



# ESQUEMAS DE DISTRIBUCIÓN DE TRABAJOS PARA UN SISTEMA ERP EN LA NUBE

110300083 CANUL CANUL CINDY GUADALUPE

110300083@UCARIBE.EDU.MX

110300097 KUMUL UC CRISTIAN ARIEL

110300097@UCARIBE.EDU.MX

110300079 PERAZA FELICIANO JONATHAN

110300079@UCARIBE.EDU.MX

**Asesor: Joel Antonio Trejo Sánchez**

**Otoño 2015**

**Ingeniería en Telemática**



Departamento de Ciencias Básicas e Ingenierías  
IT0427 / Proyecto terminal

### CARTA DE ACEPTACIÓN DE PROYECTO TERMINAL

#### TÍTULO

Esquemas de distribución de trabajo para un sistemas ERP en la nube.

#### AUTORES

110300083

Cindy Guadalupe Canul Canul

110300083@ucaribe.edu.mx

110300097

Cristian Ariel Kumul Uc

110300097@ucaribe.edu.mx

110300079

Jonathan Peraza Feliciano

110300079@ucaribe.edu.mx

#### ASESOR

Joel Antonio Trejo Sánchez

#### RESUMEN

El cómputo en la nube es una tecnología prometedora como plataforma del futuro, permite el acceso a recursos de hardware de manera remota. Dentro del funcionamiento de los servicios es necesario optimizar los recursos en el centro de datos y para esto se utilizan los esquemas de calendarización. El problema incrementa con el número de tareas, pues es un problema NP-difícil, debido a las distintas características de cada host.

En este proyecto se estudiará y se hará una comparación de varios esquemas de calendarización para proponer una mejora en la optimización de recursos de acuerdo a las peticiones de un sistema ERP en la nube.

#### FIRMAS

AUTORES

ASESOR

TITULAR DE LA  
ASIGNATURA

---

# Resumen

---

# Abstract

---

# Índice general

<b>Resumen</b>	<b>3</b>
<b>Abstract</b>	<b>4</b>
<b>Introducción</b>	<b>9</b>
Problema . . . . .	9
Propuesta de solució . . . . .	10
Objetivos . . . . .	11
Objetivo general . . . . .	11
Objetivos específicos . . . . .	12
Hipótesis . . . . .	12
<b>1. Marco contextual</b>	<b>13</b>
Introducción . . . . .	13
1.1. Situación Actual . . . . .	14
1.2. Estado del Arte . . . . .	15
1.3. Marco teórico . . . . .	16
Centro de Datos . . . . .	17
Esquemas de Distribución de Trabajos . . . . .	17
Balanceo de Cargas . . . . .	18
Migración . . . . .	19
Sistemas ERP . . . . .	19
1.4. Marco Metodológico . . . . .	19

---

<b>2. Desarrollo</b>	<b>21</b>
Introducción . . . . .	21
2.1. Aplicación del marco metodológico y de actividades de experimentación	21
<b>3. Resultados</b>	<b>22</b>
Introducción . . . . .	22
<b>Conclusiones</b>	<b>23</b>
Recomendaciones . . . . .	23
<b>Referencias</b>	<b>24</b>
<b>Glosario</b>	<b>25</b>
A . . . . .	25
<b>Anexos</b>	<b>26</b>
. . . . .	26
<b>Bibliografía</b>	<b>27</b>

---

# Índice de figuras

1.	Propuesta de arquitectura en la nube . . . . .	11
1.1.	Esquema general del Cómputo en la Nube . . . . .	17

---

## Índice de cuadros



---

# Introducción

## Antecedentes y Planteamiento del problema

El cómputo en la nube es una tecnología que ha abarcado gran parte de los negocios para dar soporte a ellos. Ésta tecnología provee a los negocios una ventaja competitiva en los costos de recursos como se ve en los negocios tradicionales, además de la versatilidad que provee al ajustarse a la necesidad de las empresas (Srinivasan, 2014).

