APLICACIÓN WEB PROTOTIPO PARA LA EVALUACIÓN DE COSTOS Y APROVISIONAMIENTO CON LIBCLOUD SOBRE AMAZON WEB SERVICES

PRICECLOUD

ANTEPROYECTO - TRABAJO PROFESIONAL

SEBASTIÁN AFANADOR FONTAL
sebastian.afanador@correounivalle.edu.co
Código 1629587

JOHN ALEXANDER SANABRIA ORDÓÑEZ PH.D
john.sanabria@correounivalle.edu.co

Facultad de Ingeniería
Escuela de Ingeniería de Sistemas y Computación
Universidad del Valle
Cali - Colombia

Marzo 2021 – Versión 1.5.0

El Web Scraping son un conjunto de técnicas y tecnologías para extraer información de la web usando algoritmos de exploración recursiva en todos los Endpoints de un sitio web. LibCloud permite generar una capa de abstracción a partir de una Application Programing Interface (API) unificada usando controladores para cada tipo de servicio que prestan muchos de los CCSPs.

RESUMEN

La aplicación web PriceCloud propone integrar la fase de evaluación de costos de los principales proveedores de servicios de Cloud Computing (CC) junto con su despligue en una única interfaz para dar un criterio informado que ayude a minimizar los costos de funcionamiento de un proyecto de software con respecto a sus necesidades iniciales.

Usando principalmente técnicas de *Web Scraping* este trabajo propone crear un servicio que sea una fuente de información veraz y actual de los costos de funcionamiento de los servicios ofrecidos por los principales Cloud Computing Provider (CCSP) y ademas una aplicación que presente esta información al usuario y le permita desplegar sus recursos usando la librería LibCloud[1].

1 https://libcloud.apache.org/about.html

ÍNDICE GENERAL

1 INTRODUCCIÓN

El Cloud Computing ha facilitado el despliegue de todo tipo de aplicaciones con un uso adaptable de recursos, esto es un punto importante del cloud, permitirle al cliente iniciar con recursos de base que luego pueden ser escalados a la medida de las necesidades que la aplicación requiera. En contraposición de una infraestructura *On-premise* las tecnologías cloud se adaptan en tiempo real a los requerimientos cambiantes de la demanda de estas aplicaciones y le permiten al cliente invertir recursos vitales en procesos mas relevantes eliminando los costos de mantenimiento e infraestructura tecnológica.

El CC es una tendencia de servicios y recursos que funcionan sobre Internet para facilitar el despliegue, uso y disponibilidad de aplicaciones de Software.

El aprovisionamiento de recursos en proveedores de servicios cloud es un proceso que cada vez toma mas relevancia en aplicaciones que han apostado por la tendencia a migrarse al CC. Sin embargo a la hora de elegir estos recursos se deben tener en cuenta aspectos como la arquitectura del proyecto, el tipo de recursos, su ubicación geográfica, las políticas donde recidirán, la reputación del proveedor y por supuesto de su costo.

En una configuración On-premise el cliente se encarga de hospedar sus aplicaciones en uno o varios servidores en su propia infraestructura.

Uno de los aspectos clave para ayudar a garantizar la continuidad de un proyecto de software son precisamente sus costos fijos; el cloud los discretiza con el modelo *Pay as You Go* en términos de peticiones, duración, poder de cómputo, espacio de almacenamiento, usuarios, y demás variables que le agregan aún mas complejidad a la decisión de escoger el proveedor adecuado para los recursos requeridos por el proyecto.

El modelo comercial Pay as You Go permite al cliente pagar solo por lo que ha usado después de haberlo usado.

PriceCloud permitirá interactuar sobre una interfaz web donde el usuario podrá tomar una decisión informada conociendo los costos de su configuración de recursos sobre varios CCSP y de manera complementaria gracias a los controladores que ofrece LibCloud podrá ser asistido en su aprovisionamiento.

