

UNIVERSIDAD MICHOCANA DE SAN NICOLAS DE  
HIDALGO.

FACULTAD DE INGENIERÍA ELÉCTRICA

**EL LENGUAJE DE PROGRAMACIÓN GO – UNA  
COMPARACIÓN CON PHP**

**TESIS**  
QUE PARA OBTENER EL TÍTULO DE:  
**INGENIERIA EN COMPUTACIÓN**

PRESENTA:  
**VICTOR OCTAVIO RAZO VERDUZCO**

ASESOR:  
**DR. JUAN JOSÉ FLORES ROMERO**

MORELIA, MICHOCÁN

**NOVIEMBRE DEL 2015**



## DEDICATORIA

*Dedico esta tesis a mi madre, que ha sabido formarme con buenos sentimientos, hábitos y valores en los momentos más difíciles, y por lo cual me a hecho la persona que soy en estos momentos.*

*A mi padre y mi abuela, que se nos adelantaron en el camino, y que desde allá están y estarán apoyándome incansablemente para lograr mis sueños y metas.*

*A mi hermano, que siempre ha estado junto a mí, brindándome su apoyo, y muchas veces poniéndose en el papel de padre.*

*A toda mi familia en general por brindarme el cariño y contagiarme las ganas por salir adelante.*

*Y finalmente a mis amigos que son la familia que elegí por creer en mí y animarme en momentos de desesperación.*

## **AGRADECIMIENTOS**

*A la Universidad Michoacana de San Nicolás de Hidalgo por a haber sido mi casa durante este lapso de tiempo.*

*A la Facultad de Ingeniería Eléctrica, a todos mis maestros y guías en mi trayecto de aprendiz, en especial a mi asesor el Dr. Juan José Flores Romero por brindarme su tiempo, amistad y gran apoyo en el desarrollo de esta investigación.*

## RESUMEN

En la actualidad la tecnología crece a pasos gigantes; una vertiente en este crecimiento es el desarrollo que ha tenido en estos últimos años el internet. Constantemente surgen nuevas y mejores tecnologías; con esta competencia entre el lanzamiento de nuevos productos y nuevas versiones, es imposible estar al día y saber sobre todo lo relacionado a la programación.

Existen tantos lenguajes nuevos, que suena apasionante navegar por todos esos caminos distintos a los que conocemos comúnmente. La innovación de muchas nuevas empresas en el mundo usando lenguajes nuevos, así con *Frameworks*, cada vez va aumentando dado a que las necesidades de los proyectos o ideas que se van haciendo de mayor tamaño, necesidad y ambición.

Motivado por esto, se desarrolla esta investigación destinada a comparar dos lenguajes de programación. El lenguaje de programación Go, uno de los nuevos lenguajes para el desarrollo que se describe como un lenguaje innovador, contra el lenguaje PHP, que está cimentado con años de experiencia y abundante información.

Tomando como reto recopilar información diversa de los dos lenguajes y haciendo una comparación entre ellos; partiendo de la información disponible se desglosa un breve manual de Go, se recopilan y se exponen las estadísticas de ambos lenguajes en la red, el número de usuarios, foros, publicaciones y códigos *OpenSource*. Posteriormente se hace un análisis respecto a características técnicas entre ambos lenguajes. Anexando por último un caso práctico de los dos lenguajes.

El resultado muestra que el lenguaje de programación Go es superior al lenguaje de programación PHP tanto en aspectos técnicos y en uso para proyectos a escala.

## **ABSTRACT**

Nowadays, technology industry grows at fast pace. Internet is one of the best examples about it. Constantly new and improved technologies are developed. With such an accelerated growth it is near to impossible to keep up to date with every field of programming.

There is a huge amount of programming languages that sounds amazing try to explore all learning paths. Innovation is one of the key factors for success in competitive enterprises. Is not rare to see firms using new languages and frameworks that started as ideas and grown up to be key players in the company success.

Using the aforementioned motivation, this investigation presents a comparison between two mainstream programming languages: the GO programming language, described as an innovation in programming languages versus PHP programming language that has been in the industry for long time and has years built over years of experience and has great documentation.

The present work collects information about both languages and does a comparison based on statistics, number of users, forums, releases and open source codes. Afterwards, a technical analysis is made based on technical features and capabilities. Finally, a use case is provided for each language.

The result shows that GO programming language is more robust in technical features and more suitable for scale projects.

## **PALABRAS CLAVE**

*Computación, lenguajes de programación, Go, Golang, Google, PHP, programación, open source, lenguajes de programación, hardware, software, framework, sistemas operativos, benchmark, HTML, web, desarrollo web, cloud, data center, algoritmos de ordenamiento.*

## TABLA DE CONTENIDO

<b>CAPÍTULO 1 - INTRODUCCIÓN</b> .....	<b>1</b>
<b>1.1. Objetivo</b> .....	<b>3</b>
<b>1.2. Justificación</b> .....	<b>3</b>
<b>1.3. Organización del documento</b> .....	<b>5</b>
<b>CAPÍTULO 2 - LENGUAJES DE PROGRAMACIÓN Y SU HISTORIA</b> .....	<b>7</b>
<b>2.1 Historia y antecedentes</b> .....	<b>8</b>
<b>2.2 No son lenguajes de programación</b> .....	<b>10</b>
<b>2.3 La nube</b> .....	<b>10</b>
2.3.1 Modelos de la nube .....	11
2.3.2 Programación en la nube.....	11
<b>CAPÍTULO 3 - GO</b> .....	<b>13</b>
<b>3.1 Características del lenguaje</b> .....	<b>14</b>
<b>3.2 Manual de introducción a Go</b> .....	<b>18</b>
3.2.1 Instalación.....	18
3.2.2 Compilación .....	21
<b>3.3 Su sintaxis</b> .....	<b>22</b>
3.3.1 Tipos de errores.....	23
3.3.2. Comentarios.....	23
3.3.3. Asignaciones.....	23
3.3.4. Variables .....	24
3.3.4.1. iota y apex .....	28
3.3.4.2. Cero o nulo .....	28
3.3.4.3 Tipos de datos compuestos.....	29
3.3.4.3.1 Arreglos .....	29
3.3.4.3.2. Slices .....	31
3.3.4.3.3. Maps .....	34
3.3.4.3.4. Structs.....	36
3.3.5. Estructuras de control .....	37
3.3.5.1. Switch .....	40
3.3.6. Funciones.....	41
3.3.7. Algunas funciones y palabras reservadas .....	43
3.3.7.1. new ( ) y make().....	43
3.3.7.2. Defer.....	45
3.3.7.3. Funciones lambda y closures .....	46
<b>3.4 Programación orientada a objetos</b> .....	<b>49</b>



<b>CAPÍTULO 4 - ESTADÍSTICAS Y COMPARACIÓN .....</b>	<b>61</b>
<b>4.2. Estadísticas de actividad en internet.....</b>	<b>62</b>
4.2.1. Comportamiento y estadísticas de ambos lenguajes en la red .....	62
4.2.1.1 Estadísticas en español.....	62
4.2.1.2 Estadísticas generales .....	68
4.2.1.3 Desarrollo de proyectos.....	79
4.2.1.4 Comparación de GO vs PHP.....	83
4.2.1.5 Desempeño de Go vs PHP .....	88
4.2.1.6. Códigos incluidos en la comparación .....	92
<b>CAPÍTULO 5 - CONCLUSIONES.....</b>	<b>100</b>
<b>BIBLIOGRAFÍA .....</b>	<b>103</b>

## INDICE DE TABLAS

TABLA 1 DEFINICIONES DE POO EN GO .....	60
TABLA 2 ESTADISTICAS GENERALES EN ESPAÑOL.....	66
TABLA 3 ESTADISTICAS GENERALES DE LOS GRUPOS DE GO .....	70
TABLA 4 ESTADISTICAS GENERALES DE LOS 12 GRUPOS MAS IMPORTANTES DE PHP .....	71
TABLA 5 ESTADISTICAS GENERALES.....	80
TABLA 6 PROYECTOS EN GO .....	81
TABLA 8 COMPARACION GENERAL GO VS PHP .....	86,87
TABLA 9 BENCHMARK GO VS PHP.....	89
TABLA 4 SEUNDO BRENCHMARK GO VS PHP.....	92

## TABLA DE ILUSTRACIONES

FIGURA 1 ANTECESORES DE GO .....	9
FIGURA 2 ESPECIFICACIONES DE GOLANG .....	20
FIGURA 3 ESTRUCTURA DE DATOS EN GO.....	25
FIGURA 4 CREAR UN SLICE DESDE UN ARREGLO .....	32
FIGURA 5 PARTES DE UN SLICE .....	33
FIGURA 6 HERENCIA DE CLASES.....	55
FIGURA 7 COMPOSICIÓN DE OBJETOS.....	56
FIGURA 8 PORCENTAJE DE CONTINENTES INTERESADOS EN GO, ESPAÑOL .....	63
FIGURA 9 PORCENTAJE DE PAISES INTERESADOS EN GO, ESPAÑOL .....	63
FIGURA 10 PORCENTAJE DE CONTIENTES INTERESADOS EN PHP, ESPAÑOL.....	64
FIGURA 11 PORCENTAJE DE PAISES INTERESADOS EN PHP, ESPAÑOL.....	64
FIGURA 12 USUARIIS EN ESPAÑOL POR MES.....	64
FIGURA 13 PUBLICACIONES EN ESPAÑOL POR MES .....	65
FIGURA 14 DOMINIOS EN ESPAÑOL POR MES.....	66
FIGURA 15 PORCENTAJE DE INTERÉS DE GO EN CONTINENTES .....	69
FIGURA 16 PORCENTAJE DE INTERÉS DE GO EN PAISES .....	69
FIGURA 17 PORCENTAJE DE INTERÉS DE PHP EN CONTINENTES .....	69
FIGURA 18 PORCENTAJE DE INTERÉS DE PHP EN PAISES .....	69
FIGURA 19 PORCENTAJE DE PUBLICACIONES EN LOS GRUPOS DE GO .....	71
FIGURA 20 PORCENTAJE DE USUARIOS EN LOS GRUPOS DE GO.....	71
FIGURA 21 PORCENTAJE DE PUBLICACIONES EN LOS GRUPOS DE PHP .....	72
FIGURA 22 PORCENTAJE DE USUARIOS EN LOS GRUPOS DE PHP .....	72
FIGURA 23 USUARIOS GENERALES EN AMBOS LENGUAJES .....	73
FIGURA 24 USUARIOS TOTALES EN PHP .....	74
FIGURA 25 PUBLICACIONES GENERALES.....	75
FIGURA 26 TOTAL DE PUBLICACIONES EN PHP .....	76
FIGURA 27 DOMINIOS CON Go.....	77
FIGURA 28 DOMINIOS CON PHP .....	78

# CAPÍTULO 1

## INTRODUCCIÓN

En la actualidad la tecnología crece a pasos gigantes; una vertiente a este crecimiento es el desarrollo que ha tenido en estos últimos años el internet. Según las últimas investigaciones, más de mil millones de páginas web están publicadas en miles de servidores en todo el mundo. Constantemente surgen nuevas y mejores tecnologías; con esta competencia entre el lanzamiento de nuevos productos y nuevas versiones, es imposible estar al día y saber todo lo relacionado a la programación [1].

En este ambiente de crecimiento se hace impredecible comparar con exactitud lenguajes o herramientas, ya que los documentos existentes se limitan a describir elementos como hardware usado, arquitectura del hardware, costos en tiempo y demás detalles que tienen peso fundamental en una comparación exacta.

Go fue basado en ideas del *Sistema Operativo Plan 9*. La mitad de sus creadores son pioneros de la programación y *OpenSource*, creando uno de los lenguajes que se considera un gran avance en el desarrollo de la programación por crear un lenguaje de programación compilado e interpretado a la vez [2]. Go es un lenguaje desarrollado por *Google*<sup>1</sup>.

Go se promociona como un lenguaje fácil y claro de leer, muy rápido de compilar, con comprobaciones estrictas, fuertemente tipado, con recolección de basura, de fácil despliegue y muchas otras características interesantes de otros lenguajes; Go puede ser instalado o simplemente usado desde la nube en su página Web.

---

<sup>1</sup> Uno de los pilares más grandes en desarrollo e innovación tecnológica de la actualidad

Go comenzó en el año 2009, y desde entonces empieza a ser tema de discusión en sitios, foros y grupos alrededor del mundo. Go, cuenta con un manual interactivo propio del lenguaje, llamado Tour por Go [3]. En las referencias de esta tesis se encuentran algunos de los pocos libros que tratan sobre el lenguaje Go. En los últimos meses este lenguaje volvió a retomar camino con la aparición de más artículos y notas en varios sitios donde dan introducciones interesantes y comentarios personales referentes a este lenguaje.

Por otro lado, PHP, uno de los pioneros en el desarrollo de programación web, basado en Perl y escrito por Rasmus Lerdorf, con la comunidad de usuarios y/o desarrolladores de mayor tamaño en el mundo, tomó un gran auge en los últimos años para el desarrollo de soluciones web *OpenSource*. PHP, combinado con el gestor de bases de datos *MySQL* permite la construcción de una manera eficiente en sitios web dinámicos pequeños. En la actualidad la mayoría de los servicios de alojamiento de web ofrecen la posibilidad de soportar PHP. PHP es un lenguaje muy simple de usar.

El rendimiento de PHP se contempla como muy bueno y eficiente. Las librerías de PHP vienen incluidas para trabajar sin necesidad de importar nada[4]. PHP soporta programación orientada a objetos<sup>2</sup> y una extensa cantidad de manipuladores de bases de datos, sin mencionar los muchos *Frameworks* que existen para este lenguaje.

A lo largo del desarrollo de esta tesis, se encontraron diversas comparaciones entre lenguajes, abundando sobre la comparación de lenguajes y aplicaciones para el desarrollo web (*PHP, javascript, node.js, Python, Rubi*) y sobre otros ambientes de desarrollo (*C, C++, java*, por mencionar algunos). Dado que Go es un lenguaje reciente, aún no se encuentra gran cantidad de información, así como una prueba a grandes rasgos. En el transcurso de esta investigación han surgido varios sitios referentes a Go. Es importante señalar que no se encontró

---

<sup>2</sup> Programación Orientada a Objetos, es un enfoque de la programación que implica organización de objetos y sus comportamiento (métodos) en grupos (clases).

antecedente de algún documento con el mismo fin que esta investigación, pero sí se encontraron múltiples comparaciones del lenguaje de programación Go contra otros.

## **1.1. Objetivo**

Esta tesis tiene como objetivo general exponer y describir a Go como un lenguaje de programación nuevo y compararlo a partir de otro lenguaje ya existente. En esta investigación se prefirió escoger el lenguaje PHP para establecer esta comparación.

En particular se plantearon los siguientes objetivos:

- Recopilar información sobre ambos lenguajes y adaptarla para formar un documento que sirva de base y/o guía para adentrar a este lenguaje.
- Tener información técnica de ambos lenguajes y opiniones diversas para poder hacer una comparación más nutrida.
- Desarrollar un ejemplo práctico para ver la comparación respecto al tiempo de ejecución y desarrollo, en ambos lenguajes.

## **1.2. Justificación**

En la actualidad el aprendizaje de nuevos lenguajes es algo indispensable para cualquier estudiante o profesionalista en áreas de computación. Por otro lado, como se mencionó anteriormente el desarrollo de hardware no crece acorde al desarrollo de nuevos lenguajes. En la actualidad los tipos de lenguajes dinámicos están siendo mucho más usados en estos años.

Existen una serie de problemáticas singulares como:

- *C/C++* no evolucionaron con el panorama de la computación, en más de una década no se ha creado un nuevo lenguaje de programación con una extensa área de uso como *C* o *Java*, por lo cual estaba la clara necesidad de nuevos lenguajes apropiados para necesidades de esta era informática[5].
- Las computadoras son mucho más rápidas, sin embargo no lo es tanto el desarrollo de software. Los contrastes con la potencia de cálculo en los desarrollos que siguen creciendo en tamaños considerados para procesar cantidades más grandes de datos hace que exista el hueco entre el lenguaje usado por el programador y el nuevo hardware, creando la necesidad indirecta de hacer un nuevo lenguaje de bajo nivel, pero equipado con conceptos y paradigmas que se necesitan actualmente.
- Existe actualmente una tendencia creciente al uso de lenguajes de tipado dinámico, como *Python* y *JavaScript*. Estos lenguajes son una gran rama de la programación inclinada al desarrollo de aplicaciones web saliendo de la forma tradicional y creando códigos de manera más fácil, rápida y en muchas ocasiones más eficiente. A la vez, existe una creciente rebelión contra los lenguajes de programación complejos como *Java* o *C++*.
- Ni la recolección de basura, ni la computación paralela, se soportan adecuadamente. En unos lenguajes que lo soportan su uso es muy robusto y complicado. En los lenguajes nuevos se tiene contemplado el uso de la recolección de basura con el cual se crea un ambiente más seguro y cómodo para el desarrollador. La computación paralela se contempla como una herramienta más amigable que en los lenguajes anteriormente conocidos.

La presente investigación se justifica en la medida que aporte elementos teórico - metodológicos para la inducción hacia este lenguaje, así como ser una base fundamental para futuros desarrollos.

### **1.3. Organización del documento**

Esta tesis expone material diverso sobre Go, recopilado en un mismo documento que tiene como objetivo dotar a los lectores de los mecanismos necesarios para conocer sobre este lenguaje. Describiendo a Go incluyendo su historia, el marco teórico con un resumido manual, marcando ventajas y características angulares en un ambiente de programación limitada y haciendo mención a temas de desarrollo web. Posterior a esto, se incluye al lector a un análisis de los dos lenguajes mencionados, que contiene una comparación de estadísticas de desarrollo, una comparación técnica y finalmente una comparación personal.

El segundo capítulo hace mención a la historia de los lenguajes de programación, así como una pequeña introducción a temas que se mencionan posteriormente.

En el tercer capítulo se expone la estructura del lenguaje Go, redactando un manual recopilado el cual describe desde su instalación hasta la descripción de una lista de aspectos de sintaxis que se consideraron de importancia para esta investigación.

El cuarto capítulo, es el núcleo de esta investigación, el cual presenta un análisis respecto a los lenguajes mencionados. En este capítulo se muestran las diferentes estadísticas, sus flujos de datos en las redes y las actividades recientes. Así mismo se tiene una comparación técnica en donde se presentan los rasgos distintivos de cada lenguaje, así como los pro y los contra de ambos lenguajes respecto a las opiniones y datos encontrados. Se anexa un benchmark ilustrativo para comparar diversos algoritmos de búsqueda creados en el lenguaje de programación Go contra los mismos algoritmos desarrollados en el lenguaje de



programación PHP, donde se muestran algunas de las diferencias y ventajas entre ambos lenguajes.

En el quinto y último capítulo se presenta una serie de conclusiones personales respecto a este marco de investigación.

## **CAPÍTULO 2**

### **LENGUAJES DE PROGRAMACIÓN Y SU HISTORIA**

Un lenguaje de programación es un idioma formal diseñado para expresar procesos que pueden ser llevados a cabo por las computadoras [41].

Estos lenguajes pueden usarse para crear programas que controlen el comportamiento físico y lógico de una máquina, para expresar algoritmos con precisión o como modo de comunicación humana con los ordenadores [6, 42].

Un lenguaje de programación consiste en un conjunto de órdenes o comandos que describen el proceso deseado. Cada lenguaje tiene sus propias instrucciones y enunciados verbales; los lenguajes de programación no son aplicaciones, sino herramientas que nos permiten construir y adecuar aplicaciones.

Al igual que los lenguajes humanos, tales como el inglés o el español, los lenguajes de programación poseen una estructura y un significado.

Hay una gran cantidad de lenguajes, algunos han evolucionado a lo largo del tiempo y siguen vigentes en el transcurso de muchos años, mientras que otros han sido operados durante periodos largos y ahora no se usan. En general un lenguaje es un método conveniente y sencillo de describir las estructuras de información y las secuencias de acciones necesarias para ejecutar una tarea concreta [7].

## 2.1 Historia y antecedentes

Los lenguajes de programación utilizan conjuntos de caracteres para comunicarse con las computadoras. Estas computadoras solo utilizaban información numérica digital mediante código o alfabeto digital, los primeros programas se escribieron en ese tipo de código, denominado lenguaje máquina[8]. La compleja tarea de programar en código máquina hizo que el alfabeto evolucionara y los lenguajes de programación comenzaron a utilizar códigos o conjuntos de caracteres similares al utilizado en los lenguajes humanos. Así, hoy en día la mayoría de las computadoras trabajan con diferentes tipos de juegos de caracteres. Uno de los más destacados es el *código ASCII* [7].

El inicio de los lenguajes de programación surgió a mediados del siglo XIX, con una idea del inventor inglés y profesor matemático de la *Universidad de Cambridge* Charles Babbage proponiendo muchas teorías en las que se basan las computadoras actualmente, llamándola en ese entonces máquinas analíticas.

La construcción de la primera máquina programadora de la historia ocurrió a mediados del siglo XX. Ada Lovelace<sup>3</sup> junto a Babbage desarrollaron cálculos con tarjetas perforadas, cabe destacar que esta máquina nunca se usó, sin embargo los programadores utilizaron las técnicas diseñadas por Babbage [8].

En el año de 1823 el gobierno Británico buscó a Babbage para crear una máquina que resolviera sumas. Con ayuda del francés Charles Jacquard realizaron las tarjetas de perforación con un telar para agujerar automáticamente las tarjetas hechas de un papel rígido. Nunca se desarrollaron completamente las ideas, ya que la época que se vivía no lo permitió.

---

<sup>3</sup> Matemática y escritora británica conocida por su trabajo sobre la máquina analítica. Se describe como la primera mujer programadora de ordenadores

El primer ordenador (ENIAC, *Electronic Numerical Integrator And Calculator*), se basaba en componentes físicos, es decir, se programaba cambiando directamente el hardware de la máquina. Lo que se hacía era cambiar cables de sitio para conseguir así la programación de la máquina. Posteriormente surgió el lenguaje máquina y así siguieron naciendo los diferentes lenguajes de programación.

Puede considerarse que a partir de ahí partió el árbol genealógico de los lenguajes de programación, derivándose desde un primer lenguaje, llamado el lenguaje máquina; después empezaron a aparecer lenguajes con el propósito de hacer el trabajo del desarrollador más fácil ya que cada vez se realizaron con un mayor número de funciones o especificaciones que crean un ambiente más rápido para crear nuevas aplicaciones [38].

Para crear Go, se tomaron de referencia a varios lenguajes existentes y junto las herramientas e ideas más sobresalientes de cada uno. La *Figura 1* contiene un fragmento del árbol genealógico de los lenguajes de programación. Se puede observar que *C*, *Python*, *Oberon*, *Perl* y *Java* fueron algunos de los lenguajes en los que Go se basó [9].

