

**CIS1530IN03**  
Grids Accesibles

Roberto Rafael García Escallón

PONTIFICIA UNIVERSIDAD JAVERIANA  
FACULTAD DE INGENIERIA  
CARRERA DE INGENIERIA DE SISTEMAS  
BOGOTÁ, D.C.  
2015



CIS1530IN03  
Grids Accesibles

**Autor:**

Roberto Rafael García Escallón

MEMORIA DEL TRABAJO DE GRADO REALIZADO PARA CUMPLIR UNO  
DE LOS REQUISITOS PARA OPTAR AL TÍTULO DE INGENIERO DE  
SISTEMAS

**Director**

Ing. Mariela J. Curiel H. PhD

**Jurados del Trabajo de Grado**

Jaime Andrés Pavlich-Mariscal

Luis Guillermo Torres Ribero

**Página web del Trabajo de Grado**

<http://pegasus.javeriana.edu.co/~CIS1530IN03>

PONTIFICIA UNIVERSIDAD JAVERIANA  
FACULTAD DE INGENIERIA  
CARRERA DE INGENIERIA DE SISTEMAS  
BOGOTÁ, D.C.  
Noviembre 2015

**PONTIFICIA UNIVERSIDAD JAVERIANA  
FACULTAD DE INGENIERIA  
CARRERA DE INGENIERIA DE SISTEMAS**

**Rector Magnífico**

Jorge Humberto Peláez Piedrahita, S.J.

**Decano Académico Facultad de Ingeniería**

Ingeniero Jorge Luis Sánchez Téllez

**Director de la Carrera de Ingeniería de Sistemas**

Ingeniero Germán Alberto Chavarro Flórez

**Director Departamento de Ingeniería de Sistemas**

Ingeniero Rafael Andrés González Rivera

**Artículo 23 de la Resolución No. 1 de Junio de 1946**

*“La Universidad no se hace responsable de los conceptos emitidos por sus alumnos en sus proyectos de grado. Solo velará porque no se publique nada contrario al dogma y la moral católica y porque no contengan ataques o polémicas puramente personales. Antes bien, que se vean en ellos el anhelo de buscar la verdad y la Justicia”*

## **AGRADECIMIENTOS**

Quisiera agradecerle, primero que todo, a la Ingeniera Mariela Curiel, mi Directora de tesis, sin ella nada de lo escrito acá hubiera sido posible. Al mismo tiempo quisiera agradecer al Ingeniero Leonardo Flórez, al brindarnos una aplicación del proyecto.

Personalmente quiero agradecerle a mi familia: mis padres y mi hermano, al igual que a mi novia; quienes me acompañaron en cada una de las etapas de este proyecto, su apoyo fue sumamente importante y sus contribuciones me ayudaron a sobrepasar las más complejas barreras.

## CONTENIDO

<b>CONTENIDO</b> .....	<b>V</b>
<b>FIGURAS</b> .....	<b>VII</b>
<b>TABLAS</b> .....	<b>VIII</b>
<b>I. INTRODUCCIÓN</b> .....	<b>1</b>
<b>II. DESCRIPCIÓN GENERAL</b> .....	<b>3</b>
1. PLANTEAMIENTO DEL PROBLEMA .....	3
1.1. <i>Formulación del problema que se resolvió</i> .....	5
1.2. <i>Justificación del problema</i> .....	5
1.3. <i>Impacto Esperado</i> .....	6
2. OBJETIVOS .....	8
2.1. <i>Objetivo general</i> .....	8
2.2. <i>Objetivos específicos</i> .....	8
<b>III. MARCO METODOLÓGICO</b> .....	<b>9</b>
1. FASE 1.....	9
2. FASE 2.....	9
3. FASE 3.....	9
<b>IV. MARCO TEORICO Y TRABAJOS RELACIONADOS</b> .....	<b>10</b>
1. CONCEPTOS FUNDAMENTALES .....	10
1.1. <i>Grid</i> .....	10
1.2. <i>Grid Emergente</i> .....	11
1.3. <i>Grid Accesible</i> .....	11
1.4. <i>Grid Ad hoc</i> .....	12
1.5. <i>Grid Inalámbrica (Wireless Grid)</i> .....	13
1.6. <i>Grid Móvil</i> .....	14
1.7. <i>Middleware</i> .....	14
1.8. <i>MPI</i> .....	14
1.9. <i>Red Ad hoc</i> .....	15
1.10. <i>Compilación Cruzada</i> .....	15

---

2.	TRABAJOS RELACIONADOS.....	15
2.1.	<i>Estado actual.....</i>	15
2.2.	<i>Trabajos relacionados en Grids Accesibles.....</i>	15
2.3.	<i>Grids integrando recursos móviles.....</i>	19
<b>V.</b>	<b>ANÁLISIS Y SELECCIÓN DE LA GRID.....</b>	<b>20</b>
1.	PRIMERA ETAPA .....	20
2.	SEGUNDA ETAPA .....	20
2.1.	<i>Matriz comparativa.....</i>	22
2.2.	<i>Justificación.....</i>	23
<b>VI.</b>	<b>DESPLIEGUE DE LA GRID SELECCIONADA.....</b>	<b>29</b>
1.	PROCESO PPDIOO.....	29
2.	DESCRIPCIÓN DETALLADA DE BOINC.....	31
3.	PREPARACIÓN PARA EL DESPLIEGUE.....	33
4.	PROCESO DE DESPLIEGUE .....	34
4.1.	<i>Despliegue del Gestor Central.....</i>	34
4.2.	<i>Cliente en la Máquina Virtual.....</i>	34
4.3.	<i>Cliente en la Máquina Host.....</i>	34
4.4.	<i>Cliente en el dispositivo Android.....</i>	35
5.	RESULTADO .....	35
<b>VII.</b>	<b>PRUEBAS.....</b>	<b>37</b>
1.	DISEÑO DE LAS PRUEBAS PARA EL CORRECTO DESPLIEGUE.....	37
1.1.	<i>Correcto despliegue de la aplicación web.....</i>	37
1.2.	<i>Inscripción y ejecución de tareas de manera local.....</i>	37
1.3.	<i>Inscripción y ejecución de tareas con la máquina host.....</i>	38
1.4.	<i>Inscripción y ejecución de tareas en los dispositivos móviles.....</i>	38
2.	VALIDACIÓN Y RESULTADOS .....	38
2.1.	<i>Resultados: Correcto despliegue de la aplicación web.....</i>	38
2.2.	<i>Resultados: Inscripción y ejecución de tareas de manera local.....</i>	39
2.3.	<i>Resultados: Inscripción y ejecución de tareas con la máquina host.....</i>	39
2.4.	<i>Resultados: Inscripción y ejecución de tareas en los dispositivos móviles.....</i>	40
<b>VIII.</b>	<b>EVALUACIÓN DE LA GRID DESPLEGADA .....</b>	<b>41</b>
1.	ANÁLISIS DE IMPACTO .....	41



2.	DISCUSIÓN DE RESULTADOS.....	42
2.1.	<i>Restricciones del despliegue.....</i>	42
2.2.	<i>Comparación con otras alternativas.....</i>	42
<b>IX.</b>	<b>CONCLUSIONES.....</b>	<b>44</b>
1.	TRABAJO FUTURO .....	44
1.1.	<i>BOINC en la Pontificia Universidad Javeriana.....</i>	45
1.2.	<i>Wrapper y administración de aplicaciones.....</i>	45
1.3.	<i>Interfaz de administración.....</i>	45
<b>X.</b>	<b>REFERENCIAS Y BIBLIOGRAFÍA .....</b>	<b>46</b>
<b>XI.</b>	<b>ANEXOS.....</b>	<b>53</b>
	ANEXO 1: PREPARACIÓN DE LA MÁQUINA VIRTUAL.....	53
	ANEXO 2: PROCESO DE DESPLIEGUE DEL GESTOR .....	53
	ANEXO 3: BOINC API .....	53
	ANEXO 4: CREACIÓN Y EJECUCIÓN DE APLICACIONES.....	53
	ANEXO 5: FUNCIONALIDADES DE BOINC.....	53

## FIGURAS

Figura 1	Capas de la Grid. Tomado de Foster, Ian (2002).....	10
Figura 2	Definición de tipos de Grids. Heba Kurdi, Maozhen Li, and Hamed Al-Raweshidy (2008).....	11
Figura 3	Grid Ad hoc. Heba Kurdi, Maozhen Li, and Hamed Al-Raweshidy (2008) .....	12
Figura 4	Grid Ad hoc Móvil. Heba Kurdi, Maozhen Li, and Hamed Al-Raweshidy (2008) .	12
Figura 5	Grid Inalámbrica. Heba Kurdi, Maozhen Li, and Hamed Al-Raweshidy (2008).....	13
Figura 6	Grid Móvil. Heba Kurdi, Maozhen Li, and Hamed Al-Raweshidy (2008).....	14
Figura 7	Modelo de Despliegue .....	30
Figura 8	Componentes BOINC.....	31

**TABLAS**

Tabla 1 Matriz de comparación DAR..... 23

Tabla 2 Ambiente de Desarrollo ..... 33

## **ABSTRACT**

This Graduation Project presents a comparative analysis between different projects implementing Accessible Grids that use mobile devices. As a result of said analysis the author decide on deploying the BOINC system, including mobile devices, in order to execute assignments. BOINC offers some limitations, like the need for a central manager and lack of control on the resources. Due to BOINC's implementation of voluntary computing there's access to more resources, but the availability of said resources depends on the users. Although the technology still in development, due to the few practical Works found.

## **RESUMEN**

El uso de los dispositivos móviles ha aumentado de forma importante alrededor del mundo en los últimos años. Por otro lado, las capacidades actuales de los teléfonos inteligentes se han incrementado de tal forma que comienzan a ser considerados como una posible infraestructura de cómputo. Por ejemplo, se ha estudiado la problemática de incorporarlos a una Grid, no sólo para mejorar el acceso de los usuarios a los recursos de la Grid, sino como proveedores de recursos; en este último caso se conocen como Grid Móviles o también Grids Accesibles. Este trabajo profundiza sobre el estado actual de la tecnología existente para Grids Móviles y se busca el sistema más adecuado que permita la ejecución de tareas, escritas en C++, en dispositivos móviles. Como resultado del análisis se decide desplegar la Grid BOINC, la cual permite ejecutar tareas en C++. Entre las principales conclusiones de este trabajo se encuentran, el confirmar la posibilidad de ejecutar tareas en C++ en teléfonos Android, realizando ciertas modificaciones a los programas, por otro lado se nota una tecnología aún en estado de maduración, por los pocos trabajos prácticos encontrados.



## I. INTRODUCCIÓN

El uso de los dispositivos móviles se ha incrementado de forma importante alrededor del mundo en los últimos años, tanto así que el uso de Internet desde dispositivos móviles superó por primera vez en 2014 a su uso desde computadores personales (Lella & Lipsman, 2014). Algunos autores consideran que el smartphone es uno de las mejores invenciones de la actualidad y su acelerada adopción hace pensar que la tecnología aún seguirá desarrollándose, a medida que los usuarios se van dando cuenta de todo lo que puede estar al alcance de uno o varios clicks (Mutchler, Shim, & Ormond, 2011).

Por otro lado, las capacidades actuales de los teléfonos inteligentes se han incrementado en los últimos años. Hoy en día existen smartphones cuyas características se comparan a las que tenían algunas estaciones de trabajo de gama alta (SGI Indigo-2) unos años atrás (Büsching, Schildt, & Wolf, 2012). Es por esta razón que los smartphones pueden ser considerados apropiados para ejecutar aplicaciones. Rodríguez y otros autores destacan que diariamente se activan alrededor de 400,000 nuevos smartphones con sistema operativo Android, y la mayoría de las capacidades de estos dispositivos están siendo subutilizadas (Rodríguez, Mateos, & Zunino, 2012). Se ha explorado su uso para la computación científica o como infraestructura de cómputo alterna para una empresa que provee smartphones a sus empleados (Vidal Aroca & Garcia Goncalves, 2012; Rodríguez, Mateos, & Zunino, 2012; Duan, y otros, 2014).

Siguiendo la misma motivación, desde la década anterior también se han desarrollado trabajos de investigación que plantean la problemática de incorporar dispositivos móviles a una Grid.

Una Grid es una arquitectura de hardware y software que provee acceso de manera confiable a recursos computacionales y de almacenamiento, los cuales están geográficamente dispersos y pueden pertenecer a distintas organizaciones (Foster, Kesselman, Nick, & Tuecke, 2002). La extensión del modelo de Grid tradicional para usar dispositivos móviles llega naturalmente debido al gran auge de la tecnología móvil en los últimos años. Surgen nuevos términos como el de Grid Accesible: Kurdi, Maozhen y Al-Raweshidy (2008) definen las Grids Accesibles como aquellas Grids cuyos recursos están disponibles sin importar su ubicación geográfica o su capacidad física. Esta definición es amplia y abarca las Grids Inalámbricas (Wireless), Ad hoc y Móviles. Siendo una Grid Inalámbrica aquella que incluye recursos o es accedida de manera inalámbrica; una Grid Móvil es aquella donde se incluyen dispositivos móviles para ejecución de tareas o para acceder a la Grid y, finalmente, una Grid Ad hoc es aquella que se construye a partir de los dispositivos disponibles sin utilizar un gestor central (Kurdi, Maozhen, & Al-Raweshidy, 2008).

El objetivo de este trabajo es explorar el uso de dispositivos móviles para la ejecución de tareas escritas en C++. Se puede decir que este Trabajo de Grado es el primer objetivo de un proyecto de investigación más ambicioso que pretende el procesamiento de imágenes médicas con ayuda de dispositivos móviles. De ahí la restricción sobre el lenguaje de programación.

Se decidió evaluar una Grid como infraestructura base, en lugar de hacer una implementación propia porque esta tecnología debe ofrecer valores agregados principalmente en lo que corresponde al descubrimiento de recursos, planificación y calidad de servicio.

Una vez tomada la decisión sobre el uso de la Grid, el siguiente paso fue evaluar el estado actual de la tecnología existente. Si bien se tiene conocimiento de varios artículos que describen proyectos e implementaciones de Grids Accesibles, no todos los sistemas descritos están disponibles para su uso, poseen buena documentación u ofrecen soporte, debido a que son proyectos que no han seguido actualizándose.

Uno de los principales objetivos de este trabajo es proveer información sobre las herramientas disponibles y seleccionar la más idónea de acuerdo a las restricciones del Trabajo de Grado y del proyecto de investigación mencionado. Una vez seleccionada la Grid se puede seguir con la estrategia de despliegue y la ejecución de tareas.

Las principales contribuciones de este trabajo son:

- a) Brindar elementos que permitan conocer el estado actual de las Grids Accesibles no sólo desde el punto de vista teórico, sino probando implementaciones reales para ejecutar tareas con ciertas características.
- b) Una vez seleccionada la herramienta se elaboran manuales de instalación y de uso que permitan seguir explorando la Grid para la ejecución de tareas desde un punto de vista más avanzado de la investigación.

Este trabajo se estructura de la siguiente forma: en el capítulo [II. DESCRIPCIÓN GENERAL](#) se detalla el planteamiento del proyecto; en el capítulo [III. MARCO METODOLÓGICO](#) se describe la metodología usada; en el capítulo [IV. MARCO TEÓRICO Y TRABAJOS RELACIONADOS](#) se definen los conceptos fundamentales utilizados a lo largo de este trabajo y se presentan artículos que han explorado temas similares a este; en el capítulo [V. ANÁLISIS Y SELECCIÓN DE LA GRID](#) se describe el proceso de comparación de varios sistemas de Grid Accesible, y se elige una de ellas para continuar; en el capítulo [VI. DESPLIEGUE DE LA GRID SELECCIONADA](#) se explica el proceso de despliegue que se realizó; en el capítulo [VII. PRUEBAS](#) se describen las pruebas que se realizaron sobre la Grid, al igual que sus resultados; en el capítulo [VIII. EVALUACIÓN DE LA GRID DESPLEGADA](#) se evalúan los resultados de las pruebas; el capítulo [IX. CONCLUSIONES](#) presenta los principales hallazgos del Trabajo de Grado, al igual que el posible trabajo futuro.

## II. DESCRIPCIÓN GENERAL

### 1. Planteamiento Del Problema

Algunos problemas computacionales requieren de una gran cantidad de recursos de procesamiento para solucionarse en un tiempo aceptable. Para resolver estos problemas en menor tiempo, se han desarrollado diversos paradigmas como la computación paralela y distribuida. La computación paralela pudiera reducir el tiempo de ejecución de una aplicación, especialmente si no hay sincronización explícita entre las tareas que la componen (paralelismo de grano grueso) (Stallings, 1995; Bargiela & Pedrycz, 2012). Un ejemplo de este tipo de aplicaciones es el procesamiento de imágenes en búsqueda de patrones (Schalkoff, 1989), las cuales se dividen en múltiples cuadros y cada computador miembro de un multiprocesador o máquina paralela toma un cuadro para procesar (Choudhary & Ponnusamy, 1991).

