

**PROTOTIPO DE UN SERVICIO DE COMPUTACIÓN EN LA NUBE PAAS  
(PLATAFORMA COMO SERVICIO) Y WEB SERVICE ORIENTADO AL  
DESARROLLO DE PROGRAMACIÓN WEB.**

**PAOLA ANDREA GARRIDO CRUZ**

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

**2017**

**PROTOTIPO DE UN SERVICIO DE COMPUTACIÓN EN LA NUBE PAAS  
(PLATAFORMA COMO SERVICIO) Y WEB SERVICE ORIENTADO AL  
DESARROLLO DE PROGRAMACIÓN WEB.**

**PAOLA ANDREA GARRIDO CRUZ**

**Trabajo de grado para optar al título de  
INGENIERO DE SISTEMAS**

**Director**

**MSC. MANUEL GUILLERMO FLÓREZ BECERRA**

**UNIVERSIDAD INDUSTRIAL DE SANTANDER  
FACULTAD DE INGENIERÍAS FÍSICO-MECÁNICAS  
ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA  
BUCARAMANGA**

**2017**

*A mis padres, Nelson Garrido Ardila y Floreliz Cruz Reyes, quienes son mi apoyo incondicional, que hicieron posible que hoy llegara donde estoy, siempre han sido mi guía y mi inspiración.*

*Al caballero sin armadura, que siempre lucho por que lograra mi objetivo y me apoyo en cada instante.*

*Paola Andrea Garrido Cruz*

## AGRADECIMIENTOS

*Al profesor Manuel Guillermo Flórez por confiar en mí para el desarrollo de este proyecto y por guiarme en el proceso.*

*A los compañeros del GID-CONUSS por su colaboración y apoyo.*

## CONTENIDO

INTRODUCCION .....	15
1. INFORMACIÓN DEL PROYECTO .....	16
1.1 DESCRIPCIÓN DEL PROBLEMA .....	16
1.2 JUSTIFICACIÓN .....	16
1.3 VIABILIDAD .....	16
2. OBJETIVOS.....	18
2.1 OBJETIVO GENERAL .....	18
2.2 OBJETIVOS ESPECÍFICOS.....	18
3. MARCO TEÓRICO .....	20
3.1 COMPUTACIÓN EN LA NUBE .....	20
3.1.1 Definición .....	20
3.1.2 Beneficios de la computación en la nube.....	21
3.1.3 modelos de Implementación .....	23
3.1.4 Modelos de servicio .....	25
3.1.5 Principios de la nube en la economía .....	27
3.1.6 Plataformas como servicios PaaS comerciales .....	29
3.1.6.1 Amazon web Services.....	29
3.1.6.2 Windows Azure .....	33
3.2. WEB SERVICE .....	36
3.2.1. Definición .....	36
3.2.2. Guía de programador como usar un Web service .....	36
4. HERRAMIENTAS .....	39
4.1 FRAMEWORK .....	39

5. DESARROLLO DEL PROTOTIPO .....	45
5.1. DEFINICIÓN DE LAS HERRAMIENTAS .....	45
6. FACTURACIÓN .....	50
6.1 TARIFAS DE SERVICIOS .....	50
6.1.1 LAMP .....	50
6.1.2 Linux postgres.....	50
6.1.3. Jommla .....	51
6.1.4. tomcat .....	51
6.1.5 Programación en la web .....	52
6.2. LIQUIDACIÓN DE SERVICIOS .....	52
6.2.1 Diseño.....	53
6.2.1.1 Modelo de Entidad Relación .....	53
6.2.2. Proceso de Obtención de los consumos.....	53
7. SLA.....	55
7.1. SLA ESTUDIANTES Y PROFESORES DE LA UNIVERSIDAD INDUSTRIAL DE SANTANDER.....	55
7.2. SLA USUARIOS EXTERNOS.....	59
8. ESCENARIOS DE DESARROLLO.....	64
8.1. HOSTING WEB PARA CURSO DE PROGRAMACIÓN WEB .....	64
8.1.1. Proceso de generación de Usuario enjaulado. ....	64
8.1.1.2. Ubicación de archivos dentro del el sistema .....	65
8.1.1.3 Creación de los usuarios de programación web dentro del sistema .....	65
8.1.1.4. Creación de los base de datos para los estudiantes de programación web. .....	65
8.1.2 Administración de los usuarios de programación web .....	67
8.1.2.1. Abrir usuario.....	67
8.1.2.2. Abrir diferentes usuarios .....	68

8.1.2.3. Cerrar usuarios. ....	68
8.1.2.4. Cerrar grupos.....	68
8.1.2.5 Backups de proyectos de usuarios. ....	69
8.1.2.6 Backups de base de datos de usuarios. ....	69
8.2 HOSTING WEB INDIVIDUAL .....	69
8.2.1 Planteamiento del escenario.....	69
8.2.2 Registro y asignación de la instancia.....	70
8.2.3 Uso de la instancia.....	70
9. CONCLUSIONES .....	72
10. RECOMENDACIONES.....	73
REFERENCIAS BIBLIOGRAFICAS.....	74
BIBLIOGRAFIA.....	75
ANEXOS.....	77

## TABLA DE FIGURAS

	<b>Pág.</b>
Figura 1 Definiciones de la computación en la nube NIST .....	21
Figura 2 Tipos de usuarios Paypal.....	36
Figura 3 Comisiones paypal para empresarios.....	37
Figura 4 Botón Paypal .....	38
Figura 5 modelo vista controlador.....	40
Figura 6 ficheros de proyectos de laravel fuente :autor .....	46
Figura 7 Modelo de Entidad Relación fuente: autor .....	53
Figura 8 Facturas generadas por la plataforma fuente: autor .....	54
Figura 9 Conexión a putty fuente:autor.....	64
Figura 10 phpmyadmin después de ejecución de scriptsql.sh fuente: autor .....	66
Figura 11 Conexión ftp del usuario fuente: autor .....	67
Figura 12 Factura para usuarios .....	71

## LISTA DE TABLAS

Tabla 1 máquinas ofrecidas en linux.....	31
Tabla 2 transferencia de datos entrantes a Amazon EC2.....	32
Tabla 3 transferencia de datos salientes de Amazon EC2 .....	32
Tabla 4 tarifas de Azure .....	35
Tabla 11 Costos LAMP .....	50
Tabla 12 Costos Linux Postgres .....	51
Tabla 13 Costos Jommla .....	51
Tabla 14 Costos Tomcat.....	51
Tabla 15 Costos de Programación web .....	52
Tabla 5 Formato de generación de usuarios .....	65
Tabla 6 Generación de bases de datos .....	66
Tabla 7 Formato de ejecución del script para abrir usuario .....	68
Tabla 8 Abrir grupo de programación web.....	68
Tabla 9 Formato de ejecución de script cerrar usuarios .....	68
Tabla 10 Formato de ejecución del script de cerrar grupo de usuarios .....	69

## **LISTA DE ANEXOS**

ANEXO A: ARTICULO ESCRITO SOBRE EL PROTOTIPO DE UN SERVICIO DE COMPUTACION EN LA NUBE PAAS (PLATAFORMA COMO SERVICIO) Y WEB SERVICE ORIENTADO AL DESARROLLO DE PROGRAMACION WEB .....	79
---	----

## RESUMEN

**TÍTULO:** PROTOTIPO DE UN SERVICIO DE COMPUTACIÓN EN LA NUBE PAAS (PLATAFORMA COMO SERVICIO) Y WEB SERVICE ORIENTADO AL DESARROLLO DE PROGRAMACIÓN WEB.

**AUTOR:** GARRIDO CRUZ, PAOLA ANDREA

**PALABRAS CLAVE:** Plataforma como servicio, Web service, Computación en la nube.

### **DESCRIPCIÓN:**

La constante investigación y evolución de las plataformas tecnológicas de información en materia de uso de recursos, y disminución de costos para las empresas que solicitan servicios de computación en la nube.

La computación en la nube ofrece tres modelos de servicio que son la infraestructura como servicio(IaaS), plataformas como servicios(PaaS) y software como servicio(SaaS) ofrecen al usuario beneficios en cada uno de ellos el autoservicio bajo demanda, acceso amplio a la red, fondo de recursos, elasticidad, costos más bajos, facilidad de utilización, calidad del servicio, viabilidad y fiabilidad.

Las plataformas tienen diferentes tipos de servicio donde el usuario administra las aplicaciones y los datos y el proveedor del servicio administra las redes, almacenamiento, servidores, virtualización, sistema operativo, middleware y runtime.

Este trabajo de investigación busca evaluar los métodos de procesamiento, almacenamiento, y la definición de costos de los servicios ofrecidos por las plataformas como servicio comercial, en base a ese estudio definir las componentes del prototipo de plataforma como servicio Webuis, donde planteamos escenario de un hosting individual y el hosting de programación web.

Para el escenario del hosting se crearon unos usuarios enjaulados con restricciones de transferencia de datos, donde se el administrador de la instancia tiene procesos de control, permitiendo generación y monitoreo de cada uno de ellos, tanto de las bases de datos como del software desarrollado por cada uno de los usuarios, limitándole la transferencia de archivos de cuando el administrador considere necesario, otro proceso que se definió la generación de backup.

Para el hosting individual se definió el proceso de facturación completo limitando los cortes mensuales de acuerdo a las políticas definidas en SLA service level agreement.

\* Trabajo de Grado

\*\* Facultad de Ingenierías Físicomecánicas. Escuela de Ingeniería de Sistemas e Informática. Director: Msc. Manuel Guillermo Flórez Becerra

## ABSTRACT

**TITLE:** PROTOTYPE OF A COMPUTER SERVICE IN THE PAAS CLOUD (PLATFORM AS A SERVICE) AND WEB SERVICE ORIENTED TO THE WEB PROGRAMMING DEVELOPMENT.

**AUTHOR:** GARRIDO CRUZ, PAOLA ANDREA

**KEY WORDS:** Platform as a service, Web service, Cloud Computing.

### DESCRIPTION:

The constant research and development of technological platforms for information on the use of resources, and cost reduction for companies requesting services in the cloud computing.

Cloud computing offers three service models that are infrastructure as a service (IaaS), platforms as services (PaaS) and software as a service (SaaS) that offer to the user benefits in each of them, self-service on demand, broad access to the network, resource pool, elasticity, lower costs, ease of use, quality of service, feasibility and reliability.

The platforms have different types of service where the user administers the applications and the data and the service provider manages the networks, storage, servers, virtualization, operating system, middleware and runtime.

This research aims to evaluate the methods of processing, storage, and cost definition of the services offered by the platforms as a commercial service, based on that study that defines the components of the platform prototype as Webuis service, where we plated the scenario of a Individual hosting and web programming hosting.

For the hosting scenario, caged users with data transfer restrictions were created, where the administrator of the instance has control processes, allowing generation and monitoring of each of them, both databases and software developed by each of the users, limiting the transfer of files when the administrator considers necessary, another process that was defined backup generation.

For individual hosting, the complete billing process was defined, limiting the monthly cuts according to the policies defined in SLA service level agreement.

\* Degree Work

\*\* Faculty of Physicomechanical Engineering. School of Systems Engineering and Informatics. Director: Msc. Manuel Guillermo Flórez Becerra

## INTRODUCCION

La constante migración de las tecnologías de información a la computación en la nube definida por el NIST(Instituto Americano de Estándares y Tecnología) como un modelo que permite acceso bajo demanda a través de la red a un conjunto compartido de recursos de computación configurables que pueden ser rápidamente provisionados con el mínimo esfuerzo de gestión o iteración del proveedor del servicio, basándonos en base a estos conocimientos previos nace la idea de generar un prototipo de plataforma como servicio partiendo de la infraestructura actual que posee el grupo GID-CONUSS.

Basándonos en los variables que tiene sus orígenes en los conceptos de virtualización, el diseño de aplicaciones, la computación paralela; facilitando el procesamiento y la administración de la información en diferentes entornos, reduciendo la inversión en infraestructura de Tecnologías de la Información y Comunicación (TIC)

## **1. INFORMACIÓN DEL PROYECTO**

### **1.1 DESCRIPCIÓN DEL PROBLEMA**

La computación en la nube está compuesta por tres modelos de servicio infraestructura como servicio, plataforma como servicio y software como servicio, actualmente el grupo de investigación GID-CONUSS de la Universidad Industrial de Santander cuenta con el servicio de computación en la nube ofreciendo el servicio de infraestructura como servicio, se pretende realizar una primera fase exploratoria de las plataformas como servicio para ampliar los servicios ofrecidos a la comunidad académica.

### **1.2 JUSTIFICACIÓN**

En el desarrollo de este proyecto se construirá un prototipo del modelo de una plataforma como servicio para los estudiantes de programación web de la escuela de ingeniería de sistemas de la Universidad Industrial de Santander para el desarrollo de los proyectos correspondientes a la respectiva materia, logrando aprovechar las ventajas de la computación en la nube, respaldados por los por seguridad, flexibilidad y disponibilidad.

### **1.3 VIABILIDAD**

La constante migración de las aplicaciones a la computación en la nube generando la necesidad de visualizar modelos de negocios ofreciendo servicios específicos, de donde se originó la idea de generar entornos inicialmente para los estudiantes de programación web, escuela de Ingeniería de Sistemas de la

Universidad Industrial de Santander, con estándares específicos para el desarrollo dentro del entorno académico.

Por otra parte el grupo de GID-CONUSS posee una infraestructura como servicios (IaaS) sobre la que se soportan servicios para la comunidad académica, se pretende visualizar el comportamiento de este en un entorno comercial con servicios específicos de las empresas que se dedican a nivel comercial, al desarrollo de plataformas como servicio, con el objetivo de a largo plazo generar ingresos para el grupo de investigación.

## 2. OBJETIVOS

### 2.1 OBJETIVO GENERAL

Implementar un prototipo de servicio de computación en la nube PAAS (plataforma como servicio) y Web Service para dar soporte al desarrollo de la programación web.

