



Universidad de San Carlos de Guatemala  
Facultad de Ingeniería  
Escuela de Ingeniería en Ciencias y Sistemas

## **GUÍA PARA LA ELECCIÓN DEL LENGUAJE DE DESARROLLO PARA PROYECTOS**

**Luis Pedro Estrada Casas**

Asesorado por el Ing. Julio Ernesto Contreras Sierra

Guatemala, octubre de 2011

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



FACULTAD DE INGENIERÍA

**GUÍA PARA LA ELECCIÓN DEL LENGUAJE DE DESARROLLO PARA  
PROYECTOS**

TRABAJO DE GRADUACIÓN

PRESENTADO A LA JUNTA DIRECTIVA DE LA  
FACULTAD DE INGENIERÍA  
POR

**LUIS PEDRO ESTRADA CASAS**

ASESORADO POR EL ING. JULIO ERNESTO CONTRERAS SIERRA

AL CONFERÍRSELE EL TÍTULO DE

**INGENIERO EN CIENCIAS Y SISTEMAS**

GUATEMALA, OCTUBRE DE 2011

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA  
FACULTAD DE INGENIERÍA



**NÓMINA DE JUNTA DIRECTIVA**

DECANO	Ing. Murphy Olympo Paiz Recinos
VOCAL I	Ing. Alfredo Enrique Beber Aceituno
VOCAL II	Ing. Pedro Antonio Aguilar Polanco
VOCAL III	Ing. Miguel Ángel Dávila Calderón
VOCAL IV	Br. Juan Carlos Molina Jiménez
VOCAL V	Br. Mario Maldonado Muralles
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

**TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO**

DECANO	Ing. Murphy Olympo Paiz Recinos
EXAMINADOR	Inga. Floriza Ávila
EXAMINADOR	Inga. Sonia Castañeda
EXAMINADOR	Ing. Marlon Pérez Turk
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

## **HONORABLE TRIBUNAL EXAMINADOR**

En cumplimiento con los preceptos que establece la ley de la Universidad de San Carlos de Guatemala, presento a su consideración mi trabajo de graduación titulado:

### **GUÍA PARA LA ELECCIÓN DEL LENGUAJE DE DESARROLLO PARA PROYECTOS**

Tema que me fuera asignado por la Dirección de la Escuela de Ingeniería en Ciencias y Sistemas, con fecha enero 2010.

Luis Pedro Estrada Casas



Guatemala, 6 de octubre de 2010

Señores  
Coordinación de Revisión de Tesis  
Escuela de Ciencias y Sistemas  
Facultad de Ingeniería  
Universidad de San Carlos de Guatemala

Estimados Señores:

Por este medio hago de su conocimiento que le he dado seguimiento al trabajo de tesis titulado "Guía para la elección del lenguaje de programación para proyectos", que el estudiante Luis Pedro Estrada Casas, carné 2005-17701 desarrolló como proyecto de graduación para optar al título de Ingeniero en Ciencias y Sistemas.

Este trabajo ha sido de mi interés y por la experiencia que cuento en el área de investigación he considerado como aprobado y finalizado satisfactoriamente dicho trabajo.

Agradeciendo su atención a la presente me despido deseándole éxitos en sus labores diarias.

  
MBA Ing. Julio Ernesto Contreras  
Asesor  
Escuela de Ciencias y Sistemas  
Colegio No. 4775



Universidad San Carlos de Guatemala  
Facultad de Ingeniería  
Escuela de Ingeniería en Ciencias y Sistemas

Guatemala, 27 de Octubre de 2010

Ingeniero  
**Marlon Antonio Pérez Turk**  
Director de la Escuela de Ingeniería  
En Ciencias y Sistemas

Respetable Ingeniero Pérez:

Por este medio hago de su conocimiento que he revisado el trabajo de graduación del estudiante **LUIS PEDRO ESTRADA CASAS** carné **2005-17701**, titulado: **"GUIA PARA LA ELECCION DEL LENGUAJE DE DESARROLLO PARA PROYECTOS"**, y a mi criterio el mismo cumple con los objetivos propuestos para su desarrollo, según el protocolo.

Al agradecer su atención a la presente, aprovecho la oportunidad para suscribirme,

Atentamente,

  
**Ing. Carlos Alfredo Azurdia**  
Coordinador de Privados  
y Revisión de Trabajos de Graduación



E  
S  
C  
U  
E  
L  
A  
  
D  
E  
  
C  
I  
E  
N  
C  
I  
A  
S  
  
Y  
  
S  
I  
S  
T  
E  
M  
A  
S

UNIVERSIDAD DE SAN CARLOS  
DE GUATEMALA



FACULTAD DE INGENIERÍA  
ESCUELA DE CIENCIAS Y SISTEMAS  
TEL: 24767644

*El Director de la Escuela de Ingeniería en Ciencias y Sistemas de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer el dictamen del asesor con el visto bueno del revisor y del Licenciado en Letras, de trabajo de graduación titulado **“GUÍA PARA LA ELECCIÓN DEL LENGUAJE DE DESARROLLO PARA PROYECTOS”**, presentado por el estudiante LUIS PEDRO ESTRADA CASAS, aprueba el presente trabajo y solicita la autorización del mismo.*

**“ID Y ENSEÑAD A TODOS”**



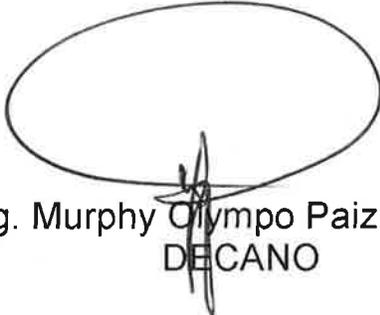
*Ing. Marlon Antonio Pérez Turk*  
*Director, Escuela de Ingeniería Ciencias y Sistemas*

*Guatemala, 01 de septiembre 2011*



El Decano de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer la aprobación por parte del Director de la Escuela de Ingeniería en Ciencias y Sistemas, al trabajo de graduación titulado: **GUÍA PARA LA ELECCIÓN DEL LENGUAJE DE DESARROLLO PARA PROYECTOS**, presentado por el estudiante universitario **LUIS PEDRO ESTRADA CASAS**, procede a la autorización para la impresión del mismo.

IMPRÍMASE.

  
Ing. Murphy Olympo Paiz Recinos  
DECANO



Guatemala, septiembre de 2011

/cc

## **AGRADECIMIENTOS A:**

<b>Dios</b>	Por darme la vida, la sabiduría, las bendiciones y las oportunidades necesarias para alcanzar esta meta dentro, de mi vida profesional.
<b>Mis padres, Luis y Rosa</b>	Por formarme como persona, darme apoyo, confianza y siempre alentarme a cumplir todas las metas que me proponga.
<b>Mis hermanos, Álvaro y María José</b>	Por su amistad, cariño y compañía. Por su buen ejemplo para hacerme crecer como persona.
<b>Mi novia, Wendy</b>	Por su apoyo incondicional y comprensión, por ser la luz de mi vida, mi compañera y amiga, por motivarme y ayudarme a lograr esta meta.
<b>Julio Ernesto Contreras y Javier Gramajo</b>	Por su asesoría, revisión y corrección durante la elaboración del presente trabajo.
<b>Víctor Orozco, Carlos Solórzano, Luis Miguel Pérez y Luis Fernando Juárez</b>	Por compartir conmigo a lo largo de la carrera y proveerme de información valiosa que me ayudo a llegar a alcanzar esta meta.

**La Facultad de  
Ingeniería de la  
Universidad de San  
Carlos de Guatemala**

Por albergarme durante mi formación como profesional dentro de esa casa de estudios.

## ÍNDICE GENERAL

ÍNDICE DE ILUSTRACIONES .....	V
GLOSARIO .....	IX
RESUMEN .....	XIII
OBJETIVOS .....	XV
INTRODUCCIÓN .....	XVII
1. LENGUAJES DE DESARROLLO .....	1
1.1. Historia .....	2
1.2. Evolución de los lenguajes .....	5
1.2.1. Primera generación .....	5
1.2.2. Segunda generación .....	5
1.2.3. Tercera generación .....	6
1.2.4. Cuarta generación .....	7
1.2.5. Quinta generación .....	8
1.2.6. Futuro de los lenguajes de desarrollo .....	9
2. UTILIZACIÓN DE LOS LENGUAJES DE PROGRAMACIÓN EN GUATEMALA .....	11
2.1. Pregunta de investigación .....	11
2.2. Metodología de la investigación .....	11
2.3. Modelo de la encuesta .....	12
2.4. Estructura de la encuesta .....	12
2.5. Instrumento de validación .....	13
2.5.1. Pregunta a todos los participantes .....	13
2.5.2. Preguntas al sector empresarial .....	13

2.5.3.	Preguntas al sector independiente .....	16
3.	RESULTADOS DE LA ENCUESTA .....	21
3.1.	Pregunta a todos los participantes .....	21
3.2.	Preguntas al sector empresarial .....	21
3.3.	Conclusiones del sector empresarial .....	25
3.4.	Preguntas al sector independiente .....	26
3.5.	Conclusiones del sector independiente .....	35
4.	LENGUAJES MÁS UTILIZADOS MUNDIALMENTE.....	37
4.1.	Java.....	39
4.1.1.	Por qué crear Java .....	40
4.1.2.	Ventajas .....	41
4.1.3.	Desventajas.....	42
4.2.	Lenguaje C .....	43
4.2.1.	Evolución.....	43
4.2.2.	Proyectos en los que se utiliza C .....	44
4.2.3.	Ventajas .....	45
4.2.4.	Desventajas.....	45
4.3.	Lenguaje C++ .....	46
4.3.1.	Aspectos técnicos .....	46
4.3.2.	Ventajas .....	47
4.3.3.	Desventajas.....	47
4.4.	Visual Basic .....	48
4.4.1.	Características y trayectoria.....	48
4.4.2.	Proyectos en los que se utiliza Visual Basic.....	50
4.4.3.	Aspectos técnicos .....	51
4.4.4.	Ventajas .....	51
4.4.5.	Desventajas.....	52

4.5.	Lenguaje C# .....	52
4.5.1.	Proyectos en los que se utiliza C#.....	53
4.5.2.	Aspectos técnicos.....	53
4.5.3.	Ventajas.....	54
4.5.4.	Desventajas.....	54
5.	PROYECTOS DE DESARROLLO.....	57
5.1.	Proyecto 1.....	59
5.1.1.	Solución.....	59
5.2.	Proyecto 2.....	61
5.2.1.	Solución.....	62
5.3.	Proyecto 3.....	63
5.3.1.	Solución.....	64
5.4.	Proyecto 4.....	65
5.4.1.	Solución.....	66
5.5.	Proyecto 5.....	66
5.5.1.	Solución.....	67
5.6.	Lenguajes de programación y sus características .....	67
	CONCLUSIONES .....	71
	RECOMENDACIONES .....	73
	BIBLIOGRAFÍA.....	75



## ÍNDICE DE ILUSTRACIONES

### FIGURAS

1.	Modelo de la encuesta .....	12
2.	Sector al que pertenece el encuestado .....	21
3.	Tipo de desarrollo al que se dedica la organización.....	21
4.	Lenguajes de programación utilizados dentro de la organización.....	22
5.	Lenguaje de programación más utilizado en la organización.....	22
6.	Número de proyectos desarrollados anualmente en la organización...	23
7.	Motivo de la elección del lenguaje más utilizado en la organización....	23
8.	Porcentaje de organizaciones que utilizarían un nuevo lenguaje de programación en sus desarrollos .....	24
9.	Cual lenguaje de programación utilizaría la organización .....	24
10.	Razón por la cual la organización utilizaría el nuevo lenguaje de programación .....	24
11.	Método por el cual la organización selecciona el lenguaje de programación al iniciar un proyecto.....	25
12.	Lenguajes de programación conocidos por el desarrollador .....	26
13.	Lenguaje de programación más utilizado por el desarrollador .....	27
14.	Número de proyectos que el desarrollador realiza anualmente con el lenguaje de programación más utilizado .....	27
15.	Razón por la cual el desarrollador selecciona el lenguaje de programación .....	28
16.	Lenguaje de programación que más le gusta utilizar al desarrollador..	28
17.	Número de proyectos que el desarrollador realiza anualmente en el lenguaje de programación de su preferencia .....	29

18.	Conocimiento que tiene el desarrollador sobre el lenguaje Java .....	29
19.	Conocimiento que tiene el desarrollador sobre el lenguaje Visual Basic .....	30
20.	Conocimiento que tiene el desarrollador sobre el lenguaje C .....	30
21.	Conocimiento que tiene el desarrollador sobre el lenguaje C++ .....	30
22.	Conocimiento que tiene el desarrollador sobre el lenguaje C# .....	31
23.	Conocimiento que tiene el desarrollador sobre el lenguaje Python .....	31
24.	Conocimiento que tiene el desarrollador sobre el lenguaje Ruby .....	31
25.	Porcentaje de desarrolladores que utilizaría un nuevo lenguaje de programación .....	32
26.	Lenguaje de programación que al desarrollador le gustaría utilizar .....	32
27.	Motivo por el cual el desarrollador aprendería un nuevo lenguaje de programación .....	33
28.	Lenguajes que el desarrollador considera que el mercado nacional actual demanda .....	33
29.	Lenguaje de programación que el mercado nacional solicitará en los próximos años .....	34
30.	Método por el cual el desarrollador selecciona el lenguaje de programación al iniciar un proyecto .....	34
31.	Porcentaje de la utilización mundial de los lenguajes de programación .....	38
32.	Porcentaje de la utilización mundial de lenguajes de programación de aplicaciones de escritorio .....	39

## TABLAS

I.	Tipo de desarrollo al que se dedica la organización .....	22
II.	Lenguajes utilizados por la organización .....	22
III.	Lenguaje más utilizado en la organización .....	23

IV.	Lenguajes conocidos por el desarrollador.....	27
V.	Lenguaje más utilizado por el desarrollado.....	27
VI.	Motivo de selección del lenguaje .....	28
VII.	Lenguaje que más le gusta utilizar.....	29
VIII.	Lenguaje que le gustaría utilizar al desarrollador.....	32
IX.	Motivación para aprendería un nuevo lenguaje .....	33
X.	Lenguajes demandados por el mercado nacional.....	34



## GLOSARIO

- ALGOL** *Algorithmic Language* (Lenguaje Algorítmico). Lenguaje de programación muy popular en las universidades en los años 60's, pero que no llegó a ser de gran popularidad comercial.
- ANSI** *American National Standards Institute* (Instituto Nacional Americano de Estándares). Organización sin ánimo de lucro que supervisa el desarrollo de estándares para productos, servicios, procesos y sistemas en los Estados Unidos. Miembro de la ISO y de la IEC.
- COBOL** *Common Business Oriented Language* (Lenguaje Orientado a Negocios Comunes). Lenguaje de programación que surge en el año 1960 con el objetivo de ser un lenguaje que pudiera ser usado por todos los ordenadores, ya que en esos años existían numerosos ordenadores incompatibles entre sí.
- Eniac** Primera computadora de propósito general. Era totalmente digital, es decir, que ejecutaba sus procesos y operaciones mediante instrucciones en lenguaje máquina, a diferencia de otras máquinas computadoras contemporáneas de procesos analógicos.
- Fortran** Primer lenguaje de programación de alto nivel, creado por IBM en 1957 para el equipo IBM 704. Orientado al cálculo por lo que tiene una gran eficiencia de ejecución.

<b>Gramática</b>	Estudio de reglas y principios que regulan el uso de los lenguajes.
<b>IDE</b>	<i>Integrated Development Enviroment</i> (Entorno de Desarrollo Integrado) o en inglés.
<b>IEC</b>	Organización de normalización en los campos de eléctrica, electrónica y tecnologías relacionadas.
<b>ISO</b>	Organismo encargado de promover el desarrollo de normas internacionales de fabricación, comercio y comunicación para todas las ramas industriales a excepción de la eléctrica y la electrónica. Su función principal es la de buscar la estandarización de normas de productos y seguridad para las empresas u organizaciones a nivel internacional.
<b>Lenguaje de máquina</b>	Es el único lenguaje que el microprocesador entiende de manera nativa, sólo utiliza los símbolos 1 y 0, debido a que los circuitos microprogamables son sistemas digitales, lo que significa que trabajan con dos niveles de tensión. De abstracción tenemos que estos dos niveles se representan con 0 o 1.
<b>Lenguaje de programación</b>	Notación para escribir programas, por medio de los cuales es posible establecer comunicación con el <i>hardware</i> y así dar órdenes adecuadas para la realización de un determinado proceso.