El procesamiento paralelo se inició en supercomputadoras de gran tamaño y costo, posteriormente se utilizaron los clústeres de computadoras y a continuación surgió el paradigma de Grid Computing (Foster & Kesselman, 2013; Foster, Carl, & Tuecke, 2001). Una Grid es un sistema de computación distribuido para resolver problemas complejos, desde el punto de vista computacional, en el cual sus componentes son heterogéneos, están geográficamente dispersos y no necesariamente pertenecen a la misma organización.

Los recursos compartidos en una Grid-computadores de alto rendimiento –redes, bases de datos, aplicaciones, instrumentos especializados (telescopios), entre otros– están bajo la administración de diferentes organizaciones, siendo su ubicación transparente para el usuario. Entre las metas de la tecnología Grid se encuentran: el proveer la ilusión de un sistema integrado de alto desempeño y alta disponibilidad; promover la colaboración de individuos e instituciones en diferentes partes del mundo, a través de las organizaciones virtuales; y usar estándares para protocolos e interfaces. Algunos campos que han aplicado las Grids para sus problemas son la simulación molecular, la física de partículas, el modelado del clima, el estudio del Genoma Humano, la física de altas energías (HEP), la biomedicina y la química computacional (Bernman, Fox, & Hey, 2003).

La extensión del modelo de Grid tradicional para usar dispositivos móviles llega naturalmente debido al gran auge de estos dispositivos en los últimos años: para 2014 existían 1.5 billardos de dispositivos móviles en el mundo, y cada año se venden más (ABI Research, 2014; Research, 2015). Estos son dispositivos cuentan con varias capacidades tales como poder de procesamiento, sistemas de comunicación, almacenamiento y sensores. Los trabajos de Lingjie Duan, Takeshi Kubo, Kohei Sugiyama, Jianwei Huang, Teruyuki Hasegawa y Jean Walrand (2014) y Juan Manuel Rodríguez, Carlos Mateos y Alejandro Zunino (2012); han

explorado las capacidades de estos dispositivos para computación científica (Duan, y otros, 2014; Rodriguez, Mateos, & Zunino, 2012). No obstante, los dispositivos móviles también poseen varias debilidades (Li, Sun, & Ifeachor, Challenges of mobile ad-hoc grids and their applications in e-healthcare, 2005):

- Alto consumo de energía: cada bit procesado o transmitido o cada segundo que permanezca la pantalla prendida se consume energía que para estos dispositivos es limitada.
- Baja disponibilidad: la red inalámbrica a la que pertenecen no ofrece la alta disponibilidad que tiene una red cableada.
- Altos costos de transmisión: la transmisión no es gratis en energía y en el caso de redes celulares podría haber un significativo costo financiero, entre otros (Shah, 2013).

Al integrarse los dispositivos móviles a las Grids, estos pueden ser tanto proveedores de servicios, los cuales contribuyen al procesamiento de información y/o datos; o tan solo consumidores de los servicios de la Grid (Furthmüller & Waldhorst, 2010). Este trabajo se enfoca en el uso de los dispositivos móviles como proveedores de recursos de cómputo de una Grid. Esta Grid, al tener elementos de naturaleza inalámbrica y en algunos casos móviles, tomaría el nombre de **Grid Accesible**. Kurdi, Maozhen, y Al-Raweshidy (2008) definen las Grids Accesibles como aquellas Grids cuyos recursos están disponibles sin importar su ubicación geográfica o su capacidad física. Esta definición es amplia y abarca las Grids Inalámbricas (Wireless), Ad hoc, y Móviles. Siendo una Grid Inalámbrica aquella que incluye recursos o es accedida de manera inalámbrica; una Grid Móvil es aquella donde se incluyen dispositivos móviles como proveedores o consumidores de servicios; y, finalmente, una Grid Ad hoc es aquella que se construye a partir de los dispositivos disponibles sin utilizar un gestor central, y no tiene un punto específico de entrada por donde se explotará (Kurdi, Maozhen, & Al-Raweshidy, 2008).

Si bien hacia el año 2010, cinco universidades colombianas, entre las que se encontraba la Pontificia Universidad Javeriana, integran la Grid Colombia (Castro, Chacón, Díaz, González, & Zuluaga, 2011). El proyecto se ha desacelerado en los últimos años y no se conocen en el país avances en la construcción de Grids Accesibles en cualquiera de sus tres tipos. Se plantea entonces la necesidad de construir una Grid Accesible con los propósitos de: conocer el estado actual de la tecnología y de apropiarse de ella, lo cual nos permitirá pensar a futuro en posibles aplicaciones para esta nueva tecnología.

El tipo de aplicaciones que se pueden beneficiar de una Grid Accesible es un tema que se menciona en la sección [II.1.3 Impacto Esperado](#). Otra posible aplicación planteada en el seno de los grupos SiDre y Takina del Depto de Ing. de Sistemas de la PUJ, es el procesamiento de imágenes médicas (IM) en una Grid Móvil Ad hoc formada parcial o totalmente por dispositivos móviles. Esto permitiría, por ejemplo que un médico pudiera –ante una



emergencia que ocurriera en un sitio lejano a un centro asistencial de salud- procesar de forma rápida los resultados de exámenes usando los dispositivos móviles que estuvieran al alcance y hacer un diagnóstico del paciente mientras se llega al Centro de Salud. Si bien en este trabajo no se tratará de forma alguna el problema del procesamiento de imágenes, la posibilidad de ejecutar programas en C++ fue un factor determinante al momento de tomar una decisión sobre la tecnología Grid a utilizar, ya que en este lenguaje están escritos varios de los algoritmos de procesamiento de imágenes utilizados al interior de la universidad. También se tomará en cuenta el uso de la librería de comunicaciones MPI que también es utilizada por los algoritmos paralelos de procesamiento de IM.

### **1.1. Formulación del problema que se resolvió**

¿Cómo se pueden explotar las capacidades de cómputo y comunicación de los dispositivos móviles para la formación de un Grid de Computación Accesible que incluya dispositivos móviles como proveedores de recursos?

### **1.2. Justificación del problema**

La incorporación de los dispositivos a la Grid, principalmente para la ejecución de trabajos, presenta grandes desafíos que tienen que ver con la duración de la batería, su movilidad, la velocidad de las redes inalámbricas, la intención de los usuarios de prestar sus recursos a la Grid, entre otros. Muchos de estos problemas se han venido estudiando desde hace algunos años y actualmente existe una amplia literatura sobre el problema, resumida en algunos artículos como: Ahuja, S. P., & Myers, J. R. (2006), Gang, L., Hongmei, S., Gao, H., Yu, H., & Cai, Y. (2009), Hijab, M., & Avula, D. (2011), Manvi, S. S. & Birje M. N. (2010), Rodríguez, J. M., Zunino, A., & Campo, M. (2011). No obstante, existen pocas implementaciones de Grids Accesibles (la mayoría son Grids Móviles) entre las que se encuentran: Akogrimo, MoGrid, MiPeG, MORE, y BOINC (Jaehnert, Wesner, & Villagra, 2007; Lima, y otros, 2005; Coronato & De Pietro, 2008; Wolff, Michaelis, Schmutzler, & Wietfeld, 2007; Anderson P. D., 2004). Infortunadamente no todas estas herramientas están disponibles o han continuado desarrollándose. Por ejemplo, en el caso de Akogrimo, la página del proyecto ya no contiene información (<http://www.mobilegrids.org/>). Por otro lado, a pesar de existir muchas propuestas de algoritmos para la gestión de recursos la mayoría de las propuestas se han probado usando la simulación, dejando de lado los problemas que se pueden presentar cuando se hace el despliegue en sistemas móviles reales. Lo anterior sugiere que la tecnología de Grids Accesibles (Móviles, Inalámbricas o Ad-hoc) no está aún madura, lo que representa sin duda una gran oportunidad de investigación (Gang, Hongmei, Huahao, Haiyan, & Yue, 2009; Ahuja & Myers, 2006; Hijab & Avula, 2011; Manvi & Birje, 2010; Rodríguez, Zunino, & Campo, 2011; Curiel, 2016).

Entre las ventajas que nos proveerían las Grids Accesibles sobre las Grids tradicionales se encuentran:

- **Disminución de costos:** Una limitante común de la tecnología Grid es el costo de sus recursos: supercomputadores, clústeres, bases de datos, etc. Este costo se pudiera disminuir en gran medida si se usa una Grid Accesible en vez de una tradicional ya que buena parte de los recursos de una Grid Móvil los proveerían personas dispuestas a compartir las capacidades de cómputo de sus dispositivos. También se ha estudiado la posibilidad de explotar los equipos móviles que las empresas le entregan a sus empleados (Vidal Aroca & Garcia Goncalves, 2012); en este caso la empresa es la propietaria y no requeriría del consentimiento de los empleados.
- **Accesibilidad:** La movilidad del dispositivo le facilita al usuario compartir con la Grid de manera transparente sus recursos, ejecutar sus tareas y obtener resultados instantáneos mientras está en movimiento. La Grid es siempre accesible, sin importar la ubicación del usuario.
- **Disponibilidad:** Si se logra implementar una Grid Ad-hoc, se pudiera pensar en la resolución de problemas con los recursos que estuvieran a la disposición (y cada día hay más personas que tienen al menos un teléfono celular), sin esperar tener a disposición un servidor o un clúster de computadoras.

Este proyecto se justifica principalmente por la necesidad de conocer el estado del arte de la tecnología de Grids Accesibles. Se quieren responder preguntas como: ¿Qué tanto de lo que está descrito en los artículos está implementado y se puede utilizar? ¿Es factible implementarlo con los recursos que se tienen a la disposición? ¿Se tienen ventajas sobre los Grids tradicionales si se usa una Grid accesible del tipo móvil o inalámbrico? ¿Qué ventajas tiene usar una Grid (ya implementada) de dispositivos móviles para ejecutar tareas, en lugar de desarrollar todo el software localmente?

### 1.3. Impacto Esperado

Se espera que los resultados del trabajo de grado puedan ser usados en trabajos futuros. El impacto se ha dividido en dos grandes grupos:

#### a. Impacto en la Comunidad Académica

**Investigación y Docencia:** los resultados de este trabajo de grado pudieran dar origen a otros trabajos de pregrado o de maestría, al igual que propuestas de proyectos de investigación, no sólo dentro del grupo SiDre sino con otros grupos del Depto. de Ingeniería de Sistemas o Electrónica. Sin duda los resultados obtenidos pueden aprovecharse en los cursos de Introducción a los Sistemas Distribuidos, Sistemas Distribuidos, Computación Móvil o Computación de Alto Desempeño.

## b. Impacto en otros ámbitos:

En la literatura se han reportado soluciones a problemas de salud, ambiente, desastres naturales, etc., con el uso de Grids Accesibles (especialmente móviles) (Gang, Hongmei, Huahao, Haiyan, & Yue, 2009; Manvi & Birje, 2010). A continuación se presentan algunos ejemplos:

**Manejo de desastres:** los desastres naturales como terremotos, inundaciones, volcanes, tsunamis, incendios forestales, etc. causan mucha destrucción y como medida de prevención se pueden instalar redes de sensores, que consisten de sensores inalámbricos y alámbricos. La Grid de sensores debidamente instalada y mantenida proveería una advertencia de desastres futuros, al igual que una respuesta ante estos (Silva, Del Luca Almeida, Poersch, & Nogueira, 2010). Sin embargo la instalación de la red de sensores toma un esfuerzo y recursos, por lo que se ha estudiado utilizar los dispositivos móviles y sus sensores para crear esta red, efectivamente formando una Grid Móvil (Chatzimilioudis, Konstantinidis, Laoudias, & Zeinalipour-Yazti, 2012).

**Salud:** las Grids inalámbricas pueden integrar como recursos, no sólo a los teléfonos inteligentes, sino a una amplia selección de dispositivos médicos para proveer un servicio personalizado de salud; este nuevo paradigma, requiere de una conexión uno a uno entre el paciente y el experto (Otto, Milenkovic, Sanders, & Jovanov, 2006). Además, el servicio de salud basado en Grids inalámbricas puede integrar tales servicios con el contexto del paciente, formando una red de centros de diagnosis, tanto estáticos como móviles, alrededor del mismo. En casos de emergencias, esta red puede ayudar para que la primera respuesta sea lo más eficiente posible. Si la emergencia ocurre en zonas remotas donde no exista una infraestructura fija, un grupo heterogéneo de dispositivos podría formar una Grid para ayudar a resolver la situación, analizando los primeros datos y/o procesando los primeros resultados de pacientes que pudieran estar en peligro (Malan, Fulford-Jones, Welsh, & Moulton, 2004; Li, Sun, & Ifeachor, Challenges of mobile ad-hoc grids and their applications in e-healthcare, 2005).

**Organizaciones:** en el caso de organizaciones también puede haber posibles usos. Se ha explorado la posibilidad de utilizar los dispositivos móviles de una determinada empresa u organización que son entregados a sus empleados. Estos dispositivos podrían usarse cuando están ociosos, por ejemplo mientras el empleado está durmiendo, en la ejecución de procedimientos pesados (Vidal Aroca & Garcia Goncalves, 2012). De manera similar se ha pensado en utilizar los dispositivos personales de los empleados, mientras están conectados a la infraestructura eléctrica de la organización en horario laboral, ayudando en tareas del día a día de la organización (Büsching, Schildt, & Wolf, 2012).

A medida que se domine la tecnología de las Grids Accesibles, pueden surgir nuevos problemas, que no solo requieran de un gran poder de procesamiento, sino necesiten de

soluciones rápidas, al alcance de uno o varios clicks, como el problema de procesamiento de médicas mencionado al comienzo del capítulo.

## **2. Objetivos**

A continuación se presentan los distintos objetivos de este Trabajo de Grado.

### **2.1. Objetivo general**

Desplegar, con la tecnología existente, una Grid Accesible que incluya dispositivos móviles Android como proveedores de recursos.

### **2.2. Objetivos específicos**

- Determinar el estado actual de la tecnología para Grids Accesibles, comparando las diversas propuestas y desarrollos.
- Establecer un modelo para desplegar una Grid Accesible en dispositivos Android.
- Evaluar el modelo establecido.

### III. MARCO METODOLÓGICO

Para resolver el problema se establecieron tres fases principales, cada una con una metodología distinta debido a los objetivos que se pretendían cumplir.

#### 1. Fase 1

Durante la primera fase metodológica se determinó el estado actual de las tecnologías de Grids Accesibles y se decidió la mejor en este caso, utilizando la metodología DAR, comparando los distintos Frameworks y herramientas existentes. Esta investigación se realizó por medio de artículos publicados en memorias de conferencias y revistas reconocidas, libros, y revisando la documentación existente de las distintas herramientas en los sitios web. Para evaluar las distintas opciones se determinaron varios atributos críticos que debieron cumplir las herramientas. Finalmente se escogió la mejor solución para su despliegue. Las actividades realizadas en esta fase fueron:

- a) Revisión bibliográfica.
- b) Análisis comparativo de los sistemas existentes usando la metodología *Decision Analysis and Resolution Process*, DAR (The Process Group, 2005).

Los resultados de estas actividades se presentan en el capítulo [V. ANÁLISIS Y SELECCIÓN DE LA GRID](#) y serán utilizados en la Fase 2.

#### 2. Fase 2

Durante la segunda fase se desplegó la Grid que fue elegida como la mejor en la fase anterior. Se diseñó una arquitectura de despliegue factible de probar en el laboratorio, utilizando los teléfonos Android, la cual fue implementada. Para el diseño de la arquitectura se utiliza la metodología PPDIIO de Cisco (Tiso & Teare, 2011). Los resultados de esta actividad se presentan en el capítulo [VI. DESPLIEGUE DE LA GRID SELECCIONADA](#).

#### 3. Fase 3

Una vez implementada la Grid se iniciaron pruebas para evaluar su funcionamiento con programas sencillos en C++. Principalmente se evaluó el correcto funcionamiento de la plataforma de ejecución, tanto en recursos fijos como móviles. Los resultados de esta actividad se presentan en el capítulo [VII. PRUEBAS](#).

## IV. MARCO TEORICO Y TRABAJOS RELACIONADOS

### 1. Conceptos Fundamentales

A continuación se presentan los conceptos fundamentales que son utilizados a lo largo del texto.

#### 1.1. Grid

Una Grid es una arquitectura de hardware y software que provee acceso de manera confiable, ubicua, consistente y barata a recursos computacionales y de almacenamiento, geográficamente dispersos, pertenecientes a distintas organizaciones (Foster, Kesselman, Nick, & Tuecke, 2002). La arquitectura de la Grid se basa en cuatro capas: la primera contiene las aplicaciones de los usuarios, que utilizan los servicios que ofrece la Grid; la segunda capa toma el papel de gestor central ofreciendo servicios de directorio, diagnóstico, monitoreo, entrega de tareas, programación de tareas, etc.; la tercera capa define los protocolos de comunicación e interfaces para realizarla; la cuarta y última capa son los recursos sobre los cuales se apoya todo el sistema, también llamada la *Fabric* o tela en español. Cada capa utiliza a la capa inmediatamente inferior, al tiempo que es utilizada por la capa inmediatamente superior (Foster, 2002).

