation Language, DML) al comando del sistema de archivos. Así el Manejador de base de datos es el responsable del verdadero almacenamiento de los datos.

- Respaldo y recuperación. Consiste en contar con mecanismos implantados que permitan la recuperación fácilmente de los datos en caso de ocurrir fallas en el sistema de base de datos.
- Control de concurrencia. Consiste en controlar la interacción entre los usuarios concurrentes para no afectar la inconsistencia de los datos.

El DBMS interpreta las peticiones de entrada-salida del usuario y son enviadas al sistema operativo para la transferencia de datos entre la unidad de memoria secundaria y la memoria principal.

Un sistema manejador de base de datos es el cerebro de la base de datos ya que se encarga del control total de los posibles aspectos que la puedan afectar.

Existen diferentes manejadores de bases de datos como MySQL, ORACLE, FoxPro, Microsoft Access y PowerBuilder.

Para el desarrollo de la aplicación web se utilizara MySQL.

2.11. MySQL.

MySQL es el Manejador de base de datos más usado y estandarizado para acceder a bases de datos relacionales en la plataforma UNIX. Es rápido y eficiente, Sus principales características son:

- Consume pocos recursos tanto de procesador como de memoria principal en una computadora.
- Su principal objetivo de diseño fue la velocidad.
- Tiene gran disponibilidad en varias plataformas y sistemas.
- Soporta gran cantidad de datos.
- Es de código abierto, puede ser usado y modificado.

Ventajas de MySQL

- Es posible manipular bases de datos enormes.
- Permite conexiones entre diferentes máquinas con distintos sistemas operativos. Es normal que servidores Linux o Unix, usando MySQL, sirvan datos para computadoras con otros sistemas operativos.
- Permite manejar multitud de tipos para columnas.
- Permite manejar registros de longitud fija o variable.
- Acceso a las bases de datos de forma simultánea por varios usuarios y/o aplicaciones.
- Seguridad, en forma de permisos y privilegios, determinados usuarios tienen permiso para consulta o modificación de determinadas tablas.
- Potencia: SQL es un lenguaje potente para consulta de bases de datos, usar un motor ahorra mucho trabajo.
- Portabilidad: SQL es también un lenguaje estandarizado, de modo que las consultas hechas usando SQL pueden hacerse fácilmente en otros sistemas y plataformas.

3. MARCO METODOLOGICO

LoginGRid hace parte del diseño de una aplicación que tiene como objetivo la implementación de servicios, se encuentra ubicado en la capa de aplicación, La función principal es un desarrollo amigable con el usuario, que sea eficiente y transparente.

Se implementará el diseño de la aplicación con una metodología tipo programación extrema (XP),⁶ está se encuentra estructurada en la metodología de programación extrema (XP), la cual es una metodología ágil, que maneja un diseño incremental en el desarrollo del software. Se seleccionó esta metodología por la comunicación continua que mantiene entre el usuario y la aplicación en desarrollo; y la permanente realización de pruebas que se realizan, ya que de esta manera se mantiene un código claro y sencillo.

La Programación Extrema es un conjunto coherente de valores, principios y prácticas para abordar problemas de desarrollo de software

En la programación extrema un ciclo corresponde a una o muchas iteraciones y una iteración constan de los siguientes pasos:

1. ANÁLISIS TEÓRICO Y CONTEXTUALIZACIÓN: En esta parte se analiza las necesidades del usuario final; que son usuarios de arquitecturas de gran escala.
2. Diseño: Se diseñan tipos de pruebas que permitan evaluar el funcionamiento de la aplicación.
3. Implementación: Generación de código fuente
4. VALIDACIÓN EXPERIMENTAL Generación de código fuente.
5. Aplicación de prueba: Se ejecutan las pruebas a la aplicación ya elaborada.
6. DIVULGACIÓN DE LOS RESULTADOS: Se analizan los resultados obtenidos de las pruebas, y junto con el usuario se evalúa el cumplimiento en el desarrollo de la aplicación.

Después de terminada la iteración si las pruebas no cumplen las expectativas, se vuelve a comenzar una nueva iteración.

⁶ Ver. Beck, Extreme Programming Explained. p.85

DIAGRAMA DE METODOLOGÍA XP

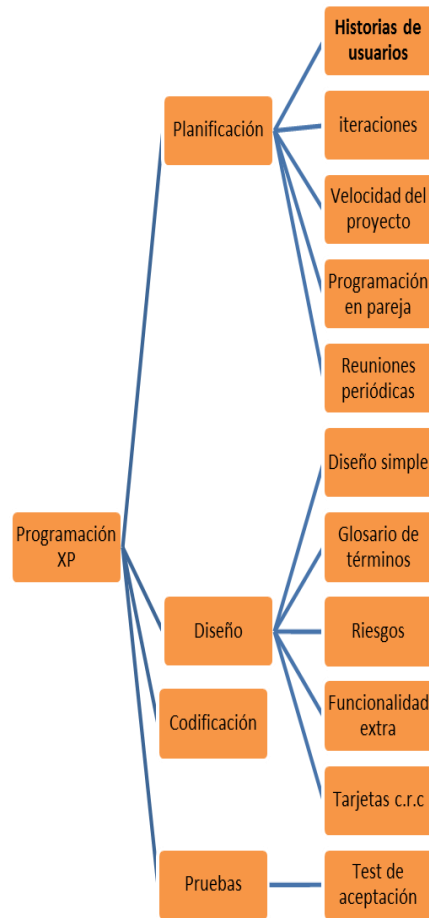


FIGURA 2. DIAGRAMA DE METODOLOGÍA XP Fuente: Autor

4. CONCEPCIÓN Y DISEÑO DE LA HERRAMIENTA Y LABORES DE ADMINISTRACIÓN.

4.1 DEFINICION UML

El lenguaje unificado de modelado o UML (Unified Modeling

Language) es la notación, principalmente gráfica que se valen los métodos de desarrollo para expresar los diseños. UML fue desarrollado por Grady Booch, Jim Rumbaugh e Ivar Jacobson, como una filosofía de diseño y análisis orientado a objetos que permitiera modelar y documentar sistemas de información y procesos empresariales. De esta manera, y con el apoyo del Object Management Group (OMG) se convierte en la notación estándar para la descripción de métodos software. Entre sus características se encuentran:

- Es un lenguaje de modelado visual que se usa para especificar, visualizar, construir y documentar los componentes de un sistema. Se emplea para entender, diseñar, configurar, mantener y controlar la información sobre los sistemas a construir.
- No es un lenguaje de programación y es independiente de cualquiera de ellos.
- Es un lenguaje de modelado de propósito general que pueden usar todos los modeladores. No tiene propietario y está basado en el común acuerdo de gran parte de la comunidad informática. Puede ser aplicado a diferentes tipos de sistemas (software y no-software), dominios (negocios Vs. Software) y métodos o procesos.
- Se ha convertido en el estándar de facto para definir, organizar y visualizar los elementos que configuran la arquitectura de una aplicación orientada a objetos. Entre los diagramas UML que se han utilizado en el desarrollo del proyecto están:

Diagramas de actividades: Modela el funcionamiento del sistema y el flujo de control entre los objetos.