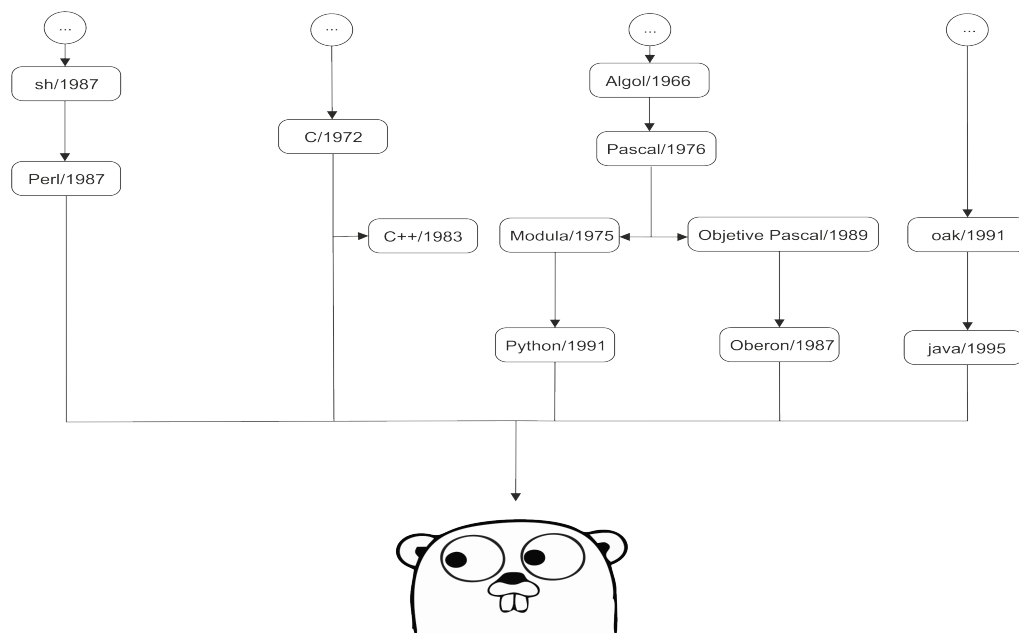


Figura 1 Antecesores de Go

## 2.2 No son lenguajes de programación

Existen elementos muy usados por algunos de nosotros que tienen algunas de las características de los lenguajes de programación, pero que no lo son. Por ejemplo los lenguajes como *XML* y/o *HTML*, en todas sus versiones, solo son lenguajes de etiquetado, ya que no se ejecutan, solo muestran cierta información.

## 2.3 La nube

La nube es un término que define el lugar donde se utiliza un sistema, aplicación o programa cuando se necesita, mientras se necesite y en el momento que lo necesite [10]. La computación en la nube, *cloud computing*, surgió entre 2008 y 2009 es el nuevo paradigma tecnológico que tiene un gran impacto social [39].

Es un conjunto muy grande de servidores de información desplegados en centros de datos, a lo largo de todo el mundo, donde se almacenan millones de aplicaciones web y enormes cantidades de datos a disposición de miles de organizaciones, empresas y de usuarios que descargan y ejecutan directamente los programas y aplicaciones de software almacenados en estos servidores, como los servicios de *Dropbox* y *Facebook* [10].

En la actualidad la nube está propiciando una nueva revolución industrial soportada en los *Data Centers* y centros de aplicaciones web. Esta nueva revolución esta produciendo un gran cambio social, económico y tecnológico, pero al contrario que otras revoluciones será silenciosa, al igual que lo ha sido la implantación de Internet y la web en la sociedad [40].

La nube es una palabra de moda que ha unido tecnologías potentes e innovadoras para construir este nuevo modelo, arquitectura de la web y la programación en ella.

### 2.3.1 Modelos de la nube

El NIST (*National Institute of Standards and Technology*) clasifica los modelos de la computación en nube en dos grandes categorías:

- Modelos de despliegue. Localización y administración de la infraestructura de la nube (Pública, Privada, Comunitaria, Híbrida).
- Modelos de servicio. Servicios específicos a los que se puede acceder en una plataforma de computación en la nube (Software, Plataforma e Infraestructura como Servicios).

El modelo de servicio en el que trabaja Go es:

**Plataforma.** Al usuario se le permite desplegar aplicaciones propias en la infraestructura de la nube de su proveedor, que ofrece la plataforma de desarrollo y las herramientas de programación.

### 2.3.2 Programación en la nube

La nube solo es un adjetivo para marcar el trabajo o almacenamiento que se hace desde servidores remotos, cuya localización es desconocida [39].

El uso de la nube en el lenguaje de programación Go se refiere a que se puede compilar cualquier programa desde una plataforma dedicada para el desarrollo de programas no tan extensos, con un fin enfocado a la parte educativa y/o para que el usuario conozca este lenguaje. El servidor de Go en la nube es <http://play.golang.org> donde puedes ejecutar un sinnúmero de pruebas, con una velocidad de compilación muy comparable a la compilación en una misma computadora local. Cada desarrollo que se haga en este sitio se queda alojado en un espacio de la nube dedicada con un ID aleatorio que se genera.

Mencionar la nube en trabajos de software es sinónimo de mencionar desarrollo escalable de forma sencilla, El lenguaje de programación Go hace también mención a la nube por el hecho de que también todo el código y manuales están almacenados en una nube *Github*, que hace el crecimiento de este lenguaje de una manera más accesible y rápida. En el año 2013 se describió a Go como la lengua emergente de la infraestructura en la nube [11].

En la historia de la computación, los nuevos lenguajes de programación toman como base algunos lenguajes existentes en busca de una versión mejorada, o una nueva fusión entre lenguajes creando uno nuevo. Es importante repasar de manera rápida la historia de los lenguajes de programación y explicar porque Go se dice que es un lenguaje en la nube.

Los requerimientos que se ocupan en la actualidad para que un nuevo lenguaje de programación tenga éxito es que el lenguaje debe permitir trabajar a escala, con un gran número de dependencias y para que un gran número de programadores pueda trabajar en un mismo proyecto; debe de ser un lenguaje familiar, las generaciones jóvenes egresan con conocimientos en *C* o *Java* y se tiene la necesidad de conseguir programadores productivos rápidamente [5].

En el siguiente capítulo se describirá Go mencionando su historia, características, ventajas y sintaxis del lenguaje.

## CAPÍTULO 3

### GO

Go es un intento de combinar los tres deseos más grandes del desarrollo, eficiente, rápido y fácil; paradigma que hasta la fecha no se ha logrado con ningún lenguaje [3]. Go es un lenguaje con menos complejidad que los lenguajes orientados a objetos, como C++ y *java*. Muchas ideas para desarrollar este nuevo lenguaje fueron retomadas del desarrollo del *Sistema Operativo Plan 9*, como la introducción del UTF-8 [43].

Proclamado por sus autores como el lenguaje C del siglo 21. Go es perteneciente a la familia de lenguajes C y está basado en lenguajes como *Java*, C y C++. de donde toma la estructura y sintaxis. De *Modula* y *Pascal* heredó las declaraciones y paquetes, de *Limbo* y *Newsqueak*, se inspiró para el mecanismo de concurrencia. Finalmente, de *Python* y *Ruby* toma la sencillez y características de un lenguaje dinámico.

Go, un lenguaje expresivo, concurrente y con recolector de basura, es también un lenguaje muy simple, rápido y seguro. Go es *OpenSource*, distribuido con la licencia BSD, por lo que puede ser utilizado por todo el mundo, inclusive para uso comercial. Go pretende mejorar la creación de software de una forma simple, confiable y eficiente [44, 45].

Go empezó a desarrollarse en Google en Septiembre del 2007 utilizando un 20% de las horas de trabajo en esta empresa. Los ingenieros Robert Griesemer<sup>5</sup>, Rob Pike<sup>6</sup> y Ken Thompson<sup>7</sup> fueron los encargados y primeros en empezar este

---

<sup>5</sup> Conocido por su trabajo en la máquina virtual *JavaHotSpot*.

<sup>6</sup> Miembro del equipo Unix en *Bell Labs*, programador de *Plan9 OS* e *inferno OS*, y del lenguaje de programación *Limbo*

<sup>7</sup> Miembro del equipo Unix en *Bell Labs*, uno de los padres de C, *Unix* y *Plan 9 OS* junto con Rob Pike.



proyecto. A ellos se les unieron Ross Cox<sup>8</sup> e Ian Lance Taylor<sup>9</sup>, para formar el equipo de desarrolladores de Google. El hecho de citar a los autores del lenguaje es saber cuál es el rumbo del mismo y qué se podría esperar de su evolución, soporte y solución a los problemas que presente. Google ha lanzado muchos lenguajes de programación, pero muchos tienen una vida muy corta en la vida real.

En noviembre del 2009 Google lanzó Go, en este mismo mes Go se convirtió en *OpenSource*.

### **3.1 Características del lenguaje**

Go fue originalmente creado con el propósito general de cubrir los procesos complejos y grandes cantidades de datos en los servidores de Google. Se necesitaba solucionar el problema de concurrencia escalable, hacer más eficiente el software que necesitaba atender muchos eventos al mismo tiempo. A Go se le agregaron otros propósitos a lo largo del desarrollo del lenguaje dentro de la empresa Google. En el camino también se pensó como un lenguaje para servidores, clientes, bases de datos, caches, y balanceo de cargas. Desde su lanzamiento, se han encontrado una amplia lista de desarrollo de aplicaciones web, juegos y herramientas de gráficos, por citar algunos.

Go promete ser un lenguaje de propósito general y muchos educadores han expresado interés en su uso para enseñarlo como lenguaje base en las escuelas . Algunas características del lenguaje de programación Go se describen en los siguientes puntos:

- Compilación muy rápida

---

<sup>8</sup> Colaborador en la creación de *Plan 9 OS* y c/o en Google Inc.

<sup>9</sup> C/O programador de Google Inc.

Basándose en el resultado de una comparación respecto al tiempo de ejecución de diferentes algoritmos computacionales entre Go y PHP, se obtuvo que Go es, por mucho, superior en todas las pruebas de ejecución de ambos *benchmarks*. La comparación de los dos lenguajes mencionados en esta tesis se describe en el Capítulo 4.

- Codificación UTF-8

Go agrega este formato de codificación de caracteres Unicode e ISO 10646<sup>10</sup> utilizando símbolos de longitud variable (definido como estándar por la RFC 3629 de la *Internet Engineering Task Force*). Gracias a esta codificación se puede escribir en cualquier lengua y permite el uso de caracteres especiales, sin causar problemas al compilar o ejecutar el código [49].

- Tipado fuerte

Con respecto al manejo de tipos de datos, Go es lo contrario al lenguaje de programación PHP, el lenguaje de programación Go se asemeja al lenguaje de programación C. No admite que una variable guarde datos diferentes al tipo con el que se declaró la variable, para lograr hacer esto se necesita una conversión previa de tipos de datos.

- Memoria Virtual

Go reserva una amplia región de la memoria virtual como un respaldo para las asignaciones. Esta memoria virtual es local para el proceso específico de Go; la reserva no priva a otros procesos de memoria.

- Punteros sin aritmética

---

<sup>10</sup> Define el Conjunto de Caracteres Universal (*Universal Character Set*) como un sistema de codificación de caracteres en varios octetos. Contiene cerca de cien mil caracteres abstractos.

Por seguridad, Go se creó como un lenguaje que no puede moverse a una dirección ilegal. Es decir, no recorre con operaciones aritméticas entre espacios de memoria. Además, la falta de la aritmética de punteros puede simplificar la implementación del recolector de basura.

- 100% concurrente<sup>11</sup>

Estas características hacen a Go muy interesante para sistemas modernos. Go se ha implementado a través de primitivas del lenguaje y no a través de librerías, creando código concurrente como parte integral del lenguaje [3]. Go tiene una función independiente llamada *goroutine* que crea la concurrencia de una manera extremadamente fácil; esta función se puede llamar las veces que se requiera y la pila de llamadas crecerá o se contraerá según sea necesario. No es un hilo, pero se puede pensar como un hilo muy barato, ya que ocupa menos recursos de los que ocupa un hilo tradicional. Su llamada es muy sencilla, como se muestra en la siguiente línea de código para crear la concurrencia solamente se incluye la palabra reservada *go* antes de cada llamada de una función [45, 49].

```
1 go name-function ()
```

Go no se basa en candados (*mutex*) o monitores, se basa en canales y el envío de información a través de ellos es mucho más limpio y sencillo. Es mucho menos propenso a problemas como *Java* o *C*. No se necesita ser experto en este tema para usar *goroutines*.

---

<sup>11</sup> Propiedad de la computación donde se pueden ejecutar varias tareas a la vez. Si usted tiene un procesador, el programa puede ser concurrente pero no paralelo.

- Servidor Web incluido (godoc)

Go toma en cuenta la documentación como una parte fundamental en la fabricación de software accesible y fácil de mantener [48].

Go cuenta con paquetes, librerías y requerimientos almacenados en un conjunto de sitios web, donde se podrán hacer uso en cualquier momento.

Godoc, instrumento de documentación. Analiza el código fuente, por medio de los comentarios y produce un documento HTML ó texto sin formato, documentando todo tu desarrollo. El diseño de *godoc* es más simple que *JavaDoc*, ya que los comentarios no necesitan tener sintaxis propia de lectura.

Para documentar se escribe un comentario descriptor previo al código. Go se encarga de ordenarlo y ponerlo en una vista adecuada [12]. Esta documentación se puede agregar a la documentación general del lenguaje alojada en *Github*.

- La nube y OpenSource

Se proclama como programa en la nube por dos razones, la primera es porque todo su código está almacenado en la nube (<http://github.com/golang>), esto hace que el código este a la vista de cualquier persona que se encuentre interesada a contribuir o a corregir algún *bug*.

La segunda razón es por que cuenta con una versión para desarrollar en una plataforma web, donde se puede crear y ejecutar código sin necesidad de instalar ningún compilador localmente. Solo se necesita acceder a su modelo de servicio en modo plataforma en la página web <http://play.golang.org/>. Todo código creado dentro de la nube se queda

almacenado con un ID generado aleatoriamente, el cual sirve de identificador para poder ingresar posteriormente a ese código.

Go es un lenguaje orientado a objetos, pero a diferencia de otros no dispone de herencia de tipos y tampoco de palabras clave que denoten claramente que esté orientado a objetos. Las “clases” se declaran separadamente (*interfaces*, *structs*, *embedded values*). Permite el uso de delegaciones de atributos a través de *embedded values* y el polimorfismo por medio de las interfaces [5].

## **3.2 Manual de introducción a Go**

En esta sección se presenta un manual de introducción del lenguaje de programación Go. Este manual explica la instalación, compilación y varios casos que muestran la sintaxis tratando de hacer una analogía de lo que sería en el lenguaje de programación PHP.

### **3.2.1 Instalación**

En esta investigación se probó e instaló el compilador del lenguaje Go en 3 diferentes sistemas operativos:

-En *Windows*, con un equipo *HP pavillion DV4*, un procesador *Intel i3 x64* y la versión de *Windows8*. Esta fue el primer fracaso para probar el lenguaje, pues no se instaló como se esperaba. Después de una búsqueda y prueba por ejecutar Go en *Windows* se concluyó que este lenguaje no se ejecuta del todo bien en la plataforma de sistema operativo *Windows*.

-En un sistema UNIX, *Pea 8 Linux* en un equipo *Notebook Acer One Series* con un procesador *Atom N270 (1.60 GHz, 533 MHz FSB, 512 KB L2 cache)*, donde se instaló y se probó posteriormente en pruebas sencillas de consola encontradas en diversos manuales así como en la documentación del lenguaje.

-Por último se probó y se concluyó la investigación en un sistema *Mac OSX Yosemite*, equipo *MacBook Pro* con procesador *Intel core i7 (2.4 GHz, 256 KB por núcleo)*, en donde se completó el desenlace de esta investigación.

Este lenguaje de programación está disponible en código fuente y binarios para las siguientes plataformas [3, 46] :

- **FreeBSD 7.** Amd64, 386.
- **Linux.** Amd64, 386.
- **Os x.** Amd64, 386.
- **Windows.** Amd64, 386.

Para la instalación de Go en Linux desde una terminal tecleamos las siguientes intruciones.

```
$ sudo apt-get install golang █
```

Esta instrucción realiza la instalación del paquete “Golang”. Si este no funciona, se tendrá que cargar el repositorio correspondiente en donde se aloja el paquete y después de actualizar sin problemas el sistema, procedemos a instalar el paquete.

```
$ sudo add-apt-repository ppa:gophers/go █
```

```
$ sudo apt-get update █
```

```
$ sudo apt-get install golang-stable █
```

Para la instalación en *Mac OSX*, se necesita descargar el archivo ejecutable en la página oficial <http://golang.org/dl/> y ejecutarlo siguiendo las instrucciones. No se

mencionará la instalación en el sistema operativo de *Microsoft*, ya que en esta investigación no se encontró viable instalarlo en ese sistema operativo.

Para verificar si está instalado solo escribimos en consola la palabra Go seguida de un salto de línea, el cual nos mostrará los comandos del lenguaje como se muestra en la *Figura 2*.

```
$ go
Go is a tool for managing Go source code.

Usage:

    go command [arguments]

The commands are:

    build      compile packages and dependencies
    clean      remove object files
    env        print Go environment information
    fix        run go tool fix on packages
    fmt        run gofmt on package sources
    generate    generate Go files by processing source
    get        download and install packages and dependencies
    install    compile and install packages and dependencies
    list       list packages
    run        compile and run Go program
    test       test packages
    tool       run specified go tool
    version    print Go version
    vet        run go tool vet on packages

Use "go help [command]" for more information about a command.

Additional help topics:

    c          calling between Go and C
    filetype   file types
    gopath     GOPATH environment variable
    importpath import path syntax
    packages   description of package lists
    testflag   description of testing flags
    testfunc   description of testing functions

Use "go help [topic]" for more information about that topic.
```

**Figura 2 Especificaciones de Golang**

La *Figura 2* muestra los catorce comandos con los que cuenta Go. En esta investigación se usaron los comandos de **get** e **install**, para buscar instalar paquetes que se requerían; **version** al momento de verificar si se instaló Go; **run**, para compilar y correr pruebas; y **fmt** para ordenar código.

Go fmt se encarga de alineación y corrección del lenguaje. Go fmt emite de forma predeterminada tabuladores y/o espacios para un orden coherente y equitativo de

todos los programas escritos en Go, esto ayuda a que sea más fácil de escribir, leer y mantener el código para otros desarrolladores.

```
$ gofmt -w tucodigo.go █
```

En Go, todos los comandos se basan en una variable de entorno llamada `$GOPATH`. Esta variable apunta al espacio de trabajo de Go en el ordenador. En los sistemas Unix, la variable debe exportarse de la siguiente forma.

```
$ export GOPATH=$HOME/work █
```

Go cuenta con *gopm*, que es un manejador de paquetes que permite buscar e instalar de manera fácil y rápida una gran y extensa variedad de paquetes. La forma de instalar *gopm* es escribiendo en consola la siguiente línea.

```
$ go get -u github.com/gpmgo/gopm █
```

Este paquete junta la funcionalidad de *get* e *install*.

### 3.2.2 Compilación

Debido a que algunas partes de Go están escritas en C y ensamblador AT&T, se tiene que instalar un compilador de C (GC y GCC)<sup>12</sup>.

Go requiere cualquiera de los anteriores compiladores para que se instale y corra sin problema. En sistemas *Mac OSX*, lo incluye el programa *Xcode*. En *Linux* es necesario instalar previamente algún compilador por consola y en *Windows OS* se necesita instalar *MinGW* para instalar *GCC*.

---

<sup>12</sup> GCC escrito en C++ con un analizador sintáctico descendente recursivo. GC, escrito en C utilizando *YACC/BISON* para el analizado.



Es un lenguaje para compilarse en la nube, pero obviamente contiene un compilador para correr en sistemas físicos. Go compila y ejecuta en una sola línea

```
$ go run archivo.go █
```

Algunas características de este compilador son:

- Generan un código 10-20% más lento a C.
- Soporta arquitecturas de 32 y 64 bits además de *ARM*.
- Se puede enlazar con *GCC* y realizar compilación junto a C.
- No soporta pilas segmentadas.

### 3.3 Su sintaxis

Go en general presenta una coincidencia sobre sus lenguajes base, sobre todo con el lenguaje C. El recolector de basura, identifica cuando una variable o declaración concreta deja de usarse y por tanto libera su memoria asociada de manera automática. Este recolector actualmente muy simple pero efectivo y de baja latencia, se basa en un marcado de barrido del recolector de *IBMtm1*.

Go se inicializa con la función *main()*, (también se puede inicializar con *init()* ) la cuál no acepta ningún parámetro, ni devuelve alguno; la inicialización se ejecuta mono-hilo.

No se requiere escribir los punto y coma al final de cada línea, el compilador lo agrega automáticamente al encontrar salto de línea. Es por esto que Go especifica que las llaves de cerrar “}” deben de ir en un renglón nuevo. Solo en algunos casos específicos como para el for si es usada la “;”.

Las variables, tipos, funciones, constantes y campos que quieran ser usados públicamente la primera letra del nombre debe ser mayúscula. Si no, su uso será solo privado.

### 3.3.1 Tipos de errores

Go tiene variables tipo *error* para efectos de mensajes de error. También cuenta con un paquete llamado *errors* para controlar los errores.

```
Go
4 Err := errors.New('error: descripción') if error != nil { fmt.Print(err) }
```

### 3.3.2. Comentarios

Los comentarios se hacen igual que en otros lenguajes de programación como, por ejemplo, en los lenguajes de programación PHP y C.

```
Go
4 //una línea
5 /*Varias líneas*/
```

### 3.3.3. Asignaciones

Existen 3 tipos:

- **Simples:** Asignar valor o variable a otra variable.
- **Múltiples:** A un grupo de variables se les asignan varios valores.
- **Múltiples de funciones:** A un grupo de variables se les asignan funciones.

```
Go
1 a = 2 //asignación simple
2 a, b c = 1, 2, 3 //asignación múltiple
3 x, y, z := function() //asignación múltiples a función
```

Se pueden asignar variables en cualquier parte del programa, ya sea en la declaración de variables, como en la declaración de una sentencia condicional o ciclo.