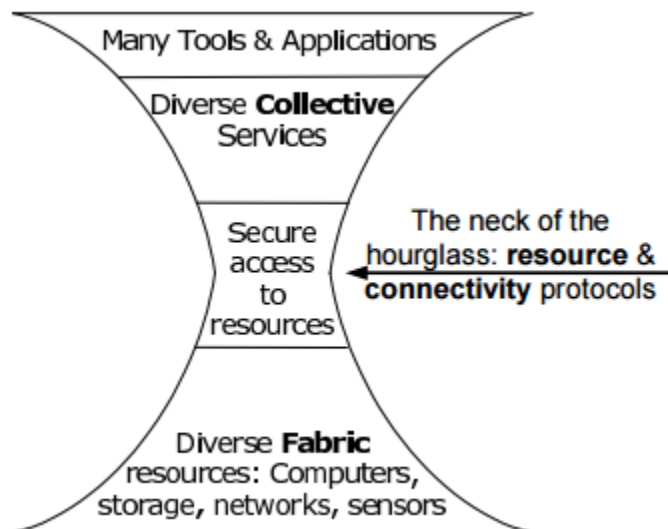


Figura 1 Capas de la Grid. Tomado de Foster, Ian (2002)

## 1.2. Grid Emergente

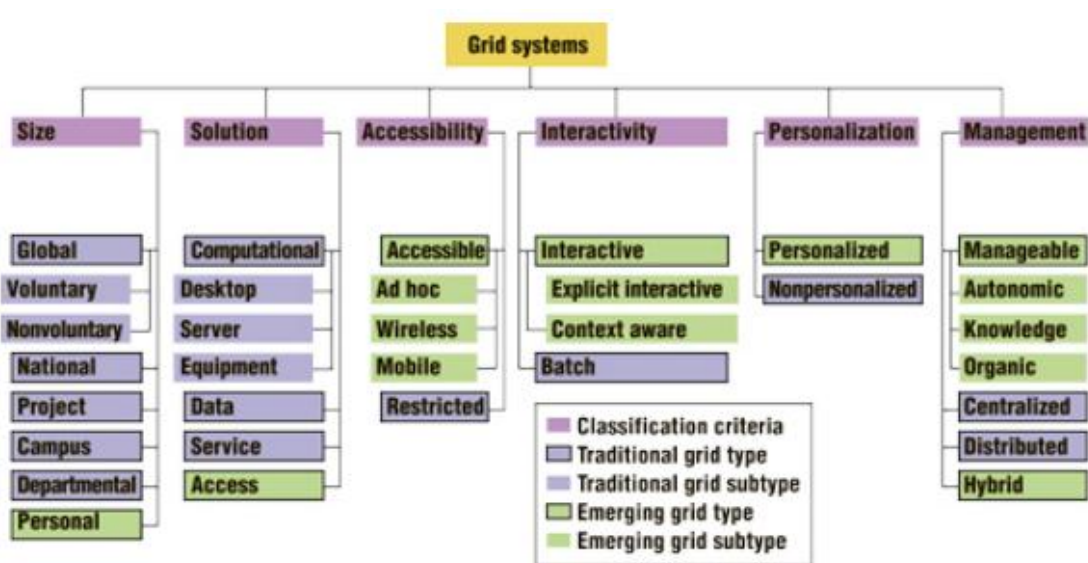


Figura 2 Definición de tipos de Grids. Heba Kurdi, Maozhen Li, and Hamed Al-Raweshidy (2008)

Las Grids Emergentes son las distintas tecnologías Grid que están surgiendo en los últimos años. Se dividen según seis dimensiones: accesibilidad, interactividad, facilidad de gestión, personalización, solución y tamaño. En la Figura 2 se muestran las dimensiones y distintos tipos de Grid bajo cada una. Según estas dimensiones se toman cuatro tipos principales de Grids emergentes: las Accesibles, las Interactivas, las Gestionables y las personalizadas. En este trabajo de grado se tratarán las Grids Accesibles, los otros tres tipos de Grids están fuera del alcance del trabajo y se definen en (Kurdi, Maozhen, & Al-Raweshidy, 2008).

## 1.3. Grid Accesible

Es un tipo de Grid emergente que se centra en el acceso a los recursos. Tomando dos aproximaciones; le facilita a los recursos unirse a la Grid; y a su vez, facilita la explotación de los recursos de la Grid a los usuarios. Este tipo de Grid se compone de las Grids Ad hoc, Inalámbricas y Móviles. Su naturaleza es dinámica, resultante de la cantidad de cambios que sufre su estructura a lo largo del tiempo debido a nodos entrando y saliendo, apagándose y moviéndose. La accesibilidad que brinda da lugar a nuevas aplicaciones en situaciones de emergencia y lugares alejados (Kurdi, Maozhen, & Al-Raweshidy, 2008). A continuación se describen detalladamente los tres tipos de Grids Accesibles.

#### 1.4. Grid Ad hoc

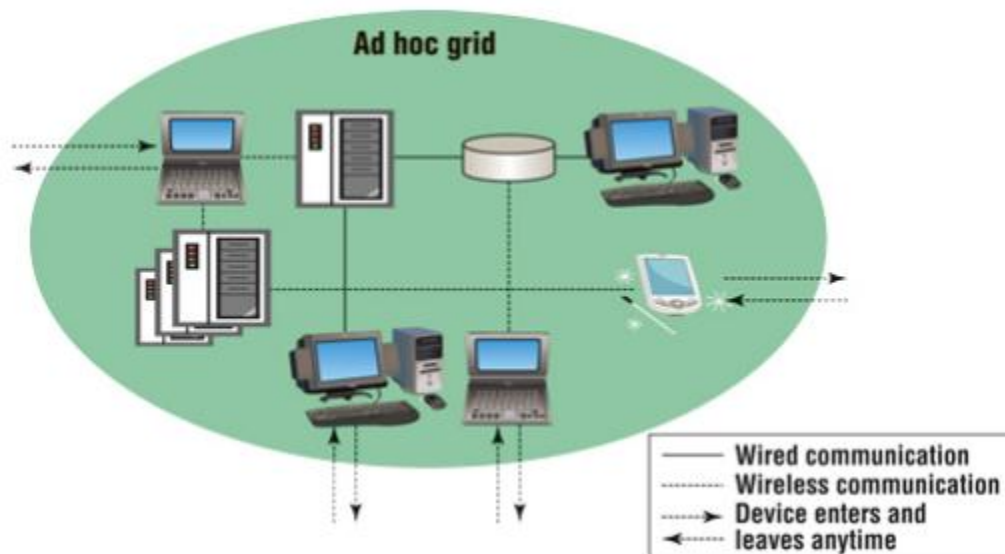


Figura 3 Grid Ad hoc. Heba Kurdi, Maozhen Li, and Hamed Al-Raweshidy (2008)

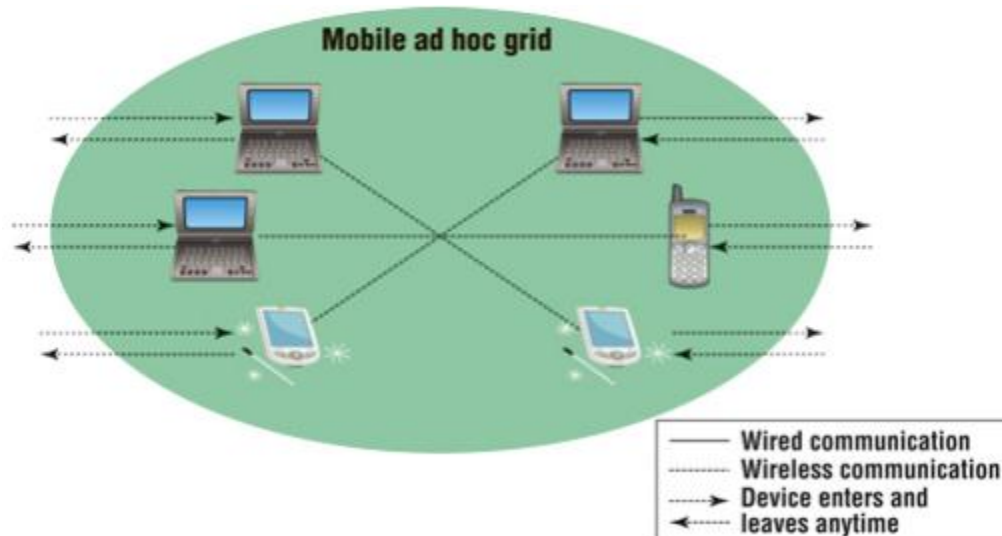


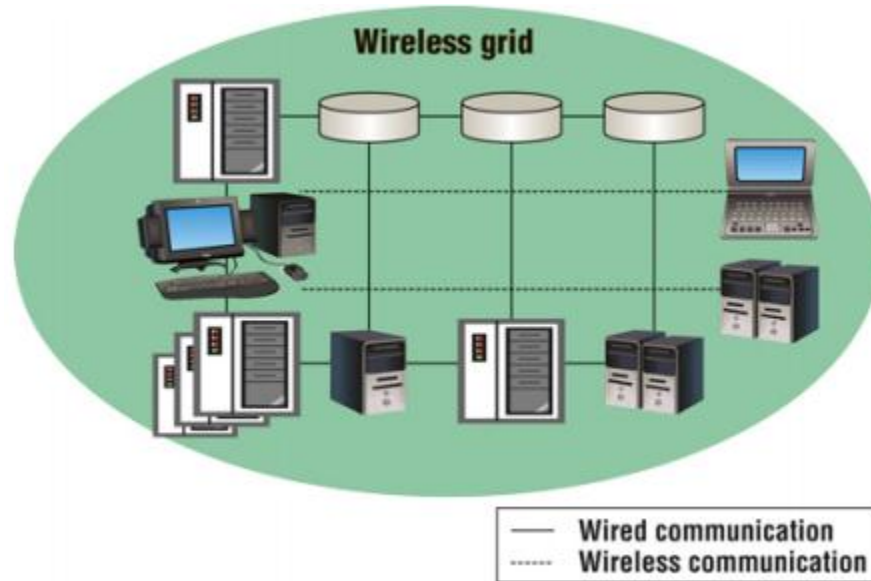
Figura 4 Grid Ad hoc Móvil. Heba Kurdi, Maozhen Li, and Hamed Al-Raweshidy (2008)

Una Grid ad hoc, como se representa en la Figura 3, es la organización espontánea de nodos heterogéneos cooperativos. Esta organización es una comunidad lógica sin infraestructura pre-configurada (Amin, Laszewski, & Mikler, 2004). También se define como un ambiente



de Grid sin infraestructura, en el que todos sus miembros son dispositivos móviles y pertenecen a distintas organizaciones, aunque esta estructura toma nombre de Grid Ad hoc Móvil, como se representa en la Figura 4 para otros autores (Kurdi, Maozhen, & Al-Raweshidy, 2008). Su principal reto es la topología dinámica que tiene, debido a los cambios que se exponen en [IV.1.3 Grid Accesible](#).

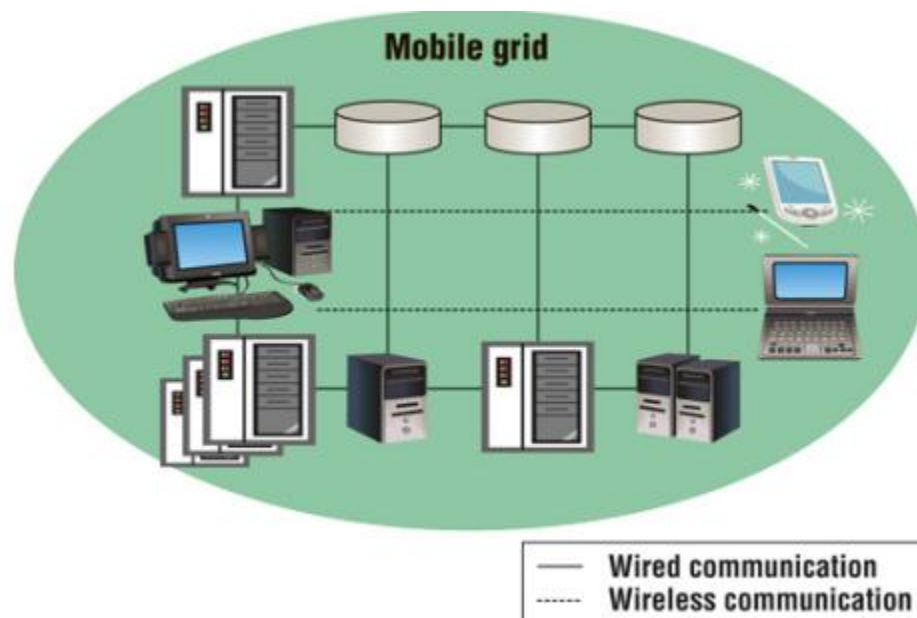
### 1.5. Grid Inalámbrica (Wireless Grid)



**Figura 5 Grid Inalámbrica. Heba Kurdi, Maozhen Li, and Hamed Al-Raweshidy (2008)**

Son aquellas Grids que entre sus miembros poseen dispositivos inalámbricos, como se observa en la Figura 5. Se hace la diferencia entre la Grid wireless y la Grid de acceso wireless. La Grid wireless es aquella que conecta recursos de manera inalámbrica, los cuales pueden ser fijos o móviles (Kurdi, Maozhen, & Al-Raweshidy, 2008). La Grid de acceso wireless presenta los dispositivos inalámbricos únicamente como dispositivos de acceso (Srinivasan, 2005). Los problemas a los que se enfrenta este tipo de Grid provienen de las características de las redes inalámbricas: ancho de banda menor que en par trenzado; vulnerabilidad de seguridad, ya que se da un acceso privilegiado a una entidad externa; latencia mayor a la que se encuentra con enlaces de par trenzado; choques de paquetes debido a la naturaleza de las ondas (Junseok & Aravamudham, 2004).

## 1.6. Grid Móvil



**Figura 6 Grid Móvil. Heba Kurdi, Maozhen Li, and Hamed Al-Raweshidy (2008)**

En este caso también existe una situación similar a las Grid wireless, es decir, se presentan las Grids móviles, y las Grids de acceso móvil. Las primeras tratan de usar dispositivos móviles como recursos computacionales, extendiendo de una Grid tradicional (Ohta, Yoshikawa, Nakagawa, & Inamura, 2005). En el segundo caso se trata de una Grid a la que se accede por medio de un dispositivo móvil (Sajjad, Jameel, Kalim, Lee, & Lee, 2005). Se han propuesto distintas técnicas para establecer este tipo de Grids; entre los enfoques se encuentran: centralizado, P2P, a través de agentes móviles, usando un middleware, entre otras (Kurdi, Maozhen, & Al-Raweshidy, 2008).

## 1.7. Middleware

En español Capa Media, es un componente de Software cuya responsabilidad yace en permitir la integración entre, al menos dos, dispositivos que originalmente no se pudieran comunicar. Tomando un lugar entre las partes involucradas, ofrece interfaces de comunicación a todas las partes y se encarga que exista una comunicación (Bernstein, 1996 ; Román, y otros, 2002).

## 1.8. MPI

La interfaz para el paso de mensajes (Message Passing Interface) es una especificación de interfaz para una librería de paso de mensajes. Su elaboración inicia en 1993 y se produce una

primera versión en 1994, se crea para facilitar la transferencia de datos de un proceso a otro en sistemas paralelos (MPI Forum, 2015).

### **1.9. Red Ad hoc**

Una red ad hoc es una red, que se construye por demanda entre dispositivos heterogéneos con el objetivo de compartir información entre ellos. Se forma sin un ente central que gestione la comunicación y sin infraestructura pre-existente (Perkins, 2008).

### **1.10. Compilación Cruzada**

Es el proceso de utilizar un cros-compilador, el cual es un compilador capaz de generar un ejecutable para una plataforma distinta a en la que se encuentra. En este texto hace referencia al proceso realizado para generar un ejecutable para la arquitectura ARM, desde una arquitectura Intel (GNU, 2014).

## **2. Trabajos relacionados**

A continuación se expondrán varios trabajos relacionados.

### **2.1. Estado actual**

Como se menciona en el capítulo anterior, el término Grid Accesible es acuñado por Kurdi, Maozhen y Al-Raweshidy (2008), donde existen varios tipos como las Grids Móviles y Grids Ad hoc (Kurdi, Maozhen, & Al-Raweshidy, 2008). En esos tipos se utilizan los dispositivos móviles como recursos para la ejecución de tareas, lo que conlleva múltiples desafíos debido a sus características: heterogeneidad, capacidad de CPU, vida corta de la batería, movilidad, conexión intermitente. Adicionalmente las redes inalámbricas se caracterizan por su bajo ancho de banda, baja confiabilidad y latencias altas. De igual forma, aunque no se debe directamente a los dispositivos, la topología de la red se torna dinámica debido a los dispositivos que entran y salen de forma impredecible. Sin embargo su inclusión es un tema que se ha venido estudiando, produciendo protocolos, middlewares y sistemas para solucionar los problemas inherentes y poder explotar efectivamente los recursos móviles (Vidal Aroca & Garcia Goncalves, 2012). La mayoría de estas incursiones se han limitado a laboratorios, pero se reportan resultados satisfactorios. Y este trabajo se adelanta a los crecimientos, tanto en capacidad como en número, que se pronostican para los móviles (ABI Research, 2014; TNS Infratest Germany, 2015).