Diagramas de casos de uso: Estos diagramas permiten definir cómo se debe comportar una parte del sistema y muestran la interacción entre los casos de uso y los actores

(Clientes o Constructores). Un caso de uso especifica un requerimiento funcional, es decir, enumera qué debe hacer cierta parte ante determinado suceso.

4.2 USUARIOS DE LA PLATAFORMA LOGINGRID:

Usuario General: Debido a que la GRIDUIS no maneja información Neurálgica interna del grupo, solo hay un único tipo de usuario del sistema

Servidor de Aplicaciones: Dispositivo que contiene implementado los servicios que van a ser de uso para la aplicación de lado del cliente

Desarrollador: Rol que ocupa el administrador de los contenidos de la aplicación; también encargado de implementar lo que el arquitecto considere necesario.

Arquitecto: Es el rol que ocupa el ingeniero de sistemas que encabeza el diseño del sistema como tal.

En la relación de usuarios de la plataforma están contemplados los tipos de usuarios que intervienen directamente en el desarrollo de la aplicación, el actor que interactúa con el sistema no tiene alguna distinción especial por parte del sistema, es decir cualquier actor que interactúe con el sistema se denomina Usuario General.

4.3 CASO DE USO GLOBAL:

Representa el diagrama de casos de uso global, el cual describe de manera general las funcionalidades características de la aplicación. El usuario selecciona un nodo en la interfaz y después de enviar la información, podrá ver la información que solicito

DIAGRAMA DE CASOS DE USO DISEÑO GLOBAL

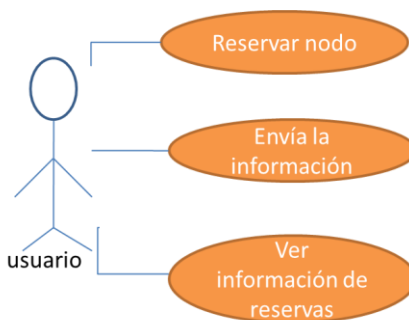


FIGURA 3. DIAGRAMA DE CASOS DE USO DISEÑO GLOBAL Fuente: Autor

Los incrementos definidos están asociados a un módulo específico. En cada incremento se realizó un diseño detallado.

4.4 MÓDULO DISEÑO INTERFAZ DEL SERVIDOR.

Se desarrolla el módulo diseño Interfaz del servidor, en el cual el usuario en Internet, puede seleccionar de una lista de clústers el nodo del cual desea conocer información.

4.4.1 DIAGRAMA DE CASOS DE USO DISEÑO INTERFAZ DEL SERVIDOR.

En los diagramas de casos de uso de este módulo se describen las funciones del usuario que ingresa a la interfaz del servidor.

Al navegar por la interfaz Web, el usuario podrá registrarse y enviar la solicitud al servidor. Posteriormente el servidor devuelve la respuesta de la solicitud y el usuario podrá utilizar la reserva en el nodo que solicito.

4.4.2 DIAGRAMA DE CASOS DE USO DISEÑO INTERFAZ DEL SERVIDOR.

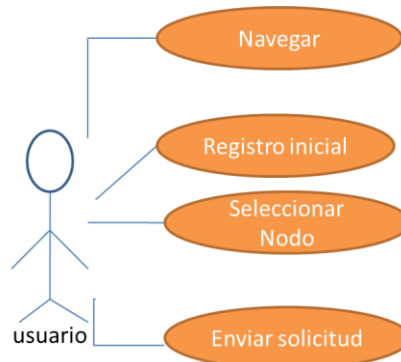


FIGURA 4. DIAGRAMA DE CASOS DE USO DISEÑO INTERFAZ DEL SERVIDOR Fuente: Autor

4.4.3 DIAGRAMA DE ACTIVIDADES PARA EL CASO DE USO ENVIAR SOLICITUD.

Como se observa en la figura 4, cuando el usuario envía la solicitud al haber seleccionado un nodo, el servidor realiza las consultas a la base de datos, genera los archivos XML correspondientes a el(los) nodos(s) seleccionados. Finalmente se le presenta al usuario un cuadro de dialogo para confirmar el envío de la información

DIAGRAMA DE ACTIVIDADES PARA EL CASO DE USO ENVIAR SOLICITUD

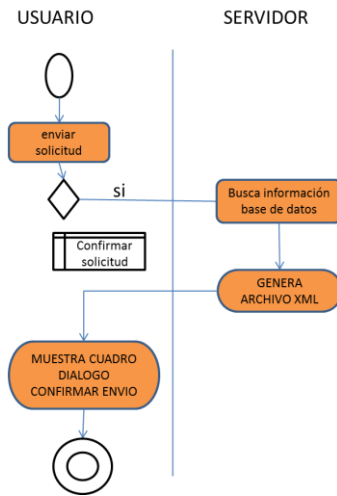


FIGURA 5. DIAGRAMA DE ACTIVIDADES PARA EL CASO DE USO ENVIAR SOLICITUD Fuente: Autor

4.4.4 DIAGRAMA DE ACTIVIDADES PARA EL CASO DE USO SELECCIONAR NODOS.

Cuando el usuario selecciona un nodo, se le notifica si este ya había sido seleccionado. En el caso de no haber sido seleccionado, el nodo se reserva satisfactoriamente.