2

PLANTEAMIENTO DEL PROBLEMA

Según lo mencionado anteriormente esta propuesta de trabajo de grado busca responder las siguientes preguntas. ¿Cómo dar un criterio informado respecto a los costos de aprovisionamiento de una aplicación facilitando su despliegue?, ¿Cómo construir un servicio que brinde información actualizada de los costos de los proveedores?, ¿Cómo proponer un proyecto colaborativo que sea extensible a nuevos proveedores y servicios?, ¿Qué tecnologías y arquitectura usar para procurar su ampliación?.

3

JUSTIFICACIÓN

La propuesta y desarrollo de este proyecto sugieren conocimientos de Desarrollo de Software, Bases de Datos Relacionales, Aplicaciones Web, Microservicios, Web Scraping, Crawling, Seguridad Web, Web Services, Redes de Computadores y Sistemas Operativos.

3.1 JUSTIFICACIÓN ECONÓMICA

Actualmente no existe una tecnología *Open Souce* que permita administrar recursos en CCSPs teniendo en cuenta los costos de servicios; gracias a que PriceCloud será una proyecto de codigo abierto podrá implementarse de manera gratuita por cualquier desarrollador para dar un punto de apoyo en este proceso que carece de un marco bien definido.

3.2 JUSTIFICACIÓN ACADÉMICA

Diseñar e implementar esta propuesta pone a prueba las destresas técnicas y formación del aspirante a grado; Este proyecto de código libre permitirá tener un punto de partida para que mas interesados puedan contribuir en la solución del problema que parece tener carencia de participación.

4

OBJETIVOS

El lenguaje XPath es un recurso que permite describir patrones para hallar coincidencias en los Endpoints de un sitio Web.

La recuperación de información en la web inicia con el Crawling que permite obtener las páginas web de objetivo para luego aplicar Web Scraping que recopila información de interés.

4.1 OBJETIVO GENERAL

Diseñar, e implementar un prototipo de aplicación web para informar las tarifas de aprovisionamiento de recursos tipo *Compute*, *Storage* y *Container* en el CC y administrar su despliegue.

4.2 OBJETIVOS ESPECÍFICOS

1. Definir un modelo general de evaluación de costos para el análisis de los servicios en el CC usando información recuperable y relevante de la Web de sus proveedores.
2. Diseñar e implementar un prototipo basado en microservicios que recopile periodicamente y almacene las tarifas de los principales CCSP usando técnicas de *Web Scraping* y *Crawling* con XPath, bases de datos relacionales y una API REST.
3. Diseñar e implementar una aplicación web basada en microservicios para acceder a la información de costos de los proveedores y administrar los recursos de un usuario usando la API de LibCloud.

5

RESULTADOS ESPERADOS

Cuando se culmine el proyecto propuesto en este escrito se espera cumplir los objetivos específicos mencionados anteriormente como se describe a continuación.

1. Este modelo permite tener un punto de referencia para medir los proveedores equitativamente teniendo en cuenta la mayor información que se pueda recuperar de manera pública, se espera poder incluir en todos los casos los costos para cada tipo de servicio, e información adicional como el tiempo en linea y la proximidad geográfica de cada provedor que permitan tener un criterio mas amplio para tomar una decisión. Este modelo se incorporará al servicio web principal del proyecto.

2. Esta parte del proyecto permite manejar de manera centralizada, actualizada y con una trazabilidad la información relevante de los CCSP, se pretende construir un servicio en linea que funcione sin interrupciones pues su responsabilidad es responder a las consultas hechas por las instancias de la interfaz web.

3. Esta parte de la aplicación puede ser desplegada a discreción de cualquier usuario sobre su propia infraestructura para poder usar la solución propuesta en una interfaz Web, se pretende que esté disponible en una imagen pública en Docker Hub con su respectiva documentación y parámetros de configuración usando Docker Compose.

El servicio principal de la aplicación también podrá ser desplegado en un stack de contenedores dispuesto en Docker Hub solo si el usuario está interesado en recuperar por sus propios médios la información de los proveedores.

La comunicación de todos los web services estará encriptada usando el protocolo HTTPS con certificados libres de Let's Encrypt, los datos sensibles almacenados usarán el algoritmo de encriptación simétrica AES.

