

**PROTOTIPO PARA SOFTWARE DE COMPETENCIAS DE PROGRAMACIÓN  
BOCA EN LA NUBE**

**YANDY TATIANA MONROY  
JOSE FERNANDO NOGUERA**



**FUNDACION UNIVERSITARIA POLITECNICO GRANCOLOMBIANO  
FACULTAD DE CIENCIAS BÁSICAS  
INGENIERA DE SISTEMAS  
PROYECTO CURRICULAR  
BOGOTÁ  
2015**

PROTOTIPO PARA SOFTWARE DE COMPETENCIAS DE PROGRAMACIÓN  
BOCA EN LA NUBE

YANDY TATIANA MONROY  
JOSE FERNANDO NOGUERA

Trabajo de Grado para optar al título de  
Ingeniero de Sistemas

Director  
Dr. Alexis Rojas Cordero  
Y  
Diego Satoba



FUNDACION UNIVERSITARIA POLITECNICO GRANCOLOMBIANO  
FACULTAD DE CIENCIAS BÁSICAS  
INGENIERA DE SISTEMAS  
PROYECTO CURRICULAR  
BOGOTÁ  
2015

Nota de Aceptación:

---

---

---

---

---

---

Firma del Director

---

---

Firma del Jurado

---

Bogotá D.C. 2015

## **AGRADECIMIENTOS**

Quiero expresar mi gratitud a Dios y a cada una de nuestras familias por darnos el apoyo y lograr el desarrollo de la presente tesis.

A nuestro tutor el Dr. Alexis Rojas por su esfuerzo, su dedicación, sus conocimientos y orientaciones.

A la Fundación Universitaria Politécnico Grancolombiano por permitirnos realizar nuestro proyecto.

# INTRODUCCION

En la presente investigación se presentan los lineamientos para establecer una oferta de los servicios de Cloud Computing y el sistema de software BOCA para maratones de programación en las universidades.

El Cloud Computing va a ser en poco tiempo un campo crítico en la evolución de las tecnologías de información, ya que ofrece eficiencia y costos reducidos en los procesos de gestión, además permite almacenar la información a través de la nube sin la necesidad de adquirir grandes equipos electrónicos.

BOCA Online Contest Administrator es un sistema de administración para celebrar competencias de programación como por ejemplo la ACM-ICPC. Este sistema se ha diseñado para utilizar PHP y PostgreSQL como backends. Las principales características que llaman la atención son: portabilidad, control de concurrencia, multi-sitio y concursos distribuidos, además de una interfaz web bastante simple e intuitiva.

Las maratones de programación promueven el aprendizaje, el trabajo en equipo, el análisis y solución de problemas, el pensamiento algorítmico y valores como la tolerancia, la honestidad y el respeto, es por esto que las universidades procuran realizar este tipo de eventos, varias veces al año en beneficio de sus estudiantes.

# LISTA DE FIGURAS

Figura No. 1 Cisco Cloud Data Center Evolution Roadmap

Figura No. 2 Nube Metáfora Computing

Figura No. 3 Infraestructura Heterogénea

Figura No. 4 Capacidad de Implementación entre las Empresas

Figura No. 5 Servicios Cloud Computing

Figura No. 6 Escalabilidad

Figura No. 7 Modelo Acoplado al Prototipo de Cloud Computing con hospedaje para Boca.

Figura No. 8 Inicio de sesión (servidor web BOCA)

Figura No. 9 Área "Site" (servidor web BOCA)

Figura No. 10 Problemas montados para la prueba de la mini maratón.

Figura No. 11 Prueba solución primer problema

Figura No. 12 Inicio de sesión como Juez

Figura No. 13 Datos de la respuesta del Equipo 1 sobre el primer problema

Figura No. 14 Respuesta correcta, tercer problema

Figura No. 15 Respuesta correcta, tercer problema.

Figura No. 16 Participantes de la Maratón de Programación Regional

Figura No. 17 Grupo de Participantes

Figura No. 18 Desarrollo de la competencia

Figura No. 19 Cantidad de Ejercicios a desarrollar

Figura No. 20 Respuestas de Autojuzgamiento

Figura No. 21 Ejercicio a juzgar

Figura No. 22 Compilación y Veredicto BOCA

# LISTA DE ANEXOS

1. Instalación apache 2
2. Instalación de PHP en el servidor web(apache)
3. Instalación del servidor de base de datos Postgresql
4. Instalación de paquetes adicionales
5. Datos de configuración de BOCA en apache
6. Datos de configuración Postgresql(/etc/postgresql/\*/main/postgresql.conf)
7. Datos de configuración Postgresql(/etc/postgresql/\*/main/pg\_hba.conf)
8. Instalación BOCA



# CONTENIDO

INTRODUCCION.....	5
LISTA DE FIGURAS.....	6
LISTA DE ANEXOS .....	8
CAPITULO I .....	11
1.    Estado del Arte .....	11
1.1.    Computación Grid .....	11
1.2.    Evolución del Almacenamiento.....	12
1.3.    Bits (b) .....	12
1.4.    Megabytes (MB).....	12
1.5.    Gigabytes (GB).....	13
1.6.    Terabytes (TB) .....	14
1.7.    Almacenamiento ilimitado .....	14
1.8.    Virtualización.....	14
CAPITULO II .....	16
2.    Cloud Computing.....	16
2.1.    La década de 1970.....	17
2.2.    La década de 1990.....	18
2.3.    El Nuevo Milenio: 2000 .....	18
3.    Arquitectura de la Nube.....	22
3.1.2.    Software como Servicio (SaaS) .....	24
4.    Cloud Server .....	25
5.    Cloud Híbrida.....	27
6.    Cloud Privada .....	29
7.    Cloud Pública.....	32
8.    Cloud Client .....	35
9.    Diseño del Modelo .....	38
CAPITULO III .....	39
3.1.    Fundamentos del Software BOCA.....	39
3.2.    ACM International Collegiate Programming Contest (ACM-ICPC).....	40
3.3.    Top Coder Algorithm Competitions Topcoder .....	40
3.4.    Resultados Parciales. (Testeo de Implementación). .....	41

3.5. Aplicación para la Implementación del Software BOCA .....	46
PROYECTOS A FUTURO.....	51
CONCLUSIONES .....	52
REFERENCIAS Y BIBLIOGRAFÍA .....	53
GLOSARIO .....	54
ANEXOS .....	58
1. Instalación apache 2.....	58
2. Instalación de PHP en el servidor web (apache).....	58
3. Instalación del servidor de base de datos Postgresql [11].....	59
4. Instalación de paquetes adicionales .....	59
5. Datos de configuración de BOCA en apache.....	60
6. Datos de configuración Postgresql(/etc/postgresql/*/main/postgresql.conf) .....	60
7. Datos de configuración Postgresql(/etc/postgresql/*/main/pg_hba.conf) .....	61
8. Datos de configuración y conexión de BOCA.....	62

# CAPITULO I

## 1. Estado del Arte

Dentro del ambiente tecnológico industrial, se entiende como "Estado del Arte", "Estado de la Técnica" o "Estado de la Cuestión", todos aquellos desarrollos de última tecnología realizados a un producto, que han sido probados en la industria y han sido acogidos y aceptados por diferentes fabricantes.

Cloud Computing se vale del Core de internet para ofrecer recursos de software y hardware, a una gran gamma de usuarios finales, desde organizaciones internacionales hasta personas del "común". En este sentido y dada su propuesta de un modelo de gestión tecnológica, que cambia la forma tradicional en el aprovisionamiento y despliegue de infraestructura de redes y recursos de hardware y software; hacia una nueva visión que los convierte en servicios, Cloud Computing se ha posicionado en el mundo de las Tecnologías de la Información y las Comunicaciones TIC's como un nuevo paradigma que parece anunciar una larga estadía.

El estado del arte de esta tecnología en constante versión beta; de manera que permita ser el preámbulo a la alquimia necesaria para poner la ciencia al servicio del ser humano.

### 1.1. Computación Grid

La Computación Grid es una tecnología innovadora que permite utilizar de forma coordinada todo tipo de recursos (entre ellos cómputo, almacenamiento y aplicaciones específicas) que no están sujetos a un control centralizado es decir una computación distribuida [1] en la cual los recursos heterogéneos como arquitecturas, supercomputadores y Cluster [2] se encuentran conectados mediante la red .El internet es desarrollado en ámbitos científicos a principios de los años 1990, su entrada al mercado comercial siguiendo la idea de la llamada Utility Computing supone una importante revolución.

El término Grid se refiere a una infraestructura que permite la integración y el uso colectivo de ordenadores de alto rendimiento, redes y bases de datos que son propiedad y están administrados por diferentes instituciones. Puesto que la colaboración entre instituciones envuelve un intercambio de datos, o de tiempo de computación, el propósito del Grid es facilitar la integración de recursos computacionales.

En la computación Grid, las redes pueden ser vistas como una forma de computación distribuida donde un “supercomputador virtual” está compuesto por una serie de computadores agrupados para realizar grandes tareas.

## **1.2. Evolución del Almacenamiento**

[3]El almacenamiento de gran cantidad de información, creó la necesidad de desarrollar dispositivos que lograrán responder a la demanda de mayor espacio y durabilidad para los datos.

### **1.3. Bits (b)**

La tecnología de almacenamiento durante la década de los 50 consistía en tarjetas perforadas que podían almacenar alrededor de 960 b de información.

### **1.4. Megabytes (MB)**

#### **1.4.1. Cinta magnética**

Surge en la década de los 60 en forma de bobina, con una longitud de hasta 4800 pies. También podía almacenar el mismo volumen de información que 10000 tarjetas perforadas.

#### **1.4.2. Disquete de 5,25"**

Su atractivo radicaba en la posibilidad de utilizarlo como dispositivo de almacenamiento más portátil para los PC domésticos. En ellos se distribuían también los juegos de PC clásicos que se popularizaron en la década de los 70.

#### **1.4.3. Disquete de 3,5"**

Estos discos mantuvieron su popularidad durante tres décadas por sus capacidades de almacenamiento, su carácter portátil y su resistencia a los daños. El disquete de 3,5" podía almacenar alrededor de 1,44 MB.

#### **1.4.4. CD**

Con una capacidad de almacenamiento 450 veces superior a la de un disquete convencional, el CD fue un producto con una gran demanda durante la década de los 90 y aún sigue utilizándose.

#### **1.4.5. Disco ZIP**

Este dispositivo podía almacenar entre 100 y 750 MB y se convirtió en la opción idónea para hacer backups de archivos.

### **1.5. Gigabytes (GB)**

Las unidades flash supusieron el siguiente paso en materia de almacenamiento de grandes volúmenes de datos, ya que pueden contener hasta 256 GB. Además de ser rentables y fáciles de utilizar, permiten almacenar una gran cantidad de datos.

## **1.6. Terabytes (TB)**

Los discos duros portátiles pueden almacenar hasta 4 TB y son útiles para hacer backups de archivos importantes o de gran tamaño.

## **1.7. Almacenamiento ilimitado**

La oportunidad de utilizar almacenamiento en Cloud ofrece a los consumidores posibilidades de almacenamiento prácticamente ilimitadas. Gracias al almacenamiento en Cloud, puede archivar, compartir, enviar y almacenar archivos, así como acceder a ellos, desde cualquier dispositivo con conexión a Internet.

## **1.8. Virtualización**

Es uno de esos neologismos emergentes que va de boca en boca entre los ejecutivos y profesionales TI y que promete ser la solución de problemas que se dan dentro de las infraestructuras de TI.

La Virtualización y Consolidación de servidores consiste básicamente en una estrategia fundamental de TI para el incremento de la utilización de plataformas, la simplificación de la infraestructura, el aumento del tiempo de servicio y la reducción de los costes de operación, así como una notable mejora en la seguridad.

A modo de introducción, hoy y con más fuerza que nunca, los responsables de las empresas de las Tecnologías de la Información (TI) deben asegurarse de que sus infraestructuras tecnológicas serán capaces de responder a los requerimientos cambiantes del mercado de manera económica, fiable y eficiente, proporcionándoles una ventaja competitiva en el mercado globalizado.

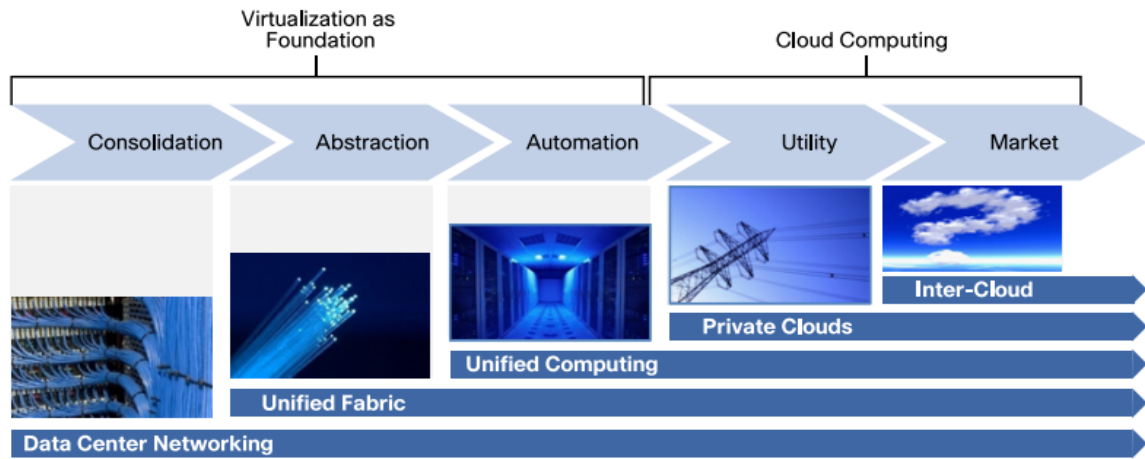


Figura No. 1 Cisco Cloud Data Center Evolution Roadmap

Fuente Cisco Center Url [https://www.cisco.com/web/strategy/docs/gov/CiscoCloud Computing\\_WP.pdf](https://www.cisco.com/web/strategy/docs/gov/CiscoCloud%20Computing_WP.pdf)

# CAPITULO II

## 2. Cloud Computing

La computación en nube, también conocido como on-demand Computing [5], es una clase de informática basada en Internet, donde se proporcionan los recursos compartidos, datos e información a los ordenadores y otros dispositivos a la carta. Es un modelo para permitir omnipresente acceso bajo demanda a un conjunto compartido de recursos informáticos configurables [4]. Soluciones de computación y almacenamiento en la nube ofrecen a los usuarios y las empresas con varias capacidades para almacenar y procesar sus datos de terceros de datos centros. Se basa en la distribución de los recursos para lograr la coherencia y las economías de escala, similar a una utilidad (como la red eléctrica) en una red. En la base de la computación en nube [9] es un concepto más amplio de la infraestructura convergente y servicios compartidos.

La computación en nube es un modelo para permitir, conveniente, acceso ubicuo a la carta de la red a un conjunto compartido de recursos informáticos configurables (por ejemplo, redes, servidores, almacenamiento, aplicaciones y servicios) que pueden ser rápidamente aprovisionados y liberados con un esfuerzo de gestión mínima.