### 2.2 OBJETIVOS ESPECÍFICOS

- Objetivos administrativos
- Administrar la actual infraestructura de alta disponibilidad y computación en la nube del grupo GID-CONUSS. Entre las labores a cumplir se encuentran:
  - ❖ Entrenamiento completo del funcionamiento de dicha infraestructura
  - ❖ Monitoreo del sistema
  - ❖ Monitoreo de los servidores, clúster de alta disponibilidad y servicios de computación en la nube
  - ❖ Realizar tareas de mantenimiento programadas y copias de seguridad
  - ❖ Recuperación del sistema o servicios en caso de fallo
  - ❖ Realizar tareas relacionadas con atención a los usuarios de los diferentes servicios

- Entrenar y asesorar los relevos administrativos para garantizar la continuidad de los procesos.
- Análisis del estado del arte de las PAAS (plataforma como servicio) orientado al desarrollo web, con el objeto de establecer cada uno de las componentes que integran la plataforma.
- Definir un modelo de service level agreement (SLA) para la plataforma de servicio (PaaS).
- Implementar una administración de usuarios y facturación de servicios de acuerdo a las políticas definidas en el uso de la plataforma como servicio (PaaS).
- Crear una máquina virtual con los elementos y herramientas que soporten la plataforma como servicios, realizándole un seguimiento continuo del desempeño y funcionalidad
- Escribir un artículo con los resultados del proyecto y las recomendaciones para próximas investigaciones y desarrollos

## 3. MARCO TEÓRICO

### 3.1 COMPUTACIÓN EN LA NUBE

Desde la aparición de la computación en la nube ha tenido múltiples definiciones para el término de computación plateando desde enfoques históricos, en los que se firmas de analistas, académicos y compañías de tecnologías de la información sin llegar a un acuerdo común. La computación en la nube toma la tecnología, los servicios y las aplicaciones que son similares a los de Internet y los convierte en una utilidad de autoservicio.

**3.1.1 Definición** “La computación en nube es un modelo para permitir un acceso por la red a un conjunto de recursos computacionales configurables (ej. Redes, servidores, almacenamiento, aplicaciones y servicios) de manera conveniente o bajo demanda, que se puede aprovisionar y liberar rápidamente con un mínimo de esfuerzo administrativo o de interacción del proveedor de servicios.” [3]

El uso de la palabra nube hace referencia a dos conceptos esenciales:

- **Abstracción:** Extrae los detalles de implementación del sistema de usuarios y programadores. Las aplicaciones se ejecutan en sistemas físicos que no se especifican, los datos se almacenan en una ubicación desconocida, la administración de sistemas esta subcontratada con otros. [1]
  
- **Virtualización:** La computación en la nube virtualiza los sistemas reuniendo y compartiendo recursos. Los sistemas y el almacenamiento pueden abastecerse según sus necesidades desde una infraestructura centralizada los costos se evalúan en base a unos contadores, se permite la multitenencia.

La computación en la nube en base a la definición del NITS separa la computación en la nube entre modelos de servicio y modelos de implementación se muestran en la figura 1

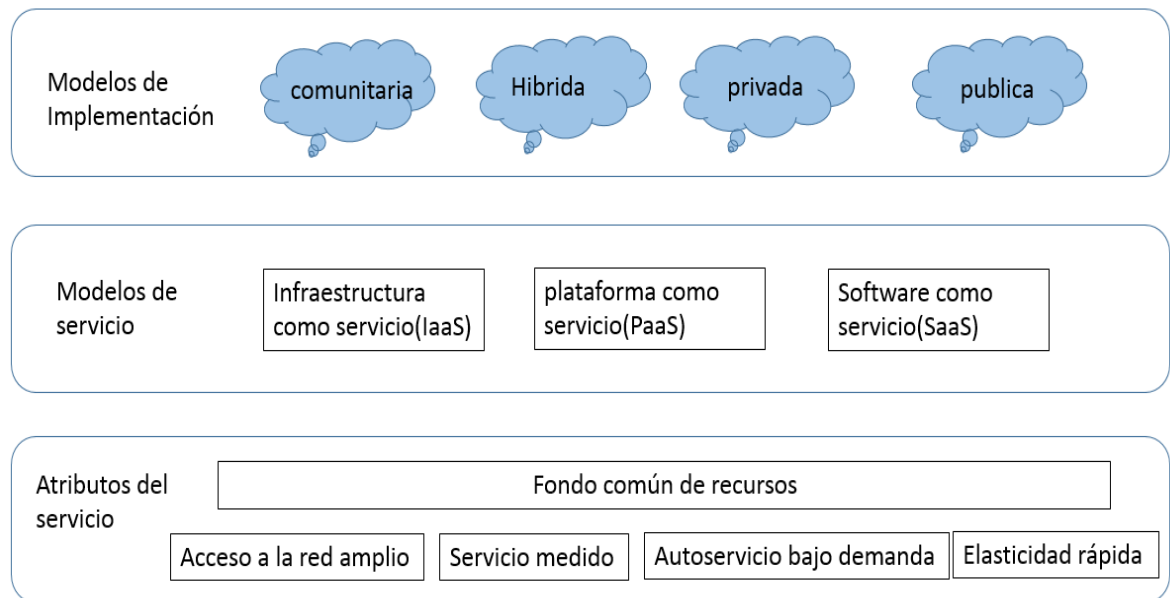


Figura 1. Definiciones de la computación en la nube NIST

Fuente: [1] Que es la nube el futuro de los sistemas de información.

### 3.1.2 Beneficios de la computación en la nube

- Autoservicio bajo demanda: Un cliente puede aprovisionar recursos unilateralmente sin interactuar con el personal proveedor de servicios. El tiempo del servidor y el almacenamiento en red, según se necesite automáticamente sin Interacción con cada proveedor de servicios.[1][3]

- Acceso amplio desde la red: Las capacidades están disponibles a través de la red y se accede a través de métodos estándar que promueven el uso por plataformas de clientes, incluyendo una mezcla de sistemas operativos heterogéneos y plataformas pesadas y ligeras como portátiles, teléfonos móviles y PDA. [1][3]
  
- Fondo de recursos: Los recursos informáticos del proveedor se agrupan para servir a múltiples consumidores Utilizando un modelo multi-tenant, con diferentes recursos físicos y virtuales dinámicamente. Asignado y reasignado de forma dinámica de acuerdo a la demanda del consumidor. En relación con este concepto esta la idea de abstracción que oculta la ubicación de recursos como máquinas virtuales, procesamiento, memoria, almacenamiento, ancho de banda y conectividad de la red [1][3]
  
- Elasticidad Rápida: Las capacidades se pueden aprovisionar y liberar elásticamente, en algunos casos automáticamente, escalar rápidamente hacia fuera y hacia adentro en consonancia con la demanda. Desde el punto de vista del cliente, los recursos de la computación en la nube deberían parecer ilimitados y poder comprarse en cualquier momento y en cualquier cantidad. [1][3]
  
- Servicio medido: Los sistemas de nube se mide, se audita y se recoge en un informe para el cliente. A un cliente se le puede cobrar el uso de recursos puede ser controlados y reportados, proporcionando transparencia tanto para el proveedor como para Consumidor del servicio utilizado. [1][3]
  
- Costos más bajos: Puesto que las redes de nubes operan con una eficacia más alta y con una utilización mayor, a menudo nos encontramos con reducción significativas de costes.

- **Facilidad de Utilización:** Dependiendo del tipo de servicio que se ofrezca quizá nos encontremos con que no se necesitan licencias de software o hardware para implementar nuestros servicios
- **Calidad del servicio:** La calidad del servicio (QoS, Quality of Service es algo que podemos obtener por contrato con nuestro proveedor
- **Fiabilidad:** La escala de la redes de computación en la nube y su capacidad para proporcionar equilibrio de carga y conmutación tras error hace que sean fiables, a menudo mucho más fiables de lo que podemos conseguir en una sola organización
- **Administración de IT subcontratada:** Una implementación de computación en la nube permite que otras personas administren nuestra infraestructura informática mientras nosotros administramos nuestros negocios. En la mayoría de los casos, conseguiremos reducciones considerables en los costes de personal de IT
- **Mantenimiento y actualización simplificados:** Puesto que el sistema esta centralizado, podemos aplicar parches y actualización de manera sencilla. Eso significa que los usuarios siempre tiene acceso a versiones más recientes de software.
- **Barrera de entrada baja:** En concreto, el desembolso de capital adelantado reduce de manera drástica. En la computación en la nube, cualquiera puede convertirse en un gigante en cualquier momento.[1]

**3.1.3 modelos de Implementación.** La definición del NIST para los cuatro modelos de implementación es la siguiente:

- Nube privada. La infraestructura de la nube está provista para su uso exclusivo por una sola organización. La nube puede administrarse a través de una organización, un tercero o alguna combinación entre ellos y puede existir dentro o fuera de las instalaciones. [1]
  
- Nube de la comunitaria. La infraestructura de la nube está provista para uso exclusivo por una comunidad de consumidores de organizaciones que tienen preocupaciones compartidas (por ejemplo, misión, requisitos de seguridad, política y consideraciones de cumplimiento). Puede ser propiedad, administrado y operado por una o más de las organizaciones de la comunidad, un tercero partido, o alguna combinación de ellos, y puede existir dentro o fuera de las instalaciones [1]
  
- Nube pública. La infraestructura de la nube está provista para uso abierto por el público en general. Puede ser propiedad, administrado y operado por una organización comercial, académica u gubernamental, o alguna combinación de ellos. Existe en las instalaciones del proveedor de la nube. [1]
  
- Nube híbrida. La infraestructura de la nube es una composición de dos o más nubes distintas Infraestructuras (privadas, comunitarias o públicas) donde se mantienen sus identidades pero se vinculan como una unidad. Una nube híbrida puede ofrecer acceso estandarizado o de propietarios a los datos y las aplicaciones, así como la portabilidad de aplicaciones. [1]

**3.1.4 Modelos de servicio.** El conjunto de servicios ofrecidos en la nube ofrece 3 modelos de adoptando la nomenclatura XaaS, ó “<Algo> como servicio”, llegando a definir la computación en la nube y la forma en que los usuarios finales pueden acceder a sus recursos y servicios. Los servicios se ha aceptado a nivel universal son:

- **Infraestructura como Servicio (IaaS):** Proporciona máquinas virtuales, almacenamiento virtual, infraestructura virtual y otros activos de hardware como recursos que los clientes pueden optimizar, El proveedor de servicios IaaS administra toda la infraestructura, mientras que el cliente es responsable de todos los demás aspectos de la implementación, que pueden incluir el sistema operativo, aplicaciones e interacciones del usuario con el sistema[1]
  
- **Plataforma como Servicio (PaaS):** Proporciona máquinas virtuales, sistemas operativos, aplicaciones, servicios, marcos de trabajo, transacciones y estructura de control. El cliente puede desplegar sus aplicaciones en la infraestructura de la nube o utilizar aplicaciones que se hayan programado mediante lenguajes y herramientas compatibles con el proveedor de servicios PaaS. El proveedor de servicios administra la infraestructura de la nube, los sistemas operativos y el software de habilitación. El cliente es responsable de la instalación y administración de la aplicación a desplegar. [1]
  
- **Software como Servicio (SaaS):** Es un entorno operativo completo con aplicaciones e interfaz de usuario. En el modelo SaaS, la aplicación proporciona al cliente a través de una interfaz de cliente ligero (por lo general un navegador), y la responsabilidad del cliente empieza y acaba con la introducción y administración de sus datos e interacción del usuario. Todo lo que va desde la aplicación hasta la infraestructura es responsabilidad del proveedor[1].

Los tres modelos de servicio en un conjunto que se conocen como el modelo SPI de computación en la nube donde se engloban todas las posibilidades, resulta útil pensar en los modelos de servicio de computación en la parte inferior de la pila están el hardware o infraestructura que comprende la red. A medida que avanzamos hacia arriba en la pila, cada modelo de servicio hereda las capacidades del que hay situado debajo. IaaS tiene los menores niveles de funcionalidad integrada y los niveles de integración más bajos y SaaS los máximos.

Para la realización de este proyecto nos centraremos en las plataformas como servicio por eso analizaremos de manera más detallada cada una de las componentes de las plataformas como servicios el cual describe un entorno compuesto por múltiples herramientas y utilidades para ayudar al diseño y la implementación de aplicaciones en una máquina virtual en la que se ejecuta nuestro software las cuales van desde las desarrolladas por compañías como Windows Azure a sistemas como Drupal, Wolf. Muchos CMS (Sistemas de Administración de Contenido) son plataformas como servicio donde obtenemos partes estándar y podemos crear sitios web, entre los servicios que se incluyen encontramos los siguientes [1] [2]:

- Desarrollo de Aplicaciones: Una plataforma proporciona los medios para utilizar programas que creamos un lenguaje compatible, o un entorno de desarrollo visual que escribe ese código para nosotros.
- Colaboración: Muchos sistemas de PaaS están configurados para permitir que varios individuos trabajen en los mismos proyectos.
- Administración de datos: Se ofrecen herramientas para acceder y utilizar datos en un almacén de datos.

- Instrumentación, rendimiento y pruebas: Existen herramientas disponibles para medir nuestras aplicaciones y optimizar su rendimiento.
- Almacenamiento: Podemos almacenar los datos en el servicio del distribuidor de PaaS o acceder a ellos desde un servicio de almacenamiento de terceros.
- Administración de transacciones: Muchos sistemas de PaaS proporcionan servicios como administración de transacciones o servicio de correduría para mantener integridad de las plataformas.