6

ALCANCES DE LA PROPUESTA

El archivo `robots.txt` ubicado en la raiz de un sitio web indica a los Bots que información es de caracter privado o no debe ser indexada.

PriceCloud pretende recuperar de la Web información de los principales CCSPs entre los que están AWS, Microsoft Azure, Google Cloud, IBM Cloud y Digital Ocean. PriceCloud también tiene en cuenta proveedores locales (en Colombia) como Host Dime, Hosting RED; Se espera que el registro de nuevos proveedores se haga por cualquier usuario y se valide por el administrador de la aplicación.

PriceCloud tiene soporte básico de admistración para los recursos de tipo *Compute*, *Storage* y *Container* solamente para el proveedor AWS pero con la posibilidad de extender el proyecto a nuevos tipos de servicios teniendo en cuenta las disponibilidad de controladores[1] de la API de LibCloud.

La implementación de la interfaz web se construye en React a partir de la librería CoreUI[2], es la razón por la que esta propuesta no considera el diseño UI y UX como un componente principal.

La información que se pretende recuperar de la Web de los CCSPs tiene en cuenta las limitaciones técnicas y legales que los mismos proveedores incluyen en sus politicas y en el archivo `robots.txt` para evitar incurrir en problemas legales.

La implementación de la aplicación con todos sus componentes tiene como objetivo mostrar la aplcación en funcionamiento, sin embargo los aspectos técnicos adicionales referentes a su despliegue como pruebas de rendimiento, pentesting y mantenimiento no se tienen en cuenta en esta propuesta.

1 https://libcloud.readthedocs.io/en/stable/supported_providers.html

2 https://github.com/coreui/coreui-react/

7

MARCO REFERENCIAL

En este apartado se intenta recopillar la información para describir el CC y los modelos de precios que usan los CCSPs para cobrar sus servicios.

7.1 ESTADO DEL ARTE

Actualmente existen recursos de software para la orquestación de recursos en la nube, estos son bien conocidos como Cloud Resource Orchestration Frameworks (CROFs), han surgido como sistemas para gestionar el ciclo de vida de los recursos de multiples CCSPs y que cada vez exigen mecanismos de orquestación de recursos capaces de tratar con la heterogeneidad subyacente. [8].

7.1.1 *Tecnologías de orquestación*

7.1.1.1 *LibCloud*

Es una librería escrita en Python tiene licencia *Apache 2.0*. Es una de las mas completas para administrar recursos de CC teniendo unicamente carencias en el soporte de DBaaSs [1]. Esta librería oculta las diferencias entre las API de los CCSPs y permite administrar diferentes recursos de la nube a través de una API unificada y facil de usar.

Esta librería divide sus funciones en seis principales categorías[2]. La primera permite administrar Servidores en la nube y *Block Storage*, este componente permite ejecutar secuencias de comandos para preparar al servidor recién creado. Otro tipo de recurso administrable es el almacenamiento de objetos en la nube *Object Storage* y la administración de CDNs, Los servicios de balanceadores de carga *Load Balancer*, la API para administración de DNSs como servicio y los servicios de administración de Contenedores que permiten a los usuarios implementar y

1 https://medium.com/@anthonypjshaw
2 https://libcloud.readthedocs.io/en/stable/index.html

administrar contenedores usando software como Docker con los proveedores que ofrecen una API de CaaS.

7.1.1.2 *pkgCloud*

Es una librería escrita en *Javascript* para usar sobre *NodeJS*, al igual *LibCloud* esta permite abstraer los diferentes mecanismos para acceder a las diferentes APIs de los CCSPs para centralizar su administración, de manera adicional es la única librería con soporte de DBaaS, permite integración con *Rackspace* usando *MySQL* y Microsoft Azure con el servicio de *Azure Tables*, esta librería tiene una licencia tipo *MIT License* y además es un proyecto de colaboración disponible en *GitHub*[3].