<b>Nemotécnicos</b>	Símbolos fáciles de reconocer, similares a palabras en idioma inglés.
<b><i>Outsourcing</i></b>	Desarrollo externo de aplicaciones de sistemas informáticos, realizado por parte de terceros.
<b>Tiobe</b>	Compañía Sueca especializada en la evaluación y el seguimiento de la calidad del <i>software</i> . Miden la calidad de los sistemas de <i>software</i> mediante la aplicación de estándares ampliamente aceptados.



## RESUMEN

La investigación pretende ayudar a los desarrolladores independientes, a los gerentes de proyectos, a las organizaciones, entre otros, en la elección del lenguaje de programación para cada uno de los proyectos que emprendan. La primera parte del trabajo está enfocada, con el fin de introducir al lector en el concepto de lenguaje de programación, también mostrar su evolución y desarrollo. Esta parte del trabajo desarrolla las generaciones de lenguajes y que marcó su punto de partida.

La segunda parte del trabajo consta de un estudio de campo realizado, por medio del cual se dio a conocer el porcentaje de personas que utilizan un lenguaje de programación respecto a otro en Guatemala. En el estudio se demuestra las razones del porqué un lenguaje es utilizado respecto a otro y las condiciones para la selección de un lenguaje de desarrollo.

La tercera parte de este trabajo presenta inicialmente los lenguajes de programación más utilizados a nivel mundial. Seguidamente de los lenguajes más utilizados se muestra su historia, beneficios y vulnerabilidades, así como un conjunto de proyectos en los cuales se puede aplicar. Con base en esta última parte del trabajo de investigación se puede establecer bajo qué características es posible utilizar determinado lenguaje de programación y las que no serian recomendables.



# OBJETIVOS

## General

Diseñar una guía para la elección del lenguaje de desarrollo para un determinado tipo de proyecto basado en sus características.

## Específicos

1. Enumerar los lenguajes de programación más utilizados en Guatemala y la razón de su uso.
2. Identificar las características, destrezas y vulnerabilidades de cada uno de los lenguajes de desarrollo más utilizados en la actualidad.
3. Plantear diferentes tipos de proyectos e identificar el lenguaje recomendable para el desarrollo del mismo.



## INTRODUCCIÓN

Los ordenadores no comprenden lo que el ser humano requiere que ellos trabajen. Por ello existen lenguajes de programación, ya que éstos sirven de canal para comunicarnos con los ordenadores y emitir la orden de qué hacer, cómo hacerlo o cuándo, entre otras operaciones.

Los lenguajes de desarrollo, no tienen mucho más de medio siglo de existir, pero han logrado evolucionar, desde la programación basada en ceros y unos, hasta lenguajes parecidos al utilizado entre los seres humanos. Mientras más tiempo pase, estos lenguajes que nos comunican con los ordenadores, serán mucho más fáciles de aprender, implementar y serán más eficientes.

Cuando se inicia un proyecto de desarrollo de *software*, se piensa en muchos aspectos, uno de ellos es la elección del lenguaje de programación. Cada lenguaje de programación presenta sus beneficios y vulnerabilidades, las cuales deben de tomarse en cuenta antes de seleccionar un lenguaje u otro.

Existen muchas razones por las cuales un proyecto no produce los resultados esperados o se catalogan como proyectos perdidos. La elección del lenguaje de programación es una de ellas. Este trabajo de investigación, pretende ayudar a los desarrolladores y organizaciones en la difícil tarea de la elección del lenguaje de desarrollo para un proyecto que inicia.



# 1. LENGUAJES DE DESARROLLO

Los ordenadores no tienen la capacidad de hablar un idioma, son máquinas y como tales, necesitan un lenguaje específico, pensado por el hombre para ellas. Es debido a esta dificultad de comunicación, entre el programador y el ordenador, que surgen los lenguajes de programación, los cuales le permiten al programador comunicarse con el microprocesador mediante términos y símbolos relacionados con el problema que debe resolver utilizando herramientas proporcionadas por la informática.

Los lenguajes de programación nos permiten comunicarnos directamente con el *hardware* y de este modo poder darle instrucciones, están definidos por un conjunto de reglas, las cuales se aplican a un alfabeto, constituido por un conjunto de símbolos válidos. Actualmente, existe un gran número de lenguajes de programación, cada uno de ellos posee su propia gramática, sintaxis y terminología particular.

Originalmente los ordenadores se programaban en lenguaje "máquina", único lenguaje que comprenden los microprocesadores de manera nativa. Éste nos permite utilizar el álgebra booleana y el sistema binario en el diseño y programación de los circuitos. El utilizar este lenguaje de programación representa un alto grado de dificultad, así como la probabilidad de cometer errores; por lo cual en la década de los 40 se concibió la posibilidad de utilizar lenguajes simbólicos.

Los primeros lenguajes de programación en aparecer fueron los ensambladores los cuales trabajaban con mnemónicos<sup>1</sup>. Para diseñar un programa, inicialmente se escribía en papel y luego mediante la utilización de unas tablas se traducía a mano, seguidamente se introducían en el ordenador en forma numérica, pero pronto aparecieron programas que se ensamblaban automáticamente.

## 1.1. Historia

En 1823, Gran Bretaña apoyó a Charles Babbage<sup>2</sup> para que creara una máquina que fuera capaz de realizar sumas repetidas, la cual fue llamada Máquina de Diferencias<sup>3</sup>. Pero Babbage se dedicó a otro proyecto, basado en la creación del Francés Charles Jacquard.

A mediados del siglo XIX, Charles Babbage crea el concepto de computadores programables, en una época en la que no existían nada parecido a la electrónica, los cuales estaban contruidos completamente de forma mecánica. Babbage concibió una máquina que podría ejecutar cualquier tipo de matemática mediante una programación basada en unas tarjetas perforadas, a esta máquina la llamo “La Máquina Analítica”, pero por motivos técnicos de la época, esta máquina no pudo construirse sino hasta mediados del siglo XX.

Con Babbage colaboró Ada Lovelace<sup>4</sup>, ella realizó programas en tarjetas perforadas para aquella supuesta máquina de Babbage, pero como la máquina analítica no pudo ser construida en esa época, los programas que Ada realizó

---

<sup>1</sup> Mnemónico: asignar un nombre a cada tipo de instrucción y cada dirección

<sup>2</sup> *Charles Jacquard*: era un fabricante de tejidos que ideó una manera de que un telar reproduciría automáticamente patrones en sus tejidos, mediante la utilización de unas tarjetas de papel rígido perforadas

<sup>3</sup> Historia: [http://html.rincondelvago.com/lenguajes-de-programacion\\_historia-y-evolucion.html](http://html.rincondelvago.com/lenguajes-de-programacion_historia-y-evolucion.html)

<sup>4</sup> Ada Lovelace: considerada como la primera programadora de la historia. Un programa creado por Ada podía calcular los números de Bernoulli.

tampoco pudieron ser ejecutados, pero esto, simboliza un punto de partida para la programación.

En 1943 surge el primer ordenador, el famoso ENIAC (*Electronic, Numerical, Integrator And Calculator*) pero su construcción finaliza en 1946. Este ordenador electrónico digital fue la máquina más grande del mundo, de la época, y tenía fines generales a gran escala. Este ordenador utilizaba una programación mediante componentes físicos, esto quiere decir que la programación de esta máquina se realizaba cambiando directamente su *hardware*. Para programarla se cambiaba de sitio sus cables y sus entradas y salidas se controlaban mediante tarjetas perforadas.

En 1945, el matemático John Von Neumann elaboró un estudio que demostró que no era necesario modificar el ordenador para que pudiera ser capaz de ejecutar cualquier tipo de programa, por lo tanto el ordenador podía tener una estructura física fija y simple. Esta innovación es conocida como “técnica de programa almacenado” y ésta inicia la era de los lenguajes de programación.

En 1949 aparece el primer lenguaje de programación que se usó en computadoras electrónicas: “*Shortcode*”. Para utilizar este lenguaje de programación era requerido que el programador compilara su programa a lenguaje máquina, 0's y 1's, de manera manual.

En 1951 inicia un proyecto para desarrollar el primer compilador, a cargo de Grace Hopper trabajando para Remington Rand. Esto representó una programación más rápida.

En 1957 surge FORTRAN, desarrollado por IBM, considerado el primero de los grandes lenguajes de programación, aún en uso en la actualidad. Originalmente sólo incluía: *FOR*, *DO*, y *GOTO*, así como definió los tipos básicos: *integer*, *real*, *double*, *booleano*.

En 1958, un profesor del M.I.T<sup>1</sup>, John McCarthy, contempló la teoría de un lenguaje de procesamiento de listas y un año más tarde aparece públicamente la primer complementación llamada LISP 1.5.

En 1958 surge el lenguaje ALGOL, creado por un comité internacional. Este lenguaje tenía la particularidad de que no definía la manera en la cual se manejaban las entradas y salidas, dejando este manejo libre para cada implementación. Este lenguaje no tuvo mucho éxito comercial, pero representa un lenguaje muy importante para la historia de la informática ya que tuvo influencia en lenguajes posteriores, como: Pascal, C, C++, y Java.

En 1959 surge COBOL, creado por CODASYL<sup>2</sup>. COBOL nace como un lenguaje de programación para negocios, el cual debiera de ser fácil de aprender y que no tuviera una formación en las ciencias de la computación. Los únicos tipos de datos manejados en este lenguaje fueron las cadenas y los números, lo cual le dio la característica de poder agrupar los datos en arreglos sencillos y de este modo poder organizarlos y seguirlos de una mejor manera. Las sentencias utilizadas en COBOL son muy similares a las usadas en el idioma inglés, lo cual facilitó la utilización de éste en los negocios promedio.

A lo largo del tiempo han surgido otros lenguajes de programación, los cuales, por regla general, se basan en los primeros lenguajes y siempre con la

---

<sup>1</sup> MIT: *Massachusetts Institute of Technology* (Instituto de Tecnología de Masachusetts). Es una de las principales instituciones dedicadas a la docencia y a la investigación en Estados Unidos, especialmente en ciencia, ingeniería y economía.

<sup>2</sup> CODASYL: Conference on Data Systems and Languages

intención de desechar lo malo, tomar lo mejor y agregar alguna novedad respecto a los lenguajes ya existentes.

## **1.2. Evolución de los lenguajes**

Los lenguajes de programación han pasado por cinco generaciones<sup>1</sup>. Las primeras tres generaciones son muy evidentes, mientras que las últimas dos generaciones son discutidas.

### **1.2.1. Primera generación**

Estos lenguajes son los lenguajes de máquina, los cuales se representan como bits o números binarios, los cuales consisten únicamente de ceros (0) o unos (1). Estos valores corresponden a que la electricidad esté encendida o apagada en la máquina, o a la presencia o ausencia de carga magnética en un medio de almacenamiento.

Partiendo de estos dos estados, es posible formar los esquemas de codificación con los cuales se puede generar letras, números, signos de puntuación y otros caracteres especiales.

### **1.2.2. Segunda generación**

Surgen a finales de los años 50's. Estos lenguajes están conformados por nemotécnicos. Inicialmente se crea un archivo que contiene todas las instrucciones que el ordenador ejecutará. Para dar origen a este "archivo fuente" se utiliza un simple editor, el cual funciona como un procesador de

---

<sup>1</sup> Evolución de los Lenguajes: <http://www.mailxmail.com/curso-componentes-pc-s/generaciones-lenguajes-programacion>

palabras. Seguidamente el “archivo fuente” es traducido a lenguaje máquina mediante la utilización de traductores.

Por la dificultad de programación, los programadores rara vez escriben programas de un tamaño relevante, esto es debido a que a pesar de ser mucho más fácil de utilizar que el código máquina, escribir programas en este tipo de lenguajes sigue siendo altamente detallado y con instrucciones básicas.

Actualmente estos lenguajes todavía siguen siendo utilizados en donde la velocidad y el rendimiento en la ejecución es la clave, como es el caso de la programación de video juegos y para afinar partes importantes y críticas de los programas que son escritos en lenguajes superiores.

Los programas desarrollados en estos lenguajes, así como los desarrollados en lenguaje máquina, son únicos para una computadora en particular. Esta dependencia del ordenador los hace ser un lenguaje de bajo nivel.

### **1.2.3. Tercera generación**

Surgen en los años 60. Son llamados lenguajes de alto nivel debido a que se asemejan al lenguaje con el que nos comunicamos los humanos. Fueron desarrollados con la finalidad de facilitar el proceso de programación. Los programas desarrollados con este tipo de lenguajes son independientes del ordenador para la cual fueron escritos, esto significa que pueden ser ejecutados en cualquier ordenador con muy pocas modificaciones o sin ellas.

Estos lenguajes facilitan la lectura, escritura y comprensión de los programas. Facilitan el aprendizaje debido a que utilizan palabras o comandos

del lenguaje natural, como por ejemplo del idioma inglés. Esta es la razón por la cual es posible compartir el desarrollo entre diferentes programadores. Estos lenguajes, así como los lenguajes ensambladores, deben ser convertidos a lenguaje máquina para poder ser utilizados en el ordenador.

Los lenguajes de tercera generación tienen la capacidad de soportar programación estructurada, esto quiere decir que permiten el uso de estructuras de programación específicas en ramificaciones y ciclos en el flujo del programa.

Existen muchos lenguajes de programación de alto nivel con sus diferentes versiones. Entre algunos de éstos encontramos: FORTRAN, COBOL, BASIC, PASCAL, C, C++, JAVA, entre otros.

#### **1.2.4. Cuarta generación**

Estos lenguajes de programación tienen propósitos específicos, se caracterizan por tener una mayor usabilidad y facilidad en comparación con los lenguajes de tercera generación, permitiendo crear rápidamente prototipos de una aplicación. Los prototipos permiten tener una idea más clara y el funcionamiento de las aplicaciones antes de que se tenga la codificación terminada.

La utilización de los prototipos permite a los involucrados obtener retroalimentación en diversos aspectos como en estructura y diseño, desde el principio del proceso. En estos lenguajes de cuarta generación se sacrifica la flexibilidad con la cual disponían lenguajes de generaciones anteriores, esto a cambio de poder trabajar velozmente.

Su característica distintiva es el énfasis en especificar qué es lo que se debe hacer, en vez de cómo debe de hacerse. Estos lenguajes se apoyan en herramientas de mucho más alto nivel, las cuales son llamadas herramientas de cuarta generación. Entre los lenguajes de cuarta generación se encuentran el SQL y el QBE. Otros tipos de lenguajes de esta generación son los siguientes:

- Lenguajes de presentación, como lenguajes de consultas y generadores de informes
- Lenguajes especializados, como hojas de cálculo y lenguajes de bases de datos
- Generadores de aplicaciones que definen, insertan, actualizan y obtienen datos de la base de datos
- Lenguajes de muy alto, generan el código de la aplicación

#### **1.2.5. Quinta generación**

Entre los lenguajes de esta generación encontramos los que incluyen la inteligencia artificial y sistemas expertos. Estos sistemas a diferencia tienen como objetivo pensar y prever las necesidades que pueda tener el usuario y no simplemente ejecutar un conjunto de órdenes previamente establecidas.

Los sistemas de quinta generación todavía se encuentran en una fase de desarrollo. Los sistemas que utilizan la inteligencia artificial han demostrado ser más complicados de desarrollar de lo previsto, pero los investigadores creen que muy pronto podrán utilizar este tipo de sistemas, los cuales tomarán como entrada hechos; aplicarán un procedimiento de datos y generarán una respuesta adecuada, del mismo modo como responden los seres humanos.

### 1.2.6. Futuro de los lenguajes de desarrollo

La tendencia más marcada de los últimos años es la creciente dependencia de reutilización de componentes de *software* por parte de equipos de desarrollo y programadores. El enfoque de la programación orientada a objetos es muestra de esta tendencia.