**3.1.5 Principios de la nube en la economía.** Las diez leyes de la nube que apuntan según Joe Wienman de AT&T Global services podrían reunirse así:

- Los servicios de utilidad cuestan menos aunque cuesten más: las utilidades cobran una prima por sus servicios, pero los clientes ahorran dinero al no pagar por los servicios que utiliza. [1]
- Pronostico de bazas bajo petición: La posibilidad de proveer y desproveer recursos supone ingresos y menos gastos [1]
- El máximo de la suma nunca supera la suma de los máximos: Una nube puede implementar menos capacidad porque los máximos de los arrendatarios individuales en un sistema compartido se ponderan en el tiempo por el grupo [1]
- La demanda agregada es más suave que la individual: La multitenencia también tiende a ponderar la variedad intrínseca a la demanda individual ya que el “coeficiente de variables aleatorias” es siempre menor o igual que el de cualquier de la variables individuales. [1]

- La media de los costes de unidad se reduce distribuyendo costos fijos entre más unidades de salida: Los proveedores de nube manejan un volumen que permite adquirir recursos por precios bastante bajos. [1]
- La superioridad numérica es el factor más importante para el resultado de un combate: Weiman sostiene que un mayor tamaño de nube tiene el poder de repeler ataques de redes de robots y DDoS mejor que los sistemas pequeños. [1]
- El espacio-tiempo es un continuum: la posibilidad de realizar una tarea en la nube utilizando procesamiento en paralelo permite a las empresas que trabajan en tiempo real responder más rápido a las condiciones de negocio y acelera la toma de decisiones. [1]
- La dispersión en es cuadrado inverso de la latencia: La latencia, o el retardo al obtener una respuesta a una petición, requiere implementaciones a gran escala y multisitio, que son características de los proveedores de nube. [1]
- No hay que jugárselo todo en una carta: La fiabilidad de un sistema con  $n$  componentes redundantes y una fiabilidad  $f$  es  $1-(1-f)^n$ . Así pues, cuando un centro de datos consigue una fiabilidad del 99 por 100 dos centos de datos redundantes tiene una fiabilidad del 99,99 por 100. [1]
- Un objeto en reposo permanece en reposo: Los centros de datos privados tienden a estar ubicados en los lugares donde se fundó o adquirió la empresa. Los proveedores de red pueden colocar sus centros de datos en sitios verdes, en estos puntos generan una red troncal la cual consta rutas de alta capacidad y routers gestionados por proveedores de servicios individuales como entidades gubernamentales o comerciales.

### **3.1.6 Plataformas como servicios PaaS comerciales**

3.1.6.1 Amazon web Services [1] Se basa en estándares SOA, incluidos HTTP, REST y los protocolos de transferencia SOAP, sistemas operativos de código comerciales, de código abierto, servidores de aplicaciones y acceso basado en el navegador dentro de los cuales se incluyen:

Amazon Elastic Compute Cloud (Amazon EC2) es un servicio web que proporciona capacidad de cómputo con tamaño modificable en la nube. Está diseñado para facilitar a los desarrolladores la informática en la nube escalable basado en web.

La sencilla interfaz de servicios web de Amazon EC2 permite obtener y configurar la capacidad con una fricción mínima. Proporciona un control completo sobre los recursos informáticos y puede ejecutarse en el entorno informático acreditado de Amazon. Amazon EC2 reduce el tiempo necesario para obtener y arrancar nuevas instancias de servidor en cuestión de minutos, lo que permite escalar rápidamente la capacidad, ya sea aumentándola o reduciéndola, según cambien sus necesidades. Amazon EC2 cambia el modelo económico de la informática, ya que solo tendrá que pagar por la capacidad que realmente utilice. Amazon EC2 proporciona a los desarrolladores las herramientas necesarias para crear aplicaciones resistentes a errores y para aislarse de los casos de error más comunes.[5].

Dentro de las herramientas que soportan los servicios de EC2 se encuentran:

- Amazon Simple Queue Service: es una cola de mensajes o transacciones para aplicaciones distribuidas basadas en internet.
- Amazon Simple Notification Service: Servicio web que puede publicar mensajes de una aplicación y entregarlo a otras aplicaciones o suscriptores.

- Amazon CloudWatch: Monitoriza EC2 con una consola o vista de línea de comandos de utilización de recursos, índices de rendimiento e indicadores operativos por factores como la demanda del procesador, la utilización del disco y la E/S.
- Elastic Load Balancing: Permite equilibrar instancias de máquinas de amazon en EC2 (Elastic Compute Cloud), detecta una instancia que está fallando y redirige el tráfico a otra que si funcione.
- Amazon Simple Storage System: Sistema de almacenamiento y copia de seguridad en línea puede transferir datos a alta velocidad.
- Amazon Elastic Block Store: Sistema para crear discos virtuales o dispositivos de almacenamiento a nivel de bloque.
- Amazon SimpleDB: Es el almacén de datos que soporta indización y consultas.
- Amazon Relational Database Service: permite crear instancias de base de datos de MySQL para soportar sitios web LAMP.
- Amazon Cloudfront: Sistema de almacenamiento de contenido que guarda datos en la cache en distintas ubicaciones físicas para mejorar el acceso al usuario. [5] [1]

Amazon ofrece una instancias generales las cuales proporcionan una especificaciones que son de uso para aplicaciones con consumos bajos de CPU las cuales se acoplan a los requerimientos para necesarios para la realización del presente proyecto, su estándar tarifario es tan variable como los servicios

ofrecidos por la multinacional. En la tabla 1 observamos los tipos de instancias ofrecidos por amazon bajo la plataforma de Linux:

Tabla 1. Máquinas ofrecidas en linux

Uso general – Generación actual	vCPU	ECU	Memoria (GiB)	Almacenamiento de la instancia (GB)	uso de Linux/unix por hora
t2.nano	1	Variable	0.5	Solo EBS	\$0.0101 por hora
t2.micro	1	Variable	1	Solo EBS	\$0.02 por hora
t2.small	1	Variable	2	Solo EBS	\$0.041 por hora
t2.medium	2	Variable	4	Solo EBS	\$0.081 por hora
t2.large	2	Variable	8	Solo EBS	\$0.162 por hora
t2.xlarge	4	Variable	16	Solo EBS	\$0.324 por hora
t2.2xlarge	8	Variable	32	Solo EBS	\$0.648 por hora
m4.large	2	6.5	8	Solo EBS	\$0.171 por hora
m4.xlarge	4	13	16	Solo EBS	\$0.343 por hora
m4.2xlarge	8	26	32	Solo EBS	\$0.685 por hora
m4.4xlarge	16	53.5	64	Solo EBS	\$1.37 por hora
m4.10xlarge	40	124.5	160	Solo EBS	\$3.425 por hora
m4.16xlarge	64	188	256	Solo EBS	\$5.48 por hora
m3.medium	1	3	3.75	1 x 4 SSD	\$0.095 por hora
m3.large	2	6.5	7.5	1 x 32 SSD	\$0.19 por hora
m3.xlarge	4	13	15	2 x 40 SSD	\$0.381 por hora
m3.2xlarge	8	26	30	2 x 80 SSD	\$0.761 por hora
Con optimización informática – Generación actual					
c4.large	2	8	3.75	Solo EBS	\$0.155 por hora
c4.xlarge	4	16	7.5	Solo EBS	\$0.309 por hora
c4.2xlarge	8	31	15	Solo EBS	\$0.618 por hora
c4.4xlarge	16	62	30	Solo EBS	\$1.235 por hora
c4.8xlarge	36	132	60	Solo EBS	\$2.47 por hora
c3.large	2	7	3.75	2 x 16 SSD	\$0.163 por hora
c3.xlarge	4	14	7.5	2 x 40 SSD	\$0.325 por hora
c3.2xlarge	8	28	15	2 x 80 SSD	\$0.65 por hora
c3.4xlarge	16	55	30	2 x 160 SSD	\$1.3 por hora
c3.8xlarge	32	108	60	2 x 320 SSD	\$2.6 por hora
Optimizadas para memoria – Generación actual					
r3.large	2	6.5	15	1 x 32 SSD	\$0.35 por hora
r3.xlarge	4	13	30.5	1 x 80 SSD	\$0.7 por hora
r3.2xlarge	8	26	61	1 x 160 SSD	\$1.399 por hora
r3.4xlarge	16	52	122	1 x 320 SSD	\$2.799 por hora
r3.8xlarge	32	104	244	2 x 320 SSD	\$5.597 por hora

Fuente: <http://aws.amazon.com/es/ec2/pricing/on-demand/>

Amazon también realiza cobros por transferencia de entrada y los de salida en la tabla 3 observamos las tarifas definidas por la compañía:

Tabla 2. Transferencia de datos entrantes a Amazon EC2

Transferencia ENTRANTE de datos a Amazon EC2	precios
Internet	\$0.00 por GB
Otra región de AWS (de cualquier servicio de AWS)	\$0.00 por GB
Amazon S3, Amazon Glacier, Amazon DynamoDB, Amazon SES, Amazon SQS o Amazon SimpleDB en la misma región de AWS	\$0.00 por GB
Instancias de Amazon EC2, Amazon RDS, Amazon Redshift y Amazon ElastiCache o interfaces de red elásticas en la misma zona de disponibilidad	
Uso de una dirección IPv4 privada	\$0.00 por GB
Uso de una dirección IPv4 pública o elástica	\$0.01 por GB
Uso de una dirección IPv6 dentro del mismo VPC	\$0.00 por GB
Uso de una dirección IPv6 de un VPC diferente	\$0.01 por GB
Instancias de Amazon EC2, Amazon RDS, Amazon Redshift y Amazon ElastiCache o interfaces de red elásticas en otra zona de disponibilidad o VPC compartida de la misma región de AWS	\$0.01 por GB

Fuente: <https://aws.amazon.com/es/ec2/pricing/on-demand/>

Tabla 3. Transferencia de datos salientes de Amazon EC2

Transferencia SALIENTE de datos de Amazon EC2 a	precios
Amazon S3, Amazon Glacier, Amazon DynamoDB, Amazon SES, Amazon SQS o Amazon SimpleDB en la misma región de AWS	\$0.00 por GB
Instancias de Amazon EC2, Amazon RDS, Amazon Redshift o Amazon ElastiCache, Amazon Elastic Load Balancing o interfaces de red elásticas en la misma zona de disponibilidad	
Uso de una dirección IPv4 privada	\$0.00 por GB
Uso de una dirección IPv4 pública o elástica	\$0.01 por GB
Uso de una dirección IPv6 dentro del mismo VPC	\$0.00 por GB
Uso de una dirección IPv6 de un VPC diferente	\$0.01 por GB
Instancias de Amazon EC2, Amazon RDS, Amazon Redshift o Amazon ElastiCache, Amazon Elastic Load Balancing o interfaces de red elásticas en otra zona de disponibilidad o VPC compartida de la misma región de AWS	\$0.01 por GB
Otra región de AWS	\$0.16 por GB
Amazon CloudFront	\$0.00 por GB
Transferencia SALIENTE de datos de Amazon EC2 a Internet	
Primer GB/mes	\$0.00 por GB
Hasta 10 TB/mes	\$0.25 por GB
Siguientes 40 TB/mes	\$0.23 por GB
Siguientes 100 TB/mes	\$0.21 por GB
Siguientes 350 TB/mes	\$0.19 por GB

Fuente: <https://aws.amazon.com/es/ec2/pricing/on-demand/>

[5]Las tarifas fijadas por amazon son de acuerdo a los sistemas operativos utilizado para nuestro estudio solo tomamos en cuenta Linux aunque la compañía

ofrece los mismos tipos de máquinas para con RHEL, SLES, Windows, Windows con SQL Standart, Windows con SQL Web, Fedora, Debian, Ubuntu, OpenSolaries, OpenSuse, Gentoo Linux los cuales poseen las misma características solo con la variación de sus sistema operativo y por ende una variación en la tarifas de acuerdo al sistema operativo ofrecido.

Actualmente están distribuidas por zonas o regiones de servicio de EC2 los cuales se listan a continuación:

- EE.UU. Este (Norte de Virginia )
- EE.UU. Este (Ohio )
- EE.UU. Oeste (Oregón )
- EE.UU. Oeste (Norte de California)
- Canadá (Central)
- UE (Irlanda)
- UE (Londres)
- UE (Fráncfort)
- Asia Pacifico (Singapur)
- Asia Pacifico (Tokio)
- Asia Pacifico (Sidney)
- Asia Pacifico (Singapur)
- Asia Pacifico (Seúl)
- Asia Pacifico (Mumbai)
- América del Sur (Sao Paulo)
- AWS GovCloud (EE.UU.)

**3.1.6.2 Windows Azure.** Azure es una infraestructura virtualizada en cuya capa superior se ha añadido una capa de servicios dentro de los que se incluyen los siguientes [6]:

- AppFabric que crea un entorno de alojamiento de aplicaciones es una versión de .NET Framework.

- Un servicio de almacenamiento no relacional de alta capacidad llamado almacenamiento.
- Un conjunto de instancias particulares de máquinas virtuales llamado proceso.
- Una versión habilitada para nube de SQL Server llamada SQL Azure Database.
- Un punto de venta de bases de datos basado en SQL Azure Database con el nombre en clave "Dallas".
- Un servicio de xRM(Gestion de relaciones de cualquier cosa) llamado Dynamics CRM, que se basa en Microsoft Dynamics.
- Un servicio de colaboración y documentos basado en SharePoint llamado SharePoint Services.
- Windows Live Services, una colección de servicios que se ejecuta en Windows Live y puede utilizarse en aplicaciones que se ejecutan en la nube.[6][1]  
 La API de azure utiliza protocolos estándar REST(Transferencia de estado Representacional), HTTP y XML que forman parte de cualquier infraestructura SOAP para permitir comunicación, permite que los programadores puedan instalar clases del lado del cliente las cuales se han añadido a Mirosoft Visual Studio como parte del entorno de desarrollo integrado, los elementos principales que la componen son:
  - Aplicación: El tiempo de ejecución de la aplicación que se está ejecutando en la nube.

- Proceso: Es el motor de directivas y computación del servidor Windows con equilibrio de carga el cual permite crear y administrar máquinas virtuales con dos roles el web role y el role del trabajador
- Almacenamiento: Es un sistema relacional para almacenamiento a gran escala, el cual se puede manipular utilizando REST API que se basa en una solicitud HTTP estándar, los almacenamientos pueden leerse por métodos GET y POST los cuales se escriben como PUT y DELETE.