7.1.1.3 *Cloudify*

Es un Framework que usa la arquitectura *Multicloud* y el concepto de Environment as a Service (EaaS) usando *Blueprints*. Los recursos y cargas de trabajo pueden ser orquestados y migrados en tiempo real entre diferentes CCSPs, su componente principal *Cloudify Manager* permite administrar los recursos via interfaz web *Cloudify Console*, este Framework esta licenciado bajo *Apache 2.0 License*, es uno de los CROFs mas completos gracias a su diseño lógico de Plug-ins [4].

7.1.1.4 *Terraform*

Es una herramienta de Infrastructure as a Codes (IaaCs) que aprovisiona y administra infraestructuras usando un lenguaje de configuración de alto nivel, *Terraform* puede administrar múltiples proveedores de nube e incluso dependencias entre CCSPs, *Terraform* aprovecha los CCSPs para proporcionar capacidades de escalado automático con disparadores de umbral en las métricas del sistema recopiladas por los servicios de monitoreo [8, p.14].

3 https://github.com/pkgcloud/pkgcloud
4 https://github.com/orgs/cloudify-cosmo/repositories

7.1.2 *Costos de servicio*

7.1.2.1 *Amazon Web Services*

Este CCSP tiene una página web en la Figura 1 dedicada para el cálculo de los costos de sus servicios [5], la página permite la estimación de los recursos seleccionando uno o varios servicios y estableciendo su configuración para después exportarla en un Portable Document Format (PDF)

Figura 1: Pagina de precios de AWS

7.1.2.2 *Google Cloud*

También conocido como Google Cloud Platform (GCP) este CCSP tiene un sitio web[6] con una calculadora de estimación en la Figura 2, también es posible acceder los precios usando una lista de precios dinámica para cada tipo de servicio mostrando su costo de operación por hora e incluso con algunas ofertas gratuitas para la capa mínima de servicios[7].

7.1.2.3 *Microsoft Azure*

Al igual que los demas CCSP Microsoft Azure tiene una calculadora y lista dinámica de precios [8] que puede ser consultada

5 https://calculator.aws
6 https://cloud.google.com/pricing
7 https://cloud.google.com/pricing/list
8 https://azure.microsoft.com/es-es/pricing/

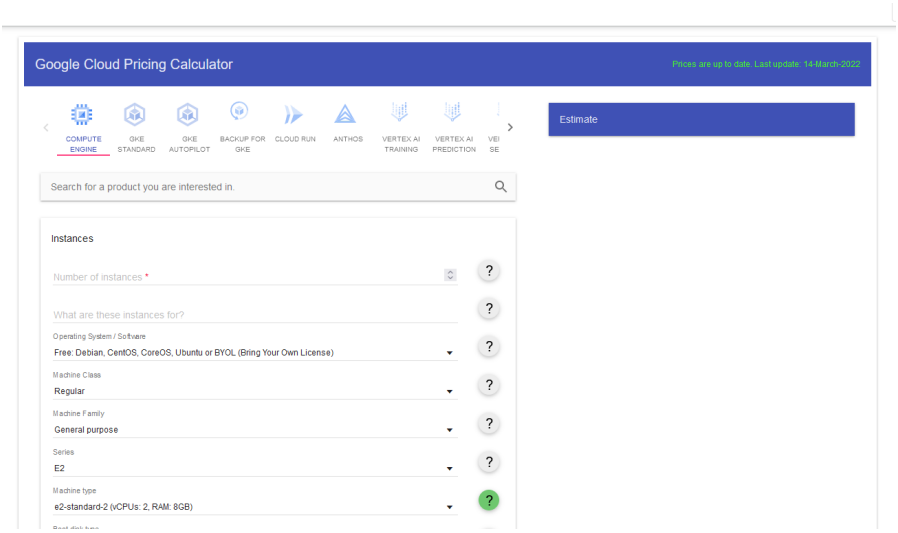


Figura 2: Pagina de precios de GCP

en términos de los servicios con el modelos de cobro por horas con se ve en la Figura 3.