El futuro no está escrito, lo único que podemos hacer, es ver las necesidades del presente e imaginar cuáles pueden ser los cambios que puedan darse en el futuro. Algunas de las especulaciones del futuro de los lenguajes de desarrollo son las siguientes:

- Es probable que en un futuro se generalice el uso de programas que se ejecuten en Internet y que se desarrollen programas que hagan uso de los inmensos recursos mundiales disponibles además de enviar y recibir mensajes y archivos. Ejemplo de esto es la tendencia de la computación en la nube, en inglés *Cloud Computing*.
- Es probable que los lenguajes de programación puedan facilitar las tareas del programador y permitan utilizar lenguaje natural, entre ellos están: comandos de voz, como herramienta para que el trabajo fluido, dinámico y flexible. Este concepto pretende disminuir la brecha entre el hombre y máquina.
- Las interfaces utilizadas serán orientadas a un lenguaje comprensible y amigable explorando al máximo los avances tecnológicos del *hardware*, aprovechando así más su capacidad.

- Otra teoría dicta que se dará un cambio paradigmático, en el cual se cambiará de procedimental a declarativo. Esto implica que se deberá aprender a describir qué queremos que suceda en un sistema o aplicación y dejaremos el cómo llevarlo a cabo a las capas subyacentes del compilador y la plataforma que estemos empleando.

## **2. UTILIZACIÓN DE LOS LENGUAJES DE PROGRAMACIÓN EN GUATEMALA**

En Guatemala se utilizan varios lenguajes de programación, algunos más que otros. Se realizó un estudio para poder determinar ¿cuál de estos lenguajes de programación es el más utilizado y por qué de su utilización?

### **2.1. Pregunta de investigación**

¿Cuál es la tendencia de la utilización de los lenguajes de programación en Guatemala, para las organizaciones y los desarrolladores independientes?

### **2.2. Metodología de la investigación**

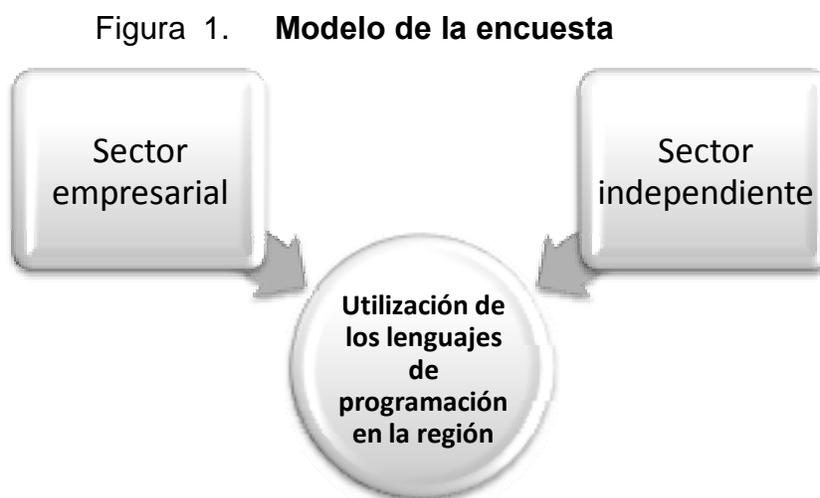
La metodología de investigación incluye un trabajo de campo por medio de la administración de una encuesta con el propósito de recolectar datos para su posterior análisis.

Para realizar este estudio se creó una encuesta electrónica, mediante la herramienta Google Forms, proporcionada por Google. A pesar de que todavía es una herramienta nueva, Google Forms permite la elaboración de reportes de manera sencilla y eficiente.

### 2.3. Modelo de la encuesta

El siguiente diagrama muestra en las partes que se encuentra dividida la encuesta titulada “Utilización de los Lenguajes de Programación en Guatemala.”

- Sector empresarial
- Sector independiente



Fuente: elaboración propia

### 2.4. Estructura de la encuesta

La encuesta se divide en dos secciones. La primera se orienta hacia las organizaciones con el fin de obtener información propia de éstos y así conocer la tendencia de la utilización de los lenguajes de programación dentro del sector empresarial.

La segunda parte de la encuesta se encuentra enfocada a desarrolladores independientes y/o estudiantes, para conocer la tendencia de la utilización de los lenguajes de programación, así como la preferencia de la población sobre un lenguaje respecto del otro.

En ambas secciones de la encuesta se intenta identificar los motivos por los cuales se elige un lenguaje de programación respecto de otro al iniciar un proyecto.

## **2.5. Instrumento de validación**

El siguiente apartado muestra el contenido de la encuesta presentada para realizar el estudio y así mismo, el valor de cada una de las preguntas presentadas en la misma.

El instrumento de evaluación tiene por nombre “Utilización de los Lenguajes de Programación en Guatemala”. En el apartado de anexo se encuentra la encuesta virtual que fue distribuida como parte del estudio.

### **2.5.1. Pregunta a todos los participantes**

Indique el sector al que pertenece.

- Sector público                       Sector privado
- Institución académica    Desarrollador independiente
- Otros

**Validación:** conocer en qué sector se ubica la persona. De este modo poder determinar cuál es el lenguaje más usado en cada sector.

### **2.5.2. Preguntas al sector empresarial**

¿A qué tipo de desarrollo se dedica la Organización?

- Insourcing*    *Outsourcing*    Otros

**Validación:** diferenciar si la encuesta es realizada por una organización que se dedica al *outsourcing* o si es realizada por el área de TI de una organización.

¿Cuáles de los lenguajes de programación se utilizan en la Organización?

- Java  Visual Basic  C  Ruby
- C#  Python  C++  Otro

**Validación:** conocer qué lenguajes de programación son los más utilizados por las organizaciones en la actualidad.

¿Cuál de esos lenguajes de programación es el más utilizado dentro de la organización?

- Java  Visual Basic  C  Ruby
- C#  Python  C++  Otro

¿Cuántos proyectos desarrolla al año en el lenguaje de programación anteriormente seleccionado?

- 1 a 5  6 a 10
- 11 a 20  Más de 20

**Validación:** identificar el lenguaje de programación que más atrae a las organizaciones o el más utilizado.

¿Cuál es la razón principal de utilizar estos lenguajes de programación?

- Eficiencia del lenguaje  Fácil aprendizaje
- Existencia de desarrolladores  Velocidad de desarrollo
- Mano de obra calificada  Demanda del mercado

- Bajo costo/ *Software* Libre
- Tendencia del mercado
- Se ajusta a las necesidades del proyecto
- Otros

**Validación:** identificar las razones por las cuales estos lenguajes de programación son los más utilizados.

¿Piensan utilizar un nuevo lenguaje de programación en la organización en los próximos años?

- Si
- No

Si su respuesta anterior fue sí ¿Qué lenguaje de programación desea utilizar?

- Java
- Visual Basic
- C
- Ruby
- C#
- Python
- C++
- Otro

¿Por qué razón?

- Eficiencia del lenguaje
- Fácil aprendizaje
- Existencia de desarrolladores
- Velocidad de desarrollo
- Mano de obra calificada
- Demanda del mercado
- Tendencia del mercado
- Bajo costo/ *Software* Libre
- Se ajusta a las necesidades del proyecto
- Otros

**Validación:** identificar las tendencias de uso de nuevos lenguajes por parte de las instituciones en los próximos años, así mismo saber qué lenguajes se desean implementar y cuáles son las razones del deseo de utilizar estos lenguajes.

Cuando inician un nuevo proyecto, ¿Cómo eligen el lenguaje de programación para llevar a cabo el proyecto?

- El lenguaje más usado en la organización
- El lenguaje que más desarrolladores involucrados conoce
- El lenguaje que más se ajuste a las necesidades del proyecto
- El lenguaje que el cliente solicite
- El lenguaje más usado en el mercado
- Otros

**Validación:** se pretende conocer cómo se asigna un lenguaje de programación a los proyectos dentro de las instituciones.

### 2.5.3. Preguntas al sector independiente

¿Cuáles de los lenguajes de programación conoce o ha oído mencionar?

- Java  Visual Basic  C  Ruby
- C#  Python  C++  Otro

**Validación:** verificar qué tan amplio es el conocimiento de quien realiza la encuesta o qué tan actualizada es la información que posee.

¿Cuál es el lenguaje de programación que prefiere utilizar?

- Java  Visual Basic  C  Ruby
- C#  Python  C++  Otro

¿Cuántos proyectos desarrolla al año en el lenguaje de programación anteriormente seleccionado?

- 1 a 5  6 a 10
- 11 a 20  Más de 20

**Validación:** conocer qué lenguaje de programación es el más utilizado por los desarrolladores y así saber el número de desarrolladores que se encuentran en el país.

¿Por qué razón?

- Lenguaje solicitado por los clientes u organización de trabajo
- El lenguaje de más rápido desarrollo
- El lenguaje más conocido
- Lenguaje más eficiente
- Otros

**Validación:** identificar las razones por las cuales los desarrolladores se mueven a utilizar más un lenguaje de programación que otro.

¿Cuál es el lenguaje de programación que más le gusta utilizar?

- Java    Visual Basic    C    Ruby
- C#    Python    C++    Otro

**Validación:** Identificar cuál es el lenguaje de programación que más atrae a los desarrolladores en el mercado guatemalteco.

Para los siguientes lenguajes, indique el nivel de conocimiento de los mismos con un número del 0 al 5, en donde 0 indica que desconoce completamente el lenguaje, 3 indica que tiene un amplio conocimiento del lenguaje y 5 indica que lo conoce en un alto porcentaje o en su totalidad.

- Java    Visual Basic    C    Ruby
- C#    Python    C++    Otro

**Validación:** identificar el nivel de conocimiento que poseen en cada uno de los lenguajes de programación.

¿Le gustaría aprender algún lenguaje de programación en los próximos 2 años?

Si  No

Si su respuesta anterior fue sí ¿Qué lenguaje de programación le gustaría aprender?

Java  Visual Basic  C  Ruby  
 C#  Python  C++  Otro

¿Por qué razón le gustaría aprender este nuevo lenguaje?

Demanda del mercado  Satisfacción propia  
 Futuro del mercado  Capacidades del lenguaje  
 Otros

**Validación:** conocer la tendencia de la mano de obra en los diferentes lenguajes, así como las razones por las cuales se da el aprendizaje de nuevos lenguajes de desarrollo.

¿Qué lenguajes de programación considera que el mercado guatemalteco solicita actualmente?

Java  Visual Basic  C  Ruby  
 C#  Python  C++  Otro

**Validación:** identificar como ven los desarrolladores la demanda del mercado actual de Guatemala.

¿Qué lenguajes de programación considera que el mercado guatemalteco demandará en los próximos años?

- Java
- Visual Basic
- C
- Ruby
- C#
- Python
- C++
- Otro

**Validación:** identificar cómo ven los desarrolladores la futura demanda del mercado en Guatemala, de este modo poder obtener una tendencia de la utilización de los lenguajes de desarrollo.

Cuándo inicia un proyecto, ¿Cómo elige el lenguaje de programación con el cual lo llevará a cabo?

- El lenguaje de preferencia
- El lenguaje sobre el cual se tiene mayor conocimiento
- El lenguaje que más se ajuste a las necesidades del proyecto
- El lenguaje que el cliente solicite
- El lenguaje más usado en el mercado
- Otros

**Validación:** se pretende conocer cómo se asigna un lenguaje de programación a los proyectos por parte de los desarrolladores.

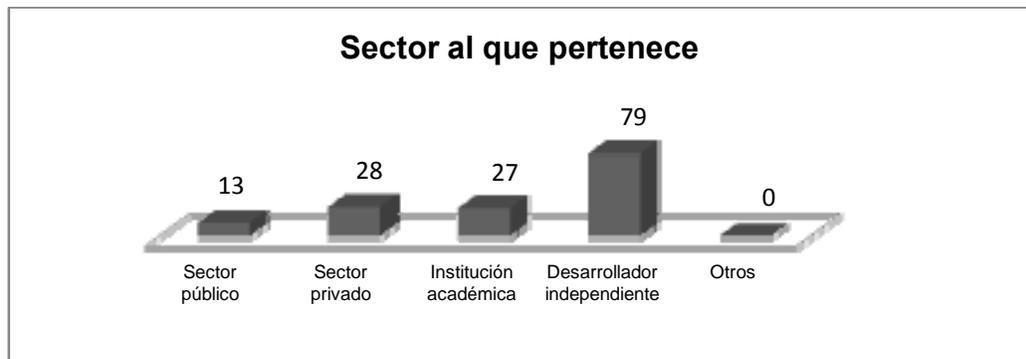


### 3. RESULTADOS DE LA ENCUESTA

La herramienta de Google Forms provee resultados instantáneos para las encuestas publicadas. A continuación se presentan los resultados obtenidos de 83 encuestados.

#### 3.1. Pregunta a todos los participantes

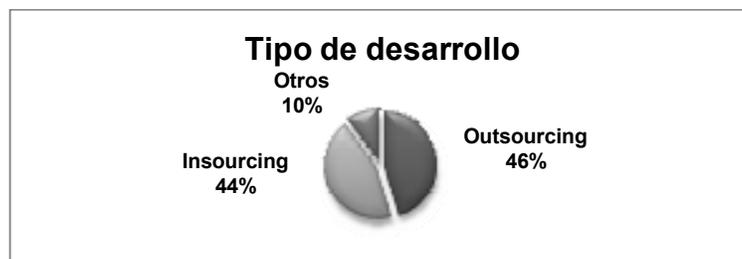
Figura 2. Sector al que pertenece el encuestado



Fuente: elaboración propia

#### 3.2. Preguntas al sector empresarial

Figura 3. Tipo de desarrollo al que se dedica la organización

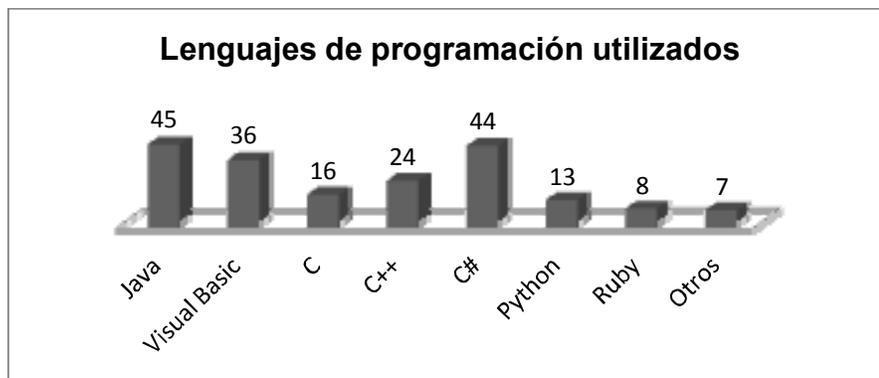


Fuente: elaboración propia

Tabla I. Tipo de desarrollo al que se dedica la organización

Respuestas para otros	
Enseñanza	4
Comercialización de aplicaciones	2
Servicios	1