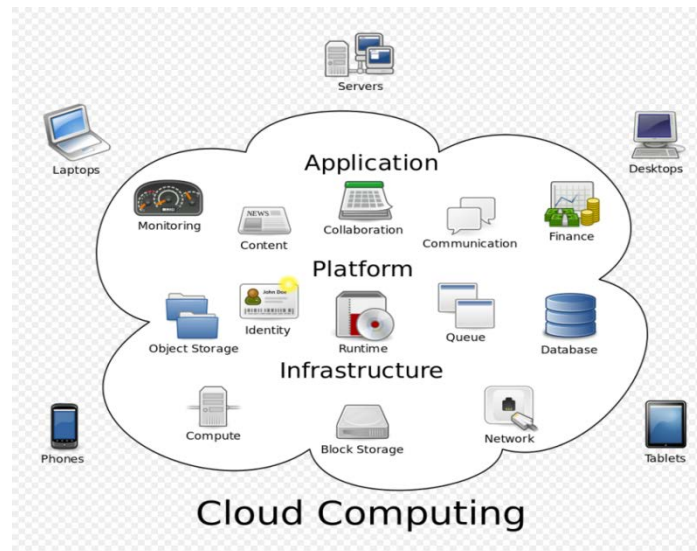


Figura No. 2 Nube Metáfora Computing

Fuente Cloud Computing Wikipedia Url [https://en.wikipedia.org/wiki/Cloud\\_Computing](https://en.wikipedia.org/wiki/Cloud_Computing)

El origen del término Cloud Computing no es claro. La expresión nube se utiliza comúnmente en la ciencia para describir una gran aglomeración de objetos que aparecen visualmente desde la distancia como una nube y describe un conjunto de cosas cuyos detalles no han sido inspeccionados más en un contexto determinado. Otra explicación es que el viejo programas que sacaban esquemas de red rodeado los iconos para servidores con un círculo, y un grupo de servidores en un diagrama de red tenido varios círculos superpuestos, que se asemejaban a una nube.

## 2.1. La década de 1970

A mediados de la década de 1970, de tiempo compartido se conoce popularmente como RJE (Remote Job Entry); esta terminología se asocia sobre todo con grandes proveedores como IBM y 12. IBM desarrolló el sistema operativo VM (primera lanzado en 1972) para proporcionar servicios de tiempo compartido a través de las máquinas virtuales.

## **2.2. La década de 1990**

En la década de 1990, las empresas de telecomunicaciones, que los circuitos dedicados principalmente de datos de punto a punto previamente ofrecidas, comenzaron a ofrecer virtuales red privada de servicios (VPN) con una calidad comparable de servicio, pero a un costo menor. Por tráfico de cambiar a su antojo para equilibrar el uso del servidor, podrían utilizar el ancho de banda de la red global con más eficacia. Comenzaron a utilizar el símbolo de la nube para indicar el punto de demarcación entre lo que el proveedor era responsable y lo que los usuarios eran responsables de la computación en nube se extiende este límite para cubrir todos los servidores, así como la infraestructura [6] de red.

Medida que las computadoras se hicieron más frecuentes, los científicos y tecnólogos explorarán formas de ganar poder de computación a gran escala a disposición de más usuarios a través de tiempo compartido. Experimentaron con algoritmos para optimizar la infraestructura, plataforma y aplicaciones para priorizar las CPU y aumentar la eficiencia para los usuarios finales.

## **2.3. El Nuevo Milenio: 2000**

Desde el año 2000 el Cloud Computing [12] ha llegado a existir. A principios de 2008, la NASA OpenNebula, mejorada en el proyecto financiado por la Comisión DEPÓSITO Europea, se convirtió en el primer software de código abierto para implementar nubes privadas e híbridas, y por la federación de nubes. En el mismo año, los esfuerzos fueron centrado en proporcionar calidad de servicio garantías (como es requerido por las aplicaciones interactivas en tiempo real) a las infraestructuras basadas en la nube, en el marco del proyecto financiado por la Comisión IRMOS Europea, dando lugar a un entorno de nube en tiempo real. A mediados de 2008, Gartner vio una oportunidad para que la computación en nube "para dar forma a la relación entre los consumidores de servicios de TI, los que utilizan los servicios de TI y los que los venden", y

observó que "las organizaciones están cambiando de hardware de propiedad de la empresa y los activos de software a los modelos basados en servicios por uso "para que el" cambio proyectado para la computación se traducirá en un crecimiento espectacular en los productos de TI en algunas áreas y reducciones significativas en otras áreas".

Microsoft Azure llegó a estar disponible a finales de 2008. En julio de 2010, Rackspace Hosting y la NASA lanzaron conjuntamente una iniciativa en la nube de software de código abierto conocido como OpenStack.

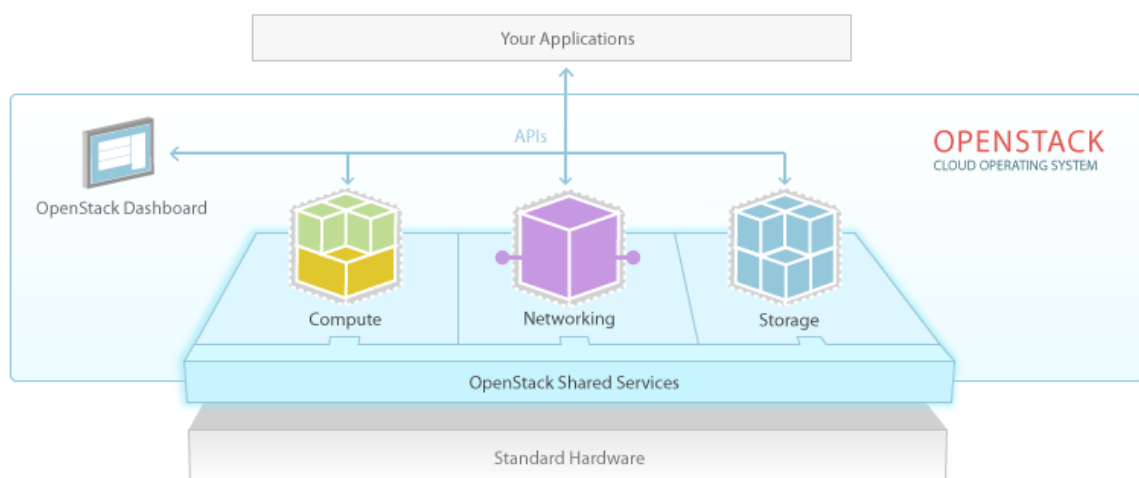


Figura No. 3 Infraestructura Heterogénea

Fuente Openstak Url <http://www.revistaCloud Computing.com/2015/03/Cloud-openstack-y-su-madurez/>

El proyecto OpenStack [7] la intención de ayudar a las organizaciones ofrecer servicios de computación en la nube se ejecutan en hardware estándar. El código temprano vino de la NASA plataforma Nebulosa así como de archivos de la nube de Rackspace plataforma.

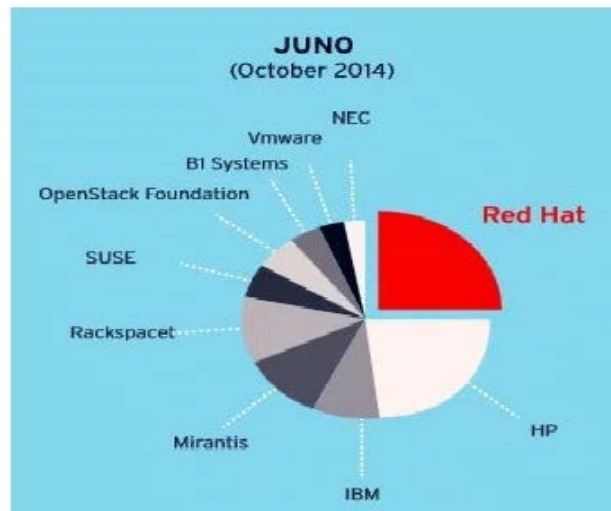


Figura No. 4 Capacidad de Implementación entre las Empresas  
 Fuente Departamento de TI Url <http://www.revistaCloud Computing.com/2015/03/Cloud-openstack-y-su-madurez/>

El 1 de marzo de 2011, IBM anunció la IBM SmartCloud marco de apoyo a Smarter Planet. Entre los diversos componentes del Smarter Computing fundación, el Cloud Computing es una pieza fundamental. El 7 de junio de 2012, Oracle anunció la nube de Oracle. Mientras que los aspectos de la nube de Oracle están todavía en desarrollo, esta oferta de nube está a punto de ser el primero en proporcionar a los usuarios el acceso a un conjunto integrado de soluciones de TI para los servicios, incluyendo en Aplicaciones (SaaS), plataforma (PaaS) e Infraestructura (IaaS).



Figura No. 5 Servicios Cloud Computing  
 Fuente Servicios de TI Url [http://www.prospecnet.com/A\\_LaNube.aspx](http://www.prospecnet.com/A_LaNube.aspx)

### **2.3.1. 2007- 2011: Los pioneros**

[8]En estos años, el mercado experimentará su fase de desarrollo. La oportunidad estará en soluciones puntuales. Se tratará de oportunidades de carácter táctico, en las que los tiempos de llegada al mercado y la productividad del desarrollo pesarán sobre la viabilidad del programa a largo plazo.

### **2.3.2. 2010- 2013: La consolidación del mercado**

A finales de junio de 2010, la consultora estadounidense, con presencia en todo el mundo, publicó un informe en que confirmaba el crecimiento de la computación en la nube. Solo los servicios vinculados a estas tecnologías registrarían ingresos de aproximadamente 68.300 millones de dólares.

A partir de 2011, el mercado ha venido mostrando una oferta mucho más amplia que se repartirá entre vendedores grandes y chicos. Para entonces y por la mayor competencia, varios de los primeros y más pequeños proveedores se verán forzados a vender o integrarse a otros más grandes. La infraestructura de servicios en general se volverá más atractiva para un mayor número de usuarios.

### **2.3.3. 2012 - 2015: Masa crítica y masificación**

De 2013 en adelante, el mercado ha estado dominado por un número más reducido de proveedores. Su oferta se centró en tecnologías de carácter propietario que fueron desarrolladas y perfeccionadas en los cinco años anteriores. Han servido como piedra angular para la entrega de interfaces que relacionan soluciones macro que vienen respondiendo y responderán a plataformas de diferentes proveedores.

En contrapartida se incrementará la oferta de Servicios de código fuente abierta. El resultado: más masa crítica y la acomodación del producto.

### **3. Arquitectura de la Nube**

La arquitectura Cloud Computing consiste de un conjunto de capas que se encuentran acopladas entre sí para brindar la funcionalidad del sistema, en este caso la arquitectura de Cloud Computing es similar a la arquitectura de red, desde un nivel físico hasta un nivel de aplicación. Esto debido a que Cloud Computing utiliza protocolos similares a los que se usan en Internet como medio de comunicación.

La nube está diseñada para proporcionar escalabilidad, desde un punto de vista conceptual, infinito. Sin embargo, no podrá aprovechar toda la escalabilidad de la infraestructura si su arquitectura no es escalable. Ambas deben trabajar mano a mano. Tendrá que identificar los componentes monolíticos y los cuellos de botella de su arquitectura, identificar aquellas áreas en las que no puede sacar partido al aprovisionamiento bajo demanda en su infraestructura, y trabajar para re-factorizar su aplicación con el objetivo de aprovechar la infraestructura escalable y sacar partido a la nube.

La computación en nube es un modelo para permitir conveniente, acceso a la red bajo demanda a un conjunto compartido de recursos informáticos configurables (por ejemplo, redes, servidores, almacenamiento, aplicaciones y servicios) que pueden ser rápidamente aprovisionados y liberados con un esfuerzo de gestión mínimos o proveedor de servicios interacción. Este modelo de nube promueve disponibilidad y se compone de cinco características esenciales (a petición de autoservicio, acceso a la red amplia, la puesta en común de recursos, elasticidad Rápido, Servicio Medido); tres modelos de servicio (la nube de software como servicio (SaaS), Cloud Platform as a Service (PaaS), Cloud Infrastructure as a Service (IaaS)); y, cuatro modelos de implementación (nube privada, nube de la Comunidad, en la nube pública, nube híbrido). Tecnologías facilitadoras clave incluyen: Redes rápidas de área

amplia, de gran alcance, equipos de servidor de bajo costo, y la virtualización de alto rendimiento para el hardware de los productos básicos.

### **3.1.1. Infraestructura como Servicio (IaaS).**

En el modelo IaaS, se parte de la idea de la externalización de servidores para espacio en disco, base de datos etc., en lugar de tener un control completo de los mismos con el DATA CENTER dentro de la empresa, u optar por un centro de datos y sólo administrarlo. Mediante este modelo de despliegue en Cloud, lo que se tiene es una solución basada en la virtualización, en la que se paga por el nivel de consumo de los recursos: espacio en disco utilizado, tiempo de CPU, espacio en base de datos, transferencia de datos.

En un modelo IaaS, un proveedor de terceros acoge hardware, software, servidores, almacenamiento y otros componentes de la infraestructura en nombre de sus usuarios. Los proveedores de IaaS también reciben solicitudes de los usuarios y manejar tareas como el mantenimiento del sistema, copia de seguridad.

Plataformas IaaS ofrecen recursos altamente escalables que se pueden ajustar a la carta por demanda. Esto hace que IaaS sea muy adecuado para las cargas de trabajo que son temporales, experimental o cambian inesperadamente.

Otras características de los entornos IaaS incluyen la automatización de tareas administrativas, escalamiento dinámico, la virtualización de escritorios y los servicios basados en políticas. Los Clientes IaaS pagan en función de cada uso, típicamente por horas, semanas o meses. Algunos proveedores también cobran a los clientes sobre la base de la cantidad de espacio de la máquina virtual que utilizan. Esto elimina el gasto de capital de desplegar en el local de hardware y software.

La ventaja más inmediata de elegir este tipo de soluciones es la de desplazar una serie de problemas al proveedor relacionados con la gestión de las máquinas y llegar a un ahorro de costes importante, ya que pagaremos solo

por lo consumido en función del nivel servicio que nos ofrezca dicho proveedor. Otro aspecto fundamental a tener en cuenta, es que las Infraestructura como servicio pueden permitir una escalabilidad automática o semiautomática, de forma que podamos contratar más recursos según los vayamos necesitando.

Como ejemplos de IaaS, se pueden citar: AbiCloud, Amazon, Web Services EC2, GoGrid, entre otros. Existen diversas soluciones de software para generar IaaS, tanto OpenSource como de ámbito privado: Vmware, Citrix, 3Tera, Abiquo, Enomaly, Eucalyptus, Proxmox, OpenNebula.org, etc.

### **3.1.2. Software como Servicio (SaaS)**

SaaS es aquella aplicación ofrecida por un fabricante de software o proveedor de servicios informáticos a través de Internet, para su uso o utilización por varios clientes.

El fabricante es el que en última instancia se ocupa del manteniendo de la privacidad de los datos y la personalización de la aplicación. En este modelo de servicio, el usuario paga por el uso y por la infraestructura necesaria (almacenamiento, seguridad, alojamiento, etc.) para el correcto funcionamiento de la aplicación y, a excepción de unos pocos parámetros de configuración, se limita a utilizar la herramienta y sus funcionalidades.

Algunos ejemplos de SaaS son: Google Apps, DocuMany, TeamBox, Kubbos, Gupigupi, Salesforce, Basecamp, Gmail, Salesforce.com, MediaWiki, Moodle, WordPress, etc.

### **3.1.3. Plataforma como Servicio (PaaS)**

Son servicios de plataformas orientadas al desarrollo, testing, despliegue, entre otros de aplicaciones del cliente. La información de los Sistemas se encuentra ubicada en la “nube pública”. Puede originar conflictos por las exigencias de seguridad en la red.



En este modelo de nube, el usuario no podrá gestionar la infraestructura de la nube, pero tendrá acceso tanto en las aplicaciones desplegadas en ella como sobre la configuración de las diversas herramientas que utilice.

Resumiendo, se puede decir que PaaS es la encapsulación de una abstracción de un ambiente de desarrollo y el empaquetamiento de una serie de módulos o complementos que proporcionan, normalmente, una funcionalidad horizontal (persistencia de datos, autenticación, mensajería, etc.). De esta forma, un arquetipo de plataforma como servicio podría consistir en una lista de componentes o APIs pre-configuradas y listas para integrarse sobre una tecnología concreta de desarrollo (por ejemplo, un servidor web, un sistema Linux, AIX, Solaris, etc. o un ambiente de programación como Perl, Fortran, C++, Java, Ruby, etc.)