A diferencia de PHP, en Go si se permite la asignación múltiple. En el lenguaje de programación Go se puede hacer un *swap* sin el uso de una tercera variable auxiliar.

```
Go
2 a, b = b, a
```

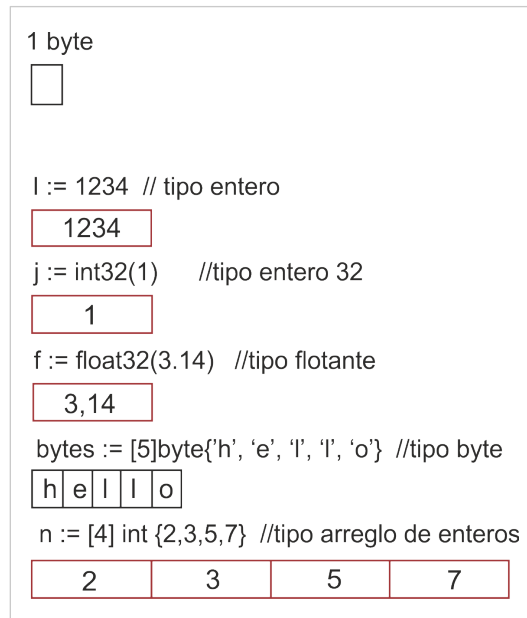
Mientras que en el lenguaje de PHP si se tiene que utilizar una variable auxiliar para lograr hacer el intercambio.

```
PHP
2 $aux = $a;
3 $a = $b;
4 $b = $aux;
```

### 3.3.4. Variables

En el lenguaje de programación Go existen 11 tipos para representar un valor: **números** (1, 2e48, 0x0FF), **booleanos** (0/1), **cadena**s (“ejemplo”), **arreglos**, **slices**, **estructuras**, **punteros**, **funciones**, **interfaces**, **canales** y **mapas**.

Cada tipo de variable en Go tiene diferente tamaño en memoria. Por ejemplo, un entero de 32 bits y un flotante ocupan 4 bytes. Las cadenas contienen los bytes que se declaren, y finalmente el tamaño en bytes de un arreglo de enteros o flotantes es el resultado de la multiplicación de la longitud del arreglo por 4, pues cada espacio del arreglo equivale a 4 bytes.



**Figura 3 Estructura de datos en Go**

- **Números:** Existen: enteros, enteros sin signo y flotantes.

Los tipos enteros en el lenguaje de programación Go son: *uint8*, *uint16*, *uint32*, *uint64*, *int8*, *int16*, *int32* y *int64*. 8, 16, 32 y 64 nos dicen cuántos bits contiene cada uno de los tipos. *Uint* significa entero sin signo y solo contienen números positivos (o cero), mientras que *int* significa número entero.

Go tiene dos tipos flotantes: *float32* y *float64* (también conocido como precisión simple y doble precisión), así como dos tipos adicionales para representar los números complejos (números con partes imaginarias): *complex64* y *complex128*. Generalmente se usa *float64* cuando se trabaja con números de punto flotante.

- **Booleanos:** True o false.
- **Strings:** Cadenas de texto pero delimitadas por su longitud, no por un carácter *null*, haciendo el *string* más seguro y eficiente. El valor debe de ir dentro de comillas (“ ”) o acentos abiertos (‘ ’). Las cadenas se pueden

declarar de dos formas, una es primero inicializar la variable y después asignarle un valor o simplemente asignarle un valor y está se convierte automáticamente en *string*.

```
Go
3  var Cadena string
4  Cadena = "cadena nueva"
5
6  //asignación
7  Cadena2 := "cadena nueva 2"
```

Un *string* no se puede modificar por bytes, previamente se tiene que realizar una conversión a *slice* para poder modificar los datos.

```
Go
9  S[0] = 'c' //error !
10 C := []byte(S) //convierte a cadena tipo byte
11 C[0] = 'c'
12 S2 = string(C) //regresa a un string
```

Al igual que en muchos otros lenguajes de programación, el signo “+” sirve para unir varias strings.

Las variables en el lenguaje de programación Go se puede declarar de las siguientes formas:

- a) Establecer que declararas todas las variables iniciando por la palabra reservada *var*; primero se ponen las variables separadas por coma (,) y luego se anexa el tipo de variable.

```
Go
3  var (
4  |  A, B, C int
5  |  )
```

b) Se pone la palabra reservada antes de cada declaración

Las variables pueden adoptar valores como:

```
Go
6 var A int
7 var B int
```

Para expresar una constante, solo es necesario aplicar la palabra reservada *const* antes de la declaración de esta.

```
Go
8 const pi = 3.14
```

*Const* es equivalente a definir una variable en PHP.

```
PHP
9 define('MIN_VALUE', '0.0');
```

El lenguaje de programación PHP cuenta con enteros, flotantes, arreglos y booleanos pero carece de tipado. Los tipos de datos se muestran en el siguiente bloque de código.

```
PHP
2 $int = 10;
3 $float = 1.5;
4 $bool = True;
5 $string = "hola";
```

### 3.3.4.1. *iota* y *apex*

Ambas son palabras reservadas dentro del lenguaje de programación Go y se usan como incrementos en una lista de variables declaradas, pero solo funcionan en los constantes. En PHP no se encontraron funciones o palabras análogas a estos conceptos.

***iota***, es un contador que comienza en 0 y aumenta en uno su valor encontrando un “;”. Se asigna a una o más variables siempre y cuando estas variables sean constantes.

```
Go
10 const{
11     Var1 = iota; //0
12     Var2 = iota; //1
13 }
```

***Apex***, también es una palabra clave que genera enumeración comenzando desde 0 e incrementando en 1. Al igual que *iota*, se genera en las constantes y colocando solo un *apex* en la primer variable, las siguientes se inicializan en el valor del anterior + 1.

```
Go
14 const{
15     Var = apex //0
16     Var1 //1
17     Var2 //2
18 }
```

### 3.3.4.2. *Cero o nulo*

Los diferentes valores en cero adquiridos en Go son:

```

Go
1 Int:      0x0
2 Float:    0.0
3 Bool:     false
4 Punteros: nil
5 String:   ""
6 Struct:   todos los valores en 0.

```

### 3.3.4.3 Tipos de datos compuestos

#### 3.3.4.3.1 Arreglos

Los arreglos son estructuras de datos que permiten tener un grupo de datos del mismo tipo distribuido uniformemente en un bloque de memoria. En Golang, los arreglos son como en el lenguaje de programación *Pascal*. Igual que cualquier variable en Go, un arreglo se puede inicializar con la palabra reservada *var* seguido por el tamaño y el tipo del arreglo; o simplemente se puede asignar valores a un arreglo sin inicializar.

```

Go
7 var array1[n]type // declarado sin inicializar
8 array2 := []type {values} // asignado sin una longitud
9 array3 := [n]type {values} // asignado con una longitud

```

En el siguiente cuadro de código se muestran las 3 diferentes formas de inicializar un arreglo en el lenguaje de programación Go. En la línea 12 simplemente se declara un arreglo llamado *array1*, que puede ser inicializado de la forma común; en la línea 14 se asigna un arreglo a la variable *array2* sin saber la longitud del arreglo, el arreglo toma como longitud el número de valores que se le asignen; en la línea 16 se asigna un arreglo a la variable *array3*, con una longitud previa declarada. Cualquier arreglo puede ser manipulado por medio de sus índices.



```

Go
12 var array1 [5]int // array declarado
13 array1[1] = 10 // inicia elementos del array
14 array2 := []int{1, 2, 3, 4, 5} // asignacion
15 array2[1] = 50 // modifica los elementos del array
16 array3 := [5]int{5, 4, 3, 2, 1} // asignacion
17 array3[1] = 100 // modifica los elementos del array

```

En PHP un arreglo se declara de la manera tradicional y se puede inicializar de forma tradicional por medio de sus índices como se muestra en las líneas 2 y 3 del siguiente cuadro de código. Para asignar un arreglo a una variable en PHP se tiene que hacer uso del constructor `array()`, la cual toma como parámetros los valores que se quieren asignar al arreglo.

```

PHP
2 $array1[5]; // declara un arreglo
3 $array1[1] = 10; // inicia elementos de un arreglo
4
5 $array2 = array(1, 2, 3, 4,5 ); // asignación sin una longitud
6 $array2[1] = 50;

```

Para generar matrices de múltiples dimensiones, el lenguaje de programación Go requiere que le especifique el tamaño de la forma `[x] [y]`, donde X son los renglones y Y son las columnas, o el tamaño de cada arreglo. X y Y se pueden dejar en blanco y tomará como X el número de sub arreglos declarados dentro del arreglo y el valor de Y será la longitud de los sub arreglos, como está en el próximo bloque de código

En el siguiente cuadro de código se muestra un arreglo de enteros de dos dimensiones llamado `doubleArray` que tiene un tamaño de 4 en cada sub arreglo. Cada sub arreglo se divide por una coma “,” y sus valores también se encuentran dentro de llaves “{ }”.

```
Go
19 multidimensional := [][]int{{1, 2, 3, 4, 5}, {6, 7, 8, 9, 0}}
```

Los arreglos multidimensionales en el lenguaje de programación Go, al igual que en Java, no necesariamente deben de ser de la misma longitud en sus sub arreglos.

```
Go
20 multidimensional := [][]int{{1, 2, 3, 4}, {5, 6, 7}, {8, 9}}
```

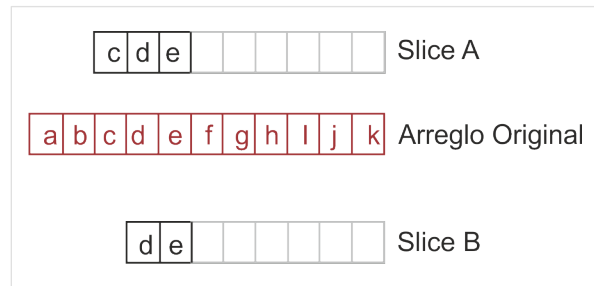
Como PHP carece de tipado, un arreglo multidimensional se construye respecto a los requerimientos del desarrollador, llamando recursivamente el constructor *array()*. En el siguiente cuadro de código se genera la misma matriz que se generó en el lenguaje de programación Go, una matriz de

```
PHP
8 $multidimensional = array(array(1, 2, 3, 4, 5), array(6, 7, 8, 9, 0));
```

#### 3.3.4.3.2. Slices

En ocasiones, el tipo matriz no es una buena opción, por ejemplo cuando no sabemos el tamaño de la matriz. Entonces necesitamos usar una matriz dinámica, llamada *slice*.

Realmente no es una matriz dinámica, es una referencia a una sección de un *array*, como se muestra en la *Figura 4*, este tipo de datos es más usado que los *array* o los *maps*.



**Figura 4 Crear un slice desde un arreglo**

Los *slices* son una estructura muy ligera que permite generarlos, aumentarlos y/o reducirlos. Además es más fácil pasar *slices* como parámetros a funciones.

Las *slices* se declaran como un arreglo, pero sin tamaño; también se puede crear vinculando una parte de un arreglo ya creado. Es importante mencionar que los cambios realizados en un *slice* creado desde otro array serán repercutidos en el arreglo original. En el siguiente bloque de código se declara un arreglo entero con nombre 'a' sin tamaño, este arreglo será un *slice* creado con los primeros 4 bits de un arreglo llamado 'name\_array'.

```

Go
4 var a[]int
5 a = name_array[0:3] // nueva slice asignando parte de un array

```

A los *slices* se les puede asignar memoria usando *make()*; los parámetros son el tipo de variable que guardará, la longitud con la que iniciará y el límite (la capacidad).

```

Go
1 var name_slice = make( [ ]tipo, longitud, capacidad )

```

Como se muestra en la *Figura 5* un *slice* contiene las siguientes partes:

- Un **puntero** donde empieza el *slice*.
- La **longitud** del *slice*.
- **Capacidad**, cuantos bytes ocupará.

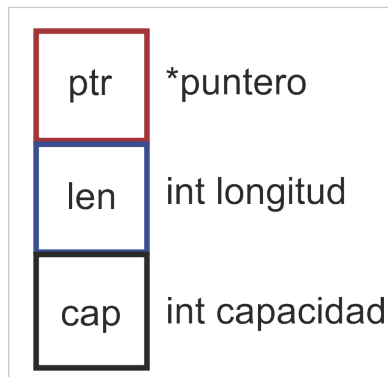


Figura 5 Partes de un Slice

Los *slices* en PHP son el resultado de un constructor llamado `array_slice` que regresa una copia de los valores de una parte de cualquier arreglo existente. A comparación de Go los valores se guardan en una nueva localidad de memoria y sus cambios no repercutirán al arreglo original. La sintaxis del *slice* se menciona en el siguiente bloque de código.

```

PHP
2  array = array_slice(arregloExistente, indice, longitud);
3  // arregloExistente -> es el arreglo de donde se extraeran los datos
4  // indice -> de donde empezará el slice
5  // longitud -> la longitud del slice, empezando de indice
6  // array = devuelve un tipo array con la copia de los datos

```

Siguiendo la sintaxis de un *slice* en PHP, en la línea 3 del siguiente bloque de código se crea un *slice* de un arreglo ya existente. Este *slice* contiene el valor de (3,4,5,6) ya que se inicio en el índice 2 y con una longitud de 4.

```

PHP
2  $arreglo = array(1,2,3,4,5,6,7,8,9,0);
3  $slice = array_slice($arreglo, 2, 4); // devuelve 3,4,5,6

```

### 3.3.4.3.3. Maps

Se comportan como los diccionarios en *Python*. Son otro tipo de referencias a otros tipos. Es un conjunto de elementos con **clave:valor**, siendo así su fácil acceso y comparación entre estos. Los mapas no tienen longitud fija, pero se puede saber su longitud con *len(map)*.

Hay tres formas de crear *maps*:

#### 1. Literal. Lista de pares "clave: valor"

```
Go
1 m = map [string] float { "uno":1 , "pi":3.14 }
```

#### 2. Creándolos con *make()*

```
Go
2 m := make(map[string]float, 10) //donde 10 es el tamaño inicial del slice
3 m["uno"] = 1
4 m["pi"] = 3.14
```

#### 3. Asignación de otro map

Para acceder al elemento con la siguiente instrucción; si no está, solo retornará un error.

```
Go
5 c := m["uno"] //correcto
6 e := m["tres"] //generá error
```

Para reemplazar, o introducir un valor

```
Go
7 m["pi"] = 3.1416
```

La formula “:=” declara e inicializa. Retorna un booleano verdadero o falso respecto a si encuentra el valor en el *map* y solo se puede usar dentro de funciones.

```
Go
1 m = map[string] float {"1" : 1 , "pi":3.14}
2 var ok boolean
3 var v int
4
5 v, ok := m[x]
```

Lo análogo a mapas en el lenguaje de programación PHP se hace con el constructor *array()* o entre corchetes a partir de PHP 5.4.

La sintaxis de la creación arreglos clave-valor en PHP empieza por declarar el constructor *array()* y dentro de este constructor se escribe la clave (siempre *string*) seguida del conector =>, y finalmente el valor asignado a esa clave.

En el siguiente bloque de código se muestra el mismo ejemplo que se realizó en el lenguaje de programación Go.

```

PHP
2  $m = array(
3      "uno" => 1,
4      "pi" => 3.14,
5  );
6
7  // a partir de PHP 5.4
8  $m = [
9      "uno" => 1,
10     "pi" => 3.14,
11 ];

```

En el lenguaje de programación PHP también se puede declarar un arreglo sin tamaño fijo y después asignarle una clave con su valor. Igualmente los valores pueden modificarse de una manera muy similar a como se hace en el lenguaje de programación Go.

```

PHP
13  $m = array(); //crear un array sin dimensión, ni valores
14  $m["uno"] = 1;
15  $m["pi"] = 3.14;
16
17  $m["pi"] = 3.1416; //modificación de valor en "pi"
18

```

A diferencia de Go, para crear un mapa en el lenguaje de programación Go no se necesita de palabras reservadas.

#### 3.3.4.3.4. Structs

Tipos de datos que contienen atributos y permiten crear datos más complejos. Su sintaxis es relativamente común declarándose como se muestra en el siguiente bloque de código.

```

Go
1  var ejemplo struct { x, y float }
2
3  type ejemplo struct { x, y float}
4  var p ejemplo

```

Los campos de un *struct* que empiezan con una letra mayúscula, indican que serán visibles fuera del paquete. Cuando no, son tipos y atributos privados. También se pueden declarar *structs* como un tipo de otro *struct*, más adelante se abordara el tema de programación orientada a objetos.

### 3.3.5. Estructuras de control

Las estructuras de control pueden dividirse en estructuras de control condicional y estructuras de control repetitivo.

Las estructuras de control condicional, *if* o *switch* son expresiones que evalúan respecto a un valor booleano, si su expresión es verdadera, se ejecuta la expresión, si no lo es la expresión se ignora. Las estructuras de control repetitivo son ciclos que ejecutan una sentencia simple o compuesta un número de veces.

Las estructuras de control en el lenguaje de programación Go son similares a las estructuras de control en PHP y también en el lenguaje de programación C, solo que no admite el uso de paréntesis en la condición, pero sí admiten el uso de llaves para enmarcar las sentencias. En el lenguaje de programación Go no existen las palabras reservadas *while* o *do while*.

En el siguiente bloque de código se muestran la estructura de control condicional *if* y la estructura de control cíclica *for* que si el desarrollador lo desea puede servir como *while* e incluso como *do while* añadiendo una condición dentro del ciclo.

```
Go
1  if a == b {
2      fmt.Print("a y b son iguales")
3  } else {
4      fmt.Print("a y b son distintos")
5  }
6
7  for i = 0; i < 10; i++ { ... }    //for común
8  for ; ; { ... }                // bucle infinito
9  for { ... }                    // bucle infinito
10 for a<5 { ... a-- }           // bucle WHILE
```



Para crear un *while* infinito en el lenguaje de programación Go basta con declarar un *for* sin atributos o con doble punto y coma “ ; ; “ como se muestra en las líneas 8 y 9 del pasado bloque de código. En el renglón 10 se genera una analogía de *while* en Go, aquí se llama a la estructura de control cíclica *for* con una condición como único parámetro.

La estructura de control condicional *if* en el lenguaje de programación PHP es la misma que en lenguaje de programación Go pero en PHP la condición es encerrada entre paréntesis. En el siguiente bloque de código se presenta el mismo ejemplo pero realizado en PHP.

```
PHP
2 if ($a == $b) {
3     print "a y b son iguales";
4 }else {
5     print "a y b son distintos"
6 }
```

En PHP si existe las estructuras de control repetitivo *while* y *do while*; todos sus ciclos o bucles contienen una sintaxis idéntica a los bucles del lenguaje de programación C.

```
PHP
2 // Bucle while
3 while (condición) {
4     ... sentencias;
5 }
6
7 // Bucle do while
8 do {
9     ...sentencias
10 } while(condición);
11
12 // Bucle for
13 for ($variable=valor_inicial; condición; incremento) {
14     ... sentencias;
15 }
```

En Go la instrucción *for* puede iterar sobre arreglos, *slices* y mapas. Por ejemplo, en el siguiente bloque se muestra el ciclo para moverse dentro de un mapa.

```
Go
1 for key , value := range m{
2     fmt.Printf("clave %s, valor %g ", _key, value)
3 }
```

Donde *key* es la etiqueta en el mapa y *value* el valor en esa posición. La palabra reservada *range* seguida de una variable, consigue crear una analogía del ciclo *foreach* existente en el lenguaje de programación PHP.

En PHP existe *Foreach*, que es un constructor que proporciona un modo sencillo de iterar sobre arreglos y objetos. Emitirá un error al intentar usarlo con algún tipo diferente o una variable no inicializada. La sintaxis de *foreach* es mostrada en el siguiente bloque de código.

```
PHP
2 foreach ($array as $valor) { /*..sentencias*/ }
3 // $array -> El arreglo a recorrer.
4 // $valor -> Apunta al valor del arreglo en esa posición.
5 // foreach aumenta en uno cada iteración hasta el fin del arreglo.
```

Siguiendo el mismo ejemplo que se mostro en el lenguaje de programación Go, en el siguiente bloque de código se muestra un ciclo que itera a lo largo de un mapa llamado *\$m*.

```
PHP
8 foreach ($m as $value) {
9     echo "Valor[$key]: $value\n";
10    $key = $key + 1;
11 }
```

En el lenguaje de programación Go los ciclos y las condicionales deben llevar la primera llave “{” en el mismo renglón de la condicional o ciclo, de lo contrario, el analizador léxico insertará un punto y coma después de la condición causando error de compilador o problemas no deseados.

### 3.3.5.1. Switch

La estructura de control condicional *switch* es similar a una serie de *if*. Usada para poder comparar una variable con una serie de valores distintos. El *switch* de Go es más potente que el del lenguaje de programación PHP, no tiene orden en los case y no necesita obligatoriamente un número o una cadena para comparar, puede tener expresiones en cada caso (*case*) simulando varios *if* en grupo.

Para poder llevar un orden dentro del Switch, se usa la palabra reservada *fallthrough* que crea una secuencia dentro de la estructura de control. En las líneas 2, 3 y 4 del siguiente bloque de código se muestra un conjunto de *cases* los cuales se ejecutan en forma secuencial y albergan condiciones simulando una serie de instrucciones.

Un *switch* puede contener una serie de condiciones en un mismo caso como es la línea 8.

```
Go
1 switch {
2     case a < b: return -1; fallthrough
3     case a == b: return 0; fallthrough
4     case a > b: return 1;
5 }
6
7 switch {
8     case 1,2,3: error()
9     default: fmt.Println("1")
10 }
```

En el lenguaje de programación PHP, la estructura de control condicional *switch* es una serie de condiciones basadas en un índice de tipo entero o en un índice

tipo cadena; esta estructura de control en el lenguaje de programación PHP también puede albergar varias condiciones para realizar alguna acción.

El siguiente bloque de código muestra un ejemplo del uso de la estructura de control condicional en PHP.

```
3 switch ($i) {
4     case "cadena": //condición tipo cadena
5         print "i es igual a una cadena\n";
6         break;
7     case 0: //condición tipo entero
8         print "i es igual a cero\n";
9         break;
10    case 1: //multiples condiciones
11    case 2:
12        print "i es menor que 3 pero mayor que uno\n";
13        break;
14 }
```

Como en otros lenguajes, *break* es una palabra reservada usada para terminar una función, mientras que *continue* es la palabra reservada que continua la ejecución en estructuras de control. Ambas palabras reservadas son usadas en los dos lenguajes de programación comentados en esta investigación.

### 3.3.6. Funciones

Una función es una parte del programa que realiza una tarea específica y puede o no retornar un valor. Opcionalmente pueden recibir valores de parámetros que modifiquen su funcionamiento, Cuando una función es invocada se le pasa el control a la misma, una vez que finaliza la tarea el control es regresado a la función principal.

Para realizar una función en el lenguaje de programación Go se usa la palabra reservada *func*, seguida del nombre, los parámetros entre paréntesis y finalmente el(los) valor(es) a regresar. En las funciones del lenguaje de programación Go también se pueden aceptar cero o más parámetros usando tres puntos "...".