La tecnología en la nube ha desarrollado una infraestructura fuerte después del surgimiento del cómputo distribuido (Chen y Deng, 2009). Para obtener las ventajas de dicha tecnología los usuarios simplemente necesitarán conectarse a internet y de esta manera tendrán el acceso al procesamiento de manera remota (Aranganathan, 2011). Sin embargo, para aprovechar el máximo potencial de éstos recursos, es necesario tener en consideración algunas variables, ya que en un entorno en la nube existe un comportamiento dinámico de los recursos a manera que se les provea a los usuarios el servicio (Shimpy y Sidhu, 2014). Una de las prácticas con mayor importancia en la nube es la calendarización, ya que tiene como objetivo administrar las tareas del centro de datos para optimizar los recursos del mismo. De esta manera la eficiencia de la carga de trabajo en la nube aumenta (Shimpy y Sidhu, 2014). En general, el objetivo de la calendarización en la nube es utilizar los recursos de manera apropiada, mientras la carga de trabajo es distribuida uniformemente para mejorar los tiempos de ejecución (Shimpy y Sidhu, 2014). Debido a la atención que se tiene en la tecnología en la nube, los centros de datos han tomado un papel muy importante para los servicios empresariales (Shimpy y Sidhu, 2014). Un centro de datos está compuesto por miles de servidores virtuales ejecutándose en una instancia de

tiempo alojando muchas tareas, al mismo tiempo el centro de datos recibe miles de peticiones a esas tareas. Es aquí en donde la programación de trabajos tiene un rol demasiado importante para el cómputo en la nube, ya que influye en el rendimiento del mismo (Srinivasan, 2014).

El problema de la calendarización pertenece a los algoritmos NP-Difícil, lo cual tiene un amplio rango de soluciones posibles y se toma mucho más tiempo de encontrar una respuesta optima, ya que no existe un método para resolver estas incógnitas. Sin embargo, es posible estar cerca de la mejor solución contemplando algunos entornos (Shimpy y Sidhu, 2014).

## Propuesta de solución y Justificación

Los centros de datos son una parte esencial en la tecnología en la nube, están conformados por varios servidores virtuales que alojan varios trabajos, ejecutándose por estancias de tiempo y a la vez recibiendo peticiones a esos trabajos (Shimpy y Sidhu, 2014). Para que el cómputo en la nube pueda rendir correctamente, se necesita una buena programación de trabajos o calendarización. Este problema tiene un gran número de soluciones posibles, toma mucho tiempo en encontrar una respuesta óptima y por eso está dentro de los algoritmos NP-Difícil. No existe un método para resolver lo anterior, sin embargo es posible obtener una solución cercana contemplando algunos entornos (Shimpy y Sidhu, 2014). En esta investigación se propone una mejora a un algoritmo de calendarización para el caso de estudio de un sistema ERP en la nube, contemplando sus posibles escenarios y peticiones heterogéneas. De esta manera se podrá contar con un esquema de distribución de trabajos que satisfaga las necesidades de un sistema ERP, mejorando el tiempo de respuesta y aprovechando los recursos. Con beneficios a los proveedores de servicios (SaaS) de sistemas ERP.

En la figura 1 se puede observar la arquitectura del centro de datos que se implementará en la investigación, el cual está compuesto de los siguientes elementos:

- **Servidor administrador:** Es el servidor que tendrá el rol de front end en el centro de datos, teniendo una interacción directa con las peticiones de los usuarios, que son especificados en éste documento como trabajos. El principal