Ejemplos de PaaS son: Velneo, Abiquo.com, SimpleDB SQS Google App Engine, entre otros.

#### **4. Cloud Server**

En ciertos aspectos, los servidores Cloud funcionan igual que los servidores físicos, pero las funciones que son capaces de ofrecer pueden llegar a ser muy distintas. Cuando optan por el Hosting Cloud, lo que hacen los clientes es alquilar espacio en un servidor virtual, en lugar de alquilar o comprar servidores físicos. Y ese servicio suele facturarse por horas, dependiendo de la capacidad que se necesite en un determinado momento.

Tradicionalmente ha habido dos grandes modalidades de Hosting: el Hosting compartido y el Hosting dedicado. Con el Hosting compartido, que es la opción más económica, los servidores se comparten entre los distintos clientes del proveedor de Hosting. La web de un cliente se alojará en el mismo servidor que las de otros clientes. Esto tiene varios inconvenientes, uno de los cuales es la inflexibilidad de la configuración y la incapacidad para atender grandes volúmenes de tráfico. El Hosting dedicado es una modalidad mucho más

avanzada, en la cual los clientes adquieren servidores físicos enteros, lo que significa que cada servidor está dedicado en exclusiva al cliente que lo ha comprado, y no se comparte con ningún otro cliente. En algunos casos, el cliente puede utilizar varios servidores a la vez, todos ellos dedicados también en exclusiva a ese mismo cliente, Los servidores dedicados proporcionan un control absoluto sobre el Hosting. Su desventaja es que obliga a predecir de antemano la capacidad que se va a necesitar, asignando potencia de cálculo y recursos suficientes para soportar los niveles de tráfico previstos. Si estos niveles se subestiman, podrían faltar recursos suficientes durante los períodos de mayor actividad, pero si se sobreestiman se estará pagando de más por una capacidad que en realidad es innecesaria.

Con los clientes de Hosting Cloud se consigue lo mejor de ambos mundos. Los recursos pueden ampliarse o reducirse según sea necesario, lo cual aporta una mayor flexibilidad y, por tanto, reduce costes. Y, cuando los servidores se vean sometidos a una demanda mayor, podrá incrementarse automáticamente la capacidad para adaptarse a esa demanda, sin necesidad de pagarla de forma permanente. Es lo mismo que la factura del gas: el usuario accede a lo que necesita, cuando lo necesita, y posteriormente paga sólo lo que realmente haya consumido.

A diferencia de los servidores dedicados, los servidores Cloud pueden ejecutarse sobre un hipervisor. La misión del hipervisor es controlar la capacidad de los sistemas operativos, para que pueda asignarse cuando se necesite. En los servicios de Hosting Cloud, hay varios servidores Cloud a disposición de cada cliente concreto. Esto permite asignar recursos de cómputo a un determinado cliente cuando éste lo necesite. Y si se produce una punta ocasional de tráfico, un servicio web, por ejemplo, podrá acceder a capacidad adicional durante un tiempo, hasta que deje de necesitarlo. Además, los servidores Cloud ofrecen un mayor grado de redundancia: si falla uno de los servidores, los otros tomarán su lugar.

Estas son las principales ventajas de los servidores Cloud:

- Flexibilidad y escalabilidad; es posible acceder a recursos adicionales cuando se necesite

- Economía; aunque los recursos estarán disponibles cuando se necesiten, los clientes sólo pagarán por lo que estén utilizando en un determinado momento
- Facilidad de configuración; los servidores Cloud no necesitan demasiada configuración inicial
- Fiabilidad; debido al gran número de servidores disponibles, si hubiera algún problema con alguno de ellos, el recurso implicado podrá trasladarse a otro servidor operativo, para que el clientes no noten ningún impacto.

## **5. Cloud Híbrida**

Una Cloud híbrida es un servicio Cloud integrado que utiliza Clouds tanto públicas como privadas para resolver diferentes funciones dentro de una misma organización. Todos los servicios Cloud deberían ofrecer un cierto grado de mejora en términos de eficiencia, pero el ahorro y la escalabilidad que ofrecen los servicios de Cloud pública son mayores que los de las Clouds privadas. Por tanto, una organización puede optimizar al máximo su eficiencia utilizando servicios Cloud públicos para todas las operaciones que no tengan carácter crítico, y recurrir únicamente a la Cloud privada en los casos en que sea realmente necesario, asegurándose de que todas sus plataformas queden integradas perfectamente.

Hay varias maneras de implementar modelos de Cloud híbrida:

- Colaboración entre proveedores diferentes de servicios Cloud para proporcionar servicios que combinen características de Cloud pública y privada como un servicio integrado
- Un paquete híbrido completo proporcionado por un único proveedor de servicios Cloud

- Organizaciones que gestionan sus propias Clouds privadas que contraten a su vez un servicio de Cloud pública y lo integren posteriormente en sus infraestructuras

En la práctica, una empresa podría implementar un servicio de Hosting Cloud híbrido para alojar su web de comercio electrónico dentro de una Cloud privada, pero su folleto en una Cloud pública, que resulta más económica (y donde la seguridad es un aspecto menos prioritario). Otra alternativa es optar por el modelo de Infraestructura como Servicio (IaaS, Infrastructure as a Service). Por ejemplo, una entidad financiera podría aplicar el modelo de Cloud híbrida alojando los datos de sus clientes en una Cloud privada, pero permitiendo la colaboración sobre los documentos de planificación de proyectos en la Cloud pública, a través de la cual muchos usuarios distintos podrán acceder a ellos desde el lugar que prefieran.

Una configuración de Cloud híbrida, por ejemplo para Hosting híbrido, puede ofrecer a sus usuarios las siguientes características:

- Escalabilidad; aunque no cabe duda de que las Clouds privadas ofrecen un cierto grado de escalabilidad, dependiendo de la configuración de que se trate (por ejemplo, de si está alojada internamente o externamente), los servicios Cloud público ofrecen escalabilidad con unas fronteras menos delimitadas, ya que los recursos proceden de la amplia infraestructura de la Cloud pública. Trasladando a la Cloud pública tantas funciones no críticas como sea necesario, la organización podrá beneficiarse de las ventajas en términos de escalabilidad que proporciona la Cloud pública y a la vez reducir la demanda ejercida sobre la infraestructura Cloud privada.
- Economía; una vez más, es mucho más probable que una Cloud pública proporcione mayores economías de escala (como la gestión centralizada), y por tanto suponga un mayor ahorro económico, que una Cloud privada. Por tanto, las Clouds híbridas permiten a las organizaciones beneficiarse de esos

ahorros en tantas funciones del negocio como sea posible, y a la vez mantener la seguridad para sus operaciones más críticas.

- Seguridad; el componente de Cloud privada dentro de un modelo de Cloud híbrida no sólo aporta la seguridad necesaria para las actividades más críticas, sino que además facilita el cumplimiento de la normativa aplicable en materia de manejo y almacenamiento de datos.
- Flexibilidad; poder disponer tanto de un recurso seguro de naturaleza privada como de un recurso público escalable y más económico brinda a las organizaciones mayores oportunidades a la hora de explorar distintos planteamientos para su operativa diaria.

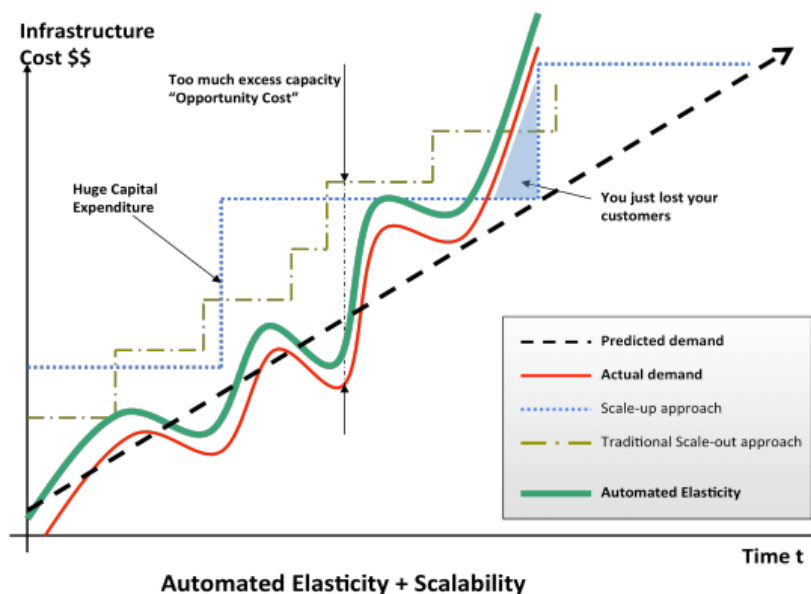


Figura No. 6 Escalabilidad

## 6. Cloud Privada

Una Cloud privada es una modalidad particular de plataforma Cloud constituida por un entorno Cloud diferenciado y seguro que sólo el cliente especificado puede utilizar. Al igual que en otras modalidades de Cloud, las Clouds privadas proporcionan capacidad de procesamiento como un servicio, que se ejecuta

dentro de un entorno virtualizado cuyos recursos proceden de un repertorio subyacente de dispositivos informáticos físicos. Sin embargo, en el modelo de Cloud privada, la plataforma Cloud (el repertorio de recursos) sólo está accesible para una organización concreta, lo cual le proporciona un mayor grado de control y privacidad.

Los mecanismos técnicos que se utilizan para proporcionar los diferentes servicios que pueden clasificarse dentro de la categoría de servicios Cloud privados pueden variar considerablemente, por lo que no resulta fácil definir qué es lo que constituye una Cloud privada desde el punto de vista técnico. En lugar de plantearlo de esta manera, este tipo de servicios suelen categorizarse por las funcionalidades que ofrecen a sus clientes. Rasgos característicos de las Clouds privadas son, por ejemplo, el aislamiento de los recursos Cloud para su uso exclusivo por parte de una única organización concreta, y el mayor nivel de seguridad, en contraposición a las Clouds públicas, en las cuales hay muchos clientes accediendo a servicios virtualizados que toman sus recursos del mismo banco de servidores y acceden a ellos a través de redes públicas. Los servicios de Cloud privada toman sus recursos de un repertorio diferenciado de ordenadores físicos, aunque éstos pueden estar alojados tanto dentro como fuera de las instalaciones de la empresa, y el acceso a ellos puede realizarse a través de líneas punto a punto privadas o de conexiones cifradas seguras que discurren por redes públicas.

El grado adicional de seguridad que proporciona el modelo Cloud con aislamiento lógico entre clientes resulta ideal para cualquier organización, por ejemplo una gran empresa, que necesite almacenar y procesar datos privados o realizar tareas delicadas. Por ejemplo, una entidad financiera podría estar interesada en utilizar un servicio de Cloud privada para almacenar internamente los datos más confidenciales, como exige la legislación, sin dejar de disfrutar dentro de su infraestructura de negocio de algunas de las ventajas que ofrece el modelo Cloud, como la asignación de recursos a demanda.

El modelo de Cloud privada se acerca más a la filosofía tradicional de funcionamiento de las redes locales individuales que han venido utilizando hasta ahora las empresas, pero con las ventajas adicionales que aporta la virtualización. Estas son, en resumen, las características y ventajas de las Clouds privadas:

- Mayor seguridad y privacidad; los servicios de Cloud pública pueden implementar un cierto nivel de seguridad, pero las Clouds privadas pueden garantizar que la operativa interna de la empresa quede fuera del alcance de ojos indiscretos, gracias a técnicas como el uso de grupos de recursos físicamente diferenciados con acceso restringido a aquellas conexiones que se establezcan desde dentro del firewall de la organización, a través de líneas punto a punto, y/o desde los entornos locales de Hosting de la propia empresa.
- Más control; como una Cloud privada sólo es accesible para una única organización, ésta puede configurarla y administrarla según sus propias necesidades, para obtener una solución de red completamente a medida. No obstante, con este grado de control desaparecen también algunas de las economías de escala que caracterizan a las Clouds públicas con su gestión centralizada del hardware
- Economía y eficiencia energética; implementar un modelo de Cloud privada puede mejorar la asignación de recursos dentro de una organización, garantizando su disponibilidad para que cada uno de los departamentos o áreas funcionales del negocio puedan responder a su demanda de una forma directa y flexible. Por tanto, aunque no sean tan económicos como los basados en Cloud pública, porque las economías de escala son menores, y mayores los costes de administración, lo que sí consiguen es aprovechar los recursos informáticos mejor que las redes locales tradicionales, ya que reducen al mínimo la necesidad de invertir en capacidad que vaya a quedar sin utilizarse, lo cual no sólo supone un ahorro económico, sino que además reduce la huella de carbono de la organización.

- Mayor fiabilidad; incluso en los casos en que los recursos (servidores, redes, etc.) se almacenan internamente, la creación de entornos virtualizados hace a la red más resistente ante fallos individuales ocurridos en cualquier posición dentro de su infraestructura física. Por ejemplo, las particiones virtuales pueden tomar sus recursos de los servidores restantes que no se hayan visto afectados por el fallo. Además, cuando la infraestructura Cloud reside en un proveedor externo, la organización puede seguir beneficiándose de la seguridad física que proporciona el hecho de que esté alojada en centros de datos.
- Puntas ocasionales de tráfico; en un servicio de Cloud privada, algunos proveedores pueden ofrecer la posibilidad de incrementar puntualmente la capacidad por encima del nivel contratado, para atender puntas ocasionales de demanda. Este servicio, conocido también como "bursting", permite al proveedor trasladar a una Cloud pública determinadas funciones menos críticas, con objeto de liberar espacio en la Cloud privada para las funciones más delicadas que puedan necesitarlo. Las Clouds privadas pueden incluso integrarse con los servicios de Cloud pública para constituir Clouds híbridas en las cuales las funciones no críticas se alojen siempre en la Cloud pública, para aprovechar al máximo las eficiencias que proporciona.

## **7. Cloud Pública**

El modelo Cloud más reconocible para la mayoría de los consumidores es el de Cloud pública, en el cual los servicios Cloud se ofrecen dentro de un entorno virtualizado construido utilizando un repertorio de recursos físicos compartidos entre varios servicios, y el acceso a ellos se realiza a través de una red pública como internet. En cierta medida, pueden definirse contrastándolas con las Clouds privadas, en las cuales el repertorio de recursos informáticos subyacentes está protegido y aislado del exterior, creando con ello una plataforma Cloud a la cual sólo tiene acceso una única organización. Las Clouds públicas, por el contrario, ofrecen servicios a varios clientes utilizando la misma infraestructura compartida.



Los ejemplos más destacados de plataformas Cloud suelen enmarcarse dentro del modelo de Cloud pública, porque, por definición, están disponibles para el público. Las propuestas de Software como Servicio (SaaS), como los servicios de alojamiento en la nube o las aplicaciones ofimáticas accesibles online, son quizás las más conocidas por los usuarios, pero están también disponibles a gran escala otras ofertas de Infraestructura como Servicio (IaaS, Infrastructure as a Service) y Plataforma como Servicio (PaaS), como los entornos de desarrollo y Hosting gestionados a través de la web, que pueden obedecer a ese mismo modelo (aunque también pueden residir en Clouds privadas). Las Clouds públicas son muy utilizadas para los servicios orientados a particulares, que generalmente no necesitan todo el nivel de infraestructura y seguridad que ofrecen las Clouds privadas. No obstante, también las empresas pueden utilizar Clouds públicas para mejorar considerablemente la eficiencia de sus operaciones, por ejemplo para almacenar contenidos no especialmente delicados, para el manejo colaborativo de documentación y para el correo web.