La sintaxis de una función en el lenguaje de programación Go se presenta en la siguiente línea de código

```
Go
1 func func_name(args ...type) (return1 ... type) {
2     //code
3     return return1 ...
4 }
5 //func -> Palabra reservada para iniciar una función
6 //args -> Puede tener uno o más argumentos
7 //type -> Tipo de variable que se agrega como atributo
8 //return1 -> Puede asignarle uno o más valores de retorno
9 //return -> Puede regresar uno o varios valores
```

Una ventaja en las funciones del lenguaje de programación Go, es que las funciones pueden regresar múltiples valores en el *return* y están inicializadas por defecto en cero. Cuando se usa una función se copia solo el puntero a memoria, pero no los valores.

```
Go
1 // Función con múltiples valores devueltos
2 func MySqrt(f float) (float, bool) {
3     if f >= 0 {
4         return math.Sqrt(f), true
5     }
6     return 0, false
7 }
```

El lenguaje de programación PHP también tiene una palabra clave para denominar que se trata de una función. A diferencia de las funciones en el lenguaje de programación Go, PHP no permite que la función retorne más de un atributo.

```

PHP
2 function name_func ($arg_1, /* ... */ $arg_n){
3     //code
4     return $retorno;
5 }
6 // function -> Palabra reservada para iniciar una función
7 // $arg_1 ... -> Puede contener uno o más atributos
8 // return -> Puede retornar solo una vez.

```

### 3.3.7. Algunas funciones y palabras reservadas

El lenguaje de programación Go cuenta con solo 25 palabras clave, contra 74 palabras reservadas que existen en PHP, 50 en *Java* y 48 en *C++*. Las palabras reservadas en el lenguaje de programación Go son:

***break, default, func, interface, select, case, defer, go, map, struct, chan, else, goto, package, switch, const, fallthrough, if, range, type, continue, for, import, return, var.***

Los lenguajes de programación también cuenta con gran cantidad de funciones previamente cargadas y predeterminadas que ayudan en la elaboración de código. Algunas de las funciones que son útiles en el lenguaje de programación Go se mencionan a continuación.

#### 3.3.7.1. *new ()* y *make()*

Función usada para crear nuevos objetos. En el lenguaje de programación Go esta función reserva memoria y devuelve un puntero a la estructura creada.

```

Go
1 var p *nw_var = new(nw_var)
2 nw_v := new(nw_v) // la asignacion se apunta a un int

```

Go no tiene funciones para liberar espacio puesto que existe su recolector de basura.



En el bloque de código anterior se representa como la letra T al tipo de dato del que se está haciendo uso en cada línea. A continuación se muestra una serie de ejemplos del uso de *make()*.

```
Go
12 s := make([]int, 10, 100) // slice con longitud == 10 y capacidad == 100
13 s := make([]int, 1e3)    // slice con longitud y capacidad == 1000
14 s := make([]int, 10, 0)  // ilegal: longitud > capacidad
15 c := make(chan int, 10)  // channel con un buffer tipo entero de tamaño 10
16 m := make(map[string]int, 100) // map con un inicial espacio para 100 elementos
```

### 3.3.7.2. Defer

Esta palabra reservada es otro truco en el lenguaje de programación Go, se usa en funciones, su función es convertir una cola en una pila, es decir la instrucción que contenga *defer* será encolado al final de las operaciones que contiene el programa. En los ciclos se puede generar una cola.

```
Go
1 func func_name() {
2     for i := 0; i < 10; i++ {
3         defer fmt.Println(i)
4     }
5 }
```

En el bloque de código anterior se muestra un ejemplo, donde una función sin atributos que contiene un *for*, imprime el índice de este ciclo las 10 veces que se llama. También incluye la palabra reservada *defer* que genera una pila.

Se genera una cola por que la primera llamada a impresión de pantalla se encola a esperar que termine las demás operaciones del ciclo, y este comportamiento lo toman todas las llamadas a impresión de pantalla haciendo que la última petición sea la primera en imprimirse.



El lenguaje de programación PHP carece de alguna función que sea análoga a *defer*.

### 3.3.7.3. Funciones lambda y closures

El cálculo lambda es un sistema formal diseñado para investigar la definición de función, la noción de aplicación de funciones y la recursión. Este concepto fue introducido por Alonzo Church<sup>13</sup> y Stephen Kleene<sup>14</sup> en la década de 1930 para denominar a funciones no regulares [13].

Las funciones lambda o funciones anónimas se conforman de dos observaciones importantes. La primera observación del cálculo lambda es que las funciones no necesitan ser explícitamente nombradas. La segunda observación es que el nombre que se asigne a los argumentos de la función es generalmente irrelevante, o bien puede no contener argumentos.

Las funciones lambda se usan cuando no se tiene sentido encapsular una lógica o para hacer alguna acción que solo tiene sentido realizar una vez. Una función puede ser el argumento de otra función o también puede jugar el papel del valor devuelto por otra función. El resultado de esta función puede ser asignado a una variable.

En el lenguaje de programación Go es posible usar directamente una función anónima como parámetro de otra función. En el siguiente bloque de código se muestra el uso de una función anónima como parámetro de otra función.

```
Go
3 func lambda(f func() string) {
4     mensaje := f()
5     fmt.Println(mensaje)
6 }
```

de Washington, se diplomó en 1924 y obtuvo su doctorado en 1927 en la Universidad de Princeton, donde ejerció como profesor entre 1929 y 1967. Su obra más conocida es el desarrollo del cálculo lambda. Su trabajo inspiró a su alumno Alan Turing. Church Murió el 11 de agosto de 1995.

<sup>14</sup> Lógico y matemático estadounidense, nacido el 5 de enero de 1909 en Madison, Wisconsin y murió el 25 de enero de 1994.

El bloque de código anterior consta de 4 líneas, la línea 3 es el inicio de la función, a comparación de una función común el parámetro que contiene esta función es de tipo *func()* indicando con esto que la función recibirá una función anónima de parámetro.

```
Go
7 func main() {
8     //parametro
9     lambda(func() string {
10        return "hola mundo"
11    })
12    //asignación
13    asignación := func() string {
14        return "hello world"
15    }
16    lambda(asignación)
```

El bloque de código anterior muestra la forma de usar las funciones anónimas. En la línea 9 se utiliza una función anónima como parámetro en la llamada de función lambda; en la línea 13 se realiza una asignación de una función anónima. Esta nueva variable creada puede ser usada de parámetro. Las dos funciones que contiene el bloque de código anterior generan la misma salida.

En el lenguaje de programación PHP también existen las funciones anónimas con su uso es muy parecido al que tienen en el lenguaje Go. En el siguiente cuadro de código se muestra la creación de una función que recibe un parámetro, la función que realiza es solo la impresión del parámetro.

```
PHP
2 function lambda ($message){
3     print $message()."\n";
4 }
```

La diferencia que existe respecto al lenguaje de programación Go es que el lenguaje PHP no requiere que las funciones conozcan previamente que recibirán

de parámetro una función anónima ya que este lenguaje no es tipado, para la función es lo mismo si se le pasa como atributo un entero, una cadena o se le pasa una función.

En la línea 6 del siguiente bloque de código se muestra el uso de la función anónima como parámetro de la función llamada `lambda`; en la línea 11 del mismo bloque de código se realiza una asignación de una función anónima.

```
PHP
5 //parametro
6 lambda(function(){
7     return "hola mundo";
8 });
9
10 //asignacion
11 $asignacion = function () {
12     return "hello world";
13 };
14 lambda($asignacion);
```

Al igual que como se mostró con el lenguaje de programación Go, la variable creada por asignación puede ser un atributo correcto para mandar llamar una función.

Las funciones *closures* son esencialmente lo mismo que un función lambda pero con la ventaja de que pueden acceder a variables fuera del entorno de la función anónima accediendo a variables del entorno padre.

En el siguiente bloque de código se muestra un ejemplo sencillo de una función *closure* escrita en el lenguaje de programación Go. En esta función ejemplifica el acceso a una variable global, en este caso llamada *name*; esta función retorna una cadena que contiene un saludo anexando la variable global *name*.

```

Go
17     name := "Octavio"
18     closure := func() string {
19         return "Hola " + name
20     }
21     fmt.Println(closure())

```

La forma en la que se puede acceder a una variable global en el lenguaje de programación PHP solamente varía en que necesita usar el constructor *use* para indicar que usará alguna variable fuera de esta función anónima. En el siguiente bloque de código se muestra el mismo ejemplo explicado en el lenguaje de programación Go.

```

PHP
15 $name = "Octavio";
16 $closure = function() use ($name) {
17     return "Hola $name";
18 };
19 print $closure()."\n";

```

Si se requiere alterar alguna variable dentro de las funciones *closures* en el lenguaje de programación PHP es necesario añadir un símbolo de unión (&), haciendo un paso por referencia y por lo tanto modificando la variable original.

### 3.4 Programación orientada a objetos

Dando una mirada hacia atrás., el concepto de POO surgió en 1960, con el primer lenguaje orientado objetos llamado Simula. Simula introdujo objetos, clases, herencia, subclases y más. Lo más importante que se introdujo fue el cambio de pensamiento y la lógica como existía [14]. Ni la herencia, ni las clases son un concepto esencial de la programación orientada a objetos, ya que se dicen que son medios de implementación. Lo esencial del paradigma de programación orientada a objetos, es el paso de los mensajes [15].

Podemos no estar familiarizados con *Simula*, pero si con C++ ó Java que están inspirados en Simula ó a su vez JavaScript, PHP y Perl que están inspirando en los anteriores.

Aunque Go tiene objetos y métodos, además permite un estilo de programación orientado a objetos, no admite la herencia. En el lenguaje de Go, el concepto de interfaz se parece más a java en el aspecto que no necesita declaración explícita.

Una clase, es un conjunto de características que definen cierto grupo de objetos. Un objeto es un tipo de dato abstracto que tiene estado y comportamiento. Un método es una capacidad o función del objeto [16].

En el lenguaje de programación PHP, como en otros lenguajes orientados a objetos, es necesario el uso de la palabra reservada *class* para indicar que se generará una clase nueva.

```
PHP
2 Class NuevaClase {
3     Var $atributo1;
4     Var $atributo2;
5 }
6
```

En el segundo renglón del pasado cuadro de código, se declara una nueva clase llamada *NuevaClase*, la cual contiene dos atributos.

El lenguaje de programación Go no tiene alguna palabra reservada que denote una “clase”. Pero cuenta con un tipo que coincide con la misma definición y comportamiento el cual se llama *struct* [14]. La encapsulación de una clase en Go, se hace mediante una estructura. En el siguiente cuadro de código se crea una estructura nueva llamada *NuevaClase* que contiene dos atributos.

```
Go
3 type NuevaClase struct {
4     atributo1 int
5     atributo2 int
6 }
```

La forma de crear un objeto de algún tipo de clase en el lenguaje de programación PHP se necesita la palabra reservada *new* posteriormente el nombre de la clase que será el objeto. Tal como se muestra en el siguiente bloque de código.

```
PHP
13 $nuevoObjeto = new NuevaClase(); //Nuevo objeto PHP
```

Existen diversas formas de iniciar un tipo (objeto) en Go, a mi punto de vista la forma más sencilla es inicializar y asignar como ilustra el siguiente cuadro de código, la variable *NuevoObjeto* corresponde a un objeto de la clase *NuevaClase*.

```
Go
9 nuevoObjeto := NuevaClase() //Nuevo Objeto en Go
```

En el lenguaje de programación PHP existen funciones que permiten iniciar y destruir objetos de una manera sencilla. Estas funciones son *\_\_construct* y *\_\_destruct*. Go no necesita de estas funciones ya que se pueden incluir automáticamente [17].

La función *\_\_construct* es una forma para generar objetos de una manera más simple y rápida. En PHP, se requiere destruir los objetos utilizando la función *\_\_destruct*, que destruye los objetos al terminar el *script*.

En el primer bloque del siguiente cuadro de código se muestra la creación de la función `__construct`, la cual debe tener en orden los parámetros que se requieren ingresar al objeto, en este ejemplo solo son dos. En las siguientes 2 líneas del código, se agregan los datos a los atributos específicos del objeto con ayuda de `this`, palabra reservada que hace mención que realizara un cambio a esa misma clase.

En el segundo bloque de código, el renglón 6 del código muestra la función `__destruct`, la cual no requiere de ningún parámetro y se ejecuta al terminar el *script* destruyendo los objetos .

```
PHP
6   function __construct($a1, $a2){
7       $this->atributo1 = $a1; //asigna valor a primer atributo del objeto
8       $this->atributo2 = $a2; //asigna valor al segundo atributo del objeto
9   }
10
11  function __destruct(){ echo "objeto destruido"; } //destruye el objeto
12 }
13
14 $nuevoObjeto2 = new NuevaClase(1,2); //nuevo objeto construido con __construct
```

El lenguaje de programación Go no necesita de funciones previas, Go acepta varias formas para declarara un nuevo objeto. Una de ellas es ingresar el valor de cada atributo del *struct* dentro de dos llaves, separándolo por comas.

```
PHP
11  nuevoObjeto := NuevaClase(432,999) //Nuevo Objeto en Go incluyendo valores al crearlo
```

Los métodos en el lenguaje de programación PHP son funciones declaradas dentro de la clase, mientras que en Go, son declaradas después de la estructura, en el siguiente cuadro de código se muestra la creación de nuevo método llamado *display*.

```

PHP
12 //función que imprime los valores del objeto
13 function display(){
14     echo $this->atributo1;
15     echo $this->atributo2;
16 }

```

En la segunda línea del bloque de código anterior, se declara la función *display*, la cual no acepta ningún parámetro, los siguientes dos renglones imprimen los dos atributos que contiene la clase. Se usa nuevamente la palabra reservada *this*, para hacer referencia a la clase en la que se encuentra.

Como se mencionó anteriormente, los métodos son funciones que realizan acciones sobre los objetos, en el lenguaje de programación Go, los métodos se declaran afuera de la estructura. En el siguiente bloque del cuadro de código posteriormente mostrado, describe la forma en que se crea un método en lenguaje Go.

```

Go
7 //función que imprime los valores del objeto
8 func (this NuevaClase) display() {
9     fmt.Println(this.atributo1, this.atributo2)
10 }

```

Primero se define una función, esta función se declara diferente a lo ya mencionado en el Capítulo anterior, en este caso se declara una función asociada. En el ejemplo se escribe *func(this NuevaClase) display()*, después de la palabra reservada *function* se le agrega un receptor. Este parámetro es del tipo de la estructura a la que se quiere incluir el método, seguido del nombre de la función, parámetros y si se requiere, tipo de variable de retorno. Finalmente, en la línea 9 del bloque de código se imprimen los atributos que contiene la estructura.

Para llamar a un método de una clase en el lenguaje de programación PHP se



crea una sentencia que incluye el nombre del objeto, el operador de clases (->) y el nombre del método que se requiere llamar. En este caso el método *dysplay()* no cuenta con ningún atributo.

```
PHP
17 $nuevoObjeto->dysplay();
```

Para llamar un método en el lenguaje de programación Go se crea una sentencia que incluye el nombre del objeto, el operador de concatenación (.) y el nombre del método que se requiere llamar. En este caso el método *dysplay()* no cuenta con ningún atributo. La llamada al método para crear un nuevo objeto se muestra en la siguiente línea de código.

```
Go
11 nuevoObjeto.display()
```

Si se requiere invocar a un método heredado de una estructura padre se invocará de la misma forma que se mostró en la línea anterior ya que en el lenguaje de programación Go la composición incluye métodos y variables.

La diferencia en la forma de llamar los métodos en ambos lenguajes de programación es el operador de concatenación, mientras que en PHP se usa el operador clásico de POO (->), en el lenguaje Go se usa un operador de concatenación para hacer uso de métodos.

La herencia es cuando una clase se basa en otra utilizando la misma forma. Existen dos implementaciones diferentes de la herencia. Herencia simple, que es donde solo se puede heredar una sola clase. Y herencia múltiple, donde se puede heredar las características de varias clases base.

Algunos lenguajes con herencia simple son: PHP, C#, Java y Ruby.

Algunos lenguajes de herencia múltiple son: Perl, Python y C ++ [16].

La herencia representa la relación “es un” o “es una”. Por ejemplo: como se ve en la *Figura 6*, un mamífero **es un** Animal. Un humano **es un** mamífero, una serpiente **es un** reptil.

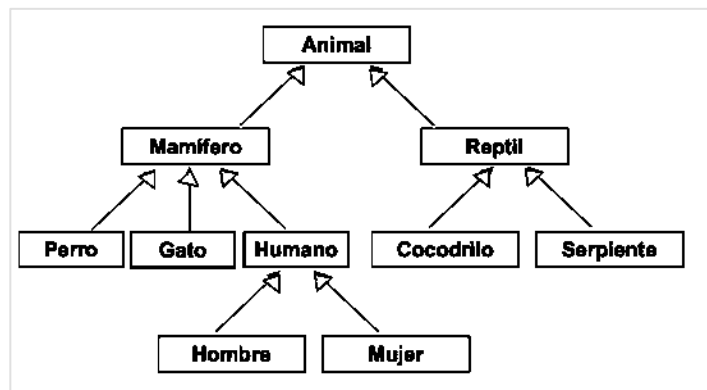


Figura 6 Herencia de clases

La composición es una relación que representa a objetos compuestos por otros objetos. El objeto en el nivel superior de la jerarquía es el todo y los que están en los niveles inferiores son sus partes o componentes. Un componente es parte esencial de una entidad. La relación es fuerte al punto que si los componente son eliminados o desaparecen, la clase mayor deja de existir.

La composición representa la relación “tiene”. Por ejemplo, la entidad mostrada en la *Figura 7* puede ser descrita en la siguiente oración. La casa **tiene** 1 baño y la habitación **tiene** 5 ventanas.

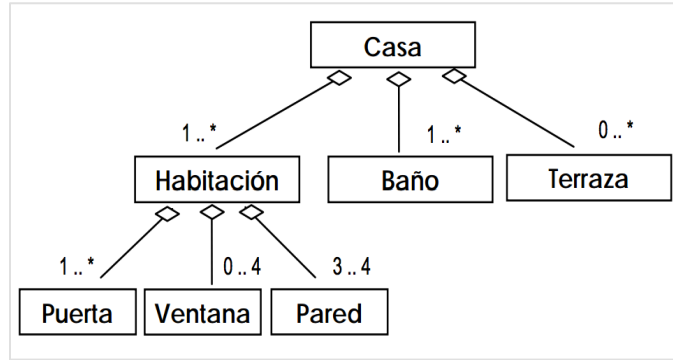


Figura 7 Composición de objetos

En el lenguaje de programación Go existen los tipos embebidos dentro de otros tipos para obtener algo análogo a la herencia y la composición. Los métodos de Go son más generales que los de C++ o Java, pueden ser definidos para cualquier tipo de datos, no solo para los *structs*.

En PHP la herencia se genera con la palabra reservada *extends*, mientras que Go utiliza la composición asemejándose a la herencia, ya que cuando se agrega un tipo dentro de otro tipo, también se le están asignando los métodos del tipo agregado.

```

18 Class ClaseHija extends NuevaClase {
19     //atributos
20     var $atributo3
21 }
  
```

En el bloque de código anterior, se muestra la herencia en PHP. Recordando la clase de nombre *NuevaClase* declarada anteriormente, la segunda línea de código crea una *ClaseHija*, esta clase hereda los atributos y métodos de la clase llamada *NuevaClase* dado que existe la palabra *extends* entre ambos nombres de las clases.

En el siguiente cuadro de código se muestra la forma en que Go crea una

estructura llamada *ClaseHija*, la cual a su vez se compone por la clase base llamada *NuevaClase*. La diferencia es que Go utiliza la composición y aquí a diferencia de lo tradicional, la composición se incluyen atributos y métodos de las clases base.

```
Go
13 type ClaseHija struct {
14     NuevaClase
15     //atributos
16 }
```

El polimorfismo se compone de dos palabras. Poli, que significa múltiple, morfismo, que se refiere a varias formas. Traducido como un objeto puede tomar diferentes formas. El polimorfismo es la capacidad que tienen los objetos de una clase para responder al mismo mensaje o evento en función de diferentes parámetros utilizados durante su invocación [14].

Existen varios tipos de polimorfismo, en esta investigación y solo para fines ilustrativos para diferenciar ambos lenguajes se incluyo el polimorfismo de inclusión también llamado de redefinición o subtipado.

Primero se crea una clase abstracta como base, clase llamada *Animal*, las dos clases siguientes heredan a la clase base y ambas clases contienen un método igual, llamado *Speak()*. Para crear polimorfismo en PHP, se crea un método general que contiene como parámetro a un tipo clase base, tipo *Animal*. Esta función es la interfaz para que cada objeto, llame a su método por la misma función pero con diferente resultado. Las llamadas a los métodos se observan en las líneas 23 y 24.

```
1 <?php
2 class Animal{}
3
4 class Perro extends Animal{
5     function Speak(){
6         print "guau guau\n";
7     }
8 }
9
10 class Gato extends Animal{
11     function Speak(){
12         print "miau miau\n";
13     }
14 }
15
16 function speak(Animal $obj){
17     $obj->Speak();
18 }
19
20 $p = new Perro();
21 $g = new Gato();
22
23 speak($p);
24 speak($g);
25
26 ?>
```

En Go la clase abstracta es una *interface* que tiene como parámetro el nombre del método que todas las estructuras posteriores en el siguiente bloque de código, se refiere al método *Speak()*. A partir de la línea 9 se crean dos nuevas estructuras (clases) y sus funciones asociadas (métodos).

De forma similar a PHP, en Go se puede crear una función que permita la generalización de todos los métodos de las clases mediante uno solo. En la línea 25 crea una función que recibe como parámetro un tipo interfaz *Animal*, su funcionamiento es equitativo al código mostrado anteriormente en PHP, esta función es la interfaz para que cada objeto llame a su método mediante la misma función, mostradas en las siguientes líneas.

```
Go
1 package main
2
3 import "fmt"
4
5 type Animal interface {
6     Speak()
7 }
8
9 type Perro struct {
10     Animal
11 }
12
13 func (Perro) Speak() {
14     fmt.Println("guauuuuu")
15 }
16
17 type Gato struct {
18     Animal
19 }
20
21 func (Gato) Speak() {
22     fmt.Println("miauuuuu")
23 }
24
25 func speak(obj Animal) {
26     obj.Speak()
27 }
28
29 func main() {
30
31     p := Perro{}
32     g := Gato{}
33     speak(p)
34     speak(g)
35 }
```

La *Tabla 1* explica la forma en que se escriben las tres bases tomadas de programación orientada a objetos en esta investigación.