### **2.2. Trabajos relacionados en Grids Accesibles**

A continuación se detallan varios proyectos de Grid Accesibles existentes en la literatura.

**a. Angels**

Basándose en las visiones de IoT, Angels propone utilizar los recursos de los dispositivos a nuestro alrededor, que denomina dispositivos borde (*Edge Devices*), para procesar de manera oportuna la información generada por IoT. Presentado como un Framework, su funcionamiento se basa en dividir las tareas en partes según el número y las propiedades de los recursos disponibles, realizado por un particionador (*Partitioner*). Luego de dividida se le envía una parte a cada dispositivo, el planificador asigna la tarea a los recursos. Cuando las diferentes partes de la tarea terminan se procede a recombinar las partes de la tarea inicial, gracias a un combinador (*Combiner*) (Datta, Dey, Paul, & Mukherjee, 2014).

**b. Akogrimo**

Proyecto financiado a través de CORDIS, el sistema de información para investigación y desarrollo de la comunidad, cuenta con participación de organizaciones de toda Europa. Tiene el objetivo de desplegar una Grid Móvil formada de recursos móviles y fijos heterogéneos para la solución de problemas complejos al acceso de todos utilizando los dispositivos móviles (Community Research and Development Information Service, 2002). Busca integrarse con redes de siguiente generación (NGN) para lograr una Grid de siguiente generación (NGG) logrando que se adapte a ambientes volátiles, con cambios en el constantes en el contexto de la Grid. (Jaehnert, Wesner, & Villagra, 2007). Se implementó en sistemas de salud, manejo de desastres y educación, puede estar compuesta por elementos inalámbricos y móviles, teniendo presente su movilidad (Gang, Hongmei, Huahao, Haiyan, & Yue, 2009). No existe información actualizada del proyecto ya que el portal web que se registra en CORDIS dejó de existir.

**c. CWC**

CWC surge por el deseo de explotar los recursos computacionales inutilizados de los smartphones, especialmente mientras su dueño está durmiendo. Se define como una infraestructura de computación distribuida utilizando smartphones. CWC realiza distintas labores para mejorar su desempeño: un perfilamiento del comportamiento de los usuarios, para optimizar la asignación de tareas según las horas a las que conectan el dispositivo para cargar la batería; asignación de tareas optimizadas según las propiedades del dispositivo y perfil del usuario; y migración de tareas entre dispositivos, para asegurar una rápida ejecución de las tareas, si un dispositivo no puede terminar la tarea, esta se asigna a un nuevo recurso (Arslan, y otros, 2015).

**d. Ibis**

Ibis es un software para computación distribuida escrito en Java y creado por la *Vrije Universiteit Amsterdam*, o Universidad VU de Ámsterdam en español. Sus creadores se fijaron como objetivo principal simplificar el proceso de implementación y despliegue

necesarios para aplicaciones distribuidas. Intentan automatizar, para los desarrolladores, el uso de los recursos de la Grid formada por Ibis (Ibis, 2011).

Se compone de múltiples módulos, los cuales pueden ser usados de manera independiente. Por usar el lenguaje Java, se puede ejecutar en múltiples plataformas de hardware y sistemas operativos. Todo el proyecto está bajo una licencia que hace su uso gratuito (Ibis, 2011).

Al ser desarrollado en Java se pudo adaptar fácilmente un cliente móvil para Android. Se formó una Grid usando teléfonos móviles, de dos maneras: como recurso computacional, y como portal para enviar tareas a la Grid. (Palmer, Kemp, Kielmann, & Bal, 2009).

**e. MADAM**

MADAM (*Mobility and AdAptation Enabling Middleware*), es un proyecto coordinado por el Simula Research Laboratory de Noruega y cuenta con miembros de toda Europa. Inicia en 2004 buscando crear un middleware que ayude a la formación de Grids móviles y ubicuas, solucionando los problemas de heterogeneidad de los dispositivos y la naturaleza dinámica de ellos (Eliassen, 2005).

**f. MiPeG**

MiPeG (*Middleware infrastructure for Pervasive Grids*) por sus siglas en inglés, presenta un middleware para Grid Ubicuas. Consiste en cuatro servicios básicos para integrar dispositivos móviles, proveer contexto, gestionar sesiones de usuarios móviles, y distribuir/ejecutar tareas en los dispositivos. Se ha desplegado de manera experimental, donde los recursos miembros ofrecen servicios a toda la Grid, como sensores (Coronato & De Pietro, 2008).

**g. Mobile OGSINET**

Extiende una implementación de OGSINET para construir una Grid Móvil, teniendo en cuenta las deficiencias de los dispositivos móviles, como sus recursos limitados y la intermitencia en la conectividad (Chu & Humpfrey, 2004). OGSINET, o infraestructura abierta de servicios de Grid, por sus siglas en inglés, define mecanismos para la creación, gestión, e intercambio de información entre entidades llamadas servicios de Grid. Los servicios de Grid se construyen utilizando servicios web, y se acomodan a comportamientos establecidos por el OGSINET (Tuecke, y otros, 2003).

**h. MoGrid**

MoGrid es una implementación de una Grid Inalámbrica Ad hoc. Es una arquitectura de Middleware para coordinar tareas distribuidas entre dispositivos móviles de manera P2P. En la arquitectura cualquier dispositivo móvil puede proponer tareas, las cuales serán distribuidas a los colaboradores, o recursos, que también pueden ser dispositivos móviles. Es concebida al unir dos proyectos: el MoCa, una arquitectura de Middleware para colaboración

entre dispositivos móviles con una infraestructura central; y InteGridade, una infraestructura Grid enfocada en la eficiencia del uso de recursos, utilizando también un gestor central. (Lima, y otros, 2005; GIGA Project, 2005; InteGridade Project, 2005).

**i. MORE**

MORE es un middleware para adaptar servicios web a sistemas embebidos, facilitando la construcción de sistemas de información móviles. Combina los servicios web con los sistemas embebidos para facilitar la integración de dispositivos heterogéneos. MORE también soporta que estos pequeños dispositivos formen grupos escalables de comunicación, los cuales se conectan por medio de redes inalámbricas (Wolff, Michaelis, Schmutzler, & Wietfeld, 2007).

**j. OurGrid**

Es un Middleware de código abierto desarrollado para crear Grids P2P. Está escrito en Java y tiene cuatro componentes: el gestor (Broker), el par (Peer), el trabajador (Worker) y el servicio de descubrimiento (Discovery Service). El gestor es el usuario que accede a la Grid para utilizar los recursos, monitorear y registrar tareas. El trabajador es el nodo que comparte sus recursos para ejecutar las tareas. El par es el nodo que coordina el uso de los Workers o trabajadores. El servicio de descubrimiento se encarga de conectar varias organizaciones de gestor-trabajador-par entre sí (OurGrid, 2013). A pesar de no ser una Grid Móvil, se incluye en la selección de herramientas por la portabilidad que le brinda el lenguaje Java en el que está construido. Por estar escrito en Java se cree que pudiera ser fácil adaptar la plataforma para que funcione en dispositivos Android sin mayor contratiempo, de la misma manera que Palmer, Kemp, Kielmann, & Bal (2009) exploraron con Ibis (Palmer, Kemp, Kielmann, & Bal, 2009). Sin embargo es una Grid Accesible, ya que se define como una Grid Ad hoc.

**k. The Berkeley Open Infrastructure for Network Computing (BOINC)**

Se trata de una infraestructura de código abierto creada en la universidad Berkeley de California para computación voluntaria, donde el público general entrega voluntariamente sus recursos computacionales (Tapparello, y otros, 2015). Actualmente existe una Grid Móvil, llamada Collatz Conjecture, que está desplegada usando BOINC y cuenta con un poder de cómputo aproximado de 9,000 TeraFLOPS (BOINC, 2014; BOINC, 2015). Esta Grid utiliza los recursos del público voluntario para investigaciones en el ámbito matemático (Sonntag, 2015). BOINC funciona de la siguiente forma: el público descarga un cliente a su dispositivo, el cual ejecutará tareas, actuando así como recurso de una Grid. Este cliente está implementado para varios tipos de dispositivos, principalmente computadores de sistema operativo Windows, pero también para dispositivos con Sistema Operativo Android, convirtiéndola así en una Grid móvil. Sobre esta infraestructura también se construyó el proyecto SETI@Home. BOINC es un proyecto de código abierto, que está bajo la licencia

de código libre *GNU Lesser General Public License* Ver.2.1 o superior. En el capítulo [VI. DESPLIEGUE DE LA GRID SELECCIONADA](#) se explica en mayor detalle.

### **2.3. Grids integrando recursos móviles**

Aunque los sistemas mencionados en la sección anterior son todos Grids Accesibles, no todos tienen en cuenta los recursos móviles. Por ejemplo el sistema OurGrid no incluye a los dispositivos móviles, sin embargo es una Grid Accesible. También se encuentran trabajos en que se simulan computadores independientes como dispositivos móviles, en aras de evitar lidiar con las limitantes de estos dispositivos desde un ambiente de laboratorio. Al igual que trabajos que solo evalúan la posibilidad de incluir dispositivos móviles.

El objetivo del proyecto es desplegar la Grid más adecuada para los propósitos actuales y que pueda ser fácilmente modificada para aplicaciones futuras. Se decide comenzar con una Grid existente, a diferencia de lo que hacen varios de los autores, quienes desarrollan su propia infraestructura. Se selecciona una Grid para aprovechar los avances hechos por estos sistemas en el uso de recursos computacionales distribuidos para la ejecución de tareas.

## V. ANÁLISIS Y SELECCIÓN DE LA GRID

En este capítulo se evaluarán varias herramientas existentes para la implementación de una Grid Accesible, y se compararán utilizando la metodología DAR. El resultado de la comparación será la herramienta seleccionada para realizar el despliegue presentado en el capítulo [VI. DESPLIEGUE DE LA GRID SELECCIONADA](#).

### 1. Primera Etapa

Inicialmente se realizó una revisión bibliográfica para identificar todas las posibles implementaciones de Grids Accesibles. Muchas de las herramientas descritas en el capítulo anterior se descartaron, principalmente debido a que no era posible acceder a sus implementaciones. Los hallazgos encontrados se mencionan a continuación:

- Akogrimo: el portal del proyecto (<http://www.mobilegrids.org>) ya no contiene información de éste, al sincero conocimiento del autor.
- MiPEG no cuenta con una implementación pública. Lo que se conoce del proyecto es lo reportado por Coronato, Antonio y Pietro, Giuseppe (2008).
- MORE se encuentra en una situación similar, se publica el trabajo de Wolff y otros (2007), pero no se expone el desarrollo al público.
- Mobile OGSINET realiza una implementación de una Grid Móvil basada en los mecanismos de OGSINET, sin embargo esta implementación no se encuentra pública.
- MADAM como lo define Frank Eliassen (2005) también se queda únicamente en la publicación.
- CWC como lo definen Mustafa Arslan y otros (2015) se encuentra reservado únicamente para los miembros del grupo de investigación y sus estudiantes inscritos (JP Infotech, 2015).
- Angels como lo define Pubali Datta y otros (2014) no se encuentra expuesto al público.

### 2. Segunda Etapa

Las herramientas que se revisan en esta segunda etapa son aquellas cuyo código fuente estaba disponible: **BOINC**, **Ibis**, **MoGrid** y **OurGrid**. Para evaluar estas herramientas se utilizará la metodología DAR (The Process Group, 2005). En la primera parte de la metodología se escogen los atributos más importantes.

Para seleccionar los atributos se realizaron sesiones de Brainstorming entre los stakeholders del proyecto de investigación: los ingenieros Mariela Curiel y Leonardo Florez al igual que el estudiante Roberto R. García. Durante las sesiones se discutieron los posibles usos futuros, tales como el procesamiento de imágenes médicas antes mencionado y otros aspectos que pudieran ser de utilidad en las fases siguientes de este Trabajo de Grado, como la facilidad de



despliegue; el alcance del proyecto; entre otros. Estas sesiones produjeron la lista de atributos que fueron evaluados para cada herramienta.

Inicialmente se pensó en distintos atributos de comparación que luego se eliminaron o se mezclaron o fusionaron para formar los atributos finales. Los atributos que pertenecen a este grupo son:

- Lenguaje en el que está escrito el sistema, se eliminó pero se tiene en cuenta en el atributo de facilidad de modificación.
- Estilo de ejecución de tareas, por ejemplo en el sistema BOINC el cliente debe pedir tareas para ejecución. Se eliminó ya que no se pudo encontrar esta información en todos los sistemas

Luego de seleccionados los atributos a evaluar, se dividieron en distintas prioridades. Durante el ejercicio se decidió dividir en dos grupos los atributos: unos de alta prioridad y otros de prioridad media. Donde prioridad alta son aquellos atributos que serían muy deseables si se encuentran, y los atributos de prioridad media son aquellos atributos que, aunque son deseables, no se consideran críticos. No se identificaron atributos de baja prioridad. Se decidió darle un peso cuantitativo de 5 a los atributos de máxima prioridad y de 2 a los de prioridad media para la ejecución de la metodología DAR.

Los atributos escogidos como de máxima prioridad son: ejecución en dispositivos móviles (smartphones), lenguaje de las aplicaciones que se pueden ejecutar en la Grid y mantenimiento del proyecto. Los atributos que se definieron como de prioridad media fueron: el tipo de Grid Accesible que se implementa, el soporte de tareas paralelas con comunicación a través de MPI, la calidad de la documentación, la facilidad del método de ejecución de aplicaciones, y la facilidad de modificación del sistema. A continuación se explica con mayor detalle el porqué se escoge cada uno de los atributos.

**Ejecución en móviles:** este atributo se relaciona con poder ejecutar el proyecto en dispositivos móviles. Es de alta prioridad por el interés y alcance del trabajo de grado que es la implementación de una Grid Accesible que contenga dispositivos móviles.

**Lenguaje de las tareas que se pueden ejecutar en la Grid:** este atributo tiene que ver con las limitaciones que existen en cuanto al lenguaje en el que estarán escritas las tareas o aplicaciones a ejecutar en la Grid. Esto cobra importancia ya que las tareas que se ejecutarán en la Grid no debieran ser reescritas en su totalidad por alguna limitante del lenguaje. Por otro lado, sería ideal que la Grid soportara la ejecución de tareas en C/C++, ya que en este lenguaje están escritas muchas aplicaciones científicas, además de las librerías para la manipulación de imágenes médicas. Mientras más lenguajes se soporten, mejor evaluada será la herramienta ya que permitirá la ejecución de diferentes tipos de aplicaciones.

**Mantenimiento del proyecto:** es importante saber en qué estado está el proyecto: si sigue en desarrollo, si ha sido desplegado en ubicaciones distintas al laboratorio o si ha logrado una versión de distribución al público y lo siguen actualizando, por lo que existe una comunidad que ha puesto a prueba el software. Estas últimas características garantizan un software más completo, libre de errores y la existencia de una comunidad que puede dar soporte en el caso de problemas. Un proyecto que ha sido abandonado no presenta estas ventajas, ni estará incorporando nuevas innovaciones tecnológicas. .

**Tipo de Grid:** este atributo está relacionado con el tipo de Grid Accesible que se implementa: Móvil, Ad hoc o Inalámbrica. Cobra importancia porque este trabajo es la primera parte de un proyecto de investigación más ambicioso cuyo objetivo es la formación de una Grid Ad hoc para el procesamiento de imágenes. Sin embargo, aunque una Grid Ad hoc sería el caso perfecto, se pudiera pensar en esta meta a mediano plazo y comenzar desplegando una Grid Móvil formada por dispositivos móviles y fijos.

**Uso de MPI:** el uso de MPI se incluye debido a uno de los propósitos que se le planea dar a la Grid en trabajos posteriores: el procesamiento de imágenes médicas. Los algoritmos que procesan imágenes en paralelo utilizan MPI. No obstante, estos algoritmos son del esquema Maestro-Esclavo y el MPI se utiliza para repartir los datos entre los nodos (esquema SIMD: single instruction multiple data). Como esta es una de las tareas que realiza BOINC, se tendría que estudiar en esos trabajos posteriores cómo portar estos algoritmos a la Grid.

**Calidad de la documentación:** para tener éxito en el despliegue de la Grid y futuras modificaciones, es importante la calidad de la documentación encontrada, es decir si existen tutoriales, wikis, documentación del código, entre otros.

**Ejecución de aplicaciones:** este atributo está relacionado con la facilidad de ejecutar una aplicación en la Grid, donde una aplicación se compone de múltiples tareas que serán ejecutadas en recursos. Mientras más fácil sea el uso de la Grid, existirán más usuarios interesados.