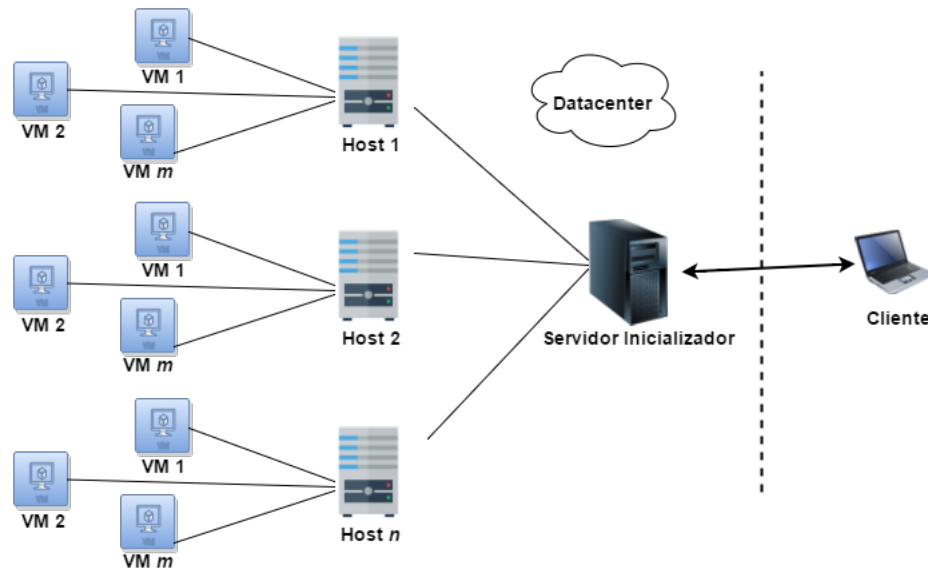


Figura 1: Propuesta de arquitectura en la nube

objetivo de éste elemento es delegar los trabajos a las máquinas virtuales en los distintos hosts del centro de datos.

- **Host:** es el recurso de hardware que se tiene en el centro de datos. Una de las características de éste elemento es que es finito.
- **Máquinas Virtuales:** son instancias dentro de los host, tienen como objetivo resolver los trabajos que se les sea asignado.

## Objetivos

### Objetivo general

Determinar y proponer un nuevo esquema que pueda resolver de manera eficiente el problema de la calendarización de un centro de datos y satisfacer las necesidades de un sistema ERP en la nube.

## Objetivos específicos

- Definir una arquitectura para un centro de datos en la nube con las necesidades de un sistema ERP.
- Realizar la simulación en base a la arquitectura del centro de datos.
- Realizar e implementar diferentes esquemas de distribución de trabajos, rescatando sus características principales.
- Reunir las características principales y proponer el nuevo esquema de distribución.
- Aplicar el nuevo esquema de distribución a las necesidades de un ERP.
- Comparar los resultados obtenidos del nuevo esquema propuesto y de los anteriores para determinar si hubo una mejora.

## Hipótesis

La modificación de un esquema de distribución mejorará el rendimiento de la calendarización para un sistema ERP en la nube.

Para realizar la verificación de validez de la hipótesis se usarán:

- **Pruebas de hardware y software:**
  - **Prueba de desempeño:** Validar el tiempo de respuesta para las tareas bajo condiciones normales y máximas.
  - **Prueba de carga:** Verificar el tiempo de respuesta del sistema, bajo diferentes condiciones de carga. Se mide el tiempo de respuesta y otros requisitos sensibles al tiempo.
- **Estadísticas:** en base a los resultados que se obtendrán en las pruebas de desempeño y de carga, se va realizar comparaciones y posteriormente visualizarlas por medio de gráficas y tablas.

---

# Capítulo 1

## Marco contextual

### Introducción

En este capítulo se describen algunos marcos conceptuales necesarios para fundamentar y analizar los requisitos de la investigación, las secciones son las siguientes: Situación actual, Estado del Arte, Marco Teórico y el Marco Metodológico. En la primera sección se muestra cómo se encuentra la investigación en la actualidad, cómo se encuentra nacional e internacionalmente y cuál es el punto de partida del proyecto; la siguiente sección se presenta la información sobre los diferentes esquemas de distribución que serán implementados durante la elaboración del proyecto. El Marco teórico nos describe los conceptos necesarios para fundamentar y comprender mejor el proyecto y su propósito, finalmente en la última sección se describe los pasos que se van a seguir para realizar y alcanzar los objetivos del proyecto.