En programación a orientada a objetos existen 2 técnicas pilares para construir clases a partir de otras ya existentes, estas dos técnicas son la herencia y la composición. En el diseño y programación orientada a objeto se debe decidir cual de estos dos tipos usar en las relaciones entre las clases que modelas las entidades del mundo real. Se debe comprender que ambos tipos de relaciones pueden existir al mismo tiempo en un problema en donde hay varias clases y diversos tipos de relaciones [18].

**Tabla 1 Definiciones de POO en Go**

Go	PHP	Definición en POO Clásico
<i>struct</i>	class	clase con campos, únicos métodos no virtuales
<i>interface</i>	class abstract	clase sin campos, sólo los métodos virtuales
<i>embedding</i>	extends / composición	herencia múltiple y la composición

Para poder aprender acerca de un nuevo lenguaje de programación es necesario darle una vista a su historia y saber sobre su documentación.

En este capítulo se presentó una recopilación de historia, así como un manual del lenguaje de programación Go. Si se quiere conocer más de partes técnicas, consultar la bibliografía de esta investigación y/o entrar al sitio oficial <http://golang.org>.

El próximo capítulo realiza una comparación entre ambos lenguajes. Las estadísticas en internet que tiene cada uno de estos lenguajes, su comportamiento entre los desarrolladores del mundo, un *benchmark* poniendo a prueba sus características y finalmente se desarrolla una comparación de varios algoritmos en ambos lenguajes.

## CAPÍTULO 4

### ESTADÍSTICAS Y COMPARACIÓN

Este capítulo contiene un conjunto de comparaciones entre el lenguaje Go y el lenguaje PHP. Comienza con una estadística donde se describe el porcentaje de actividad respecto a países y continentes, actividad en grupos, usuarios activos, publicaciones de ambos lenguajes, dominios y proyectos que se desarrollaron en los dos lenguajes comparados en esta investigación. Posteriormente se analiza ambos lenguajes de manera más técnica, incluyendo las ventajas y desventajas de cada lenguaje. Por último se incluye un *benchmark* donde se compara el rendimiento de ambos lenguajes y se anexa un código en ambos lenguajes donde se probó de manera práctica lo que aquí se menciona.

Cada generación se adapta a un lenguaje de programación distinto, así como cada generación sabrá apoyar o inclinarse hacia un lenguaje diferente desde puntos de vista meramente personales, laborales o técnicos a los que se dedique.

Go es un lenguaje formulado por uno de los gigantes de la computación en estos años. Go, se plantea como uno de los lenguajes del futuro que tendrá más auge en el desarrollo.

En un inicio PHP fue un proyecto para tener un curriculum personal que se llamaba *Personal Home Page* basado en Perl. Después la librería estándar se creó con parámetros de C, y las partes orientadas a objetos se diseñaron con una combinación de C++ y Java (Holzner, 2009). PHP carece de orden en el lenguaje, haciendo que el desarrollo pueda contener partes repetidas, inutilizadas y otros detalles [19]. PHP es un lenguaje que se ejecuta en el servidor y cada que se pide una petición por el cliente, los códigos PHP no son visibles para los clientes. PHP es un lenguaje que es criticado por muchos; una crítica que sobresale es que no



tiene una estructura real de lenguaje, ya que su idea inicial nunca fue ser un lenguaje de programación.

En las universidades del mundo se enseña C, Java o en algunos Python como lenguajes principales, pero C es demasiado complejo, Python es comparablemente lento y Java demasiado voluminoso. Go desempeña el papel que tradicionalmente pertenecía a PHP, pero mucho más rápido, simple, seguro y fácilmente escalable, y a la vez se pretende ser un lenguaje de base educativa [20].

## **4.2. Estadísticas de actividad en internet**

El internet es la fuente más importante para recopilar estadísticas sobre el comportamiento de ambos lenguajes brindándonos características específicas y reales sobre los dos lenguajes.

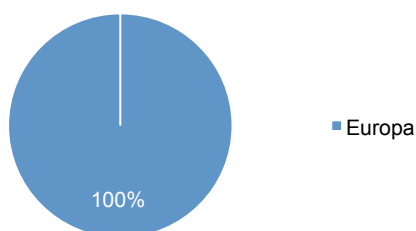
### **4.2.1. Comportamiento y estadísticas de ambos lenguajes en la red**

#### *4.2.1.1 Estadísticas en español.*

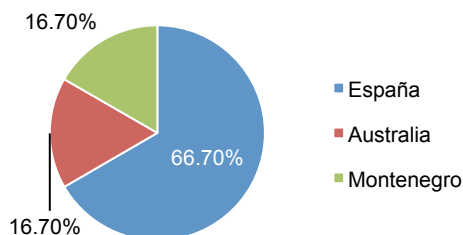
Hoy en día, las compañías como Google registran todos los movimientos que realizamos en la red. Esto produce una inmensa cantidad de datos, donde se pueden comparar el comportamiento de los usuarios y de los programas. esta información es la base para poder medir estadísticas en la actualidad. Con esta información se conoce el número de personas interesadas en buscar algún tema en específico, así como también se tiene el número de dominios que se crearon y se consultaron sobre estos lenguajes, el número de grupos de desarrolladores y datos semejantes.

Los datos del lenguaje de programación Go en español crearon una máxima punta de interés a mediados del 2013 y principios del 2014. Se percibió que muchos grupos de desarrolladores Mexicanos en Golang, están parcial o totalmente abandonados.

El interés sobre el lenguaje de programación Go que existe al número de usuarios en español se centra a su 100% en el continente Europeo, mostrado en las Figuras 8 y 9 [21].



**Figura 8** Porcentaje de continentes interesados en Go español

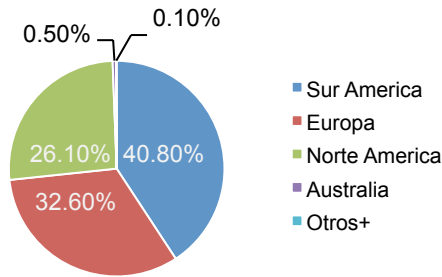


**Figura 9** Porcentaje de países interesados en Go español

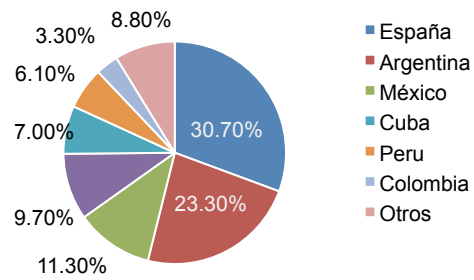
El que no existan numerosos sitios o grupos en español no dice que no existen hispanos que estén interesados en este lenguaje, como es un lenguaje relativamente nuevo, no existen tantas traducciones.

En el lenguaje de programación Go existen muy pocos usuarios registrados en español a comparación del lenguaje de programación PHP. Ya que PHP es un lenguaje de mucho uso escolar y de aprendizaje, de igual manera el gran número de desarrolladores que existen en todos los continentes hace posible tener numerosas traducciones a varios idiomas, entre ellos el español. En la Figura 10 y 11, se muestran las estadísticas sobre PHP en español.

En la Figura 11, se menciona a México como el 3er país con más usuarios activos en grupos relacionados a PHP. Estas gráficas solo ilustran el peso de usuarios hispanos en ambos lenguajes, pero como se mencionó anteriormente, PHP no se puede comparar equitativamente en el número de usuarios de Go, porque Go tiene en existencia el 10% del tiempo que tiene PHP en el mercado.



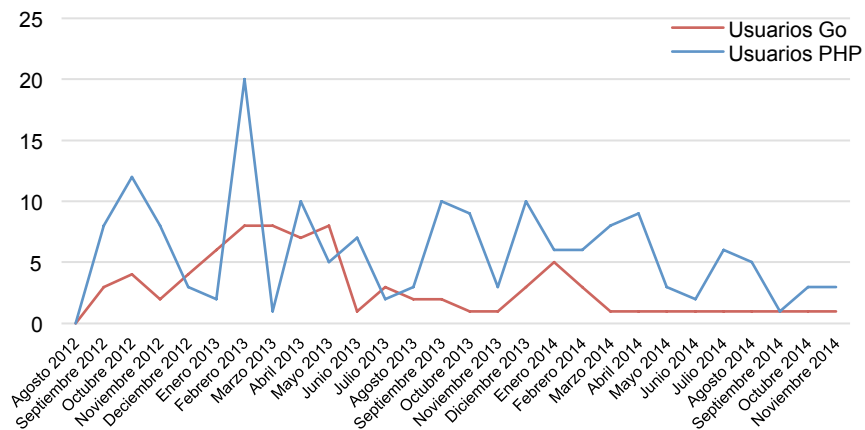
**Figura 10 Porcentaje de continentes interesados en PHP español**



**Figura 11 Porcentaje de países interesados en PHP español**

Después del lanzamiento del lenguaje Go, muchos programadores empezaron a conocerlo convirtiéndose en los nuevos usuarios en las estadísticas. Pero fue hasta inicios del 2013 que se disparó el número de desarrolladores y usuarios interesados en este lenguaje.

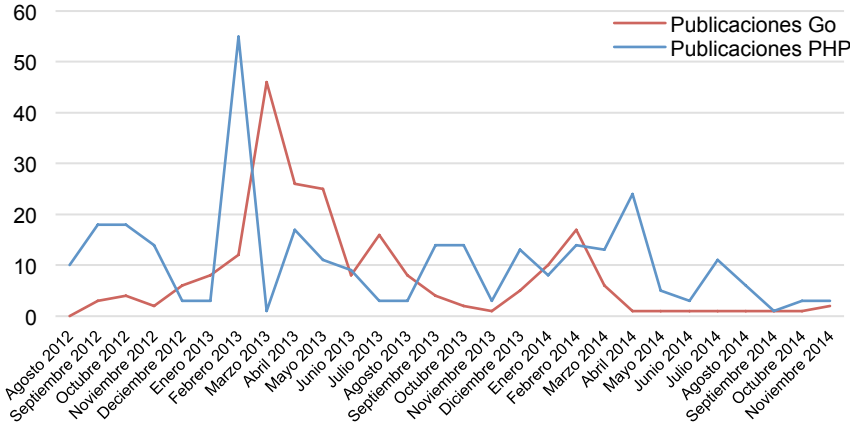
Como se muestra en la *Figura 12*, el interés de los usuarios nuevos en el lenguaje de programación Go en español es casi nulo con un pico máximo de 8 usuarios en Febrero, Marzo y Mayo del año 2013, después de esto decayó bastante hasta principios del año 2014 donde en el mes de Enero se volvieron a incluir 5 usuarios más como parte de este grupo hispano [21].



**Figura 12 Usuarios en español por mes**

Tomando los últimos dos años de ingresos de nuevos usuarios en el lenguaje de programación PHP, se aprecia en la *Figura 12* que PHP siempre estuvo por encima con el número de usuarios interesados sobre Go. El promedio de ingreso mensual fue el doble. Mientras que el lenguaje de programación Go promedio 3 usuarios nuevos por mes, PHP promedio 6 nuevos usuarios por mes durante estos dos últimos años.

En la *Figura 13* se muestran las estadísticas respecto a las publicaciones mensuales en ambos lenguajes. El pico de interés por el lenguaje de programación Go en zonas de habla hispana se dio a mediados del año 2013 logrando un máximo de 46 publicaciones realizadas en marzo 2013, a principios del año 2014 las estadísticas por el interés de hispanos en grupos en español sobre el lenguaje de programación Go comenzó a incrementar alcanzando un total de 17 publicaciones en el mes de febrero [21, 22].

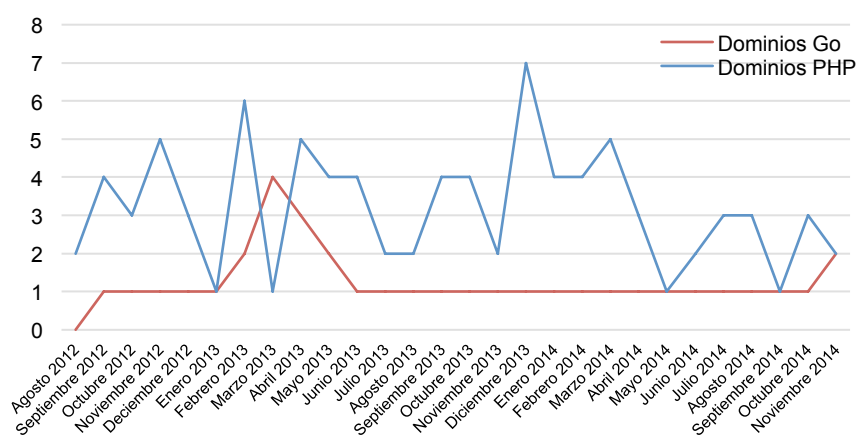


**Figura 13 Publicaciones en español por mes**

El total de publicaciones registradas sobre el lenguaje de programación Go hasta finales de 2014 alcanzó un total de 180, mientras que el número de publicaciones en el lenguaje de programación PHP este par de años de 290, nuevamente el

lenguaje de programación PHP se muestra por encima de Go en las estadísticas de publicaciones realizadas en español.

Las estadísticas de ambos lenguajes respecto a los dominios creados con cada lenguaje y para la difusión de los mismos, mostrados en la *Figura 14* vuelven a mostrar que el lenguaje de programación PHP supera a Go con 55 sitios generados.



**Figura 14 Dominios en español por mes**

Todos los datos estadísticos de ambos lenguajes de programación graficados anteriormente se muestran de una manera más legible en la *Tabla 2*. Go carece de información en algunos meses en este periodo.

México es un país donde no estamos al día con las nuevas tendencias tecnológicas. Desgraciadamente, a lo largo de la investigación no se pudo localizar en Michoacán ninguna empresa o persona que desarrolle en este lenguaje. Sin embargo, la búsqueda se expandió a estados vecinos, donde se encontró la comunidad Go de Guadalajara, Jalisco que sirvió de ayuda en esta investigación.

**Tabla 2 Estadísticas generales en español**

Período		Publicaciones		Nuevos Usuarios		Nuevos Dominios	
		PHP	Go	PHP	Go	PHP	Go
Noviembre	2014	3		3		2	
Octubre	2014	3	1	3	1	3	1
Septiembre	2014	1	1	1	1	1	1
Agosto	2014	6		5		3	
Julio	2014	11		6		3	
Junio	2014	3		2		2	
Mayo	2014	5	1	3	1	1	1
Abril	2014	24	1	9	1	3	1
Marzo	2014	13	6	8	1	5	1
Febrero	2014	14	17	6	3	4	1
Enero	2014	8	10	6	5	4	1
Diciembre	2013	13		10		7	
Noviembre	2013	3	1	3	1	2	1
Octubre	2013	14		9		4	
Septiembre	2013	14		10		4	
Agosto	2013	3		3		2	
Julio	2013	3	16	2	3	2	1
Junio	2013	9	8	7	1	4	1
Mayo	2013	11	25	5	8	4	2
Abril	2013	17	26	10	7	5	3
Marzo	2013	1	46	1	8	1	4
Febrero	2013	55	12	20	8	6	2
Enero	2013	3		2		1	
Diciembre	2012	3		3		3	
Noviembre	2012	14	2	8	2	5	1
Octubre	2012	18	4	12	4	3	1
Septiembre	2012	18	3	8	3	4	1
Total		290	180	165	58	88	24

En todos los meses que se muestran en la *Tabla 2*, el lenguaje de programación PHP está por encima de Go. Es de esperar que el lenguaje de programación Go aún está en etapas de inicio y más si se trata de traducciones.

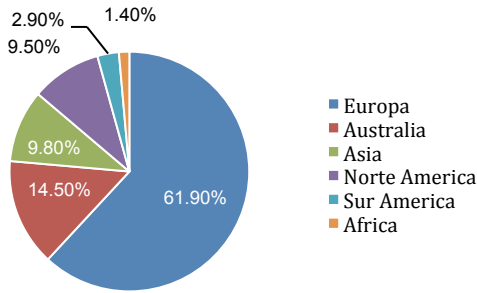
#### 4.2.1.2 Estadísticas generales

Gracias a la existencia de estos sitios web especializados a comparar el flujo de datos respecto a tendencias tecnológicas, se puede realizar una comparación más equitativa entre el lenguaje de programación Go y el lenguaje de programación PHP [21]. Se tomaron datos de ambos lenguajes de 6 tópicos de interés: localidad, grupos creados, usuarios, publicaciones, número de dominios y proyectos realizados.

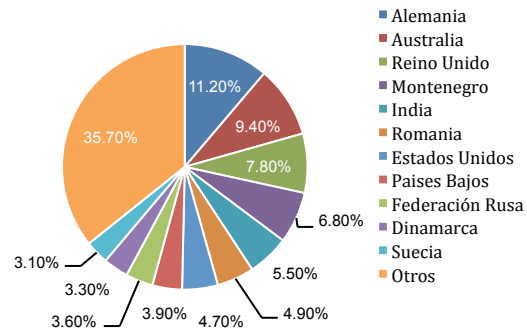
En esta investigación se plasmó una estadística respecto al tráfico en Internet del desarrollo, evolución y alcance actual de los dos lenguajes de programación mencionados en esta investigación. El lenguaje de programación PHP contiene una base de estadísticas en *Grokbase* donde se tiene almacenado un poco más de 15 años de información sobre este lenguaje.

En un principio se pensó en tomar los primeros años del lenguaje de programación PHP y de esta manera realizar una comparación de ambos lenguajes en sus periodos iniciales. Esta propuesta se descartó ya que las formas de comunicación, la expansión del lenguaje y el interés en la rama de computación hace 20 años son muy diferentes a hoy en día. Por este motivo se tomarán de base el período de tiempo de 2 años que es el período que lleva hasta ahora el lenguaje de programación Go.

El interés de usuarios sobre el lenguaje de programación Go mostrado en las *Figuras 15 y 16* posiciona al continente Europeo como el continente que contiene el mayor porcentaje de interés en el lenguaje de programación Go y en países como Alemania e India las estadísticas los enlistan como los países con más interés [21].



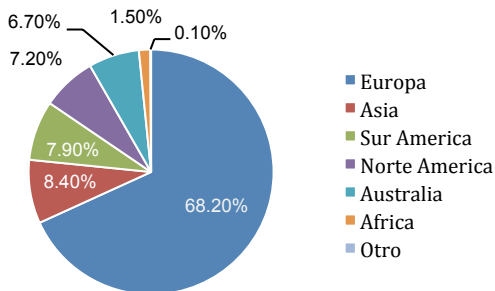
**Figura 15** Porcentaje de interés de Go en continentes



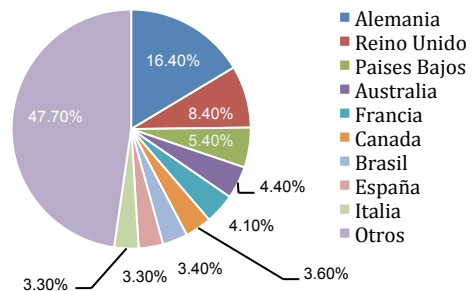
**Figura 16** Porcentaje de interés de Go en países

De igual manera se graficaron las estadísticas globales del lenguaje de programación Go, en las Figuras 17 y 18 se muestran las estadísticas de interés globales del lenguaje PHP. En las dos siguientes figuras se observa que el continente Europeo encabeza la lista de continentes con un mayor porcentaje de interés y consulta sobre el lenguaje de programación PHP.

También, igual que las estadísticas con Go, Alemania, es el país con un mayor interés en discusiones respecto al lenguaje de programación PHP.



**Figura 17** Porcentaje de intereses de PHP en continentes



**Figura 18** Porcentaje de intereses de PHP en países



La contribución de México es tan pequeña que se encuentra agrupada en el porcentaje de *otros*. Aparentemente no existen gran variedad de grupos ni temas de discusión liberados por desarrolladores mexicanos, la información en computación esta generalmente en el idioma inglés, causando la concentración de publicaciones y usuarios latinos e hispanos en grupos de Norteamérica y Europa.

Los grupos son una herramienta que te permiten establecer contacto con otros usuarios de Internet y generar comunicación sobre tópicos diversos.

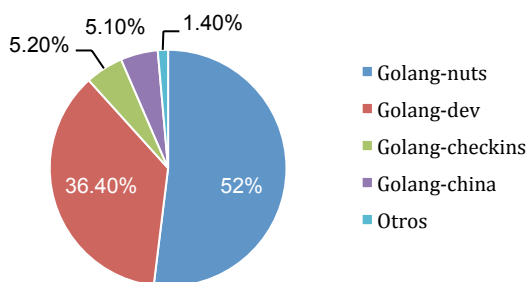
Los grupos también funcionan como una fuente importante de información a la hora de realizar consultas y buscar asesoramiento de primera mano. En ambos lenguajes existen abundantes grupos de discusión.

Go actualmente cuenta con un total de 10 grupos y un poco más de 138 mil usuarios activos en los grupos de desarrollo y de preguntas de este lenguaje, datos descritos en la *Tabla 3* [21].

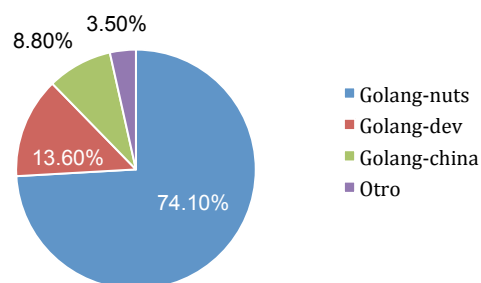
**Tabla 3 Estadísticas generales de los grupos de Go**

<b>Grupos</b>	<b>Publicaciones</b>	<b>Usuarios</b>	<b>Dominios</b>	<b>Publicaciones / Usuarios</b>	<b>Publicaciones / Dominio</b>
golang-nuts	70553	4969	981	14.2	71.9
golang-dev	50741	880	196	57.7	256.3
golang-checkins	7396	2	2	3698	3698
golang-china	6959	608	33	11.5	210.9
golang-ru	1293	130	13	9.9	99.5
goclipse	266	39	10	6.8	26.6
golang-spanish	180	29	5	6.2	36
golang-korea	41	17	3	2.4	13.7
golang-goweb	29	11	5	2.6	5.8
golang-weekly	6	2	2	3	3

De los grupos que hablan sobre Go solo sobresalen 4 de ellos. Como se muestra en las Figuras 19 y 20. *Golang-nuts* es el grupo con más usuarios y por tanto mayor actividad, teniendo el 74.1% del total de los usuarios integrados a grupos, y abarca más de la mitad en publicaciones y temas de discusión, con un 52% del total de todas las publicaciones incluyendo todos los grupos.



**Figura 19** Porcentaje de publicaciones en los grupos de Go



**Figura 20** Porcentaje de usuarios en los grupos de Go

*Golang-nuts* y *Golang-dev* generan temas de discusión sobre preguntas y desarrollos Go.