Figura 4. Lenguajes de programación utilizados dentro de la organización

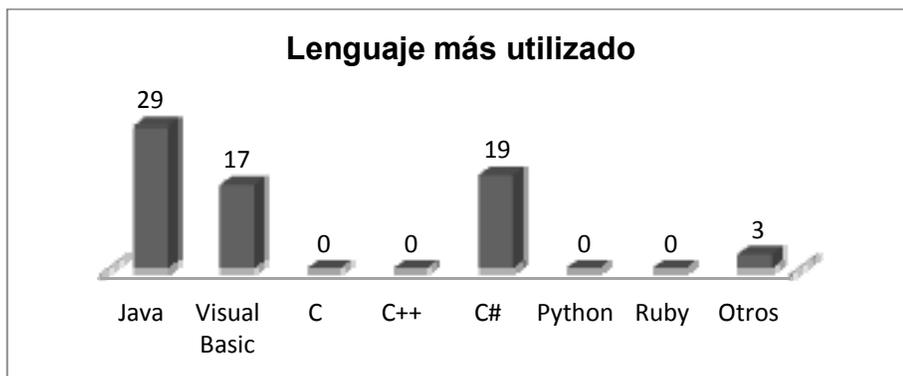


Fuente: elaboración propia

Tabla II. Lenguajes utilizados por la organización

Respuestas para otros	
Delphi	5
Objective-C	2

Figura 5. Lenguaje de programación más utilizado en la organización

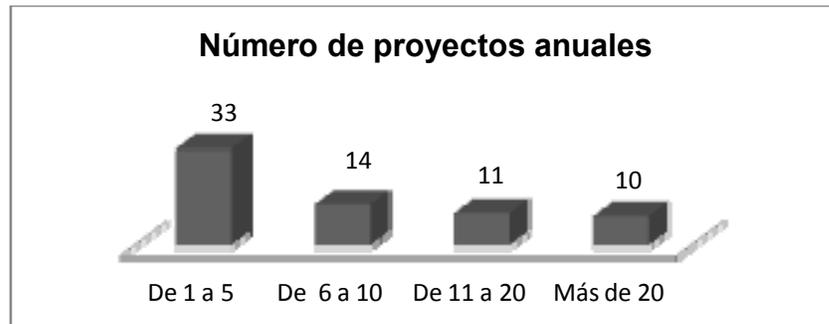


Fuente: elaboración propia

Tabla III. **Lenguaje más utilizado en la organización**

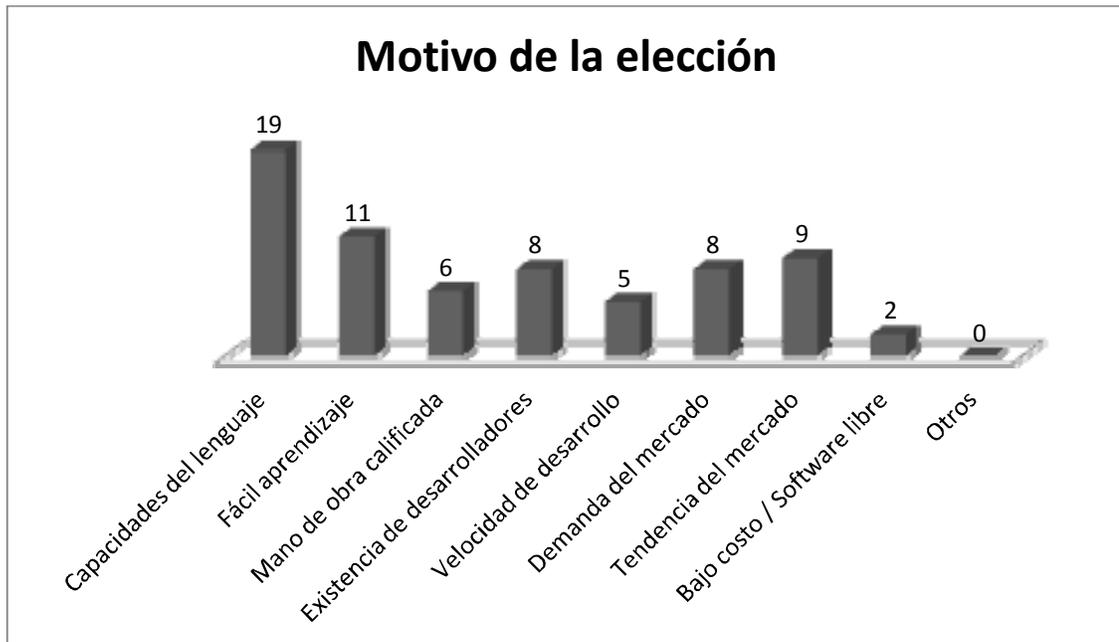
Respuestas para otros	
Delphi	3

Figura 6. **Número de proyectos desarrollados anualmente en la organización**



Fuente: elaboración propia

Figura 7. **Motivo de la elección del lenguaje más utilizado en la organización**



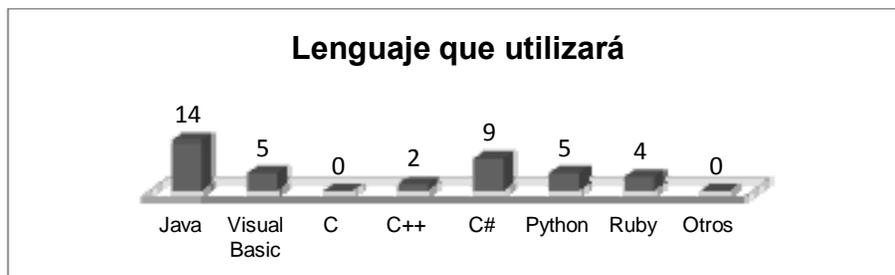
Fuente: elaboración propia

Figura 8. **Porcentaje de organizaciones que utilizarían un nuevo lenguaje de programación en sus desarrollos**



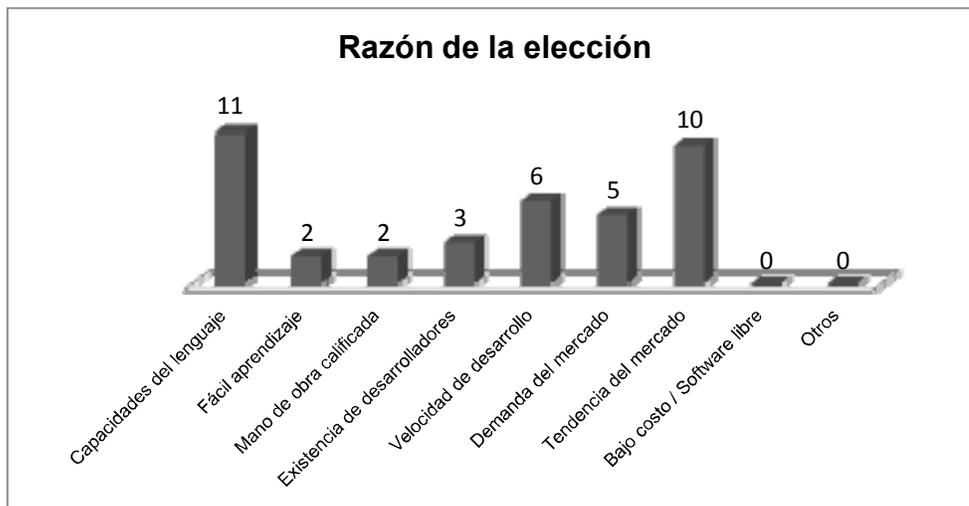
Fuente: elaboración propia

Figura 9. **Lenguaje de programación que utilizaría la organización**



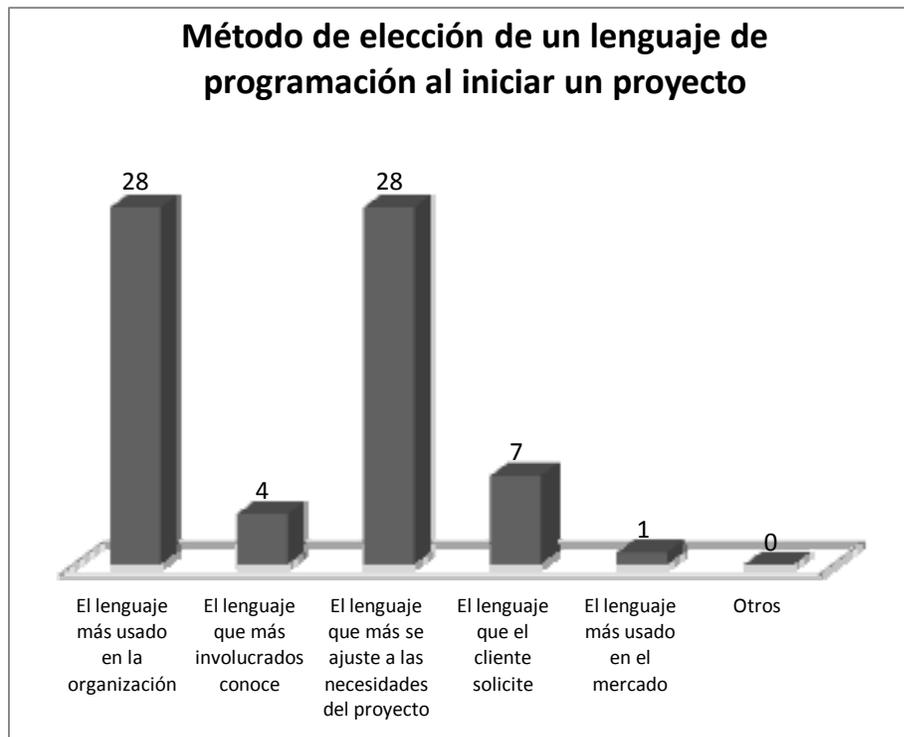
Fuente: elaboración propia

Figura 10. **Razón por la cual la organización utilizaría el nuevo lenguaje de programación**



Fuente: elaboración propia

Figura 11. **Método por el cual la organización selecciona el lenguaje de programación al iniciar un proyecto**



Fuente: elaboración propia

### 3.3. Conclusiones del sector empresarial

En general, las organizaciones utilizan Java, VB, C, C++, y C# en sus proyectos, pero la utilización de los lenguajes de programación se encuentra encabezada por Java seguido de C# y luego VB con gran ventaja sobre C, C++, Ruby y Python. Las organizaciones eligen el lenguaje de programación por las siguientes situaciones:

- Capacidades del lenguaje
- Fácil aprendizaje
- Existencia de desarrolladores

Esto da una perspectiva que la principal razón de la utilización de un lenguaje de desarrollo es debido a la existencia de mano de obra en el mercado y/o la facilidad de capacitar a la misma en determinado lenguaje.

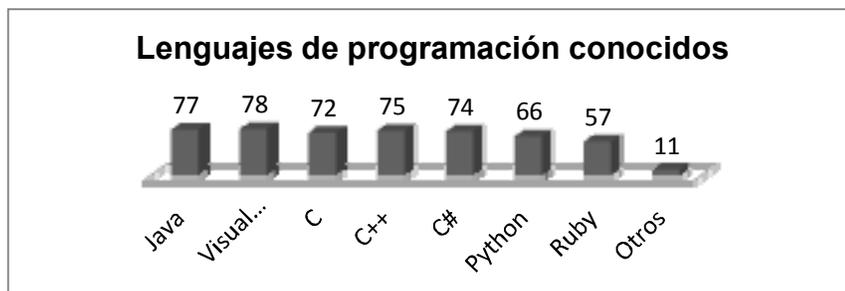
La gran mayoría de las organizaciones pretenden utilizar nuevos lenguajes de programación en los próximos años, principalmente pretenden introducir Java en sus desarrollos seguido de C#. La razón principal de esta tendencia es debido a la eficiencia presentada por estos dos lenguajes de programación.

Cuando las organizaciones se embarcan en nuevos proyectos, eligen el lenguaje de programación seleccionando el más utilizado dentro de la organización y otros lo eligen porque es el que se ajusta a las necesidades del proyecto, sobresaliendo la primer elección.

A manera de resumen, los lenguajes más utilizados en las organizaciones son Java, C# y VB. Las organizaciones eligen actualmente estos lenguajes debido a la capacidad del lenguaje y a la facilidad de encontrar o capacitar la mano de obra. Se demuestra una marcada tendencia a la utilización de Java y C# en los próximos años en las organizaciones de Guatemala.

### 3.4. Preguntas al sector independiente

Figura 12. Lenguajes de programación conocidos por el desarrollador

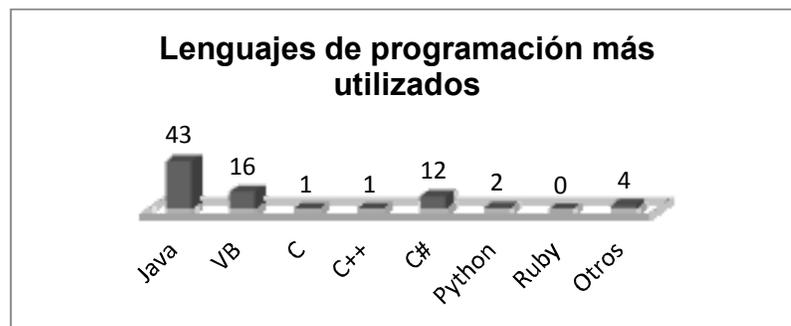


Fuente: elaboración propia

Tabla IV. **Lenguajes conocidos por el desarrollador**

Respuestas para otros	
Cobol	2
Fortran	1
Delphi	4
X++	1
D	2
Ensamblador	1

Figura 13. **Lenguaje de programación más utilizado por el desarrollador**

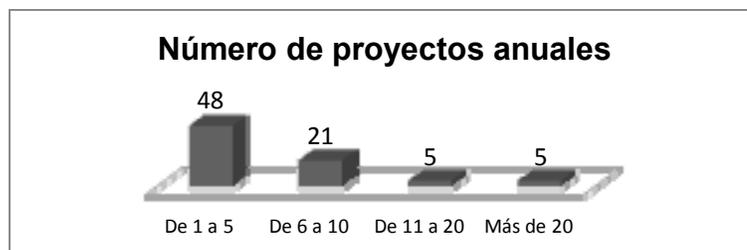


Fuente: elaboración propia

Tabla V. **Lenguaje más utilizado por el desarrollo**

Respuestas para otros	
X++	1
Delphi	3

Figura 14. **Número de proyectos que el desarrollador realiza anualmente con el lenguaje de programación más utilizado**



Fuente: elaboración propia

Figura 15. **Razón por la cual el desarrollador selecciona el lenguaje de programación**

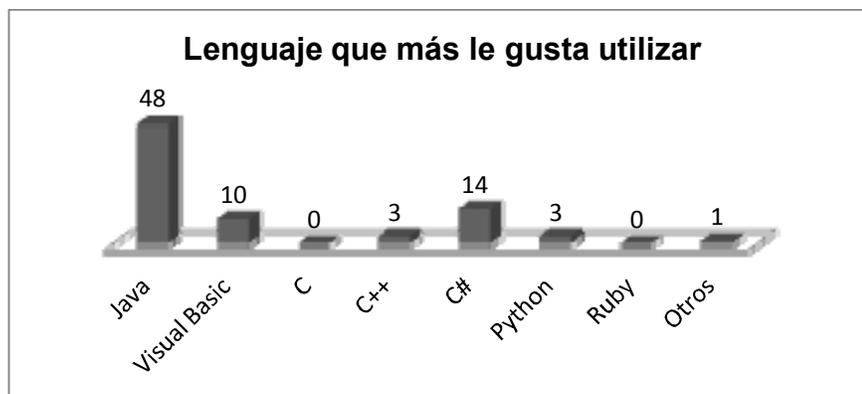


Fuente: elaboración propia

Tabla VI. **Motivo de selección del lenguaje**

Respuestas para otros	
Lenguaje potente, multiplataforma, para el cual existen varios <i>frameworks</i> de trabajo	1

Figura 16. **Lenguaje de programación que más le gusta utilizar al desarrollador**

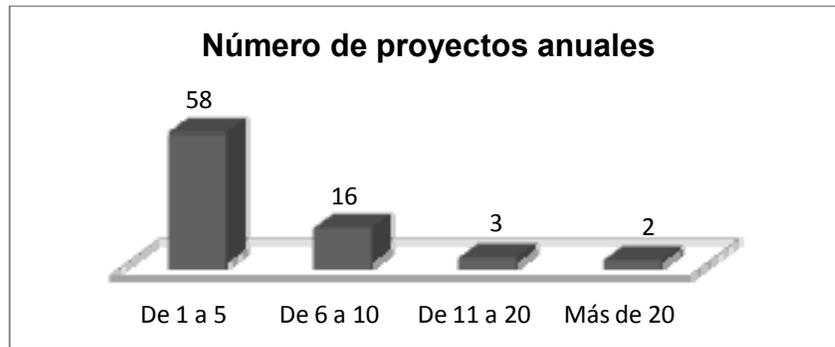


Fuente: elaboración propia

Tabla VII. **Lenguaje que más le gusta utilizar**

Respuestas para otros	
Objective-C	1

Figura 17. **Número de proyectos que el desarrollador realiza anualmente en el lenguaje de programación de su preferencia**

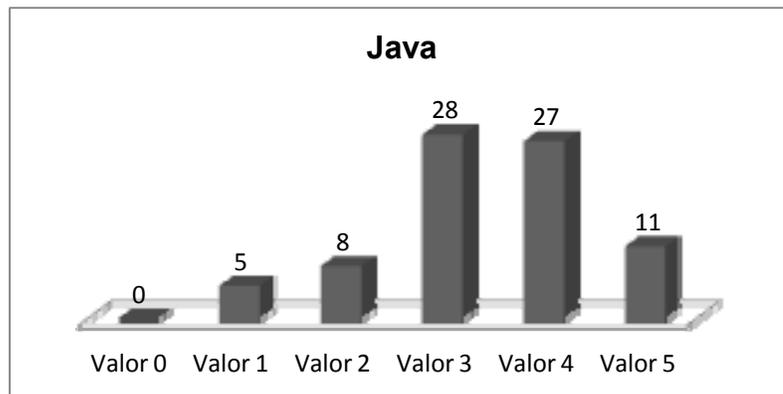


Fuente: elaboración propia

Para los siguientes lenguajes, indique el nivel de conocimiento de los mismos, en donde 0 indica que desconoce completamente el lenguaje, 3 indica que tiene un amplio conocimiento del lenguaje y 5 indica que lo conoce en un muy alto porcentaje o en su totalidad.