**Facilidad de modificación del Sistema:** este atributo se refiere a qué tan compleja es la estructura del sistema, y su documentación. Se tiene en cuenta ya que es probable que se modifique el sistema para cumplir con las necesidades del proyecto de Investigación del que forma parte este trabajo, o de otros proyectos de investigación de la Facultad de Ingeniería o de la Universidad.

## 2.1. Matriz comparativa

En esta sección se presentan el proceso y los resultados de aplicar la metodología DAR después de realizar las sesiones de brainstorming, seleccionar los atributos y sus prioridades se procedió a organizar los cuatro sistemas según cuál cumplía de mejor manera cada

atributo, de acuerdo a su definición. Al más alto se le da el indicador 4, y al más bajo 1. Por último se calcula el puntaje total para cada herramienta sumando la multiplicación de coeficiente e indicador. Este proceso es el definido por el DAR (The Process Group, 2005).

Atributos	BOINC	Ibis	MoGrid	OurGrid
Ejecución en dispositivos móviles	4	3	2	1
Lenguaje de las tareas que se pueden ejecutar en la Grid	4	3	3	2
Mantenimiento del proyecto	4	2	1	3
Tipo de Grid	3	2	4	1
Uso de MPI	3	4	2	2
Calidad de la documentación	4	1	2	3
Ejecución de aplicaciones	4	1	2	3
Facilidad de modificación del sistema	1	4	2	3
<b>Total</b>	90	64	54	54

**Tabla 1 Matriz de comparación DAR**

En la tabla 1 se puede observar que la herramienta que obtuvo la más alta puntuación fue BOINC (BOINC, 2013), por esta razón se decide continuar con la siguiente fase metodológica utilizando el software **BOINC**.

## 2.2. Justificación

En los siguientes subsecciones se presentarán los argumentos que explican la puntuación asignada a cada herramienta para cada uno de los atributos:

### a. Ejecución en dispositivos móviles

Debido al objetivo del proyecto: la implementación de una Grid Accesible con dispositivos móviles como proveedores de recursos, la ejecución en este tipo de dispositivo, era fundamental en cualquier decisión que se tomara.

BOINC toma el primer puesto, ya que incluye una implementación de computación voluntaria que está actualmente en funcionamiento en el mundo con decenas de miles de instalaciones en Android (Fritsch, 2014). Los dispositivos le pertenecen al público que quiere participar en el proyecto. Para minimizar el consumo de recursos como batería y transmisión de datos, el dispositivo sólo entrará como recurso a la Grid cuando se encuentra cargando batería y conectado a una red Wi-Fi (BOINC, 2014).

Ibis toma el segundo lugar, ya que todos sus módulos están implementados en Java. Valiéndose de esta propiedad se ha explorado llevar las partes principales del sistema a

Android. Aunque los resultados reportados fueron positivos (Palmer, Kemp, Kielmann, & Bal, 2009) , al momento de desarrollar este trabajo, no se observó código disponible para Android en el portal de Ibis.

MoGrid toma el tercer puesto, porque el proyecto al día de hoy no cuenta con clientes móviles disponibles para uso público. Aunque existen artículos que presentan el despliegue y pruebas de MoGrid con dispositivos móviles (Lima, y otros, 2005; Gomes, Ziviani, Lima, Endler, & Chelius, 2008), éstas probablemente fueron realizadas en entornos controlados (ambientes de laboratorio). En el código que cedieron los autores no se observó una implementación para dispositivos móviles.

OurGrid toma el último puesto ya que no cuenta con un cliente móvil implementado y tampoco se reporta su uso en dispositivos móviles. Sin embargo, al estar construido en Java y por su arquitectura de P2P sería posible adaptar un cliente móvil a partir de los componentes que ya existen (OurGrid, 2013).

#### **b. Lenguaje de las tareas que se pueden ejecutar en la Grid**

Las limitaciones que existan con respecto al código que se ejecutará en la Grid son importantes, ya que se desea tener la mayor libertad en las tareas que pueden ser desplegadas. Es importante también que permita la ejecución en C++ por el plan futuro de procesar imágenes médicas.

BOINC toma el primer lugar por su estructura de wrappers. La función de los wrappers es enlazar un ejecutable con un empaque o envoltorio que permitirá su ejecución en la plataforma objetivo (BOINC, 2015). De esta forma hay tanta libertad en los lenguajes de programación como wrappers existan. Actualmente existen wrappers para C/C++, Java y Python (BOINC, 2012; BOINC, 2012). Para Fortran existe un método de convertir el código a C y luego tratarlo como tal (BOINC, 2015).

OurGrid toma segundo lugar, ya que en sí mismo no limita ni soporta lenguajes. La ejecución de tareas se realiza por medio de líneas de comandos que se ejecutarán en los trabajadores. Los sistemas operativos limitan los comandos que se pueden enviar, y las tareas que se pueden ejecutar. Las particularidades de la forma de ejecutar líneas de comando las brinda el programa trabajador en cada recurso. Si el comando a ejecutarse tiene restricciones, por ejemplo funcionará en máquinas con sistema operativo Linux, se puede configurar en la definición de la tarea (OurGrid, 2013).

Ibis y MoGrid empatan en el tercer lugar, porque sólo soportan tareas escritas en Java. (Ibis, 2011; Gomes A. T., 2015).

### **c. Mantenimiento del proyecto**

El mantenimiento del proyecto es importante debido a que si ya fue abandonado será difícil obtener respuestas ante posibles dificultades en la implementación y/o despliegue de la Grid. Por otro lado un proyecto que no se actualice no se adaptará al surgimiento de nuevas tecnologías. Un proyecto abandonado no tiene nadie encargado de él, por tanto en casos extremos podría llegar a perderse.

BOINC tomó el primer lugar debido al trabajo que se hace sobre él. El último hito fue una conferencia, BOINC:FAST 2015 que se llevó acabo en septiembre de ese año. Se ha formado una comunidad alrededor de BOINC que se ha involucrado y ha enriquecido el proyecto. Parte del interés generado puede deberse al proyecto SETI@home que utiliza BOINC para formar una Grid de computación voluntaria, cuyos nodos colaboran para procesar señales en busca de vida extraterrestre (Anderson, Cobb, Korpela, Lebofsky, & Werthimer, 2002).

OurGrid toma el segundo lugar dado que su última versión data de finales de 2013 (OurGrid, 2013). El proyecto, al unir los recursos computacionales de múltiples laboratorios pequeños y medianos, tiene una comunidad participando y mejorando el producto (OurGrid, 2013).

Ibis toma el tercer lugar, debido a que la última actualización que se hizo al proyecto fue en 2012, y el último hito ocurrió en 2014 que fue la entrega del premio Euro-Par Achievement Award 2014 al profesor Henri Bal, miembro del proyecto. (Ibis, 2014).

MoGrid toma el último puesto, debido a que el proyecto no cuenta actualmente con un portal web para informar al público. El código fuente se obtuvo después de contactar a los autores, los cuales desplegaron el sistema en un repositorio online.

### **d. Tipo de Grid**

El tipo de Grid Accesible que logre instalarse cobra importancia por el interés futuro en procesar imágenes en una Grid Móvil Ad hoc

MoGrid toma el primer puesto, debido a que es el único sistema que plantea una Grid tipo ad-hoc (Lima, y otros, 2005).

BOINC toma el segundo lugar, porque no se plantea una Grid ad-hoc, dado que su despliegue necesita de un gestor central (BOINC, 2013). No obstante, permite claramente implementar una Grid móvil porque dentro del proyecto se ha desarrollado un cliente móvil para dispositivos Android que estén en uso alrededor del mundo (BOINC, 2014).

Ibis toma el tercer lugar, y aunque posee una estructura similar a la de BOINC, en cuanto a que se necesita un gestor central, no cuenta con un cliente móvil desplegado; sólo se reportan

pruebas con dispositivos móviles en laboratorio como se mencionó anteriormente (Palmer, Kemp, Kielmann, & Bal, 2009).

OurGrid toma el último lugar ya que en su implementación actual no soporta recursos móviles, sólo una estructura P2P (OurGrid, 2013).

#### **e. Uso de MPI**

Se incluyó el soporte de la librería MPI como atributo secundario, por el posible uso futuro para procesamiento de imágenes. Los resultados de este atributo en los distintos sistemas se exponen a continuación:

Para este atributo Ibis resulta la mejor opción, debido a que ofrece MPJ. La cual es una implementación en Java de MPI, que ofrece alto rendimiento y se puede desplegar en varias plataformas, desde clústeres hasta Grids. En la documentación no hay referencia al uso de MPI en dispositivos móviles (Ibis, 2009; Bornemann, van Nieuwpoort, & Kielmann, 2005).

BOINC se decide como la segunda mejor opción debido a que sí implementa MPI pero con limitaciones importantes. Todo proceso que necesite comunicarse por medio de MPI deberá ser ejecutado en un mismo nodo, por lo que se pierde las ventajas de la computación distribuida (BOINC, 2011). Para lograr la implementación se utiliza la librería MPICH2, una implementación portable y de alto desempeño del estándar MPI (MPICH, 2012).

OurGrid y MoGrid toman el 3er puesto, ya que no implementan MPI de ninguna forma. (OurGrid, 2013; Lima, y otros, 2005).

#### **f. Calidad de la documentación**

La calidad de la documentación de la herramienta toma importancia para resolución de posibles problemas que surjan en la siguiente fase metodológica de este trabajo: el despliegue de la herramienta.

BOINC toma el primer lugar, ya que cuenta con una wiki con información tanto general como detallada sobre el sistema. Entre la documentación se encuentran guías de despliegue y ayudas para lograr una implementación de BOINC en un corto tiempo (BOINC, 2013).

OurGrid toma el segundo lugar ya que su portal de información contiene documentación general del proyecto al igual que guías para despliegue, instalación y uso (OurGrid, 2013). No obstante las guías no son muy detalladas.

MoGrid toma el tercer lugar debido a que solo cuenta con una serie de pasos para el despliegue del prototipo (Gomes A. T., 2015).

Ibis toma el último lugar, debido a que la documentación que presenta en los distintos módulos no es muy detallada. No presenta guías para el despliegue (Ibis, 2011).

No se tuvo en cuenta la documentación del código, ya que ésta se evalúa en otro atributo.

#### **g. Ejecución de aplicaciones**

Para su uso posterior, la herramienta seleccionada debe ser fácil de usar.

BOINC toma el primer lugar, debido a la extensa documentación, ejemplos y guías que existen al respecto (BOINC, 2013). La ejecución de aplicaciones se facilita debido a la documentación de archivos de configuración detallada y completa. Apoyándose en esta documentación los usuarios pueden de manera más fácil elaborar los archivos de configuración para la ejecución de aplicaciones, establecer qué tipos de recursos realizan las tareas, determinar archivos de entrada y salida, entre otras.

OurGrid toma el segundo lugar. Aunque cuenta con menos documentación, y menos detalles, de igual forma se transmite el conocimiento mínimo necesario para realizar la ejecución de comandos (OurGrid, 2013). De manera similar a BOINC, se debe elaborar un archivo de configuraciones.

MoGrid toma el tercer lugar debido a que, aunque la ejecución de las tareas es simple – se entrega la ubicación de ejecutables Java por medio de un script- no hay archivos que describan en detalle el proceso de ejecución de aplicaciones (Gomes A. T., 2015).

Ibis toma el último lugar. La ejecución de aplicaciones o tareas es más difícil debido a que existe más de una opción para hacerlo y hay poca documentación sobre estas opciones. Las dos formas de ejecución existentes son: 1) Se pueden desplegar aplicaciones con o sin dependencias, utilizando comandos, como lo hace OurGrid; y b) Ejecutables Java (Ibis, 2011).

#### **h. Facilidad de modificación del sistema**

La segunda fase metodológica requiere del despliegue de un prototipo, y en esta actividad puede que se realicen modificaciones del código fuente. De igual forma, en un futuro se podrá ampliar el sistema para ajustar la Grid a las necesidades de otros proyectos de Investigación.

Ibis toma el primer lugar para este atributo debido a su estructura modular que facilita encontrar más rápidamente las zonas de código a modificar y la depuración del código también se hace más sencilla. Adicionalmente, los proyectos están escritos en Java y cuentan con javadoc de acceso público (Ibis, 2011).

OurGrid toma el segundo lugar debido a que no cuenta con un documento javadoc, aunque las anotaciones en el código existen. También divide las responsabilidades entre los distintos componentes, por lo que se puede identificar con facilidad el punto exacto a modificar, y el cambio estará encapsulado en ese componente específico (OurGrid, 2013).

MoGrid toma el tercer lugar, y aunque no cuenta con un documento javadoc, las anotaciones en el código existen (Gomes A. T., 2015).

BOINC toma el último lugar, ya que por estar implementado en varios lenguajes (C, C++, Java, PHP, entre otros), la documentación del código es más compleja. Por esa misma razón realizar cualquier cambio puede tornarse sumamente complejo dependiendo del componente que se vaya a modificar. Por ejemplo un cambio del cliente sería necesario replicarlo en las distintas implementaciones del cliente (Windows, Linux, etc) (BOINC, 2013).



## VI. DESPLIEGUE DE LA GRID SELECCIONADA

En este capítulo se explica en detalle la herramienta seleccionada en la fase anterior, BOINC. Se explican en detalle todos sus componentes y la estrategia utilizada para el despliegue.

BOINC es una infraestructura para la computación voluntaria distribuida, desarrollada originalmente en la Universidad de Berkeley, sobre la cual se basan proyectos como SETI@home. Actualmente hay otros proyectos usando BOINC en diversos campos como la física, medicina nuclear, climatología, astronomía, etc. (BOINC, 2014). BOINC es un medio que aprovecha una capacidad enorme de cómputo, utilizando, inicialmente, computadores personales alrededor del mundo y posteriormente también el poder de procesamiento de dispositivos móviles (BOINC, 2014).

BOINC está diseñado para ser una plataforma libre para cualquier persona o institución que quiera crear un proyecto de computación distribuida, y ese es el uso que se le dará en la institución.

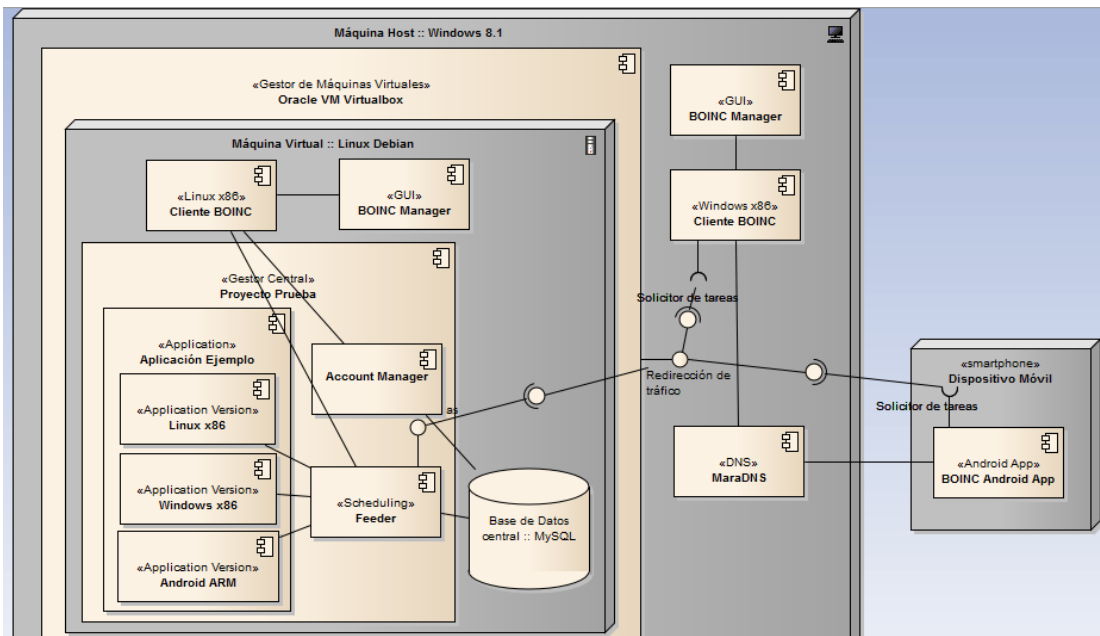
BOINC contiene dos componentes principales: el proyecto, o gestor central; y el cliente. El gestor representa un único proyecto, por ejemplo SETI@home, y cada recurso puede ser parte de uno o más proyectos, cada recurso es un dispositivo ejecutando el cliente BOINC. Sobre los proyectos se despliegan Aplicaciones (Applications) que definen el esqueleto de las tareas que se delegarán a los recursos. En la Figura 7 se puede observar el gestor y un usuario administrador de la Grid; distintos tipos de máquinas, todas brindando sus capacidades de cómputo por medio de un cliente BOINC; por último un usuario que configura cómo se comparten sus recursos por medio de la interfaz gráfica BOINC Manager. Para elaborar el diseño de despliegue se aplica la metodología PPDIOO (Tiso & Teare, 2011):

### 1. Proceso PPDIOO

El método se basa en realizar las acciones: Preparar, Planear, Diseñar, Implementar, Operar y Optimizar de manera iterativa al interior de una organización. Aunque el método fue diseñado para el diseño y gestión de redes de comunicación se utilizó en este caso debido a que se preveía que los problemas ajenos al uso y gestión del sistema BOINC estarían relacionados con la red desplegada al tiempo. En mayor detalle las fases son las siguientes:

- a) **Preparar:** en esta fase se definen los requerimientos de negocio del despliegue la organización, en este caso la organización se define como este Trabajo de Grado. Se tomaron en cuenta los requerimientos: el despliegue debe ser fácil de replicar, o mejor aún, portátil; y el despliegue debe tener en cuenta conectividad con dispositivos móviles.
- b) **Planear:** en esta fase se define un plan de proyecto, determinando los recursos que serán necesarios en el diseño e implementación. Debido a los requerimientos, se

- determinan los recursos necesarios, que son básicamente a) tecnologías de máquina virtual, ya que se puede generar una máquina virtual pre-configurada que facilite replicar los resultados de este Trabajo de Grado; y, b) un switch inalámbrico, para conectar los componentes en una red local.
- c) **Diseñar:** en esta fase se define el diseño detallado del despliegue a realizar. El diseño detallado se puede observar en la Figura 8.
  - d) **Implementar:** en esta fase se implementa el diseño realizado en fases anteriores. La preparación para el despliegue se puede observar en el capítulo [VI.3. Preparación para el Despliegue](#) y la implementación detallada en [VI.4 Proceso de Despliegue](#). El despliegue se realiza en tres etapas: primero se despliega BOINC a nivel local en la máquina virtual, luego se agregan los recursos de la máquina host y por último se agregan los dispositivos móviles.
  - e) **Operar:** en esta fase, como su nombre lo indica, se opera el sistema desplegado en un contexto de día a día.
  - f) **Optimizar:** en esta fase, a partir de lo observado en la operación del sistema se realizan pequeñas optimizaciones. También se evalúan los requerimientos de negocio originales y se asegura que no hayan cambiado, de lo contrario se inicia el proceso de nuevo.