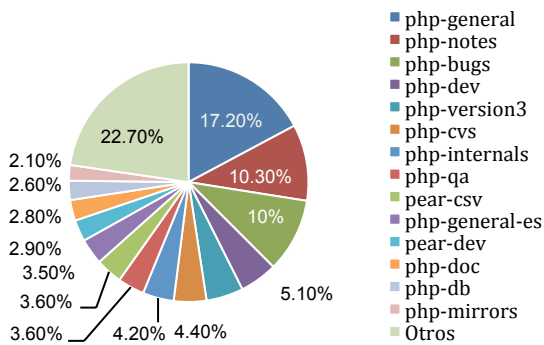
La razón porque China tiene su grupo como el cuarto grupo de más actividad y el primer grupo del país con más movimiento sobre temas del lenguaje de programación Go en el mundo, es que existe mucho interés por el lenguaje de Google en estas ciudades asiáticas. Más adelante se describe la historia de Go en China.

El lenguaje de programación PHP logra tener el total de 133 grupos que sobrepasan por más de 10 veces al lenguaje de programación Go. Podemos decir que Go está en una etapa aun de iniciación, comparándolo en base el tiempo que tienen de vida del lenguaje de programación PHP. En la *Tabla 4*, se enlistan los 12 más grupos con mayor número de usuarios y publicaciones.

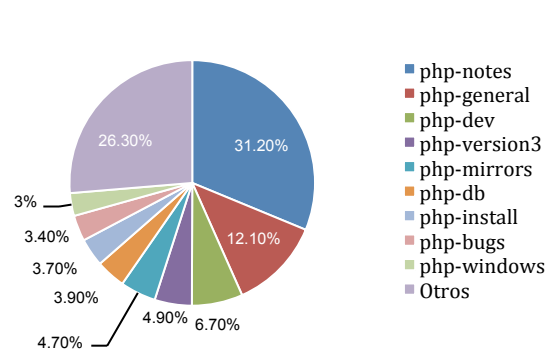
**Tabla 4 Estadísticas generales de los 12 grupos más importantes de PHP**

Grupos	Publicaciones	Usuarios	Dominios	Publicaciones / / Usuarios	Publicaciones / Dominio
<a href="#">php-general</a>	323,747	25,558	15,883	12.7	20.4
<a href="#">php-notes</a>	194,225	65,853	24,982	2.9	7.8
<a href="#">php-bugs</a>	187,922	7,067	4,886	26.6	38.5
<a href="#">php-dev</a>	95,075	14,083	10,020	6.8	9.5
<a href="#">php-version3</a>	93,615	10,365	7,925	9	11.8
<a href="#">php-cvs</a>	82,018	853	490	96.2	167.4
<a href="#">php-internals</a>	78,364	4,055	2,222	19.3	35.3
<a href="#">php-qa</a>	67,440	1,910	1,342	35.3	50.3
<a href="#">pear-cvs</a>	67,232	3,094	1,760	21.7	38.2
<a href="#">php-general-es</a>	65,531	3,789	2,025	17.3	32.4
<a href="#">pear-dev</a>	55,391	2,712	1,797	20.4	30.8
<a href="#">php-doc</a>	52,682	3,446	2,062	15.3	25.5

En las *Figuras 21 y 22* se ven los grupos más importantes de discusión respecto a temas PHP. En PHP, el grupo con más audiencia y flujo de información es el llamado PHP-general que es un grupo de preguntas enfocadas en el uso del lenguaje.



**Figura 21** Porcentaje de publicaciones en los grupos de PHP

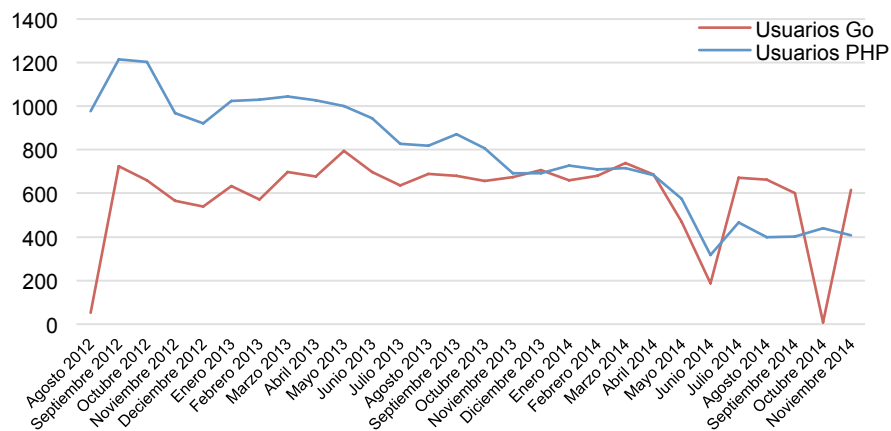


**Figura 22** Porcentaje de usuarios en los grupos de PHP

Los grupos de asistencia sobre el desarrollo y mantenimiento del lenguaje PHP tienen un menor flujo de información comparándolos con los grupos del lenguaje de programación Go, dado que el lenguaje de programación Go es nuevo, tiene muchas vertientes de desarrollo de nuevas librerías y soporte a nuevas versiones.

Las estadísticas sobre el ingreso de nuevos interesados en el lenguaje de programación Go, ha tenido una estabilidad desde principios del 2013, excepto por una caída enorme de los meses de marzo a junio de este año donde bajo de 739 usuarios e ingreso 186 nuevos usuarios para el mes de junio.

La media de usuarios en este lenguaje tiende a 606 usuarios por mes, esto se puede ver en la *Figura 23*. En mayo 2013, Go alcanzó su mayor auge hasta la fecha, con un total de 726 nuevos desarrolladores.



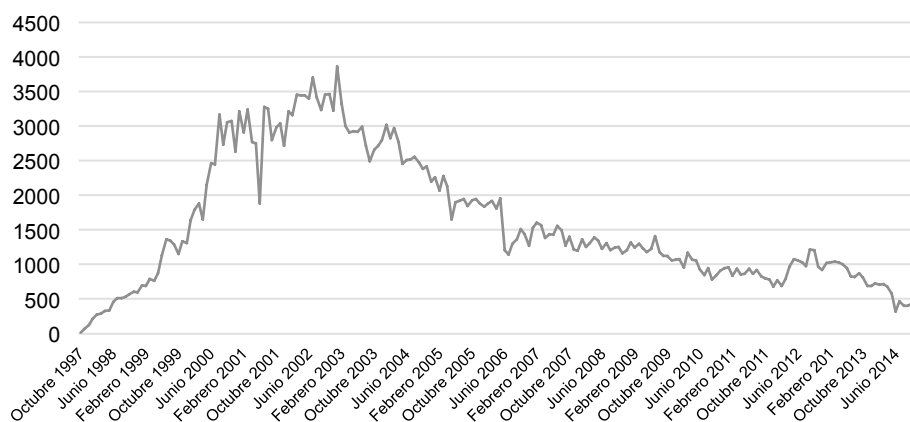
**Figura 23 Usuarios generales en ambos lenguajes**

En estos dos últimos años de estadísticas, el lenguaje de programación PHP ha perdido una afluencia considerable de nuevos usuarios registrados en las estadísticas consultadas y mostradas en la *Figura 23*. Se debe de mencionar y tomar en cuenta que desde el año 2012 hasta el año 2014 PHP ya contaba con un

número muy grande de personas que conocían y desarrollaban en este lenguaje y por consecuencia el número de nuevos interesados registrados en las estadísticas consultadas son más bajos que años pasados [21].

La media de usuarios del lenguaje de programación PHP es de 773, solo 167 usuarios más que la media de usuarios del lenguaje de programación Go.

Los datos completos respecto al ingreso de nuevos usuarios en grupos sobre el lenguaje de programación PHP desde su inicio al año 2014 están graficadas en la *Figura 24* y muestra que el interés sobre PHP ha decaído en más de 50%; esto suceso empezó entre marzo y abril de 2003 hasta el mes de noviembre de 2014. En noviembre de 2014, último mes donde se capturaron estadísticas, PHP incremento solo 72 nuevos usuarios.



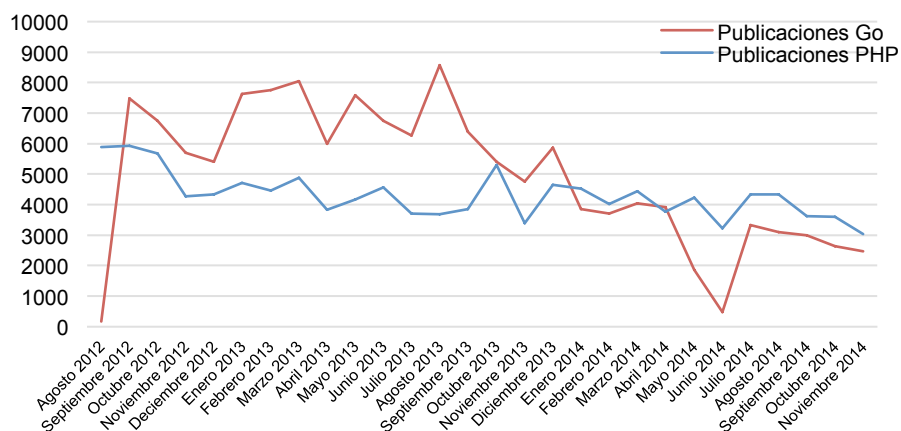
**Figura 24 Usuarios totales en PHP**

Respecto a las publicaciones totales del lenguaje de programación Go, se puede observar que el máximo número de publicaciones se realizó en agosto del 2013 con un total de 8,575 publicaciones. A partir de ahí, las publicaciones en el mayor número de meses empezaron a descender hasta junio 2014 donde hubo solo 472 publicaciones.

La media en estos dos años de publicaciones del lenguaje de programación Go realizadas en los grupos y/o páginas registradas con los dominios afines a este tema es de 4,909. Las estadísticas de actividad en los grupos, las publicaciones sobre el lenguaje de programación Go, bajaron en un 75% desde septiembre de 2012.

El lenguaje de programación PHP promedió en 4460 sus publicaciones para estos dos últimos años quedando por 449 puntos debajo del promedio que tiene el lenguaje de programación Go en estos dos últimos años. Después de septiembre de 2012, las publicaciones no volvieron a subir más de 5304, esto en octubre de 2014.

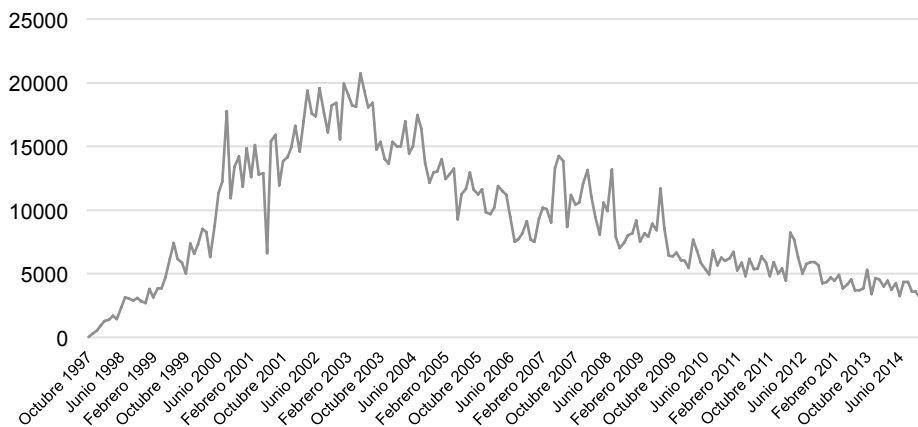
En el último mes del que se tiene datos en esta estadística, el nivel de publicaciones sobre temas del lenguaje de programación PHP suman un total de 3,027 publicaciones obtenidas en el mes de noviembre de 2014. A pesar de estas estadísticas PHP se posiciono con 2,465 publicaciones más que las obtenidas en este mes en el lenguaje de programación Go.



**Figura 25 Publicaciones generales**

El lenguaje de programación PHP se mantiene estable con el número de publicaciones desde noviembre hasta septiembre de 2014, donde las publicaciones sobre este lenguaje empiezan a descender. Go, por otro lado no a podido tener una estabilidad, pero es notable en la *Figura 25* que en los primeros dos años de inicio del lenguaje de programación Go estuvo por encima de PHP en el número de publicaciones.

El número total de publicaciones que se tienen en la base de datos sobre las estadísticas del lenguaje de programación PHP reflejan el mismo patrón de las *Figuras 23 y 24*. Después del mes de abril de 2003 PHP tuvo una notable baja en las publicaciones mostrada en la *Figura 26*.



**Figura 26 Total de publicaciones en PHP**

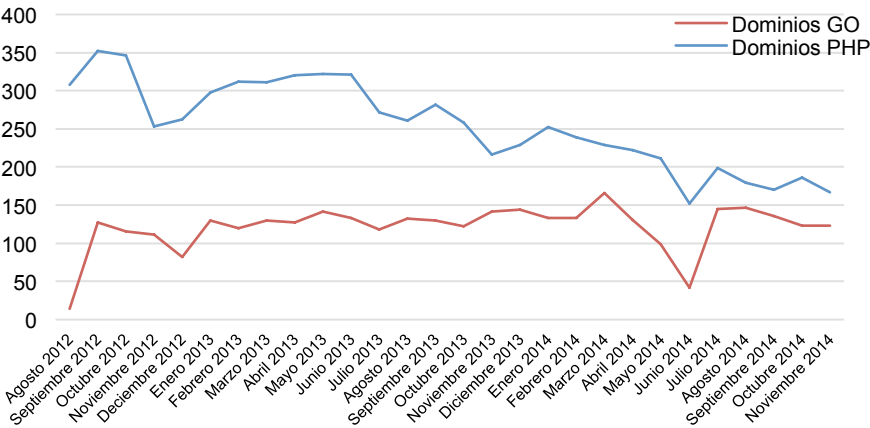
Un dominio de Internet es una identificación más fácil de memorizar asociada a uno o un grupo de dispositivos conectados a una red, los cuales contienen una dirección IP. Los dominios cuentan con extensiones para campos distintos y con abreviaciones específicas para cada país llamada CCtld<sup>18</sup>. Los dominios

<sup>18</sup> Country code-top-level-domain, dominio de nivel superior geográfico. Es el indicador geográfico de los dominios. Abreviación de letras o palabras para referirse a cada país.

registrados son una parte fundamental para saber la cantidad de sitios que se alojan para discusión, desarrollo, investigación y/o dialogo sobre ambos lenguajes.

Las estadísticas de un lenguaje se basan, entre otras cosas, en saber cuantos dominios existen activos y cuantos se están creando en los últimos meses. El promedio máximo de dominios en el lenguaje de programación Go mostrados en la *Figura 27* ha sido alcanzado en marzo 2014, con un total de 166 dominios nuevos abiertos ese mes.

Teniendo estas estadísticas de base se puede calcular la media de los dominios nuevos en este período, dando un valor de 121.6 dominios en Go.



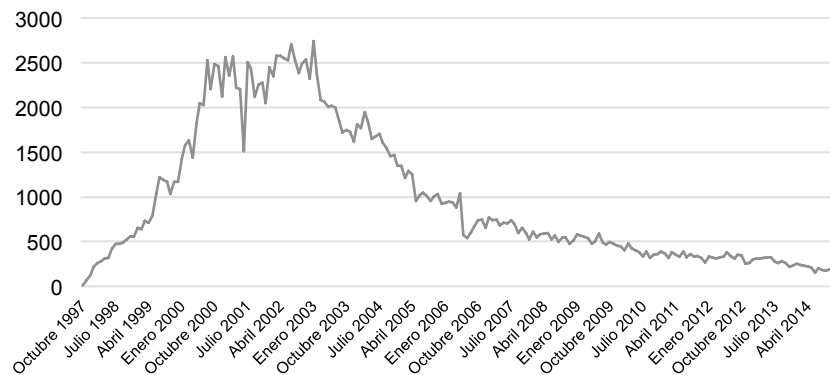
**Figura 27 Dominios con Go**

En el ambiente general hubo una pérdida de interés de abril a junio del 2014 bajo en un 70% la creación de sitios sobre el lenguaje de programación Go, después tuvo un intentó de ascenso en el mes de agosto, pero en los últimos meses del 2014 volvió a decaer el interés sobre Go [21]. Se tienen registrados 18,861 dominios creados con y/o para temas del lenguaje de programación Go [21].



El promedio de dominios creado en estos dos años por el lenguaje de programación PHP es de 264 dominios por mes, el cual le gana en un 45% al promedio mensual que tiene el lenguaje de programación Go.

PHP es un lenguaje con mucho tiempo en la red, quizá a estas alturas ya no se traten ni la décima parte de las publicaciones que se tenían en los años 90's. (w3School, 2014).



**Figura 28 Dominios en PHP**

Como se muestra en la *Figura 28*, en el mes de marzo del 2003, los desarrolladores y público aficionado empezó a dejar de interesarse por crear páginas de información y grupos de discusión respecto al lenguaje de programación PHP.

Una forma legible de presentar todas las estadísticas graficadas anteriormente de ambos lenguaje en estos dos años es anexar todos los datos en la *Tabla 5*.

#### 4.2.1.3 Desarrollo de proyectos

PHP nació en el año de 1995, desarrollado por Lerdorf, del cual se han registrado movimientos desde 1997. Número uno en el desarrollo Web y convertido en el estándar de desarrollo de sitios Web dinámicos con bases de datos [23].

PHP es el lenguaje más popular para aplicaciones web, aproximadamente el 82.2% de los sitios web están basados en PHP [24]. pero relativamente lento y no adecuado para sistemas grandes. Existe una gama impresionante de herramientas para el desarrollo web; esto hace imposible conocer todo el conjunto. En el año 2003 se registraron más de 200 millones de sitios web y proyectos realizados con el lenguaje de programación PHP [25], es por esta razón que no es posible enlistar todos los proyectos de PHP realizados hasta la fecha actual.

El lenguaje de programación Go se creó para solventar las necesidades y problemáticas internas que tenía Google; problemáticas que no eran sencillas. El uso de Go empieza a hacer cada vez mayor dentro de Google y esto ha dado pie al uso de este lenguaje en grandes empresas y/o proyectos nuevos.

El primer programa escrito en Go fue escrito por Rob Pike, en febrero de 2008. En él se transcribió código que ya tenían en el lenguaje de programación C. Este código sirvió para hacer pruebas de la primera versión de Go y su compilador.

Los proyectos desarrollados en Go, mostrados en la *Tabla 6*, registran un gran número en su página oficial. Todos estos proyectos, así como los que no están documentados, se encuentran en su mayoría alojados en *Github*, proyectos *OpenSource* y privados. Esta lista de proyectos solo es la punta del iceberg. El número de proyectos *OpenSource* es asombrosa, es como Python en 1995, Java en 1999 o C++ en 1987 [26].

**Tabla 5 Estadísticas generales**

Período		Publicaciones		Usuarios		Dominios		Publicaciones / Usuario		Publicaciones / Dominio	
		PHP	Go	PHP	Go	PHP	Go	PHP	Go	PHP	Go
Todo		65,531	137,464	3,789	5,896	2,025	1,074	17.3	23	32.4	128
Promedio mensual		374	4,909	88	606	64	119	3.1	7.7	5	39.8
Noviembre	2014	3	973	3	336	2	64	1	2.9	1.5	15.2
Octubre	2014	3	2,641	3	607	3	123	1	4.4	1	21.5
Septiembre	2014	1	2,991	1	602	1	136	1	5	1	22
Agosto	2014	6	3,101	5	664	3	147	1.2	4.7	2	21.1
Julio	2014	11	3,329	6	673	3	145	1.8	4.9	3.7	23
Junio	2014	3	472	2	186	2	42	1.5	2.5	1.5	11.2
Mayo	2014	5	1,869	3	469	1	99	1.7	4	5	18.9
Abril	2014	24	3,910	9	685	3	141	2.7	5.7	8	29.8
Marzo	2014	13	4,044	8	739	5	166	1.6	5.5	2.6	24.4
Febrero	2014	14	3,717	6	681	4	133	2.3	5.5	3.5	27.9
Enero	2014	8	3,864	6	659	4	133	1.3	5.9	2	29.1
Diciembre	2013	13	5,859	10	706	7	144	1.3	8.3	1.9	40.7
Noviembre	2013	3	4,747	3	675	2	142	1	7	1.5	33.4
Octubre	2013	14	5,414	9	658	4	122	1.6	8.2	3.5	44.4
Septiembre	2013	14	6,393	10	680	4	130	1.4	9.4	3.5	49.2
Agosto	2013	3	8,575	3	689	2	132	1	12.4	1.5	65
Julio	2013	3	6,268	2	636	2	118	1.5	9.9	1.5	53.1
Junio	2013	9	6,755	7	699	4	133	1.3	9.7	2.3	50.8
Mayo	2013	11	7,590	5	794	4	142	2.2	9.6	2.8	53.5
Abril	2013	17	5,996	10	679	5	127	1.7	8.8	3.4	47.2
Marzo	2013	1	8,044	1	697	1	130	1	11.5	1	61.9
Febrero	2013	55	7,759	20	572	6	120	2.8	13.6	9.2	64.7
Enero	2013	3	7,627	2	635	1	130	1.5	12	3	58.7
Diciembre	2012	3	5,403	3	539	3	82	1	10	1	65.9
Noviembre	2012	14	5,704	8	566	5	111	1.8	10.1	2.8	51.4
Octubre	2012	18	6,745	12	659	3	116	1.5	10.2	6	58.1
Septiembre	2012	18	7,490	8	726	4	127	2.3	10.3	4.5	59

El total es de aproximadamente mil proyectos documentados que están en marcha. La información contenida en la *Tabla 7* fue formulada en base a consultas realizadas en diciembre de 2014.

En la actualidad existen varios proyectos desarrollados en Go en producción dentro de Google. Entre estos proyectos se encuentra el desarrollo de aplicaciones nativas de Google, como es el caso de Gmail, el servidor detrás de Golang.org.