## 1.1. Situación Actual

Actualmente, hay un mayor número de organizaciones que utilizan el cómputo en la nube, ya que trae beneficios en cuanto a gastos y recursos. Ya no hay necesidad de invertir tanto en la administración de TI (Information Technology) y existe una mejor capacidad de almacenamiento para los recursos que el ámbito empresarial necesita. El cómputo en la nube utiliza los servicios SaaS (Software as a Service) y HaaS (Hardware as a Service) para hacer más eficiente y flexible el uso de la infraestructura de la nube (Mariscal y Gil-Garcia, 2013).

México utiliza, en cuanto a la infraestructura, los servicios ofrecidos por Amazon, Google y Triada (TELMEX). Estos proporcionan almacenaje de información virtual, plataformas de hospedaje de aplicaciones y servicios en línea (Mariscal y Gil-Garcia, 2013).

Al proveer el servicio se tiene dos escenarios diferentes: cuando hay una mayor demanda de recursos y cuando no la hay, eso va dependiendo de las necesidades de las empresas. Para que se tenga una mejor eficacia durante los servicios, se tiene que mejorar el uso de los recursos en el centro de datos. Dentro del centro de datos hay muchos servidores virtuales que están recibiendo trabajos, mientras la nube las mantiene en los lotes de solicitud de trabajos. Es aquí, donde resaltamos la importancia de la distribución de trabajos dentro del centro de datos (Shimpy y Sidhu, 2014).

En este momento, la distribución de trabajos es un problema NP- difícil, pues que no se ha encontrado una solución óptima. Sin embargo, existen diferentes propuestas para encontrar una mejor solución, utilizando diferentes esquemas de distribución o dicho de otra forma algoritmos (Shimpy y Sidhu, 2014).

Existen una gran variedad de algoritmos utilizados en el problema de distribución de trabajos. Los algoritmos más utilizados en las investigaciones son: FCFS, Round Robin, Min-Min, Max-Min y Metaheurística. Cabe mencionar que en las investigaciones solo hacen una selección de una serie de algoritmos para que, por medio de pruebas, determinen quién es el mejor de ellos.

## 1.2. Estado del Arte

Los siguientes algoritmos de calendarización actualmente están prevaleciendo en la nube:

- **Resource-Aware-Scheduling Algorithm (RASA):** Parsa, Entezari-Maleki (2009) proponen el algoritmo RASA, el cual utiliza las ventajas de dos algoritmos tradicionales (Max-min y Min-min) y cubre sus desventajas. Aunque el tiempo limite, la tasa de llegada, costo de ejecución y costo de comunicación no están considerados (Parsa y Entezari-Maleki, 2009).
- **RSDC (Reliable Scheduling Distributed In Cloud Computing):** Delavar, Javanmard, Shabestari y Talebi (2012) proponen un algoritmo confiable en un entorno en la nube, en este algoritmo los trabajos importantes son divididos en sub trabajos, de tal manera que se puedan balancear las peticiones (Delavar et al., 2012).
- **An Optimal Model for Priority based Service Scheduling Policy for Cloud Computing Environment:** Dakshayini, Gurupasad (2011), proponen un nuevo algoritmo de calendarización que se basa en la prioridad y un esquema de control de admisión. En este algoritmo, la prioridad se asigna a cada proceso admitido en la cola (Dakshayini y Guruprasad, 2011) .
- **Pre-emptable Shortest Job Next Scheduling algorithm (PSJN) :** Este algoritmo se propone en una nube privada. Utiliza la técnica de suscripción preferente del algoritmo de Round Robin junto con el siguiente proceso más corto (PSN). Brinda beneficios de costos y mejora tiempo de respuesta y tiempo de ejecución (nis, 2015).
- **User-priority Guided Min-min scheduling algorithm:** Se realiza una mejora para el algoritmo de balanceo de cargas, a través del algoritmo Min-min para la calendarización de trabajos con el fin de minimizar el tiempo de terminación del ultimo trabajo (makespan) y maximizar la utilización de los recursos (Chen et al., 2013).