Tabla 4. Tarifas de Azure

INSTANCIA	NÚCLEOS	RAM	TAMAÑOS DE DISCO	PRECIO
A0	1	0,75 GB	20 GB	\$0,022/h
A1	1	1,75 GB	40 GB	\$0,054/h
A2	2	3,50 GB	60 GB	\$0,112/h
A3	4	7,00 GB	120 GB	\$0,232/h
A4	8	14,00 GB	240 GB	\$0,464/h
A1 v2	1	2,00 GB	10 GB	\$0,061/h
A2 v2	2	4,00 GB	20 GB	\$0,129/h
A4 v2	4	8,00 GB	40 GB	\$0,27/h
A8 v2	8	16,00 GB	80 GB	\$0,567/h
A2m v2	2	16,00 GB	20 GB	\$0,208/h
A4m v2	4	32,00 GB	40 GB	\$0,437/h
A8m v2	8	64,00 GB	80 GB	\$0,917/h
A1 v2	1	2,00 GB	10 GB	\$0,061/h
A2 v2	2	4,00 GB	20 GB	\$0,129/h
A4 v2	4	8,00 GB	40 GB	\$0,27/h
A8 v2	8	16,00 GB	80 GB	\$0,567/h
A2m v2	2	16,00 GB	20 GB	\$0,208/h
A4m v2	4	32,00 GB	40 GB	\$0,437/h
A8m v2	8	64,00 GB	80 GB	\$0,917/h
D1	1	3,50 GB	50 GB	\$0,095/h
D2	2	7,00 GB	100 GB	\$0,19/h
D3	4	14,00 GB	200 GB	\$0,38/h
D4	8	28,00 GB	400 GB	\$0,76/h

Fuente:[6][https://azure.microsoft.com/en-us/support/legal/sla/virtual-machines/v1\\_4/#](https://azure.microsoft.com/en-us/support/legal/sla/virtual-machines/v1_4/#)

## 3.2. WEB SERVICE

**3.2.1. Definición.** [13] Es un servicio ofrecido por un dispositivo electrónico a otro dispositivo electrónico, comunicándose entre sí a través de la World Wide Web. En un servicio Web, la tecnología Web, tal como HTTP, diseñada originalmente para la comunicación de persona a máquina, se utiliza para la comunicación de máquina a máquina, más específicamente para transferir formatos de archivo legibles por máquina como XML y JSON. En la práctica, el servicio web proporciona normalmente una interfaz basada en web orientada a objetos a un servidor de base de datos, utilizado por ejemplo por otro servidor web, o por una aplicación móvil, que proporciona una interfaz de usuario al usuario final. Otra aplicación común ofrecida al usuario final puede ser un mashup, en el que un servidor web consume varios servicios web en diferentes máquinas y compila el contenido en una sola interfaz de usuario.

**3.2.2. Guía de programador como usar un Web service.** La plataforma de pagos segura que este servicio ofrece tres tipos de cuentas la cuenta personal, la cuenta premier y la cuenta de Business estas cuentas ofrecen los servicios que se observan en la figura 3 de acuerdo al tipo de usuario [13]

Ventajas de la cuenta	Personal	Premier	Empresa comercial
Enviar dinero	✓	✓	✓
Vigilancia 24 horas contra fraude	✓	✓	✓
Disponibilidad del Servicio de Atención al Cliente	✓	✓	✓
Herramientas de eBay	Limitada	✓	✓
Servicios para vendedores	Limitada	✓	✓
Administrar usuarios			✓

Figura 2. Tipos de usuarios Paypal

fuelle: <https://www.paypal.com/es/webapps/mpp/paypal-account-types>:

3.2.2.1. Tarifas de paypal. Para el caso del desarrollador tomaremos la opción de empresas dentro de sus características está el certificado PCI el cual procesa los datos de tu negocio que te negocio, se definen las unas tarifas de acuerdo a los consumos de servicio los cuales se especifican en la figura 4. [13]

Si las ventas de tu negocio superan los 3,000 USD, y creaste tu cuenta hace más de 90 días, podemos ofrecerte una tasa más baja. Una vez que hayas creado tu negocio, solo tienes que [Quiero vender](#) y esperar a que la aprobemos. [Solicitar ahora.](#)

Tus ventas mensuales	Comisión por transacción	Ejemplos
>100,000.00 USD	4.4 % + 0.30 USD	Comisión de 4.70 USD en una venta de 100 USD
10,000.01 USD – 100,000.00 USD	4.7 % + 0.30 USD	Comisión de 5.00 USD en una venta de 100 USD
3,000.01 USD – 10,000.00 USD	4.9 % + 0.30 USD	Comisión de 5.20 USD en una venta de 100 USD
0.00 USD – 3,000.00 USD	5.4 % + 0.30 USD	Comisión de 5.70 USD en una venta de 100 USD

Conversión Monetaria: Se cobrará un costo adicional de hasta un 3.5 % por cualquier conversión de moneda. La tarifa con descuento por volumen para los organizaciones sin fines de lucro que cualifiquen puede ser tan baja como de un 4.4 %.

[Ver todos las comisiones y descuentos](#)

Figura 3. *Comisiones paypal para empresarios*

fuelle: <https://www.paypal.com/co/webapps/mpp/paypal-fees>

**3.2.2.1. Registro de paypal** Para el proceso de registro, con los datos personales en cada uno de los menús hasta finalizar el proceso de creación de un nuevo usuario, y asociamos una tarjeta de débito o crédito [13]

Para generar los pagos de paypal debemos adjuntar el botón de pagos online para eso debemos seguir las siguientes instrucciones.

1. Inicie sesión en su cuenta PayPal.
2. Haga clic en la pestaña Servicio para comerciantes
3. Haga clic en el vínculo Botones comprar ahora.
4. Especifique el nombre, precio y otros datos del artículo que desea vender.

5. Añada más información alternativa como los campos de envío, impuesto de ventas y opción (tamaño, color, etc.).
6. Haga clic en **Crear botón ahora** y la Fábrica de botones generará un código HTML personalizado.
7. Pegue el código HTML en su sitio Web para crear su botón **Comprar ahora**. Cuando un comprador haga clic en el botón **Comprar ahora** se le conducirá a una página de pago seguro de PayPal en la que puede iniciar sesión en una cuenta PayPal existente o registrarse para abrir una nueva y completar así rápidamente la compra.

Un ejemplo de código de creación de botones es el siguiente script

```
<form name="_xclick" action="https://www.paypal.com/co/cgi-bin/webscr"
method="post">
<input type="hidden" name="cmd" value="_xclick">
<input type="hidden" name="empresas" value="me@mybusiness.com">
<input type="hidden" name="currency_code" value="USD">
<input type="hidden" name="item_name" value="Teddy Bear">
<input type="hidden" name="amount" value="12.99">
<input type="image" src="http://www.paypal.com/es_XC/i/btn/x-click-but01.gif"
border="0" name="submit" alt="Haga pagos con PayPal: es rápido, sin costo y
seguro">
</form>
```

El cual generara el siguiente botón



Figura 4. *Botón Paypal*

fuelle:[https://www.paypalobjects.com/es\\_XC/i/btn/x-click-but01.gif](https://www.paypalobjects.com/es_XC/i/btn/x-click-but01.gif)

## 4. HERRAMIENTAS

### 4.1 FRAMEWORK

Los frameworks tienen como objetivo principal ofrecer una funcionalidad definida, auto-contenida, siendo construidos usando patrones de diseño, y su característica principal es su alta cohesión y bajo acoplamiento. Para acceder a esa funcionalidad, se construyen piezas, objetos, llamados objetos calientes, que vinculan las necesidades del sistema con la funcionalidad que este presta. Esta funcionalidad, está constituida por objetos llamados fríos, que sufren poco o ningún cambio en la vida del framework, permitiendo la portabilidad entre distintos sistemas, donde lo esencial para ser denominados frameworks es estar constituidos por objetos casi estáticos con funcionalidad definida a nivel grupo de objetos y no como parte constitutiva de estos, por ejemplo en sus métodos, en cuyo caso se habla de un API o librería. Algunas características notables que se pueden observar [13] :

- La inversión de control: en un framework, a diferencia de las bibliotecas, el flujo de control no es dictado por el programa que llama, sino por el mismo.<sup>2</sup>
  - La funcionalidad o comportamiento predeterminado: un marco tiene un comportamiento predeterminado. Este comportamiento por defecto debe ser un comportamiento útil, definido e identificable.
  - [4] Su extensibilidad: un marco puede ser ampliado para proporcionar una funcionalidad específica. El frame, en general, no se supone que deba ser modificado, excepto en cuanto a extensibilidad. Los usuarios pueden ampliar sus características, pero no deben ni necesitan modificar su código.  
La mayoría de los frameworks poseen un mvc (modelo vista controlador) es un patrón de arquitectura que separa los datos de y la lógica de negocio de una

aplicación de la interfaz de usuario y el modulo encargado de gestionar eventos y las comunicaciones.

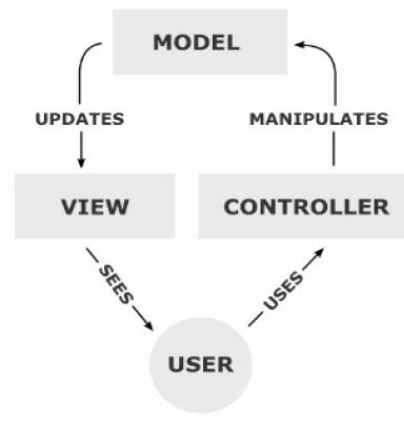


Figura 5. *Modelo vista controlador*

*Fuente: manual de laravel 5.1*

En la figura tres visualizamos los componentes de MVC y definimos sus componentes:

- El Modelo: Es la representación de la información con la cual el sistema opera, por lo tanto gestiona todos los accesos a dicha información, tanto consultas como actualizaciones.
- El Controlador: Responde a eventos (usualmente acciones del usuario) e invoca peticiones al 'modelo' cuando se hace alguna solicitud de Por tanto se podría decir que el 'controlador' hace de intermediario entre la 'vista' y el 'modelo'.

Para el presente proyecto tomaremos en cuenta los frameworks de desarrollo para programar en php dentro las posibles opciones que contemplaremos se encuentran los siguientes:

- CakePHP: es un framework para el desarrollo rápido OpenSource, entre sus características se encuentran [14]:

- Compatible con PHP4 y PHP5
- CRUD integrado para la interacción con la base de datos
- Soporte de la aplicación
- Generación de código
- Arquitectura modelo vista controlador (MVC)
- Despachador de peticiones, con URLs y rutas personalizadas y limpias
- Validación integrada
- Plantillas rápidas y flexibles
- Ayudantes para AJAX, Javascript, formularios HTML y mas
- Componentes de Email, Cookie, Seguridad, Sesión y manejo de solicitudes
- Listas de control de acceso flexibles
- Limpieza de datos
- Cache flexible
- Localización
- Funciona en cualquier subdirectorio de sitio web, con poca o ninguna configuración de Apache.

- Yiiframework: es un framework genérico de programación web es liviano de correr y está equipado con soluciones de cache sofisticadas es adecuado para desarrollar aplicaciones de gran trafico como portales, foros, sistema de administración de contenidos (CMS), sistemas de comercio electrónico, etc. [14]
  
- Laravel: Fue creado en 2011 por Taylor Otwell, inspirándose en Ruby on Rails y SymfonySe basa en arquitectura mvc y resuelve necesidades actuales como manejo de eventos y autenticación de usuarios, sus características principales son [14]:
  - Modular y extensible: permite agregar todo lo que necesitas a través de su directorio Packalys que cuenta con más de 5.500 paquetes donde puedes encontrar lo que necesitas implementado.
  
  - Micro-servicios y APIs: Lumen es un micro-framework derivado de Laravel para desarrollar de forma fácil y rápida micro-servicios y APIs. Lumen integra todas las características de Laravel con mínima configuración
  
  - HTTP rounting: Posee un sistema de enrutamiento rápido y eficiente, similar al que se usa en Ruby on Rails. Este nos permite relacionar las partes de nuestra aplicación con las rutas que ingresa al usuario en el navegador.
  
  - HTTP Middleware: El uso de Middleware se encarga de analizar y filtrar las llamadas
  
  - HTTP en tu servidor: Puedes instalarlo para que se encargue de verificar que se trate de un usuario registrado, de evitar problemas de tipo Cross-Site-Scripting(XSS) y otras medidas de seguridad.

- Cache: Robusto sistema de cache configurable
- Autenticación: autenticación de usuarios de forma nativa
- Tareas automatizadas: Con el APIElixirde Laravel se pueden crear tareas de Gulp con las que podemos definir el uso pre-procesadores para comprimir nuestro CSS y JavaScript
- Encriptacion: Uso de seguridad OpenSSL y cifrado AES-256-CBC
- Eventos: Definir, registrar y escuchar eventos en tu aplicación es muy sencillo
- Objet-Relation-Map(ORM)
- Unit Testing
- Cola de tareas: opción de ejecutar procesos largos y complejos en segundo plano usando lista de tareas.
- Symfony: Es un framework de tipo full-stack construido con varios componentes independientes creando aplicaciones y sitios web seguros y de forma profesional entre sus características principales se encuentran [14]:
  - Fácil de instalar y configurar en la mayoría de plataformas
  - Independiente del sistema gestor de bases de datos. Su capa de abstracción y el uso de ORM
  - Utiliza programación orientada a objetos y características

- Basado en mvc con variantes en la capa de abstracción de base de datos
- Sigue la mayoría de mejoras prácticas y patrones de diseño para la web
- Ideal para aplicaciones empresariales. Es lo suficientemente estable como para desarrollar aplicaciones a largo plazo
- Fácil de extender, lo que permite su integración con las Bibliotecas de otros fabricantes.
- URLs inteligentes para el uso de direcciones amigable de las páginas de la aplicación
- Uso templates y layouts.