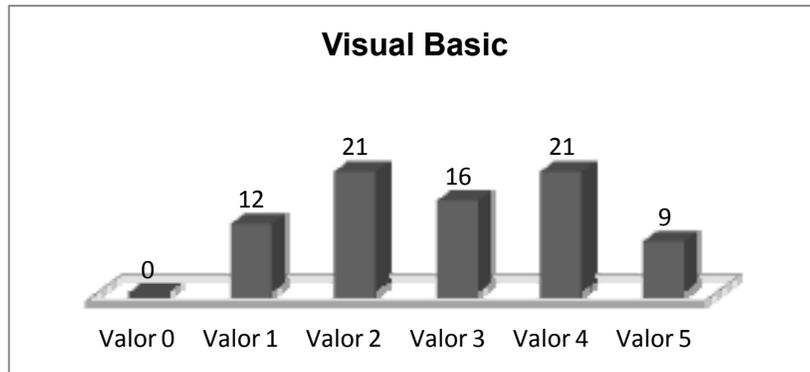
Figura 18. **Conocimiento que tiene el desarrollador sobre el lenguaje**

**Java**



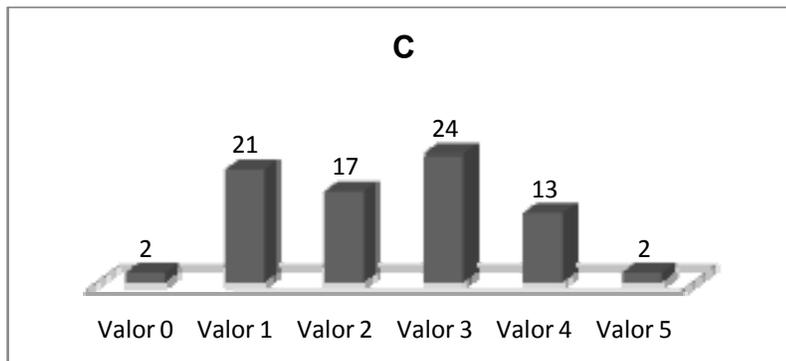
Fuente: elaboración propia

Figura 19. **Conocimiento que tiene el desarrollador sobre el lenguaje Visual Basic**



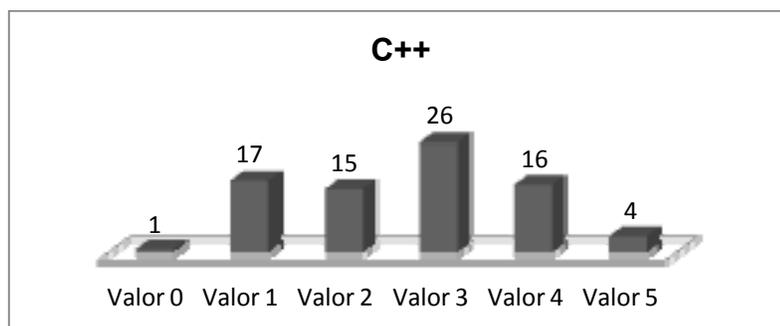
Fuente: elaboración propia

Figura 20. **Conocimiento que tiene el desarrollador sobre el lenguaje C**



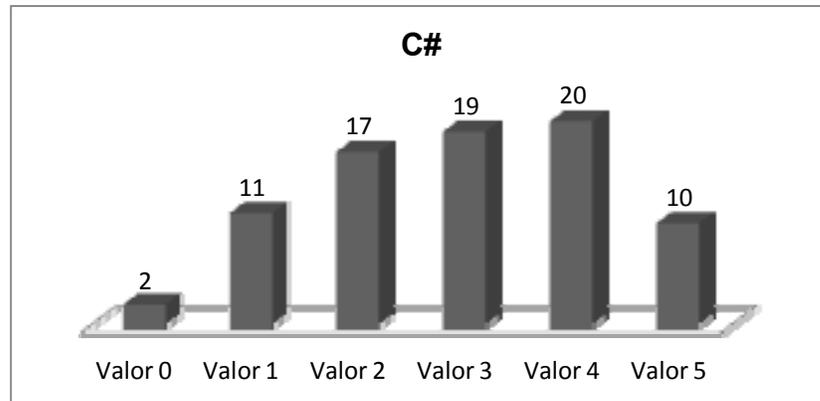
Fuente: elaboración propia

Figura 21. **Conocimiento que tiene el desarrollador sobre el lenguaje C++**



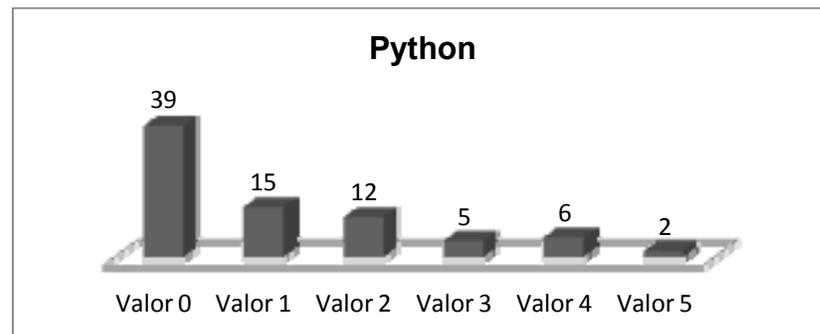
Fuente: elaboración propia

Figura 22. **Conocimiento que tiene el desarrollador sobre el lenguaje C#**



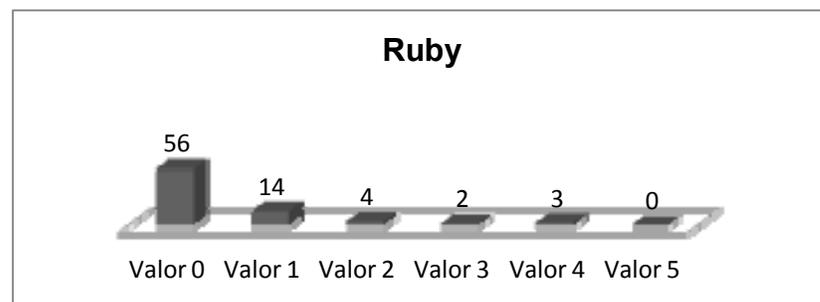
Fuente: elaboración propia

Figura 23. **Conocimiento que tiene el desarrollador sobre el lenguaje Python**



Fuente: elaboración propia

Figura 24. **Conocimiento que tiene el desarrollador sobre el lenguaje Ruby**



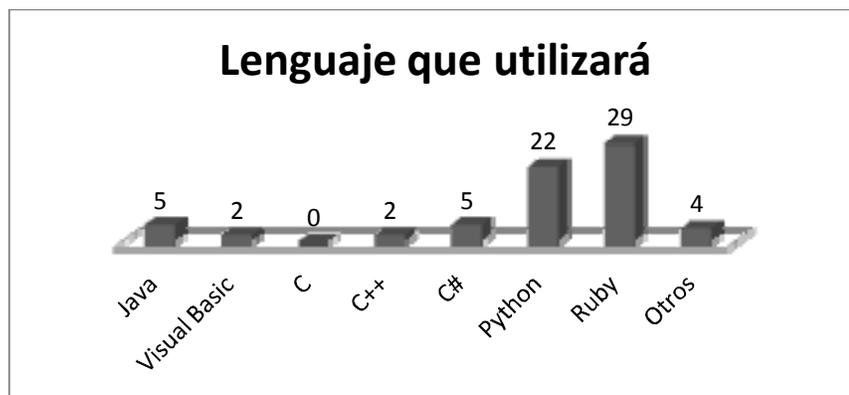
Fuente: elaboración propia

Figura 25. **Porcentaje de desarrolladores que utilizaría un nuevo lenguaje de programación**



Fuente: elaboración propia

Figura 26. **Lenguaje de programación que al desarrollador le gustaría utilizar**



Fuente: elaboración propia

Tabla VIII. **Lenguaje que le gustaría utilizar al desarrollador**

Respuestas para otros	
Vala	1
Scala	1
D	2

Figura 27. **Motivo por el cual el desarrollador aprendería un nuevo lenguaje de programación**

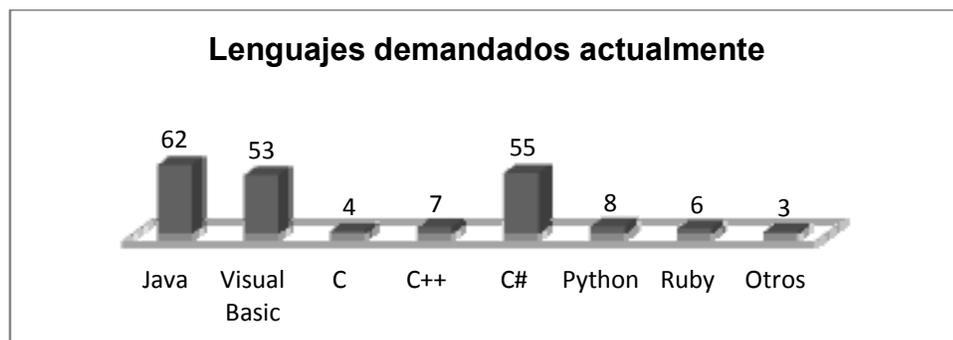


Fuente: elaboración propia

Tabla IX. **Motivación para aprendería un nuevo lenguaje**

Respuestas para otros	
Agilidad de desarrollo	3
Software libre	5
Deseo de aprendizaje	11
Tendencia del mercado	3
Utilizado en animaciones en blender	1

Figura 28. **Lenguajes que el desarrollador considera que el mercado nacional actual demanda**

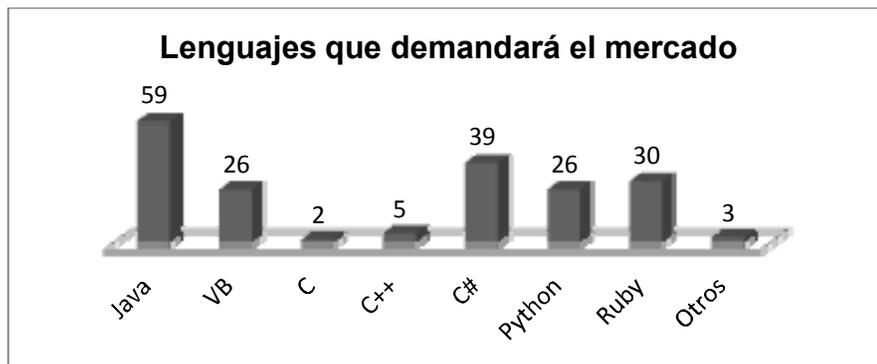


Fuente: elaboración propia

Tabla X. **Lenguajes demandados por el mercado nacional**

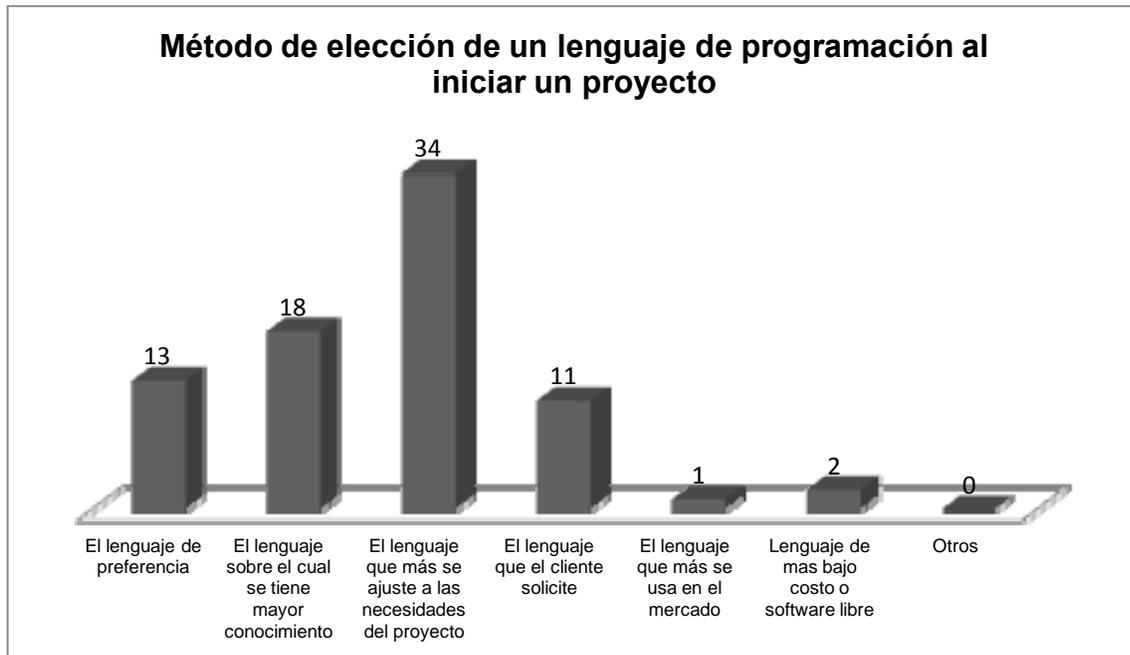
Respuestas para otros	
No sabe, no responde	3

Figura 29. **Lenguaje de programación que el mercado nacional solicitará en los próximos años**



Fuente: elaboración propia

Figura 30. **Método por el cual el desarrollador selecciona el lenguaje de programación al iniciar un proyecto**



Fuente: elaboración propia

### **3.5. Conclusiones del sector independiente**

Los desarrolladores independientes de Guatemala se muestran más afines a lenguajes como Java, VB y C#. La encuesta refleja que existe un alto número de desarrolladores independientes con gran conocimiento de lenguajes de programación como Java, VB y C#, pero no así, de Ruby o Python en donde el número de desarrolladores con cierto conocimiento es muy bajo y el número de desarrolladores con alto conocimiento es más bajo aún.

Se puede observar una tendencia muy marcada sobre los lenguajes de programación que desean aprender los desarrolladores guatemaltecos. La tendencia se inclina hacia Ruby, principalmente seguida de Python. Las razones de esta tendencia apuntan hacia la rapidez de desarrollo que estos dos lenguajes proveen.

A pesar de la tendencia a que los desarrolladores aprendan Python o Ruby, la mayoría de desarrolladores concuerdan en que la tendencia de utilización de los lenguajes de programación en Guatemala continuará enfocada en Java, C# y VB.

La mayoría de desarrolladores elige el lenguaje de programación con base en las necesidades del proyecto, pero otro gran porcentaje lo elige con base en los conocimientos que ellos posean sobre el lenguaje, el lenguaje de mayor preferencia o el que los clientes les soliciten.



## 4. LENGUAJES MÁS UTILIZADOS MUNDIALMENTE

TIOBE *Programming Community* provee cada mes un indicador actualizado de los lenguajes de programación más populares. Este estudio está basado en el número de ingenieros calificados, cursos y vendedores.

Motores de búsquedas populares como Google, MSN, Yahoo y Youtube son utilizados para calcular el puntaje de cada lenguaje. Este índice no es acerca del mejor lenguaje de programación o el que posee más líneas de código sino del más popular y más usado.