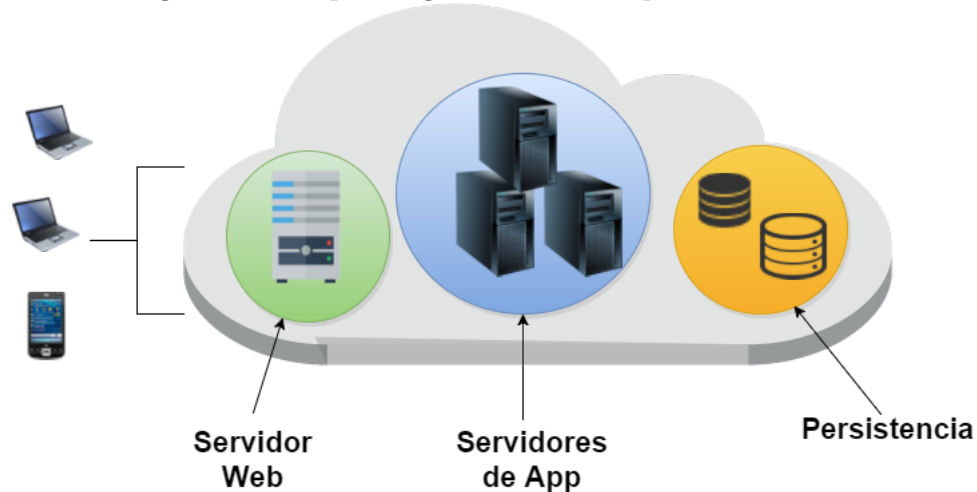
### 1.3. Marco teórico

El cómputo en la nube es una tecnología emergente, la cual está compuesta por un grupo de recursos heterogéneos que proveen servicios a través de internet (Agarwal et al., 2014). Esta tecnología permite a los consumidores y negocios utilizar aplicaciones sin necesidad de instalación y con acceso a sus archivos personales en cualquier computadora con acceso a internet (Ahmed et al., 2012). El cómputo en la nube se puede clasificar de dos maneras:

- Por la ubicación:
  - **Nube Pública:** La infraestructura de cómputo se puede compartir entre cualquier organización (Ahmed et al., 2012).
  - **Nube Privada:** La infraestructura de cómputo es dedicada a una organización en particular y no se comparte con otras organizaciones (Ahmed et al., 2012).
  - **Nube Híbrida:** Las organizaciones pueden albergar sus aplicaciones críticas en nubes privadas y las aplicaciones con menos problemas de seguridad las puede albergar en nubes públicas (Ahmed et al., 2012).
- Por el tipo de servicios ofrecidos:
  - **Infrastructure as a Service (IaaS):** En éste nivel, la infraestructura se ofrece como servicio hacia los solicitantes en forma de Máquinas Virtuales (VM) (Agarwal et al., 2014).
  - **Platform as a Service (PaaS):** Es una plataforma de desarrollo de aplicaciones que se provee como servicio hacia los desarrolladores para crear aplicaciones basadas en web (Agarwal et al., 2014).
  - **Software as a Service (SaaS):** En éste nivel el proveedor en la nube provee las aplicaciones de software (Agarwal et al., 2014).



Figura 1.1: Esquema general del Cómputo en la Nube



## Centro de Datos

Un centro de datos (datacenter) es un espacio dedicado donde las organizaciones almacenan y operan la mayoría de su información y la infraestructura de las tecnologías de comunicaciones que respaldan sus negocios (EPI, 2015).