## 5. DESARROLLO DEL PROTOTIPO

### 5.1. DEFINICIÓN DE LAS HERRAMIENTAS

Para el desarrollo definimos una línea base dado que muchos desarrolladores crean aplicaciones web en la nube basados en una pila de LAMP tomaremos las siguientes condiciones básicas:

- Ubuntu 14.0, sistema operativo
- Apache HTTP Server, el servidor web basado en el trabajo de Apache Software Foundation
- MySQL, la base de datos desarrollada por la firma sueca MySQLAB, propiedad de Oracle Corporation.
- PHP, el lenguaje de programación preprocesador de hipertexto desarrollado por PHP Group.

[14]Se realizó la respectiva instalación y configuración de cada una de estas componentes, partiendo de este punto se decidió por las características ofrecidas por el framework Laravel con el siguiente proceso:

- En primer lugar se verifico el cumplimiento de los requerimientos mínimos para su instalación donde se solicitan las siguientes condiciones para su correcto funcionamiento:
  - PHP >= 5.6.4
  - OpenSSL PHP Extensión
  - DOP PHP Extensión
  - Mbstring PHP Extensión
  - Tokenizer PHP Extensión

- XML PHP Extensión
- Laravel utiliza para gestionar sus dependencias el composer
  - \$ curl -sS <https://getcomposer.org/installer.php>
  - \$ sudo mv composer.phar /usr/local/bin/composer
- Instalación del laravel
  - \$ composer global require "laravel/installer"

```
drwxr-xr-x 10 root www-data 4096 ene 12 11:59 app
-rw-r--r-- 1 root www-data 1646 ene 12 11:59 artisan
drwxr-xr-x 3 root www-data 4096 ene 12 11:59 bootstrap
-rw-r--r-- 1 root www-data 1272 ene 12 11:59 composer.json
-rw-r--r-- 1 root www-data 114403 ene 12 12:05 composer.lock
drwxr-xr-x 2 root www-data 4096 ene 12 11:59 config
drwxr-xr-x 5 root www-data 4096 ene 12 11:59 database
-rw-r--r-- 1 root www-data 503 ene 12 11:59 gulpfile.js
-rw-r--r-- 1 root www-data 212 ene 12 11:59 package.json
-rw-r--r-- 1 root www-data 1026 ene 12 11:59 phpunit.xml
drwxr-xr-x 2 root www-data 4096 ene 12 11:59 public
-rw-r--r-- 1 root www-data 1918 ene 12 11:59 readme.md
drwxr-xr-x 5 root www-data 4096 ene 12 11:59 resources
-rw-r--r-- 1 root www-data 567 ene 12 11:59 server.php
drwxrwxr-x 5 root www-data 4096 ene 12 11:59 storage
drwxr-xr-x 2 root www-data 4096 ene 12 11:59 tests
drwxr-xr-x 29 root www-data 4096 ene 12 12:05 vendor
```

*Figura 6. Archivos de proyectos de laravel fuente :autor*

[15] Cuando creamos cada uno de los proyectos laravel nos crea una estructura de ficheros para organizar nuestro código como observamos en la figura 4. Al crear un nuevo proyecto de Laravel se nos generará una estructura con los siguientes ficheros:

- app – Contiene el código principal de la aplicación. Esta carpeta a su vez está dividida en muchas subcarpetas que analizaremos en la siguiente sección.

- config – Aquí se encuentran todos los archivos de configuración de la aplicación: base datos, cache, correos, sesiones o cualquier otra configuración general de la aplicación.
- database – En esta carpeta se incluye todo lo relacionado con la **definición de la base de datos** de nuestro proyecto. Dentro de ella podemos encontrar a su vez tres carpetas: *factores*, *migrations* y *seeds*. public – Es la única carpeta pública, la única que debería ser **visible** en nuestro servidor web. Todo las peticiones y solicitudes a la aplicación pasan por esta carpeta, ya que en ella se encuentra el *index.php*, este archivo es el que inicia todo el proceso de ejecución del *framework*. En este directorio también se alojan los archivos CSS Javascript, imágenes y otros archivos que se quieran hacer públicos.
- resources – Esta carpeta contiene a su vez tres carpetas: *assets*, *views* y *lang*: resources/views – Este directorio contiene las vistas de nuestra aplicación. En general serán plantillas de HTML que usan los controladores para mostrar la información. Hay que tener en cuenta que en esta carpeta no se almacenan los Javascript, CSS o imágenes, ese tipo de archivos se tienen que guardar en la carpeta public
- artisan: Esta utilidad nos va a permitir realizar múltiples tareas necesarias durante el proceso de desarrollo o despliegue a producción de una aplicación, por lo que nos facilitará y acelerará el trabajo.
- bootstrap – En esta carpeta se incluye el código que se carga para procesar cada una de las llamadas a nuestro proyecto. Normalmente no tendremos que modificar nada de esta carpeta.
- composer.json – Este fichero es el utilizado por Composer para realizar la instalación de Laravel. En una instalación inicial únicamente se especificará la

instalación de un paquete, el propio *framework* de Laravel, pero podemos especificar la instalación de otras librerías o paquetes externos que añadan funcionalidad a Laravel.

- tests – Esta carpeta se utiliza para los ficheros con las pruebas automatizadas. Laravel incluye un sistema que facilita todo el proceso de pruebas con PHPUnit.
- vendor – En esta carpeta se alojan todas las librerías y dependencias que conforman el *framework* de Laravel. Esta carpeta tampoco la tendremos que modificar, ya que todo el código que contiene son librerías que se instalan y actualizan mediante la herramienta Composer.
- storage – En esta carpeta Laravel almacena toda la información interna necesarios para la ejecución de la web, como son los archivos de sesión, la caché, la compilación de las vistas, meta información y los logs del sistema. Normalmente tampoco tendremos que tocar nada dentro de esta carpeta, unicamente se suele acceder a ella para consultar los logs.
- bootstrap – En esta carpeta se incluye el código que se carga para procesar cada una de las llamadas a nuestro proyecto. Normalmente no tendremos que modificar nada de esta carpeta.[8]

Al concluir el proceso de configuración del laravel se procede a la configuración de sponly con el enjaular los usuarios que tenga acceso a la instancia virtual el proceso de instalación y configuración es el siguiente [16]:

- Prime se instalará make para poder instalar el paquete sponly binario / binarios.

- Editar / etc / shells e incluir la líneas, incluyendo la ruta completa.
  - /usr/local/bin/scponly
  - /usr/local/sbin/scponly
  
- Utilice el comando adduser o pw del sistema para crear el usuario.
  - adduser -d /pub -s /usr/local/bin/scponly scpdemo
  - adduser -d /pub -s /usr/local/sbin/scponly scpdemo
  
- Instalar algunos directorios, archivos passwd, bibliotecas y binarios en su ruta de chroot para que scponly
  
- Los Scripts de ejecución se encuentran en la siguiente ruta
  - ../scponly/
  
- Se cambiaron las condiciones del script generar\_usuarios.sh donde se definió que cada uno de los usuarios con software de 185MB y hardware de 200MB.
  
- Se creó el generacion\_base.sh donde se genera el usuario de php con el objetivo que puedan crear las bases de datos de cada uno.

## 6. FACTURACIÓN

### 6.1 TARIFAS DE SERVICIOS

En base al estudio previo de las diferentes plataformas como salesforce, amazon web service y Microsoft azure definimos las tasas de cobros por usuarios de acuerdo al tiempo de uso y los bytes transferidos por cada uno de los usuarios definidos por sistemas operativos, tamaño de memoria y tipo de complementos.

**6.1.1 LAMP.** Instancias definidas con tarifas de subida y bajada de bytes de 27.94, 20.94 respectivamente las tarifas por tiempo de uso las definimos en la tabla 11.

Tabla 5. *Costos LAMP*

<b>Nombre</b>	<b>Memoria GB</b>	<b>Sistema Operativo</b>	<b>Costos /hora</b>
Lamp_nano	1	Ubuntu 14	17.2
Lamp_micro	2	Ubuntu 14	35
Lamp_small	5	Ubuntu 14	137.08
Lamp_largue	10	Ubuntu 14	274.16
Lamp_Xlargue	20	Ubuntu 14	1096.64

*Fuente: Autor*

**6.1.2 Linux postgres.** Instancias definidas con tarifas de subida y bajada de bytes de 29.94, 30.94 respectivamente las tarifas por tiempo de uso las definimos en la tabla 12.

Tabla 6. Costos Linux Postgres

Nombre	Memoria GB	Sistema Operativo	Costos/hora
Linuxpostg_nano	1	Ubuntu 14	18.2
Linuxpostg_micro	2	Ubuntu 14	40
Linuxpostg_small	5	Ubuntu 14	150.08
Linuxpostg_largue	10	Ubuntu 14	300.16
Linuxpostg_Xlargue	20	Ubuntu 14	1196.64

*Fuente: Autor*

### 6.1.3. Jommla

Instancias definidas con tarifas de subida y bajada de bytes de 28.94, 31.94 respectivamente las tarifas por tiempo de uso las definimos en la tabla 13.

Tabla 7. Costos Jommla

Nombre	Memoria GB	Sistema Operativo	Costos/hora
Joomla_nano	1	Ubuntu 14	19.4
Joomla_micro	2	Ubuntu 14	40
Joomla_small	5	Ubuntu 14	128.08
Joomla_largue	10	Ubuntu 14	240.16
Joomla_Xlargue	20	Ubuntu 14	1256.64

*Fuente: Autor*

**6.1.4. tomcat.** Instancias definidas con tarifas de subida y bajada de bytes de 35.94, 40.94 respectivamente las tarifas por tiempo de uso las definimos en la tabla 14.

Tabla 8. Costos Tomcat

Nombre	Memoria GB	Sistema Operativo	Costos/hora
tom_nano	1	Ubuntu 14	22.4
tom_micro	2	Ubuntu 14	45
tom_small	5	Ubuntu 14	138.08
tom_largue	10	Ubuntu 14	250.16
tom_Xlargue	20	Ubuntu 14	1400.64

**6.1.5 Programación en la web.** Instancias definidas con tarifas de subida y bajada de bytes de 43.94, 45.94 respectivamente las tarifas por tiempo de uso las definimos en la tabla 15.

Tabla 9. Costos de Programación web

<b>Nombre</b>	<b>Memoria GB</b>	<b>Sistema Operativo</b>	<b>Costos/hora</b>
programa_nano	1	Ubuntu 14	20,4
programa_micro	2	Ubuntu 14	28
programa_small	5	Ubuntu 14	168.08
programa_largue	10	Ubuntu 14	340.16
Programa_Xlargue	20	Ubuntu 14	1020.64

## **6.2. LIQUIDACIÓN DE SERVICIOS**

Para la liquidación de los servicios en base a las tarifas obtenemos los consumos por cada uno de las instancias y procedemos a emitir la factura a cada uno de los usuarios.

## 6.2.1 Diseño

### 6.2.1.1 Modelo de Entidad Relación

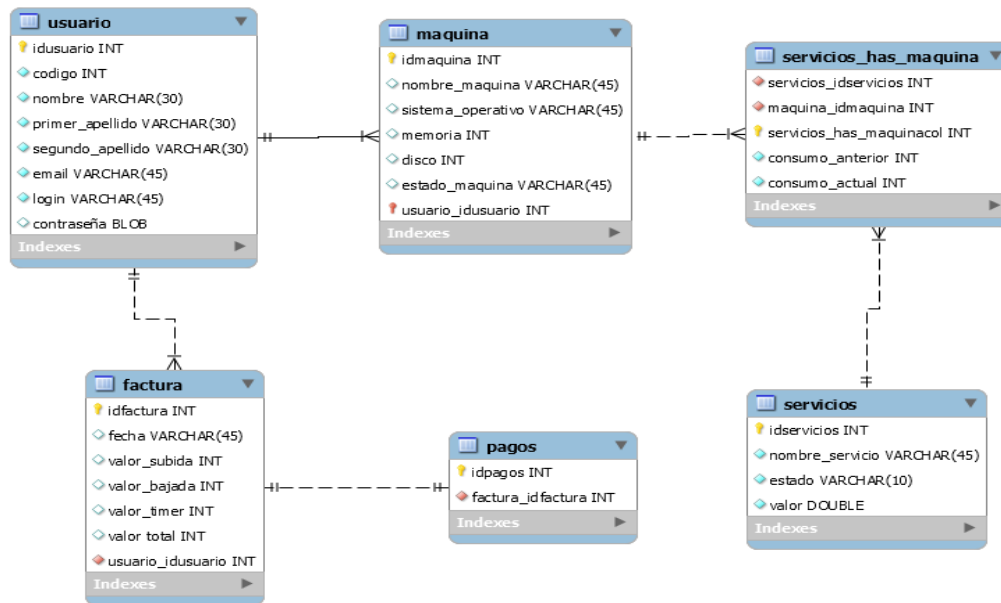


Figura 7. *Modelo de Entidad Relación fuente: autor*

**6.2.2. Proceso de Obtención de los consumos.** Se definieron unos scripts donde se obtenemos las variables de consumo de la instancia los cuales se ejecutarán mensualmente de acuerdo a las políticas definidas en el SLA obteniendo con ellos las variables de tiempo de usos de la máquina, los bytes de subida y bajada los cuales generan dos archivos txt:

- Bytes.txt
- Timer.txt
- El siguiente paso es la captura de estos datos para emitir la factura de cada uno de los usuarios con sus instancias asignadas, este proceso se realiza con

un script encargado de leer los txt generados de consumos de la máquina virtual el archivo es:

- Cargadatos.php

El cual se encarga de procesar los consumos y hacer la carga de los datos, para la emisión final de la factura de cada uno de los usuarios, que pueden ser consultadas en la plataforma como se observa en la siguiente figura 8

The screenshot shows a web application interface. At the top, there is a dark blue header with the text "PAAS" on the left and "CLIENTES FACTURACION CERRAR SESIÓN" on the right. Below the header, there is a link "Nuevo Usuario" in green. The main content area features a table with the following data:

Codigo	Nombre	Primer Apellido	Segundos Apellido	Correo	
2101717	manuel	guillermo	florez	manuel@hotmail.com	<a href="#">Facturar</a>
2101721	jerson	villa	jaimes	gato	<a href="#">Facturar</a>
2103210	jeferson	642b9224c70a270e4443125d05e066	galves	j@hotmail.com	<a href="#">Facturar</a>
1	paola	garrido	cruz	polag1@hotmail.com	<a href="#">Facturar</a>

At the bottom of the page, there is a dark blue footer. On the left, it says "LOCALIZACION" followed by "Universidad Industrial de Santander" and "Paola Garrido". In the center, there are five circular social media icons: Facebook, Google+, Twitter, LinkedIn, and YouTube. On the right, it says "GID-CONUSS" followed by "Grupo de investigación y desarrollo en computación en la nube, servicios,".