DIAGRAMA DE ACTIVIDADES PARA EL CASO DE USO SELECCIONAR NODOS

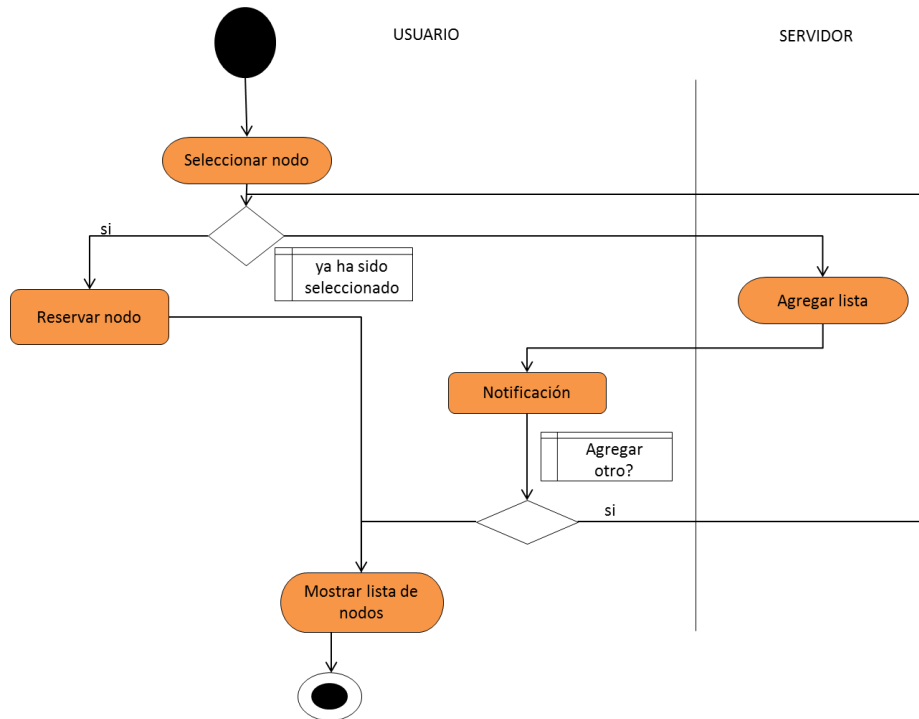


FIGURA 6. DIAGRAMA DE ACTIVIDADES PARA EL CASO DE USO SELECCIONAR NODO Fuente: Autor

4.5 MÓDULO DISEÑO DE SINCRONIZACIÓN.

En el desarrollo del módulo de diseño de sincronización, el usuario envía la información que previamente descarga del pc a la aplicación del dispositivo, además se permite la eliminación de información y el reconocimiento de las características del dispositivo: procesador, sistema operativo, batería y memoria disponible.

4.5.1 DIAGRAMA DE CASO DE USO DEL MÓDULO DISEÑO DE SINCRONIZACIÓN.

En el diagrama de caso de uso de este módulo se describe la función del usuario que utiliza la aplicación de sincronización.

DIAGRAMA DE CASO DE USO DEL MÓDULO DISEÑO DE SINCRONIZACIÓN

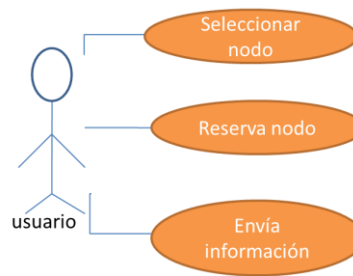


FIGURA 7. DIAGRAMA DE CASOS DE USO DE SINCRONIZACIÓN. Fuente: Autor

1.1.1 DIAGRAMA DE ACTIVIDADES PARA EL CASO DE USO ENVIAR INFORMACIÓN.

Cuando la aplicación detecta que el dispositivo está conectado copia los archivos XML y las imágenes correspondientes a la carpeta de datos en la aplicación del dispositivo móvil y muestra una lista con los archivos transferidos.

DIAGRAMA DE ACTIVIDADES PARA EL CASO DE USO ENVIAR INFORMACIÓN

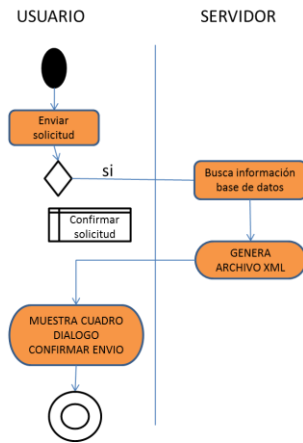


FIGURA 8. DIAGRAMA DE ACTIVIDADES PARA EL CASO DE USO ENVIAR INFORMACIÓN Fuente: Autor

4.6 Módulo aplicación para el dispositivo

El diagrama de caso de uso de este módulo, describen las funciones del usuario, el cual al seleccionar un nodo determina el tipo de información a consultar: información general, podrá escoger entre diferentes nodos.

Usuarios del sistema

ACTOR	Usuario general
CASO DE USO	Registro, Usuario, Reservas
TIPO	Primario
DESCRIPCIÓN	Cualquier actor que interactúe con el sistema.

Tabla 4 Usuarios del sistema Fuente: Autor

4.6.1 MODELO DE CASOS DE USOS

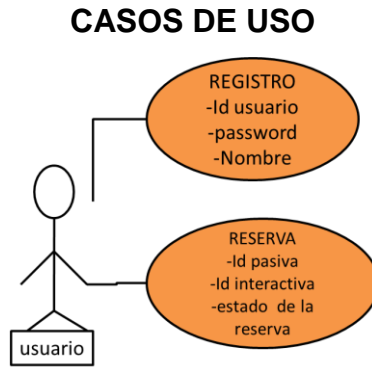


FIGURA 9. CASOS DE USO.Fuente: Autor

4.6.2 MODELO CASO DE USO REGISTRO

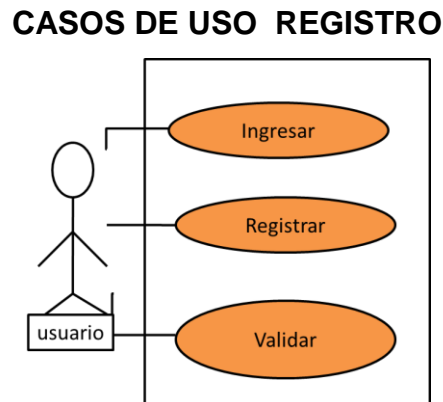


FIGURA 10.CASOS DE USO REGISTRO.Fuente: Autor

CASO DE USO REGISTRO

CASO DE USO	Registro
ACTOR	Usuario General
TIPO	Generalización.
PROPOSITO	Gestiona y valida los datos necesarios que deben ser enviados al servidor de aplicaciones.
RESUMEN	Ubicado en la capa de aplicaciones del modelo general del

	sistema GRIDUIS, encargado de administrar la información que fluye entre el usuario y las herramientas hardware.
PRECONDICIONES	Acceder a la plataforma del grupo de Investigación GRIDUIS por medio de la aplicación.
FLUJO PRINCIPAL	Se presenta al usuario la pantalla Simulación. El usuario puede seleccionar entre las siguientes actividades: ingresar, registrarse, reservar nodo.
SUBFLUJOS	Ninguno.
EXCEPCIONES	Ninguno.