La siguiente tabla muestra el índice de utilización de los lenguajes de programación, obtenido por el estudio de TIOBE<sup>1</sup>. Un estudio similar se encuentra en el sitio de Langpop<sup>2</sup>. (Vease: Figura 31)

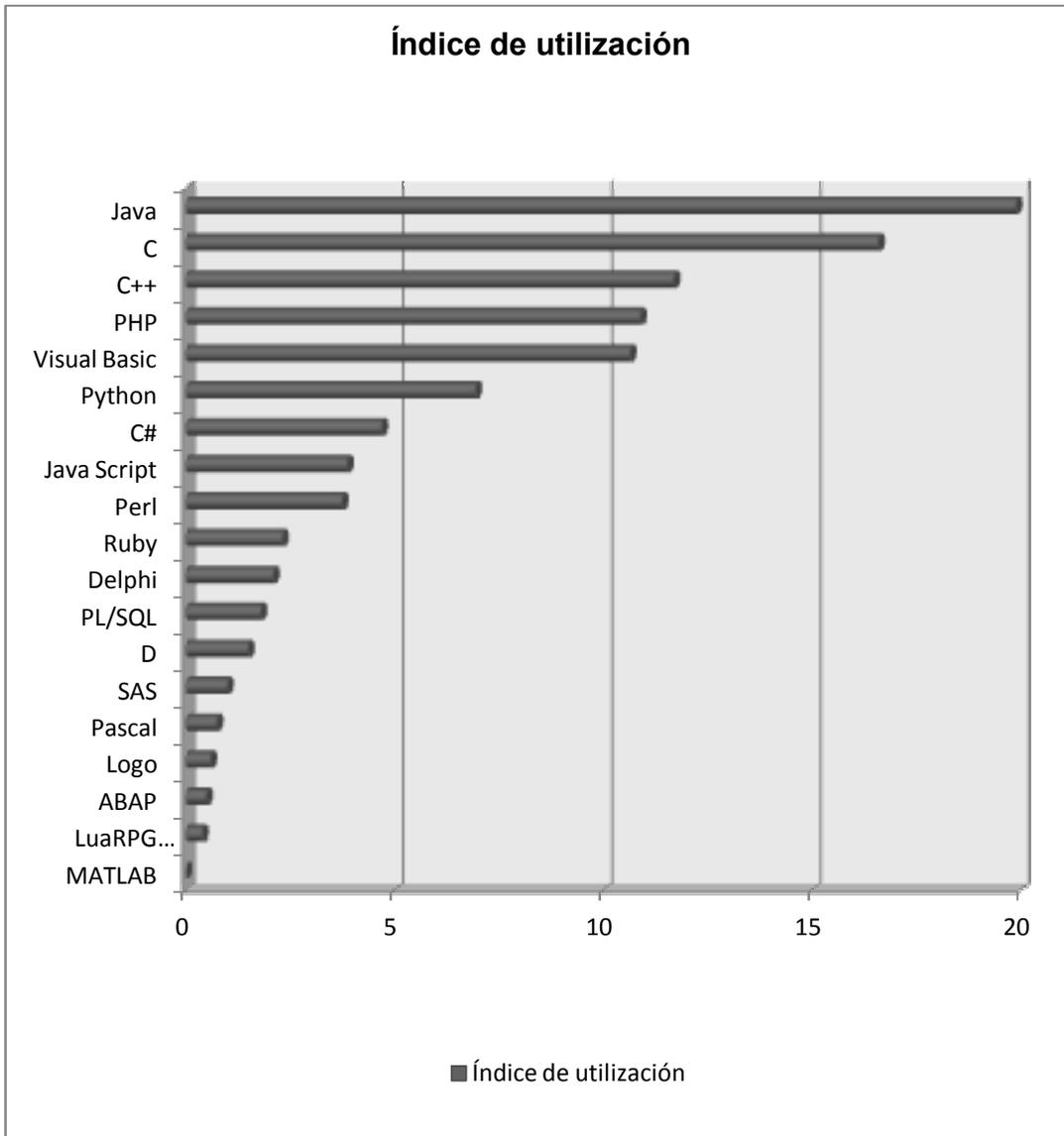
El presente documento está centrado en los lenguajes de programación de aplicaciones de escritorio, por tal razón el siguiente gráfico muestra los 10 lenguajes de programación, de aplicaciones de escritorio, más utilizados a nivel mundial.

---

<sup>1</sup> TIOBE: <http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html>

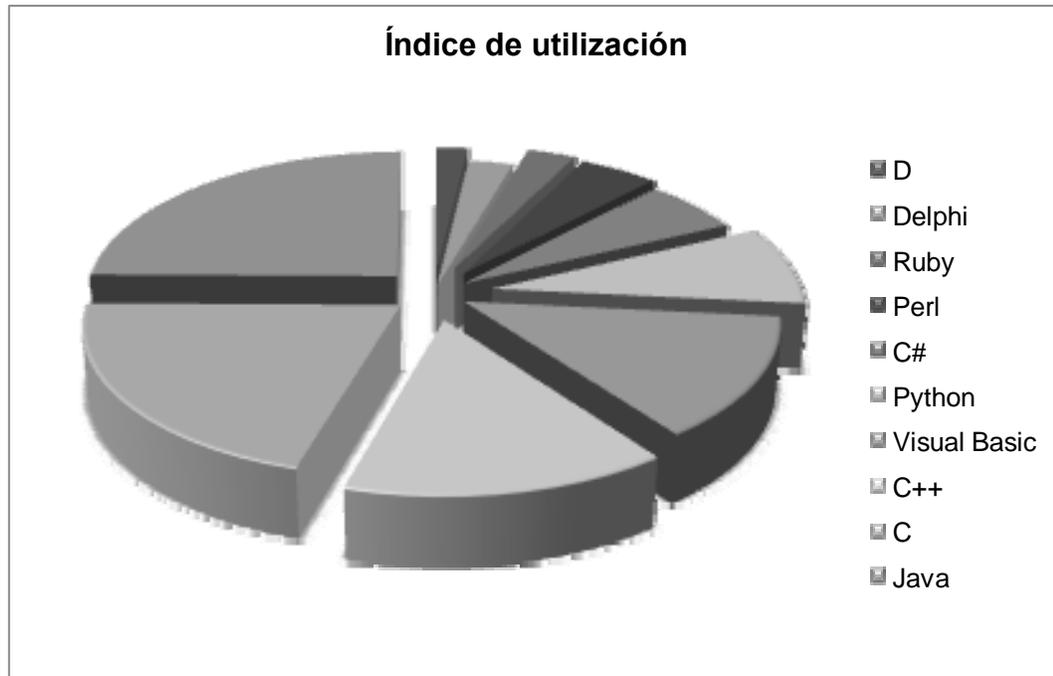
<sup>2</sup> Langpop: <http://www.langpop.com/>

Figura 31. **Porcentaje de la utilización mundial de los lenguajes de programación**



Fuente: Langpop. 2010. Recopilado el 01 de junio de 2010. <http://www.langpop.com/>.

Figura 32. **Porcentaje de la utilización mundial de lenguajes de programación de aplicaciones de escritorio**



Fuente: TIOBE. 2010. Recopilado el 01 de junio de 2010.

<http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html>.

#### 4.1. Java

Fue diseñado en 1990 como un *software* para dispositivos electrónicos de consumo, por Sun Microsystems<sup>1</sup> quien a principios de los 90's intentó ingresar al mercado de la electrónica de consumo desarrollando programas para pequeños dispositivos electrónicos.

Este lenguaje de programación fue nombrado inicialmente como *Oak* (roble en inglés), debido a la existencia de un roble en las cercanías del lugar de trabajo de los promotores de este lenguaje, pero luego tuvo que ser cambiado porque dicho nombre se encontraba registrado por otra empresa.

<sup>1</sup> Sun Microsystems: Empresa de informática fabricante de semiconductores y *software*, fundada en 1982.

#### **4.1.1. Por qué crear Java**

Uno de los factores determinantes de la creación de Java, fue la creciente necesidad de interfaces mucho más cómodas e intuitivas que los sistemas basados en ventanas que se encontraban hasta ese momento.

Los dispositivos electrónicos son controlados utilizando microprocesadores de bajo precio y pocas capacidades, los cuales varían cada cierto tiempo y que utilizan diversos conjuntos de instrucciones. Por esta razón se necesitaba un lenguaje que presentara fiabilidad del código, facilidad de desarrollo y sobre todo una enorme diversidad de controladores electrónicos, características que presenta Java, el cual permite escribir un código común para todos los dispositivos.

Los creadores de Java estudiaron gran cantidad de lenguajes existentes y se basaron en el conocimiento obtenido para recoger las características esenciales que debería de tener un moderno y potente lenguaje de programación, para realizar esto eliminaron todas aquellas funciones que no eran imprescindibles.

El primer proyecto en el cual se aplicó Java fue “El Proyecto Green”. Este proyecto consistía en la creación de un sistema para el control total de los aparatos electrónicos y el entorno de un hogar. Poco tiempo después, Java fue aplicado a otro proyecto, llamado VOD, Video (*Video On Demand*). En este proyecto, Java se utilizaba como una interfaz entre el usuario y la televisión interactiva. Se pensaba que el desarrollo de aplicaciones para la televisión interactiva sería el principal campo de aplicación de este lenguaje. Estos dos proyectos iniciales nunca llegaron a ser comercializados, pero sí fueron desarrollados enteramente con Java primitivo.

Sun Microsystems se dio cuenta que la televisión interactiva no tendría un gran éxito a corto plazo, por lo cual era necesario desarrollar nuevas estrategias que produjeran beneficios. Una de sus nuevas estrategias fue la aplicación de Java en Internet, pero en ese momento no se consideró muy productiva.

En 1996 Sun Microsystems distribuye su primer entorno de programación, el *Java Development Kit* (JDK). En el transcurso de ese año mejora el lenguaje de programación drásticamente al incluir nuevas funcionalidades como el estándar JDBC, para el acceso a las bases de datos, los JavaBeans, las funcionalidades de impresión, entre otros.

Con el pasar del tiempo Java se convierte en un poderoso entorno de programación y de ejecución de programas para producir aplicaciones de valor comercial, las cuales son comparables hasta con aplicaciones escritas en lenguajes de programación más recientes. Las aplicaciones escritas en Java proveen independencia de la plataforma, robustez y fiabilidad.

#### **4.1.2. Ventajas**

- Orientado a objetos
- Es sencillo y práctico, provee toda la funcionalidad de un lenguaje potente, pero sin las características menos utilizadas y confusas de éstos
- Sus sistemas de verificación le proveen robustez, ya que realiza verificaciones de problemas tanto en tiempo de ejecución como en tiempo de corrida, por medio del cual detecta errores en el ciclo de desarrollo

- La ejecución del código Java es segura y fiable. Los programas no acceden directamente a la memoria del ordenador, siendo imposible que un programa escrito en Java pueda acceder a los recursos del ordenador sin que esta operación le sea permitida de forma explícita
- Es completamente multi-plataforma
- Es un entorno de programación multi-hilo
- Más que un lenguaje, es toda una plataforma y se encuentra apoyada por muchas empresas
- Soporta el desarrollo rápido de aplicaciones, y su *toolkit* resuelve muchas de las tareas del programador
- Está construido con extensas capacidades de interconexión TCP/IP. Existen librerías de rutinas que permiten interactuar con protocolos como http y ftp
- Provee una administración automática de la memoria. El recolector de basuras es quien determina cuándo se puede liberar la memoria ocupada por un objeto

#### **4.1.3. Desventajas**

- Debido a la capacidad de portabilidad que Java presente, el código de Java es convertido a un código intermedio, llamado Código *Byte*, antes de convertirlo a código máquina. Este paso adicional provoca que los

programas escritos en java tiendan a ser más lentos que los programas escritos en otros lenguajes como C

## 4.2. Lenguaje C

Este lenguaje nació en los Laboratorios Bell de AT&T<sup>1</sup>, se le ha asociado estrechamente con el Sistema Operativo UNIX<sup>2</sup> debido a que su desarrollo se realizó en este sistema operativo y tanto UNIX como el propio compilador de C y casi la totalidad de los programas y herramientas de UNIX fueron escritos en C. La eficiencia de C y su claridad hicieron que el lenguaje ensamblador apenas haya sido utilizado en UNIX.

El lenguaje de programación C fue inspirado en el lenguaje B<sup>3</sup> creado en 1969, por Ken Thompson. En 1972, Dennis Ritchie modifica el lenguaje B creando el así el lenguaje C con el cual reescribe UNIX. C proporcionó una novedad respecto B, la cual fue el diseño de tipos y estructuras de datos.

Una de las peculiaridades de C es su riqueza de operadores. Puede decirse que prácticamente dispone de un operador para cada una de las posibles operaciones en código máquina.

### 4.2.1. Evolución

En 1967, Martin Richard diseña el lenguaje BCPL con el objetivo de utilizarlo para escribir compiladores y *software* de sistemas operativo. En 1969,

---

<sup>1</sup> Laboratorios Bell de AT&T: Son varios centros de investigación científica y tecnológica ubicados en más de diez países y que pertenecen a la empresa estadounidense Lucent Technologies.

<sup>2</sup> UNIX: Es un sistema operativo portable, multitarea y multiusuario; desarrollado, en principio, en 1969 por un grupo de empleados de los laboratorios Bell de AT&T.

<sup>3</sup> B: Surge en los laboratorios de Bell en 1969, creado por Ken Thompson con contribución de Dennis Ritchie con intención de recodificar el sistema operativo UNIX, que en la fase de arranque está escrito en lenguaje ensamblador.

Ken Thompson escribe el Lenguaje B, en los laboratorios de Bell, con el objetivo de recodificar el sistema operativo UNIX, con un lenguaje de alto nivel más portable y flexible. UNIX se encontraba hasta ese momento escrito en lenguaje ensamblador.

En 1972, Dennis Ritchie modifica el lenguaje B, creando el lenguaje C y reescribe el sistema operativo UNIX en dicho lenguaje. A este lenguaje le añade características nuevas como diseño de tipos y estructuras de datos. 1983 ANSI estandariza C. Entre 1983 y 1984 surge el "C con Clases", lenguaje C++. En 1985 Bjarne Stroustrup crea C++ en los laboratorios de Bell.

#### **4.2.2. Proyectos en los que se utiliza C**

Esencialmente el lenguaje C está diseñado para crear proyectos en los cuales se destaca la optimización y rendimiento, para muchos programadores, este lenguaje es altamente transportable y permite programar lo inprogramable.

La compilación de un programa escrito en C se realiza en varias etapas, las cuales generalmente son automatizadas y ocultas por los entornos de desarrollo:

- Pre procesado: modifica el código fuente en C mediante una serie de instrucciones simplificando así el trabajo del compilador
- Compilación: genera el código objeto a partir del código ya pre procesado
- Enlazado: une los códigos objeto de los distintos módulos y bibliotecas externas para generar el programa ejecutable final

### **4.2.3. Ventajas**

- Lenguaje eficiente, permite utilizar sus características de bajo nivel para realizar implementaciones óptimas
- A pesar de ser un lenguaje de bajo nivel, es portable en existencia, habiendo compiladores para casi todos los sistemas operativos conocidos
- Facilita la realización de programas modulares y/o utilizar código o bibliotecas existentes
- Economía de las expresiones
- Posee abundancia en operadores y tipos de datos
- Presenta una codificación en alto y bajo nivel simultáneamente
- Reemplaza ventajosamente la programación en ensamblador
- Utilización natural de las funciones primitivas del sistema
- No está orientado a ningún área en especial

### **4.2.4. Desventajas**

- Carece de instrucciones de entrada/salida, de instrucciones para manejo de cadenas de caracteres, por lo que este trabajo queda para la biblioteca de rutinas, lo cual produce cierta pérdida de portabilidad
- La excesiva libertad en la escritura de los programas puede llevar a errores en la programación, que por ser correctos sintácticamente, no se detectan a simple vista
- Las precedencias de los operadores producen, en ocasiones, complicaciones en la construcción de las expresiones
- El mayor problema que presenta el desarrollo con el lenguaje C frente a otros lenguajes es su diferencia de velocidad de desarrollo, siendo más lento desarrollando en C
- El mantenimiento es más difícil y costoso que con el resto de lenguajes

### **4.3. Lenguaje C++**

A mediados de 1980, Bjarne Stroustrup diseña este lenguaje de programación con la intención de extender el exitoso lenguaje C dotando a este con mecanismos que le permitieran manipular objetos. Desde el punto de vista de los lenguajes orientados a objetos, C++ es un lenguaje híbrido.