El modelo público ofrece las siguientes características y ventajas:

- Máximo grado de escalabilidad; los recursos Cloud están disponibles a demanda, a través de los enormes repertorios de recursos que componen las Clouds públicas, lo que permite a las aplicaciones que se ejecutan sobre ellos responder con total fluidez a las fluctuaciones en la actividad
- Economía; las Clouds públicas reúnen mayores volúmenes de recursos, lo que les permite beneficiarse de mayores economías de escala. La centralización de la gestión y el mantenimiento de los recursos subyacentes permite compartir costes entre todos los servicios Cloud que dependen de ellos, mientras que los componentes físicos, como los servidores, necesitan menos trabajo de configuración a medida. Algunos modelos de servicio disponibles en el mercado pueden incluso ser gratuitos para el cliente, al utilizar la publicidad como fuente de ingresos para sufragar su coste.

- Tarificación similar a la de los suministros públicos. como la luz y el gas; los servicios basados en Cloud pública suelen aplicar un modelo de tarificación en función del consumo, que permite al cliente acceder al recurso que necesite, en el momento en que lo necesite, y a partir de ese momento pagar sólo por lo que realmente utilice, evitando desaprovechar capacidad contratada.
- Fiabilidad; por el enorme número de servidores y redes que intervienen en la creación de una Cloud pública, y la redundancia en sus configuraciones, en caso de fallo en un componente físico concreto, el servicio Cloud podrá seguir funcionando sin interrupción sobre los componentes restantes. En algunos casos en los que los recursos de la plataforma Cloud proceden de varios centros de datos, incluso un centro de datos completo podría quedar fuera de servicio sin que los servicios Cloud individuales sufrieran un impacto significativo. En otras palabras, no existe un punto único de fallo que pueda hacer vulnerable a un servicio basado en Cloud pública.
- Flexibilidad; existen en el mercado infinidad de servicios IaaS, PaaS y SaaS que están basados en el modelo de Cloud pública y a los que se puede acceder directamente desde cualquier dispositivo que tenga conexión a internet. Estos servicios pueden satisfacer la mayoría de las necesidades de procesamiento y ofrecer sus ventajas a todo tipo de clientes, tanto particulares como empresas. Las empresas pueden incluso integrar sus servicios de Cloud pública con Clouds privadas, en aquellos casos en que necesiten realizar funciones especialmente críticas para la empresa, creando así Clouds híbridas.
- Independencia de la localización; el hecho de que los servicios de Cloud pública estén disponibles a través de una conexión a internet significa que cualquier usuario podrá acceder a ellos independientemente del lugar donde se encuentre, lo cual brinda valiosas oportunidades a las empresas, por ejemplo la posibilidad de acceder de forma remota a su infraestructura informática (en caso de emergencia o situaciones similares) o

de colaborar a través de internet en la elaboración de documentos desde varios lugares distintos.

## **8. Cloud Client**

Un cliente de la nube se compone de hardware y / o software que se basa en la computación en nube para la entrega de aplicaciones, o que está diseñado específicamente para la entrega de servicios en la nube y que, en cualquier caso, es esencialmente inútil sin ella. Los ejemplos incluyen algunos ordenadores, teléfonos y otros dispositivos, sistemas operativos y navegadores.

Computación cliente Cloud es para todo el mundo en busca de una solución de TI que reemplaza los sistemas obsoletos del PC con un PC simplificada que se ejecuta en un entorno de nube virtualizada.

La computación en nube es cliente de la solución de TI moderna para centralizada infraestructuras, ofreciendo reducen el TCO, simplificaron Gestión, mejorar la seguridad, la funcionalidad, la movilidad y flexibilidad para finales alternativos dispositivos ya sea de hardware o firmware.

Mejor eficiencia Computación cliente en nube es la mejor alternativa a la herencia modelo de computación de clientes pesados, que son más caros, tienen una huella de carbono más grande y representan un mayor riesgo de seguridad. El nuevo modelo de computación de los clientes en la nube ofrece total seguridad, la fiabilidad y la experiencia del usuario en un coste total de propiedad y el consumo energético mínimo. Computación cliente Nube abarca todo, desde software de cliente de la nube y el hardware a los servicios.

Tecnología probada Una serie de organizaciones y empresas ya utilizar clientes en la nube, también conocido como delgada y cero clientes, y nube PCs. Todas las clases de productos están respaldados por la nube, virtualización y software de gestión de suites. Cliente Nube Computing

simplemente conecta todos los puntos: delgada y cero computación cliente, comunicaciones unificadas, virtualización de escritorio y la Web, por lo que puede llegar a las nubes en una organización privada, pública, del gobierno o de nube híbrida. Un comprensivo software y el hardware de baño, ya que ofrece por Wyse, da la completa libertad del usuario para acceder a todo el contenido, todo aplicaciones en cualquier entorno, desde cualquier lugar y sin limitación ni conflictos. Es en los negocios. Está en casa. Es sobre la marcha.

Global Computación cliente Nube conduce no sólo a la mejora de la eficiencia y una mayor productividad, sino que también contribuye a la cambio de paradigma positivo de la sociedad moderna. Cliente Nube Computing reduce los costos, complejidad y elimina hace que sea posible acceder a cualquier dispositivo final desde cualquier parte en el mundo.

Esta es la época del año para pontificar sobre el año pasado y hacer predicciones para el futuro. Hay un montón de tendencias que se están hablando en este contexto, aquellos cuyo tiempo ha llegado y otros que están empezando a mirar a través y prometedores para el futuro.

Me gustaría hablar de uno en particular, que creo que va a tener un impacto significativo en la industria en general y perturbar el espacio de entrega de aplicaciones, en particular, la creación de la próxima modelo de cálculo y aplicación. Nube / Client Computing.

Las semillas se sembraron en 2014 y yo iremos por las ramas para decir que 2015 será el año de Cloud Computing / Cliente.

Cloud y arquitecturas móviles están permitiendo una nueva forma de diseñar y entregar aplicaciones y contenidos que proporcionan una rápida comercialización, la experiencia del usuario final personalizado, y la mejora de la agilidad del negocio. Con el aumento de las expectativas de los consumidores para contenido en línea que se puede acceder desde una gran variedad de dispositivos móviles, desde cualquier lugar y en cualquier

momento, las empresas tienen que repensar su estrategia en línea o el riesgo está quedando atrás y perder ventaja competitiva.

En este nuevo mundo en el que los departamentos de TI a aprovechar la nube privada y pública, y las aplicaciones están diseñadas móvil primero, basado en una arquitectura de nube / cliente, nos enfrentamos a nuevos e interesantes desafíos en torno a la velocidad, la seguridad y la escala.

Soluciones de entrega de aplicaciones legado que fueron diseñados en una arquitectura centralizada no pueden hacer frente. Para acelerar la creación de empresas de aplicaciones utilizadas tradicionalmente servicios de CDN que aceleraron la entrega de contenido desde el centro de datos hasta el borde de la Internet. Ahora con la nube / cliente, el cuello de botella tiene shifted- la última milla inalámbrica entre el borde y el cliente ahora es el cuello de botella de rendimiento de capital.

Para asegurar las aplicaciones, las empresas utilizan aplicaciones de firewall que protegían el perímetro del centro de datos donde la aplicación corrió. Con arquitectura de nube / cliente, el perímetro de la red ha desaparecido y la superficie de ataque ha aumentado para incluir a clientes y tercera backend fiesta servicios, que están más allá del alcance del firewall. Ahora necesita una solución unificada, integral que ofrece la vista y control de extremo a extremo.

Para escalar las aplicaciones, las empresas utilizan un ADC (Application Delivery Controller) que funcionaba como un equilibrador de carga y reduce el coste de la ampliación de la aplicación. Con la nube / arquitectura cliente los servicios de back-end se ejecutan a través de diferentes nubes y por lo tanto, junto con los servicios al cliente, hacen de electrodomésticos ADC ineficiente.

Para hacer frente a estos desafíos, un servicio de entrega de aplicaciones en sí tiene que estar basada en una nube / arquitectura cliente. Software-Defined entrega de aplicaciones (SDAD), un enfoque innovador fuimos pioneros, es la solución de infraestructura de Cloud Computing / cliente. Se trata de una

solución de entrega única que se puede utilizar para aplicaciones de velocidad, seguras y escala.

Gartner está de acuerdo claramente conmigo en base a sus 10 principales tendencias estratégicas de tecnología en 2015.

## 9. Diseño del Modelo

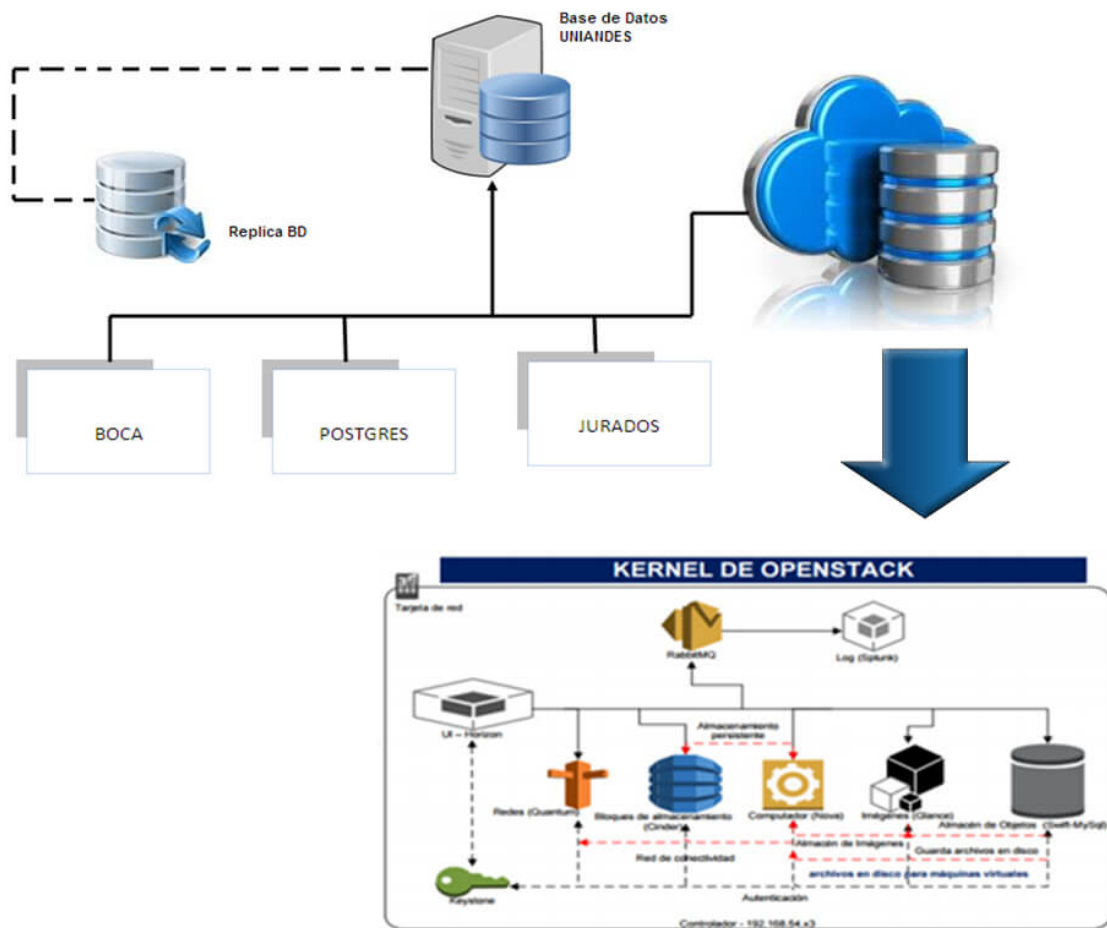


Figura No. 7 Modelo Acoplado al Prototipo de Cloud Computing con hospedaje para Boca.  
Fuente Tesis Modelo, diseño y técnicas básicas para implantar un sistema de computación en la nube

# CAPITULO III

## 3.1. Fundamentos del Software BOCA

Con el objetivo de ampliar y mejorar la participación de los estudiantes del programa de Ingeniería de Sistemas en las competencias oficiales de maratones de programación se tiene planeada la conformación de un grupo de estudio que va ser parte del Semillero de Investigación SILUX, en el cual se van a preparar y aprender las estrategias para afrontar una competencia oficial de programación, en este grupo pueden participar estudiantes de cualquier semestre.

El objetivo de esta actividad es adquirir habilidades y conocimientos que nos permitan prepararnos para la maratón nacional que se llevará a cabo en el mes de septiembre en la ciudad de Bucaramanga, a la cual clasifican 3 equipos del programa. La competencia se realiza en equipos de tres personas, se recomienda que por lo menos uno de los tres tenga habilidades para la traducción del enunciado de los ejercicios. Se proponen entre 8 y 10 ejercicios de diferente dificultad los cuales deben ser desarrollados en el menor tiempo posible. Cada equipo dispone de un computador con las herramientas necesarias para implementar los problemas. Los lenguajes de desarrollo son Java, C y C++.

Este es un sistema de administración de los concursos celebrados de programación (por ejemplo, ACM-ICPC, Maratón de Programación da SBC). Se ha diseñado para utilizar PHP y PostgreSQL como backends. Las principales características deseadas son: portabilidad, control de concurrencia, multi-sitio y concursos distribuidos, simple interfaz web [5].

### **3.2. ACM International Collegiate Programming Contest (ACM-ICPC)**

Cualquier estudiante que este matriculado en una universidad puede llegar a participar en estas competencias, habiendo restricciones sobre el número de veces que el concurso permite a cada estudiante participar en cada una de las rondas de la competencia. Los participantes compiten en equipos de tres integrantes representando solamente una institución. La configuración de las rondas está dada, por una clasificación previa en la universidad, una posible competencia subregional.

### **3.3. Top Coder Algorithm Competitions Topcoder**

Se caracteriza por ser un sitio online que administra diferentes maratones de programación, estableciendo en promedio competencias cada 2 semanas. Cualquier persona puede participar en Topcoder excluyendo sus propios empleados, aunque hay restricciones en las ocasiones en que se entregan premios en efectivo, donde solo pueden participar personas mayores a 18 años. Una competencia consiste en varias rondas; en la primera ronda los participantes se les entrega tres problemas de naturaleza algorítmica con diferentes grados de dificultad y por ende diferentes puntuaciones por la solución de cada problema, contando con 75 minutos para resolverlos.

Tipos de problemas A la hora de un participante abordar un problema, es necesario analizar inicialmente la naturaleza del problema y por ende su complejidad, dependiendo de esto se podría determinar que técnicas o algoritmos seguir con el fin de llegar a una solución óptima para el problema.