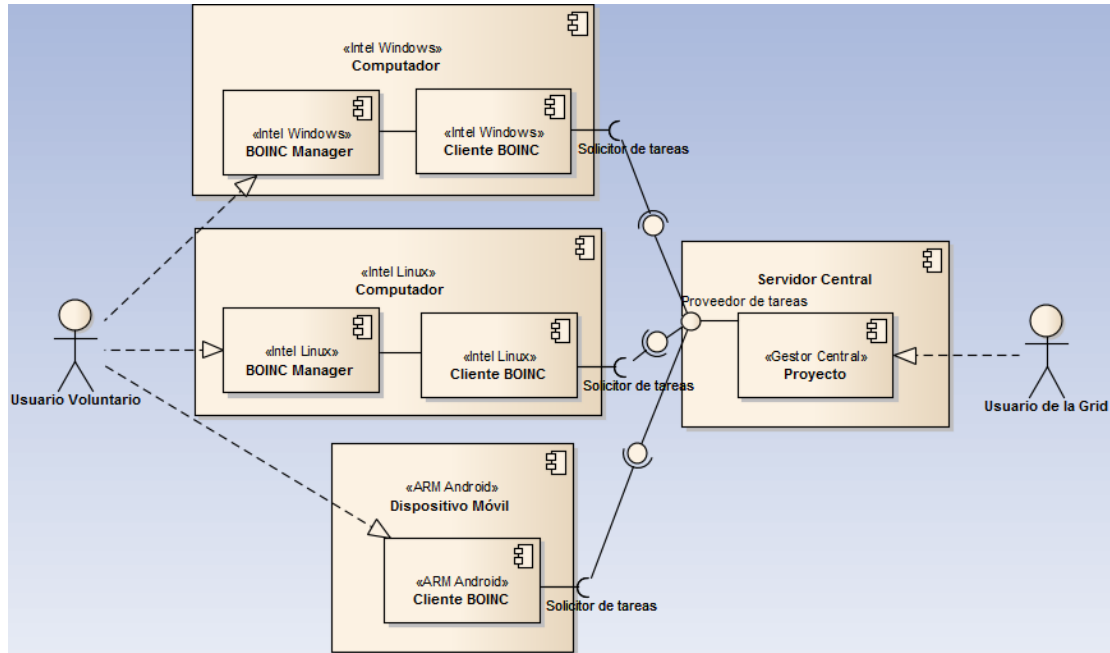


**Figura 7 Modelo de Despliegue**

En la figura 7 se presenta el modelo de despliegue, el cual fue implementado y probado.

En las siguientes secciones se explicarán las actividades relacionadas con la implementación y operación del despliegue. En el alcance del Trabajo de Grado no se realizaron labores de Optimización, pero se presentan posibilidades en la sección [IX.1. Trabajo Futuro](#).

## 2. Descripción detallada de BOINC



**Figura 8 Componentes BOINC**

**Proyecto o Gestor:** el proyecto es la pieza central de la Grid BOINC, su principal función es la de gestionar los recursos. Contiene las tareas que se deben ejecutar, y conoce a los recursos disponibles, con esta información delega tareas a los recursos y espera a que sean ejecutadas. Desde el proyecto, o gestor, se administran las Aplicaciones y sus versiones, las tareas pendientes, las completadas y canceladas, los usuarios, la asignación de recursos, etc. Se administra la cara pública de la Grid, por medio de una página web de portada, foros, etc. es el corazón latente de la Grid. Dentro del proyecto se encuentran múltiples daemons, encargados de la asignación de tareas, el envío de tareas, la limpieza de archivos y generadores de tareas.

**Cliente:** se trata del software cliente de BOINC que es ejecutado en la máquina voluntaria o en el recurso que va a realizar el trabajo. El cliente se encarga de ejecutar las tareas que le son asignadas por el gestor. Dependiendo de las propiedades de la máquina se tiene un cliente

específico, por ejemplo, se tienen clientes para Windows, para Android, para Linux, entre otros.

**Recurso:** son las máquinas o dispositivos móviles donde se ejecutan las tareas.

**Aplicación (Application):** se trata de la aplicación desplegada sobre la Grid, la cual se crea en el gestor. Cada aplicación tendrá una versión (Application Versions) por cada plataforma de ejecución (arquitectura, sistema operativo) que se quiera utilizar. Por ejemplo, para que una Aplicación utilice un recurso ARM Android deberá existir la versión de la aplicación específica para ARM Android. Se debe tener al menos una versión de aplicación, para que el gestor asigne esta Aplicación en recursos con las características adecuadas.

**Scripts y otras herramientas:** el sistema BOINC, al ser de código abierto, nos da acceso al código fuente, alojado en la plataforma de administración de código fuente github (Beer & Walton, 2015). Dentro del proyecto está, primero que todo, el código fuente para los clientes y el gestor; luego se encuentran las herramientas necesarias para instalar el gestor, por ejemplo scripts; al igual que herramientas para crear Aplicaciones, por ejemplo código ejemplo y makefiles ejemplo, etc.

**API:** el API de BOINC, definido en el lenguaje C/C++, contiene los métodos que deben ser ejecutados por una tarea para iniciar y finalizar ejecución, comunicar avances, acceder a archivos de entrada y salida, reportar errores, entre otros. Para que una Aplicación pueda ser ejecutada en la Grid BOINC debe utilizar el API o los wrappers. La definición detallada del API se encuentra en el Anexo 3: BOINC API.

**Wrappers:** o envoltorios, le permiten a los programas ejecutables no escritos para BOINC (porque no usan el API), ser desplegados en la Grid BOINC. Si están disponibles los Wrappers no es necesario reescribir, o modificar la Aplicación para ejecutarla en la Grid. El wrapper se utiliza a la hora de crear una Versión de Aplicación. El proceso de creación de una aplicación se detalla en el Anexo 4: Creación y ejecución de aplicaciones. Para ejecutar en clientes móviles no existe un wrapper, en esos casos es necesario modificar el código fuente de la aplicación utilizando el API de BOINC. Aunque el código fuente del wrapper se encuentra en el código fuente de BOINC, aún no se ha adaptado para utilizarse con dispositivos móviles. Actualmente se encuentran publicados los wrappers para las siguientes plataformas: Windows Intel x86, Windows 64 bits x86, Linux i686, Linux 64 bits x86, Apple i686, Apple 64 bits x86 (BOINC, 2015).

**BOINC Manager:** Se trata de una interfaz gráfica que permite la administración y configuración del cliente BOINC de una manera más natural para el usuario.

### 3. Preparación para el Despliegue

Para esta Fase se cuenta con un ambiente de desarrollo compuesto por los elementos que se encuentran en la Tabla 2. Para realizar el despliegue se decide utilizar el sistema operativo Linux Debian, debido a que gran parte de la documentación de BOINC está escrita específicamente para este sistema. Dada la facilidad que ofrece, tanto en instalación, como en la posibilidad de realizar una copia fácil de desplegar en los laboratorios de la Universidad, se decide preparar la máquina Debian como virtual. Los detalles específicos se encuentran en el Anexo 1: Preparación de la máquina virtual.

ELEMENTO	DESCRIPCIÓN
2x Celulares Samsung S4	Arquitectura ARM, Sistema operativo Android versión 4.2, BOINC versión 7.4.41
Servidor DNS MaraDNS	Servidor DNS desplegado sobre la máquina host, definiendo la dirección local con el nombre de host de la máquina virtual
Máquina host	Windows 8.1, 6GB de RAM, Intel i7 2360QM
Gestor de máquinas virtuales	Oracle VM Virtualbox versión 4.3.20, haciendo port forwarding por los puertos 80 y 443
Máquina virtual	Linux Debian, con servidor web apache, MySQL

**Tabla 2 Ambiente de Desarrollo**

Se instaló un sistema operativo Debian 32 bits, debido a que es el preferido por el equipo de BOINC, y gran parte de la documentación está escrita para él. Como primer paso se instalan las librerías y programas necesarios y se descarga el código fuente de BOINC. Se crean los usuarios boincadm y www-data, el primero para la ejecución de la Grid y el segundo se crea para el servidor web apache. Es necesario otorgarles los permisos de escritura, lectura y ejecución, sobre la ubicación del proyecto a ambos usuarios.

## 4. Proceso de Despliegue

Siendo BOINC un sistema para la creación de Grids, que luego se expandió para incluir recursos móviles, se decidió hacer el despliegue de esa misma manera, es decir, se prueba el despliegue primero en computadores de escritorio y luego en dispositivos móviles. Además, de esa forma el estudiante se familiarizó con el uso de la herramienta, pasando de lo más sencillo a lo más complejo. Los pasos fueron los siguientes:

- a) Desplegar el gestor, en la máquina virtual Debian.
- b) Instalar el cliente BOINC en la máquina virtual Debian. Gestor y Cliente se ejecutan en la misma máquina virtual, siendo la máquina virtual el único recurso.
- c) Instalar el cliente BOINC en la máquina host Windows. El Gestor se ejecuta en la máquina virtual y el cliente, afuera de la máquina virtual, en la máquina host. El recurso será la máquina host.
- d) Instalar el cliente BOINC en los dispositivos Android. El Gestor se ejecuta en la máquina virtual y el cliente en los dispositivos Android que son los recursos para la ejecución.

A continuación se describen cada una de las estrategias de despliegue:

### 4.1. Despliegue del Gestor Central

Para el despliegue del gestor es necesario la preparación del sistema: la compilación del código fuente en la máquina virtual y luego la instalación. Una vez compilado e instalado se puede seguir con el procedimiento de creación de un proyecto. Para crear un proyecto se debe ejecutar un script cuyo resultado es una aplicación web alojada en el servidor web de la máquina virtual. Esta aplicación web es el gestor que se ha mencionado anteriormente y que está asociado a un proyecto.

### 4.2. Cliente en la Máquina Virtual

Para agregar como recurso la máquina virtual donde reside el gestor es necesaria la instalación del cliente de BOINC, incluyendo el BOINC Manager para Debian. Luego se inscribe al recurso al proyecto local.

### 4.3. Cliente en la Máquina Host

Para agregar la máquina host es necesario realizar unas configuraciones previas. Ya que la máquina virtual está dentro de la máquina host, se debe configurar para que esté escuchando en los puertos que utilizará la Grid. Esto se logra con una redirección del tráfico desde el gestor de máquinas virtuales, de esta manera el tráfico que reciba la máquina host en esos puertos será enviado hacia la máquina virtual. Luego es necesario agregar el nombre de host

de la máquina virtual al servidor MaraDNS para asociar la dirección local con el nombre de host, para acceder a la Grid por medio de la dirección [www.gridpuj.com](http://www.gridpuj.com). Por último es necesaria la instalación del cliente BOINC; al igual que el BOINC Manager, para simplificar el proceso de inscripción del recurso a la Grid.

#### 4.4. Cliente en el dispositivo Android

Para que la Grid desplegada sea móvil se procede a agregar dos dispositivos Android suministrado por la universidad, las características de los cuales se presentan en la Tabla 2. En cada uno de los dispositivos se instaló el cliente BOINC. Este se encuentra en la Play Store de Google. Se debe tener en cuenta que este cliente requiere una versión de Android 4.1 o superior, de igual forma sólo se soportan dispositivos de arquitectura ARM, MIPS y x86.

Como se mencionó al comienzo de la sección [VI.1. Descripción detallada de BOINC](#) en un proyecto se despliega una Aplicación, y dentro de ella, se configuran diferentes versiones para varios tipos de clientes: Windows x86, Linux x86, y Android ARM.

Infelizmente para recursos Android no se tiene un wrapper que permita ejecutar de forma automática programas en C++ en la Grid Móvil. Se debe modificar la aplicación, utilizando el API de BOINC para controlar el inicio y final de la tarea, al igual que las operaciones de entrada/salida.

Una vez escrita la nueva aplicación, esta se debe compilar usando el proceso de compilación cruzada de croscompilación para la arquitectura Android deseada, en este caso ARM.

Para la compilación cruzada se debe realizar un nuevo procedimiento de preparación que consiste en los siguientes pasos:

- a) Se descarga el Android Native Development Kit. Este Kit de desarrollo contiene un conjunto de herramientas para generar un Toolchain.
- b) Una vez que se tiene el Toolchain se hace primero la compilación cruzada del código fuente de BOINC, incluyendo las librerías.
- c) Por último se realiza la compilación cruzada de la aplicación, lo cual nos produce el compilado. Con éste se crea una versión de Aplicación para Android ARM en el proyecto.

## 5. Resultado

El resultado del proceso es la Grid Accesible Móvil utilizando el sistema BOINC. La Grid se compone de: un proyecto, desplegado sobre una máquina virtual con sistema operativo Debian, actuando como gestor central; un recurso en la máquina virtual; el sistema gestor de máquinas virtuales; la máquina host que alberga el sistema gestor de máquinas virtuales; un recurso en la máquina host; dos dispositivos móviles Android de arquitectura ARM como

recursos. Y sobre esta estructura se despliega una Aplicación la cual contiene versiones para Linux (Linux Debian de la máquina virtual), Windows y Android ARM tal como Cómo se muestra en la Figura 8. Los detalles específicos del proceso se encuentran en el Anexo 2: Proceso de despliegue del gestor, y en el Anexo 3: Proceso de despliegue de una Aplicación para recursos Android.

Entre las principales limitaciones de la solución desplegada tenemos:

- No hay en la actualidad wrappers para dispositivos móviles por lo que cualquier código se tiene que modificar para incluir las instrucciones de BOINC.
- El kit de desarrollo solo tiene compiladores cruzados para Android.
- Es necesario un despliegue complejo para cualquier prueba sobre los dispositivos, principalmente por usar tecnologías de máquinas virtuales.



## VII. PRUEBAS

En este capítulo se detallarán las pruebas que se realizaron para validar el correcto funcionamiento del despliegue del sistema BOINC.

### 1. Diseño de las Pruebas para el correcto Despliegue

Para realizar la validación del despliegue descrito en el capítulo [VI. DESPLIEGUE DE LA GRID SELECCIONADA](#) se realizan pruebas incrementales, en distintos momentos. Los componentes que se probaron fueron los siguientes:

- a) Correcto despliegue de la aplicación web
- b) Inscripción y ejecución de tareas de manera local,
- c) Inscripción y ejecución de tareas con la máquina host,
- d) Inscripción y ejecución de tareas en los dispositivos móviles.

#### 1.1. Correcto despliegue de la aplicación web

Para probar el correcto despliegue de la aplicación web, que representa al proyecto o gestor central, se ingresó a ella, primero de manera local, luego desde la máquina host, y por último desde los dispositivos móviles. Esto se realiza por medio de un explorador, accediendo al nombre de host del servidor Debian y el nombre del proyecto; se utiliza una URL de la forma [http://hostname/nombre\\_de\\_proyecto](http://hostname/nombre_de_proyecto). Esta prueba servía para confirmar primero que la aplicación estuviera corriendo, y segundo que se tuviera acceso a ella. Estas pruebas son necesarias para todos los tipos de despliegue.