**Tabla 6 Proyectos en Go**

<b>Categoría</b>	<b>Proyectos</b>	<b>Categoría</b>	<b>Proyectos</b>	<b>Categoría</b>	<b>Proyectos</b>
Analizadores de archivos de configuración	7 y 9 sub proyectos	Línea de Comandos	16	Computación en la nube	7
Redes	112	Construcción	14	Documentación	6
Diversos	100	Simulación	1	P2P	6
Librerías Web	100	Caching	12	Programación	6
Controladores de Bases de datos	76	Compresión	12	Windows	6
Gráficos y Audio	70	Widget	12	Procesamiento de datos	5
Estructura de datos	58	Lenguajes	12	Manejo de errores	5
Matemáticas	33	Inicio de sesión	12	Interfaces de SO	5
Herramientas de desarrollo	28	Editores	11	Ordenamiento	5
Sin categoría	25	GIS	11	Astronomía	4
Pruebas	23	Juegos	10	FTP	4
Criptografía	22	Mensajería instantánea	9	Distribución	3
Codificaciones y conjuntos de caracteres	22	Generadores de escáner y analizador	9	Hardware	3
Maquinas virtuales	22	Aprendizaje automático	8	Unix	2
Aplicaciones Web	22	Música	8	Otros	17
Consola	21	DNS	8		

En 2007, Google tenía su desarrollo en C++ que tomaba 45 minutos en resolver un algoritmo interno utilizando un sistema de construcción distribuido. La versión actual desarrollada en el lenguaje de programación Go tarda ahora 25 minutos sin mencionar que de 2007 a la actualidad los requerimientos del algoritmo han crecido notoriamente [5].

En 2012, el lenguaje de programación Go se empezó a usar en algunos procesos de youtube.com y en servidores dentro de la producción de *Google App Engine* [27]. Posteriormente se implementó en proyectos como VItes<sup>19</sup>, y el servidor de descarga de Google dl.google.com que es el que entrega binarios a Chrome y otros instalables desde apt-get en el servidor de Chrome.

En el ámbito privado, el lenguaje de programación Go también empezó a usarse. Las empresas más mencionadas que migraron todo o gran parte de sus servicios a este lenguaje son: Heroku, GitHub, Canonical, Nokia, SoundCloud, BBC Worldwide, CloudFlare, Novartis, SmugMug, StarHat, Tinkercad, DigitalOcean y a finales del 2014 Oracle, Intel y Microsoft empezaron a usar Go en servicios internos [28]. Otra aplicación exitosa desarrollada en el lenguaje de programación Go es Twitterstream el cual publica las actualizaciones de Twitter en Facebook. Esta aplicación se ejecuta en una sola instancia en un servidor privado virtual con 2Gb de RAM y con cerca de 90,000 usuarios [29].

El lenguaje de programación Go empezó a usarse en China cuando Qihoo 360<sup>20</sup> empezó a desarrollar en el año 2012 un servicio que satisficiera una carga excesiva en tiempo real. Este servicio inició corriendo en 20 servidores, con 20 millones de conexiones en tiempo real y enviando 2,000,000 de mensajes al día. Meses después todo Qihoo se migró a Go y se usa una cantidad arriba de 400 servidores y da apoyo a más 200 millones de conexiones en tiempo real y envía más de 10 mil millones de mensajes al día. A Qihoo se le unió DaoCloud<sup>21</sup>,

---

<sup>19</sup> Sistema de instalaciones en SQL a gran escala.

<sup>20</sup> Uno de los principales proveedores de internet, servicios, productos y seguridad tanto web como móvil.

<sup>21</sup> Empresa Asiática de nivel mundial. Se encarga de mover automáticamente el software actualizado de GitHub al servidor.

seguida por Qiniu, empresa de almacenamiento en la nube que atiende más de 150,000 empresas y desarrolladores, almacena más de 50 mil millones de archivos y atendiendo más de 500 millones de archivos por día.

El uso del lenguaje de programación Go en empresas de talla internacional en el continente Asiático, disparó el interés de los desarrolladores. En el 2012 Wei Hsu, desarrollador de Qiniu, creó el primer libro chino de Go.

Hoy en día, compañías Asiáticas como Alibaba, Baidu, Tencent y Weibo están usando uno o más servicios desarrollados en Go.

#### *4.2.1.4 Comparación de Go vs PHP*

En esencia los dos lenguajes realizan una lógica de programación, y manipulan una entrada para crear cierta salida. Pero para ello, los dos toman caminos muy diferentes. Las diferencias son más profundas que sólo la sintaxis y bibliotecas disponibles. Partiendo de algo inicial, Go tiene una filosofía de ser pequeño y simple para lograr grandes cosas.

Una gran cantidad de desarrolladores que probaron seriamente Go, están muy contentos con él. Muchos de ellos reportan una reducción de hasta el 50% del código, más fácil de leer y por lo tanto fácil de mantener. Sobre todo cuando viene de C, C++ o Java. Desarrolladores que migraron de Python a Go, indican estar contentos con la ganancia de rendimiento [5]. Go, un lenguaje en pañales, comparado con el desarrollo que tiene PHP. Si ya se ha manejado PHP, no se tendrá menor problema al manejar Go, pues Go es parecido en su sintaxis.

Cuando un programa desarrollado en PHP crece, también crece su complejidad y dimensión. Se dice que el lenguaje de programación PHP es difícil de adaptar a proyectos posteriores, así como obtener mantenimiento pues por más que sea estricto; resultará más común repetir código y hacer un lenguaje más complicado de leer para un futuro. En el lenguaje de programación Go puede ser más efectiva

la reutilización de código ya que no permite incluir código híbrido, es decir, mezclar lenguaje de etiquetado con un lenguaje de programación.

El concepto de velocidad es un punto por el que el lenguaje de programación Go sobresale. Es la comparación más significativa, en opiniones diversas se comenta que Go realiza una operación por debajo de  $\frac{1}{4}$  del tiempo que tarda en resolverla PHP. Posteriormente especificaré a detalle el tiempo que tardan los dos lenguajes en un ejemplo.

Go no solo es más rápido, sino también es altamente concurrente. En un ejemplo PHP vs Go, el lenguaje de programación Go atiende aproximadamente a 348,555 solicitudes por segundo, mientras que *Laravel*, el *Frameworks* PHP que cuenta con la mayor cantidad de uso en la actualidad, resuelve 3,270 solicitudes, dejándolo por más de un mil por ciento PHP pierde bastante tiempo en las peticiones *http* simples, en la carga de miles de archivos (ya sea de disco o de la memoria cache). El lenguaje de programación Go evita este problema ya que desde que se compila ya no necesita recibir peticiones *http* [45, 51].

El lenguaje de programación PHP realiza las consultas de manera secuencial, desperdiciando así el uso de los demás procesadores y haciendo más tardía la consulta. Mientras que en Go se puede usar la concurrencia para realizar y/o atender uno o varios procesos a la vez.

Go centra la lógica en un archivo, y genera un programa binario estático. En cambio el lenguaje de programación PHP compila dinámicamente en cada petición desde la raíz del servidor.

El paso por valor es copiar el valor de la variable a una variable que solo vivirá dentro de la función predeterminada. Mientras que el paso por referencia apunta a la variable y su valor puede ser modificado por la función. En el lenguaje de

programación Go y en el lenguaje de programación PHP cuentan con ambos tipos de paso de parámetros.

El lenguaje de programación PHP también utiliza una sintaxis basada en C. Permite trabajar con valores enteros, reales, cadenas, arreglos y objetos pero no se necesita declararlos previamente. Existen variables globales y locales, es orientado a objetos y también permite crear funciones [30]. Go también tiene una sintaxis basada en C, respecto a la declaración de valores, se ponen declarar o simplemente asignar un valor a la nueva variable. Finalmente Go también cuenta con variables públicas y privadas.

En los tipos de datos, el lenguaje PHP adquiere variables dinámicas, es decir las variables aquí pueden ser de cualquier tipo y declaradas en cualquier lugar, al igual que pueden ser reutilizadas y modificadas sin mayor complejidad. Mientras que en Go las variables se pueden declarar sin tipo de variable, pero si se declara una variable de un tipo, esa misma variable no puede cambiar de tipo, para esto tiene que ser reasignada.

PHP hasta su versión 4, no contaba con recolector de basura, pero en la versión 5 ya incluye un recolector de basura; Go también cuenta con un recolector de basura.

El lenguaje de PHP es maravilloso para trabajar con pequeños proyectos, pero se viene abajo cuando un proyecto crece, la reutilización de código se vuelve limitada y es más propenso a crear partes confusas en el código. Este efecto, conocido también como crear un código "espagueti", siendo este un problema más común cuando el desarrollador carece de experiencia. Go, por el contrario, crea una plantilla más estricta donde podrá reutilizar código y/o consultas, creando una programación con mayor limpieza.



**Tabla 7 Comparación General Go vs PHP**

Descripción	Go	PHP
<b>Categoría</b>	Lenguaje de programación	Lenguaje de programación
		Lenguaje de plantillas
<b>Sitio Web</b>	www.golang.org	www.php.net
<b>Licencia</b>	BSD	BSD
<b>Lenguaje Script</b>	×	✓
<b>Interpretado</b>	×	✓
<b>Multiusuario</b>	✓	✓
<b>Sistema Operativo</b>	Multiplataforma	Multiplataforma
<b>Número de versión</b>	1.3.3	5.5
<b>Multilenguaje</b>	✓	Condicional
<b>Nivel de dificultad</b>	Principiante	Principiante
	Intermedio	Intermedio
	Avanzado	Avanzado
<b>Soporte de Adobe Flash</b>	×	×
<b>Bases de datos soportadas</b>		
PostgreSQL	✓	✓
MySQL	✓	✓
MongoDB	✓	✓
Microsoft Access	×	✓
Microsoft BI	×	✓
MariaDB	✓	✓
MemcacheDB	✓	×
SQLite	✓	✓
Oracle	✓	✓
ODBC	×	✓
Cassandra	✓	×
CUBRID	×	✓
CouchDB	✓	✓
BangDB	×	✓
IBM DB2	×	✓
InterBase	✓	✓
ORM	✓	✓
Aerospike	✓	✓
Redis	✓	✓
<b>Uso</b>	Base de aprendizaje	Desarrollo web
	Desarrollo Web desarrollo de aplicaciones y computación en la nube	

**Tabla 7 Comparación General Go vs PHP (continuación)**

Descripción	Go	PHP
Fecha de lanzamiento	28-mar-12	1995
Fecha de creación	2009	1994
Nivel de Documentación	★★★★★	★★★★★
Lenguajes que tuvo influencia para crearse	Python	Perl
	C	C
	Java	Java
Tipado dinámico	×	✓
Paradigma de programación	OOP	OOP
	Programación Imperativa	Programación Imperativa
	Orientado a concurrencia	Funcional
Sintaxis HTML	✓	✓
Lenguaje Compilado	✓	No
Oportunidad de trabajo	★★★★☆	★★★★☆
Uso comercial	Gratis	Gratis

Existen una inmensa cantidad de comparaciones entre estos dos lenguajes, se puede hacer una prueba sobre los dos con la profundidad que el interesado le quiera dar. Centrando sólo en algunas características que se creyeron importantes, se recopiló información encontrada en diversos grupos y páginas consultadas para finalmente se generó la *Tabla 7* de comparación entre estos dos lenguajes mencionados [3, 32]<sup>22</sup>.

PHP, por ser un lenguaje que tiene ya una antigüedad muy superior al lenguaje de programación Go, el lenguaje PHP tiene un mayor alcance y es el lenguaje más

<sup>22</sup> Sitio web donde se comparan una gran cantidad de objetos. La forma en que se genera la comparación es en base a la aportación de conocimiento de los clientes del sitio.

popular para el desarrollo web de lo que lo es aún Go. El lenguaje de programación PHP cuenta con una mayor compatibilidad de manejadores de bases de datos, y para muchos es una herramienta más factible para laborar en el desarrollo de aplicaciones. Ambos lenguajes tienen un amplio panorama para el desarrollo y se pueden llegar a ser tan complejos como el desarrollador quiera.

#### 4.2.1.5 Desempeño de Go vs PHP

Un *benchmark* es un conjunto de procedimientos para evaluar el rendimiento de un ordenador. Hay cuatro categorías generales de pruebas de comparación, pero solo usaremos la primera.

- **Application-based.** Ejecuta y cronometra sistemas.
- **Playback test.** Ejecuta y cronometra ejecución de gráficos o uso de disco en sistemas.
- **Synthetic test.** Ejecuta pruebas aisladas en partes específicas del sistema.

En esta investigación se incluyeron dos *benchmarks*. El primero fue una prueba realizada por *Benchmarkgame* la cual se desarrolló en una máquina con procesador *X64 quad-duo*, corriendo un sistema operativo Unix donde se probaron los siguientes algoritmos [33].

- **Fasta**, un algoritmo de bio-computación usado para el alineamiento de secuencias de ADN y/o proteínas [34].
- **Árbol binario**, una estructura de datos, donde cada nodo del árbol tiene solo dos hijos [35]
- **N-body**, problema clásico de la mecánica celeste, describe un número de cuerpos que cuentan con una fuerza de interacción propia [36].

En base a los resultados mostrados en la *Tabla 8*, podemos apreciar que el lenguaje Go es por mucho más veloz que PHP. En el primer algoritmo consultado, el algoritmo *Fasta*, PHP termina 2.2 minutos después, y termina de usar el CPU 2.1 minutos también después de Golang. El trabajo de la solución del algoritmo realizado en PHP se centró sobre 3 de los 4 CPUs, mientras que en GO se distribuyó equitativamente sobre los 4 CPUs comprobando la eficiencia en distribución de procesos en el procesador con la que cuenta Go.

En cuanto a memoria, el lenguaje de programación Go utiliza 472 Kb menos que PHP. Finalmente sobre el tamaño del código, PHP desarrolla el algoritmo en un menor número de líneas, claro está que esto depende de la habilidad y forma de programar del desarrollador.

**Tabla 8 Comparación Go vs PHP**

Algoritmo	Segundos en CPU	Memoria KB	Líneas de código	≈ trabajo de CPU
<b>Fasta</b>				
Go	6.20	2,856	1344	85% 85% 86% 89%
PHP	116.02	3,328	1030	60% 36% 4% 0%
<b>Árbol binario</b>				
Go	24.31	265,788	752	86% 86% 85% 85%
PHP	604.80	1,025,300	472	0% 0% 0% 100%
<b>N-body</b>				
Go	22.93	1,020	1310	1% 0% 100% 0%
PHP	712.83	3,336	1082	100% 0% 0% 0%

En el algoritmo de árbol binario, se tienen resultados a favor de Golang, la velocidad de resolución es mayor en 10.27 minutos, donde PHP tarda poco más de 30 veces de lo que tarda Go. PHP utiliza 741.7 MB más de memoria que Go. PHP solo usa el 100% de un CPU, mientras que Go divide el trabajo equitativamente haciendo que todos los CPUs trabajen.

En el algoritmo de los  $N$  cuerpos, el *benchmark* muestra las diferencias en los aspectos más relevantes, dejando por mucho en tiempo, memoria y en uso de los CPUs.

El hecho que PHP sea un lenguaje interpretado y a la vez lineal hace que PHP esté muy lejos de competir con Go. Como se mencionó en el primer capítulo, la concurrencia de Go, es una de las características que respaldan esta nueva tecnología de desarrollo.

Para finalizar esta investigación se realizó una comparación y un *benchmark* elaborado por mí, con el objetivo de lograr poner en práctica todo lo mencionado anteriormente. Este benchmark se realizó en una computadora con un procesador *x64 Intel i7 a 2.4 Ghz*, en un sistema operativo *Mac OSX Yosemite*, basado en *Unix*. La estructura en ambos programas se pretendió fuera la misma, para poder comparar de una manera más clara las diferencias y similitudes en ambos lenguajes, así como también se hizo con el fin de que ambos lenguajes contengan las mismas librerías para analizar la diferencia en tiempos.

Los algoritmos que se desarrollaron en ambos lenguajes fueron:

- **Algoritmo de ordenamiento de burbuja.** Este método de ordenación consiste en recorrer los elementos siempre en la misma dirección, el nombre de este algoritmo trata de reflejar cómo el elemento mínimo siempre sube como una burbuja. El algoritmo es de complejidad cuadrática.

Este algoritmo funciona de forma parecida al de selección, pero haciendo más trabajo para llevar cada elemento a su posición. De hecho es el peor método de ordenamiento [37].

- **Algoritmo de búsqueda QuickSort.** Este método es probablemente el algoritmo de ordenación más utilizado, pues es muy fácil de implementar,

trabaja bien en casi todas las situaciones y consume en general menos recursos, memoria y tiempo, que otros métodos [37]. Este algoritmo cuenta con una complejidad de  $\theta(n \log n)$ .

Su diseño está basado en la técnica de Divide y Vencerás, y consta de dos partes:

a) En primer lugar el vector a ordenar  $numeros[0..n]$  es dividido en dos subvectores no vacíos  $numeros[0..pivote-1]$  y  $numeros[pivote+1..n]$ , tal que todos los elementos del primero son menores que los del segundo. El elemento de en medio se denomina pivote y se calcula como parte del procedimiento de partición.

b) A continuación, los dos subvectores son ordenados mediante llamadas recursivas a Quicksort. Como los subvectores se ordenan sobre ellos mismos, no es necesario realizar ninguna operación de combinación.

La construcción en ambos lenguajes es muy parecida. El principal punto de este *benchmark* es la comparación de velocidad; los dos programas cuentan con funciones y métodos de programación análogos entre ellos. Se consiguió una lista de resultados muy semejantes al *benchmark* investigado, en nuestra primera prueba con el algoritmo de ordenamiento de burbuja se generaron tres arreglos, la primera fue una matriz con una dimensión de 100, la segunda prueba fue con una matriz con una dimensión de 1,000 y finalmente se probó con una matriz de dimensión de 100,000. Todas las anteriores creadas con valores aleatorios entre 0 a 100.

En la primera comparación, la diferencia es nula. En ambos programas y en ambos algoritmos el tiempo de ejecución es el mismo.

En una matriz de 1,000 elementos, el algoritmo de ordenamiento burbuja

implementado en el lenguaje de programación Go reporta 1.9 milisegundos, mientras que el tiempo de ejecución en la implementación del algoritmo burbuja en PHP es de 32.2 segundos. Enorme diferencia en tiempo de ejecución que existe en la prueba 1,000 elementos.

En la última prueba donde se procesan a ordenar 100,000 elementos Go reportó un tiempo de 27.5 segundos, mientras que la ejecución de los mil elementos implementados en el método de ordenamiento burbuja realizado en el lenguaje PHP se tuvo que suspender después de 24 horas de ejecución.

#### 4.2.1.6. Códigos incluidos en la comparación

Con el mismo orden en el cual se presentaron los algoritmos en la pasada *Tabla 9*. El primer código analiza es el algoritmo de ordenamiento burbuja realizado en Go.

**Tabla 9 Segunda comparación Go vs PHP**

Algoritmo	Datos	Tiempo transcurrido(seg)	Líneas de código
<b>Burbuja</b>			
Go	100	0.00003	29
	1,000	0.00190	
	100,000	27.50060	
PHP	100	0.00004	11
	1,000	32.24780	
	100,000	No terminado	
<b>QuickSort</b>			
Go	100	0.00002	40
	1,000	0.00010	
	100,000	0.00935	
PHP	100	0.00020	30
	1,000	0.00210	
	100,000	0.36730	

El programa del siguiente cuadro de código se divide en 3 funciones, la primera función, es una función que genera una matriz aleatoria entre 0 y 100, el único parámetro que recibe es el tamaño que se quiere de la matriz lineal.

La segunda función, que se escribe a partir de la línea 17 del código, es la implementación del algoritmo de burbuja, el cual consta de dos ciclos. El primer ciclo es el índice que llevará el número de pasadas que será igual a la longitud del arreglo 1; el segundo ciclo es el encargado de verificar si existen números desacomodados y aplicar el intercambio entre ellos cuando los encuentra. Go, permite la declaración múltiple

```
Go
1 package main
2
3 import (
4     "fmt"
5     "math/rand"
6     "time"
7 )
8
9 func create_array(n int) []int {
10     numbers := make([]int, n)
11
12     for i := 0; i < n; i++ {
13         numbers[i] = rand.Intn(101)
14     }
15     return numbers
16 }
17
18 func o_burbuja(numbers []int) {
19     for i := 1; i < len(numbers); i++ {
20         for j := 0; j < len(numbers)-1; j++ {
21             if numbers[j] > numbers[j+1] {
22                 //aux := numbers[j]
23                 //numbers[j] = numbers[j+1]
24                 //numbers[j+1] = aux
25                 numbers[j], numbers[j+1] = numbers[j+1], numbers[j]
26             }
27         }
28     }
29 }
30
31 func main() {
32     t1 := time.Now()
33     rand.Seed(time.Now().UTC().UnixNano())
34
35     numbers := create_array(100)
36     o_burbuja(numbers)
37
38     t2 := time.Now()
39     fmt.Println("Tiempo empleado: ", t2.Sub(t1))
40 }
```

La tercera función es la función principal, donde se mandan llamar las dos funciones anteriores. El tiempo que el programa dura es contado al iniciar el



programa capturando el valor de la fecha, esto mismo se hace antes de terminar y se saca la diferencia con la función *Sub* escrita dentro del *Print* en la línea 29.

El desarrollo del algoritmo de ordenamiento burbuja en el lenguaje de programación PHP, muestra análogamente lo mismo al código desarrollado en Go mostrado anteriormente, pero con la diferencia de que en PHP desde su versión 4.0, no requiere de una función principal.

El código, se divide en dos funciones, que hacen lo mismo que se explico en el pasado código. Go usa el paso de parámetros por valor, es decir siempre copia el valor a usar, hace esto hasta con los punteros. En el lenguaje de programación PHP se debe de manejar paso por referencia en el parámetro usado para la función del algoritmo para que pueda modificar los valores de la matriz original.

```
PHP
1 <?php
2 function create_array($n){
3     for ($i=0; $i <$n ; $i++) {
4         $numbers[$i] = rand(0, 100);
5     }
6     return $numbers;
7 }
8
9 function o_burbuja(&$numbers) {
10     for ($i = 1; $i < sizeof($numbers); $i++) {
11         for ($j = 0; $j < sizeof($numbers)-1; $j++) {
12             if ($numbers[$j] > $numbers[$j+1]) {
13                 $aux = $numbers[$j];
14                 $numbers[$j] = $numbers[$j+1];
15                 $numbers[$j+1] = $aux;
16             }
17         }
18     }
19 }
20
21 $tiempo_inicio = microtime(true);
22
23 $numbers = create_array(100000);
24
25 o_burbuja($numbers);
26
27 $tiempo_fin = microtime(true);
28 $tiempo = bcsub($tiempo_fin, $tiempo_inicio, 20);
29 print "Tiempo empleado:\t" . number_format($tiempo*1000000). " uS\n";
30 ?>
```

A partir de la línea 21 del bloque de código, se hace lo que en Go se realizó dentro de la función principal. Y al igual que en el lenguaje de programación Go, para

calcular el valor que duró el programa se captura el tiempo cuando comienza y cuando termina el programa y se saca una diferencia de ambos tiempos.

El segundo algoritmo que se programó, fue el algoritmo de ordenamiento *QuickSort*, al igual que en código del algoritmo de burbuja, se usó la misma función para generar una matriz de una sola dimensión con valores aleatorios anteriormente mencionados.