Tabla 5 Caso de uso REGISTRO. Fuente: Autor

4.6.3 MODELO CASO DE USO RESERVA

CASOS DE USO RESERVA

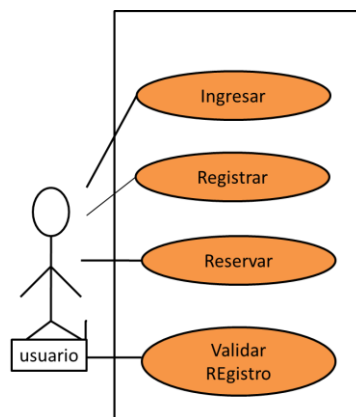


FIGURA 11. CASOS DE USO RESERVA Fuente: Autor

CASO DE USO RESERVA

CASO DE USO	Reserva
ACTOR	Usuario General
TIPO	Generalización.
PROPOSITO	Gestiona y valida los datos necesarios que deben ser enviados al servidor de aplicaciones.
RESUMEN	Ubicado en la capa de aplicaciones del modelo general del

	sistema GRIDUIS, encargado de administrar la información que fluye entre el usuario y las herramientas hardware.
PRECONDICIONES	Acceder a la plataforma del grupo de Investigación GRIDUIS por medio de la aplicación.
FLUJO PRINCIPAL	Se presenta al usuario la pantalla Simulación. El usuario puede seleccionar entre las siguientes actividades: ingresar, registrarse, reservar nodo.
SUBFLUJOS	Ninguno.
EXCEPCIONES	Ninguno.

Tabla 6Caso de uso RESERVA Fuente: Autor

4.7 DIAGRAMA DE PAQUETES

El diagrama de paquetes UML del sistema, muestra la relación entre las agrupaciones lógicas que fueron implementadas, adicionalmente muestra la versatilidad para integrar nuevos módulos al sistema, representando una sección que será agregada a la plataforma posteriormente en la fase de desarrollo

DIAGRAMA DE NUEVO MODULO

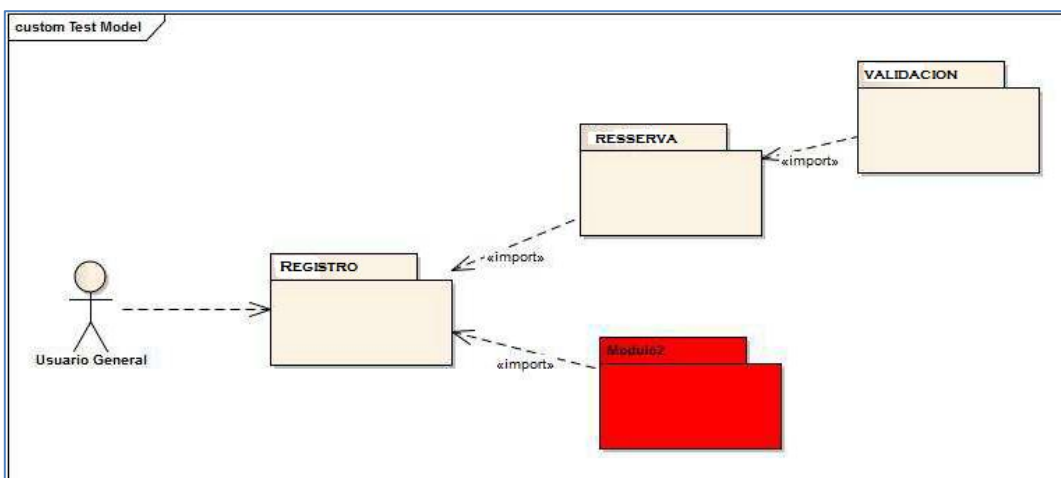


FIGURA 12. DIAGRAMA DE NUEVO MODULO Fuente: Autor

4.8 DIAGRAMA DE COMPONENTES.

Entre la documentación generada, se encuentra la representación del sistema mediante diagramas de componentes, encargados de mostrar la conformación del software, componentes y dependencias entre estas secciones. Como se ha mencionado a lo largo del documento, los componentes se relacionan entre sí y con otros componentes, y se observa que antes de llegar a la estación de trabajo, el proceso de manejo y transformación de datos está representado de manera esquemática de tal manera que si es necesario modificar algún componente, se pueda realizar dicha acción sin ningún reparo, siendo necesario solamente configurar los nombres o direcciones adecuadas para dicha actividad.

El diagrama de componentes que modela la estructura del sistema, muestra como está diseñada la aplicación, esta representa el dispositivo Estación de trabajo que opera sobre el explorador web e interactúa con el servidor de aplicaciones por medio de protocolos de transferencia de datos http, donde se localiza la lógica del negocio implementada en java y la persistencia de los datos en MYSQL.