## Esquemas de Distribución de Trabajos

La distribución de trabajo (calendarización) es uno de los puntos principales además de los más retadores problemas en el cómputo en la nube (Li et al., 2014). Esta consiste en un conjunto de políticas para controlar el orden en el cual los procesos serán ejecutados por el sistema (Agarwal et al., 2014).

Existen varios tipos de algoritmos de calendarización, su principal ventaja es obtener un alto rendimiento al cumplir con las tareas. Los principales ejemplos de algoritmos de calendarización son (Salot, 2013):

- **FCFS (First Come First Serve) Algorithm:** significa que el trabajo que llegue primero será el primero en ser ejecutado.
- **Round Robin Algorithm:** consiste en agregar un tiempo de ejecución para cada trabajo, si dicho tiempo se agota y el trabajo actual no ha finalizado, pasa a un estado de espera y continúa con el siguiente trabajo, al finalizar el

último trabajo regresa a revisar si existen trabajos en espera para proceder a ejecutarlas.

- **Min-Min Algorithm:** se va ejecutando una por una iniciando por los trabajos de menor tamaño.
- **Max-Min Algorithm:** se va ejecutando una por una iniciando por los trabajos de mayor tamaño.
- **Most fit task scheduling algorithm:** este algoritmo busca los elementos más aptos en la cola de trabajos para ejecutarlo primero. Tiene un alto rádio de error.
- **Priority scheduling algorithm:** la idea básica es que a cada proceso se le asigna una prioridad, y dependiendo a las prioridades de los demás procesos es cómo se ejecutarán cada uno de ellos.

## Balanceo de Cargas

Dentro de nuestro entorno en la nube cada host tiene recursos de hardware finitos y son susceptibles a fallos. Para mitigar contra las fallas y el exhaustivo uso de los recursos, los host son agrupados en clusters, los cuales son esencialmente un agrupamiento de recursos compartidos. El administrador (manager) es capaz de asegurarse que ninguno de los host dentro del cluster es responsable de todas las máquinas virtuales dentro de ese cluster ([Dover, 2013](#)).

Un entorno de virtualización responde a los cambios en la demanda para los recursos en cada host haciendo uso del balanceo de cargas, calendarización de trabajos y migración.

Una política de balanceo de cargas es configurada para un cluster, que a su vez contiene multiples hosts, cada uno de ellos con distintos recursos de hardware y disponibilidad de memoria ([Dover, 2013](#)). Existen tres políticas para el balanceo de cargas:

- Sin Balanceo.
- Distribución Uniforme.

- Ahorro de Energía.

## Migración

La migración se utiliza para hacer cumplir las políticas dentro del balanceo de cargas. La migración de una máquina virtual toma lugar de acuerdo a las políticas de balanceo de cargas para un cluster y la demanda actual sobre los hosts dentro de dicho cluster. La migración de igual manera puede ser configurada para ocurrir automáticamente cuando un host es bloqueado o movido a modo de mantenimiento (Dover, 2013).

## Sistemas ERP

La sigla ERP, en inglés Enterprise Resource Planning, significa Planificación de los Recursos de la Empresa. Un sistema ERP constituye un marco de trabajo que incluye aplicaciones comerciales, administrativas (finanzas, contabilidad), recursos humanos, planeamiento de manufactura y gestión de proyectos (Saroka, 2002).

### 1.4. Marco Metodológico

A continuación se propone la siguiente metodología para llevar a cabo el proyecto:

- **Simular:** Se implementará a manera de simulación un centro de datos con un entorno en la nube, las máquinas virtuales y el servidor inicializador que lo conforman.
- **Implementar:** En el centro de datos se desarrollarán los algoritmos de calendarización que se mencionan con antelación.
- **Evaluar:** Se simulará el comportamiento de las peticiones de un sistema contemplando los distintos escenarios del ERP y consumiendo el servicio en la nube (SaaS), para conocer el rendimiento en tiempo de ejecución de los algoritmos.
- **Mejorar:** Se propondrá una mejora a algún algoritmo de acuerdo a las necesidades y comportamiento de un sistema ERP en la nube.