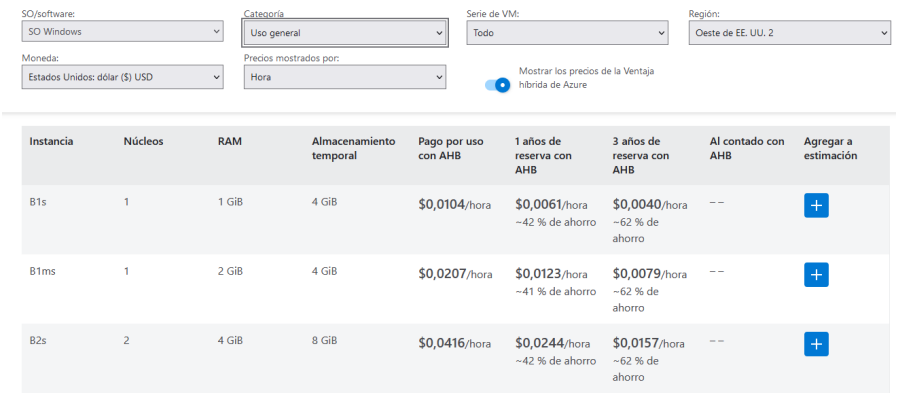
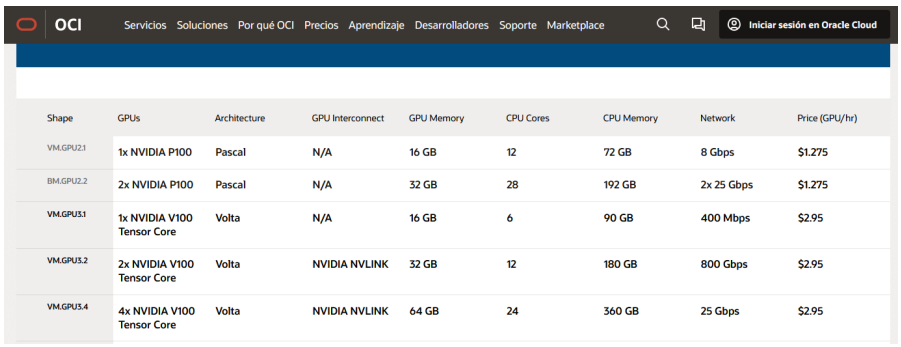


Figura 3: Lista de precios de Microsoft Azure para VMs con OS *Windows*

7.1.2.4 *Oracle Cloud Infrastructure*

Orace tiene un estimador de costos y lista de precios en su sitio web.[9] como se muestra en la Figura 4.

9 https://www.oracle.com/co/cloud/price-list.html



Shape	GPUs	Architecture	GPU Interconnect	GPU Memory	CPU Cores	CPU Memory	Network	Price (GPU/hr)
VM.GPU2.1	1x NVIDIA P100	Pascal	N/A	16 GB	12	72 GB	8 Gbps	$1.275
BM.GPU2.2	2x NVIDIA P100	Pascal	N/A	32 GB	28	192 GB	2x 25 Gbps	$1.275
VM.GPU3.1	1x NVIDIA V100 Tensor Core	Volta	N/A	16 GB	6	90 GB	400 Mbps	$2.95
VM.GPU3.2	2x NVIDIA V100 Tensor Core	Volta	NVIDIA NVLINK	32 GB	12	180 GB	800 Gbps	$2.95
VM.GPU3.4	4x NVIDIA V100 Tensor Core	Volta	NVIDIA NVLINK	64 GB	24	360 GB	25 Gbps	$2.95

Figura 4: Muestra de precios para instancias de GPU de *Oracle Cloud*

7.2 MARCO TEÓRICO

7.2.1 *Cloud Computing*

Según el NIST la computación en la nube está definida como un modelo que permite el acceso de red conveniente y bajo demanda a un grupo compartido de recursos informáticos configurables como redes, servidores, almacenamiento, aplicaciones y servicios que pueden aprovisionarse y librerarse rápidamente con un mínimo esfuerzo de gestión o interacción con el CCSP[4].