Figura 8. Facturas generadas por la plataforma fuente: autor

## 7. SLA

Plantearemos dos SLA uno para la comunidad académica de Universidad Industrial de Santander y el otro caso para los usuarios externos.

### 7.1. SLA ESTUDIANTES Y PROFESORES DE LA UNIVERSIDAD INDUSTRIAL DE SANTANDER

Este acuerdo establece los términos y condiciones de uso aplicables para usted (el "USUARIO") en su uso de los servicios prestados (los "SERVICIOS") por el Grupo de Investigación y Desarrollo en Computación en la Nube, Seguridad, Servidores y Servicios ("GID-CONUSS") para la plataforma como servicio (PaaS) de computación en la nube cloudEISI.

Al usar hacer uso de los SERVICIOS, el USUARIO; automáticamente acepta estos Términos y Condiciones de Uso y acuerda registrarse por ellos.

#### 1. Usuarios elegibles.

Los servicios; están abiertos para el uso de cualquier miembro de la comunidad académica de la Escuela de Ingeniería de Sistemas e Informática de la Universidad Industrial de Santander.

#### 2. Cuenta de usuario, contraseña y seguridad. Como parte de la creación de una cuenta de usuario se otorgará un nombre de cuenta y una contraseña. El USUARIO; es completamente responsable por mantener la confidencialidad de su cuenta o contraseña. Si el USUARIO tiene conocimiento de que la privacidad de su contraseña ha quedado expuesta, debe notificar

inmediatamente a GID-CONUSS; e informar acerca de cualquier fallo de seguridad.

3. Contenidos del usuario. El USUARIO acepta usar los SERVICIOS sólo para fines legales y queda prohibida la publicación, distribución y alojamiento en los SERVICIOS de:

(a) cualquier material ilegal, perjudicial, amenazante, abusivo, acosador, difamatorio u obsceno de cualquier tipo, incluyendo, entre otros, cualquier material que fomente conductas que constituirían una ofensa penal, que produjera responsabilidad civil o que de otro modo infrinja cualquier ley aplicable local, estatal, nacional o internacional.

(b) del cual el USUARIO no tenga el derecho de poner a disposición conforme a ninguna ley o relación contractual

(c) infrinja cualquier derecho de patente, marca comercial, secreto de mercado, derecho de autor u otros derechos de propiedad de cualquier parte.

El USUARIO; acepta no transmitir, subir, publicar, enviar por email o poner a disposición de otro modo virus informáticos, publicidad no solicitada o no autorizada, material promocional o de venta, lo que incluye cadenas de correspondencia, correos masivos o cualquier otra forma de spam.

El USUARIO acepta NO

- (i) Falsificar encabezados o manipular de otro modo los identificadores de archivos con el objeto de ocultar el origen de cualquier contenido del USUARIO
- (ii) Recopilar o almacenar información de identificación personal acerca de otros usuarios sin la debida autorización y consentimiento de estos
- (iii) Utilizar herramientas de interceptación de datos
- (iv) Hacer uso de software malicioso o de obstaculización de redes

- (v) Realizar accesos abusivos a los equipos informáticos que permiten la prestación de los SERVICIOS o la prestación los SERVICIOS; El USUARIO; reconoce que GID-CONUSS; tiene el derecho (pero no la obligación), de eliminar el contenido del USUARIO; que viole estos Términos y Condiciones, y nos reservamos el derecho de denegar el servicio; y/o cancelar cuentas sin previo aviso para cualquier usuario que viole estos Términos y Condiciones o infrinja los derechos de terceros

#### 4. Seguridad de los servicios.

La seguridad de las instancias asignadas por parte de los SERVICIOS prestados recaen bajo la administración del USUARIO, así como el manejo de las copias de seguridad de los datos, código fuente, aplicaciones y bases de datos que hagan parte de sus contenidos, GID-CONUSS; no se responsabiliza en casos de falla total o parcial de los servidores y que imposibiliten la recuperación de los contenidos por los sistemas de almacenamiento implementados

#### 5. Quejas por derecho de autor.

GID-CONUSS respeta la propiedad intelectual de terceros. Si usted cree que su trabajo ha sido copiado en este sitio de manera que constituya una infracción a sus derechos de autor, puede escribir [cloudeisi@gmail.com](mailto:cloudeisi@gmail.com)"; para notificarnos acerca de la situación.

#### 6. Descargo de responsabilidad

USUARIO reconoce y acepta que utiliza este servicio sin cargo alguno y que por lo tanto este uso se ofrece “tal como está GID-CONUSS expresamente niega toda garantía de cualquier tipo, ya sea expresa o implícita. Como USUARIO; acepta de forma expresa que todo uso de la información disponible en las los SERVICIOS; asignados son para su uso personal o académico y que dicha información se utilizará bajo su propio riesgo y responsabilidad GID-CONUSS expresamente niega toda responsabilidad por errores u omisiones en la información, los materiales y las funciones que se incluyen en nuestros servidores o en los sitios de terceros enlazados desde o hacia nuestros servicios por parte del USUARIO.

#### 7. Indemnización

El USUARIO acepta indemnizar y liberar de toda responsabilidad a GID-CONUSS sus directivos, desarrolladores, investigadores, y administradores en cuanto a cualquier reclamación o demanda, lo que incluye los honorarios razonables de representación legal, entablada por terceros debida a, o que resulte de, el uso por parte del USUARIO de los SERVICIOS prestados.

#### 8. Potestad jurídica.

Los SERVICIOS son administrados por GID-CONUSS desde sus oficinas dentro de la República de Colombia. Al hacer uso de los SERVICIOS, tanto el USUARIO como GID-CONUSS aceptan que todos los asuntos relacionados con su uso de los SERVICIOS estarán regidos e interpretados de acuerdo con la legislación de la República de Colombia, sin considerar los principios de conflictos de leyes, como si se realizaran enteramente dentro de la República de Colombia.

## 9. Modificación de los términos de uso

EL servicio GID-CONUSS puede cambiar los términos y condiciones de los Términos de Uso del Servicio GID-CONUSS hará las notificaciones de los cambios y publicará la versión más reciente en la sección Términos de Uso del sitio <http://cloudeisi.uis.edu.co> si el USUARIO no está de acuerdo con las modificaciones puede dejar de usar los SERVICIOS informando a GID-CONUSS.

## 10. Confidencialidad de la Información

GID-CONUSS no compartirá ni revelará la información confidencial con terceros, excepto que tenga expresa autorización de parte del USUARIO o cuando ha sido requerido por orden judicial o legal.

## **7.2. SLA USUARIOS EXTERNOS**

Este acuerdo establece los términos y condiciones de uso aplicables para usted (el "USUARIO") en su uso de los servicios prestados (los "SERVICIOS") por el Grupo de Investigación y Desarrollo en Computación en la Nube, Seguridad, Servidores y Servicios ("GID-CONUSS") para la plataforma como servicio (PaaS) de computación en la nube cloudEISI.

Al usar hacer uso de los SERVICIOS, el USUARIO; automáticamente acepta estos Términos y Condiciones de Uso y acuerda registrarse por ellos.

### 1. Usuarios elegibles.

Los servicios; están abiertos para el uso de cualquier miembro de la externo a la comunidad académica de la Universidad Industrial de Santander.

2. Cuenta de usuario, contraseña y seguridad. Como parte de la creación de una cuenta de usuario se otorgará un nombre de cuenta y una contraseña. El USUARIO; es completamente responsable por mantener la confidencialidad de su cuenta o contraseña. Si el USUARIO tiene conocimiento de que la privacidad de su contraseña ha quedado expuesta, debe notificar inmediatamente a GID-CONUSS; e informar acerca de cualquier fallo de seguridad.
  
3. Contenidos del usuario. El USUARIO acepta usar los SERVICIOS sólo para fines legales y queda prohibida la publicación, distribución y alojamiento en los SERVICIOS de:
  - (a) Cualquier material ilegal, perjudicial, amenazante, abusivo, acosador, difamatorio u obsceno de cualquier tipo, incluyendo, entre otros, cualquier material que fomente conductas que constituirían una ofensa penal, que produjera responsabilidad civil o que de otro modo infrinja cualquier ley aplicable local, estatal, nacional o internacional.
  - (b) Del cual el USUARIO no tenga el derecho de poner a disposición conforme a ninguna ley o relación contractual
  - (c) Infrinja cualquier derecho de patente, marca comercial, secreto de mercado, derecho de autor u otros derechos de propiedad de cualquier parte. El USUARIO acepta no transmitir, subir, publicar, enviar por email o poner a disposición de otro modo virus informáticos, publicidad no solicitada o no autorizada, material promocional o de venta, lo que incluye cadenas de correspondencia, correos masivos o cualquier otra forma de spam.El USUARIO acepta NO
  - (vi) Falsificar encabezados o manipular de otro modo los identificadores de archivos con el objeto de ocultar el origen de cualquier contenido del USUARIO

- (vii) Recopilar o almacenar información de identificación personal acerca de otros usuarios sin la debida autorización y consentimiento de estos
- (viii) Utilizar herramientas de interceptación de datos
- (ix) Hacer uso de software malicioso o de obstaculización de redes
- (x) Realizar accesos abusivos a los equipos informáticos que permiten la prestación de los SERVICIOS o la prestación los SERVICIOS, El USUARIO reconoce que GID-CONUSS; tiene el derecho (pero no la obligación), de eliminar el contenido del USUARIO; que viole estos Términos y Condiciones, y nos reservamos el derecho de denegar el servicio y/o cancelar cuentas sin previo aviso para cualquier usuario que viole estos Términos y Condiciones o infrinja los derechos de terceros

#### 4. Seguridad de los servicios.

La seguridad de las instancias asignadas por parte de los SERVICIOS prestados recaen bajo la administración del USUARIO, así como el manejo de las copias de seguridad de los datos, código fuente, aplicaciones y bases de datos que hagan parte de sus contenidos, GID-CONUSS; no se responsabiliza en casos de falla total o parcial de los servidores y que imposibiliten la recuperación de los contenidos por los sistemas de almacenamiento implementados

#### 5. Quejas por derecho de autor.

GID-CONUSS respeta la propiedad intelectual de terceros. Si usted cree que su trabajo ha sido copiado en este sitio de manera que constituya una

infracción a sus derechos de autor, puede escribir [cloudeisi@gmail.com](mailto:cloudeisi@gmail.com) para notificarnos acerca de la situación.

#### 6. Descargo de responsabilidad

USUARIO reconoce y acepta que utiliza este servicio con los cargos de tráfico de bytes de subida y bajada, el tiempo de uso de la instancia asignada al USUARIO y se acepta los cargos económicos generados por uso de su máquina virtual de GID-CONUSS. Como USUARIO; acepta de forma expresa que todo uso de la información disponible en los SERVICIOS; asignados son para su uso personal y que dicha información se utilizará bajo su propio riesgo y responsabilidad. GID-CONUSS expresamente niega toda responsabilidad por errores u omisiones en la información, los materiales y las funciones que se incluyen en nuestros servidores o en los sitios de terceros enlazados desde o hacia nuestros servicios por parte del USUARIO.

#### 7. Indemnización

El USUARIO acepta indemnizar y liberar de toda responsabilidad a GID-CONUSS sus directivos, desarrolladores, investigadores, y administradores en cuanto a cualquier reclamación o demanda, lo que incluye los honorarios razonables de representación legal, entablada por terceros debida a, o que resulte de, el uso por parte del USUARIO de los SERVICIOS prestados.

#### 8. Potestad jurídica.

Los SERVICIOS son administrados por GID-CONUSS desde sus oficinas dentro de la República de Colombia. Al hacer uso de los SERVICIOS, tanto el USUARIO como GID-CONUSS aceptan que todos los asuntos

relacionados con su uso de los SERVICIOS estarán regidos e interpretados de acuerdo con la legislación de la República de Colombia, sin considerar los principios de conflictos de leyes, como si se realizaran enteramente dentro de la República de Colombia.

#### 9. Modificación de los términos de uso

EL servicio GID-CONUSS puede cambiar los términos y condiciones de los Términos de Uso del Servicio GID-CONUSS hará las notificaciones de los cambios y publicará la versión más reciente en la sección Términos de Uso del sitio <http://cloudeisi.uis.edu.co> si el USUARIO no está de acuerdo con las modificaciones puede dejar de usar los SERVICIOS informando a GID-CONUSS.

#### 10. Confidencialidad de la Información

GID-CONUSS no compartirá ni revelará la información confidencial con terceros, excepto que tenga expresa autorización de parte del USUARIO o cuando ha sido requerido por orden judicial o legal.

## 8. ESCENARIOS DE DESARROLLO

### 8.1. HOSTING WEB PARA CURSO DE PROGRAMACIÓN WEB

Para este primer escenario plantearemos el uso de la instancia para los estudiantes y profesores de la materia de programación web de la escuela de Ingeniería de sistemas de la Universidad Industrial de Santander donde al profesor de la materia de programación web, con todos los complementos necesarios generándoles un servicio de hosting.

Para efectos de administración se generaran los procesos de generación de los entornos y la posteríos administración de los mismos.