#### 1.2. Inscripción y ejecución de tareas de manera local

Para probar la inscripción y ejecución de tareas de manera local, en la máquina virtual Debian, se ejecutaba el BOINC Manager, que se explica en el capítulo anterior.

Desde el Manager se realiza la inscripción del recurso a la Grid desplegada, utilizando la URL utilizada en la primera prueba: [http://hostname/nombre\\_de\\_proyecto](http://hostname/nombre_de_proyecto). La inscripción marca el ingreso del recurso a la Grid. El Manager accede a esta Grid para realizar la autenticación del usuario, o registro de uno nuevo. Una vez autenticado el usuario, el Manager descarga las tareas, compatibles con la arquitectura del recurso, de la Grid para ser ejecutadas en el cliente. Al finalizar la ejecución se envía el resultado a la Grid. Con esta prueba se validan las funcionalidades de gestión de recursos, seguridad, scheduling, ejecución de tareas y reporte de resultados de la Grid.

### **1.3. Inscripción y ejecución de tareas con la máquina host**

Ya que en las pruebas anteriores se valida el funcionamiento completo de la Grid las siguientes pruebas se realizan para validar la ejecución de tareas en recursos externos, empezando en recursos fijos para luego integrar dispositivos móviles.

La siguiente prueba se realiza desde la máquina host, para validar que el flujo de información de la Grid hacia los recursos externos sea el correcto, y que la Grid siga funcionando de manera correcta. Esto se debe a que el despliegue de la Grid se realizó en una máquina virtual, y el recurso a probar se encuentra en la máquina host, por lo que se debe redirigir el tráfico de la Grid desde la máquina host hacia la máquina virtual. Este tráfico es toda la comunicación entre el recurso y la Grid, desde la autenticación hasta el reporte de resultados. Esta prueba se realiza para confirmar que la máquina host es capaz de redirigir el tráfico hacia la máquina virtual de manera correcta.

### **1.4. Inscripción y ejecución de tareas en los dispositivos móviles**

Por último se prueba que los clientes de los dispositivos móviles puedan inscribirse y ejecutar tareas. Las pruebas anteriores nos demuestran que la Grid funciona correctamente, y la máquina host es capaz de redirigir el tráfico hacia ésta. Este último caso sirve para probar las funcionalidades que se extienden hasta dispositivos móviles, al igual que el ejecutable compilado-cruzado durante el despliegue.

## **2. Validación y Resultados**

### **2.1. Resultados: Correcto despliegue de la aplicación web**

Los resultados de la primera prueba podían ser varios: correcto, parcial o fallido. En el caso correcto se proseguía con la siguiente etapa. Sin embargo se obtuvo resultado fallido por las siguientes razones:

- a) el tráfico de información no estaba llegando hasta la Grid, debido a un Cortafuegos. Para solucionar este problema se desactivó el cortafuegos de la máquina host.
- b) el gestor de máquinas virtuales no estuviera redirigiendo el tráfico. Para solucionar este problema se agregaron reglas de redireccionado de puertos al gestor de máquinas virtuales

También se obtuvo un resultado parcial: donde la portada de la aplicación web era accesible, pero con elementos de estilo faltantes, o sin la funcionalidad de login. Esto se debió

principalmente a dos razones: la primera es que varios elementos de la aplicación web se referencian de manera directa por el nombre de host, por lo que era necesario desplegar un servidor DNS que enlazara ese nombre con la dirección de la máquina host, de esta forma la máquina host y los móviles acceden por medio del nombre y no de la dirección. La segunda razón se debe al tráfico HTTPS que fluye por el puerto 443 que también debió ser redireccionado.

## **2.2. Resultados: Inscripción y ejecución de tareas de manera local**

Los resultados de la segunda prueba fueron: correcto, parcial o fallido. En el caso correcto se proseguía con la siguiente etapa.

El único resultado fallido que se obtuvo se atribuye principalmente a los permisos de los usuarios Debian, debido a que conviven múltiples usuarios y cada uno debe tener acceso a distintas partes del sistema. Ya que el de servidor web se ejecuta bajo el usuario www-data, y debe acceder al directorio donde se aloja la aplicación web, que era propiedad del usuario boincadm.

El único resultado parcial que se obtuvo se relacionó con que el cliente se podía inscribir a la Grid pero no recibía tareas para ejecución. Este problema se presentó porque el programa que se encargaba de realizar el envío de tareas no estaba corriendo. Este programa era el daemon llamado feeder, el cual, por defecto, se ejecuta al inicializar la Grid. Sin embargo en este caso el procedimiento de inicialización no detectaba al daemon, por lo que era necesario ejecutarlo manualmente.

## **2.3. Resultados: Inscripción y ejecución de tareas con la máquina host**

Los resultados de la tercera prueba eran similares a la segunda ya que eran similares, sin embargo las razones que provocaron los fallos eran distintos.

En resultados fallidos, se debía principalmente a que el tráfico https no se estaba redirigiendo, por lo que no se podía autenticar el usuario. Lo cual se debía a que no se estaba redireccionando el tráfico del puerto 443.

En el caso de resultados parciales, era el mismo caso que la 2da prueba, y se solucionó al mismo tiempo y de la misma manera.

## **2.4. Resultados: Inscripción y ejecución de tareas en los dispositivos móviles**

En la cuarta prueba los posibles resultados eran fallido, parcial y correcto. Donde un resultado fallido se debía a la configuración de la red, en cuanto a que no se lograba comunicación entre los dispositivos y el gestor. Esto se debía a que el servicio DNS, antes mencionado, no era accesible desde los dispositivos, por lo que no lograban encontrar al gestor. Se solucionó al corregir la configuración del DNS.

Los resultados parciales podían ser por las mismas razones que las pruebas anteriores, sin embargo la principal razón era un error de ejecución en la tarea por errores propios de programación de la tarea, principalmente en la configuración del archivo de entrada de la tarea, donde se definía la ejecución de una línea de comando que no era soportada en los dispositivos.

## VIII. EVALUACIÓN DE LA GRID DESPLEGADA

La principal evaluación de la Grid que se realizó fue la de su funcionamiento, después de confirmado su correcto funcionamiento en la fase anterior. El proceso detallado se puede encontrar en el Anexo 4 – Creación y ejecución de aplicaciones.

Después de desplegada la Grid es necesario crear una aplicación, la cual se escribe en C importando las librerías de BOINC para el control de su ejecución. La aplicación inicia la ejecución, procesa un ciclo de un millón de iteraciones para luego escribir “Hello World” en el archivo resultado y terminar la ejecución devolviendo el resultado.

Para su ejecución se prepara el ambiente de cros-compilado para Android, y se elabora el archivo Makefile, el cual contiene los comandos necesarios para la compilación del código. Se compila para la arquitectura Android y Windows para así utilizar recursos de ambos tipos. Se crea una Aplicación y sus dos versiones, que luego se configuran para que sean ejecutadas por la Grid. Por último se crean tareas combinando la aplicación y archivos de entrada vacíos.

Se observa como en los distintos recursos se ejecutan las tareas y reportan su resultado, el cual se puede observar dentro del proyecto prueba en la Grid.

### 1. Análisis de Impacto

A continuación se presentan los distintos impactos que tendrá este proyecto.

Desde el punto de vista del grupo Sidre, la Grid desplegada podrá ser utilizada en las asignaturas de sistemas distribuidos para ilustrar cómo funciona una Grid en el mundo real.

Desde el punto de vista económico se producen los manuales detallados que explican paso a paso las acciones que hay que realizar para desplegar el gestor central para la Grid BOINC. Esta Grid Móvil, dependiendo de la cantidad de recursos que se inscriban, podría representar un ahorro de cientos de miles de dólares para la universidad, debido a que no es necesario invertir en la infraestructura necesaria para proveer esa cantidad de recursos. Es necesario agregar, que el posible ahorro es completamente dependiente de los recursos inscritos a la Grid.

Desde el punto de vista social, con la tecnología de la Grid Móvil cuando esté madura, se podrán resolver necesidades de gran poder de cómputo, que no se podían satisfacer. Esto incluye cualquier simulación o cálculo que ahora la universidad podrá realizar utilizando la Grid para el interés de la sociedad. De igual manera, al poder formar una Grid con recursos

voluntarios, móviles y estáticos, se puede desplegar incluso en lugares de pocos recursos, ya que los móviles empiezan a tener mayor presencia que los computadores. La cual es una tendencia global ya que cada vez más personas están usando móviles inteligentes, como los usados en este proyecto, y aumenta en mayor proporción que las personas que usan computadores (TNS Infratest Germany, 2015).

## **2. Discusión de resultados**

Como se puede observar en el capítulo I. PRUEBAS los resultados del despliegue fueron satisfactorios. Sin embargo es necesario discutir ciertos aspectos detalladamente.

### **2.1. Restricciones del despliegue**

Aunque el sistema BOINC permitió hacer el despliegue de una Grid de computación voluntaria y explotar recursos móviles de manera satisfactoria, existen ciertas restricciones que se mencionan a continuación:

- a) La estructura centralizada de la Grid. Debido a que es necesario un gestor central, toda la estructura de la Grid, y la disponibilidad de la misma, depende del gestor. Si se pierde acceso al gestor no hay forma de participar o explotar la Grid. Esto hace imposible la creación de una Grid Ad hoc, y Ad hoc Móvil.
- b) En la versión actual de BOINC el uso de MPI se limita únicamente a la ejecución de tareas en un mismo nodo (con múltiples cores), por lo que no se tienen limitaciones para la computación paralela.
- c) En la actualidad, aunque se dispone de wrappers para varias arquitecturas, y se dispone del código fuente, no existe un wrapper para dispositivos Android. Por lo que el código fuente de cualquier aplicación que vaya a ser ejecutada en recursos móviles tiene que ser modificado, para agregar ciertas operaciones API de BOINC, cuyo objetivo es marcar el inicio y final de la ejecución al igual que el acceso a archivos de entrada y salida.
- d) La estructura de computación voluntaria le entrega mucho poder al usuario, ya que decide en qué momento ejecutará las tareas de la Grid. Por lo que en cualquier momento sólo se cuenta con los recursos que los usuarios decidieron entregar. Esto hace que el sistema BOINC, tal y como está implementado en la actualidad, no pueda “obligar” a los recursos a ejecutar tareas si el usuario no lo desea.

### **2.2. Comparación con otras alternativas**

Durante el proyecto se argumenta las bondades de las Grids Accesibles, y se despliega el sistema BOINC que, al incluir recursos móviles, se puede catalogar como una Grid Móvil. Sin embargo no se evalúa lo necesario para realizar un desarrollo que cumpliera las funcionalidades que ofrece BOINC y sobrepasara las restricciones nombradas en el capítulo anterior. De igual forma cuenta con funcionalidades más avanzadas que quedaron fuera del

alcance del proyecto. Desarrollar estas funcionalidades, de tal manera que estén listas para ser utilizadas por el público general como BOINC, sería más complejo que el proceso que se realizó en este proyecto. Hay que tener en cuenta que, además de las funcionalidades que se pusieron a prueba en el transcurso del proyecto, BOINC cuenta con una comunidad activa que lo ha extendido con el tiempo. Una enumeración de las funcionalidades generales de BOINC se encuentra en el Anexo 5: Funcionalidades de BOINC

## IX. CONCLUSIONES

Durante el desarrollo de este trabajo se evaluaron varias herramientas para el despliegue de una Grid Accesible que incluyera recursos móviles para la ejecución de tareas. Después de evaluar con cierto nivel de profundidad a cuatro herramientas se escogió el sistema BOINC, que es una infraestructura para la computación voluntaria distribuida desarrollada en la Universidad de Berkeley. El cual es de código libre por lo que cualquier institución puede crear un proyecto de computación distribuida utilizándolo. Entre las primeras ventajas de BOINC, en el contexto de este Trabajo de Grado, se encontraron:

- Permite la inclusión de dispositivos móviles entre sus recursos. Al utilizar recursos móviles es una Grid Móvil, sin embargo su base principal se encuentra en recursos fijos, por lo que es más cercano definirlo como una Grid Híbrida siendo un paso intermedio entre una Grid clásica y una Grid Móvil.
- Puede ejecutar tareas en varios lenguajes de programación. Sin embargo la ejecución de código en C++ requiere de su modificación para incluir operaciones del API, relacionadas con el inicio y ejecución de tareas, así como el uso de archivos. También vale la pena decir que BOINC ofrece herramientas para la compilación cruzada, lo cual es complejo si se realiza manualmente.
- Posee buena documentación.
- Es un proyecto con una comunidad activa, y recursos para resolver dificultades en su despliegue y uso.
- El proceso de despliegue y validación tardó poco en comparación al tiempo que se tomaría desarrollar las funcionalidades que se explotaron.
- Ofrece los principales servicios de una Grid: descubrimiento de servicios, scheduling de tareas, calidad de servicio y seguridad.

Entre sus desventajas se encuentran: no puede ser usada en la formación de una Grid ad hoc, debido a su necesidad de un gestor central; el uso de MPI se limita a un nodo únicamente; Es necesaria la modificación del cliente para que no sea voluntario.

Finalmente, después de una búsqueda bibliográfica extensa, y de evaluar varias herramientas, encontramos que muy pocas se adaptaban a los requerimientos de este Trabajo de Grado. La Grid seleccionada, aunque ofrece varias ventajas, tendría que ser modificada para adaptarse a todos los requerimientos planteados. Se nota una tecnología aún inmadura, debido a los pocos trabajos realizados, y menos sistemas publicados aún, pero un campo de investigación fértil.

### 1. Trabajo futuro

A continuación se exponen los posibles trabajos futuros que surgen a partir de este proyecto:



### **1.1. BOINC en la Pontificia Universidad Javeriana**

Continuando con el proyecto se puede desplegar el sistema BOINC en la infraestructura de la universidad. Luego ingresando los computadores de escritorio de la universidad para que brinden sus recursos cuando no están siendo utilizados. Y por último extenderle a los estudiantes la invitación a compartir sus recursos, móviles o fijos.

Para incentivar a los estudiantes se podría utilizar las herramientas de créditos de BOINC y que ellos tengan algún tipo de retribución según estos créditos.

Para adaptarse a los requerimientos del proyecto de procesamiento de imágenes médicas en la Grid Móvil sería necesario realizar una modificación a los clientes para que los usuarios no pusieran límites a la disponibilidad de los recursos.

### **1.2. Wrapper y administración de aplicaciones**

El wrapper de BOINC, aunque facilita la labor de ejecución de aplicaciones, no se encuentra en todas las arquitecturas que BOINC soporta. Por lo tanto, crear los wrappers para Android y iOS sería beneficioso no sólo para la universidad sino para el sistema BOINC.

También se podría realizar cambios al wrapper para encapsular aplicaciones que utilicen MPI para comunicarse de tal forma que se pueda ejecutar de manera distribuida, y sin reescribir la aplicación.

### **1.3. Interfaz de administración**

El método de administración de la Grid, por medio de cual se realizan las tareas de creación de aplicaciones, de archivos de entrada y salida y creación de tareas, está basado en. Este método no es amigable para un usuario que no tenga conocimientos en administración de sistemas Linux. Sin embargo existe un API para las acciones de administración más comunes, por lo que es posible el desarrollo de un sistema que le permita al administrador de la Grid ingresar aplicaciones, tareas y demás tareas de manera fácil.