7.2.1.1 *Modelos de despliegue*

En el CC existen varios modelos de despligue dependiendo de cual es el alcance de los recursos. La *Nube privada* donde los datos y procesos son administrados dentro de la organización quien también se encarga de dar mantenimiento a la infraestructura que recide en las mismas instalaciones o en un lugar remoto administardo por la organización. En la *Nube pública* los recursos son aprovisionados dinamicamente a través de Internet, el CCSP le permite acceder a los todos los tipos de servicios bajo esquemas de pago por uso. En una *Nube híbrida* los recursos o servicios de la organización pueden ser complementarios entre ambas ubicaciones e incluso pueden incluir infraestructura de *Edge Computing* como dispositivos de IoT [2, p.2].

7.2.1.2 *Tipos de servicios*

Los servicios servicios de CC ofrecen mas beneficios que la computación tradicional, ahorro de costos, escalabilidad, almacenamiento movil, acceso desde cualquier momento o cual-

quier lugar, mejor seguridad, ahorro de energía[6]. En terminos de taxonomía el CC está divido en tres principales tipos de servicios, Infrastructure as a Service (IaaS) que puede verse como una evolución de los Virtual Private Servers (VPSs) la cual ofrece plataformas de virtualización donde los clientes deben configurar sus propio software, estos servicios son facturados con base en los recursos consumidos, el modelo de pago por alquiler permite usar hasta mínimo una hora en algunos casos donde la cantidad de instancias se duplican para satisfacer las necesidades los clientes. Entre los tipos de soluciones que se encuentran en esta categoría están Amazon Elasctic Compute Cloud (EC2), *Rackspace Cloud*, *IBM Smart Bussiness Cloud solutions*, *Oracle Cloud Computing*, *Google Compute Engine*. [6, p.2]

Otro esquema de servicios ofrecidos por el CC son las Platform as a Services (PaaSs), como características principales tenemos que son proveidas a los clientes por una interfaz en un navegador para editar, depurar, desplegar y monitorear una aplicación.[3, p.1] A diferencia de las IaaSs permiten una interacción de alto nivel con el cliente porque abstraen la configuración del entorno donde se ejecutan las aplicaciones, ademas de permitir la preconfiguración de un entorno de ejecución para un lenguaje de programación específico. Entre los CCSPs mas conocidos se encuentra también EC2 con la posibilidad de preconfigurar los entornos, *Salesforce.com* y *Google App Engine*.

El siguiente nivel en terminos de abstracción son las Function as a Services (FaaSs) conocidas también como computación *Serverless* [5, p.1] que emerge como un paradigma conveniente para el despliegue de aplicaciones y servicios donde el desarrollador no se preocupa acerca de los aspectos operacionales, de despliegue, y mantenimiento y espera de este que sea tolerante a fallos y auto escalable especialmente el código que es escalable a cero donde no hay servidores corriendo cuando la función del usuario no está siendo usada. FaaS a diferencia de Platform as a Services (PaaSs) evita que el cliente reciba cobros en los periodos de inactividad de la función.[1, p.5] Entre los servicios mas conocidos están *Azure Functions*, AWS *Lambda*, *Google Cloud Functions* y *Oracle Cloud Functions*.

Por último y mas cerca del usuario se encuentran el Software as a Service (SaaS), en este modelo de servicios el cliente interactúa con una aplicación que da soporte a los procesos de su modelo de negocío o le ofrece un servicio, el cliente paga recurrentemente por las funcionalidades y la cantidad de usuarios

que usan el servicio, el SaaS presenta ventajas frente al software tradicional, el cliente compra una suscripción y no una licencia perpetua, además el software tradicional tarda algunos años en publicar una nueva versión que el SaaS puede poner a disposición en cuanto esté completa [2, p.1]. Algunos ejemplos de SaaSs son *Google Workspace*, *Microsoft 365*.