DIAGRAMA DE ESTRUCTURA DEL SISTEMA

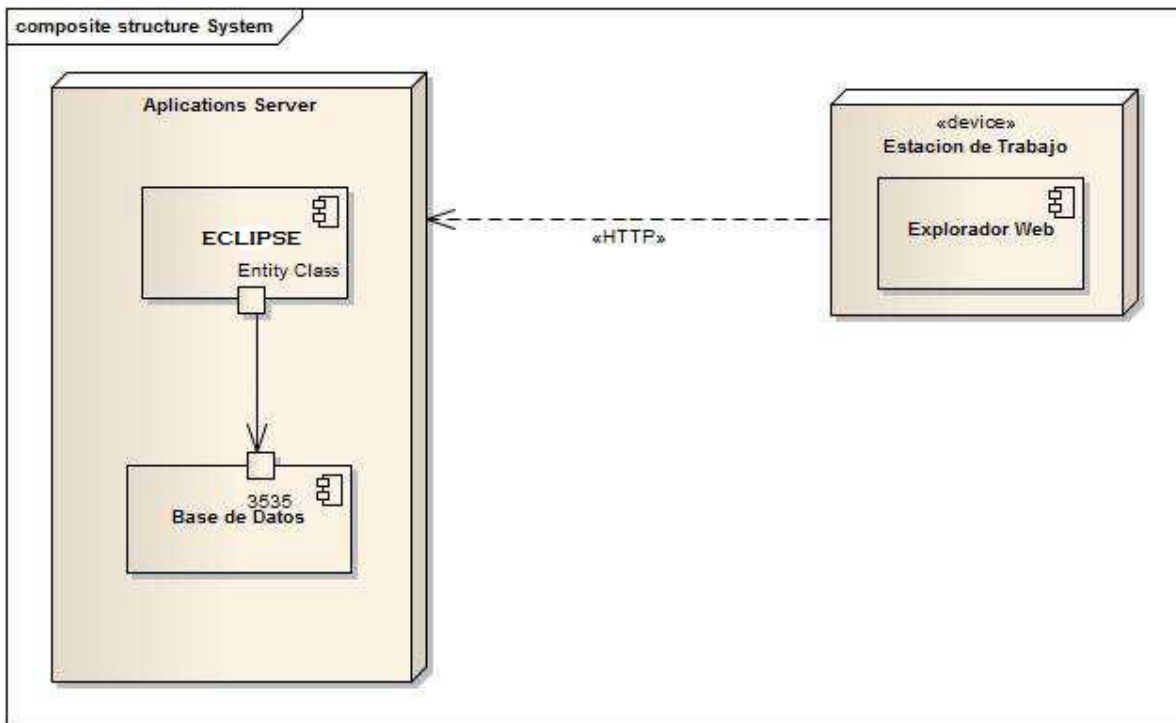


FIGURA 13DIAGRAMA DE ESTRUCTURA DEL SISTEMA Fuente: Autor

A grandes rasgos la capa de acceso consta de varios tipos de interfaces y dispositivos asociados como dispositivos móviles celulares, GPS, desktops entre otros, en esta sección se mencionaran en el desarrollo de un aplicativo cliente para móviles.

Por otra parte, en la capa de entidad se alojan todas aquellas clases que garantizan un mapeo objeto/relacional de los datos usando el API de persistencia de Java, permitiendo que la información alojada en la base de datos relacional pueda ser utilizada de forma transparente por los desarrolladores que usan objetos.

En la capa de servicios, se encuentran los Servicios Web que tiene la función de exponer los métodos del negocio hacia el exterior. Esta capa está dividida en servicios básicos y servicios especializados, en el primer grupo encontramos todas aquellas funcionalidades genéricas que pueden ser utilizadas por los desarrolladores para crear nuevos servicios, como por ejemplo los servicios de identificación de usuario o el servicios de adaptación de contenidos.

En la capa de entidad se alojan todas aquellas clases que garantizan un mapeo objeto/relacional de los datos, usando el API de persistencia de Java, permitiendo que la información alojada en la base de datos relacional pueda ser utilizada de forma transparente por los desarrolladores que usa objetos.

DIAGRAMA DE SECUENCIA

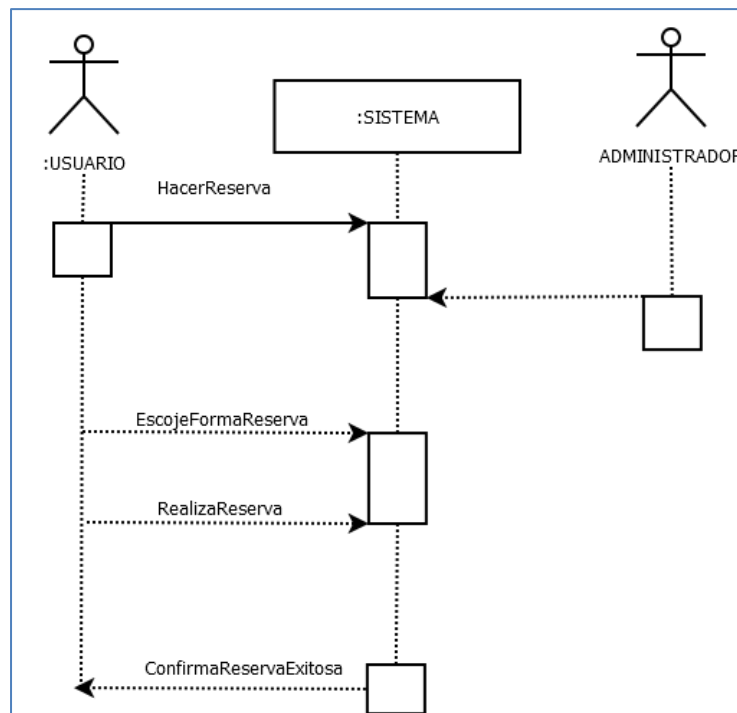


FIGURA 14. DIAGRAMA DE SECUENCIA Fuente: Autor

Para representar de manera más esquemática la interacción del sistema con el usuario, se presentan los diagramas de secuencias de las clases principales que interactúan con el módulo diseño interfaz de aplicación.

El funcionamiento de la aplicación se caracteriza por tener solo tres grandes acciones, la primera es realizar registro “por parte del Usuario”, la segunda es la de realizar reserva “labor realizada por el usuario también” y la tercera es enviar los datos a ser procesados.

Como se puede apreciar, para el caso de uso reservar nodo es el usuario General quien activa la secuencia de eventos que llevan a una correcta selección No es

muy diferente el caso de uso registrar usuario; se gestionan los datos y en el servidor ocurre toda la labor de procesamiento y diligenciamiento de Información.