#### 8.1.1. Proceso de generación de Usuario enjaulado.

8.1.1.1. Conexión por ssh. Para el uso de esta instancia se le hace entrega de usuario y password para el ingreso a la respectiva instancia, donde el profesor estará en las condiciones de generar los usuarios enjaulados que considere necesarios, el cual describiremos a continuación:

- Debe generar una conexión ssh con la ip asignada puede usar el programa de putty como se observa en la figura 6

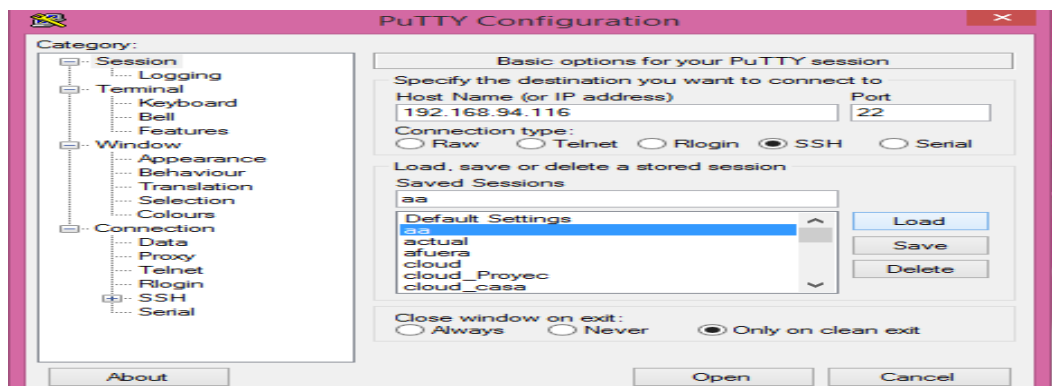


Figura 9. Conexión a putty fuente: autor

- El usuario se loguea donde se le dan privilegios de root

8.1.1.2. Ubicación de archivos dentro del sistema. Posteriormente nos movemos dentro de las carpetas del sistema operativo como se observa donde se encuentran los archivos necesarios para iniciar el proceso de generación de cada uno de los usuarios del sistema en la siguiente ruta

- `usr/scponly`

8.1.1.3 Creación de los usuarios de programación web dentro del sistema

- Para generar los usuarios enjaulados debemos ejecutar el script `generación de usuarios` el cual se puede ejecutar así:

- `./script_generar_usuarios.sh usuario`

Al ejecutar el script se sugiere el siguiente formato con el objeto de tener un control cronológico el cual se observa en la tabla 5

Tabla 10. *Formato de generación de usuarios*

USUARIO	FORMATO	20171B101
AÑO	XXXX	2017
SEMESTRES	X	1
GRUPO	XX	B1
SUBGRUPO	XX	01

Este usuario nuevo dentro del sistema con privilegios de sftp, nos almacena su usuario y contraseña en el archivo `misusuarios.txt`

8.1.1.4. Creación de la base de datos para los estudiantes de programación web.

- El proceso a seguir es la creación de la base de datos y el usuario de la base de datos.

- `./generacion_base.sh`

Tabla 11. *Generación de bases de datos*

SCRIPT	BASE DE DATOS	USUARIO	PASSWORD
generacion_base.sh	20171B101	20171B101	Y6PggOipwf

Como se observa en la siguiente imagen el usuario ya tiene creada una base de datos, en la cual se puede logear y acceder a la generación de su base de datos como se observa en la figura .

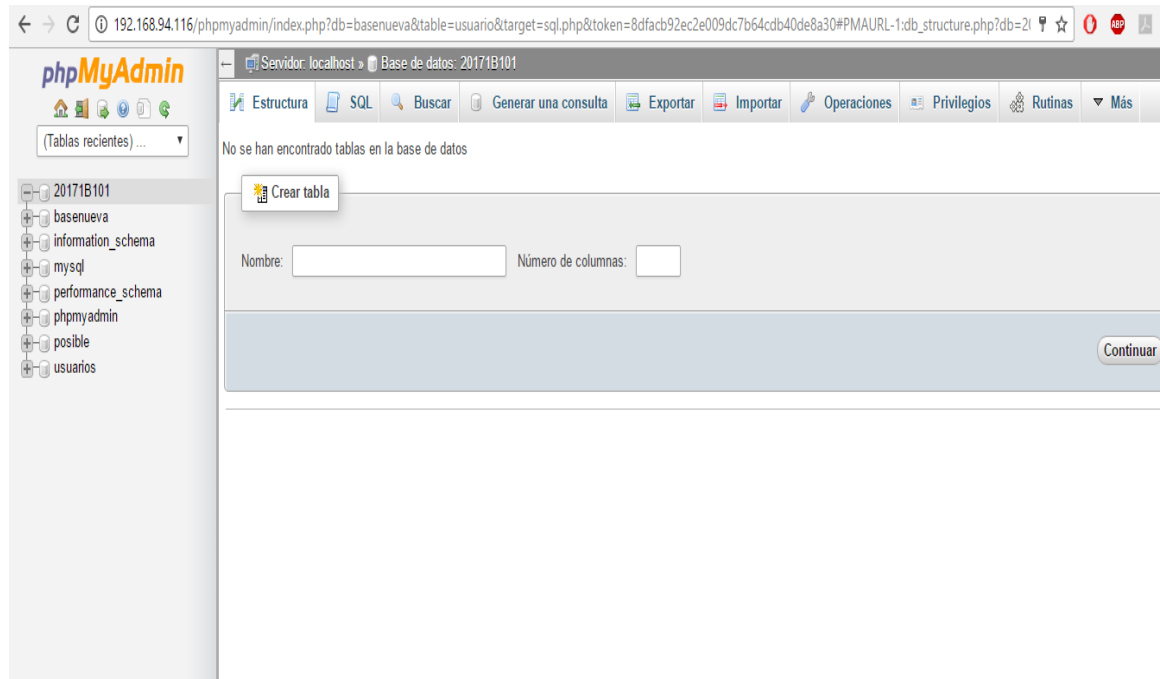


Figura 10. *phpmyadmin después de ejecución de scriptsql.sh fuente: autor*

Con el proceso anterior damos por finalizado el procedimiento de generación de las instancias para los usuarios de programación web donde los estudiantes pueden acceder y transmitir sus archivos por ftp como se observa en la figura 10 ofreciéndoles un entorno real de hosting en producción

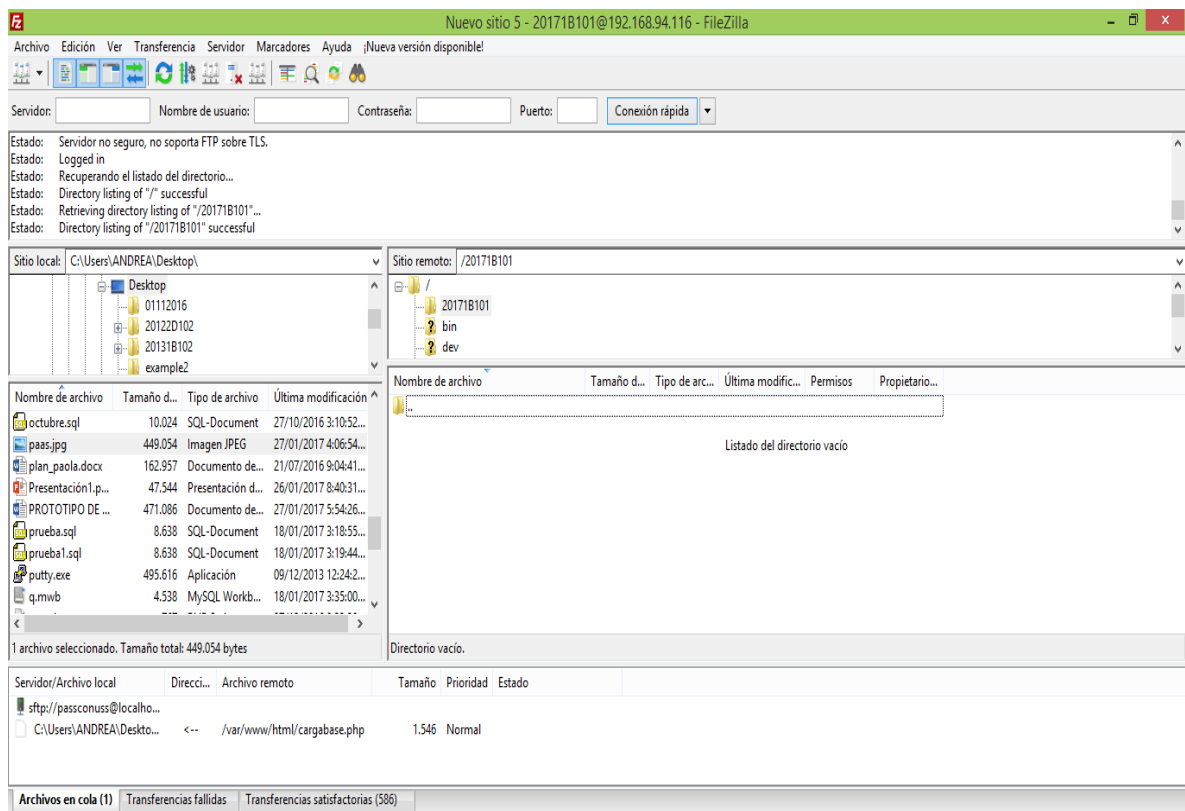


Figura 11. *Conexión ftp del usuario fuente: autor*

**8.1.2 Administración de los usuarios de programación web.** Con el objetivo de mantener un control sobre los límites de transferencia de archivos, se crearon unos scripts los cuales ofrecen diferentes opciones de control al administrador de la instancia, en este caso el profesor de la materia de programación web.

8.1.2.1. Abrir usuario. Se pueden abrir un usuario durante un periodo o indefinidamente para esto tenemos el siguiente formato como se observa en la tabla donde definiremos en primer lugar por un lapso de tiempo e interminablemente como se observa en la tabla 7, el formato es el siguiente

- xxxxx.sh USUARIO TIEMPO

Tabla 12. *Formato de ejecución del script para abrir usuario*

<b>SCRIPT</b>	<b>USUARIO</b>	<b>TIEMPO(min)</b>
abrir_usuario.sh	20171B101	10
abrir_usuario.sh	20171B101	

8.1.2.2. Abrir diferentes usuarios Con los formatos previamente definidos para la generación de los usuarios le damos la opción al administrador de activar la transferencia de datos de un grupo de usuarios que posean las mismas características a nivel de grupo como se observa en la tabla 8, con el siguiente formato:

- xxxxx.sh USUARIO TIEMPO

Tabla 13. *Abrir grupo de programación web*

<b>SCRIPT</b>	<b>USUARIO</b>	<b>TIEMPO(min)</b>
abrir_gusuario.sh	B1	10
abrir_gusuario.sh	B1	

8.1.2.3. Cerrar usuarios. *Con este script impedimos la transferencia de archivos sftp para la ejecución lo realizamos de acuerdo al formato que observamos en la tabla 9, y con el siguiente formato*

- xxxxx.sh USUARIO

Tabla 14. *Formato de ejecución de script cerrar usuarios*

<b>SCRIPT</b>	<b>USUARIO</b>
cerar_usuario.sh	20171B101

8.1.2.4. Cerrar grupos. En base al formato definido en la generación de usuarios para evitar estar cerrando cada uno de los usuarios lo realizaremos de acuerdo a la característica de subgrupo facilitando el trabajo del administrador el cual se ejecuta como se observa en la tabla 10.

- xxxxx.sh USUARIO

Tabla 15. Formato de ejecución del script de cerrar grupo de usuarios

SCRIPT	USUARIO
cerar_gusuario.sh	B1

8.1.2.5 Backups de proyectos de usuarios. Permite almacenar y comprimir los archivos de proyectos de los usuarios con el siguiente formato de ejecución, el cual se almacena dentro de la carpeta backups\_proyectos.

- ./baskup.sh

8.1.2.6 Backups de base de datos de usuarios. Permite mover los archivos de las copias de las bases de datos generadas por el webmin en el cual se define dentro de un cron de ejecución acción que puede ser modificada de acuerdo a los parámetros que considere el administrador de la instancia.

- ./baskupsql.sh

## 8.2 HOSTING WEB INDIVIDUAL

**8.2.1 Planteamiento del escenario.** Para este segundo escenario tomaremos el caso en el que el usuario no pertenezca a la comunidad académica de la escuela de ingeniería de sistemas de la Universidad Industrial de Santander, hecho en el cual aunque las condiciones de la instancia son similares este posee un estándar de cobro por los servicios ofrecidos por el grupo de investigación GID-CONUSS.

**8.2.2 Registro y asignación de la instancia.** El usuario se registra en la plataforma, posteriormente el administrador se pondrá en contacto con el usuario interesado en solicitar los servicios de la plataforma como servicio, con el objetivo de realizar la asignación de la instancia de acuerdo a los servicios que solicito.

**8.2.3 Uso de la instancia.** El usuario realiza los procesos que considere pertinentes dentro de la máquina virtual que le fue asignada sin incumplir el SLA definido para este tipo de usuario, realizando las modificaciones necesarias a su instancia de acuerdo a sus necesidades. Posteriormente se realiza la liquidación del servicio de acuerdo a los consumos realizados por el usuario de la instancia, con una facturación de consumo mensual la cual puede ser consultada en la plataforma como se observa en la imagen

## PLATAFORMA COMO SERVICIO

Codigo:	2101717	GID-CONUSS
Nombres:	manuel	Cra 27 #9 - 11,
Apellidos:	guillermo florez	Bucaramanga, Colombia
Correo:	manuel@hotmail.com	316-4734415
		<a href="mailto:cloudeisi@gmail.com">cloudeisi@gmail.com</a>
Datos	Descripcion	
Fecha de consumo	25-01-17	
Subida (bits)	14844	
Bajada (bits)	4105224	
Tiempo (horas)	50.61	
Total	13,989	

Figura 12. *Factura para usuarios*

## 9. CONCLUSIONES

- El ahorro en costos de servicio de hosting, con el método de cobro por tiempo de uso es más rentable para los usuarios, otro factor que beneficia a los usuarios es la existencia de funciones dedicadas de acuerdo a las necesidades, evitándole el proceso tedioso de la instalación y configuración de la instancia.
- Las políticas SLA, se definieron de acuerdo a los tipos usuarios, con las diferentes características de acuerdo a los escenarios en el que se plantea el uso de las instancias.
- La administración de la infraestructura del grupo de investigación GID-CONUSS, nos brinda un entorno de aprendizaje similar al entorno real usado a nivel corporativo manejado por compañías como red hat, amazon, ibm generando capacidades nuevas dentro del entorno de la administración de cloud computing
- La instancia de desarrollo de programación en la web nos demuestra la iteración con el entorno real de un hosting, para los estudiantes con toda la funcionalidad para el desarrollo de los proyectos del semestre y el control por parte del docente o el administrador de la máquina virtual asignada, junto con la reutilización del contiguo facilitando la labor de el programador de aplicaciones.