1. Programación Dinámica
2. La ruta más corta
3. Técnicas de búsquedas recursivas
4. Problemas geométricos
5. Problemas con números grandes
6. Búsquedas de aproximación
7. Búsquedas Heurísticas



### 3.4. Resultados Parciales. (Testeo de Implementación).

Para probar la funcionalidad del sistema BOCA [13] separado en 3 (tres) máquinas virtuales, se realizó una maratón de programación, en la cual existieron 2 (dos) equipos los cuales tenían que resolver una serie de problemas, a continuación el proceso y los resultados de la mini maratón:

1. Iniciamos sesión (Fig. No. 8) como administradores en el aplicativo, para activar un “contest” para empezar con la maratón:

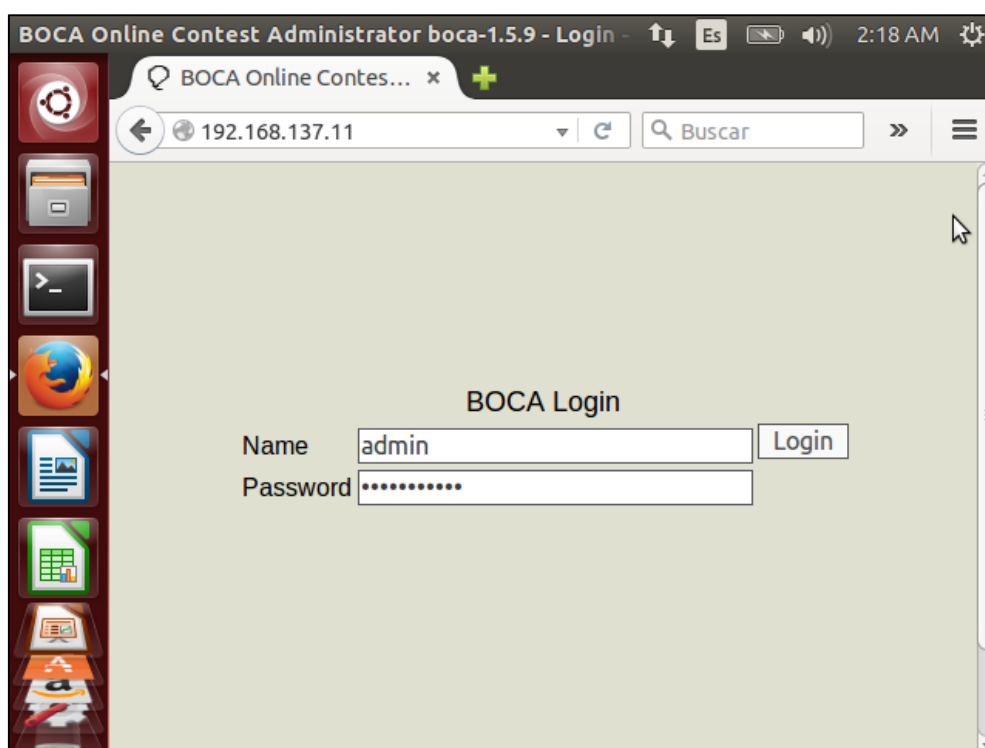


Figura No. 8 Inicio de sesión (servidor web BOCA).  
Fuente Autores del Libro

2. Ahora ingresamos al área de “Site” (Fig. No. 9), para activar un nuevo sitio con un “contest” activo:

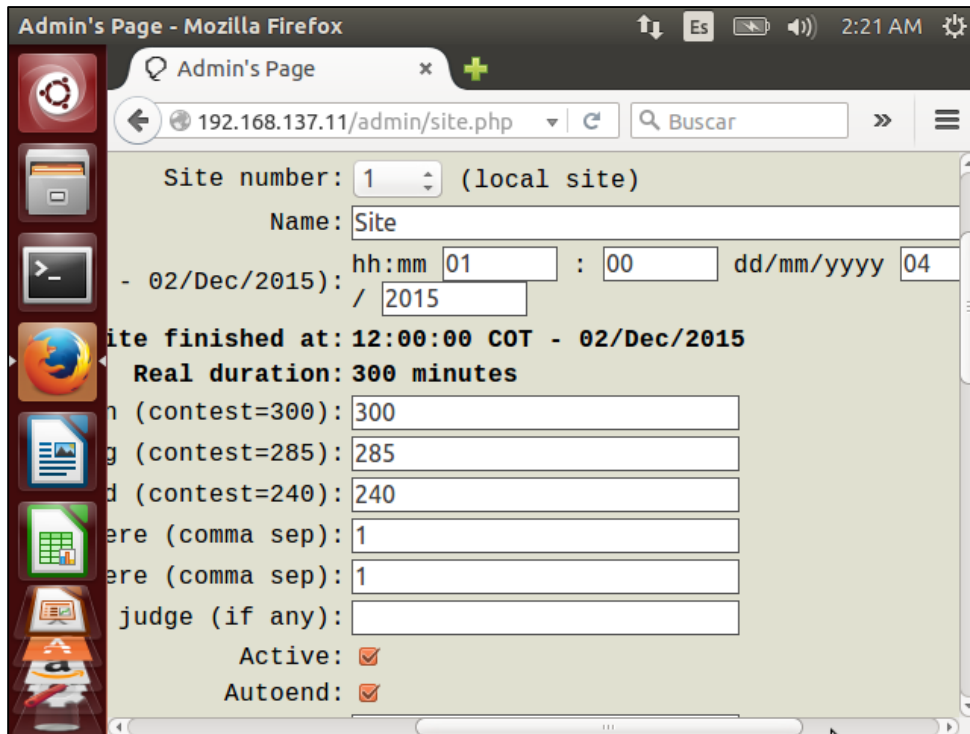


Figura No. 9 Área "Site" (servidor web BOCA).  
Fuente Autores del Libro

- Una vez creado nuestro "Site", ahora debemos montar los problemas que se resolverán en la maratón, para esta prueba solo montamos 3 (tres) problemas (Fig. No. 10) para resolver:

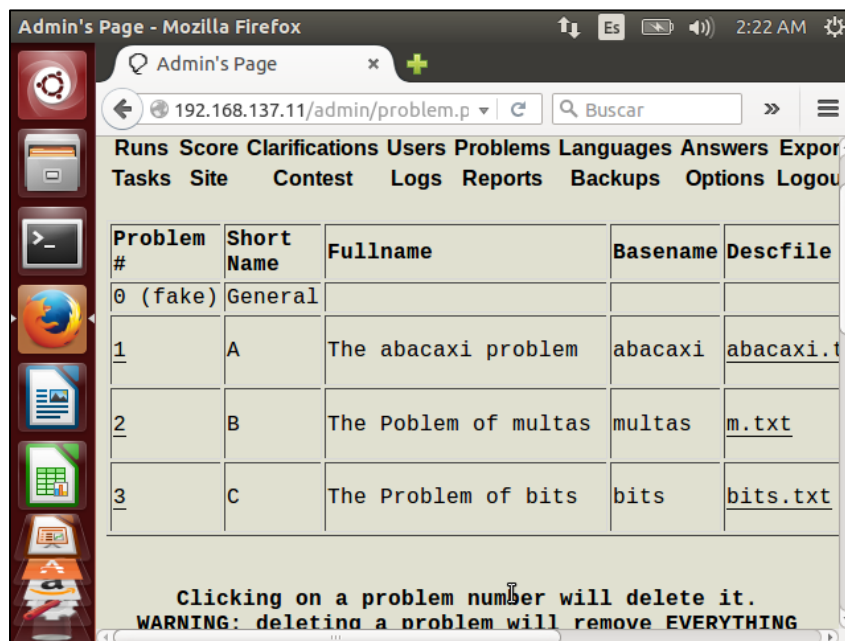


Figura No. 10 Problemas montados para la prueba de la mini maratón.  
Fuente Autores del Libro

- Una vez que el “Contest” está activo, iniciamos sesión con los equipos participantes para empezar a resolver los problemas, para esto se crearon 2 (dos) usuarios de tipo “team”, ingresamos con el primer equipo y montamos una prueba para el primer problema (Fig. No. 11):

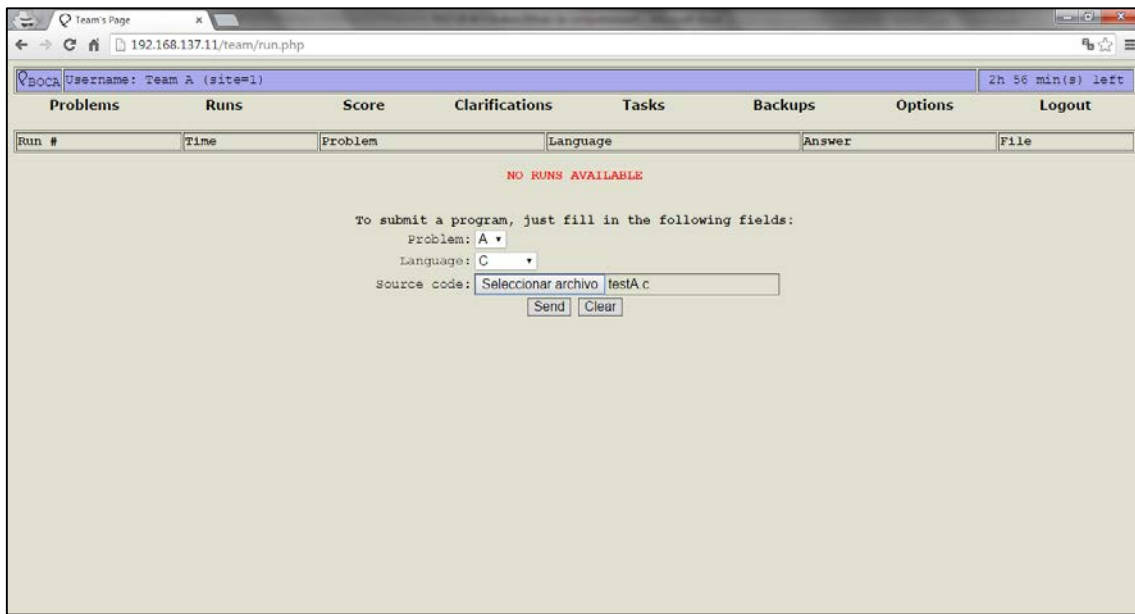


Figura No. 11 Prueba solución primer problema.  
Fuente Autores del Libro

- Ingresamos con una cuenta de Juez (Fig. No. 12) para validar la solución del Equipo 1 al primer problema:

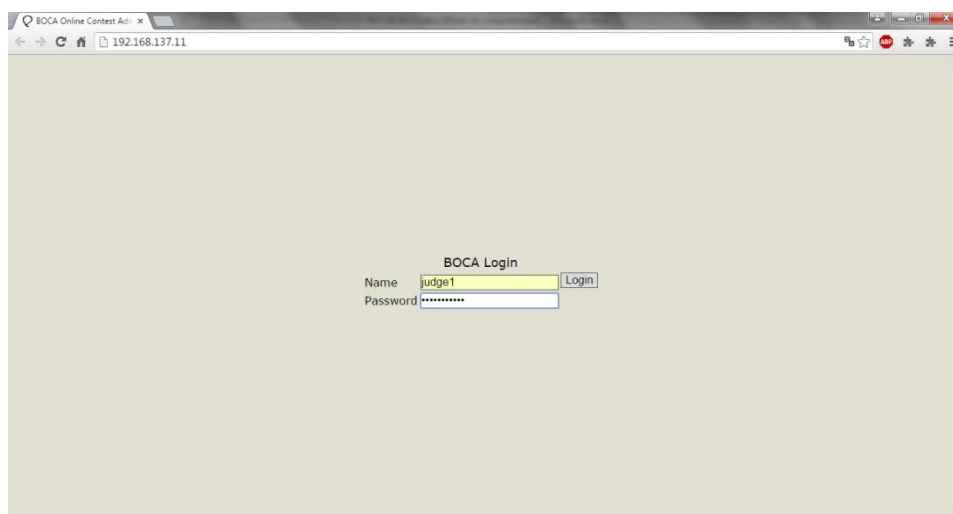


Figura No. 12 Inicio de sesión como Juez.  
Fuente Autores del Libro

- Ingresamos al área de “Runs” para validar la respuesta de Equipo1 sobre el primer problema y seleccionamos el “run” del usuario:

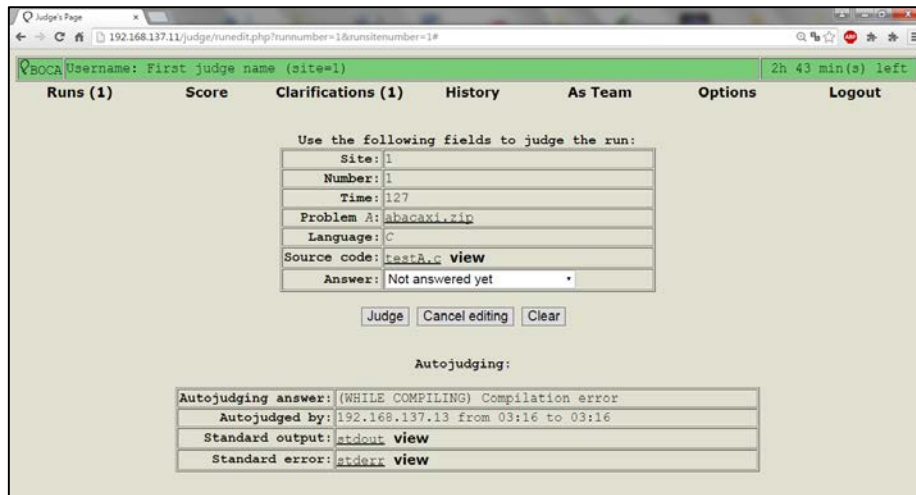


Figura No. 13 Datos de la respuesta del Equipo 1 sobre el primer problema.  
Fuente Autores del Libro

Como se observa en la Figura No. 13, el auto-juzgador está funcionando correctamente y nos una respuesta de “Error de compilación”, el auto-juzgador ayuda a los jueces a tomar decisiones sobre las soluciones montadas por los usuarios.

- Ahora realizamos una prueba, montando una respuesta correcta al tercer problema.

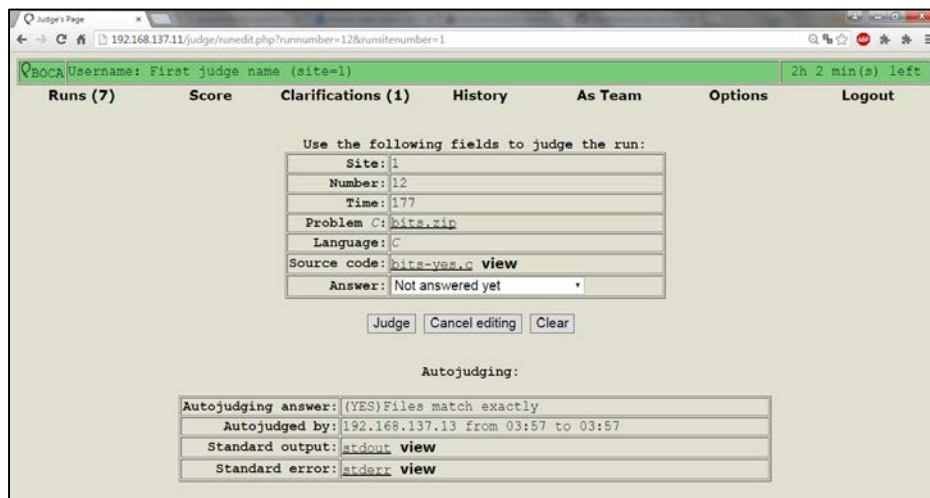
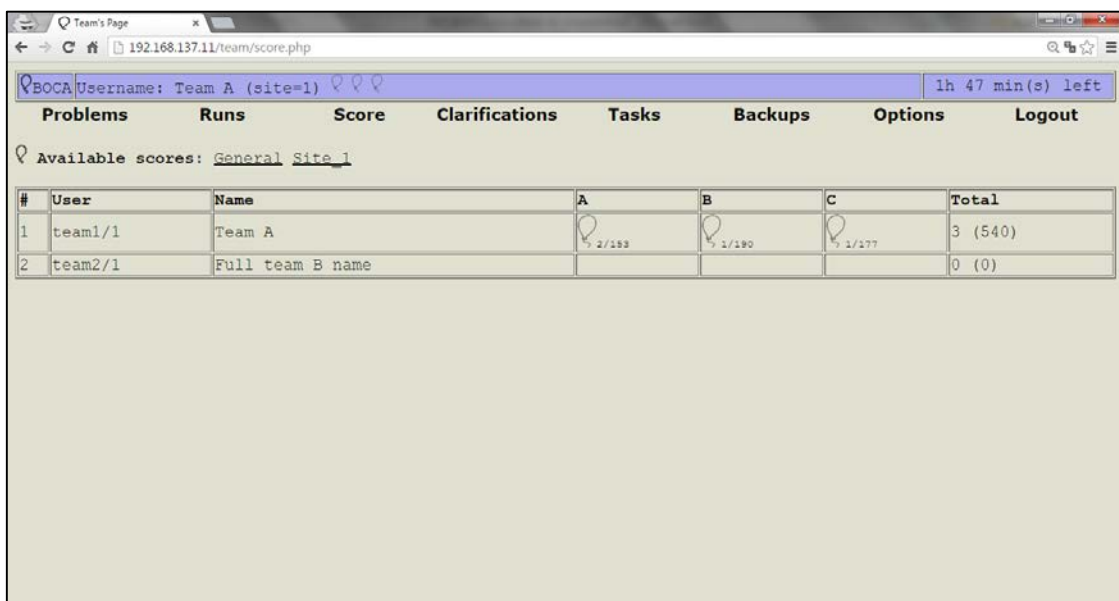


Figura No. 14 Respuesta correcta, tercer problema.  
Fuente Autores del Libro

Como se ve en la figura No. 14 en el área de “Autojudging answer”, el auto-juzgador nos respondió de manera afirmativa a la solución montada por el Equipo1 para el tercer problema.