DIAGRAMA DE REGISTRO

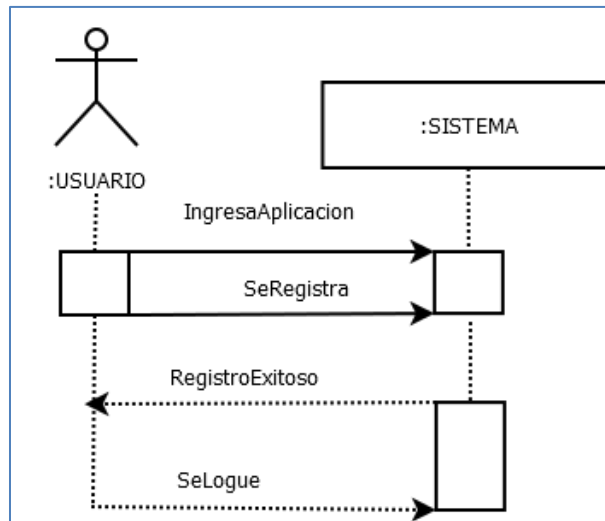


FIGURA 15. DIAGRAMA DE REGISTRO Fuente: Autor

DIAGRAMA DE RESERVA

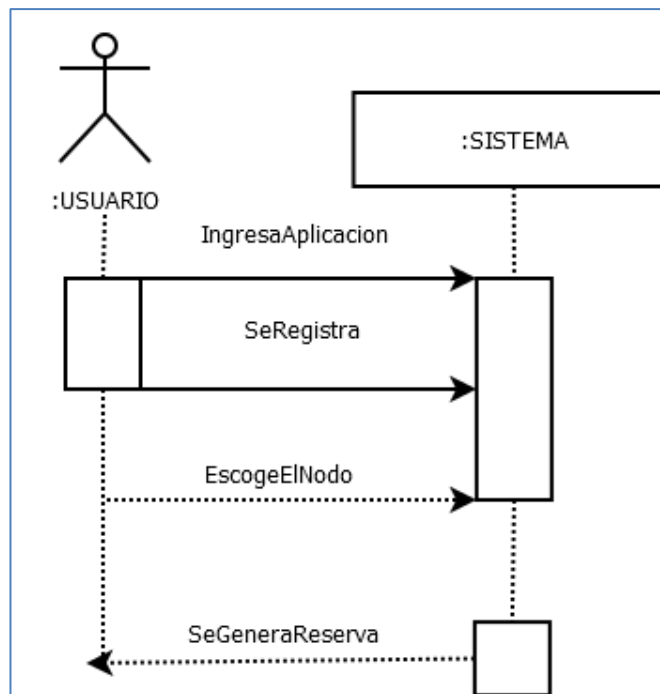


FIGURA 16. DIAGRAMA DE RESERVA Fuente: Autor

4.9 DISEÑO DE COMPONENTES MODULARES QUE INTEGRAN LA APLICACIÓN WEB

Se mencionó anteriormente, LoginGrid está integrado por distintos módulos, como por ejemplo menús, registro, reserva. En esta sección se detalla la fase de implementación y documentación de cada uno de estos componentes.

COMPONENTES MODULARES EN JAVA

La aplicación LoginGrid se desarrollara en java, es un ambiente que permite crear aplicaciones orientadas a objetos pero especializando en aplicaciones de tipo XML, LoginGrid fue diseñado, inicialmente para albergar un módulo de registro y reserva, pero con capacidad de incluir fácilmente nuevos módulos según lo requiera la plataforma.

COMPONENTES MODULARES EN JAVA

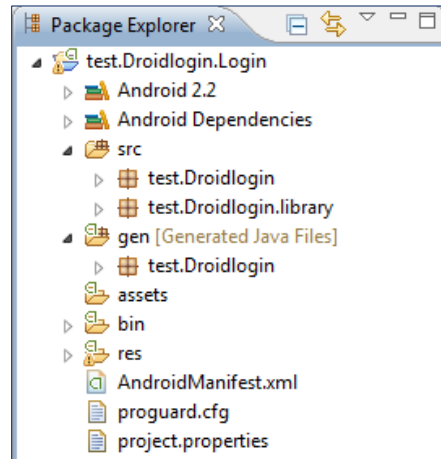


FIGURA 17.DIRECTORIO RAÍZ DEL PROYECTO Fuente: Autor

Como se observa en la figura 16, la raíz del proyecto al que denominamos test.Droidlogin, está constituida por 2 paquetes:

TEST.DROIDLOGIN: Contiene lo que corresponde a la página de presentación de la aplicación que en este caso está compuesto por un menú con 2 enlaces a diferentes sitios de interés relacionado al portal como la wiki del grupo LoginGrid, también se comparte información de contacto con el grupo. Para efectos del proyecto, no se hizo un diseño exhaustivo en este módulo pues, el diseño de este tipo de interfaz gráfica de usuario se pretende sea llevado a cabo directamente por diseñadores gráficos especializados en tal tarea.

INICIO DE SESIÓN



GRIDUIS

Usuario:

Password:

login

FIGURA 18. INICIO DE SESIÓN Fuente: Autor

4.10 ANÁLISIS DE REQUERIMIENTOS.

El análisis de requerimientos busca analizar la especificación de la aplicación y el rendimiento, al igual que el establecer las restricciones. El objetivo es describir lo que el usuario quiere, formando una base para el diseño del software, y la definición de los requisitos que puedan ser verificados en la finalización del desarrollo de la aplicación.

4.10.1 MÓDULO INTERFAZ DEL SERVIDOR.

En este módulo se creará la interfaz Web que permitirá al usuario presentar credenciales de registro a la aplicación. Las funciones realizadas en este módulo serán:

Requisito 1: Permitir al usuario ingreso a la aplicación presentando la información de usuario y password registrado anteriormente en grid.uis.edu.co.

Requisito 2: Permitir al usuario el envío de la información registrada.

Requisito 3: Realizar la consulta a la base de datos y la depuración a documentos XML.

Requisito 4: El usuario podrá ingresar a satisfactoriamente a la aplicación.

4.10.2 MÓDULO SINCRONIZACIÓN

Con este módulo se busca la creación de la aplicación que permita al usuario enviar la información. Como funciones de esta aplicación se tendrá:

Requisito 1: Permitir al usuario detectar el estado de conexión del dispositivo a la web.

Requisito 2: Permitir al usuario la selección de la información.

Requisito 3: Listar las características del dispositivo.

REPRESENTACIÓN GENERAL DEL SISTEMA.

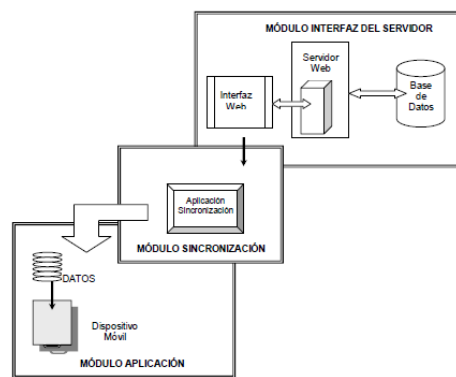


FIGURA 19. REPRESENTACIÓN GENERAL DEL SISTEMA. Fuente: Autor

Servidor:

La funcionalidad de la interfaz Web, es permitir la reserva de nodos por parte del usuario de los cuales este requiere información. Esta solicitud es enviada al servidor, el cual debe tener las siguientes especificaciones:

- Intel Pentium IV 1.6 Mhz o superior.
- Memoria de 256 Mb.
- Tarjeta de red Ethernet a 10/100 Mbps

Dispositivo Móvil:

Para la aplicación, es necesario que el dispositivo móvil tenga las siguientes especificaciones:

-El dispositivo móvil debe tener sistema operativo Android con un microprocesador:

Microprocesador - CPU - 1,4 GhzQuad-coreExynox (cuadruplenucleo) 1 Gb de RAM.