## 10. RECOMENDACIONES

Se propone como un futuro proyecto la iniciación de software como servicio para los estudiantes de la escuela utilizando de base la infraestructura adquirida hasta el momento.

Configuración de los módulos de telemetría del opensatck con el objetivo de visualizar de manera más concreta todas las variables para la administración de los usuarios con instancias virtuales.

Inclusión de nuevo hardware y adecuación del laboratorio de pruebas que permitan a los nuevos integrantes del grupo de investigación realizar labores de adiestramiento como futuros administradores de la infraestructura.

## REFERENCIAS BIBLIOGRAFICAS

- [1] Sosinsky, B. (2011). *¿Qué es la nube? El futuro de los sistemas de información*. Anaya Multimedia-Anaya Interactiva.
- [10] Gallego Sanchez, A. J. (s.f.). Obtenido de <https://ajgallego.gitbooks.io/laravel-5/content/index.html>
- [11] Lizaraso, J. E., & Nogera, D. (2013). *Administración, mantenimiento, configuración de los equipos servidores del grupo GID-CONUSS con énfasis en el análisis y reestructuración de los modelos de alta disponibilidad y computación en la nube*. Bucarmanga.
- [12] Lopez, w., & Moreno, c. (2013). *MIGRACIÓN DE VIRTUALIZACIÓN EN LA INFRAESTRUCTURA*.
- [12] Riehle, D. (2000). *wikipedia*. Obtenido de <https://es.wikipedia.org/wiki/Framework>
- [13] wikipedia. (s.f.). Obtenido de [https://en.wikipedia.org/wiki/Web\\_service](https://en.wikipedia.org/wiki/Web_service)
- [14] eduteka. (s.f.). Obtenido de <http://eduteka.icesi.edu.co/proyectos.php/2/29384>
- [2] CARREÑO DIAZ, E. (2012). *Modelo y prototipo de servicios de computación en la nube para estudiantes y profesores de la escuela de Ingeniería de Sistemas e Informática de la Universidad Industrial de Santander*. Bucaramanga: Universidad Industrial de Santander. Facultad de Ingenierías Fisicomecánicas. Escuela de Ingeniería de Sistemas e Informática.
- [3] Mell, P. &. (2011). *The NIST Definition of cloud Computing. NIST Special Publication 800-145*. <http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf>.
- [5] Services, A. W. (s.f.). Obtenido de <https://aws.amazon.com/es/ec2/>
- [6] azure.microsoft. (s.f.). Obtenido de <https://azure.microsoft.com/es-es/>
- [9] Ambrust, m. (2009). *Abode the Clouds: A Berkeley View of Cloud Computing EECS*. Obtenido de <http://www.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-28.pdf>
- gitbook. (s.f.). Obtenido de <https://www.gitbook.com/book/ajgallego/laravel-5/details>
- github. (s.f.). Obtenido de <https://github.com/scponly/scponly/wiki>

## BIBLIOGRAFIA

Ambrust, m. (2009). Abode the Clouds: A Berkeley View of Cloud Computing EECS. Obtenido de <http://www.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-28.pdf>

azure.microsoft. (s.f.). Obtenido de <https://azure.microsoft.com/es-es/>

CARREÑO DIAZ, E. (2012). Modelo y prototipo de servicios de computacion en la nube para estuidantes y profesores de la escuela de Ingeniera de Sistemas e Informatica de la Universidad Industrial de Santander. Bucaramanga: Uniersidad Industrial de Santander. Facultad de Ingenierias Fisicomecanicas.Escuela de ingeniera de sistemas e informatica.

eduteka. (s.f.). Obtenido de <http://eduteka.icesi.edu.co/proyectos.php/2/29384>

Gallego Sanchez, A. J. (s.f.). Obtenido de <https://ajgallego.gitbooks.io/laravel-5/content/index.html>

gitbook. (s.f.). Obtenido de <https://www.gitbook.com/book/ajgallego/laravel-5/details>

github. (s.f.). Obtenido de <https://github.com/scponly/scponly/wiki>

Lizaraso, J. E., & Nogera, D. (2013). Administracion, mantenimiento, configuracion de los equipos servidores del grupo GID-CONUSS con enfasis en el analisis y reestructuracon de lo modelos de alta disponibilidad y computacion en la nube. Bucarrmanga.

lopez, w., & moreno, c. (2013). MIGRACIÓN DE VIRTUALIZACIÓN EN LA INFRAESTRUCTURA.

Mell, P. &. (2011). The NIST Definition of cloud Computing.NIST Special Publication 800-145. <http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf>.

Riehle, D. (2000). wikipedia. Obtenido de <https://es.wikipedia.org/wiki/Framework>

Services, A. W. (s.f.). Obtenido de <https://aws.amazon.com/es/ec2/>

Sosinsky, B. (2011). ¿Ques es la nube? El futuro de los sistemas de informacion. Anaya Multimedia-Anaya Interactiva.

wikipedia. (s.f.). Obtenido de [https://en.wikipedia.org/wiki/Web\\_service](https://en.wikipedia.org/wiki/Web_service)

## ANEXOS

# ANEXO A: ARTICULO ESCRITO SOBRE EL PROTOTIPO DE UN SERVICIO DE COMPUTACION EN LA NUBE PAAS (PLATAFORMA COMO SERVICIO) Y WEB SERVICE ORIENTADO AL DESARROLLO DE PROGRAMACION WEB



Prototipo de un Servicio de Computación en la nube y web service orientado al desarrollo de programación web  
Garrido Cruz Paola Andrea.

Paola Andrea Garrido Cruz  
polag1@hotmail.com

## 2 CARACTERISTICAS

### 2.1 CARACTERÍSTICAS GENERALES

a) Prototipo de un Servicio de Computación en la nube y web service orientado al desarrollo de programación web  
**TÍTULO**

b) **RESUMEN.** Partimos del estudio del estado del arte de los aspectos y componentes de la computación en la nube centrándonos en las plataformas como servicio y sus modelos estándar de servicios y sus modelos de facturación, con estas variables definimos los servicios posibles a ofrecer para los usuarios de la comunidad académica, de la Universidad Industrial de Santander, como los posibles usuarios externos, definiendo unos estándares de servicios.

c) **ABSTRACT**  
*We start from the study of the state of the art aspects and components of cloud computing focusing on the platforms as a service and its standard models of services and their models of billing, with these variables we define the possible services to be offered to the users of the Academic community, of the Industrial University of Santander, as potential external users, defining service standards.*

d) **PALABRAS CLAVE (KEY WORDS)** Plataforma como servicio, Fraenwork, computación en la nube

e) **INTRODUCCIÓN.** Las plataformas como servicio ofrecen una amplia gama de servicios nosotros nos centraremos en la generación de un modelo para los estudiantes de programación web de la Universidad Industrial de Santander.

f) **MATERIAL Y MÉTODOS.** En primera instancia realizamos un proceso investigado de la computación en la nube y las condiciones necesarias para la generación de un prototipo de plataforma como servicio y definimos nuestras limitaciones en la construcción del mismo. La computación en nube es un modelo para permitir un acceso por la red a un conjunto de recursos computacionales configurables (ej. Redes, servidores, almacenamiento, aplicaciones y servicios) de manera conveniente o bajo demanda, que se puede

**RESUMEN:** *Las computación en la nube a tenido una evolución en el paradigma de la computación en la nube plateándonos nuevos esquemas para el desarrollo de aplicaciones, Nos centraremos en las plataformas como servicio las cuales estudiaremos, analizando sus modelos e interpretando sus componentes, se plantea la generación de una instancia virtual para el desarrollo de aplicaciones para los estudiantes de programación web de la escuela de Ingeniería de Sistemas de la Universidad Industrial de Santander*

**PALABRAS CLAVE:** Plataforma como servicio, Fraenwork, computación en la nube.

### **ABSTRACT.**

*Cloud computing with an evolution in the paradigm of cloud computing plated new schemes for the development of applications, focused on the platforms as a service which we will study, analyzing their models and interpreting their components, consider the generation of A virtual instance for the development of applications for students of web programming of the School of Systems Engineering of the Industrial University of Santanderabstract*

**Keywords.** Platform as a service, Fraenwork, cloud computing

## 1 INTRODUCCIÓN

La constante migración de las tecnologías de información a la computación en la nube definida por el NIST(Instituto Americano de Estándares y Tecnología) como un modelo que permite acceso bajo demanda a través de la red a un conjunto compartido de recursos de computación configurables que pueden ser rápidamente provisionados con el mínimo esfuerzo de gestión o iteración del proveedor del servicio, basándonos en base a estos conocimientos previos nace la idea de generar un prototipo de plataforma como servicio partiendo de la infraestructura actual que posee el grupo GID-CONUSS.



## Prototipo de un Servicio de Computación en la nube y web service orientado al desarrollo de programación web

Garrido Cruz Paola Andrea.

aprovisionar y liberar rápidamente con un mínimo de esfuerzo administrativo o de interacción del proveedor de servicios.”

La definición del NIST para los cuatro modelos de implementación la Nube privada, Nube de la comunitaria, Nube pública y la Nube híbrida

El conjunto de servicios ofrecidos en la nube ofrece 3 modelos de adoptando la nomenclatura XaaS, ó “<Algo> como servicio”, llegando a definir la computación en la nube y la forma en que los usuarios finales pueden acceder a sus recursos y servicios. Los servicios se ha aceptado a nivel universal son infraestructura como servicio IaaS, plataforma como servicio PaaS y software as a service SaaS.

Para el desarrollo de este proyecto se consideró la Infraestructura del grupo de investigación GID-CONUSS y sus servicios ofrecidos, se realizó la configuración de la instancia virtual con los componentes para el desarrollo y administración de la máquina virtual, definiendo como usuario administrador al profesor encargado de dictar la materia de programación web, la cual posee el sistema operativo Ubuntu 14.04, php5, mysql, el framework laravel y el scponly.

Con la scponly se realizó la configuración y la definición de los usuarios enjaulados para software y hardware, el proceso siguiente fue la creación del usuario en la máquina virtual configurada para los estudiantes de programación web, permitiéndole una interacción continua dentro de un entorno de hosting.

Se definió una administración para el profesor donde se crearon scripts de modificación en la transferencia de datos de acceso sftp, para limitar el acceso en determinadas ocasiones como son días límites para la entrega de proyectos de semestre por parte de los estudiantes.

Se realizaron las configuraciones necesarias para generar un módulo de facturación automático de los servicios de máquinas virtuales ofrecidos por el grupo de investigación GID-CONUSS.

Para los usuarios de externos que no hacen parte de la comunidad académica se definió un estándar tarifario donde se tuvieron en cuenta características como el sistema operativo, los complementos funcionales pre configurados.

**RESULTADOS.** La funcionalidad de la generación de usuarios con acceso sftp para los estudiantes de programación web cumple con todo el entorno necesario para el desarrollo de la actividad académica y facilita la administración y control al profesor encargado, ofreciendo un entorno real y contacto directo con un hosting.

Donde se generó una facturación para los usuarios con maquina virtuales asignadas con el respectivo tarifas de acuerdo a los servicios utilizados del hosting, los cuales se generan de forma automática de acuerdo a los ciclos de facturación que definíamos para la plataforma para este caso se planteó un periodo mensual.

Servicio	Descripcion	Precio
Servicio 001		1000
Servicio 002		2000
Servicio 003		3000
Total		15000

h) **DISCUSION.** En base a los modelos de facturación ofrecidos por las plataformas internacionales y en base a los servicios ofrecidos por cada una de ellas desarrollamos un estándar que cumple a cabalidad con los modelos de plataformas como servicios, la ventaja que tenemos frente a estas plataformas es la migración a otra hosting ya que los paquetes que se manejan dentro de la plataforma son de software libre y sin ningún tipo de restricción para la actualización de los repositorios facilitando la modificación de la instancia de acuerdo a las necesidades de los usuarios.

**CONCLUSIONES.** Tenemos una visión general de los modelos de facturación utilizado nos ofrece variables básicas las cuales se pueden ampliar para nuestro prototipo de acuerdo a los nuevos servicios que deseen ofrecer para este caso tomamos las variables que consideramos de acuerdo a las políticas definidas en el SLA generando dos escenarios de acuerdo al entorno en el que se planteó el uso de las instancias



Prototipo de un Servicio de Computación en la nube y web  
service orientado al desarrollo de programación web  
Garrido Cruz Paola Andrea.

El ahorro en costos de servicio de hosting, con el método de cobro por tiempo de uso es más rentable para los usuarios, otro factor que beneficia a los usuarios es la existencia de funciones dedicadas de acuerdo a las necesidades, evitándole el proceso tedioso de la instalación y configuración de la instancia.

j) **LITERATURA CITADA.** osinsky, B. (2011). *¿Qué es la nube? El futuro de los sistemas de información.* Anaya Multimedia-Anaya Interactiva.

Mell, P. &. (2011). *The NIST Definition of cloud Computing.* NIST Special Publication 800-145.  
<http://nvlpubs.nist.gov/nistpubs/Leqacv/SP/nistspecialpublication800-145.pdf>.