- **Comparar:** Se realizará una comparativa de tiempo de ejecución entre la versión mejorada y la original para el caso de estudio de un sistema ERP.

---

## Capítulo 2

### Desarrollo

#### Introducción

#### 2.1. Aplicación del marco metodológico y de actividades de experimentación

---

## Capítulo 3

# Resultados

### Introducción

---

# Conclusiones

## Recomendaciones

---

## Referencias



---

# Glosario

## A

---

## **Anexos**

---

# Bibliografía

- [1] Pre-emptable shortest job next scheduling in private cloud computing in journal of information, knowledge and research computer engineering. En *Knowledge and research computer engineering*. 2015.
- [2] Dr Agarwal, Saloni Jain, et al. Efficient optimal algorithm of task scheduling in cloud computing environment. *arXiv preprint arXiv:1404.2076*, 2014.
- [3] Mohiuddin Ahmed, ASMR Chowdhury, Mustaq Ahmed, y Md Mahmudul Hasan Rafee. An advanced survey on cloud computing and state-of-the-art research issues. *IJCSI International Journal of Computer Science Issues*, 9(1):1694–0814, 2012.
- [4] KM Aranganathan, S y Mehata. An aco algorithm for scheduling data intensive application with various qos requirements. *Int. J. Comput. Appl*, 27(10):10–5120, 2011.
- [5] Huankai Chen, Frank Wang, Na Helian, y Gbola Akanmu. User-priority guided min-min scheduling algorithm for load balancing in cloud computing. En *Parallel Computing Technologies (PARCOMPTECH), 2013 National Conference on*, págs. 1–8. IEEE, 2013.
- [6] Quan Chen y Qianni Deng. Cloud computing and its key techniques. *Journal of Computer Applications*, 29(9):2565, 2009.
- [7] Dr M Dakshayini y Dr HS Guruprasad. An optimal model for priority based service scheduling policy for cloud computing environment. *International Journal of Computer Applications*, 32(9):23–29, 2011.

- 
- [8] Arash Ghorbannia Delavar, Mahdi Javanmard, Mehrdad Barzegar Shabestari, y Marjan Khosravi Talebi. Rsdsc (reliable scheduling distributed in cloud computing). *International Journal of Computer Science, Engineering and Applications (IJCSEA)* Vol, 2, 2012.
  - [9] Gordon S. Hildred T Dover, Z. The technical architecture of red hat enterprise virtualization environments. 2013.
  - [10] EPI.
  - [11] Ji Li, Longhua Feng, y Shenglong Fang. An greedy-based job scheduling algorithm in cloud computing. *Journal of Software*, 9(4):921–925, 2014.
  - [12] J Mariscal y JR Gil-Garcia. El cómputo en la nube en México: Alcances y desafíos para los sectores público y privado. cide. 2013.
  - [13] Saeed Parsa y Reza Entezari-Maleki. Rasa-a new grid task scheduling algorithm. *JDCTA*, 3(4):91–99, 2009.
  - [14] Pinal Salot. A survey of various scheduling algorithm in cloud computing environment. *IJRET: International Journal of Research in Engineering and Technology, ISSN*, págs. 2319–1163, 2013.
  - [15] Raúl Horacio Saroka. *Sistemas de información en la era digital*. Fundación OSDE, 2002.
  - [16] Er Shimpy y Mr Jagandeep Sidhu. Different scheduling algorithms in different cloud environment. *algorithms*, 3(9), 2014.
  - [17] S Srinivasan. Cloud computing evolution. En *Cloud Computing Basics*, págs. 1–16. Springer, 2014.