El algoritmo se resuelve dentro de la función *qs*, esta función recibe un arreglo de enteros a ordenar, un valor entero que indica el extremo izquierdo y otro que indica el extremo derecho del arreglo.

Los pasos a seguir al realizar el algoritmo son:

1. Seleccionar el elemento central, llamándolo pivote.  $P$  es igual al valor en la posición resultante de la división de la suma de los dos extremos del array entre 2.
2. Dividir elementos restantes en particiones *left* y *right*, si un elemento es mayor al valor de  $P$ , se pasa a *right* y si es menor se pasa a la parte de *left*.

En la línea 21, se inicia un ciclo que solo será parado cuando la posición del índice  $i$  supere a la posición del índice  $j$ . En Go, como se vio en el capítulo 3, todos los ciclos son *for*, y este funciona respecto a los parámetros que se le introduzcan. En este caso, el *for* significaría un *while*, pero como contiene una condición dentro después del contenido del ciclo, se convierte en un *do while*.

Dentro de este ciclo, existen dos ciclos más, estos ciclos se encargan de mover la posición de los índices  $i$  y  $j$  hasta que encuentran un número fuera de lugar. Se detiene  $j$  cuando encuentra a un número menor que el valor de

$P$ , y se detiene  $i$  cuando encuentra un número mayor al valor de  $P$ . Cuando los dos se detienen y aun no se cumple la condición de salida, genera un intercambio entre los valores de la matriz.

3. Llama recursivamente la función con las dos partes divididas por el pivote. En la línea 39 y en la línea 42, se muestra que envía ambas partes del arreglo pero siempre y cuando la longitud sea mayor a 1.

```
Go
1 package main
2
3 import (
4     "fmt"
5     "math/rand"
6     "time"
7 )
8
9 func create_array(n int) []int {
10     numbers := make([]int, n)
11
12     for i := 0; i < n; i++ {
13         numbers[i] = rand.Intn(101)
14     }
15     return numbers
16 }
17
18 func qs(numbers []int, left int, right int) {
19     i := left
20     j := right
21     P := numbers[int((left+right)/2)]
22     for {
23         for numbers[i] < P {
24             i++
25         }
26         for numbers[j] > P {
27             j--
28         }
29         if i <= j {
30             //aux := numbers[i]
31             //numbers[i] = numbers[j]
32             //numbers[j] = aux
33             numbers[i], numbers[j] = numbers[j], numbers[i]
34             i++
35             j--
36         } else {
37             break
38         }
39     }
40     if left < j {
41         qs(numbers, left, j)
42     }
43     if i < right {
44         qs(numbers, i, right)
45     }
46 }
```

En la función principal, se repiten todas las líneas que se hicieron en el código del algoritmo de ordenamiento burbuja, excepto la llamada a la función del algoritmo de ordenamiento *QuickSort*, en esta función se piden dos parámetros más, la primera y la última posición del arreglo.

Go

```
46 func main() {
47     t1 := time.Now()
48     rand.Seed(time.Now().UTC().UnixNano())
49
50     numbers := create_array(500000)
51     qs(numbers, 0, len(numbers)-1)
52
53     t2 := time.Now()
54     fmt.Println("Tiempo empleado: ", t2.Sub(t1))
55 }
```

Para desarrollar el algoritmo de ordenamiento *QuickSort* en el lenguaje de programación PHP, se siguió la misma lógica de programación que en Go. Como se mencionó, el este código contiene la misma función *create\_array* y las variables para calcular el tiempo.

PHP

```
1 <?php
2 function create_array($n){
3     for ($i=0; $i <$n ; $i++) {
4         $numbers[$i] = rand(0, 100);
5     }
6     return $numbers;
7 }
8
9 function qs(&$numbers, $left , $right ){
10     $i = $left;
11     $j = $right;
12     $P = $numbers[(int)( ($left+$right)/2 )];
13     do{
14         while( $numbers[$i] < $P )
15             $i++;
16         while( $numbers[$j] > $P )
17             $j--;
18
19         if( $i <= $j ){
20             $aux = $numbers[$i];
21             $numbers[$i] = $numbers[$j];
22             $numbers[$j] = $aux;
23             $i++; $j--;
24         }
25     }while( $i <= $j );
26
27     if( $left < $j )
28         qs($numbers, $left, $j);
29     if( $i < $right )
30         qs($numbers, $i, $right);
31 }
32
33 $tiempo_inicio = microtime(true);
34 $numbers = create_array(50);
35 qs($numbers, 0, count($numbers)-1);
36
37 $tiempo_fin = microtime(true);
38 $tiempo = bcsub($tiempo_fin, $tiempo_inicio, 20);
39 print "Tiempo empleado:\t" . number_format($tiempo*1000000). " uS\n";
40 ?>
```

La función de *qs* en PHP también se hace por paso de referencia; En PHP, si existe *while* y *do while*.

En el lenguaje de programación Go se puede escribir casi igual a PHP, los cambios mínimos encontrados en el desarrollo entre ambos códigos son:

- Las variables en el lenguaje de programación PHP siempre llevan un identificador previo (\$), esto no es necesario en el lenguaje de programación Go.
- En Go, no es obligatorio poner punto y coma, en el lenguaje de programación PHP son obligatorios.
- En PHP pueden obviarse las llaves en ciclos o condicionales de que solo contengan una sola instrucción, en el lenguaje de programación Go es obligatorio su uso por su sintaxis sin punto y coma, la primera llave en las condicionales o ciclos va en el mismo renglón donde inicie, de no ser así el compilador marcará error de sintaxis porque lo tomaría como una declaración u otra instrucción.
- En el lenguaje de programación Go se puede usar asignación múltiple y así evitar el uso de variables auxiliares en algún intercambio de valores; en PHP esto no está permitido.
- En PHP existen múltiples funciones para saber la longitud del arreglo, en Go la más indicada es *len*.
- Las funciones de tiempo se usaron diferentes por que se encontró de esa forma, la forma más eficiente para registrar el tiempo. PHP también cuenta con la función *date*, pero *time* es más correcta para este ejemplo.

La prueba benchmark realizada a ambos lenguajes de programación es la parte más esencial de la investigación poniendo en evidencia la hipótesis anteriormente mencionada.

El lenguaje de programación PHP es un lenguaje que carece de facilidad en funciones y métodos para desarrollar sistemas o aplicaciones de gran tamaño y exigencia.

En este capítulo se presentó una serie de estadísticas respecto al número de usuarios, publicaciones y sitios web que existen en el periodo de dos años en ambos lenguajes de programación; también se presentó una comparación de rendimiento entre estos dos lenguajes de programación.

En estadísticas de usuarios, publicaciones y sitios web, el lenguaje de programación PHP se posiciona por encima de Go. Pero en aspectos técnicos el lenguaje de programación Go supera por bastante la capacidad de PHP.

En todo o casi todo lenguaje de programación se puede lograr el mismo objetivo, en unos lenguajes quizá es más complejo realizarlo y en otros es más sencillo, todo dependerá de la práctica que tenga el desarrollador con el lenguaje. Como se menciona en el primer capítulo, cada vez están naciendo lenguajes de programación que se asemejan cada vez más al lenguaje humano y por lo tanto hacen más sencilla la programación.

El uso del lenguaje dependerá del gusto y fin para lo que se requiera.

## CAPÍTULO 5

### CONCLUSIONES

La presente investigación se ha desarrollado con el objetivo de mostrar un entorno amplio sobre un nuevo lenguaje con tan solo 6 años de desarrollo. Cabe resaltar que Michoacán es el estado con menor desarrollo e innovación en cuanto a nuevos lenguajes se refiere. Fue muy complicado conseguir personas y empresas que desarrollan sistemas en este lenguaje (ninguna en Michoacán). Este es el motivo por el cual se invita a nuevos programadores a tomar este trabajo como un primer escalón para el desarrollo e investigación para próximos trabajos. O bien, para la búsqueda de nuevos lenguajes.

Se desarrollo un documento de base para la introducción al lenguaje de programación Go. Go es un lenguaje que tiene el apadrinamiento de un gigante como lo es Google. El lenguaje de programación Go tiene ventajas tentadoras para el desarrollo ante los lenguajes antiguos. El desarrollo en este lenguaje es más amigable para programadores novatos, ya que tiene simplicidad en su estructura y sintaxis.

En la actualidad los problemas computacionales en los servicios web exigen una mayor velocidad, una mayor área de manejo de datos y una forma de trabajo concurrente. Es por esto que Go se plantea como una excelente elección para el desarrollo de nuevos servicios web, la creación de *goroutines* y la creación de un archivo binario compilado son conceptos que el lenguaje de programación PHP carece por completo.

Si bien, el presente trabajo abordó una comparación teórico-práctico, no se puede decir que está regida por algún marco de referencia sobre el cual se establezca la

comparación. Las comparaciones siempre serán tan grandes como el interesado quiera hacerlas; en esta investigación se abarcaron aspectos que se consideraron interesantes, dados el tiempo y forma del proyecto.

Los gustos de cada desarrollador son distintos, no existen formas para elegir el lenguaje para trabajar; así como tampoco argumentos desde un punto de vista ajeno a la publicidad para convencer que se use cualquier lenguaje exhibido.

Se concretó en esta investigación una comparación entre el comportamiento de ambos lenguajes en la red y una comparación sintaxis y de rendimiento de los dos lenguajes exhibidos.

El poner en marcha una investigación donde no tienes un camino marcado a seguir y no cuentas con recursos confiables vuelve muy complicado el proceso para conseguir una meta que se planteó desde un principio como sencilla pero muy interesante.

En mayor parte, este trabajo determinó el desarrollo de una investigación teórica, dejando así el desarrollo progresivo de un panorama más técnico para siguientes fases. Este proyecto podrá tener un seguimiento con un sin fin de etapas más técnicas.

Las ventajas más sobresalientes del lenguaje de programación Go en la ejecución y desarrollo de los programas es la inmensa diferencia que existe en velocidad y la exigente sintaxis te crea disciplina para programar en cual quiera sea el lenguaje.

Desde el punto de vista personal, no me decepcioné en entrar más a fondo al lenguaje de programación Go y fue algo relativamente sencillo con las bases que tenía sobre el lenguaje de programación C y PHP, los cuales fueron enseñados a lo largo de mi formación en la Universidad. Es primordial el hecho de saber las



bases de programación para que de esta forma el migrar entre un lenguaje y otro no cause ningún conflicto.

La parte mas confusa quizá del lenguaje de programación Go puede ser la analogía de su paradigma de lenguaje orientado a objetos. Pero en lo personal el hecho que no supiera tanto sobre este tema me hizo comprenderlo más rápido.

En conclusión final, el lenguaje de programación PHP es un excelente lenguaje para crear sistemas poco extensos pero entre más se requiera expandir el proyecto, más complejo se empieza a volver el uso de este lenguaje, más aun si se hace sin el uso de algún *Framework* en Go. Por otro lado, el lenguaje de programación Go, es un muy buen lenguaje de programación si vienes de aprender C o Java pues contiene mucha de la lógica y sintaxis que estos lenguajes ya manejan. Go es un potente lenguaje de programación para proyectos de mayor escala y quizá demasiado si se trata de desarrollar un sistema o aplicación relativamente sencilla aunque existen también *frameworks* que permiten ayudarte en el desarrollo de proyectos dependiendo su escala y dimensión.

## BIBLIOGRAFÍA

1. Lic Francisco Javier Diaz y Lic Claudia Mariana Banchoff T. (2000), *LINTI*, Facultad de informática, Universidad Nacional de la Plata. Uruguay.
2. Golang-book, Caleb Doxsey. (2012). <http://www.golang-book.com/>
3. Pagina oficial de Go. (2009-2014). Golang.org  
Golang. (2009). *Documentacion Golang*. Recuperado el 10 de Diciembre de 2014, de Golang: <https://golang.org/doc/>  
Golang. (2009). *Golang*. Recuperado el 19 de Marzo de 2015, de Wiki Golang: <https://golang.org/doc/articles/wiki/>
4. PHP y MySQL: Tecnología para el desarrollo de aplicaciones web. (2005), Ángel Cobo, Patricia Gómez Daniel Pérez, Rocío Rocha, Ediciones Díaz de Santos, ISBN: 84-7978-706-6
5. Conferencia magma 2015, William Kennedy, Manzanillo, Colima (mayo 2015)
6. Lutz, M. (2010). *Inc. Learning Python*. US: O'Reilly Media.
7. Larousse. (1999). *Diccionario El pequeño Larousse Ilustrado*. -: Larousse .
8. Lenguajes de programación, informática IV, Programación funcional, Hernández castillo Vicente, UNAM. Facultad de contaduría y administración (2008)
9. Éric Lévénez, (2013), *The History of Programming Languages* - O'Reilly Media, (Mayo 2015),  
[http://archive.oreilly.com/pub/a/oreilly//news/languageposter\\_0504.html](http://archive.oreilly.com/pub/a/oreilly//news/languageposter_0504.html)
10. Reese, G. (2012). Computacion en la nube. *Revista del Instituto Español de Estudios Estratégicos* , 92.

11. Rob Pike (2012), Recuperado en Abril 2015, *Google Inc*  
<http://talks.golang.org/2012>.
12. Andrew Gerrand, (23 enero 2013), Fmt with Go code, Recuperado Julio 2015.  
<http://blog.golang.org/go-fmt-your-code>
13. A Tutorial Introduction to the Lambda Calculus. Raúl Rojas. (2011). FU Berlin,  
WS-97/98
14. Steve Francia (9 junio 2014), Is Go object oriented?, <http://spf13.com/post/is-go-object-oriented/>.
15. Go (lenguaje de programación), (27 octubre 2013)  
[https://es.wikipedia.org/wiki/Go\\_\(lenguaje\\_de\\_programaci%C3%B3n\)](https://es.wikipedia.org/wiki/Go_(lenguaje_de_programaci%C3%B3n))
16. Métodos orientados a objetos, Graham, IAN. Editorial ADDISON-Wesley & David de santos. 1 edición, México (1996)
17. PHP manual de referencias, Recuperado Agosto 2015, <http://php.net/>
18. Hilda Contreras (2006), Libro OOP PHP, Universidad de los andes Venezuela.
19. Munroe, A. (- de - de 2008). *alpha* . Recuperado el 5 de Marzo de 2015, de <http://blog.alphasmanifesto.com/2013/04/15/php-un-fractal-de-un-mal-diseno/#comment-25276>,
20. Wei Hsu, Qiniu, (2014), <http://blog.golang.org/Qiniu>.
21. Análisis y estadísticas. (2014).  
*Grokbase*. Recuperado Diciembre de 2014, <http://grokbase.com/s/php/>  
*Grokbase*. Recuperado Diciembre de 2014, de <http://grokbase.com/s/go>
22. Grupos de Google. (2009-2014). <https://groups.google.com>

*Golang-spanish*. Recuperado el 10 de Diciembre de 2015,  
[https://groups.google.com/forum/?utm\\_source=digest&utm\\_medium=email#!forum/golang-spanish](https://groups.google.com/forum/?utm_source=digest&utm_medium=email#!forum/golang-spanish).

23. Davis, M., & Phillips, J. (2008). *PHP y Mysql*. España: Anaya Multimedia.
24. Surveys, W. T. (- de - de 2009). *w3techs*. Recuperado el 2 de Marzo de 2015, de [http://w3techs.com/technologies/overview/programming\\_language/all](http://w3techs.com/technologies/overview/programming_language/all)
25. PHP just grows & grows, (2013),  
<http://news.netcraft.com/archives/2013/01/31/php-just-grows-grows.html>
26. Keith Rarick (noviembre 2013), 4 years of Go , <https://blog.golang.org/4years>
27. Russ Cox , The Go Blog, (10 nov 2012) Recuperado Julio 2015.  
<http://blog.golang.org/go-turns-three>  
Russ Cox, Open source comunidad, (8 julio 2015), Recuperado Julio 2015.  
<http://blog.golang.org/open-source>
28. Andrew Gerrand (10 nov 2014), Articulo Media década de Go,  
<http://blog.golang.org/Qiniu>.
29. Michael Hoisie (2010), Recuperado en Abril 2015,  
<http://blog.golang.org/twitterstream>.
30. Pegar objetos de otras clases dentro de una nueva clase. , Prof. Hilda Contreras, Universidad de los andes Venezuela, (2006), Recuperado Marzo 2015.  
<http://webdelprofesor.ula.ve/ingenieria/gilberto/pr2/practica7-OxO-herencia-composicion.pdf>
31. VSchart (2014). <http://vschart.com/>
32. The comparison wiki. (2014). <http://vschart.com/compare/go-language/vs/php>.

33. Benchmarksgame. (Febrero de 2015). *Benchmark PHP vs Go*. Recuperado el Marzo de 2015, de Benchmark PHP vs Go:  
<http://benchmarksgame.alioth.debian.org/u64q/compare.php?lang=go&lang2=php>.
34. Diaz, R. P. (2012). *Biología computacional - Lecciones*. Recuperado el 20 de Marzo de 2015, de Cursos virtuales Universidad Nacional de Colombia,  
<http://www.virtual.unal.edu.co/cursos/ingenieria/2001832/lecciones/fasta.html>
35. Graff, M. (2013). *Notas de Analisis de Algoritmos*. Universiada Michoacana de San Nicolas de Hidalgo. Morelia: UMSNH.
36. Aguilar, L. A. (23 de junio de 1992). El problema de N cuerpos en la astronomía. *Revista Mexicana de Física* , 701.
37. Técnicas de Diseño de Algoritmos Segunda Edición (Mayo 2000), Rosa Guerequeta y Antonio Vallecillo , Editorial Servicio de Publicaciones de la Universidad de Málaga. ISBN: 84-7496-666-3
38. Universidad de alicante, departamento de ciencias de la computación e inteligencia artificial, Historia y evolución de los lenguajes de programación. (2008-09). <http://www.dccia.ua.es/dccia/inf/ asignaturas/LPP/2008-2009/tema-01.html>
39. Artículo COMPUTACIÓN EN LA NUBE, Notas para una estrategia española en cloud computing (2008) Luis Joyanes Aguilar, Catedrático de Lenguajes y Sistemas Informáticos en la Escuela de Ingeniería y Arquitectura de la Universidad Pontificia de Salamanca.
40. Artículo Computación en Nube (Cloud Computing) y Centros de Datos: la nueva revolución industrial. ¿Cómo cambiará el trabajo en organizaciones y empresas?
41. Universidad de Alicante. (2009). Lenguajes y Paradigmas de programación. *Notas escolares de la Universidad*.

42. Luis joyanes Aguilar, Ignacio Zahonero Martinez, (2000) Programación en C, Metodología, algoritmos y estructura de datos
43. LarryBattle, Build-web-application-with-golang Book. (2013).  
[https://github.com/Unknwon/build-web-application-with-golang\\_EN/tree/master/code](https://github.com/Unknwon/build-web-application-with-golang_EN/tree/master/code)
44. Google's Go Programming Language: Taking Cloud Development By Storm, Matt Asay. (2014). <http://readwrite.com>.
45. Carlos Díaz, Why should consider Golang for your next web app. (2014).  
<http://www.cyb.co.uk/blog/development/2014/07/why-you-should-consider-golang-for-your-next-web-application>
46. Instalación de Go en diferentes sistemas operativos, Recuperado Enero 2015,  
<http://golang.org/doc/install>
47. Tour sobre la sintaxis, praxis del lenguaje Go, configuración y entornos según sistema operativo para Go, Recuperado Enero 2014,  
<http://golang.org/doc/code.html>
48. IDE de desarrollo para GoGolang Lite IDE, Recuperado Agosto 2015,  
<https://code.google.com/p/golangide>
49. Go FAQ, (2008 – 2015), Preguntas y respuestas frecuentes sobre el lenguaje Go (FAQ), <http://golang.org/doc/faq>
50. Antonio Cortez (abril 2015), <http://www.antoniocortes.com/que-me-gusta-de-go/>.
51. Carlos Díaz, comparative PHP vs Go source. (2014).  
<http://www.cyb.co.uk/blog/development/2014/08/golang-example-for-beginners#.VPpDuYGUeci>
52. Holzner, S. (2009). *PHP: Manual de Referencia*. Mexico: Mac Graw Hil.

53. ICANN. *ICANN FAQs*. Recuperado el - de - de 2014, de icann.org:  
<https://www.icann.org/resources>
54. Gannaway, M. (2014). *PHP vs Golang*. Lead Software Engineer at Cyber Interactive Ltd. Francia: -.
55. What are php lambdas and closures , Artículo, Philip Brown, (25 marzo 2013), Recuperado Mayo 2015 <http://culttt.com/2013/03/25/what-are-php-lambdas-and-closures/>
56. Funciones lambda, Recuperado Mayo 2015, <http://www.inf.fu-berlin.de/lehre/WS03/alpi/lambda.pdf>
57. Funciones anónimas y clousures, (1 Junio 2013), <http://blog.denevell.org/golang-closures-anonymous-functions.html>
58. Lucio M. Tato (2013), OOP Golang, <https://github.com/luciotato/golang-notes/blob/master/OOP.md>.
59. Sección de código de practica para profundizar conceptos del lenguaje “Codewalks”, Recuperado febrero 2015,  
<http://golang.org/doc/codewalk/functions/>
60. Programación orientada a objetos, UNAM, Facultad de contaduría y administración. Luis Arenas Hernández,  
<http://fcasua.contad.unam.mx/apuntes/interiores/docs/98/opt/java.pdf>
61. introducción a Go, Antonio Nicolás Pina, 2014, Recuperado Marzo 2015  
<http://es.slideshare.net/scaliamedia/introduccion-a-google-go>
62. Structs in Go instead of classes in , Sathish VJ, (2001),  
<http://golangtutorials.blogspot.mx/2011/06/structs-in-go-instead-of-classes-in.html>

63. Polimorfismo, Recuperado Marzo 2015, <http://c2.com/cgi/wiki?PolyMorphism>
64. Java con programación orientada a objetos y aplicaciones en la www. Wang. Pauls. Editorial internacional Thomson Editores. 1 edición. México (1999)
65. Análisis y diseño de orientado a objetos. Martin, James. Editorial Prentice hall. 1 edición. México (1994).
66. Revista: Sociedad y Utopía: Revista de Ciencias Sociales; (36). (Nov 2010). Luis Joyanes Aguilar
67. Oscar Toro (2011), Qué es un benchmark?, Apuntes, <http://www.monografias.com/trabajos/benchmark/benchmark.shtml>,
68. Lic. Francisco Javier Díaz, UNLO, Una solución OpenSource para el desarrollo de paginas web dinámicas, PDF, (2000)
69. Lenguajes de programación: diseño e implementación. Terréense W. Prat, Marvin V. Zelkowitz. Prentice-Hall Hispanoamericana S.A., era edición. México (1998)