Al realizar varias pruebas, podemos comprobar que el sistema funciona correctamente, tanto la creación de “Contest” para las maratones, la creación de problemas para que los equipos los resuelvan, la participación del servidor Juez al momento de auto-juzgar y la participación de los usuarios jueces.



#	User	Name	A	B	C	Total
1	team1/1	Team A	2/183	1/180	1/177	3 (540)
2	team2/1	Full team B name				0 (0)

Figura No. 15 Respuesta correcta, tercer problema.  
Fuente Autores del Libro

En la figura No. 15 se puede observar como el Equipo1 de nombre “Team A” ha finalizado con los 3 (tres) problemas planteados para esta mini maratón, con esto damos por concluida la prueba del BOCA.

### 3.5. Aplicación para la Implementación del Software BOCA



Figura No. 16 Participantes de la Maratón de Programación Regional

Fuente: Página de la IUPG, Url <https://www.facebook.com/poligran/photos/pb.310454127055.-2207520000.1449188702./10153117934082056/?type=3&theater>

Se efectuó una maratón de programación el día 14 de noviembre, los mejores programadores de Latinoamérica se reunieron en el campus del Politécnico Gran Colombiano para resolver durante cinco horas la mayor cantidad de problemas mediante programación de computadores.

Este importante encuentro auspiciado por IBM, ACIS [10] (Asociación Colombiana de Ingenieros de Sistemas) y REDIS (Red Nacional de Programas de Ingeniería de Sistemas).

1.075 Equipos, 3.225 participantes, más de 600 entrenadores y 600 universidades de 12 países en Latinoamérica como se ve en la figura No. 17 este fue uno de los equipos participantes, probaran sus habilidades en la solución de problemas compitiendo en simultánea contra los mejores programadores de Argentina, Bolivia, Brasil, Chile, Cuba, Ecuador, Haití, Jamaica, México, Panamá, Perú, República Dominicana, Uruguay y Venezuela. Un reto que les permitirá a los ganadores representar a su país en la Maratón

Mundial que se realizará en la ciudad de Phuket, Tailandia, del 15 al 20 de mayo de 2016.



Figura No. 17 Grupo de Participantes

Fuente Página de la IUPG, Url <https://www.facebook.com/poligran/photos/pb.310454127055.-2207520000.1449188719./10153117933337056/?type=3&theater>

“En estas competencias participan estudiantes de ingeniería a quienes les gustan las matemáticas, son hábiles trabajando en equipo y compiten resolviendo problemas en un tiempo limitado; Colombia ha tenido gran representación en los mundiales de programación, a pesar del déficit de ingenieros que hemos registrado en los últimos años”, comentó Rafael García, Decano de la Facultad de Ingeniería y Ciencias Básicas.

Los estudiantes de ingeniería de sistemas más sobresalientes en esta disciplina, mostrarán sus habilidades y destrezas para la ejecución de códigos en donde las empresas desarrolladoras de software como: Google, Microsoft, IBM, Facebook y Twitter, convocan a los participantes más destacados de las competencias mundiales de programación y los invitan a formar parte de su equipo de trabajo.

REDIS, red académica que reúne a 99 universidades de Colombia, genera estos espacios de motivación y de esta manera establece dinámicas que promueven el interés en la formación en Ingeniería de Sistemas.

“El desarrollo de un país está ligado a la profesionalización de sus jóvenes en las diferentes disciplinas, el reto para nosotros está en motivar y establecer espacios para estas competencias, en donde las grandes empresas multinacionales están atentas a identificar a los mejores programadores”, agregó el Decano.

El CCPL (Colombian Collegiate Programming League) junto con Google, realizan un evento anual para conocer a los mejores programadores del país; principalmente se fijan en ellos desde lo que realizan y cómo se proyectan.

Las competencias de programación son la mejor experiencia para los jóvenes participantes; aprender, divertirse y dar lo mejor para aprovechar el crecimiento personal y profesional como se observa en la figura No. 18 en la que podemos ver que la decoración de los globos hace referencia al software Boca el cual representa cada problema con globos de colores. El Politécnico Grancolombiano realiza este evento como una de las diversas acciones para apoyar e incentivar a los estudiantes de ingeniería del país.



Figura No. 18 Desarrollo de la competencia

Página de la IUPG, Url <https://www.facebook.com/poligran/photos/pb.310454127055.-2207520000.1449188702./10153118330317056/?type=3&theater>



Los diferentes participantes evidencian los ejercicios a desarrollar en el aplicativo como se observa en la figura No. 19

Problem #	Short Name	Fullname	Username	Descfile	Package file	Answers Options	Export Logout
0 (fake)	General						
1	A	Tornado	tornado	tornado.pdf	problemTornado.zip	Color	
2	B	Odd or Even	odd	odd.pdf	problemOdd.zip	black	
3	C	Weekend Lottery	lottery	lottery.pdf	problemLottery.zip	blue	
4	D	He is offside	offside	he.pdf	problemCr74J_d.zip	greenblue	
5	E	Feynman	feynman	feynman.pdf	problemFeynman.zip	green	
6	F	Almost Shortest Path	path	path.pdf	problemAlmost.zip	red	
7	G	Electricity	electricity	electricity.pdf	problemElectricity.zip	white	
8	H	Pole Position	pole	pole.pdf	problemPole.zip	yellow	

Clicking on a problem number will delete it.  
 WARNING: deleting a problem will remove EVERYTHING related to it.  
 It is NOT recommended to change anything while the contest is running.  
 To import a problem, fill in the following fields.  
 To replace the data of a problem, proceed as if it did not exist (data will be replaced without removing it!).  
 Number:   
 Short Name (usually a letter):

Figura No. 19 Cantidad de Ejercicios a desarrollar

Fuente Autores del Libro

Proceso de juzgamiento de los ejercicios enviados por los competidores, el proceso que realiza el BOCA llamado autojuzgamiento es el encargado de encapsular los ejercicios desarrollados y dar un veredicto, luego el juez entra a evaluar cada ejercicio e indica un resultado final. En el BOCA encontramos las siguientes respuestas como se evidencia en la figura No. 20, a los ejercicios desarrollados que son con las que se evalúa el proceso

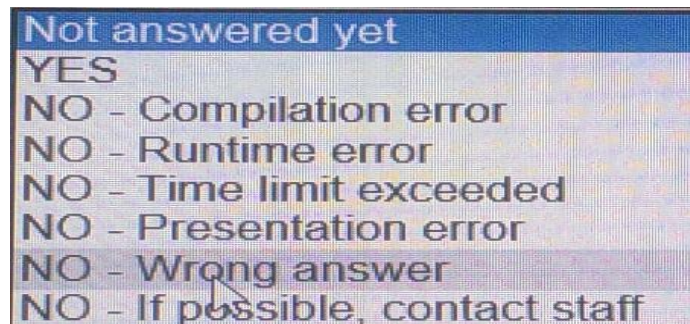


Figura No. 20 Respuestas de Autojuzgamiento

Fuente Autores del Libro

Muestra de ejercicios enviados en los cuales se evalúa dependiendo el proceso de cada ejercicio y teniendo en cuenta las entradas y las salidas (figura No. 21), también el veredicto que da el BOCA (figura No. 22)

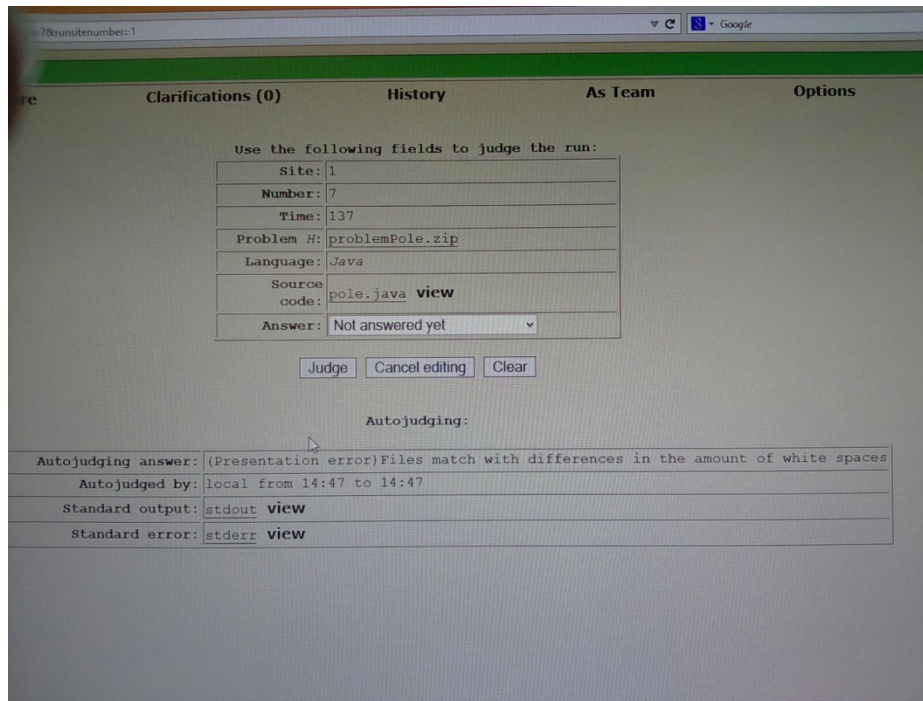


Figura No.21 Ejercicio a juzgar

Fuente Autores del Libro

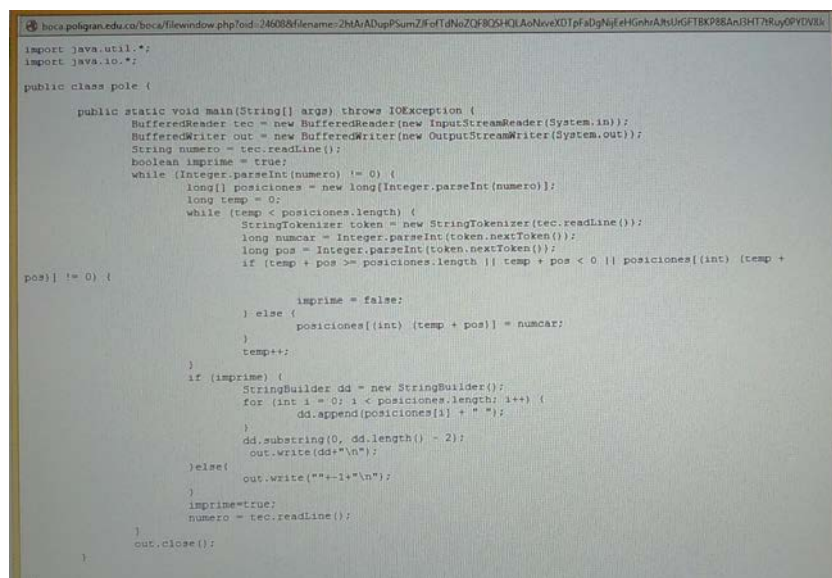


Figura No. 22 Compilación y Veredicto BOCA

Fuente Autores del Libro

# PROYECTOS A FUTURO

## **Sistema de maratones de programación conectada a través de una VPN [14]:**

Un proyecto muy interesante, consistiría en adaptar el sistema de maratones de programación y conectarlo a través de una VPN, para que de esta manera no solo las personas que están dentro de la red puedan participar, si no, que todas las personas desde cualquier parte del mundo a través de la internet puedan participar.

Este tipo de sistema es bastante útil para dar a conocer y promover las maratones de programación.

## **Sistema de maratones de programación como una herramienta en la enseñanza de la programación:**

Consistiría en utilizar el sistema de las maratones de programación como una herramienta de enseñanza, donde los estudiantes puedan practicar los conocimientos adquiridos en programación, evaluando los conocimientos, ya sea, a manera de quices, practica para parciales, notas adicionales, etc.

Este proyecto podría ayudar a que más estudiantes se interesen en participar en las maratones de programación y lleguen más preparados a las mismas.

# CONCLUSIONES

El principal objetivo de este trabajo fue montar un sistema de administración de maratones de programación sobre un sistema de Cloud Computing, con el fin de probar la funcionalidad, estabilidad y robustez del mismo.

Se logra evidenciar que el uso de las nuevas tecnologías (como Cloud Computing) nos ayuda a aumentar la eficacia y a reducir costos, al no tener que usar servidores físicos los cuales representan un gasto para cualquier institución.

Un punto fundamental que se logra con el presente trabajo fue la creación de un sistema sólido, que brinda la tranquilidad que las competencias realizadas no presenten bloqueos o errores, los cuales podrían afectar seriamente los resultados de los participantes.

Se comprobó que al dividir el sistema de BOCA en 3 (tres) servidores distintos el rendimiento y la estabilidad mejoraron sustancialmente, ya que al mantener los servicios separados en diferentes máquinas permitió que estos usen todo el potencial de la máquina para sí mismos.

# REFERENCIAS Y BIBLIOGRAFÍA

- [1] <http://www.unirioja.es/cu/fgarcia/sd/pub/teo/01-IntroduccionALaComputacionDistribuida.pdf>
- [2]<http://clusterfoodmasi.es/cluster/que-son-los-clusters/>
- [3]Evolución del Almacenamiento: Tesis “Model, Diseño y técnicas básicas para implementar un sistema de computación en la nube”, Autores: Viviana Yiseth Cardozo Escobar, María Trinidad Bernal Celis y Josemar David Sierra Ramírez
- [4] Hassan, Qusay (2011). "Desmitificando Cloud Computing " (PDF). El Diario de Defensa Ingeniería de Software (crosstalk) 2011 (enero / febrero): 16-21. Obtenido once 12 2014.
- [5] The NIST Definition of Cloud Computing, Peter Mell, Timothy Grance  
<http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf>
- [6] Infraestructura tecnológica, Asociación Catalana de Universidades Públicas (ACUP)  
[http://www.uoc.edu/portal/es/tecnologia\\_uoc/infraestructures/](http://www.uoc.edu/portal/es/tecnologia_uoc/infraestructures/)
- [7] OpenStack,  
<https://www.openstack.org/>
- [8] QUE ES EL CLOUD COMPUTING?, Kit Ingeniería Electrónica SRL  
<http://www.kit.com.ar/boletines-a.php?id=0000032>
- [9] Computación en la nube, Wikipedia®  
[https://es.wikipedia.org/wiki/Computaci%C3%B3n\\_en\\_la\\_nube](https://es.wikipedia.org/wiki/Computaci%C3%B3n_en_la_nube)
- [10] Maratón de programación, ACIS  
<http://acis.org.co/portal/content/acis-marat%C3%B3n-de-programaci%C3%B3n>
- [11] PostgreSQL, Ubuntu Documentation  
<https://help.ubuntu.com/community/PostgreSQL>
- [12]The NIST Definition of Cloud Computing,  
<http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf>
- [13] BOCA Online Contest Administrator,  
<http://www.ime.usp.br/~cassio/boca/>
- [14] Red privada virtual, Wikipedia®  
[https://es.wikipedia.org/wiki/Red\\_privada\\_virtual](https://es.wikipedia.org/wiki/Red_privada_virtual)

# GLOSARIO

**Almacenamiento:** Para cualquier sistema ordenado, las unidades de almacenamiento son aquellas que permiten guardar física o virtualmente archivos de datos de todo tipo.