7.2.2 *Modelos de precios*

De manera general existen dos grupos de modelos de precios, los modelos de *precios dinámicos* donde el costo es flexible y depende de factores como el número de peticiones, el espacio usado, el tráfico generado, el poder de cómputo requerido o la cantidad de instancias necesarias. En los modelos de *precios fijos* se dice que el cliente tiene mayor seguridad sobre el costo del servicios porque tiene un cargo fijo periodicamente para pagar, sin embargo el beneficio nunca es real porque depende del uso que tenga el servicio contratado donde algunas veces puede llegar al sobreuso.[7, p.4].

Los CCSPs usan modelos de precios mixtos que incluyen el esquema de pago por suscripción y el pago por uso, *Pay-as-you-go* que usan las IaaSs y las PaaSs.

Existen otros modelos como el *Modelo de pago por suscripción* donde los precios se establecen de acuerdo con el nivel de la suscripción, son comunmente usados por los provedores de servicios de Hosting. El *modelo de pago basado en costos* permite al CCSP calcular los precios dinámicos dependiendo de la demanda y las operaciones del los centros de datos del CCSP. Entre los demás modelos de precios dinámicos están los *Modelos genéticos de precios*, *Modelos de precios basados en la competencia*, *Modelos de precios basados en el cliente*, *Modelos de precios con subasta dinámica* y los *Modelos de precios híbridos* que establecen el precio con la convinación de los mencionados anteriormente.

8

METODOLOGÍA

8.1 ACTIVIDADES A REALIZAR

8.2 CRONOGRAMA DE ACTIVIDADES

9 PRESUPUESTO

GLOSARIO

Amazon Web Services Es una empresa lider mundial en servicios de CC. 9, 18

Bootstrap Es una biblioteca de estilos para aplicaciones web definidos principalmente en lenguaje SCSS. 16

Bot Es un programa de computador o script que realiza tareas periódicas de recopilación de datos generalmente sobre Internet y la web. 6

Cloud Computing Provider Empresa que presta servicios de Computación en la nube. ii, 18

Cloud Computing El CC es una tendencia de servicios y recursos que funcionan sobre Internet para facilitar el despliegue, uso y disponibilidad de aplicaciones de Software. ii, 1, 18

CoreUI Es una biblioteca de React que ofrece una versión de uso libre de componentes gráficos escritos en Javascript y TypeScript y apoyada en en los estilos de Bootstrap. 6

Digital Ocean Es un proveedor de servicios de CC que proviene de New York fundada en el año 2011. 6

Docker Es una plataforma de contenerización para la implementación microservicios, abstrae recursos como redes, volumenes de almacenamiento, y nodos de cómputo. 8, 16

Docker Compose Es una tecnología para la automatización y orquestación de aplicaciones a partir de contenedores de Docker. 5

Docker Hub Es el nombre del sitio web que contiene las librerías de imágenes de contenedores de Docker. 5

Endpoint Comunmente las rutas en un servicio web que responden a las peticiones que le son solicitadas. ii, 4

Framework Conjunto de tecnologías que trabajan juntas para darle desarrollo a un proyecto que tiene objectivos específicos. 8

Google Cloud Es una empresa de CC fundada por Google en el 2008. 6

Host Dime Esta empresa tiene presencia de infraestructura de servicios de CC en Colombia. 6

Hosting Es un servicio relacionado generalmente con el alojamiento de un sitio web en una empresa que presta el servicio. 13

Hosting RED Es una empresa de CC con centros de datos ubicados en Bogotá Colombia además de Estados Unidos y Canadá. 6

Hypertext Media Transfer Protocol Secure El protocolo que usa la web para enviar y recibir tráfico encriptado a través de una capa de sockets seguros. 18

IBM Cloud Es el servicio de CC prestado por la empresa IBM. 6

Let's Encrypt Es una entidad certificadora gratuita de lucro para expedición de cetificados SSL/TLS. 5

LibCloud Es una de las bibliotecas escrita en lenguaje Python mas completa hasta el momento para interactuar con los servicios de los CCSPs. ii, 1, 4, 6

Microsoft Azure Es una empresa de Microsoft que ofrece servicios de CC. 6, 8, 9, 10