Pantalla - 4.8 pulgadas 720x1280 HD SuperAmoled - GorillaGlass 2

- Sistema operativo del Android 2.3 gingerbread o superior

-La comunicación entre el dispositivo y el PC se hace a través de un cable desincronización USB.

Características del sistema

En la aplicación realizada para visualizar la información en el dispositivo móvil, se debe tener en cuenta el tamaño de la pantalla, ya que el espacio es reducido para la distribución de los elementos. La presentación para un usuario debe ser cómoda, en sentido de presentar facilidad en las consultas y ubicación de los elementos.

Limitaciones del sistema

El sistema ofrece inicialmente la consulta directa de datos, por lo cual no se requieren herramientas específicas de análisis.

No se desarrollará un módulo administrador que maneje la información y controle usuarios que accedan a la interfaz del servidor.

No se tendrá en cuenta la actualización de datos dentro de la aplicación.

5. ANALISIS EVALUACIÓN DE RENDIMIENTO

5.1 PRUEBAS DE INTEGRACIÓN.

En esta etapa se realizaron las pruebas de diseño y prototipo del sistema las cuales se clasificaron en dos grupos, pruebas realizada por los desarrolladores y pruebas de verificación.

Fueron realizadas de manera informal y periódica, en la etapa de diseño del prototipo. Estas pruebas permitieron verificar la funcionalidad de cada módulo y fueron realizadas durante toda la etapa y empalme de los módulos que conforman el diseño, por esta razón no tienen un cronograma ni un orden definido.

5.2 MODULO DISEÑO INTERFAZ DEL SERVIDOR

5.2.1 DESCRIPCIÓN DE LA PRUEBA.

Como objetivo de la prueba a este módulo se verifica el correcto funcionamiento de la interfaz y poder encontrar posibles fallas y errores en:

- La presentación de la interfaz
- Enlace entre la página
- Errores en script.
- La ejecución de las consultas a la base de datos
- Estructura de la información presentada al usuario

5.2.2 PROCEDIMIENTO DE LA PRUEBA

Procedimiento de la prueba Modulo diseño interfaz del servidor
<ol style="list-style-type: none">1. Al ingresar a la interfaz se debe presentar información de registro previamente realizado en la página grid.uis.edu.co2. Hacer clic sobre el botón enviar.

Tabla 7Procedimiento de la prueba. Módulo interfaz.Fuente: Autor

5.2.3 RESULTADOS DE LA PRUEBA

CRITERIO	OBSERVACIÓN
Presentación de la interfaz	CORRECTO
Enlaces con otras páginas	CORRECTO
Estructura de la información presentada al usuario.	AGRADABLE

Tabla 8 Resultado primera prueba. Módulo Interfaz del servidor Fuente: Autor

Los errores que se detectaron en esta prueba (tabla 8), fueron corregidos en su totalidad. Por esto se hizo necesaria la realización de otras pruebas, luego de hacer algunos cambios en la implementación:

5.3 MÓDULO DE SINCRONIZACIÓN

5.3.1 DESCRIPCIÓN DE LA PRUEBA.

Como objetivo de la prueba a este módulo se tenía verificar el correcto funcionamiento de la aplicación de sincronización

- Establecimiento de comunicación con el dispositivo
- Envío de la carpeta con la información.
- Errores de la aplicación.

5.3.2 PROCEDIMIENTO DE LA PRUEBA.

Procedimiento de la prueba Modulo sincronización
<ol style="list-style-type: none">1. Instalación de la aplicación de sincronización en el PC.2. Instalación de la aplicación en el equipo de prueba.3. Conectar el dispositivo con el PC a través del cable de sincronización USB.4. Esperar a que la aplicación reconozca el dispositivo.5. Seleccionar carpeta de datos.6. Enviar la aplicación.

Tabla 9 Procedimiento de la prueba. Módulo sincronización.

Fuente: Autor

5.3.3 RESULTADOS DE LA PRUEBA

Criterio	Observación
Reconocimiento del estado conectado o desconectado del dispositivo.	Correcto
Envío de información.	Correcto
Reconocimiento de las características del dispositivo.	Correcto

Tabla 10. Resultado prueba. Módulo sincronización Fuente: Autor

5.4 MÓDULO APLICACIÓN

5.4.1 DESCRIPCIÓN DE LA PRUEBA.

Como objetivo de la prueba a este módulo se tenía verificar las funciones de la aplicación en el equipo:

- Presentación de las interfaces de la aplicación.
- Visualización de la información en el formulario.

5.4.2 PROCEDIMIENTO DE LA PRUEBA

Procedimiento de la prueba Modulo aplicación
1. Ejecución de la aplicación. 2. Seleccionar y ver la información general, 3. Realizar búsquedas.

Tabla 11 Procedimiento de la prueba Aplicación Fuente: Autor

6. CONCLUSIONES

- Mediante este proyecto se definieron requerimientos fundamentales para el diseño de una aplicación utilizando arquitectura de computo avanzado desde un dispositivo móvil sin tener en cuenta el sistema operativo, como lo son los requerimientos de modulo interfaz del servidor, módulo de sincronización y módulo de aplicación. La propuesta permitirá a los desarrolladores, implementar módulos y aplicaciones de producción, siguiendo los lineamientos escritos en este documento, teniendo en cuenta que se sigue un esquema de servicio dirigido al usuario.

Si bien estos lineamientos y criterios de diseño, están enmarcados dentro del proyecto GridUIS-2 y los servicios de computo avanzado de la Universidad Industrial de Santander, es importante mencionar que pueden extenderse a servicios similares en otras instituciones y garantiza, igualmente la interoperabilidad con otros sistemas posibles, como sería el caso del Sistema de Computo Avanzado para América Latina y el Caribe, el proyecto GISELA, entre otros.