**Arquitectura:** La Arquitectura del Software es el diseño de más alto nivel de la estructura de un sistema.

**Backends:** Este término que se refieren a la separación de intereses entre una capa de presentación y una capa de acceso a datos, respectivamente.

**Backups:** Es una copia de los datos originales que se realiza con el fin de disponer de un medio para recuperarlos en caso de su pérdida. Las copias de seguridad son útiles ante distintos eventos y usos: recuperar los sistemas informáticos y los datos de una catástrofe informática, natural o ataque; restaurar una pequeña cantidad de archivos que pueden haberse eliminado accidentalmente, corrompido, infectado por un virus informático u otras causas.

**Bases de Datos:** Bancos de información que contienen datos relativos a diversas temáticas y categorizados de distinta manera, pero que comparten entre sí algún tipo de vínculo o relación que busca ordenarlos y clasificarlos en conjunto.

**BOCA:** (BOCA Online Administrador del Concurso) es un soporte de competencias de programación del sistema diseñado para ser utilizado en la sociedad brasileña Maratón de Programación Informática. El sistema también se utiliza en el apoyo a las disciplinas que hacen uso de la sumisión y la corrección de trabajo durante la clase.

**Cloud:** Nube de cómputo o nube de conceptos (del inglés Cloud Computing), es un paradigma que permite ofrecer servicios de computación a través de una red, que usualmente es Internet.

**Cluster:** Se aplica a los conjuntos o conglomerados de computadoras construidos mediante la utilización de hardwares comunes y que se comportan como si fuesen una única computadora.

**Computación Distribuida:** La computación distribuida o informática en malla es un modelo para resolver problemas de computación masiva utilizando un gran número de ordenadores organizados en clústeres incrustados en una infraestructura de telecomunicaciones distribuida.

**Computación Grid:** La computación grid es una tecnología innovadora que permite utilizar de forma coordinada todo tipo de recursos entre ellos cómputo, almacenamiento y aplicaciones específicas.

**Core:** Tipo de microprocesador en donde coexisten múltiples procesadores.

**Hosting:** El alojamiento web es el servicio que provee a los usuarios de Internet un sistema para poder almacenar información, imágenes, vídeo, o cualquier contenido accesible vía web.

**Infraestructura:** Una infraestructura tecnológica cualquiera siempre será la base que definirá la vida de un sistema cualquiera. En el campo de la Informática o de los Sistemas de Información la decisión sobre una infraestructura tiene una importancia estratégica ya que limitará o potenciará el crecimiento y desarrollo de una organización.

**Maratones de Programación:** Es una competencia de programación y algoritmia llevada a cabo anualmente alrededor del mundo, bajo el auspicio de la Association for Computing Machinery (ACM).

**Openstack:** Es un proyecto de computación en la nube para proporcionar una infraestructura como servicio (IaaS). Es un software libre y de código abierto distribuido bajo los términos de la licencia Apache.

**PHP:** Es un lenguaje de programación de uso general de código del lado del servidor originalmente diseñado para el desarrollo web de contenido dinámico.

**PostgreSQL:** Es un Sistema de gestión de bases de datos relacional orientado a objetos y libre.

**Tecnología:** Es el conjunto de conocimientos técnicos, científicamente ordenados, que permiten diseñar, crear bienes, servicios que facilitan la adaptación al medio ambiente y satisfacer tanto las necesidades esenciales como los deseos de la humanidad.



**TopCoder:** Es una empresa que administra concursos en programación de computadoras. TopCoder acoge programación quincenal en línea competencias conocidos como materiales especificados de riesgo o "ronda de partidos individuales" -como así como competencias semanales en el diseño gráfico y el desarrollo.

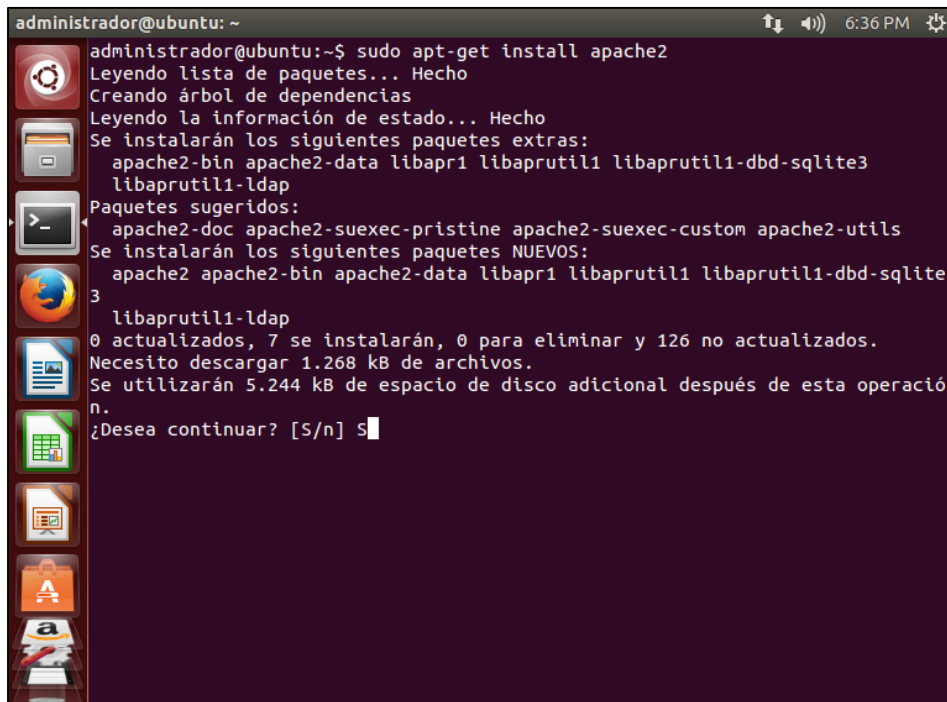
**Utility computing:** Se define como el suministro de recursos computacionales, como puede ser el procesamiento y almacenamiento.

**VPN:** (Virtual Private Network) Una Red Privada Virtual, es una tecnología de red que permite una extensión segura de la red local (LAN) sobre una red pública o no controlada como Internet.

**Wyse:** Es un fabricante líder de los sistemas de computación en nube.

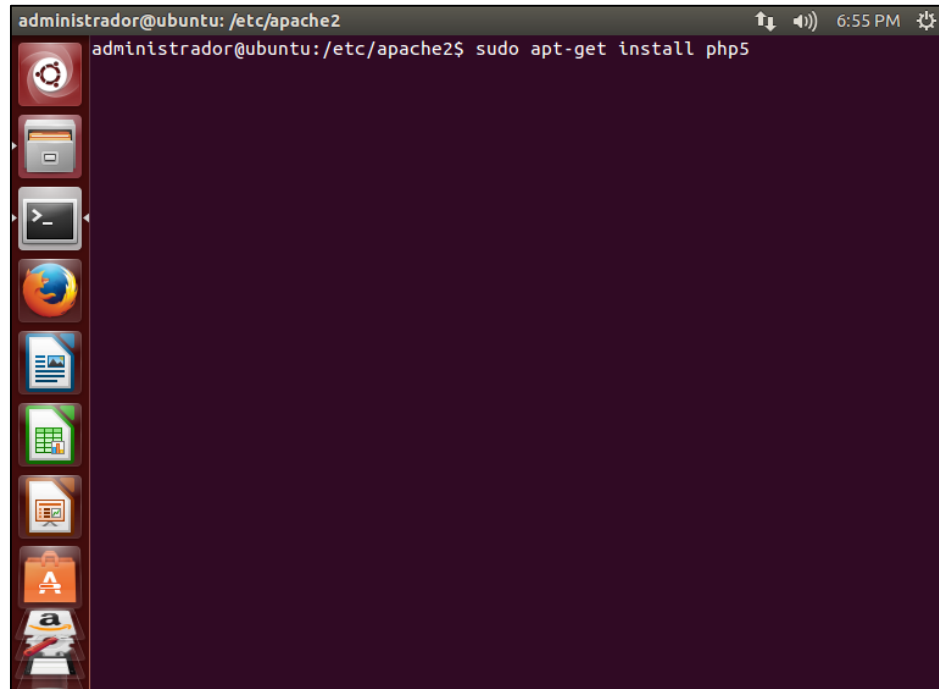
# ANEXOS

## 1. Instalación apache 2



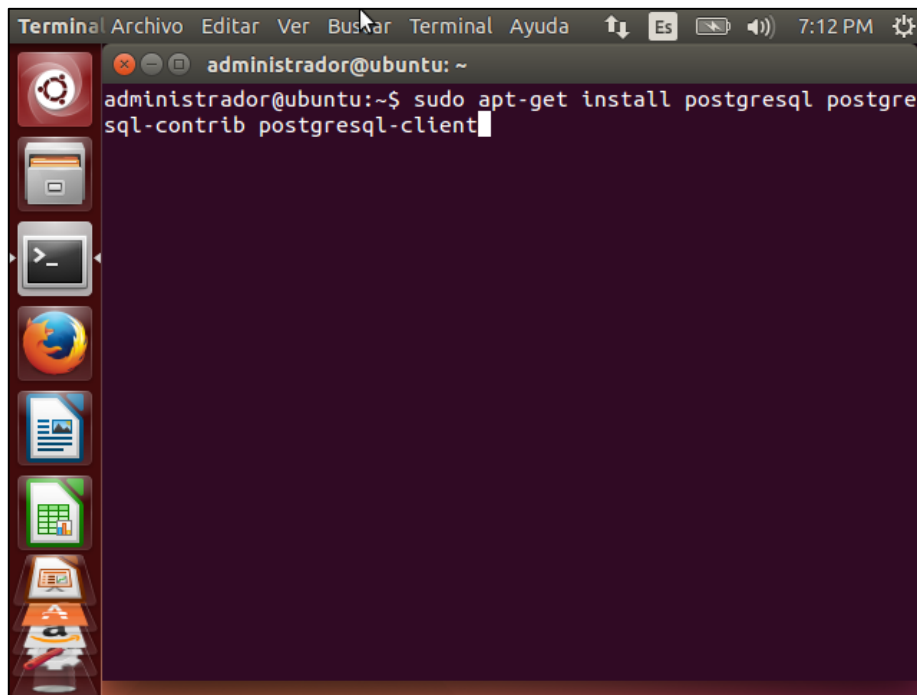
```
administrador@ubuntu: ~  
administrador@ubuntu:~$ sudo apt-get install apache2  
Leyendo lista de paquetes... Hecho  
Creando árbol de dependencias  
Leyendo la información de estado... Hecho  
Se instalarán los siguientes paquetes extras:  
  apache2-bin apache2-data libapr1 libaprutil1 libaprutil1-dbd-sqlite3  
  libaprutil1-ldap  
Paquetes sugeridos:  
  apache2-doc apache2-suexec-pristine apache2-suexec-custom apache2-utils  
Se instalarán los siguientes paquetes NUEVOS:  
  apache2 apache2-bin apache2-data libapr1 libaprutil1 libaprutil1-dbd-sqlite3  
  libaprutil1-ldap  
0 actualizados, 7 se instalarán, 0 para eliminar y 126 no actualizados.  
Necesito descargar 1.268 kB de archivos.  
Se utilizarán 5.244 kB de espacio de disco adicional después de esta operación.  
¿Desea continuar? [S/n] S
```

## 2. Instalación de PHP en el servidor web (apache)



```
administrador@ubuntu: /etc/apache2  
administrador@ubuntu:/etc/apache2$ sudo apt-get install php5
```

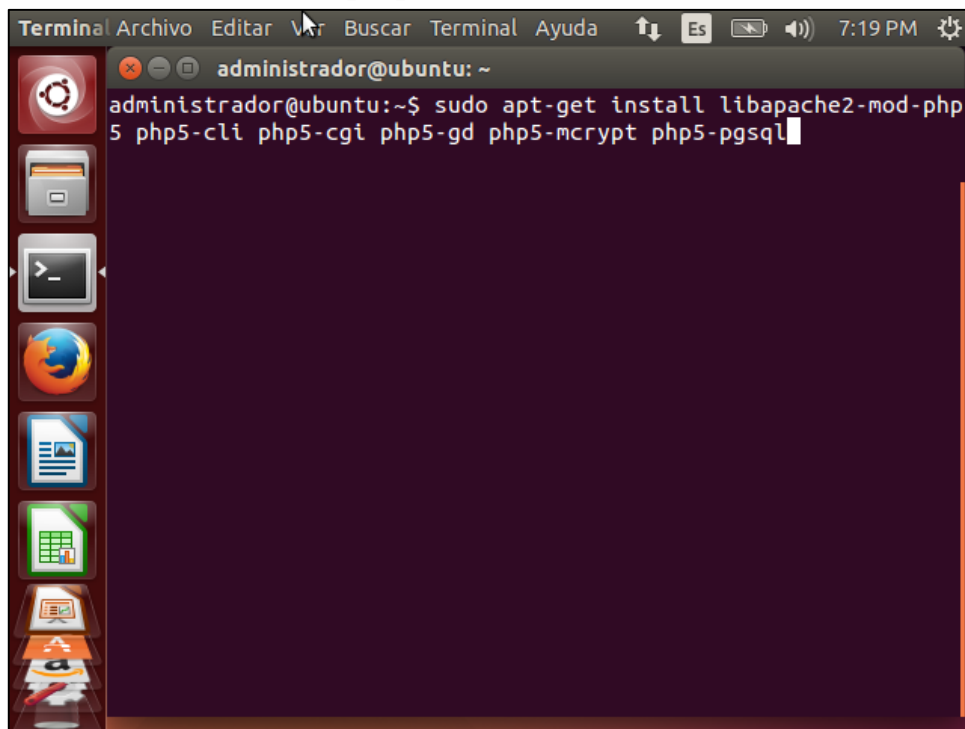
### 3. Instalación del servidor de base de datos Postgresql [11]



A terminal window titled 'Terminal' with a menu bar containing 'Archivo', 'Editar', 'Ver', 'Buscar', 'Terminal', and 'Ayuda'. The window shows the user 'administrador@ubuntu: ~' and the command `sudo apt-get install postgresql postgresql-contrib postgresql-client` entered at the prompt. The terminal background is dark purple with a vertical dock on the left side containing icons for the Dash, Home, Dash to Home, Firefox, LibreOffice Writer, LibreOffice Calc, and LibreOffice Impress. The system tray at the top right shows the time as 7:12 PM and various system icons.

```
administrador@ubuntu:~$ sudo apt-get install postgresql postgresql-contrib postgresql-client
```