Posteriormente se le añadieron ciertas facilidades de programación genérica, las cuales fueron sumadas a los otros dos paradigmas que ya se encontraban admitidos. Esta es la razón por la cual se suele decir que C++ es un lenguaje multiparadigma.

En C++ existe la posibilidad de redefinir los operadores, esto es una particularidad del lenguaje, también es posible crear nuevos tipos que se comporten como tipos fundamentales y con C++ es posible trabajar tanto a alto como a bajo nivel.

En 1983, Rick Mascitti propone el nombre de C++ para el nuevo lenguaje de programación cuando éste fue utilizado por primera vez fuera del laboratorio científico. Inicialmente se le había llamado “C con clases”, pero luego quedó aceptado el término C++, el cual significa “Incremento de C” y se refiere a que C++ es una extensión de C.

#### **4.3.1. Aspectos técnicos**

Combinando la utilización de plantillas y un estilo específico para denotar tipos y variables, la STL (*Standard Template Library*) ofrece una serie de funciones que representan operaciones comunes, y cuyo objetivo es la parametrización de las operaciones en las que estas funciones se ven

involucradas, de modo que su lectura, comprensión y mantenimiento sean más fáciles de realizar.

#### **4.3.2. Ventajas**

- Al compilar un programa en C++ se genera código objeto, nativo de cada máquina.
- Es una extensión de C, por tanto, muchos programadores encontrarán sencilla la transición. Esto les permite seguir haciendo cosas a la antigua usanza
- Permite controlar la memoria y provee al programador de capacidad de programar a bajo nivel

#### **4.3.3. Desventajas**

- No es un lenguaje multiplataforma. Es posible lograr que las aplicaciones se ejecuten en varios sistemas operativos, pero se requiere de cierto esfuerzo
- No presenta una arquitectura estándar de desarrollo orientado a Internet
- Bastantes dogmas de la programación orientada a objetos se sacrifican para hacer que C++ sea una extensión de C
- No presenta un *toolkit* rico como otros lenguajes
- Aunque existen varias librerías en la red para C++, no son estándar del lenguaje, y algunas deben de ser pagadas
- Es más complicado de aprender que otros lenguajes, debido a que es en parte C, y el exceso de libertad puede ocasionar ciertos problemas

## 4.4. Visual Basic

En 1990 se introduce Visual Basic 1.0. Antes de este lenguaje de programación, construir un simple programa basado en Microsoft Windows era bastante complicado, pero con la introducción de Visual Basic los programadores pudieron, por primera vez crear aplicaciones de Windows en un ambiente gráfico e intuitivo.

Visual Basic trajo consigo el concepto de “*Drag and Drop*”<sup>1</sup>, bajo el cual los controladores no deben de ser programados, en cambio sólo deben de ser arrastrados sobre un formulario. Visual Basic trajo consigo un renacimiento del desarrollo de las aplicaciones basadas en Microsoft Windows.

La comunidad de Visual Basic ha crecido en los últimos años, hasta llegar a ser la mayor comunidad de desarrolladores de *software* para aplicaciones Microsoft Windows del mundo.

Visual Basic fue pensado para ser un lenguaje muy táctico, para convertirse en la herramienta de programación a largo plazo, estratégica, gráfica y orientada a objetos.

### 4.4.1. Características y trayectoria

Visual Basic 1.0 (mayo 1991). Incluyó la Tecnología Embebida de Basic, la cual había sido desarrollada originalmente en Microsoft QuickBasic 4.0. Además incluía una herramienta compiladora de diseño simple, originalmente diseñada para Windows 3.0 pero que nunca fue utilizada con tal fin.

---

<sup>1</sup> Drag and Drop: es una expresión informática que se refiere a la acción de mover con el ratón objetos de una ventana a otra o entre partes de una misma ventana

Visual Basic 2.0 (noviembre 1992). Fue distribuida de dos maneras, la versión estándar y la profesional. Proveía a los desarrolladores de un funcionamiento mejorado, una mayor capacidad para crear aplicaciones de más sofisticadas y de mayor tamaño.

Visual Basic 3.0 (mayo 1993). Combina el motor de la base de datos de Microsoft Access 1.1 con un conjunto rico de controles. Estos cambios permitieron a los desarrolladores conectar fácilmente a las bases de datos en un ambiente cliente/servidor usando un diseñador visual intuitivo. También incluyó un motor para la creación de reportes llamado Crystal Reports.

Visual Basic 4.0 (septiembre 1995). Fue distribuida en Edición Estándar y Edición Profesional, así como una nueva edición destinada al nivel empresarial y el desarrollo en equipo. Presentaba nuevas capacidades tales como: automatización remota, control de datos remoto, y una versión integrada de Microsoft Visual SourceSafe para la dirección de la configuración y realización de diferentes versiones.

Visual Basic 5.0 (marzo 1997). Programación en nuevos niveles de funcionamiento en el ambiente libre que representa internet. Las características, tales como, el compilador del código nativo introdujeron aumentos del funcionamiento hasta el 2 000 por ciento.

Visual Basic 6.0 (junio 1998). En esta versión de VB fue incluido un diseñador de clases Web el cual simplificó la creación de las aplicaciones Web proporcionando un modelo intuitivo del objeto para el servidor Web. Incluyó un diseñador de páginas *DHTML*<sup>1</sup> que permitió la creación de aplicaciones para

---

<sup>1</sup> DHTML: designa el conjunto de técnicas que permiten crear sitios web interactivos utilizando una combinación de lenguaje HTML estático, un lenguaje interpretado en el lado del cliente, el lenguaje de hojas de estilo en cascada y la jerarquía de objetos de un sistema de datos.

Microsoft Internet Explorer<sup>1</sup> que combinarán la riqueza de HTML dinámico (DHTML) con el funcionamiento y la productividad de Visual Basic.

Visual Basic .Net (actualidad): se continúa ampliando las posibilidades del desarrollador aplicaciones en Visual Basic. La orientación a objetos que presenta Visual Basic, como la herencia, manejo estructural excepcional y construcciones con parámetros, han hecho que la programación en Visual Basic pueda hacerse de manera elegante, simple y de mantenimiento sencillo.

#### **4.4.2. Proyectos en los que se utiliza Visual Basic**

A continuación se describen los proyectos en los cuales se utiliza Visual Basic:

- Sistemas Bancarios
- Sistema para envío de contenidos vía bluetooth
- Extracción de datos de la Web
- Gestión, control y facturación de PYME's
- Control de inventarios
- Aplicaciones generales de escritorio

El principal uso de Visual Basic es la creación de aplicaciones de gestión de empresas, esto es debido a la rapidez con la cual se puede hacer un programa que utilice una base de datos sencilla, además de la abundancia de programadores de este lenguaje.

En realidad podríamos continuar con una gran lista de proyectos en los cuales podría aplicarse Visual Basic, pero esencialmente son proyectos que

---

<sup>1</sup> Internet Explorer (IE): es un navegador web desarrollado por Microsoft para el sistema operativo Microsoft Windows desde 1995

están dedicados a trabajar bajo ambientes Windows o que utilizarán alguna API de Windows. Debido a la compatibilidad con los sistemas Windows, y que la mayoría de la población mundial utiliza sistemas operativos de Microsoft Windows, Visual Basic, es uno de los lenguajes de programación más populares en el presente.

#### **4.4.3. Aspectos técnicos**

Visual Basic constituye un IDE<sup>1</sup> que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un programa llamado depurador el cual corrige errores en el código fuente para que pueda ser bien compilado, un compilador y un constructor de interfaz gráfica o GUI que no es más que una forma de programar en la que no es necesario escribir el código para la parte gráfica del programa, sino que se puede realizar de forma visual.

#### **4.4.4. Ventajas**

- Es un lenguaje RAD (*Rapid Application Development*)
- La curva de aprendizaje es muy rápida
- Integra el diseño e implementación de formularios de Windows
- Permite usar con facilidad la plataforma de los sistemas Windows
- El código en VB es fácilmente migrable a otros lenguajes compatibles

---

<sup>1</sup> IDE: Un entorno de desarrollo integrado, es un programa informático compuesto por un conjunto de herramientas de programación.

#### **4.4.5. Desventajas**

- Sin soporte de Microsoft desde el 4 de abril de 2008, pero el MSDN es completo
- No es multiplataforma
- Por defecto permite la programación sin declaración de variables
- No permite programación a bajo nivel ni incrustar sección de código en ASM
- Sólo permite el uso de funciones de librerías dinámicas (DDL)
- Algunas funciones están indocumentadas
- Es un lenguaje basado en objetos y no orientado a objetos
- No maneja muy bien los punteros a memoria
- No soporta tratamiento de procesos como parte del lenguaje
- No incluye operadores de desplazamiento de bits como parte del lenguaje
- No permite el manejo de memoria dinámica y punteros como parte del lenguaje
- No avisa de ciertos errores o advertencias
- No tiene instrucciones de pre procesamiento
- El tratamiento de mensajes de Windows es básico e indirecto
- La gran gama de controles incorporados son generales
- Los controles personalizados no mejoran la potencia de la API de Windows

#### **4.5. Lenguaje C#**

En el 2000, Microsoft presenta su plataforma .NET junto con este nuevo lenguaje, el cuál servirá de lenguaje principal de la plataforma. C# es un híbrido de C++ y Java, por tanto es una variante de C, que fusiona la capacidad de combinar operadores con la plena orientación a objetos.

La orientación a objetos es tal, que el propio programa está encapsulado en una clase. A pesar de su corta historia, ha recibido la aprobación del estándar de dos organizaciones, en el 2001 se aprueba el ECMA y en el 2003 el ISO.

#### **4.5.1. Proyectos en los que se utiliza C#**

Es utilizado para desarrollar aplicaciones para el entorno .NET de Microsoft, ofrece una alternativa al desarrollo basado en Java. A continuación se enumeran los tipos de aplicaciones que se pueden construir utilizando C#:

- Aplicaciones de consola
- Aplicaciones móviles
- Aplicaciones basadas en Web
- Aplicaciones GUI de Windows
- Aplicaciones de ASP.NET
- Aplicaciones de escritorio
- Servicios Web XML
- Servicios de Windows

#### **4.5.2. Aspectos técnicos**

El código fuente de C#, así como el de otros lenguajes .NET, es compilado a un código intermedio llamado MSIL (*Microsoft Intermediate Language*). C# deriva principalmente de los lenguajes C, C++ y Java con algunas características de Visual Basic de Microsoft.

### **4.5.3. Ventajas**

- Independiente de la máquina donde vaya a ejecutarse, siempre que se ambiente Windows
- Recolección de basura automática
- Eliminación del uso de punteros
- No hay que preocuparse por archivos de cabecera “.h”
- No importa el orden en que hayan sido definidas las clases ni las funciones
- No existen las dependencias circulares
- Concepto formalizado de los métodos get y set, con lo que se consigue código mucho más legible
- Gestión de eventos, usando delegados, mucho más limpia
- Soporta la sobrecarga de operadores, que aunque pueden complicar el desarrollo son opcionales y algunas veces útiles
- .NET ofrece mucha documentación de ayuda incluida en el IDE y de soporte. Esto simplifica el desarrollo y la implementación
- Todos los códigos que se ejecutan en el ambiente .NET son compilados, lo cual proporciona un gran rendimiento a diferencia de versiones interpretadas
- Rápido aprendizaje por parte de los desarrolladores. Es sencillo de aprender por la documentación y el soporte de ayuda

### **4.5.4. Desventajas**

- El mantenimiento de proyectos en múltiples lenguajes es costoso
- .NET no es multiplataforma, sólo está disponible para la familia Windows
- Es un código cerrado, no hay licencias libres. La infraestructura para desarrollar en .NET representa un alto costo para las empresas

- Hay productos que ofrecen mucha más portabilidad que .NET, la cual sólo está preparada para ejecutarse sobre plataformas Microsoft



## 5. PROYECTOS DE DESARROLLO

Cada día los desarrolladores de *software* se encuentran con nuevos proyectos, los cuales representan un reto a superar. Cada proyecto posee características propias y sería imposible enumerarlas o mostrar un ejemplo, para cada uno de los proyectos que se puedan presentar.

Los proyectos de desarrollo de *software* pueden ser realizados en varios lenguajes de programación de manera satisfactoria, pero en ocasiones la elección de un lenguaje en comparación a otro puede facilitar su desarrollo o solventar el problema de una manera eficiente. Es importante que el gerente del proyecto seleccione detenidamente el lenguaje a utilizar teniendo en cuenta las características del proyecto y las capacidades y vulnerabilidades de cada lenguaje.

En el caso de existir varios lenguajes de programación, que puedan dar solución a un proyecto de manera satisfactoria, la elección del lenguaje estará dictado por las capacidades y/o deseos del equipo de trabajo o empresa, entre estos factores es posible enumerar las siguientes:

- Número de integrantes con control de cierto lenguaje de programación
- Acceso a las herramientas de desarrollo
- Afinidad hacia el lenguaje
- Elección del cliente
- Acceso a documentación
- Tiempo de desarrollo
- Presupuesto para el proyecto

Este capítulo muestra un listado de proyectos, cada uno con características particulares, para ejemplificar los tipos de proyectos que se le pueden presentar a un equipo de desarrollo y la elección óptima del lenguaje a utilizar para dicho proyecto.

Nota: las elecciones del lenguaje presentadas para cada proyecto en este capítulo no son definitivas, pueden darse circunstancias distintivas en el negocio, las cuales puedan inclinar la balanza hacia un lenguaje u otro, estas circunstancias deben de ser consideradas detenidamente.

Nota: recordar que este documento se centra en aplicaciones de escritorio, por lo que todos los proyectos presentados en este capítulo están enfocados a un desarrollo de aplicaciones de este tipo y no así de aplicaciones Web. Los lenguajes de programación comparados para dar solución a los problemas presentados son los lenguajes descritos en el capítulo anterior, por lo cual no se compara con otros lenguajes como Ruby, Python, Pearl, ni lenguajes más antiguos como Pascal, D o el Ensamblador.

Para las ejemplificaciones se utilizará la empresa ficticia “Productos Lácteos S.A.”. Esta es una empresa familiar, pero que ha obtenido un gran crecimiento en la última década, dedicada a la cría de ganado vacuno y procesamiento de lácteos. Esta empresa se encuentra actualmente en expansión, por lo que necesita de nuevos sistemas para dar soporte a los procesos de la empresa.

## **5.1. Proyecto 1**

La empresa “Productos Lácteos S.A.” desea desarrollar un sistema que controle los siguientes aspectos del negocio:

- Inventario de herramientas
- Inventario de personal
- Sistema de reportes

La empresa cuenta con cuatro ordenadores con sistema operativo Microsoft Windows XP, con sus respectivas licencias, los cuales serán utilizados por diferentes trabajadores: bodegueros, secretarias y recursos humanos. Posee un servidor con un sistema operativo GNU/Linux, como servidor de base de datos. Los 4 ordenadores se conectarán a este servidor para llevar el control de la empresa y consultar la información. Los empleados que utilizarán el sistema han sido capacitados de antemano en la utilización del sistema operativo Windows.

Debido a que existen otros proyectos dentro de la empresa, no se desarrollan estos módulos con el departamento de tecnología; sino que se desea contratar a un equipo para que solucione estos sistemas. La empresa cuenta con poco presupuesto para la realización de estos dos módulos y el plazo para obtener estos dos sistemas es de 2 meses. Se solicita al gerente de proyectos de TI de la empresa, elegir el lenguaje de programación.