Plug-in Son complementos a las funcionalidades de base que tiene una aplicación de software, pueden ser incorporados por el usuario a discreción. 8

React Es una biblioteca de Javascript para la construcción de aplicaciones web reactivas. 6

Representational State Transfer API Es una interfaz de programación de aplicaciones que permite la comunicación a través de la web usando el protocolo HTTP. 18

User Interface Referente al diseño que tienen las interfaces de un proyecto de software o tecnología. 18

XPath Es un lenguaje de programación para formular patrones que permiten encontrar coincidencias en documentos de marcado de etiquetas. 4

SIGLAS

AES Advanced Encryption Standard. 5
API Application Programing Interface. ii, 4, 7, 8
API REST Representational State Transfer API. 4
AWS Amazon Web Services. 6, 9, 12

CaaS Container as a Service. 8
CC Cloud Computing. ii, 1, 4, 7, 11, 12, 16, 17
CCSP Cloud Computing Provider. ii, 1, 3, 4, 5, 6, 7, 8, 9, 11, 12, 13, 17
CDN Content Delivery Network. 7
CROF Cloud Resource Orchestration Framework. 7, 8

DBaaS Data Base as a Service. 7, 8
DNS Domain Name Server. 7

EaaS Environment as a Service. 8
EC2 Amazon Elasctic Compute Cloud. 12

FaaS Function as a Service. 12

GCP Google Cloud Platform. 9, 10
GPU Graphic Processor Unit. 11

HTTPS Hypertext Media Transfer Protocol Secure. 5

IaaC Infrastructure as a Code. 8
IaaS Infrastructure as a Service. 12, 13
IoT Internet of Things. 11

NIST National Institute of Standards and Technology. 11

OS Operative System. 10

PaaS Platform as a Service. 12, 13
PDF Portable Document Format. 9

SaaS Software as a Service. 12, 13

UI User Interface. 6
UX User Experience. 6

VM Virtual Machine. 10
VPS Virtual Private Server. 12

BIBLIOGRAFÍA

[1] Ioana Baldini y col. «Serverless Computing: Current Trends and Open Problems». En: *Research Advances in Cloud Computing*. Ed. por Sanjay Chaudhary, Gaurav Somani y Rajkumar Buyya. Singapore: Springer Singapore, 2017, págs. 1-20. ISBN: 978-981-10-5026-8. DOI: 10.1007/978-981-10-5026-8_1. URL: https://doi.org/10.1007/978-981-10-5026-8_1.

[2] Vidyanand Choudhary. «Software as a service: Implications for investment in software development». En: *2007 40th Annual Hawaii International Conference on System Sciences (HICSS'07)*. IEEE. 2007, 209a-209a.

[3] George Lawton. «Developing software online with platform-as-a-service technology». En: *Computer* 41.6 (2008), págs. 13-15.

[4] Fang Liu, Jin Tong, Jian Mao, Robert Bohn, John Messina, Lee Badger, Dawn Leaf y col. «NIST cloud computing reference architecture». En: *NIST special publication* 500.2011 (2011), págs. 1-28.

[5] Theo Lynn, Pierangelo Rosati, Arnaud Lejeune y Vincent Emeakaroha. «A preliminary review of enterprise serverless cloud computing (function-as-a-service) platforms». En: *2017 IEEE International Conference on Cloud Computing Technology and Science (CloudCom)*. IEEE. 2017, págs. 162-169.

[6] Ayob Sether. «Cloud computing benefits». En: *Available at SSRN 2781593* (2016).

[7] Aishwarya Soni y Muzammil Hasan. «Pricing schemes in cloud computing: a review». En: *International Journal of Advanced Computer Research* 7.29 (2017), pág. 60.

[8] Orazio Tomarchio, Domenico Calcaterra y Giuseppe Di Modica. «Cloud resource orchestration in the multi-cloud landscape: a systematic review of existing frameworks». En: *Journal of Cloud Computing* 9.1 (2020), págs. 1-24.