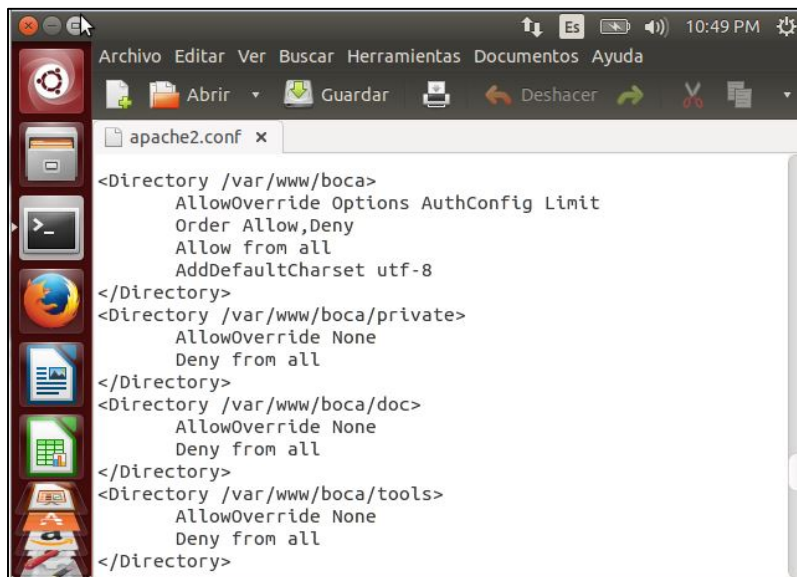
### 4. Instalación de paquetes adicionales



A terminal window titled 'Terminal' with a menu bar containing 'Archivo', 'Editar', 'Ver', 'Buscar', 'Terminal', and 'Ayuda'. The window shows the user 'administrador@ubuntu: ~' and the command `sudo apt-get install libapache2-mod-php5 php5-cli php5-cgi php5-gd php5-mcrypt php5-pgsql` entered at the prompt. The terminal background is dark purple with a vertical dock on the left side containing icons for the Dash, Home, Dash to Home, Firefox, LibreOffice Writer, LibreOffice Calc, and LibreOffice Impress. The system tray at the top right shows the time as 7:19 PM and various system icons.

```
administrador@ubuntu:~$ sudo apt-get install libapache2-mod-php5 php5-cli php5-cgi php5-gd php5-mcrypt php5-pgsql
```

## 5. Datos de configuración de BOCA en apache



```
Archivo Editar Ver Buscar Herramientas Documentos Ayuda
Abrir Guardar Deshacer
apache2.conf x
<Directory /var/www/boca>
    AllowOverride Options AuthConfig Limit
    Order Allow,Deny
    Allow from all
    AddDefaultCharset utf-8
</Directory>
<Directory /var/www/boca/private>
    AllowOverride None
    Deny from all
</Directory>
<Directory /var/www/boca/doc>
    AllowOverride None
    Deny from all
</Directory>
<Directory /var/www/boca/tools>
    AllowOverride None
    Deny from all
</Directory>
```

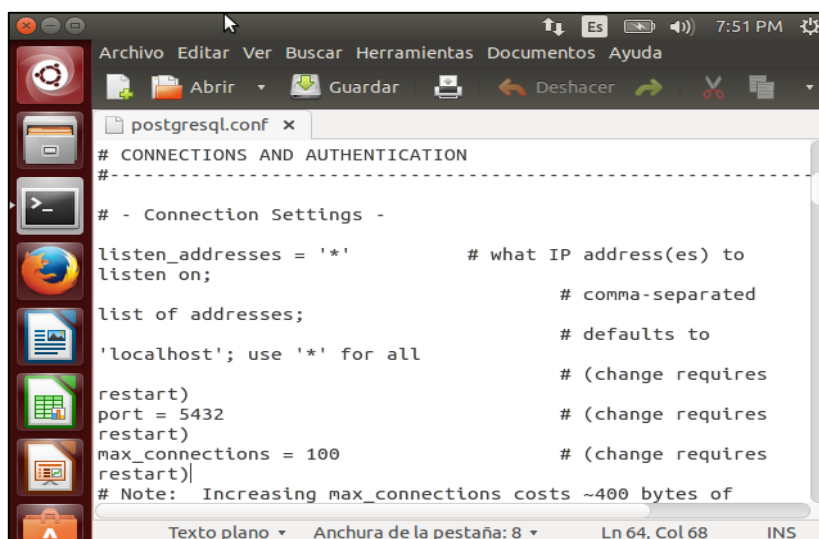
## 6. Datos de configuración PostgreSQL(/etc/postgresql/\*/main/postgresql.conf)

```
***** /etc/postgresql/*/main/postgresql.conf: *****
tcpip_socket = true #if using tcp to connect to database (older
postgresqls)
listen_addresses = '*' #newer postgresqls

# THE FOLLOWING MAY BE GOOD FOR PERFORMANCE
max_connections = 100
maintenance_work_mem = 32MB
shared_buffers = 512MB      ## USE AROUND 1/3 OF YOUR RAM
work_mem = 10MB
effective_cache_size = 512MB  ## USE AROUND 1/3 OF YOUR RAM

Note that for doing these changes you will need to increase the SHMMAX
parameter of your kernel, e.g. in file /etc/sysctl.d/10-shmmax.conf

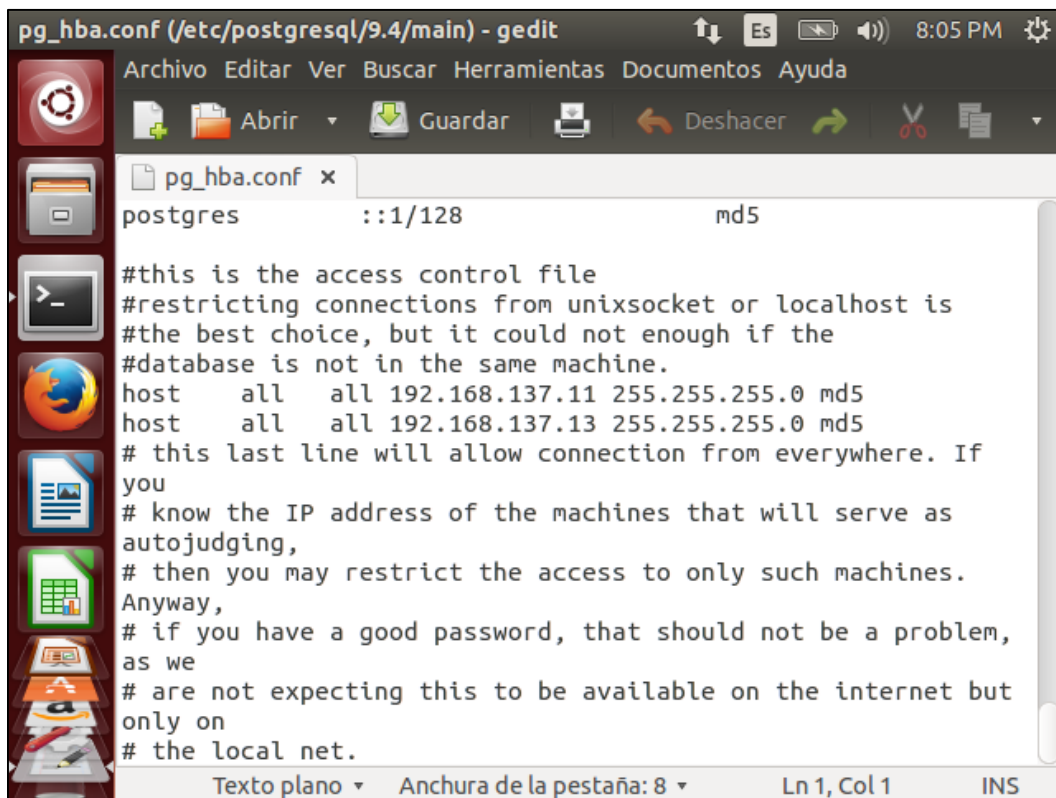
kernel.shmmax = 805416960
kernel.shmall = 196635
```



```
Archivo Editar Ver Buscar Herramientas Documentos Ayuda
7:51 PM
postgresql.conf x
# CONNECTIONS AND AUTHENTICATION
#-----
# - Connection Settings -
listen_addresses = '*'          # what IP address(es) to
listen on;
                                # comma-separated
list of addresses;              # defaults to
'localhost'; use '*' for all   # (change requires
restart)
port = 5432                      # (change requires
restart)
max_connections = 100           # (change requires
restart)
# Note: Increasing max_connections costs ~400 bytes of
Texto plano Anchura de la pestaña: 8 Ln 64, Col 68 INS
```

## 7. Datos de configuración Postgresql(/etc/postgresql/\*/main/pg\_hba.conf)

```
***** /etc/postgresql/*/main/pg_hba.conf: *****
#this is the access control file
#restricting connections from unixsocket or localhost is
#the best choice, but it could not enough if the
#database is not in the same machine.
host    all    all 127.0.0.1 255.255.255.255 md5
host    all    all 0.0.0.0   0.0.0.0     md5
# this last line will allow connection from everywhere. If you
# know the IP address of the machines that will serve as autojudging,
# then you may restrict the access to only such machines. Anyway,
# if you have a good password, that should not be a problem, as we
# are not expecting this to be available on the internet but only on
# the local net.
```



The screenshot shows a gedit editor window titled "pg\_hba.conf (/etc/postgresql/9.4/main) - gedit". The window has a menu bar with "Archivo", "Editar", "Ver", "Buscar", "Herramientas", "Documentos", and "Ayuda". Below the menu bar is a toolbar with icons for "Abrir", "Guardar", "Deshacer", and other functions. The main text area displays the content of the pg\_hba.conf file, which is identical to the text shown in the previous block. The status bar at the bottom indicates "Texto plano", "Anchura de la pestaña: 8", "Ln 1, Col 1", and "INS".

```
pg_hba.conf (/etc/postgresql/9.4/main) - gedit
Archivo Editar Ver Buscar Herramientas Documentos Ayuda
pg_hba.conf x
postgres      :::1/128      md5
#this is the access control file
#restricting connections from unixsocket or localhost is
#the best choice, but it could not enough if the
#database is not in the same machine.
host    all    all 192.168.137.11 255.255.255.0 md5
host    all    all 192.168.137.13 255.255.255.0 md5
# this last line will allow connection from everywhere. If
you
# know the IP address of the machines that will serve as
autojudging,
# then you may restrict the access to only such machines.
Anyway,
# if you have a good password, that should not be a problem,
as we
# are not expecting this to be available on the internet but
only on
# the local net.
Texto plano Anchura de la pestaña: 8 Ln 1, Col 1 INS
```

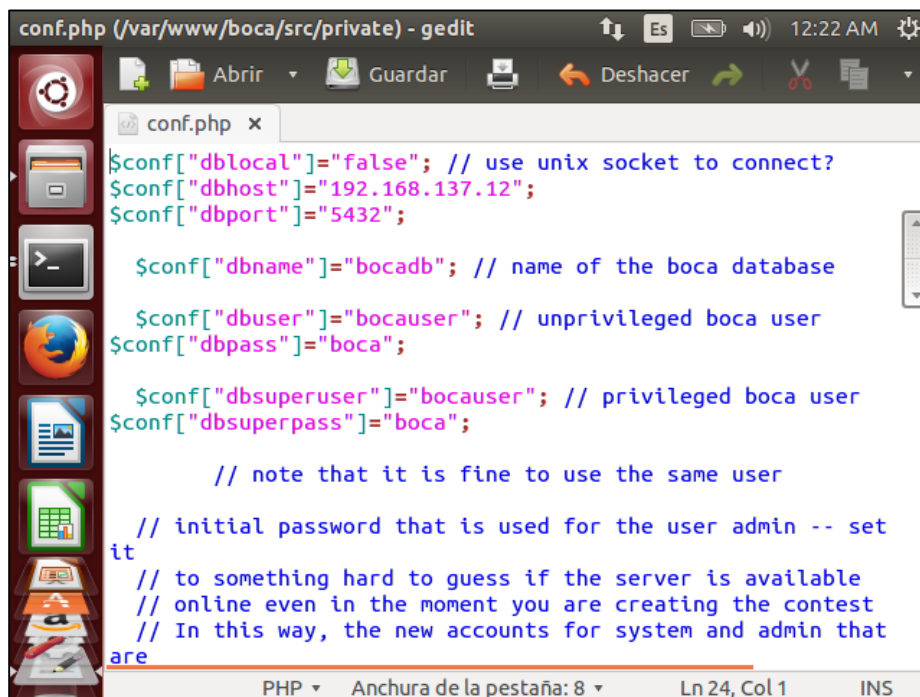
## 8. Datos de configuración y conexión de BOCA

```
$conf["dblocal"]="false"; // use unix socket to connect?
$conf["dbhost"]="localhost"; // ip address of the db
$conf["dbname"]="bocadb"; // name of the boca database
$conf["dbuser"]="bocausers"; // unprivileged boca user
$conf["dbpass"]="boca"; // unprivileged boca password
$conf["dbsuperuser"]="bocausers"; // privileged boca user
$conf["dbsuperpass"]="boca"; // privileged boca password
    // note that it is just fine to use the same user for
    // unpriv and priv access

// secret key to be used in HTTP headers
// you MUST set it with any random large enough sequence
// DONT LEAVE IT AS YOU GOT WHEN UNPACKED THE SOFTWARE
// CHOOSE ANOTHER NUMBER/STRING AND REPLACE BELOW
$conf["key"]="secretKey:23894091237589234759234723489";

// initial password that is used for the user admin -- set it
// to something hard to guess if the server is available
// online even in the moment you are creating the contest
// In this way, the new accounts for system and admin that are
// eventually created come already with the password set to this
// value. It is your task later to update these passwords to
// some other values within the BOCA web interface.
$conf["basepass"]="boca";

// the following field is used by the autojudging script
// set it with IP address (or other short description)
// of the computer allocated for
// autojudging during the competition
// this is only useful for debug purposes when multiple
// autojudges are being used
$conf["ip"]='10.10.10.10';
```



The screenshot shows a text editor window titled "conf.php (/var/www/boca/src/private) - gedit". The editor displays the configuration file content, which is a PHP array of configuration options. The visible code includes database connection settings, user credentials, a secret key, and initial password information. The editor interface includes a menu bar with "Abrir", "Guardar", "Deshacer", and "Cortar" options, and a status bar at the bottom showing "PHP", "Anchura de la pestaña: 8", "Ln 24, Col 1", and "INS".

```
conf.php (/var/www/boca/src/private) - gedit
$conf["dblocal"]="false"; // use unix socket to connect?
$conf["dbhost"]="192.168.137.12";
$conf["dbport"]="5432";

$conf["dbname"]="bocadb"; // name of the boca database

$conf["dbuser"]="bocausers"; // unprivileged boca user
$conf["dbpass"]="boca";

$conf["dbsuperuser"]="bocausers"; // privileged boca user
$conf["dbsuperpass"]="boca";

    // note that it is fine to use the same user

// initial password that is used for the user admin -- set it
// to something hard to guess if the server is available
// online even in the moment you are creating the contest
// In this way, the new accounts for system and admin that are
// eventually created come already with the password set to this
// value. It is your task later to update these passwords to
// some other values within the BOCA web interface.
$conf["basepass"]="boca";

// the following field is used by the autojudging script
// set it with IP address (or other short description)
// of the computer allocated for
// autojudging during the competition
// this is only useful for debug purposes when multiple
// autojudges are being used
$conf["ip"]='10.10.10.10';
```

```
conf.php (/var/www/boca/src/private) - gedit
// to something hard to guess if the server is available
// online even in the moment you are creating the contest
// In this way, the new accounts for system and admin that
are
// eventually created come already with the password set to
this
// value. It is your task later to update these passwords to
// some other values within the BOCA web interface.
$config["basepass"]="boca";

// secret key to be used in HTTP headers
// you MUST set it with any random large enough sequence
$config["key"]="secretKey:23894091237589234759234723489";

// the following field is used by the autojudging script
// set it with the ip of the computer running the script
// The real purpose of it is only to differentiate between
// autojudges when multiple computers are used as autojudges
$config["ip"]='192.168.137.13';
```

BOCA Online Contest Administrator boca-1.5.9 - Login

localhost

### BOCA Login

Name  Login

Password