## X. REFERENCIAS Y BIBLIOGRAFÍA

- ABI Research. (2014). *Market research*. Obtenido de Mobile Devices Market Sizing and Share: <https://www.abiresearch.com/market-research/product/1021334-mobile-devices-market-sizing-and-share>
- Ahuja, S. P., & Myers, R. J. (2006). A Survey on Wireless Grid Computing. *The Journal of Supercomputing*, 37(1), 3-21.
- Amin, K., Laszewski, G., & Mikler, A. R. (2004). Toward an Architecture for Ad Hoc Grids. *IN 12TH INTERNATIONAL CONFERENCE ON ADVANCED COMPUTING AND COMMUNICATIONS (ADCOM 2004)*, (págs. 15-18). AHMEDABAD.
- Anderson, D. P., Cobb, J., Korpela, E., Lebofsky, M., & Werthimer, D. (2002). SETI@home: an experiment in public-resource computing. *Communications of the ACM* (págs. 56-61). New York, NY, USA: ACM.
- Anderson, P. D. (2004). Boinc: A system for public-resource computing and storage. *Grid Computing, 2004. Proceedings. Fifth IEEE/ACM International Workshop on* (págs. 4-10). IEEE.
- Arslan, M. Y., Singh, I., Singh, S., Madhyastha, H. V., Sundaresan, K., & Krishnamurthy, S. V. (2015). Cwc: A distributed computing infrastructure using smartphones. *Mobile Computing, IEEE Transactions on* , 1587 - 1600.
- Bargiela, A., & Pedrycz, W. (2012). *Granular computing: an introduction*. Springer Science & Business Media.
- Beer, C., & Walton, R. (22 de Noviembre de 2015). *BOINC: Open-source software for volunteer computing and grid computing - Repository*. Obtenido de Github: <https://github.com/BOINC>
- Bernman, F., Fox, G., & Hey, A. (2003). *Grid Computing: Making the Global Infrastructure a Reality*. John Wiley and Sons.
- Bernstein, P. A. (1996 ). Middleware: a model for distributed system services. *Communications of the ACM*, 86-98 .
- BOINC. (2011). *MPI applications*. Obtenido de Boinc: <http://boinc.berkeley.edu/trac/wiki/MpiApps>

- BOINC. (2012). *Python applications*. Obtenido de Boinc Project: <https://boinc.berkeley.edu/trac/wiki/PythonApps>
- BOINC. (2012). *Using BOINC with Java applications*. Obtenido de Boinc Project: <https://boinc.berkeley.edu/trac/wiki/JavaApps>
- BOINC. (21 de Julio de 2013). *How BOINC works*. Obtenido de BOINC - Open-source software for volunteer computing and grid computing: [https://boinc.berkeley.edu/wiki/How\\_BOINC\\_works](https://boinc.berkeley.edu/wiki/How_BOINC_works)
- BOINC. (17 de Diciembre de 2014). *Android FAQ*. Obtenido de Boinc Client Wiki: [http://boinc.berkeley.edu/wiki/Android\\_FAQ](http://boinc.berkeley.edu/wiki/Android_FAQ)
- BOINC. (14 de Septiembre de 2014). *Choosing BOINC projects*. Obtenido de Boinc: <http://boinc.berkeley.edu/projects.php>
- BOINC. (2015). *FORTRAN applications*. Obtenido de Boinc Project: <https://boinc.berkeley.edu/trac/wiki/FortranApps>
- BOINC. (22 de Noviembre de 2015). *Home*. Obtenido de BOINC: All Projects Stats: <http://www.allprojectstats.com/>
- BOINC. (3 de Noviembre de 2015). *Project statistics - Collatz Conjecture*. Obtenido de All Project Stats: <http://www.allprojectstats.com/po.php?projekt=90>
- BOINC. (2015). *The BOINC Wrapper*. Obtenido de Boinc Project: <https://boinc.berkeley.edu/trac/wiki/WrapperApp>
- Bornemann, M., van Nieuwpoort, R. V., & Kielmann, T. (2005). MPJ/Ibis: a flexible and efficient message passing platform for Java. *Proceedings of 12th European PVM/MPI Users' Group Meeting*, (págs. pages 217-224). Sorrento, Italy.
- Büsching, F., Schildt, S., & Wolf, L. (2012). DroidCluster: Towards Smartphone Cluster Computing--The Streets are Paved with Potential Computer Clusters. *Distributed Computing Systems Workshops (ICDCSW), 2012 32nd International Conference on* (págs. 114-117). IEEE.
- Castro, H., Chacón, J., Díaz, C., González, E., & Zuluaga, J. (Junio de 2011). Grid Colombia: Soporte para Investigaciones de Avanzada. *Renata, 1*.
- Chatzimilioudis, G., Konstantinidis, A., Laoudias, C., & Zeinalipour-Yazti, D. (2012). Crowdsourcing with Smartphones. *Internet Computing, IEEE*, 36 - 44.

- Choudhary, A. N., & Ponnusamy, R. (1991). Implementation and evaluation of Hough transform algorithms on a shared-memory multiprocessor. *Journal of Parallel and Distributed Computing*, 178-188.
- Chu, C. D., & Humpfrey, M. (2004). Mobile OGSI.NET: Grid Computing on Mobile Devices. *Proceedings of the 5th IEEE/ACM International Workshop on Grid Computing*. Pittsburgh, PA.: IEEE.
- Community Research and Development Information Service. (2002). *Access to knowledge through the Grid in a mobile world - Akogrimo*. Community Research and Development Information Service.
- Coronato, A., & De Pietro, G. (2008). MiPeG: A middleware infrastructure for pervasive grids. *Future Generation Computer Systems*, 24(1), 17–29.
- Curiel, M. (2016). Wireless Grids: Recent Advances in Resource and Job Management. En G. Gürsel, *Mobility in Healthcare: M-Health* (págs. 293-320). IGI-Global.
- Datta, P., Dey, S., Paul, H. S., & Mukherjee, A. (2014). ANGELS: A framework for mobile grids. *Applications and Innovations in Mobile Computing (AIMoC)*, 15-20.
- Duan, L., Kubo, T., Sugiyama, K., Huang, J., Hasegawa, T., Walrand, & Jean. (2014). Incentive mechanisms for smartphone collaboration in data acquisition and distributed computing. *INFOCOM, 2012 Proceedings IEEE* (págs. 1701-1709). IEEE.
- Eliassen, F. (2005). *MADAM: Mobility and adaptation*. Oslo, Noruega: Simula Research Laboratory.
- Foster, I. (2002). The grid: A new infrastructure for 21st century science. *Physics today*.
- Foster, I., & Kesselman, C. (2013). The History of the Grid. *Cloud Computing and Big Data*.
- Foster, I., Carl, K., & Tuecke, S. (2001). The anatomy of the grid: Enabling scalable virtual organizations. *International Journal of High Performance Computing Applications*, 200-222.
- Foster, I., Kesselman, C., Nick, J. M., & Tuecke, S. (2002). The physiology of the grid. En F. Berman, G. Fox, & T. Hey, *Grid computing: making the global infrastructure a reality* (págs. 217-249). Chichester: Jon Wiley & Sons Ltd.
- Fritsch, J. (30 de Septiembre de 2014). *BOINC/Android:Status*. Budapest: 10th BOINC Workshop. Obtenido de BOINC.

- Furthmüller, J., & Waldhorst, O. P. (2010). Survey on Grid Computing on Mobile Consumer Devices. En N. Antonopoulos, G. Exarchakos, L. Maozhen, & A. Liotta, *Handbook of Research on P2P and Grid Systems for Service-Oriented Computing: Models, Methodologies* (págs. 313-363). Hershey: IGI Global.
- Gang, L., Hongmei, S., Huahao, G., Haiyan, Y., & Yue, C. (2009). A Survey on Wireless Grids and Clouds. *Grid and Cooperative Computing, 2009. GCC '09. Eighth International Conference on* (págs. 261 - 267). Lanzhou, Gansu: IEEE.
- GIGA Project. (2005). Obtenido de GIGA Project: <http://www.projetogiga.org.br/>
- GNU. (31 de Diciembre de 2014). *GNU Automake: 2.2.8 Cross-Compilation*. Obtenido de GNU Automake Manual: [https://www.gnu.org/savannah-checkouts/gnu/automake/manual/html\\_node/Cross\\_002dCompilation.html](https://www.gnu.org/savannah-checkouts/gnu/automake/manual/html_node/Cross_002dCompilation.html)
- Gomes, A. T. (24 de Junio de 2015). *MoGrid*. Obtenido de Github: <https://github.com/atagomes/MoGrid>
- Gomes, A. T., Ziviani, A., Lima, L. S., Endler, M., & Chelius, G. (2008). Mitigating reply implosions in query-based service discovery protocols for mobile wireless ad hoc networks. *Ad-hoc, Mobile and Wireless Networks* (págs. 29-42). Sophia Antipolis: Springer Berlin Heidelberg.
- Hijab, M., & Avula, D. (2011). Resource discovery in wireless, mobile and ad hoc grids — Issues and challenges. *Advanced Communication Technology (ICACT), 2011 13th International Conference on* (págs. 502 - 505). Seoul: IEEE.
- Ibis. (20 de Noviembre de 2009). *Ibis MPJ User's Guide*. Obtenido de The Ibis Project: <http://www.cs.vu.nl/ibis/downloads/mpj/usersguide.pdf>
- Ibis. (2011). *Ibis Javadoc*. Obtenido de The Ibis Project: <http://www.cs.vu.nl/ibis/javadoc.html>
- Ibis. (2011). *Ibis Overview*. Obtenido de The Ibis Project: <http://www.cs.vu.nl/ibis/projects.html>
- Ibis. (Agosto de 2014). *News*. Obtenido de The Ibis Project: <http://www.cs.vu.nl/ibis/news.html>
- InteGridade Project. (2005). *InteGridade Project*. Obtenido de <http://integridade.lncc.br/>
- Jaehnert, J. M., Wesner, S., & Villagra, V. A. (2007). The akogrimo mobile grid reference architecture-overview. *Akogrimo tech. rep.*

- Junseok, H., & Aravamudham, P. (2004). Middleware services for P2P computing in wireless grid networks. *Internet Computing, IEEE*, 8(4), 40 - 46.
- Kurdi, H., Maozhen, L., & Al-Raweshidy, H. (2008). A Classification of Emerging and Traditional Grid Systems. *IEEE Distributed Systems Online*, 9(3).
- Lella, A., & Lipsman, A. (2014). *The US mobile app report*. Obtenido de Comscore: <http://www.comscore.com/Insights/Presentationsand-Whitepapers/2014/The-US-Mobile-App-Report>
- Li, Z., Sun, L., & Ifeachor, C. E. (2005). Challenges of mobile ad-hoc grids and their applications in e-healthcare. *Proceedings of Second International Conference on Computational Intelligence in Medicine and Healthcare*.
- Lima, L. S., Gomes, A. S., Ziviani, A., Endler, M., Soares, L. F., & Schulze, B. (2005). Peer-to-peer resource discovery in mobile Grids. *MGC '05 Proceedings of the 3rd international workshop on Middleware for grid computing* (págs. 1-6). New York, NY, USA: ACM.
- Malan, D., Fulford-Jones, T., Welsh, M., & Moulton, S. (2004). Codeblue: An ad hoc sensor network infrastructure for emergency medical care. *International workshop on wearable and implantable body sensor networks*.
- Manvi, S. S., & Birje, M. N. (2010). A Review on Wireless Grid Computing. *International Journal of Computer and Electrical Engineering*, 2(3), 469-474.
- MPI Forum. (4 de Junio de 2015). *MPI: A Message-Passing Interface Standard*. Obtenido de MPI Documents: <http://www.mpi-forum.org/docs/mpi-3.1/mpi31-report.pdf>
- MPICH. (2012). *MPICH Overview*. Obtenido de MPICH: <http://www.mpich.org/about/overview/>
- Mutchler, A. L., Shim, P. J., & Ormond, D. K. (2011). Exploratory Study on Users' Behavior: Smartphone Usage. *AMCIS 2011 Proceedings*.
- Ohta, K., Yoshikawa, T., Nakagawa, T., & Inamura, H. (2005). Design and implementation of mobile grid middleware for handsets. *Parallel and Distributed Systems, 2005. Proceedings. 11th International Conference on (Volumen 2)* (págs. 679 - 683). IEEE.
- Otto, C., Milenkovic, A., Sanders, C., & Jovanov, E. (2006). System architecture of a wireless body area sensor network for ubiquitous health monitoring. *Journal of Mobile Multimedia*, 307-326.

- OurGrid. (30 de Agosto de 2013). *Downloads*. Obtenido de OurGrid: <http://www.ourgrid.org/download.php>
- OurGrid. (2013). *Overview*. Obtenido de OurGrid: <http://www.ourgrid.org/overview.php>
- Palmer, N., Kemp, R., Kielmann, T., & Bal, H. (2009). Ibis for Mobility: Solving Challenges of Mobile Computing Using Grid Techniques. *Proceedings of the 10th Workshop on Mobile Computing Systems and Applications* (págs. 17:1–17:6). New York, NY, USA: ACM.
- Perkins, C. E. (2008). *Ad Hoc Networking*. Addison-Wesley Professional.
- Research, A. (2015). *Mobile Devices Market Sizing and Share*. Obtenido de <https://www.abiresearch.com/market-research/product/1021336-mobile-devices-market-sizing-and-share/>
- Rodriguez, J. M., Mateos, C., & Zunino, A. (2012). Are Smartphones Really Useful for Scientific Computing? *Lecture Notes In Computer Science*. Springer.
- Rodriguez, J., Zunino, A., & Campo, M. (2011). Introducing mobile devices into Grid systems: a survey. *International Journal of Web and Grid Services*, 7(1), 1-40.
- Román, M., Hess, C., Cerqueira, R., Ranganathan, A., Campbell, R. H., & Nahrstedt, K. (2002). A middleware infrastructure for active spaces. *IEEE pervasive computing*, 74-83.
- Sajjad, A., Jameel, H., Kalim, U., Lee, Y.-K., & Lee, S. (2005). A component-based architecture for an autonomic middleware enabling mobile access to grid infrastructure. *EUC'05 Proceedings of the 2005 international conference on Embedded and Ubiquitous Computing* (págs. 1225-1234 ). Heidelberg, DE: Springer-Verlag Berlin.
- Schalkoff, R. J. (1989). *Digital image processing and computer vision*. Michigan: John Wiley & Sons Australia, Limited.
- Shah, S. C. (2013). Mobile Ad Hoc Computational Grid: Opportunities and Challenges. *MILCOM 2013 - 2013 IEEE Military Communications Conferense*, (págs. 848-857).
- Silva, R. I., Del Luca Almeida, V., Poersch, A. M., & Nogueira, J. M. (2010). Wireless sensor network for disaster management. *Network Operations and Management Symposium (NOMS), 2010 IEEE* (págs. 870 - 873). Osaka: IEEE.

- Sonntag, J. (2015). *About Collatz Conjecture*. Obtenido de Collatz Conjecture: <http://boinc.thesonntags.com/collatz/>
- Srinivasan, S. H. (2005). Pervasive wireless grid architecture. *Wireless On-demand Network Systems and Services, 2005. WONS 2005. Second Annual Conference on* (págs. 83-88). IEEE.
- Stallings, W. (1995). *Operating Systems: Internals and Design Principles*. Englewood Cliffs: Prentice Hall.
- Tapparello, C., Karaoglu, B., Funai, C., Ba, H., Hijazi, S., Shi, J., . . . Heinzelman, W. (2015). Volunteer Computing on Mobile Devices: State of the Art and Future. En T. Soyata, *Enabling Real-Time Mobile Cloud Computing through Emerging Technologies* (págs. 152-181). Hershey: Information Science Reference.
- The Process Group. (October de 2005). DECISION ANALYSIS AND RESOLUTION PROCESS. *newsletter*, págs. Vol. 12, No. 2.
- Tiso, J., & Teare, D. (2011). Chapter 1: The Cisco Enterprise Architecture. En J. Tiso, & D. Teare, *Designing Cisco Network Service Architectures (ARCH): Foundation Learning Guide* (págs. 1-21). Cisco Press.
- TNS Infratest Germany. (10 de Junio de 2015). The Connected Consumer - Public Data. *Global Connected Consumer Survey (CCS) Global Device Usage and Internet Stats*. Obtenido de [http://www.google.com.sg/publicdata/explore?ds=dg8d1eetcqsb1\\_&hl=en\\_US&dl=en\\_US](http://www.google.com.sg/publicdata/explore?ds=dg8d1eetcqsb1_&hl=en_US&dl=en_US)
- Tuecke, S., Czajkowski, K., Foster, I., Frey, J., Graham, S., Kesselman, K., . . . Snelling, D. (2003). Open Grid Services Infrastructure (OGSI). *GLOBAL GRID FORUM*.
- Vidal Aroca, R., & Garcia Goncalves, L. M. (2012). Towards green data centers: A comparison of x86 and ARM. *Journal of Parallel and Distributed computing*, 1770-1780.
- Wolff, A., Michaelis, S., Schmutzler, J., & Wietfeld, C. (2007). Network-centric Middleware for Service Oriented Architectures across Heterogeneous Embedded Systems. *EDOC Conference Workshop, 2007. EDOC '07. Eleventh International IEEE* (págs. 105 - 108). Annapolis, MD: IEEE.



## XI. ANEXOS

### **Anexo 1: Preparación de la máquina virtual**

En este anexo se detalla el proceso llevado a cabo para la preparación de una máquina virtual, y la instalación del sistema BOINC.

<http://pegasus.javeriana.edu.co/~CIS1530IN03/docs/Anexo%201%20-%20Preparacion%20de%20la%20maquina%20virtual.pdf>

### **Anexo 2: Proceso de despliegue del gestor**

En este anexo se detalla el proceso llevado a cabo para desplegar el gestor de la Grid.

<http://pegasus.javeriana.edu.co/~CIS1530IN03/docs/Anexo%202%20-%20Proceso%20de%20despliegue%20del%20gestor.pdf>

### **Anexo 3: BOINC API**

En este anexo se detalla el uso del API ofrecido por BOINC.

<http://pegasus.javeriana.edu.co/~CIS1530IN03/docs/Anexo%203%20-%20BOINC%20API.pdf>

### **Anexo 4: Creación y ejecución de aplicaciones**

En este anexo se detalla el proceso para la creación y ejecución de aplicaciones en la Grid.

<http://pegasus.javeriana.edu.co/~CIS1530IN03/docs/Anexo%204%20-%20Creaci%C3%B3n%20y%20ejecuci%C3%B3n%20de%20aplicaciones.pdf>

### **Anexo 5: Funcionalidades de BOINC**

En este anexo se listan las funcionalidades generales del sistema.

<http://pegasus.javeriana.edu.co/~CIS1530IN03/docs/Anexo%205%20-%20Funcionalidades%20BOINC.pdf>