- Se analizó como se presenta la interacción de dispositivos móviles con una plataforma de alto rendimiento. Al revisar las opciones más influyentes que existen en el mercado, fundamentalmente aquellas basadas en IOS de Apple, Android y Windows Mobile se concluyó que el diseño de la aplicación debía basarse en el sistema operativo Android, como se muestra en el documento por posibilidades económicas, impacto, facilidad en el proceso de desarrollo y verificación. Este sistema operativo cuenta con licencia de software libre, con una comunidad de desarrolladores muy activa además que la mayoría de los usuarios “activos” que requerirían eventualmente la interacción móvil con la plataforma, cuenta con dispositivos basados en Android.

No obstante, es importante aclarar, que a pesar de la selección tecnológica, los lineamientos planteados en este diseño, permiten la implementación en cualquiera de las otras dos opciones descartadas para este proyecto. Igualmente el prototipo para los test de interacción, realizado con Android, sirve como ejemplo o base de prueba que nos permitió en el desarrollo del proyecto observar nuevas características de interacción que pueden implementarse a futuro en otros prototipos con otras tecnologías.

- Como se plantea en el documento y se resaltó en la conclusión anterior, los lineamientos de desarrollos propuestos permiten una base para la concepción e implementación de cualquier aplicación que busque interactuar con dispositivos móviles y plataformas de cálculo de alto rendimiento. En este punto es muy importante establecer que “la aplicación” no es un “aplicativo”, es decir, los lineamientos están definidos como se observa en el documento, para la interacción con todos los componentes del sistema de cómputo de alto rendimiento.
- El prototipo de aplicación desarrollado seleccionado, hace la interconexión de recursos de una infraestructura de cómputo avanzado, permitiendo las pruebas realizadas y presentadas en este documento, principalmente, de operación y seguimiento de los lineamientos. Las pruebas de rendimiento, no se tuvieron en cuenta con rigurosidad, debido a que esta fuera del alcance de este proyecto, pero se observa un costo muy ligero tanto de la aplicación como de la interacción misma entre el dispositivo y la plataforma.. En síntesis, el prototipo es el fruto del seguimiento de los pasos y lineamientos planteados en el diseño logrando la implementación de la estrategia propuesta y que sirviera de base para realizar las diferentes modificaciones y pruebas para probar su efectividad.

RECOMENDACIONES

A partir de los lineamientos planteados, se recomienda realizar dos productos que permitan, por un lado administrar ciertos procesos de la plataforma y por el otro, la interacción completa con los servicios de usuario base permitidos (reservación, despliegue de imagen, descargada/cargada de datos, selección de aplicación).

Por otro lado, se recomienda observar los criterios de identificación y federación, así como definir un esquema de seguridad más robusto, que garantice no solo la interacción con las plataformas de GridUIS-2, sino también con las de los proyectos asociados, principalmente observando los lineamientos de seguridad, federación e identificación del Science Gateway de GISELA y EDUROAM.

Una limitación fuerte que se tiene es precisamente la posibilidad de interacción entre múltiples dispositivos con diferentes sistemas operativos. Es necesario tanto en los dispositivos como en los servidores de acceso, frontends y gateways, tener en cuenta la definición de niveles jerárquicos que permitan definir cada una de las políticas y requerimientos para comunicación, servicios colectivos, interacción, interfaces y ejecución de aplicaciones, todo esto en una arquitectura de sistema que permita esa heterogeneidad sin sacrificar simplicidad.

VERIFICACIÓN DEL CUMPLIMIENTO DE LOS OBJETIVOS

Los objetivos que fueron planteados al inicio del proyecto (Numeral 2), se cumplieron en su totalidad durante el desarrollo. A continuación se presenta una tabla donde se clarifica lo entregado:

CUMPLIMIENTO DE LOS OBJETIVOS

Objetivo del proyecto	Diseño de prototipo
<ul style="list-style-type: none"> Analizar la interacción de plataformas de cómputo de alto rendimiento con dispositivos móviles, identificando posibilidades tecnológicas 	Fase análisis del proyecto ver 2.1 grid computing
<ul style="list-style-type: none"> Definir requerimientos, que necesita una aplicación que interactúe con arquitecturas de cómputo avanzado desde dispositivos móviles. 	Módulo interfaz del servidor. ver 4.10 análisis de requerimientos
<ul style="list-style-type: none"> Proponer lineamientos de desarrollo para aplicaciones localizadas en plataformas de cómputo avanzado que requieran acceso desde dispositivos móviles. 	Módulo sincronización. Ver 4.10.2
<ul style="list-style-type: none"> Desarrollar un prototipo de aplicación que permita acceder a recursos específicos en infraestructura de plataformas de cómputo avanzado bajo un modelo de infraestructura como servicio. 	Analizado en las diferentes fases del proyecto y su correspondiente diseño del prototipo.

Tabla 12 Cumplimiento de objetivos Fuente: Autor

BIBLIOGRAFIA Y FUENTES DE INFORMACION

[1] Creation of Web-based user interface for supercomputing environment, Hong Wu, Xue-Bin Chi, Feng Xu. Proceedings on 5th International Conference on Algorithms and Architectures for Parallel Processing, 2002.

[2] DAUM, BERTHOLD; Profesional eclipse 3 para desarrolladores java/
BERTHOLD DAUM.

[3] FOWLER, Martin y SCOTT, Kendall. UML Gota a Gota. México: Addison Wesley Longman, 1999. 203 p.

[4] PRESSMAN, Roger S. Ingeniería del software: un enfoque práctico.
McGrawHill

[5] The Grid 2, Second Edition: Blueprint for a New Computing Infrastructure (The Elsevier Series in Grid Computing), Ian Foster (Editor), Carl Kesselman (Editor), KaufmannEditors, USA 2004.

[6] Página Oficial de Grupo SC3:
http://sc3.uis.edu.co/index.php/P%C3%A1gina_Principal.

[7] Computación Grid, Ventajas y desventajas, Aplicaciones, Textos científicos, 2007:
<http://www.textoscientificos.com/redes/computaciongrid/ventajasdesventajas-aplicaciones>.

[8] Computación distribuida, Ordenadores y portátiles. Disponible en
<http://www.ordenadores-y-portatiles.com/computacion-distribuida.html>.

[9] Developer Android site: <http://developer.android.com/index.html>

[10] Extreme Science and Engineering Discovery Environment
<https://www.xsede.org/>

[11] GISELA Science Gateway, web site <http://gisela-gw.ct.infn.it>

[12] The Grid5000 Project <http://www.grid5000.fr>

[13] Science Gateways Portal <http://sciencegateways.org/>.