### **5.1.1. Solución**

Este proyecto no muestra características muy específicas, un proyecto de este tipo puede llevarse a cabo con varios lenguajes de programación. Lo

primero que se debe de realizar es enumerar las características del proyecto, para así poder llegar a una conclusión. Las características de este proyecto son las siguientes:

- Capacidad de conexión por red
- Compatible con sistemas operativos Windows
- Utilización de ventanas
- Bajo presupuesto
- Escaso tiempo de desarrollo
- Desarrollo externo

Estas características nos dejan un margen bastante amplio para la elección del lenguaje de programación, pero el lenguaje de programación que se aplicaría en este proyecto sería Visual Basic. Los motivos de la elección de este lenguaje son las siguientes:

- El número de programadores con conocimiento de este lenguaje es mayor en Guatemala, por lo tanto los costos de realizar estos sistemas en este lenguaje, podrían ser menores en comparación a otros lenguajes
- Es compatible solamente con Windows
- Permite un desarrollo rápido
- El rendimiento del sistema no es un punto crítico en este proyecto
- Permite conexión mediante la red y de base de datos
- Existen herramientas para la creación de reportes
- No existe problemas de licenciamiento. El licenciamiento sólo existe para la empresa desarrolladora del *software* no para el comprador de este

Nota: para este proyecto pudieron haberse elegido Java o C#. El punto diferenciador en este proyecto es el acceso a la mano de obra, ya que este puede hacer que el costo disminuya,

## **5.2. Proyecto 2**

La empresa “Productos Lácteos S.A.” necesita un sistema con el cual sea posible el control de la línea de procesos de la empresa. Durante el proceso de los lácteos se tomarán mediciones de varios tipos, los cuales deben de ser registrados de manera diaria.

Los puntos de control están alojados a lo largo de toda la línea del proceso. Este sistema es importante para la empresa, por lo cual no se puede aceptar ningún tipo de fallos. Existen cuatro ordenadores destinados a la toma de estos datos, estos ordenadores poseen un sistema operativo Microsoft Windows 98, cada uno con su respectiva licencia, y se encuentran conectados al servidor de base de datos, GNU/Linux, mencionado en el ejemplo anterior.

Debido a que existen otros proyectos más críticos dentro de la empresa no se desarrollará este sistema con el departamento de tecnología, sino que se desea contratar a un equipo para que den solución a estos sistemas. La empresa cuenta con el presupuesto necesario para la realización de este sistema, el tiempo para la obtención de este sistema no es una limitante. Se solicita al gerente de proyectos de TI de la empresa, elegir el lenguaje de programación.

### 5.2.1. Solución

Este proyecto muestra ciertas características específicas aún este proyecto puede llevarse a cabo con varios lenguajes de programación. Las características de este proyecto son las siguientes:

- Capacidad de conexión por red
- Compatible con sistemas operativos Windows
- Utilización de ventanas
- Presupuesto aceptable
- Tiempo de desarrollo aceptable
- Desarrollo externo
- Sistema de alta importancia para el negocio, sin tolerancia a fallos

El lenguaje de programación que se aplicaría en este proyecto sería C#, Los motivos de la elección de este lenguaje son las siguientes:

- El número de programadores con conocimiento de este lenguaje es bastante amplio en Guatemala, por lo tanto los costos de realizar estos sistemas en este lenguaje podrían ser menores en comparación a otros lenguajes
- Es compatible solamente con Windows
- Permite un desarrollo rápido
- Provee de buen rendimiento en sus aplicaciones
- Permite conexión mediante la red y de base de datos
- No existe problemas de licenciamiento. El licenciamiento solo existe para el desarrollador del *software* no para el comprador de este

Nota: para este proyecto pudieron haberse elegido Java y Visual Basic. En comparación C# contra Java, el punto diferenciador en este proyecto es el acceso a la mano de obra, ya que este puede hacer que el costo disminuya y al rendimiento necesario para este sistema. En comparación C# contra Visual Basic, el punto diferenciador es que el lenguaje C# es más estable, completo y provee de mayor eficiencia en sus aplicaciones.

### **5.3. Proyecto 3**

Actualmente existen tres procesos que corren en tres ordenadores distintos, separados físicamente. Estos procesos corren en ordenadores GNU/Linux. Los tres procesos tienen como objetivo el control de la maquinaria que aplica los tratamientos a los lácteos. La ejecución de los procesos es de manera lineal, el proceso 1 se ejecuta primero, luego el proceso 2 y finalmente el proceso 3.

Estos procesos se encuentran controlados por el personal, pero esto ha ocasionado muchos problemas debido al factor humano. Se desea que este proceso sea automatizado y adicionalmente se desea que estos procesos se ejecuten en el mismo ordenador. El problema que existe es que cada proceso consume todos los recursos del ordenador cuando se encuentra ejecutando y no se cuenta con capital para comprar un servidor de mayor capacidad.

El gerente del área de TI le indica a Ud. que para que el sistema funcione adecuadamente se deben de instalar los tres sistemas en un mismo ordenador, para controlar cual proceso se ejecuta se utilizarán semáforos del sistema y para la comunicación entre un proceso y otro se utilizarán los mensajes del sistema y así eliminar el factor humano.

El desarrollo de este sistema se realizará en el área de TI de la empresa. El plazo para la realización de este sistema es de seis meses. Se le solicita al gerente de proyectos de TI de la empresa, que seleccione el lenguaje de programación a utilizar en este proyecto.

### **5.3.1. Solución**

Este proyecto muestra características específicas, pocos lenguajes de programación nos permitirían realizar lo solicitado. Las características que presenta este proyecto son las siguientes:

- Ejecutarse sobre un sistema operativo GNU/Linux
- Utiliza cola de mensajes del sistema operativo
- Utiliza semáforos del sistema
- Desarrollo interno
- Tiempo de desarrollo amplio
- Sistema crítico para el negocio

El lenguaje de programación que se aplicaría en este proyecto sería C/C++, Los motivos de la elección de este lenguaje son las siguientes:

- Permite un control total del sistema operativo
- Facilidad de desarrollar en comparación con otros lenguajes como el ensamblador
- Bastante acceso a documentación

#### **5.4. Proyecto 4**

La empresa “Productos Lácteos S.A.” necesita un sistema con el cual se procese una cantidad “n” de datos provenientes de varios archivos de texto, los cuales contienen información referente a los puntos de control de los procesos realizados a la leche extraída del ganado y sus tratamientos. Los archivos que utilizará este sistema tienen un peso promedio de 100 Mb.

Este sistema pretende encontrar tendencias en el aprovechamiento de la leche, el aprovechamiento de los productos utilizados en el tratamiento, rendimiento del ganado, entre otros. Este proceso es un punto crítico para el negocio. El sistema deberá generar un archivo de texto con los datos obtenidos del proceso. El sistema no tendrá interfaz gráfica, simplemente se ejecutará el sistema, proveyéndole de entrada los archivos y especificándole la ruta de los archivos de salida.

El gerente de TI identifica que algunos de los componentes utilizados en el sistema expuesto en el punto anterior, pueden ser utilizados en este nuevo proyecto, por lo que le indica que se debe reutilizar el código para agilizar el tiempo de desarrollo.

Para mejorar el rendimiento de este sistema, se utilizará un ordenador dedicado, el cual tendrá un sistema operativo GNU/Linux. El desarrollo de este sistema estará a cargo del equipo de TI de la empresa.

### **5.4.1. Solución**

Sistema importante para la empresa y demanda un buen rendimiento. Las características que presenta este proyecto son las siguientes:

- Ejecutarse sobre un sistema operativo GNU/Linux
- Proyecto crítico para el negocio
- Necesidad de buen rendimiento del sistema
- Proveer compatibilidad con el lenguaje C, debido a que en el proyecto anterior el lenguaje de programación aplicado fue este
- Plazo amplio de desarrollo

El lenguaje de programación que se aplicaría en este proyecto sería C/C++, Los motivos de la elección de este lenguaje son las siguientes:

- Provee de un buen rendimiento de las aplicaciones
- En el caso de C++, es una extensión de C, por lo cual el código del proyecto anterior puede ser reutilizado
- Bastante acceso a documentación
- El factor tiempo no es crítico

### **5.5. Proyecto 5**

La junta directiva solicita un sistema con el cual se pueda obtener la información referente a los procesos realizados en un día específico, mes, año, etc., así como el estado de las tareas realizadas por cada empleado. Este sistema sólo es de carácter informativo, se conectará al servidor de base de datos y obtendrá los datos solicitados por la aplicación.

La junta directiva se encuentra formada por tres personas, las cuales tienen gustos distintos, uno de ellos es aficionado al diseño gráfico por lo cual cuenta con una computadora personal con sistema operativo Mac Snow Leopard. Otro integrante de la junta directiva es aficionado al *Open Source*, este posee un computador de escritorio con un sistema operativo Slackware. El último integrante no tiene intereses especiales respecto al sistema operativo, es por esto que tiene un sistema operativo Windows 7.

El sistema solicitado será de uso único y exclusivamente para estos tres individuos. El desarrollo de este sistema de consulta será realizado por el departamento de TI de la empresa. En un plazo no mayor a ocho meses.

#### **5.5.1. Solución**

Este sistema nos muestra una característica particular, la cual no puede ser solventada por los lenguajes de programación presentados en este documento. El único lenguaje de programación de dar solución a este sistema es Java. Esto es debido a la necesidad de que sea una aplicación multiplataforma.

#### **5.6. Lenguajes de programación y sus características**

En este apartado se muestran un listado de proyectos de manera general y se especifica los lenguajes de programación que pueden dar solución a cada proyecto.

El objetivo de este apartado es mostrar las características más importantes para las cuales un lenguaje de programación es apto y así proveer

al desarrollador de una forma fácil de elección del lenguaje de programación mediante comparación.

Características del proyecto y lenguajes aplicables:

- Poco tiempo de desarrollo
- Sistema no crítico
- La eficiencia no es una determinante
- Necesidad de desarrolladores
- Sistema operativo Windows
- Lenguajes aplicables: Visual Basic, C#, Java

Características del proyecto y lenguajes aplicables:

- Poco tiempo de desarrollo
- Sistema importante para el negocio
- Sistema Operativo Windows
- Toolkit que permitan al desarrollador agilizar su trabajo
- Lenguajes aplicables: Java y C#

Características del proyecto y lenguajes aplicables:

- Amplio tiempo de desarrollo
- Control excesivo de los recursos del ordenador
- Lenguajes aplicables: C

Características del proyecto y lenguajes aplicables:

- Sistema sin GUI
- Sistema Operativo GNU/Linux
- Tiempo de desarrollo medio
- Lenguajes aplicables: Java y C++

Características del proyecto y lenguajes aplicables:

- Reutilización de código de C
- Control de recursos del ordenador
- Lenguajes aplicables: C++
- Programación orientada a objetos
- Utilización de patrones de diseño
- Sistema Operativo GNU/Linux
- Lenguajes aplicables: Java

Características del proyecto y lenguajes aplicables:

- Multiplataforma
- Lenguajes aplicables: Java

Éstas son algunas de las características que puede tener un proyecto para que un lenguaje de programación sea apto para darle solución. Para poder comparar detenidamente un lenguaje de programación es necesario ver sus ventajas y desventajas, las cuales se encuentran en el punto 3 de este documento.



## CONCLUSIONES

1. Existen varios lenguajes de programación, más de los que se presentan en este documento. Algunos ya son obsoletos, otros inician sus pasos; hay algo que tienen en común, es que cada uno tiene sus fortalezas y debilidades.
2. Cuando se inicia un proyecto es necesario contemplar muchos aspectos: el análisis, el diseño, la arquitectura, pero existe un aspecto importante como lo es el lenguaje de programación. Este aspecto es por lo general olvidado por los desarrolladores, los cuales dan por sentado que el cambio de un lenguaje a otro no tiene repercusión en el proyecto.
3. Dependiendo del lenguaje de programación, el desarrollo de un proyecto se facilita o complica. Debe tomarse en cuenta las características y comparar los lenguajes; para poder decidir qué lenguaje se debe utilizar y no tomar esta decisión a la ligera.
4. Java es el lenguaje más utilizado a nivel mundial así como en Guatemala. Este lenguaje presenta beneficios como: recolector de basura, independencia de plataforma, entre otros. Este lenguaje de programación es apto para proyectos de gran tamaño, el aprendizaje de este lenguaje no es el más rápido pero es el menos complicado.
5. Otro lenguaje de programación utilizado a nivel mundial es el lenguaje C, éste es utilizado en proyectos en los que se requiera gran control sobre el sistema operativo o el ordenador. Principalmente se utiliza en porciones

del desarrollo no en la elaboración completa de un proyecto. Es un lenguaje de programación un tanto complejo de aprender, por lo cual el desarrollo se limita a ciertos proyectos, en los cuales sea necesario. En Guatemala no es muy utilizado este lenguaje.

6. Visual Basic, es un lenguaje sencillo de aprender, esto permite que sea un lenguaje muy utilizado en Guatemala, pero no es recomendable en proyectos de gran envergadura. Principalmente es utilizado en sistemas bancarios y en aplicaciones de escritorio.

## RECOMENDACIONES

1. La elección del lenguaje de programación, cuando se inicia un proyecto es un aspecto importante. Es importante que durante la fase de análisis y diseño del proyecto, se establezcan todas las características que debe de llenar un lenguaje de programación, para que, con base en estas características se puedan identificar los lenguajes que mejor se acoplen al proyecto a realizar.
2. No conviene que se elija un lenguaje de programación a la ligera, tampoco se debe de elegir el lenguaje que el cliente sugiere; sin antes verificar las destrezas del mismo contra las características del proyecto. No se recomienda utilizar el lenguaje más conocido, sin antes verificar que éste sea el óptimo.
3. En ocasiones no se contempla la elección de un lenguaje, debido a que se tiene el temor de iniciar un proyecto con un lenguaje nuevo. Un nuevo lenguaje, representa aprendizaje y tiempo invertido para conocer el mismo. El aprender un nuevo lenguaje no es complicado, siempre que se tengan conocimientos de otros, en general el primer lenguaje aprendido resulta complicado. Aprender uno nuevo es bueno y permite actualizarse. Hay que vencer el temor al cambio y aceptar a nuevos retos.
4. Las instituciones académicas deben de contemplar incluir la enseñanza, en los cursos de programación que impartan a los estudiantes, qué ventajas y debilidades presentan cada lenguaje de programación, para que puedan tomar decisiones al iniciar un proyecto.



## BIBLIOGRAFÍA

1. BARCELÓ, Miquel. *Una historia de la informática*. Marcín Wichary (ed. lit.). 2a ed. Barcelona: UOC, 2008. 161 p. ISBN: 978-84-9788-709-0.
2. CARRALES MORA, Maricruz. *Lenguaje Logo III, Explorando la programación*. San José: EUNED, 1994. 224 p. ISBN: 9977-64-762-3.
3. CERESO LÓPEZ, Yolanda. *Iniciación a la programación en C#, un enfoque práctico*. Fernando M. Garcia Tomé (ed. lit.). Madrid: Delta Publicaciones, 2007. 301 p. ISBN: 84-96477-53-3.
4. DEL JUNCO LORENTE, Carlos. *Manual de Programación Lenguaje C++*. Madrid: MAD, S.L., 2005. 133 p. ISBN: 54-665-4456-9.
5. GOSLING, James. *The Java Language Specification*. California: Sun Microsystems, 2000. 505 p. ISBN: 0-201-31008-2.
6. KERNIGHAN, Brian W. y Ritchie, Dennis M. *El lenguaje de programación C*. 2a ed. Naucalpan de Juárez: Prentice Hall, 1991. 287 p. ISBN: 0-13-110362-8.
7. LOUDEN, Kennet C. *Lenguajes de programación: principios y práctica*. 2a ed. California: Thomson, 2003. 625 p. ISBN: 970-686-284-6.

8. PERRY, Greg. *Aprendiendo Visual Basic 6 en 21 días*. A. David Garza Marín (ed. lit.); Rebeca Alicia Sánchez López (trad.). México DF: Prentice Hall, 2003. 851 p. ISBN: 0-13-120532-5.
  
9. RODRÍGUEZ SALA, Jesús Javier. *Introducción a la programación: teoría y práctica*. Oscar Martínez Bonastre (ed. lit.). 2a. ed. Madrid: ECU, 2006. 211 p. ISBN: 84-8454-274-2.